

**DESARROLLO DE SOFTWARE
BASADO EN MODELOS**

Año 2019

Carrera/ Plan: (Dejar lo que corresponda)*Licenciatura en Informática Plan 2015/Plan 2012/Plan 2003-07
Licenciatura en Sistemas Plan 2015/Plan 2012/Plan 2003-07
Ingeniería en Computación***Año:** 4to, 5to**Régimen de Cursada:** Semestral**Carácter (Obligatoria/Optativa):** Optativa**Correlativas:** Orientación a Objetos 2, Ingeniería de Software 2**Profesor/es:** Roxana Giandini**Hs. semanales:** 6hs**FUNDAMENTACIÓN**

El Desarrollo Dirigido por Modelos (MDD) se ha convertido en un nuevo paradigma de desarrollo de software que promete una mejora de la productividad y de la calidad del software a través de un proceso guiado por modelos y soportado por potentes herramientas que generan código a partir de modelos.

La iniciativa MDD cubre un amplio espectro de áreas de investigación: lenguajes para la descripción de modelos, definición de lenguajes de transformación entre modelos, construcción de herramientas de soporte a las distintas tareas involucradas, aplicación de los conceptos en métodos de desarrollo y en dominios específicos, etc. y para esto MDD propone el uso de un conjunto de estándares como MOF, UML y QVT. Actualmente, algunos de estos aspectos están bien fundamentados y se están empezando a aplicar con éxito, otros sin embargo están todavía en proceso de definición.

En este contexto, el aporte de esta asignatura es brindar conocimientos sobre el proceso de desarrollo de software iterativo e incremental basado en modelos, como así también un panorama general sobre los componentes del MDD. Se abarcará también el área de Pruebas de software introduciendo el paradigma de Testing Basado en Modelos.

Esta materia Optativa está dirigida tanto a alumnos avanzados en la Carrera de Licenciatura en Informática y Licenciatura en Sistemas como a alumnos de Posgrado, que cumplan el requisito previo de manejar conceptos esenciales del paradigma de orientación a objetos y conocimientos básicos del lenguaje UML.

OBJETIVOS GENERALES

- Comprender y aplicar el proceso de desarrollo de software Orientado a Objetos iterativo e incremental, utilizando el lenguaje de modelado UML.
- Comprender y aplicar el proceso de desarrollo de software al estilo Larman, aplicándolo a un ejemplo, desarrollando todas sus etapas y aplicando el lenguaje de restricciones para objetos OCL.
- Analizar la base formal de UML y herramientas para edición y validación de diagramas UML.
- Introducir al alumno en la filosofía del Desarrollo Dirigido por Modelos, brindando un panorama general sobre esta metodología de desarrollo.
- Introducir al alumno en el modelado de pruebas de software aplicando Testing basado en Modelos.

COMPETENCIAS

- LI-CE4- Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.

-
- LI-CE7- Planificar, dirigir, realizar y/o evaluar proyectos de sistemas de seguridad en el almacenamiento y procesamiento de la Información. Especificación, diseño, desarrollo, implementación y mantenimiento de los componentes de seguridad de información embebidos en los sistemas físicos y en los sistemas de software de aplicación. Establecimiento y control de metodología de procesamiento de datos que mejoren la seguridad y privacidad incluyendo datawarehousing.
 - LS-CE1- Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
 - LS-CE6- Planificar, dirigir, realizar y/o evaluar los sistemas de seguridad en el almacenamiento y procesamiento de la información. Realizar la especificación, diseño, desarrollo, implementación y mantenimiento de los componentes de seguridad de información embebidos en los sistemas físicos y en los sistemas de software aplicados. Establecer y controlar las metodologías de procesamiento de datos orientadas a seguridad, incluyendo data-warehousing.

CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)

Introducción a Lenguajes de modelado: El lenguaje Unificado de modelado (UML), El Lenguaje para restricciones de objetos OCL. El proceso de desarrollo de software iterativo e incremental RUP, al estilo LARMAN. Uso de OCL para especificar operaciones y restricciones (invariantes, etc.) del modelo. Introducción a la filosofía del Desarrollo de software Conducido por Modelos MDD. Técnicas de Metamodelado. El estándar MOF (Meta Object Facility). Lenguajes de transformación. El estándar QVT. Herramientas para MDD. Testing basado en Modelos.

PROGRAMA ANALÍTICO

Unidad 1: Lenguajes Gráficos de modelado- Introducción

Utilidad de los modelos. Los modelos a través del proceso de desarrollo de software.
Distintas vistas del modelo a través de diferentes diagramas.
Características generales de los lenguajes de modelado gráficos.

Unidad 2: El lenguaje Unificado de modelado (UML)

- Casos de Uso
- Diagrama de Estáticos. Conceptos básicos y Avanzados
- Diagramas de Comportamiento

Unidad 3: El Lenguaje para restricciones de objetos OCL

- Sintaxis de OCL
- Uso de OCL integrado a modelos
- Ejemplos

Unidad 4: El proceso de desarrollo de software

- La filosofía y etapas del proceso de desarrollo de software
- Procesos de desarrollo iterativos e incrementales (el Proceso Unificado- RUP)
- Desarrollo de un Caso de Estudio aplicando el proceso iterativo e incremental y UML al estilo LARMAN.
- Uso de OCL para especificar operaciones y restricciones (invariantes, etc.) del modelo en general.

Unidad 5: El Desarrollo de software Conducido por Modelos

- La filosofía MDD
- Técnicas de Metamodelado
- El estándar MOF (Meta Object Facility)
- Ejemplos de transformaciones en MDD
- Anatomía de las transformaciones
- Lenguajes de transformación. -
- El estándar QVT y otros lenguajes para transformaciones
- Herramientas para MDD

Unidad 6: El Testing Basado en Modelos

- El proceso MBT y sus etapas.
- UTP: UML Testing Profile
- Modelado de Tests de Unidad en MBT

BIBLIOGRAFÍA

1. Jacobson, I.,Booch, G Rumbaugh, J., The Unified Software Development Process. Addison Wesley. ISBN 0-201-57169-2 (1999).
2. Craig Larman. UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. 2da. Edición. Prentice Hall. (2003).
3. Alistar Cockburn, Writing Effective Use Cases, Addison Wesley. (2001).
4. Stahl, T. and Völter, M. Model-Driven Software Development. John Wiley & Sons, Ltd. (2006)

5. Pons, Claudia, Giandini Roxana y Pérez, Gabriela. Desarrollo de Software Dirigido por Modelos. Conceptos teóricos y su aplicación práctica 1er.Edición. EDULP & McGraw-Hill Educacional, Argentina (2010).
6. Meta Object Facility (MOF) 2.0. OMG Adopted Specification. <http://www.omg.org>. (2005)
7. The Object Constraint Language Specification – Version 2.0, for UML 2.0, revised by the OMG, <http://www.omg.org>, April 2004.
8. The Unified Modeling Language Version2.0., OMG Final Adopted Specification. formal/2005-07-04. <http://www.omg.org> (2005).
9. The Eclipse Project <http://www.eclipse.org>
10. Kleppe, Anneke G. and Warmer Jos, and Bast, Wim. MDA Explained: The Model Driven Architecture: Practice and Promise. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.
11. Object Management Group, MDA Guide, v1.0.1, omg/03-06-01, June 2003. <http://www.omg.org>.
12. QVT 2.0 Query/View/Transformations - OMG Adopted Specification. March 2005. <http://www.omg.org>.
13. Blackburn, M., Busser, R., Nauman, A.: "Why model-based test automation is different and what you should know to get started". International Conference of Practical Software Quality & Testing (2004).
14. Baker, P. et all: Model-Driven Testing Using the UML Testing Profile. Springer-Verlag (2009)
15. Correa, N.: "Proceso de Generación de Casos de Prueba en el Contexto MDD/ MBT". Tesina de grado. Facultad de Informática. UNLP (2015)

METODOLOGÍA DE ENSEÑANZA

Con el fin de cumplir los objetivos planteados, la actividad seguirá el siguiente esquema:

En el dictado de las clases, los nuevos conceptos serán introducidos mediante el desarrollo de ejemplos que guarden relación con casos conocidos por el alumno. Se guiará al alumno en el desarrollo de un sistema a través de la cursada, utilizando las técnicas aprendidas. Con el fin de incentivar la participación del alumno y dado que los cursos no son demasiado numerosos, se formarán grupos de 2 o 3 integrantes como máximo. Las actividades prácticas se llevarán a cabo en la Sala de PC. Se utilizarán distintas herramientas de modelado. Cada grupo utilizará una herramienta diferente de código abierto para construir los modelos producto de cada etapa del desarrollo del sistema. Los distintos grupos tendrán entregas parciales obligatorias por etapa, con devolución del docente y posibilidad de re-entrega. Al final de la cursada, cada grupo hará una presentación a sus compañeros exponiendo ventajas y desventajas del uso de la herramienta de modelado asignada.

Las clases se impartirán con diapositivas. Se utilizará la sala de PC con herramientas instaladas adecuadamente. Para agilizar la comunicación con la cátedra, los alumnos tendrán acceso al material, enunciado de los trabajos prácticos, apuntes, horarios, bibliografía, novedades e información general de la materia a través del entorno web de la asignatura que mantendrán actualizada los docentes.

Descripción de la actividad curricular alternativa:

Con el fin de flexibilizar el régimen de aprobación de la materia, la propuesta de enseñanza permitirá una actividad curricular alternativa siguiendo una modalidad semi-presencial con el siguiente esquema:

Se creará un curso en un entorno virtual donde los miembros tendrán acceso a temas específicos correspondientes a las distintas unidades del programa. En cada tema se habilitarán los complementos: foro de discusión (donde se tratarán exclusivamente los contenidos de dicha unidad), y opcionalmente un Chat (donde el profesor estará disponible en un tiempo establecido, y se plantearán consignas a resolver durante la sesión). En los foros se propondrán debates, se intercambiarán ideas y se darán consignas a resolver. Se estimulará la participación de todos los alumnos a distancia.

Esta modalidad también incluye la resolución de un trabajo práctico (desarrollado en forma individual en este caso), en una herramienta de modelado especificada por la cátedra y dividido en etapas que definen entregas parciales obligatorias.

Alguna de estas entregas será presencial, mientras que en otras el alumno podrá hacer uso de la plataforma para la presentación de la entrega. En todos los casos, estas presentaciones tendrán la devolución correspondiente por parte de la cátedra y podrán ser corregidas en una próxima instancia.

Al final de la cursada, el alumno en esta modalidad, cumplirá con la entrega final obligatoria del trabajo práctico y expondrá en forma presencial a sus compañeros de cursada, ventajas y desventajas del uso de la herramienta de modelado asignada.

Adicionalmente, para esta modalidad, el alumno deberá analizar otra herramienta de modelado que le será asignada, por lo cual su exposición incluirá la comparación de funcionalidades entre las dos herramientas asignadas.

De esta manera el entorno virtual del curso quedará conformado de la siguiente manera:

1. Foro de Noticias (acceso a todos los miembros)
2. Foro para Alumnos (acceso para todos los alumnos)
3. Foro de Consultas de temas Teóricos y/o Prácticos (acceso a todos los miembros)
4. Material de teóricos y prácticos (acceso a todos los miembros)
5. Recursos adicionales por unidad, exclusivo para la modalidad semi-presencial.
 - a. Recurso: Material Teórico adicional
 - b. Recurso: Material Práctico adicional
 - c. Foro de Discusión de la Unidad (sobre los contenidos del material adicional)
 - d. Chat de la Unidad

EVALUACIÓN

Debido a que la actividad es una materia optativa, con alumnado no demasiado numeroso y avanzado en la carrera, se espera una alta participación de ellos.

Tanto los alumnos en la modalidad presencial como los semi-presenciales, al final de la cursada presencial, cumplirán con la entrega obligatoria de un trabajo práctico (individual en caso de la modalidad semi-presencial y grupal en la modalidad presencial) y comentarán, mediante una presentación oral, a sus compañeros las ventajas y desventajas del uso de la herramienta de modelado asignada. Esta actividad, para la aprobación de los trabajos prácticos, cuya aprobación final se completa mediante un coloquio, se tomará de forma presencial.

Los alumnos que opten por la alternativa semi-presencial, tendrán que analizar adicionalmente otra herramienta y presentar comparaciones entre ambas el día de la exposición.

La aprobación final de la materia se logrará mediante la resolución de un trabajo final donde el alumno aplique los conceptos adquiridos durante la cursada. Será evaluado por el profesor en forma presencial.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
1	21/03/19	El lenguaje Unificado de modelado (UML) Diagramas de Comportamiento, Casos de Uso, Diagrama de Estáticos. Conceptos básicos y Avanzados. Ejercitación
2	28/03/19	El lenguaje Unificado de modelado (UML) Consulta y resolución de Ejercicios Prácticos
3	4/04/19	Lenguajes Gráficos de modelado- Introducción Utilidad de los modelos. Los modelos a través del proceso de desarrollo de software. Distintas vistas del modelo a través de diferentes diagramas. Características generales de los lenguajes de modelado gráficos.
4	11/04/19	El proceso de desarrollo de software La filosofía y etapas del proceso de desarrollo de software Procesos de desarrollo iterativos e incrementales (el Proceso Unificado- RUP) Resolución Trabajo Práctico RUP
5	25/04/19	Desarrollo de un Caso de Estudio aplicando el proceso iterativo e incremental y UML al estilo LARMAN. Uso de OCL para especificar operaciones y restricciones (invariantes, etc.) del modelo en general. Resolución Trabajo Práctico RUP (cont.)
6	02/05/19	El Lenguaje para restricciones de objetos OCL Sintaxis de OCL Uso de OCL integrado a modelos Ejemplos, Ejercicios Prácticos
7	9/05/19	El Desarrollo de software Conducido por Modelos La filosofía MDD. El estándar MOF (Meta Object Facility), Ejemplos de transformaciones en MDD.
8	16/05/19	Testing Basado en Modelos (TBM). Generación de modelos de testing: diagramas de estado, de actividades, de interacción, de objetos.
9	23/05/19	Presentación y uso de Herramientas para MDD
10	30/05/19	El Metamodelado en Ingeniería de Software Técnicas de Metamodelado Resolución Trabajo Práctico Metamodelado
11	06/06/19	Resolución Trabajo Práctico RUP
12	13/06/19	Resolución Trabajo Práctico RUP (cont.)/ Resolución Trabajo Práctico RUP incorporando MBT.

Evaluaciones previstas	Fecha
Evaluación/ Presentaciones de trabajos grupales	20/06/19



Evaluación (2da fecha)	27/06/19
Evaluación (3ra fecha)	11/07/19

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

Profesora: Roxana Giandini

giandini@info.unlp.edu.ar

Ayudante: Natalia Correa

natalia.correa@lifa.info.unlp.edu.ar

Plataforma virtual Moodle: <https://catedras.info.unlp.edu.ar/> (el curso se denomina "Desarrollo De Software Basado En Modelos")

Firma del/los profesor/es