Tutorial Introductorio a Cabal

San Simon Haskell Hackathone

Carlos Gomez
carliros.g@gmail.com

Abril, 2011

Motivacion

- Experiencia: Hace mucho tiempo, quise probar el juego Mario Bross hecho en haskell.
  No pude hacerlo, porque no sabia como instalarlo.
- Desarrollas una aplicacion y quieres compartirlo.

  Reflexion:
  "La facilidad de instalacion puede facilmente incrementar la adopcion de software en los usuarios destinos"

Make & Cabal al rescate

Make es una herramienta estandard de unix/linux el cual determina que partes del Software deben ser recompilados, tambien colecciona los comandos que usara para recompilarlos.

Cabal provee una arquitectura comun que puede ser utilizada para constuir e instalar una herramienta desarrollada en Haskell (ejecutables y librerias).

Make & Cabal al rescate

El archivo Makefile puede ser simple, o tambien complejo:

```
all: uuagc
  ghc --make -i$(SRC) -o s2hk $(SRC)Main.hs
  -outputdir $(OUT)
uuagc:
  uuagc --data --catas --semfuns -signatures $
  (SRC)Estructura.ag -P $(SRC)
```

Y muy utilizable, especialmente en el desarrollo de Software:

```
make configure
make build
make install
```

Make & Cabal al rescate

Pero, hoy aprenderemos sobre: 

Ademas, teoricamente Cabal es mucho mas poderoso que Make, porque:

- Provee un archivo meta-data de descripcion de paquete.
- Es estandard en todas las plataformas donde esta Haskell. No asi con Make (~solo similares a unix).
- Facil de escribir un archivo.
- Su arquitectura comun permite manejar distintas herramientas.
- Provee resolucion de dependencias de paquetes, muy util al instalar los paquetes.
- .....

Distribuccion de Software con Cabal

- Necesitamos 2 archivos:
  - Setup.hs o Setup.lhs
  - NombrePaquete.cabal
- El contenido de Setup.hs normalmente es:

```
import Distribution.Simple

main :: IO()
main = defaultMain
```

“This module isn't called "Simple" because it's simple. Far from it. It's called "Simple" because it does complicated things to simple software.” Cabal's Documentation

Descripcion del Paquete con Cabal

Ejemplo:

```
Name                 : pkgName
Version              : 1.0
Build-Depends        : OtherPackage >= 1.1
Copyright            : (c) 2005 Isaac Jones
License              : BSD3
Maintainer           : Isaac Jones
Synopsis             : Brief description.
Exposed-Modules      : MyPackage
Other-Modules        : MyPackage.SubModule ,
                       MyPackage.OtherModule
C-Sources            : cfile1.c, c_src/cfile2.c
Extensions           : ForeignFunctionInterface
Executable packageExe
    Main-is          : Main.hs
    Other-Modules    : MyPackage
    Extensions       : OverlappingInstances
```

Descripcion del Paquete con Cabal

- El archivo `NombrePaquete.cabal` es la informacion meta-data de descripcion de nuestro software.
- Puede estar dividido en 3 partes:
  - Descripcion generica
  - Descripcion de Libreria
  - Descripcion de Ejecutale
- En cada parte tiene campos claves permitidos.

Descripcion del Paquete con Cabal

- La descripcion generica puede tener los campos:

```
name version cabal-versionbuild-
type license  license-file
copyright  maintainer build-depends
  stability   homepage package-url
 bug-reports    synopsis
description   category   author
tested-with data-files data-dir
extra-source-files extra-tmp-files
```

Descripcion del Paquete con Cabal

- La descripcion de libreria puede tener los campos:

```
exposed-modules exposed buildable
build-tools cpp-options cc-options
ld-options pkgconfig-depends
frameworks c-sources extensions
extra-libraries extra-lib-dirs
includes install-includes
include-dirs  hs-source-dirs
other-modules ghc-prof-options ghc-
shared-options ghc-options hugs-
options nhc98-options jhc-options
```

Descripcion del Paquete con Cabal

- La descripcion de ejecutable puede tener los campos:

```
executable main-is buildable
build-tools cpp-options cc-options
ld-options pkgconfig-depends
frameworks c-sources extensions
extra-libraries extra-lib-dirs
includes install-includes
include-dirs hs-source-dirs
other-modules ghc-prof-options
ghc-shared-options ghc-options
hugs-options nhc98-options
jhc-options
```

Crear los archivos de descripcion

- Se puede crear de forma manual.
- Tambien se puede usar la herramienta “cabal” Quiza es la forma mas sencilla de hacer:

```
cabal init
```

- Luego modificamos los archivos generados a nuestro gusto.

Pruebas con “cabal init”

```
-> cabal init
Package name [default "carlos"]?
Package version [default "0.1"]?
Please choose a license:
   1) GPL      2) GPL-2    3) GPL-3    4) LGPL
    5) LGPL-2.1 6) LGPL-3  *7) BSD3     8) BSD4
   9) MIT       10) PublicDomain    11) AllRightsReserved
  12) OtherLicense 13) Other (specify)
Your choice [default "BSD3"]?
Author name? carlos
Maintainer email? carlos@gmail.com
Project homepage/repo URL? comunidadhaskell.org
Project synopsis? test de cabal init
Project category:
   1) Codec  2) Concurrency   3) Control        4) Data
   5) Database  6) Development  7) Distribution  8) Game
   9) Graphics  10) Language   11) Math  12) Network
  13) Sound   14) System    15) Testing   16) Text
  17) Web     18) Other (specify)
Your choice? 18
Please specify? CabalTest
What does the package build:
   1) Library        2) Executable
Your choice? 2
Generating LICENSE...
Generating Setup.hs...
Generating hcal.cabal...
```

Algunas recomendaciones

- Si hay un error, no es el fin del mundo!! Puede modificar tus archivos y intentar de nuevo.

```
Distribution quality warnings:
    No 'category' field.
    No 'synopsis' field.
Warning: Cannot run preprocessors.
Run 'configure' command first.
Building source dist for SimpleLenguajeLogo-1.1...
Source tarball created: dist/SimpleLenguajeLogo-1.1.tar.gz
```

```
[carlos@cathrina slogo]$ runhaskell Setup.hs configure
Configuring SimpleLenguajeLogo-1.1...
[carlos@cathrina slogo]$
```

Algunas recomendaciones

No conoces alguna dependencia?

```
[carlos@cathrina slogo]$ runhaskell Setup.hs build
Preprocessing executables for SimpleLenguajeLogo-1.1...
Building SimpleLenguajeLogo-1.1...

src/Procesar.hs:14:7:
    Could not find module `Graphics.SOE':
      It is a member of the hidden package `HGL-3.2.0.2'.
      Perhaps you need to add `HGL' to the build-depends in your .cabal file.
      Use -v to see a list of the files searched for.
[carlos@cathrina slogo]$
```

Intenta hacer Prueba y Error. En el Error te muestra lo que te falta, Modificas el .cabal y vuelves a probar.

Instalacion de Paquetes

- Probablemente la forma mas sencilla:

```
cabal instal paquete
```

- Forma manual:

```
runhaskell Setup.hs configure
runhaskell Setup.hs build
runhaskell Setup.hs install
```

- Para documentacion:

```
runhaskell Setup.hs haddock
```

- Coloreado de sintaxis

```
runhaskell Setup.hs hscolour
```

Instalacion de Paquetes

- Mas comandos?

configure	Preparar el paquete.
build	Contruir el paquete.
install	Instalar el paquete (copy y register).
copy	Copia a una ubicacion.
haddock	Genera documentacion.
clean	Limpia.
sdist	Genera el paquete para su distribucion.
hscolour	Coloreado de sintaxis.
register	Registrar la libreria en el compilador.
unregister	Desregistrar.
test	
help	Help de los comandos.

Pruebas de instalacion

```
[carlos@cathrina dbjava]$ runhaskell Setup.hs configure
Configuring dbjava-1.7...
[carlos@cathrina dbjava]$ runhaskell Setup.hs build
Preprocessing library dbjava-1.7...
Preprocessing executables for dbjava-1.7...
Building dbjava-1.7...
[1 of 3] Compiling Jvm.Data.ClassFormat ( src/Jvm/Data/ClassFormat.hs, dist/build/Jvm/Data/ClassFormat.o
)
[2 of 3] Compiling Jvm.BinaryClass  ( src/Jvm/BinaryClass.hs, dist/build/Jvm/BinaryClass.o )
[3 of 3] Compiling Jvm.PrettyClass  ( src/Jvm/PrettyClass.hs, dist/build/Jvm/PrettyClass.o )
Registering dbjava-1.7...
[1 of 4] Compiling Jvm.Data.ClassFormat ( src/Jvm/Data/ClassFormat.hs, dist/build/dbjava/dbjava-
tmp/Jvm/Data/ClassFormat.o )
[2 of 4] Compiling Jvm.BinaryClass  ( src/Jvm/BinaryClass.hs, dist/build/dbjava/dbjava-
tmp/Jvm/BinaryClass.o )
[3 of 4] Compiling Jvm.PrettyClass  ( src/Jvm/PrettyClass.hs, dist/build/dbjava/dbjava-
tmp/Jvm/PrettyClass.o )
[4 of 4] Compiling Main             ( src/Main.hs, dist/build/dbjava/dbjava-tmp/Main.o )
Linking dist/build/dbjava/dbjava ...
[carlos@cathrina dbjava]$ runhaskell Setup.hs haddock
Running Haddock for dbjava-1.7...
Preprocessing library dbjava-1.7...
Preprocessing executables for dbjava-1.7...
Warning: The documentation for the following packages are not installed. No
links will be generated to these packages: binary-0.5.0.2, ffi-1.0, rts-1.0
Warning: Jvm.Data.ClassFormat: could not find link destinations for:
    Data.Binary.Binary
Warning: Jvm.BinaryClass: could not find link destinations for:
    Data.Binary.Binary Data.Binary.Get.Get
Documentation created: dist/doc/html/dbjava/index.html
[carlos@cathrina dbjava]$ sudo runhaskell Setup.hs install
Installing library in /usr/local/lib/dbjava-1.7/ghc-6.12.3
Installing executable(s) in /usr/local/bin
Registering dbjava-1.7...
[carlos@cathrina dbjava]$ runhaskell Setup.hs clean
cleaning...
```

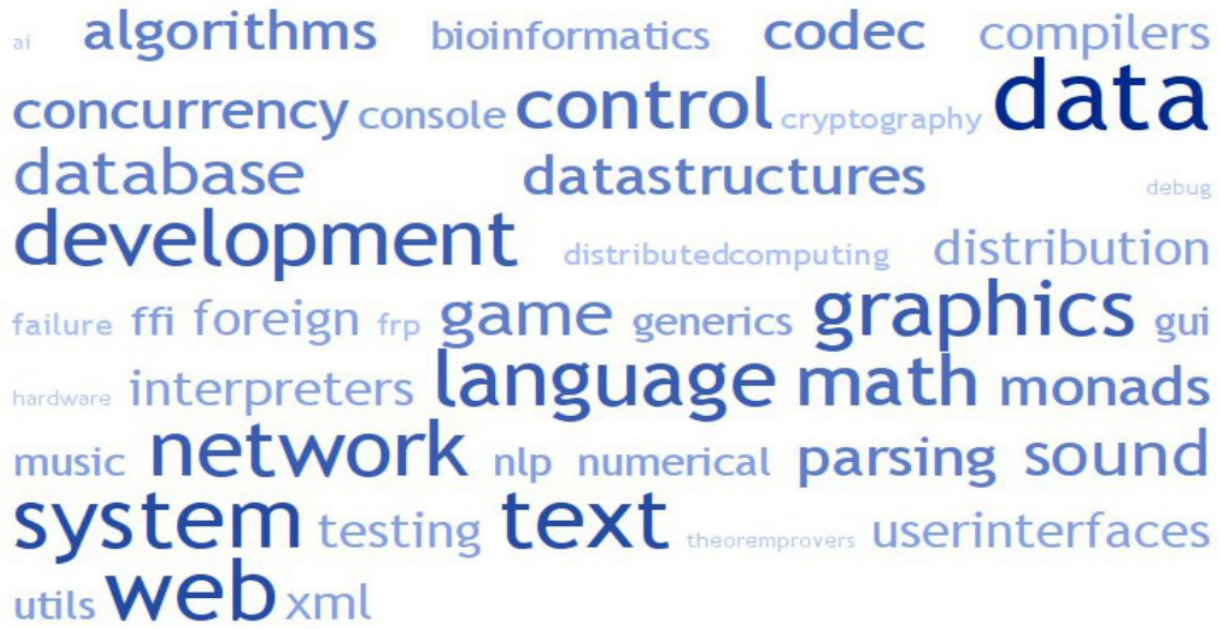
Distribucion de un Paquete

- Podemos distribuir un paquete de una manera estandar (Forma en que maneja Cabal).
- Simplemente hacemos uso de un comando de cabal:

```
runhaskell Setup.hs sdist
```

- Ahora, podemos compartir nuestro paquete e incluso subirlo a Hackage.

Hackage



- Es Lugar donde viven todos los paquetes de Haskell.
- URL: hackage.haskell.org
- Puedes descargar los paquetes, e instalarlos como habiamos aprendido.

Algunas recomendaciones

- Si no sabes los argumentos de un comando:
  `runhaskell Setup.hs haddock --help`
- Para descomprimir un paquete en linux:
  `tar xzvf paqueteHaskell.tar.gz`
- Para instalar en otro lugar (diferente a /usr/local/)

  ```
  runhaskell Setup.hs copy --copy-prefix=/usr
  ```

Algunas recomendaciones

- Haddock y HsColour en ejecutables:

```
runhaskell Setup.hs hscolour
--executables

runhaskell Setup.hs haddock
--executables
```

- Enlazar codigo fuente en la documentacion:

```
runhaskell Serup.hs haddock
--hyperlink-source
```

Herramientas sobre Cabal

- Cabal-get: Para descargar e instalar paquetes desde Hackage.
- dh_haskell: Convierte paquetes Haskell en paquetes para Debian.
- Cabal2rpm: Convierte paquetes Haskell en paquetes para RedHat (RPM)
- Hmake: Es una herramienta inteligente para administrar el proceso de compilacion de un programa de Haskell. Tambien provee

Referencias

- The Haskell Cabal, Isaac Jones, Galois Connections
- Documentacion de Cabal
- Haskell-Cabal, Cesar Flores, Comunidad Haskell

Mas Preguntas ?