



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>
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). <ul style="list-style-type: none"> • Implementar la solución algorítmica mediante un



	lenguaje de programación imperativo.
--	--------------------------------------

Metodología	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 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	<ul style="list-style-type: none"> - Cognitivas (Saber): <ul style="list-style-type: none"> - Idioma - Matemáticas - Nuevas tecnologías TIC - Conocimientos de informática - Procedimentales / Instrumentales (Saber hacer): <ul style="list-style-type: none"> - Redacción en interpretación de documentación técnica - Estimación y programación del trabajo - Planificación, organización y estrategia. - Actitudinales (Ser): <ul style="list-style-type: none"> - Calidad - Toma de decisión - Capacidad de iniciativa y participación
Estrategias de aprendizaje	<p>Las técnicas docentes que se van a utilizar son:</p> <ol style="list-style-type: none"> a. Clases de teoría b. Exposiciones sobre trabajos de casos prácticos. c. Desarrollo en clase, de proyectos dirigidos por el docente d. Tutorías colectivas de teoría e. Clases de prácticas f. Corrección de las prácticas g. Tutorías colectivas de prácticas <ul style="list-style-type: none"> - Tutorías individualizadas

Contenido de la asignatura



Unidad 1	Fundamentos de la programación imperativa. a. Los pseudo lenguajes b. La asignación c. El tipado de datos d. Técnicas solución de problemas (divide y venceras)
Unidad 2	Fundamentos generales (constructores básicos parte I) 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	Fundamentos generales (constructores básicos parte II) a. Las funciones y sus parámetros. b. procedimientos y sus parámetros.
Unidad 4	Procesos iterativos frente a procesos recursivos a. Ciclos (para , hace_mientras, mientras_hacer) b. Recursión
Unidad 5	Arreglos a. Vectores b. Algoritmos de Ordenamiento y búsqueda c. Cadenas de caracteres d. Matrices
Unidad 6	Manejo dinámico de memoria 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.

Semana	Tema	Actividades	Referencias Bibliográficas
---------------	-------------	--------------------	-----------------------------------



1	Fundamentos de la programación imperativa. a. Los pseudo lenguajes b. La asignación c. El tipado de datos	Clase magistral	1,2,3,4
2	Fundamentos de la programación imperativa. d. Direccionamiento de memoria física: Modelos de memoria. e. Interpretadores y compiladores.	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
3	Fundamentos generales (constructores básicos) a. Variables y constantes b. Operadores y palabras reservadas c. Control del flujo de programas	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
4	Fundamentos generales (constructores básicos parte I) d. Estructuras de selección simple	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
5	Fundamentos generales (constructores básicos parte I) e. Estructuras de selección anidadas	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
6	Fundamentos generales (constructores básicos parte I) f. Verificación, documentación y pruebas	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
7	Fundamentos generales (constructores básicos parte II) a. Las funciones y sus parámetros.	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4



	b. procedimientos y sus parámetros.		
8	Procesos iterativos frente a procesos recursivos (parte I simple) a. Ciclos (para , hacer mientras, mientras)	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
9	Procesos iterativos frente a procesos recursivos (parte II Anidados) a. Ciclos (para , hacer mientras, mientras)	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
10	Procesos iterativos frente a procesos recursivos b. Recursión (divide y vencerás)	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
11	Arreglos a. Vectores b. Algoritmos de Ordenamiento y búsqueda	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
12	Arreglos c. Cadenas de caracteres d. Matrices	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
13	Manejo dinámico de memoria a. Apuntadores b. Indirección múltiple	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
14	Manejo dinámico de memoria c. Parámetros por referencia en las funciones	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
15	Archivos a. Secuenciales b. Indizados	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4



16	Complejidad a. Introducción a la complejidad algorítmica. b. tipos de complejidad.	Clase magistral, talleres, laboratorio, quiz, consultas.	1,2,3,4
----	--	--	---------

Evaluación	Porcentaje	Objetivo
Parcial 1	20%	Prueba individual de carácter obligatorio, con los que se espera evaluar el nivel de conocimientos desarrollados por el estudiante a lo largo de las <i>unidades 1, 2 y 3</i>
Parcial 2	20%	Prueba individual de carácter obligatorio, con los que se espera evaluar el nivel de conocimientos desarrollados por el estudiante a lo largo de las <i>unidades 4 y 5</i>
Proyecto	15%	Proponer un problema algorítmico de alta complejidad y que el estudiante diseñe un algoritmo, mediante un enfoque funcional que lo solucione y que lo implemente en un



		lenguaje de programación imperativa.
Talleres y laboratorios	25%	El objetivo es instruir y entrenar al estudiante en el uso de las herramientas computacionales (lenguaje de programación y compilador) y evaluar el nivel de conocimientos desarrollado por el estudiante.
Exámen final	20%	Prueba individual de carácter obligatorio, con los que se espera evaluar el nivel de conocimientos desarrollados por el estudiante a lo largo todo el curso.

Texto Guía (si se tiene)	
---	--

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.