ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

GUIA DOCENTE

CURSO 2015/2016

Grado en Ingeniería Informática

DATOS DE LA ASIGNATURA				
Nombre:				
Diseño de Compiladores				
Denominación en inglés:				
Compilers Design				
Código:			**Carácter:**	
606010309			Optativo	
Horas:				
	Totales		**Presenciales**	**No presenciales**
Trabajo estimado:	150		60	90
Créditos:				
	Grupos reducidos			
Grupos grandes	**Aula estándar**	**Laboratorio**	**Prácticas de campo**	**Aula de informática**
3	0	0	0	3
Departamentos:			**Áreas de Conocimiento:**	
Tecnologías de la Información			Ciencia de la Computación e Inteligencia Artificial	
Curso:			**Cuatrimestre:**	
4º - Cuarto			Primer cuatrimestre	

DATOS DE LOS PROFESORES			
Nombre:	**E-Mail:**	**Teléfono:**	**Despacho:**
*Francisco José Moreno Velo	francisco.moreno@dti.uhu.es	87659	Edificio Torreumbría, despacho 14.

*Profesor coordinador de la asignatura

Consultar los horarios de la asignatura

DATOS ESPECÍFICOS DE LA ASIGNATURA

1. Descripción de contenidos

1.1. Breve descripción (en castellano):

Introducción a los Compiladores
Tabla de símbolos.
Árbol de Sintáxis Abstracta
Generación de código intermedio
Organización y gestión de la memoria
Optimización de código
Técnicas básicas de optimización
Optimización local
Generación de código final
Paralelización de programas secuenciales
Aspectos específicos de los LOO
Aspectos específicos de los LF

1.2. Breve descripción (en inglés):

Introduction to Compilers
Table of Symbols.
Abstract Syntax Tree.
Intermediate code generation.
Memory management and organization.
Code Optimization.
Basic techniques.
Local optimization.
Object code generation.
Parallelization of sequential programs.
Specific aspects of OOL.
Specific aspects of FL.

2. Situación de la asignatura

2.1. Contexto dentro de la titulación:

Asignatura optativa de 4 curso de la titulación donde se profundiza en las técnicas para el diseño adecuado de compiladores y traductores.

2.2. Recomendaciones:

Se recomienda para esta asignatura tener conocimientos de programación orientada a objetos y estructuras de datos.

3. Objetivos (Expresados como resultados del aprendizaje):

Conocer e implementar las diferentes fases del proceso de compilación de los lenguajes imperativos: análisis, gestión de memoria y generación de código.
Conocer e implementar las técnicas básicas de optimización de código.
Conocer las técnicas básicas de gestión de memoria dinámica.

4. Competencias a adquirir por los estudiantes

4.1. Competencias específicas:

4.2. Competencias básicas, generales o transversales:

- **CB5:** Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía
- **CG0:** Capacidad de análisis y síntesis: Encontrar, analizar, criticar (razonamiento crítico), relacionar, estructurar y sintetizar información proveniente de diversas fuentes, así como integrar ideas y conocimientos.
- **G02:** Capacidad de comunicación oral y escrita en el ámbito académico y profesional con especial énfasis, en la redacción de documentación técnica
- **G03:** Capacidad para la resolución de problemas
- **G04:** Capacidad para tomar decisiones basadas en criterios objetivos (datos experimentales, científicos o de simulación disponibles) así como capacidad de argumentar y justificar lógicamente dichas decisiones, sabiendo aceptar otros puntos de vista
- **G05:** Capacidad de trabajo en equipo.
- **G06:** Capacidad para el aprendizaje autónomo así como iniciativa y espíritu emprendedor
- **T02:** Conocimiento y perfeccionamiento en el ámbito de las TIC’s

5. Actividades Formativas y Metodologías Docentes

5.1. Actividades formativas:

- Sesiones de Teoría sobre los contenidos del Programa.
- Sesiones Prácticas en Laboratorios Especializados o en Aulas de Informática.
- Actividades Académicamente Dirigidas por el Profesorado: seminarios, conferencias, desarrollo de trabajos, debates, tutorías colectivas, actividades de evaluación y autoevaluación.

5.2. Metologías docentes:

- Clase Magistral Participativa.
- Desarrollo de Prácticas en Laboratorios Especializados o Aulas de Informática en grupos reducidos.
- Resolución de Problemas y Ejercicios Prácticos.
- Tutorías Individuales o Colectivas. Interacción directa profesorado-estudiantes.
- Planteamiento, Realización, Tutorización y Presentación de Trabajos.
- Evaluaciones y Exámenes.

5.3. Desarrollo y justificación:

Se desarrollarán clases teóricas donde se impartirán los conceptos fundamentales de la asignatura y sus aplicaciones. Estos conceptos se reforzarán en unas series de prácticas que deberán realizar en grupos reducidos. Finalmente se realizará una prueba donde el alumno deberá de reflejar los conocimientos adquiridos tanto a nivel teórico como práctico.

6. Temario desarrollado:

TEMARIO TEÓRICO
Tema 1: Introducción a los Compiladores
Repaso a las etapas de compilación
Tema 2: Tabla de símbolos.
Operaciones frecuentes.
Integración semántica.
Gestión de ámbitos.
Reglas de visibilidad: Importación, Exportación.
Declaraciones implícitas.
Sobrecarga.
Referencias adelantadas
Tema 3: Árbol de Sintáxis Abstracta
Estructura del ASA.
Construcción (recursiva) del ASA.
Tema 4: Generación de código intermedio
Lenguajes Intermedios: Tipos.
Código dos direcciones.
Código tercetos indirectos.
Código tres direcciones.
Código asociado a las instrucciones comunes.
Código para máquinas abstractas.
Tema 5: Organización y gestión de la memoria
Organización de la memoria en tiempo de ejecución
Zona de código
Memoria estática
Memoria de pila
Memoria con reserva dinámica
Recolección de basura
Tema 6: Optimización de código
Optimización .vs. óptimo.
Etapas de optimización.
Dependencia de la máquina.
Tema 7: Técnicas básicas de optimización
Cálculo previo de constantes.
Reducción de fuerza y frecuencia.
Optimización de ciclos.
Eliminación de código.
Reacomodos.
Tema 8: Optimización local
Bloques Básicos.
Grafo Dirigido Acíclico.
Eliminación código redundante.
Expresiones comunes.
Adjudicación de registros.
Tema 9: Generación de código final.
Resolución de referencias simbólicas.
Modos de direccionamiento.
Entornos de ejecución.
Selección de instrucciones.
TEMARIO PRÁCTICO
Práctica 1: Completar el conjunto de instrucciones del lenguaje Tinto
Práctica 2: Añadir operadores de bit
Práctica 3: Añadir operadores de incremento, decremento y asignación
Práctica 4: Añadir tipos de datos enteros (byte, short y long)
Práctica 5: Añadir tipos de datos en coma flotante (float y double)
Práctica 6: Optimización por paso de argumentos por registro
Práctica 7: Optimización por reordenación de código
Práctica 8: Optimización por alojamiento en registro
Práctica 9: Añadir tipo de dato array
Práctica 10: Liberación de memoria con comtadores de uso
Práctica 11: Constantes y variables globales
Práctica 12: Añadir clases

7. Bibliografía

7.1. Bibliografía básica:

Libro: .Compiladores. Principios, Técnicas y Herramientas.
Autores: A.V. Aho.
Editorial: Addison-Wesley Iberoamericana. Año: 1998
Libro: .Compiladores e Intérpretes: teoría y práctica.
Autores: Manuel Alfonseca Morena y otros
Editorial: Pearson . Prentice Hall. Año: 2006
Libro: .Modern compiler implementation in Java(second edition).
Autores: Andrew W. Appel
Editorial: Cambridge. Año: 2002

7.2. Bibliografía complementaria:

Libro: .Construcción de compiladores.
Autores: Kenneth C. Louden
Editorial: Thomson Año: 2004
Libro: .Compiladores. Teoría y Construcción.
Autores: Sanchís Llorca y Galán Pascual
Editorial: Paraninfo Año: 1986
Libro: .Compiladores e Intérpretes. Un enfoque pragmático.
Autores: Sanchez Dueñas y Valverde Andreu
Editorial: Diaz de Santos, S.A. Año: 1984
Libro: .Compiladores. Conceptos fundamentales.
Autores: Teufel, Schmidt y Teufel
Editorial: Addison-Wesley Iberoamericana Año: 1995

8. Sistemas y criterios de evaluación.

8.1. Sistemas de evaluación:

- Examen de teoría/problemas
- Defensa de Trabajos e Informes Escritos
- Seguimiento Individual del Estudiante
- Examen de prácticas

8.2. Criterios de evaluación y calificación:

- Examen final que constará de preguntas teóricas y problemas. - Realización de prácticas de laboratorio. - Actividades académicas dirigidas (individuales o en grupo). Trabajo final.
Las calificaciones serán ponderadas según la siguiente relación: 50% examen, 40% trabajo final y 10% participación en clase.
El alumno deberá superar en, al menos, un 40% cada una de las partes y obtener un mínimo del 50% de la media aritmética de las partes.

9. Organización docente semanal orientativa:

Semanas	Grupos Grandes	Grupos Reducidos Aula Estándar	Grupos Reducidos Aula de Informática	Grupos Reducidos Laboratorio	Grupos Reducidos prácticas de campo	Pruebas y/o actividades evaluables	Contenido desarrollado
#1	2	0	0	2	0		Presentación
#2	2	0	0	2	0		Tema 1
#3	2	0	0	2	0	Entrega práctica 1	Tema 2
#4	2	0	0	2	0	Entrega práctica 2	Tema 3
#5	2	0	0	2	0	Entrega práctica 3	
#6	2	0	0	2	0	Entrega práctica 4	Tema 4
#7	2	0	0	2	0	Entrega práctica 5	
#8	2	0	0	2	0	Entrega práctica 6	Tema 5
#9	2	0	0	2	0	Entrega práctica 7	Tema 6
#10	2	0	0	2	0	Entrega práctica 8	
#11	2	0	0	2	0	Entrega práctica 9	Tema 7
#12	2	0	0	2	0	Entrega práctica 10	
#13	2	0	0	2	0	Entrega práctica 11	Tema 8
#14	2	0	0	2	0	Entrega práctica 12	
#15	2	0	0	30	0		Tema 9
	30	0	0	58	0		