



**Universidad de Ingeniería y Tecnología**  
**Escuela Profesional de**  
**Ciencia de la Computación**  
**Silabo del curso – Periodo Académico 2017-II**

1. **Código del curso y nombre:** CS221. Arquitectura de Computadores
2. **Créditos:** 3
3. **Horas de Teoría y Laboratorio:** 2 HT; 2 HP;
4. **Docente(s)**

Atención previa coordinación con el profesor

**5. Bibliografía**

- [Den05] Peter J. Denning. "The locality principle". In: *Commun. ACM* 48.7 (July 2005), pp. 19–24. ISSN: 0001-0782. DOI: 10.1145/1070838.1070856. URL: <http://doi.acm.org/10.1145/1070838.1070856>.
- [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.
- [EA05] Hesham El-Rewini and Mostafa Abd-El-Barr. *Advanced Computer Architecture and Parallel Processing*. Hoboken, NJ: John Wiley & Sons, 2005. ISBN: 0-471-46740-5.
- [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.
- [PH04] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 3rd ed. San Mateo, CA: Morgan Kaufman, 2004.
- [Sta10] William Stalings. *Computer Organization and Architecture: Designing for Performance*. 8th. Upper Saddle River, NJ: Prentice Hall, 2010.

**6. Información del curso**

- (a) **Breve descripción del 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 se instala 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.

- (b) **Prerrequisitos:** CS1D2. Estructuras Discretas II. (2<sup>do</sup> Sem)

- (c) **Tipo de Curso:** Obligatorio

**7. Competencias**

- 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.

## 8. Contribución a los resultados (*Outcomes*)

- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (**Evaluar**)

## 9. Competencias (IEEE)

- C4.** Una comprensión del hardware de la computadora desde la perspectiva del software, por ejemplo, el uso del procesador, memoria, unidades de disco, pantalla, etc.⇒ **Outcome i**
- C8.** Entendimiento de lo que las tecnologías actuales pueden y no pueden lograr.⇒ **Outcome b,i**
- C9.** Comprensión de las limitaciones de la computación, incluyendo la diferencia entre lo que la computación es inherentemente incapaz de hacer frente a lo que puede lograrse a través de un futuro de ciencia y tecnología.⇒ **Outcome b**
- CS3.** Analizar el grado en que un sistema basado en el ordenador cumple con los criterios definidos para su uso actual y futuro desarrollo.⇒ **Outcome i**

## 10. Lista de temas a estudiar en el curso

1. Lógica digital y sistemas digitales
2. Representación de datos a nivel máquina
3. Organización de la Máquina a Nivel Ensamblador
4. Organización funcional
5. Organización y Arquitectura del Sistema de Memoria
6. Interfaz y comunicación
7. Multiprocesamiento y arquitecturas alternativas
8. Mejoras de rendimiento

## 11. Metodología y Evaluación

### Metodología:

#### Sesiones Teóricas:

El desarrollo de las sesiones teóricas está focalizado en el estudiante, a través de su participación activa, resolviendo problemas relacionados al curso con los aportes individuales y discutiendo casos reales de la industria. Los alumnos desarrollarán a lo largo del curso un proyecto de aplicación de las herramientas recibidas en una empresa.

#### Sesiones de Laboratorio:

Las sesiones prácticas se desarrollan en laboratorio. Las prácticas de laboratorio se realizan en equipos para fortalecer su comunicación. Al inicio de cada laboratorio se explica el desarrollo de la práctica y al término se destaca las principales conclusiones de la actividad en forma grupal.

**Exposiciones individuales o grupales:**

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.

**Lecturas:**

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

**Sistema de Evaluación:****12. Contenido**

Unidad 1: Lógica digital y sistemas digitales (18)	
Competences esperadas: C8	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> <li>• Describir el avance paulatino de los componentes de la tecnología de computación, 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, sistematizar, 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>	<ul style="list-style-type: none"> <li>• Revisión e historia de la Arquitectura de Computadores.</li> <li>• Lógica combinacional vs. secuencial/Arreglos de puertas de campo programables como bloque fundamental de construcción lógico combinacional-secuencial.</li> <li>• Múltiples representaciones / Capas de interpretación (El hardware es solo otra capa)</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>• Restricción física (Retrasos de Entrada, fan-in, fan-out, energía/poder)</li> </ul>
<b>Lecturas :</b> [Par05], [PH04]	

Unidad 2: Representación de datos a nivel máquina (8)	
Competences esperadas: C9	
Objetivos de Aprendizaje	Tópicos
<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>	<ul style="list-style-type: none"> <li>• Bits, Bytes y Words.</li> <li>• Representacion 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>
<b>Lecturas :</b> [Par05], [Sta10]	

Unidad 3: Organización de la Máquina a Nivel Ensamblador (8)	
Competences esperadas: C4,CS3	
Objetivos de Aprendizaje	Tópicos
<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>	<ul style="list-style-type: none"> <li>• Organización Básica de la Máquina de Von Neumann.</li> <li>• Unidad de Control.</li> <li>• Paquetes de instrucciones y tipos (manipulación de información, control, I/O)</li> <li>• Assembler / 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>
Lecturas : [Par05], [PH04], [HP06]	

Unidad 4: Organización funcional (8)	
Competences esperadas: C9	
Objetivos de Aprendizaje	Tópicos
<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>	<ul style="list-style-type: none"> <li>• Implementación de rutas de datos simples, incluyendo la canalización de instrucciones, detección de riesgos y la resolución.</li> <li>• Control de unidades: Realización Cableada vs Realización Microprogramada.</li> <li>• Instrucción (Pipelining)</li> <li>• Introducción al paralelismo al nivel de instrucción (PNI)</li> </ul>
<b>Lecturas :</b> [Par05], [HP06]	

Unidad 5: Organización y Arquitectura del Sistema de Memoria (8)	
Competences esperadas: CS3	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> <li>• Identifique las principales tecnologías de memoria (Por ejemplo: SRAM, DRAM, Flash, Disco Magnético) y su relación costo beneficio [Familiarizarse]</li> <li>• Explique el efecto del retardo de la memoria en tiempo de ejecución [Familiarizarse]</li> <li>• Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Usar]</li> <li>• Describa 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>	<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 intercalación.</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> <li>• Error de codificación, compresión de datos y la integridad de datos.</li> </ul>
<b>Lecturas :</b> [Par05], [PH04], [Den05]	

<b>Unidad 6: Interfaz y comunicación (8)</b>	
<b>Competences esperadas: C4,C9,CS3</b>	
<b>Objetivos de Aprendizaje</b>	<b>Tópicos</b>
<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>	<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>
<b>Lecturas :</b> [Par05], [Sta10]	

<b>Unidad 7: Multiprocesamiento y arquitecturas alternativas (8)</b>	
<b>Competences esperadas: C9</b>	
<b>Objetivos de Aprendizaje</b>	<b>Tópicos</b>
<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 electricos 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>	<ul style="list-style-type: none"> <li>• Ley potencial.</li> <li>• Ejemplos de juego 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>
<b>Lecturas :</b> [Par05], [Par02], [EA05]	

Unidad 8: Mejoras de rendimiento (8)	
Competences esperadas: C8,C9	
Objetivos de Aprendizaje	Tópicos
<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>	<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>
<b>Lecturas :</b> [Par05], [Par02], [PH04], [Don06], [Joh91]	





**Universidad de Ingeniería y Tecnología**  
**Escuela Profesional de**  
**Ciencia de la Computación**  
**Silabo del curso – Periodo Académico 2017-II**

1. **Código del curso y nombre:** CS113. Programación Orientada a Objetos II
2. **Créditos:** 4
3. **Horas de Teoría y Laboratorio:** 3 HT; 2 HP;
4. **Docente(s)**

Atención previa coordinación con el profesor

**5. Bibliografía**

- [Nak13] S. Nakariakov. *The Boost C++ Libraries: Generic Programming*. CreateSpace Independent Publishing Platform, 2013.
- [Str13] B Stroustrup. *The C++ Programming Language, 4th edition*. Addison-Wesley, 2013.

**6. Información del curso**

- (a) **Breve descripción del curso** Este es el tercer curso en la secuencia de los cursos introductorios a la informática. En este curso se pretende cubrir los conceptos señalados por la Computing Curricula IEEE(c)-ACM 2001, bajo el enfoque funcional-first. El paradigma orientado a objetos nos permite combatir la complejidad haciendo modelos a partir de abstracciones de los elementos del problema y utilizando técnicas como encapsulamiento, modularidad, polimorfismo y herencia. El dominio de estos temas permitirá que los participantes puedan dar soluciones computacionales a problemas de diseño sencillos del mundo real.
- (b) **Prerrequisitos:** CS112. Programación Orientada a Objetos I. (2<sup>do</sup> Sem)
- (c) **Tipo de Curso:** Obligatorio

**7. Competencias**

- Introducir al alumno a los fundamentos del paradigma de orientación a objetos, permitiendo asimilar los conceptos necesarios para desarrollar un sistema de información

**8. Contribución a los resultados (Outcomes)**

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Usar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Usar**)
- d) Trabajar efectivamente en equipos para cumplir con un objetivo común. (**Usar**)

**9. Competencias (IEEE)**

- C1.** La comprensión intelectual y la capacidad de aplicar las bases matemáticas y la teoría de la informática (*Computer Science*).⇒ **Outcome a**
- C3.** Una comprensión intelectual de, y el aprecio por el papel central de los algoritmos y estructuras de datos.⇒ **Outcome a**
- C18.** Capacidad para participar de forma activa y coordinada en un equipo.⇒ **Outcome d**
- CS1.** Modelar y diseñar sistemas de computadora de una manera que se demuestre comprensión del balance entre las opciones de diseño.⇒ **Outcome a**