

# Programación Funcional Avanzada

Instituto de Computación, Facultad de Ingeniería  
Universidad de la República, Uruguay

curso 2023

# ¿Por qué Programación Funcional?

- Novedad
  - programación desde una perspectiva diferente
  - pensar en funciones en lugar de modificación de datos
  - distinción fuerte entre código puro y comunicación con el “mundo real”
- Poder expresivo
  - código más corto, en menos tiempo y con menos bugs
  - tipos y polimorfismo
  - funciones de alto orden
  - evaluación perezosa
- Disfrute
  - concentrarse en la parte interesante del código
  - una forma completamente distinta de programar
  - mayor relación con conceptos matemáticos

Leer **“Why functional programming matters”**

# Uso en la industria

Algunos ejemplos:

- ABN AMRO
- AT&T
- Credit Suisse
- Deutsche Bank
- Galois
- Well-Typed
- Tweag I/O
- Facebook
- Google
- Intel
- Microsoft
- IOHK

Más en:

[http://www.haskell.org/haskellwiki/Haskell\\_in\\_industry](http://www.haskell.org/haskellwiki/Haskell_in_industry)

<https://haskellcosm>

Al finalizar el curso se debería:

- poder utilizar varias herramientas y bibliotecas de Haskell
- saber como estructurar y escribir programas de mediano y gran porte
- poder entender artículos de investigación sobre conceptos de programación funcional

- El poder de la programación funcional:
  - funciones de alto orden
  - tipos y polimorfismo
  - evaluación perezosa
- Herramientas para programación funcional
- Técnicas de programación
- Bibliotecas
- Conceptos avanzados y experimentales

Dos clases teórico/prácticas semanales

- Martes de 9:00 a 11:00 hs
- Viernes de 9:00 a 11:00 hs

Clases usualmente basadas en uno o más papers

- Leer antes de la clase.

Tres grandes bloques

- Desarrollo
- Bibliotecas
- Lenguaje

Todo lo necesario para desarrollar proyectos Haskell

- Ambiente de desarrollo
- Administración de paquetes
- Herramientas útiles
- Control de versiones
- Testeo y debugging
- Documentación
- Técnicas de programación

## Usando Haskell para problemas del mundo real

- Estructuras de datos
- Patrones de Diseño
- Mónadas, Functores aplicativos
- Combinadores de parsing

## Conceptos avanzados de lenguajes funcionales

- Inferencia de tipos
- Módulos
- Type classes avanzadas
  - Múltiples parámetros
  - Dependencias funcionales
  - Tipos asociados
- Tipos de datos avanzados
  - Kinds
  - Existenciales
  - GADTs

Componentes:

- Prueba Escrita
- Prácticos entregables
- Presentación
- Participación en clase

Los cuatro componentes influyen en la calificación final.

- Individual
- Incluye todo lo visto en las clases y prácticos
- Se permite la consulta de material

- Grupos de 1 o 2 estudiantes
- Aproximadamente bisemanales
- Formados por ejercicios prácticos, algunos de programación

- Grupos de 1 o 2 estudiantes
- Se tocarán tópicos como:
  - herramientas
  - bibliotecas
  - conceptos del lenguaje
  - otros lenguajes
  - etc
- Se proveerá una lista de posibles temas
- Se aceptarán sugerencias de los estudiantes

- GHC  
<http://www.haskell.org/ghc/>
- Compilador: *ghc*
- Intérprete: *ghci*
- GHC soporta **Haskell 2010** y una gran variedad de **extensiones**
- Se puede experimentar con cualquier extensión
- Intente NO utilizar las operaciones *unsafeXX* (al menos en los ejercicios de práctico).

## Curso

- EVA del curso

<https://eva.fing.edu.uy/course/view.php?id=965>

## Haskell

- Listas de correo

[http://www.haskell.org/haskellwiki/Mailing\\_Lists](http://www.haskell.org/haskellwiki/Mailing_Lists)

- Canales IRC

[http://www.haskell.org/haskellwiki/IRC\\_channel](http://www.haskell.org/haskellwiki/IRC_channel)