

## **PROGRAMA de Organización de Computadoras**

**Carrera/s:** *Tecnicatura Universitaria en Programación Informática / Licenciatura en Informática*

**Asignatura:** *Organización de Computadoras*

**Núcleo al que pertenece:** *Sistemas Informáticos*

**Profesor/es:** *María Dalponte - Pablo Nieloud - Denise Pari - Federico Martínez*

**Asignaturas previas necesarias para favorecer el aprendizaje:**

*Elementos de Lógica y Programación*

**Objetivos:**

*Se espera que el estudiante:*

- (a) *Comprenda, maneje y se familiarice con conceptos fundamentales de la ejecución de programas y el funcionamiento de la computadora, en los distintos niveles de abstracción. En particular:*
  - 1. *Cómo evolucionaron los sistemas de cómputo a través de la historia*
  - 2. *Cómo se representa la información: Sistemas de Numeración y representación de texto.*
  - 3. *Cómo se construyen los circuitos de la computadora: Lógica digital.*
  - 4. *Cómo se ejecuta un programa: Enfoque de Von Neumann, ciclo de ejecución de instrucciones.*
  - 5. *Cómo son las unidades constructivas de los programas: Repertorio de instrucciones de las distintas arquitecturas.*
  - 6. *Qué componentes son necesarios para que una computadora funcione: Unidad de Control, Unidad Aritmético-Lógica, Registros, Memoria Principal, subsistema de interconexión.*
  - 7. *Cómo puede optimizarse el desempeño de una computadora: Memoria Caché, RAIDs de discos.*
  - 8. *Cómo se comunica una computadora con el exterior: Subsistema de Entrada/Salida*
- (b) *Sea capaz de diseñar circuitos elementales para entender cuales son los componentes elementales a partir de los cuales se construye la computadora, y comprender que si sabe construir estos componentes, puede construir un sistema de computos.*
- (c) *Sea capaz de confeccionar programas sencillos en lenguajes y relacionarlos con los componentes y abstracciones necesarios para que esos programas puedan ser ejecutados.*
- (d) *Adquiera habilidades actitudinales que lo prepare de mejor manera para transitar la vida universitaria y laboral.*
  - 1. *Autogestión del tiempo de estudio.*
  - 2. *Autoevaluación.*
  - 3. *Solidaridad con los compañeros / Trabajo en equipo.*

4. *Inquietud por la integración de contenidos.*

5. *Responsabilidad*

**Contenidos mínimos:**

- *Representación de la información: alfanumérico, numérico, punto fijo y flotante, ASCII. Sistema de numeración binario.*
- *Aritmética de las computadoras: Unidades. Funcionamiento y organización (modelo de Von Neumann).*
- *Unidades funcionales: Unidad Central de Proceso, Unidad de Control, memorias, ciclo de instrucciones, direccionamiento, subsistema de Memoria. Periféricos: conceptos y principio de funcionamiento. Procesadores de Entrada/Salida.*
- *Lógica digital: tablas de verdad, equivalencia de fórmulas proposicionales, circuitos combinatorios, circuitos secuenciales*
- *Arquitectura del computador: Componentes de la CPU, memoria principal y secundaria, jerarquía de memorias.*
- *Subsistema de Entrada/Salida.*
- *Lenguaje Máquina. Código fuente y código objeto.*

**Carga horaria semanal: 6 hs**

**Programa analítico:**

*Unidad 0: Evolución de las computadoras y sistema binario*

1. *Evolución de las computadoras*
2. *Sistemas posicionales: binario, octal, hexadecimal.*
3. *Sistemas de representación: alfabetos, cadenas válidas, función de interpretación, función de codificación y aritmética.*
4. *Arquitectura de Von Neumann: programa almacenado, ejecución secuencial, celda direccionable.*
5. *Componentes: CPU, Unidad de control, Unidad Aritmético-Lógica, Subsistema de Memoria, Subsistema de Entrada-Salida, Subsistema de interconexión.*

*Unidad 1: Lógica digital*

1. *Tablas de verdad*
2. *Equivalencia de fórmulas proposicionales*
3. *Circuitos combinatorios, circuitos secuenciales*
4. *Circuitos aritméticos*

*Unidad 2: Funcionamiento elemental de un sistema de cómputo*

1. *Ciclo de vida de un programa. Código fuente y código máquina*
2. *Ciclo de ejecución de instrucción. Registro de uso específico Program Counter (PC)*
3. *Modo de direccionamiento por registro*
4. *Formato de las instrucciones*
5. *Q1: repertorio de instrucciones. Registros visibles al programador. Formato de instrucción de dos operandos*

### *Unidad 3: Memoria principal y bus del sistema*

- 1. Memoria principal: Operaciones, capacidad, método de acceso, relación con circuitos digitales combinatorios*
- 2. Subsistema de interconexión: Buses. dirección, control y datos.*
- 3. Ciclo de ejecución de instrucción ampliado. Registros de uso específico Memory Buffer Register, Memory Address Register y registro de instrucción*
- 4. Q2: Modo de direccionamiento directo*

### *Unidad 4: Modularización y definición de rutinas*

- 1. Modularización de los programas. Definición de rutinas teniendo en cuenta el reuso.*
- 2. Documentación de las rutinas*
- 3. Estructura de datos Pila*
- 4. Q3: Instrucciones CALL y RET. Codificación de las etiquetas. Formato de instrucciones con un operando origen y sin operandos. Registros de uso específico: Stack Pointer*

### *Unidad 5: Sistemas de numeración enteros*

- 1. Sistema Signo magnitud: Representación de números, interpretación de cadenas, aritmética.*
- 2. Sistema Complemento a dos: Representación de números, interpretación de cadenas, aritmética.*
- 3. Sistema Exceso: Representación de números, interpretación de cadenas, aritmética.*

### *Unidad 6: Punto fijo*

- 1. Sistemas de representación de números fraccionarios: punto fijo. Representación de números en punto fijo, interpretación de cadenas*
- 2. Concepto de error: relativo y absoluto.*
- 3. Concepto de Resolución*

### *Unidad 7 : Punto Flotante*

- 1. Sistemas de representación de números fraccionarios: punto flotante. Representación de números en punto flotante, interpretación de cadenas. Resolución variable.*
- 2. Mantisa entera, fraccionaria y normalizada. Bit implícito.*
- 3. Estándar IEEE 754*

### *Unidad 8 Flujo Condicional*

- 1. Flags Zero, Carry, Overflow, Negative. Condiciones de cómputo y relación con las instrucciones aritméticas.*
- 2. Saltos absolutos y relativos. Saltos condicionales e incondicionales. Etiquetas*
- 3. Estructuras condicionales (if-then-else) en un lenguaje ensamblador.*
- 4. Q4: Instrucciones de salto relativo condicional. Instrucciones CMP y JMP*

### *Unidad 9 Iteraciones/Arreglos/Mascaras*

- 1. Modo de direccionamiento indirecto*

2. Estructura de repetición (*while/for*) en un lenguaje ensamblador
3. Estructura de datos: arreglo. Recorrido de arreglos
4. Máscaras: definición y operaciones lógicas
5. Q5: Operaciones lógicas. Formato de instrucción. Modo indirecto

#### Unidad 10: Subsistema de memoria

1. Métodos de acceso, volatilidad, métodos de escritura, métodos de borrado. Costo
2. Jerarquía de memorias.
3. Memoria cache: Función de correspondencia, políticas de reemplazo, políticas de escritura, tasas de acierto y fallo

#### Unidad 11 Subsistema de Entrada/Salida

1. Mecanismo de entrada salida programado. Mecanismo de entrada salida por interrupciones.
2. Almacenamiento externo: discos, cintas y discos ópticos. Métodos de acceso, velocidad de transferencia, capacidad, densidad.
3. RAID de discos.
4. Interfaces de video: gráfica y alfanumérica. Representación de colores y atributos.
5. Comunicación serie sincrónica y asincrónica, comunicación paralelo.

#### **Bibliografía obligatoria:**

1. "Organización y arquitectura de computadoras", Williams Stallings.
2. "Organización de Computadoras – un enfoque estructurado", A. Tanenbaum. Editorial Prentice Hall (4ta edición)

#### **Bibliografía de consulta:**

1. "Arquitectura de Computadores - Un enfoque cuantitativo", Hennessy & Patterson., Editorial Mc Graw Hill (1ra edición).
2. "Lógica para matemáticos", A. G. Hamilton, Cambridge. University Press, 1988.

#### **Organización de las clases:**

*Cada comisión tiene dos clases teórico prácticas y constan de cuatro momentos: la conexión con el tema a desarrollar y con los saberes anteriores, la conceptualización y motivación donde el docente es el mas activo y presenta los saberes nuevos, la práctica concreta donde los estudiantes trabajan individualmente o en grupo (según las particularidades de la clase) y la conclusión donde el estudiante debe afianzar los conocimientos adquiridos de manera colectiva con los compañeros.*

#### **Modalidad de evaluación:**

*La materia posee dos instancias de evaluación parcial y una instancia final integradora. Los alumnos que obtienen un promedio de 7 o más en los parciales (teniendo al menos un 6 en cada instancia) son promocionados, quedando eximidos del examen integrador. Aquellos alumnos que obtienen un promedio entre 4 y 6 deben rendir la instancia integradora. Todas las instancias de evaluación tienen dos fechas, la principal y otra de recuperación. Los alumnos que luego de agotar las instancias de recuperación parciales no*

*alcanzan el promedio mínimo de 4 son reprobados sin acceder al integrador. La nota final es el promedio entre la mejor nota de cada parcial y el integrador, si fue rendido.*

## CRONOGRAMA TENTATIVO

Semana	Tema/unidad	Actividad*			Evaluación
		Teórico	Práctico		
			Res Prob.	Lab.	
14/3	<i>Unidad 0: Evolución de las computadoras y sistema binario</i>	x	x		
21/3	Unidad 1: Lógica digital	x	x		
38/3	Unidad 2: Funcionamiento elemental de un sistema de cómputo	x	x	x	
4/4	Unidad 3: Memoria principal y bus del sistema	x	x	x	
11/4	Unidad 4: Modularización y definición de rutinas	x	x	x	
18/4	Unidad 5: Sistemas de numeración enteros	x	x		
25/4	Unidad 6: Punto fijo	x	x		
2/5	Unidad 7 : Punto Flotante	x	x		
9/5	<b>Primer parcial, primera fecha</b>				x
16/5	Unidad 8: Flujo Condicional	x	x	x	
23/5	Unidad 9: <i>Iteraciones/Arreglos/Mascaras</i>	x	x	x	
30/5	Unidad 10: Subsistema de memoria	x	x	x	
6/6	Unidad 11: Subsistema de Entrada/Salida	x	x		
13/6	<b>Segundo parcial, primera fecha</b>				x
20/6	<b>Primer parcial, segunda fecha</b>				x
27/6	<b>Segundo parcial, segunda fecha</b>				x
4/7					
11/7	<b>Integrador</b>				x

\*INDIQUE CON UNA CRUZ LA MODALIDAD