



UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

ORIENTACIÓN A OBJETOS II

Carrera: *Licenciatura en Informática 2003*

Año: 3°

Año 2005

Duración: *Semestral*

Profesor: *Dr. Rossi, Gustavo*

Objetivos:

Lograr las habilidades y reconocer los problemas vinculados a la construcción de un sistema computacional basado en el paradigma de orientación a objetos. Para ellos se requiere del conocimiento de un proceso par el desarrollo de software adecuado a la orientación a objetos que permita la comprensión y manejo de los mecanismos de reutilización como son las clases abstractas, frameworks y patrones de diseño.

También se proveerán con técnicas que permitan el mejoramiento del diseño de sistemas existentes.

Experimentar en ambientes de construcción de software apropiados con el objeto de lograr las destrezas prácticas que consoliden los conocimientos teóricos.

Programa:

El proceso de desarrollo de software

1. La filosofía del proceso de desarrollo de software. Las etapas del proceso de desarrollo de software. Procesos de desarrollo iterativos e incrementales, basados en modelos: Utilidad de los modelos. Los modelos a través del proceso de desarrollo de software. Cualidades y clasificación de los modelos.
2. Introducción al Rational Unified Process (RUP). Proceso Larman como un adaptación práctica del RUP. Introducción al Proceso Larman.
3. Proceso Larman - Etapa de Análisis: a) Requerimientos, Use Cases, identificación de Use Cases, Describir y rankear use cases. b) Modelo Conceptual: descubrir conceptos, agregar atributos, descubrir asociaciones. c) Comportamiento del sistema: diagramas de Interacción a nivel Análisis, Contratos de operaciones.
4. Desarrollo de un Caso de Estudio aplicando el proceso iterativo e incremental y UML.



UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

5. Proceso Larman - Etapa de Diseño: uso de los patterns GRASP para asignación de responsabilidades, Diagramas de Colaboración y Secuencia – Diagrama de Clases del diseño para el caso de estudio

Patrones de Diseño:

6. Introducción a Patrones. Definición de Patrón. Descripción de un patrón. Catálogo de Patrones.
7. Patrones de diseño. Definición. Descripción de un patrón de diseño. Organización del Catálogo de patrones de diseño. Utilidad de los patrones de diseño. Selección de los patrones de diseño. Uso de los patrones de diseño.
8. Patrones creacionales: Abstract Factory, Singleton
9. Patrones estructurales: Composite, Bridge, Decorador, Adapter
10. Patrones de comportamiento: Observer, State, Strategy, Template Method, Command, Mediator, Iterator

Refactoring

11. Introducción a Refactoring. Utilidad del refactoring. Técnica de aplicación del refactoring. Casos de uso. Catálogo de refactoring.
12. Manipulación de métodos largos: *Extract Method*, *Inline Method*, *Replace Temp with Queries*, *Replace Method with Method Object*. Mover aspectos entre objetos: *Move Method*, *Move Field*, *Extract class*. Organización de datos. *Self Encapsulate Field*, *Replace Data Value with Objects*, *Replace Type Code with Class / Subclass / State-Strategy*. Simplificación de invocación de métodos: *Rename Method*, *Replace Constructor with Factory Method*, *Parameterize Method*. Simplificación de expresiones condicionales: *Replace Conditional with Polimorfism*. Manipulación de la generalización: *Pull Up Method*, *Push Down Method*, *Extract Subclass*, *Extract Superclass*, *Form Template Method*, *Replace Inheritance with Delegation*.

Frameworks

13. Introducción a Frameworks. Reutilización de software vs, reutilización de diseño. Frameworks basados en herencia (*white box frameworks*): ejemplo Hotdraw. Frameworks basados en composición (*black box frameworks*): ejemplo MVC.
14. Frameworks como Modelos + Arquitecturas. Modelo de dominio de aplicación. Frameworks: Arquitecturas genéricas orientadas a objetos.
15. Instanciación de aplicaciones en un frameworks.
16. Documentación de frameworks. Técnicas: Descripción del Uso del Framework Descripción del Diseño Detallado. Técnicas específicas: Patrones, Patrones de diseño, Contratos, Meta-Patrones: hot-spots cards.



UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

17. Los ejemplos como herramienta para la comprensión del framework.
18. Diseño de frameworks: conducido por ejemplos, por refactoring o por puntos de flexibilización.

Bibliografía

1. *The Unified Software Development Process*. Jacobson, I., Booch, G., Rumbaugh, J., Addison Wesley. ISBN 0-201-57169-2 (1999)
2. *Applying Uml and Patterns: An Introduction to Object-Oriented Analysis and Design*. Craig Larman - Computers - 2001 - 656 pages
3. *El Proceso Unificado de Desarrollo de Software*. LARMAN CRAIG. Editorial PRENTICE-HALL, Edición 2003, en Rústica ISBN 8420534382
4. *Design Patterns. Elements of Reusable Objects Oriented Software*. Gamma, Helm, Johnson, Vlissides, Addison-Wesley, Professional Computing Series.
5. *Head First Design Patterns*. Elisabeth Freeman, Bert Bates, Kathy Sierra - Computers - 2004 - 676 pages.
6. *Refactoring: Improving the Design of Existing Code*. Fowler, Martin. Addison-Wesley, 1999.
7. *Refactoring to Patterns*. Joshua Kerievsky. Addison Wesley, 2004, ISBN : 0-321-21335-1
8. *Object-Oriented Application Frameworks: Problems & Perspectives* Vol. 1 – Editors Mohamed Fayad, Douglas Schmidt & Ralph Johnson, Wiley, 1998.
9. *Object-Oriented Application Frameworks: Applications & Experiences* Vol. 2 - Editors Mohamed Fayad, Douglas Schmidt & Ralph Johnson, Wiley, 1998