

Programa de Asignatura

1. Identificación Asignatura

Nombre:	Arquitectura de Computadores		Código:	IF1014
Carrera:	Ingeniería Civil Informática	Unidad Académica:	Ciencias Naturales y Tecnología	
Ciclo Formativo:	Ciclo Licenciatura	Línea formativa:	Especializada	
Semestre	VI	Tipo de actividad:	Obligatoria	
N° SCT:	3	Horas Cronológicas Semanales		
		Presenciales:	3	Trabajo Autónomo:
Pre-requisitos	Teoría de la Computación			

2. Propósito formativo

El curso "Arquitectura de Computadores" tiene como objetivo principal que los estudiantes desarrollen una comprensión integral de los conceptos fundamentales que subyacen en la evolución y el diseño de las arquitecturas de sistemas computacionales modernos. A través de este curso, los estudiantes explorarán tanto los principios organizativos como las relaciones funcionales entre los diferentes componentes de hardware y software que conforman un sistema computacional, comprender estos principios es fundamental para el curso de "Sistemas Operativos", que se imparte en el séptimo semestre, asegurando que los estudiantes estén bien equipados para analizar, diseñar, y optimizar sistemas computacionales complejos en su futuro profesional.

En la primera parte del curso, se abordan los elementos esenciales de la estructura y funcionamiento de los computadores, comenzando con la descripción de la estructura básica, continuando con el concepto de maquina multinivel de manera de comprender la "distancia conceptual" que existe entre el lenguaje entendido por la maquina y el lenguaje de programación usado por los programadores. Luego se entrara en el desarrollo de los principios básicos en los cuales se funda el diseño, correspondientes a las ideas que se han seguido desde los primeros computadores. Se adentrara en la relación existente entre la arquitectura y el juego de instrucciones de la CPU (RISC y SISC). Por ultimo en esta primera parte se observara la organización interna de la CPU y su relación con el rendimiento (Ley de Amdahl).

La segunda parte del curso se enfoca en la CPU, basándose en la arquitectura MIPS64, comenzando con una descripción de esta arquitectura y su juego de instrucciones para posteriormente plantear posibles implementaciones, microarquitecturas monociclo y segmentada, para continuar con paralelismo y microarquitecturas superescalar, en este punto y desde la perspectiva del rendimiento se analiza la ley de Moore y sus implicaciones. Para terminar con esta parte del curso se analizaran los CPU multihilo (taxonomía de Flynn, paralelismo y procesadores multinucleo).

La tercera parte del curso se enfoca en la memoria, mostrando como el sistema de memoria del computador se construye empleando diferentes tecnologías de almacenamiento, tratando de obtener un sistema de memoria rápido, barato y de gran capacidad. En esta parte se desarrollaran conceptos como SRAM, DRAM, almacenamiento secundario como tecnologías y memoria cache y memoria virtual como estrategias

Para completar el curso con un bloque conocido como sistema de entrada/salida dado que corresponde a una parte fundamental de la arquitectura de cualquier computador, tomando tópicos como interfaces, sistemas de interconexión y periféricos.

3. Contribución al perfil de egreso

Esta asignatura contribuye a los siguientes desempeños o resultados de aprendizaje globales declarados en el Perfil de Egreso de la carrera:

1. Evalúa la implementación de soluciones computacionales, utilizando métodos analíticos y experimentales, para estudiar su eficiencia y rendimiento.

4. Resultados de aprendizaje específicos

Resultado de Aprendizaje Específico	Criterios de evaluación	Evidencia
RA1. Comprende y aplica los principios organizativos de los sistemas computacionales, identificando y explicando las relaciones y funciones de sus distintos componentes de hardware y software.	<ol style="list-style-type: none"> 1. Evalúa y describe las relaciones funcionales básicas entre los componentes principales de los sistemas computacionales (CPU, memoria, dispositivos de E/S), identificando cómo estos interactúan para ejecutar operaciones computacionales de manera eficiente. 1.2. Demuestra comprensión del concepto de rendimiento y las relaciones que tiene este con el juego de instrucciones implementado 1.3. Analiza y compara rendimiento de cpu usando métricas y Benchmarks. 	Laboratorios, guías de ejercicio.
RA2. Comprende conceptualmente las características fundamentales en las que se basa el diseño de las CPU actuales	<ol style="list-style-type: none"> 2.1. Demuestra comprensión avanzada de los principios de operación en las que se basa el diseño de un CPU 2.2. Utiliza software para la programación de bajo nivel de sistemas computacionales, utilizando la arquitectura MIPS64 	Laboratorios, guías de ejercicio, https://web.edumips.org/
RA3. Evalúa el rendimiento del diseño de un sistema computacional. Integrando el conocimiento de los componentes y sus funciones.	<ol style="list-style-type: none"> 3.1 Analiza y Aplica de manera crítica el problema de evaluar el rendimiento de un computador, al tiempo que cómo las interacciones impactan en el rendimiento y la eficiencia del sistema. Además, propone alternativas basadas en un entendimiento de las dependencias y colaboraciones entre software y hardware en contextos específicos. 	Laboratorios, guías de ejercicio.

1. Unidades de Aprendizaje

1. Evolución del Procesamiento de Datos y Fundamentos de Arquitectura

- 1.1. Introducción y clasificación de las computadoras
- 1.2. Estructura básica
- 1.3. Maquina multinivel
- 1.4. Principios básicos de diseño
- 1.5. Arquitectura del juego de instrucciones
- 1.6. microarquitectura
- 1.7. Rendimiento

2. La CPU

- 2.1. CPU
- 2.2. Arquitectura MIPS64
- 2.3. Micro arquitectura monociclo
- 2.4. Micro arquitectura segmentada
- 2.5. Emisión múltiple de instrucciones
- 2.6. paralelismo
- 2.7. micro arquitectura superescalar
- 2.8. Ley de Moore
- 2.9. CPU multihilo

3. Jerarquía de Memoria

- 3.1. Memoria Cache
 - Estrategias de correspondencia, reemplazo y escritura
 - Organización
 - Coherencia
- 3.2. Memoria Virtual
 - Memoria virtual paginada
 - el TLB
 - Soporte

4. El sistema E/S

- 4.1. Interfaces
 - Ubicación de espacios de direcciones
 - protección
 - Técnicas de E/S
- 4.2. Sistema de interconexión
 - Topologías
 - Características
- 4.3. Periféricos
- 4.4. Visualización de E/S

1. Recursos de Aprendizaje

Bibliografía:

[1] Julio Ortega, Mancia Anguita, Alberto Prieto. Arquitectura de computadores. Paraninfo, 2005. ISBN: 849-7322746.

[2] W. Stallings, A.C. Vargas. Organización y arquitectura de computadores: diseño para optimizar prestaciones. Prentice Hall, 2001. ISBN: 978-8420529936.

Bibliografía sugerida:

[1] D.A. Patterson, J.L. Hennessy. Computer organization and design. The hardware/-software interface, 5th edition. Morgan Kaufmann, 2014. ISBN: 978-0124077263.

[2] J.L. Hennessy, D.A. Patterson. Computer architecture. A quantitative approach, 5th edition. Morgan Kaufmann, 2012. ISBN: 978-0123838728.

[3] G.E. Moore. Moore's law at 40. Understanding Moore's law: four decades of innovation. Chemical Heritage Foundation, 2006. ISBN: 978-0941901413.

[4] R.R. Schaller. Moore's law: past, present and future. IEEE Spectrum, vol. 34(6), pp. 52-59, 1997.

[7] V.C. Hamacher, Z.G. Vranesic, S.G. Zaky, M.L.F. García, G.Q. Vieyra. Organización de computadoras. McGraw-Hill, 1987. ISBN: 968-4220588.

Simulación:

[1] <https://web.edumips.org/>

Editor Argentino.

Recursos materiales e infraestructura:

- Laboratorio de computación.
- Acceso a Ucampus.

Computadores debidamente equipados para utilizar lenguajes de alto nivel (por ej.: Python, Verilog, VHDL).

7. Comportamiento y ética académica:

Se espera que, en el marco de sus actividades académicas y estudiantiles, los estudiantes demuestren un compromiso inquebrantable con los estándares de ética y honestidad académica que distinguen a nuestra comunidad universitaria. Este compromiso debe reflejarse en todas las acciones y trabajos realizados dentro del ámbito de la Universidad de Aysén, siguiendo rigurosamente las directrices establecidas en el Reglamento de Estudiantes, particularmente lo indicado en los artículos 23°, 24°, y 26°.

Además, es imperativo que todas las acciones se alineen con los reglamentos institucionales aplicables y las normativas internas vigentes, incluido el código de ética de la universidad. Estos documentos no solo rigen nuestros comportamientos cotidianos sino que también establecen las expectativas para nuestra convivencia y trabajo académico.

Cualquier infracción a estos principios de honestidad académica, detectada durante la realización, presentación, o entrega de cualquier actividad evaluativa del curso, acarreará consecuencias serias. Estas incluyen la suspensión inmediata de la actividad en cuestión y la asignación de la nota mínima posible (1.0) como reflejo de la gravedad del acto. Esta política se aplica de manera estricta y sin excepciones, reafirmando nuestro compromiso con una formación académica íntegra y responsable.

8. Responsables:

Académico (s) Responsable (s) y equipo docente	Profesor: Claudio Alejandro Herranz Ibarro		
Contacto	Correo: claudio.herranz@docentes.uaysen.cl		
Año	2025	Periodo Académico	Segundo semestre
Horario clases	Cátedra: Lu 8:30 – 10:00 hrs. Sesión Prácticas: Mi 12:00 – 13:30 hrs.	Horario de atención estudiantes	Contactar previamente al profesor mediante email.
Sala / Campus	Sala Virtual Ucampus		

9. Metodología de Trabajo:

En el curso se contemplan cuatro tipos de actividades docentes, las cuales se asocian a requerimientos de sala y al nivel de intervención del profesor/ayudante:

Actividad docente	Descripción	Intervención del profesor/ayudante	Requerimiento de sala
Exposición conceptual	El profesor introduce los fundamentos preliminares de Arquitectura de Computadores y conceptos esenciales para el desarrollo de habilidades prácticas, con una participación activa en el aula tradicional o entornos virtuales.	Alta	Sala de clases UCampus Online UCampus Offline
Programación expositiva	Mediante la resolución de problemas específicos de Arquitectura de Computadores, el profesor guía a los estudiantes a profundizar en el entendimiento de conceptos clave, empleando ejemplos prácticos en un entorno de sala de clases o plataformas online.	Alta	Sala de clases UCampus Online UCampus Offline
Programación tutorial	Esta modalidad incorpora pausas estratégicas en la exposición, permitiendo a los estudiantes completar tareas específicas bajo la supervisión del docente, favoreciendo el aprendizaje interactivo en laboratorios de computación o con uso de equipos personales	Media	Laboratorio de computación Computador personal
Actividad práctica / Programación autónoma	Fomenta la independencia de los estudiantes al enfrentar y resolver desafíos de Arquitectura de Computadores, en grupos o individualmente, con mínima intervención docente, ideal para el trabajo en laboratorios de computación o mediante dispositivos personales.	Baja	Laboratorio de computación Computador personal

Durante el semestre, **se programarán evaluaciones menores** en cualquier momento, con el fin de revisar y reforzar los conocimientos adquiridos hasta la fecha. Estas evaluaciones podrán adoptar diversas formas:

Control : Evaluación, ya sea individual o en grupo, realizada en papel durante los primeros 45 minutos de la clase.

Trabajo en Clase: Actividad evaluativa individual o grupal, llevada a cabo en computador. Podrá ocupar los primeros 45 minutos o extenderse durante toda la sesión.

Entrega de Tareas: Evaluación grupal realizada en computador y fuera del horario de clase, promoviendo el trabajo colaborativo y la gestión del tiempo.

Adicionalmente, como componente de la calificación final, los estudiantes deberán presentar, un proyecto que abarque el análisis, diseño e implementación de una solución tecnológica. Para esta tarea, se proporcionará un instructivo detallado que guiará a los estudiantes en su preparación y entrega, el desarrollo del proyecto se gestará durante todo el semestre, evidenciando una metodología clara de forma progresiva.

7. Evaluaciones:

Evaluación	Ponderaciones específicas	Ponderación nota presentación
Pruebas de cátedra	<ul style="list-style-type: none"> Prueba 1 (P1): 40% Prueba 2 (P2): 40% Controles (Q1): 20% 	80%
Evaluaciones menores	El promedio simple entre las notas consideradas corresponde al 100%.	20%

Calificación final:

- Nota de presentación: 70%
- Examen Final: 30 % (Exposición de proyecto de aplicación)

Requisito de Aprobación

- Asistencia: 65%
- Nota Final: 4,0
- La entrega de cualquiera de las evaluaciones indicadas en el programa de la asignatura por fuera del plazo definido serán evaluadas con nota mínima (1,0).
- Sólo para el caso de las evaluaciones presenciales, se permitirá comenzar con un atraso máximo de 15 minutos después de comenzada la misma. El tiempo de retraso no se podrá recuperar. Aquel/la estudiante que se presente a rendir la evaluación después de pasados los 15 minutos, se evaluará con nota mínima (1,0).
- En caso de inasistencia o no entrega en plazo de alguna evaluación, se podrán justificar mediante el mismo procedimiento antes descrito para la justificación de inasistencias actividades lectivas.
- Las clases teóricas tienen un 65% de asistencia mínima obligatoria, siguiendo el mínimo requerido por el Reglamento General de Estudios de Pregrado . La asistencia a las clases prácticas (laboratorios y terrenos) es del 100%. El no cumplimiento de estos porcentajes de asistencia será causal de reprobación de la asignatura.
- En casos debidamente justificados ante el Registro Académico, el/ la estudiante que no haya asistido a una salida a terreno o laboratorio tendrá derecho a rendir examen. Se considerarán debidamente justificadas las inasistencias ante el Registro Académico aquéllas que estén respaldadas con certificados médicos, laborales o algún documento validado por la Unidad de Acceso y Desarrollo Estudiantil. Las actividades de terrenos y laboratorios no podrán ser recuperadas.
- Se permitirá el ingreso posterior a la hora de inicio con un máximo de 15 minutos, siempre y cuando no sea una acción repetida por la/el estudiante (se aceptará máximo de 3 veces). Para las salidas a terreno, el tiempo de espera máximo será de 5 minutos. Para casos donde las actividades lectivas contemplen más de un módulo, el/la estudiante que no haya asistido al módulo anterior, podrá ingresar al comienzo del nuevo módulo.
- En caso de que ningún estudiante se presente a la actividad lectiva después de 15 minutos de comenzada, ésta se suspenderá. Los contenidos programados para dicha actividad se darán por dictados, será responsabilidad del estudiante ponerse al día con los contenidos de dicha clase. Los contenidos de dicha clase, y ejercicios, si así lo hubiera, serán enviados para ser realizados como trabajo autónomo.

Otros aspectos asociados al funcionamiento del curso:

- Toda la coordinación del curso (comunicaciones, actualización de notas, material, etc.) será realizada a través de UCampus.
- **Es deber del estudiante mantenerse informado de las noticias, avisos y material entregado por el profesor a través de estos medios, se sugiere instalar en su dispositivo móvil la aplicación de UCampus.**
- Recordar que los correos electrónicos serán respondidos en horario laboral (lunes a viernes de 9:00 a 18:00hs), no se responderán correos fuera de ese horario.
- Durante el desarrollo de las actividades lectivas, los teléfonos celulares deberán estar en silencio y guardados, a menos que el/la profesor/a específicamente requiera de estos equipos para la realización de su clase o durante algunos casos excepcionales conversados previamente con el/la docente a cargo.
- Las actividades lectivas se dictarán de forma presencial, salvo excepciones sujeto a contingencias presentes durante el transcurso de la asignatura.

7. Planificación de las actividades de enseñanza- aprendizaje y de evaluación:

Semana	Resultado (s) de Aprendizaje (*)	Tema (Unidades de aprendizaje) y actividades	Recursos utilizados o lecturas (*)	Actividad(es)
1	RA1	<ul style="list-style-type: none"> • Estructura del computador (Arquitectura von Neumann vs. Harvard, componentes: CPU, memoria, E/S). • Máquina multi-nivel (Niveles de abstracción, traducción vs. interpretación). • Principios de diseño (Optimización del caso común, paralelismo, jerarquía de memoria). 	B1, B2, B3	
2	RA1	Arquitectura del juego de instrucciones (ISA) <ul style="list-style-type: none"> • Modelos de máquina (Pila, acumulador, registro-memoria, carga/almacenamiento). • Clasificación ISA (RISC vs. CISC), ejemplos (MIPS64, ARM, x86). • Modos de direccionamiento (Inmediato, registro, indexado, relativo al PC). 	B1, B2, B3	
3	RA1	Rendimiento y métricas <ul style="list-style-type: none"> • Conceptos de rendimiento (Tiempo de respuesta, productividad, aceleración). • Ley de Amdahl (Impacto de mejoras parciales). • Rendimiento de la CPU (CPI, frecuencia, ley de hierro). 	B1, B2, B3	Prueba (P1) ()
4	RA1, RA2	Microarquitectura monociclo <ul style="list-style-type: none"> • Unidades funcionales (ALU, registro PC, fichero de registros). • Camino de datos monociclo (Etapas: IF, ID, EX, MEM, WB). • Limitaciones y deficiencias (Ciclo de reloj largo, falta de optimización). 	B1, B2, B3	

5	RA2	<p>Microarquitectura segmentada (I)</p> <ul style="list-style-type: none"> • Concepto de pipeline (Analogía con procesos industriales, etapas balanceadas). • Camino de datos segmentado (Registros de segmentación, flujo de instrucciones). • Cronogramas de ejecución (Ganancia teórica vs. práctica). 	B1, B2, B3	
6	RA2	<p>Microarquitectura segmentada (II)</p> <ul style="list-style-type: none"> • Riesgos estructurales (Conflictos de recursos). • Riesgos por dependencias de datos (RAW, WAR, WAW). • Riesgos de control (Saltos condicionales/incondicionales, detenciones). 	B1, B2, B3	Prueba 1 Introducción, ISA, rendimiento, microarquitectura monociclo/segme ntada, riesgos del pipeline.
7	RA2	<p>Optimización del pipeline</p> <ul style="list-style-type: none"> • Reducción de detenciones por datos (Rutas de reenvío/forwarding). • Planificación de instrucciones (Reordenamiento por hardware/compilador). • Renombrado de registros (Eliminación de dependencias WAR/WAW). 	B1, B2, B3	
8	RA2	<p>Operaciones multiciclo y excepciones</p> <ul style="list-style-type: none"> • Unidades de ejecución multiciclo (Multiplicación, punto flotante). • Gestión de excepciones (Excepciones precisas, terminación en orden). • Ejecución fuera de orden (Ventajas y complejidad). 	B1, B2, B3	
9	RA2	<p>Emisión múltiple y paralelismo</p> <ul style="list-style-type: none"> • Paralelismo a nivel de instrucción (ILP). • Microarquitectura superescalar (Múltiples unidades de ejecución). • Ley de Moore (Impacto histórico y limitaciones actuales). 	B1, B2, B3	
10	RA2	<p>CPU multihilo y multinúcleo</p> <ul style="list-style-type: none"> • Taxonomía de Flynn (SISD, SIMD, MIMD). • Procesadores multinúcleo (Arquitecturas homogéneas/heterogéneas). • Paralelismo a nivel de hilo (Multihilo simultáneo - SMT). 	B1, B2, B3	
11	RA2, RA3	<p>Soporte a sistemas operativos</p> <ul style="list-style-type: none"> • SO multitarea (Cambio de contexto, protección). • Mecanismos hardware (Modos privilegiado/usuario, interrupciones). • Virtualización (Hipervisores, soporte en x86). 	B1, B2, B3	Prueba 2 Optimización del pipeline, operaciones multiciclo, excepciones, emisión múltiple, CPU
12	RA3	<p>Jerarquía de memoria (I)</p> <ul style="list-style-type: none"> • Conceptos básicos (Localidad, coste/rendimiento). • Memoria caché (Estrategias de correspondencia: directa, asociativa). • Políticas de reemplazo (LRU, FIFO, aleatorio). 		

13	RA3	<p>Jerarquía de memoria (II)</p> <ul style="list-style-type: none"> • Políticas de escritura (Write-through, write-back). • Coherencia de caché (Protocolos MESI, MOESI). • Memoria virtual (Paginación, tablas de páginas). 	B1, B2, B3	
14	RA3	<p>Memoria virtual y TLB</p> <ul style="list-style-type: none"> • Traducción de direcciones (MMU, fallos de página). • TLB (Translation Lookaside Buffer). • Soporte a virtualización en memoria (Nested page tables). 	B1, B2, B3	
15	RA3	<p>Sistema de E/S (I)</p> <ul style="list-style-type: none"> • Interfaces de E/S (Memoria mapeada, puertos aislados). • Técnicas de E/S (Polling, interrupciones, DMA). • Sistema de interconexión (Buses, PCI Express). 	B1, B2, B3	
16	RA3	<p>Sistema de E/S (II) y cierre</p> <ul style="list-style-type: none"> • Periféricos (Discos duros, SSD, comparativas). • Virtualización de E/S (Passthrough, emulación). • Repaso general y tendencias futuras (Computación cuántica, neuromórfica). 	B1, B2, B3	prueba 3 (énfasis en memoria y E/S) + integración de conceptos previos
17		Periodo de exámenes		
18		Periodo de exámenes		

(*) Para referencias de Resultados de Aprendizaje y Recursos, consultar el programa del curso.