

# Nivel de conocimiento en programación y el desarrollo de sistemas de información automatizados

Sandy Romero, Universidad de La Guajira, Colombia  
Jaider Quintero, Universidad de La Guajira, Colombia

**Resumen:** El objetivo de estudio fue analizar el nivel de conocimiento en programación y su relación con el desarrollo de Sistemas de Información (SI) en los estudiantes de Ingeniería de Sistemas de la Universidad de la Guajira. Se utilizó una metodología descriptiva, correlacional, un diseño no experimental, transaccional de campo; la población fue 320 estudiantes, aplicándose un muestreo probabilístico, estratificado y proporcional, quedando la muestra conformada por 120 estudiantes, para la recolección de los datos se aplicó un instrumento constituido por 45 ítems tipo Liker. Para el tratamiento de los datos se utilizó la estadística descriptiva. Los resultados obtenidos muestran que el nivel de conocimiento de los estudiantes de ingeniería es medio, porque no todos tienen una buena actitud frente a la programación y el desarrollo de software, en la mayoría sus conocimientos teóricos son buenos; pero en la práctica y experiencia en programación el resultado es bajo, pues practican poco y tienen poca experiencia. En cuanto a la relación entre el nivel de conocimiento en programación y el desarrollo de software es directamente proporcional porque mientras mayor sea el nivel de conocimiento de los estudiantes serán más competentes para desarrollar SI y viceversa. En consecuencia se recomienda a todos los programas ingeniería de sistemas y carreras afines, que sus estudiantes adquieran los conocimientos y competencias necesarias en la programación, para desarrollar SI de calidad, pues estos están jugando un papel cada vez más innovador en la Sociedad, posibilitando soluciones a las empresas e introduciendo cambios significativos en cada una de las áreas del saber.

**Palabras clave:** software, programación, sistema, información, conocimiento

**Abstract:** The aim of study was to analyze the level of knowledge in programming and its relationship to the development of information systems (IS) students in Systems Engineering from the University of La Guajira. We used a descriptive methodology, correlational, non-experimental design, transactional field, the population was 320 students, applying probability sampling, stratified proportional sample being made up of 120 students, for the data collection instrument was applied consists of 45 Likert-type items. For the treatment of the data was used descriptive statistics. The results show that the level of knowledge of engineering students is medium, because not everyone has a good attitude towards programming and software development, most are good theoretical knowledge, but in practice and experience scheduling the result is low, as practiced little and have little experience. Regarding the relationship between the level of knowledge in programming and software development is directly proportional because the higher the level of knowledge of the students will be more competent to develop SI and vice versa. Consequently it is recommended that all systems engineering programs and related careers that students acquire the knowledge and skills required in programming, to develop SI quality, as these are playing an ever more innovative in the Company, enabling solutions companies and introducing significant changes in each of the areas of knowledge.

**Keywords:** Software, Programming, System, Information, Knowledge

## Introducción

En esta nueva sociedad del conocimiento y de la información ya no son válidos ciertos modelos y conceptos tradicionales, pues el mundo se rige actualmente por un nuevo orden, los cambios introducidos por la tecnología en nuestra forma de vida hacen necesario replantear los modelos tradicionales en todos los campos siendo uno de ellos la informática y los sistemas, fundamentos de este proceso de transición Castells (2006).

Todo esto ha hecho que el manejo de la información es cada vez más dependiente de la tecnología, ya que los crecientes volúmenes de la misma que se manejan y su carácter claramente multimedia obligan a un tratamiento con medios cada vez más sofisticados. El acceso a redes como Internet mediante ordenadores personales o la complejidad de los sistemas bancarios y de reservas aéreas totalmente informatizadas son pruebas evidentes de que sin la tecnología el uso de la



información sería imposible en la actualidad. Es por esta razón que el software y los Sistemas de Información son conceptos importantes cuando se habla de las Tecnologías. Estos están presentes en todos los procesos de información, ya que dichas funciones son realizadas cada vez con mayor intensidad por ordenadores.

El software está jugando un papel cada vez más innovador en la Sociedad del Conocimiento, posibilitando soluciones a las empresas e introduciendo cambios significativos en los comportamientos de los usuarios finales, tanto en casa, como en el trabajo y la implementación de sistemas de información en las organizaciones, brindan la posibilidad de obtener información confiable, oportuna, eficaz y grandes ventajas competitivas.

Si analizamos el uso de la tecnología como un fenómeno determinante en el desarrollo de hombres y mujeres del presente y futuro, podemos inferir que representan uno de los paradigmas sociales más importantes de los últimos treinta años; pero no se puede olvidar que no solamente debemos enfocarnos a utilizarlas, sino también a crearlas y es aquí donde juega un papel importante la programación y el profesional de sistemas, con los conocimientos necesarios, para crear, facilitar, gestionar y dar soluciones tecnológicas a cada una de las áreas del saber y en consecuencia los programas de ingeniería de sistemas y afines, que son los llamados directos a liderar estos procesos de cambio.

Por lo tanto, partiendo de estas afirmaciones, esta investigación se orienta a fundamentar y sustentar Nivel de conocimiento en programación y su relación con el desarrollo de sistemas de información automatizados en los estudiantes de ingeniería de Sistemas de la Universidad de la Guajira.

## Fundamentos teóricos

### *Actitud y conocimiento de los estudiantes frente a la programación*

Son muchos los autores que han conceptualizado la actitud, ya que la definición de éste data del año 1935, pero es Allport (1969: 77) quien agrupa todas estas definiciones para explicarlas como “un estado neural y mental de la disposición a responder, organizado a través de la experiencia y que ejerce una influencia directiva y/o dinámica de la conducta”.

Al analizar esta definición, se observa un conjunto de términos y expresiones que resultan importantes para la presente investigación el primero de ellos radica en el uso de las palabras, estado neural y mental”, de donde se puede inferir que, la actitud ha sido orientada a las mediciones fisiológicas (neural). Esto significa que se pretende medir directamente la presencia de reacciones emocionales ante la presentación de un objeto determinado.

Así mismo, la palabra organizada, es entendida por el autor como un término que conduce directamente a identificar una forma de organización entre un conjunto de componentes como son el cognitivo, el cual incluye la información de que se dispone acerca del objeto, necesaria para efectuar la evaluación correspondiente, el componente afectivo, responsable de las evaluaciones de las emociones, considerando la inclinación a evaluar en términos positivos y negativos que se tiene frente un determinado objeto, y el componente conductual, referido a la conducta o los comportamientos que manifiesta el individuo frente al objeto, partiendo de una información previa del objeto y luego de haber realizado la evaluación emocional del mismo.

Sin embargo, es importante destacar, que los tres componentes mencionados anteriormente coinciden en un punto, en que todos ellos son evaluaciones del objeto de la actitud. De hecho, las percepciones o la información pueden ser favorables o desfavorables, por su parte, los sentimientos pueden ser positivos o negativos y finalmente la conducta o intenciones de conductas de aceptación o de rechazo. En relación a la expresión, a través de la experiencia, muestra que las actitudes, son consideradas, aprendidas, aunque su debilidad radica en determinar el mecanismo mediante el cual se adquieren y los estudios realizados demuestran que la manera como se aprenden las actitudes es a través de las experiencias.

Por otra parte, Eagly & Chaiken, (1993: 1), afirman que “la actitud es una tendencia psicológica que se expresa mediante la evaluación de una identidad concreta con cierto grado de favorabilidad y desfavorabilidad”.

Los tres tipos principales de la actitud han sido identificados por Rebolledo (1999: 195) como, “el componente cognitivo, el componente afectivo y el componente conductual” y las definiciones encontradas en cualquier texto de Psicología Social, plantean a las actitudes como la disposición para evaluar objetos favorables o desfavorablemente, al igual que la tendencia a evaluar un objeto en términos positivos o negativos y por último la postura de evaluar dicho objeto mediante la aceptación o el rechazo del mismo. Es por ello, cada componente que conforma la actitud, posee aspectos que lo integran, y a continuación se explican cada uno de ellos:

**Actitud Cognitiva:** Este tipo se refiere a las creencias que tienen los individuos, consideradas éstas por Salazar y otros (1992: 153) como, “el juicio probabilística que conecta a un objeto o concepto con algún atributo”, asignando así características particulares a cualquier objeto, tendiendo un firme asentamiento y conformidad para aceptarlo o rechazarlo.

**Actitud afectiva:** mediante el cual, el individuo evalúa una respuesta emocional, entendiéndose por tal, la emoción que conduce a la persona a acercarse a cualquier cosa evaluada positivamente y de la misma forma alejarse de las evaluadas de manera negativa. Pero para que el estímulo provoque una respuesta emocional en el individuo, éste primero debe evaluar el significado de dicho estímulo, por lo tanto dicha evaluación se hace en base a los planteamientos cognitivos de la emoción.

**Actitud Conductual:** hace referencia a conductas o comportamientos que el individuo tiene ante una determinada situación, respuesta que resulta producto de un estímulo y que varía de acuerdo al aprendizaje que posee, porque es el aprendizaje lo que cambia la conducta y causa el desarrollo.

Es necesario mencionar, que los tres tipos de actitud antes señalados, ejercen mutua influencia hacia un estado de armonía y cualquier cambio que se registre en uno de estos tres componentes, modificará a los otros dos, puesto que todo el sistema es accionado cuando uno de sus integrantes es alterado, puede crear un estado de incongruencia entre los componentes cognitivo, afectivo y conductual, resultando en un cambio de actitud.

Para la presente investigación, la actitud se toma como una disposición de aceptación o rechazo hacia los lenguajes de programación, considerando los tres componentes, el cognitivo, el afectivo y el conductual, como elementos preponderantes para el manejo de los mismos en el programa de ingeniería de sistemas de la Universidad de la Guajira.

La actitud cognitiva la vamos a analizar como las creencias que tienen los estudiantes sobre el significado, la conveniencia y la importancia de aprender o no los lenguajes de programación. La actitud afectiva la vamos a revisar teniendo en cuenta el Interés, disposición, agrado y sentimientos hacia la programación y por último la actitud conductual se va analizar teniendo en cuenta su iniciativa y su liderazgo para aprender, gestionar y utilizar lenguajes de programación en la resolución de problemas.

Como se mencionó anteriormente en la definición de actitud conductual, la actitud está relacionada y puedes ser modificada por el aprendizaje que posee la persona, porque es el aprendizaje y el conocimiento sobre el objeto lo que cambia la conducta y causa el desarrollo.

El conocimiento es “el conjunto de saberes de un individuo que le permiten lograr un buen desempeño o tarea, e indican su suficiencia o idoneidad para el citado desempeño” (Bueno, 1996). La habilidad es “la destreza, talento, experiencia o gracia para ejecutar una cosa o capacidad y disposición para negociar y conseguir unos objetivos con personas, tanto en grupos como individualmente” (Bueno, 1996).

Por esta razón también es importante estudiar en esta investigación los conocimientos teóricos, prácticos y experiencia en programación que tienen los estudiantes del programa de ingeniería de sistemas. Estos conocimientos implican el aprendizaje y el desarrollo de habilidades sobre los siguientes aspectos: Arquitectura Interna y externa del Computador, Historia y Evolución del hardware y el software, Algoritmia y Programación estructurada y Orientada a Objetos, Estructuras de datos, Base de datos, Sistemas de Información, Compiladores, Sistemas Operativos, Simulación, Métodos Numéricos, Diseño y Desarrollo de Software, Sistemas de Información y el desarrollo de software como parte importante del ciclo de vida de los SI, Lenguajes de Programación estructurados y Orientados a Objetos.

La programación es una actividad que implica un proceso mental, generalmente complejo y creativo, exige del programador: inteligencia, conocimiento, habilidades y disciplina. La inteligencia es un recurso natural que en la mujer y el hombre normales es suficiente para adquirir los otros tres atributos necesarios para programar y poder resolver problemas utilizando una computadora. La adquisición del conocimiento se logra estudiando los conceptos, fundamentos y las técnicas básicas de programación.

En cuanto a la adquisición de las habilidades, la voluntad del estudiante es fundamental, ya que sólo con la práctica y la experimentación personal podrá lograrse. La disciplina, como observancia de las normas y el buen proceder, la podrá adquirir el estudiante con la guía valiosa y motivación de los profesores del área y la dedicación que él disponga para esto.

### ***La programación y el desarrollo de software***

La programación es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación. El propósito de la programación es crear programas que exhiban un comportamiento deseado (Joyanes, 2006). El proceso de escribir código requiere frecuentemente conocimientos en varias áreas distintas, además del dominio del lenguaje a utilizar, algoritmos especializados y lógica formal. Un algoritmo es una secuencia no ambigua, finita y ordenada de instrucciones que han de seguirse para resolver un problema. Un programa normalmente implementa (traduce a un lenguaje de programación concreto) uno o más algoritmos. Un algoritmo puede expresarse de distintas maneras: en forma gráfica, como un diagrama de flujo, en forma de código como en pseudocódigo o un lenguaje de programación, en forma explicativa, etc.

En general, al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación (Joyanes, 1998).

Aunque el propósito principal debe ser que el estudiante aprenda a programar, es conveniente que pueda codificar los algoritmos en algún lenguaje de programación e implementarlo utilizando la herramienta de desarrollo disponible y que se considere adecuada.

Tal como cita Guillermo Levine en su libro, sobre programar y codificar “la diferencia entre ambos conceptos es fundamental, y no está entendida del todo en el medio profesional de la computación ni por completo, por desgracia, en el medio académico”. Más de una vez hemos escuchado decir “ya aprendí a programar en Pascal” o bien “se requiere experiencia en programación en C”, lo correcto sería “ya aprendí a codificar en Pascal” y “se requiere experiencia en el manejo del lenguaje C”.

Hay un consenso creciente entre los profesores de informática en que los estudiantes deben aprender cómo se resuelven los problemas lógicamente (desarrollo de algoritmos) antes de preocuparse por la puntuación y otros detalles de un lenguaje de programación.

En el ciclo de la programación, la codificación solamente representa una de las etapas posteriores al diseño de la lógica y permite escribir en los términos de un lenguaje de programación la serie de pasos que realizará la computadora al seguir paso a paso el programa. El estudiante debe adquirir el conocimiento sobre las técnicas de programación, a la definición adecuada de los datos y al desarrollo de la habilidad necesaria para el diseño y construcción de los algoritmos con un estilo disciplinado que les dé orden y claridad.

### ***Los lenguajes de programación***

Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana (Mark, 2010).

Los lenguajes de programación sirven para escribir programas que permitan la comunicación usuario – máquina. Estos son como un lenguaje cualquiera, pero simplificado y con ciertas normas, para poder transmitir las instrucciones a la computadora. (Joyanes, 2008).

Los lenguajes de programación son la herramienta informática que nos permite establecer una correcta comunicación entre la computadora y el usuario de la misma. Sin lenguaje de programación no sería posible expresar las instrucciones que queremos que la computadora ejecute, ni tampoco podría mostrarnos los resultados de dichas instrucciones. Su función es proporcionar instrucciones al sistema de la computadora para que pueda realizar una actividad de procesamiento.

Cada lenguaje de programación define y utiliza un grupo de símbolos o reglas que describen como deben escribirse las sentencias y comandos de programación. Estas reglas son de tipo semántico y sintáctico, la primera es el significado que tiene cada palabra y sentencia y la segunda la forma y el orden como se deben escribir (Gómez, 2009).

### *Antecedentes y evolución de los lenguajes de programación*

Los primeros lenguajes de programación surgieron de la idea de Charles Babagge quien desarrollo una excelente idea mediados del siglo XIX. Consistía en lo que él denominaba la máquina analítica, pero que por motivos técnicos no pudo construirse hasta mediados del siglo XX. Con él colaboró en el desarrollo de esta idea, Ada Lovedby, la cual es considerada como la primera programadora de la historia, ella realizó programas para aquella supuesta máquina de Babagge, en tarjetas perforadas. Como la máquina nunca a construirse, los programas de Ada, lógicamente, tampoco llegaron a ejecutarse.

Cuando surgió el primer ordenador, el famoso ENIAC (Electronic Numerical Integrator And Calculator), su programación se basaba en componentes físicos, o sea, que se programaba, cambiando directamente el Hardware de la máquina, exactamente lo que se hacía era cambiar cables de sitio para conseguir así la programación de la máquina. La entrada y salida de datos se realizaba mediante tarjetas perforadas.

Con el advenimiento de los computadores aparecen las secuencias de posiciones de llaves eléctricas que debían conectarse para obtener una acción determinada. Los lenguajes de computadoras toman diferentes formas; los de las primeras computadoras, como la ENIAC y la EDSAC, se componían en el lenguaje real de las máquinas mismas. Para que la computadora entienda nuestras instrucciones debe usarse un lenguaje específico conocido como código máquina, el cual la máquina comprende fácilmente, pero que lo hace excesivamente complicado para las personas. De hecho sólo consiste en cadenas extensas de números 0 y 1.

La necesidad de recordar secuencias de programación para las acciones usuales llevó a denominarlas con nombres fáciles de memorizar y asociar: ADD (sumar), SUB (restar), MUL (multiplicar), CALL (ejecutar subrutina), etc. A esta secuencia de posiciones se le denominó "instrucciones", y a este conjunto de instrucciones se le llamó lenguaje ensamblador. Este fue el primer tipo de lenguaje conocido.

A mediados de los años 60's aparecieron los primeros lenguajes de propósito general, que son en los que se basa este trabajo, como FORTRAN, FORTRAN IV, ALGOL, COBOL, BASIC, PL/I, ADA, C, C++, PASCAL, etc. pero el desarrollo de nuevas tecnologías, tanto en arquitectura de computadoras como en lenguajes de programación, continúa a paso acelerado, cada vez con mayor velocidad, el panorama está cambiando de una etapa de sistemas y lenguajes especialmente desarrollados para aplicaciones individuales. Los lenguajes de programación actuales son los conocidos como Lenguajes visuales, como por ejemplo Visual Fox, Visual Basic, Visual C. Wilson, Leslie B. (1993).

Cada una de estas etapas de la historia de los computadores y los lenguajes de programación que desde sus inicios ha mostrado que su desarrollo y evolución ha ido de la mano, Según Gómez (2009) las podemos resumir en lo que se conoce como las generaciones de los lenguajes de programación:

**Primera generación:** Lo constituyen los lenguajes máquina. Estos se consideran como de bajo nivel porque no existe un programa de codificación menos complicado que el que utiliza los símbolos binarios 1 y 0.

- Ascii, utiliza ceros y unos para representar letras del alfabeto.

Como este es el lenguaje del CPU, los archivos de texto traducidos a los grupos binarios ASCII pueden leerse por casi cualquier plataforma de sistemas de computadoras.

**Segunda generación:** se crearon los primeros lenguajes ensambladores.

Los lenguajes ensambladores usan códigos cortos y precisos, como a para agregar o mvc para mover, y así sucesivamente. Los programas de software de sistemas tales como los sistemas operativos y los programas de utilidad se escriben con frecuencia en un lenguaje ensamblador.

**Tercera generación:** Son relativamente independientes del hardware de la computadora. Esto significa que el mismo programa puede utilizarse en varias computadoras diferentes de distintos fabricantes, son los primeros lenguajes de alto nivel. (C, Pascal, Cobol, etc.) Estos son más fáciles de aprender y usar que los lenguajes máquina y el lenguaje ensamblador, pues su similitud con la comunicación y comprensión humana cotidiana es mayor. Aunque son más fáciles de programar, no son tan eficientes en términos de rapidez operacional y memoria.

**Cuarta generación:** Son los lenguajes capaces de generar código por si solos, son los llamados RAD, con lo cuales se pueden realizar aplicaciones sin ser un experto en el lenguaje. Son lenguajes que se relacionan menos con procedimientos y que son aún más parecidos al inglés que los lenguajes de tercera generación. Algunas características incluyen capacidades de consulta y base de datos, de creación de códigos y capacidades gráficas.

Aquí también se encuentran los lenguajes orientados a objetos, que permiten la interacción de objetos de programación incluyendo elementos de datos y las acciones que se realizan en ellos.

En la programación orientada al objeto, los datos, instrucciones y otros procedimientos de programación se agrupan en un elemento denominado objeto, y estos objetos presentan las siguientes características:

- Encapsulación. Receso de reagrupar elementos dentro de un objeto.
- Polimorfismo. Receso que le permite al programador desarrollar una rutina o grupo de actividades que operaran sobre objetos múltiples.
- Herencia. Propiedad utilizada para describir objetos en un grupo de este tomando características de otros en el mismo grupo o clase de objetos.
- Código reutilizable. Código de instrucciones dentro de un objeto que se puede usar repetidamente en diferentes programas de diversas aplicaciones.

En este grupo también están los lenguajes de programación visual o gráficos, que son aquellos que usan el ratón, iconos o símbolos en la pantalla y menús despegables para desarrollar programas.

Tales como: Visual C++, Visual Basic, Power Builder, Delphi, Forte y muchos otros.

**Quinta generación:** Alrededor de la mitad 1998 surgieron grupos de herramientas de lenguajes de quinta generación, los cuales combinan la creación de códigos basadas en reglas, la administración de reutilización y otros avances. Son los lenguajes orientados a la inteligencia artificial.

Programación basada en conocimiento. Método para el desarrollo de programas de computación en el que se le ordena a la computadora realizar un propósito en vez de instruirla para hacerlo. Estos lenguajes todavía están poco desarrollados. Ej. LISP.

### ***Los sistemas de información***

Un sistema de información (SI) es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad u objetivo.

Dichos elementos formarán parte de alguna de las siguientes categorías: personas, datos, actividades o técnicas de trabajo, recursos materiales en general. Generalmente recursos informáticos y de comunicación, aunque no necesariamente.

“Sistema de información es un conjunto de componentes interrelacionados que reúne, almacena, y distribuye información para apoyar a la toma de decisiones y el control de una organización” (Laudon y Laudon, 2004).

Para Guzmán (2002), los sistemas de información de una organización están conformados por el ser humano y cosas materiales, encargados del trato de información de un negocio mediante el acceso de datos, su desarrollo, acumulamiento y posterior salida.

Tabla 1: Clasificación de los lenguajes de programación

<i>Clasificación</i>	<i>Tipo</i>
<i>Según su nivel de abstracción</i>	<p><i>Lenguajes de Bajo Nivel.</i> Son aquellos que se acercan al funcionamiento de una computadora:</p> <ul style="list-style-type: none"> <li>• <i>Lenguajes máquina:</i> ordena a la máquina las operaciones fundamentales para su funcionamiento</li> <li>• <i>Lenguajes ensambladores:</i> Con la aparición de este lenguaje se crearon los programas traductores para poder pasar los programas escritos en lenguaje ensamblador a lenguaje máquina.</li> </ul>
	<p><i>Lenguajes de Medio Nivel.</i> Tienen características que los acercan a los lenguajes de bajo nivel pero, al mismo tiempo, ciertas cualidades que lo hacen un lenguaje más cercano al humano y, por tanto, de alto nivel.</p>
	<p><i>Lenguajes de Alto Nivel.</i> Se tratan de lenguajes independientes de la arquitectura del ordenador. C, Pascal, Cobol, Fortran.</p>
<i>Según la forma de ejecución</i>	<p><i>Lenguajes compilados:</i> Los compiladores son aquellos cuya función es traducir un programa escrito en un determinado lenguaje a un idioma que la computadora entienda.</p>
	<p><i>Lenguajes interpretados:</i> Se puede también utilizar una alternativa diferente de los compiladores para traducir lenguajes de alto nivel.</p>
<i>Según el paradigma de programación.</i>	<p><i>Lenguajes imperativos:</i> Aquellos en los cuales se le ordena a la computadora cómo realizar una tarea siguiendo una serie de pasos o instrucciones. Los programas que usan un lenguaje imperativo especifican un algoritmo, usan declaraciones, expresiones y sentencias.</p>
	<p><i>Lenguajes orientados a objetos:</i> La programación orientada a objetos expresa un programa como un conjunto de objetos, que colaboran para realizar tareas. En la programación orientada al objeto, los datos, instrucciones y otros procedimientos de programación se agrupan en un elemento denominado objeto.</p>
	<p><i>Lenguajes declarativos:</i> Los programas que usan un lenguaje declarativo especifican las propiedades que la salida debe conocer y no especifica cualquier detalle de implementación. Dos amplias categorías de lenguajes declarativos son los lenguajes funcionales y los lenguajes lógicos. Los lenguajes funcionales no permiten asignaciones de variables no locales, así, se hacen más fáciles, por ejemplo, programas como funciones matemáticas. El principio detrás de los lenguajes lógicos es definir el problema que se quiere resolver (el objetivo) y dejar los detalles de la solución al sistema.</p>

Fuente: Elaboración propia basado en Gómez, 2009.

Puede ser cualquier combinación organizada de personas, hardware, software, redes de comunicación y recursos de información que almacene, recupere, transforme y disemine información en una organización. (O'Brien y Maracas, 2006).

Es una conjunción de elementos como el talento humano, el hardware, el software, redes que trabajen en forma armónica, organizada, sincronizada e integrada para procesar datos, ordenándolos, transformándolos y analizándolos, a través de los diferentes subsistemas organizacionales para producir información. (Rodríguez, 2012).

Es importante aclarar que en todas las definiciones anteriores todos los elementos interactúan para procesar los datos y dan lugar a información más elaborada, que se distribuye de la manera más adecuada posible en una determinada organización, en función de sus objetivos, y esto se aplica tanto los procesos manuales como a los procesos automáticos. Habitualmente se cree que todos los sistemas de información hacen su procesamiento de forma automática.

Muchos sistemas de información son inicialmente sistemas manuales que después se convierten en sistemas computarizados. En esta investigación, cuando hablemos de Sistemas de información nos vamos a referir siempre a los sistemas de información computarizados, por lo cual nuestro concepto de este tipo de sistema es: "Sistema de información automatizado, es un conjunto de elementos, como: software, hardware, base de datos, telecomunicaciones, personas y procedimientos específicamente configurados para recolectar, almacenar, y procesar datos para ser convertidos en información oportuna y pertinente para la toma de decisiones". (O'Brien, 2003).

### *Componentes o elementos de un sistema de información*

Según Fuentes (2005: 17-21), los sistemas de información se componen de los siguientes elementos:

- a) **Software.** Es el componente lógico, está constituido por los programas de computación, las rutinas e instrucciones que conforman el sistema de información.
- b) **Hardware.** Equipo de computación que se utiliza para llevar a cabo las actividades de entrada, procesamiento y salida, y pueden ser teclados, dispositivos entre otros.
- c) **Base de Datos.** Es un conjunto organizado de datos e información
- d) **Telecomunicaciones.** Son la transmisión electrónica de señales de comunicación que permite a las organizaciones crear redes de sistemas de computación. Ya que hacen posible que las personas se comuniquen entre por diferentes canales y medios, y facilite así el trabajo y el acceso a la información.
- e) **Personas.** Son el elemento más importante de la mayoría de los sistemas de información basados en computadoras. El personal de sistemas de información incluye a todos los individuos que administran, operan, programan y mantienen el sistema.  
Son todos aquellos que utilizan el sistema para obtener resultados, como pueden ser los ejecutivos financieros, operadores de manufactura, etc.
- f) **Procedimientos.** Son las estrategias, políticas, métodos y reglas para el uso de los sistemas de información automatizados.

### *Actividades de los Sistemas de Información*

Senn (2005: 23), considera que: "las finalidades de los sistemas de información, como la de cualquier sistema dentro de una organización son: procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas".

Según Fuentes (2003: 21-24), los sistemas de información realizan las siguientes actividades principales basadas en la computadora:

- **Entrada de información:** proceso en el cual el sistema toma los datos que requiere para procesar la información, por medio de estaciones de trabajo, teclado, diskettes, cintas magnéticas, código de barras, etc.
- **Almacenamiento de información:** es una de las actividades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sesión o proceso anterior.



- **Procesamiento de la información:** esta característica de los sistemas permite la transformación de los datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general en un año base.
- **Salida de información:** es la capacidad de un SI para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, graficadores, cintas magnéticas, diskettes, la voz, etc.
- **Retroalimentación:** es aquí, donde sirve para hacer cambios en actividades de entrada o procesamiento. Y se da cuando existe la presencia de errores o problemas, y se tiene la necesidad de corregir datos de entrada o también puede ser cuando se quiere modificar o mejorar un proceso. Esta actividad es de gran importancia para administradores y tomadores de decisiones.

**Clasificación de los Sistemas de Información.**

Kendall y Kendall (2003: 4-5) consideran que los sistemas de información se desarrollan con diferentes propósitos, los cuales dependen de las necesidades de la empresa. A continuación presentamos una tabla que resume todas las categorías según sus propósitos:

Tabla 2: Clasificación de los Sistemas de Información

<i>Según los niveles organizacionales</i>	Sistemas Departamentales
	Sistemas Institucionales
	Sistemas Inter-Organizacionales
<i>Según las Áreas funcionales(Actividades de Misión Central)</i>	Sistemas de Contabilidad
	Sistemas de Finanzas
	Sistemas de Manufacturas(Operaciones/Producción
	Sistemas de Mercadeo y Ventas
	S. de Administración de personal.
<i>Según el tipo de apoyo que proveen-</i>	Sistema de procesamiento de transacciones (TPS).
	Sistemas de información gerencial (MIS).
	Sistemas de soporte a decisiones (DSS).
	Sistemas de información ejecutiva (EIS).
<i>Según la Arquitectura Computacional</i>	Centralizado basado en mainframes
	Centralizado basado en red de servidores
	Distribuidos(Cliente-Servidor, Web-Intranet, Orientado a Servicios)
	PC Independientes
<i>Según el Nivel de Actividad Soportada</i>	Estratégico
	Táctico/Gerencial-Medio
	Operacional

Fuente: Elaboración propia basado en Casas, 2010.

## *Ciclo de vida clásico del desarrollo de sistemas*

Según James Senn (2005), el método de ciclo de vida para el desarrollo de sistemas es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. El método del ciclo de vida para el desarrollo de sistemas consta de las siguientes Etapas y cada una de ellas de diferentes fases: Planificación, Análisis, Diseño, Implementación, Pruebas, Implantación o Instalación y Mantenimiento.

### *1. Planificación*

El propósito de la planeación es buscar información suficiente para determinar si se debe continuar con el Ciclo de Vida del Desarrollo del Sistema.

Las tareas iniciales que se realizarán esta fase inicial del proyecto incluyen actividades tales como la determinación del ámbito del proyecto, la realización de un estudio de viabilidad, el análisis de los riesgos asociados al proyecto, Los beneficios que se obtendrán con el proyecto, una estimación del coste del proyecto, su planificación temporal y la asignación de recursos a las distintas etapas del proyecto.

La planificación es fundamental en la gestión de un proyecto de desarrollo de software. Se puede decir que esta etapa es una investigación exploratoria, porque es el primer acercamiento al problema.

### *2. Análisis*

La etapa de análisis en el ciclo de vida del software corresponde al proceso mediante el cual se intenta descubrir qué es lo que realmente se necesita y se llega a una comprensión adecuada de los requerimientos del sistema (las características que el sistema debe poseer). Lo primero que debemos hacer para construir un sistema de información es averiguar qué es exactamente lo que tiene que hacer el sistema. Los analistas, al trabajar con los empleados y administradores, deben estudiar los procesos de una empresa para dar respuesta a las siguientes preguntas clave:

¿Qué es lo que hace?

¿Cómo se hace?

¿Con que frecuencia se presenta?

¿Qué tan grande es el volumen de transacciones o decisiones?

¿Cuál es el grado de eficiencia con el que se efectúan las tareas?

¿Existe algún problema? ¿Qué tan serio es? ¿Cuál es la causa que lo origina?

### *3. Diseño del sistema*

El diseño de un sistema de información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. Los especialistas en sistemas se refieren, con frecuencia, a esta etapa como diseño lógico en contraste con la del desarrollo del software, a la que denominan diseño físico.

En la fase de diseño se han de estudiar posibles alternativas de implementación para el sistema de información que hemos de construir y se ha de decidir la estructura general que tendrá el sistema (su diseño arquitectónico). El diseño de un sistema de información también presenta distintas facetas:

- Abordar el diseño de la base de datos, diseñar las aplicaciones que permitirán al usuario utilizar el sistema de información. Tendremos que diseñar la interfaz de usuario del sistema y los distintos componentes en que se descomponen las aplicaciones.

### *4. Implementación del software*

Una vez que sabemos qué funciones debe desempeñar nuestro sistema de información (análisis) y hemos decidido cómo vamos a organizar sus distintos componentes (diseño), es el momento de pasar a la etapa de implementación. Para la fase de implementación hemos de seleccionar las

herramientas adecuadas, un entorno de desarrollo que facilite nuestro trabajo y un lenguaje de programación apropiado para el tipo de sistema que vayamos a construir. La elección de estas herramientas dependerá en gran parte de las decisiones de diseño que hayamos tomado hasta el momento y del entorno en el que nuestro sistema deberá funcionar.

Además de las tareas de programación asociadas a los distintos componentes de nuestro sistema, en la fase de implementación también hemos de encargarnos de la adquisición de todos los recursos necesarios para que el sistema funcione (por ejemplo, las licencias de uso del sistema gestor de bases de datos que vayamos a utilizar). Usualmente, también desarrollaremos algunos casos de prueba que nos permitan ir comprobando el funcionamiento de nuestro sistema conforme vamos construyéndolo.

#### *4. Prueba de sistemas*

Durante la prueba de sistemas, el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir, que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga.

Errar es humano y la etapa de pruebas tiene como objetivo detectar los errores que se hayan podido cometer en las etapas anteriores del proyecto (y, eventualmente, corregirlos). Lo suyo, además, es hacerlo antes de que el usuario final del sistema los tenga que sufrir. De hecho, una prueba es un éxito cuando se detecta un error (y no al revés, como nos gustaría pensar). La búsqueda de errores que se realiza en la etapa de pruebas puede adaptar distintas formas, en función del contexto y de la fase del proyecto en la que nos encontremos.

#### *5. Implantación*

La implantación es el proceso de verificar e instalar nuevo equipo, entrenar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarla. Una vez instaladas, las aplicaciones se emplean durante muchos años. Sin embargo, las organizaciones y los usuarios cambian con el paso del tiempo, incluso el ambiente es diferente con el paso de las semanas y los meses. De cara a su instalación, hemos de planificar el entorno en el que el sistema debe funcionar, tanto hardware como software: equipos necesarios y su configuración física, redes de interconexión entre los equipos y de acceso a sistemas externos, sistemas operativos (actualizados para evitar problemas de seguridad), bibliotecas y componentes suministrados por terceras partes, etcétera.

#### *6. Mantenimiento*

La etapa de mantenimiento consume típicamente del 40 al 80 por ciento de los recursos de una empresa de desarrollo de software. Dada la naturaleza del software, que ni se rompe ni se desgasta con el uso, su mantenimiento incluye tres facetas diferentes:

- Eliminar los defectos que se detecten durante su vida útil (mantenimiento correctivo), lo primero que a uno se le viene a la cabeza cuando piensa en el mantenimiento de cualquier cosa.
- Adaptarlo a nuevas necesidades (mantenimiento adaptativo), cuando el sistema ha de funcionar sobre una nueva versión del sistema operativo o en un entorno hardware diferente, por ejemplo.
- Añadirle nueva funcionalidad (mantenimiento perfectivo), cuando se proponen características deseables que supondrían una mejora del sistema ya existente.

Si examinamos las tareas que se llevan a cabo durante la etapa de mantenimiento, nos encontramos que en el mantenimiento se repiten todas las etapas que ya hemos visto del ciclo de vida de un sistema de información. Al tratar principalmente de cómo añadirle nueva funcionalidad a un sistema ya existente, el mantenimiento repite "en miniatura" el ciclo de vida completo de un sistema de información. Es más, a las tareas normales de desarrollo hemos de añadirle una nueva, comprender el sistema que ya existe, por lo que se podría decir que el mantenimiento de un sistema es más difícil que su desarrollo (Glass, 2003).

## *La programación y los Sistemas de información*

La programación es una de las áreas profesionales que más ha aumentado en popularidad durante los últimos años. Esto se debe por supuesto al rápido crecimiento que ha tenido Internet y en general el mundo de las tecnologías de la comunicación. Por ello cada vez son más los interesados en aprender lenguajes de programación ya que estas son herramientas que ayudan en el camino de cualquier profesional. (Vega y Espinel, 2010).

La importancia de la programación radica a través del tiempo el conocimiento del hombre ha evolucionado y hoy en día, existe la necesidad de utilizar la tecnología para poder estar en un mundo globalizado y actualizado. Pero no sólo es necesaria la inclusión de las más modernas tecnologías de información y comunicación para lograr sistemas de información eficaces, sino que además es imprescindible tener las herramientas y el conocimiento para desarrollarlas y es allí donde juega un papel importante la programación.

Es necesario comprender que la razón principal para que aprenda la programación es para resolver problemas. Pero también es muy importante que conozca el tipo de problemas que deberá resolver en su campo profesional, por ejemplo: en la banca, las instituciones gubernamentales, hospitales, el comercio, la industria, servicios turísticos, en distintas áreas, que necesitan el desarrollo de Sistemas de información, para el manejo eficiente de la información. (Vega y Espinel, 2010).

Castell (1998) nos dice: “En el Nuevo sistema tecnológico, económico y social, el incremento de la productividad no depende del incremento cuantitativo de los factores de producción (capital, trabajo, recursos naturales), sino de la aplicación de conocimientos e información a la gestión, producción y distribución, tanto en los procesos como en los productos”.

Este Nuevo Sistema se caracteriza por “el uso intensivo de las nuevas tecnologías en todos los sectores sociales y económicos, como herramientas para lograr la modernización de los mismos, la competitividad y el desarrollo auto sostenido del Territorio, modernizando la industria tradicional y todo esto es posible si se tiene conocimiento de la programación como una herramienta fundamental en el proceso de determinación de conceptos, procedimientos, modelos y algoritmos informáticos relacionados con las vertientes de desarrollo de la informática satisfaciendo las exigencias sociales caracterizada por la generación, la extensión, la flexibilidad y la autonomía en todas las áreas y procesos”.

Si analizamos el ciclo de vida clásico de los SI vemos que en la etapa de implementación, el concepto de programación es realmente importante porque es el que permite diseñar, codificar, depurar y mantener el código fuente del programa. Este es escrito en un lenguaje de programación y una vez terminado es el realmente lo que permitió la creación del SI con el propósito o función deseada.

La programación vienen entonces a ser en este nuevo entorno una herramienta fundamental para las diferentes ramas del conocimiento y la ingeniería y su aprendizaje es una necesidad imperiosa para los estudiantes, pues esta será la herramienta fundamental para avanzar en su proceso de formación profesional.

Hoy en día, existe la necesidad de utilizar la tecnología para poder estar en un mundo globalizado y actualizado. Los Sistemas de Información (SI) y las Tecnologías de Información (TI) han cambiado la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos, suministran una plataforma de información necesaria para la toma de decisiones y, lo más importante, su implantación logra ventajas competitivas o reducir la ventaja de los rivales. (Castell, 2006).

Para continuar operando en estos nuevos mercados y mantener la competitividad es necesario un firme control de las operaciones y una dirección coordinada globalmente, para esto es esencial contar con sistemas de información flexibles y rápidos. (Cohen y Asin, 2000).

Kendall y Kendall (2004) resaltan que los sistemas de información son reconocidos como un arma estratégica que se usa para la obtención y mantenimiento de una ventaja competitiva, bien actuando sobre el producto o sobre el proceso: para de esta manera mejorar el rendimiento de la empresa, y la competitividad, facilitando nuevas maneras de organizarse, gestionar y para llevar a cabo nuevos negocios.

Según Guzmán (2002), los sistemas de información se encargan específicamente de:

- a. Proporcionar, facilitar y ejecutar automáticamente procesos que constantemente se realizan manualmente.
- b. Dar información y datos para ayudar a la toma de decisiones.
- c. Interaccionar con el usuario de manera más profunda.

Según Fuentes (2005), los usos que se dan a los sistemas de información dentro de las organizaciones son los siguientes:

- Respaldo a procesos operativos u operaciones empresariales
- Generación de información de apoyo al proceso de toma de decisiones
- Obtención de ventajas competitivas mediante su implantación y uso

Curiel (1999) nos dice que todo proceso o modelo de innovación tecnológica está ligado a un proceso de toma de decisiones que requiere de sistemas y servicios de información altamente eficiente.

En la actualidad la información es considerada como un recurso estratégico para las actividades tecnológicas, de innovación y empresarial en sentido general, constituyendo un instrumento básico para diagnosticar la ventaja competitiva y encontrar maneras para identificarla. Para lograr esta ventaja competitiva son necesarios los sistemas de información, como elemento clave en el proceso de toma de decisiones.

## Metodología de estudio

La investigación es de tipo descriptivo- correlacional, ya que describe, el nivel de conocimiento en programación y su relación con el desarrollo de sistemas de información automatizados en los estudiantes de ingeniería de Sistemas de la Universidad de la Guajira, basado en el criterio de Fernández, Hernández & Baptista (2003).

La población del estudio estuvo constituida por todos los estudiantes del programa de ingeniería de Sistemas, de la Facultad de ingeniería de la Universidad de la Guajira que son 321 en total.

Se seleccionó el muestreo probabilístico porque permite reducir al mínimo el tamaño del error en las predicciones. De esta manera, esta técnica es esencial en los diseños de investigación por encuesta, por cuanto posibilita la realización de estimaciones perteneciente al programa de Ingeniería de sistemas, ya que tiene la misma probabilidad de ser elegido. El tamaño de la muestra de los estudiantes, se determinó aplicando, la fórmula recomendada por Sierra Bravo (1996), la cual consiste en:

$$n = \frac{4 \cdot p \cdot q \cdot N}{E^2 (N-1) + 4 \cdot p \cdot q}$$

Dónde: n = es el tamaño de la muestra que se calculará.

4 = es una constante.

p y q = son las probabilidades de éxito que tiene una valor asignado de 50% cada una.

N = es el tamaño de la población.

E2 = es el error seleccionado por el investigador.

Remplazando los valores, en la formula obtenemos:

$$n = \frac{4 \cdot 321 \cdot 50 \cdot 50}{52 (321-1) + 4 \cdot 50 \cdot 50} = \frac{3,210.000}{26.640} = 120,4 \sim 120$$

Se obtuvo una muestra total de ciento veinte (120) alumnos del total de la población de estudiantes de ingeniería de sistemas de la cual se realizó una estratificación aleatoria de tres grupos, representadas por los diferentes semestres de estudiantes del programa. Por lo que se utilizó la fórmula de Shiffer, citada por Chávez (2001), lo cual permitió calcular el tamaño de los tres grupos considerando los semestres así: de 1-4, de 5-8 y de 9-10.

$$n_i = \frac{nh * n}{N}, \text{ de donde:}$$

- Ni= Estrato que se determinará
- n= Tamaño adecuado de la muestra
- nh= Tamaño del estrato de la población
- N= Tamaño de la población

De la aplicación de la fórmula reemplazando cada uno de los valores se obtuvo una muestra estratificada. (Ver Tabla 3).

Tabla 3: Muestra estudiantes

<i>Semestre Estudiantes</i>	<i>N°. de Estudiantes</i>	<i>%</i>	<i>Muestra de Estudiantes</i>	<i>%</i>
<i>Del 1 al 4 semestre</i>	153	47	57	47
<i>Del 5 al 8 semestre</i>	86	27	31	27
<i>Del 9 al 10 semestre</i>	82	26	32	26
<i>Total:</i>	321	100%	120	100%

Fuente: Romero, 2014.

En este estudio se aplicó como instrumento un cuestionario conformado por 45 ítems con cinco (5) alternativas de respuestas con un escalamiento tipo Lickert 5) siempre; 4) casi siempre; 3) algunas veces; 2) casi nunca; 1) Nunca, se diseñó tomando en consideración, tanto el objetivo general como cada uno de los ítems, indicadores y dimensiones que precisan las variables.

Para realizar el procesamiento y tratamiento estadístico de los resultados arrojados por el cuestionario aplicado a los estudiantes del programa ingeniería de Sistemas, se utilizó el programa SPSS versión 21 que permitió tabular de una manera organizada y automatizada los datos y se utilizó el método de Estadística Descriptiva, para hacer un análisis cuantitativo y la descripción de las variables de estudio, específicamente a través del uso de las técnicas: medidas de tendencia (media aritmética) y de variabilidad(desviación estándar).

El nivel de conocimiento se midió a través de dos subvariables: La actitud de los estudiantes y los conocimientos en programación. El desarrollo de Software se midió analizando las generalidades de los SI y la implementación de SI.

### **Análisis e interpretación de los resultados**

Para realizar la interpretación de los resultados, se construyó una tabla de rango, intervalo y de categoría, la cual se muestra a continuación (ver Tabla 4).

En lo que respecta a la medida de variabilidad, para indicar el grado de dispersión de las respuestas con relación a la escala de medición utilizada y su rango, representado por las puntuaciones mayor y menor obtenidos, es decir, cinco(5) y uno (1) respectivamente. (Vid. *infra* Tabla 5).

Una vez tabulados y procesados los datos obtenidos de la muestra propuesta se procedió al cálculo del promedio y la desviación estándar de cada uno de las dimensiones planteadas con respecto a las variables tratadas; luego estos resultados se compararon con las Tablas 4 y 5 para interpretar su significado.

Tabla 4: Media

<i>Intervalo</i>	<i>Categoría</i>
4,20 < x̄ ≤ 5,00	Muy alta

<i>Intervalo</i>	<i>Categoría</i>
$3,40 < \bar{x} \leq 4,20$	Alta
$2,60 < \bar{x} \leq 3,40$	Moderada
$1,80 < \bar{x} \leq 2,60$	Baja
$1,00 < \bar{x} \leq 1,80$	Muy baja

Fuente: Romero, 2014.

Tabla 5: Desviación estándar

<i>Intervalo</i>	<i>Categoría</i>
$1,60 < \sigma \leq 2,00$	Alta dispersión
$1,20 < \sigma \leq 1,60$	Dispersión
$0,80 < \sigma \leq 1,20$	Moderada dispersión
$0,40 < \sigma \leq 0,80$	Baja dispersión
$0,00 < \sigma \leq 0,40$	Muy baja dispersión

Fuente: Romero, 2014.

A continuación se presenta la Tabla 6. Con los resultados que corresponden a las subdimensiones contenidas en la dimensión Actitud de los estudiantes frente a la programación y que dan respuesta al primer objetivo.

Tabla 6: Dimensión: Actitud de los estudiantes hacia la programación

<i>Sub dimensiones</i>	<i>Promedio</i>	<i>Desviación</i>
<i>Actitud cognitiva</i>	4.10	0.66
<i>Actitud emocional o afectiva</i>	2.91	1.02
<i>Actitud conductual</i>	2.70	1.12
<i>Promedio general</i>	3.23	0.93

Fuente: Romero, 2014.

Para la dimensión actitud de los estudiantes frente a la programación de la variable nivel de conocimientos en programación de los estudiantes de ingeniería de sistemas de la Universidad de la Guajira, se obtuvo un promedio con categoría moderada de 3.23, con una muy baja dispersión de las respuestas de 0.93; la cual se ubica dentro de una frecuencia moderada, con una muy alta confiabilidad; estos resultados reflejan que de acuerdo a la opinión de los Estudiantes, a pesar que ellos tienen creencias positivas sobre la programación y reconocen que es una herramienta importante en el desarrollo de software y SI y son conscientes de las múltiples ventajas que estos ofrecen en todas las áreas del saber; no se sienten motivados a utilizar esta herramienta, les parece aburrido, difícil y por lo tanto no programan, ni promueven el uso de la programación entre sus compañeros.

En la siguiente Tabla, tenemos los resultados de las subdimensiones contenidas en la dimensión Conocimientos de programación de los estudiantes de ingeniería de sistemas que dan respuesta al segundo objetivo.

Tabla 7. Dimensión: Conocimientos en programación de los estudiantes

<i>Sub dimensiones</i>	<i>Promedio</i>	<i>Desviación</i>
<i>Conocimientos teóricos</i>	3.82	0.56
<i>Conocimientos prácticos</i>	2.12	0.51
<i>Experiencia</i>	2.10	0.59
<i>Promedio general</i>	2.91	0.55

Fuente: Romero, 2014.

Se observa en esta Tabla, que los conocimientos teóricos obtuvieron un promedio con categoría alta de 3.82 y una muy baja dispersión de 0.56 la cual se ubica en una frecuencia alta y una muy alta confiabilidad; los conocimientos prácticos obtuvieron un promedio con categoría baja de 2.12 y una muy baja dispersión de 0.51 la cual se ubica en una frecuencia baja y una muy alta confiabilidad, la

experiencia obtuvo un promedio con categoría baja de 2.10 y una muy baja dispersión de 0.59 la cual se ubica en una frecuencia baja y una muy alta confiabilidad; y el promedio general con categoría baja de 2.91 y una muy baja dispersión de 0.55 la cual se ubica en una frecuencia baja y una muy alta confiabilidad. Estos resultados nos dicen que los estudiantes tienen conocimientos teóricos de la programación, tienen claros los conceptos de algoritmos, metodologías para la solución de problemas, Codificación; pero a pesar de esto, no práctica con mucha frecuencia ni el desarrollo de algoritmos para desarrollar la lógica de programación, ni la codificación de programas en algún lenguaje. Por esta misma razón no tienen ni experiencia ni experticia en programación que al final se traduce en un nivel bajo de conocimientos de programación.

En la Tabla 8. Se presenta los resultados del variable independiente nivel de Conocimiento en Programación teniendo en cuenta las dimensiones Actitud y Conocimientos de programación de los Estudiantes del programa de ingeniería de Sistemas.

Tabla 8: Variable: Nivel de conocimientos en Programación

<i>Dimensiones</i>	<i>Promedio</i>	<i>Desviación</i>
<i>Actitud de los Estudiantes</i>	3.23	0.93
<i>Conocimientos en Programación</i>	2.91	0.55
<i>Promedio general</i>	3,07	0.74

Fuente: Romero, 2013.

En lo que respecta a la variable nivel de conocimiento de los estudiantes del programa de ingeniería de sistemas, se obtuvo un promedio con categoría moderada de 3,07, con una muy baja dispersión de las respuestas de 0.74; la cual se ubica dentro de una frecuencia moderada, con una muy alta confiabilidad; estos resultados, reflejan que de acuerdo a la opinión de los estudiantes del programa el nivel de conocimientos en programación es moderado, porque tienen una actitud apática frente a la programación y a pesar de que reconocen la importancia de la programación en su carrera y en la solución de problemas en el mundo real y además entienden todos los conceptos teóricos de la programación, no practican y por esta misma razón carecen de experiencia y experticia en programación.

En la tabla 9. Se muestran los resultados de la variable dependiente Desarrollo de SI y sus Dimensiones Conocimientos Generales de SI, Implementación de SI.

Tabla 9: Variable: Desarrollo de SI

<i>Dimensiones</i>	<i>Promedio</i>	<i>Desviación</i>
<i>Conocimientos Generales de SI</i>	3.50	0.93
<i>Implementación de SI</i>	2.62	0.55
<i>Promedio general</i>	3.06	0.74

Fuente: Romero, 2014.

Estos resultados nos muestran que los estudiantes en los conocimientos generales sobre SI obtuvieron un promedio con categoría alta de 3.50 y una muy baja dispersión de 0.93 la cual se ubica en una frecuencia Alta y una muy alta confiabilidad; en la implementación de SI, obtuvieron un promedio con categoría media de 2.62 y una muy baja dispersión de 0.51 la cual se ubica en una frecuencia media y una muy alta confiabilidad, Estos resultados nos dicen que los estudiantes tienen conocimientos Generales de SI, reconocen la importancia y las ventajas de los SI en todas las áreas, tienen claros los conceptos de SI, Clasificación De los SI, Actividades, Tipos y conceptos en general, pero a pesar de esto, son pocos los que han desarrollado SI; principalmente por el nivel de conocimientos que tienen en programación.



Para el tercer objetivo se analizó la relación entre el nivel de conocimiento y el desarrollo de SI, se tomaron los resultados del nivel de Conocimientos en programación y el desarrollo de SI, para determinar esta relación se utilizó la correlación de Pearson, obteniendo un valor de 0,743 que está en el rango de correlación positiva de media a considerable, Según la tabla 10, que muestra los rangos de correlación de Pearson y su descripción. Lo cual quiere decir que existe una relación directamente proporcional entre las variables nivel de Conocimientos en programación y desarrollo de SI, A menor conocimiento en programación menor desarrollo de SI. En general hay una correspondencia directa entre las dos variables, los conocimientos en programación estimulan el desarrollo de SI de calidad. (Ver Tabla 10 y 11).

Tabla 10: Coeficiente de correlación de Pearson

<i>Rango de correlación</i>	<i>Descripción</i>
r = - 1.00	Correlación negativa perfecta
r = - 0.90	Correlación negativa muy fuerte
r = - 0.75	Correlación negativa considerable
r = - 0.50	Correlación negativa media
r = - 0.10	Correlación negativa débil
r = 0.00	No existe correlación alguna
r = +0.10	Correlación positiva débil
r = +0.50	Correlación positiva media
r = +0.75	Correlación positiva considerable
r = +0.90	Correlación positiva muy fuerte
r = +1.00	Correlación positiva perfecta

Fuente: Romero, 2014.

Tabla 11: Correlación de Pearson. Conocimientos en programación y desarrollo de SI

CONOCIMIENTOS EN PROGRAMACION- DESARROLLO SI	CORRELACION DE PEARSON
	0,743

Fuente: Romero, 2013.

Esta relación nos demuestra que el nivel de conocimientos en programación de 3,07 que fue medio, se traduce directamente en una categoría media de 3,06 para el desarrollo de sistemas de información.

## Conclusiones

La programación es una actividad que implica un proceso mental, generalmente complejo y creativo, exige del programador: inteligencia, conocimiento, habilidades y disciplina. La inteligencia es un recurso natural que tienen todas las personas normales. La adquisición del conocimiento se logra estudiando los conceptos, fundamentos y las técnicas básicas de programación.

En cuanto a la adquisición de las habilidades, la voluntad del estudiante es fundamental, ya que sólo con la práctica y la experimentación personal podrá lograrse. La disciplina, como observancia de las normas y el buen proceder, la podrá adquirir el estudiante con la motivación e interés que tenga en esta área.

Los resultados encontrados evidenciaron un nivel de conocimientos en programación, moderado, este resultado tiene que ver directamente con la actitud y los conocimientos teóricos, prácticos y la experiencia de los estudiantes. Que reflejan en su mayoría apatía por la programación, carencia de habilidades para programar, carencia de disciplina en programación; muy a pesar que conocen y entienden la importancia y ventajas de la programación en todas las áreas del saber y que algunos manejan los conocimientos teóricos necesarios.

Estos resultados a su vez se reflejan directamente en nuestra segunda variable, desarrollo de SI que muestran también una categoría moderada precisamente porque el conocimiento de los estudiantes en los lenguajes de programación presenta un nivel medio.

Por todo lo anterior el nivel de conocimiento de los estudiantes de ingeniería de sistemas es moderado y el desarrollo de SI de información también es moderado, les falta más práctica, más disciplina y más motivación hacia la programación y el desarrollo de SI como herramientas fundamentales en su carrera y en el mundo real para la solución de problemas en todas las áreas del saber.

## Recomendaciones

Es muy fundamental, tener en cuenta que en este mundo globalizado donde prima el uso de la tecnología, no solamente debemos enfocarnos a utilizarla, sino también a crearla y es aquí donde juega un papel importante los lenguajes de programación y el profesional de sistemas, con los conocimientos necesarios, para crear, facilitar, gestionar y dar soluciones tecnológicas a cada una de la áreas del saber.

Es por esto que los estudiantes de los programa de ingeniería de sistemas y carreras afines deben, adquirir conocimientos y competencias en la programación, para el desarrollo de software y sistemas de información que cumplan todas las exigencias de las necesidades de información actuales.

Para alcanzar un buen nivel en programación que permita el desarrollo de Software y SI de calidad se recomienda tener en cuenta los siguientes aspectos:

- Se debe explicar a los estudiantes desde el principio de la carrera la importancia de la programación en su formación profesional y en el ejercicio profesional.
- Enfatizar en nuestros estudiantes que la programación requiere de inteligencia, conocimientos; pero también de habilidades que solo se consiguen con la práctica y la disciplina que ellos tomen al respecto.
- Ubicar la programación en la fase de desarrollo del ciclo de vida de los sistemas de información.
- Diferenciar plenamente la programación de la codificación.
- Privilegiar la enseñanza de la programación sobre los lenguajes de programación.
- Motivar hacia el uso de la programación y el desarrollo de Sistemas de Información
- Mejorar las estrategias y la metodología tradicional en la enseñanza de la programación, para ello es necesario apoyarse en las nuevas tecnologías.
- Utilizar lenguajes y herramientas de programación de última generación.

## REFERENCIAS

- Aparisi, J. y Ripoll F (2000). “Relevancia de la tecnología de la Información y de los sistemas de información estratégica”. I Encuentro Iberoamericano de Contabilidad de Gestión Valencia. España. Fuente: <http://www.observatorio-iberoamericano.org/>.
- Allport, G. (1967). *Teorías de la Personalidad*. Buenos Aires: Ed. Eudeba.
- Bueno (2004). *El mito de la cultura*. Barcelona: Editorial Prensa Ibérica.
- Casas (2010). *Definición y clasificación de los sistemas de información*. Pontificia Universidad Católica de Chile.
- Castell (2006). *La era de la Información: economía, sociedad y cultura* (vol. 3). España: Alianza editorial.
- Cohen y Asin (2000) *Sistemas de Información un enfoque de toma de decisiones*. México: Mc Graw Hill.
- Curiel (1999). “Los Sistemas de Información y la Innovación Tecnológica”. *Revista avanzada y científica IDIC*. Centro de Información y Gestión Tecnológica. CITMA. Matanzas Cuba 1999.
- Eagly, A., & Chaiken, S. (1993). *Psychology of Attitudes*. NY: Harcourt.
- García, Chamorro y Molina (2000). *Informática de Gestión y Sistemas de Información*. 1ª Edición. Madrid: Mc Graw Hill.
- Gómez, Rodrigo (2009). *Curso de Programación y Lógica*. Madrid: Ediciones Santillana.
- González H., Walfredo (2000). “Hacia un enfoque sistémico en la enseñanza de la Informática”. *Colección Matemática I*. MINED.
- Hernández, Fernández y Batista (2003). *Metodología de la Investigación*. Cuarta Edición. México: Mc. Graw Hill.
- Joyanes A., Luis (1998). *Fundamentos de Programación. Algoritmos y Estructura de Datos*. España: McGraw Hill.
- (2000). *Cibersociedad. Los retos Sociales ante un Nuevo Mundo Digital*. Primera Edición. Bogotá: Mc. Graw Hill.
- (2006). *Programación en c++*. Algoritmos, estructuras de datos y objetos (2ª ed.). España: McGraw Hill.
- (2008). *Fundamentos de la programación*. España: McGraw Hill
- Kalakota, Ravi (2000). “Prólogo” en *Del e-Commerce al e-Business. El siguiente paso*. España: Editorial Rama
- Laudon y Laudon (2000). *Administración de los Sistemas de Información. Organización y Tecnología*. 3ª Edición. México: Prentice Hall.
- (2004). *Sistemas de información gerencial: Administración de la empresa digital*. México. Prentice Hall.
- Levine, G. (2002). *Computación y programación moderna. Perspectiva integral de la informática*. México: Pearson educación.
- Kendall y Kendall (2005). *Análisis y Diseño DE Sistemas E*. México: Editorial Pearson.
- O’Brien, James (2003). *Sistemas de información gerencial*. Cuarta Edición. Colombia: Irwin-McGraw Hill.
- O'Brien, y Marakas (2006). *Sistemas de información gerencial*. Colombia: McGraw-Hill/Irwin.
- Pressman, Roger (2006). *Ingeniería del Software. Un enfoque práctico*. Sexta Edición. México: McGrawHill.
- Piatinni, Mario y Emilio del Peso (2001). *Auditoría Informática* 2da Edición, España: Editorial RA-MA.
- Rastrollo H. Ana María y Ana M. Castillo (2004). “Nuevas TIC y Estructura Organizativa: de la burocracia vertical a la empresa red. Dirección y organización” *Revista de dirección, organización y administración de empresas*, 30. pp. 134-144. Fuente: <http://dialnet.unirioja.es/servlet/articulo?codigo=876906>
- Salazar, J. et al. (1992). *Psicología social*. México. Editorial Trillas.
- Stair, M.N. / Reynolds G. W. (2000) *Principios de Sistemas de Información*. 4ta Edición. México: International Tomson Editores.

Senn (2005). *Análisis y Diseños de Sistemas de Información*. Madrid: Pearson Prentice Hall.

Vega, Espinel (2010). “Aspectos Importantes de la enseñanza de la programación”. *Revista Avances en Sistemas e Informática*. 7 (1).

Urrutia, Amaia (2000). “Comunicación en la empresa. Importancia de la información interna en la empresa”. *Revista Latina de Comunicación Social*. 3(27). Fuente: [http://redalyc/uaemex.mx](http://redalyc.uaemex.mx)

## **SOBRE LOS AUTORES**

**Sandy Romero Cuello:** Sandy Romero Cuello, Docente Investigadora de la Universidad de la Guajira (Colombia), adscrita a la Facultad de Ingeniería, programa de Ingeniería de Sistemas. Ingeniero de sistemas. Especialista en Administrador de Empresas. Magister en Telemática. Doctorando en Ciencia y Tecnología.

**Jaider Quintero Mendoza:** Jaider Quintero Mendoza, Docente- investigador de la Universidad de la Guajira (Colombia). Adscrito a la Facultad de ingeniería, Programa de ingeniería de Sistemas. Ingeniero de sistemas. Especialista en Gerencia Financiera. Magister en Telemática. Doctorando en Ciencia y Tecnología.