

Planificación de  
**Arquitectura del Computador**



Código/s: R-222

**Identificación y características del Espacio Curricular**

Carrera/s:	Licenciatura en Ciencias de la Computación		
Plan de Estudios:	2010, TO2024	Carácter:	Obligatoria
Bloque/Campo:		Área:	Arquitectura, Sistemas Operativos y Redes
Régimen de cursado:	Cuatrimestral		
Cuatrimestre:	4º [LCC], 4º [LCC]		
Carga horaria:	105 hs. / 7 hs. semanales	Formato curricular:	Asignatura
Escuela:	Ciencias Exactas y Naturales	Departamento:	Ciencias de la Computación
Docente responsable: FEROLDI, Diego			

**Programa Sintético**

Historia e introducción. Representación de datos. Instrucciones y modos de los procesadores. Organización de la memoria. Procesadores CISC. Familia iAPX\*86. Procesadores RISC. Familia MIPS. Arquitecturas no von Neumann. Arquitecturas multiprocesadores.

**Espacios Curriculares Relacionados**

Previos Aprobados:	R-212 - Estructuras de Datos y Algoritmos I
Simultaneos Recomendados:	
Posteriores:	R-323 - Comunicaciones, R-412 - Sistemas Operativos II

**Vigencia desde 2024**

\_\_\_\_\_  
Firma Profesor

\_\_\_\_\_  
Fecha

\_\_\_\_\_  
Firma Aprob. Escuela

\_\_\_\_\_  
Fecha

Con el aval del Consejo Asesor:

## Fundamentación

Arquitectura del Computador es una asignatura del cuarto cuatrimestre (segundo cuatrimestre del segundo año) del plan de estudios de la Licenciatura en Ciencias de la Computación. El motivo de la asignatura dentro del plan de estudios es que se enseñan nociones sólidas de cómo funciona una computadora, cómo se representan los datos a nivel computacional, cómo se organizan los datos en memoria y cómo se programa a bajo nivel. Es una asignatura fundamental para poder cursar otras asignaturas tales como Comunicaciones, Sistemas Operativos y Compiladores, donde se adquieren conocimientos y competencias fundamentales para toda/o Licenciada/o en Ciencias de la Computación. También se adquieren otros conocimientos generales, útiles para otras asignaturas.

## Resultados del aprendizaje

Al finalizar el cursado los/las estudiantes serán capaces de:

RA1 Interpretar el funcionamiento de las computadoras desde el punto de vista lógico.
RA2 Utilizar las técnicas de representación computacional de tipos de datos fundamentales y aplicar dichas representaciones en códigos de programación.
RA3 Reconocer el diseño de diferentes de tipos de arquitecturas de computadoras (CISC y RISC).
RA4 Desarrollar programas elementales utilizando lenguajes de bajo nivel (lenguaje ensamblador x86-64 y lenguaje ensamblador ARM) para diferentes tipos de arquitecturas.
RA5 Interpretar la organización de los datos en memoria en relación con el funcionamiento de programas, compiladores y sistemas operativos.
RA6 Comunicar en forma adecuada los conocimientos y competencias adquiridos.
RA7 Interactuar y colaborar con otros estudiantes expresando sus puntos de vista durante la resolución de problemas.

## Competencias / Ejes transversales y Resultados del Aprendizaje

Competencia/Eje transversal al que tributa	Nivel	Resultados del Aprendizaje
CGT4-Utilización de técnicas y herramientas de aplicación en la informática	Bajo	RA1, RA2, RA3, RA4 y RA5
CGS1-Fundamentos para el desempeño en equipos de trabajo	Bajo	R7
CGS2-Fundamentos para la comunicación efectiva	Bajo	R6

## Programa Analítico

En sintonía con las expectativas de logro, resultados de aprendizaje y ejes transversales señalados, consignar los contenidos que se prevén desarrollar en el espacio curricular. Indicarlos mediante el siguiente formato:

### Unidad 1

#### Introducción a Arquitectura del Computador

- 1.1. Definición de computadora
- 1.2. Breve historia de la computación
- 1.3. Definición de Arquitectura del Computador
- 1.4. Descripción básica de la arquitectura de una computadora
- 1.5. Procesador (ALU, registros, memoria caché, contador de programa, buses, dispositivos de entrada/salida)
- 1.6. Memoria principal y memoria secundaria

- 1.7. Arquitecturas de Von Neumann y Harvard
- 1.8. Conjunto de instrucciones
- 1.9. Microarquitectura
- 1.10. Ejecución de un programa
- 1.11. Representación de programas a nivel de máquina

## Unidad 2

### Representación computacional de datos

- 2.1. Introducción
- 2.2. Organización de los datos
- 2.3. Sistemas de numeración posicionales
- 2.4. Sistema binario (formatos, conversión entre sistemas, operaciones)
- 2.5. Números con signo
- 2.6. Complementos a la base y a la base menos uno
- 2.7. Operaciones en complemento a dos
- 2.8. Otras representaciones
- 2.9. Sistema hexadecimal
- 2.10. Operaciones en sistema hexadecimal
- 2.11. Sistema octal
- 2.12. Operaciones para números de precisión arbitraria
- 2.13. Operadores de bits en lenguaje C
- 2.14. Representación de números reales con punto fijo
- 2.15. Representación de números reales con punto flotante
- 2.16. Notación científica normalizada
- 2.17. Redondeo de un número real
- 2.18. Representación computacional de números en punto flotante
- 2.19. Standard IEEE 754 para números en punto flotante
- 2.20. Operaciones de números en punto flotante
- 2.21. Manejo de errores de representación
- 2.22. Precauciones al operar con números en punto flotante
- 2.23. Densidad de los números en punto flotante

## Unidad 3

### Ensamblador y Arquitectura x86-64

- 3.1. La arquitectura x86-64
- 3.2. Lenguaje de máquina
- 3.3. Lenguaje ensamblador de x86-64
- 3.4. Registros (propósito general, especiales, SSE)
- 3.5. Endianness
- 3.6. Instrucciones
- 3.7. Comparaciones, saltos y estructuras de control
- 3.8. Manejo de arreglos y cadenas
- 3.9. Acceso a datos en memoria
- 3.10. Gestión de la pila
- 3.11. Aritmética de punto flotante
- 3.12. Instrucciones packed
- 3.13. Funciones y convención de llamada
- 3.14. Compilación de código ensamblador con GNU Assembler

## Unidad 4

### Organización y gestión de la memoria

- 4.1. Introducción

- 4.2. Jerarquía de memoria
- 4.3. Cuello de botella de von Neumann
- 4.4. Las memorias de acceso aleatorio
- 4.5. Memoria caché
- 4.6. Espacios de direccionamiento
- 4.7. Memoria virtual
- 4.8. Segmentación
- 4.9. Paginación
- 4.10. Tamaño de página
- 4.11. La tabla de paginación
- 4.12. La TLB
- 4.13. Protección de memoria
- 4.14. Políticas de reemplazo
- 4.15. Tablas de páginas multinivel
- 4.16. Caso de estudio: Intel Core i7

## Unidad 5

### Arquitecturas RISC

- 5.1. Concepto de RISC y pipelines
- 5.2. Principales características
- 5.3. Algunas arquitecturas
- 5.4. Ventajas y desventajas
- 5.5. RISC comparado con arquitecturas CISC

## Unidad 6

### Conceptos básicos del ensamblador y arquitectura ARM

- 6.1. La arquitectura ARM
- 6.2. Registros
- 6.3. Endianness
- 6.4. CPSR
- 6.5. Ejecución condicional
- 6.6. Barrel shifter
- 6.7. Valores inmediatos
- 6.8. Instrucciones ARM
- 6.9. La pila
- 6.10. Llamada a función

## Unidad 7

### Arquitecturas multinúcleo y otras arquitecturas

- 7.1. Introducción a arquitecturas multinúcleo
- 7.2. Problemas de rendimiento (hardware y software)
- 7.3. Organización multinúcleo (homogénea y heterogénea)
- 7.4. Caso de estudio: Intel Core i7
- 7.5. Arquitectura GPU
- 7.6. Introducción a OpenCL

## Modalidades de enseñanza

Clases teóricas presenciales en aula con y sin presentación audiovisual.

Clases prácticas presenciales en laboratorio resolviendo ejercicios de aplicación de los conceptos estudiados bajo la supervisión y ayuda de los docentes.

Trabajos prácticos grupales basados en la resolución de problemas de computación. Estos trabajos se desarrollarán en el laboratorio de informática con la guía de los docentes, así como también fuera del horario de clases fomentando el autoaprendizaje y el trabajo en equipo.

Preparación de temas complementarios por parte de los alumnos, y posterior discusión en clases, utilizando bibliografía previamente asignada por los docentes.

## Recursos

Detallar los recursos utilizados en el desarrollo del espacio curricular: Espacios físicos (aula, laboratorio, equipamiento informático, etc.), Recursos tecnológicos de apoyo (proyector multimedia, software, equipo de sonido, aulas virtuales, plataforma educativa, etc.), Transporte, seguros, elementos de protección para desarrollar actividades en laboratorios, empresas, fábricas, etc.

Mencionar, si lo considera conveniente, recursos didácticos especiales utilizados. Si el Departamento / Carrera / Espacio Curricular / Docente tiene algún sitio web / plataforma de referencia, indicarlo.

## Actividades de Formación Práctica

Las actividades de formación práctica consisten en la realización de trabajos prácticos en laboratorio de computación con la asistencia de los docente de la cátedra. Estos trabajos prácticos consisten en la resolución mediante algoritmos de programación de ejercicios propuestos por la cátedra. Finalmente, se realiza un trabajo práctico final de tipo globalizador que consiste en la resolución de un problema general. Los alumnos pueden optar por un tema dentro de la lista propuesta por la cátedra o proponer a la cátedra un tema que les resulte de interés.

Nº	Título	Descripción
1	Representación Computacional de Datos	Ejercicios para familiarizarse con la representación de datos orientada a la computación pero independizándose de la implementación computacional propiamente dicha. Son ejercicios para resolver con papel y lápiz aunque algunos ejercicios luego pueden realizarse utilizando computadora para chequear resultados.
2	C y sistemas de numeración posicional	Trata los sistemas de representación de números enteros en el lenguaje de programación C y los operadores de bits para manipular estos números a bajo
3	Arquitectura y lenguaje ensamblador de x86-64	Ejercicios para abordar la arquitectura de CPU x86 64 y su lenguaje de ensamblador, usando la sintaxis AT&T y la herramienta de compilación GNU. Se utilizan las operaciones más relevantes y se hace uso de los diferentes registros y subregistros. Se muestra además cómo utilizar la pila y cómo realizar llamadas a funciones.
4	Números en punto flotante usando lenguaje ensamblador	Ejercidos para trabajar la representación de números en punto flotante. En particular, trata la representación de números reales en formato de punto flotante utilizando la norma IEEE 754. Se realiza utilizando lenguaje C y Assembler. Se hace hincapié en los formatos, precisiones y utilidades. Se utilizan también instrucciones SSE.

5	Arquitectura ARM	Ejercicios para abordar la arquitectura de CPU ARM y su lenguaje de ensamblador, usando la herramienta de compilación GNU. Se hace uso de diferentes instrucciones específicas para desarrollar funciones con el objetivo de remarcar las ventajas inherentes de esta arquitectura.
6	Memoria virtual	Ejercicios básicos de memoria virtual para aplicar los conceptos básicos vistos en teoría.

## Evaluación

A lo largo del cuatrimestre los estudiantes realizan seis actividades prácticas y dos exámenes parciales. Para regularizar la asignatura se deben entregar y aprobar el 80% de las prácticas en las fechas convenidas y aprobar los dos parciales con nota mínima seis en cada uno de ellos. El primer parcial abarca las Unidades 1 y 2, mientras que el segundo parcial abarca las unidades 3 y 4. En caso de no aprobación de uno de los dos parciales hay una instancia de examen recuperatorio al final del cuatrimestre donde se puede recuperar el parcial no aprobado. El examen final en condición REGULAR consta de dos instancias: un examen escrito teórico-práctico de todos los temas de la asignatura y luego un coloquio integrador, debiendo aprobar cada instancia para acceder a la siguiente. El examen final en condición LIBRE abarca de manera más exhaustiva todos los temas de la asignatura y además incluye una instancia de examen práctico en computadora previa al examen teórico-práctico.

Resultado de Aprendizaje	Actividades/Modalidad de Enseñanza	Modalidad de Evaluación
RA1	Clases teóricas presenciales en aula con y sin presentación audiovisual. Clases prácticas presenciales en laboratorio resolviendo ejercicios de aplicación de los conceptos estudiados bajo la supervisión y ayuda de los docentes. Trabajos prácticos grupales basados en la resolución de problemas.	Exámenes parciales Trabajos prácticos Examen final Coloquio final
RA2	Clases teóricas presenciales en aula con y sin presentación audiovisual. Clases prácticas presenciales en laboratorio resolviendo ejercicios de aplicación de los conceptos estudiados bajo la supervisión y ayuda de los docentes. Trabajos prácticos grupales basados en la resolución de problemas.	Exámenes parciales Trabajos prácticos Examen final Coloquio final
RA3	Clases teóricas presenciales en aula con y sin presentación audiovisual. Clases prácticas presenciales en laboratorio resolviendo ejercicios de aplicación de los conceptos estudiados bajo la supervisión y ayuda de los docentes. Trabajos prácticos grupales basados en la resolución de problemas.	Exámenes parciales Trabajos prácticos Examen final Coloquio final

RA4	Clases teóricas presenciales en aula con y sin presentación audiovisual. Clases prácticas presenciales en laboratorio resolviendo ejercicios de aplicación de los conceptos estudiados bajo la supervisión y ayuda de los docentes. Trabajos prácticos grupales basados en la resolución de problemas.	Exámenes Parciales Trabajos prácticos Examen final Coloquio final
RA5	Clases teóricas presenciales en aula con y sin presentación audiovisual. Clases prácticas presenciales en laboratorio resolviendo ejercicios de aplicación de los conceptos estudiados bajo la supervisión y ayuda de los docentes. Trabajos prácticos grupales basados en la resolución de problemas.	Exámenes Parciales Trabajos prácticos Examen final Coloquio final
RA6	Trabajos prácticos grupales basados en la resolución de problemas.	Coloquio final
RA7	Clases prácticas presenciales en laboratorio resolviendo ejercicios de aplicación de los conceptos estudiados bajo la supervisión y ayuda de los docentes. Trabajos prácticos grupales basados en la resolución de problemas.	Trabajos prácticos Coloquio final

### Bibliografía básica

<b>Autores (Apellido, Inicial nombre)</b>	<b>Año de edición</b>	<b>Título de la obra</b>	<b>Editorial o Revista</b>	<b>Ejemplares disponibles o sitio web</b>
Andrew Tanenbaum	1992	Organización de Computadoras: Un enfoque estructurado	Prentice Hall Hispanoamericana	1
Hennessy, J.; Patterson, D.	2011	Estructura y diseño de computadores: la interfaz hardware/software	Reverté	1
Stallings, W.	1996	Computer organization and architecture: designing for performance	Prentice Hall	1
G. Wolf, E. Ruiz, F. Bergero, E. Meza	2016	Fundamentos de Sistemas Operativos	Universidad Nacional Autónoma de México	<a href="https://ru.iiec.unam.mx/2718/1/sistemas_operativos.pdf">https://ru.iiec.unam.mx/2718/1/sistemas_operativos.pdf</a>
ARM	2023	Learn the architecture - Introducing the Arm architecture Version 2.1 Release information	ARM Limited	<a href="https://developer.arm.com/documentation/">https://developer.arm.com/documentation/</a>

AMD64 Technology	2020	AMD64 Architecture Programmer's Manual Volume 1: Application Programming	AMD64 Technology	<a href="https://www.amd.com/system/files/TechDocs/24592.pdf">https://www.amd.com/system/files/TechDocs/24592.pdf</a>
------------------	------	--	------------------	---

### Bibliografía complementaria

<b>Autores (Apellido, Inicial nombre)</b>	<b>Año de edición</b>	<b>Título de la obra</b>	<b>Editorial o Revista</b>	<b>Ejemplares disponibles o sitio web</b>
Jorgenson, E.	2022	X86-64 Assembly Language Programming with Ubuntu	Ed Jorgensen	<a href="https://open.umn.edu/opentextbooks/textbooks/733">https://open.umn.edu/opentextbooks/textbooks/733</a>
Seyfarth, R.	2011	Introduction to 64 Bit Intel Assembly Language Programming for Linux	Ray Seyfarth	<a href="http://library.bagrintsev.me/ASM/Introduction%20to%2064bit%20Intel%20Assembly%20Language%20Programming%20for%20Linux.2011.pdf">http://library.bagrintsev.me/ASM/Introduction%20to%2064bit%20Intel%20Assembly%20Language%20Programming%20for%20Linux.2011.pdf</a>
Randal E. Bryant - David R. O'Hallaron	2015	x86-64 Machine-Level Programming		<a href="https://www.cs.cmu.edu/~fp/courses/15213-s07/misc/asm64-handout.pdf">https://www.cs.cmu.edu/~fp/courses/15213-s07/misc/asm64-handout.pdf</a>
Carter, P.	2006	PC Assembly Language		<a href="https://www.scs.stanford.edu/05au-cs240c/lab/reference.html">https://www.scs.stanford.edu/05au-cs240c/lab/reference.html</a>
Goldberg, D.	1991	What every computer scientist should know about floating point arithmetic	ACM Computing Surveys (CSUR), (23), 5-48	<a href="https://dl.acm.org/doi/abs/10.1145/103162.103163">https://dl.acm.org/doi/abs/10.1145/103162.103163</a>
IEEE Computer Society	2019	IEEE Standard for Floating-Point Arithmetic	IEEE	<a href="https://ieeexplore.ieee.org/document/8766229">https://ieeexplore.ieee.org/document/8766229</a>
AMD64 Technology	2021	AMD64 Architecture Programmer's Manual Volume 4: 128-Bit and 256-Bit Media Instructions	AMD64 Technology	<a href="https://www.ucr.edu/results">https://www.ucr.edu/results</a>
Kernighan, B.; Ritchie, D.	2017	The C programming language	Prentice Hall	3
Drepper, U.	2007	What every programmer should know about memory	Red Hat, Inc.	<a href="https://people.freebsd.org/~lstewart/articles/cpumemory.pdf">https://people.freebsd.org/~lstewart/articles/cpumemory.pdf</a>
Tanenbaum, Andrew S.	2003	Sistemas operativos modernos	Pearson Educación	1

Huang, S.	2009	GDB Tutorial A Walkthrough with Examples	<a href="https://www.cs.umd.edu/~srhuang/teaching/cmsc212/gdb-tutorial-handout.pdf">https://www.cs.umd.edu/~srhuang/teaching/cmsc212/gdb-tutorial-handout.pdf</a>
-----------	------	--	---

### Distribución de la carga horaria

#### Presenciales

Teóricas		40 Hs.
Prácticas	Formación Experimental	
	Resolución de Problemas vinculados a la Profesión	
	Resolución de Problemas y Ejercicios	60 Hs.
	Actividades de Proyecto y Diseño	
	Formación en la Práctica Profesional	
Evaluaciones		5 Hs.
	<b>Total</b>	<b>105 Hs.</b>

#### Dedicadas por el alumno fuera de clase

	Preparación Teórica	40 Hs.
	Preparación Práctica	60 Hs.
	Elaboración y redacción de informes, trabajos, presentaciones, etc.	
	<b>Total</b>	<b>100 Hs.</b>

### Cronograma de actividades

Semana	Unidad	Tema	Actividad
1	1	Introducción a Arquitectura del Computador	Clases teóricas. Ejercicios en papel y en computadora. Plancha de ejercicios 0.
2	2	Representación computacional de datos: de la Sección 2.1 a la 2.6	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora. Plancha de ejercicios 1.
3	2	Representación computacional de datos: de la Sección 2.1 a la 2.13.	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora. Plancha de ejercicios 1.
4	2	Representación computacional de datos: de la Sección 2.14 a la 2.19.	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora. Plancha de ejercicios 3.
5	2	Representación computacional de datos: de la Sección 2.20 a la 2.23.	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora. Plancha de ejercicios 3.
6	3	Ensamblador y Arquitectura X86-64: de la sección 3.1 a la 3.5	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora. Plancha de ejercicios 2.
7	3	Ensamblador y Arquitectura X86-64: de la sección 3.6 a la 3.8.	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora.
8	3	Ensamblador y Arquitectura X86-64: de la	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora. Plancha de ejercicios 2.

9	3	Ensamblador y Arquitectura X86-64: de la sección 3.11 a la 3.12.	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora. Plancha de ejercicios 3.
10	3	Ensamblador y Arquitectura X86-64: de la sección 3.13 a la 3.14.	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora. Plancha de ejercicios 2.
11	4	Organización y gestión de la memoria: de la Sección 4.1 a la 4.8.	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora. Plancha de ejercicios 5
12	4	Organización y gestión de la memoria: de la Sección 4.9 a la 4.16.	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora. Plancha de ejercicios 5.
13	5 - 6	Arquitecturas RISC. Conceptos básicos del ensamblador y arquitectura ARM: de la Sección 6.1 a la 6.4	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora. Plancha de ejercicios 4
14	6	Conceptos básicos del ensamblador y arquitectura ARM: de la Sección 6.5 a la 6.10	Clases teóricas. Ejemplos en pizarrón y en computadora. Trabajos prácticos en computadora. Plancha de ejercicios 4.
15	7	Arquitecturas multinúcleo y otras arquitecturas.	Clases teóricas. Ejemplos en pizarrón y en computadora.