PROGRAMA de Programación Funcional Avanzada

Carreras: Tecnicatura Universitaria en Programación Informática - Licenciatura en Informática

Asignatura: Programación Funcional Avanzada

Núcleo al que pertenece: Orientación

Profesor: Pablo E. Martínez López

Asignaturas Correlativas: Programación Funcional

Objetivos:

Que la persona que cursa la materia:

- Aprenda un abanico de técnicas y prácticas modernas sobre el desarrollo de software en lenguajes de programación puramente funcionales y avanzados.
- Pueda utilizar herramientas de sistemas de tipos avanzados, principalmente de lenguajes de programación funcionales.
- Aprenda a diseñar software mediante un análisis basado en las propiedades de los tipos de datos que el programa manipula.
- Comprenda la relevancia de las propiedades matemáticas de los datos y funciones en el diseño de software.

Contenidos mínimos:

- Tipos algebraicos, polimorfismo paramétrico, alto orden, typeclasses. Lazy evaluation, memoization.
- Sistema de tipos Hindley–Milner y variantes más avanzadas. Recursión polimórfica. Rank-N Types. Nociones básicas sobre sistemas de tipos dependientes.
- Técnicas funcionales de parsing. Parsing combinators, parsing mónadico.
- Tipos de recursión. Fold, unfold. Anamorfismos, catamorfismos, hilomorfismos. Fixed-point combinator.

- Estructuras algebraicas. Functors, Applicative Functors, Mónadas, Arrows, Lenses. Monad Transformers, Free monads, F-Algebras.
- Programación funcional reactiva y su uso en el diseño de interfaces gráficas.
- Testing automatizado mediante pruebas aleatorias sobre propiedades (Quickcheck).

Carga horaria semanal: 4 hs

Programa analítico:

Unidad 1
Introducción a las construcciones y propiedades de lenguajes puramente funcionales: tipos algebraicos, polimorfismo, alto orden, lazy evaluation.

Unidad 2
Sistemas de tipos basados en Hindley-Milner.

Unidad 3
Técnicas funcionales de parsing.

Unidad 4
Tipos de recursión y sus propiedades y características particulares.

Unidad 5
Definición y uso de estructuras algebraicas utilizadas en la programación funcional.

Unidad 6:
Programación funcional Reactiva. Implementación de interfaces de usuario utilizando técnicas funcionales.

Unidad 7
Técnicas de testing sobre programas puramente funcionales. Quickcheck.

Bibliografía obligatoria:

- Lipovaca, M. (2011). Learn you a Haskell for great good!: A beginner's guide. No Starch Press. ISBN-10: 1593272839; ISBN-13: 978-1593272838
- Allen, C., & Moronuki, J. (2017). Haskell Programming from First Principles. Gumroad (ebook). URL: http://haskellbook.com/
- Bird, R. & Wadler, P. (1988). Introduction to Functional Programming. Prentice Hall International (UK). ISBN:0-13-484189-1.
- Bird, R. (2014). Thinking functionally with Haskell. Cambridge University Press. ISBN-10: 1107452643; ISBN-13: 978-1107452640

Bibliografía de consulta:

- Michaelson, G. (2011). An introduction to functional programming through lambda calculus. Dover Publications. ISBN-10: 9780486478838; ISBN-13: 978-0486478838

Organización de las clases:

Las clases serán en modalidades teórico-práctico dependiendo del tema a desarrollar. Se dará a quienes cursan ejercicios para realizar con el objetivo de asentar los conceptos trabajados en clase. Algunos de estos ejercicios formarán parte de su evaluación.
Durante las últimas 6 semanas de clases, se trabajará con las personas cursando en un trabajo práctico grupal, con entregas semanales incrementales, donde tendrán oportunidad de poner en práctica los temas desarrollados durante el curso de la materia. El trabajo finalizará con una exposición de cada grupo frente al resto de sus compañeres.

Trabajos Prácticos

Práctica 1: **Repaso**
Los objetivos de esta práctica son repasar las características de las construcciones y propiedades de lenguajes puramente funcionales: tipos algebraicos, polimorfismo, alto orden, lazy evaluation.

Práctica 2: Sistemas de tipos
Los objetivos de esta práctica son conseguir familiaridad en la inferencia de tipos en el sistema Hindley-Milner y sus límites y posibles extensiones (recursión polimórfica,

rank-N types, etc.), y presentar y practicar otros sistemas de tipos, tales como System F (Lambda cálculo de 2do orden), sistemas de tipos dependientes, etc.

Práctica 3: Combinadores de parsing
Los objetivos de esta práctica son conocer los combinadores de parsing más utilizados, y confeccionar pequeños parsers para lenguajes simples, utilizando las técnicas funcionales.

Práctica 4: Álgebra de la recursión
Los objetivos de esta práctica son profundizar en las propiedades y características particulares de los diferentes tipos de recursión: anamorfismos, catamorfismos, hilomorfismos, paramorfismos y presentar el combinador de punto fijo y su utilización para expresar las recursiones anteriores.

Práctica 5: Estructuras algebraicas
Los objetivos de esta práctica son introducir la definición y uso de estructuras algebraicas utilizadas en la programación funcional: Functors, Applicative Functors, Mónadas, Arrows, Lenses. Monad Transformers, Free monads, F-Algebras.

Práctica 6: Programación funcional reactiva
Los objetivos de esta práctica son introducir la programación funcional reactiva e implementación de interfaces de usuario utilizando técnicas funcionales.

Práctica 7: Generación de testings automatizados
Los objetivos de esta práctica son presentar técnicas de testing automatizado sobre programas puramente funcionales y presentación de Quickcheck.

Modalidad de evaluación:

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los artículos del Régimen de estudios de la UNQ (Res. CS 201/18). En la modalidad de libre se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral, un desarrollo de programa en el entorno de trabajo e instancias de evaluación similares a las realizadas en la modalidad presencial.

CRONOGRAMA TENTATIVO

Semana	Tema/unidad	Actividad*				Evaluación
		Teórico	Práctico			
			Res Prob.	Lab.	Otros Especificar	
1	Conceptos básicos de programación funcional	x		x		
2	Sistema de tipos Hindley-Milner	x				
3	Sistemas de tipos avanzados	x			Dinámica	
4	Sistemas de tipos dependientes	x	x			
5	Tipos de Recursión: fold, recursión primitivia, unfold	x	x			
6	Fixed-point combinator	x	x			
7	Estructuras algebraicas simples	x	x			
8	Estructuras algebraicas avanzadas	x				
9	Monad Transformers			x	Dinámica	
10	Técnicas funcionales de parsing			x		
11	Programación Funcional Reactiva	x		x		
12	Free Monads		x			
13	QuickCheck			x		
14	Seguimiento de trabajos prácticos		x			
15	Clase de repaso					
16	Evaluación					x
17	Actividad de retrospectiva				Dinámica	

***INDIQUE CON UNA CRUZ LA MODALIDAD**

Firmado digitalmente por: SUAREZ Mariana Alejandra
Directora del Departamento de Ciencia y Tecnologla
UNIVERSIDAD NACIONAL DE QUILMES