

Nombre y código de la asignatura		Algoritmia y Programación - IM313					
Área académica		Ciencias Básicas					
Semestre	Créditos	Requisitos	Horas presenciales (HP)			Horas de trabajo independiente	Total de horas
			Teóricas	Prácticas	HP Totales		
1	3	Ninguno	2	3	5	4	144

Año de actualización de la asignatura: 2023

1. Breve descripción

En este curso se estudian los conceptos básicos de algoritmos, procesos computacionales y programación, con el propósito de reforzar las capacidades y habilidades necesarias, entendiendo los problemas en un contexto en general y estructurando las soluciones basándose en los requerimientos del caso.

2. Objetivo general

Adquirir los conocimientos básicos de la lógica de programación, Utilizar herramientas y metodologías de programación básicas en los diferentes lenguajes de programación. Introducir al alumno en la programación estructurada y orientada objetos.

3. Resultados de aprendizaje de asignatura

El estudiante:

1. Aplica características básicas y fundamentales de un lenguaje de programación, en la solución de problemas mediante la aplicación de pensamiento lógico.
2. Conoce y aplica la terminología utilizada en la creación de algoritmos y diagramas, para la solución de problemas de forma estructurada.
3. Implementa y utiliza herramientas de programación existentes, que permitan plasmar los conceptos de la programación funcional, en aplicaciones ingenieriles que requieran solución de problemas mediante la ejecución de metodologías recursivas e iterativas.
4. Aprende y trabaja de forma autónoma y colaborativa.

4. Contenido

I. INTRODUCCIÓN A LA PROGRAMACIÓN ^[3] ^[6] ^[10] (~5 horas)

Terminología básica en informática. Comprensión global de un problema de cómputo (análisis y estructuración). Componentes de un computador digital. Diferentes tipos de Software: Sistemas Operativos, lenguajes de Programación, Paquetes, entre otros.

II. REPRESENTACIÓN DE DATOS Y OPERADORES (~15 hora)

Análisis, estructuración y solución de problemas. Constantes y tipos de constantes. Variables y tipos de variables. Conversión de tipos. Operadores de asignación, aritméticos, relacionales, lógicos, condicionales y de precedencia.

III. LENGUAJE DE PROGRAMACIÓN PYTHON ^[7] ^[11] (~15 horas)

Elementos de lenguaje. 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. Comparación de los lenguajes de programación.

IV. ESTRUCTURAS DE DECISIÓN, SELECCIÓN E ITERACIÓN (~20 horas)

Condicionales múltiples. Anidación de condicionales. Estructuras iterativas. Bucles anidados. Manejo de archivos. Vectores y matrices como listas y tablas. Aplicaciones ingenieriles.

V. PROGRAMACIÓN ESTRUCTURADA Y MODULAR (~20 horas)

Definición y aplicaciones. Funciones, abstracción de problemas, profundización del concepto divide y vencerás. Modularización y manejo de librerías externas.

VI. INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS (~5 horas)

Definición. Atributos y comportamiento. Código base del programa.

5 Recursos

Biblioteca, Internet, recursos audiovisuales y multimedia.
Lenguajes de programación: C, C++, Python, Lenguaje M.

Bibliografía:

1. Acedo S., J. A. Instrumentación y control básico de procesos. Diaz de Santos, S. A., Madrid, 2013.
2. Jorquera, H., Weston, C. A., Métodos numéricos aplicados a ingeniería: casos de estudio en ingeniería de procesos usando MATLAB. Alfa Omega. Santiago 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. Walter, R., Villalobos, M. Fundamentos de programación C++ : Más de 100 algoritmos codificados. Editorial Macro, 2014.
5. McRobert, M. s, Beginning Arduino. Apress, 2013.
6. Felleisen, M., Findler, R. B., Flatt, M., Krishnamurthi, S. 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., Whiteley, J. Guide to Scientific Computing in C++ (Undergraduate Topics in Computer Science). Londres: Springer 2012.
9. Purdum, J., Beginning C for Arduino. Apress, 2015.
10. Trejos, O. I., Lógica de programación, 1ra ed. Ecoe Ediciones, 2017.
11. Zhang, Y. An introduction to python and computer programming, vol. 353. Singapur: Springer, 2015.

6. Metodología

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.
Trabajo independiente del alumno con asesoría personalizada del profesor.

7. 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.