

Asignatura	Programación II
Código	IS284
Créditos	4
Intensidad semanal	6 horas semanales para 96 horas totales
Requisitos	IS105 Programación I

Justificación	<p>El proceso formativo del Ingeniero de Sistemas, requiere como elemento de formación básica, el conocimiento de los diferentes paradigmas de programación y sus modelos computacionales.</p> <p>Nuestra perspectiva incorpora el estudio del paradigma imperativo, después de alcanzar cierta familiaridad con el paradigma funcional.</p> <p>El control de las técnicas básicas de programación de algoritmos es una herramienta fundamental para adquirir el dominio de la descomposición de problemas y plantear una solución posible y eficiente. Es crítico para cualquier ingeniero el poder definir estrategias de solución con orden, secuencia, iteración, recursividad y análisis profundo de la naturaleza del problema planteado, y así generar un modelo (algoritmo) que pueda ser solución a una necesidad cualquiera.</p>
Competencias Previas	<ul style="list-style-type: none"> - Saber contar: sucesiones, sumatorias y relaciones de recurrencia. - Aplicación de conocimientos en programación funcional:

	<p>concepto de función y su invocación, recursión, concepto de Lista.</p> <ul style="list-style-type: none"> - Aplicación de conocimientos en programación funcional: manejo de funciones, tipos de recursión, condicionales, vectores, cadenas estructuras de datos, conceptos de ordenamiento y búsqueda. - Conocimiento básico del idioma inglés.
Objetivo general	<p>Para un problema algorítmico mediana y alta complejidad el estudiante diseñe un algoritmo, mediante un enfoque funcional elemental, que lo solucione y por el otro implementarlo en un lenguaje de programación imperativa.</p>
Objetivos Específicos	<ul style="list-style-type: none"> • Conocer los fundamentos básicos de la programación imperativa. • Generar programas algorítmicos en pseudocódigo. • Conocer y aplicar sintaxis de los diagramas de flujo. • Conocer y aplicar las técnicas básicas de programación modular. • Gestión dinámica de memoria (apuntadores y referencias). • Fundamentos de archivos. • Conocer los fundamentos básicos de la algoritmia (complejidad). <p>Implementar la solución algorítmica mediante un lenguaje de programación imperativo.</p>
Metodología	<p>Se expondrá el contenido teórico en el aula de clase, se realizarán talleres que permitan poner en práctica la teoría expuesta. Planteamiento de problemas de diversa índole que abarquen desde perspectivas antiguas hasta las modernas. Investigación de casos de otras áreas del conocimiento, no</p>

	necesariamente las abordadas por la ingeniería de sistemas.
Competencias Genéricas	<ul style="list-style-type: none"> - Aprendizaje autónomo - Capacidad de análisis y síntesis. - Capacidad de aplicar los conocimientos a la práctica. - Resolución de problemas. - Trabajo individual y por parejas. - Comunicación oral y escrita.
Competencias específicas	<p>Cognitivas (Saber):</p> <ul style="list-style-type: none"> - Idioma - Matemáticas - Nuevas tecnologías TIC - Conocimientos de informática <p>Procedimentales / Instrumentales (Saber hacer):</p> <ul style="list-style-type: none"> - Redacción en interpretación de documentación técnica - Estimación y programación del trabajo - Planificación, organización y estrategia. <p>Actitudinales (Ser):</p> <ul style="list-style-type: none"> - Calidad - Toma de decisión - Capacidad de iniciativa y participación

Contenido de la asignatura	
Unidad 1	Fundamentos de la programación imperativa. a. Los pseudo lenguajes

	<ul style="list-style-type: none"> b. La asignación c. El tipado de datos
Unidad 2	<p>Fundamentos generales (constructores básicos parte I)</p> <ul style="list-style-type: none"> a. Variables y constantes b. Operadores y palabras reservadas c. Control del flujo de programas. d. Estructuras de selección simple. e. Estructuras de selección anidadas f. Verificación, documentación y pruebas.
Unidad 3	<p>Fundamentos generales (constructores básicos parte II)</p> <ul style="list-style-type: none"> a. Las funciones y sus parámetros. b. procedimientos y sus parámetros.
Unidad 4	<p>Procesos iterativos frente a procesos recursivos</p> <ul style="list-style-type: none"> a. Ciclos (para , hace_mientras, mientras_hacer) b. Recursión (divide y vencerás)
Unidad 5	<p>Arreglos</p> <ul style="list-style-type: none"> a. Vectores b. Algoritmos de Ordenamiento y búsqueda c. Cadenas de caracteres d. Matrices
Unidad 6	<p>Manejo dinámico de memoria</p> <ul style="list-style-type: none"> a. Modelo de Memoria b. Apuntadores c. Indirección múltiple d. Parámetros por referencia en las funciones

Unidad 7	Archivos a. Secuenciales b. Texto
Unidad 8	Complejidad a. Introducción a la complejidad algorítmica. b. tipos de complejidad.

Texto Guía	Si se tiene, en revisión de nuevo material.
-------------------	---

Referencia	Bibliografía
1	“Introducción a la programación con C++”, Autor: Deitel Y Deitel., Editorial Prentice Hall.
2	“C Manual de Referencia” cuarta edición Autor: Herbert Schildt Editorial: Mc Graw Hill
3	“El Lenguaje de Programación C”, Brian W Autor: Kernighan, Dennis M. Ritchie, Editorial Prentice Hall.
4	“Programación en C”, Autor: Luis Joyanes Aguilar - Editorial: Mc Graw Hill