
MAESTRÍA EN INGENIERÍA EN SISTEMAS DE INFORMACIÓN
ESPECIALIZACIÓN EN INGENIERÍA EN SISTEMAS DE INFORMACIÓN

1. Nombre de la actividad curricular: Modelado Conceptual de Sistemas de Información

2. Año Académico: 2020

3. Docente: Silvio Gonnet

4. Fundamentación

Los modelos conceptuales de información capturan conocimiento sobre un universo de discurso en un nivel de abstracción mas cercano a la conceptualización humana del dominio que a la implementación computacional. Esta captura se realiza mediante la descripción de las funcionalidades del sistema de información a desarrollar de manera completa y precisa, independiente de particularidades tecnológicas del diseño y la implementación. Los modelos conceptuales se desarrollan empleando un lenguaje de modelado, el cual es especificado mediante un meta-modelo.

El curso presenta al alumno en profundidad los conceptos, técnicas y métodos modernos de modelado conceptual de sistemas de información, abordando diferentes patrones conceptuales a partir de los cuales es posible crear modelos de los sistemas de información. En particular se aborda el lenguaje de modelado UML, incluyendo el lenguaje de especificación de restricciones OCL ("Object Constraint Language") para la representación de las reglas de negocio y las restricciones del problema.

5. Objetivos

Este curso presenta el estado del arte para la investigación y práctica en la disciplina modelado conceptual, siendo el objetivo general del mismo que los alumnos adquieran habilidades en la construcción y validación de modelos conceptuales de sistemas de información.

Objetivos específicos:

- Desarrollar estado del arte en modelado conceptual, distintos tipos de lenguajes de modelado, representación e ingeniería de conocimiento.
- Desarrollar los elementos necesarios y principios de modelado para la construcción de un modelo conceptual, aplicando los lenguajes UML+OCL.
- Abordar diferentes patrones conceptuales a partir de los cuales es posible crear modelos de los sistemas de información.
- Validar correctitud y completitud de los modelos conceptuales, evitando consideraciones de diseño e implementación.
- Asociar cada uno de los patrones conceptuales (espacio del problema) con representaciones de software asociadas (espacio de la solución).
- Desarrollar capacidades de comparación crítica de lenguajes específicos ya existentes.

6. Contenidos

Contenidos mínimos:

Antecedentes del modelado conceptual en el desarrollo de sistemas de información.

Modelado Estructural. Tipo de entidades. Tipos de relaciones. Restricciones de cardinalidad. Reificación. Tipos derivados. Restricciones de integridad. Taxonomías.

Modelado dinámico. Eventos, restricciones, efectos. Máquinas de estados. Diagramas de actividades. Casos de Uso.

Desarrollo de metamodelos.

Contenidos analíticos:

1 – Introducción al Modelado Conceptual de Sistemas de Información:

¿Qué es el modelado de información? ¿Qué son los modelos conceptuales? Historia de los modelos conceptuales. Primeros lenguajes de modelado. Introducción a los lenguajes actuales. Lenguajes formales e informales. Diferentes empleos de los modelos conceptuales. UML. OCL.

2 – Modelado Estructural

Tipos de entidades: teorías de concepto y clasificación, diseño de conceptos, clasificación y tipo de entidades, significado de los tipos de datos. Tipos de relaciones. Restricciones de cardinalidad. Tipos particulares de relaciones. Reificación. Tipos de relaciones genéricas: representación, relaciones parte-todo, relaciones de agrupamiento, relaciones de materialización, roles. Tipos derivados. Restricciones de integridad. Taxonomías: especialización, especialización restringida, generalización, refinamiento, especialización de tipos de relaciones.

3 – Modelado Dinámico

Eventos, restricciones, efectos.

4 – Metamodelado

Tipos de meta-entidades: nivel de clasificación, relaciones “instance-of” e “isA”. Powertypes. Tipos de relaciones “clase” y “meta”. Meta-esquemas. MOF como lenguaje de modelado conceptual. Metamodelado con Eclipse.

7. Metodología de Enseñanza y Formación práctica

El curso se llevará a cabo en 12 clases de 5 horas cada una. 8 clases serán del tipo teórico-práctico, en donde se expondrán los conceptos teóricos y se realizará la resolución de cuatro guías de problemas con el propósito de afianzar los conocimientos.

Cuatro clases serán prácticas y se efectuarán en un laboratorio de computadoras donde se resolverán problemas de modelado de aspectos estáticos, y aspectos dinámicos, especificación de restricciones, y de especificación de un meta-modelo. Durante el laboratorio se emplearán las herramientas USE (<https://sourceforge.net/projects/useocl/>) y Eclipse (<http://www.eclipse.org>).

Durante el cursado se realizarán dos trabajos prácticos. El primer trabajo práctico aborda el modelado estático y dinámico de un dominio empleando el lenguaje UML y el lenguaje de restricciones OCL. El segundo trabajo práctico consiste en la especificación de un lenguaje en particular y su utilización empleando Ecore de Eclipse.

8. Carga horaria total

Carga horaria teórica	Carga horaria práctica	Carga horaria total
40	20	60

9. Modalidad de Evaluación

La evaluación se realizará por medio de dos trabajos prácticos y un examen escrito integrador. Los trabajos prácticos y el examen son de carácter individual. Para la aprobación del curso se requerirá que los alumnos obtengan una ponderación mínima del 60% en cada instancia de evaluación.

10. Requisitos de aprobación y promoción

La calificación se expresará en escala numérica de cero (0) a diez (10) sin decimales. Para la promoción se requerirá la norma mínima de siete (7). (Extraído de la Ordenanza N° 1313)

11. Infraestructura y equipamiento

De acuerdo con la formación prevista en la actividad curricular, indicar el/los ámbitos en los que se desarrolla, mencionando el principal equipamiento a emplear.

12. Bibliografía

Balaban, M., Maraee, A. (2019). Removing redundant multiplicity constraints in UML class models. *Software & Systems Modeling*, 18, 2717–2751.

Brambilla, M., Cabot, J., y Wimmer, M. (2017). *Model-Driven Software Engineering in Practice*, second edition. Morgan & Claypool Publishers.

Gogolla, M., Büttner, F., Richters, M. (2007). USE: A UML-Based Specification Environment for Validating UML and OCL. *Science of Computer Programming*, 69, 27-34.

Gogolla, M., y Doan, K-H (2017). Quality Improvement of Conceptual UML and OCL Schemata through Model Validation and Verification. En J. Cabot, C. Gomez, O. Pastor, M. Ribera Sancho, y E. Teniente (Eds.), *Conceptual Modeling Perspectives* (pp. 155-168). Springer.

Guizzardi, G. (2005). *Ontological Foundations for Structural Conceptual Models*. Universal Press.

Jeusfeld, M. (2009). Metamodeling for method engineering with ConceptBase. En M. Jeusfeld, M., J. Mylopoulos (Eds.), *Metamodeling for Method Engineering*. The MIT press.

Object Constraint Language™ (OCL™), Version 2.4, 2014, disponible en: <http://www.omg.org/spec/OCL/2.4/>.

Olivé, A. (2005). Conceptual Schema-Centric Development: A Grand Challenge for Information Systems Research. *Advanced Information Systems Engineering, Lecture Notes in Computer Science*, 3520, 59-77.

Olivé, A. (2007). *Conceptual Modeling of Information Systems*. Springer-Verlag.

Pérez, B., Porres, I. (2019). Reasoning about UML/OCL class diagrams using constraint logic programming and formula. *Information Systems*, 81, 152-177.

Unified Modeling Language™ (UML®) Version 2.5.1, 2017, disponible en: <http://www.uml.org/>.

Wahler, M., Basin, D., Brucker, A., Koehler, J. (2010). Efficient analysis of pattern-based constraint specifications. *Software & Systems Modeling*, 9, 225–255.