



Galileo
UNIVERSIDAD
La Revolución en la Educación

FACULTAD DE INGENIERÍA DE SISTEMAS,
INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN
**CIENCIAS DE LA COMPUTACIÓN III:
ESTRUCTURAS DE MÁQUINAS**

Catedrático:	Ali Lemus (alilemus@galileo.edu) Luis Jimenez (luisclu@galileo.edu)
Auxiliares de clase:	Iván Juárez (alexanderj@galileo.edu) Carlos Hermosilla (efrainh12@galileo.edu) Rocío Santos (14003112@galileo.edu)
Auxiliares de laboratorio:	Juan Cardona (15000787@galileo.edu) Kevin Hernández (hernandez.kevin@galileo.edu) Andrés Castellanos (andres.cv@galileo.edu) Francisco Hernández (francisco.hernandez@galileo.edu) Guillermo Maldonado (guiller@galileo.edu) Daniel Passarelli (17001393@galileo.edu) Diego Calderón (diegocdl@galileo.edu) Evelyn Cruz (evecruz@galileo.edu)

Objetivo del curso

En cursos anteriores han aprendido a programar en un nivel bastante alto de abstracción. Acá nos concentraremos en la máquina y cómo esta ejecuta los programas que ustedes escriben.

En este curso nos preocuparemos no tanto por el proceso de escribir un programa de computadora, sino más bien en cómo la computadora ejecuta los programas. Es decir, el tema principal de este curso es la estructura de la máquina: los circuitos electrónicos, y las operaciones computacionales que estos circuitos llevan a cabo.

Para concretizar estas ideas, vamos a estudiar con cierto detalle el ISA RISC-V. Usamos este ISA pues es más sencillo que otras familias, mientras que posee características deseables como paralelismo.

El detalle semanal de los temas puede ser consultado en la sección de Material de Apoyo del GES.

Competencias

- Comprende todas las fases del proceso de ejecución de un programa computacional escrito en código de alto nivel, incluyendo manejo de memoria, arquitectura del set de instrucciones y las operaciones computacionales que se llevan a cabo en los circuitos electrónicos.
- Identifica los componentes principales que forman parte del diseño de la arquitectura de una computadora.
- Diseña sistemas de hardware y software apoyándose en su conocimiento de los diferentes niveles de abstracción de la estructura y organización de una computadora.
- Implementa sistemas de hardware y software, a través de simulaciones, apoyándose en su conocimiento de los diferentes niveles de abstracción de la estructura y organización de una computadora.

Prerrequisitos

Para este curso deben sentirse cómodos con la idea de programar, y en la capacidad de aprender lenguajes de programación nuevos sin demasiado esfuerzo. Todo el código que no sea escrito en lenguaje ensamblador será escrito en lenguaje C y Python. Es importante que tengan muy claros los conceptos aprendidos en el curso de estructuras de datos que tomaron en el segundo semestre.

Horario del curso

El curso consta de 7 créditos académicos (CA) distribuidos en el siguiente horario:

- Clases Magistrales (4 horas a la semana)
- Laboratorios (2 horas a la semana)
- Hoja de trabajo o discusión (1 hora a la semana)
- Lecturas y cuestionarios semanales (vía GES)
- Proyectos, tareas y estudio personal

Evaluación del curso

Exámenes	30
Proyectos	30
Laboratorios	20
Cuestionarios semanales	10
Discusiones	10
EPA	5
Extras	5
Nota Final	110

Adicionalmente se tendrán actividades con puntos extras. Los puntos extras solo serán tomados en cuenta si su nota en el curso es de 55 puntos o superior.

Requisitos adicionales de aprobación

Para aprobar el curso, el estudiante debe de cumplir con los siguientes requisitos adicionales:

- Asistencia mínima a clase: 80%
- Asistencia mínima a laboratorio: 80%

Entregas tarde / Casos especiales

Utilizamos un sistema de redundancia y puntos extras con el objetivo que el estudiante que falta a un par de tareas o cuestionarios aun pueda obtener la nota de 100 puntos en la clase en la clase, sin embargo, si el estudiante completa todas las tareas y nunca falta, podrá obtener una nota superior a 100 puntos (por cuestiones administrativas la nota máxima en expediente es de 100 puntos).

Además, implementamos un sistema de “Slip Days” con el cual el estudiante tiene 3 días de entrega tarde en cualquier proyecto. El estudiante decide en que proyecto utilizar sus 3 días de entrega tardía.

Bibliografía

The C Programming Language (2nd Edition)

Autores: Kernighan y Ritchie

Editorial: Prentice Hall

Alias: K&R

El estándar es K&R, escrito por los inventores de C. El libro contiene muchos ejemplos que deberían transcribir y no solo leer, además de muchos ejercicios para practicar. K&R no enseña a programar, sino que le enseña a utilizar C a alguien que ya sabe programar.

Computer Organization and Design – RISC-V Edition

Autores: Patterson y Hennessy

Editorial: Elsevier

Alias: COD

David Patterson y John Hennessy son diseñadores de arquitecturas RISC y sus libros de COD son la mejor opción para pregrado. El libro describe la interfaz entre software y hardware, es decir lleva desde un lenguaje de alto nivel hasta cómo está diseñado el hardware para ejecutar cada instrucción.

Guía Práctica de RISC-V: El Atlas de una Arquitectura Abierta

Autores: Patterson y Waterman

Introducción concisa y referencia para programadores de sistemas embebidos, estudiantes y curiosos a una arquitectura moderna, popular y abierta. RISC-V abarca desde los microcontroladores embebidos más baratos de 32 bits hasta las computadoras más rápidas de 64 bits en la nube. El texto muestra cómo RISC-V siguió las buenas ideas de arquitecturas del pasado, evitando al mismo tiempo sus errores.

Descarga gratuita: <http://riscvbook.com/espanol/>

The Datacenter as a Computer

Autores: Barroso y Hölzle

El tercer libro es una introducción al diseño de “Warehouse Scale Machines” o Maquinas del tamaño de un almacén. Estos nuevos centros de cómputo masivos ya no pueden verse como una colección de servidores puesto que se necesita que trabajen al unísono para poder prestar sus servicios de forma eficiente.

Descarga gratuita: <http://inst.eecs.berkeley.edu/~cs61c/resources/WSCBarrosoHolzle.pdf>