



**INSTITUTO/S:** Tecnología e Ingeniería

**CARRERA/S:** Tecnicatura Universitaria en Redes y Operaciones Informáticas

**MATERIA:** Organización de computadoras II

**NOMBRE DEL RESPONSABLE DE LA ASIGNATURA:** Martín Vernengo

**EQUIPO DOCENTE:** Martín Vernengo

**CUATRIMESTRE:** 2<sup>do</sup>

**AÑO:** 1<sup>er</sup>

**PROGRAMA N°:** 6

**(Aprobado por Consejo directivo fecha XX)**

**Instituto/s:** Tecnología e Ingeniería

**Carrera/s:** Tecnicatura Universitaria en Redes y Operaciones Informáticas

**Nombre de la materia:** Organización de computadoras II

**Responsable de la asignatura y equipo docente:** Martín Vernengo

**Cuatrimestre y año:** 2<sup>do</sup> cuat. del 1<sup>er</sup> año

**Carga horaria semanal:** 6 hs

**Programa N°:** 6

**Código de la materia en SIU:** 795

## Organización de computadoras II

### 1. Fundamentación

Esta materia estudia las relaciones funcionales entre las distintas partes de una computadora que le permiten actuar como tal y las interacciones entre dicha computadora (hardware) y los sistemas informáticos (software) que se ejecutan en la misma. Este conocimiento permite entender no solo las múltiples posibilidades que esto ofrece sino también sus limitaciones y hasta muchos de los errores que pueden ocurrir.

La temática cubierta es altamente idónea para desarrollar en el estudiante capacidades analíticas, lógicas y de modelización que le serán útiles durante toda su carrera y ejercicio profesional.

### 2. Propósitos y/u objetivos

#### Objetivos

Son objetivos de esta materia que los/as estudiantes:

- Interpreten el concepto de información desde el punto de vista estructural de la computadora.
- Conozcan los fundamentos lógicos que sirven como base a los sistemas de computación.
- Asocien los conceptos asimilados a la "Teoría de sistemas" con el comportamiento de la computadora y los conceptos físicos y tecnológicos que la sustentan.
- Terminen de desarrollar las competencias genéricas sobre las que se construye el proceso educativo universitario.

### 3. Programa sintético:

Combinacionales genéricos. Circuitos combinacionales específicos: Sumador, decodificador, multiplexor, demultiplexor, detector de paridad, comparador de magnitud y codificador de prioridades. Biestables. Circuitos secuenciales. Celda de memoria, registro de desplazamiento y contador. Organización y arquitectura del computador. Unidades funcionales. Unidad aritmético lógica (ALU). Memoria y sus niveles de jerarquía. Subsistemas de entrada y salida. Unidad de control y camino de datos.

Arquitectura. ISA. Instrucciones en código de máquina. Von Neumann. Harvard. RISC. CISC. Ciclos de instrucción. Tipos de direccionamiento. Lenguaje de transferencia de registros (RTL). Lenguaje de máquina. Lenguaje ensamblador. Código fuente y código objeto. Ensambladores, intérpretes y compiladores. Conceptos de arquitecturas superescalares y multi-procesamiento.

#### **4. Programa analítico**

##### **4.1 Organización del contenido:**

El contenido de la materia se inscribe dentro de tres grandes ejes:

- Circuitos combinacionales y circuitos secuenciales
- Unidades funcionales, caminos de datos y organización
- Lenguajes de máquina y arquitectura

##### **Unidad 1 - Lógica digital combinacional**

Repaso de álgebra de Boole. Axiomas y propiedades. Repaso de compuertas lógicas: NOT, AND, OR, XOR, NAND, NOR, XNOR. Formas normales o canónicas de una función. Minitérminos y Maxitérminos. Mapas de Karnaugh. Repaso de circuitos combinacionales específicos utilizados dentro de una CPU: comparador de magnitud, semisumador y sumador completo, decodificadores, multiplexores y demultiplexores. Repaso de Unidad Aritmético-Lógica (ALU): estructura interna. Alternativas de implementación. Funciones aritméticas unarias y binarias. Funciones lógicas unarias y binarias. Restadores con sumadores en complemento a dos. Repaso de banderas de la ALU: registro de estado de una CPU. Combinaciones nuevos abordados: “Shifter” (desplazador) divisor, multiplicador y “pass-through”. Multiplexores con vías o entradas de varios bits. Simulaciones con LOGISIM.

##### **Unidad 2 –Lógica digital secuencial.**

Circuitos secuenciales: definición. “Latch” S-R con NOR y con NAND. “Latch D” con NOR y con NAND. Desventajas de los “Latches”: necesidad de escritura sincrónica: “Flip-Flops”. Reset asincrónico. Registros desplazamiento entradas/salidas serie/paralelo. Diagramas de estado de un circuito secuencial. Contadores – diseño utilizando diagramas de estado y Mapa de Karnaugh. Simulaciones con LOGISM.

##### **Unidad 3 - Unidades de memoria.**

La función de almacenamiento y su implementación. Consideraciones tecnológicas: memorias estáticas versus dinámicas. Tiempo de acceso: latencia de decodificación de dirección y tiempo de carga. Jerarquías en las estructuras de memoria de un computador. Tipos de memoria: electrónicas volátiles y no volátiles. ROM, PROM, EPROM, EEPROM,

FLASH. Evolución histórica y diferencias. Memoria principal. Capacidad y estructura. Vinculación con los otros bloques de la computadora. Características circuitales. Dispositivos de alta impedancia. Bus: Concepto. Buses uni y bidireccionales. Bus síncrono y asíncrono. Configuraciones típicas. Señales de control: R/W, Memory Enable, diagramas de temporización. Bancos de Memoria – diseño de lógica de control. Memoria caché. Capacidad y estructura, distintos niveles posibles. Fundamentos de su utilización: principio de “localismo” de los programas informáticos. Vinculación con los otros bloques de la computadora. Caché de Datos y Caché de Programa. Administración de la memoria caché. Técnicas de asignación. Técnicas de reemplazo. Influencia sobre el rendimiento de la CPU. Tiempo medio de acceso a memoria. Elementos para almacenamiento secundario. Características. Principios de funcionamiento. Simulaciones con LOGISIM.

#### **Unidad 4 – Programación en Assembler y código de máquina.**

Diagrama en bloques o microarquitectura básica de una CPU: ALU, banco de registros, unidad de control, interfaz con memoria externa. Arquitecturas de Von Neumann y Harvard. Lenguajes ensambladores. Código de máquina. Arquitectura del Conjunto de Instrucciones (ISA): concepto general y ejemplificación con CPU simulada (Logisim) desarrollada por la cátedra. Código de operación. Operando. Instrucciones de doble y único operando. Tipos de instrucción. Vinculación del juego de instrucciones de un procesador con la estructura del mismo. Direccionamiento absoluto y relativo. Instrucciones que no acceden a memoria. Direccionamiento implícito: instrucciones de transferencia entre registros. Instrucciones LOAD para lectura de memoria con direccionamiento inmediato, directo e indirecto. Instrucciones STORE para escritura de memoria. Instrucciones aritmético-lógicas: ADD, ADDC (“ADD con Carry”), AND, OR, XOR, XNOR. Instrucciones de desplazamiento de bits. Instrucciones de salto incondicional y condicional en función de las banderas de ALU: ZF, NF, OF, CF. Implementación de programas simples. Relación del código de máquina con los lenguajes de alto nivel: compiladores e intérpretes: diferencias, ventajas y desventajas de cada uno. Pila (“Stack”), Puntero de Pila (SP) y llamado a subrutinas. Interrupciones. Simulación con CPU de la cátedra utilizando LOGISIM. Arquitecturas de acumulador, banco de registros y stack.

#### **Unidad 5 - Sistemas de entrada-salida.**

Elementos o sistemas periféricos. Concepto. Interfaces de entrada-salida(E/S). Manejo de entrada-salida. Vinculación de la CPU con sus interfaces de E/S. Control por programa de encuesta o “Polling”. Control por interrupciones. Mapeo de E/S en Memoria. Espacio de direccionamiento de E/S. CPU’s Control por acceso directo a memoria (DMA). Simulación con CPU de la cátedra utilizando LOGISIM. Priorización de periféricos. Criterios de priorización. Implementación. Prioridad establecida por hardware o software. Técnicas de implementación de interrupciones. Tiempo real. Periféricos estándar. Vinculación entre la interfaz de entrada salida y el periférico controlado. Distintos tipos de interfaz. Interfaces de comunicaciones. Comunicación serie. Breve reseña de Tecnologías de dispositivos de E/S: almacenamiento magnético. Pistas y Sectores del disco. Sistemas RAID. Tecnologías de pantalla: LCD, LED, PDP, OLED.

#### **Unidad 6–Ciclo de Instrucción y Unidad de control.**

Registros básicos para las etapas del ciclo de instrucción: PC (ProgramCounter), IR (InstructionRegister), MAR (MemoryAddressRegister), entre otras posibles implementaciones. Conmutación del camino de datos y control de las restantes unidades funcionales. Etapas del ciclo: búsqueda de instrucción, decodificación de instrucción, búsqueda de operandos, ejecución. Esquema secuencial de señales de control requeridas para algunos ejemplos de instrucciones de código de máquina/Assembler previamente vistas. Unidad de Control: implementaciones combinacionales y secuenciales (micro-programadas).

### **Unidad 7 - Introducción a las arquitecturas modernas.**

Arquitecturas CISC versus RISC: comparación de características principales y principios de funcionamiento. Ejemplos: Intel versus ARM / RISC V. Medidas de velocidad de procedimiento: megainstrucciones por segundo o MIPS. "Benchmarks" o referencias de comparación: Dhrystone MIPS. Paralelismo a nivel de instrucción: Pipelining o Segmentación de procesamiento. Arquitecturas superescalares. Riesgos de Pipeline: riesgos por saltos. Riesgos de lectura después de escritura ("ReadAfterWrite" - RAW). Otros: WAR, WAW. Pipelines con predicción de saltos y ejecución especulativa. Arquitecturas combinadas Von Neuman para acceso a memoria principal / Harvard para Caché. Otras arquitecturas de computadoras: multiprocesamiento y "procesadores multinúcleo". "Hilos de Programa".

#### **4.3 Bibliografía y recursos obligatorios:**

- **Tocci., R.** (1993). "Sistemas digitales, principios y aplicaciones" - 5ta edición. México, Prentice Hall Hispanoamericana.
- **Tanenbaum, A.** (2000) "Organización de computadoras" - 7ma edición. México, Editorial Prentice Hall.

#### **4.4 Bibliografía optativa:**

- **Null, L.; Lobur, J.** (2015). "Computer Organization and Architecture". 4ta. U.S.A, Edición. Jones & Bartlett Learning.
- **Quiroga, P.** (2010). "Arquitectura de computadoras". 1ra edición. Buenos Aires: Ed Alfaomega, Grupo Editor Argentino.
- **Morris Mano, M.** (1994). "Arquitectura de computadoras". México, Ed. Prentice Hall
- **Morris Mano, M.** (1991). "Ingeniería computacional: diseño del hardware". México, Ed. Prentice Hall.
- **Floyd, T.** (2006). "Fundamentos de sistemas digitales" 7ma edición. España, Madrid: Ed. Prentice Hall.

### **5. Metodología de enseñanza:**

El curso seguirá la dinámica de la clase invertida, estructurada sobre estos 3 momentos:

- Clases teóricas virtuales del tipo sincrónico, dictadas por los profesores de la asignatura.
- Clases presenciales, con prácticas dirigidas guiadas los profesores o auxiliares de la asignatura.

- Prácticas individuales no dirigidas, realizadas en formato virtual asincrónico.

Las prácticas asociadas a la mayoría de las unidades se apoyarán en el uso de simuladores de circuitos digitales para fortalecer no solo la comprensión de dichos temas sino fomentar el desarrollo de competencias.

### **Plan de trabajo en el campus:**

El campus virtual es un espacio fundamental para el desarrollo de la asignatura. En el aula virtual se propondrá material educativo consistente en: documentos con apuntes de clase, videos introductorios o ampliatorios de los temas de clase, ejercicios resueltos, guías de trabajos prácticos y tareas para la entrega de los mismos. También incluirá un foro de consultas, el programa y el cronograma de la asignatura.

### **6. Actividades de investigación y extensión (si hubiera)**

No aplica

### **7. Evaluación y régimen de aprobación**

#### **7.1 Aprobación de la cursada**

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos. Todas las instancias evaluativas deberán tener una instancia de recuperatorio. Podrán acceder a la administración de esta modalidad solo aquellos y aquellas estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial.

Siempre que se realice una evaluación de carácter recuperatorio, la calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

El estudiante deberá poseer una asistencia no inferior al 75% en las clases presenciales y tener entregados y aprobados el 75% de los trabajos prácticos realizados durante la cursada.

#### **7.2 Aprobación de la materia**

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

**Promoción directa:** tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.

**Evaluación integradora:** tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos estudiantes que hayan aprobado la cursada con una nota de entre cuatro(4) y seis (6) puntos.

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrado, preferentemente, por el/la docente a cargo de la comisión. Se

aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia.

La nota obtenida se promediará con la nota de la cursada.

**Examen final:** Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.

### 7.3 Criterios de calificación

La calificación de cada evaluación se determinará en la escala 0 a 10, con los siguientes valores: 0, 1, 2 y 3: insuficientes; 4 y 5 regular; 6 y 7 bueno; 8 y 9 distinguido; 10 sobresaliente.

## 8. Cronograma

El siguiente cronograma establece el marco de trabajo en función de los temas a abordar, su importancia y complejidad. Las actividades se conformarán de manera presencial, virtual o combinadas y se comunicarán al inicio de la cursada de manera de fijar los encuentros.

Semana	Tema	Actividad
1	Introducción: objetivos de la materia. Repaso sobre combinacionales. Decodificador, Multiplexores de 1 bit por vía. ALU, suma con y sin signo. Banderas: ZF, NF, CF, OF.	Teoría/Práctica
2	Combinacionales: Shifter. Multiplexor de varios bits por vía. Secuenciales: definición, Latch SR con NOR y NAND. Latch D con NAND. Flip-Flop D con NAND de flanco positivo y negativo. Conformación de un banco de registros. Registros de desplazamiento.	Teoría/Práctica
3	Secuenciales: diagramas de estado. Contadores: análisis y diseño. Buses unidireccionales. Estado de alta impedancia y buses bidireccionales. Memoria: tipos (estáticas y dinámicas, volátiles y no volátiles). Tiempo de Acceso. Diagramas de tiempo de escritura y lectura.	Teoría/Práctica
4	Memoria Caché. Almacenamiento magnético. Jerarquías de almacenamiento de datos. Introducción a arquitectura de computadoras y lenguaje Assembler. Tipos de instrucciones. LOAD y STORE en modo implícito, inmediato y directo. Programas simples.	Teoría/Práctica
5	Traducción de sentencias aritméticas de alto nivel con y sin signo a pequeños programas Assembler. Saltos condicionales. Traducción de sentencias de alto nivel con toma de decisión (de tipo "if" y "while") a Assembler.	Teoría/Práctica
6	Corrección y análisis de TPs ya realizados en clase. Pila o "stack" y "stack pointer". Saltos a subrutinas.	Teoría/Práctica

7	Instrucciones de Entrada/Salida (E/S). Espacio de memoria de E/S. Acceso por Polling y por mapeo. Introducción a interrupciones. Repaso de temas antes del parcial (hasta clase 6).	
8	Primer evaluación parcial.	Evaluación
9	Repaso errores del parcial (consolidación). Repaso de sistemas de E/S. Acceso a E/S vía interrupciones. Repaso modos de direccionamiento e introducción a modo de direccionamiento indirecto.	Teoría/Práctica
10	Utilización de direccionamiento indirecto en programas Assembler. Práctica de programas Assembler para control de dispositivos de E/S con diversos métodos (Polling / interrupciones). Compiladores e Intérpretes: ventajas y desventajas.	Teoría/Práctica
11	Unidad de control: estructura básica. Camino de datos: arquitectura básica con ALU, Shifter, banco de registro y multiplexores. Ciclo de Instrucción. Señales de control: implementación del "Instruction fetch". Microprogramas de instrucciones simples de tipo LOAD y STORE.	Teoría/Práctica
12	Microprograma de instrucciones aritméticas: ADD, ADDC (ADD con carry), INC, DEC. Implementación paso a paso con maqueta de CPU. Segmentación de instrucciones o "pipeline". Riesgos de Pipeline. Pipeline Superescalar.	Teoría/Práctica
13	Pipeline con predicción de saltos y ejecución especulativa. Medición de performance: MIPS. Arquitecturas CISC y RISC: historia, diferencias y aplicaciones. Ejemplos: Intel, ARM, RISC V. Multiprocesamiento y CPUs multinúcleos.	Teoría/Práctica
14	Segunda evaluación parcial.	Evaluación
15	Práctica en Assembler y microinstrucciones.	Teoría/Práctica
16	Recuperatorios.	Evaluación