

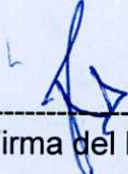
Computación  
2002

CARAL.DOC

**UNIVERSIDAD DE BUENOS AIRES**  
**FACULTAD DE CIENCIAS EXACTAS Y NATURALES**

1. DEPARTAMENTO: Computación.
  2. CUATRIMESTRE: Segundo 2002
  3. ASIGNATURA: Introducción a la Programación Funcional Genérica
  4. CARRERA: Licenciatura en Ciencias de la Computación
  5. CARÁCTER DE LA MATERIA: Optativa
  6. NUMERO DE CÓDIGO DE CARRERA: 18
  7. NUMERO DE CÓDIGO DE MATERIA: C .....
  8. PUNTAJE: 1 p
  9. PLAN DE ESTUDIOS AÑO: 1993 - 1984
  10. DURACIÓN DE LA MATERIA: Cuatrimestral
  11. HORAS DE CLASE SEMANAL:
    - a) TEÓRICAS/PRACTICAS: 15 hs
    - b) LABORATORIO:
    - c) PROBLEMAS HS.
    - d) SEMINARIOS HS
  12. CARGA HORARIA TOTAL: 15 hs
  13. ASIGNATURAS CORRELATIVAS: conocimientos básicos de álgebra, programación funcional y tipos de datos.
  14. FORMA DE EVALUACIÓN: trabajo final
  15. PROGRAMA: Se adjunta
  16. BIBLIOGRAFÍA: no fue especificada por el docente
- FECHA: julio 2002.

  
-----  
Firma del Profesor

  
-----  
Firma del Director  
Dr. Guillermo Duran  
Director Adjunto  
Depto. de Computación  
F. C. E. y N. UBA

## **Introducción a la Programación Genérica**

Horario: 14 a 17hs.

Dr. Alberto Pardo

El Dr. Alberto Pardo es profesor de la Universidad de la República, Uruguay. Obtuvo su doctorado en la Darmstadt University of Technology, Alemania. Sus intereses incluyen la transformación de programas, la programación genérica, la fusión de programas, la programación funcional, la semántica de lenguajes de programación y la teoría de tipos. Ha publicado numerosos artículos en importantes conferencias y revistas internacionales. También ha sido miembro del comité de programa de congresos especializados. Es el autor de una herramienta experimental MPG que genera parser monádicos escritos en Haskell/Gofer a partir de especificaciones de gramáticas en BNF.

Programa:

El objetivo de este curso es introducir los conceptos fundamentales relativos al diseño y transformación de programas funcionales sobre la base de una interpretación algebraica de tipos de datos. En particular, se muestra como dicho enfoque algebraico de tipos permite, en forma natural, la construcción de programas genéricos (también llamados politípicos) que son programas paramétricos en la signatura (estructura) de los tipos de datos que manipulan.

Introducción y Motivación.

Conceptos preliminares.: Tipos de datos básicos (producto, suma, etc.) y sus leyes algebraicas, El concepto de functor; funciones polimórficas.

Modelado algebraico de tipos de datos. Algebras de términos, signaturas de tipos como functores, Algebra de functores, álgebras iniciales. El operador fold y sus leyes algebraicas. Functores de tipos.

Funciones politípicas: Ejemplos de funciones politípicas,

Lenguajes de programación genérica: PolyP y Generic Haskell.

Operadores recursivos más avanzados. Aplicaciones de programación genérica.

Prerrequisitos: Conocimientos básicos de álgebra, programación funcional y tipos de datos.



Dr. Guillermo Duran  
Director Adjunto  
Depto. de Computación  
F. C. E. y N. UBA

com 2002  
13