



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

---

## PROGRAMACIÓN FUNCIONAL

Carrera: *Licenciatura en Informática*

Año 2007

Año: 4°

Duración: *Semestral*

Profesor: *Prof. Martínez López*

---

### OBJETIVOS GENERALES:

Desarrollar los conceptos y metodologías de programación asociadas con el paradigma funcional. Analizar técnicas y herramientas de Programación Funcional. Realizar trabajos experimentales de resolución de algoritmos con PF.

### CONTENIDOS MINIMOS:

- Conceptos básicos.
- Modelo de Computación del Paradigma Funcional.
- Técnicas Formales
- Aplicación de conceptos: Listas.
- Sistemas de Tipos.
- Técnicas de Diseño Funcional.
- Lambda Cálculo

### Programa

#### 1. **Conceptos Preliminares.**

- Revisión de la noción de programación y el concepto de programa.
- Propiedades deseables de los programas. Razonamiento y demostración de dichas propiedades.
- Dificultades del modelo clásico de programación para el razonamiento sobre programas.
- Descripción del modelo de programación funcional.
- Características principales de los lenguajes funcionales:
  - transparencia referencial,



- alto orden y currificación, y
- sistemas de tipos.

## 2. Modelo de Computación del Paradigma Funcional.

- Valores y expresiones. Las funciones como valores.
- Mecanismos de definición de expresiones y valores. Ecuaciones orientadas para definir funciones. Sintaxis.
- Visión denotacional y operacional de las expresiones. Modelos de computación mediante reducción. Semántica.
- Órdenes de reducción: reducción aplicativa y reducción normal.
- Sistema de Tipos Hindley-Milner. Tipos básicos. Constructores de tipos. Polimorfismo. Sintaxis para valores de cada tipo (caracteres, tuplas, listas, strings, funciones).
- Funciones parciales y totales.
- Funciones de alto orden. Currificación.

## 3. Técnicas Formales

- Demostración de propiedades
  - Noción de propiedad y de demostración. Diferentes formas de garantizar propiedades: por construcción, por chequeo automático, por demostración manual.
  - Algunas propiedades interesantes de los programas: corrección, terminación, equivalencia de programas.
- Inducción/Recursión.
  - Definición inductiva de conjuntos.
  - Definición recursiva de funciones sobre esos conjuntos.
  - Demostraciones inductivas sobre dichas funciones.
  - Ejemplos: programas, expresiones aritméticas, listas.

## 4. Aplicación de conceptos: Listas.

- Listas por comprensión. Definición y ejemplos. Semántica de listas por comprensión mediante reducción.
- Listas como tipo inductivo. Funciones básicas sobre listas (append, head, tail, take, drop, reverse, sort, elem, etc.).
- Funciones de alto orden para listas. Patrón de recorrido: map. Patrón de selección: filter. Patrón de recursión: foldr.



- Demostración de propiedades de las listas y de funciones sobre listas.

#### 5. Sistemas de Tipos.

- Nociones básicas. Sistemas de tipado fuerte. Ventajas y limitaciones de los lenguajes de programación con tipos.
- Lenguaje de tipos. Asignación de tipos a expresiones. Propiedades interesantes de esta asignación. Algoritmo de inferencia.
- Mecanismos de definición de tipos nuevos y de funciones sobre ellos. Tipos algebraicos. Ejemplos: enumeraciones, listas, árboles binarios, árboles generales.

#### 6. Técnicas de Diseño Funcional.

- Transformación y Síntesis de Programas .
  - Motivación. Obtención de programas a partir de especificaciones. Mejoramiento de eficiencia, con corrección por construcción.
  - Transformación de expresiones que utilizan listas por comprensión en expresiones que utilizan map, filter y concat.
  - Técnicas de transformación y síntesis de programas. Ejemplos.
- Combinadores y Transformadores.
  - Noción de combinador y transformador.
  - Estructuración de bibliotecas de funciones basadas en combinadores y transformadores. Ejemplos: parsing, pretty-printing, etc.
- Aplicaciones en Funcional
  - HyCom: construyendo hypermedia en Haskell.
  - Fudgets: interfases gráficas funcionales.
  - Lava: diseño de hardware.
  - HaskellScript: usando Haskell para scripting.

#### 7. Lambda Cálculo

- Definición del lenguaje. Sintaxis. Definición de sustitución.
- Modelo de computación. Nociones de alfa, beta y eta reducción. Semántica operacional.
- Lambda cálculo como modelo teórico de los lenguajes funcionales. Representación de booleanos, pares, números, listas y otras construcciones.



**8. Temas Adicionales (opcionales)**

- Sistemas inductivos. Órdenes parciales completos (CPOs).
- Formalización del sistema de tipos HM monomórfico.
- Lambda cálculos con tipos y con sustituciones explícitas.