

Córdoba, 01 de agosto de 2004

Universidad Tecnológica Nacional - Facultad Regional Córdoba - Departamento de Sistemas

## **Modalidad Académica**

Asignatura: **Paradigmas de Programación**

Area: **Programación**

Coordinador de cátedra: **Ing. Tymoschuk, Jorge Pablo**

## **Introducción**

En prácticamente todas las disciplinas del saber humano alguna vez hubo más de un camino para llegar al mismo destino. En esto la programación no a sido una excepción. El destino, el fin en última instancia siempre ha sido el mismo. En cambio los caminos, los "vehículos" para transportarse en ellos han ido cambiando.

En un principio había un camino, el de la programación llamada "lineal", que en un buen momento "cambió de traje". Se transformó en estructurada, sin abandonar la vertiente imperativa. A su lado surgió el paradigma declarativo, con dos estilos: la programación lógica, y el funcional. Una importante evolución posterior, tal vez habría que hablar de revolución, de la programación imperativa es la orientada a objetos.

El tiempo siguió y hay nuevos caminantes. La programación controlada por eventos, la concurrencia, el paralelismo y la sincronización, todos ellos presentes en la programación multihilos. Además tenemos la WEB

En estos últimos años la cátedra a estado realizando una permanente actualización de contenidos. En 1999 reemplazamos Ada por Java., como el lenguaje soporte del paradigma concurrente, y en el Paradigma Orientado a Objetos incorporamos también el lenguaje Java, por parecernos importante el cambio de tratamiento que en una década se produce en este último paradigma. El 2001 incorporamos el concepto de applets, con algo de control de eventos. En 2002 sistematizamos la parte de programación controlada por eventos, introduciendo los conceptos de componente y contenedor e incorporamos una batería de casos prácticos en la página del Laboratorio de Sistemas. En 2003 reemplazamos la interfase gráfica AWT por la Swing e incorporamos algo de trabajo en red (Tecnología CGI) en el Paradigma Concurrente. Y este año reemplazamos esta última parte por la tecnología Servlets.

## **Objetivos Generales**

- Satisfacer, en lo que le compete, los aspectos del Perfil Profesional relacionados con la Programación, de manera que le quede al alumno un máximo provecho del conocimiento impartido.
- Desarrollar la capacidad de razonamiento y lógica, abordando problemas reales, analizándolos, definiendo la estrategia, codificando los algoritmos necesarios a su resolución.
- Ser capaz de plantear estos problemas en más de una metodología.
- Ser capaz de establecer criterios comparativos de soluciones en distintas metodologías.
- Utilizar estos criterios para efectuar análisis comparativo de casos concretos.

## **Contenidos**

### **Tópicos introductorios**

Concientizar al alumno de que el "Arte de la Programación", ya lleva décadas de constante perfeccionamiento. Reseñar los sucesivos paradigmas que se fueron alcanzando en esta evolución. Comparar las estrategias de resolución de problemas, vistas en la asignatura A.E.D. (Programación Imperativa) con las introducidas por los Paradigmas que son objeto de tratamiento en la presente asignatura.

## Unidad 1 – PROGRAMACION ORIENTADA A OBJETOS

### Objetivos :

Introducir en el alumno la idea de Clase, cuyos atributos y comportamiento constituyen la implementación efectiva de un Concepto. Concientizarlo de la importancia y ventajas de este enfoque.

Entrenarlo en el efectivo uso de las herramientas que lo posibilitan.

Familiarizar el alumno en la utilización de esta técnica, permitiendo la generalización del comportamiento pertenecientes a clases bases mediante el mecanismo de la herencia. Concientizarlo de las ventajas que esta reutilización de código presupone.

### Contenidos :

Introducción. Beneficios de la herencia. Heurísticas para crear subclases. Herencia y jerarquía de clases. Tipos de herencia. Clases derivadas (declaración, atributos y métodos, redefinición de métodos). Herencia simple (constructores y destructores). Herencia múltiple (constructores y destructores, clases base virtuales). Polimorfismo (funciones virtuales, métodos abstractos o funciones virtuales puras, clases abstractas, polimorfismo en c++).

Semana	Clase	Contenidos
31 02/08	T	Presentación. Tópicos introductorios. Conceptos generales de la programación orientada a objetos: introducción, que es la programación orientada a objetos, elementos o mecanismos, características, ejemplo ilustrativo: una manera de ver el mundo, ventajas y desventajas, lenguajes de la programación orientada a objetos.
31 02/08	P	Repaso de AED, funciones, vectores, punteros y listas.
32 09/08	T	<b>Programación orientada a objetos en java:</b> Introducción, clases (estructura general, declaración y definición, cuerpo, declaración de miembros de clases, modificadores de acceso a miembros de clases, atributos y métodos). Objetos o instancias de clases.
32 09/08	P	Identificación de clases enunciando atributos y métodos. Declaración de clases simples (por ejemplo: Persona) con tres o cuatro atributos y los métodos inicializar, mostrar todos los atributos, cargar todos los atributos y modificar o retornar algún atributo en particular. Utilización de objetos en el main() llamando a los métodos desarrollados.
33 16/08	T	Objetos e instancias de clases. Array de objetos y punteros a objetos. Puntero this. Funciones miembro, en línea, constructores, destructores, amigas, sobrecargadas, operador. .
33 16/08	P	Declaración e implementación de una clase simple. Por ejemplo, un vector, una lista. Declaración e implementación de clases con constructores (por defecto, con argumento y de copia). Utilización objetos en el main(). Declaración, implementación y utilización de clases amigas.

Semana	Clase	Contenidos
34 23/08	T	Introducción. Beneficios de la herencia. Heurísticas para crear subclases. Herencia y jerarquía de clases. Tipos de herencia. Clases derivadas (declaración, atributos y métodos, redefinición de métodos).
34 23/08	P	Declaración e implementación de clases bases y clases derivadas.
35 30/08	T	Herencia simple (constructores y destructores). Herencia múltiple (constructores y destructores, clases base virtuales). Polimorfismo.
35 30/08	P	Declaración e implementación de una clase base y su derivada utilizando distintos tipos de constructores. Polimorfismo con vector de punteros a la clase base

**Tiempo de dictado :** 5 (Cinco) semanas, finalizando el 03 de setiembre

## Unidad 2 PROGRAMACIÓN ORIENTADA A EVENTOS

Introducción. Los Applets. La clase JApplet. El ciclo de vida de un Applet. Páginas HTML y etiquetas. Componentes y Swing. Componentes y contenedores. Clases JComponent, JPanel. Componentes básicos: etiquetas, botones, campos de edición, casillas de verificación, listas desplegables, listas.

Contenedores básicos: paneles, paneles desplazables, ventanas, ventanas de diálogo, diálogos de opción.

Un applet con componentes.

Manejo de eventos: modelo de delegación, respondiendo al ratón, escuchas y adaptadores.

Principales tipos de eventos: ratón, teclado, acción, ventana.

Un Applet con eventos. Una ventana con eventos.

Parte práctica. Enunciados y ejercicios resueltos.

Tiempo de dictado : 2 (Dos semanas, finalizando el 17 de setiembre)

## Unidad 3 – CONCURRENCIA Y TRABAJO EN RED

### Objetivos :

Familiarizar al alumno en la noción de proceso y sus interacciones : comunicación, sincronización. Introducirlo en la problemática del procesamiento en paralelo, los recursos compartidos y la seguridad en la concurrencia.

Contenidos /Cronograma:

- 1. INTRODUCCION**
- 2. CONCEPTO DE CONCURRENCIA**
- 3. UN EJEMPLO TEORICO**
- 4. PROGRAMAS DE FLUJO UNICO**
- 5. PROGRAMAS DE FLUJO MULTIPLE**
- 6. PLANIFICACION DE HILOS**
- 7. LA CLASE THREAD**
  - 7.1 METODOS DE CLASE** // Multihola Java, contar
  - 7.2 METODOS DE INSTANCIA**
    - 7. 2.1 CREACION DE UN THREAD**
      - 7.2.1.1 Extendiendo Thread, 10 hilos contando**
      - 7.2.1.2 Implementando Runnable, 10 hilos contando**
    - 7. 2.2 ARRANQUE DE UN THREAD**
- 8. GRUPOS DE HILOS**
- 9. SCHEDULING**
- 10. HILOS DEMONIO**
- 11. SINCRONIZACION**
  - 11.1 METODOS INSTANCIA SINCRONIZADOS** // class Maxguard
  - 11.2 METODOS ESTATICOS SINCRONIZADOS**
  - 11.3 LA SENTENCIA synchronized**
  - 11.4 DISEÑOS DE SINCRONIZACION**
    - 11.4.1 SINCRONIZACIÓN EN LA PARTE DEL CLIENTE**
    - 11.4.2 SINCRONIZACIÓN EN LA PARTE DEL SERVIDOR**
- 12. COMUNICACION ENTRE HILOS**
  - 12.1 wait, notifyAll y notify**
    - 12.1.1 Productor y consumidor (Métodos instancia sincr.)**
    - 12.1.2 Servidor de impresión (Métodos de instancia y clase sincr.)**
  - 12.2 MAS SOBRE ESPERA Y NOTIFICACIÓN**
- 13. FINALIZACIÓN DE LA EJECUCION DE HILOS**
  - 13.1 CANCELACIÓN DE UN HILO**
  - 13.2 ESPERA A QUE UN HILO FINALICE**
  - 13.3 FINALIZACION DE LA EJECUCION DE APLICACIONES**

#### **14. CASOS RESUELTOS DE APPLETS MULTITHREADS**

- 14.1 - JApplet, JRootPane, contentPane - Introducción
- 14.2 - 3 hilos, 3 contadores independientes, eventos globales
- 14.3 - 3 hilos, 3 contadores independientes, eventos globales(bis)
- 14.4 - 3 hilos, 3 series aritméticas, eventos paso a paso
- 14.5 - 3 hilos, 3 series aritméticas, eventos comandan ciclos
- 14.6 - 3 hilos, 2 productores, 1 consumidor, eventos, sincronización

#### **15. DEMO DE ORDENAMIENTO (Verlo en [www.labsys.frc.utn.edu.ar](http://www.labsys.frc.utn.edu.ar))**

#### **16. MAS EJERCICIOS RESUELTOS**

- 16.1 - Reunion de amigos
- 16.2 - La liebre, la tortuga y el guepardo
- 16.3 - El sumador
- 16.4 - Controla llegada de personal
- 16.5 - Sincronizacion de un semáforo
- 16.6 - Testea eventos

#### **TRABAJO EN RED**

##### **1.1 INTRODUCCIÓN A INTERNET/INTRANET**

- 1.1.1 Introducción histórica
- 1.1.2 Redes de ordenadores
- 1.1.3 Protocolo TCP/IP
- 1.1.4 Servicios

##### **1.2 PROTOCOLO HTTP Y LENGUAJE HTML**

##### **1.3 URL (UNIFORM RESOURCE LOCATOR)**

- 1.3.1 URLs del protocolo http
- 1.3.2 URLs del protocolo FTP

##### **1.4 CLIENTES Y SERVIDORES**

- 1.4.1 Clientes (clients)
- 1.4.2 Servidores (servers)

##### **1.5 TENDENCIAS ACTUALES PARA LAS APLICACIONES EN INTERNET**

INVERSIÓN DE UNA CADENA DE CARACTERES (Ejemplos de uso tecnol. CGI)

INVERSIÓN DE UNA CADENA DE CARACTERES, ahora usando unJApplet y CGI

PROCESANDO UNA ENCUESTA

El "gateway" o programa CGI-bin que atiende a los alumnos encuestados

##### **2 DIFERENCIAS ENTRE LAS TECNOLOGÍAS CGI Y SERVLET**

##### **3 CARACTERÍSTICAS DE LOS SERVLETS**

##### **4.1 VISIÓN GENERAL DEL API DE JSDK 2.0**

##### **4.2 LA APLICACIÓN SERVLETRUNNER**

##### **4.3 FICHEROS DE PROPIEDADES**

##### **4.4 EJECUCIÓN DE LA APLICACIÓN SERVLETRUNNER**

##### **5.1 INSTALACIÓN DEL JAVA SERVLET DEVELOPMENT KIT (JSDK 2.0)**

##### **5.3 CÓDIGO DEL SERVLET**

##### **6 EL SERVLET API 2.0**

##### **6.1 EL CICLO DE VIDA DE UN SERVLET: CLASE GENERICSERVLET**

##### **6.1.2 El método service() en la clase GenericServlet**

##### **6.2 EL CONTEXTO DEL SERVLET (SERVLET CONTEXT)**

##### **6.2.1 Información durante la inicialización del servlet**

##### **6.2.2 Información contextual acerca del servidor**

##### **6.3 TRATAMIENTO DE EXCEPCIONES**

##### **6.4 CLASE HTTPSERVLET: SOPORTE ESPECÍFICO PARA EL PROTOCOLO HTTP**

##### **6.4.1 Método GET: codificación de URLs**

##### **6.4.2 Método HEAD: información de ficheros**

##### **6.4.3 Método POST: el más utilizado**

##### **6.4.4 Clases de soporte HTTP**

##### **6.4.5 Modo de empleo de la clase HttpServlet**

##### **7 FORMAS DE SEGUIR LA TRAYECTORIA DE LOS USUARIOS (Introducción)**

Tiempo de dictado : 5(Cinco) semanas, finalizando el 29 de Octubre.

#### **Unidad 4 – EL PARADIGMA LOGICO**

**Objetivos :**

Formar en los alumnos hábitos de programación relacional. Introducirlo en la utilización de una herramienta práctica para aplicaciones que implican búsqueda de padrones, búsqueda con rastreo inverso o información incompleta.

**Contenidos :**

Introducción. Cálculos con relaciones : Relaciones, reglas y hechos, consultas, qué es la programación lógica ?. Introducción a Prolog. Términos. Interacción con Prolog. Consulta de existencia. La negación como fracaso. Estructuras de datos en Prolog : Listas, términos como datos. Técnicas de programación : Propone y verifica, Variables para datos, listas de diferencias. Control en Prolog. Árboles de búsqueda. Cortes. ([4],305,...)

**Cronograma:**

Semana	Clase	Contenidos
40 28/10	T	Introducción a la programación lógica. Introducción a Prolog, cálculo con relaciones: relaciones y soluciones en Prolog.
40 28/10	P	Programación en Prolog. Entorno de programación. Sintaxis. Búsqueda de soluciones. Enunciados simples.
41 04/11	T	Estructura de un programa en Prolog: domains, predicates, goals y clauses. Ejemplificación. Objetos compuestos.
41 04/11	P	Programa en Prolog. Ejercicios con condiciones. Utilización de variables. Objetos compuestos.
42 11/11	T	Listas y recursividad.
42 11/11	P	Ejercicios con listas y recursividad. Repaso para el parcial.

**Tiempo de dictado :2/3** (Dos/tres) semanas, finalizando el 19 de noviembre

**Bibliografía**

- **Principal**

- [1] Apunte teórico/práctico de la cátedra
- [2] Ceballos, Fco Javier; "P.O.O. con C++"
- [3] Hekmatpour, Sharam ;"C++ Guía para programadores en C"
- [4] Ravi, Sethi; "Lenguajes de Programación conceptos y constructores"
- [5] Guía de prácticos en BBS/WEB.
- [6] Wang, Paul S., "Java, con programación orientada a objetos y aplicaciones en la WWW"
- [7] F.Alonso Amo y F.J.Segovia Perez, "Entornos y Metodologías de Programación"
- [8] Ken Arnold, James Gosling y David Olmes. El lenguaje de Programación JAVA

- **Complementaria**

- [1] Mac Graw Hill;"Aplique el Prolog"
- [2] Alvaro, Tasistro y Vidal, Jorge;"Programación lógica y funcional"
- [3] Winston, Patrick H. y Berthold K. Horn; "Lisp, Second Edition"
- [4] Wirth, N; "Algoritmos + Estructuras de Datos = Programas"
- [5] Martínez, Raúl; "Programación en Reglas de Producción"
- [6] Mattson, S.E.;"Conceptos de un modelo de estructura"
- [7] Clark,K.L. y Tarlund, S.A.;"Programación lógica : historia y teoría"
- [8] Barendregt, Henk y Barendsen, Erik; "Introduction to Lambda Calculus"

**Plantel docente**

**Situación al 30/07/2004**

**P.P.R. Distribución de docentes**

Curso	<<==HORARIOS ==>>>	A CARGO	1er ayudante	2do ayudante
Mañana				
03-227	lun 1,2,3 mar 1,2, jue 1,2	Tymoschuk, Jorge	Ferreira, Javier	Guemes, Marcelo Colacioppo, Nicolás
04-228	lun 1,2,3 mie 3,4,5,6	Ligorria, Laura	Colazo, Guillermo	Alborno, Soledad

06-230	mie 4,5,6	vie 1,2,3,4	Ligoria, Laura	Ferreyra, Javier	Solana, Dante
07-219	mie 1,2,3	jue 3,4,5,6	Ligorria, Laura	Ligorria, Karina	Colacioppo, Nicolás
<b>Tarde</b>					
01-220	lun 1,2,3,4	jue 4,5,6	Guzmán, Analía	Serra, Silvio	Paredes, Marisa
02-218	mar 1,2,3	vie 1,2,3,4	Tymoschuk, Jorge	Ligorria, Karina	Guemes, Marcelo
05-219	lun 1,2,3,4	mie 1,2,3	Guzmán, Analía	Serra, Silvio	Paredes, Marisa

#### Noche

08-2188	Jue 4,5,6	vie 3,4,5,6	Ligorria, Karina	Ferreyra, Javier	Guemes, Marcelo Martinez, Ramiro	
09-208	lun 4,5,6	mar 1,2,3,4	García, Gustavo	Ferreira, Javier	Solana, Dante	
10-219	mie 5,6	jue 1,2,3	vie 5,6	García, Gustavo	Ferreyra, Javier	Alborno, Soledad García, Graciela
90 (C.Paz)	mar 1,2,3	vie 2,3,4,5	Ligorria, Laura	Ligorria, Laura		

80 Mar 17,20 – 20,40 Mier 17,20 a 19,45 ¿????????? ¿??????????????  
(Rio II)

### Condiciones de Regularidad

Lo que sigue es copia de lo decidido en reunión de cátedra del 15-04-2002, conforme consta en acta.

#### Enunciados para evaluaciones parciales.

Se consensa que los jefes de prácticos elaborarán los enunciados de parciales (3 modelos con varios temas) en tiempo y forma para ser presentados a la respectiva reunión de cátedra, prevista para una semana antes del parcial unificado. Se recomienda que los enunciados no sean demasiado restrictivos, ni tampoco dejen librado totalmente a criterio del alumno la resolución. Como ejemplo, en el P.O.O, se especificarían las clases y el eventual esquema hereditario, y sería responsabilidad del alumno la distribución de métodos y atributos en ellas, la definición de las restricciones de acceso, etc.)

**Esquema de Regularización/Promoción.** Se debaten y definen los siguientes puntos.

- El recuperatorio no recubre nota de parcial. Es una nota mas y se promedia.
- El criterio de promoción de examen final práctico, promedio 8 y nota mínima 7 se mantiene sin cambios.

**Acerca de los prácticos (Grupales)** Se observan vicios en el esquema vigente, en el cual el alumno puede presentar un mismo práctico reiteradas veces, hasta su aprobación o vencimiento del plazo. Los alumnos se movilizan a último momento y se dan frecuentes casos de presentación de programas que ni siquiera fueron compilados. Esto tiene como consecuencia una saturación de la capacidad de corrección de los auxiliares docentes. Corrigiendo este vicio, se consensa en :

- Estipular el número de prácticos en un mínimo de 4(cuatro) y hasta 5(cinco). Los prácticos se receptan una única vez. De ellos:
- 1(Una) y solo uno puede ser rechazado.
- 1(Una) y solo uno puede ser devuelto para corrección, y recuperarse.
- Los restantes deben estar correctos en su recepción inicial.
- El número de prácticos grupales correctos para la regularización será de 3(tres) a 4(cuatro), según criterio del Jefe de Prácticos.

4) Los temas de los parciales son unificados por la cátedra.

### Examen final

- 1) Regularización parte práctica: examen práctico/teórico.
- 2) Promoción parte práctica: examen teórico, consistiendo en una evaluación de conceptos teóricos.
- 3) Se toma cualquier punto del programa.
- 4) Cualquier profesor de la cátedra puede examinar a cualquier alumno.

### **Trabajos de investigación**

Grupos de hasta 3(tres) alumnos pueden promover la materia aprobando un trabajo de investigación. El tema puede ser indicado por el profesor a cargo del curso o ser una iniciativa del grupo. 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.

El profesor a cargo de curso será auxiliado en algunos de estos puntos por el Docente Investigador, figura aprobada en el Consejo Departamental en sesiones del año 1996. Este docente será también responsable por el archivamiento de estos trabajos. Como motivación a estos trabajos, la cátedra considerará promovidos a los alumnos que :

- 1) Promocionen parte práctica.(Criterio ya descripto)
- 2) Se apruebe el trabajo de investigación.

Lo que sigue es copia de lo decidido en reunión de cátedra del 12-07-2002, conforme consta en acta.

**Evaluaciones parciales.** Quedan establecidas la siguientes:

1er parcial. Unidades 1, (Paradigma Objetos), teórico/práctico, **sábado 18 de setiembre.**

2do parcial. Unidades 2,3 (Interfases y Paradigmas Concurrente), **sábado 6 de noviembre.**

**Recuperatorio: 20 Noviembre + Fecha exámenes generales 6to llamado, opción del alumno, única vez.**

**Ing. TYMOSCHUK, Jorge**  
**Coordinador de Cátedra**