

Código de asignatura: IE313

Nombre del programa académico	Ingeniería Eléctrica
Nombre completo de la asignatura	Algoritmia y Programación
Area académica o categoría	Profesionales y específicas
Semestre y año de actualización	Semestre 1 – 2022
Semestre y año en que se imparte	Semestre 3 – Año 2
Tipo de asignatura	[x] Obligatoria [] Electiva
Número de créditos ECTS	5
Director o contacto del programa	José Germán López Quintero
Coordinador o contacto de la asignatura	Germán Andrés Holguín Londoño

Descripción y contenidos

1. Breve descripción La asignatura de Algoritmia y Programación es de naturaleza teórica, el propósito es el análisis, estructuración, diseño e implementación de soluciones a problemas de cómputo para aplicaciones de ingeniería eléctrica, con base en lenguajes de programación estandarizados.
2. Objetivos Al finalizar este curso se espera que el estudiante esté en capacidad de identificar y comprender el léxico propio de la programación, Comprender las arquitecturas básicas de un computador digital, Captar la estructura básica de un programa de computador, Conocer los diferentes paradigmas clásicos y modernos de la programación. Analizar y diseñar soluciones a problemas computacionales. Lo anterior se corresponde con el objetivo del programa (OP-2).
3. Resultados de aprendizaje - RA1: Identificar el léxico propio de la programación - RA2: Distinguir entre las diferentes arquitecturas básicas de un computador digital. RA3: Identificar la estructura de un programa de computador digital. - RA4: Aplicar técnicas para la solución de un problema de cómputo. RA5: Analizar soluciones a problemas de cómputo. RA6: Capacidad de trabajo en equipo - RA7: Capacidad de abstracción análisis y síntesis RA8: Capacidad para tomar decisiones - RA9: Habilidad para buscar, procesar y analizar información procedente de fuentes diversas. RA10: Capacidad para identificar y resolver problemas. - RA11: Capacidad para comprender literatura en idioma inglés sobre algoritmia y programación. - Los anteriores se corresponden con los siguientes resultados de aprendizaje del programa: RAP-7, RAP-12, RAP-14
4. Contenido - T1: Historia de la informática (2 h). - T2: Análisis y estructuración de problemas de cómputo (4h). - T3: Programa de cómputo y algoritmos (6 h). - T4: Representación de datos en sistema de procesamiento (4 h) - T5: Programación estructurada (12 h) - T6: Arreglos y punteros (8 h) - T7: Programación modular (4h) - T8: Estructuras (4h) - T9: Introducción a la programación orientada a objetos (POO) (8h) - T10: Herencia y polimorfismo (4h) - T11: Manejo de memoria y tópicos especiales (4h)
5. Requisitos - Asignaturas: Álgebra lineal (código CB223). - Competencias blandas: Capacidad de abstracción, análisis y síntesis, (Pensamiento crítico). Capacidad para identificar, planear y resolver problemas. Capacidad para tomar decisiones. Habilidades para buscar, procesar y analizar información procedente de fuentes diversas.

6. Recursos

Libros de texto:

- C. HORSTMANN, R. NECAISE. *Python for Everyone*. Wiley 3 edición. 2018.
- SANDEEP NAGAR. *Introduction to Python for Engineers and Scientists*. Apress. 2018.
- ROB MASTRODOMENICO. *The Python Book*. John Wiley & Sons Ltd. 2022.
- SEBASTIEN CHAZALLET. *Python 3 – Los fundamentos del lenguaje*. 3ra Edición. Editorial Eni. 2020.

Herramientas informáticas

- Software de desarrollo: PYTHON.
- Interfaces de desarrollo integradas: Pycharm version community, Google Colab, Visual Studio Community para Python.

Recursos de internet:

- <https://www.ieee.org/>
- <https://www.learnpython.org/>

7. Herramientas técnicas de soporte para la enseñanza

- Talleres en clase opcionales enfocados al desarrollo del pensamiento lógico programativo y algoritmia.
- Trabajos prácticos opcionales enfocados a la resolución de problemas de cómputo.
- Talleres de análisis y estructuración de problemas de cómputo. Otras herramientas se presentan en 6.

8. Trabajos en laboratorio y proyectos

- Proyecto al finalizar la asignatura implementado en Python (8 h).
- Trabajos prácticos de programación en Python (8h).

9. Métodos de aprendizaje

- Cátedra magistral. Se efectúa planteamiento y debates sobre problemas y diseños propuestos.
- Aula extendida. Se dejan temáticas específicas para ser estudiadas y profundizadas por los estudiantes a través del trabajo individual.
- Aprendizaje basado en problemas. Se presentan problemas reales de aplicación al diseño de automatismos.
- Trabajos colaborativos. Se desarrollan actividades independientes, personalizadas y grupales en forma de trabajos prácticos. Existe proyecto al finalizar la asignatura.
- Tutorías.
- Ejercicios de programación planteados durante las sesiones de clase.

10. Métodos de evaluación

Para la obtención de la nota definitiva se realizan diferentes pruebas escritas individuales en el aula durante el semestre, de las cuales están previstas:

- Examen 1: Historia de la informática (T1), Análisis y estructuración de problemas de cómputo (T2), Programa de cómputo y algoritmos (T3), Representación de datos en sistema de procesamiento (T4). (30%) (RA1, RA2, RA3, RA4).
- Examen 2: Programación estructurada (T5), Arreglos y punteros (T6), Programación modular (T7), Estructuras (T8). (30%) (RA3, RA4, RA5, RA8).
- Examen 3: Introducción a la programación orientada a objetos (POO) (T9), Herencia y polimorfismo (T10), Manejo de memoria y tópicos especiales (T11). (30%) (RA3, RA4, RA5, RA7, RA8, RA9, RA10). Y un proyecto grupal, donde se realiza el diseño y programación de un aplicativo simple con programación orientada a objetos y el uso de estructuras de datos.
- El trabajo debe ser presentado en la fecha establecida, no se admiten entregas posteriores a la fecha establecida. El trabajo debe ser sustentado y su exposición hará parte de la evaluación del trabajo.
- En esta asignatura se evalúa la competencia transversal de trabajo en equipo (RA6) y la competencia transversal de Análisis, evaluación y uso de la información para la solución de problemas (RA7, RA9, RA10). La evaluación se realiza en el proyecto y tiene un peso de 10%.