

# Programa de Asignatura

## Ingeniería en Computación - In.Co.

<b>Nombre de la Asignatura</b>	Introducción a la Programación Funcional.
<b>Créditos</b>	7
<b>Docentes</b>	Guillermo Calderón y Sylvia da Rosa
<b>Objetivo de la Asignatura</b>	<p>Introducir los conceptos básicos de la Programación Funcional usando Haskell como lenguaje de programación.</p> <p>Ilustrar, desarrollando aplicaciones simples y de mediano porte, la versatilidad de Haskell como herramienta de prototipación y programación.</p> <p>Desde el punto de vista de metodología de la programación se hará hincapié en: el uso generalizado de funciones de alto orden y polimorfismo para la obtención de programas modulares y fácilmente reutilizables. La definición y uso de estructuras perezosas, y en particular de objetos infinitos, como metodología alternativa para la solución de problemas.</p>
<b>Metodología de enseñanza</b>	<p>Las horas en las que se trabaja en la asignatura serán distribuidas de la siguiente manera: en la primera mitad de las mismas aproximadamente, se trabajará sobre los conceptos teóricos y prácticos, utilizando el material que está especialmente elaborado para su uso en el curso, donde cada tema está acompañado de abundantes ejercicios planteados en forma ordenada de acuerdo a su complejidad.</p> <p>En la segunda mitad de las horas de la asignatura, los estudiantes trabajarán en una tarea en máquina planteada previamente y dispondrán de clases de consulta sobre la misma.</p>
<b>Temario</b>	<p>I. Introducción.</p> <p>    Qué es la programación funcional?     Haskell y Hugs.     Introducción al Cálculo Lambda.</p> <p>2. Tipos de Datos Básicos.</p> <p>    Enteros, Booleanos, Flotantes, Tuplas.     Definición de funciones.     Definiciones locales.</p>

### 3.Listas.

Listas en Haskell. Definición de funciones sobre listas.  
Pattern matching y definición de listas por comprensión.  
Aplicaciones.

### 4.Abstracción.

Alto orden y polimorfismo.  
Composición de funciones. Funciones como argumentos y resultados.  
Aplicación parcial de funciones.  
La noción de clase en Haskell.  
Chequeo automático de tipos.

### 5.Programación "in the large".

Tipos algebraicos (polimórficos). Tipos recursivos.  
Diseño de programas usando tipos algebraicos.  
Caso de estudio: códigos de Huffman.  
Tipos Abstractos de Datos.

### 6.Programación perezosa.

Evaluación perezosa.  
Listas infinitas.  
Caso de estudio: Algoritmos generales sobre grafos.

### 7.Entrada/Salida.

Mónadas para la programación funcional.  
Programación "imperativa" sin efectos laterales.  
La mónada IO.

## Bibliografía

The Craft of Functional Programming, Simon Thompson, Addison-Wesley,1996.  
ISBN: 0-201-40357-9

The Haskell Interpreter Hugs: <http://haskell.org/hugs/> , 1998.

A Gentle Introduction to Haskell, Paul Hudak. (<http://haskell.cs.yale.edu/tutorial/>)

## Conocimientos previos exigidos y recomendados

Experiencia en Programación, fundamentos de Lenguajes de Programación,  
nociones de Matemática Discreta.

## Anexo:

### 1) Modalidad de evaluación

Al finalizar la primera parte del curso, los estudiantes deberán rendir una prueba individual que los habilitará para realizar la tarea en máquina..

2) Cronograma tentativo

semana 1 : capítulos 1, 2 y 4

semana 2 : capítulo 6,7,8 y 9

semana 3 : capítulos 10,13 y 14

semana 4 a 7 : trabajo en máquina y consultas

3) Materia

Programación.

4) Previaturas

Lógica (examen).

Programación 2 (examen).

Teoría de lenguajes (examen).

Matemática Discreta I (examen).

**Aprobado por Res. del Consejo de Facultad el 13.3.06 – Exp. 060120-000075-06**