

Curso Académico: (2019 / 2020)

Fecha de revisión: 09-05-2020

Departamento asignado a la asignatura: Departamento de Ingeniería Telemática

Coordinador/a: SANCHEZ FERNANDEZ, LUIS

Tipo: Obligatoria Créditos ECTS : 6.0

Curso : 3 Cuatrimestre : 2

REQUISITOS (ASIGNATURAS O MATERIAS CUYO CONOCIMIENTO SE PRESUPONE)

- Programación
- Programación de Sistemas

OBJETIVOS

El objetivo de esta asignatura es que los alumnos adquieran los conocimientos y capacidades básicos para procesar documentos en formatos intercambiados en aplicaciones telemáticas. Se considera tanto el enfoque tradicional (procesadores de lenguajes) como el enfoque basado en tecnología XML, y se relacionan ambos. Para lograr estos objetivos, el alumno debe adquirir una serie de conocimientos y capacidades.

Por lo que se refiere a conocimientos, al finalizar el curso el estudiante será capaz de:

- Conocer las definiciones de alfabeto, cadena y lenguaje.
- Entender las partes en que se divide un compilador, las tareas que se realizan en cada parte y sus interrelaciones.
- Conocer algunos formalismos para definir lenguajes: expresiones regulares y gramáticas.
- Conocer las funciones de un analizador léxico y como especificarlo.
- Conocer las funciones de un analizador sintáctico y como especificarlo.
- Conocer el funcionamiento de un generador de analizadores sintácticos LR(k).
- Conocer las definiciones de atributo, atributo heredado y atributo sintetizado.
- Conocer como especificar un atributo.
- Conocer el lenguaje XML
- Conocer el formato DTD para especificación de aplicaciones XML
- Conocer el entorno para procesar documentos XML DOM
- Conocer el lenguaje de hojas de estilo XSLT

En cuanto a las capacidades, éstas las podemos clasificar en dos grupos, uno de capacidades específicas y otro de capacidades más genéricas o destrezas.

En cuanto a las capacidades específicas, al finalizar el curso el alumno será capaz de:

- Entender cuál es el lenguaje definido por una expresión regular o una gramática.
- Definir la especificación de un analizador léxico y un analizador sintáctico para un lenguaje dado.
- Programar un conjunto de clases Java que representen los árboles de sintaxis abstracta para cadenas de un lenguaje dado.
- Utilizar generadores de analizadores para producir analizadores léxicos y sintácticos.
- Implementar una tabla de símbolos.
- Implementar un atributo sintetizado o heredado a partir de su especificación.
- Implementar un comprobador de tipos.
- Implementar otras comprobaciones requeridas por el analizador semántico.
- Diseñar e implementar un generador de código en Java.
- Determinar si un documento XML es válido según un DTD.
- Escribir programas sencillos que procesen documentos XML.
- Escribir hojas de estilo XSLT sencillas.

En cuanto a las capacidades genéricas o destrezas, los alumnos deberán ser capaces de:

- Identificar, formular y resolver problemas ingenieriles. (PO: e)
- Usar técnicas, destrezas y modernas herramientas ingenieriles necesarias para la práctica ingenieril. (PO: k)

DESCRIPCIÓN DE CONTENIDOS: PROGRAMA

El contenido se divide en dos grandes bloques:

1. Definición, análisis y procesado de formatos genéricos
 - Definición de alfabeto, cadena y lenguaje
 - Gramáticas y expresiones regulares
 - Fases de un compilador
 - Análisis léxico y sintáctico
 - Generadores de analizadores
 - Análisis semántico y generación de código
2. XML como formato interno, de intercambio y presentación en aplicaciones telemáticas
 - El lenguaje XML
 - DTD
 - Procesado de documentos XML con DOM
 - XPath
 - XSLT

ACTIVIDADES FORMATIVAS, METODOLOGÍA A UTILIZAR Y RÉGIMEN DE TUTORÍAS

Las actividades que se llevan a cabo en la impartición de la asignatura son:

- Clases magistrales. Presentación por parte del profesor de los principales conceptos a modo de resumen. Se fomentan en este tipo de sesiones tanto la interactividad como la discusión de los principales problemas planteados.
- Clases de ejercicios prácticos: Sesiones en las que se plantean problemas reales y se permite a los estudiantes su análisis así como el planteamiento de posibles soluciones.
- Laboratorios de programación: En estas sesiones los alumnos deberán desarrollar un compilador sencillo así como realizar pequeños desarrollos en los que se trabajará con las tecnologías XML. Con la supervisión de un profesor, se procede a su implementación. Estas sesiones suponen un trabajo adicional del estudiante durante un período de varios días en los que tiene a su disposición a la plantilla docente por múltiples canales para resolver las dudas que se hayan planteado.

SISTEMA DE EVALUACIÓN

La evaluación de la adquisición de competencias se hará por medio de:

- Un examen parcial sobre análisis léxico y sintáctico compuesto de preguntas tipo test, cuestiones teóricas y/o problemas (peso 30 puntos).
- Un examen parcial sobre análisis semántico y generación de código compuesto de cuestiones teóricas y/o problemas (peso 20 puntos).
- Examen final sobre XML (peso 30 puntos).
- Evaluación de las prácticas realizadas por los alumnos en los laboratorios (peso 10 puntos).
- Examen de prácticas (peso 10 puntos).

Mejora

A los alumnos que hayan obtenido simultáneamente una calificación igual o superior a 5 puntos en la evaluación de las prácticas, una calificación igual o superior a 5 puntos en el examen de prácticas y una calificación mayor a 20 puntos entre los 2 exámenes parciales se les calificará la evaluación continua, caso de que les fuera más favorable, de la siguiente forma:

Siendo:

NP: suma de la nota de los 2 exámenes parciales (sobre 50 puntos)
NL: nota de la evaluación de las prácticas (sobre 10 puntos)
NEL: nota del examen de prácticas (sobre 10 puntos)

NEC: nota en la evaluación continua

- Si $20 < NP \leq 30$

$$NEC = NP + NL + NEL + (NL - NP/5) * (NP - 20)/10$$

- Si $30 < NP \leq 45$

$$NEC = NP * 4/5 + NEL + NL * (2 + (NP - 30)/15)$$

- Si $NP > 45$

$$NEC = NP * 4/5 + NEL + NL * 3$$

Requisitos para aprobar

Los alumnos que hayan seguido la evaluación continua deberán cumplir los siguientes requisitos para aprobar la asignatura:

- 1) Obtener una calificación entre la evaluación continua y el examen final mayor o igual a 50 puntos
- 2) Obtener una calificación en la evaluación de las prácticas mayor o igual a 5 puntos (sobre 10)
- 3) Obtener una calificación en el examen final mayor o igual a 12 puntos

Peso porcentual del Examen Final: 30

Peso porcentual del resto de la evaluación: 70

BIBLIOGRAFÍA BÁSICA

- Scott E. Hudson, modificado por Frank Flannery, C. Scott Ananian y Dan Wang, guiados por Andrew W. Appel CUP User's Manual, <http://www.cs.princeton.edu/~appel/modern/java/CUP/manual.html>, 1999
- A.V. Aho, R. Sethi, J.D. Ullman Compilers: principles, techniques and tools, Pearson Education.
- Andrew W. Appel Modern Compiler Implementation in Java, Cambridge University Press.
- Atul Kahate XML & Related Technologies, Pearson Education, 2009
- Carlos Delgado Kloos, Peter T. Breuer Definición de lenguajes de programación, <http://www.it.uc3m.es/luis/fo1/es/parte1.pdf>.
- E.R. Harold, W.S. Means XML in a Nutshell, Third edition, O'Reilly.
- Elliot Berk JLex: A lexical analyzer generator for Java(TM), <http://www.cs.princeton.edu/~appel/modern/java/JLex/current/manual.html>, 2000
- Erik T. Ray Learning XML, Second Edition, O'Reilly, 2003
- Evan Lenz XSLT 1.0 Pocket Reference, O'Reilly, 2005
- Gregorio Fernández Fernández, Fernando Sáez Vacas Fundamentos de informática: [lógica, autómatas, algoritmos y lenguajes], Anaya Multimedia.
- Luis Sánchez Fernández Introduction to Compiler Development with Java, Apuntes de la asignatura.
- World Wide Web Consortium XML Technology. <http://www.w3.org/standards/xml/>, World Wide Web Consortium.