

# UNIVERSIDAD NACIONAL DE ROSARIO

## INSTITUTO POLITECNICO SUPERIOR

### “GRAL SAN MARTIN”

**PROGRAMA ANALITICO DEL ESPACIO CURRICULAR:** Informática IV  
**CURSO** : Cuarto Año.

**PLAN DE ESTUDIOS:**  
*Educación Técnico-Profesional de Nivel Secundario.*

**CARRERA:**  
*Técnico Mecánico*

**DEPARTAMENTO:**  
*Informática*

**VIGENCIA AÑO:** 2013

**CANTIDAD DE HORAS CATEDRA**  
**SEMANALES:** 04 hs

**PLAN DE ESTUDIOS RESOLUCION C.S. N°: 0237/10**

**RESOLUCION MINISTERIO DE EDUCACION N°: ...**

#### OBJETIVOS GENERALES

Al finalizar esta materia los estudiantes estarán en condiciones de encarar estrategias de resolución de problemas y resolver pequeños problemas de programación, esencialmente de carácter didáctico mostrando conocimientos en:

- El dominio de las estructuras de control y tipos de datos elementales.
- La utilización de ambientes de programación imperativa y herramientas de depuración de programas.

Esta materia introduce al estudiante en los esquemas de razonamiento necesarios para incorporar rudimentos de programación, que le faciliten la interpretación del funcionamiento del *software*.

Las capacidades planteadas por esta materia se vinculan con la optimización del ambiente informático de trabajo del usuario y el desarrollo de programas, o la adaptación y el complementar sus funcionalidades, utilizando las herramientas puestas a disposición de los usuarios.

#### FUNDAMENTOS

La problemática abordada por esta materia se relaciona con los conocimientos relativos al desarrollo de programas.

La resolución automática de un problema obliga a analizar previamente y en forma exhaustiva las diferentes situaciones y condiciones que puede presentar. La forma que adopta la solución es un algoritmo que computa la función pretendida y que, por su complejidad, debe ser verificado metódicamente para asegurar su corrección y validez.

El instrumento utilizado para programar es un lenguaje que tiene características de los lenguajes formales y que, de acuerdo al tipo de problema que intenta representar y las estrategias en que se basa, tiene estructuras,

reglas, operaciones y objetos propios.

Programar es una actividad compleja que combina procesos teóricos (propios de la matemática), de abstracción o experimentales (típicamente de la ciencia) y de diseño (de la ingeniería).

Esta materia, está orientada a desarrollar un esquema de razonamiento lógico apropiado no sólo para las actividades de programación sino que sirva de base para la comprensión de los objetos del trabajo y la capacidad profesional de abstracción requerida del técnico.

## **CONTENIDOS**

### **UNIDAD 1:**

- 1.1. Programas, estructura funcional genérica.
- 1.2. Operaciones y sentencias. Líneas y bloques de sentencias.
- 1.3. Funciones de entrada y salida.

### **UNIDAD 2:**

- 2.1. Diseño de programas. Programación estructurada.
- 2.2. Método Top-Down o de Refinamiento Progresivo.
- 2.3. Comentarios de una línea y bloques de comentarios.
- 2.4. Procedimientos y funciones.

### **UNIDAD 3:**

- 3.1. Proceso de construcción de un programa. Código fuente. Código objeto. Código ejecutable.
- 3.2. Editor.
- 3.3. Compilador. Enlazador (Linker).
- 3.4. Depuración. Errores sintácticos, lógicos y funcionales.

### **UNIDAD 4:**

- 4.1. Estructura genérica de un programa (en lenguaje C) basado en funciones
  - 4.1.1. Comentarios.
  - 4.1.2. Directivas al compilador
  - 4.1.3. Declaración y definición de funciones
  - 4.1.4. Declaración y definición de constantes y variables
  - 4.1.5. Ingreso de datos, procesamiento, impresión de resultados
  - 4.1.6. Funciones de entrada y salida de datos con formato (scanf() y printf() en lenguaje C).

### **UNIDAD 5:**

- 5.1. Operadores aritméticos, orden de prelación.
- 5.2. Expresiones aritméticas, evaluación y asignación de resultados (en C).
- 5.3. Operadores de relación y lógicos, orden de prelación.
- 5.4. Expresiones lógicas-relacionales.
- 5.5. Evaluación del resultado, valores posibles.

### **UNIDAD 6:**

- 6.1. Toma de decisiones simple, bifurcación. Estructura if()... else.
- 6.2. Toma de decisiones múltiples. Estructura switch()...case...default.
- 6.3. Control de iteraciones. Estructuras while() y for().
- 6.4. Ruptura de la secuencia natural de en selecciones múltiples e iteraciones. Propositiones break y continue.

### **UNIDAD 7:**

- 7.1. Índices, contadores y acumuladores.
- 7.2. Algoritmos de búsqueda de menor, mayor y dicotómica.
- 7.3. Algoritmo de cálculo de promedio.

### **UNIDAD 8:**

- 8.1.Organización de datos en arreglos, dimensiones.
  - 8.1.1. Arreglo de variables.
  - 8.1.2. Concepto de puntero a variables y arreglos.
- 8.2. Cadenas de caracteres.
  - 8.2.1. El tipo string como arreglo de caracteres.
  - 8.2.2. Manejo de cadenas a través de funciones.
  - 8.2.3. Funciones de biblioteca.
- 8.3.Arreglos bidimensionales.
  - 8.3.1. Arreglos de cadenas como matrices de caracteres.
- 8.4.Tablas de datos.
  - 8.4.1. Arreglos vinculados.
  - 8.4.2. Estructura de datos, conceptos básicos.
  - 8.4.3. Arreglos de estructuras.

#### UNIDAD 9:

- 9.1.Funciones, conceptos, declaración, prototipo, definición.
- 9.2.Argumentos reales y formales, valor y tipo de retorno.
- 9.3.Variables externas e internas.
  - 9.3.1. Alcance de las variables.
  - 9.3.2. Variables estáticas.
  - 9.3.3. Inicialización de variables.
- 9.4.Comunicación de datos entre funciones.
- 9.5.Argumentos por valor y por referencia (punteros).
- 9.6.Archivo de encabezamiento (header).
- 9.7.Organización de funciones en archivos de encabezamiento.
- 9.8.Archivos, conceptos, modos de acceso.

#### UNIDAD 10:

- 10.1. Algoritmos de ordenamiento
  - 10.1.1. Selección
  - 10.1.2. Burbuja
  - 10.1.3. Inserción
  - 10.1.4. Shell
  - 10.1.5. Quicksort.
- 10.2. Algoritmos de búsqueda de ceros de polinomios
- 10.3. Algoritmos para resolución de sistemas de ecuaciones
  - 10.3.1. método de los determinantes
  - 10.3.2. método de Gauss.
- 10.4. Algoritmos para cálculo de integrales definidas
  - 10.4.1. métodos de los trapecios
  - 10.4.2. Simpson y Romberg.

### ACTIVIDAD PRÁCTICA

La práctica desarrollada a lo largo de todo el curso, tiene por objeto afianzar los nuevos conocimientos relativos a los contenidos unidad por unidad. Dependiendo de los temas abordados puede consistir en trabajos individuales o colectivos, de los siguientes tipos: a) trabajos de investigación a cargo de los alumnos con ejecución de informes, b) talleres de práctica y entrenamiento en clase, c) desarrollo de ejercitación en clase, d) desarrollo de ejercitación por internet utilizando tics del campus virtual de la unr, e) visitas a lugares de interés para algunos de los puntos desarrollados.

Para desarrollar las capacidades que la materia plantea se recomienda organizar actividades formativas tales como:



Planificar, desarrollar y depurar programas que reflejen problemas extraídos de la matemática y

de la vida real.



Buscar, interpretar y relacionar información sobre programas, comandos y estructuras de datos.



Analizar las ventajas y desventajas de distintas formas de representación y de estructuras de datos, según las especificaciones previas.



Poner en común lo desarrollado para abstraer aspectos comunes y señalar diferencias.

## METODOLOGÍA

En principio las opciones metodológicas para desarrollar los contenidos son facultades del docente y generalmente derivan de sus capacidades y recursos para ejercer el arte de la enseñanza en este tema. Sin embargo conviene establecer algunas pautas generales como recomendación, que son:

- Propiciar el aprendizaje a través del dialogo, la discusión y la resolución de problemas,
- En el desarrollo de las clases, utilizar los recursos comunicacionales orales, escritos y audiovisuales más eficaces que se tengan en el momento de desarrollar el curso,
- Promover acciones y trabajos individuales y/o grupales que motiven la investigación y faciliten la lectura comprensiva y crítica de textos y todo otro material de apoyo para el estudio (como por ejemplo películas, publicaciones, publicidades, etc.),
- Generar el intercambio de opiniones desde diferentes puntos de vista, para practicar el análisis reflexivo que lleve al alumno a fijar conceptos, ensayar y profundizar interpretaciones y tomar posiciones fundamentadas, en búsqueda de consensos grupales,
- Promover la práctica de la escritura técnica precisa en la ejecución de trabajos en los cuales se deba abordar el asunto de redactar con coherencia el pensamiento en forma válida para la comunicación con terceras partes, perfeccionando en la medida posible la expresión de las ideas y sus fundamentos.

## EVALUACIÓN.

La evaluación de esta materia es un proceso que debe desarrollar el docente a través de todos los actos de aprendizaje y entrenamiento en que participa el alumno. En tal sentido deberán tenerse en cuenta no sólo las pruebas que reglamentariamente proponga la institución, sino también la interacción y participación de cada sujeto con la clase, así como el desempeño en los trabajos individuales y grupales, ya sean obligatorios u optativos que se elaboren durante el año y que deberían recorrer todas las vertientes de presentación/exposición que se consideren procedentes para cada tema.

## Bibliografía:

Kernighan; Ritchie. Lenguaje de programación C.Ed. Pearson

Oualline, Steve. Practical C Programming. 3ra. Edición. Ed. O'Reilly.

### Observación sobre la bibliografía:

*Sólo se ha consignado una posible bibliografía de soporte inicial para el desarrollo de la asignatura. Es necesario hacer notar que en este tema, los conceptos y las prácticas evolucionan tan rápidamente y son tan cambiantes que los más adecuados para cada tema dependen totalmente del momento en que se dicte/course la materia; incluso parte de la bibliografía consignada tiene partes que resultan inaplicables y otras que sirven de apoyo al desarrollo de temas puntuales. Por lo tanto será el docente quien deberá proponer con que recursos bibliográficos (soporte papel o digital), software(s) y posibles prestaciones de sitios de Internet será conveniente desarrollar el curso.*