



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA  
Secretaría de Postgrado

---

## ***Tópicos de Ingeniería de Software II***

**Año 2010**

Carrera: **MAGÍSTER EN INGENIERÍA DE SOFTWARE**

**Año: 2010**

**Duración: Agosto – Diciembre**

**Profesor a Cargo: Dra. Alejandra Garrido**

**Hs. semanales : 2 hrs**

---

### **OBJETIVOS GENERALES:**

Brindar a los asistentes elementos y técnicas avanzados del desarrollo de software, que permiten construir software robusto, mantenible, extensible y reusable. Se introducirá al alumno a las metodologías actuales de desarrollo y se estudiará en profundidad el diseño de software orientado a objetos, focalizándose en arquitecturas y soluciones adaptables que permitan alcanzar las características mencionadas.

### **MODALIDAD DE EVALUACION**

Desarrollo de un trabajo práctico y un proyecto de mediana envergadura, donde los alumnos tengan la oportunidad de aplicar los conocimientos adquiridos.

### **Programa**

#### **Unidad 1.**

Principios del Diseño Orientado a Objetos. Reutilización de software. Definición de clases y responsabilidades. Definición de jerarquías de clase. Polimorfismo. Relaciones de conocimiento y composición. Herencia vs. Composición. Delegación vs. Colaboración. Cohesión y Acoplamiento.

#### **Unidad 2.**

Introducción a Patrones de Diseño. Patrones en otras áreas. Origen de los patrones de diseño. Patrones en el diseño de software. Descubriendo un patrón de diseño. Definición. Partes esenciales.

#### **Unidad 3.**



Patrones de Diseño. Análisis de diversos patrones de diseño.

#### **Unidad 4.**

Uso de Patrones. Formato “a la GOF”. Clasificación de los patrones del GOF. Criterios de Uso, Selección y Aplicación de Patrones. Comparación entre Patrones. Ventajas y consideraciones.

#### **Unidad 5.**

Metodologías Ágiles. Motivación. Diferencias entre las metodologías ágiles y las metodologías estructuradas de desarrollo. Extreme Programming (XP): valores, principios y prácticas. Scrum: roles, reuniones y artefactos; sprints. Patrones de organización en el marco de las metodologías ágiles.

#### **Unidad 6.**

Refactoring y Test Driven Development como prácticas dentro de las metodologías ágiles. Análisis de los refactorings más importantes. Relación entre refactoring y patrones. Refactoring hacia patrones.

#### **Unidad 7.**

Frameworks. Características. Ejemplos. Documentación de los distintos “concerns” de un framework. Integración de frameworks.

#### **Unidad 8.**

Adaptive Object Models. Motivación. Elementos principales. Patrones en el diseño de Adaptive Object Models. Especificación de las reglas de negocio a través de Adaptive Object Models y Metadata.

### **Bibliografía**

- “Design Patterns: Elements of Reusable Object-Oriented Software”. Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Addison-Wesley. 1994.
- “A Pattern Language: Towns, Buildings, Construction”. Christopher Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King y S. Angel, New York: Oxford University Press, 1977.



- “A Timeless Way of Building”. Christopher Alexander. Oxford University Press, 1979.
- “Pattern Languages of Program Design 3”. Software Patterns Series. Addison-Wesley. 1998.
- “Extreme Programming Explained. Embrace change”. Kent Beck. Addison-Wesley. 2004.
- “Refactoring: Improving the Design of Existing Code”. Martin Fowler. Addison-Wesley. 1999.
- “Refactoring to Patterns”, Joshua Kerievsky . Addison-Wesley. 2004.
- “Building Application Frameworks: Object-Oriented Foundations of Framework Design”. Mohamed Fayad, Ralph Johnson y Douglas Schmidt.
- “Architecture and Design of Adaptive Object Models”. Joseph W. Yoder, Federico Balaguer, & Ralph Johnson. Intriguing Technology, Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '01), ACM SIGPLAN Notices, ACM Press, December 2001.