

Córdoba, 11-Agosto-2009

Universidad Tecnológica Nacional - Facultad Regional Córdoba - Departamento de Sistemas

**Modalidad Académica****Asignatura: Paradigmas de Programación****Area: Programación****Coordinador de cátedra: Ing. Tymoschuk, Jorge Pablo**

**Nota:** El presente programa cumple con los contenidos mínimos especificados por Rectorado para la Implementación 2008 del Plan 1995.

**Objetivos Generales**

- Comprender los fundamentos de los paradigmas de programación básicos que son utilizados por los lenguajes de programación actuales.
- Conocer el modelo formal u semiformal subyacente de cada paradigma y la forma en que el mismo es incorporado en un lenguaje de programación concreto.
- Aplicar los diferentes paradigmas en la solución de problemas.

**Contenidos****Unidad I - Programación Orientada a Objetos avanzada**

Introducción a colecciones . . . . .	3
Interfaces de colecciones . . . . .	4
Interfaces de colección e implementaciones separadas . . . . .	5
Colecciones y las interfaces de iteración en la librería Java . . . . .	5
Colecciones concretas . . . . .	8
Listas enlazadas (LinkedList). . . . .	8
public class LinkedListTest . . . . .	12
public class SortedList extends LinkedList . . . . .	13
Array de listas (ArrayList). . . . .	16
Acceso a los elementos de un ArrayList. . . . .	17
Inserción y eliminación de elementos intermedios . . . . .	18
Una implementación de ArrItems extendiendo ArrayList . . . . .	21
Conjuntos de hash . . . . .	22
Usando class HashSet . . . . .	23
Arboles . . . . .	25
Árboles rojo-negro . . . . .	25
Comparación de objetos . . . . .	26
Class TreeSet . . . . .	28
Arbol TreeSet ordenado según interfaz Comparable . . . . .	28
Arbol TreeSet ordenado según objeto Comparator . . . . .	29
Conjuntos de bits . . . . .	31
public class CribaEras extends BitSet . . . . .	31
Herencia y Polimorfismo en Java. Caso concreto. . . . .	33
public class Progression . . . . .	33
class ArithProgression extends Progression . . . . .	34
class GeomProgression extends Progression. . . . .	35
class FibonacciProgression extends Progression . . . . .	35
class Tester . . . . .	36
<b>Programación dirigida a eventos.</b>	
Introducción. . . . .	38
Creando el proyecto . . . . .	38
Primeros pasos, Agregando un par de JPanel's al JFrame . . . . .	39
Agregando títulos de borde (title borders) al panel . . . . .	40
Incorporando componentes individuales, rótulo (JLabel). . . . .	42
Agregando, alineando, anclando . . . . .	43
Dimensionado de componentes . . . . .	43
Indentar varios JRadioButtons debajo de un JLabel . . . . .	43
Tratamiento de eventos . . . . .	45
Preparando la incorporación del tratamiento. . . . .	45
Incorporando el tratamiento de eventos. . . . .	48

Trazando una ejecución del Proyecto Contactos 50  
**Tiempo de dictado. 3 semanas, finalizando el 28/08**

### Unidad II - Programación concurrente

Programación Multithread , ventajas, desafíos . . . . .	2
Exclusión mutua, sincronización . . . . .	3
Calendarización de hilos, punto muerto, hilos en Java	4
class ParImpar extends Thread . . . . .	5
public class BounceThread . . . . .	6
La interfaz Runnable, Interrupción de threads . . . . .	8
Estados de thread: nuevos, ejecutab. bloqueados, muertos	9
Grupos de threads, prioridades, egoistas, . . . . .	16
Sincronización, bloqueo, Los métodos wait y notify . . . . .	17
Resumen, puntos muertos . . . . .	21
Espera a que un hilo finalice . . . . .	24
Métodos estáticos sincronizados . . . . .	26
El modelo productor/consumidor. . . . .	29
Usando Swing como interfaz de usuario . . . . .	32
<b>Tiempo de dictado. 3 semanas, finalizando el 18/09</b>	

### Unidad III - Programación Distribuida

Breve historia de Java y la Red . . . . .	2
Common Gateway Interface (CGI), alternatives . . . . .	3
Java y la Red . . . . .	4
JavaScript, . . . . .	5
Elementos scriplet, Comentarios, Declaraciones. . . . .	5
Expresiones, Scriptlets . . . . .	6
<b>Obtención de datos desde el navegador . . . . .</b>	<b>8</b>
Recuperación de datos utilizando formularios HTML . . . . .	8
Utilización del elemento <form>,El atributo action . . . . .	8
Utilización de Get y Post . . . . .	9
Utilización de controles HTML . . . . .	10
El elemento HTML <input> . . . . .	11
Atributos maxlength, checked, value . . . . .	12
Añadir botones y casillas de verificación . . . . .	13
Procesamiento de peticiones . . . . .	14
Utilización del objeto request . . . . .	15
Funcionamiento, El valor null . . . . .	16
<b>Presentación de JavaBeans . . . . .</b>	<b>18</b>
Separación de papeles, componentes, Introducción . . . . .	18
Propiedades, Construcción de un JavaBean . . . . .	19
Utilización de un JavaBean . . . . .	20
Etiquetas bean . . . . .	21
Proyecto PPRBean01 . . . . .	23
Proyecto SimulaAuditoriamedicaWe . . . . .	28
<b>Introducción a las bibliotecas de etiquetas . . . . .</b>	<b>32</b>
Dentro de una biblioteca de etiquetas . . . . .	34
Los manipuladores de etiquetas . . . . .	35
El descriptor de bibliotecas de etiquetas . . . . .	35
La directriz taglib . . . . .	35
Utilización de una biblioteca de etiquetas . . . . .	37
<b>Cómo escribir bibliotecas de etiquetas . . . . .</b>	<b>45</b>
Construcción de una etiqueta personalizada . . . . .	45
El archivo descriptor de bibliotecas de etiquetas . . . . .	48
El elemento < tag> . . . . .	50
Crearemos un proyecto PPRTag01 . . . . .	51
Crear etiquetas con atributos . . . . .	54
Etiqueta que retorna una línea de texto invertida . . . . .	55
Procesar una línea de texto cumpliendo una serie... . . . .	58
Herramienta Palette para diseñar páginas (html/jsp) 62	
Proyecto ReservaOnline . . . . .	66

<b>Seguir la pista a los usuarios, qué es una sesión?</b>	73
Cookies, Utilización de sesiones en JSP.	76
El objeto Session en detalle	76
Proyecto ContarAccesos usando un scriptlet	78
Proyecto ContarAccesos01 usando un JavaBean	79
Un carrito de compra	80

**Servlets y JSP, entrando en detalle**

Tecnología Java Servlet,
Funcion del un servlet en una aplicación web
Arquitectura de un servlet,
Procesamiento de solicitudes, un servlet sencillo
Ciclo de vida servlet, objetos Request y Response
Que sucede en una página JSP?
Aplicaciones Web y web.xml
Información de inicialización
Colaboración entre servlets

**Tiempo de dictado 6 semanas, finalizando el 30/10**

**Unidad V - Programación Funcional**

Un poco de Historia	3
El lenguaje Haskell	5
A. INTRODUCCIÓN	5
B. ¿Que es Haskell	5
C. ¿Por qué usar Haskell?	5
D. Desarrollo futuro de Haskell	6
E. Haskell en acción	6
1 - Tipos	6
1.1 Información de tipo	7
1.2 Tipos predefinidos	7
1.3 Funciones	7
2 - El entorno de Haskell - HUGS	7
3 - Funciones	9
4 - Listas	9
5 - Tuplas	10
6 - Ecuaciones con guardas	10
7 - Definiciones locales	10
8 - Expresiones lambda	11
9 - Disposición del código	11
10 - Tipos definidos por el usuario	12
11 - Tipos Recursivos	14
12 - Entrada/Salida	15
13 - Sobrecarga y Clases en Haskell	17
14 - Evaluación Perezosa (Lazy)	19
15 - Ejercicios Prácticos	19
Anexo "La enseñanza de Haskell"	21
Bibliografía	22

**Tiempo de dictado 1 semanas, finalizando el 6/11**

**Unidad VI - PROGRAMACION LOGICA**

1 - INTRODUCCION A LA PROGRAMACION LOGICA	2
2 - LOGICA PROPOSICIONAL	3
3 - INTRODUCCION A PROLOG	4
4 - CALCULO DE RELACIONES	6
4.1 RELACIONES	8
4.2 RELACIONES EN PROLOG	8
5 - BUSCANDO LAS SOLUCIONES	15
5.1 EL CONTROL EN PROLOG	16
5.2 EL BACKTRACKING	17
5.3 EL CORTE(!)	23
5.4 LA NEGACIÓN COMO FRACASO	26
5.5 REALIZANDO EL SEGUIMIENTO DE UN PROGRAMA	26
6 - ESTRUCTURA DE UN PROGRAMA EN PROLOG	26
7 - OBJETOS COMPUESTOS	36
8 - RECURSIVIDAD	37

9 - LISTAS	.	.	.	.	.	.	.	40
PARTE PRACTICA	.	.	.	.	.	.	.	45
SOLUCIONES	.	.	.	.	.	.	.	50
Referencias y fuentes	.	.	.	.	.	.	.	64
Tiempo de dictado 1 semana, finalizando el 13/11								

## Bibliografía Principal

[1] Apunte teórico/práctico de la cátedra

[2] Guía de prácticos en BBS/WEB.

### Bibliografía de Consulta

[3] Java 2, Cay S. Horstman/Gary Cornell, Ed. Prentice Hall

[4] Wang, Paul S., "Java, con POO y aplicaciones en la Web"

[5] Fco Javier Ceballos Java 2 Interfaces gráficas y aplicaciones Internet

[6] David J. Barnes/Michael Kolling, POO con Java

[7] Desarrollo Web con JSP Jayson Falkner, Ben Galbraith, 6 mas, Ed. Anaya

### Esquema de Regularización/Promoción:

- **Prácticos** (Grupales hasta 3 alumnos) Mínimo de 4(cuatro) y hasta 5(cinco). El promedio de prácticos es una nota a considerarse en el esquema de Regularización.
- **Parciales** (Individuales) Dos parciales aprobados, siendo que 1(uno) puede recuperarse. El recuperatorio no recubre la nota del parcial. Es una nota más y se promedia.

### Modalidad de examen final

- Los alumnos con promedio 8 y nota mínima 7 tienen promovida la parte práctica del examen final. O sea que rinden un examen de evaluación de conceptos, estrategias, interpretación, etc. También pueden optar por presentar un trabajo de investigación, constituyendo un grupo de a lo sumo 3(tres) integrantes.
- Los alumnos con promedio inferior a 8 rinden previamente parte práctica. Aprobada, pasan al teórico.
- Se toma cualquier punto del programa.
- Cualquier profesor de la cátedra puede examinar a cualquier alumno.

### Fechas

- **Evaluaciones parciales.** Quedan establecidas las siguientes fechas:
  - **1er parcial.** Unidades I y II, semana del 28/9 (2da semana después de concluida la unidad II - Concurrencia)
  - **2do parcial.** Unidades III - Programación Distribuida, semana del 02/11, (Semana inmediata siguiente a la finalización de Unidad III - Distribuida)
- **Recuperatorio:** Coincidente con el examen final, 5to o 6to llamado, a opción del alumno, una única vez (Solo se recupera un parcial).

### Trabajos de investigación

El tema puede ser indicado por el profesor a cargo del curso o ser una iniciativa del grupo, autorizada por el profesor. Se entiende que se tratará de programación funcionando en computadora. El profesor a cargo es responsable por:

- 1) Definir la forma de presentación del trabajo.
- 2) Controlar su originalidad.
- 3) Orientar/Asesorar el grupo de trabajo.
- 4) Evaluar lo realizado.
- 5) Presentar un informe al director de cátedra.