

**UNIVERSIDAD NACIONAL DEL CENTRO
DE LA PROVINCIA DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS**

DEPARTAMENTO: Computación y Sistemas.

Asignatura: Introducción a la Programación II.

Carrera: Ingeniería de Sistemas.

Curso: 1^{er} año.

Profesor: Ing. Hernán Cobo.

Año de vigencia: 2009.

Programa Analítico - Contenidos

Unidad 1: Tipos enumerados y subrangos.

Objetivos: Que el alumno aprenda a programar utilizando la posibilidad de definir nuevos tipos de datos. Que programe utilizando los tipos enumerados y los subrangos como tipos de datos simples definidos por el programador. Que valore estos tipos de datos como documentación de los programas. Que trabaje con arreglos con índices de diferentes tipos ordinales y no sólo enteros.

Contenidos: Tipos ordinales. Tipos enumerados como tipos ordinales definidos por el usuario. Subrangos. Uso de la definición de tipos de datos. Utilización de tipos enumerados, otros tipos ordinales y subrangos, en particular con arreglos.

Unidad 2: Registros.

Objetivos: Que el alumno conozca y aprenda a utilizar los registros como estructura de datos. Que comprenda la diferencia entre registros y arreglos.

Contenidos: Tipos de datos estructurados estáticos. Registros como tipos de datos estructurados de componentes heterogéneos: su utilización. El acceso a los campos de datos. Registros vs. Arreglos.

Unidad 3: Archivos.

Objetivos: Que el alumno comprenda las características y diferencias entre memoria primaria y secundaria. Que entienda la necesidad de almacenar datos en un medio no volátil. Que el alumno aprenda a manejar archivos de datos. Que comprenda la diferencia entre las distintas formas de acceder a un archivo: secuencial y directa.

Contenidos: Características y diferencias de memoria primaria y secundaria. Archivos como almacenamiento de datos. Apertura, cierre, lectura y escritura. Acceso secuencial y directo. Algoritmos de búsqueda, agregado, inserción y eliminación en archivos. Consideraciones sobre la eficiencia en el uso de los archivos.

Unidad 4: Introducción al uso dinámico de memoria. Listas vinculadas.

Objetivos: Que el alumno comprenda las limitaciones en el uso de tipos de datos estáticos. Que comprenda la necesidad de trabajar con tipos de datos dinámicos. Que comprenda el concepto del tipo de dato puntero. Que aprenda a utilizar los punteros para la utilización de estructuras dinámicas de datos. Que conozca las listas vinculadas y aprende a programar los algoritmos que los manipulan: búsqueda, inserción, eliminación.

Contenidos: Limitaciones en el uso de memoria estática. Motivos para el uso dinámico de memoria. Punteros. Indirección en el acceso a los datos en memoria primaria. Apropiación y liberación de memoria. Listas vinculadas. Algoritmos de búsqueda, inserción y eliminación.

Unidad 5: Recursión.

Objetivos: Que el alumno aprenda a codificar soluciones recursivas, con procedimientos y funciones. Que aprenda a manejar adecuadamente los pasajes de parámetros en procedimientos / funciones recursivos. Que el alumno sepa identificar problemas recursivos.

Contenidos: Definición de recursión. Características de los problemas recursivos. Estructura de una solución recursiva: rama explícita y rama recursiva. Recursión con procedimientos y funciones. Seguimiento de soluciones recursivas. Pasaje de parámetros en soluciones recursivas.

Unidad 6: Estructuras dinámicas de datos.

Objetivos: Que el alumno conozca las estructuras dinámicas de datos más típicas: listas vinculadas, listas doblemente vinculadas y árboles binarios. Que comprenda y sepa elegir el tipo de dato más adecuado al problema. Que el alumno sepa codificar los algoritmos clásicos de manejo de estas estructuras dinámicas: búsqueda, inserción, eliminación.

Contenidos: Estructuras dinámicas de datos típicas: listas simplemente vinculadas, listas doblemente vinculadas, árboles binarios. Representación gráfica de estas estructuras. Algoritmos de búsqueda, inserción, eliminación y ordenamiento en las estructuras dinámicas. Árboles binarios de búsqueda. Recorridos de árboles binarios: preorden, enorden, postorden. Ventajas y desventajas en el uso de las estructuras dinámicas y criterios de eficiencia en tiempo y espacio.

Unidad 7: Combinación y construcción de estructuras de datos.

Objetivos: Que el alumno aprenda a construir tipos de datos estructurados a partir de los tipos de datos clásicos, adecuándolos a los problemas. Que sepa modularizar los programas para el manejo de las estructuras complejas. Que comprenda y aprenda a utilizar el concepto de tipos abstractos de datos.

Contenidos: Construcción y combinación de tipos de datos estáticos y dinámicos. Modularización de código para el manejo de estructuras complejas. Criterios de elección y diseño de estructuras de datos complejas. Noción de Tipo Abstracto de Datos.

Unidad 8: Implementación de un sistema.

Objetivos: Que el alumno implemente, documente y pruebe un sistema mediano a partir de una especificación detallada del diseño. Que el alumno pueda volcar en esta experiencia todos los conocimientos y habilidades de programación imperativa aprendidas en la asignatura. Que comprenda a través de una experiencia concreta la necesidad de escribir programas de buena calidad. Que tenga la experiencia de trabajar con unidades y tipos abstractos de datos, y de extender y modificar un sistema. Que comprenda el rol de la programación en el ciclo de vida de los sistemas.

Contenidos: Comprensión del dominio de un sistema determinado. Un caso práctico. Documentación de estructuras de datos y módulos funcionales. Planificación de una implementación. Implementación de un sistema. Documentación del código. Criterios de calidad de código: modularización, reusabilidad, legibilidad, mantenibilidad. Prueba de programas: “caja negra” y “caja de cristal”. Casos de prueba.

Bibliografía

- **Salmon, William I.**; *Introducción a la computación con Turbo Pascal: Estructuras y Algoritmos*. Addison-Wesley Iberoamericana. 1993.
- **Shackelford, Russell L.** *Introduction to Computing and Algorithms*. Addison-Wesley. 1998.
- **Bishop, Judy**; *Pascal Precisely, 3rd Edition*. Addison-Wesley. 1993.
- **Wirth, N.** *Introducción a la programación sistemática*.
- **Jensen, K. y Wirth, N.** *Pascal*.
- **Gries, D.** *A Science of Programming*.
- **Sitio web de FreePascal**
<http://www.freepascal.org/>.