

## Programa Regular

### Metodología de Programación

I

**Carga horaria:** 5 hs.

**Modalidad de la Asignatura:** Teórico-práctica.

#### Objetivos:

el estudiante tendrá los conocimientos de técnicas y herramientas que le permitan realizar software modular, reusable y extensible. Las técnicas mencionadas incluyen conocimientos teóricos y prácticos, habilidades, experiencias y sentido crítico, todas ellas fundamentadas en teorías y técnicas sólidas, comprobadas y bien establecidas.

Profundización en los conceptos de la programación orientada a objetos articulando vertical y horizontalmente con otras materias de la carrera.

Introducción en el uso de un lenguaje de modelado gráfico orientado a objetos (UML), que permite construir diagramas especificando distintos aspectos de un sistema.

Proporcionar al estudiante más experiencia en el campo de la programación orientada a objetos mediante la realización de prácticas.

#### Contenidos:

Análisis y diseño Orientado a Objetos. Procesos de desarrollos iterativos e incrementales. UML como Lenguaje de modelado. Patrones de Diseño. Patrones Creacionales. Patrones Estructurales. Patrones de comportamiento. Frameworks. MVC. Refactoring.

#### Unidades temáticas:

##### Unidad I

Diseño Orientado a Objetos: La filosofía del proceso de desarrollo de software. Las etapas del proceso de desarrollo de software. Procesos de desarrollos iterativos e incrementales. UML como Lenguaje de modelado. Diagrama de clases. Diagrama de estados. Diagrama de interacción.

##### Unidad II

Introducción a los Patrones de Diseño: Introducción a Patrones. Definición de Patrón. Descripción de un patrón. Catálogo de Patrones. 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.

### **Unidad III**

Patrones Creacionales: Descripción de los patrones Abstract Factory y Singleton.

### **Unidad IV**

Patrones Estructurales: Descripción de los patrones Composite, Bridge, Decorador y Adapter.

### **Unidad V**

Patrones de comportamiento: Descripción de los patrones Observer, State, Strategy, Template Method, Command, Mediator y Iterador.

### **Unidad VI**

Frameworks : 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.

### **Bibliografía:**

- Gamma, E. y otros. Patrones de Diseño. Editorial Addison Wesley (ISBN: 84-7829-059-1). Ed. 1º. Año 2002.
- Kerievsky, J. Refactoring to Patterns. Editorial Addison Wesley (ISBN: 0-321-21335-1). Año 2004.
- Freeman, E.; Bates, B.; Sierra, K. Head First Design Patterns. Editorial O Reilly & Associates (ISBN: 978-0596007126). Ed. 1º. Año 2004.
- Fayad, M.; Johnson, R.; Schmidt, D. Building Application Frameworks: Object-Oriented Foundations of Framework Design. Editorial Wiley (ISBN: 978-0471248750). Ed. 1º. Año 1999.
- Fayad, M.; Johnson, R. Domain-Specific Application Frameworks: Frameworks Experience by Industry. Editorial Wiley (ISBN: 978-0471332800). Ed. 1º. Año 1999.
- Fayad, M.; Schmidt, D.; Johnson, R. Implementing Application Frameworks: Object-Oriented Frameworks at Work. Editorial Wiley (ISBN: 978-0471252016). Ed. 1º. Año 1999.
- Johnson, R. E. Components, frameworks, patterns. In Proceedings of the 1997 Symposium on Software Reusability (Boston, Massachusetts, United

States, May 17 - 20, 1997). M. Harandi, Ed. SSR '97. ACM Press, New York, NY, 10-17. [PDF]Año 1997.

- Foote, B.; Johnson, R. Designing Reusable Classes. Journal of Object-Oriented Programming. Año 1998.

**Propuesta didáctica:** Las clases se desarrollarán en el Laboratorio de Informática. Se organizarán en modalidades teórico-prácticas con soporte de presentaciones digitales y prácticas en función de cada clase.

En las clases se presentan los contenidos teóricos y se van exponiendo situaciones reales en forma de ejemplos que ayuden a comprender los nuevos conceptos introducidos.

La formación práctica está basada en la resolución de problemas tipo y de problemas abiertos de ingeniería, cuyas resoluciones se realizan principalmente en las computadoras, utilizando aplicaciones de uso en la industria que permitan un contacto directo con las tecnologías actuales.

En cuanto a los problemas abiertos de ingeniería, se realizarán trabajos relacionados con la implementación de una aplicación determinada, que haga uso de las técnicas y herramientas presentadas en la asignatura que permitan la construcción de software modular, extensible y reusable y la profundización de los conceptos de la programación orientada a objetos. Cada trabajo debe incluir el diagrama UML del sistema, un detalle de los problemas encontrados, las formas de solucionarlos, las condiciones de ejecución, formato de los datos de entrada e ideas o sugerencias para realizar una versión mejorada del mismo. La realización de los trabajos implica la identificación de un problema dado y la solución del mismo, a partir de la aplicación de los conocimientos adquiridos hasta entonces, lo cual constituye la base formativa para que el estudiante adquiera las habilidades que le permitan encarar proyectos y diseños de ingeniería.

Los estudiantes deben realizar entregas de al menos un trabajo integrador durante el desarrollo de la cursada el cual será revisado por los docentes y devuelto a los estudiantes para su corrección.

El material correspondiente a las clases teóricas, así como los documentos de la práctica se encontrarán disponibles a través de un grupo Web al cual los estudiantes tendrán acceso. Este mecanismo también será utilizado para realizar consultas simples.

**Actividades extra-áulicas:** Se establecerán guías de actividades prácticas para

que el estudiante pueda ejercitar, a fin de consolidar los conceptos aprendidos en clase.

**Evaluación:** La evaluación integradora de las instancias teórico-prácticas se realiza a través de un parcial teórico-práctico en máquina. Además, los estudiantes deberán desarrollar un trabajo final donde se integren los temas vistos en la materia. Las clases son obligatorias ya que implican participación y debate que forman parte de la evaluación.