

## PROGRAMA DE ASIGNATURA

<b>ASIGNATURA:</b> Algoritmos y estructuras de datos I	<b>AÑO:</b> 2012
<b>CARÁCTER:</b> Obligatoria	
<b>CARRERA:</b> Licenciatura en Ciencias de la Computación	
<b>RÉGIMEN:</b> cuatrimestral	<b>CARGA HORARIA:</b> 120 hs.
<b>UBICACIÓN en la CARRERA:</b> Primer año – Segundo cuatrimestre	

### FUNDAMENTACIÓN Y OBJETIVOS

Se pretende que el alumno adquiera:

- capacidad para analizar algoritmos
- capacidad y hábito de identificar abstracciones al abordar un problema
- familiaridad con técnicas frecuentes de diseños de algoritmos

### CONTENIDO

Unidad I: Expresiones cuantificadas: repaso de especificaciones con cuantificadores lógicos, revisión de la sustitución y la regla de Leibniz, reglas generales para las expresiones cuantificadas, cuantificadores aritméticos y lógicos.

Unidad II: Construcción y corrección de programas: especificaciones formales, ejemplos y resolución de problemas.

Unidad III: Técnicas elementales para la construcción de programas: definiciones recursivas, reemplazo de constantes por variables, modularización, uso de tuplas, generalización por abstracción.

Unidad IV: Tipos abstractos de datos: aritmética de precisión arbitraria, especificaciones implícitas, especificaciones implícitas con invariante de representación.

Unidad V: Recursión final: funciones recursivas finales, recursión lineal y final, recursión final para listas.

Unidad VI: Programación imperativa: estados y predicados, transformador de

predicados wp, definición de un lenguaje de programación imperativo (Skip, Abort, Asignación, Composición, Alternativa, Repetición).

Unidad VII: Cálculo de programas imperativos: derivación de ciclos. Técnicas para determinar invariantes (tomar términos de una conjunción, reemplazo de constantes por variables, fortalecimiento de invariantes).

Unidad VIII: Recursión final y programación imperativa; programas imperativos sobre listas.

Trabajos prácticos especiales

Proyecto 1: Tipos de datos en Haskell.

Proyecto 2: Recursión, alto orden, expresiones lambda y otros elementos de Haskell.

Proyecto 3: Lenguaje C primera parte.

Proyecto 4: Lenguaje C segunda parte.

## **BIBLIOGRAFÍA**

### BIBLIOGRAFÍA BÁSICA

- Cálculo de programas. Javier Blanco, Silvina Smith, Damián Barsotti, Córdoba: Universidad Nacional de Córdoba, c2008.

### BIBLIOGRAFÍA COMPLEMENTARIA

- Programming: the derivation of algorithms, Anne Kaldewaij, Prentice-Hall, 1990.

## **METODOLOGÍA DE TRABAJO**

La asignatura se organiza en clases teóricas y prácticas, de cuatro horas reloj cada una.

Las clases teóricas son de carácter expositivo, con participación de los estudiantes en la resolución de problemas planteados.

Se asignan además trabajos prácticos especiales que son desarrollados por los alumnos de manera grupal.

En las clases prácticas los alumnos trabajan de manera individual o grupal, bajo la supervisión y acompañamiento de los docentes.

En las clases prácticas de taller los alumnos acceden al laboratorio de informática, desarrollando las prácticas en computadora con software libre (Haskell, GCC).

## EVALUACIÓN

Se toman dos parciales, evaluándose respectivamente conceptos de programación funcional y programación imperativa.

### Requisitos para obtener la regularidad

- Aprobación de los dos parciales o un parcial y un recuperatorio.

### Requisitos para la aprobación:

- Evaluación integral de la materia, o promoción directa.

### Promoción:

- Aprobación de ambos parciales con nota no inferior a 7 (siete).