

**PLAN DE ESTUDIOS:** 2004 Ajuste 2011  
**AÑO ACADÉMICO:** 2013  
**CARRERA:** Ingeniería Electrónica

## **1. Objetivos**

Esta materia constituye la primera que trata de la programación de computadoras; se dicta luego de la materia **Lógica** y sirve de base para la materia siguiente **Programación II**.

Por lo tanto, utilizando los conceptos de **Lógica**, debe preparar al alumno para que pueda afrontar los problemas de aplicación que se tratarán en la materia **Programación II**.

En este sentido, al finalizar exitosamente esta materia, el alumno debe ser capaz de:

- Explicar y generar algoritmos para la resolución de problemas.
- Usar y adaptar algoritmos conocidos.
- Diseñar algoritmos correctos y eficientes.
- Aplicar las técnicas de diseño modular para la descomposición de problemas.
- Conocer un lenguaje de programación<sup>1</sup> con cierta profundidad, de manera de aplicarlo eficientemente a la implementación de los algoritmos.
- Conocer las herramientas de generación de programas, logrando una razonable eficiencia para poder expresar la solución de los algoritmos y proceder a la verificación de los mismos.
- Conocer el funcionamiento de los programas que resuelven dichos algoritmos en el marco de un dado Sistema Operativo<sup>2</sup>, con una razonable interfase con el mismo.
- Desarrollar un estilo de programación que permita la expresión clara de los algoritmos involucrados.
- Desarrollar la solución de problemas en ambientes multiplataforma.

---

<sup>1</sup> Se usará el lenguaje C por su importancia para la Ingeniería Electrónica

<sup>2</sup> Se usará un Sistema Operativo tipo Unix.

## 2. Contenidos

- **Unidad 1:** Historia de la computación. Representación de los datos en una Computadora. Arquitectura Harvard & Von Neuman. Unidad Central de proceso. Unidad Aritmético-Lógica. Unidad de Memoria. Instrucciones, tipos. Clases de direccionamiento y formato de los datos.
- **Unidad 2:** Introducción a la Programación. Programación modularizada y estructurada. Buen estilo de programación. Concepto de Algoritmo. Pseudocódigo y refinamiento. Lenguajes de alto nivel. Compiladores y preprocesadores. Prueba de Software: caja negra, caja blanca y caja de Pandora.
- **Unidad 3:** Tipos, Operadores y Expresiones. Tipos y tamaño de datos. Constantes. Declaraciones. Operadores: aritméticos, relacionales, lógicos, de incremento y decremento, para manejo de bits y de conversión. Operadores de asignación y expresiones. Precedencia de operadores. Instrucciones básicas de E/S: *printf* y *getchar*. Redireccionamiento de la entrada y de la salida.
- **Unidad 4:** Estructuras de Control. Preposiciones y bloques. Almacenamiento y alcance de variables. Variables externas y estáticas. Estructuras de decisión: *if-else* y *switch*. Estructuras de repetición: *while*, *for* y *do-while*. Uso de *break* y *continue*. Instrucción *goto*.
- **Unidad 5:** Funciones. Concepto de Función. Prototipación. Parámetros de entrada. Biblioteca estándar de C (funciones básicas). Archivos de encabezamiento. El preprocesador de C. Concepto básico de puntero. Parámetros de entrada y salida. Entrada formateada: *scanf* y *sscanf*. Construcción de Bibliotecas. Funciones con argumentos variables. Recursividad.
- **Unidad 6:** Arreglos. Arreglos unidimensionales. Arreglos de caracteres. Ordenamiento de arreglos. Búsqueda en arreglos unidimensionales. Arreglos multidimensionales.
- **Unidad 7:** Temas avanzados del lenguaje C. Punteros. Aritmética de punteros. Punteros a caracteres. Estructuras. Uniones. Uso del Heap.

### 3. Bibliografía

#### 3.1. **Básica**

- 3.1.1. *El Lenguaje de Programación C – Segunda Edición*  
Brian Kernighan – Dennis M. Ritchie  
Prentice Hall Hispanoamericana
- 3.1.2. *The vim book*  
<ftp://ftp.vim.org/pub/vim/doc/book/vimbook-OPL.pdf>
- 3.1.3. *An introduction to the make utility*  
[www.mtsu.edu/~csdept/FacilitiesAndResources/make.htm](http://www.mtsu.edu/~csdept/FacilitiesAndResources/make.htm)
- 3.1.4. *gcc tutorial*  
<http://cs.acadiau.ca/~jdiamond/comp2103/beginner-tutorials/LinuxTutorialGcc.html>
- 3.1.5. *Vínculos de Internet*

#### 3.2. **Adicional**

- 3.2.1. *El Entorno de Programación Unix*  
Brian Kernighan – Rob Pike  
Prentice Hall Hispanoamericana

### 4. Metodología de la enseñanza

Para que el alumno, al completar exitosamente la materia, cumpla los objetivos establecidos, la enseñanza estará establecida sobre las siguientes actividades, las cuales están íntimamente relacionadas.

- Clases teórico-prácticas
- Prácticas de resolución de problemas.
- Prácticas de programación en computadora
- Prácticas de diseño y proyecto.

#### 4.1. **Clases teóricas o teórico-prácticas.**

Las clases son del tipo teórico-prácticas, donde el docente a cargo desarrolla los temas, ejemplificando el concepto y la aplicación de los mismos en forma interactiva con el alumno. En dichas clases, se utilizan los mismos elementos instrumentales que el alumno, luego, deberá utilizar en las actividades prácticas.

## **4.2. Actividades Prácticas**

El objetivo de las actividades prácticas es permitir que el alumno se enfrente con los temas de la materia, resolviendo problemas de complejidad creciente en cada uno de los temas que se ataquen y luego de haber concurrido y participado a la respectiva clase teórica-práctica. De esta manera, el alumno puede realmente verificar sus conceptos, afianzarlos y colocarlos en la justa dimensión de la actividad del área.

Por la característica de la materia, las prácticas versan, no solamente sobre los temas específicos de la materia, sino también sobre las herramientas necesarias a conocer y utilizar para obtener una razonable eficiencia en la construcción de programas.

### **4.2.1. Prácticas de resolución de problemas**

El objetivo de estas prácticas es que el alumno, luego de la clase teórico-práctica sobre un tema, pueda realizar las primeras aplicaciones de los conceptos fundamentales del tema y obtenga, antes de realizar la práctica de programación correspondiente, la verificación de sus conceptos.

En estas prácticas, se resuelven fundamentalmente los algoritmos que luego se llevarán a la práctica en las Prácticas de programación en computadora,

En la misma, se realizarán los siguientes pasos de resolución

- ❖ Análisis del problema
- ❖ Planteo de solución en papel
- ❖ Escritura de un programa que se ajuste a la solución en papel.
- ❖ Verificación del programa en papel (aplicando métodos de prueba)

### **4.2.2. Prácticas de programación en computadora**

En estas prácticas los alumnos aplican los conocimientos adquiridos a través de la resolución de los ejercicios presentes en las guías de trabajos prácticos. Dichos ejercicios están especialmente pensados para ayudar al aprendizaje de los puntos importantes en la materia, con una complejidad creciente.

Sobre los problemas planteados y luego de haber seguido los pasos correspondientes en las Prácticas de resolución de problemas, el alumno procederá a editar, compilar y verificar el funcionamiento.

No sólo se tratan los temas conceptuales de la materia en forma práctica, sino que también el alumno debe desarrollar una eficiencia en el manejo de las herramientas que contribuyen al desarrollo de programas.

Los docentes están a disposición de los alumnos para resolver las dudas que surgen en este proceso y al finalizar la clase pueden solicitar la entrega de ejercicios indicados al comienzo de la misma, a modo de seguimiento. La entrega de dichos ejercicios es optativa y se calificará con una nota de concepto.

El alumno puede consultar en la semana en la cual se desarrolla cada guía de trabajos prácticos. Dada la extensión de las mismas se entiende que el alumno **debe completarlas fuera del horario de clase.**

#### **4.2.3. Prácticas de diseño y proyecto**

Si bien esta es una materia inicial sobre la programación de computadoras, hacia el final de la materia se lo enfrentará al alumno a pequeños problemas de diseño y proyecto, de manera de prepararlo para las sucesivas materias (en especial el caso de Programación II).

### **5. Evaluación.**

Para estar en condiciones de dar el examen final, el alumno debe:

- ❖ Cumplir con la condición de presentismo fijada y verificada por la Universidad.
- ❖ Aprobar un parcial (recuperable) de acuerdo a las condiciones fijadas por la Universidad
- ❖ Aprobar el curso de Trabajos Prácticos

La evaluación del curso de Trabajos Prácticos se realiza a través de:

- ❖ el **desempeño del alumno** en las clases prácticas y de laboratorio, el cual será verificado en forma continuada y generará el **concepto de desempeño.**
- ❖ el resultado de los informes de cada uno de los trabajos prácticos

#### **5.1. Aprobación del curso práctico de la materia**

Para poder aprobar el curso práctico de la materia, se deberán cumplir las siguientes condiciones:

- Tener como nota de **concepto de desempeño 4** (cuatro) como mínimo
- Haber entregado en tiempo y forma y haber aprobado, aproximadamente, el 60% de los Informes de Trabajos Prácticos dentro del curso de la asignatura.
  - El eventual 40% de los Trabajos Prácticos que el alumno no haya presentado o no haya aprobado, podrán ser recuperados en las oportunidades y los plazos que la Universidad fije para los mismos.
  - En cada año lectivo en que se dicte la asignatura y al comienzo de clases, se especificará cuántos Trabajos Prácticos definen el 60% antes aludido