

Presentación de la asignatura

Estructura de Computadores

1ª Semana

Bibliografía:

Fichas grado Informática <http://grados.ugr.es/informatica/pages/infoacademica/por cursos>

Ficha grado EC http://grados.ugr.es/informatica/pages/infoacademica/guias_docentes/comunesrama/estructuradecomputadores

Calendario académico <http://www.ugr.es/pages/estudios/calendario>

Presentación de la asignatura

- **Ubicación de EC en el Plan de Estudios**
- Temario de la asignatura
- Horarios, Temporización
- Evaluación, Metodología
- Otra información

Estructura de Computadores

■ **Continuación** de asignaturas previas del área (ATC)

- Tecnología y Organización de Computadores (2º semestre)
 - Se siguen estudiando las 5 unidades funcionales: E/S, M, CPU (=ALU+UC)

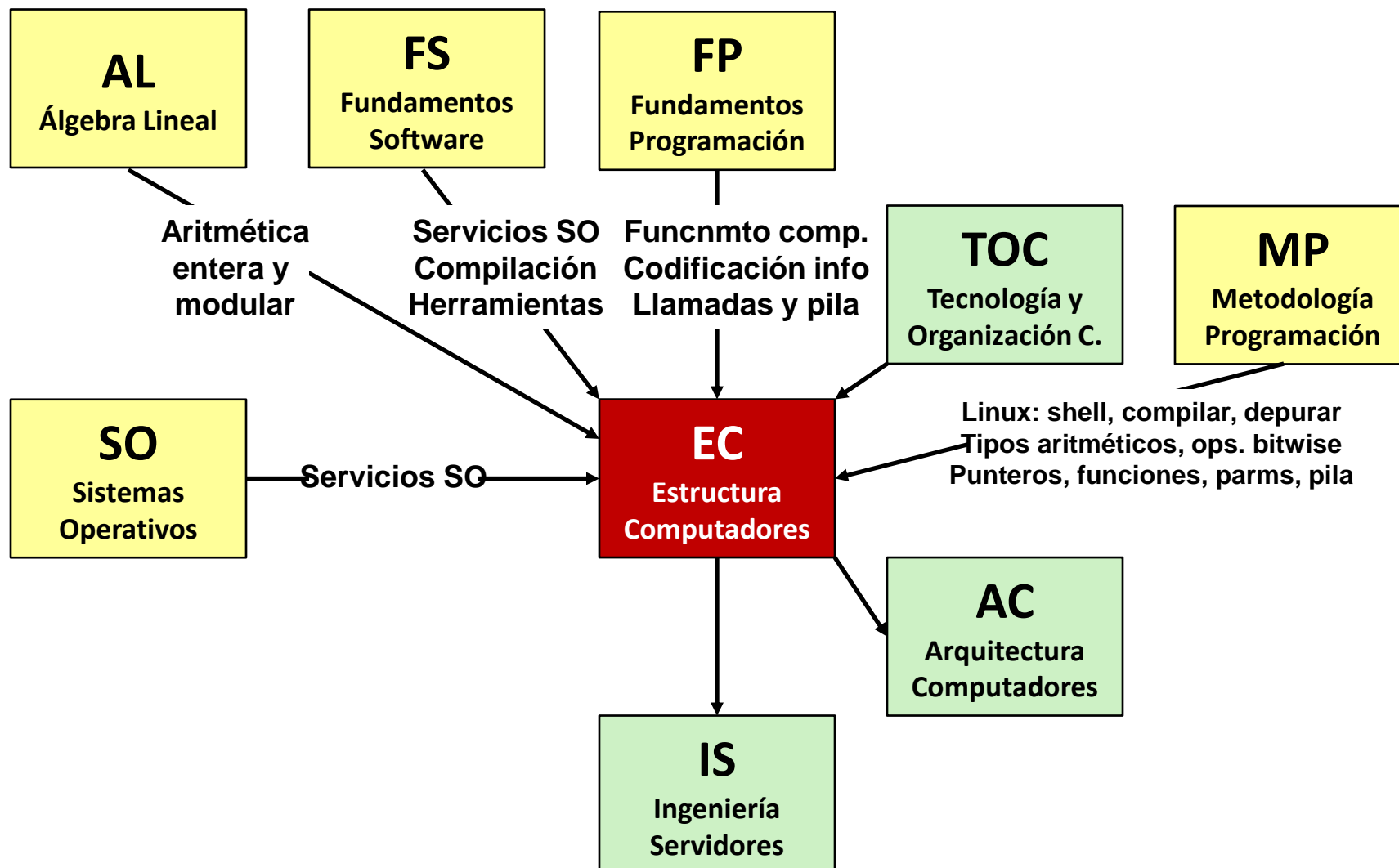
■ **Preparación para asignaturas posteriores** del área

- Arquitectura de Computadores (4º semestre)
- Ingeniería de Servidores (5º semestre)

■ **Prácticas** se apoyan en asignaturas de otras áreas

- Álgebra Lineal y Estructuras Matemáticas (1º semestre)
- Fundamentos del Software (1º semestre)
- Fundamentos de Programación (1º semestre)
- Metodología de la Programación (2º semestre)

Ubicación de EC en el Plan de Estudios



Apoyo prácticas

■ Algebra Lineal (1º semestre)

- Tema 2: Aritmética Entera y Modular
- Práctica 2: Práctica de Aritmética Entera y Modular

■ Fundamentos de Programación (1º semestre)

- Objetivos:
 - Comprender el funcionamiento de un computador...
 - Mostrar la necesidad de codificar la información que maneja internamente un computador, enfatizando posibilidades y limitaciones cuando se resuelve un problema.
 - Entender la gestión de llamadas a funciones mediante la pila.

Apoyo prácticas

■ Fundamentos del Software (1º semestre)

■ Contenidos:

- **Servicios** del Sistema Operativo: **llamadas al sistema** e intérpretes de órdenes.
- **Compilación, enlazado y carga de programas.**
- Entornos y **herramientas** de desarrollo de aplicaciones.

■ Objetivos:

- Manejar los servicios que ofrece el sistema operativo vía llamadas al sistema, intérpretes de órdenes o programas de utilidad.
- Comprender los pasos necesarios para construir un programa de aplicación desde su programación hasta la generación del código ejecutable y su posterior ejecución por el sistema operativo.
- Utilizar un entorno de desarrollo de aplicaciones, teniendo en cuenta las herramientas necesarias para trabajar con distintos módulos de código fuente, interdependencias entre estos, portabilidad de la aplicación y depuración de errores.

Apoyo prácticas

■ Fundamentos del Software (cont)

- Tema 1. Sistema de computo.
 - 1.1. Componentes de los sistemas de computo.
 - 1.2. Capa hardware: **interrupciones y excepciones**, protección, y entradas/salidas.
- Tema 2. Introducción a los sistemas operativos
 - 2.1. Componentes de un sistema operativo multiprogramado.
 - 2.2. **Servicios** del sistema operativo: API y shell.
- Tema 3. **Compilación y enlazado de programas**
- Tema 4. **Depuración** de programas
 - 4.1. Ejecución supervisada del programa. Visualización de estado.
 - 4.2. Seguimiento de la asignación/desasignación dinámica de memoria. Memory leaks.
 - 4.3. **Volcado de registros. Volcado de memoria.**

- Práctica 1. Órdenes básicas del sistema operativo e interprete de órdenes.
- Seminario práctico 1: Instalación de un sistema operativo y utilidades básicas.
- Seminario práctico 2: Compilación y enlazado de programas.
- Seminario práctico 3: Depuración de programas.

Apoyo prácticas

■ Metodología de la Programación (2º semestre)

■ Objetivos

- Comprender la **relación** entre **tipos de alto nivel** y la representación **a bajo nivel** de dicha información
- **Distinguir** y manejar correctamente las **referencias** y los **objetos** referenciados.
- Manejar correctamente **herramientas** de depuración, pruebas y validación.

■ TEMARIO TEÓRICO

- TEMA 1. **Punteros** y memoria dinámica.
- TEMA 2. Funciones.
 - 2.1. La función main
 - 2.2. La responsable de que todo funcione: La **Pila**
 - 2.3. **Paso de parámetros y devolución de resultados**
 - ...
 - 2.6. Variables locales static

Apoyo prácticas

■ Metodología de la Programación (cont)

■ SEMINARIOS

- Seminario 1.-Linux
 - 1.1 El sistema operativo linux.
 - 1.2 Órdenes básicas.
 - 1.3 Compilación y enlazado en linux.
 - 1.4 Depuración en linux.
- Seminario 3.- Tipos aritméticos. Representación y conversiones
 - 3.1. Tipos integrales y en coma flotante.
 - 3.2. Características de los tipos
 - 3.3. Conversiones
 - 3.4. Operadores lógicos a nivel de bit

Asignaturas simultáneas

■ Sistemas Operativos

(3º semestre)

■ Contenidos

- Programación de aplicaciones utilizando los **servicios** del sistema operativo.

■ Objetivos

- Describir diferentes formas de asignar memoria a los procesos y al propio sistema operativo, con especial atención a los sistemas **paginados y segmentados** que utilizan gestión de **memoria virtual**.
- Describir proceso realización de una operación **entrada/salida** desde su inicio hasta su conclusión.

■ TEMARIO DE TEORÍA

- Tema 3. Gestión de memoria
 - 3.3. Memoria virtual.
- Tema 5. Gestión de entradas y salidas
 - 5.3. Manejadores de dispositivos.

■ TEMARIO DE PRÁCTICAS

- Práctica 1. Administración de sistemas operativos:
 - 1.1 Herramientas básicas de administración.
 - 1.2 Monitorización del sistema.
 - 1.3 Automatización de tareas.
- Práctica 2. Uso de servicios del sistema operativo mediante la API

Asignaturas previas del área

■ Tecnología y **Organización** de Computadores (2º semestre)

■ Contenidos

- Organización y componentes del computador. Prestaciones básicas. Niveles conceptuales de descripción de un computador. Representación de información en el computador. Componentes básicos. .. Descripción de las operaciones de un computador en el nivel de transferencia entre registros

■ Objetivos

- Conocer la organización y **componentes** de un computador.
- Identificar los factores que determinan las **prestaciones** básicas de un computador.
- Comprender la conveniencia de describir un computador en diferentes niveles de abstracción para facilitar su comprensión, su diseño y su utilización.
- Conocer las distintas formas básicas de **representación de la información** en un computador.
- Conocer la organización de los sistemas diseñados en el nivel de transferencia de registros, comprendiendo la misión del camino de datos y de la **unidad de control**, y su interacción.
- Deducir las operaciones de transferencia entre registros que puedan realizarse en un **camino de datos** dado.

Asignaturas previas del área

■ Tecnología y **Organización** de Computadores (cont)

■ TEMARIO TEÓRICO:

■ 1. Introducción

- 1.1 Conceptos básicos
- 1.2 Estructura funcional de un computador
- 1.3 Niveles conceptuales de descripción de un computador
- 1.4 Clasificación de computadores
- 1.5 Parámetros que caracterizan las prestaciones de un computador

■ 2. Unidades funcionales de un computador

- 2.1. El procesador
- 2.2. La memoria
- 2.3. Periféricos de E/S
- 2.4. Estructuras básicas de interconexión

■ 3. Representación de la información en los computadores

- 3.1 Representación de textos
- 3.2 Representación de sonidos
- 3.3 Representación de imágenes
- 3.4 Representación de datos numéricos

Asignaturas previas del área

■ Tecnología y **Organización** de Computadores (cont)

■ TEMARIO TEÓRICO:

- 7. Sistemas en el nivel transferencia entre registros (**RTL**)
 - 7.1 Introducción y definiciones generales
 - 7.2 Unidad de procesamiento o camino de datos
 - 7.3 **Unidad de control**
 - 7.4 Introducción a lenguajes de descripción hardware
 - 7.5 Fases de diseño

■ TEMARIO PRÁCTICO:

- Seminarios (S):
 - S1 Herramientas de análisis de la configuración de un computador personal.
 - S2 Identificación componentes computador personal. Montaje e interconexión.
- Prácticas de laboratorio (P):
 - P4 Diseño de una unidad aritmético-lógica.
 - P8 Comprobar el funcionamiento de un camino de datos sencillo.

Presentación de la asignatura

- Ubicación de EC en el Plan de Estudios
- **Temario de la asignatura**
- Horarios, Temporización
- Evaluación, Metodología
- Otras informaciones

Estructura de Computadores

■ TEMARIO TEÓRICO:

- Tema 1. Introducción (repaso-recordatorio de TOC)
 - Unidades funcionales.
 - Conceptos básicos de funcionamiento.
 - Estructuras de bus.
 - Rendimiento.
 - Perspectiva histórica.
- Tema 2. Representación de programas a nivel máquina
 - Codificación de programas. Lenguajes ensamblador y máquina. Ficheros fuente, objeto y ejecutable.
 - Arquitectura del repertorio (ISA). Formatos de datos (tipos y tamaños). Modos de direccionamiento.
 - Instrucciones de transferencia, de E/S, aritmético-lógicas, de control.
 - Procedimientos y subrutinas. Marco de pila. Convenciones de llamada.
 - Arrays. Aritmética de punteros.
 - Estructuras de datos heterogéneas: estructuras, uniones y su alineamiento.

Estructura de Computadores

- Tema 3. **Unidad de control**
 - Camino de datos.
 - Unidades de control **cableadas y microprogramadas**.
 - Control microprogramado.

- Tema 4. Segmentación de cauce
 - Conceptos básicos.
 - **Riesgos** de datos.
 - Riesgos de instrucciones.
 - Influencia en el repertorio de instrucciones.
 - Funcionamiento **superescalar**.
 - Consideraciones relativas a las prestaciones.
 - Ejemplo de funcionamiento.

Estructura de Computadores

- Tema 5. **Entrada/Salida** y buses
 - Funciones del sistema de E/S. Interfaces de E/S.
 - **E/S programada.**
 - **Interrupciones.**
 - **DMA (Acceso directo a memoria).**
 - Estructuras de bus básicas.
 - Especificación de un **bus**: Transferencias. Temporización. Arbitraje.
 - Ejemplos y estándares.
- Tema 6. **Memoria**
 - **Jerarquía** de memoria
 - Concepto de localidad
 - Memorias RAM semiconductoras
 - Memorias de sólo lectura
 - **Prestaciones**: velocidad, tamaño y coste
 - Configuración y diseño de memorias utilizando varios chips
 - Memorias asociativas
 - Memoria **cache**
 - Influencia en las prestaciones

Estructura de Computadores

■ TEMARIO PRÁCTICO:

- Práctica 1: Programación en **ensamblador IA-32**.
 - Verificación de diversos ejercicios en lenguaje C y ensamblador IA-32.
 - Programas aritméticos, representación de datos, marco de pila, etc.
- Práctica 2: Combinar código escrito en distintos **lenguajes, C/ASM**.
 - Llamar desde C a rutina ASM, ensamblador inline, etc.
 - Llamar desde ASM a rutina C, llamada librería C, etc.
- Práctica 3: Bomba digital
 - Ejercicios de **desensamblado y depuración**.
- Práctica 4: Prácticas de procesadores segmentados con **WinDLX**.
 - Ejercicios de optimización.

Estructura de Computadores

■ TEMARIO PRÁCTICO:

- Seminario práctico 1:
 - Entorno de programación. **Ensamblador** en Linux. Depuración. Llamadas al sistema.
- Seminario práctico 2:
 - Llamada y retorno de subrutinas. **Convenciones** de llamada.
- Seminario práctico 3:
 - Desensambladores y **editores** hexadecimal.
- Seminario práctico 4:
 - Simulador **WinDLX**.

Presentación de la asignatura

- Ubicación de EC en el Plan de Estudios
- Temario de la asignatura
- **Horarios, Temporización**
- Evaluación, Metodología
- Otras informaciones

Horarios

(curso 2012-2013)

	Lunes	Martes	Miércoles	Jueves	Viernes
8-09				A3 (2.6) jgp	C3 (2.6) fgm
9-10	A (0.1) acv	tut acv	A (0.1) acv		
10-11	tut acv	A1 (2.6) acv	C (0.3) acv	C1 (2.6) grl	C2 (2.6) fgm
11-12					
12-13	C (0.3) acv	A2 (2.6) acv			
13-14	tut acv				
14-15		tut acv			
15-16					
16-17					
17-18					
18-19					
19-20					
20-21					

Profesorado

(curso 2012-2013)

Profesor	iniciales	grupos
Antonio Cañas Vargas	acv	A , A1 , A2 , C
Francisco Gómez Mula	fgm	C3 , C2
Jesús González Peñalver	jgp	A3
Gustavo Romero López	grl	C1

Aulas

(curso 2012-2013)

- **Grupo A:** **0.1**
- **Grupo C:** **0.3**

- Prácticas: 2.6

Calendario (curso 2012-2013)

Octubre 2012

	L	M	X	J	V	S	D
Semana 2	1	2	3	4	5	6	7
Semana 3	8	9	10	11	12	13	14
Semana 4	15	16	17	18	19	20	21
Semana 5	22	23	24	25	26	27	28
Semana 6	29	30	31				

Diciembre 2012

	L	M	X	J	V	S	D
Semana 10						1	2
Semana ?	3	4	5	6	7	8	9
Semana 11	10	11	12	13	14	15	16
Semana 12	17	18	19	20	21	22	23
	24	25	26	27	28	29	30

Septiembre 2012

L	M	X	J	V	S	D	
24	25	26	27	28	29	30	Semana 1

Noviembre 2012

L	M	X	J	V	S	D	
			1	2	3	4	Semana 6
5	6	7	8	9	10	11	Semana 7
12	13	14	15	16	17	18	Semana 8
19	20	21	22	23	24	25	Semana 9
26	27	28	29	30			Semana 10

Enero 2013

L	M	X	J	V	S	D	
31	1	2	3	4	5	6	
7	8	9	10	11	12	13	Semana 13
14	15	16	17	18	19	20	Semana 14
21	22	23	24	25	26	27	Semana 15
28	29	30	31				

Temporización

(curso 2012-2013)

Fecha	Semana	Temas	Prácticas
24 Sep	Sem. 1	Present. / T1	
1 Oct	Sem. 2	Tema 1 / T2	Seminario 1
8 Oct	Sem. 3	Tema 2	Práctica 1
15 Oct	Sem. 4		
22 Oct	Sem. 5		
29 Oct	Sem. 6		Seminario 2
5 Nov	Sem. 7		Práctica 2
12 Nov	Sem. 8	Tema 3	
19 Nov	Sem. 9		
26 Nov	Sem. 10	Tema 4	Seminario 3
3 Dic	Sem. ?		Práctica 3
10 Dic	Sem. 11	Tema 5	
17 Dic	Sem. 12		Seminario 4
7 Ene	Sem. 13	Tema 6	Práctica 4
14 Ene	Sem. 14		
21 Ene	Sem. 15		

■ Presencial

- Grupo grande: 2h/sem.
 - Teoría
 - Aulas 0.1, 0.3
- Grp. pequeño: 2h/sem.
 - Prácticas y Seminarios
 - Laboratorio 2.6

■ No presencial

- 4h/semana (= presencial)
- estudio, lecturas, ejercicios, actividades...

Presentación de la asignatura

- Ubicación de EC en el Plan de Estudios
- Temario de la asignatura
- Horarios, Temporización
- **Evaluación, Metodología**
- Otras informaciones

Evaluación

■ Teoría: 6p

- Examen final
- Entregas de trabajos durante el curso
- Notas de clase (participación activa, etc.)

■ Prácticas: 4p

- Examen escrito (junto con examen final teoría)
- Entregas de prácticas durante el curso
- Notas de laboratorio (entrevistas, participación activa, etc.)

■ Aprobar requiere...

- «aprobar» (40%) separadamente teoría y prácticas
 - Teoría $\geq 6 \times 0.4 = 2.4$
 - Prácticas $\geq 4 \times 0.4 = 1.6$
- Nota **total** = Teoría+Prácticas ≥ 5.0

Metodología

- **Clases** (2h/s «grupo grande»)
 - Explicación de los **puntos más importantes** del temario
 - Indicación de tareas a realizar no presencialmente (estudio/trabajos)
- **Prácticas** (2h/s «grupo pequeño»)
 - Seminarios (prácticas dirigidas)
 - Guiones de prácticas (a realizar personalmente)
- **Actividades no presenciales (4 horas/semana)**
 - Lectura de material antes de clase/prácticas
 - **Estudio** de material **después** de clase/prácticas
 - Ejercicios, problemas, trabajo de prácticas
 - Actividades propuestas en clase
- **Tutorías**
 - 6h/s cada profesor (de teoría y de prácticas), consultar su horario

SWAD

■ <http://swad.ugr.es>

- El profesor probablemente ya habrá dado de alta a todos los matriculados
- Poner contraseña y foto (si es la primera asignatura en SWAD)
- **Escoger grupo** de prácticas (correspondiente al de teoría)
 - comienzan en la **semana 2**

■ La asignatura aprovecha SWAD para

- Apuntes y **transparencias** de clase
 - Y otros materiales
- Entrega de **actividades**
- Avisos (Post-It)
 - **mensajería** SWAD personalizada
- etc.

Presentación de la asignatura

- Ubicación de EC en el Plan de Estudios
- Temario de la asignatura
- Horarios, Temporización
- Evaluación, Metodología
- Otra información

Bibliografía



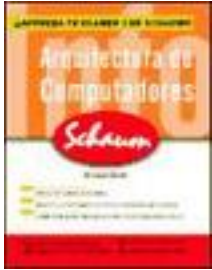
- Computer systems: a programmer's perspective.
 - R.E. Bryant, D.R. O'Hallaron
 - 2ª Ed. International. Pearson, 2011. [ESIIT/C.1 BRY com](#)
 - CS:APP – hay 11 ejemplares en biblioteca – 3 de 1ª ed

- C.V. Hamacher, Z. Vranesic, S. Zaky
 - Organización de Computadores
 - 5ª Ed. McGraw-Hill, 2003. [ESIIT/C.1 HAM org](#)

- W. Stallings
 - Organización y Arquitectura de Computadores
 - 7ª Ed. Pearson Educación, 2008. [ESIIT/C.1 STA org](#)

- Arquitectura de Computadores
 - J. Ortega, M. Anguita, A. Prieto
 - Paraninfo, 2006. [ESIIT/C.1 ORT arq](#)

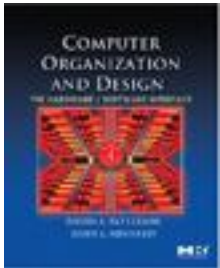
Bibliografía



- N. Carter.
 - Arquitectura de Computadores
 - McGraw-Hill, 2004. [ESIIT/C.1 CAR arq](#)



- A.S. Tanenbaum
 - Structured Computer Organization
 - 5ª Ed. Pearson Education, 2006. [ESIIT/C.1 TAN str](#)



- Patterson/Hennessy
 - Computer Organization and design: the hardware-software interface.
 - 4ª Ed. Elsevier, 2009. [ESIIT/C.1 PAT com](#)

ECTS (European Credit Transfer System)

■ En UGR / ETSIIT

- 1 crédito = 10h presenciales. **1ECTS = 25h**
 - EC: 6 créditos, 6 ECTS, en 15 semanas
 - 6 créditos = 2h/s teoría + 2h/s prácticas = 30 + 30 = 60h presenciales
 - 6ECTS= 60h + 90h no presenciales (60h no pres. + 30h preparación examen)
 - A **cada hora** teoría/prácticas hay que dedicarle **otra hora no presencial**
 - » Y 30h preparación examen
- EC tiene 4h/semana trabajo personal
 - 1ª semana:
 - repasar apuntes **TOC**, estudiar **Tema 1** (Cap.1 del Hamacher)
 - Ir leyendo guión **práctica 1** y **Tema 2** (Cap.2 del libro Bryant/O'Hallaron)
 - Es **importante** leer guión y empezar Tema 2 **para** aprovechar el **Seminario 1**

Otros detalles

■ Prácticas

- Laboratorios:
 - Requiere cuenta ETSIT / arrancar con **Ubuntu**
- Portátil
 - ¿Todos disponen de **portátil o PC** en casa?
 - Instalarse Wubi 10.04 ó VirtualBox + Ubuntu 10.04 LTS 64bit
 - Harán falta 3 paquetes adicionales: *ia32-libs*, *gcc-multilib*, *ddd*
 - Activar firewall con “sudo ufw enable”
- Normas entrega prácticas
 - En principio cada guión indica qué hay que entregar...
 - ...pero guiones comunes a todos los grupos
 - ...**profesores teoría/prácticas** pueden introducir modificaciones
 - Fecha: fin de semana de la última semana de cada práctica (domingo 23:59)
 - Una semana adicional con **penalización por entrega tardía** (siguiente domingo)
 - en SWAD, como «Actividades» (con límite de tiempo)

Teoría 2º A y C



■ Profesor: Antonio Cañas Vargas

- Departamento:
 - ATC, Arquitectura y Tecnología de Computadores
- Despachos:
 - 2D-29 ETSIIT
 - CEVUG (Real de Cartuja, 36)
- Teléfono despacho ETSIIT: 958 24 05 84
- Móvil: 630 87 93 81
- Tutorías:
 - Lu 10-12h, Lu 13-15 h, Ma 9-10 h, Ma 14-15 h
 - acordar cita previamente (swad, tlfno, verbalmente)
- Prácticas
 - Grupos A1 y A2

Teoría 2º A y C

■ Otros profesores de prácticas

- Grupo A3: Jesús González Peñalver
- Grupo C1: Gustavo Romero López
- Grupos C3 y C2: Francisco Gómez Mula

