
Formulario de Aprobación Curso de Posgrado 2014

Asignatura: Introducción a la Programación Genérica

Profesor de la asignatura ¹: Dr. Alberto Pardo, grado 5 - 35 horas DT, Inco

Profesor Responsable Local ¹:

Otros docentes de la Facultad:

Docentes fuera de Facultad:

Instituto ó Unidad: Instituto de Computación

Departamento ó Area: Programación

¹ Agregar CV si el curso se dicta por primera vez.

(Si el profesor de la asignatura no es docente de la Facultad se deberá designar un responsable local)

Fecha de inicio y finalización: primer semestre

Horario y Salón: Martes y Viernes de 9:00 a 11:00, salón 111.

Horas Presenciales: 73 horas

Nº de Créditos: 10

Público objetivo y Cupos: Sin Cupo

Objetivos:

Los *programas genéricos* son programas parametrizados en la estructura de los datos que manipulan. Este mecanismo de parametrización permite escribir programas que pueden trabajar sobre una importante clase de diferentes estructuras de datos, liberando así al programador de escribir código adicional de similar funcionalidad cada vez que se quiera computar el programa sobre una nueva estructura de datos. Programas genéricos, tales como analizadores sintácticos o pretty-printers pueden ser escritos una sola vez y luego especializados para trabajar sobre diferentes estructuras de datos sin esfuerzo adicional por parte del usuario. Esto simplifica en forma considerable el trabajo de construcción y mantenimiento de los sistemas de software ya que los programas se pueden adaptar automáticamente a cambios en la representación de los datos, reduciendo de esta manera el impacto de los cambios.

El objetivo de este curso es introducir los conceptos fundamentales que hacen posible la construcción y derivación de programas genéricos en el contexto de programación funcional. El abordaje a programación genérica tratado en el curso tiene como base el modelado algebraico de tipos y programas, el cual demuestra ser un marco apropiado para el desarrollo de programas funcionales genéricos. Este abordaje hace especial énfasis en las propiedades algebraicas de los tipos de datos las cuales dan origen a una gama de operadores recursivos, como *map* y *fold*, entre

otros, de uso frecuente en programación funcional. En el curso se presenta la definición genérica de varios operadores recursivos y se discute el papel que dichos operadores juegan en el diseño y construcción de programas genéricos. Asimismo, se discuten las propiedades algebraicas de los operadores, mostrando su aplicación en la transformación de programas funcionales.

Conocimientos previos exigidos:

Conocimientos previos recomendados: Programación Funcional

Metodología de enseñanza:

Semanalmente se dictarán 4 horas de clase teórico-prácticas en las cuales se presentarán los conceptos y técnicas fundamentales. El estudiante deberá profundizar los temas haciendo uso de la bibliografía sugerida. Se recomienda al estudiante dedicar, en promedio, 4-5 horas semanales complementarias de estudio, incluyendo la realización de ejercicios prácticos.

A lo largo del curso se marcarán ejercicios prácticos para realizar, algunos de los cuales serán obligatorios (y realizados en forma personal) con el objetivo de luego ser corregidos.

En el curso habrá práctica de laboratorio. Como mínimo se hará uso del lenguaje PolyP el cual permite escribir programas genéricos y obtener instancias específicas de los mismos en el lenguaje funcional Haskell. Se estima que el laboratorio requiera una dedicación aproximada de 30 horas.

Se introducirá al estudiante en la actividad de investigación por medio de la lectura y presentación de artículos técnicos de revistas y conferencias. De esta forma se espera que el estudiante no sólo adquiera conocimientos en el área específica del curso sino también tenga una actitud crítica acerca de los desarrollos existentes en el tema.

En resumen:

- Horas clase (teórico): 60
- Horas clase (práctico):
- Horas clase (laboratorio):
- Horas consulta: 10
- Horas evaluación: 3
 - o Subtotal horas presenciales: 73
- Horas estudio: 30
- Horas resolución ejercicios/prácticos: 50
- Horas proyecto final/monografía:
 - o Total de horas de dedicación del estudiante: 153

Forma de evaluación:

Entrega de ejercicios prácticos.
Presentación de un tema relacionado al curso.
Examen final.

Temario:

- Introducción y Motivación.
 - **Conceptos preliminares:** tipos de datos básicos (producto, suma) y sus leyes algebraicas; el concepto de functor, functores polinomiales; transformaciones naturales.
 - **Tipos inductivos:** interpretación algebraica de tipos de datos, functores como firmas, álgebras de functores; álgebras iniciales y tipos inductivos; el operador fold y sus leyes algebraicas; functores de tipo; functores regulares.
 - **Tipos coinductivos:** co-álgebras; co-álgebras finales y tipos coinductivos; el operador unfold y sus leyes algebraicas; functores de tipo.
 - **Tipos recursivos:** interpretación de tipos de datos en teoría de dominios; el operador hylomorfismo y sus leyes algebraicas.
 - Lenguajes para la programación genérica (PolyP, Generic Haskell, etc).
-

Bibliografía:

Algebra of Programming, Richard Bird y Oege de Moor, Prentice Hall, 1997. ISBN-13: 978-0135072455

Introduction to Functional Programming using Haskell, Richard Bird, Prentice Hall, 1998. ISBN-13: 978-0134843469.

Generic Programming - An Introduction -, Roland Backhouse, Patrick Jansson, Johan Jeuring y Lambert Meertens, Advanced Functional Programming, Lecture Notes in Computer Science, 1608, Springer-Verlag, 1998. ISBN 3540662413.

Polytypic Programming, Johan Jeuring y Patrick Jansson, Advanced Functional Programming, Lecture Notes in Computer Science, 1129, Springer-Verlag, 1996. ISBN 3540616284.

Durante el curso se hace referencia también a otras publicaciones técnicas relacionadas.
