

## **DISEÑO AVANZADO ORIENTADO A OBJETOS**

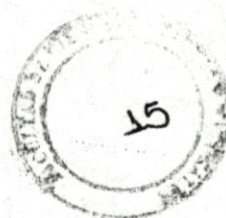
### **11.- PROGRAMA ANALÍTICO (adjuntarlo).**

#### **Objetivo:**

Esta asignatura es una continuación natural de Programación Orientada a Objetos, y permite formar Profesionales de acuerdo con las tendencias actuales en el desarrollo de Arquitecturas de Software.

Entre los principales elementos conceptuales de arquitectura se encuentran los patrones de diseño, que brindan soluciones óptimas a problemas de diseño recurrentes, independientemente del dominio de problema y del lenguaje de programación a utilizar. Dichas soluciones están garantizadas por el consenso de toda una comunidad internacional





El objetivo de esta materia es brindar a los alumnos los elementos conceptuales de arquitectura que ofrece la Orientación a Objetos, y que permiten construir software robusto, mantenible, extensible y reusable, independientemente de la plataforma de desarrollo que se utilice.

Programa:

1. Caracterización del Diseño Orientado a Objetos
  - 1.1. Polimorfismo como clave de diseño
  - 1.2. Clases Abstractas y Concretas
  - 1.3. Definición de Protocolos
  - 1.4. Contratos
  - 1.5. Jerarquías Polimórficas
  - 1.6. Tipos
  - 1.7. Double Dispatching
  - 1.8. Arquitecturas Orientadas a Objetos
    - 1.8.1. MicroArquitecturas
    - 1.8.2. MacroArquitecturas
2. Frameworks
  - 2.1. Concepto de Framework
    - 2.1.1. Caja Negra
    - 2.1.2. Caja Blanca
  - 2.2. Ciclo de vida
  - 2.3. Framework MVC
3. Patrones de Diseño
  - 3.1. Concepto de Patrón
  - 3.2. Descripción de Patrones
  - 3.3. Influencia de los Patrones en el Diseño
    - 3.3.1. Identificación de Objetos
    - 3.3.2. Granularidad
    - 3.3.3. Especificación de Interfases
    - 3.3.4. Especificación de Implementaciones
    - 3.3.5. Mecanismos de reúso
      - 3.3.5.1. Herencia VS. Composición
      - 3.3.5.2. Delegación
    - 3.3.6. Diseño para el cambio
  - 3.4. Catálogo de Patrones
    - 3.4.1. Patrones Creacionales
      - 3.4.1.1. Builder
      - 3.4.1.2. Factory Method
      - 3.4.1.3. Prototype
      - 3.4.1.4. Singleton
    - 3.4.2. Patrones Estructurales
      - 3.4.2.1. Adapter
      - 3.4.2.2. Composite
      - 3.4.2.3. Decorator
      - 3.4.2.4. Facade






- 3.4.2.5. Iterator
- 3.4.2.6. Proxy
- 3.4.3. Patrones de Comportamiento
  - 3.4.3.1. Command
  - 3.4.3.2. Mediator
  - 3.4.3.3. Observer
  - 3.4.3.4. State
  - 3.4.3.5. Strategy
  - 3.4.3.6. Template Method
  - 3.4.3.7. Visitor
- 4. Reflexión y Metaprogramación
  - 4.1. Características reflexivas de un Ambiente de Objetos
  - 4.2. Reificación
    - 4.2.1. Clases
    - 4.2.2. Mensajes
    - 4.2.3. Contextos de Ejecución
  - 4.3. MOP (Metaobject protocol)
  - 4.4. Lenguajes y Meta-lenguajes
  - 4.5. Clases y Metaclases

12.- BIBLIOGRAFÍA (indicar título del libro, autor, Editorial y año de publicación)

- Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. *Design Patterns. Elements of Reusable Object-Oriented Software*. Reading, MA, Addison Wesley, 1995.  
[ [Amazon](#) ] [ [Cúspide](#) ]
- Sherman R. Alpert, Kyle Brown y Bobby Woolf. *The Design Patterns Smalltalk Companion (Software Patterns Series)*. Reading, MA, Addison Wesley, 1998.  
[ [Amazon](#) ]
- Rebecca Wirfs-Brock, Brian Wilkerson, Lauren Wiener. *Designing Object-Oriented Software*. Englewood Cliffs, NJ, Prentice Hall, 1990.  
[ [Amazon](#) ]

  
Dr. Alejandro N. Ríos  
Departamento de Computación  
FCEyN UBA

INGRESADO

UNIVERSIDAD DE BUENOS AIRES	
FACULTAD DE CIENCIAS EXACTAS Y NATURALES	
ENTRO	SALIO
13 FEB. 2006	