

## Curso avanzado sobre abordaje funcional a EDSLs

Dr. Alberto Pardo (Profesor Visitante FCEN-UBA) con la colaboración del Dr. Pablo Barenbaum (Profesor Interino Adjunto)

### Programa:

El objetivo de este curso es introducir a los estudiantes a los conceptos de Lenguajes de Dominio Específico (DSL) y lenguajes de Dominio Específico Embebidos (EDSL), y presentar una serie de técnicas para el diseño e implementación de este tipo de lenguajes desde una perspectiva de programación funcional.

### Temario:

- Introducción a los Lenguajes de Dominio Específico (DSL) y lenguajes de Dominio Específico Embebidos (EDSL) en el contexto de lenguajes funcionales en general y Haskell en particular.
- Beneficios y clasificación de los DSLs y EDSLs.
- Integración de EDSLs a lenguajes de propósito general. Ejemplos de EDSLs: QuickCheck, Finger Trees, Arrays, Streams, HaXml, Lava, Parsec, Haskore.
- Modelado de EDSLs a través de abordajes interno (shallow embedding) y externo (deep embedding).
- Ejemplos de shallow embedding: analizadores sintácticos funcionales.
- Ejemplos de deep embedding: lenguajes de expresiones.
- EDSLs fuertemente tipados y extensiones al sistema de tipos de Haskell: polimorfismo, tipos fantasma, tipos dependientes, tipos existenciales, tipos de datos algebraicos generalizados (GADTs).
- Clases de tipos en Haskell: funtores, funtores aplicativos, mónadas. Secuenciación del control y modelado de efectos.
- Uso de clases de tipos para modelar EDSLs. Combinadores de parsers, parsers aplicativos y parsers monádicos.
- Reporte de errores en EDSLs.

### Bibliografía:

- Gibbons, J. (2015). Functional Programming for Domain-Specific Languages. In: Zsóck, V., Horváth, Z., Csató, L. (eds) Central European Functional Programming School. CEFP 2013. Lecture Notes in Computer Science(), vol 8606. Springer, Cham. [https://doi.org/10.1007/978-3-319-15940-9\\_1](https://doi.org/10.1007/978-3-319-15940-9_1)
- McBride, Conor and Paterson, Ross (2008). Applicative programming with effects. J. Funct. Program. 18, 1 (January 2008), 1–13.
- Wadler, P. (1995). Monads for Functional Programming. Advanced Functional Programming (p./pp. 24–52), London: Springer. ISBN: 3-54059451-5
- Graham Hutton and Erik Meijer. 1998. Monadic parsing in Haskell. J. Funct. Program. 8, 4 (July 1998), 437–444. <https://doi.org/10.1017/S0956796898003050>
- García-Garland, J., Pardo, A., Viera, M. (2019). Attribute grammars fly first-class... safer!: dealing with DSL errors in type-level programming. In Stutterheim, J. and Chin, W., editors, IFL '19: Implementation and Application of Functional Languages, Singapore, September 25-27, 2019, pages 10:1–10:12. ACM.
- Kiselyov, O. (2010). Typed Tagless Final Interpreters. SSGIP 2010: 130-174

- Huttner, L., Merigoux, D. (2022) "Catala: Moving Towards the Future of Legal Expert Systems", Artif. Intell. Law.