Aprobado por resolución N°113 del CFI de fecha 04.07.2017

FACULTAD DE INGENIERIA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Programa de PROGRAMA FUNCIONAL AVANZADA

1. NOMBRE DE LA UNIDAD CURRICULAR

Programación Funcional Avanzada

2. CRÉDITOS

12 créditos

3. OBJETIVOS DE LA UNIDAD CURRICULAR

El objetivo de este curso es introducir al estudiante en algunos de los últimos avances en la práctica de la programación funcional. Se presentarán varias técnicas avanzadas de programación funcional, patrones, bibliotecas y herramientas.

4. METODOLOGÍA DE ENSEÑANZA

Se darán 48hs. de clases teórico-prácticas divididas en veinticuatro clases de dos horas cada una, a razón de dos clases por semana.

Se espera que el estudiante le dedique 10 horas semanales de estudio.

Al final del curso los estudiantes realizarán presentaciones de artículos técnicos de revistas y conferencias y se realizará una prueba teórico-práctica.

Resumen de la dedicación esperada por parte del estudiante:

- 48 horas de clases teórico-práctica.
- 120 horas de estudio.
- 3 horas de la prueba final.
- 10 horas de preparación de las presentaciones, prueba y presentación.

5. TEMARIO

1. Desarrollo
    a. Ambiente de Desarrollo
    b. Corrección y Testing de Programas
    c. Profiling
2. Estructuras de Datos
    a. Arrays
    b. Coalgebras, Stream Fusion, ByteString
    c. Finger Tree
3. Lenguajes de Dominio Específico Embebidos
    a. EDSL
    b. Parsing
4. Efectos computacionales
    a. Mónadas, Transformadores de Mónadas
    b. Functores Aplicativos
5. Cálculo Lambda
    a. Cálculo Lambda Puro
    b. Cálculo Lambda Tipado
6. Tipos Avanzados
    a. GADTs
    b. Type Classes

6. BIBLIOGRAFÍA

Tema	Básica	Complementaria
Desarrollo	1, 2, 3	15, 16
Estructuras de Datos	4, 5, 6	15, 16
Lenguajes de Dominio Específico Embebidos	7	15
Efectos computacionales	8, 9, 10, 11	15, 16
Cálculo Lambda	12	
Tipos Avanzados	13, 14	17, 18, 19, 20

6.1 Básica

1. Hughes, John (1989). Why functional programming matters. Comput. J. 32, 2, 98–107.
2. Claessen, Koen and Hughes, John (2011). QuickCheck: a lightweight tool for random testing of Haskell programs. SIGPLAN Not. 46, 4 (April 2011), 53–64.
3. Gill, Andy and Runciman, Colin (2007). Haskell program coverage. In Proceedings of the ACM SIGPLAN workshop on Haskell workshop (Haskell '07). Association for Computing Machinery, New York, NY, USA, 1–12.

4. Coutts, Duncan, Leshchinskiy, Roman, and Stewart, Don (2007). Stream fusion: from lists to streams to nothing at all. SIGPLAN Not. 42, 9 (September 2007), 315–326.
5. Coutts, Duncan, Stewart, Don, and Leshchinskiy, Roman (2007). Rewriting haskell strings. In Proceedings of the 9th international conference on Practical Aspects of Declarative Languages (PADL'07). Springer-Verlag, Berlin, Heidelberg, 50–64.
6. Hinze, Ralf and Paterson, Ross, "Finger trees: a simple general-purpose data structure", Journal of Functional Programming 16:2 (2006) pp 197-217.
7. Gibbons, Jeremy. 2013. "Functional Programming for Domain-Specific Languages." In: Zsók, V., Horváth, Z., Csató, L. (eds) Central European Functional Programming School. CEFP 2013. Lecture Notes in Computer Science(), vol 8606. Springer, Cham.
8. Wadler, P. (1993). Monads for functional programming. In: Broy, M. (eds) Program Design Calculi. NATO ASI Series, vol 118. Springer, Berlin, Heidelberg.
9. Hutton, G., & Meijer, E. (1998). Monadic parsing in Haskell. Journal of Functional Programming, 8(4), 437-444.
10. McBride, Conor and Paterson, Ross (2008). Applicative programming with effects. J. Funct. Program. 18, 1 (January 2008), 1–13.
11. Liang, Sheng, Hudak, Paul, and Jones, Mark (1995). Monad transformers and modular interpreters. In Proceedings of the 22nd ACM SIGPLAN-SIGACT symposium on Principles of programming languages (POPL '95). Association for Computing Machinery, New York, NY, USA, 333–343.
12. Selinger, Peter (2018). Lecture Notes on the Lambda Calculus. ISBN 10: 0359158854 - ISBN 13: 9780359158850.
13. Cheney, James; Hinze, Ralf (2003). "First-Class Phantom Types". Technical Report CUCIS TR2003-1901. Cornell University.
14. Jones, Mark. (1992). A theory of qualified types. In Symposium proceedings on 4th European symposium on programming (ESOP'92). Springer-Verlag, Berlin, Heidelberg, 287–306.

6.2 Complementaria

15. Stewart, Donald Bruce y Goerzen, John (2008). Real World Haskell. Code You Can Believe In. Bryan O'Sullivan. O'Reilly Media. ISBN-13: 978-0596514983.
16. Bird, Richard (1998). Introduction to Functional Programming using Haskell. Prentice Hall ISBN-13: 978-0134843469
17. Haskell 98 Language and Libraries: The Revised Report. Simon Peyton Jones (editor). Cambridge University Press. ISBN-13: 978-0521826143. Diciembre 2002

18. Pierce, Benjamin (2002). Types and Programming Languages. MIT Press. ISBN-13: 978-0262162098.

19. Pierce, Benjamin (2004). Advanced Topics in Types and Programming Languages. MIT Press. ISBN-13: 978-0262162289.

20. Fun of Programming. Jeremy Gibbons y Oege de Moor (editores). Palgrave Macmillan. ISBN-13: 978-0333992852. Marzo 2003

21. En el curso se va a hacer referencia a otros artículos de investigación.

7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS

7.1 Conocimientos Previos Exigidos: Programación Funcional

7.2 Conocimientos Previos Recomendados: Lógica.

ANEXO A
Para todas las Carreras

Esta primera parte del anexo incluye aspectos complementarios que son generales de la unidad curricular.

A1) INSTITUTO

Instituto de Computación

A2) CRONOGRAMA TENTATIVO

Semana 1	Introducción. Ambiente de Desarrollo.
Semana 2	Corrección y Testing de Programas.
Semana 3	Profiling. Listas y Arreglos.
Semana 4	Coalgebras, Stream Fusion.
Semana 5	ByteString. Finger Tree.
Semana 6	EDSLs. Parsing.
Semana 7	Mónadas. Ejemplos de Mónadas.
Semana 8	Functores Aplicativos
Semana 9	Composición de Functores Aplicativos. Transformadores de Mónadas.
Semana 10	Cálculo Lambda Puro.
Semana 11	Cálculo Lambda Tipado.
Semana 12	GADTs. Type Classes.
Semana 13	Presentaciones de los estudiantes.
Semana 14	
Semana 15	

A3) MODALIDAD DEL CURSO Y PROCEDIMIENTO DE EVALUACIÓN

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.

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

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. Sobre el final del curso se realizarán presentaciones de artículos técnicos por parte de los estudiantes. Las clases de presentaciones serán por fuera de las 24 clases teórico-prácticas.

Se recomienda al estudiante dedicar, en promedio, 10 horas semanales complementarias de estudio, incluyendo la realización de ejercicios prácticos y la lectura de artículos.

Al finalizar el curso se realizará una prueba teórico-práctica de 3 horas de duración.

La preparación de la presentación y de la prueba insumirá alrededor de 10 horas.

Las instancias de evaluación se ponderan de la siguiente forma: 60% prueba y 40% prácticos, presentaciones y participación en clase. En todos los casos hay un mínimo de aprobación de un 60%.

A4) CALIDAD DE LIBRE

Esta unidad curricular no adhiere a la calidad de libre.

A5) CUPOS DE LA UNIDAD CURRICULAR

No tiene

ANEXO B para la carrera Ingeniería en Computación (Plan 1997)

B1) ÁREA DE FORMACIÓN

Programación

B2) UNIDADES CURRICULARES PREVIAS

Curso: Curso aprobado de Introducción a la Programación Funcional (1328) o curso aprobado de Programación Funcional (1354),

Examen: No corresponde.

ANEXO B para la carrera Licenciatura en Computación (Plan 2012)

B1) ÁREA DE FORMACIÓN

Programación

B2) UNIDADES CURRICULARES PREVIAS

Curso: Curso aprobado de Introducción a la Programación Funcional (1328) o curso aprobado de Programación Funcional (1354),

Examen: No corresponde.

ANEXO B para la carrera Ingeniería en Computación (Plan 1987)

B1) ÁREA DE FORMACIÓN

No corresponde

B2) UNIDADES CURRICULARES PREVIAS

Curso: Previas comunes a las electivas técnicas.

Examen: No corresponde.

Esta unidad curricular vale por una electiva técnica.