

INFORMATICA I

Análisis de la asignatura

1. Análisis de los objetivos

Dentro del campo de acción del ingeniero electrónico hay una importante cantidad de problemas que requieren tener sólidos conocimientos de programación para poder resolverlos. Algunos ejemplos son: aplicaciones basadas en microprocesadores y microcontroladores, procesamiento digital de señales, procesamiento digital de imágenes, sistemas de control en tiempo real, sistemas de comunicación de datos, etc. Incluso cabe mencionar que otras actividades dentro de la ingeniería electrónica, como el diseño de circuitos digitales y el diseño y fabricación de circuitos a partir de componentes electrónicos discretos está siendo reemplazado por la programación de circuitos integrados programables, como por ejemplo las FPGA (Field Programmable Gate Array). Uno de los requerimientos para poder aprender a analizar o desarrollar circuitos electrónicos puede requerir del uso de paquetes o software que incluyan módulos de programación para su utilización (MatLab, LabView, etc.) Estos requieren tener cierta experiencia y habilidad en el uso de lenguajes de programación estructurada y/o orientados a objetos.

Uno de los lenguajes de programación predominantes para las aplicaciones mencionadas es el lenguaje C. Una característica muy importante de la misma es que combina la abstracción de un lenguaje de alto nivel con la facilidad de acceso a los recursos de bajo nivel como periféricos, dispositivos de entrada/salida, y registros entre otros.

Asimismo, como paso previo, es necesario introducir al alumno en el diseño de algoritmos para la resolución de problemas. Se puede citar como ejemplo los algoritmos de distinto grado de complejidad para la solución de problemas típicos de la computación, como son la búsqueda y la ordenación de datos, entre otros. Además, teniendo en cuenta la condición de materia integradora, se analizan problemas basados en conceptos de álgebra, análisis numérico y física.

2. Competencias que desarrolla.

Las competencias que se pretende desarrollar en esta asignatura son:

Competencias genéricas:

- Competencias Sociales:
 - ⤴ Trabajo en equipo.
 - ⤴ Formación del juicio crítico y toma de decisión.

- Competencias metodológicas:
 - ⤴ Buscar, seleccionar y utilizar estratégicamente los recursos disponibles para el estudio.
 - ⤴ Resolver problemas a partir del uso estratégico y heurístico de los saberes construidos.
 - ⤴ Utilizar pensamiento lógico-formal para obtener conclusiones a partir de datos.

- Competencias científico-técnicas:
 - ⤴ Manejar el lenguaje propio de la disciplina para comprender, producir e informar resultados.
 - ⤴ Planificar estrategias para la resolución de problemas a partir de la identificación de los datos, la representación de estos y el establecimiento de relaciones integrando los saberes.
 - ⤴ Identificar y aplicar la información correspondiente a cada situación.
 - ⤴ Utilizar modelos de simulación simples.

- △ Aplicar la creatividad, la iniciativa personal y la innovación en el área tecnológica.

- Competencias específicas:

- △ Manejar herramientas auxiliares para la escritura de programas y su ejecución.
- △ Elaborar estrategias lógicas (programas) para resolver problemas.
- △ Depurar, probar y validar los programas escritos contra las especificaciones originales.

3. Análisis de los contenidos.

Los contenidos de la asignatura están orientados a introducir al alumno en el diseño e implementación de algoritmos para la solución de diversos tipos de problemas. En primer lugar, se introduce la noción de procedimiento computacional y los requerimientos teóricos para su correcta formulación. En esta etapa también se introducen algunas nociones sobre la estructura y funcionamiento de una computadora, bases del modelo computacional imperativo y la programación estructurada.

Otra parte importante de la materia está dedicada a conocer la sintaxis del lenguaje C, elegida como la herramienta a ser utilizada para el desarrollo de programas. Paralelamente se van introduciendo algoritmos de uso frecuente como las de búsqueda, ordenación y los que faciliten el desarrollo de aplicaciones ingenieriles basados en algebra y física.

Como los algoritmos serán implementados mediante programas ejecutados por computadoras digitales, se estudia el importante concepto de la digitalización y la forma de representar información del mundo real en la computadora. Se analiza la codificación binaria para la representación de valores numéricos de punto fijo y de punto flotante, se estudia la representación de caracteres a través de la tabla ASCII y del manejo de cadenas de caracteres como arreglos en la memoria de la computadora. Finalmente se indica como el programador puede obtener sus propias representaciones para diversos conceptos físicos reales a través del uso de tipos definidos por el usuario mediante estructuras.

A lo largo del curso se incorporan conceptos sobre la estructura y manejo de las funciones del sistema operativo para la asignación de memoria dinámica, acceso a archivos secuenciales y binarios. Para ello es importante entender los diferentes modelos de direccionamiento de la Unidad Central de Procesos, en especial aquellos referidos al direccionamiento indirecto que el lenguaje presenta al usuario en la forma de punteros.

Todo programa debe interactuar con el sistema operativo, en especial resulta de interés el manejo de la memoria dinámica y la persistencia de la información a través de archivos secuenciales y de acceso directo (binarios).

Por último, se trabaja sobre las posibilidades que brinda el lenguaje para la manipulación de bits y el acceso a bajo nivel, aspectos de importancia para la especialidad. Estas cuestiones son básicas para el desarrollo de programas para la interacción con el hardware y dispositivos de una computadora y permite introducir conceptos de procesamiento de audio, imágenes y de diferentes periféricos.

Finaliza el dictado de la materia introduciendo el concepto modular de desarrollo de aplicaciones de mayor porte a través de los conceptos de desarrollo parcial de unidades independientes por parte de equipos de desarrolladores.

4. Metodología a emplear en el cursado.

Las actividades que se desarrollan durante el cursado son:

- Clases con exposición por parte del docente. Se fomenta a los alumnos para que formulen preguntas y participen activamente en el desarrollo de las clases, como así también se les efectúa preguntas para motivarlos a pensar en el tema que se está tratando. Si la situación generada por el COVID-19 no lo permite se emplearán técnicas de presentación on-line a través del aula virtual y otros recursos disponibles para clases no presenciales. Basándose en los conceptos de la educación centrado en el alumno se brindará acceso a videos pregrabados realizados por el profesor que permitirán la revisión o aprendizaje de los temas sin la necesidad específica de asistir a clase.

- Guías de problemas, para que los alumnos profundicen conceptos y estrategias de resolución de problemas.

- Laboratorios de programación presenciales, en la sala de cómputos, cuyas finalidades son:

- △ ejercitarse en algún aspecto puntual de la programación
- △ resolver una problemática relativamente compleja, integrando conocimientos y habilidades, tanto de esta asignatura como también de otras.
- △ Realizar trabajos en grupo.

De no poder realizarse en forma presencial se desarrollarán en forma virtual a través del uso de software de videoconferencia grupal en los horarios establecidos.

-En la medida de lo posible se harán prácticas de introducción a los microprocesadores del tipo Arduino o Launch Pad que se programan en C conectándolos a dispositivos como LEDs, sensores de temperatura, y otros dispositivos de bajo costo. Cabe señalar que estas placas son de bajo costo, fácilmente adquiribles por el alumno, y con muchas páginas en Internet con proyectos hogareños de fácil ejecución. Estas experiencias, sin evaluación formal, se usarán como motivadoras para iniciar a los alumnos al desarrollo electrónico y para ilustrar las aplicaciones prácticas de los temas vistos en clase.

5. Técnicas de evaluación.

Por tratarse de una materia integradora, está previsto que haya una evaluación permanente de la realización de los trabajos prácticos de laboratorio. Se pretende evaluar:

- △ la adquisición de habilidades para el diseño de algoritmos y el desarrollo de programas,
- △ el orden y la metodología utilizada para la codificación
- △ el dominio de los elementos del lenguaje C

Durante el dictado de la materia se tomarán 4 instancias de evaluación teórico-prácticos y aproximadamente 20 actividades de formación práctica (laboratorios en el centro de cómputos – virtuales si lo impide la pandemia). Cada instancia de evaluación está compuesta por un parcial y su correspondiente recuperatorio. Los laboratorios no poseen recuperatorio, pero se dejará la actividad publicada y se responderán consultas si el alumno intenta resolverlo por cuenta propia.

Los parciales y recuperatorios se evaluarán con puntuación numérica asignada en forma lineal. Se lo considera aprobado si el alumno reúne el 60% del puntaje total de la evaluación. La nota del recuperatorio reemplaza a la nota del parcial desaprobado para los fines del promedio. Las actividades de formación práctica se evalúan como “aprobado”, “desaprobado” o, “ausente”.

La aprobación directa de la materia se obtiene cuando el alumno apruebe las cuatro instancias de evaluación (parcial o recuperatorio), tenga con un 60% de las actividades de formación práctica aprobadas (laboratorios), y cumpla con el porcentaje de asistencia exigido por la Universidad Tecnológica Nacional en sus estatutos. La nota final será la obtenida por promedio de las 4 notas de las instancias de evaluación.

El cursado (condición regular) de la materia se obtiene cuando el alumno aprueba 3 de las instancias de evaluación teórica o práctica con el 60%, tenga con un 60% de las actividades de formación práctica aprobadas (laboratorios), y cumpla con el porcentaje de asistencia exigido por la Universidad Tecnológica Nacional en sus estatutos. En este caso debe rendir un examen final en forma oral, escrito o en computadora en el que se evalúa el dominio de conceptos y el grado de competencia teórico-práctico exigido por la materia. Estos exámenes finales se rinden en las fechas predeterminadas por la UTN en el calendario anual.

Se considera que un alumno NO ha alcanzado los niveles mínimos y básicos de aprendizaje (condición LIBRE) cuando no logra dar cumplimiento al 60% de las actividades de formación práctica, o desaprueba el 50% de los exámenes teóricos-prácticos (2 parciales y sus recuperatorios).

Un alumno ABANDONA la materia si se encuentra AUSENTE en 2 o más parciales

Las actividades presenciales descriptas en este punto están sujetas a las restricciones generadas por el virus COVID-19. De no poder realizarse en forma presencial se los hará en forma virtual a través del aula virtual.

6. Análisis sobre la articulación horizontal y vertical, teniendo en cuenta el área, el régimen de correlatividades y el alcance del título establecidos en el diseño curricular.

Esta asignatura está articulada en forma vertical con Informática II, y en tal sentido hay una continuidad en los contenidos desarrollados por ambas materias. En Informática I se introduce al alumno en el mundo de los algoritmos y la programación, se lo familiariza con el lenguaje de programación C y su aplicación a la resolución de distintos problemas, algunos afines a otras materias que cursa en paralelo. Posteriormente, en Informática II se estudia con más profundidad el lenguaje C para aplicarlo a la programación avanzada. Asimismo, en Informática II se trabaja sobre algoritmos más complejos y se introducen los conceptos de programación orientada a objetos en C++.

En cuanto a la integración horizontal, la misma se realiza particularmente en los ejemplos que se dan en clase como así también en los trabajos de laboratorio, debido a que se utiliza a la programación para resolver problemas que provienen de otras disciplinas (p. ej.: física, y matemática) Se da especial énfasis en problemas que requieren algebra, geometría y análisis para resolver problemas de física e ingeniería. En estos casos no solamente es necesario saber programar, sino interpretar el problema y saber plantear una solución.

Esta asignatura guarda una gran relación con el alcance del título, ya que dentro de las actividades reservadas al título de ingeniero electrónico en el libro Rojo de CONFEDI figura lo siguiente (Anexo 11 – Actividades reservadas del Ing. Electrónico)

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.

Además, impacta en todas las competencias específicas que incluyen tareas de diseño de sistemas de control, procesamiento y de comunicaciones señaladas en dicho documento ya que en la ingeniería moderna estas tareas se desarrollan con algún tipo de desarrollo de software o de programación.

Mg. Ricardo J. Coppo
Profesor Adjunto Ordinario