

PROGRAMACIÓN EN HASKELL

Septiembre, 2016

Instructora: Andrea Gómez

Email: hi@daedra.ml

Temario:

1. Introducción a Haskell
 - (a) Lenguajes de programación
 - i. Funcionales e imperativos.
 - ii. Sistema de tipos.
 - (b) Haskell
 - i. ¿Qué es Haskell?
 - ii. ¿Por qué usar Haskell?
 - iii. ¿Quién usa Haskell?
 - iv. Lo necesario para programar en Haskell.
2. Usando GHCi
 - (a) Operadores
 - i. Aritméticos.
 - ii. Álgebra Booleana.
 - (b) Funciones
 - i. Tipos de funciones
 - A. Prefijas
 - B. Infijas
 - ii. Funciones predefinidas
 - iii. Funciones en tiempo de ejecución
 - iv. Funciones desde un archivo
3. Listas
 - (a) Introducción a las listas
 - i. Definición de lista
 - ii. Creando listas
 - iii. Operaciones con listas
 - iv. Listas anidadas
 - v. Funciones básicas en listas
 - (b) Rangos texanos
 - i. Utilidad de los rangos texanos
 - ii. Orden de los rangos

- iii. Listas infinitas

- (c) Listas intencionales

- i. Similitud con conjuntos definidos de forma intensiva
- ii. Dentro de funciones
- iii. Listas intencionales con múltiples predicados
- iv. Funciones con múltiples listas intencionales
- v. Listas intencionales y rangos texanos

4. Tuplas

- (a) Introducción a las tuplas

- i. Definición de tupla
- ii. Diferencia entre tupla y lista
- iii. Ventajas de tuplas frente a listas

- (b) Trabajando con tuplas

- i. Tipos de tupla
- ii. Listas unitarias, tuplas unitarias
- iii. Comparación de tuplas
- iv. De tuplas a listas

5. Tipos y clases de tipos

- (a) Introducción a tipos

- i. Inferencia de tipos
- ii. Examinando tipos en GHCi
- iii. Tipos de funciones
- iv. Tipos comunes
- v. Tipos de tuplas

- (b) Variables de tipo

- i. Tipos y variables de tipo
- ii. Funciones polimórficas

- (c) Clases de tipo

- i. Introducción a las clases de tipos
- ii. Diferencia de clases entre lenguajes de programación funcionales y orientados a objetos
- iii. Restricción de clase
- iv. Clases de tipos básicas
 - A. Eq
 - B. Ord
 - C. Show
 - D. Read
 - E. Enum
 - F. Bounded
 - G. Num

- H. Integral
- I. Floating

6. Sintaxis de funciones

- (a) Correspondencia de patrones
 - i. Introducción a la correspondencia de patrones
 - ii. Tipos de datos válidos
 - iii. Ventaja de correspondencia de patrones frente a árboles if then else
 - iv. Recursión básica
 - v. Correspondencia de patrones y otros tipos de datos
 - vi. Correspondencia de patrones y tuplas
 - vii. Correspondencia de patrones y listas intencionales
 - viii. Implementando correspondencia de patrones
- (b) Guardas
 - i. Introducción a las guardas
 - ii. Aplicación de las guardas
 - iii. otherwise.otherwise
 - iv. Guardas y funciones que reciben parámetros
 - v. Ligando valores a variables para usarse en expresiones
 - vi. Múltiples "where"

Instalando lo necesario en Arch Linux: Recomiendo encarecidamente llevar instalados los siguientes paquetes antes de comenzar el curso para poder hacer un uso eficiente del tiempo. Si no usan Arch Linux, bastará con adecuar los comandos al gestor de paquetes usado por su distribución preferida:

```
1 $ sudo pacman -S ghc
```

Si bien GHC es todo lo que necesitaremos pues el curso es introductorio, recomiendo instalar los siguientes paquetes en adición a GHC para quien desee adentrarse al hermoso mundo de Haskell.

```
1 $ sudo pacman -S cabal-install haskell-haddock-api haskell-haddock-library happy alex  
   emacs
```

Recomiendo también los siguientes paquetes para Emacs pues enriquecerán el ya increíble editor de texto Emacs. Solo requieren tener habilitado el repositorio Melpa para instalarlos dentro de Emacs y posteriormente, editar el archivo .emacs para inicializarlos.

```
1 M-x package-install RET haskell-mode  
2 M-x package-install RET auto-complete  
3 M-x package-install RET ac-haskell-process
```