

Programa Regular de asignatura

- **Asignatura:** Metodologías de programación I
- **Carrera/s:** Ingeniería en Informática
- **Ciclo lectivo:** 2018
- **Docente/s:**

- **Coordinador:** Dr. Lic. Waldo Hasperué
- **Integrantes del plantel docente:** Ing. Pablo Andres González Casco
- **Carga horaria semanal:** 5 (cinco) horas
- **Tipo de asignatura:** Teórico - Práctica

Fundamentación:

Metodología de Programación I es una materia obligatoria correspondiente al tercer año de la carrera de Ingeniería en Informática.

En la materia los alumnos estudiarán las técnicas y herramientas para la construcción de software modular, extensible y reusable. Se profundiza la programación en el paradigma de la programación orientada a objetos estudiando diferentes patrones de diseño. Para cada patrón estudiado se analiza su categoría y se ejemplifica con un caso concreto su utilidad.

Los estudiantes completan los temas vistos durante el segundo año de estudio mediante la aplicación de nuevas herramientas que le permiten resolver problemas de mayor complejidad. Los temas abordados en esta materia permiten al alumno comprender la importancia de la buena práctica de programación para la construcción de grandes sistemas de software.

Los patrones de diseño estudiados durante la materia son ampliamente utilizados en el ámbito laboral. Finalizada la materia, el alumno habrá tenido su primera experiencia en el desarrollo de software utilizando patrones de diseño, siendo de utilidad en el resto de la carrera, ya que puede utilizar los conceptos vistos en otras materias del área como así también en el trabajo correspondiente a su PPS. La experiencia obtenida durante la materia también es de valiosa ayuda a aquellos alumnos que, finalizada su carrera, se dediquen al desarrollo de software orientado a objetos.

Objetivos:

- Profundizar en el paradigma de programación orientada a objetos que sirve para complementar los conocimientos introductorios que se ven en segundo año.
- Estudiar los patrones de diseño como herramienta para la solución de problemas permitiendo al alumno conocer y saber utilizar las herramientas que se utilizan en el ámbito laboral.
- Proporcionar al estudiante más experiencia en el campo de la programación orientada a objetos mediante la realización de diferentes prácticas. Cada práctica le permite al alumno conocer el uso de un patrón de diseño en particular, teniendo conocimiento de una amplia batería de herramientas al finalizar la materia
- Estudiar las técnicas desde la teoría y la práctica analizando los casos de uso. Con este objetivo se busca que el estudiante realice una rápida asociación entre un patrón de diseño y los tipos de problema que puede resolver con su uso.

Contenidos mínimos:

Análisis y diseño orientado a objetos. Proceso de desarrollos iterativos e incrementales. UML como lenguaje de modelado. Patrones de diseño. Patrones creacionales, estructurales y de comportamiento. Frameworks. MVC. Refactoring.

Contenidos :

Unidad I

Diseño orientado a objetos: La filosofía del proceso de desarrollo de software. Las etapas del proceso de desarrollo de software. Proceso 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 del patrón 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, Decorator y Adapter.

Unidad V

Patrones de comportamiento. Descripción de los patrones Observer, State, Strategy, Template method, Command, Mediator e Iterator.

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 frameworks): ejemplos MVC.

Bibliografía Obligatoria:

- Gamma, E. y otros. Patrones de Diseño. Editorial Addison Wesley (ISBN: 84-7829-059-1). Ed. 1°. 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. 1°. Año 2004.
- Fayad, M.; Johnson, R.; Schmidt, D. Building Application Frameworks: Object-Oriented Foundations of Framework Design. Editorial Wiley (ISBN: 978-0471248750). Ed. 1°. Año 1999.
- Fayad, M.; Johnson, R. Domain-Specific Application Frameworks: Frameworks Experience by Industry. Editorial Wiley (ISBN: 978-0471332800). Ed. 1°. Año 1999.
- Fayad, M.; Schmidt, D.; Johnson, R. Implementing Application Frameworks: Object-Oriented Frameworks at Work. Editorial Wiley (ISBN: 978-0471252016). Ed. 1°. 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 pedagógica –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 resolviendo en forma conjunta ejemplos que ayuden a comprender los nuevos conceptos introducidos.

La materia tiene un alto contenido de trabajos prácticos ya que los estudiantes comienzan a analizar y discutir, para cada problema presentado, que estrategia de resolución es la más conveniente. 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 en base a la programación orientada a objetos. Cada trabajo debe incluir el diagrama UML de las clases desarrolladas, 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 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. El resultado de estos trabajos son programas ejecutables que los propios alumnos realizan y que les permiten apreciar todo el poder que les brinda una computadora como herramienta para la resolución de problemas.

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 del campus virtual al cual los estudiantes tendrán acceso. Este mecanismo también será utilizado para realizar consultas simples.

Régimen de aprobación:

Para aprobar la materia los alumnos deben contar con el 75% de asistencia a clases y aprobar tres instancias de evaluación

- Un trabajo integrador

- Dos exámenes escritos

Cada examen escrito tiene un recuperatorio. Al final del cuatrimestre se cuenta con una fecha especial (examen flotante) donde los alumnos podrán recuperar uno o ambos exámenes.

El trabajo integrador se puede presentar en cualquier clase dentro del cuatrimestre. Se defiende con un coloquio y tiene un único recuperatorio (dos fechas en total).

La nota del trabajo, más la nota del parcial, más la nota conceptual del alumno, según su participación en clase determina la nota final de la materia según la siguiente fórmula:

$$0.4 * \text{Nota_Trabajo} + 0.4 * \text{Nota_Parcial} + 0.2 * \text{Nota_Concepto}$$

- Nota 7 o más se promociona la materia. Además para poder promocionar, y como lo dispone el reglamento, debe sacar como mínimo 6 en cada una de las instancias de evaluación (parcial y trabajo)
- Nota de 4 a 6.99 solo se aprueba la cursada
- Nota menor a 4 se desaprueba la cursada