



Ministerio de Cultura y Educación  
 Universidad Nacional de San Luis  
 Facultad de Ciencias Físico-Matemáticas y Naturales  
 Departamento: Informática  
 Area: Area V: Automatas y Lenguajes

(Programa del año 2005)  
 (Programa en trámite de aprobación)  
 (Presentado el 29/09/2005 12:35:08)

### I - Oferta Académica

Materia	Carrera	Plan	Año	Período
ANALISIS COMPARATIVO DE LENGUAJES	LIC. CS. COMP.	11/98	2	2c
ANALISIS COMPARATIVO DE LENGUAJES	PROF.CS.COMP.	13/04	2	2c

### II - Equipo Docente

Docente	Función	Cargo	Dedicación
KAVKA, CARLOS	Prof. Responsable	P.ADJ EXC	40 Hs
ERRECALDE, MARCELO LUIS	Prof. Co-Responsable	P.ADJ EXC	40 Hs
CAGNINA, LETICIA CECILIA	Responsable de Práctico	JTP EXC	40 Hs
FUNEZ, DARIO GUSTAVO	Auxiliar de Práctico	A.1RA SEM	20 Hs
GATICA, CLAUDIA RUTH	Auxiliar de Práctico	A.1RA EXC	40 Hs

### III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
0 Hs	4 Hs	4 Hs	0 Hs	8 Hs

Tipificación	Periodo
C - Teoría con prácticas de aula	2 Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
06/09/2005	02/12/2005	13	104

### IV - Fundamentación

Este curso trata sobre los aspectos principales del diseño e implementación de los lenguajes de programación. También se incluye la teoría y los modelos formales subyacentes que forman la base para las decisiones tomadas en la creación de este tipo de lenguajes.

En este sentido, el curso complementa los contenidos introducidos en Programación II, donde el énfasis fue puesto en los aspectos de programación utilizando lenguajes representativos de algunos de los principales paradigmas de programación.

Ahora la descripción de los lenguajes se realiza por medio de la arquitectura de hardware y software subyacente que se requiere para la ejecución de programas escritos en estos lenguajes.

Los modelos formales que dan origen a los principales paradigmas

de programación también son analizados con una visión más general.

Los contenidos del curso se corresponden con las unidades de conocimiento recomendadas por la ACM/IEEE Computer Society Joint Curriculum Task Force en 1991, para el área de lenguajes de programación.

En el convencimiento de que los aspectos básicos del diseño de compiladores deberían formar parte del conocimiento general de cualquier buen programador, este curso muestra de que manera son compiladas las distintas construcciones de los lenguajes.

Los aspectos abordados, relacionados a la teoría formal de Lenguajes y parsing servirán además como introducción a tópicos más avanzados a desarrollarse en cursos de la carrera mayor como Autómatas y Lenguajes y Compiladores.

## V - Objetivos

Al finalizar el curso los alumnos serán capaces de:

Evaluar en forma crítica distintos lenguajes de programación existentes y futuros.

Responder a cuestiones tales como las motivaciones de la existencia de tantos lenguajes de programación, cómo y porqué fueron desarrollados, en que se asemejan y difieren.

Realizar un curso avanzado sobre el estudio del diseño y construcción de compiladores.

Reconocer los paradigmas claves usados en el desarrollo de lenguajes de programación modernos, sus bases teóricas, aplicativas y de implementación.

Entender la implementación de distintos lenguajes con suficiente detalle como para reconocer la relación entre un programa fuente y su comportamiento en ejecución.

Extender sus conocimientos sobre los tópicos anteriores con bibliografía adecuada y mínima supervisión.

## VI - Contenidos

### UNIDAD 1:

Razones del estudio de lenguajes de programación. Historia de los lenguajes de programación. Características de un buen lenguaje. Estándards. Efectos de los ambientes sobre los lenguajes. Ambientes Batch, interactivos, embebidos, de programación y marcos de trabajos de ambientes.

### UNIDAD 2

La estructura y operación de una computadora. Computadora de hardware, de firmware y simulada por software. Traductores.

Computadoras virtuales y ligaduras. Tiempos de ligadura. Paradigmas de lenguajes.

### **UNIDAD 3**

Sintaxis de los lenguajes de programación. Criterios sintácticos generales. Elementos sintácticos de un lenguaje. Estructura programa-subprograma general. Etapas en la traducción. Modelos de traducción formales. Gramáticas BNF. Automatas finitos. Expresiones regulares. Automatas Pushdown. Parsing.

### **UNIDAD 4**

Propiedades de tipos y objetos. Tipos de datos. Especificación e implementación de tipos de datos elementales. Declaraciones. Chequeo de tipos. Conversión de tipos. Asignación e inicialización. Tipos de datos numéricos. Enumeraciones. Booleanos. Caracteres. Internacionalización. Tipos de datos estructurados. Especificación e implementación. Vectores y arreglos. Registros. Cadenas de caracteres. Punteros. Conjuntos. Archivos.

### **UNIDAD 5**

Tipos de datos abstractos. Evolución del concepto de tipo de datos. Ocultamiento de la información. Encapsulamiento mediante subprogramas. Definiciones de tipo. Equivalencia por nombre y estructural. Administración de memoria. Administración de memoria estática. Administración de memoria basada en Stack. Heap con elementos de tamaño fijo y de tamaño variable.

### **UNIDAD 6**

Control de secuencia explícito e implícito. Secuenciamiento en expresiones aritméticas. Representación de árbol. Representación en tiempo de ejecución. Control de secuencia entre sentencias. Sentencias estructuradas.

### **UNIDAD 7**

Control de subprogramas. Llamada-retorno simple. Subprogramas recursivos. Variaciones en control de subprograma. Excepciones. Corutinas. Subprogramas planificados. Comandos en guardia. Tareas. Sincronización. Control de datos. Ambientes de referenciación. Alcance estático y dinámico. Estructura de bloques. Datos compartidos en subprogramas. Parámetros. Pasaje de parámetros. Ambientes comunes explícitos. Alcance dinámico. Alcance estático. Desplegado.

### **UNIDAD 8**

Tipos de datos abstractos genéricos. Programación Orientada a Objetos: Herencia, Polimorfismo y ligadura dinámica. Subclases y subtipos. Herencia simple y múltiple. Aspectos de diseño en Smalltalk, C++ y Java.

### **UNIDAD 9**

Programación Lógica. Prolog. Elementos básicos: hechos, consultas y reglas. Variables. Variables anónimas. Unificación. Reglas recursivas. Proceso de backtracking. Listas. Functores. Predicados fail y cut.

## **UNIDAD 10**

Lenguaje Java. Características básicas. Tipos. Estructuras de control. Arreglos. Clases. Herencia. Polimorfismo. Control de acceso. Paquetes. Interfases. Excepciones. Entrada - salida.

## **VII - Plan de Trabajos Prácticos**

Práctico 1: Programación Lógica - Prolog.

Práctico 2: Diseño de Lenguajes

Práctico 3: Lenguajes - Aspectos Sintácticos.

Práctico 4 (A): Tipos de Datos Elementales.

Práctico 4 (B): Tipos de Datos Estructurados.

Práctico 5: Tipos de Datos Abstractos - Administración de Memoria.

Práctico 6: Control de Secuencia.

Práctico 7: Control de Secuencia y Datos en Subprogramas.

## **VIII - Regimen de Aprobación**

Para regularizar la materia, los alumnos deberán aprobar 1 (un) parcial que incluye todos los prácticos, con al menos el 70% correcto. Este parcial tendrá 1 (una) recuperación.

La materia se podrá rendir como libre, con un exámen que incluye todos los contenidos teóricos y prácticos.

## **IX - Bibliografía Básica**

- [1] "Programming Languages - Design and Implementation". Pratt, Terrence y Zelkowitz, Marvin. Cuarta edición. Prentice Hall, 2001.
- [2] "Lenguajes de Programación - Diseño e Implementación". Pratt, Terrence y Zelkowitz, Marvin. Tercera edición. Prentice Hall, 1999.
- [3] "Concepts of Programming Languages". Sebesta, Robert. Cuarta Edición. Addison-Wesley, 1999.
- [4] Edición. Addison-Wesley, 1999.
- [5] "Prolog, programming for artificial intelligence". Bratko, Ivan. Addison-Wesley. Tercera Edición, 2001.
- [6] Ivan. Addison-Wesley. Tercera Edición, 2001.
- [7] "El lenguaje de Programación C". Kernighan, Brian y Ritchie, Dennis. Prentice Hall, 1991.
- [8] Dennis. Prentice Hall, 1991.
- [9] Apuntes de la materia de los lenguajes C, C++ y Lisp.
- [10] "Introduction to Java". Carlos Kavka. ICTP, 2004.

## **X - Bibliografía Complementaria**

- [1] "Programming Languages - A Grand Tour". Editor: Horowitz, Ellis.
- [2] Tercera Edición. Computer Science Press, 1987.
- [3] "Smalltalk-80. The Language and its implementation". Goldberg, Adele y Robson, David. Addison-Wesley, 1985.
- [4] Adele y Robson, David. Addison-Wesley, 1985.

## **XI - Resumen de Objetivos**

El curso tiene como objetivo introducir al alumno a la problemática del diseño e implementación de lenguajes, incluyendo fundamentos teóricos y modelos formales. El estudio se realiza teniendo en cuenta todos los paradigmas actuales de programación, realizando un estudio comparativo de las técnicas de implementación de cada uno de ellos.

## **XII - Resumen del Programa**

Historia de los lenguajes de programación. Evolución de los paradigmas de programación. Computadoras virtuales. Sistemas de traducción. Modelos sintácticos: gramáticas, expresiones regulares, autómatas, autómatas pushdown. Modelos semánticos: gramaticales, imperativos, aplicativos, axiomáticos y de especificación, gramáticas de atributos. Características esenciales de los lenguajes de programación y su implementación: tipos de datos y su representación, control de secuencia y datos. Programación orientada a objetos. Programación lógica. Prolog. Lenguaje Java.

## **XIII - Imprevistos**

--

### **ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA**

**Profesor Responsable**

Firma:

Aclaración:

Fecha: