

## Grado en Ingeniería Informática

### DATOS DE LA ASIGNATURA

**Nombre:**

Diseño de Compiladores

**Denominación en inglés:**

Compilers Design

**Código:**

606010309

**Carácter:**

Optativo

**Horas:**

	Totales	Presenciales	No presenciales
<b>Trabajo estimado:</b>	150	60	90

**Créditos:**

Grupos reducidos				
Grupos grandes	Aula estándar	Laboratorio	Prácticas de campo	Aula de informática
3	0	0	0	3

**Departamentos:**

Tecnologías de la Información

**Áreas de Conocimiento:**

Ciencia de la Computación e Inteligencia Artificial

**Curso:**

4º - Cuarto

**Cuatrimestre:**

Primer cuatrimestre

### DATOS DE LOS PROFESORES

**Nombre:**

\*Moreno Velo, Francisco  
José

**E-Mail:**

francisco.moreno@dti.uhu.es

**Teléfono:**

87659

**Despacho:**

Edificio Torreumbria,  
despacho 14.

\*Profesor coordinador 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. Metodologías 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:

- Clase Magistral Participativa

Las clases teóricas tendrán una duración de 2 horas. En ellas se expondrá y explicará, con ayuda del cañón de proyecciones y la pizarra, los contenidos asociados a cada tema. Habrá bibliografía específica de cada tema disponible en la web de la asignatura con antelación suficiente.

- Resolución de Problemas y Ejercicios Prácticos

Al finalizar las sesiones de teoría de cada tema se desarrollarán las sesiones de problemas correspondientes al tema desarrollado. Para cada tema de teoría se facilitará un boletín de problemas. En estas sesiones se resolverán los problemas más representativos de cada boletín.

- Desarrollo de Prácticas en Laboratorios Especializados o Aulas de Informática en grupos reducidos

Las sesiones de prácticas se desarrollarán en aulas provistas de ordenadores y tendrán una duración de 2 horas. En estas prácticas se implementarán aplicaciones gráficas haciendo uso de las características de la librería OpenGL descritas en las sesiones teóricas.

- Planteamiento, Realización, Tutorización y Presentación de Trabajos

A lo largo del curso se planteará uno trabajo práctico a desarrollar por los alumnos de manera individual. El trabajo se referirá al desarrollo de una aplicación gráfica que permita navegar e interactuar sobre un escenario virtual. Este trabajo se considera una actividad académica dirigida y su explicación se realizará en el horario de las sesiones de prácticas. El seguimiento de este trabajos se realizará en tutorías individualizadas.

## 6. Temario desarrollado:

### TEMARIO TEÓRICO

#### Tema 1: Introducción a los Compiladores

- 1.1 Introducción
- 1.2 Estructura de un compilador
- 1.3 Una herramienta de ayuda al diseño de compiladores
- 1.4 Un ejemplo de compilador

#### Tema 2: Análisis léxico

- 2.1 Características
- 2.2 Análisis léxico basado en Autómatas No Deterministas
- 2.3 Contextos léxicos

#### Tema 3: Análisis sintáctico

- 3.1 Características
- 3.2 Análisis descendente con gramáticas extendidas
- 3.3 Lookahead adaptativo
- 3.4 Análisis ascendente LR(1)
- 3.5 Análisis ascendente LALR

#### Tema 4: Análisis semántico

- 4.1 Características del análisis semántico
- 4.2 Analizadores semánticos descendentes
- 4.3 Analizadores semánticos ascendentes
- 4.4 Generación automática de analizadores semánticos

#### Tema 5: Organización y gestión de la memoria

- 5.1 Organización de la memoria en tiempo de ejecución
- 5.2 Zona de código, memoria estática y memoria de pila
- 5.3 Memoria con reserva dinámica
- 5.4 Recolección de basura

#### Tema 6: Optimización de código

- 6.1 Definición
- 6.2 Tipos de optimización
- 6.3 Optimización local
- 6.4 Optimización de bucles

#### Tema 7: Alojamiento en registro

- 7.1 Planteamiento del problema
- 7.2 Algoritmo de coloreado de mapas
- 7.3 Algoritmo de análisis de supervivencia
- 7.4 Convenciones sobre el uso de registros

#### Tema 8: Aspectos avanzados

- 8.1 Paralelización de programas secuenciales
- 8.2 Aspectos específicos de los lenguajes orientados a objetos
- 8.3 Aspectos específicos de los lenguajes funcionales

### TEMARIO PRÁCTICO

#### Práctica 1: El compilador del lenguaje Tinto

#### Práctica 2: Conjunto de instrucciones

#### Práctica 3: Operadores de bit

#### Práctica 4: Operadores de incremento, decremento y asignación

#### Práctica 5: Tipos de datos byte y short

#### Práctica 6: Tipo de dato long

#### Práctica 7: Tipos de datos en coma flotante (float y double)

#### Práctica 8: Constantes y variables globales

#### Práctica 9: Optimización local

#### Práctica 10: Clases y objetos

#### Práctica 11: Tipo de dato array

#### Práctica 12: Mecanismos de herencia

## 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

### 8.2. Criterios de evaluación y calificación:

Examen de teoría/problemas: 50%  
Defensa de Trabajos e Informes Escritos: 50%  
La evaluación de la asignatura consta de una parte teórica y una parte práctica.  
La parte teórica (50% de la nota final) se evalúa por medio de las convocatorias oficiales de exámenes (junio y septiembre).  
La parte práctica (50% de la nota final) se evalúa por medio de un trabajo individual.  
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:**

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