

Córdoba, 10-febrero-2008

Universidad Tecnológica Nacional - Facultad Regional Córdoba -
Departamento de Sistemas

Modalidad AcadémicaAsignatura: **Algoritmos y Estructuras de Datos**Area: **Programación**Coordinador de cátedra: **Ing. Tymoschuk, Jorge Pablo****Objetivos Generales.**

- *Identificar problemas algorítmicos*
- *Conocer el proceso de diseño e implementación de software*
- *Aplicar las herramientas fundamentales representativas de los procesos, integrando la sintaxis elemental de un lenguaje de programación en el laboratorio asociado.*

Objetivos Específicos

. Desarrollar la capacidad de razonamiento y lógica, abordando problemas reales, analizándolos, definiendo las clases requeridas a su tratamiento, detallando el comportamiento necesario a nivel de métodos y atributos, verificando su correcto funcionamiento.

CONTENIDOS

Nota: Los contenidos mínimos especificados por Rectorado para la Implementación 2008 del Plan 1995 están subrayados para su fácil identificación.

Unidad I - Algoritmos con tipos de datos simples

Objetivos de la Unidad	3
Pasos a seguir en la resolución de un problema	3
Introducción a la POO	4
Que es la programación Orientada a Objetos	5
Componentes básicos de la POO	5
Características de la POO	5
Programación estructurada, Programación orientada a objetos	9
Lenguaje Java, características	11
Instalación de java	13
Compilación y ejecución de un programa	13
Gramática del lenguaje Java	14
Comentarios	14
Identificadores, palabras clave y reservadas	15
<u>Concepto de Dato</u>	15
Concepto de Información, dif. entre datos e información	16
<u>Tipos de Datos Simples</u> , Variables	17
Genero de las variables, asignación, inicialización	18
Operadores: unarios, binarios	19
Separadores	21
<u>Estructuras de control: secuencial, selectivas</u>	21

Alternativa simple, doble	21
Alternativa múltiple . . .	23
<u>Estructuras de Control: Bucles</u> . . .	25
El Bucle While . . .	25
Terminaciones anormales de un ciclo .	26
Bucles controlados por centinela .	27
Diseño eficiente de bucles . . .	27
Bucles controlados por banderas . .	29
Sentencias de quiebre de control .	30
Repetición: El Bucle For . . .	31
Repetición: El Bucle do-while . . .	34
Bucles Anidados . . .	35
Ejercicios propuestos para el alumno .	36

Tiempo previsto: 6 (Seis) semanas, concluyendo el 18 de abril/2008

Unidad II - Algoritmos con Tipo Abstracto de Datos

Objetivos de la Unidad . . .	3
Programación orientada a Objetos, Principios de diseño .	3
Abstracción, Tipo Abstracto de Datos . . .	3
Encapsulamiento, Modularidad . . .	4
Clases, (<u>Abstracciones con procedimientos y funciones</u>)	5
Estructura general . . .	5
Declaración y Definición, El Cuerpo de la Clase .	6
Acceso a miembros, Ciclo de Vida de los Objetos .	7
Declaración de los Miembros de una Clase . .	8
Modificadores de acceso a miembros de clases .	8
Separación de la interfaz . . .	10
Atributos de una Clase . . .	12
Métodos de una clase . . .	13
Llamadas a métodos . . .	14
El objeto actual (puntero this) . . .	15
<u>Pasaje de Parámetros</u> . . .	16
Métodos sobrecargados . . .	18
Resolución de llamada a un método . . .	18
Métodos constructores . . .	19
Constructores, Por defecto, Con argumentos, Copiadores	19
Caso especial . . .	20
public class paridad . . .	22
Interfaces . . .	23
Implementación de interfaces . . .	23
public class Character implements Caract01 . .	24

Tiempo previsto: 6 (Seis) semanas, concluyendo el 30 de mayo/2008

Unidad III - Estrategias de Resolución

Interacción de objetos, Abstracción . . .	3
Abstracción en software, Relaciones entre objetos .	4
Composición usando una sucesión de objetos Número .	6
Composición usando una sucesión de objetos Carácter, Clase Character	8
class SuceCar, Composición usando una progresión de caracteres	10
Tratamiento de frases . . .	12
Agrupar objetos. . .	14
Colecciones de tamaño fijo - <u>Arreglos</u> . . .	15
Declaración, creaciOn, uso de variables arreglo	16
Composición usando un arreglo de objetos Elemento	17
Composición usando una secuencia de números .	18

Colecciones de tamaño flexible, agenda personal .	21
Características importantes de la clase ArrayList .	23
Eliminar un elemento .	24
Procesar una colección completa, ciclo for-each .	24
Recorrer una colección .	26
Herencia .	29
Usar herencia .	31
Jerarquías de herencia .	32
Herencia e inicialización .	34
Agregando nuevos elementos a una jerarquía existente.	35
Ventajas de la herencia, Subtipos, Subtipos y asignación	36
Variables polimórficas, Enmascaramiento de tipos .	37
La clase Object .	38
Tipo estático y tipo dinámico .	39
Búsqueda dinámica del método .	40
Llamada a super en métodos .	42
Métodos polimórfico .	43
Paquetes (package) .	46
Tratamiento de Excepciones .	48
Lanzamiento de excepciones .	49
Atrapado de excepciones .	50
Generando nuestras propias excepciones.	53
<u>Algoritmos de búsqueda, recorrido y ordenamiento</u>	
Búsqueda en arreglos .	55
Búsqueda secuencial .	57
Búsqueda binaria .	58
<u>Nociones de Complejidad Computacional ,</u>	
Operaciones Primitivas .	59
Conteo, Notación asintótica .	61
<u>Ordenamiento</u> - Introducción,	
Algoritmos básicos y mejorados.	63
class OrdBurb, OrdSac, OrdPein, ordPeinCol .	64
<u>Determinación del Orden de Complejidad.</u> Tiempos .	68
Arrays multidimensionales .	70
Acceso a elementos mediante bucles .	71
Matriz de objetos Item. .	73
<u>Estructuras lineales.</u> .	79
<u>Pila</u> , concepto, diversas implementaciones .	79
public class Tiempos .	81
<u>Cola</u> , concepto, diversas implementaciones .	83
Implementación usando nodos enlazados. .	84
<u>Recursividad</u> , concepto, formulación recursiva del algoritmo	85

Tiempo previsto: 12 (doce) semanas, concluyendo el 19 de setiembre/2008

Unidad IV - Flujos y Archivos

Flujos .	.3
Clases File Input/Output Stream .	.5
Procesamiento Básico .	.5
Clases ByteArray Input/Output Stream .	.6
Clases Pipe Input/Output Stream .	.6
Clases Filtro .	.7
Clases Data Input/Output Stream .	.7
La Clase File, ejemplos. .	.10
Archivos secuenciales .	12
Creación de un archivo secuencial .	.11
Consulta de un archivo secuencial .	.14
Actualización de un archivo secuencial .	.16

Archivos de acceso aleatorio16
Creación de un Random AccessFile17
Métodos de posicionamiento17
Creando un archivo directo19
Class MiCasa19
Class CreaCountry20
Consultas en un archivo directo21
Class MiCasona22
Class ConsultaCountry22
Flujos de tipo Objeto24
Almacenamiento de objetos de distinto tipo.29
Recordando interfaces30
Implementando interfaces30
Trabajo Ing. Valerio Fritelli	
Relaciones entre objetos, interfaces, paquetes30
Class Principal32
Class Consola35
Interface Grabable36
Class Register.37
Class Archivo40
Class Alumno implements Grabable.48
Class Artículo implements Grabable.51

Tiempo previsto: 7(siete) semanas, concluyendo el 7 de noviembre/2008

Condiciones de Regularización

a) Realizar los trabajos prácticos individuales y grupales propuestos por el Jefe de Trabajos Prácticos respectivo, individuales (Escrito) y grupales (Medio magnético).

A partir del año 2004, se dispone en el sitio [labsys.frc.utn.edu.ar /Sitios de las cátedras/ AED](http://labsys.frc.utn.edu.ar/Sitios%20de%20las%20cátedras/AED) Año del siguiente material:

- Programa / Cronograma.
- Material por unidad
 - Carpeta Apunte: texto teórico / practico, formato pdf
 - Carpeta Java: fuentes .java, por proyecto, preparadas para ser descargadas por los alumnos y usadas en sus prácticos.

El promedio de prácticos proporciona una nota, la cual será promediada con las notas de los parciales. Esta nota, resultado de la entrega y aprobación de todos los prácticos solicitados, es un **requisito imprescindible para la regularización**.

b) Aprobar 2 (dos) parciales individuales. Existirá un parcial recuperatorio para quien haya aprobado un único parcial, el tema a ser desarrollado será independiente del parcial a recuperar. La nota del recuperatorio **no reemplaza** la del parcial recuperado. Se regulariza materia con dos parciales y prácticos aprobados y promedio no inferior a 4(cuatro).

Las fechas previstas son :

1er. parcial Unidades I y II completas. **9 al 20 de junio**, correspondientes a las semanas 14 y 15 del Calendario Académico.

2do. parcial Unidades III y IV, incluyendo, de ésta última, archivos secuenciales y de acceso directo. 27 de Octubre al 7 de noviembre, correspondientes a las semanas 30 y 31 del calendario académico.

Nota. La Cátedra abandonó el criterio de los parciales unificados del sábado a partir de cuando comienzan a evaluarse desempeños en máquina. El enunciado del parcial es sometido a aprobación del titular; normalmente el alumno baja de Autogestión parte de la programación fuente, configura un proyecto y trabajando en un entorno de desarrollo completa o modifica estos fuentes, compila y ejecuta el proyecto. La evaluación se realiza "on line" sobre el desarrollo del alumno. Previendo eventuales "fallas técnicas", disponemos siempre de una opción papel.

Recuperatorio: se tomará la **3ra semana de noviembre**. Los temas a evaluar son independientes del parcial a recuperar.

Examen final

Los temas de los exámenes finales son elaborados por una comisión interna de la cátedra y distribuidos a los tribunales, contemplándose la existencia de dos horarios.

El examen consta de parte práctica y teórica

Se toman todos los temas del programa, tanto en el práctico como teórico. Cualquier profesor puede evaluar a cualquier alumno.

Dependiendo del promedio de la regularización:

- Promedio igual o mayor a 8 (ocho) y ninguna nota menor a 7(siete), se promueve la evaluación práctica.
- Promedio inferior a 8 (ocho): examen práctico y teórico completo.

TRABAJOS DE INVESTIGACIÓN

La idea es la de encauzar y premiar la dedicación a la investigación en aquellos alumnos que están dispuestos a invertir tiempo en esta actividad. El tema podría ser una iniciativa del grupo o de la cátedra. En todo caso, será una programación funcionando en computadora en lenguaje Java, interactuando objetos de diversas clases, usando herencia, polimorfismo.

El profesor a cargo del curso y el jefe de prácticos son responsables por:

- Definir la forma de presentación del trabajo.
- Controlar la originalidad del trabajo.
- Orientar, asesorar al grupo.
- Evaluar el trabajo realizado.
- Presentar un informe al director de cátedra.

Como motivación a estos trabajos, la cátedra considerará promovidos a aquellos alumnos que:

- Promedio igual o mayor a 8 (ocho) y ninguna nota menor a 7(siete)

- Se apruebe el trabajo de investigación.

Bibliografía**Principal:**

- Apunte de la cátedra publicado por el CE y disponible en el sitio labsys.frc.utn.edu.ar, Sitios de las Cátedras, Algoritmos y Estructuras de Datos 2008 conjuntamente con todo el de código Java de la ejercitación incluida en el apunte.

De consulta:

- Estructuras de Datos y Algoritmos en Java, Goodrich/Tamassia, CECSA
- Algoritmos Y Estructuras de Datos en Java, Javier Cevallos

Hay un par mas, incluirlos

Entornos de programación sugeridos

Net Beans IDE, SUN,
BlueJ, www.bluej.org,

Ing. TYMOSCHUK, Jorge
Director de cátedra