

UNIVERSIDAD NACIONAL DE RÍO CUARTO
FAC. DE CS. EXACTAS, FCO-QCAS Y NATURALES

DEPARTAMENTO DE COMPUTACION

Carrera/s: Analista en Computación, Profesorado en Computación y Licenciatura en Computación.

Asignatura: PROGRAMACIÓN AVANZADA

Código: 1948

Equipo docente:

- **Profesor responsable:** Dr. Javier Blanco
- **Auxiliares:** AC. María Marta Novaira, AC. Sonia Permigiani, AC. Gastón Scilingo y Lic. Pablo Ponzio.

Año Académico: 2009

Régimen de la asignatura:

- Régimen de regularidad: Las condiciones para la regularidad de la asignatura son las siguientes:
 - Aprobar una serie de ejercicios prácticos obligatorios que consisten en implementaciones usando el lenguaje funcional Haskell.
- Régimen de promoción:
 - Aprobar una serie de ejercicios prácticos obligatorios que consisten en implementaciones usando el lenguaje funcional Haskell.
 - Aprobar dos exámenes parciales o un recuperatorio.
 - La nota promedio de las instancias de evaluación (exámenes parciales y trabajos prácticos) debe ser superior o igual a 7.
- Examen Final: El examen final para los alumnos regulares se llevará a cabo mediante evaluación escrita u oral dependiendo del número de alumnos inscriptos. Abarcará la totalidad de los contenidos de la asignatura, verificando que los alumnos hayan adquirido los conocimientos teóricos y puedan aplicarlos en casos concretos en la práctica.
Por otra parte, el examen final para alumnos libres estará compuesto por una primera instancia escrita, en la cual se evaluarán las capacidades de los alumnos para construir programas utilizando las técnicas de programación aprendidas en la asignatura. La segunda instancia del examen final para alumnos libres será similar al examen final para alumnos regulares.

Carga horaria semanal: 4hs. teóricos, 4hs. Prácticos (2 hs. Taller y 2 hs. Practico en aula) y 1hs. consultas.

Objetivos Generales:

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 de programación rigurosa.
- Desarrollar especificaciones de problemas simples.
- Comprender una teoría elemental de tipos abstractos de datos y su uso en el desarrollo de programas.

Contenidos:

Unidad 1: 'Lógica y Sistemas formales'

Expresiones booleanas. Cálculo de predicados. Expresiones cuantificadas. Especificaciones. Inducción y recursión.

Unidad 2: 'Programación funcional'

Formalismo básico. Modelo computacional. Técnicas de programación elementales: reemplazo de constantes por variables, modularización, tuplas (programación dinámica), generalización por abstracción. Ejemplos.

Tipos abstractos de datos: conjuntos y listas. Recursión final.

Unidad 3: 'Autómatas y Lenguajes'

Autómatas finitos. Expresiones regulares. Gramáticas. Jerarquía de Chomsky

Unidad 4: 'Programación Imperativa'

Estados y predicados. El transformador de predicados WP. Lenguaje de comandos guardados. Invariantes. Función de cota. Derivación de ciclos. Técnicas elementales: tomar términos de una conjunción, reemplazo de constantes por variables, fortalecimiento de invariantes y recursión final.

Formas Metodológicas de Enseñanza y Aprendizaje:

En las clases teóricas se desarrollarán los conceptos fundamentales, con sus demostraciones y ejemplos.

En las clases prácticas los alumnos resolverán ejercicios conteniendo problemas en los que aplicarán los temas teóricos aprendidos.

Habrará consultas para temas teóricos y prácticos, dónde cada alumno

podrá plantear y evacuar sus dudas, como así también recibir explicaciones adicionales sobre los temas en que tenga dificultades. Se hará un uso intensivo de la página web de la materia para la difusión de material, novedades, etc, y una lista de distribución por correo electrónico para que los alumnos puedan discutir problemas, realizar consultas o hacer entregas de trabajos. Además habrá un taller en el que los alumnos aprenderán el lenguaje de programación funcional Haskell. La metodología de trabajo en este taller será grupal, con no más de tres integrantes por grupo.

Bibliografía:

- Richard, Bird - "Introduction to functional programming using Haskell". 1998.
- Blanco Javier, Smith Silvina y Barsotti Damián - "Cálculo de Programas". 2008 (Cba.)
- Simon Thompson - "Haskell: The Craft of Functional Programming". 1999.
- Dijkstra - "A discipline of programming". 1976.
- Gries, D. - "The science of programming".
- Gries, D. y Schneider, F. - "A logic approach to discrete math". Sprieger. 1993.
- Kaldewaij, A - "Programming the derivation of algorithms"
- B.Ruiz Jiménez, F.Gutiérrez López, J.Gallardo Ruiz y P.Guerrero García - "Razonando con Haskell. Un curso sobre Programación Funcional"
- P.H.Hartel y R.Plasmeijer - "Functional Programming Languages in Education"

Vigencia del programa: primer cuatrimestre del 2009