

**Programa Regular**

**Asignatura:** Metodologías de Programación II

**Carrera/s:** Ingeniería en Informática

**Ciclo Lectivo:** 2018

**Docente/s:**

**Coordinador de la materia:** Lic. Claudia Cappelletti

**Plantel docente de la materia:** Lic. Claudia Cappelletti

**Carga horaria semanal:** 6 horas semanales

**Tipo de Asignatura:** teórico-práctica

**Fundamentación:** Metodologías de Programación II es una materia obligatoria correspondiente al tercer año de la carrera Ingeniería Informática. En ella, se abordarán tópicos avanzados en el análisis y diseño de programación orientada a objetos. Además se presentarán diversos frameworks y se desarrollará un software de mediana complejidad.

**Objetivos:**

Que los alumnos:

- Distingan las características de las metodologías ágiles.
- Aprendan cuáles son los métodos y prácticas ágiles.
- Determinen las fases de la Programación extrema y distingan sus tipos de implementación.
- Comprendan la diferencia entre Reflexión e Integración continua.
- Comprendan cuál es el beneficio de utilizar un Framework.
- Distingan las características de cada Framework.
- Establezcan la diferencia entre un Framework y un Patrón de diseño.
- Entiendan qué es y para qué sirve el Refactoring.
- Comprendan la necesidad de realizar testeos en el software.
- Comprendan principios avanzados de la programación orientada a objetos.
- Comprendan en qué medida las técnicas orientadas a objetos favorecen la calidad del software.
- Analicen cómo la programación orientada a objetos facilita la reutilización y extensibilidad del código.

- Adquieran habilidad para aprender el lenguaje de programación orientado a objetos Smalltalk, junto a su entorno de desarrollo.
- Analicen, diseñen e implementen una aplicación orientada a objetos utilizando el lenguaje de programación Smalltalk.
- Apliquen las técnicas y herramientas de desarrollo del lenguaje de programación aprendido en la resolución de problemas de diferentes niveles de dificultad.

#### Contenidos mínimos:

Metodologías Ágiles. Reflexión. Integración continua. Frameworks. Frameworks basados en herencia. Frameworks basados en composición. MVC. Refactoring. Testing. Test Driven Development, JUnit framework Maven.

#### Unidades temáticas:

- **Unidad I:**  
Metodologías ágiles. Características. Filosofía. Métodos ágiles. Prácticas ágiles. Medición de agilidad. Resultados. Extreme Programming y Scrum. Su filosofía. Características de cada una.
- **Unidad II:**  
Programación extrema (XP). Fases. Conceptos. Diseño. Variables. Implementación. Pruebas. Valores en XP. Roles. Scrum. Historia. Actividades. Roles. Herramientas.
- **Unidad III:**  
Reflexión. Introducción. Características. Implementación en distintos lenguajes. Integración continua.
- **Unidad IV:**  
Frameworks: Introducción a Frameworks. Reutilización de software vs. Reutilización de diseño. Frameworks basado en herencia (while box frameworks): ejemplo Hotdraw. Frameworks basados en composición (black box frameworks): ejemplo MVC.
- **Unidad V:**  
Refactoring. Características. Proceso de refactoring. Catálogo de refactoring. Refactoring en distintos lenguajes y paradigmas de programación. Refactoring de modelos.
- **Unidad VI:**  
Testing. Test Driven Development (Desarrollo guiado por

pruebas). Estrategias para testear. Pruebas de software de tipo XUnits. JUnit frameworks. Maven.

#### **Bibliografía Obligatoria:**

#### **Bibliografía correspondiente a la Unidad I: Metodologías ágiles**

“Extreme Programming Explained” Beck, K. Editorial Addison Wesley. (ISBN-978-0321278654). Ed. 2º. Año 2004.

#### **Bibliografía correspondiente a la Unidad II: Programación extrema**

“Extreme Programming Explained” Beck, K. Editorial Addison Wesley (ISBN-978-0321278654). Ed. 2º. Año 2004.

"Succeeding with agile: software development using Scrum" Cohn, M. (2010).

"Agile product management with Scrum: creating products that customers love" Pichler, R. (2010)

"Essential Scrum: A Practical Guide to the Most Popular Agile Process" Rubin, K. (2012)

"Agile Software Development with Scrum" Schwaber, K. / Beedle, M. (2002)

"Agile Project Management with Scrum" Schwaber, K. (2004)

"Agile Development: Lessons learned from the first Scrum" Sutherland, J. (2004)

"The New Product Development Game" Takeuchi, H. y Nonaka, I. (1986)

#### **Bibliografía correspondiente a la Unidad III: Reflexión e Integración continua.**

"Object Oriented Design: With Applications" Booch, G. (1991)

"A Recipe for Build Maintainability and Reusability" Flowers, J. (2006)

"Java Reflection in Action" Forman I. (2005)

"Continuous Integration: The Cornerstone of a Great Shop" Richardson, J. (2006)

#### **Bibliografía correspondiente a la Unidad IV: Frameworks**

“Building application frameworks: Object-Oriented Foundations of Framework” Fayad, M.E.; Johnson, R.E.; Schmidt, D. C. - Design. Editorial Wiley- (ISBN 978-0471248750). Ed. 1º. Año 1999.

“Frameworks Experience by Industry” - Fayad, M.E.; Johnson, R.E. Domain-Specific Application Frameworks: Editorial Wiley (ISBN: 978-0471252016). Año 1999.

**“Implementing Application Frameworks: Object-Oriented Frameworks at Work”** Fayad, M.E.; Schmidt, D.C.; Johnson, R.E. – Editorial Wiley (ISBN: 978-0471332800). Año 1999.

#### Bibliografía correspondiente a la Unidad V: Refactoring

**“Pattern Languages of Program Design”** Manolescu. D.; Voelter, M; Noble, J. Editorial Addison-Wesley (ISBN: 978-0321321947). Año 2006.

**“Refactoring to Patterns”** Kerievsky. J. Editorial Addison Wesley (ISBN: -0321-21335-1). Ed. 2°. Año 2004.

**“Head First Design Patterns”** Freeman, E.; Bates. B.; Sierra, K. Editorial O’ Really Media (ISBN: 978-0-596-51668’0). Ed. 2°. Año 2008.

#### Bibliografía correspondiente a la Unidad VI: Testing

**“Simple Smalltalk Testing with Patterns”** Kent Beck.

<http://www.xprogramming.com/testfram.htm>

**“JUnit Explained”**

<http://www.iam.unibe.ch/~ducasse/Programmez/OnTheWeb/SUnitEnglish2.pdf>

**“Test Driven Development by Example”** Kent Beck.

**“JUnit Test Patterns : Refactoring Test Code”** Gerard Meszaros.

#### Bibliografía de consulta:

**“Building application frameworks: Object-Oriented Foundations of Framework”** Fayad, M.E.; Johnson, R.E.; Schmidt, D. C. – Design. Editorial Wiley- (ISBN 978-0471248750). Ed. 1°. Año 1999.

**“Frameworks Experience by Industry”** – Fayad, M.E.; Johnson, R.E. Domain-Specific Application Frameworks: Editorial Wiley (ISBN: 978-0471252016). Año 1999.

**“Implementing Application Frameworks: Object-Oriented Frameworks at Work”** Fayad, M.E.; Schmidt, D.C.; Johnson, R.E. – Editorial Wiley (ISBN: 978-0471332800). Año 1999.

**“Pattern Languages of Program Design”** Manolescu. D.; Voelter, M; Noble, J. Editorial Addison-Wesley (ISBN: 978-0321321947). Año 2006.

**“Refactoring to Patterns”** Kerievsky. J. Editorial Addison Wesley (ISBN: -0321-21335-1). Ed. 2°. Año 2004.

**“Extreme Programming Explained”** Beck, K. Editorial Addison Wesley (ISBN-978-0321278654). Ed. 2°. Año 2004.

**“Java Reflection in Action”** Forman I. (2005)

**“Continuous Integration: The Cornerstone of a Great Shop”** Richardson, J. (2006).

**“Introducción a la programación Orientada a Objetos”**. Budd, T. Addison Wesley. (2002).

“Construcción de Software Orientado a Objetos”. Meyer, B. Prentice Hall. (2000).  
“Análisis y diseño orientado a objetos con aplicaciones”. Booch, G. Addison-Wesley Iberoamericana. (2007).  
“El lenguaje unificado de modelado”. Booch, G., Rumbaugh, J. y Jacobson, I. Addison-Wesley Iberoamericana. (2007). Cap. 1, 2, 3 y 4.  
“Descubra SmallTalk”. Lalonde, W. Addison-Wesley Iberoamericana. (2004).  
“Smalltalk Width Style”. Skublics, S, Klimas, E., Thomas, D y Pugh, J. Pearson Education. (2001).

#### Propuesta Pedagógico-Didáctica:

Por tratarse de una materia con un importante respaldo conceptual, la enseñanza de la teoría ocupa un lugar destacado. Para facilitar el aprendizaje de cada uno de los temas se recurre a relacionarlos con ejemplos vinculados con la realidad. En la práctica los alumnos aprenderán un nuevo lenguaje de programación junto con su ambiente de desarrollo. Además se introducirán conceptos avanzados de la Programación orientada a objetos. Para ello se ha seleccionado material de manera de cubrir un amplio espectro temático y se han desarrollado Trabajos Prácticos donde los ejercicios a resolver cuentan con distintos niveles de dificultad. Algunos ejercicios son resueltos completamente en clase, otros son encaminados para ser completados por los alumnos y otros son propuestos, donde su solución queda a cargo de los alumnos.

La metodología adoptada para el desarrollo de la asignatura es la siguiente:

Las 6 horas semanales que tiene asignadas la materia se distribuyen de la siguiente manera:

**Clases teóricas:** 3 horas para presentación y exposición de los temas, con el uso de pizarrón y apoyo de proyección de diapositivas desarrolladas por el coordinador de la materia

**Clases prácticas:** 3 horas para la resolución de los Trabajos prácticos en un aula informatizada. Durante las clases prácticas se formarán grupos de hasta 2 personas para realizar un Trabajo Práctico Grupal final que consistirá en un desarrollo orientado a objetos de mediana complejidad en el lenguaje de programación Smalltalk.

Las clases teóricas como las clases prácticas serán dictadas por el profesor a cargo del curso.

Además se propone entre 3 y 5 horas extra-aula para las actividades planteadas a los alumnos: búsqueda de información bibliográfica, lecturas, cuestionarios, análisis previo de trabajos prácticos, desarrollo del Trabajo Práctico Grupal, etc.

El ambiente de desarrollo a utilizar es Pharo, software que se encuentra instalado en las aulas informatizadas.

Régimen de aprobación:

Al finalizar la cursada, el alumno:

- deberá haber cumplimentado con el 75% de asistencia a clases.
- contará con una nota final, que será obtenida a partir de las calificaciones de:
  - un primer parcial teórico-práctico
  - un segundo parcial teórico-práctico
  - el desarrollo de un proyecto orientado a objetos de mediana envergadura
  - una nota de concepto que evalúa el desempeño del alumno durante toda la cursada.

Para rendir el segundo parcial el alumno debe haber aprobado el primer parcial.

La reglamentación vigente de la de la Universidad estipula que la materia:

- Se promociona: con nota final mayor o igual a 7 y menor o igual a 10. En esta situación el alumno NO deberá rendir examen final.
- Se aprueba: con nota final mayor o igual a 4 y menor a 7. En esta situación el alumno deberá rendir examen final.
- Se desaprueba: con nota final menor a 4. En esta situación el alumno deberá recurrar la materia.

Para promocionar la materia ninguna de las notas que componen la nota final debe ser inferior a 6.

Se valorará especialmente en la instancia de promoción la participación activa del alumno durante toda la cursada.

Claudia Cappelletti

Firma y Aclaración

Inicializar cada hoja y firma completa con aclaración en la última página