



ALGORITMOS Y ESTRUCTURAS DE DATOS

PROGRAMA ANALÍTICO

| | |
|-------------------------|------|
| PLAN DE ESTUDIOS | 2008 |
| ORDENANZA CSU. N° | 1150 |
| HORAS/AÑO: | 160 |
| OBLIGATORIA | X |
| ELECTIVA | |
| ANUAL | X |
| PRIMER CUATRIMESTRE | |
| SEGUNDO CUATRIMESTRE | |
| NIVEL / AÑO | 1° |
| HORAS CÁTEDRA SEMANALES | 5 |

OBJETIVOS

OBJETIVO GENERAL

Esta asignatura constituye una introducción al proceso de Diseño e Implementación de Software, particularmente a la disciplina PROGRAMACIÓN, en lo que respecta a especificación y expresión de algoritmos, analizando su corrección y eficiencia, combinados con el tratamiento de las estructuras de datos básicas (simples y compuestas) de los lenguajes de programación.

El paradigma de programación adoptado es el Imperativo - procedural, con énfasis en la modularización y la programación estructurada.

El enfoque aplicado es combinar el tratamiento algorítmico con la elección de las estructuras de datos, contemplando aspectos de documentación y reusabilidad.

Se utiliza pseudocódigo para desarrollo en papel y se emplea el lenguaje de programación Turbo/Borland Pascal 7.0 en taller de programación (presentándose las características básicas).

OBJETIVOS ESPECÍFICOS

OBJETIVOS DE LA UT I. CONCEPTOS BÁSICOS – ALGORITMOS Y DATOS

Valorar la importancia de la correcta expresión oral y escrita, bases ciertas de la comunicación necesaria en el camino de la especificación. Aprender a escuchar y debatir ideas en forma positiva tendiendo siempre a la resolución de un problema concreto.

Comprender y emplear conceptos básicos de algorítmica. Comprender y reconocer los elementos básicos de especificaciones de algoritmos (paradigma imperativo). Especificar algoritmos. Preparar al estudiante brindándole una visión amplia de las herramientas y métodos más usuales para la solución de problemas. Introducir al alumno en el diseño e

COPIA FIEL DEL ORIGINAL

MARIA EUGENIA LAVORATTO
DIRECTORA
DIRECCIÓN ACADÉMICA



implementación de algoritmos con las técnicas y métodos más usados: desarrollo de habilidades en el uso de las técnicas diseño de algoritmos computacionales. Adquirir destreza en la utilización de tipos de datos y estructuras de control.

OBJETIVOS DE LA UT II. ABSTRACCIÓN PROCEDURAL

Comprender el significado de la abstracción procedural. Adquirir destreza en el diseño descendente y en el uso de la recursión. Aprender a discernir entre el uso de procedimientos y funciones. Afianzar el uso de parámetros como metodología de comunicación entre módulos. Introducir al alumno en el análisis de la eficiencia de algoritmos iterativos y recursivos.

OBJETIVOS DE LA UT III. ESTRUCTURAS DE DATOS COMPUESTAS

Utilizar diferentes estructuras de datos compuestas: arreglos, registros, archivos, listas, pilas y colas. Analizar y ejercitar las operaciones que sobre las estructuras de datos se pueden realizar. Resolver problemas de aplicación de complejidad gradualmente creciente con las herramientas disponibles.

OBJETIVOS DE LA UT IV. ANÁLISIS DE ALGORITMOS - COMPLEJIDAD COMPUTACIONAL

Introducir al alumno en el análisis de la complejidad de los algoritmos. Conocer principios para verificar corrección de algoritmos, determinar y analizar su complejidad. Desarrollar habilidades en el uso de las técnicas de análisis de algoritmos computacionales. Emplear las herramientas vistas para realizar análisis comparativos entre diferentes algoritmos.

OBJETIVOS DE LA UT V. LENGUAJES DE PROGRAMACIÓN

Realizar una introducción a un lenguaje de programación. Comprender y experimentar que la formulación de una solución real mediante el empleo de una computadora deberá respetar las reglas impuestas por el lenguaje (sintaxis y semántica) y los equipos a utilizar. Lograr que el alumno conozca y utilice un lenguaje de programación estructurado y se familiarice con un ambiente de programación. Permitir que el alumno pruebe sus soluciones propuestas, las optimice, de manera tal que pueda reconocer los beneficios de la programación modular, re-usar código y afianzar el proceso de documentación.

CONTENIDOS

CONTENIDOS SINTÉTICOS

- Concepto de Dato.
- Tipos de Datos Simples.
- Tipo Abstracto de datos.
- Estructuras de Control Básicas: secuencial, condicional, cíclica.
- Estrategias de Resolución.
- Estructuras de Datos: registros, arreglos y archivos.
- Abstracciones con procedimientos y funciones.
- Pasaje de Parámetros.
- Estructuras de Datos lineales (Pilas-Colas).
- Algoritmos de Búsqueda, Recorrido y Ordenamiento.
- Archivos de Acceso Secuencial y Aleatorio: organizaciones y accesos. Procesamiento Básico.
- Recursividad.
- Nociones de Complejidad Computacional.
- Noción de Orden de Complejidad.



MARIA EUGENIA LAVORATTO
DIRECTORA
DIRECCIÓN ACADÉMICA
U.T.N. F.R.L.P.



CONTENIDOS ANALÍTICOS

UNIDAD TEMÁTICA I. CONCEPTOS BÁSICOS – ALGORITMOS Y DATOS

- 1.1. Conceptos Introdutorios: Problema, Dato, Acción, Informática, Computadora, Programa. Paradigma de Programación. Comprensión de problemas. Etapas del proceso de resolución de problemas. Estrategias de resolución de problemas. Introducción a las principales técnicas de diseño: divide y vencerás, partición en bloques, refinamientos sucesivos, algoritmos ávidos (método voraz), programación dinámica (recursión), vuelta atrás (backtracking). Comparación de técnicas.
- 1.2. Concepto de Datos. Estructuras de datos simples: variable, constantes, declaración y operaciones de manipulación. Tipos de datos simples: numéricos, lógicos y caracteres. Operaciones y expresiones. Entrada y salida estándar. Formateo.
- 1.3. Concepto de acción y condición. Estructuras de Control: Secuencia, Selección (simple y múltiple) e Iteraciones. Composición de estructuras. Estructura de programas.
- 1.4. Algoritmos: Documentación, Depuración de errores, Ejecución y Mantenimiento.

TIEMPO ASIGNADO: 30 HORAS

UNIDAD TEMATICA II. ABSTRACCIÓN PROCEDURAL

- 2.1. Aplicación de la técnica de diseño Descendente. Principios de la Programación Modular. Concepto de subprograma. Descomposición y Modularización. Pasaje de parámetros. Procedimientos y Funciones. Utilización práctica de la técnica Divide y Vencerás.
- 2.2. Tiempo de vida y Alcance de datos. Variables locales y globales. Reglas de alcance.
- 2.3. Introducción a la técnica de diseño Programación Dinámica: concepto de Recursión. Aplicación y verificación. Procedimientos y funciones recursivas.
- 2.4. Análisis de los algoritmos recursivos: comparación de la eficiencia con algoritmos iterativos.

TIEMPO ASIGNADO: 35 HORAS

UNIDAD TEMÁTICA III. ESTRUCTURAS DE DATOS COMPUESTAS

Tema 1: Estructuras de datos estáticas

- 3.1. Tipos de datos definidos por el usuario. Constructores. Conceptos básicos de estructuras de datos compuestas. Clasificaciones. Noción de tipo abstracto de datos.
- 3.2. Arreglos: Datos compuestos indexados. Arreglos unidimensionales, bidimensionales y multidimensionales. Operaciones básicas de carga, consulta, recorridos. Arreglos como parámetros.
- 3.3. Tratamientos básicos de arreglos unidimensionales: algoritmos clásicos de búsqueda, ordenamiento, mezcla.
- 3.4. Registros: Accesos a campos. Anidamiento de registros. Operaciones sobre registros. Arreglos de registros.

Tema 2: Estructuras de datos dinámicas

- 3.5. Archivos: Conceptos generales. Archivos de acceso secuencial y Aleatorio. Organizaciones y accesos.
- 3.6. Concepto de puntero. Datos compuestos enlazados por punteros. Definiciones de datos recursivas.
- 3.7. Listas lineales enlazadas simples: Declaraciones y operaciones básicas.
- 3.8. Aplicación de listas en pilas y colas. Operaciones básicas.

TIEMPO ASIGNADO: 50 HORAS

UNIDAD TEMATICA IV. ANÁLISIS COMPUTACIONAL

- 4.1. Concepto de Eficiencia. Medidas de Eficiencia: análisis *a priori* y *a posteriori*.

DIRECCION ACADEMICA
ES COPIA FIEL DEL ORIGINAL

MARIA EUGENIA LAVORATTO
DIRECTORA
DIRECCION ACADEMICA



- 4.2. Concepto de Complejidad. Tamaño o talla del problema. Tiempo de ejecución: aproximación por funciones. Función de complejidad como medida de la complejidad espacial/ temporal de un algoritmo, relativa al tamaño del problema. Órdenes de Complejidad: funciones de complejidad más comunes.
- 4.3. Cálculo del Orden de Complejidad de los algoritmos: aplicación práctica sobre métodos de Ordenamiento y Búsqueda. Análisis del peor caso, mejor caso y caso promedio. Tamaño de los datos. Comparación de eficiencia.
- 4.4. Corrección de programas. Errores más comunes en programación.
- TIEMPO ASIGNADO: 15 HORAS

UNIDAD TEMATICA V. LENGUAJES DE PROGRAMACIÓN

- 5.1. Empleo de un lenguaje imperativo estructurado: Turbo/Borland Pascal. Introducción: características, entorno, compilación y ejecución, errores. Formato de programas en Pascal.
- 5.2. Estructuras de control en Pascal.
- 5.3. Subprogramas y unidades de programación.
- 5.4. Resolución de problemas básicos (sumas y conteos, máximos y mínimos, manejos de caracteres, etc.) con aplicación adecuada de las estructuras de control y de datos.
- TIEMPO ASIGNADO: 30 HORAS

BIBLIOGRAFÍA OBLIGATORIA

| TITULO | AUTOR | EDITORIAL | AÑO DE EDICIÓN /ISBN | EJEMPLARES DISPONIBLES |
|---|--|--------------------------|----------------------|------------------------------|
| Fundamentos de Programación. Algoritmos y Estructuras de Datos. | Luis Joyanes Aguilar | Mc. Graw - Hill | 2004 | 2 |
| Algoritmos, Datos y Programas. Conceptos básicos | De Giusti, Madoz, Bertone, Naiouf, Lanzarini, Gorga, Russo | Pearson | 2001 | 8 |
| Fundamentos de algoritmos y programación | De Lage- Cataldi - Salgueiro. | Editorial Nueva Librería | 2008 | 1 |
| Programación en Turbo/Borland Pascal 7.0 | Luis Joyanes Aguilar | Mc. Graw - Hill | 1995/ 1999 | 8 ejemp/1995 5 ejemp/1999 |

BIBLIOGRAFÍA COMPLEMENTARIA

| TITULO | AUTOR | EDITORIAL | AÑO DE EDICIÓN /ISBN | EJEMPLARES DISPONIBLES |
|--|------------------------|-----------------|----------------------|------------------------|
| Introducción a la Programación | Braunstein - Gioia | Eudeba | 1998 | 2 |
| Algorítmica. Diseño y Análisis de algoritmos funcionales e imperativos | Frances Galve y otros. | Adison - Wesley | 1993 | 4 |

DIRECCIÓN ACADEMICA
ES COPIA FIEL DEL ORIGINAL
1998
1993
MARIA EUGENIA LAVORATTO
DIRECTORA
DIRECCIÓN ACADEMICA
U.T.N. F.R.L.P.



CARACTERÍSTICAS DE LA ACTIVIDAD CURRICULAR

DESCRIPCIÓN

A continuación se describen brevemente las actividades curriculares, las tareas a realizar por los docentes y alumnos, como así también los materiales didácticos que se requieran para desarrollarla.

El considerar los problemas básicos como punto de partida del proceso enseñanza aprendizaje, posibilita la actividad autogestionaria por parte del alumno y le permite aproximarse a las situaciones problemáticas realizando los procesos característicos de su profesión.

La propuesta es acercarse a dichos problemas integrando la teoría y práctica al modo de trabajo profesional, parafraseando a Albert Einstein "no hay mejor práctica que una buena teoría y viceversa". Es necesario encarar lo teórico-práctico como forma de generación de conocimiento. Si se producen aprendizajes verdaderamente significativos, se consigue uno de los objetivos principales de la educación: asegurar la funcionalidad de lo aprendido.

De acuerdo con estos objetivos:

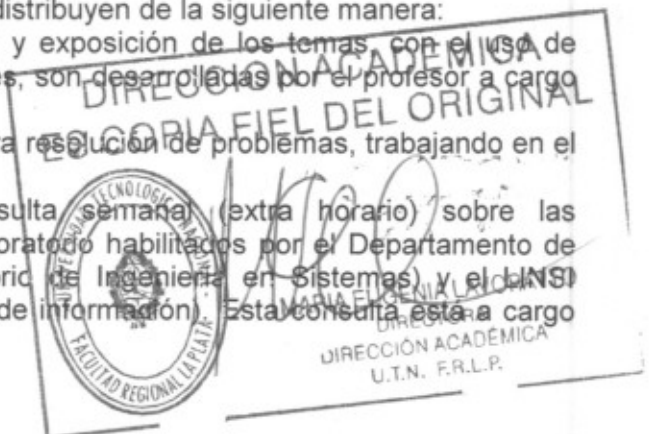
- ✓ Las clases teóricas incluirán gran contenido de práctica, serán participativas y de debate. Se desarrollarán de manera expositiva los conceptos fundamentales, con presentación de casos de estudio prácticos de aplicación inmediata del tema presentado. Se promoverá la investigación y la búsqueda de información. Se exigirá un trabajo continuo al alumno en la actividad áulica y fuera de ella.
- ✓ Las clases prácticas incluirán presentación del tema, casos de estudio abiertos a debate y de construcción colectiva y consulta individual de ejercicios.
- ✓ Los trabajos prácticos contendrán ejercicios-modelos resueltos y una lista de ejercicios a resolver. A medida que avance el tratamiento de las unidades temáticas, un trabajo práctico incluirá los conceptos aprendidos en los anteriores de modo que desde la presentación hasta el contenido serán cada vez más completos a medida que se avance en la cursada.
- ✓ Las clases en laboratorio funcionan como talleres, en donde se presentan las herramientas a emplear y se orienta para su instalación y aplicación; se desarrolla tomando como base 2 ejercitaciones globalizadoras (cuyos contenidos se puede descargar de la página Web de la materia: www.frlp.utn.edu.ar/materias/algoritmos aquí también se halla disponible el software Pascal Free)
- ✓ La cátedra dispone de un reglamento interno, el cual se adjunta y se entrega a los alumnos al comienzo de la cursada.

MODALIDAD DE LA ENSEÑANZA

La metodología adoptada para el desarrollo de la asignatura es la siguiente:

Las 5 horas semanales que tiene asignadas se distribuyen de la siguiente manera:

- ✓ Clases teóricas: 2 horas para presentación y exposición de los temas con el uso de pizarrón y apoyo de proyección de imágenes, son desarrolladas por el profesor a cargo del curso.
- ✓ Clases prácticas: 3 horas de ejercitación para resolución de problemas, trabajando en el aula, a cargo de un auxiliar.
- ✓ Clases de Laboratorio: 1 hora de consulta semanal (extra horario) sobre las Ejercitaciones de pascal (1 y 2), en los laboratorios habilitados por el Departamento de Sistemas de Información: LINES (Laboratorio de Ingeniería en Sistemas) y el LINSI (Laboratorio de innovaciones en Sistemas de información). Esta consulta está a cargo





de un Auxiliar Docente, quien será el responsable de llevar adelante la clase de Laboratorio, y de manera que trabajen 2/3 alumnos por máquina.

EVALUACIÓN

La cursada se regulariza mediante la aprobación de 2 (dos) parciales conceptuales prácticos, cada uno de los cuales tendrá dos fechas de recuperación, y se dispondrá de una fecha flotante única a la que se accede habiendo aprobado alguno de los 2 parciales; y de la aprobación de un trabajo en Pascal.

En el laboratorio de Pascal

- ✓ Los alumnos deben resolver e implementar en Pascal los ejercicios correspondientes a las 2 Ejercitaciones globalizadoras de programación. Se evalúan los avances de los trabajos en clases de taller sobre papel y en máquina (evaluación de seguimiento). Este desarrollo supervisado le permite adquirir los conocimientos necesarios para resolver un problema e implementar su solución en Pascal.
- ✓ Se evalúa el trabajo individual en forma escrita. Esta instancia se lleva a cabo una vez obtenida la aprobación del 1º y 2º parcial de la materia. Se establece un cronograma de fechas donde el alumno puede concurrir y ser evaluado por el Ayudante Diplomado a cargo de la comisión (5 fechas en total incluyendo recuperatorios). La certificación individual obtenida definirá la aprobación o no de la cursada.

La materia se aprueba mediante un examen final escrito, de carácter teórico-práctico, que constituye una evaluación sumativa final.

