



## Universidad Nacional de Ingeniería (UNI)

Programa Profesional de  
Inteligencia Artificial  
Sílabo 2024-I

### 1. CURSO

CS221. Arquitectura de Computadores (Obligatorio)

### 2. INFORMACIÓN GENERAL

2.1 Curso	:	CS221. Arquitectura de Computadores
2.2 Semestre	:	3 <sup>er</sup> Semestre.
2.3 Créditos	:	3
2.4 horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS1D2. Estructuras Discretas II. (2 <sup>do</sup> Sem)

### 3. PROFESORES

Atención previa coordinación con el profesor

### 4. INTRODUCCIÓN AL CURSO

Es necesario que el profesional en Ciencia de la Computación tenga sólido conocimiento de la organización y funcionamiento de los diversos sistema de cómputo actuales en los cuales gira el entorno de programación. Con ello también sabrá establecer los alcances y límites de las aplicaciones que se desarrollen de acuerdo a la plataforma siendo usada.

Se tratarán los siguientes temas: componentes de lógica digital básicos en un sistema de computación, diseño de conjuntos de instrucciones, microarquitectura del procesador y ejecución en *pipelining*, organización de la memoria: caché y memoria virtual, protección y compartición, sistema I/O e interrupciones, arquitecturas super escalares y ejecución fuera de orden, computadoras vectoriales, arquitecturas para *multithreading*, multiprocesadores simétricos, modelo de memoria y sincronización, sistemas integrados y computadores en paralelo.

### 5. OBJETIVOS

- Este curso tiene como propósito ofrecer al estudiante una base sólida de la evolución de las arquitecturas de computadores y los factores que influenciaron en el diseño de los elementos de *hardware* y *software* en sistemas de computación actuales.
- Garantizar la comprensión de cómo es el *hardware* en sí y cómo interactúan *hardware* y *software* en un sistema de cómputo actual.
- Tratar los siguientes temas: componentes de lógica digital básicos en un sistema de computación, diseño de conjuntos de instrucciones, microarquitectura del procesador y ejecución en *pipelining*, organización de la memoria: caché y memoria virtual, protección y compartición, sistema I/O e interrupciones, arquitecturas super escalares y ejecución fuera de orden, computadoras vectoriales, arquitecturas para *multithreading*, multiprocesadores simétricos, modelo de memoria y sincronización, sistemas integrados y computadores en paralelo.

### 6. RESULTADOS DEL ESTUDIANTE

- 1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (**Usar**)
- 6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (**Evaluar**)

### 7. TEMAS

Unidad 1: Lógica digital y sistemas digitales (18 horas)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Revisión e historia de la Arquitectura de Computadores.</li> <li>• Lógica combinacional y secuencial/<i>field programmable gate arrays</i> como bloque fundamental de construcción lógico combinacional secuencial.</li> <li>• Modelos de representación(abstracción)</li> <li>• Herramientas de diseño asistidas por computadora que procesan hardware y representaciones arquitecturales.</li> <li>• Registrar transferencia notación / Hardware language descriptivo (Verilog/VHDL)</li> <li>• Restriccion física (Retrasos de Entrada, fan-in, fan-out, energía/potencia)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir el avance de la tecnología de dispositivos, desde los tubos de vacío hasta VLSI, desde las arquitecturas mainframe a las arquitecturas en escala warehouse [Familiarizarse]</li> <li>• Comprender que la tendencia de las arquitecturas modernas de computadores es hacia núcleos múltiples y que el paralelismo es inherente en todos los sistemas de hardware [Usar]</li> <li>• Explicar las implicancias de los límites de potencia para mejoras adicionales en el rendimiento de los procesadores y también en el aprovechamiento del paralelismo [Usar]</li> <li>• Relacionar las varias representaciones equivalentes de la funcionalidad de un computador, incluyendo expresiones y puertas lógicas, y ser capaces de utilizar expresiones matemáticas para describir las funciones de circuitos combinacionales y secuenciales sencillos [Familiarizarse]</li> <li>• Diseñar los componentes básicos de construcción de un computador: unidad aritmético lógica (a nivel de puertas lógicas), unidad central de procesamiento (a nivel de registros de transferencia), memoria (a nivel de registros de transferencia) [Usar]</li> <li>• Usar herramientas CAD para capturar, sistetizar, y simular bloques de construcción (como ALUs, registros, movimiento entre registros) de un computador simple [Familiarizarse]</li> <li>• Evaluar el comportamiento de un diagrama de tiempos y funcional de un procesador simple implementado a nivel de circuitos lógicos [Evaluar]</li> </ul>
<b>Lecturas :</b> [Harris12], [Sanjay05], [Patterson2004], [Ashenden07], [HP06], [Par05], [Stallings2010], [Pong06]	

Unidad 2: Representación de datos a nivel máquina (8 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Bits, Bytes y Words.</li> <li>• Representación de datos numérica y bases numéricas.</li> <li>• Sistemas de punto flotante y punto fijo.</li> <li>• Representaciones con signo y complemento a 2.</li> <li>• Representación de información no numérica (códigos de caracteres, información gráfica)</li> <li>• Representación de registros y arreglos.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar porqué en computación todo es datos, inclusive las instrucciones [Evaluar]</li> <li>• Explicar las razones de usar formatos alternativos para representar datos numéricos [Familiarizarse]</li> <li>• Describir cómo los enteros negativos se almacenan con representaciones de bit de signo y complemento a 2 [Usar]</li> <li>• Explicar cómo las representaciones de tamaño fijo afectan en la exactitud y la precisión [Usar]</li> <li>• Describir la representación interna de datos no numéricos como caracteres, cadenas, registros y arreglos [Usar]</li> <li>• Convertir datos numéricos de un formato a otro [Usar]</li> </ul>
<b>Lecturas :</b> [Harris12], [Sanjay05], [Patterson2004], [Ashenden07], [HP06], [Par05], [Stallings2010], [Pong06]	

Unidad 3: Organización de la Máquina a Nivel Ensamblador (8 horas)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Organización Básica de la Máquina de Von Neumann.</li> <li>• Unidad de Control.</li> <li>• <i>Instruction sets</i> y tipos (manipulación de información, control, I/O)</li> <li>• Assembler y Programación en Lenguaje de Máquina.</li> <li>• Formato de instrucciones.</li> <li>• Modos de direccionamiento.</li> <li>• Llamada a subrutinas y mecanismos de retorno.</li> <li>• I/O e Interrupciones.</li> <li>• Montículo (Heap) vs. Estático vs. Pila vs. Segmentos de código.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar la organización de la maquina clásica de von Neumann y sus principales unidades funcionales [Familiarizarse]</li> <li>• Describir cómo se ejecuta una instrucción en una máquina de von Neumann con extensión para hebras, sincronización multiproceso y ejecución SIMD (máquina vectorial) [Familiarizarse]</li> <li>• Describir el paralelismo a nivel de instrucciones y sus peligros, y cómo es esto tratado en pipelines de proceso típicos [Familiarizarse]</li> <li>• Resumir cómo se representan las instrucciones, tanto a nivel de máquina bajo el contexto de un ensamblador simbólico [Familiarizarse]</li> <li>• Demostrar cómo se mapean los patrones de lenguajes de alto nivel en notaciones en lenguaje ensamblador o en código máquina [Usar]</li> <li>• Explicar los diferentes formatos de instrucciones, así como el direccionamiento por instrucción, y comparar formatos de tamaño fijo y variable [Usar]</li> <li>• Explicar como las llamadas a subrutinas son manejadas a nivel de ensamblador [Usar]</li> <li>• Explicar los conceptos básicos de interrupciones y operaciones de entrada y salida (I/O) [Familiarizarse]</li> <li>• Escribir segmentos de programa simples en lenguaje ensamblador [Usar]</li> <li>• Ilustrar cómo los bloques constructores fundamentales en lenguajes de alto nivel son implementados a nivel de lenguaje máquina [Usar]</li> </ul>
<b>Lecturas :</b> [Harris12], [Sanjay05], [Patterson2004], [Ashenden07], [HP06], [Par05], [Stallings2010], [Pong06]	

Unidad 4: Organización funcional (8 horas)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Implementación de <i>datapath</i>, incluyendo un <i>pipeline</i> de instrucciones, detección de <i>hazards</i> y la resolución.</li> <li>• Control de unidades: Microprogramada.</li> <li>• Instrucción (Pipelining)</li> <li>• Introducción al paralelismo al nivel de instrucción (PNI)</li> </ul>	<ul style="list-style-type: none"> <li>• Comparar implementaciones alternativas de ruta de datos [Evaluar]</li> <li>• Discutir el concepto de puntos de control y la generación de señales de control usando implementaciones a nivel de circuito o microprogramadas [Familiarizarse]</li> <li>• Explicar el paralelismo a nivel de instrucciones básicas usando pipelining y los mayores riesgos que pueden ocurrir [Usar]</li> <li>• Diseñar e implementar un procesador completo, incluyendo ruta de datos y control [Usar]</li> <li>• Calcular la cantidad promedio de ciclos por instrucción de una implementación con procesador y sistema de memoria determinados [Evaluar]</li> </ul>
Lecturas : [Harris12], [Sanjay05], [Patterson2004], [Ashenden07], [HP06], [Par05], [Stallings2010], [Pong06]	

Unidad 5: Mejoras de rendimiento (8 horas)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Arquitectura superescalar.</li> <li>• Predicción de ramificación, Ejecución especulativa, Ejecución fuera de orden.</li> <li>• Prefetching.</li> <li>• Procesadores vectoriales y GPU's</li> <li>• Soporte de hardware para multiprocesamiento.</li> <li>• Escalabilidad.</li> <li>• Arquitecturas alternativas, como VLIW / EPIC y aceleradores y otros tipos de procesadores de propósito especial.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las arquitecturas superescalares y sus ventajas [Familiarizarse]</li> <li>• Explicar el concepto de predicción de bifurcaciones y su utilidad [Usar]</li> <li>• Caracterizar los costos y beneficios de la precarga prefetching [Evaluar]</li> <li>• Explicar la ejecución especulativa e identifique las condiciones que la justifican [Evaluar]</li> <li>• Discutir las ventajas de rendimiento ofrecida en una arquitectura de multihebras junto con los factores que hacen difícil dar el máximo beneficio de estas [Evaluar]</li> <li>• Describir la importancia de la escalabilidad en el rendimiento [Evaluar]</li> </ul>
Lecturas : [Par05], [Par02], [PH14], [Don06], [Joh91]	

Unidad 6: Organización y Arquitectura del Sistema de Memoria (8 horas)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Sistemas de Almacenamiento y su Tecnología.</li> <li>• Jerarquía de Memoria: importancia de la localización temporal y espacial.</li> <li>• Organización y Operaciones de la Memoria Principal.</li> <li>• Latencia, ciclos de tiempo, ancho de banda e <i>inter-leading</i>.</li> <li>• Memorias caché (Mapeo de direcciones, Tamaño de bloques, Reemplazo y Políticas de almacenamiento)</li> <li>• Multiprocesador coherencia cache / Usando el sistema de memoria para las operaciones de sincronización de memoria / atómica inter-core.</li> <li>• Memoria virtual (tabla de página, TLB)</li> <li>• Manejo de Errores y confiabilidad.</li> </ul>	<ul style="list-style-type: none"> <li>• Identificar las principales tecnologías de memoria (Por ejemplo: SRAM, DRAM, Flash, Disco Magnético) y su relación costo beneficio [Familiarizarse]</li> <li>• Explique el efecto de latencia de memoria en tiempo de ejecución [Familiarizarse]</li> <li>• Describir como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el atraso efectivo en la memoria [Usar]</li> <li>• Describir los principios de la administración de memoria [Usar]</li> <li>• Explique el funcionamiento de un sistema con gestión de memoria virtual [Usar]</li> <li>• Calcule el tiempo de acceso promedio a memoria bajo varias configuraciones de caché y memoria y para diversas combinaciones de instrucciones y referencias a datos [Evaluar]</li> </ul>
Lecturas : [Harris12], [Sanjay05], [Patterson2004], [Ashenden07], [HP06], [Par05], [Stallings2010], [Pong06]	

Unidad 7: Interfaz y comunicación (8 horas)	
Resultados esperados: 1,6	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• Fundamentos de I/O: Handshaking, Bbuffering, I/O programadas, interrupciones dirigidas de I/O.</li> <li>• Interrumpir estructuras: interrumpir reconocimiento, vectorizado y priorizado.</li> <li>• Almacenamiento externo, organización física y discos.</li> <li>• Buses: Protocolos de bus, arbitraje, acceso directo a memoria (DMA).</li> <li>• Introducción a Redes: comunicación de redes como otra capa de acceso remoto.</li> <li>• Soporte Multimedia.</li> <li>• Arquitecturas RAID.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar como las interrupciones son aplicadas para implementar control de entrada-salida y transferencia de datos [Familiarizarse]</li> <li>• Identificar diversos tipos de buses en un sistema computacional [Familiarizarse]</li> <li>• Describir el acceso a datos desde una unidad de disco magnético [Usar]</li> <li>• Comparar organizaciones de red conocidas como organizaciones en bus/Ethernet, en anillo y organizaciones conmutadas versus ruteadas [Evaluar]</li> <li>• Identificar las interfaces entre capas necesarios para el acceso y presentación multimedia, desde la captura de la imagen en almacenamiento remoto, a través del transporte por una red de comunicaciones, hasta la puesta en la memoria local y la presentación final en una pantalla gráfica [Familiarizarse]</li> <li>• Describir las ventajas y limitaciones de las arquitecturas RAID [Familiarizarse]</li> </ul>
Lecturas : [Harris12], [Sanjay05], [Patterson2004], [Ashenden07], [HP06], [Par05], [Stallings2010], [Pong06]	

Unidad 8: Multiprocesamiento y arquitecturas alternativas (8 horas)	
Resultados esperados: 6	
Temas	Objetivos de Aprendizaje
<ul style="list-style-type: none"> <li>• <i>Power Law</i>.</li> <li>• Ejemplos de <i>sets</i> de instrucciones y arquitecturas SIMD y MIMD.</li> <li>• Redes de interconexión (Hypercube, Shuffle-exchange, Mesh, Crossbar)</li> <li>• Sistemas de memoria de multiprocesador compartido y consistencia de memoria.</li> <li>• Coherencia de cache multiprocesador.</li> </ul>	<ul style="list-style-type: none"> <li>• Discutir el concepto de procesamiento paralelo mas allá del clásico modelo de von Neumann [Evaluar]</li> <li>• Describir diferentes arquitecturas paralelas como SIMD y MIMD [Familiarizarse]</li> <li>• Explicar el concepto de redes de interconexión y mostrar diferentes enfoques [Usar]</li> <li>• Discutir los principales cuidados en los sistemas de multiprocesamiento presentes con respecto a la gestión de memoria y describir como son tratados [Familiarizarse]</li> <li>• Describir las diferencias entre conectores eléctricos en paralelo backplane, interconexión memoria procesador y memoria remota via red, sus implicaciones para la latencia de acceso y el impacto en el rendimiento de un programa [Evaluar]</li> </ul>
Lecturas : [Harris12], [Sanjay05], [Patterson2004], [Ashenden07], [HP06], [Par05], [Stallings2010], [Pong06]	

## 8. PLAN DE TRABAJO

### 8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

### 8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

### 8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

## 9. SISTEMA DE EVALUACIÓN

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Don06] J. Dongarra. "Trends in high performance computing: a historical overview and examination of future developments". In: *Circuits and Devices Magazine, IEEE* 22.1 (2006), pp. 22–27. ISSN: 8755-3996. DOI: 10.1109/MCD.2006.1598076
- [HP06] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. 4th. San Mateo, CA: Morgan Kaufman, 2006.
- [Joh91] M. Johnson. *Superscalar microprocessor design*. Prentice Hall series in innovative technology. Prentice Hall, 1991. ISBN: 9780138756345.
- [Par02] Behrooz Parhami. *Introduction to parallel processing: algorithms and architectures*. Plenum series in computer science. Plenum Press, 2002. ISBN: 9780306459702.
- [Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers*. New York: Oxford Univ. Press, 2005. ISBN: ISBN 0-19-515455-X.
- [PH14] David A. Patterson and John L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 5th ed. San Mateo, CA: Morgan Kaufman, 2014.