



**UNIVERSIDAD NACIONAL DE RÍO CUARTO**

**FACULTAD DE CIENCIAS EXACTAS, FÍSICO-QUÍMICAS Y NATURALES**

**DEPARTAMENTO DE COMPUTACIÓN**

**CARRERA/S:** Analista en Computación (Cód. 12), Profesorado en Ciencias de la Computación (Cód. 13), Licenciatura en Ciencias de la Computación (Cód. 14).

**PLAN DE ESTUDIOS:** 1999 Versión 1 (para las 3 carreras)

**ASIGNATURA:** Análisis Comparativo de Lenguajes      **CÓDIGO:** 1956

**DOCENTE RESPONSABLE:** Mg. Ariel Gonzalez.

**EQUIPO DOCENTE:** Lic. Maria M. Novaira, Dra. Valeria Bengolea y Profesora Sandra Angeli.

**AÑO ACADÉMICO:** 2017

**REGIMEN DE LA ASIGNATURA:** cuatrimestral.

**RÉGIMEN DE CORRELATIVIDADES:**

<i>Aprobada</i>	<i>Regular</i>
	1948 – Programación Avanzada
	1949 – Organización del Procesaor

**CARGA HORARIA TOTAL:** 112

**TEÓRICAS:** 42hs      **PRÁCTICAS y LABORATORIO:** 70 hs

**CARÁCTER DE LA ASIGNATURA:** Obligatoria

## **A. CONTEXTUALIZACIÓN DE LA ASIGNATURA**

La asignatura pertenece al segundo cuatrimestre del tercer año de cursado para las tres carreras (Analista, Licenciatura y Profesorado).

Los temas tratados incluyen el estudio en profundidad de las herramientas de programación (lenguajes, compiladores, etc.) necesarias para el correcto desenvolvimiento en el ámbito laboral.

## **B. OBJETIVOS PROPUESTOS**

Lograr que el alumno adquiera la capacidad crítica en base a fundamentos técnicos sobre el análisis y elección del lenguaje de programación mas adecuado de acuerdo al tipo de problema. Además se espera que el alumno consiga adquirir los conocimientos básicos sobre los detalles de implementación de los diferentes aspectos de un lenguaje de programación.

## **C. CONTENIDOS BÁSICOS DEL PROGRAMA A DESARROLLAR**

Los contenidos mínimos son enfocados y divididos en estilos de programación (Paradigmas). El estudio de cada estilo comprende la definición y aplicación de los conceptos subyacentes, y de un conjunto de lenguajes de aplicación de cada paradigma.

Los mismos son estructurados de la siguiente manera:

- Introducción a los Lenguajes y Herramientas de Programación.
- Lenguajes y Modelos de Programación
- El modelo declarativo, Lenguajes Funcionales y Relacionales.
- El modelo con estado (statefull) y Lenguajes de Programación Imperativos.
- Manejo de Memoria.
- Programación Orientada a Objetos.
- Programación Concurrente.

## **D. FUNDAMENTACIÓN DE LOS CONTENIDOS**

Los temas tratados se fundamentan en la necesidad de comprender los conceptos subyacentes de los lenguajes de programación, para su posterior selección y utilización en la resolución de problemas.

La heterogeneidad y complejidad de los problemas actuales, obligan al profesional a estar al tanto de las herramientas de programación que automatizan los procesos del negocio

En este sentido, los lenguajes de programación son la herramienta principal del programador, y pueden ser categorizados de acuerdo a sus estilos o paradigmas. La comprensión de las características, técnicas de programación y técnicas de verificación de programas en cada uno de estos estilos, es de suma importancia para una selección adecuada del lenguaje que

implementará la solución de un problema determinado. La correcta elección de un lenguaje de programación brindará enormes beneficios relacionados con la mantenibilidad de los sistemas, la detección de errores, la legibilidad de los programas y otros más.

Los requisitos mínimos para la comprensión de los temas, incluyen conceptos relacionados a: Administración de Memoria, Estructuras de datos y Algoritmos, una mínima noción de los mecanismos para la definición de lenguajes de programación y la noción de recursividad.

Durante el desarrollo del curso los alumnos resolverán ejercicios prácticos pertinentes, siguiendo el apunte de cátedra elaborado por los propios docentes. La elaboración del apunte sigue el estilo de un libro en cuanto a estructura y descripción de los temas.

Para la comprobación del aprendizaje de los temas, se hará mediante dos exámenes parciales de la parte práctica (con sus respectivos recuperatorios) durante el curso, y posteriormente un examen final que abarcará los conceptos teóricos.

## **E. ACTIVIDADES A DESARROLLAR**

**CLASES TEÓRICAS:** Clases presenciales de 3 horas semanales.

### **CLASES PRÁCTICAS y CLASES DE TRABAJOS PRÁCTICOS DE LABORATORIO:**

Clases presenciales de 5 horas semanales divididas en dos clases de 2 y 3 horas. Las actividades prácticas y de Laboratorio se desarrollan en conjunto. Los ejercicios prácticos consisten en el planteo de problemas y realización de pequeños experimentos en diferentes lenguajes de programación que deberán implementar en el laboratorio.

## **F. NÓMINA DE TRABAJOS PRÁCTICOS**

La bibliografía de la asignatura está basada en un apunte (libro) confeccionado por los docentes. Al final de cada capítulo se describen una serie de ejercicios relacionados con los contenidos del mismo.

Listado de ejercicios por temática:

Capítulo 1: Introducción a los Lenguajes y Herramientas de Programación.

Capítulo 2: Lenguajes y Modelos de Programación

Capítulo 3: El modelo declarativo.

Capítulo 4: Lenguajes Funcionales.

Capítulo 5: Programación Relacional.

Capítulo 6: El modelo con estado (statefull).

Capítulo 7: Lenguajes de Programación Imperativos.

Capítulo 8: Manejo de Memoria.

Capítulo 9: Programación Orientada a Objetos.

## **G. HORARIOS DE CLASES:**

### **Teóricos:**

Miércoles de 11 a 14hs.  
Aula: aun no asignada.

### **Comisión 1:**

Lunes de 9 a 12 hs., Lab. 102 Pab. 2  
Viernes de 10 a 12 hs., Lab. 101 Pab. 2

### **Comisión 2:**

Martes de 16 a 19hs., Lab. 101 Pab. 2  
Jueves de 14 a 16 hs., Lab. 101 Pab. 2

## **HORARIO DE CLASES DE CONSULTAS:**

Miércoles 14:30hs. (Sujeto a cambios)  
Jueves 10hs (Sujeto a cambios)  
Viernes 9hs (Sujeto a cambios)  
Lunes 14hs (sujeto a cambios)

## **H. MODALIDAD DE EVALUACIÓN:**

### **Evaluaciones Parciales:**

Dos exámenes parciales con sus respectivos exámenes recuperatorios, los cuales consisten en ejercicios prácticos del tipo a los vistos en la parte práctica.

### **Evaluación Final:**

*Alumnos regulares:* evaluación, oral o escrita sobre conceptos teóricos.

*Alumnos libres:* en una primera instancia deberán aprobar ejercicios prácticos, y luego se procede a una evaluación similar para los alumnos regulares.

- **CONDICIONES DE REGULARIDAD:**

Aprobación de los dos exámenes parciales o sus recuperatorios.

- **CONDICIONES DE PROMOCIÓN:**

No tiene.

## PROGRAMA ANALÍTICO

### A. CONTENIDOS

**Unidad 1:** Lenguajes como herramientas de programación. Características generales. Elementos de un lenguaje. Especificación de un lenguaje de programación: Sintaxis y semántica. Declaraciones, definiciones, expresiones y comandos. Mecanismos para la abstracción funcional y de datos. Herramientas de programación: intérpretes, compiladores, enlazado (linking). Archivos objeto, bibliotecas y ejecutables.

**Unidad 2:** El modelo declarativo. Asignación única. Valores y tipos de datos primitivos y estructurados. Variables e identificadores. Sintaxis y semántica de un lenguaje núcleo declarativo. Adornos sintácticos y abstracciones lingüísticas. Tipado estático y dinámico. Manejo de la memoria. Unificación y ligadura.

**Unidad 3:** Lenguajes funcionales. El estilo de la programación funcional. Polimorfismo paramétrico. Fundamentos teóricos. Cálculo Lambda y lógica combinatoria. LISP. Lenguajes funcionales modernos: Haskell y ML.

**Unidad 4:** Programación relacional. El modelo de computación no determinístico. Sentencias *choice* y *fail*. Árbol de búsqueda. Programación lógica. Prolog. Forma clausal. Mecanismos de inferencia. Resolución. Características extra-lógicas: Cut, aritmética, entrada-salida. El problema de la negación.

**Unidad 5:** El modelo con estado. Extensión del lenguaje núcleo con celdas mutables. Semántica de celdas. Punteros y referencias. Aliasing. Igualdad. Asignación. Razonando con estado. Abstracción procedural. Sobrecarga. Efectos colaterales. Referencia transparenencial. Lenguajes de programación imperativos. El lenguaje C. Manejo manual de la memoria y sus problemas: Referencias colgadas y lagunas de memoria. Manejo automático de la memoria: Recolectores de basura y contadores de referencias.

**Unidad 6:** Programación orientada a objetos. Objetos y clases. Clases como módulos y tipos. Herencia. Sistemas de tipos. Control de acceso a métodos y atributos. Polimorfismo basado en herencia. Ligadura dinámica. Redefinición de métodos. Clases abstractas e interfaces. Herencia múltiple. Implementación de la ligadura dinámica.

**Unidad 7:** Polimorfismo por instanciación. Tipos parametrizados. Generecidad (clases parametrizadas). Funciones parametrizadas. Parámetros de tipo. Plantillas (templates). C++ templates. Generecidad en Java. Programación genérica. Computación estática. Herencia y clases parametrizadas.

**Unidad 8:** Concurrencia y paralelismo. Modelos de memoria compartida y mensajes. Threads, eventos, corrutinas, procesos secuenciales y concurrentes.

Problemas que plantea la concurrencia: no determinismo, dependencia de velocidad, bloqueos (deadlocks). Progreso finito (starvation). Interacción entre procesos. Procesos independientes, competitivos y comunicantes. Primitivas para la creación y destrucción de procesos y threads. Primitivas de bajo nivel de sincronización para exclusión mutua, semáforos. Constructores para el manejo de concurrencia de alto nivel: monitores y rendezvous. Ejemplos en Java y C++ . Concurrencia en Erlang. Mecanismos de comunicación entre procesos: tuberías (pipes), colas de mensajes, FIFOs, protocolos de comunicación entre procesos remotos (TCP/IP). Sockets.

## B. CRONOGRAMA DE CLASES Y PARCIALES

Semana	Día/Fecha	Teóricos	Prácticos y Laboratorio (*). Día/Fecha COM. 1 (lunes y viernes) Día: Tema	Prácticos y Laboratorio (*). Día/Fecha COM. 2 (martes y jueves)	Parciales / Recup.
1	Miércoles 16-08	X	Viernes 18-08	Jueves 17-08	
2	Miércoles 23-08	X	Lunes 21-08: : Feriado Viernes 25-08	Martes 22-08 Jueves 24-08	
3	Miércoles 30-08	X	Lunes 28-08 Viernes 01-09	Martes 29-08 Jueves 31-08	
4	Miércoles 06-09	x	Lunes 04-09 Viernes 08-09	Martes 05-09 Jueves 07-09	
5	Miércoles 13-09	X	Lunes 11-09: Asueto Viernes 15-09	Martes 12-09 Jueves 14-09	
6	Miércoles 20-09		Lunes 18-09 Viernes 22-09	Martes 19-09 Jueves 21-09: Asueto	
7		X	Lunes 25-09 Miércoles 27-09	Martes 26-09 Jueves 28-09	Primer Parcial Viernes 29-09 de 9-12hs
8	Miércoles 04-10	X	Lunes 02-10 Viernes: 06-10	Martes 03-10	Recuperatorio jueves 05-10 13-18hs
9	Miércoles 11-10	X	Lunes 09-10 Viernes 13-10	Martes 10-10 Jueves 12-10	

10	Miércoles 18-10	X	Lunes 16-10: Feriado Viernes 20-10	Martes 17-10 Jueves 19-10	
11	Miércoles 25-10	X	Lunes 23-10 Viernes 27-10	Martes 24-10 Jueves 26-10	
12	Miércoles 01-11		Lunes 30-10 Viernes 03-11	Martes 31-10 Jueves 02-11	
13	Miércoles 08-11		Lunes 06-11 Viernes 10-11	Martes 07-11	Seg. Parcial Jueves 09-11 de 13-16hs
14			Lunes 13-11 Miércoles 15-11	Martes 14-11	Recuperatorio: Jueves 16-11 de 13-16hs

**(\*) Las clases prácticas y de laboratorio se desarrollan en conjunto.**

### C. BIBLIOGRAFÍA

- o *Concepts, Techniques and Models of Computer Programming*. Van Roy and Haridi. The MIT Press. ISBN: 0-262-22069-5. 2004.
- o *Programming Language Concepts and Paradigms*. David Watt. Prentice Hall. 1990. ISBN: 0-13-728874-3.
- o *Programming Languages Design and Implementation (Third Edition)*. Terrence Pratt, Marvin Zelkowitz. Prentice Hall. 1996. ISBN: 0-13-678012-1
- o *Foundations on Logic Programming*. J. W. Lloyd. Second, extended edition. Springer Verlag. 1987. ISBN: 3-540-18199-7.
- o *Programming Language Concepts*. Carlo Guezzi, Medhi Jazayeri. 3rd edition. ISBN:0-471-10426-4.
- o Apunte de la Materia, elaborado por los docentes de la Asignatura.