



2019 10 al 13 de septiembre - Cartagena de Indias, Colombia

RETOS EN LA FORMACIÓN DE INGENIEROS EN LA ERA DIGITAL

PROGRAMACIÓN CIENTÍFICA: UNA PROPUESTA DIDÁCTICA PARA LA ENSEÑANZA DE MÉTODOS NUMÉRICOS Y PROGRAMACIÓN

Whady Felipe Flórez Escobar, Diego Andrés Flórez Londoño, Raúl Adolfo Valencia Cardona

**Universidad Pontificia Bolivariana
Medellín, Colombia**

Resumen

Los actuales avances en la computación científica ofrecen cada día más oportunidades de modelar y simular toda una variedad de problemas en diferentes campos. Algunos ejemplos se encuentran en la evaluación de materiales en diferentes escalas, predicción de riesgos y desastres, en el entendimiento de patologías y optimización de dispositivos médicos, entre otros. Incluso, varias universidades de Estados Unidos y Europa lideran proyectos en sus currículos para el entrenamiento de la nueva generación de ingenieros expertos en computación científica, que les permita complementar los pilares tradicionales de la teoría y experimentación. En este sentido, la programación científica es una herramienta clave que le permitirá a los futuros ingenieros explorar nuevas teorías o diseñar experimentos computacionales cuando el fenómeno no es observable, riesgoso o demasiado costoso. En el contexto local, existe una dificultad en el estudiante de desarrollar un pensamiento lógico-algorítmico para analizar y descomponer problemas, buscar soluciones mediante el uso de herramientas computacionales e interpretar resultados. La razón posiblemente se deba a la carencia de más espacios de apropiación de metodologías para la implementación computacional de procesos de solución de problemas de ingeniería, tanto en el ciclo de las ciencias básicas, donde aún el estudiante no posee las competencias para enfrentar problemas específicos, como en los ciclos disciplinares, donde la conceptualización y aplicación de los saberes propios de la asignatura ocupan un gran porcentaje de tiempo en las actividades académicas. En el caso de ingeniería mecánica de la Universidad Pontificia Bolivariana, la programación científica se introduce en el curso de Programación y Métodos Numéricos, espacio donde estudiantes y docentes se familiaricen con la programación científica y puedan articular saberes previos en el modelamiento matemático de un problema, técnicas de solución numéricas modernas, la implementación de la solución en un lenguaje de programación o software de

simulación y la visualización y análisis de datos a una variedad de problemas físicos y aplicaciones de ingeniería gobernados por ecuaciones diferenciales. Por lo tanto, es de suma importancia generar competencias en el uso de herramientas computacionales en el proceso formativo del ingeniero UPB con el objetivo de facilitar su desenvolvimiento en el campo laboral.

Palabras clave: programación científica; didácticas de enseñanza; métodos numéricos

Abstract

Recent advances in scientific computing offer every day more opportunities to model and simulate a variety of problems in different fields. Some examples are found in the evaluation of materials at different scales, in the prediction of risks and disasters, in the understanding of pathologies and optimization of medical devices, among others. Indeed, several universities in the United States and Europe are leading projects in their curriculum for training of the next generation of engineers who are experts in scientific computing and for complementing of the traditional pillars of theory and experimentation. In this sense, scientific programming is a key tool that will allow the engineers of the future to explore new theories or to design computational experiments when the phenomenon is not observable, risky or too expensive. In the local context, there is a difficulty for the student to learn logical-algorithmic thinking, to analyze and to break down problems, to find solutions by means of computational tools and to interpret results. The reason may be the lack of more spaces with appropriate methodologies for developing computational methods and solving engineering problems, both in the basic stages, where the student does not have the skills to face specific problems, as in the disciplinary stages, where the conceptualization and application of knowledge of the subject spend a large percentage of time in academic activities. In the case of mechanical engineering department at Universidad Pontificia Bolivariana (UPB university), scientific programming is introduced in the course of Programming and Numerical Methods, a space where students and teachers become familiar with scientific programming and can articulate previous knowledge in the mathematical modeling of a problem, modern numerical solution techniques, the implementation of the solution in a programming language or simulation software and the visualization and analysis of data. A wide variety of physical problems and engineering applications governed by differential equations are introduced in the course. Therefore, it is very important to generate skills in the use of computational tools for the training process of the UPB undergraduate engineers with the aim of facilitating their development in their work field.

Keywords: *scientific programming; didactic teaching; numerical methods*

1. Contexto en el que surgió la experiencia

La experiencia integra dos saberes, Métodos Numéricos y Lenguajes de Programación, con el computador como herramienta de apoyo y en un aprendizaje basado en problemas (ABP). El aula se convierte en un laboratorio computacional que permite fortalecer saberes previos y lograr implementar una representación matemática de un modelo a un código, simulando y validando una variedad de problemas de ingeniería.

La experiencia surgió a partir de un proceso de transformación curricular en la facultad de Ingeniería Mecánica de la UPB, programa creado desde 1956, y donde actualmente cuenta con más de 2000 egresados, alrededor de 350 estudiantes activos y 23 docentes internos (UPB, 2017). Aunque el proceso de transformación ha sido un ejercicio permanente, en el 2011, el proceso se basó en un nuevo Modelo Pedagógico Integrado donde el estudiante es el protagonista del proceso de transformación, permitiéndole realizar procesos de ideación, validar ideas mediante procesos de experimentación y generar productos que impactaran en la sociedad. Adicionalmente, el Proyecto Innovacampus, auspiciado por Ruta N, le dio una dimensión innovadora a la transformación curricular del programa (RutaN, 2015). Curricularmente, en el 2015 culminó el proceso de transformación en la facultad. En la malla anterior a la transformación curricular, se encontraban dos cursos: Métodos Numéricos del Ciclo Básico y Programación para Ingenieros del Ciclo Profesional, ambos con tres créditos. Luego en la malla posterior a la transformación curricular, ambos cursos quedaron integrados en uno solo. El curso actual pertenece al Ciclo Profesional y cuenta con tres créditos. La propuesta de integrar ambos cursos, surge con el fin de articular ambos contenidos en un contexto de curso más general enmarcado en la programación científica. La importancia de esta unión radica que los estudiantes podrán articular saberes previos mejorando la conceptualización de las ciencias básicas (cálculo, física, ecuaciones diferenciales) con el modelamiento matemático de un problema, la implementación de la solución en un lenguaje de programación o software de simulación, la visualización y análisis de datos en una variedad de problemas físicos y aplicaciones de ingeniería.

2. Necesidades que pretende afrontar la experiencia

Los actuales avances en la programación científica ofrecen cada día nuevas posibilidades en las formas de enseñar y hacer matemáticas aplicadas, mediante el modelamiento matemático y simulación computacional de toda una variedad de problemas en diferentes campos (National Science Foundation, 2006), (Lin, 2003). Algunos ejemplos se encuentran en la evaluación de materiales en diferentes escalas, procesos de manufactura y diseño de nuevos productos, predicción de riesgos y desastres para el diseño de planes de emergencias, en el entendimiento de patologías y optimización de dispositivos médicos, entre otros. En este sentido, la programación científica es una herramienta clave que le permitirá a los futuros ingenieros complementar los pilares tradicionales de la teoría y experimentación.

En el contexto de ingenierías UPB, existe una dificultad en el estudiante de desarrollar un pensamiento lógico-algorítmico para analizar problemas, descomponerlos, traducirlos en algoritmos y códigos, buscar soluciones e interpretar resultados, mediante el uso de herramientas computacionales. Este escenario puede deberse a varias situaciones: en el ciclo de ciencia básica, el estudiante todavía no posee las competencias y el conocimiento necesario para enfrentar problemas aplicados y donde conceptos, teoremas y demostraciones pueden quedar en un nivel de abstracción difícil de asimilar y aplicar por los estudiantes; en el ciclo disciplinar, la conceptualización y aplicación de los saberes propios de la asignatura, ocupan un gran porcentaje de tiempo en las actividades presenciales y autónomas, quedando poca oportunidad de utilizar herramientas computacionales. Muchas veces esta situación se puede dificultar aún más por no

tener experiencias previas del uso del computador, colocando la herramienta en un segundo plano para actividades optativas y no como una herramienta de apoyo en su aprendizaje.

En el caso de la ingeniería mecánica, la programación científica se introduce en el curso de "Programación y métodos numéricos" y sus herramientas se emplean en cursos posteriores como Mecanismos, Elementos de Máquinas, Control, Mecánica de fluidos y Transferencia de calor, pero en un nivel básico dado las dificultades mencionadas previamente. Ante la situación de no conformarse con la solución de ejercicios simplificados de libros, se evidenció la necesidad de transformar el salón de clase en un laboratorio para la solución de **problemas aplicados de ingeniería, retos propuestos por la industria y temáticas de proyectos de investigación**, integrando el modelamiento matemático, métodos numéricos y herramientas computacionales, que permitieran un aprendizaje más dinámico con un enfoque teórico-práctico.

Por otro lado, el apoyo de parte de la institución y del programa para dar origen a esta iniciativa ha sido significativo, reflejado en varios aspectos: la formación especializada de los docentes en el área de la programación científica ha permitido que con sus vivencias enriquezcan el proceso de enseñanza, la actualización permanente de salas de cómputo y software, la oportunidad de participar a estudiantes y docentes en congresos y eventos de carácter internacional alrededor de estas temáticas, entre otros.

Adicionalmente, la prueba piloto de integrar Métodos Numéricos y Lenguajes de Programación fue un ejercicio con profesores y egresados, donde se logra evaluar la pertinencia del curso, los saberes previos, su ubicación dentro de la malla curricular y las competencias a desarrollar, lográndose el diseño de una nueva carta descriptiva.

3. ¿Qué pretende lograr la experiencia en relación con el aprendizaje o desarrollo de capacidades humanas o competencias de los estudiantes?

El objetivo desde la perspectiva del estudiante, es la capacidad de desarrollar un pensamiento lógico-algorítmico, con fundamentos en técnicas numéricas y uso de herramientas computacionales, para el modelado matemático y simulación de una variedad de problemas físicos y de ingeniería gobernados por ecuaciones diferenciales, siguiendo una metodología que va desde la formulación matemática del problema y elaboración de algoritmos, hasta la búsqueda de la solución de una manera creativa usando métodos numéricos y lenguajes de programación. Otros elementos que se buscan fomentar son habilidades de investigación mediante el desarrollo de un proyecto de curso, visualizar y analizar datos con sus respectivas unidades para su posterior verificación y validación, comunicar apropiadamente los resultados mediante reportes escritos tipo artículo en Latex, presentaciones orales, wikis, entre otros.

Desde la perspectiva del docente, uno de los objetivos es el poder explicar una serie de conceptos matemáticos y problemas de difícil abstracción en muchos casos, en un tiempo relativamente corto y con una mayor comprensión (aprendizaje significativo), apoyado en los lenguajes de programación como un recurso didáctico.

4. Marco teórico de la experiencia

La ONU, la UNESCO y muchos gobiernos del mundo como la India, USA, Reino Unido, entre otros, han declarado la programación científica como una de las competencias más importantes para la humanidad, hasta el punto de definir el desconocimiento de la programación como analfabetismo algorítmico o computacional (Marin, 2019), (Lomba, 1996). La programación es un proceso, con 3 fases muy importantes: Identificación de problemas y fase de resolución, fase de implementación y fase de mantenimiento. Los tres son importantes para que los ingenieros, científicos y desarrolladores de software puedan resolver problemas. Muchos desarrolladores definirían el proceso de programación sólo con la implementación del algoritmo, lo cual es incorrecto (National Science Foundation, 2006). Traducir un algoritmo en un lenguaje de programación, que luego puede ser compilado o interpretado y finalmente entendido por una computadora, es solo una parte del proceso de programación. En resumen, la Programación es un proceso mediante el cual se resuelve un problema. En el mundo moderno, con la cantidad de problemas y su complejidad aumentando cada vez más, es una herramienta clave para obtener soluciones, cuidadosamente empujando las tareas repetitivas a la computadora y logrando así la "automatización".

Así mismo, las exigencias laborales son cada vez más cambiantes buscando gente más preparada y con habilidades distintas en áreas de la ciencia y la tecnología, donde la robótica y los lenguajes de programación son claves (Portafolio, 2016). En este sentido, algunos países como Estados Unidos, Canadá y Francia vienen adaptando desde hace diez años en sus sistemas educativos, la metodología de enseñanza-aprendizaje STEM: Ciencia (Science), Tecnología (Technology), Ingeniería (Engineering) y Matemáticas (Mathematics). Mediante la combinación de estos cuatro pilares, un profesional podrá utilizar la ciencia para observar y experimentar con problemas del mundo real; con la tecnología podrá manipular equipos y manejar grandes volúmenes de datos; con la ingeniería y las matemáticas podrá estudiar y solucionar problemas con el uso numérico, espacial y lógico.

Por otro lado, la capacidad de pensamiento lógico-algorítmico y su competencia primordial, la programación, tiene dos pasos: en primer lugar, debe analizar el problema o sistema y diseñar una solución (Council, 2000), (Lagoswki, 1990). Este proceso utilizará el razonamiento lógico, la descomposición, la abstracción y los patrones para diseñar algoritmos para resolver el problema o modelar el sistema. En segundo lugar, necesita expresar estas ideas en un lenguaje de programación que comprende instrucciones precisas e inequívocas donde no hay lugar para la interpretación o el debate sobre el significado de una línea particular de código. Este segundo paso incluye la utilización de herramientas de cómputo para cálculos específicos, por ejemplo, macros y complementos de hojas de cálculo, software de análisis de datos, programas de computación simbólica, entre otros.

Adicionalmente, en el mundo actual todos están rodeados de tecnología. La mayoría de las personas usa teléfonos inteligentes y otras computadoras diariamente. Sin embargo, existe una diferencia entre saber cómo usar los programas que se ejecutan en estas tecnologías y comprender cómo funcionan. La comprensión de cómo funcionan les revela más sobre el mundo digital en el que viven. Mirando hacia el futuro, parece inevitable que la cantidad de tecnología y la

dependencia de ella continúen aumentando. Por lo tanto, es importante que la educación de los alumnos les proporcione una comprensión de estos sistemas y, potencialmente, la capacidad de adaptar o desarrollar nuevos sistemas. En palabras de Douglas Rushkoff: "programe o sea programado (Rushkoff, 2011).

Es así como la programación proporciona la motivación para aprender diferentes saberes, entre ellos los métodos numéricos. En este sentido, hay una gran sensación de logro cuando una computadora hace exactamente lo que el usuario le pide y ejecuta el algoritmo numérico de solución. Es posible enseñar el pensamiento computacional sin programar los métodos numéricos y viceversa, pero los dos parecen funcionar mejor juntos. Enseñar el pensamiento computacional sin dar a los alumnos la oportunidad de probar sus estrategias de solución numérica como código en una computadora es como enseñar ciencias sin hacer ningún experimento (CAS, 2019).

Un curso combinado de métodos numéricos y programación, cubre una variedad de temas que incluyen análisis numéricos clásicos (interpolación, integración, álgebra lineal, ecuaciones diferenciales, etc.), procesamiento de señales, tratamiento estadístico de datos, entre otros. La metodología es accesible y tiene un carácter informal en contraste con un curso convencional de métodos numéricos clásico enfocado sólo en las matemáticas. Este enfoque subgéneris ha probado su éxito a nivel mundial y como prueba de ello existen textos reconocidos que utilizan este mismo enfoque, como *Numerical recipes* (William H. Press, 2007) de Cambridge University, que históricamente ha sido el más vendido de todos los tiempos sobre métodos de programación científica.

Finalmente, todo lo anterior va alineado con el Modelo Pedagógico Integrado de la Universidad pues permite que el estudiante tenga una participación más activa en la construcción de su propio conocimiento utilizando el computador como un recurso didáctico y con el profesor como mediador. Además, la combinación de ambos cursos fomenta la integración del currículo pues los contenidos no van por caminos diferentes (UPB, 2009).

5. Metodologías activas de la experiencia

El énfasis de esta experiencia no es solo en el saber de conceptos netamente teóricos, sino que además hay que aprender hacer, para poder luego aplicarlos en el entendimiento y solución de problemas en su vida profesional. También se resalta, como se había mencionado previamente, en la posibilidad de explicar una serie de conceptos matemáticos y problemas aplicados, muchas veces de difícil comprensión para los estudiantes, apoyado en los lenguajes de programación como un recurso didáctico.

El curso se basa en una de las derivaciones del constructivismo, en un aprendizaje basado en problemas (ABP) donde se promueve la exploración libre del estudiante en el aula de clase visto como un laboratorio computacional. Entre las actividades que se desarrollan durante el semestre se tienen actividades tanto grupales como de trabajo autónomo. Algunas de ellas son: clases teórica-prácticas con solución de casos, resolución de talleres, tareas extraclases debidamente orientados que promuevan la construcción del conocimiento, incentivar la participación activa en clase,

participación en charlas técnicas y defensas de tesis de los grupos de investigación y en el desarrollo de un proyecto de curso. Algunas de estas actividades son formativas que le permitirán ir afianzando los conceptos vistos en clase. Otras son evaluativas y son incluidas en el seguimiento del curso: quices, evaluaciones, tareas extraclases y proyecto de curso. En el caso de los quices, algunos son presenciales y otros son virtuales, cuyo desarrollo se hace a través de la plataforma digital Moodle. Además, el curso cuenta con un espacio en el aula digital donde encuentra allí todo el material trabajado durante el semestre y donde podrán subir las diferentes actividades evaluativas. Otros elementos que también se motivan dentro del desarrollo del curso, es la responsabilidad, la puntualidad, el respeto y la disciplina. Adicionalmente, los estudiantes cuentan con un foro para preguntas y con el acompañamiento de un monitor que normalmente es un estudiante que ya vio el curso en el semestre anterior y que tuvo un buen desempeño académico.

En cuanto al proyecto del curso, que busca incentivar la actividad investigativa en los estudiantes, al comienzo del semestre se conforman una serie de equipos de trabajo y se les plantea una serie de problemas aplicados a la ingeniería (**temáticas de proyectos de investigación, retos de la industria, problemas clásicos de ingeniería, entre otros**) para ser resueltos en varias etapas: formulación del problema a resolver en el cual deben hacer una búsqueda sistematizada en diferentes fuentes como artículos de bases de datos, capítulos de libros, memorias de congresos, cursos online, charlas con expertos, entre otros. Una de las bases de datos más utilizadas en el curso es ScienceDirect. Luego deben elaborar un marco teórico-conceptual donde se contextualice el problema y se analizan las ecuaciones diferenciales gobernantes que rigen, indicando simplificaciones al modelo matemático seleccionado y describiendo características relevantes como condiciones de frontera e iniciales, dominio de estudio, dimensiones, propiedades de los materiales y fluidos, entre otros. Posterior a esta etapa, se comienza la implementación de la solución mediante la selección de métodos numéricos, el esbozo de algoritmos y la codificación en un lenguaje de programación. Allí se deben simular una serie de escenarios y realizar un análisis de sensibilidad de algunos parámetros con las variables de respuesta, que les permita hacer un análisis comprensivo y discusión de los resultados del caso de estudio. Posteriormente, se hace una presentación a todo el grupo bajo la metodología de "elevator pitch", con un espacio para preguntas y donde se cuenta con la presencia de invitados externos, egresados y docentes internos de ingeniería, que evalúan la entrega final. Por último, se elabora un informe tipo artículo o un poster de alta calidad tipográfica en un editor Latex, aprovechando que el software funciona también con código.

Durante el desarrollo del curso también se han utilizado metodologías apoyadas en la web o cuatro modos diferentes de impartir instrucción, a saber:

- 1) Conferencia tradicional,
- 2) Conferencia mejorada en la web,
- 3) Auto-estudio basado en web, y
- 4) Auto-estudio basado en la web, discusión y análisis de diversas simulaciones.

6. ¿Por qué se considera relevante dar a conocer esta experiencia?

La experiencia de integrar los dos saberes, Métodos Numéricos y Lenguajes de Programación, ha permitido a los estudiantes no solo entender toda una serie de conceptos matemáticos alrededor de los Métodos Numéricos, sino el poder manejar un número mayor de datos, realizar operaciones complejas de cálculo y visualizar diferentes formas para llegar a la solución de un problema en un tiempo menor, con el uso de herramientas computacionales. Adicionalmente, se ha visto una participación mucho más activa y creativa en el aula de clase por parte de los estudiantes.

Otra parte fundamental de la experiencia es la introducción de una variedad de problemas aplicados basados en ecuaciones diferenciales, tanto ordinarias como parciales, que parten de temáticas de proyectos de investigación, retos de la industria y problemas clásicos de ingeniería. En este sentido, el curso podría beneficiar no solo a estudiantes de pregrado sino de posgrado para que tengan la oportunidad de integrar saberes previos e involucrar las competencias adquiridas en el desarrollo de trabajos de investigación, en la toma de decisiones y pensamiento crítico para orientar el componente experimental y en la posibilidad de ampliar el espectro de su vida laboral.

Otro aporte del curso es en el uso de tecnologías audiovisuales y computacionales (visualización científica, consulta de fuentes de internet, consulta de algoritmos, etc). Más aún, el curso trasciende dichas tecnologías pues empodera al estudiante para crear sus propias herramientas de visualización, solución y análisis; además de la obvia ventaja del curso en mostrar como conceptos matemáticos y físicos pueden ser aplicados.

También debe darse a conocer porque es basada en modelos internacionales; para hacer bien las cosas hay que mirar hacia afuera al resto de la comunidad académica y científica para determinar si las prácticas que se diseñan originalmente, si concuerdan con la dinámica académica mundial y las prácticas correctas globales. También debe darse a conocer por la importancia de la computación en las ingenierías. Por otro lado, le aporta al Modelo Pedagógico Integrado de la UPB, las competencias computacionales declaradas por la Unesco como vitales para la sociedad presente y futura.

Finalmente, todavía quedan retos por trabajar: por ejemplo, se pone en evidencia saberes previos desarticulados o que aún hay vacíos en algunas competencias. También otro reto a trabajar es en la motivación, tanto a estudiantes como docentes, a darle continuidad al uso de estos saberes en cursos superiores de la malla curricular.

Referencias

- RutaN. (2015). *Reto Innovacampus: La Transformación Curricular como detonante del cambio*. Medellín.
- National Science Foundation. (2006). *Report of the Blue Ribbon Panel on Simulation-Based Engineering Science*. Obtenido de Revolutionizing Engineering Science through Simulation: https://www.nsf.gov/pubs/reports/sbes_final_report.pdf

- UPB. (2017). *Proyecto Educativo del Programa*. Documento institucional, Escuela de Ingenierías. Ingeniería Mecánica. Universidad Pontificia Bolivariana, Medellín.
- Marin, J. (2019). Recuperado el 2019, de El analfabetismo tecnológico: http://www.iar.unicamp.br/lab/luz/ld/Linguagem%20Visual/el_analfabetismo_tecnologico.pdf
- Lomba, J. B. (1996). *Sociedad digital del homo sapiens al homo digitalis*. Madrid, España: Alianza.
- *Portafolio*. (2016). Recuperado el 2019, de STEM Education for the Future (Educación STEM para el futuro): <https://www.portafolio.co/economia/stem-education-for-the-future-educacion-stem-futuro-492939>
- Rushkoff, D. (2011). *Program or Be Programmed: Ten Commands for a Digital Age*. New York, USA: OR Books.
- CAS. (2019). Recuperado el 2019, de Case barefoot program. Computing at School (CAS): <https://www.barefootcomputing.org/about-barefoot>
- William H. Press, S. A. (2007). *Numerical Recipes: The Art of Scientific Computing* (3 ed.). New York, USA: Cambridge University Press.
- UPB. (2009). Recuperado el 2019, de Modelo pedagógico integrado. Vicerrectoría Académica. Universidad Pontificia Bolivariana: http://cmap.upb.edu.co/rid=1MZRMJ8L6-V9T7JP-2L5/Modelo_Pedagogico-integrado-upb.pdf
- Lin, J. F. (2003). Individual differences in learning entrepreneurship and their implications for web-based instruction in e-business and e-commerce. *British Journal of Educational Technology*, 34, 455-465.
- Council, N. R. (2000). *How People Learn: Brain, Mind, Experience, and School*. Washington, DC: The National Academies Press.
- Lajoswski, J. (1990). Retention rates for Student Learning. *Journal of Chemical Engineering*, 67, 811.

Sobre los autores

- **Whady Felipe Flórez Escobar**: docente investigador de la Facultad de Ingeniería Mecánica. Doctor de Wessex Institute Of Technology University Of Wales. Líder del Grupo de Investigación Energía y Termodinámica (GET). Áreas de trabajo: Modelación Matemática y Simulación, Mecánica de Fluidos Computacional, Matemáticas Aplicadas.
- **Diego Andrés Flórez Londoño**. Director del programa y docente de la Facultad de Ingeniería Mecánica. Magister en Gestión Tecnológica. Área de trabajo: Diseño Mecánico.
- **Raúl A. Valencia**: Docente titular de la Facultad de Ingeniería Nanotecnología y Mecánica. Líder del Grupo Automática y Diseño. Especialista en Diseño Mecánico y Maestría en Ingeniería de la Universidad Eafit. Doctor en Ingeniería con énfasis en Computación Científica de la Universidad de Eafit. Áreas de trabajo: Modelación Matemática y Simulación, Mecánica Computacional, manufactura digital, ingeniería inversa y procesamiento de imágenes.

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.

Copyright © 2019 Asociación Colombiana de Facultades de Ingeniería (ACOFI)