



Asignatura	Paradigmas de Programación
Ciclo Lectivo	2010
Vigencia del programa	Ciclo lectivo 2010
Plan	2008
Área	Programación
Carga horaria semanal	8 hs
Anual/ cuatrimestral	Cuatrimstral
Coordinador de Cátedra	Ing. Tymoschuk, Jorge Pablo
Objetivos de la Materia	Comprender los fundamentos de los paradigmas de programación básicos de los lenguajes de programación. Conocer el modelo formal u semiformal subyacente de cada paradigma y la forma en que el mismo es implementado en un lenguaje de programación concreto. Aplicar estos lenguajes en la solución de problemas.

Programa Analítico

Unidad Nro. 1: Programación Orientada a Objetos Avanzada

Objetivos Específicos: Completar la formación del alumno en recursos de la programación actual:

- Colecciones, para tratar en forma genérica diversas estructuras en memoria dinámica.
- Herencia y Polimorfismo, para tratar genéricamente comportamiento de diferente implementación.
- Programación interactiva: Uso de interfaces gráficas y respuesta a eventos

Contenidos:

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
Programación dirigida a eventos.	
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

Bibliografía:

Material de la cátedra publicado por Educa y disponible en el sitio labsys.frc.utn.edu.ar, Sitios de las Cátedras, Paradigmas de Programación, incluyendo:

- Documentos .pdf conteniendo el mismo material citado
- Proyectos en entorno NetBeans conteniendo ejemplos de codificación Java.

Criterio de Evaluación:

La resolución del problema planteado por el enunciado implica varias etapas:

- Analizar y comprender la codificación previa lista que el alumno recibe en el pre-proyecto
- Codificar e integrar con la codificación previa los puntos específicamente pedidos en el enunciado
- Ejecutar el proyecto completo

En este proceso estamos evaluando los siguientes aspectos:

- Capacidad de comprensión de codificación no propia.
- Capacidad de complementación con la codificación propia
- Capacidad de depurar errores detectados en la fase de ejecución

Unidad Nro. 2: Programación concurrente

Objetivos Específicos: Hacer conocer al alumno las ventajas y características de la programación de flujo múltiple. Entrenarlo en las técnicas de sincronización, comunicación e interrupción de hilos.

Contenidos:

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



Bibliografía: La misma descrita para la Unidad 1
Criterio de Evaluación: El mismo descrito para la Unidad 1

Unidad Nro. 3:

Objetivos Específicos: Entrenar al alumno en las técnicas y características de la programación distribuida:

- La finalidad de la programación multicapas y su implementación mediante el Patrón de Diseño Modelo-Vista-Controlador.
- Distintas formas de implementar el controlador.
- Técnicas de seguimiento de sesiones.

Contenidos:

Breve historia de Java y la Red	2
Common Gateway Interface (CGI), alternativas	3
Java y la Red	4
JavaScript,	5
Elementos scriptlet, Comentarios, Declaraciones.	5
Expresiones, Scriptlets	6
ARQUITECTURA WEB. Patrón de Diseño Modelo-Vista-Controlador.	
Obtención de datos desde el navegador	8
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
Presentación de JavaBeans	18
Separación de papeles, componentes, Introducción	18
Propiedades, Construcción de un JavaBean	19
Utilización de un JavaBean	20
Etiquetas bean	21
Proyecto PPRBean01	23
Proyecto SimulaAuditoriamedicaWeb.	28
Introducción a las bibliotecas de etiquetas	32
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
Cómo escribir bibliotecas de etiquetas	45
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
Seguir la pista a los usuarios, qué es una sesión?	73



Cookies, Utilización de sesiones en JSP	76
El objeto Session en detalle	76
Proyecto ContarAccesos usando un scriptlet	78
Proyecto ContarAccesos01 usandoun 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	

Bibliografía: La misma descrita para la Unidad 1
Criterio de Evaluación: El mismo descrito para la Unidad 1

Unidad Nro. 4: Programación Funcional

Objetivos Específicos: Suministrar al alumno conocimientos sobre las características de la programación funcional y sus diferencias de enfoque con respecto a la programación imperativa. Suministrarle entrenamiento en el uso de un lenguaje de este tipo, trabajando con un entorno adecuado.

Contenidos:

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: La misma descrita para la Unidad 1
Criterio de Evaluación: Verificar la capacidad del alumno de resolver planteos de dificultad similar a los tratados en clase, usando el mismo lenguaje y entorno.



Unidad Nro. 5: PROGRAMACION LOGICA

Objetivos Específicos: Suministrar al alumno conocimientos sobre las características de la programación lógica y sus diferencias de enfoque con respecto a los paradigmas anteriores. Suministrarle entrenamiento en el uso de un lenguaje de este paradigma, trabajando con un entorno adecuado.

Contenidos:

1 - INTRODUCCION A LA PROGRAMACION LOGICA	2
2 - LOGICA PROPOSICIONAL	3
3 - INTRODUCCION A PROLOG	4
3.1 LOGICA DE PREDICADOS DE PRIMER ORDEN y FORMAS RESTRINGIDAS	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
REGLA INFERENCIA DE RESOLUCION	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 PRÁCTICA	45
SOLUCIONES	50
Referencias y fuentes	64

Bibliografía: La misma descrita para la Unidad 1

Criterio de Evaluación: Verificar la capacidad del alumno de resolver planteos de dificultad similar a los tratados en clase, usando el mismo lenguaje y entorno.

Metodología de enseñanza y aprendizaje	Teórico y práctico desarrollado en laboratorio. Los conceptos teóricos son inmediatamente llevados a prácticos en el lenguaje de programación base del paradigma y luego son profundizados en clases prácticas (laboratorio) y prácticos grupales (Domicilio).
Sistema de evaluación	<p>Prácticos grupales son aceptados una vez que funcionen a satisfacción del Jefe de Prácticos.</p> <p>Exámenes Parciales se evalúan de acuerdo al grado de cumplimiento de la meta propuesta, que normalmente consiste en completar un proyecto parcialmente codificado y editado en el entorno de programación usado en el paradigma. El alumno debe interpretar el enunciado, analizar la programación ya existente y codificar los puntos pedidos, integrando todo el código. La evaluación se realiza directamente sobre el trabajo del alumno en la computadora, consistiendo en:</p> <p>1ro - Correcto funcionamiento de lo solicitado.</p> <p>2do – Explicar satisfactoriamente la codificación utilizada</p>



Condiciones de regularidad	Totalidad de prácticos grupales (mínimo cuatro) aceptados por el Jefe de Prácticos mas dos parciales aprobados. Se puede recuperar un parcial. Se promedian los prácticos grupales, obteniéndose 1 nota y los parciales(2 notas)
Modalidad de examen final	Práctico, si aprobado se pasa al teórico. El práctico se realiza en máquina siguiéndose la modalidad detallada para los exámenes parciales. Si el alumno tiene promedio ≥ 8 (ocho) y ninguna nota < 7 (siete) rinde examen teórico o puede presentar un trabajo de investigación consistente en un proyecto, normalmente un desarrollo de tópicos que representen una novedad o un avance sobre lo visto en el aula.
Actividades en laboratorio	La totalidad de las clases se desarrollan en el laboratorio de computación, tanto los horarios asignados a contenidos teóricos como la parte practica. Se utilizan los siguientes lenguajes y entornos: Unidades I, II y III: Lenguaje Java, entorno Netbeans Unidad IV: Lenguaje Haskell, entorno Hugs Unidad V: Lenguaje y entorno Prolog.
Horas/año totales de la asignatura	128
Cantidad de horas prácticas totales	84 (Agregar 8 horas para las evaluaciones parciales)
Cantidad de horas teóricas totales	36
Tipo de formación práctica (marque la que corresponde y si es asignatura curricular -no electiva-)	<input checked="" type="checkbox"/> Formación experimental – Asignatura curricular, no optativa <input type="checkbox"/> Resolución de problemas de ingeniería <input type="checkbox"/> Actividades de proyecto y diseño <input type="checkbox"/> Prácticas supervisadas en los sectores productivos y /o de servicios
Cantidad de horas afectadas a la formación práctica indicada	84
Descripción de los prácticos	Unidad Nro. 1: Programación Orientada a Objetos Avanzada A continuación una lista de los proyectos NetBeans conteniendo algunos ejemplos prácticos vistos en clase o a desarrollar por los alumnos como prácticos grupales: . <u>ArbolRojoNegro</u> . Árbol de búsqueda binario balanceado en altura usando la colección Java TreeSet . <u>CribaEras</u> : Algoritmo de Erastones (Cálculo de números primos) almacenados en una colección Java BitSet . <u>LinkedListTest</u> : manipulación de una lista de nodos vinculados implementada usando la colección Java LinkedList. . <u>ArrItems</u> : Manipulación de Objetos Item almacenados en una colección Java ArrayList. . <u>TreeSetTestGUI</u> : Gestión mediante interfaz gráfica y eventos de una colección TreeSet.



	<p>Unidad Nro. 2: Programación concurrente A continuación una lista de los proyectos NetBeans conteniendo algunos ejemplos prácticos vistos en clase o a desarrollar por los alumnos como prácticos grupales:</p> <ul style="list-style-type: none">. DemoJoin: Demostrativo del resultado de concatenar o la ejecución de hilos. HilosContando: Hilos contando/exhibiendo en una interfaz gráfica controlada por eventos.. ProdCons: Hilos productores/consumidores mostrando avances en una interfaz gráfica controlada por eventos.. ProgressBar: Qúmetro mostrando el avance de una actividad, numérica y gráficamente.. ThreadBall: Hilos manipulando una bola rebotante en los bordes, controlada por eventos en interfaz.. ThreadInterr. Demo de interrupción de hilos. <p>Unidad Nro. 3: Programación Distribuida. A continuación una lista de los proyectos NetBeans conteniendo algunos ejemplos prácticos vistos en clase o a desarrollar por los alumnos como prácticos grupales:</p> <ul style="list-style-type: none">. AnálisiDeTextoWeb: Aplicación web usando formularios HTML/JSP y un Java Bean para atender consultas de cantidad de caracteres, dígitos, palabras, existencia de palíndromos en un texto.. ReservasOnLine: Aplicación web usando formularios HTML/JSP y un Java Bean para atender requerimientos de reservas en un restaurante.. PPRTag01: Aplicación web usando formularios HTML/JSP y una biblioteca de etiquetas de usuario para con diversos manipuladores de etiquetas (handlers Tags), como ser invertir hileras, Tratamiento de caracteres, etc. RegularidadPPRWeb: Aplicación web usando formularios JSP y un Java Bean para informar la situación de regularidad de un alumno según su actuación.. SoporteTecnicoWeb: Aplicación web usando formularios JSP, imágenes, un servlet como controlador y varias clases para la lógica de negocios. Implementa el modelo MVC. Cookies: Aplicación web usando formularios HTML y un Java Bean para seguir la pista de transacciones de clientes en una sesión. <p>Unidad Nro. 4: Programación Funcional A continuación una lista de pequeños programas en lenguaje Haskell usando entorno Hugs. Array.hs</p>
--	---



	<p>Bits.hs Bool.hs Complex.hs Dynamic.hs</p> <p>Unidad Nro. 4: Programación Lógica A continuación una lista de pequeños programas en lenguaje Prolog usando su propio entorno. COMPRAS.PRO FACTORIA.PRO INTERVAL.PRO MATERIA.PRO PERSONAS.PRO SUMALIS.PRO LISCANT.PRO</p>
Criterios de evaluación de los prácticos	Mínimo de un 80% de los puntos especificados en el enunciado funcionando correctamente (en ejecución).
Formato de presentación de los prácticos	Proyectos de Programación en soporte magnético/óptico.
Cronograma de actividades de la asignatura , incluyendo semana prevista para cada práctico	Unidad Nro I - 3(tres) semanas, finaliza el 20 de agosto de 2010 Unidad Nro II - 3(tres) semanas, finaliza el 10 de septiembre de 2010 Unidad Nro III - 6(seis) semanas, finaliza el 22 de octubre de 2010 Unidad Nro IV - 1(una) semanas, finaliza el 29 de octubre de 2010 Unidad Nro V - 1(una) semanas, finaliza el 5 de noviembre de 2010
Descripción de metodología propuesta de consultas y cronograma de consultas	Uso de email y del foro del aula virtual, todo el año. No existe un cronograma que limite o restrinja las consultas de alumnos.
Plan de integración con otras asignaturas	El contenido de esta asignatura es pre-requisito del contenido dictado en Gestión de Datos.
Bibliografía Obligatoria	Material de la cátedra publicado por Educa y disponible en el sitio labsys.frc.utn.edu.ar, Sitios de las Cátedras, Paradigmas de Programación, incluyendo: - Documento .pdf conteniendo el mismo material citado - Proyectos en entorno NetBeans conteniendo codificación Java.
Bibliografía Complementaria	Java 2, Cay S. Hortsman/Gary Cornell, Ed. Prentice Hall Java 2, Francisco Javier Ceballos, Interfaces Gráficas y Aplicaciones Internet



Distribución de docentes por curso	Curso	Turno	Día y Horas	Profesor	JefeTrab.Práct.	Ayudante
	2k01	Mañana	mar 4,5,6,7 jue 1,2,3,4	Tymoschuk Jorge	Colacioppo Nicolas	Polliotto Martin
2k02	Mañana	mie 1 jue 1,2,3 vie 1,2,3,4	Ligorria Laura	Colacioppo Nicolas		
2k03	Mañana	lun 1,2,3,4 mie 4,5,6,7	Ligorria Laura	Colacioppo Nicolas		
2k04	Mañana (Cuatr.)	mie 1,2,3,4 jue 4,5,6,7 alterno)	Ligorria Laura	Polliotto Martin	Colacioppo Nicolas	
2k05	Tarde	lun 3,4,5,6 vie 1,2,3,4	Tymoschuk Jorge	Serra Silvio		
2k06	Tarde	mar 1,2,3,4 mie 3,4,5,6	Guzman Analia	Castillo Julio	Frias Pablo	
2k07	Tarde (Cuatr.)	mie 0,1,2,3 jue 3,4,5,6 alterno)	Guzman Analía	Corso Cynthia	Frias Pablo	
2k08	Noche	mar 0,1,2,3 vie 3,4,5,6	Garcia Gustavo	Ligorria Karina	Cargnelutti Fernando	
2k09	Noche	lun 3,4,5,6 jue 0,1,2,3	Garcia Gustavo	Ligorria Karina	Cargnelutti Fernando	
2k10	Noche (Cuatr.)	mar 3,4,5,6 mie 3,4,5,6 alterno)	Ligorria Karina	Romani German		
2k90	Noche (Carlos)	lun 3,4,5,6 mar 0,1,2,3 Paz)	Ligorria Laura	Ligorria Laura		