



Arquitectura de Computadoras

1. Características generales

Nombre:	Arquitectura de Computadoras
Sigla:	CI-0120
Créditos:	4
Horas lectivas:	5 horas de teoría
Requisitos:	CI-0114 Fundamentos de Arquitectura, CI-0118 Lenguaje Ensamblador
Correquisitos:	Ninguno
Clasificación:	Curso propio
Ciclo de carrera:	I ciclo, 3er año (énfasis de Ciencias de la Computación)
Docente(s):	Ricardo Gang
Datos de contacto:	ricardo.gang@ucr.ac.cr, Oficina 3-25
Grupo:	1
Semestre y año:	I ciclo 2025
Horario y lugar de clases:	Aula 203 L13-15:30, J13-14:30
Horario y lugar de consulta:	Of 3-25, M1:30PM-4PM

2. Descripción

Este curso está dirigido a estudiantes que inician su tercer año de bachillerato. Se enfoca en el estudio de distintas arquitecturas de computadoras para permitirle al estudiante tomar decisiones acertadas sobre las capacidades de un computador para ejecutar un determinado programa de manera eficiente, especialmente en ambientes de ejecución paralela. Se requieren conocimientos amplios de programación en C/C++, ensamblador y lenguaje de máquina. Durante el curso, el estudiante adquiere un mayor criterio que le permite evaluar y comparar el rendimiento de diferentes arquitecturas de computadoras. El curso puede darse bajo un enfoque por proyectos o mediante una combinación de clases magistrales, estudio independiente y tareas con el uso de laboratorios.





3. Objetivos

Objetivo General

El *objetivo general* del curso es que cada estudiante comprenda los principios de diseño que rigen las principales arquitecturas de computadoras para determinar su impacto en el rendimiento de los programas.

Objetivos Específicos

Durante este curso cada estudiante desarrollará habilidades para:

1. Relacionar la arquitectura de una computadora con su conjunto de instrucciones.
2. Utilizar técnicas que permitan obtener estadísticas dinámicas sobre los tipos de instrucciones ejecutadas.
3. Realizar y evaluar diferentes mediciones de desempeño de procesadores con el fin de comprender cómo y por qué la arquitectura del procesador en el que se trabaja es un elemento determinante en el comportamiento de las aplicaciones de software que ahí se ejecuten.
4. Analizar críticamente un microprocesador calculando medidas de rendimiento con base en sus elementos y la organización de su arquitectura.
5. Comparar y contrastar distintas organizaciones de memoria.
6. Aplicar técnicas que aprovechen el paralelismo inherente a un proceso.
7. Explicar las diferencias que existen entre distintos procesadores y multiprocesadores con implementaciones de paralelismo de hilos, de procesos y de datos.

4. Contenidos

Objetivo específico	Eje Temático	Desglose
1 y 2	Fundamentos de diseño	El diseño de computadoras. Clases de computadoras. Definición de la arquitectura de una computadora. Tendencias en la tecnología de circuitos integrados y sus costos, consumo de energía y eficiencia. Medición del desempeño. Principios cuantitativos en el diseño de computadores.
1 y 3	Arquitectura basada en el conjunto de instrucciones	Clasificación. Direccionamiento de memoria. Modos de direccionamiento para procesamiento de señales. Tipo y tamaño de los operandos.



		Operandos para procesamiento de señales y multimedia. Operaciones para procesamiento de señales y multimedia. Instrucciones para el control de flujo. Codificación de un conjunto de instrucciones. Rol de los compiladores. Relación con lenguajes de alto nivel. Arquitectura MIPS.
1, 6 y 7	Paralelismo a nivel de instrucciones y ejecución segmentada (<i>pipeline</i>)	Definición de <i>pipelining</i> . Problemas con el pipeline: conflictos de datos y de control. Dificultades para implementar <i>pipelining</i> (excepciones). Pipeline para operaciones multiciclo y ejemplo del <i>pipeline</i> para el procesador MIPS. Procesadores superescalares (de emisión múltiple).
5	Jerarquías de memoria	Introducción a las memorias caché. <i>Translation lookaside buffers</i> (TLB). Tecnología de memoria y optimizaciones. Desempeño de la caché. Reducción de tasa de fallos de caché. Diseño de jerarquías de memoria y memoria virtual. Protección y ejemplos de memoria virtual.
4, 6 y 7	Paralelismo a nivel de datos	Arquitecturas vectoriales, SIMD (<i>single instruction, multiple data</i>) y GPU (<i>graphics processing unit</i>). Extensiones SIMD del conjunto de instrucciones para multimedios.
6 y 7	Paralelismo a nivel de hilos	Arquitecturas de memoria compartida centralizada. Multiprocesadores simétricos (SMP), Acceso uniforme a memoria (UMA). Desempeño de los multiprocesadores de memoria compartida simétrica. Coherencia de caché mediante protocolos <i>snooping</i> . Arquitecturas de memoria compartida distribuida no uniforme (NUMA). Coherencia de caché mediante protocolos de directorios.
5	Almacenamiento externo	Discos duros, memorias <i>flash</i> y discos de estado sólido (SSD).
3	Redes de interconexión y clusters	Creación de procesos en ambientes de multiprocesadores.



5. Metodología

Este es un curso principalmente teórico. La teoría es, entonces, el marco de trabajo y la programación el fin y el medio para alcanzar los objetivos. La participación del estudiante le permitirá adquirir destreza y claridad para el éxito en el curso. Se evaluará la comprensión de los conceptos vistos en clase así como iniciativa propia y creatividad para la resolución de problemas. El curso es BAJO VIRTUAL y los contenidos se cubrirán, en horario de clase, por medios varios incluyendo conferencias y material en multimedios. Se usará la plataforma institucional de Mediación Virtual para enunciados, entregas y el correo electrónico de la UCR para comunicaciones oficiales. Las clases virtuales serán por medio de este enlace <https://udecr.zoom.us/j/84770135063>. La consulta se hará por correo, videoconferencia y/o mensajes de la plataforma.

6. Evaluación

Toda actividad de evaluación medirá dominio teórico y práctico de manera acumulativa además de originalidad y creatividad. El curso tendrá actividades teóricas y prácticas, una investigación y dos exámenes. Se recuerda que es prohibido suplantar una respuesta original y propia por una generada, total o parcialmente, por una herramienta de generación automática de respuestas.

TRABAJO	
INVESTIGACIÓN	20%
TAREAS	20%
EVALUACIONES	
PRUEBAS CORTAS	20%
EXAMEN 1	20%
EXAMEN FINAL	20%

Aprobación: 6.75+ (ó 7+ en Ampliación)

Derecho de Ampliación: 5.75-6.74



7. Cronograma

	FECHA
I	29/5 FECHA DE ENTREGA FINAL: 26/6
E1	5/5
EF	30/6

Incluir al menos fechas de exámenes y de elementos de evaluación cuyo puntaje singular sea significativo.

8. Bibliografía

[1] William Stallings. Computer Organization and Architecture. Pearson, 10ª edición (2015).

[2] David A. Patterson y John L. Hennessy. Computer Organization and Design. Elsevier Inc. / Morgan Kaufmann Publishers, San Francisco, 5.a edición (2014).

[3] John L. Hennessy y David A. Patterson. Computer Architecture: A Quantitative Approach. Elsevier Inc. / Morgan Kaufmann Publishers, San Francisco, 5ª edición (2012)

9. Recursos estudiantiles

Para información sobre recursos estudiantiles disponibles en la UCR, incluyendo el Sistema de bibliotecas y la normativa universitaria vigente, favor visitar la página: <https://www.ecci.ucr.ac.cr/vida-estudiantil/servicios-institucionales-para-estudiantes/guia-de-re-cursos-estudiantiles-de-la-ucr>.