



*Universidad Autónoma “Gabriel René Moreno”*  
*FACULTAD DE INGENIERIA EN CIENCIAS*  
*DE LA COMPUTACIÓN Y TELECOMUNICACIONES*



## **PROGRAMA ANALÍTICO DE ASIGNATURA**

### **1. DATOS GENERALES**

<b>ASIGNATURA</b>	: Programación II
<b>PRE-REQUISITOS</b>	: INF-120, MAT-103
<b>SIGLA Y CODIGO</b>	: INF-210
<b>CURSO</b>	: Tercer Semestre
<b>HORAS</b>	: 6
<b>CREDITOS</b>	: 5
<b>REVISADO EN</b>	: Enero 2011

### **2. JUSTIFICACION**

La materia de programación II es un nexo entre las materias de Programación I y Estructuras de Datos I, pues complementa los conceptos de Programación orientada a objetos iniciado en Programación I e introduce los conceptos de Recursividad y Análisis de algoritmos que son la base fundamental para el diseño de estructuras de datos.

### **3. OBJETIVOS**

#### **3.1. OBJETIVO GENERAL**

Al finalizar el curso el estudiante será capaz de:

- Capacidad para programar en pequeña escala aplicando los conceptos de programación orientada a objetos, realizando previamente un diagrama de clases que modela la aplicación.
- Fomentar la participación del alumno en el desarrollo de trabajos en equipo.



*Universidad Autónoma “Gabriel René Moreno”*  
**FACULTAD DE INGENIERIA EN CIENCIAS**  
**DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



- Podrá comparar, resolver y evaluar algoritmos desde el punto de vista de complejidad y costo asintótico, como también conocer las limitaciones de los mismos.
- Aplicar técnicas de recursión para la solución de problemas algorítmicos.

### **3.2. OBJETIVOS ESPECÍFICOS**

Al finalizar el curso el estudiante será capaz de:

- Describir los conceptos que caracterizan al modelo OO.
- Valorar en que medida las técnicas OO favorecen la calidad del software, analizando sobre todo cómo facilitan la reutilización, extensibilidad, herencia y polimorfismo.
- Contrastar como diferentes lenguajes de programación OO (Object Pascal, C++ y Java) reflejan los conceptos del paradigma.
- Modelar problemas con un diseño de diagramas de clases.
- Utilizar correctamente la recursividad en el diseño de algoritmos.
- Contrastar soluciones iterativas y recursivas.
- Podrá comparar, resolver y evaluar algoritmos desde el punto de vista de complejidad y costo asintótico, como también conocer las limitaciones de los mismos.
- Programar en C++, realizando aplicaciones propias en el ámbito de los contenidos.

### **4. CONTENIDO MÍNIMO**

Fundamentos de algoritmia; Recursividad; Programación Orientada a objetos.

### **5. UNIDADES DEL PROGRAMA ANALÍTICO**

#### **UNIDAD I: Análisis de algoritmos**

**Tiempo:** 26 horas

#### **Objetivos**

- Conocer los conceptos de análisis de algoritmos, tiempo y complejidad



*Universidad Autónoma “Gabriel René Moreno”*  
**FACULTAD DE INGENIERIA EN CIENCIAS**  
**DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



- Analizar de algoritmos de complejidad lineal, cuadrática, logarítmica y n-logarítmica.

### **Contenido**

1. Introducción
  1. Historia de los algoritmos
  2. Definición de algoritmo
2. Cálculo de tiempo (T(n))
3. Cálculo de complejidad ( O(n) )
4. Ejercicios.

### **UNIDAD II: Recursividad**

**Tiempo:** 30 horas

#### **Objetivos**

- Recordar el concepto del principio de inducción completa como base para los algoritmos recursivos
- Resolver algoritmos utilizando recursión

#### **Contenido**

1. Concepto
2. Principio de Inducción Completa (P.I.C.)
3. Algoritmos recursivos usando el primer principio de inducción
  - Algoritmos con Números
  - Algoritmos con cadenas
  - Algoritmos con Vectores
  - Algoritmos con Matrices
4. Algoritmos recursivos usando el segundo principio de inducción.
5. Aplicaciones de algoritmos recursivos

### **UNIDAD III: Programación Orientada a Objetos Avanzada**

**Tiempo:** 40 horas

#### **Objetivos**

- Construir un diagrama de clases aplicando los conceptos de programación orientada a objetos y diagrama de clases
- Implementar en lenguaje C++ una aplicación pequeña a partir de el diagrama de clases.

#### **Contenido**

1. Introducción
2. Conceptos Básicos
  - Clases y Objetos
  - Atributos y Métodos
  - Ligadura estática y dinámica
3. Polimorfismo



*Universidad Autónoma “Gabriel René Moreno”*  
**FACULTAD DE INGENIERIA EN CIENCIAS**  
**DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



- Sobrecarga de métodos y operadores
- Reescritura de métodos
- 4. Diagrama de Clases
  - Conceptos
  - Relaciones entre clases
    - a. Relación de Herencia
      - o Simple
      - o Múltiple
    - b. Relación de Asociación
    - c. Relación de Agregación
    - d. Relación de Composición
    - e. Relación de Dependencia
- 5. Implementación a partir de un diagrama de clases
- 6. Proyecto.

## **6. METODOLOGÍA**

Clase magistral

Clases de prácticas dirigidas en la que el estudiante es el principal protagonista

Trabajos de aplicación de tipo práctico y aplicativo

## **7. CRONOGRAMA**

<b>SEMANA ACTIVIDAD</b>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Presentación de la materia <b>Unidad I</b> , conceptos																		
Análisis de alg. Lineales																		
Análisis de alg. Cuadráticos																		
Análisis de alg. Logarít., n- logaritm.																		
Primera evaluación																		
<b>Unidad II</b> , conceptos																		
Algoritmos con números																		
Algoritmos con cadenas																		
Algoritmos con vectores																		
Algoritmos con matrices																		
Segunda evaluación																		
<b>Unidad III</b> , conceptos clase																		
Implem. una clase, sobrecarga																		





*Universidad Autónoma “Gabriel René Moreno”  
FACULTAD DE INGENIERIA EN CIENCIAS  
DE LA COMPUTACIÓN Y TELECOMUNICACIONES*



5. Brassard Bratley; “Fundamentos de Algoritmia”;
6. Baeza Yates ricardo; “Algoritmia”
7. Pareja Flores Cristobal, Ojeda Aciego Manuel, Adeyro Quezada Ángel, Rossi Jimenez Carlos; “Algoritmos y programación en Pascal”.
8. Instituto Nacional de Capacitación; “Manual de análisis y diseño de algoritmos”

### **REFERENCIAS BIBLIOGRAFICAS EN LA WEB**

- 1. Blog de la materia**  
[www.pro2uagrm.blogspot.com](http://www.pro2uagrm.blogspot.com)
- 2. Sobre el origen del principio de inducción matemática:**  
<http://www.panchonet.net/educacion/140inducccion.htm>
- 3. Ejercicios de inducción sobre estructuras numéricas:**  
<http://www.eneayudas.cl/indmat.htm>
- 4. Problemas sobre teoría de números:**  
<http://www.geocities.com/jespinos57/>
- 5. Mathematical Induction:**  
<http://www.math.csusb.edu/notes/proofs/pfnot/node10.html>
- 6. Programacion en C++ como si estuviera en primero**  
<http://fcapra.ceit.es/AyudaInf/Index.htm>
- 7. Manual de análisis y diseño de algoritmos.**  
<http://www.informatica.inacap.cl>