



CARRERA: Ing. Electrónica

PROGRAMA ANALÍTICO DE ASIGNATURA

NOMBRE DE LA ACTIVIDAD CURRICULAR: INFORMÁTICA I

Año Académico: 2023

Área: Técnicas Digitales

Bloque: Tecnologías Básicas

Nivel: 1

Tipo: Obligatoria

Modalidad: Anual

Cargas horarias totales:

<i>Horas reloj</i>	<i>Horas cátedra</i>	<i>Horas cátedra semanales</i>
120	160	5

COMPOSICIÓN DEL EQUIPO DE CÁTEDRA:

Director de Cátedra:

Prof. Asociado Ing. Marcelo Ángel Trujillo

Profesores:

Prof. Adjunto: Ing. Daniel Pellettieri

Prof. Adjunto: Lic. Andrea Fleischman

Prof. Adjunto: Ing. Jerónimo Atencio

Prof. Adjunto: Ing. Oscar Paniagua

Prof. Adjunto: Ing. Nahuel González

Prof. Adjunto: Ing. Gabriel Mandrut

Prof. Adjunto: Ing. Mariano Gonzalez

Prof. Adjunto: Ing. Ignacio Bonelli

Prof. Adjunto: Ing. Héctor Spataro

Prof. Adjunto: Ing. Martín Marino

Prof. Adjunto: Ing. Roberto Gómez

JTP:

Ing. Héctor Levi

ATP 1°:

Ing. Misael Cudek

Ing. Marcelo Howlin



Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

Ing. Javier Ballofet
Ing. Matías Amitrano
Ing. Alexander Jiricny
Ing. Gaston Coustau
Ing. Durante Diego
Ing. Fernando Posse
Ing. Pablo Sebastián Sisevich

ATP 2°:

Sr. Ignacio Ziccardi
Srta. Milagros Rojas
Sr. Sebastián Sisevich

FUNDAMENTACIÓN

Informática I es la asignatura integradora del 1° nivel de la carrera Ing. Electrónica y pertenece al bloque de asignaturas de Tecnologías Básicas y al área de Técnicas Digitales.

El área de las Técnicas Digitales está constituida por:

- Informática I
- Informática II
- Técnicas Digitales I
- Técnicas Digitales II
- Técnicas Digitales III

Informática I es una asignatura que reúne varias condiciones particulares que le confieren un alto valor estratégico dentro de la carrera de Ingeniería Electrónica: Es una asignatura Integradora del primer nivel de la carrera, condición que le impone tener que articular sus contenidos con las restantes asignaturas de dicho nivel. Asimismo conforma, junto a Diseño Asistido por Computadoras, el bloque de asignaturas del primer nivel de la carrera que depende del Departamento de Electrónica, y que por lo tanto tiene contenidos directamente relacionados con la especialidad que los estudiantes han elegido, de modo que es su primer contacto con ella.

Por ello su dictado debe ser encarado, no tan solo como una asignatura formativa para enseñarle al alumno a programar en C, sino además y muy especialmente como un elemento movilizador para los estudiantes, que les permita lograr resultados concretos en la aplicación de los contenidos, de modo de transformar éstos en un poderoso estímulo para avanzar a los siguientes niveles de la carrera en busca de más. También es importante que como materia integradora, los alumnos puedan encontrar aplicación



a los conceptos de Álgebra y Geometría Analítica y Análisis Matemático I, para comprobarlos en forma práctica, y encontrar así cabal sentido a su estudio. Por ello si bien el tema de cálculo numérico se formaliza en una de las unidades temáticas de la asignatura, su aplicación práctica se extenderá a lo largo de todo el programa, ya que la teoría se ha abordado en las asignaturas mencionadas, e Informática I se enfoca en su aplicación en la práctica de programación, para resolver problemas en el campo de la Ingeniería Electrónica.

COMPETENCIAS DE EGRESO ESPECÍFICAS A LAS QUE CONTRIBUYE:

Competencia	Competencias de Actividades reservadas			Competencias de Alcances
	Baja	Media	Alta	
CE 1.1: Diseñar, proyectar y calcular sistemas, equipos y dispositivos de generación, transmisión y/o procesamiento de campos y señales analógicos y digitales; circuitos integrados; hardware de sistemas de cómputo de propósito general y/o específico y el software a él asociado; hardware y software de sistemas embebidos y dispositivos lógicos programables; sistemas de automatización y control; sistemas de procesamiento y de comunicación de datos y sistemas irradiantes, para brindar soluciones óptimas de acuerdo a las condiciones técnicas, legales, económicas, humanas y ambientales.		X		
CE 1.2: Plantear, interpretar, modelar y resolver los problemas de ingeniería descritos en CE 1.1		X		

COMPETENCIAS DE EGRESO GENÉRICAS A LAS QUE CONTRIBUYE:

Competencia	Baja	Media	Alta
Identificar, Formular y resolver problemas de Ingeniería Electrónica	X		
Utilizar de manera efectiva las técnicas y herramientas de aplicación de la ingeniería Electrónica.	X		
Desempeñarse de manera efectiva en equipos de trabajo.	X		
Comunicarse con efectividad.	X		
Aprender en forma continua y autónoma.	X		



OBJETIVOS (APRENDIZAJE/LOGROS A ALCANZAR)

- Aprender las bases y elementos constitutivos de un sistema de cómputo tanto en sus componentes de hardware como de software y sus sistemas de representación numérica.
- Adquirir las bases metodológicas de desarrollo de software que les permitan diseñar algoritmos para resolver problemas determinados, discriminando entre las diferentes alternativas para resolverlo cuál es la más eficiente con fundamento teórico.
- Implementar dichos algoritmos en forma práctica utilizando lenguajes de programación estructurada, empleando diferentes recursos de este, manejo de datos complejos y recursos de bajo nivel.
- Dominar el funcionamiento, las posibilidades, y el empleo de las diferentes herramientas de desarrollo.
- Comprender la necesidad de un sistema operativo en un sistema computacional complejo para poder dar un contexto a las plataformas sobre las que desarrollaremos nuestros programas.
- Entender los diferentes sistemas de representación para números en un sistema de cómputo, las diferentes bases numéricas de uso común, sus reglas de conversión, tanto para números enteros con y sin signo, como las representaciones estándar de la notación científica tanto en punto fijo como punto flotante, para comprender la diferente codificación de las variables en un programa.
- Comprender cabalmente el funcionamiento, las posibilidades, y empleo de las diferentes herramientas de compilación, linkeo, debugging, documentación y control de versiones, y como automatizar y configurar su utilización, para poder implementar programas en diferentes entornos de desarrollo.
- Utilizar efectivamente las interfaces de programación (APIs, Syscalls. o como se las llegue a denominar en el futuro) para poder interactuar en forma adecuada con el Sistema Operativo que administra los recursos de hardware en un sistema computacional.
- Aplicar los conocimientos adquiridos en las asignaturas de ciencias básicas para la resolución de problemas reales.



- Desarrollar el concepto de buenas prácticas de programación de manera incipiente pero sostenida, para desarrollar esta competencia a lo largo de su trayecto académico.

CONTENIDOS

Contenidos mínimos

- Estructura de un sistema computacional.
- Sistemas de numeración y aritmética binaria.
- Interpretación y resolución de problemas, herramientas de representación de algoritmos.
- Lenguaje de programación estructurada.
- Contenedores de datos complejos.
- Uso del lenguaje en aplicaciones de bajo nivel.

Contenidos analíticos

Unidad Temática 1 - Estructura de un sistema computacional

Definición de unidades fundamentales (bit y byte) y sus múltiplos. Definición de memoria. Capacidad de memoria. Tipos de memoria. Buses. Unidad Central del Proceso (C.P.U). Unidad Aritmética y Lógica (ALU). Unidad de Control. Contador de Programa. Dispositivos de entrada/salida (I/O). Ejecución de instrucciones. Fase de búsqueda de instrucciones. Secuencia de Instrucciones: Programa. Conceptos de Hardware y Software. Interacción entre ambos elementos. Asimilación de los conceptos teóricos enunciados en el modelo teórico, de modo que los estudiantes puedan verificarlos tangiblemente en diferentes plataformas: PC, Embedded PC, Microcontroladores.

Unidad Temática 2 - SISTEMAS DE NUMERACIÓN Y REPRESENTACIÓN NUMÉRICA

Aritmética Binaria: introducción. Los sistemas de numeración y su evolución histórica. Sistemas de numeración decimal, binario, octal y hexadecimal. Pasajes entre sistemas de números enteros y positivos. Representación de Números signados: Convenio de signo y magnitud. Convenio de complemento a uno. Convenio de complemento a dos. Operaciones de adición y de sustracción utilizando el convenio de complemento a dos. Representación de números fraccionarios. Punto flotante. Precisión y truncado. Formato IEEE 754. Representación de caracteres: Binario Codificado Decimal (BCD), ASCII.

Unidad Temática 3 - INTRODUCCIÓN A LA PROGRAMACIÓN ESTRUCTURADA

Interpretación de enunciados. Ideas sobre programas y datos. Algoritmos. Estructuras de control de flujo: Secuencia, selección, iteración, salto incondicional. Eliminación del Salto Incondicional. Primer Paradigma de Programación: Programación estructurada. Implementación de algoritmos sencillos mediante pseudocódigo y/o métodos gráficos.



Utilización del Modelo Top Down y Bottom-Up para desarrollo de problemas de programación paso a paso.

Unidad Temática 4 - EL SOPORTE DEL SISTEMA OPERATIVO

Conceptos Básicos sobre Sistemas Operativos modernos. Objeto y Funciones principales de un sistema operativo, Tipos de sistemas operativos. Referencias a Sistemas de uso cotidiano: Introducción a Linux. Herramientas de desarrollo: El Compilador, el Linker. Su relación con el Hardware y el Sistema Operativo. Uso de utilidad make. Ejemplos de Makefile. Introducción al desarrollo de software: Programa fuente, programa objeto, programa ejecutable. Fases de compilación y vinculación de programas, en el entorno de un sistema operativo.

Unidad Temática 5 - INTRODUCCIÓN AL LENGUAJE C

Introducción a la sintaxis del lenguaje C. Primer ejemplo: Hola Mundo. Identificación de los elementos de la sintaxis. Preprocesador. Archivos de cabecera, biblioteca estándar del lenguaje C. Herramientas: Uso de un editor de texto, compilador, linker y entornos integrados de desarrollo.

Unidad Temática 6 - VARIABLES

Concepto de variable, el tamaño según la arquitectura, declaraciones (tipos de datos). Constantes. Variables enteras. Representación en memoria según almacenamiento: little endian y big endian. Variables de punto flotante. Variables de almacenamiento de direcciones (punteros). Reglas de alcance (scope): Variables globales y globales de módulo (static). Modificador extern: visibilidad de una variable global para todos los módulos que conformen un proyecto. Variables locales, locales de bloque y locales estáticas. Cast. Enumeraciones. Definición de nuevos tipos (typedef)

Unidad Temática 7 - OPERADORES A NIVEL DE VARIABLE

Operadores de asignación. Operadores aritméticos. Operadores de incremento y decremento. Operadores unarios. Operadores relacionales y lógicos; expresiones condicionales. Valor resultante a partir de su ejecución. Operadores de evaluación (expresiones condicionales). Precedencia.

Unidad Temática 8. CONTROL DE FLUJO EN LENGUAJE C

Implementación de la Estructura de Selección: Proposición if. Proposición if-else. Proposición switch-case. Estructura de iteración: ciclo while, ciclo do-while, ciclo for. Sentencias break y continue. Control de flujo con el preprocesador: compilación condicional. Implementación de algoritmos en lenguaje C.

Unidad Temática 9. - FUNCIONES EN LENGUAJE C

Definición de una función. Prototipo de una función. Argumentos y tipo de retorno. Proposición return. Llamada con el contenido de variables, con la dirección de variables y con la dirección de funciones. Archivos de cabecera. Programación modular. Funciones estáticas (visibilidad). Recursividad: introducción, funciones elementales, comportamiento del stack. Macros.



Unidad Temática 10 - ARREGLOS EN C

Declaración de arreglos (char, int, long, estructuras etc.). Concepto de vector (arreglo de una dimensión) y de matriz (arreglo de dos dimensiones). Inicialización de los arreglos. Algoritmos de ordenamiento. Ordenamiento por más de un criterio. Algoritmos de búsqueda en arreglos. Algoritmos de búsqueda e inserción en arreglos. Pasaje de la dirección del comienzo de arreglos como argumentos a funciones. Ejemplos de uso de arreglos: desarrollo de histogramas, cálculo de media y desvío. Integración con álgebra: detección de bordes mediante el máximo de la derivada. Caso de aplicación de vectores de variables enteras de un byte: strings. Fundamentos de aplicación de strings y carácter. Desarrollo de funciones asociadas a strings, asociadas. Argumentos por línea de comandos.

Unidad Temática 11- ESTRUCTURAS Y UNIONES

Definición de estructura. Acceso a los miembros de una estructura. Punteros a estructuras. Funciones con Estructuras. Estructuras autoreferenciadas. Definición de Unión. Acceso a los miembros de una unión. Funciones con Uniones.

Unidad Temática 12 - PUNTEROS EN C

Concepto de dirección. Aritmética de punteros. Inicialización de punteros. Punteros a punteros. Vector de punteros. Relación entre punteros y arreglos. Ordenamiento de estructuras utilizando vector de punteros. Inicialización de punteros y reserva de memoria: malloc y free. Punteros a función. Declaraciones complejas.

Unidad Temática 13 - OPERADORES A NIVEL DE BIT Y CAMPOS DE BIT

Operadores lógicos, Operadores de desplazamiento. Campos de bit. Definición. Acceso a los miembros de un campo de bits Campos de bit vs. operadores a nivel de bits. Uniones y campos de bit.

Unidad Temática 14 - STREAMS

Funciones del kernel. Archivos. Concepto de Flujo de datos (streams). Tipos de stream conocidos: stdin, stdout, stderr. Clase de nodo más común: el archivo. Funciones del kernel: File I/O Creación, apertura y cierre de archivos: funciones creat, open, close. Concepto y significado de File Descriptor. Funciones para lectura y escritura de bajo nivel: read , y write. Accesos secuencial función lseek. Funciones fcntl e ioctl. Concepto “everything is a file”. Uso de las funciones de archivos para acceder a dispositivos de E/S (acceso a E/S de alto nivel). Espera del dispositivo de E/S. Ejemplos de acceso al dispositivo de audio de una PC. Funciones de la librería estándar de C para el manejo de archivos, concepto de archivos de “texto” y “binario”

Unidad Temática 15. – MULTIPROCESAMIENTO: HILOS

Introducción al multiprocesamiento. Creación de threads, necesidad de la herramienta. Aplicaciones típicas (espera sincrónica de respuesta, paralelismo para performance, etc.). Funciones asociadas. Espera a la finalización de threads y threads detached .



Implicancias de espera a la finalización de threads. Uso y riesgo de threads detachados. Retorno de resultados de threads: Retorno mediante argumentos al thread. Retorno mediante variables globales. Retorno mediante finalización del thread. Creación y destrucción de mutex. Atributos de los mutex. Su utilización como sincronización entre threads y sus implicancias. Bloqueo indefinido (deadlock). Concepto de deadlock. Cómo evitarlos.

Unidad Temática 16. - IPC: SEÑALES y SOCKETS

Concepto de redes. Intercomunicaciones de procesos residentes en computadores diferentes a través de una red de datos: sockets. Acceso de lectura y escritura mediante read () write (). Ejemplos Cliente – Servidor. Señales

Unidad Temática 17 - APLICACIONES DE MÉTODOS NUMÉRICOS

Articulación con las asignaturas Álgebra y Geometría Analítica y Análisis Matemático I, desarrollando en lenguaje C aplicaciones que lleven a casos reales de aplicación los temas siguientes: Determinación de raíces reales. Diversos Métodos de Resolución Aproximación de funciones. Interpolación polinomial. Interpolación de Hermite. Integración Numérica. Método de los trapecios. Método de Simpson.

DISTRIBUCIÓN DE CARGA HORARIA ENTRE ACTIVIDADES TEÓRICAS Y PRÁCTICAS

Modalidad organizativa de las clases	Horas Reloj totales presenciales	Horas reloj virtuales totales	Horas totales
Teórica	48	0	48
Formación práctica	72	0	72

Tipo de prácticas	Horas Reloj totales presenciales	Horas reloj totales virtuales	Lugar donde se desarrolla la práctica
Formación experimental	-	-	-
Problemas abiertos de Ingeniería (ABP)	60	0	Laboratorios 101, 104, 105, 108, 109, 110 del Dpto de Ing. Electrónica UTN.BA sede Medrano y Laboratorios 61, 62, 63 y 64 del Dpto de Ing. Electrónica
Proyecto y diseño	12	0	Laboratorios 101, 104, 105, 108, 109, 110 del Dpto de



			Ing. Electrónica UTN.BA sede Medrano y Laboratorios 61, 62, 63 y 64 del Dpto de Ing. Electrónica
Otras:	-	-	-
Práctica supervisada	-	-	-
Total de horas	72	0	

ESTRATEGIAS DE ENSEÑANZA Y ACTIVIDADES DE APRENDIZAJE

La aproximación a la enseñanza toma los aportes de los enfoques pedagógicos constructivistas y cognitivistas. Desde estas concepciones se trata de generar situaciones problemáticas (principalmente en la primera mitad del año) para que los estudiantes pongan en juego sus ideas previas e interactúen con los nuevos saberes de forma tal que puedan construir nuevos conocimientos producto de esta interacción.

Como otra estrategia metodológica se utiliza las presentaciones como disparador temático en soportes de diapositivas (ppt) y videos para poner en disponibilidad ideas previas por parte de los estudiantes a modo fomentar aprendizajes significativos.

Para alcanzar las metas propuestas se debe conferir a la implementación práctica de la asignatura una plataforma de desarrollo a un entorno operativo, estándar, de fácil instalación, ampliamente documentado y disponible sin costo para los estudiantes. Por tal motivo se ha elegido trabajar sobre el sistema operativo Linux, ya que se entendió que estos requisitos eran plenamente satisfechos por cualquiera de sus distribuciones.

Respecto de los estudiantes, es muy factible que en algunos casos este sea su primer contacto con Linux, sin embargo, al igual que en todas las carreras que lo implementan desde su primer año, contando con la guía adecuada por parte de los profesores y auxiliares, podrán adoptar su uso sin dificultades, ya que en cuanto a capacidades gráficas es muy similar a las versiones actuales de Windows, con las que seguramente los estudiantes estarán familiarizados.

Se propone trabajar no solo sobre los aspectos relacionados con la calidad de los contenidos, sino al mismo tiempo motivando a los estudiantes a través del desarrollo de aplicaciones que despierten su vocación y den sentido al resto de los contenidos impartidos desde el área de Ciencias Básicas haciendo que el tránsito por los primeros años de la carrera les sea más estimulante. Y por supuesto tener una fuerte interacción con las asignaturas del área de Matemáticas a través del desarrollo de trabajos prácticos que incluyan los métodos de cálculo numérico formalizados en las asignaturas Análisis Matemático I, Álgebra y Geometría Analítica, y Física I, y permitan a los estudiantes comprobar su aplicación práctica, y los movilicen a la formulación de conclusiones acerca de su uso.



A lo largo de la cursada los estudiantes realizan cuatro trabajos prácticos de clase y laboratorio que se irán evaluando en cada curso en el marco de la evaluación continua.

Los ejercicios van creciendo en dificultad y promueven procesos cognitivos que permiten un abordaje integral de los temas involucrados en la asignatura, para poder ir avanzando hacia la construcción de un ejercicio integrador cuya presentación se realizará en forma individual sobre la finalización el segundo cuatrimestre.

A partir de la experiencia de virtualidad sincrónica realizada durante los dos años de Pandemia, y considerando las posibilidades tecnológicas, a partir del CL2022 Informática I incorpora la modalidad de dictado donde se contempla la virtualidad sincrónica.

La propuesta consiste entonces en incorporar en la programación didáctica de Informática I la estrategia de clases dictadas en virtualidad sincrónica, apoyándonos en los recursos institucionales y tecnológicos con los que ya contamos donde por supuesto se destaca la plataforma Moodle, punto de encuentro off line entre estudiantes y docentes a partir del Aula Virtual.

A través de esa estrategia, se persigue el objetivo de impulsar la metodología de Aula Invertida que dio buenos resultados durante el período de Pandemia, especialmente en 2021 dado que el profesorado de la Cátedra contaba con la valiosa experiencia de 2020 que supo capitalizar.

Esta metodología se basa en considerar diferentes momentos para el aprendizaje que no quedan circunscritos exclusivamente al momento de la clase sincrónica.



La gran generación de materiales didácticos en video y diapositivas realizados durante 2020 y 2021 que, ahora editados, constituyen verdaderos objetos de aprendizaje, serán la base para el momento ANTES. La posibilidad de incorporar en la previa estos materiales, permite, entre otras cosas, que el estudiante pueda asimilar los conceptos a su ritmo. El video se puede ver, parar, volver a revisar. Esto colabora con privilegiar el ritmo del propio estudiante. Además, los estudiantes deberán realizar actividades específicas de evaluación formativa (autoevaluaciones) prevista en el Aula Virtual al efecto.



El momento DURANTE se constituirá en el espacio de reflexión y consolidación de los temas, así como también el del pasaje a la práctica de aquellos temas que así lo requieran. Este momento, siempre sincrónico, podrá estar mediado por tecnologías de video comunicación, tipo Zoom.

Específicamente los encuentros planificados para realizar actividad práctica, teórica y las evaluaciones (aproximadamente un 80% de la carga horaria), se realizarán de manera presencial física preferentemente en el laboratorio del Departamento de Ing. Electrónica.

El 20% restante se realizará, también de manera sincrónica, por medio de una plataforma de video comunicación (Zoom) cumpliendo con las condiciones requeridas para ser considerada una presencialidad remota.

El momento DESPUÉS será, como en la modalidad tradicional, de mayor responsabilidad por parte de quien aprende. Se incluye allí la realización nuevamente de la evaluación formativa prevista para cada módulo a los efectos de poder valorar el aprendizaje realizado. Las métricas del resultado de las evaluaciones citadas son generadas automáticamente por Moodle y de fácil revisión por parte del equipo docente.

Desde estas concepciones, se trata de generar situaciones problemáticas para que los estudiantes pongan en juego sus ideas previas e interactúen con los nuevos saberes de forma tal que puedan construir nuevos conocimientos producto de esta interacción.

En la programación didáctica establecida en el apartado “Cronograma estimado de clases” se establece una planificación para cubrir 32 encuentros que es la duración estándar de la cursada. A partir de esta propuesta, cada profesor a cargo de cada curso deberá adecuarlo a las circunstancias propias del día de cursado específico. Como definición general, las actividades prácticas y las evaluaciones se consideran presenciales.

MODALIDAD DE EVALUACIÓN

Modalidad

- Dos parciales individuales con dos recuperatorios por cada parcial tanto en diciembre como en febrero del año siguiente al cursado.
- Un trabajo práctico integrador
- Trabajos prácticos de aula y laboratorio.

La aprobación del trabajo práctico integrador consistirá en:



- Presentación documentada y funcional sobre la finalización del segundo cuatrimestre.

Requisitos de regularidad:

- Los dos parciales aprobados con 6 o más puntos.
- Trabajo práctico integrador aprobado.
- 75 % de los trabajos prácticos de laboratorio y aula aprobados.
- Cumplir con los requisitos de asistencia establecidos por la Universidad

Requisitos de aprobación directa:

Criterio 3A Resolución 216-17 del CD FRBA	<u>Característica principal:</u> 2 (dos) instancias de evaluación con nivel de exigencia equivalente. <u>Instrumento de evaluación para la promoción:</u> las evaluaciones mencionadas. Primer parcial: 6 (o más). Segundo parcial 8 (o más) Se puede alcanzar la promoción hasta incluso en la recuperación de las evaluaciones parciales (ver configuración RPAD) Sumatoria 15 o más, siempre que el primer parcial tenga como mínimo un 6 3 opcionales: AN – PC y RPAD
---	--

RPAD (Cantidad de Recuperatorios permitidos para la Aprobación Directa): 1(uno) solo durante diciembre del año de cursada.

PC (Pruebas Complementarias): Es una nueva evaluación para indagar exclusivamente sobre los temas que en algún examen parcial no alcanzaron el nivel suficiente. Se aplica exclusivamente a estudiantes que ya tienen aprobados los parciales (notas 6 o 7) pero no les alcanza para promocionar. Se admite el complemento de una sola de las evaluaciones parciales.

AN (Actualización de notas): Si el estudiante se presenta a la recuperación de un parcial con el objeto de levantar la nota aspirando a promocionar, pone en juego la calificación obtenida previamente (en caso de que tuviera el parcial aprobado (notas 6 o 7) en primera instancia.

ARTICULACIÓN HORIZONTAL Y VERTICAL CON OTRAS MATERIAS

Como asignatura integradora del primer nivel, debe tener relación con las asignaturas del mismo nivel (Integración Horizontal), encontrando aplicaciones a los conceptos de Álgebra y Geometría Analítica y Análisis Matemático I, para comprobarlos en forma práctica, y darle así cabal sentido a su estudio. Para ello si bien el tema de cálculo numérico se formaliza en una de las unidades temáticas de la asignatura, su aplicación práctica se diseminará a lo largo de todo el programa ya que la teoría se ha impartido



en las asignaturas mencionadas. Asimismo, debe ser fluida la relación secuencial de un nivel al siguiente, colaborando así con la Integración Vertical; por pertenecer al área de las Técnicas Digitales, Informática I articula naturalmente con su sucesora (continuación de temas en el segundo nivel) Informática II.

CRONOGRAMA ESTIMADO DE CLASES

N	Tema	UT	Modalidad de dictado
1	Introducción a la Programación Estructurada Interpretación de enunciados	3	presencial
2	Introducción a la Programación Estructurada Interpretación de enunciados	3	presencial
3	Introducción a la Programación Estructurada Interpretación de enunciados	3	presencial
4	Ejercitación integradora- Evaluación		presencial
5	El soporte del Sistema Operativo Herramientas de desarrollo: El Compilador. El Linker. : Programa fuente, programa objeto, programa ejecutable. Introducción al Lenguaje C: Algoritmos elementales. Concepto de variables	4 , 5	presencial
6	Control de flujo en lenguaje c	8	presencial
7	Sistemas de numeración y representación numérica	2	Virtualidad sincrónica
8	Variables. operadores a nivel de bit y operadores a nivel de variable	13 ,7,6	Virtualidad sincrónica
9	Funciones en lenguaje c	9	presencial
10	Ejercitación integradora- Evaluación		presencial
11	Arreglos en C	10	presencial
12	Arreglos en C- strings	10	presencial
13	Punteros en c	12	presencial
14	Punteros en c - Ejercitación	12	presencial
15	Argumentos del main - Recursividad	10 11	Virtualidad sincrónica
16	Primer parcial		presencial
17	Estructuras y uniones y campos de bit	11,13	presencial
18	Estructuras y uniones y campos de bit-Ejercitación	11,13	presencial
19	Estructuras autorreferenciadas - memoria dinámica	11,12	presencial
20	Sistema computacional por dentro - formalización de: prototipo, llamado a función y desarrollo de la función	1	Virtualidad sincrónica
21	Streams	14	presencial
22	Streams - Ejercitación	14	presencial



23	Ejercitación integradora- Evaluación		presencial
24	Señales	16	Virtualidad sincrónica
25	Multiprocesamiento - hilos	15	presencial
26	Multiprocesamiento - hilos	15	presencial
27	Concepto de redes	16	Virtualidad sincrónica
28	Sockets	16	presencial
29	Sockets	16	presencial
30	Ejercitación integradora - Evaluación		presencial
31	Métodos numéricos	17	presencial
32	Segundo parcial		presencial

BIBLIOGRAFÍA OBLIGATORIA

- [1] DEITEL & DEITEL. (2014) Como programar en C/C++, 9na edición. Ed. Pearson
- [2] KERNIGHAM & RITCHIE (1991). El lenguaje de programación C segunda edición. Prentice-Hall.
- [3] RICHARD STEVENS (2013). Programación Avanzada en el entorno UNIX. 3aª Edición. Ed Pearson

BIBLIOGRAFÍA COMPLEMENTARIA

- [1] B. Hall (2020) "Beej's Guide to Network Programming Using Internet Sockets," p. 134,[http://beej.us/guide/bgnet/pdf/bgnet_usl_c_1.pdf]
- [2] LINDA NULL & JULIA LOBUR (2018). Essentials of Computer Organization and Architecture, fifth edition. Pennsylvania State University. Jones and Bartlett Publishers
- [3] WILLIAM STALLINGS (2006). Organización y Arquitectura de Computadores 7ma.. Edición. Ed. Prentice Hall.
- [4] WILLIAM H. PRESS, SAUL A. TEUKOLSKY, WILLIAM T. VETTERLING, BRIAN P. FLANNERY (2007). Numerical recipes in C, third edition. Cambridge University Press.