

**Nombre:** Bibliotecas digitales

**Créditos:** 2.5

**Semestre:** A

**Objetivos:**

- Clarificar la noción de BiD y sus implicaciones
- Presentar los modelos más aceptados de BiD
- Introducir los principales aspectos a tener en cuenta en el desarrollo de Bibliotecas Digitales Multimedia, incluyendo, entre otros:
  - mecanismos de publicación electrónica
  - arquitecturas de BiD
  - almacenamiento y recuperación de información
  - interoperabilidad
  - catalogación, conservación
- Revisar el estado del arte en investigación y desarrollo en el campo

**Contenidos:**

PARTE 1. Fundamentos

- Introducción a las Bibliotecas Digitales
- Arquitecturas
- Metadatos
- Búsqueda y recuperación de información
- Recuperación de información distribuida: interoperabilidad y búsquedas en la Web
- Conservación a largo plazo de los contenidos digitales
- Estado del arte en la investigación sobre Bibliotecas Digitales

PARTE 2. Temas de investigación del Grupo ISSI

- Gestión de bibliografía: Bibshare
- Sistemas de Gestión de Emergencias
- Otros temas relacionados

**Metodología de enseñanza y aprendizaje:** clases teóricas, con participación de los alumnos. Uso de recursos en red.

**Criterios y procedimientos de evaluación:** la evaluación consistirá en el desarrollo de un examen teórico, más la presentación individual de un tema relacionado con la materia, elegido por consenso entre el profesor y el alumno.

**Bibliografía relevante y actualizada:**

- Lesk, Michael. Understanding Digital Libraries, 2nd edition. Morgan Kaufman, 2005.
- Lesk, Michael. Practical Digital Libraries: books, bytes & bucks. Morgan Kaufman, 1997.
- Arms, William Y. Digital Libraries. MIT Press, 2000.
- Borgman, Christine L. From Gutenberg to the Global Information Infrastructure. MIT Press, 2000.
- Stefik, Mark. Internet Dreams. MIT press, 1996.
- Baeza, R. and Ribeiro, B. Modern Information Retrieval. Addison Wesley, 1999.

**Nombre:** Desarrollo de aplicaciones en Java

**Créditos:** 2.5

**Semestre:** B

**Objetivos:** El lenguaje de programación Java es uno de los lenguajes de programación modernos, orientado a objetos, que está llamado a tener una importancia fundamental, presente y futura, tanto en el desarrollo de software y de aplicaciones complejas como en el de las futuras plataformas de programación. En este contexto, el curso aborda como principal objetivo, desde un enfoque pragmático, el diseño, programación y desarrollo (en un entorno de programas genéricos, ampliamente reutilizables), dentro de la plataforma Java, de sistemas informáticos integrales, entendiendo por éstos aquellos sistemas (multihilo) que operen de un modo global en contextos multimedia con interconexiones remotas seguras a otros sistemas y configuraciones informativas distribuidas, en el entorno de Internet, con las adecuadas interfaces gráficas de usuario.

**Contenidos:**

- Tema 1: Introducción a la plataforma Java. Generalidades.
- Tema 2: Lenguaje de programación Java. Introducción a Eclipse.
- Tema 3: Tipos de datos avanzados: Colecciones parametrizadas.
- Tema 4: Programación multihilo.
- Tema 5: Entradas y salidas.
- Tema 6: Serialización y persistencia de objetos.
- Tema 7: Interconectividad.
- Tema 8: Programación de sistemas distribuidos: RMI.
- Tema 9: Interfaces gráficas de usuario I: AWT.
- Tema 10: Interfaces gráficas de usuario II: Swing.
- Tema 11: Applets.
- Tema 12: Beans.
- Tema 13: Introducción a Ant y Maven.
- Tema 14: Introducción a los servlets, JSPs y Enterprise Java Beans.

**Metodología de enseñanza y aprendizaje:** De cada tema se realizará una exposición de sus características más importantes y se suministrarán unas notas completas que lo amplíen convenientemente, dejando para el horario de consultas y tutorías las posibles cuestiones, dudas, aclaraciones y demás pormenores que puedan surgir tanto en el estudio de la materia presentada como en la resolución de los ejercicios propuestos.

**Criterios y procedimientos de evaluación:** A partir de los ejercicios y trabajos propuestos durante el curso.

**Bibliografía relevante y actualizada:**

- Tutorial de Java (J2SE): <http://java.sun.com/docs/books/tutorial/>
- Tutorial de J2EE: <http://java.sun.com/javaee/reference/tutorials/index.jsp>
- B. Eckel, Thinking in Java, Prentice-Hall, 2006.
- J. Gosling, B. Joy, G. Steele, G. Bracha The Java language specification, Addison-Wesley, 2005.
- C.S. Horstmann, G. Cornell, Core Java Vol. 1 y Vol. 2, Prentice Hall, 2005.
- K. Mukhar, C. Zelenak, J.L: Weaver, J. Crume, Beginning Java EE 5. From novice to professional, Apress, 2006.
- J. Tilly, E.M. Burke, Ant. The Definitive Guide, O'92>reilly, 2002.

**Nombre:** Descripción, simulación y verificación de sistemas concurrentes

**Créditos:** 2.5

**Semestre:** A

**Objetivos:**

El curso plantea la presentación de diversos formalismos, tanto gráficos como textuales, que permiten la descripción de sistemas concurrentes, entendidos como sistemas de procesos que interactúan y se comunican entre sí.

La primera parte está centrada en las redes de Petri y sus diversas extensiones, de gran aplicabilidad y uso en el mundo de la computación. En su desarrollo, se incide en los métodos de verificación de propiedades habituales sobre las mismas, así como la presentación de diversas herramientas de simulación y verificación automática de propiedades que existen en la literatura.

A continuación, en la segunda parte del curso, se muestran en detalle diferentes álgebras de procesos como otros posibles modelos para la descripción de la concurrencia. En concreto, se presentan el lenguaje LOTOS, CCS, CSP, el cálculo pi y la Máquina Química Abstracta, entre otros. Una vez conocidos los distintos formalismos para describir sistemas concurrentes, se presentarán las lógicas usadas en la actualidad para la verificación de este tipo de sistemas.

Por último, se plantean propuestas con redes de Petri y álgebras de procesos aplicadas a la Biocomputación y a la Computación Ubicua.

**Contenidos:**

1. Redes de Petri

- 1.1. Introducción / Historia / Aplicaciones
- 1.2. Definición / Representación gráfica y matricial
- 1.3. Conceptos dinámicos
- 1.4. Interpretación / Ejemplos de modelización
- 1.5. Propiedades de comportamiento y estructurales
- 1.6. Métodos de análisis
- 1.7. Clases de redes de Petri
- 1.8. Herramientas de simulación y verificación

2. Álgebras de Procesos

- 2.1. CCS
- 2.2. CSP
- 2.3. LOTOS
- 2.4. Máquina Química Abstracta
- 2.5. Cálculo PI
- 2.6. Verificación

3. Otras propuestas

**Metodología de enseñanza y aprendizaje:** Clases magistrales combinadas con resolución de problemas.

**Criterios y procedimientos de evaluación:** Desarrollo y/o presentación de un trabajo relacionado con la materia impartida.

**Bibliografía relevante y actualizada:**

- Tadao Murata. Petri nets: Properties, Analysis and Applications. Proc. of the IEEE, 77(4), 1989.
- James L. Peterson. Petri Net Theory and the Modeling of Systems. Prentice-Hall, 1981.
- <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>
- <http://lamp.epfl.ch/mobility/>

**Nombre:** Extracción automática de conocimiento en bases de datos e ingeniería del software

**Créditos:** 2.5

**Semestre:** B

**Objetivos:** Ante el problema de los grandes volúmenes de información se plantea la necesidad de nuevas técnicas y herramientas para un análisis inteligente y automático de la información. La extracción de conocimiento en bases de datos se basa en técnicas inductivas y de aprendizaje automático. Esta parte del proceso se conoce como prospección de datos (data-mining). La programación lógica inductiva (ILP) permite abordar este proceso con un nivel de automatización mucho mayor ya que se minimiza el preprocesamiento de datos y la interpretación de los resultados, debido a la expresividad y la comprensibilidad del conocimiento extraído. Estas características permiten extender la extracción automática de conocimiento a otras fuentes de información no estructurada (web) y al desarrollo de sistemas software diseñados para el tratamiento de la información.

**Contenidos:**

1. Introducción.

- 1.1. El Problema de la Extracción Automática de Conocimiento.
- 1.2. Técnicas de Aprendizaje Automático.
- 1.3. Evaluación de Hipótesis

2. Minería de Datos y Extracción de Conocimiento de Bases de Datos

- 2.1. Nuevas Necesidades del Análisis de Grandes Volúmenes de Datos. Los Data-Warehouses y el KDD.
- 2.2. El Proceso de Extracción de Conocimiento de Bases de Datos.
- 2.3. Métodos Específicos de Prospección de Datos.

3. Extracción de Conocimiento a Partir de Información No Estructurada

- 3.1. Los Problemas de la Extracción de Conocimiento de Información No Estructurada.
- 3.2. Extracción de Conocimiento a partir de Documentos No Estructurados (Text Mining, Web Content Mining y Web Structure Mining).
- 3.3. XML, DTDs y Consultas. Extracción de Conocimiento a partir de Información Semi-Estructurada

(XML).

- 3.4. Intercambio de Conocimiento (PMML y RuleML).
- 3.5. Extracción de Conocimiento a partir de Patrones de Uso (Web Usage Mining).
- 3.6. Personalización y Asistentes (Web) "Inteligentes".

4. Programación Inductiva e Ingeniería del Software

- 4.1. La Inducción en la Generación de Software.
- 4.2. Aplicaciones del Aprendizaje Automático para el Desarrollo de Software.
- 4.3. Inducción Automática de Requerimientos.
- 4.4. Modelado (Semi-)Automático.
- 4.5. Aplicaciones a Otras Fases del Ciclo de Vida.

**Metodología de enseñanza y aprendizaje:** Lección magistral, aprendizaje autónomo, y resolución de ejercicios prácticos dirigidos por el profesor.

**Criterios y procedimientos de evaluación:** Presentación de un resumen de un trabajo de investigación del área.

**Bibliografía relevante y actualizada:**

- J. Hernández, M. J. Ramírez, C. Ferri "Introducción a la Minería de Datos" Prentice Hall Addison-Wesley, ISBN 84 205 4091 9
- Tom Mitchell, Machine Learning, , McGraw Hill, 1997.
- Ian Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations (2000), ISBN 1-55860-552-5
- Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, 2<sup>nd</sup> ed. Morgan Kaufmann, March 2006. ISBN 1-55860-901-6

**Nombre:** Fundamentos lógicos de la ingeniería del software y los sistemas de información

**Créditos:** 3

**Semestre:** A

**Objetivos:**

- Estudiar los fundamentos de las técnicas de transformación de programas.
- Estudiar técnicas de especialización de programas, con especial énfasis en la evaluación parcial.
- Presentar el problema del aprendizaje a partir de datos almacenados en varias tablas o relaciones.
- Estudiar el paradigma de la Programación Lógica Inductiva (ILP), sus técnicas así como presentar diferentes sistemas ILP.
- Extender el paradigma ILP para tratar con funciones.
- Presentar las extensiones de otras técnicas de aprendizaje para abordar el tratamiento de información estructurada.

**Contenidos:**

1. Introducción.
2. Transformación de programas.
  - 2.1. Conceptos básicos.
  - 2.2. Revisión de las principales técnicas.
  - 2.3. Evaluación parcial.
  - 2.4. Transformación basada en plegado/desplegado.
  - 2.5. Otras técnicas
3. Programación Lógica Inductiva.
  - 3.1. Introducción a la programación lógica.
  - 3.2. Aprendizaje inductivo de conceptos.
  - 3.3. Búsqueda de las cláusulas del programa.
  - 3.4. Métodos bottom-up.
  - 3.5. Métodos top-down.
  - 3.6. Bias inductivos.
  - 3.7. Sistemas ILP.
  - 3.8. Aprendizaje multirelacional.
  - 3.9. Programación Lógico-funcional inductiva.
  - 3.10. Aplicaciones

**Metodología de enseñanza y aprendizaje:** Exposición de contenidos mediante presentación o explicación por parte del profesor; resolución de problemas reales por parte de los alumnos organizados en pequeños grupos y bajo la supervisión del profesor; conjunto de pruebas escritas, orales, proyectos/trabajos, etc. utilizados para la evaluación del progreso del estudiante.

**Criterios y procedimientos de evaluación:** Examen escrito tipo test para evaluar los conceptos generales de la asignatura; resolución a nivel individual de problemas o trabajos teóricos o prácticos propuestos por el profesor.

**Bibliografía relevante y actualizada:**

- Neil D. Jones, Carsten K. Gomard, Peter Sestoft. *Partial evaluation and automatic program.* Prentice-Hall. 2003.
- Leon Sterling, Ehud Shapiro. *The art of Prolog : Advanced programming techniques.* The MIT Press. 1994.
- Simon Thompson. *Haskell : the craft of functional programming.* Addison-Wesley. 1996.
- Shan-Hwei Nienhuys-Cheng and Ronald de Wolf. *Foundations of inductive logic programming.* Springer Verlag. 1997.
- Francesco Bergadano, Daniele Gunetti. *Inductive logic programming : from machine learning to software engineering.* The MIT Press. 1995.
- S. Muggleton and L. De Raedt. *An Inductive Logic Programming: Theory and Methods*, Journal of Logic Programming 19,20:629-679, 1994.
- S. Muggleton, editor. *Inductive Logic Programming*, Vol. 38 of APIC Series, Academic Press, 1992.
- N. Lavrac and S. Dzeroski. *Inductive Logic Programming: Techniques and Applications.* Ellis Horwood, New York, 1994.

**Nombre:** Gestión de modelos

**Créditos:** 2.5

**Semestre:** B

**Objetivos:**

Presentar los Modelos como los artefactos software adecuados para desarrollar, recuperar, migrar, reutilizar... software. En particular dentro del marco OMG y MDA. Un álgebra de modelos permite resolver problemas difíciles de Ingeniería del Software con elegancia a un elevado nivel de abstracción. En la Asignatura se presenta además nuestra aproximación MOMENT a la Gestión de Modelos comparándola con otras aproximaciones

**Contenidos:**

- 1- Ingeniería del Software con Modelos
- 2- Aproximación OMG-MDA
- 3- Compilación de Modelos
- 4- Evolución del Software con Modelos
- 5- Recuperación de software Legado
- 6- Migración de datos
- 7- MOMENT

**Metodología de enseñanza y aprendizaje:** Clases magistrales acompañadas de trabajos de los alumnos de presentación y discusión de distintas aproximaciones a la Gestión de Modelos.

**Criterios y procedimientos de evaluación:** Los criterios son la comprensión operativa de las ideas expuestas y su aplicación en el contexto de los trabajos de los alumnos. La evaluación se realiza mediante el desarrollo, presentación y discusión de un trabajo del alumno en el contexto de la clase.

**Bibliografía relevante y actualizada:**

Initial:

- A. Boronat, J. Á. Carsí, I. Ramos: Una plataforma semántica para la gestión de modelos. Jornadas de Ingeniería del Software y Bases de Datos, JISBD 2003. Alicante, noviembre 2003.
- A. Boronat, J. Pérez, J. Á. Carsí, I. Ramos: Two experiences in software dynamics. Journal of Universal Science Computer. Special issue on Breakthroughs and Challenges in Software Engineering, 2005 [http://www.jucs.org/jucs\\_10\\_4/two\\_experiences\\_in\\_software](http://www.jucs.org/jucs_10_4/two_experiences_in_software).

Visio:

- A. Boronat, Julian Pedrós, J. Á. Carsí, I. Ramos: Una Arquitectura para la Definición de Metáforas Gráficas para Metamodelos. Jornadas de Ingeniería del Software y Bases de Datos, JISBD 2004. Málaga, 10-12 noviembre 2004
- A. Boronat, Julian Pedrós, J. Á. Carsí, I. Ramos: Una Arquitectura para la Definición de Metáforas Gráficas para Metamodelos. Revista IEEE América Latina, Special Edition - JISBD'2004 - IX Jornadas de Ingeniería del Software y Bases de Datos

F# (transformations)

- A. Boronat, I. Ramos, J. Á. Carsí. Automatic Model Generation in Model Management. LNCS. Intelligent Information Technology: 7th International Conference on Information Technology, CIT 2004, Hyderabad, India, December 20-23, 2004. Proceedings
- Artur Boronat, José Á. Carsí, Isidro Ramos, Julián Pedrós. An Approach for Cross-Model Semantic Transformation on the .NET Framework. 3rd International Conference on .NET Technologies. May 30 - June 1, 2005. Plzen (Pilsen), Czech Republic

MOMENT

- A. Boronat, I. Ramos, J. Á. Carsí: An Algebraic Baseline for Automatic Transformations in MDA. Software Evolution Through Transformations: Model-based vs. Implementation-level Solutions Workshop (SETra'04), Second International Conference on Graph Transformation (ICGT2004), Electronic Notes in Theoretical Computer Science (ENTCS), Rome (Italy)
- A. Boronat, J. Á. Carsí, I. Ramos: Automatic Reengineering in MDA Using Rewriting Logic as Transformation Engine. IEEE Computer Society Press. 9th European Conference on Software Maintenance and Reengineering, March 21 - 23, 2005. The Manchester Conference Centre, Manchester, UK.
- Artur Boronat, Isidro Ramos, José Á. Carsí. Is Mapping-Based Model Merging a Suitable Approach for Integrating Models from a Generic Point of View? First Conference on Algebra and Coalgebra in Computer Science. LNCS. (pendiente)
- Artur Boronat, José Á. Carsí, Isidro Ramos, Patricio Letelier: Generic Model Merging Applied to Class Diagram Integration. 16th International Conference on Database and Expert Systems Applications. LNCS (pendiente)
- Artur Boronat, José Á. Carsí, Isidro Ramos: Automatic Support for Traceability in a Generic Model Management Framework. ACM / IEEE 8th International Conference on Model Driven Engineering Languages and Systems. LNCS. (pendiente)

**Nombre:** Ingeniería del lenguaje natural

**Créditos:** 2.5

**Semestre:** A

**Objetivos:**

El alumno conocerá la problemática del Procesamiento del Lenguaje Natural (PLN): los tipos de ambigüedad, cómo abordar los distintos fenómenos lingüísticos y los problemas de eficiencia.

Aprenderá las técnicas básicas del tratamiento automático del Lenguaje Natural, la utilización de formalismos gramaticales, los algoritmos de análisis clásicos y los mecanismos para representar la semántica oracional.

El alumno aprenderá el manejo de herramientas de PLN existentes para las distintas fases de análisis.

Asimismo, conocerá las aplicaciones de PLN más usuales: aplicaciones de diálogo hombre-máquina, como el acceso a bases de conocimiento, aplicaciones de tratamiento de grandes volúmenes de información textual, como recuperación de información, minería de textos o búsqueda de respuestas, y la aplicación del PLN en fases iniciales del desarrollo del software como el modelado conceptual.

**Contenidos:**

PARTE I: Técnicas de PLN

Tema 1. Introducción al PLN.

Tema 2. Tipos de Ambigüedad.

Tema 3. Análisis Léxico.

Tema 4. Análisis Sintáctico.

Tema 5. Interpretación Semántica.

Tema 6. Interpretación Contextual.

PARTE II: Aplicaciones

Tema 7. Recuperación de Información.

Tema 8. Minería de textos. Extracción de información.

Tema 9. Búsqueda de respuestas.

Tema 10. Modelado conceptual.

**Metodología de enseñanza y aprendizaje:** Lección magistral por parte del profesor y puesta en práctica de los conocimientos adquiridos por parte de los alumnos, además del uso de herramientas disponibles.

**Criterios y procedimientos de evaluación:** La calificación de la asignatura vendrá determinada por la calidad de los trabajos que el alumno realice sobre los contenidos de la asignatura.

**Bibliografía relevante y actualizada:**

- Moreno, L., Palomar, M., Molina, A., Fernández, A. (1999) *Introducción al procesamiento del Lenguaje Natural*. Publicaciones de la Universidad de Alicante.
- Allen, J. (1995) *Natural Language Understanding*. Benjamin Cummings series in Computer Science.
- R. Mitkov ed. (2003) *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2003.
- Jurafsky, D., Martin, J.H. (2000) *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, N. J.
- Strzalkowski, T. (1999) *Natural Language Information Retrieval*. Kluwer Academic Publishers.
- Manning, C. D., Schütze, H. (1999) *Foundations of Statistical Natural Language Processing*. The MIT Press.

**Nombre:** Ingeniería del software automática

**Créditos:** 4

**Semestre:** A

**Objetivos:** El curso se centra en el estudio de métodos formales, y las técnicas y herramientas automáticas asociadas, para dar soporte sistemático y racional al desarrollo del software. Siguiendo un enfoque moderno, que tiene en cuenta los tres elementos de la trilogía del software -programas, datos y propiedades-, estudiamos los procesos formales que transforman dichas componentes de forma automática, en particular los de mayor impacto industrial. Esto incluye, entre otros, los siguientes mecanismos:

- Análisis y especificación (modeling)
- Verificación (model-checking)
- Certificación (proof-carrying code)
- Síntesis (machine learning)
- Optimización (análisis y transformación)
- Depuración (diagnóstico y corrección)

**Contenidos:**

Programa de teoría:

+ Introducción a la Ingeniería del Software Automática

- \* La trilogía del desarrollo del software.
- \* Métodos formales, la aproximación lightweight.
- \* Lógicas para aplicaciones software

+ Procesos formales y tecnología ágil

- Análisis y especificación
  - \* Sistemas de reescritura de términos; Propiedades de los Sistemas de Reescritura
  - \* Herramientas de especificación y verificación automática
  - \* Integración de la programación lógica y funcional
- Verificación (model-checking)
- Depuración (diagnóstico y reparación)
- Certificación (proof-carrying code)

Prácticas de Laboratorio: Se realizarán tres grupos de prácticas:

1. Análisis de terminación
2. Modelización y verificación automática con SMV; El entorno Maude
3. Programación multiparadigma: El lenguaje Curry

**Metodología de enseñanza y aprendizaje:** Clases de teoría, resolución de problemas, y prácticas de laboratorio.

**Criterios y procedimientos de evaluación:** La asignatura se evalúa mediante la resolución de cuestiones de carácter teórico-práctico, con dos partes separadas: A) Sistemas de reescritura de términos (S. Lucas): cuestiones cortas; B) Métodos ágiles (M. Alpuente): test de respuesta múltiple de 20 preguntas (aprox.). El peso de cada parte es del 50%, necesiándose una nota umbral de 4 (en cada parte) para promediar. En ambas partes se evalúan tanto los contenidos teóricos como los prácticos. Para la parte B se estudiará, de forma alternativa y particularizada a cada caso, la posibilidad de realizar una presentación o un desarrollo práctico.

**Bibliografía relevante y actualizada:**

- María Alpuente, Salvador Lucas. Introducción a la Ingeniería del Software Automática. UPV, 2002.
- F. Baader and T. Nipkow. Term Rewriting and All That. Cambridge University Press, 1998.
- R. Bird. Introducción a la Programación Funcional con Haskell. Prentice-Hall, 2000.
- N. Dershowitz. Termination of rewriting. Journal of Symbolic Computation:3:96-116, 1987.
- M. Muller-Olm, D. A. Schmidt, and B. Steffen. Model Checking: A Tutorial Introduction. In Proc. SAS'99. Springer LNCS 1694:330-354, 1999.
- M. Hanus. Functional Logic Programming: From Theory to Curry. Technical Report, Christian-Albrechts-Universität Kiel, 2005
- D. Le Metayer et al. Exploring the Software Development Trilogía. Journal of Logic Programming, Vol. 19&20, pp. 583-628, 1994

**Nombre:** Integración Semántica de Datos

**Créditos:** 3

**Semestre:** B

**Objetivos:** Estudio de los fundamentos del modelado conceptual. Revisión de lenguajes de modelado para S.I y web. Estudio de aproximaciones a la integración semántica de datos: estructurados y semiestructurados.

**Contenidos:**

1. Fundamentos de modelado conceptual.
2. Lenguajes de modelado para S.I
3. Lenguajes de modelado para la web
4. Ontologías
5. Integración semántica de datos.

**Metodología de enseñanza y aprendizaje:** Clases teóricas, discusión de casos de estudio, revisión de bibliografía sobre la materia.

**Criterios y procedimientos de evaluación:** Presentación de un trabajo sobre algún tema tratado en el curso.

**Bibliografía relevante y actualizada:**

- Modelització conceptual de sistemes d'informació. Antoni Olivé. Ediciones UPC. 2001
- Conceptual Modeling. Boman et al. Prentice Hall. 1997.
- Information modeling and relational databases. From conceptual analysis to logical design. Halpin, T. Morgan Kaufman. 2001
- Object-oriented methods: a foundation. Martin, J.; Odell, J. Prentice Hall. 1995.
- The object data standard: ODMG 3.0. Cattel, R.G. Morgan Kaufman, 2006
- The unified modelling language user guide. Booch, G.; Rumbaugh, J., Addison Wesley. 1999
- Common Warehouse Metamodel (CWM). <http://www.omg.org/>
- Unified Modelling Language (UML). <http://www.omg.org/>
- Entity-relationship modelling. Foundations of database technology. Thalheim, B. Springer Verlag. 2000.

**Nombre:** Modelado, diseño e implementación de servicios web

**Créditos:** 2.5

**Semestre:** B

**Objetivos:** Presentar los principios ingenieriles y tecnologías subyacentes a los servicios web. Se estudian los fundamentos y conceptos, el soporte tecnológico, los estándares, las arquitecturas, los principios de diseño, las técnicas de especificación de procesos complejos, los entornos de desarrollo y los estándares WS\*-. Se presentan y analizan líneas de investigación actuales y futuras para el desarrollo e integración de sistemas de información basados en servicios web.

**Contenidos:**

1. Servicios WEB y Escenarios de Aplicación.
2. Arquitecturas Orientadas a Servicios. Un Marco Conceptual para SOA.
3. Estándares y Tecnologías Subyacentes a los Servicios WEB.
4. Consumo, Implementación y Publicación de Servicios Web. Ejemplos prácticos.
5. Principios de Diseño para Servicios WEB. Frameworks y Patrones de Diseño.
6. Coordinación, Composición y Orquestación de Servicios WEB. Conceptos, Especificación y Modelado.
7. Las Extensiones WS- y Otros Estándares.
8. Entornos de Desarrollo de Servicios Web.

**Metodología de enseñanza y aprendizaje:** La enseñanza se lleva a cabo mediante clases magistrales que incluyen demostraciones de tecnologías y aplicaciones reales de los servicios web en el entorno industrial y académico.

**Criterios y procedimientos de evaluación:** La evaluación se llevará a cabo mediante el desarrollo y la presentación pública de trabajos. Se podrán llevar a cabo:

- Presentaciones sobre Estándares, Publicaciones, Extensiones de Servicios Web y/o Aplicaciones de los Servicios Web en entornos académicos e industriales.
- Trabajos de Investigación y desarrollo/preparación de artículos de investigación
- Desarrollos Prácticos. Presentación y Demos

**Bibliografía relevante y actualizada:**

- G. Alonso et al. Web Services. Concepts, Architectures and Applications. Springer, 2003.
- F. Curbera, F. Leymann, T. Storey, and D. Ferguson. Web Services Platform Architecture: Soap, WSDL, WS-Policy, WS-Addressing, WS-Bpel, WS-Reliable Messaging and More- Prentice Hall 2005.
- M. Stal. Web Services: Beyond Component-Based Computing. Communications of the ACM 45, 2002
- E. Newcomer et al. Understanding SOA with Web Services. Addison-Wesley, 2004.
- M. Fowler. Patterns of Enterprise Application Architecture. Addison-Wesley, 2005.
- G. Hohpe et al. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions
- M. Young. XML Step by Step, Second Edition, Microsoft Press.
- K. Ballinger. .NET Web Services. Architecture and Implementation. Addison-Wesley. 2003
- W3C Consortium: <http://www.w3c.org>.
- Artículos Revistas ACM e IEEE
- Sitios Web (IBM, Microsoft, BEA, Sun, etc?)

**Nombre:** Técnicas avanzadas en ingeniería de requisitos

**Créditos:** 2.5

**Semestre:** B

**Objetivos:** Al finalizar la asignatura el alumno debería ser capaz de:

- Realizar la gestión de requisitos de software utilizando herramientas específicas
- Configurar la trazabilidad de requisitos para un proceso de software
- Describir algunas técnicas avanzadas para ingeniería requisitos, incluyendo enfoques orientado a objetivos, orientado a aspectos y ingeniería de requisitos para líneas de productos
- Describir las principales tendencias y desafíos en ingeniería de requisitos

**Contenidos:**

- Introducción a Ingeniería de Requisitos
- Trazabilidad de Requisitos
- Caso de Estudio
  - Introducción a RequisitePro
  - Configuración de trazabilidad
  - Elaboración de artefactos y establecimiento de trazabilidad
  - Plan de Desarrollo
- Tendencias en Ingeniería de Requisitos
- Seminario de presentación de trabajos
- Conclusiones

**Metodología de enseñanza y aprendizaje:**

- Presentaciones del profesor, temas de teoría
- Sesiones de trabajo en grupo, caso de estudio
- Seminario, presentación de trabajos en grupos

**Criterios y procedimientos de evaluación:**

- Caso de Estudio: 60 %
- Presentación en seminario: 20 %
- Examen: 20 %

**Bibliografía relevante y actualizada:**

- Dean Leffingwell, Don Widrig. Managing Software Requirements: A Use Case Approach. Addison Wesley. 2003
- Alan M. Davis. Just Enough Requirements Management: Where Software Development Meets Marketing. Dorset House Publishing. 2005
- Lawrence Chung et al. Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers. 2000.

**Nombre:** Tecnología software para ambientes web

**Créditos:** 4

**Semestre:** B

**Objetivos:**

- 1- Introducir a los alumnos en el análisis integral de Sistemas de Información, en particular en Sistemas de Información para Ambientes Web, haciendo hincapié en la necesidad de usar procesos de producción de software basados en modelos conceptuales donde las perspectivas estática, dinámica y de interacción se gestionen de manera unificada
- 2.- Analizar los principios básicos de las herramientas CASE basadas en la noción de Compilación de Esquemas Conceptuales, usando como ejemplo práctico el método OO-Method y herramientas que lo soportan como Olivanova
- 3.- Diseñar e Implementar Sistemas de Información con herramientas de producción automática de código a partir de Modelos, considerando de manera especial las particularidades propias de los entornos web.
- 4.- Estudiar principios de medida de la calidad y del tamaño funcional para entornos de producción automática de aplicaciones web, usando la plataforma OO-Method/OOWS como sistema de referencia.

**Contenidos:**

Teoría:

- Introducción: Generalidades de Modelado Conceptual: El Paradigma Orientado a Objetos
- Entornos de modelado basados en UML
- La aproximación OO-Method: Generación de Código a partir de Modelos Conceptuales
- Modelado Conceptual de Aplicaciones Web
- Conclusiones
- Anexo 1: Ingeniería de Requisitos
- Anexo 2: Medida del Tamaño Funcional de Aplicaciones Web

Prácticas

- Resolución de un Caso de Estudio con la herramienta OlivaNova

**Metodología de enseñanza y aprendizaje:** El curso se plantea desde una perspectiva que combina presentaciones teóricas con clases prácticas. Los contenidos teóricos se presentarán combinando clase magistral con ejercicios que hagan posible la discusión y puesta en práctica de los conocimientos expuestos. Un caso de estudio resuelto con una herramienta de compilación de modelos será resuelto de manera colectiva en el laboratorio para proyectar en entornos reales las ideas expuestas durante el desarrollo del programa de teoría

**Criterios y procedimientos de evaluación:** La evaluación del curso se hará a través de dos entregables:

- un trabajo breve (10 págs. aproximadamente) de investigación relacionado con cualquiera de los temas expuestos en el curso. El objetivo del mismo es introducir al alumno en las técnicas y capacidades requeridas para escribir artículos básicos de temas de investigación (exposición de un problema concreto, estado del arte para resolverlo, contribución original y relevante)
- la entrega del caso práctico realizado en prácticas totalmente acabado, tanto a nivel de modelo como de producto software correspondiente.

**Bibliografía relevante y actualizada:**

- O.Pastor,J.Gómez,E.Insfran,V.Pelechano. The OO Method Approach for Information Systems Modeling: From Object Oriented conceptual Modeling to Automatic Programming. Information Systems Journal, Elsevier Science, October 2001, Vol 26/7, pp 507-534.
- Conceptual Modeling of Web Applications: The OOWS Approach. Pastor O., Fons J., Pelechano V., Abrahão S. Web Engineering - Theory and Practice of Metrics and Measurement for Web Development <http://www.amazon.com/exec/obidos/ASIN/3540281967/qsucess-20/104-9548070-0512712?dev-t=D1GQGDLFYU3D4>, E. Mendes and N. Mosley (Eds.), 2005, ISBN: 3540281967, Springer Verlag.
- O.Pastor,J.C.Molina. Model-Driven Architecture in Practice. A Software Production Environment Based on Conceptual Modeling. Springer-Verlag, 2007, ISBN: 978-3-540-71867-3
- Web Engineering: Modelling and Implementing Web Applications. Series: Human-Computer Interaction Series <http://www.springer.com/west/home/computer/user+interfaces?SGWID=4-154-69-173624645-0>, Vol. 12. Rossi, G.; Pastor, O.; Schwabe, D.; Olsina, L. (Eds.) 2008, Approx. 460 p., 169 illus., Hardcover. ISBN: 978-1-84628-922-4