



## **Programa de NOMBRE DE LA UNIDAD CURRICULAR**

### **1. NOMBRE DE LA UNIDAD CURRICULAR**

Programación para Ingeniería Eléctrica

### **2. CRÉDITOS : 7**

### **3. OBJETIVOS DE LA UNIDAD CURRICULAR**

Fortalecer y complementar la formación en programación de los egresados de la carrera de ingeniería eléctrica en todos sus perfiles, con especial énfasis en las habilidades particulares requeridas en ingeniería eléctrica. En particular, se hará foco en el paradigma de programación estructurada aplicado al desarrollo de programas de pequeño y mediano porte, y para sistemas embebidos.

Se utilizarán ejemplos de aplicación típicos de la Ingeniería Eléctrica, buscando una vinculación entre ese campo y la programación.

Habilidades adquiridas

- Conocimientos básicos sobre uso y programación de computadoras y procesadores programables en general.
- Desarrollar una metodología de implementación y prueba de programas.
- Implementación de algoritmos mediante programación estructurada.

### **4. METODOLOGÍA DE ENSEÑANZA**

Se dictarán 2 horas semanales de teórico durante todo el semestre. Durante las primeras 8 semanas y en paralelo con el teórico se dictarán 2 horas de práctico. Esto será complementado por el desarrollo de un proyecto obligatorio de programación de mediano porte, entregado en varias etapas durante el curso.

Se buscará que el alumno aprenda, ejercite y profundice sobre aquellos aspectos y técnicas de la programación que resultan de particular importancia en el ámbito de la ingeniería eléctrica. Ejemplos de esto son el manejo eficiente de recursos (memoria, operaciones), programación de flujos de datos, y aplicaciones específicas de la ingeniería eléctrica como por ejemplo el tratamiento de señales, la robótica, las telecomunicaciones. Teniendo en cuenta esta especificidad este curso se basa en el lenguaje de programación C.

En la parte práctica se propondrán pequeñas rutinas a implementar por los estudiantes.

El trabajo obligatorio consistirá en un proyecto a desarrollar de manera incremental a lo largo del curso (usando herramientas de dominio público), en la medida que los estudiantes adquieran las distintas habilidades necesarias para completar cada etapa. El trabajo será supervisado por los docentes y evaluado en entregas parciales con fecha límite predeterminada.

Tanto en la parte práctica como en el trabajo obligatorio se hará énfasis en problemas relacionados con la ingeniería eléctrica, como ser (y sin limitarse a) la implementación de filtros digitales, modulación/demodulación, control y procesamiento de señales por ejemplo audio, imágenes, o señales biológicas.

## 5. TEMARIO

- Introducción
- Repaso de programación estructurada. Primer programa en C. Sintaxis básica.
- Conceptos Básicos de C. Tipos básicos. Operadores y expresiones relevantes (e.g. operadores de bits). Particularidades del control de flujo en C.
- Funciones (declaración e implementación), pasaje de parámetros.
- Modularización y encapsulamiento
- Preprocesador. Compilación, Linkeditor, etc.
- Programación modularizada. Makefile, generación bibliotecas. Recursividad
- Punteros. Relación entre punteros y arreglos. Caso particular de cadenas de caracteres. Pasaje por referencia
- Debug: gdb y nemiver.
- Tipos avanzados (struct, union).
- Estructuras autoreferenciadas. Por ejemplo Listas, arboles de decisión, etc.
- Entrada y Salida
- Manejo de memoria. Eficiencia. Profiling. Valgrind.

## 6. BIBLIOGRAFÍA

Identificación de las publicaciones básicas y complementarias adecuadas para el buen seguimiento del curso. Se debería observar la disponibilidad de estos textos, tanto en la Biblioteca de Facultad como en el mercado. En caso de existir varios textos principales, indicar para qué tema aporta cada uno. La referencia bibliográfica deberá darse de la siguiente forma:

Tema	Básica (1)
Repaso de programación estructurada. Primer programa en C. Sintaxis básica.	Cap 1
Conceptos Básicos de C. Tipos básicos. Operadores y expresiones relevantes (e.g. operadores de bits). Particularidades del control de flujo en C.	Cap. 2 y 3
Funciones (declaración e implementación), pasaje de parámetros. Modularización y encapsulamiento Preprocesador. Compilación, Linkeditor, etc.	Cap. 4
Programación modularizada. Makefile, generación bibliotecas. Recursividad	Cap 4
Punteros. Relación entre punteros y arreglos. Caso particular de cadenas de caracteres. Pasaje por referencia	Cap. 5
Debug: gdb y nemiver.	
Tipos avanzados (struct, union).	Cap. 6
Estructuras autoreferenciadas. Por ejemplo Listas, arboles de decisión, etc.	Cap. 6
Entrada y Salida	Cap. 7
Manejo de memoria. Eficiencia. Profiling. Valgrind.	

### 6.1 Básica

1. A. Kernighan, A. y Ritchie, D. (1988) "The C Programming Language". Segunda edición. Prentice Hall. ISBN 0-13-110362-8. 1988.

### 6.2 Complementaria

1. H. M. Deitel, H. M. y Deitel, P. J. (1998) "Como Programar en C/C++". Prentice Hall. ISBN: 9688804711. 2da Edición. Feb. 1998.
2. ISO/IEC. "The C99 Standard".  
<http://www.open-std.org/jtc1/sc22/WG14/www/docs/n1256.pdf>

## 7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS

### 7.1 Conocimientos Previos Exigidos:

### 7.2 Conocimientos Previos Recomendados:

- Programación estructurada
- Nociones de programación científica
- Cálculo Numérico

## ANEXO A

### Para todas las Carreras

Esta primera parte del anexo incluye aspectos complementarios que son generales de la unidad curricular.

#### A1) INSTITUTO

Instituto de Ingeniería Eléctrica

#### A2) CRONOGRAMA TENTATIVO

Semana 1	Introducción Repaso de programación estructurada. Primer programa en C. Sintaxis básica.
Semana 2	Conceptos Básicos de C. Tipos básicos. Operadores y expresiones relevantes (e.g. operadores de bits). Particularidades del control de flujo en C.
Semana 3	Funciones (declaración e implementación), pasaje de parámetros. Modularización y encapsulamiento Preprocesador. Compilación, Linkeditor, etc.
Semana 4	Programación modularizada. Makefile, generación bibliotecas. Recursividad
Semana 5	Punteros. Relación entre punteros y arreglos. Caso particular de cadenas de caracteres. Pasaje por referencia
Semana 6	Debug: gdb y nemiver. Tipos avanzados (struct, union).
Semana 7	Estructuras autoreferenciadas. Por ejemplo Listas, arboles de decisión, etc. Entrada y Salida
Semana 8	Manejo de memoria. Eficiencia. Profiling. Valgrind.
Semana 9	Obligatorio
Semana 10	Obligatorio
Semana 11	Obligatorio
Semana 12	Obligatorio
Semana 13	Obligatorio
Semana 14	Obligatorio
Semana 15	Obligatorio

#### A3) MODALIDAD DEL CURSO Y PROCEDIMIENTO DE EVALUACIÓN

El curso se aprobará por la aprobación de dos exámenes teórico parciales y la entrega de dos o tres obligatorios (depende del año), que pueden ser vistos como tres partes de un obligatorio general.

Para lograr la aprobación se deberá obtener un 60% de los puntos totales del curso entre las dos pruebas y las entregas obligatorias, y un mínimo del 25% en cada una de las cuatro o cinco partes (en cada parcial y cada entrega de obligatorios). Las pruebas parciales evaluarán conceptos teóricos relacionados con los obligatorios.

En caso de no aprobar la asignatura, deberá recursarse de manera integra.

#### **A4) CALIDAD DE LIBRE**

No se contempla calidad de libre en esta asignatura.

#### **A5) CUPOS DE LA UNIDAD CURRICULAR**

No tiene.

### **ANEXO B para la carrera de Ingeniería Eléctrica**

#### **B1) ÁREA DE FORMACIÓN**

Informática

#### **B2) UNIDADES CURRICULARES PREVIAS**

Curso: Programación 1

Examen: Curso.