

## **Grado en Ingeniería Informática**

### **30205 - Arquitectura y organización de computadores 1**

**Guía docente para el curso 2011 - 2012**

**Curso: 1, Semestre: 2, Créditos: 6.0**

---

### **Información básica**

---

#### **Profesores**

- **Carlos José Pérez Jiménez** cjperez@unizar.es
- **Francisco José Martínez Domínguez** fcomardo@unizar.es
- **Darío Suárez Gracia** dario@unizar.es
- **María Villarroya Gaudó** maria.villarroya@unizar.es
- **Juan Pablo López Grao** jpablo@unizar.es
- **Luis Carlos Gállego Rodríguez** lcgalleg@unizar.es

#### **Recomendaciones para cursar esta asignatura**

Para cursar esta asignatura es prerequisito haber cursado la asignatura Introducción a los Computadores.

##### **PROFESORADO**

Profesorado del área de Arquitectura y Tecnología de Computadores, del Departamento de Informática e Ingeniería de Sistemas.

Escuela de Ingeniería y Arquitectura del Campus Río Ebro

María Villarroya Gaudó, maria.villarroya@unizar.es, despacho 0.14 (edificio Ada Byron, Centro Politécnico ) es la profesora responsable de la asignatura en Zaragoza.

Carlos José Pérez Jiménez, cjperez@unizar.es, despacho D.2.12 (edificio Ada Byron).

Escuela Universitaria Politécnica del Campus de Teruel

Francisco José Martínez Domínguez, f.martinez@unizar.es, despacho 3.3 (Escuela Universitaria Politécnica) es el profesor responsable de la asignatura en Teruel.

#### **Actividades y fechas clave de la asignatura**

El calendario de exámenes y las fechas de entrega de trabajos de evaluación se anunciará con suficiente antelación.

---

## Inicio

---

### Resultados de aprendizaje que definen la asignatura

**El estudiante, para superar esta asignatura, deberá demostrar los siguientes resultados...**

**1:**

Conoce a nivel básico los parámetros que definen la arquitectura de lenguaje máquina (repertorio, formato y codificación de instrucciones, almacenes, tipos de datos, modos de direccionamiento, control del secuenciamiento y transferencias de control, gestión de excepciones).

**2:**

Conoce y puede manejar la arquitectura de lenguaje máquina de un procesador de referencia.

**3:**

Distingue los conceptos de lenguaje máquina y ensamblador.

**4:**

Conoce los métodos de representación y codificación de la información y sus operaciones básicas. Es capaz de traducir estructuras de datos y control de lenguajes de alto nivel a ensamblador. Utiliza llamadas a procedimiento.

**5:**

Entiende el modelo genérico de registros de un controlador de dispositivo periférico y los métodos básicos de sincronización y transferencia. Puede programar cualquier dispositivo de E/S y sabe cómo tratar las excepciones.

**6:**

Sabe integrar código ensamblador y rutinas de librería en programas escritos en lenguajes de alto nivel.

## Introducción

### Breve presentación de la asignatura

En esta asignatura se comprenderá cuál es la estructura interna de un computador, en un primer acercamiento a una arquitectura real dentro del Grado en Ingeniería Informática.

---

## Contexto y competencias

---

### Sentido, contexto, relevancia y objetivos generales de la asignatura

**La asignatura y sus resultados previstos responden a los siguientes planteamientos y objetivos:**

Esta asignatura de introducción busca que cada estudiante sea capaz de comprender una arquitectura de lenguaje máquina y diseñar programas en lenguaje ensamblador capaces de comunicarse con periféricos.

### Contexto y sentido de la asignatura en la titulación

Esta asignatura forma parte de la materia básica de Arquitectura de Computadores en el Grado de Ingeniería Informática. La asignatura enlaza con Introducción a los Computadores y es requisito para cursar Arquitectura y Organización 2.

## **Al superar la asignatura, el estudiante será más competente para...**

- 1:**  
Resolver problemas y tomar decisiones con iniciativa, creatividad y razonamiento.
- 2:**  
Aplicar las tecnologías de la información y las comunicaciones en ingeniería.
- 3:**  
Utilizar conocimientos básicos sobre el uso y programación de los ordenadores, bases de datos y programas informáticos con aplicación en ingeniería.
- 4:**  
Aplicar conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.

Estas dos últimas competencias se desarrollan de forma específica para reforzar las dos primeras, persiguiendo los resultados de aprendizaje de la asignatura (léase apartado Introducción, resultados de aprendizaje de esta guía).

## **Importancia de los resultados de aprendizaje que se obtienen en la asignatura:**

Esta asignatura fundamenta el diseño, programación y uso eficiente del computador, ya sea este de propósito general o específico (empotrado, supercomputación, etc.).

---

## **Evaluación**

---

### **Actividades de evaluación**

#### **El estudiante deberá demostrar que ha alcanzado los resultados de aprendizaje previstos mediante las siguientes actividades de evaluación**

- 1:**

La evaluación de la asignatura se realiza en base a dos pruebas:

**P1.** Prueba escrita en la que responder cuestiones y resolver ejercicios y problemas. Se requiere una nota mínima de 5.0 puntos en esta prueba para aprobar la asignatura. Si se obtiene esta nota mínima, entonces la prueba pondera un 70% en la nota de la asignatura y, si no se alcanza este mínimo, entonces la calificación en la asignatura es la de esta prueba.

**P2.** Trabajos y prueba de laboratorio. Esta prueba pondera un 30% en la nota de la asignatura. Cada alumno deberá entregar los trabajos que se indiquen en las prácticas de la asignatura y presentarse a una prueba de laboratorio donde deberán realizar alguna tarea relacionada con una parte del trabajo entregado. Se valorará tanto el trabajo entregado como el realizado en la prueba, con pesos análogos. El alumno que no presente los trabajos de programación que se indiquen o que no se presente a la prueba será calificado con un cero.

**Pruebas con carácter voluntario.** A lo largo del cuatrimestre se plantearán varias pruebas voluntarias consistentes en la resolución, por escrito en clase o en casa, de ejercicios y problemas. El 10% de la calificación de estas pruebas, es decir, entre 0 y 1.0 puntos, se sumará a la calificación obtenida por el alumno en la prueba **P1**, siempre y cuando haya aprobado la asignatura ( $0.7*P1+0.3*P2>=5.0$ ).

**Convocatoria de Septiembre.** La evaluación de la asignatura se realiza en base a dos pruebas análogas a las de la convocatoria de Junio, con las mismas ponderaciones y exigencia de notas mínimas. Las calificaciones del alumno obtenidas en la convocatoria de Junio en cualquier de las pruebas (P1 y P2) se mantienen en Septiembre, salvo que el alumno opte por presentarse a la prueba correspondiente en esta nueva convocatoria, en cuyo caso prevalecerá la

nueva calificación.

---

## Actividades y recursos

---

### Presentación metodológica general

**El proceso de aprendizaje que se ha diseñado para esta asignatura se basa en lo siguiente:**

1. Estudio y trabajo continuado desde el primer día de clase.
2. Aprendizaje de conceptos a través de las clases magistrales impartidas en el aula.
3. Aplicación de esos conocimientos a la resolución de ejercicios y/o aplicaciones de teoría en las clases de problemas (grupos reducidos).
4. Clases de prácticas en laboratorio en las que el alumno aprenderá la tecnología necesaria para poner a punto programas en ensamblador utilizando un entorno de programación. Será preciso trabajo personal del alumno de preparación de las actividades y entregas en las sesiones prácticas.
5. Así mismo se realizarán tutorías no académicas y podrían complementarse con otro tipo de actividades formativas voluntarias (seminarios, por ejemplo).

### Actividades de aprendizaje programadas (Se incluye programa)

**El programa que se ofrece al estudiante para ayudarle a lograr los resultados previstos comprende las siguientes actividades...**

**1:**

1. En las clases impartidas en el aula se desarrollará el programa de la asignatura.
2. En las clases de problemas se resolverán problemas de aplicación de los conceptos y técnicas presentadas en el programa de la asignatura.
3. Las sesiones de prácticas se desarrollan en un laboratorio informático. En cada sesión el alumno deberá realizar un trabajo, individualmente o en equipo, poniendo a punto uno o varios programas.

## Planificación y calendario

### Calendario de sesiones presenciales y presentación de trabajos

La organización docente de la asignatura es la siguiente.

- Clases teóricas (2 horas semanales)
- Clases de problemas (1 hora semanal)
- Clases prácticas de laboratorio (2 horas cada dos semanas). Son sesiones de trabajo de programación en laboratorio, tuteladas por un profesor, en las que participan los alumnos en grupos reducidos.
- Tutorías y actividades de evaluación

Los horarios de todas las clases y las fechas de las sesiones de prácticas se anunciarán con suficiente antelación a través de las webs del centro y de la asignatura.

## Programa de la asignatura

1. Arquitectura del Procesador: Interpretación y traducción, lenguaje máquina y ensamblador, entorno de desarrollo,

- representación y codificación de la información, operaciones básicas, almacenes, modos de direccionamiento, repertorio de instrucciones, traducción de estructuras de datos y control de lenguajes de alto nivel, llamadas a procedimiento.
2. Subsistema de E/S. Modelo genérico de registros de controlador de dispositivo. Métodos básicos de sincronización y transferencia. Excepciones. Integración de periféricos en microcontroladores.
  3. Caso práctico. Integración de código de alto nivel con código ensamblador y rutinas de biblioteca.

## Trabajo del estudiante

La dedicación del estudiante para alcanzar los resultados de aprendizaje en esta asignatura se estima en 150 horas distribuidas del siguiente modo:

- 56 horas, aproximadamente, de actividades presenciales (clases teóricas, de problemas y prácticas en laboratorio)
- 51 horas de estudio personal efectivo (estudio de apuntes y textos, resolución de problemas, preparación clases y prácticas, desarrollo de programas)
- 40 horas de trabajo de programación en equipo
- 3 horas de examen final escrito

## Bibliografía

### Bibliografía de la asignatura

Bibliografía básica:

1. **ARM Assembly Language: Fundamentals and Techniques**, William Hohl. CRC press. 2009

Bibliografía complementaria:

1. **ARM System-on-Chip Architecture** (2nd ed.), Steve Furber. Addison-Wesley. 2000
2. **ARM Architecture Reference Manual** (2nd ed.), David Seal. Addison-Wesley. 2000
3. **ARM System Developer's Guide: Designing and Optimizing System Software**, Andrew N. Sloss, Dominic Symes, Chris Wright. Elsevier. 2004
4. **Organización y Arquitectura de computadores** (7<sup>a</sup> ed.). W. Stallings. Prentice Hall. 2006

### Referencias bibliográficas de la bibliografía recomendada

- Furber, Stephen B.|q(Stephen Bo). ARM system-on-chip architecture / Steve Furber . - 2nd ed. Harlow : Addison-Wesley, 2000
- Hohl, William. ARM Assembly language : fundamentals and techniques / William Hohl Boca Raton (Florida) : CRC Press, cop. 2009
- Sloss, Andrew. ARM system developer's guide : designing and optimizing system software / Andrew Sloss, Dominic Symes, Chris Wright ; whit a contribution by John Rayfield San Francisco, CA : Elsevier/ Morgan Kaufman, cop. 2004