

Guía para el entorno de programación funcional ghc

En el primer bloque de la materia vamos a estar trabajando con programas funcionales, es decir, basados en funciones. Para ello vamos a trabajar con el lenguaje funcional haskell (para saber más, vean el [artículo sobre haskell en la wikipedia](#)).

Para trabajar haskell lo primero que vamos a tener que hacer es instalar haskell en nuestra computadora. En las computadoras del laboratorio de FaMAF ya lo tenemos instalado. Para instalarlo en su casa, pueden seguir la guía para instalar [ghc](#) en su computadora personal que se encuentra más abajo.

Para trabajar con haskell vamos a necesitar un **intérprete**, es decir, un programa que entiende haskell, por lo tanto, puede entender y ejecutar los programas que vamos a hacer en haskell. En este curso vamos a estar usando el entorno de programación en haskell [ghc](#) y su intérprete. Otros años habíamos trabajado con el intérprete hugs, si les interesa pueden ver [algunas clases del taller de años anteriores](#).

Cómo vamos a trabajar con haskell? Veamos un paso a paso:

1. Lo primero que vamos a hacer es llamar al intérprete de haskell. En el entorno ghc, ese intérprete se llama ghci. Hay que ejecutar una terminal y de ahí hay que ejecutar el comando ghci. Al hacer esto, entraremos dentro del programa ghci, que es un intérprete de haskell, es decir, un programa que entiende y ejecuta otros programas escritos en haskell. El entorno de este programa se ve así:

```
user@lab60:~$ ghci
GHCi, version 6.10.4: http://www.haskell.org/ghc/ :? for help
Loading package ghc-prim ... linking ... done.
Loading package integer ... linking ... done.
Loading package base ... linking ... done.
Prelude>
```

2. El intérprete de haskell tiene muchos parecidos con la terminal. Al igual que la terminal, tiene un prompt (en este caso, `Prelude>`) que identifica en qué situación estamos, y que es donde el intérprete espera que escribamos nuestros comandos. Podemos pensarlo como que el intérprete de haskell es una calculadora, y el prompt es su entrada, su invitación a que introduzcamos una expresión para evaluarla e imprimir el resultado. Después, nuestra calculadora queda lista para volver a pedir una expresión. Cuando recién ejecutamos ghci, el prompt dice `Prelude>`, porque como no hemos cargado ningún programa, únicamente disponemos de los programas (o funciones) que vienen preestablecidos en el [Preludio Estándar](#) de haskell. Gracias al preludio estándar tenemos muchas funciones ya definidas, entre ellas, todas las aritméticas y las lógicas básicas. Pero también tenemos otras muchas, como por ejemplo "reverse". Veamos algunos ejemplos de cosas que podemos hacer con las funciones que vienen incluidas en el preludio:

```
Prelude> 2+2
4
```

```

Prelude> reverse [1,2,3]
[3,2,1]
Prelude> True && False
False
Prelude> 2*4 == 10 `div` 5 && True
False
Prelude> reverse2 "dabale arroz a la zorra el abad"
"daba le arroz al a zorra elabad"
Prelude>

```

- Podemos salir del intérprete con `CTRL-D` o con el comando `":q"`. Así volvemos a la terminal.
- Para poder dar nuevas definiciones y/o funciones, además de las que se encuentran en el preludio estándar, podemos escribir programas en archivos separados, programas que luego cargaremos en el intérprete. Para escribir programas, tenemos que ejecutar un editor de textos cualquiera. Por ejemplo, podemos abrir otra terminal y ejecutar `kate`, con el nombre del archivo donde vamos a escribir nuestros programas:

```
kate hola.hs
```

- En este editor podemos escribir un programa cualquiera, por ejemplo, el programa `sgn`, de la siguiente forma:

```

sgn :: Int -> Int      -- esta línea es la signatura de tipos de la
función sgn
sgn x | 0<x  = 1
      | x<0  = -1
      | x==0 = 0

```

- Una vez que tenemos escrito el programa, lo guardamos **en el mismo directorio donde ejecutamos ghci**, y volvemos a `ghci`.
- En `ghci`, ejecutamos el comando `:l` mas el nombre del archivo para cargar el archivo y poder usar los programas que hemos escrito en él:

```

Prelude> :l hola.hs
[1 of 1] Compiling Main          ( hola.hs, interpreted )
Ok, modules loaded: Main.
*Main>

```

- Una vez que tenemos cargado el archivo con los programas sin problemas, el prompt cambia: ahora dice `*Main>`, e incluye todos los programas del preludio estándar y además todos los programas que están en el archivo que acabo de cargar.

Cómo ejecutar una terminal

- en la barra de menús del sistema (arriba o abajo de la pantalla) vamos al menú Aplicaciones, accedemos al submenú Accesorios y de ahí seleccionamos el item

Terminal.

2. cuando hayamos seleccionado Terminal, nos aparecerá una ventana con la clásica barra de menús arriba y como único contenido, una línea de texto y un cursor que titila:

Esta ventana es la terminal, y la línea de texto que aparece es el `prompt`. El `prompt` nos dice el usuario que somos, luego, separado por el símbolo `@`, el nombre de la máquina en la que estamos y luego, separado por dos puntos (`:`), el directorio en el que estamos. En el ejemplo, el nombre de la máquina es `debian-pica`, y el directorio es el directorio raíz, que se simboliza con `~`. Pero si cambiamos a otro directorio nos aparecerá distinto:

```
user@debian-pica:~$ cd haskell
```

```
user@debian-pica:~/haskell$
```

El símbolo `$` (o algún otro símbolo en diferentes versiones de la terminal) indica el final del `prompt`. Ahí es donde el usuario (o sea, vos) puede empezar a escribir comandos para realizar acciones. Después de cada comando, se presiona la tecla `enter` para que la terminal ejecute el comando. En la [página de la facultad](#) podrán encontrar una [lista de básica de comandos de linux](#).

Para cualquier otro problema técnico, pueden consultar esta [lista de preguntas frecuentes sobre cuestiones informáticas en los laboratorios de la facultad](#).

Guía para instalar haskell (ghc) en tu computadora personal

Para instalar `ghc` en su computadora, pueden ir a la [página de bajadas de ghc](#), http://www.haskell.org/ghc/distribution_packages.html. Ahí pueden seleccionar el sistema operativo que tengan en su computadora, por ejemplo, seleccionan [ghc para Ubuntu](#).

Para instalar `ghc` en Ubuntu, podemos usar el gestor de paquetes [Synaptic](#), e instalar los paquetes **`ghc6`**, **`ghc6-prof`** y **`ghc6-doc`**.

1. Para iniciar Synaptic navega a **Sistema -> Administración -> Gestor de Paquetes Synaptic**.
2. La ventana principal está dividida en tres secciones: un explorador de paquetes a la izquierda, la lista de paquetes en la parte superior derecha, y los detalles del paquete en la parte inferior derecha. La barra de estado muestra el estado del sistema de un vistazo.
3. Para instalar `ghc`, haz clic con el botón derecho en los paquetes **`ghc`**, **`ghc6-prof`** y **`ghc6-doc`** y selecciona **Marcar para instalar** en el menú contextual, o presiona **Ctrl + I**. Si el paquete requiere la instalación de otros paquetes, aparecerá

una ventana de diálogo. Haz clic en **Marcar** para permitir la instalación de paquetes adicionales.

4. Una vez estés satisfecho con tu selección, haz clic en **Aplicar** en la barra de herramientas o presiona **Ctrl + P**. Aparecerá otra ventana de diálogo con un resumen de los cambios que se van a aplicar. Confirma los cambios haciendo clic en **Aplicar**
5. Para saber más sobre Synaptic, vean la [página de documentación sobre Synaptic](#) en castellano.

[Ubuntu](#) es la misma distribución de linux que tienen en la mayor parte de computadoras del laboratorio. Se trata de un sistema operativo libre, de código abierto y es muy fácil de instalar. En una misma computadora se puede instalar windows y linux (en este caso, ubuntu). Para saber cómo instalar ubuntu en una computadora, pueden consultar [alguno de los muchos blogs sobre el tema](#) o tratar por su cuenta a partir de las instrucciones en la [web de descargas oficial de ubuntu](#)