

**LA IMPORTANCIA DE LA INFORMÁTICA EN LA INGENIERÍA
MECATRÓNICA: UNA EXPERIENCIA REFLEXIVA**

**THE IMPORTANCE OF COMPUTERS IN MECHANICAL ENGINEERING: A
REFLECTIVE EXPERIENCE**

Autores: Pablo Benavides ^[1], Washington Mosquera ^[1], Diego Peluffo ^[2] y Diego Terán ^[1]

[1]: Carrera de Ingeniería en Mecatrónica, Universidad Técnica del Norte, Ibarra, Ecuador

[2]: Carrera de Ingeniería en Telecomunicaciones, Universidad Técnica del Norte, Ibarra,
Ecuador

LA IMPORTANCIA DE LA INFORMÁTICA EN LA INGENIERÍA MECATRÓNICA: UNA EXPERIENCIA REFLEXIVA

Pablo A. Benavides Bastidas^[1]

pabenavides@utn.edu.ec

Washington G. Mosquera Terán^[1]

wgmosquera@utn.edu.ec

Diego H. Peluffo Ordoñez^[2]

dhpeluffo@utn.edu.ec

Diego F. Terán Pineda^[1]

dfteran@utn.edu.ec

[1]: Carrera de Ingeniería en Mecatrónica, Universidad Técnica del Norte, Ibarra, Ecuador

[2]: Carrera de Ingeniería en Telecomunicaciones, Universidad Técnica del Norte, Ibarra,
Ecuador

RESUMEN

La ingeniería mecatrónica es una especialidad relativamente nueva, la cual se conforma por la sinergia y aprovechamiento de características y fortalezas de otras especialidades que ya tienen un largo trayecto a nivel profesional. Particularmente, la formación de un estudiante de mecatrónica involucra técnicas de programación que constituye el cerebro o parte inteligente para el desarrollo de cualquier sistema automático. Por tanto, el área de programación es de uso transversal y debe ser adecuadamente enseñada a los estudiantes de mecatrónica. No obstante, hay evidencia de que algunos educadores en mecatrónica han tomado la errada posición de que los futuros profesionales, al no ser desarrolladores de software, no deberían recibir enseñanza de la programación a profundidad, sino por el contrario, solamente una formación en conceptos básicos.

El método aplicado para la realización de este artículo es el experimental, aplicado a partir de experiencia en la práctica profesional y como docentes de las asignaturas de programación

básica, avanzada y sistemas de control, en la carrera de ingeniería en mecatrónica en la Universidad Técnica del Norte, lo cual ha permitido presentar una reflexión sobre la importancia y pertinencia del área de programación informática en la enseñanza de mecatrónica.

Como conclusión de este trabajo se presenta elementos de discusión y aspectos técnicos para tomar una posición ante la necesidad y pertinencia de la enseñanza de la programación informática para mecatrónica de forma adecuada, es decir, en contexto y a nivel requerido por un profesional en ingeniería, abarcando en partes iguales la sinergia de los saberes que conforman la carrera de Ingeniería en Mecatrónica.

PALABRAS CLAVE

Informática, mecatrónica, programación, computación.

ABSTRACT

Mechatronics engineering is a relatively new specialty, which is shaped by the synergy and utilization of characteristics and the strengths of other specialties that already have a long journey at the professional level. In particular, the training of a mechatronics student involves programming techniques that constitute the brain or intelligent part for the development of any automatic system. Therefore, the programming area is transversal and must be adequately taught to mechatronics students. However, there is evidence that some mechatronics educators have taken the position of what future professionals, there are no software developers, no treatment receive programming from a depth, but on the contrary, just a training in basic concepts.

The applied method for the accomplishment of this article is experimental, applied from the experience in the professional practice and as teachers of the subjects of basic programming, advanced and control systems, in the race of engineering in mechatronics in the Technical

University of the North, which has allowed to present a reflection on the importance and relevance of the area of computer programming in the teaching of mechatronics.

As a conclusion of this work the elements of discussion and the technicians are presented to take a position before the necessity and the pertinence of the teaching of the computer programming for mechatronics in an appropriate way, that is to say, in the context and level required by a professional In engineering, covering in equal parts the synergy of the acquaintances who make up the Engineering career in Mechatronics.

KEYWORDS

Informatic, mechatronic, programming, computing.

I. INTRODUCCIÓN

La Ingeniería Mecatrónica se presenta como una cooperación de varios saberes para la creación de resultados que aprovechen y maximicen las potencialidades de cada uno de ellos, de forma dinámica, y por tanto debe ser estudiada con criterios de enseñanza-aprendizaje claramente definidos. Por un lado, es una ventaja que las otras especialidades ya tengan su trayectoria altamente fortalecida, y, por otro lado, se constituye en una gran desventaja, al no poder determinar qué tan profunda debe ser la enseñanza en cada una de ellas.

La idea de que un estudiante de Ingeniería Mecatrónica no tiene la destreza ni el conocimiento necesario para profundizar en la elaboración de programas computacionales, considerando que los componentes necesarios a nivel de tiempo y profundidad de los temas no son adecuados y, de igual manera, que el perfil del Ingeniero Mecatrónico no se enfoca a ser un desarrollador puro (debido a que debe hacerse una distribución más justa del tiempo para desarrollar las otras áreas del conocimiento), es una suposición común en el contexto universitario y carece de argumentos sólidos. Esta idea se ha extendido a diferentes esferas tanto a nivel directivo, como docente e inclusive, a nivel de estudiantes. En efecto, a nivel docente es fundamental reconocer y fomentar la noción de que la formación del Ingeniero Mecatrónico debe estar a cargo de especialistas en áreas específicas del conocimiento,

quienes deberían sintetizar los conocimientos de su área específica, de acuerdo con el perfil del Ingeniero Mecatrónico y no exclusivamente de dicha área.

Se ha detectado que el principal inconveniente de un docente de especialidad que enseña en una carrera que está conformada por la sinergia de varias especialidades, se encuentra enmarcado en una percepción que abarca dos posibles escenarios: Por un lado, no puede apartarse fácilmente de sus raíces y trata de llevar a sus estudiantes por la línea en la que el mismo fue formado; y por otro lado, considera que sus estudiantes al ser de otra especialidad, no necesitan una enseñanza profunda de conceptos y habilidades.

Como docentes especializados de la Universidad Técnica del Norte, por casi cuatro décadas en conjunto, a nivel personal, se ha podido comprobar que esta percepción está muy equivocada, ya que, con una buena formación y orientación de los estudiantes en el ámbito de la programación de software, los estudiantes de Ingeniería Mecatrónica, pueden llegar a desarrollar habilidades para el diseño e implementación de programas informáticos, con igual calidad que la que normalmente desarrollan los estudiantes de Ingeniería de Software durante los primeros niveles de formación académica. No obstante, después de estos niveles, las carreras en general desarrollan la formación específica y, por tanto, es natural que los estudiantes de Ingeniería de Software, al especializarse en esta formación, despunten en conocimientos y habilidades en desarrollo de software, así como los estudiantes de ingeniería mecatrónica en temas de electrónica y mecánica.

Esto ha sido puesto en práctica en varios encuentros de programación básica, donde los estudiantes de Ingeniería Mecatrónica han obtenido buenos resultados al enfrentar a estudiantes de otras carreras de especialidades netamente dedicadas al desarrollo de software, y, en muchas ocasiones, los estudiantes de Ingeniería Mecatrónica han incluso brindado asesoría a los estudiantes de Desarrollo de Software de los primeros niveles.

Adicionalmente, la formación de los primeros niveles de Ingeniería, en lo referente a ciencias básicas y específicamente en este caso, en la programación, tiene como objetivo formar e inculcar en el estudiante de Ingeniería habilidades como: Analizar, diseñar e implementar

soluciones informáticas. Particularmente, en proyectos de Ingeniería Mecatrónica, habitualmente se requiere elaborar programas computacionales que se relacionen y puedan ser integrados, utilizando y aprovechando el uso de modernas herramientas de desarrollo informático. Por tanto, en este artículo se presenta una reflexión sobre la importancia de la programación en el desarrollo curricular de la carrera en Ingeniería Mecatrónica. Dicha reflexión se realiza desde un punto de vista de currículo, técnico-conceptual y experiencia personal.

El resto de artículo está estructurado así: En la sección II, se presenta las definiciones básicas relacionadas con Ingeniería Mecatrónica. En la sección III, se presenta la metodología aplicada. En la sección IV, se presenta el análisis de resultados. Finalmente, en la sección V, se menciona los comentarios finales, las conclusiones, el trabajo futuro de la presente reflexión y las referencias utilizadas para la elaboración del presente artículo.

II. DEFINICIONES BÁSICAS

En este apartado se define conceptos necesarios para la reflexión y la discusión posteriores, abarcando definiciones de: Ingeniería mecatrónica, y software.

Ingeniería mecatrónica

La ingeniería mecatrónica es una sinergia (combinación necesaria), entre ingeniería mecánica, electrónica, sistemas de control e informática, para crear procesos y maquinas inteligentes e innovadores que permitan solucionar problemas en maquinaria, robots, procesos industriales, autos, naves, aviones y demás relacionados que permitan mayor utilidad y desarrollo en la sociedad tanto en la ingeniería y en la ciencia y tecnología. [13]



Figura 1. Relación de la Mecatrónica con otras especialidades

Un error muy común es pensar que en esta especialidad, el uso de las computadoras no va más allá del diseño y simulación de piezas, la realización de documentos, escuchar música y ver fotos; sin embargo la realidad es completamente distinta, ya que el uso del dispositivos informáticos abarca más que un simple computador de escritorio, una laptop o una Tablet; en la actualidad la mayoría de equipos de uso cotidiano, contienen componentes electrónicos muy avanzados para poder ser controlado, así un micro ondas, una cocina, un refrigerador, una lavadora, un celular y hasta una simple cafetera puede contener avanzados sistemas en su interior.

Sistema Informático

Un sistema informático (conocido también como sistema de cómputo), se define como un conjunto de elementos organizados, que interactúan unos con otros, para lograr ciertos objetivos operando sobre la información. Estos elementos son componentes físicos (Hardware), los programas (Software), los usuarios (Humanware), algunos autores incluyen en esta organización los datos. Todos estos componentes son importantes y cada uno de ellos juega un papel fundamental para el correcto funcionamiento del sistema, de tal manera que ellos deben complementarse, ser compatibles y evolucionar a la par.



Figura 2. Componentes de un Sistema Informático

En mecatrónica, los dispositivos más usados además de un PC de escritorio son: Microprocesadores, Microcontroladores y PLC's; cada uno con capacidades y potencialidades diferentes, y su uso dependerá del sistema donde se quiera incorporar.

Software

Corresponde a la parte intangible de un computador, la palabra software se refiere a las instrucciones que se incorporan a un sistema informático para que este lleve a cabo una determinada función.

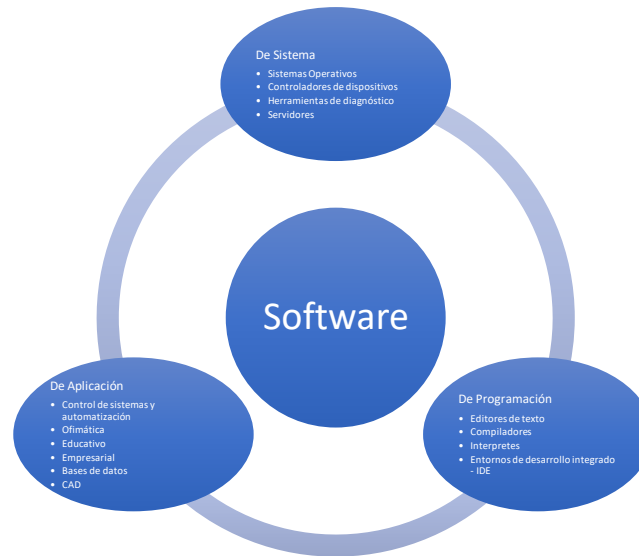


Figura 3. Clasificación de Software

Programación

Es el proceso de diseñar, codificar, depurar y mantener el código fuente de un programa computacional, el cual fue desarrollado en un lenguaje de programación, con el propósito de solucionar o automatizar algún proceso específico. Puede orientarse a un funcionamiento netamente a nivel de software, inclusive en forma multiplataforma, de la misma manera puede implementarse para controlar y/o recibir información de otros programas o dispositivos hardware, para trabajar con los datos recibidos.

PLC (Controlador Lógico Programable)

PLC es el acrónimo de Controlador Lógico Programable (en inglés Programmable Logic Controller). Aparecen debido a la necesidad de dispositivos electrónicos cada vez más pequeños y funcionales, que enlacen las entradas y salidas de un proceso industrial y se

procesen estos datos permitiendo controlar dicho proceso a diferencia de sistemas cableados con una mejor prestancia y velocidades de reacción extremadamente grandes, se ha logrado desarrollar dispositivos electromecánicos programables, utilizados especialmente en industria automática o automatización industrial. Los cuales a diferencia de los pc convencionales que son de propósito general, los PLC están diseñadas para trabajar con múltiples señales de entrada y salida, rangos de temperaturas específicos y ampliados, resistencia a factores como vibración, campos electromagnéticos e inclusive resistencia a impactos. Una característica importante de estos dispositivos es su trabajo en tiempo real, para que los resultados de salida, deben ser producidos como respuesta a las condiciones de entrada, dentro de un tiempo específico y limitado, para que produzca el resultado y comportamiento esperado.

Entre las principales ventajas de trabajar con PLC's tenemos:

- Menor tiempo empleado en su elaboración.
- Realizar modificaciones sin cambiar cableado.
- La lista de materiales y componentes, es muy reducida.
- Se requiere un mínimo espacio de aplicación.
- Menor costo.
- El mantenimiento es económico por tiempos de paro reducidos

Entre las potencialidades de los plcs podemos también agregar las formas de “programarlos” es decir, determinar su funcionamiento y el tratamiento de las señales de entrada y salida. Estas formas son definitivamente computación aplicada tanto en interfaces graficas especiales o formatos de programación con un formato de funciones lógicas.

- Gráfico secuencial de funciones (Grafcet)
- Lista de instrucciones.
- Texto estructurado.
- Diagrama de flujo.
- Diagrama de contactos o Lógica de Escalera o Ladder Logic.

Comunicación y protocolos de Comunicación entre PLC

Las comunicaciones a este nivel deben poseer unas características particulares para responder las necesidades de intercomunicación en tiempo real que deben ser capaces de resistir ambientes hostiles con ruido y electromagnetismo de tal forma que permita relacionar todos los dispositivos para lograr que los procesos industriales cumplan con sus funciones y con una debida intercomunicación entre ellos como se muestra en la figura 4

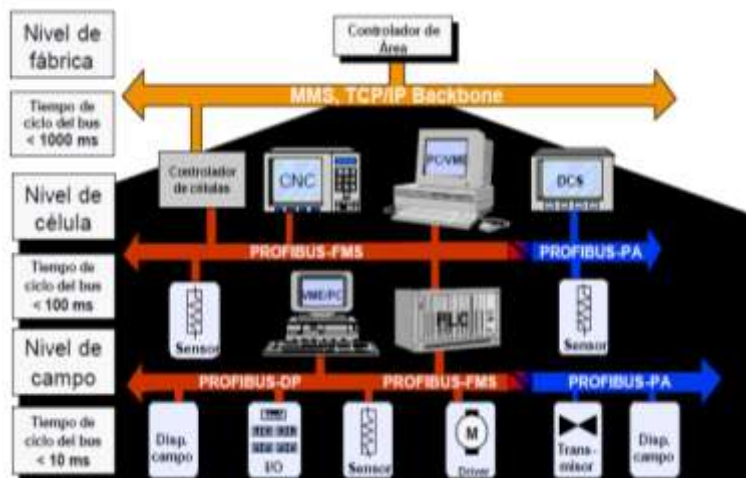


Figura 4. Comunicación y redes entre PLC

Para la comunicación entre cada equipo básicamente trabajan a nivel computacional como se muestra en la figura 5 en la cual se muestran las diferentes capas que permiten esta interconexión.



Figura 5. Ubicación de las capas ISO/OSI de DeviceNet y empleo del protocolo CAN en DeviceNet

III. METODOLOGÍA

Para la realización de este estudio, se aplicó la metodología experimental desde el punto de vista de la docencia universitaria en la carrera de Ingeniería Mecatrónica en la Universidad Técnica del Norte, gracias al análisis de funcionamiento y rendimiento por parte de los estudiantes en la elaboración e implementación de proyectos integradores, entre diferentes asignaturas, en los cuales se puede aprovechar las destrezas adquiridas en varias asignaturas, y para este caso, principalmente con las asignaturas de programación. El cual se complementará con el método de la observación de los procesos realizados en asignaturas de niveles superiores, en las cuales se aplica los principios de programación informática.

Como indican Poikela [10], Barrios-Ananibar [3] y Barrios-Ananibar [4], se ha comprobado que el aprendizaje por parte del estudiante en el desarrollo de nuevas tecnologías, es un proceso que implica la elaboración de muchas prácticas y un correcto entendimiento de los elementos de un programa informático y sus diferentes iteraciones. De la misma manera la necesidad de implementar programas informáticos que satisfagan los requerimientos específicos, utilizando estructuras estáticas de almacenamiento y aprovechando las características de la programación orientada a objetos, en la malla original estaba orientada en la asignatura de programación avanzada, cubrirla durante todo un semestre, sin embargo, debido a que se ha detectado que al implementar los principios de Programación Orientada a Objetos en la programación básica, se puede culminar con el pensum en medio semestre, y dedicar el final del mismo para avanzar en conocimientos de otro lenguaje que será utilizado a nivel superior, como lo es Matlab.

De esta manera al aplicar esta metodología durante siete semestres consecutivos, se ha logrado cubrir las necesidades de programación en los niveles superiores, donde se ha comprobado que los estudiantes tienen una lógica estructurada más firme y conocimientos más sólidos en cuanto a desarrollo de aplicaciones informáticas. Lo cual es una necesidad en el nuevo actual ingeniero, como lo indica Jimenez [8], y mucho más en el ámbito de la mecatrónica.

En base a estos resultados, se ha brindado apoyo en la elaboración de la nueva malla de estudio aprobada en la Reforma de la carrera de Ingeniería en Mecatrónica de la Universidad Técnica del Norte, aprobada por el Consejo de Educación Superior del Ecuador en el 2016, en la cual se implementará las asignaturas de programación para el tercer y cuarto nivel, donde los estudiantes ya cuentan con la lógica básica perfeccionada por medio de la aprobación de las asignaturas básicas de ingeniería. Y de esta manera poder elaborar aplicaciones más avanzadas, debido al grado de conocimiento de los estudiantes a ese nivel. Esta nueva reforma entrará en funcionamiento a finales del 2017. A la par, se ha tomado en cuenta esta aplicación a nivel de la red de Ingeniería en Mecatrónica del Ecuador, para implementarla de igual manera en las Universidades miembros de dicha red.

IV. ANÁLISIS DE RESULTADOS

El conocer claramente el ámbito de desarrollo de la Ingeniería Mecatrónica, nos permite orientar la formación de los futuros ingenieros, desde sus bases, aplicando ejercicios prácticos y fortaleciendo ciertas capacidades en ellos, que serán explotadas más adelante durante su formación académica y luego en su ambiente laboral, lo cual concuerda con el estudio Diagnóstico y Prospectiva de la Mecatrónica en México, elaborado por la Secretaría de Economía (SE) y la Fundación para el desarrollo de las nuevas tecnologías (Funtec) [6] y con el punto de vista de Braxton [1].

Esto ha permitido reestructurar el contenido del syllabus de las asignaturas de programación básica y de programación avanzada, las cuales son las dos asignaturas de desarrollo de programación informática que reciben los estudiantes de Ingeniería Mecatrónica en la Universidad Técnica del Norte, donde se ha desarrollado el presente artículo.

En principio las bases de programación informática, es idéntica en los estudiantes de Ingeniería Mecatrónica con los estudiantes de Ingeniería de Software, ya que los principios de desarrollo de software son universales.

La programación informática al igual que muchos conceptos matemáticos entran muchas veces en el paradigma por parte de los estudiantes e incluso muchos profesionales, que dicen no saber porque reciben esta asignatura, o que nunca han aplicado o utilizado estos conceptos o fórmulas, sin comprender que al cursar estas asignaturas, están fortaleciendo y mejorando su forma de automatizar tareas y enfrentar problemas, que más adelante aplicarán no solo en su vida profesional, sino inclusive en su diario vivir. Tal y como lo afirma Orler [9]

El avance desmesurado de la tecnología, hace que esta controle cada vez más nuestras vidas, de tal manera que se ha vuelto indispensable que el profesional de hoy, accedan a una serie de herramientas para crear algo, desde lo básico a lo complejo.

Esto ha permitido a la industria ofrecer posibilidades de desarrollo de aplicaciones mucho más fáciles orientadas al público en general, quienes sin mayores conocimientos de programación pueden desarrollar aplicaciones muy interesantes. Tal es el caso de los lenguajes de programación visuales intuitivos, tipo Scratch o Kodu, en los cuales su desarrollo es muy distinto a otros lenguajes de programación tipo texto, donde el usuario debe conocer la sintaxis, estructura y forma de escribir, tal es el caso de Logo, C#, Matlab, entre otros.

Mark Guzdial en su libro *Learner-centered design of computing education: Research on Computing for Everyone* [7], afirma que los lenguajes visuales producen una falta de conexión con la profesión informática, o una falta de autenticidad y verdadero conocimiento de lo que se está desarrollando, justamente por la facilidad con la que permiten desarrollar aplicaciones. Y aclara que los lenguajes de programación tipo texto, fomentan habilidades, como creatividad, capacidad de resolución de problemas y trabajo en equipo, gracias al desarrollo del pensamiento computacional, y que se ve reflejado en el estudio de le FUNTEC [6].

Desde este punto de vista, se puede afirmar que la programación textual exige un mayor esfuerzo de abstracción por parte del programador, así como una mayor atención a evitar el

error en el desarrollo de su programa, y por otro lado su código se vuelve más escalable y más fácil de migrarlo a otro lenguaje de programación.

Al compartir este punto de vista, se ha logrado ir adaptando el contenido de las asignaturas de programación para lograr, los resultados esperados, dando una mayor importancia a la programación informática en la carrera de Ingeniería Mecatrónica.

Para la asignatura de Programación Básica, se aporta al perfil profesional del estudiante ya que se fomenta la capacidad para desarrollar aplicaciones de software, la sensibilidad y conocimientos para hacer uso eficiente de las herramientas de programación de alto nivel. En competencias específicas, se promueve el diseñar e implementar estrategias y programas para el control de los dispositivos en los sistemas Mecatrónicos. Diseñando interfaces gráficas a través de lenguajes de programación actuales como Java y un desarrollo a nivel de código y consola. Para lograrlo, se ha implementado proyectos de fin de asignatura que fortalecen estas habilidades, que fortalecen su preparación para la carrera, tal es el caso de:

- Manejo de menú con varias opciones y niveles.
- Simulación de un tablero de leds, a nivel de código y con salida ASCII.
- Elaboración de Chat Bots, sin uso de procesos de inteligencia avanzados.

De la misma manera en la asignatura de Programación Avanzada, se aporta al perfil profesional reforzando mucho más la capacidad para desarrollar aplicaciones de software, así como la sensibilidad y conocimientos para hacer uso eficiente de las herramientas de programación de alto nivel. En competencias específicas, se promueve diseñar e implementar estrategias y programas para el control de los dispositivos en los sistemas Mecatrónicos, diseñar interfaces gráficas que permitan el manejo de datos utilizables, a través de lenguajes de programación actuales como es el caso de C# y Matlab, a nivel de código y consola. De esta manera se ha logrado desarrollar con los estudiantes de esta asignatura aplicaciones como:

- Automatización de procesos de Autocad, como proyecto de fin de nivel, de las asignaturas de Programación Avanzada y Dibujo Mecánico 1.
- Desarrollo de aplicaciones que utilicen una combinación de varios lenguajes de programación, por ejemplo, el uso de Matlab para la realización de cálculos complejos y de C#, aprovechando los pilares de la Programación Orientada a Objetos, para el desarrollo de interfaces de comunicación con el usuario.
- Desarrollo de aplicaciones que simulen el comportamiento de un dispositivo mecatrónico solo a nivel de software, por ejemplo, el desarrollo de un programa que simule el comportamiento de un objeto que pueda evadir obstáculos de forma básica.
- Desarrollo de simuladores de dispositivos electromecánicos como PLC, electro válvulas, motores eléctricos, entre otros.

De esta manera, el estudiante de Ingeniería Mecatrónica, al realizar desarrollos a nivel de texto y consola tiene una visión más clara del conocimiento que podrá explotar más adelante en su carrera, y cuando utilice herramientas de desarrollo gráficas, tendrá un entendimiento más claro de lo que está realizando, similar a lo que dice Riedijk [11] en su postulado.

V. CONCLUSIONES

Las actividades de investigación han contribuido a la actualización del syllabus de las asignaturas de programación básica y avanzada de la carrera de Ingeniería en Mecatrónica de la Universidad Técnica del Norte, de una forma positiva, ya que se ha visto la necesidad de fortalecer su contenido por medio del refuerzo de tópicos de análisis y desarrollo de programas.

El futuro profesional debe conocer el funcionamiento y comportamiento de las instrucciones y rutinas básicas de programación, así como de su correcta aplicación, lo cual será aplicado en los niveles superiores de la carrera en otras asignaturas, e inclusive en el desarrollo de trabajos finales y trabajo profesional.

Si bien es cierto, en la actualidad se puede contar con mucho software de desarrollo de tipo bloque o visual, los cuales permiten crear fácilmente soluciones software, estos a la vez crean

una errada idea de facilidad de desarrollo de programas, ya que automatizan muchos procesos de forma totalmente oculta para el desarrollador, el cual solo se da cuenta de ello cuando quiere personalizar o ahondar un poco más en el desarrollo de la aplicación que está diseñando.

Se ha comprobado que los estudiantes de Programación Básica de Ingeniería Mecatrónica, están al mismo nivel de desarrollo de aplicaciones que los estudiantes de nivel 1 en la carrera de Ingeniería de Software, lo cual ha sido demostrado a nivel de debates académicos elaborados a nivel de Facultad dentro de la Universidad Técnica del Norte. De la misma forma, los estudiantes de Programación Avanzada de Ingeniería Mecatrónica, están en capacidad de tener una visión más clara de la aplicabilidad de estas asignaturas en su futuro académico y profesional.

Los estudiantes de Ingeniería Mecatrónica, no por ser de una carrera diferente a Ingeniería de Software, deben sentirse en menos capacidad que los estudiantes de otras especialidades, al contrario, tienen la capacidad de entablar debates y conversaciones de alto nivel, respecto a principios y fundamentos de programación con estudiantes y profesionales de Desarrollo de Software.

En el aspecto social, el nivel actual con el que cuenta el estudiante de Ingeniería Mecatrónica, respecto a programación informática, ha permitido fortalecer su autoestima personal, ya que por ejemplo, al fortalecer los conocimientos básicos de programación, los estudiantes en los niveles superiores pueden avanzar más rápido en el uso de herramientas de desarrollo informático, lo que permite a la vez que los docentes de las asignaturas posteriores puedan especializarse más en conceptos avanzados.

VI. REFERENCIAS

- [1] Braxton, J. (1996). Contrasting perspectives on the relationship between teaching and research. *New Directions for Institutional Research*.

- [2] Barrios-Aranibar D, Goncalves L Y Alsina P (2008). Learning by Experience and by Imitation in Multi-Robot Systems, *Frontiers in Evolutionary Robotics*, Hitoshi Iba (Ed.), InTech, DOI: 10.5772/5448. Disponible en: https://www.intechopen.com/books/frontiers_in_evolutionary_robotics/learning_by_experience_and_by_imitation_in_multi-robot_systems
- [3] Barrios-Aranibar D. “Estratégias Baseadas em Aprendizado para Coordenação de uma Frota de Ro- bôs em Tarefas Cooperativas”. *Disertación de Maestria*. 2005. Universidade Federal de Rio Grande do Norte, Natal, Brasil. <http://www.ppgeec.ufrn.br/teses.php>.
- [4] Barrios-Aranibar, Dennis ; Goncalves, Luis M. G. ; Alsina, Pablo J. . “Learning by Experience and by Imitation in Multi-Robot Systems”. En: *International Journal of Advanced Robotic Systems*. (Org.). <http://zums.ac.ir/files/research/site/ebooks/Robotics/Evolutionary%20Robotics.pdf>.
- [5] Berbey A, Caballero R, Galán R y Sanz Bobi J. A fuzzy logic inference approach for the estimation of the passengers flow demand. In *Proceedings of the International Conference on Fuzzy Computation and 2nd International Conference on Neural Computation, ICFC 2010 - International Conference on Fuzzy Computation*, pages 125-129, DOI: 10.5220/0003057701250129, Copyright (c) SciTePress, Valencia, Spain.2010.
- [6] FUNTEC, (2013) *Diagnóstico y Prospectiva de la mecatrónica en Mexico*. Disponible en: http://www.economia.gob.mx/files/comunidad_negocios/industria_comercio/Estudios/Diagnostico_Prospectiva_Mecatronica_Mexico.PDF
- [7] Guzdial, M. (2015). *Learner-Centered Design of Computing Education: Research on Computing for Everyone*. *Synthesis Lectures on Human-Centered Informatics*, [online] 8(6), pp.1-165. Available at: <http://www.morganclaypool.com/doi/abs/10.2200/S00684ED1V01Y201511HCI033> [Accessed 29 Mar. 2017].
- [8] Jiménez E, García L., Nuñez E, Bojorquez G, Navarro L, Juarez H, Luna G, Amavizca L. *Experiencias en investigación y docencia en la carrera de Ingeniería*

Mecatrónica en la Universidad la Salle Noroeste. “Twelfth LACCEI Latin American and Caribbean Conference for Engineering to Enhance a Country’s Productivity” July 22-24, 2014, Guayaquil, Ecuador. 2014.

- [9] Orler J. (2012). Docencia- Investigación: ¿Una relación antagónica, inexistente o necesaria? Academia. Revista sobre Enseñanza del Derecho. Año10, Número 19, pp.289-301. Buenos Aires, Argentina (ISSN 1667-4154).2012.
- [10] Poikela, E. & Poikela, S. (1997). Concepts of learning and the implementation of Problem-based learning. Zeitschrift für Hochschuldidaktik, Special Issue. Problem- based learning: theory, practice and research. Editors F. Eitel & W. Gijsselaers. Volume 21.
- [11] Rietdijk, J. (1989). Ten propositions on mechatronics, en Mechatronics in Products and Manufacturing Conference. Inglaterra: Lanceter
- [12] Sotomayor, B. (2003). Grid Computing. Un nuevo paradigma de computación distribuida.
http://www.eside.deusto.es/eventos/semana/eventos/pdf/grid_computing.pdf (25 Jul. 2004)
- [13] Upv.es. (2017). Máster Universitario en Ingeniería Mecatrónica: UPV. [online] Available at: <https://www.upv.es/titulaciones/MUIMECA/> [Accessed 29 Mar. 2017].