



Universidad  
Tecnológica  
de Pereira



**UNIVERSIDAD TECNOLÓGICA DE PEREIRA**  
**FACULTAD DE INGENIERÍAS: ELÉCTRICA, ELECTRÓNICA, FÍSICA Y**  
**CIENCIAS DE LA COMPUTACIÓN**  
**PROGRAMA INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

<b>ASIGNATURA:</b>	<b>PROGRAMACION II</b>
<b>CODIGO:</b>	<b>IS284</b>
<b>CREDITOS:</b>	<b>4</b>
<b>INTENSIDAD:</b>	<b>6 horas semanales para 96 horas totales</b>
<b>REQUISITOS:</b>	<b>IS105 Programación I</b>

#### **JUSTIFICACION**

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. Nuestra perspectiva incorpora el estudio del paradigma imperativo, después de alcanzar cierta familiaridad con los lenguajes funcionales.

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 y análisis profundo de la naturaleza del problema planteado, y generar un modelo que pueda ser solución a una necesidad cualquiera.

#### **OBJETIVO GENERAL**

Estudiar el paradigma de computación imperativo, conocer los mecanismos de administración de memoria dinámica y de archivos. Tener un acercamiento a temas básicos abordados por la algoritmia.

## **OBJETIVOS ESPECIFICOS**

- Conocer los fundamentos básicos de la programación imperativa.
- Generar programas algorítmicos en pseudocódigo
- Conocer y aplicar sintaxis de los diagramas de flujo.
- Gestión dinámica de memoria (apuntadores y referencias)
- Fundamentos de archivos
- Introducción a gráficos por computadora
- Conocer y aplicar las técnicas básicas de programación modular
- Conocer los fundamentos básicos de la algoritmia.
- Conocer fundamento en de la programación imperativa en otro lenguaje: PYTHON

## **CONTENIDO**

1. Fundamentos de la programación imperativa.
  - a. Los pseudo lenguajes
  - b. La asignación
  - c. El tipado de datos
  - d. Direccionamiento de memoria física: Modelos de memoria.
  - e. Interpretadores y compiladores.
2. Fundamentos de algoritmia (Divide y Vencerás)
  - a. Divide y vencerás
3. Fundamentos generales (Caso de estudio: Lenguaje C)
  - a. Variables y constantes
  - b. Operadores y palabras reservadas
  - c. Control del flujo de programas
  - d. Las funciones y sus parámetros
4. Procesos iterativos frente a procesos recursivos
  - a. Ciclos
  - b. Recursión
5. Manejo dinámico de memoria
  - a. Apuntadores
  - b. Indirección múltiple
  - c. Parámetros por referencia en las funciones
6. Arreglos
  - a. Vectores
  - b. Algoritmos de Ordenamiento y búsqueda
  - c. Cadenas de caracteres
  - d. Matrices
7. Archivos
  - a. Secuenciales
  - b. Indizados
8. Gráficos
  - a. Memoria de video
  - b. Funciones de gráficos
  - c. Interrupciones

## **METODOLOGIA**

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.

## **TÉCNICAS DOCENTES**

Las técnicas docentes que se van a utilizar son:

- Clases teóricas
- Exposiciones sobre trabajos de casos prácticos.
- Talleres
- Prácticas en lenguaje C
- Investigación

## **DESARROLLO Y JUSTIFICACIÓN:**

### **Clases de teoría:**

Se expondrá la información básica necesaria para abordar el tema, se entregarán referencias bibliográficas para que el estudiante profundice en su tiempo de estudio.

### **Talleres:**

Se propondrán problemas para resolver en clase buscando la interacción entre el expositor y el estudiante, permitiendo además la interacción entre los mismos estudiantes.

### **Acerca de las prácticas:**

Se realizará al menos una práctica guiada en salas de cómputo buscando implementación del pseudocódigo en lenguajes de programación.

## **MECANISMOS DE CONTROL Y SEGUIMIENTO**

El profesor podrá comprobar el grado de seguimiento de la asignatura mediante:

- La asistencia a las clases de teoría y prácticas
- Las exposiciones de temas de teoría.
- La corrección de las prácticas.
- Los parciales
- Los exámenes de corta duración (Quiz).

## EVALUACIÓN

Según el reglamento estudiantil vigente, en sus artículos 72 y 73. "**ARTÍCULO 72o.:** Se entiende por Prueba Parcial aquella que se realiza individualmente para verificar el logro de los objetivos de las diferentes unidades o temas en que se divide cada asignatura. Estas **no podrán ser menos de dos para cada asignatura...** **ARTÍCULO 73o.:** Se entiende por Prueba Final aquella que se realiza individualmente para verificar el logro de los objetivos generales de cada asignatura. Esta prueba se realizará con estricta observancia de las fechas establecidas en el calendario académico...", subrayado y resaltado nuestro.

El profesor hará como mínimo dos (2) pruebas parciales y una (1) prueba ó examen final, estando en libertad de tomar el resultado de otras actividades y trabajos como calificaciones de la asignatura. Se ha sugerido el desarrollo de dos proyectos durante el semestre.

Según acuerdo realizado por los profesores integrantes del área se tomarán cinco notas de 20% cada una.

## ORGANIZACIÓN SEMANAL

Nro. Sem ana	Temas	Clases de teoría (Hora)	Clases Prácticas (Horas)	Tutorías Profesor (Horas)	Examen (Horas)
1	1. Fundamentos de la programación imperativa. a. Los seudolenguajes	4	2	2	
2	b. La asignación c. El tipado de datos	4	2	2	
3	d. Direccionamiento de memoria física: Modelos de memoria.	2	2	2	
4	e. Interpretadores y compiladores	4	2	2	
5	2. Fundamentos de algoritmia (Divide y Vencerás)	4	2	2	
6	3. Fundamentos generales (Caso de estudio: Lenguaje C) a. Variables y constantes b. Operadores y palabras reservadas	4	2	2	
7	c. Control del flujo de programas d. Las funciones y sus parámetros	4	2	2	
8	4. Procesos iterativos frente a procesos recursivos a. Ciclos b. Recursión	4	2	2	
9	5. Manejo dinámico de memoria a. Apuntadores	4	2	2	
10	b. Indirección múltiple c. Parámetros por referencia en las funciones	4	2	2	
11	6. Arreglos a. Vectores b. Algoritmos de Ordenamiento y búsqueda	4	2	2	
12	c. Cadenas de caracteres d. Matrices	2	2	2	
13	7. Archivos a. Secuenciales	4	2	2	
14	b. Indizados	2	2	2	



15	8. Gráficos a. Memoria de video b. Funciones de gráficos	4	2	2	
16	c. Interrupciones	2	2		

## BIBLIOGRAFIA

- Joyanes Aguilar, Luis, Martinez Ignacio, “Fundamentos de Programación en C”, Ed. McGraw-Hill, 2.003, España.
- Joyanes Aguilar, Luis, “C: Algoritmos, Programación y Estructuras de Datos”. Schaum, 2005, España.
- Gottfried, B., “Programación en C (2a. edición)”. Ed. Schaum, Mexico, 2005.
- L. Joyanes. “C. Algoritmos, programación y estructuras de datos”. Schaum, 2005.
- Kernighan, Brian W. y Ritchie, Dennis M., “Lenguaje de programación C”, Ed. Prentice-Hall, Mexico.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, “Introduction to Algorithms”, The MIT Press; 2nd edition (September 1, 2001).
- Donald E Knuth , “The Art of Computer Programming, Volumes 1-3”, Ed. Addison-Wesley Professional; 2 edition (October 15, 1998).

## LINKS DE INTERÉS

[www.python.org](http://www.python.org)

[www.mundogeek.net/tutorial-python](http://www.mundogeek.net/tutorial-python)