



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Programa de Taller de Ingeniería Dirigida por Modelos

1. NOMBRE DE LA UNIDAD CURRICULAR

Taller de Ingeniería Dirigida por Modelos

2. CRÉDITOS

8 créditos

3. OBJETIVOS DE LA UNIDAD CURRICULAR

La Ingeniería Dirigida por Modelos (Model Driven Engineering, MDE) es un paradigma de Ingeniería de Software que jerarquiza el modelado como principal actividad del ciclo de vida de un sistema de software (construcción, mantenimiento, ingeniería inversa, etc.) El paradigma propone la construcción de modelos (abstracciones) de diferentes aspectos de un sistema y la transformación de dichos modelos de forma (semi)automática.

Varios términos vinculados han surgido a lo largo de los años, como Computer-Aided Software Engineering (CASE) en los 90s y, más recientemente, los términos Low-code y No-code. Todos ellos persiguen la idea de reducir la codificación manual de un sistema, a través de combinar técnicas de desarrollo visual (modelos) con la generación de código (un tipo particular de transformación). De esta forma, se apunta a reducir errores en el proceso de ingeniería de software al aumentar el nivel de abstracción en la especificación del sistema y posibilitar la verificación y reuso de los modelos y las transformaciones. Asimismo, se enfoca en aumentar la productividad reduciendo tiempos de desarrollo a través de mecanismos automáticos de construcción.

El objetivo de esta unidad curricular es brindar una visión general de MDE describiendo sus fundamentos, técnicas, y herramientas para su aplicación práctica.

Al finalizar el curso el estudiante será capaz de:

- Identificar la aplicación de acciones relacionadas con MDE en diversos contextos
- Describir los principios y técnicas básicas de MDE
- Especificar modelos y transformaciones sencillas utilizando herramientas disponibles
- Discutir los beneficios y limitaciones de la aplicación de MDE
- Elaborar una propuesta de aplicación de MDE

4. METODOLOGÍA DE ENSEÑANZA

La unidad tiene una modalidad de dictado de taller con realización de ejercicios y trabajos de laboratorio tanto de forma individual como grupal. Habrá entre uno y dos encuentros semanales que combinarán exposiciones teóricas con trabajo en máquina y actividades grupales. Además, se estimulará la investigación, la lectura de artículos provistos, y la discusión y reflexión sobre las problemáticas presentadas. Se busca involucrar a los participantes en discusiones activas, por ejemplo a través de la entrega de ejercicios o controles de lectura domiciliarios y posterior discusión en clase.

Se desarrollarán proyectos grupales con entregas obligatorias que pondrá en práctica los conceptos vistos en el curso. A cada grupo se le realizará un seguimiento y eventual evaluación por parte de los docentes. Los proyectos grupales serán defendidos. Las entregas y defensas realizadas durante la asignatura tendrán un puntaje asignado y un nivel de suficiencia definido.

Se incluye una discriminación de dedicación en horas:

- Horas clase (teórico): 12
- Horas clase (práctico): 6
- Horas clase (laboratorio): 8
- Horas consulta: incluidas en teórico y práctico
- Horas evaluación: incluidas en teórico (control de lectura, defensas, etc.)

Subtotal horas presenciales: 26

- Horas estudio: 36
- Horas resolución ejercicios/prácticos: incluidas en teórico
- Horas proyecto final/monografía: 60

Total de horas de dedicación del estudiante: 122 hs

5. TEMARIO

1. Conceptos básicos de MDE: Se presenta en amplitud el paradigma, introduciendo los conceptos básicos que serán profundizados a lo largo del curso, particularmente:
 - a. Modelado y desarrollo de software
 - b. Objetivos de MDE
 - c. Elementos básicos de MDE
 - d. Ejemplos de adopción de MDE

2. Modelos y Metamodelos: Se profundiza en los conceptos de modelo y metamodelo, centrales para la aplicación del paradigma, particularmente:
 - a. Lenguajes de modelado de propósito general y de dominio específico
 - b. Definición de metamodelos: sintaxis (gráfica y textual) y semántica
 - c. Herramientas de soporte al modelado y metamodelado

3. Transformaciones: Se profundiza en el concepto de transformación de modelos, central para la aplicación del paradigma, particularmente:
 - a. Clasificación de enfoques de transformación
 - b. Transformaciones de modelo a modelo (M2M)
 - c. Transformaciones de modelo a texto (M2T)
 - d. Lenguajes y herramientas para la realización de transformaciones

4. Práctica de MDE: Se presentan aspectos prácticos a tener en cuenta para la aplicación de MDE en proyectos reales, particularmente:
 - a. Arquitectura Dirigida por Modelos (MDA)
 - b. Integración de MDE al proceso de desarrollo
 - c. Gestión de modelos (repositorios, evolución, calidad)

6. BIBLIOGRAFÍA

Tema	Básica	Complementaria
Conceptos básicos de MDE	(1)(2)	(5)(6)
Modelos y Metamodelos	(1)(3)	(5)(6)(7)
Transformaciones	(1)(4)	(5)(6)
Práctica de MDE	(1)	(5)(6)(8)

6.1 Básica

1. Brambilla, Marco; Cabot, Jordi; Wimmer, Manuel (2012). Model-Driven Software Engineering in Practice. Morgan Claypool. ISBN: 9781608458820. [Recursos parciales disponibles en la web]
2. Schmidt, Douglas (2006). Guest editor's introduction: Model-Driven Engineering. IEEE Computer, 39(2):25–31. [Disponible en TIMBO]
3. Clark, Tony; Evans, Andy (2015). Applied Metamodelling: A Foundation for Language Driven Development. CoRR abs/1505.00149. [Acceso abierto en la web]
4. Czarnecki, Krzysztof; Helsen, Simon (2006). Feature-based survey of model transformation approaches. IBM Systems Journal, 45(3):621–646. [Disponible en TIMBO]

6.2 Complementaria

5. Pons, Claudia; Giandini, Roxana; Perez, Gabriela (2010). Desarrollo de software dirigido por modelos: Conceptos teóricos y su aplicación práctica. McGraw-Hill. ISBN: 9789503406304. [Acceso abierto en la web]
6. Durán, Francisco; Troya, Javier; Vallecillo, Antonio (2013). Desarrollo de software dirigido por modelos. Universidad Oberta de Catalunya. CC-BY-NC-ND, PID_00184466. [Acceso abierto en la web]
7. Bézin Jean (2005). On the unification power of models. Software and System Modeling, 4(2):171–188. [Disponible en TIMBO]
8. OMG (2003). MDA guide version 1.0.1. Technical report, Object Management Group. [Acceso abierto en la web]

7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS

7.1 Conocimientos Previos Exigidos: Conocimientos básicos de desarrollo de software y modelado de sistemas.

7.2 Conocimientos Previos Recomendados: Conocimientos básicos de ingeniería de software

No incluye la información de previaturas. Las unidades curriculares previas serán definidas por cada carrera que tome la unidad curricular y serán incluidas en el anexo B.

ANEXO A

Para todas las Carreras

A1) INSTITUTO

Instituto de Computación

A2) CRONOGRAMA TENTATIVO

Cronograma de avance semanal con detalle de las horas de clase asignadas a cada tema.

Semana 1	Conceptos básicos de MDE (4 hs)
Semana 2	Modelos y Metamodelos (4 hs)
Semana 3	Transformaciones M2M (4 hs)
Semana 4	Transformaciones M2T (4 hs)
Semana 5	Proyecto grupal (Monitoreo 1hs)
Semana 6	Proyecto grupal (Monitoreo 1hs)
Semana 7	Proyecto grupal (Monitoreo 1hs)
Semana 8	Proyecto grupal (Monitoreo 1hs)
Semana 9	Proyecto grupal (Monitoreo 1hs)
Semana 10	Proyecto grupal (Monitoreo 1hs)
Semana 11	Proyecto grupal (Monitoreo 1hs)
Semana 12	Proyecto grupal (Monitoreo 1hs)
Semana 13	Práctica de MDE (2 hs)

A3) MODALIDAD DEL CURSO Y PROCEDIMIENTO DE EVALUACIÓN

Si bien el taller es principalmente grupal, la evaluación podrá realizarse individualmente en caso de ser necesario. En este caso, se podrá determinar la insuficiencia del trabajo de un estudiante en base a los resultados de las evaluaciones grupales, al seguimiento del grupo que realizan los docentes y las evaluaciones individuales.

El procedimiento de evaluación

- Asistencia a clase (10 %)
- Participación activa en actividades de discusión y ejercicios (20 %)
- Realización de trabajos grupales (70 %)

Para la aprobación final del curso se requiere un mínimo de 60% de los puntos en cada parte y un mínimo de 60% en el total.

A4) CALIDAD DE LIBRE

Este curso no adhiere a la Calidad de Libre.

A5) CUPOS DE LA UNIDAD CURRICULAR

No tiene

ANEXO B para las carreras Ingeniería en Computación (plan 97) y Licenciatura en Computación

B1) ÁREA DE FORMACIÓN

Ingeniería de Software

B2) UNIDADES CURRICULARES PREVIAS

Para el Curso: Curso aprobado de Taller de Programación

Para el Examen: No Aplica

APROB RES CONSEJO DE FAC. ING.

Fecha 2/8/22 Exp. 061130-000091-22