**Programa Analítico**

**PROGRAMA ANALÍTICO DE ASIGNATURA**

**1. DATOS GENERALES**

**ASIGNATURA**              **:** Programación II

**PRE-REQUISITOS**        **:** INF-120, MAT-103

**SIGLA Y CODIGO**                  **:** INF-210

**CURSO**                         **:** Tercer Semestre

**HORAS**                    **:** 6

**CREDITOS**                    **:** 5

**PROFESOR**                   **:** Ing. Juan Carlos Contreras Villegas

**REVISADO EN**          **:** Enero 2011

**2. JUSTIFICACIÓN**

         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 es diseño de estructuras de datos.

**3. OBJETIVOS**

**1.**

**2.**

**3.**

**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.

-      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 recursion 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
* Análizar de algoritmos de complejidad lineal, cuadrática, logarítmica y n-logaritmica.

**Contenido**

1. Introducción
   1. Historia de los algoritmos
   2. Definicion 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 recursion

**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

1. Algoritmos recursivos usando el segundo principio de inducción.
2. 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

1. Polimorfismo

-      Sobrecarga de métodos y operadores

-      Reescritura de métodos

1. Diagrama de Clases

-      Conceptos

-      Relaciones entre clases

* 1. Relación de Herencia

o   Simple

o   Múltiple

* 1. Relación de Asociación
  2. Relación de Agregación
  3. Relación de Composición
  4. Relación de Dependencia

1. Implementación a partir de un diagrama de clases
2. 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**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SEMANA**  **ACTIVIDAD** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| Presentación de la materia  **Unidad I**, conceptos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Análisis de alg. Lineales |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Análisis de alg. Cuadráticos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Análisis de alg. Logarít., n-logaritm. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Primera evaluación |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Unidad II**, conceptos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Algoritmos con números |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Algoritmos con cadenas |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Algoritmos con vectores |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Algoritmos con matrices |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Segunda evaluación |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Unidad III**, conceptos clase |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implem.una clase, sobrecarga |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Conceptos de diagrama de clases |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementacion de herencia |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Planteamiento de un problema |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Diagrama de clases del problema |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementación 1ra parte |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementación 2da parte |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Evaluacion final |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Examen recuperatorio |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Entrega de notas finales |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**8. SISTEMA DE EVALUACIÓN**

         Evaluaciones teórico –práctico de acuerdo al reglamento de la carrera

**EVALUACIÓN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Descripción** | **Tipo Eval.** | **Temas** | **Calific.** |
| 1er Parcial | Escrito | I | 30 |
| 2do Parcial | En Máquina | II | 30 |
| 3er Parcial | Proyecto | III | 40 |

**9. RECURSOS DIDÁCTICOS**

Pizarrón acrílico

Marcadores

Proyector multimedia

Computador para los estudiantes

Internet

**10.    BIBLIOGRAFÍA**

1.   Diego G. Ruiz;”C++ Programación Orientada a Objetos”; MP Ediciones

2.   Francisco Moreno;”Introducción a la OOP”; Editorial grupo EIDOS

3.   Javier García de Jalón, José Ignacio Rodríguez, José María Sarriegui, Alfonso Brazales; “Aprenda C++ Como si estuviera en primero”;

4.   Ceballos Francisco Javier.; “Programación Orientada a Objetos con C++”, RAMA.

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/140induccion.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://mat21.etsii.upm.es/ayudainf/aprendainf/Cpp/manualcpp.pdf>

**7.   Manual de análisis y diseño de algoritmos.**

[http://www.informatica.inacap.cl](http://www.informatica.inacap.cl/)