

**UNIVERSIDAD NACIONAL DE RÍO CUARTO**

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

**DEPARTAMENTO DE COMPUTACIÓN**

**CARRERA/S: Analista Computación, Profesorado en Computación, Licenciatura en Ciencias de la Computación.**

**PLAN DE ESTUDIOS:**

**ASIGNATURA: Programación Avanzada                      CÓDIGO:1948**

**DOCENTE RESPONSABLE: Dr. Pablo F Castro**

**EQUIPO DOCENTE:**

**Maria Marta Novaira (Licenciada en Ciencias de la Computación),  
Ernesto Cerdá (Profesor en Computación),  
Pablo Ponzio (Doctor en Ciencias de la Computación).**

**AÑO ACADÉMICO: 2019**

**REGIMEN DE LA ASIGNATURA: Cuatrimestral**

**RÉGIMEN DE CORRELATIVIDADES:**

<b>Aprobada</b>	<b>Regular</b>
<b>Lógica Mat. El.</b>	
	<b>Intr. Alg. y Programación</b>

**CARGA HORARIA TOTAL:**

**TEÓRICAS:56 hs**

**PRÁCTICAS: 56 hs**

**LABORATORIO: 56hs**

**CARÁCTER DE LA ASIGNATURA: Obligatoria**

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

**La materia se encuentra en el segundo año de las carreras de Analista en Computación y Licenciatura en Ciencias de la Computación.**

## **B. OBJETIVOS PROPUESTOS**

**Que los alumnos sean capaces de:**

- **Desarrollar habilidades para el desarrollo formal de programas.**
- **Comparar críticamente los paradigmas de programación funcional e imperativo, desde el punto de vista de los métodos rigurosos de desarrollo de programas.**
- **Desarrollar especificaciones de programas simples.**
- **Comprender los conceptos básicos de la teoría de automatas y lenguajes.**
- **Comprender una teoría básica de estructuras de datos y su utilización para realizar programas simples.**

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

**Los contenidos de la materia incluyen las nociones básicas para que los alumnos puedan comprender, y usar en la práctica, el concepto de programa correcto. Para este propósito, como primer objetivo, se introduce a los alumnos a un lenguaje lógico que después se utiliza para la demostración de propiedades sobre programas. En particular, se ve un cálculo para la lógica proposicional y de primer orden., haciendo hincapié en su utilización para la resolución de problemas lógicos.**

**Como segundo tema se aborda el paradigma de programación funcional, los alumnos son introducidos a los conceptos básicos de la programación funcional: modelo computacional, ordenes de reducción, evaluación lazy, el lenguaje Haskell. Además, se introduce un cálculo de programas funcionales, que permite que el alumno se inicie en el mundo de las derivaciones de programas correctos. Para esto, se hace trabajar al estudiantado con problemas de mediana complejidad, lo cual posibilita que se puedan desarrollar las habilidades matemáticas necesarias para aplicar la teoría en la práctica; además, se requiere a los estudiantes la resolución de un trabajo práctico que incluye la implementación de un programa en un lenguaje funcional avanzado, lo que permite brindar al alumno una perspectiva clara de la utilización de los lenguajes funcionales en la actualidad.**

**Como tercer tema se aborda la lógica de Hoare y la derivación de programas imperativos correctos. Con este fin, se introducen los conceptos de pre/postcondición, invariantes, aserciones y el transformador de predicados WP. Para lograr que los estudiantes obtengan una perspectiva adecuada de la utilización de dichas técnicas para la programación, se utilizan problemas de mediana complejidad para los cuales se obtienen programas correctos utilizando las técnicas enseñadas.**

**Finalmente, se brinda una introducción a los temas básicos de lenguajes y autómatas: Automátas y lenguajes, Automátas finitos, expresiones regulares, gramáticas y las nociones básicas de computabilidad. Estos temas proveen a los estudiantes de una visión más amplia sobre la teoría de la computación.**

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

Los contenidos abordados permiten que los estudiantes adquieran los conocimientos básicos para poder desarrollar especificaciones de programas y programas correctos con respecto a sus especificaciones. De la misma forma, se introduce a los estudiantes al paradigma de programación funcional, lo cual hace posible que los alumnos conozcan otro paradigma de programación el cual les provee otra perspectiva sobre la programación, y a la vez permite una aplicación directa de los conocimientos lógicos y matemáticos obtenidos durante la materia.

El trabajo práctico, cuya aprobación es requisito para la regularidad, tiene por objetivo lograr que los alumnos puedan aplicar la teoría aprendida en la resolución de problemas mediante la aplicación de las distintas representaciones de datos estudiadas. Además, permiten integrar los contenidos de esta asignatura y afianzar las capacidades adquiridas en las materias de lógica y matemática discreta. Se intentará utilizar ejemplos y problemas interesantes, en los cuales las representaciones de datos más adecuadas sean difíciles de reconocer, intentando estimular al alumnado.

Se fomentará la lectura de material adicional y la auto organización de los alumnos en sus actividades. Además, se dejará en manos de los alumnos la instalación y manejo de las herramientas de software utilizadas en la asignatura, como una manera de estimular la práctica en cuestiones más técnicas (no necesariamente ligadas a los tópicos que la asignatura abarca), y la experiencia en la utilización de herramientas nuevas.

#### **E. ACTIVIDADES A DESARROLLAR**

**CLASES TEÓRICAS:** 2 clases teóricas de 2hs cada una por semana.

**CLASES PRÁCTICAS:** 2 clases prácticas de 2hs cada una por semana.

**CLASES DE TRABAJOS PRÁCTICOS DE LABORATORIO:** 2 clases de laboratorio por semana, en los laboratorios los alumnos tienen la posibilidad de implementar las resoluciones obtenidas para los trabajos prácticos de la materia.

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

La materia contará con dos trabajos prácticos obligatorios, sin recuperación, que serán evaluado por los docentes de la materia y su aprobación es condición para la regularidad.

El objetivo de los trabajos prácticos obligatorios es poder evaluar la aplicación de las técnicas aprendidas en problemas concretos de mediana complejidad, y la correcta comprensión de los fundamentos teóricos subyacentes a las técnicas estudiadas y que los estudiantes pueden iniciarse en la programación de un lenguaje funcional. También detectar y corregir problemas, y así afianzar los conocimientos adquiridos en materias anteriores, de las áreas de matemática discreta, lógica y programación.

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

Teóricos: Miércoles de 14hs a 16hs.

Viernes de 10hs a 12hs.

**Prácticos: Lunes de 14hs a 16hs.  
Jueves de 8 a 10hs**

**Laboratorios: Martes de 10 a 12hs.  
Martes 16hs a 18hs.**

**HORARIO DE CLASES DE CONSULTAS:** Tres horarios de consulta por semana de 1 hora cada una, a convenir con los docentes de la materia.

#### **H. MODALIDAD DE EVALUACIÓN:**

- **Evaluaciones Parciales: 2 parciales.**
- **Evaluación Final: El oral consta de una evaluación oral o escrita.**
- **CONDICIONES DE REGULARIDAD: Aprobación de los parciales y el trabajo práctico con nota de más de 5 (cinco).**
- **CONDICIONES DE PROMOCIÓN: Aprobación de los parciales y trabajo práctico con nota de más de 8 (ocho).**

## PROGRAMA ANALÍTICO

### A. CONTENIDOS

Lógica y Sistemas Formales, Expresiones booleanas, cálculo proposicional, cálculo de primer orden, resolución de problemas lógicos. Inducción y recursión.

Programación Funcional, Formalismo básico. Modelo computacional. Especificación de programas funcionales. Tipos de datos: listas, árboles y pilas. Ordenes de Reducción. Evaluación Perezosa. Funciones de Orden Superior. Programación básica en el lenguaje Haskell.

Verificación y Especificación de Programas Imperativos, Lógica de Hoare. Construcción de Programas Correctos. El transformador de predicados WP. Un lenguaje simple con guardas. Invariantes. Derivación de Ciclos. Metodología de Programación Dijkstra-Gries.

Autómatas y Lenguajes, Autómatas Finitos. Expresiones Regulares. Gramáticas. Jerarquía de Chomsky. Nociones básicas de computabilidad.

### B. CRONOGRAMA DE CLASES Y PARCIALES

Semana	Día/Fe cha	Teóricos	Día/Fe cha	Prácticos	Día/F echa	Laboratorios	Parciales / Recuperatorios
1	13/03 15/03	Introducción, Nociones Básicas	18/03	Practico1 introdutorio	19/03	Repaso	
2	20/03 22/03	Nociones Básicas, Funcional	25/03 29/03	Practico2 Funcional-Re cursión	26/03	Introducción a Funcional y Recursión	
3	27/03 29/03	Funcional, Haskell	1/04 4/04	Practico3 Funcional-Re cursión	02/04	Programación en Haskell: Ejercicios básicos	
4	3/04 5/04	Haskell, Lógica	8/04 11/04	Practico 4 Modelo Computacion al	09/04	Programación en Haskell: Ejercicios de mediana complejidad	
5	10/04 12/04	Lógica	15/04 18/04	Practico5-Lóg ica	16/04	Programación en Haskell: Definición de Tipos en Haskell	
6	17/04 19/04	Especificacione s	22/04 25/04	Practico5-Log ica/Practico6 Especificacion es	23/04	Programación en Haskell: Funciones de Alto Orden	
7	24/04 26/04	Inducción	9/04 2/05	Practico5 Especificacion es	30/04	Programación en Haskell, repaso.	1er Parcial:09/05 1er Rec: 16/05

8	<del>01/05</del> 03/05	Derivaciones	6/05 9/05	Practico6 Derivaciones	07/05		
9	08/05 10/05	Derivaciones	13/05 16/05	Practico6 Derivaciones	14/05	Proyecto Haskell	
10	15/05 18/05	Derivaciones	20/05 22/05	Practico6 Derivaciones	21/05	Proyecto Haskell	
11	22/05 24/05	Prog.Imperativa	27/05 30/06	Practico7 Imperativo	28/06	Proyecto Haskell	
12	29/05 31/06	Prog.Imperativa	03/06 06/06	Practico7 Imperativo	04/06	Proyecto Haskell	
13	05/06 07/06	Automatas	10/06 12/06	Practico8 Automatas			
14	12/06	Repaso Imperativo					2 Parcial:11/06 2 Rec:18/06

### C. BIBLIOGRAFÍA

#### Bibliografía Obligatoria:

- **Program Construction, Calculating Implementations from Specifications.** Roland Backhouse. Wiley&Sons.
- **Cálculo de Programas.** Javier Blanco, Silvina Smith y Damián Barsotti. Facultad de Matemática , Astronomía y Física. Universidad Nacional de Cordoba.
- **Introduction to the Theory of Computation.** Michael Sipser. PWS Pub. Co.

#### Bibliografía de Consulta:

- **Introduction to Functional Programming using Haskell.** Richard Bird. Prentice Hall Series in Computer Science.