

Nombre y código de la asignatura			Algoritmia y Programación (IS313)				
Área académica			Ciencias Básicas				
Semestre	Créditos	Requisitos	Horas presenciales (HP)			Horas de trabajo independiente	Total de horas
			Teóricas	Prácticas	HP Totales		
3	3	CB223	32	48	80	64	144

Año de actualización de la asignatura: 2019

1. Breve descripción

Esta asignatura está incluida en el grupo de asignaturas profesionales en el área de ciencias básicas. El Ingeniero de hoy, dentro de su campo de trabajo, se ve obligado a manejar, en forma directa o indirecta, su información a través de un computador. Por ello es importante que dentro de su vida universitaria exista un espacio donde adquiera las bases sobre informática incluyendo los conceptos básicos de algoritmos, procesos computacionales y programación. Con esto se refuerza la capacidad y habilidad para entender problemas en general y estructurar soluciones para los mismos.

2. Objetivos

Adquirir los conocimientos básicos necesarios para utilizar y programar en diferentes lenguajes de programación y tarjeta de desarrollo de Arduino, además de su entorno de desarrollo integrado y electrónica requeridos.

Correspondencia con los objetivos del programa:

Preparar profesionales con una sólida formación en los conceptos, en la lógica, en los métodos y la teoría disciplinaria y profesional

Formar profesionales que actúen integralmente, tanto en su ejercicio profesional como en su tiempo libre, y que busquen desarrollar su potencial

Formar profesionales con las competencias genéricas y disciplinares, de tal manera que éstas le permitan desempeñarse con idoneidad en diferentes contextos, bajo principios éticos y morales, con compromiso y responsabilidad económica, social y ambiental, promoviendo el desarrollo sostenible del país y de la comunidad

3. Resultados de aprendizaje de asignatura

1. Conocer y aplicar la terminología utilizada en el área de procesamiento electrónico de datos, en la creación de algoritmos, diagramas y pseudocódigo, para la solución de problemas de forma estructurada.
2. Aplicar características básicas y fundamentales de un lenguaje de programación, en la solución de problemas mediante la aplicación de pensamiento lógico.
3. Implementar y utilizar herramientas de programación existentes, que permitan plasmar los conceptos de la programación funcional, en aplicaciones ingenieriles que requieran solución de problemas numéricos.

Otras competencias por formar:

1. Demostrar la creatividad a través de la solución de problemas de ingeniería.
2. Aprender y trabajar de forma autónoma

4. Contenido

Capítulo I. Introducción a la Programación ^{[3] [6] [10]} (~10 horas)

Terminología básica en informática. Componentes de un computador digital. Diferentes tipos de Software: Sistemas Operativos, lenguajes de Programación, Paquetes, entre otros

Capítulo II. Lenguaje de Programación C Y C++ ^{[4] [8] [9]} (~15 horas)

Elementos de lenguaje. Diferencia entre C y C++. Caracteres en C. Tipos de datos. Declaración de constantes y variables. Manejo y herramientas del software de simulación. Librerías. Funciones. Ciclos. Estructuras de control. Ejercicios y ejemplos de simulación.

Capítulo III. Lenguaje de Programación Python ^{[7] [11]} (~15 horas)

Elementos de lenguaje. Diferencia entre Python y C++. Caracteres en Python. Tipos de datos. Declaración de constantes y variables. Manejo y herramientas del software de simulación. Librerías. Funciones. Ciclos. Estructuras de control. Ejercicios y ejemplos de simulación.

Capítulo IV. Matlab ^{[1] [2]} (~20 horas)

Elementos de lenguaje. Caracteres en MatLab Tipos de datos. Declaración de constantes y variables. Manejo y herramientas del software de simulación. Librerías. Funciones. Ciclos. Estructuras de control. Ejercicios y ejemplos de simulación. Entornos gráficos (GUI). Simulink.

Capítulo V. IDE Arduino ^[5]^[9] (~20 horas)

Descripción del IDE. Instalación en Windows. Entorno y capacidades. Comunicación con y desde Arduino. Despliegue de información con Arduino. Elementos básicos. Componentes habituales. Sensores de fenómenos físicos. Sensores digitales y analógicos. Medición de magnitudes principales. Sensores de temperatura. Sensores de luz. Sensores de movimiento y proximidad. Acople de sensores con Arduino. Protecciones necesarias. Funcionamiento de interruptores y pulsadores. Comunicación con MatLab.

Nota: las horas son aproximadas y no contemplan el tiempo destinado a las evaluaciones

5. Recursos

Recursos:

Biblioteca, Internet, recursos audiovisuales y multimedia.

Bibliografía:

1. Acedo S., José A, Instrumentación y control básico de procesos. Diaz de Santos, S. A., Madrid, 2013
2. Jorquera, H. y Gelmi Weston, C. A., Métodos numéricos aplicados a ingeniería: casos de estudio en ingeniería de procesos usando MATLAB. Pontificia Universidad Católica de Chile, 2014.
3. Kelsey, R., Clinger, W, Rees, J. y otros: “Revised⁶ Report on the Algorithmic Language Scheme”, 2009. <https://users.cs.northwestern.edu/~robby/pubs/papers/jfp2009-sdfsfmkcr.pdf>
4. Marcelo Villalobos, R. W., Fundamentos de programación C++ : Más de 100 algoritmos codificados. Editorial Macro, 2014.
5. McRobert, M. s, Beginning Arduino. Apress, 2013.
6. Matthias, Felleisen, Findler Robert Bruce, FLATT Matthew, KRISHNAMURTHI Shriram, “How to Design Programs An Introduction to Computing and Programming”, The MIT Press, Cambridge, Massachusetts. London, England Last modified: Wednesday, February 23th, 2019 US/Eastern, tomado de internet en: <https://htdp.org//2019-02-24/>, el 24 de Octubre de 2006.
7. Langtangen, H. P., A Primer on Scientific Programming with Python, vol. 6. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.
8. Pitt-Francis, J. and Whiteley, J., Guide to Scientific Computing in C++ (Undergraduate Topics in Computer Science). 2012.
9. Purdum, J., Beginning C for Arduino. Apress, 2015.
10. Trejos Buriticá, O. I., Lógica de programación, 1ra ed. Ediciones de la U, 2017.
11. Zhang, Y., An introduction to python and computer programming, vol. 353. 2015.

6. Actividades

- Clases magistrales, interactuando con el estudiante, haciéndolo partícipe de su proceso de formación.
- Solución de problemas enfocados a afianzar los conceptos y a desarrollar habilidades analíticas.
- Desarrollo dirigido de talleres en forma individual o en grupo.
- Planteamiento, análisis y solución de problemas en forma dirigida.

7. Trabajos en laboratorio y proyectos

No se realizan trabajos en laboratorio

8. Métodos de enseñanza-aprendizaje

Se usará la metodología basada en competencias, donde su énfasis se concentrará en la aplicación de la teoría, las ecuaciones y la asimilación de los conceptos estudiados, con el fin de lograr en el estudiante la adecuada adquisición de conocimiento, el desarrollo de habilidades y la solidificación de hábitos de trabajo colaborativo.

9. Evaluación

- Evidencia de conocimiento: tareas (65%), con las que se evalúe la idoneidad con la cual se ejecutan las competencias del proyecto formativo.
- Evidencias de producto: Proceso de seguimiento de la formación (no superior al 25 %): se busca aplicar los siguientes criterios con el fin de promover la evaluación continua: a) realización de procesos consulta e investigación sobre temas complementarios a la formación, b) desarrollo de problemas tipo, c) entrega de reportes e informes sobre temas tratados en clase, d) quices y evaluaciones orales.
- Trabajo Final (35%) que involucre una aplicación de la simulación computacional en un sistema mecánico según el área de profundización tratada. Evidencias de desempeño: realizar una presentación y discusión sobre temas complementarios a la formación.
- Estas evaluaciones estarán diseñadas teniendo en cuenta las competencias, los criterios de desempeño, el rango de aplicación y los saberes esenciales.