

Fecha: enero del 2026

**PROGRAMA ACADÉMICO:** INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

**SEMESTRE:** PRIMERO

**ASIGNATURA:** ALGORITMOS Y PROGRAMACIÓN

**CÓDIGO:** 8107550

**NÚMERO DE CRÉDITOS:** 4

### PRESENTACIÓN

Esta asignatura se orienta a la formación profesional del Ingeniero de Sistemas y Computación, teniendo como punto de partida los diferentes perfiles en los cuales se desempeña el profesional de Ingeniería y los diferentes campos de acción de la Ingeniería de Sistemas y Computación para tener una idea precisa de los temas que se desarrollarán teniendo en cuenta el plan de estudios del programa académico y que hace parte fundamental de la formación del estudiante para aplicarlos en cada una de las asignaturas y de su mismo desempeño académico y laboral.

Durante el desarrollo de la asignatura se debe conocer detalladamente la representación de la información en sistemas digitales, para luego enfocarse en el desarrollo de la lógica necesaria para aplicarla en la solución de problemas. A continuación, se hará una introducción a metodologías y herramientas que permitan representar soluciones a problemáticas reales (por ejemplo, algoritmos y representación mediante objetos).

En la segunda parte del curso se abordarán temáticas relacionadas con el proceso de desarrollo de software, especialmente la etapa de codificación, es decir los componentes que intervienen en el proceso de programación. Cabe aclarar que, aunque se utilizará como base el lenguaje de programación Java, se dará mayor relevancia a los conceptos fundamentales de programación dejando en un segundo plano las herramientas utilizadas.

### JUSTIFICACIÓN

En el mundo laboral, se requieren profesionales capaces de afrontar situaciones reales y dinámicas, que estén focalizados en la gestión y desarrollo de las mejoras continuas, por ello esta asignatura contribuye a formar las bases fundamentales, del futuro Ingeniero de Sistemas y Computación, que esté en la capacidad de aplicar los conocimientos adquiridos en la construcción de soluciones a los problemas presentados, haciendo uso de diferentes herramientas computacionales.

Luego de analizar el ámbito internacional es evidente que una de las habilidades más demandadas por las compañías de Tecnología Informática, es la de desarrollar aplicaciones, según la "Encuesta de Salarios IT 2017", realizada por el *Portal Computer World*; en el campo nacional el panorama es similar teniendo en cuenta el "Estudio de salarios y profesionales del sector de Software y TI de Colombia 2016", elaborado por MinTIC y FEDESOFTE, donde se presentan los perfiles más solicitados en la industria nacional, destacándose los relacionados con el área de desarrollo, lo que es corroborado con otros estudios que señalan el déficit de profesionales con estos perfiles, por ello la estructura de esta asignatura contribuye de manera efectiva a responder a las necesidades de esta industria.

Por otra parte, es justo mencionar que las habilidades que se desarrollan en esta asignatura, sirven como base para el abordaje de asignaturas de otras áreas que hacen parte del plan de estudios del Programa.

Por último, es importante resaltar que la lógica de programación actualmente se considera una habilidad transversal, debido a que contribuye al desarrollo del pensamiento lógico y matemático, que se requiere en cualquier campo de la Ingeniería. Existen, además, comunidades académicas que han identificado la necesidad de incluir temas de lógica de programación, desde tempranas edades en los primeros niveles de escolaridad.

### **COMPETENCIAS**

Adquirir una visión global de la Ingeniería de sistemas en el campo laboral y construir modelos de solución algorítmicas focalizadas en las bases de la lógica y la programación, con el fin de construir soluciones reales, viables y de eficaz implementación en un lenguaje de programación.

- Adopta una cultura de la actualización permanente teniendo en cuenta los avances tecnológicos
- Desarrolla la habilidad de abstraer los elementos y relaciones de un sistema real a una representación formal
- Diseña algoritmos a partir del análisis de problemas que ameritan una solución informática
- Utiliza los elementos de los lenguajes de programación orientada a objetos
- Identifica y aplica las estructuras básicas de programación para implementarlas en cualquier lenguaje
- Conoce y aplica los procesos de codificación y compilación
- Utiliza herramientas de software para soportar los procesos de las etapas del ciclo de vida del desarrollo de software
- Utiliza de forma adecuada las estructuras de datos básicas para dar solución a problemas simples
- Conoce y aplica los principios de única responsabilidad y separación de conceptos
- Trabaja en equipo para dar solución a problemas planteados
- Genera código de calidad aplicando buenas prácticas de programación

### **RESULTADOS DE APRENDIZAJE**

- Identifico las etapas básicas del ciclo de vida del desarrollo de software (análisis, diseño y construcción) que se deben aplicar para obtener un producto de software en diferentes contextos de aplicación y la forma como estas etapas pueden ser soportadas con herramientas especializadas
- Conozco conceptos fundamentales (clase, objeto) y características de la programación orientada a objetos como la abstracción, única responsabilidad y separación de conceptos.
- Aplico las estructuras básicas de programación en el marco del paradigma de orientación a objetos, a partir del trabajo en equipo y aplicando buenas prácticas de codificación.
- Adopto una cultura de la actualización permanente teniendo en cuenta los avances tecnológicos

### **METODOLOGÍA**

Esta asignatura será guiada en los espacios presenciales por la complementación conceptual del docente al trabajo de preparación previo, que los estudiantes han realizado sobre la temática particular a tratar en la sesión; por lo tanto, un tema será abordado en cuatro momentos:

1. Preparación, consulta e investigación conceptual por cuenta del estudiante y su grupo de trabajo.

2. Tratamiento conceptual del tema en sesión del gran grupo junto con el docente.
3. Aplicación de talleres individuales y cooperativos a nivel tutorial.
4. Desarrollo de actividades de refuerzo en sesiones autónomas.
5. Se tendrá en cuenta el aprendizaje basado en proyectos

De lo anterior se verifica que en la actividad 1, el estudiante constituirá conflictos conceptuales de baja complejidad, a solucionar en el transcurso de la actividad 2, entre tanto, la actividad 4 generará conflictos cognitivos orientados a la aplicación, a subsanar con la actividad 3.

Se considera que el estudiante debe alcanzar un amplio trabajo autónomo, que posteriormente será complementado por el trabajo cooperativo de su pequeño grupo. La generación de conflictos cognitivos es importante y necesaria, para que las sesiones de gran grupo cumplan con su objetivo de afianzamiento del conocimiento.

### INVESTIGACIÓN

La asignatura por sus condiciones particulares, promueve en el estudiante la investigación formativa, de modo sea posible obtener los fundamentos con los que podrá posteriormente vincularse en el tratamiento de áreas específicas de la disciplina Informática, por medio de los grupos y semilleros de investigación dinamizados al interior de la Escuela de Ingeniería de Sistemas y Computación.

### MEDIOS AUDIOVISUALES

Aula virtual Moodle.  
Salas de informática

La asignatura es eminentemente práctica, de manera que se hace necesario que todas sus clases sean cumplidas en salas de informática.

### EVALUACIÓN

#### EVALUACIÓN COLECTIVA

No aplica.

#### EVALUACIÓN INDIVIDUAL

La evaluación en la asignatura está orientada a determinar el nivel de desarrollo de los procesos lógicos en el estudiante, junto con su capacidad para abstraer problemas y generar soluciones informáticas.

#### RANGOS DE VALORACIÓN (%)

El cálculo de la nota final se hará de la siguiente manera:

Primer corte

- 60% Evaluaciones
- 20% Talleres y exposiciones
- 20% Trabajos

Segundo corte

- 60% Evaluaciones
- 20% Talleres y exposiciones
- 20% Trabajos

### CONTENIDOS TEMÁTICOS CENTRALES

#### 1. PENSAMIENTO COMPUTACIONAL

- 1.1.Descomposición
- 1.2.Reconocimiento de patrones
- 1.3.Abstracción
- 1.4.Algoritmos

#### 2. DISEÑO ORIENTADO A OBJETOS

- 2.1.Clases
- 2.2.Atributos (constantes y variables)
- 2.3.Métodos (parámetros y retornos)
- 2.4.Encapsulamiento básico
- 2.5.Diagramas de clases (UML)

#### 3. ALGORITMOS

- 3.1.Algoritmos en lenguaje natural
- 3.2.Estructura de un algoritmo
- 3.3.Proceso (entender el problema, abstraer una posible solución usando orientación a objetos, detallar los algoritmos necesarios y realizar pruebas de escritorio)
- 3.4.Entradas y salidas
- 3.5.Diagramas de secuencia (BPMN)

#### 4. CODIFICACIÓN

- 4.1.Generalidades del lenguaje, palabras reservadas y estructuras básicas POO
- 4.2.Instalación y configuración de herramientas de programación
- 4.3.Proceso de compilación de un programa
- 4.4.Identificación y corrección de errores
- 4.5.Pruebas
- 4.6.Buenas prácticas de programación

#### 5. EXPRESIONES Y DATOS

- 5.1.Tipos de datos
- 5.2.Operadores
  - 5.2.1.Aritméticos
  - 5.2.2.Lógicos
  - 5.2.3.Unitarios
- 5.3.Expresiones compuestas

#### 6. ESTRUCTURAS DE ALGORITMIZACIÓN

- 6.1.Lineal o secuencial
- 6.2.Toma de decisión
  - 6.2.1.Simple
  - 6.2.2.Operador ternario
  - 6.2.3.Anidada
  - 6.2.4.Múltiple
- 6.3.Ciclos
  - 6.3.1.Bucle n veces
  - 6.3.2.Bucle condicional
  - 6.3.3.Recursividad básica

#### 7. ARREGLOS UNIDIMENSIONALES

#### 8. INTERACCION CON EL USUARIO USANDO PATRON MVP Y CONSOLA DEL SISTEMA

### LECTURAS MÍNIMAS

Artículos y apartados bibliográficos de los diferentes temas a tratar, proporcionados por el docente. Consulta permanente a los foros de las comunidades de desarrolladores.

### BIBLIOGRAFÍA

- Bell Tim; Witten Ian y Fellows Mike, CS Unplugged, e CS Education Research Group at the University of Canterbury, NZ, 2015.
- Freund Jakob y Rucker Bernd, Real-Life BPMN: Using BPMN, CMMN and DMN to Analyze, Improve, and Automate Processes in Your Company, CreateSpace Independent Publishing Platform, 2016.
- Jenkins Tony, Hardman Graham, How to Program Using Java, Macmillan Education, 2004.
- Deitel, Harvey M; DEITEL Paul J. Cómo programar en C/C++ y Java. Pearson. 1113 p., 2004.
- Joyanes A., Luis. Algoritmos y estructuras de datos: una perspectiva en C. McGraw-Hill. 2004, 655 p
- Wu, Introducción a la programación orientada a objetos con Java. McGraw-Hill.2001
- Sznajdleder, Pablo. Java a fondo: estudio del lenguaje y desarrollo de aplicaciones. Alfaomega. 2013.
- Torres, Manuel. Desarrollo de aplicaciones con Java. Macro EIRL. 2014
- Robert C. Martin. Código limpio. Manual de estilo para el desarrollo ágil de software. Prentice Hall. 2009
- Martin Fowler y Kendall Scot. UML gota a gota. Publisher, Pearson Educación, 1999.

En línea:

- Deitel, P y Deitel, H. (2021). Java cómo programar objetos tardíos – Introducción a Jshell. (11a. ed.) Pearson Educación. Disponible en <https://www-ebooks7-24-com.biblio.uptc.edu.co/?il=16477&pg=1>
- Moreno, Juan Carlos (2015). Programación orientada a objetos. Ediciones de la U. Disponible en <https://www-ebooks7-24-com.biblio.uptc.edu.co/?il=7927&pg=10>
- Blasco, F (2019). Programación Orientada a Objetos en Java. Ediciones de la U. Disponible en <https://www-ebooks7-24-com.biblio.uptc.edu.co/?il=10026&pg=1>
- Joyanes, L. (2020). Fundamentos de programación. (5a. ed.) McGraw-Hill - Plus. Disponible en [https://www-ebooks7-24-com.biblio.uptc.edu.co/?il=33138&pg=epubcfi\(/6/6\[data-uuid-c5071462f4ee4aa1bee5670c6ffa357a\]!/4/2\[P%C3%A1gina+I\]/1:0\)](https://www-ebooks7-24-com.biblio.uptc.edu.co/?il=33138&pg=epubcfi(/6/6[data-uuid-c5071462f4ee4aa1bee5670c6ffa357a]!/4/2[P%C3%A1gina+I]/1:0))
- Java, API especification Oracle. Disponible en <https://docs.oracle.com/en/java/javase/17/docs/api/index.html>

Nombre del docente responsable: Ing. Jorge Gabriel Hoyos Pineda