

PROGRAMA DE CURSO ARQUITECTURA DE COMPUTADORES

A. Antecedentes generales del curso:

Departamento	Ingeniería Eléctrica (DIE)					
Nombre del curso	Arquitectura de computadores	Código	EL4102	Créditos	6	
Nombre del curso en inglés	<i>Computer Architecture</i>					
Horas semanales	Docencia	3	Auxiliares	1,5	Trabajo personal	5,5
Carácter del curso	Electivo: Núcleo línea de especialización					
Requisitos	EL4002/EL3102					

B. Propósito del curso:

Los procesadores son sistemas digitales que se han desarrollado de una forma vertiginosa desde los años 80 y han tomado su camino propio. En este contexto, el curso tiene como propósito que el estudiantado analice y diseñe softwares para programación de bajo nivel (lenguaje assembly) de procesadores, utilizando herramientas de simulación como MARS 4.5 (simulador para MIPS) y RARS 1.5 (simulador para RISC-V), disponibles en la web, a fin de resolver problemas concretos en forma eficiente y efectiva.

Para alcanzar este propósito, se analizan las diferentes arquitecturas de diseño de un procesador desarrolladas a través del tiempo, considerando su evolución en la actualidad.

Además, se identifican, comprenden y se usan terminología y conceptos para la descripción y especificaciones de estos dispositivos.

El curso tributa a las siguientes competencias específicas (CE) y genéricas (CG):

CE2: Concebir y aplicar conocimientos de ciencias físicas y matemáticas para el desarrollo de soluciones tecnológicas a problemáticas de la Ingeniería Eléctrica y áreas afines.

CE5: Resolver problemas y optimizar soluciones en el ámbito de la Ingeniería Eléctrica utilizando conceptos, enfoques y metodologías apropiadas.

CE7: Concebir, implementar y gestionar proyectos tecnológicos en el ámbito de la Ingeniería Eléctrica, considerando para tal efecto requerimientos técnicos, económicos, ambientales, sociales y éticos.

CG1: Comunicación académica y profesional

Comunicar en español de forma estratégica, clara y eficaz, tanto en modalidad oral como escrita, puntos de vista, propuestas de proyectos y resultados de investigación fundamentados, en situaciones de comunicación compleja, en ambientes sociales, académicos y profesionales.

CG2: Comunicación en inglés

Leer y escuchar de manera comprensiva en inglés una variedad de textos e informaciones sobre temas concretos o abstractos, comunicando experiencias y opiniones, adecuándose a diferentes contextos y a las características de la audiencia.

CG3: Compromiso ético

Actuar de manera responsable y honesta, dando cuenta en forma crítica de sus propias acciones y sus consecuencias, en el marco del respeto hacia la dignidad de las personas y el cuidado del medio social, cultural y natural.

CG4: Trabajo en equipo

Trabajar en equipo, de forma estratégica y colaborativa, en diversas actividades formativas, a partir de la autogestión de sí mismo y de la relación con el otro, interactuando con los demás en diversos roles: de líder, colaborador u otros, según requerimientos u objetivos del trabajo, sin discriminar por género u otra razón.

C. Resultados de aprendizaje:

Competencias específicas	Resultados de aprendizaje
CE7	RA1: Analiza el desarrollo y evolución de los procesadores a lo largo del tiempo, considerando las tecnologías más utilizadas en la actualidad y los nuevos estándares exigidos por la industria.
CE5, CE7	RA2: Determina cómo funciona un procesador, considerando criterios asociados a diferentes arquitecturas de diseño, así como las relaciones y funciones de los distintos componentes de software y hardware para tomar decisiones respecto de la programación.
CE5	RA3: Evalúa el funcionamiento y la performance o rendimiento de sistemas computacionales, considerando relaciones y funciones de los distintos componentes del software y hardware.
CE2, CE5	RA4: Desarrolla programas en un lenguaje de bajo nivel (assembly) para procesadores, en niveles de complejidad creciente en el desarrollo de cada tarea, a fin de resolver problemas concretos en forma óptima y eficiente, utilizando herramientas de simulación, por ejemplo, MARS 4.5 y RARS 1.5, entre otras.
Competencias genéricas	Resultados de aprendizaje
CG1	RA5: Elabora reporte de las experiencias de laboratorio y programación de softwares de procesadores desarrollados y hardware, donde informa, con claridad, el análisis de los resultados obtenidos, considerando si el o los algoritmos funcionan, cómo se comporta dicho programa, y qué ocurre en la entrada y salida.

CG2	RA6: Lee de manera comprensiva en inglés manuales y textos sobre arquitectura de computadores, sintetizando información (jerga técnica y especificaciones) aplicable a la programación y avances tecnológicos.
CG3, CG4	RA7: Elabora, de manera individual o con su equipo, tareas y ejercicios solicitados, trabajando de manera responsable y con respeto por las ideas de sus pares al momento de establecer acuerdos y tomar decisiones sobre una actividad.

D. Unidades temáticas:

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
1	RA1	Niveles de Abstracción en los Computadores y Tecnologías	1,5 semanas
Contenidos		Indicador de logro	
1.1. La revolución del computador. 1.2. Clases de Computadores (desde 1960). 1.3. El significado de “performance”. 1.4. Los ocho grandes conceptos en arquitectura de computadores. 1.5. Anatomía de un computador. 1.6. Niveles de abstracción. 1.7. Tendencias de la Tecnología. 1.8. El porqué del paso de uniprosesadores a multiprosesadores. 1.9. Análisis de “Benchmarks” para los procesadores. 1.10. Errores de conceptos y falacias. 1.11. Síntesis de los aspectos más relevantes sobre nivel de abstracción y tecnologías de los computadores.		El/la estudiante: 1. Analiza el desarrollo y evolución de los procesadores, considerando el porqué del paso de uniprosesadores a multiprosesadores, las tendencias tecnológicas más utilizadas, así como estándares exigidos por la industria. 2. Identifica y analiza conceptos básicos relacionados con la arquitectura de los computadores, tales como performance o rendimiento, niveles de abstracción, entre otros. 3. Determina la importancia de la performance o rendimiento en un procesador, explicando su importancia para la programación o desarrollo de softwares.	
Bibliografía de la unidad		[1] Texto base: Patterson y Hennessy (2012), capítulo 1. Patterson y Hennessy (versión 2018), capítulo 1*.	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
2	RA2, RA3, RA4, RA5, RA6, RA7	Instrucciones: el lenguaje del computador	2 semanas
Contenidos		Indicador de logro	
2.1. El Lenguaje del Computador. 2.1.1. Operaciones del Hardware del Computador. 2.1.2. Operandos del Hardware del Computador. 2.1.3. Números Con y Sin Signo. 2.2. Representación de Instrucciones en el Computador. 2.3. Operaciones Lógicas. 2.4. Instrucciones para tomar decisiones. 2.5. Procedimientos en el Hardware del Computador. 2.6. Comunicación con las personas. 2.7. Direccionamiento para Constantes y Direcciones Inmediatas en MIPS y RISC-V. 2.8. Paralelismo e Instrucciones: sincronización. 2.9. Traducción y ejecución de un programa. 2.10. Casos reales: instrucciones del x86. 2.11. Errores de conceptos y falacias. 2.12. Conclusiones sobre la importancia del lenguaje para procesadores, MIPS y RISC-V, entre otros.		El/la estudiante: 1. Identifica y analiza las principales operaciones del hardware de un computador, considerando las relaciones y funciones de sus distintos componentes. 2. Utiliza diferentes herramientas de simulación, disponibles en la Web, tales como MARS 4.5 (simulador para MIPS) y RARS 1.5 (simulador para RISC-V), entre otros, para programar. 3. Programa un software en un lenguaje de bajo nivel (assembly) para resolver un problema o desafío dado, considerando la performance o rendimiento. 4. Lee en inglés, de manera comprensiva, sobre el desarrollo de programas en lenguaje assembly, sintetizando información aplicable para programar. 5. Reporta, por escrito, los resultados la programación de un software simple de bajo nivel, informando sobre el comportamiento del programa y si el o los algoritmos funcionan. 6. Trabaja, de manera individual o en equipo, en las tareas o actividades académicas, cumpliendo con responsabilidad la solicitud.	
Bibliografía de la unidad		[1] Texto base: Patterson y Hennessy (2012), capítulo 2. Patterson y Hennessy (versión 2014), capítulo 2*. Patterson y Hennessy (versión 2018), capítulo 2*.	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
3	RA2, RA3, RA4, RA5, RA6, RA7	Aritmética para computadores	1 semana
Contenidos		Indicador de logro	
<p>3.1. Aritmética para Computadores.</p> <p>3.1.1. Sumas y restas.</p> <p>3.1.2. Multiplicaciones.</p> <p>3.1.3. Divisiones.</p> <p>3.2. Punto flotante (estándar IEEE 754).</p> <p>3.3. Paralelismo y aritmética de computadores: asociatividad y Paralelismo a nivel de datos.</p> <p>3.4. Errores de conceptos y falacias.</p> <p>3.5. Conclusiones sobre la importancia de la aritmética para computadores.</p>		<p>El/la estudiante:</p> <ol style="list-style-type: none"> 1. Representa los números decimales en <i>bits</i>, a partir de un problema que debe resolver. 2. Resuelve problemas asociados a la programación, considerando esquemas de representación interna de números enteros, reales, entre otros. 3. Utiliza diferentes herramientas de simulación, disponibles en la Web, como MARS 4.5 (simulador para MIPS) y RARS 1.5 (simulador para RISC-V) para programar. 4. Programa en assembly, en un nivel creciente de complejidad respecto de los requerimientos de la tarea, para resolver un problema dado, considerando las operaciones aritméticas. 5. Reporta en un informe el programa desarrollado, enunciando y describiendo, con claridad, las fases involucradas en el proceso. 6. Lee en inglés sobre la programación de la aritmética de computadores y ejemplos asociados, extrayendo información que utiliza para el desarrollo de sus programas. 7. Ejecuta las actividades, de manera individual o en equipo según corresponda, cumpliendo con las entregas en los plazos asignados. 	
Bibliografía de la unidad		<p>[1] Texto base: Patterson y Hennessy (2012), capítulo 3.</p> <p>Patterson y Hennessy (versión 2014), capítulo 3*.</p> <p>Patterson y Hennessy (versión 2018), capítulo 3*.</p>	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
4	RA2, RA3, RA4, RA5, RA6, RA7	Arquitecturas y evolución de los procesadores	6 semanas
Contenidos		Indicador de logro	
4.1. El procesador. 4.2. La Máquina de Turing; el procesador más básico. 4.3. Convenciones del diseño lógico. 4.4. Diseño e Implementación del Sistema Controlado (“Datapath”). 4.5. La Arquitectura “Single-Cycle”. 4.6. La Arquitectura “Multi-Cycle”. 4.7. Introducción a la arquitectura “Pipeline”. 4.8. “Datapath” y control de la Arquitectura “Pipeline”. 4.9. “Hazards” de datos: “Forwarding” versus “Stalling”. 4.10. “Hazards” de control. 4.11. Excepciones e Interrupciones. 4.12. Paralelismo y paralelismo avanzado a nivel de Instrucciones. 4.13. Ejemplos de casos reales. 4.14. Errores de conceptos y falacias.		El/la estudiante: 1. Identifica diferentes arquitecturas de diseño de un procesador. 2. Analiza cómo se implementa un procesador, considerando el datapath y el controlador. 3. Compara el comportamiento de diferentes arquitecturas de un procesador (Single-Cycle, Multi-Cycle y pipeline). 4. Elabora un informe sobre el comportamiento de diferentes estructuras de un procesador (Single-Cycle, Multi-Cycle y pipeline), considerando el uso de lenguaje formal y técnico, así como claridad, organización y precisión en la presentación de sus ideas y resultados. 5. Lee en inglés textos para extraer conceptos y especificaciones técnicas con las cuales trabajar en una tarjeta de desarrollo. 6. Trabaja, con su equipo, en forma física con un procesador considerando el uso de una aplicación para programar. 7. Resuelve un problema, programando una tarjeta de desarrollo.	
Bibliografía de la unidad		[1] Texto base: Patterson y Hennessy (2012), capítulo 4. Patterson y Hennessy (versión: 2018), capítulo 4*.	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
5	RA2, RA3, RA4, RA5	La memoria y su esquema jerárquico	2 semanas
Contenidos		Indicador de logro	
5.1. La Memoria. 5.2. Introducción a la Memoria Cache. 5.3. Medidas y Mejoras de la Performance de la Memoria Cache. 5.4. Memoria Virtual. 5.5. Una Infraestructura Común para las Jerarquías de Memoria (memorias caché L1, L2 y L3). 5.6. Máquinas Virtuales. 5.7. El uso de una Máquina de estado finito (FSM) para implementar el control de una Memoria Cache Básica. 5.8. Paralelismo y jerarquías de memoria: Coherencia de la Memoria Cache. 5.9. Ejemplo del diseño e implementación de un controlador para una Memoria Cache. 5.10. Ejemplos de casos reales: jerarquías de memorias en ARM e Intel I7. 5.11. Errores de conceptos y falacias.		El/la estudiante: 1. Identifica y compara diferentes arquitecturas de memoria asociadas al diseño de un procesador. 2. Analiza cómo se implementan los diferentes tipos de memorias cache, considerando, según su uso, las distintas performance o rendimiento. 3. Compara y evalúa el comportamiento de diferentes arquitecturas de memorias caché. 4. Diseña e implementa el control de una memoria caché básica. 5. Informa los resultados del diseño e implementación de un controlador, justificando con argumentos claros y precisos sobre el procedimiento ejecutado, los ajustes realizados, entre otros aspectos.	
Bibliografía de la unidad		[1] Texto base: Patterson y Hennessy (2012), capítulo 5. Patterson y Hennessy (versión 2018), capítulo 5*.	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
6	RA2, RA3, RA4	Almacenamientos y dispositivos de entrada/salida (E/S)	1,5 semana
Contenidos		Indicador de logro	
6.1. Almacenamientos y dispositivos de E/S. 6.2. Fiabilidad, confiabilidad y disponibilidad de los dispositivos de E/S. 6.3. Almacenamiento en discos. 6.4. Almacenamiento en memorias flash. 6.5. Interconexión entre procesadores, memorias y dispositivos de E/S. 6.6. Interfaz entre Dispositivos de E/S, procesadores, memoria y el sistema operativo. 6.7. Medidas de performance de los dispositivos de E/S: Ejemplos de discos y sistemas de archivos. 6.8. Diseño de un sistema de E/S. 6.9. Paralelismo y E/S: estructuras de discos RAID. 6.10. Conceptos de redes de computadores. 6.11. Errores de conceptos y falacias.		El/la estudiante: <ol style="list-style-type: none"> 1. Evalúa la implementación de dispositivos de almacenamiento y dispositivos de entrada y salida (E/S). 2. Compara la <i>performance</i> entre dispositivos de E/S, procesadores, memoria y el sistema operativo, en base a su eficiencia. 	
Bibliografía de la unidad		[1] Texto base: Patterson y Hennessy (2012): capítulo 6.	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
7	RA1, RA3	Multicores, multiprocesadores y clusters	1 semana
Contenidos		Indicador de logro	
<p>7.1. Procesadores paralelos: desde el cliente hasta la “Nube”.</p> <p>7.2. La dificultad de crear programas con procesamiento paralelo.</p> <p>7.3. Multiprocesadores con memoria compartida.</p> <p>7.4. <i>Clusters</i> y otros multiprocesadores “Pasa-Mensajes”.</p> <p>7.5. Implementación en Hardware del “Multithreading”.</p> <p>7.6. Arquitecturas SISD, MIMD, SIMD, SPMD y Vectoriales.</p> <p>7.7. Introducción a las Unidades de Procesamiento Gráfico (GPU).</p> <p>7.8. Introducción a las topologías de redes de multiprocesadores.</p> <p>7.9. Benchmarks para multiprocesadores.</p> <p>7.10. “Roofline”: un modelo simple de la medida de la performance de multiprocesadores.</p> <p>7.11. Errores de conceptos y falacias. Conclusiones.</p>		<p>El/la estudiante:</p> <ol style="list-style-type: none"> 1. Analiza la problemática de si el reemplazar procesadores grandes e ineficientes por otros más pequeños y eficientes unidades de proceso, favorece una mejor performance. 2. Comprende el desarrollo de softwares, considerando el uso de multiprocesadores y la complejidad de dicho desarrollo, dados los avances tecnológicos. 3. Extrae conclusiones acerca del uso de multiprocesadores y su performance, tomando <i>en cuenta</i> restricciones asociadas al consumo de energía. 	
Bibliografía de la unidad		<p>[1] Texto base: Patterson y Hennessy (2012), capítulo 7.</p> <p>Patterson y Hennessy (versión revisada (2018), capítulo 6.</p>	

E. Estrategias de enseñanza -aprendizaje:

El curso considera el uso de diversas estrategias:

- **Clases expositivas:** donde se presentan los principales conceptos asociados a la sesión de trabajo.
- **Trabajo de laboratorio computacional:** programan softwares y hardware de acuerdo a especificaciones y criterios dados. Se trabaja con tarjetas de desarrollo basadas en FPGA (circuito de alta integración).
- **Simulaciones:** uso de diferentes herramientas de simulación, por ejemplo, MARS 4.5 (simulador para MIPS) y RARS 1.1 (simulador para RISC-V), disponibles en la Web, entre otros.
- **Resolución de problemas:** el estudiantado desarrollará tareas de programación en lenguaje *assembly* de procesadores MIPS y RISC – V que irán de menor a mayor nivel de complejidad en la ejecución de cada actividad.

F. Estrategias de evaluación:

Al inicio del semestre, el cuerpo docente informará sobre los tipos de evaluación, cantidad, así como ponderaciones correspondientes para la calificación final y aprobación del curso.

Para esta propuesta, el curso considera las siguientes instancias de evaluación:

Tipo de evaluación	Resultado de aprendizaje asociado a la evaluación
<ul style="list-style-type: none"> • Tareas (tareas de programación en lenguaje <i>Assembly</i>, con su respectivo reporte). 	Evalúan RA1, RA2, RA4, RA5, RA6, RA7
<ul style="list-style-type: none"> • Controles (2). 	Evalúan RA1, RA2, RA3
<ul style="list-style-type: none"> • Examen. 	Evalúa RA1, RA2, RA3

G. Recursos bibliográficos:

Bibliografía Obligatoria:

[1] Patterson, D.A. and John L. Hennessy, J.L (2012). *Computer Organization and Design: The Hardware/Software Interface*". Morgan Kaufmann Publishers: Fourth Edition Revised.

**La bibliografía considera ediciones revisadas del mismo autor y texto, tomando en cuenta la referencia de diferentes tipos de procesadores: año 2014 (quinta edición; se hace una actualización del texto, pero se hace necesario mantener ambos textos); versión 2018 (es el mismo texto basado en el nuevo procesador RISC – V).*

Por último, la versión 2017 del texto de Patterson es importante, pues está basada en el procesador ARM.

H. Datos generales sobre elaboración y vigencia del programa de curso:

Vigencia desde:	Primavera, 2022
Elaborado por:	Francisco Rivera
Validado por:	Revisión académico par: Javier Ruiz del Solar Validación CTD ampliado de Eléctrica
Revisado por:	Área de Gestión Curricular