

1. **ASIGNATURA:** Diseño de Datos y Algoritmos

2. **UNIDAD ACADÉMICA QUE OFRECE LA ASIGNATURA:** Centro de Estudios de Ingeniería de Software

3. **CÓDIGO
MNEMÓNICO:** DDYA

4. **NÚCLEO AL QUE PERTENECE LA ASIGNATURA:**

Núcleo de Formación Común Institucional	Núcleo de Formación Común por Campo de Conocimiento	Núcleo de Formación Básica Profesional	Núcleo de Formación Profesional Específica
		X	

5. **REQUISITOS ACADÉMICOS:**

Nombre de la asignatura que es requisito previo	Código Mnemónico
Introducción a la Programación	IAPR

6. **INTENSIDAD SEMANAL:**

Clase dirigida por el profesor	Laboratorio	Taller	Otras (especificar)	Total horas presenciales a la semana
4.5				4.5

7. **CRÉDITOS ACADÉMICOS:**

Tiempo presencial (horas a la semana)	Tiempo independiente (horas a la semana)	Tiempo total (horas a la semana)	Créditos académicos
4.5	7.5	12.0	4

8. **JUSTIFICACIÓN:**

La capacidad de desarrollar soluciones informáticas efectivas a problemas puntuales es esencial para la producción de software de calidad. Este curso cubre los conceptos, las técnicas y las herramientas fundamentales considerando los dos componentes básicos de las soluciones informáticas: las estructuras de datos y los algoritmos.

9. COMPETENCIAS A LAS QUE APUNTA LA ASIGNATURA:

Competencias generales comunes para todos los programas	Competencias específicas
<p>Al finalizar su programa académico, un estudiante de la Escuela habrá desarrollado las siguientes competencias:</p> <ol style="list-style-type: none">1. <input checked="" type="checkbox"/> De comunicación: facilidad de interacción personal, de selección y clasificación de información y de comprensión e intercambio de información verbal, escrita y simbólica.2. <input checked="" type="checkbox"/> De productividad intelectual: dominio adecuado de los conocimientos, habilidades de pensamiento y metodologías que corresponden a los niveles de pregrado o posgrado.3. <input type="checkbox"/> De colaboración y liderazgo: demostración de actitudes éticas y sociales; de capacidades de trabajo en equipo, organización y desarrollo de proyectos, y de realización de tareas administrativas y de dirección relacionadas con la profesión. Implica, además, la disposición para el trabajo en redes, según la cual cada uno asume su responsabilidad individual en un proceso de producción colectiva.4. <input checked="" type="checkbox"/> De innovación y cambio: capacidad para identificar problemas y generar ideas que contribuyan a su solución en condiciones de riesgo e incertidumbre.5. <input type="checkbox"/> De compromiso solidario: visión para comprender, interpretar y proponer soluciones a los problemas sociales, especialmente los que afectan la dignidad humana.6. <input type="checkbox"/> De autonomía personal: capacidad para reflexionar y discernir sobre su proyecto de vida personal y profesional en el contexto social que lo rodea y tomar decisiones personales y laborales.	<p>Esta asignatura que pertenece al Núcleo de Formación Básica Profesional (NFBP) aporta al desarrollo de las competencias transversales indicadas en la columna izquierda y a las siguientes competencias específicas:</p> <ol style="list-style-type: none">1. Diseñar y construir soluciones digitales de tamaño pequeño aplicando los principios de ingeniería de software.2. Reconocer la necesidad de apropiar nuevo conocimiento y definir estrategias adecuadas para lograrlo.

10. RESULTADOS DE APRENDIZAJE:

RESULTADOS DE APRENDIZAJE DEL PROGRAMA (RAP)	RESULTADOS DE APRENDIZAJE DE LA ASIGNATURA (RAA)
<p>Al finalizar el programa, un ingeniero de sistemas de la Escuela habrá logrado los siguientes resultados de aprendizaje:</p> <ol style="list-style-type: none"><input checked="" type="checkbox"/> Comprende, apropia y aplica teorías, metodologías, procesos, técnicas, tecnologías y herramientas de ciencias de la computación y la ingeniería de sistemas en la identificación y solución de problemas propios de la profesión.<input type="checkbox"/> Desarrolla e integra arquitecturas de negocio, datos, software e infraestructura en el modelamiento de la solución de problemas.<input type="checkbox"/> Participa efectivamente en equipos interdisciplinarios en la concepción, diseño, construcción e implantación de soluciones computacionales.<input checked="" type="checkbox"/> Se comunica fluida y efectivamente en español, en los ámbitos cotidiano y profesional.<input checked="" type="checkbox"/> Se comunica fluida y efectivamente en inglés, en los ámbitos cotidiano y profesional, de acuerdo con lo establecido institucionalmente.<input type="checkbox"/> Comprende los principios fundamentales de la ingeniería de sistemas y sus áreas de aplicación y aplica un enfoque sistémico a la solución de problemas complejos apoyados en TI cumpliendo con estándares éticos y profesionales.	<p>Esta asignatura que pertenece al Núcleo de Formación Profesional Básica (NFPB) aporta al desarrollo de los siguientes resultados de aprendizaje y a los resultados de aprendizaje del programa marcados en la columna de la izquierda:</p> <p>RAA. Desarrolla soluciones informáticas efectivas y modulares a problemas puntuales reales aplicando conceptos, técnicas y herramientas adecuadas.</p> <ol style="list-style-type: none">Modela problemas usando estructuras de datos apropiadas.Desarrolla algoritmos aplicando técnicas básicas de diseño.Identifica y reutiliza los tipos abstractos y los algoritmos clásicos pertinentes a las soluciones.Construye módulos siguiendo principios de diseño.Justifica las decisiones de diseño con argumentos de corrección y eficiencia.Desarrolla soluciones siguiendo las etapas de diseño, análisis de algoritmos, codificación y prueba.Maneja con propiedad las facilidades de un lenguaje de programación y un ambiente de desarrollo.

11. CONTENIDO RESUMIDO:

Conceptos y técnicas fundamentales de la programación imperativa modular para desarrollar soluciones software a problemas puntuales. Está organizado en tres temas: (i) técnicas de diseño y análisis de algoritmos; (ii) estructuras de datos; y (iii) tipos abstractos de datos y algoritmos clásicos.

12. PROGRAMACIÓN DETALLADA.

- Técnicas de diseño y análisis de algoritmos.
Objetivos específicos:
Al finalizar esta unidad, el estudiante:
 - Desarrolla algoritmos aplicando técnicas básicas de diseño.
 - Justifica las decisiones de diseño con argumentos de corrección y eficiencia.
 - Desarrolla soluciones siguiendo las etapas de diseño, análisis de algoritmos, codificación y prueba.
 - Maneja con propiedad un lenguaje de programación y un ambiente de desarrollo (Python, Pycharm).
- Contenido:
 - Especificación y corrección: pre y poscondición.
 - Métodos de solución: fuerza bruta, incremental y dividir y conquistar.
 - Técnicas de diseño de algoritmos: iteración (invariantes), recursión (funciones recursivas) y programación dinámica.
 - Técnicas de análisis de algoritmos: análisis asintótico y teorema maestro.
 - Pruebas de programa.

2. Estructuras de datos.

Objetivos específicos:

Al finalizar esta unidad, el estudiante:

- Modela problemas usando estructuras de datos apropiadas.
- Justifica las decisiones de diseño con argumentos de corrección y eficiencia.
- Desarrolla soluciones siguiendo las etapas de diseño, análisis de algoritmos, codificación y prueba.
- Utiliza con propiedad un lenguaje de programación y un ambiente de desarrollo

Contenido:

Estructuras de datos básicas: lineales, árboles y grafos.

Estructuras lineales: conceptos básicos, especificación, representación, recursión estructural, búsqueda y ordenamiento.

Árboles: conceptos básicos, especificación, representación, recursión estructural, búsqueda (niveles y profundidad - preorden, inorden y posorden – voraz y reintento), ordenamiento y balanceo.

Grafos: conceptos básicos, especificación, representación, recursión estructural y búsqueda (niveles y profundidad – voraz y reintento).

3. Tipos abstractos de datos y algoritmos clásicos.

Objetivos específicos:

Al finalizar esta unidad, el estudiante:

- Identifica y reutiliza los tipos abstractos y los algoritmos clásicos pertinentes a las soluciones.
- Construye módulos siguiendo principios de diseño.
- Desarrolla soluciones siguiendo las etapas de diseño, análisis de algoritmos, codificación y prueba.
- Maneja con propiedad un lenguaje de programación y un ambiente de desarrollo.

Contenido:

- Tipos abstractos de datos: Montones binarios, colas de prioridad, tabla de hash y conjuntos disyuntos.
- Algoritmos clásicos: árboles de cubrimiento y búsqueda de camino mínimo.
- Principios de diseño de módulos: separación de responsabilidades y ocultación de información.
- Recursión estructural y generativa.

13. METODOLOGÍAS DE ENSEÑANZA-APRENDIZAJE:

La Escuela reconoce que cada asignatura exige un tratamiento didáctico coherente con la naturaleza del conocimiento que se enseña, con las competencias correspondientes y con el nivel de conocimiento de los estudiantes. Por tal razón, no privilegia un modelo didáctico particular.

Las metodologías utilizadas en el curso son:

- **Estudio dirigido:** para que los estudiantes adquieran conocimiento inicial de los temas del curso. Esta actividad es de aprendizaje individual. Una guía de estudio semanal. (3 horas semanales no presenciales)
- **Clases:** para que los estudiantes afiancen los conocimientos correspondientes a lecturas previas, mediante discusión teórica y ejercicios de práctica. Esta actividad es de aprendizaje grupal e individual. Una sesión de clase a la semana. (1.5 horas semanales presenciales)
- **Laboratorios:** para que los estudiantes aprendan a solucionar problemas con el acompañamiento del profesor. Esta actividad es de aprendizaje grupal e individual. Dos sesiones semanales. (3 horas semanales presenciales)
- **Arenas de programación:** para que los estudiantes practiquen la solución de problemas. Esta actividad es de aprendizaje individual. Los estudiantes deben resolver mínimo dos problemas a la semana (4.5 horas semanales no presenciales)
- **Solución de problemas:** para que los estudiantes evidencien el logro de los objetivos de aprendizaje y el desarrollo de las competencias. En los exámenes, la solución de problemas incluye preguntas sobre los conceptos aplicados. Seis sesiones al semestre. (9 horas presenciales al semestre)

14. SISTEMAS DE EVALUACIÓN:

En la Escuela, la evaluación del aprendizaje se asume como un proceso formativo y complejo que debe trascender la intención de utilizarse únicamente para certificar los conocimientos adquiridos por los estudiantes y evitar su uso como instrumento de poder y control.

Mecanismos de evaluación:

- Estudio dirigido: los estudiantes presentan comprobaciones del aprendizaje logrado siguiendo las guías de estudio que son evaluadas por el profesor.
- Clases: los estudiantes reciben realimentación, tanto del profesor como de sus compañeros, durante la discusión teórica, solución de ejercicios y desarrollo de trabajos por pares. Adicionalmente, los resultados de trabajos en clase son evaluados por el profesor.
- Laboratorios: los estudiantes reciben orientación para el desarrollo de sus laboratorios del profesor y sus monitores. Los trabajos realizados por los equipos en los laboratorios son presentados para revisión de par al monitor y evaluados por el profesor.
- Arenas de programación: Los estudiantes reciben realimentación de sus monitores con respecto al trabajo realizado en las arenas de programación.
- Solución de problemas: los estudiantes participan en el análisis de las posibles soluciones a los problemas planteados y reciben realimentación del profesor.

15. CALIFICACIONES:

Porcentaje calificación del primer tercio	Porcentaje de calificación del segundo tercio	Porcentaje de calificación del tercer tercio	Porcentaje de calificación valorativa (si aplica)	Porcentaje de calificación examen final (si aplica)	Porcentaje de calificación Laboratorio (si aplica)
30	30	40	---	---	---

16. RECURSOS:

16.1 Texto(s) guía:

Introduction to Algorithms. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2009). MIT press.

16.2 Otras referencias bibliográficas:

1. Data Structures and Algorithms in Python. Goodrich, M. T., Tamassia, R., Goldwasser, M. H. (2013). Hoboken: Wiley.
2. Algorithm Design. Kleinberg, J., Tardos, E. (2006). Pearson Education India.
3. Data Structures and Algorithms. Hopcroft, J. E., Ullman, J. D. (1983).
4. Data Structures and Algorithms Using Python. Wiley Publishing. Necaise, R. D. (2010).

16.3 Sitios web relevantes para la asignatura:

Campus Virtual. Curso DDYA.

16.4 Otros recursos:

Publicados en el Campus Virtual. Curso DDYA.

17. VIGENCIA Y MODIFICACIONES:

Contenidos vigentes desde	202X-X
Última fecha de actualización	2023-2

Aprobado

Claudia Patricia Santiago Cely
Directora del Centro de Estudios de Ingeniería de Software

Firma:



Aprobado

Claudia Patricia Santiago Cely
Decana de Ingeniería de Sistemas

Firma:

