

Primeros pasos con Apache Cassandra.

0. Índice de contenidos.

- 1. Entorno.
- 2. Introducción.
- 3. Descargar el software.
- 4. Preparación del entorno.
- 5. Instalando la versión correcta de Java para Cassandra.
- 6. Interactuamos desde consola con la nueva base de datos.
- 7. Instalación de la consola Helenos para Cassandra.
- 8. Instalación de DataStax Development Center.
- 9. Conclusiones.

1. Entorno

Este tutorial está escrito usando el siguiente entorno:

- Hardware: Ordenador iMac 27" (3.2 GHz Intel Core i5, 8 GB DDR3)
- Sistema Operativo: Mac OS X Mavericks 10.9

2. Introducción.

Dentro del mundo de las bases de datos NoSql existen muchas opciones sobre las que elegir: clave-valor, columnares, orientadas a documentos, orientadas a grafos, etc.

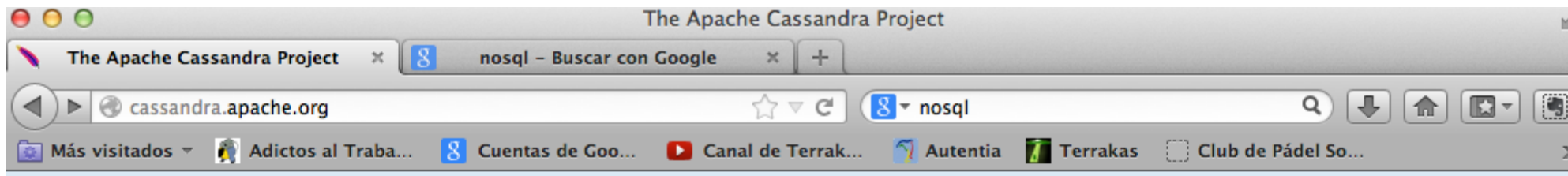
Me gusta saber las cosas de primera mano y "oler" la madurez de los productos, entornos y tecnologías y por ello vamos a dar los primeros pasos con Cassandra que es una base de datos "orientada a columnas".

Siempre digo que la investigación sin un tangible es tirar gran parte del tiempo: este tutorial es el tangible del proceso de investigación que he seguido con la idea de que en mucho menos tiempo cualquier compañero de Autentia (o de fuera), sea capaz de llegar a las mismas conclusiones en mucho menos tiempo que yo y, si hace un tutorial adicional, yo podré a su vez avanzar con menos esfuerzo. Como diría Newton: "Si he logrado ver más lejos, ha sido porque he subido a hombros de gigantes."

Si en el tutorial anterior de [MongoDB](#) decía que me recordaba a Tamino, de mis años en Software AG, ésta me recuerda, siempre desde de distancia y el cariño, a AdabasC, con campos periódicos y múltiples, estructuras desnormalizadas para alcanzar una eficiencia en almacenamiento y recuperación especiales (verhttp://jmpeco.es/personal/usr_docs/adabas_2.pdf). Todo en la vida se parece algo a otra cosa aunque obviamente estas bases de datos aportan un valor nuevo: el crecimiento horizontal :-)

3. Descargar el software.

Vamos al portal de Cassandra en Apache: cassandra.apache.org/



[Home](#) [Download](#) [Getting Started](#) [Planet Cassandra](#) [Contribute](#)

[Welcome](#) [Video](#) [Slides](#)

Welcome to Apache Cassandra

The Apache Cassandra database is the right choice when you need scalability and high availability without compromising performance. [Linear scalability](#) and proven fault-tolerance on commodity hardware or cloud infrastructure make it the perfect platform for mission-critical data. Cassandra's support for replicating across multiple datacenters is best-in-class, providing lower latency for your users and the peace of mind of knowing that you can survive regional outages.

Cassandra's data model offers the convenience of [column indexes](#) with the performance of log-structured updates, strong support for [denormalization](#) and [materialized views](#), and powerful built-in caching.

Download

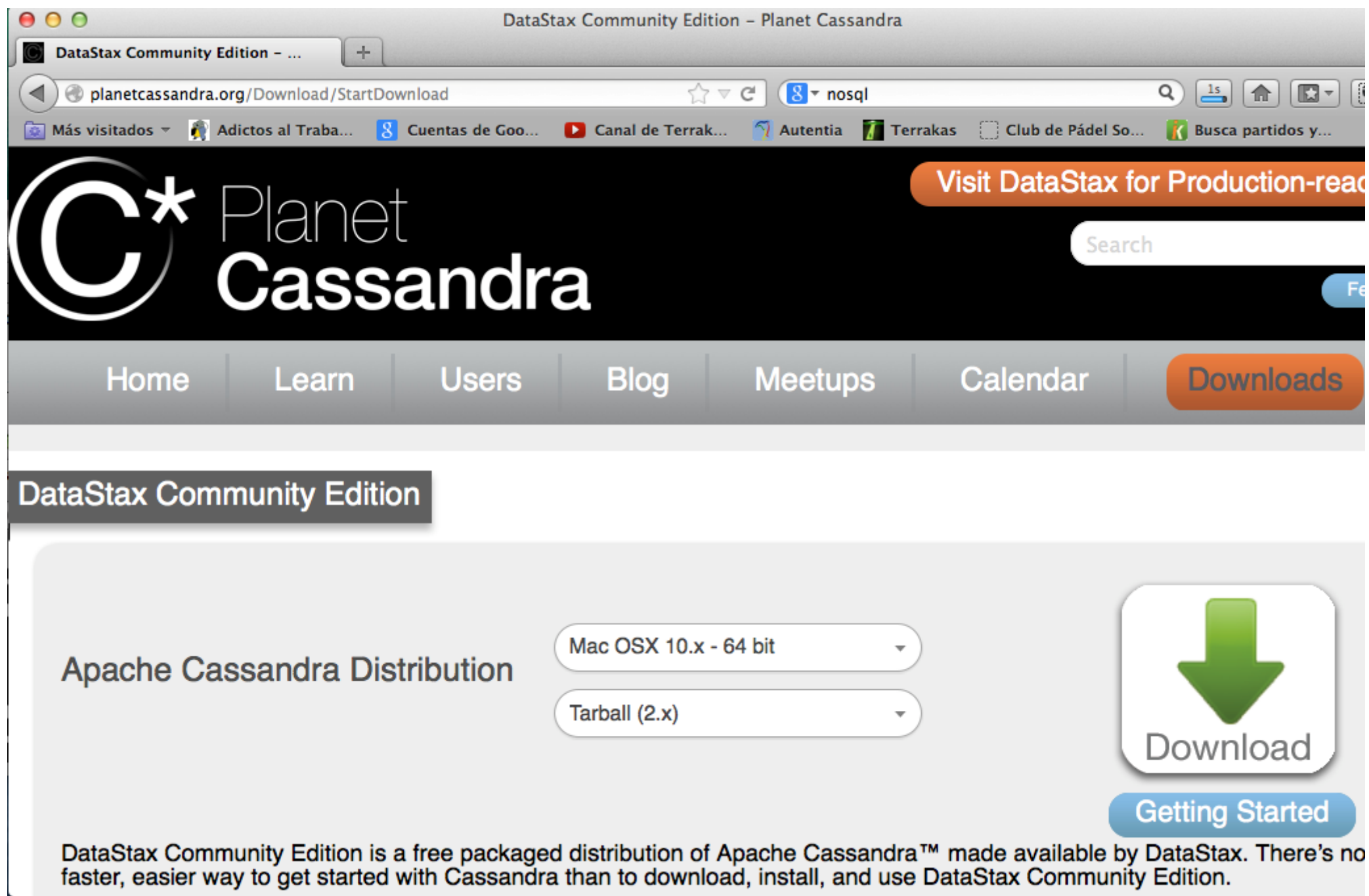
The latest release is **2.0.2**

[\(Changes\)](#)



[Download options](#)

En el propio Web nos sugiere que nos descarguemos una distribución ya compilada que podemos conseguir en DataStax.



4. Preparación del entorno.

Ejecutamos los Scripts para instalar. Simplemente seguimos las instrucciones para crear un directorio de trabajo, desplegar los binarios, añadir el path a la variable \$PATH del entorno, etc.

Es recomendable ir siempre a la fuente original y seguir los pasos: <http://wiki.apache.org/cassandra/GettingStarted>

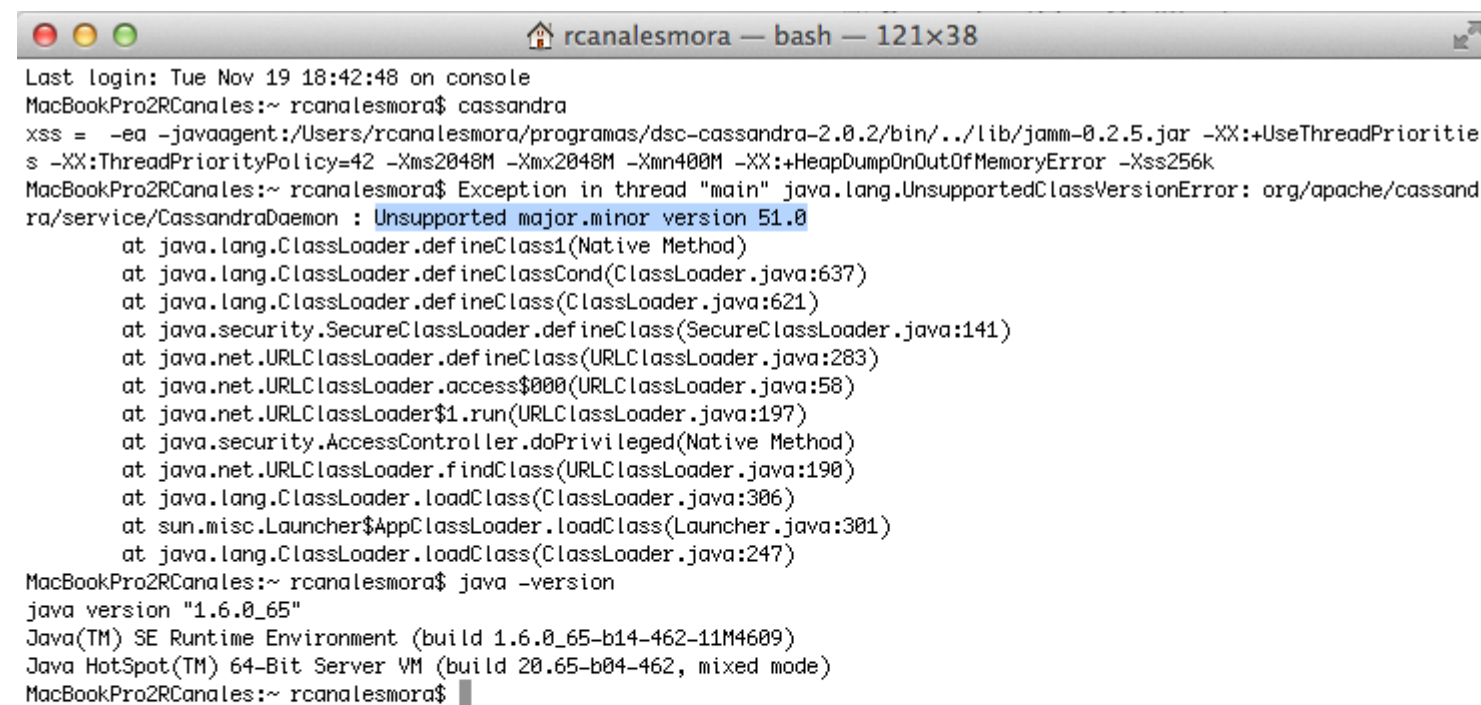
```
Last login: Tue Nov 19 18:29:48 on ttys000
MacBookPro2RCanales:~ rcanalesmora$ pwd
/Users/rcanalesmora
MacBookPro2RCanales:~ rcanalesmora$ cd programas/
MacBookPro2RCanales:programas rcanalesmora$ cd dsc-cassandra-2.0.2/
```

```

MacBookPro2RCanales:dsc-cassandra-2.0.2 rcanalesmora$ sudo mkdir -p /var/log/cassandra
Password:
MacBookPro2RCanales:dsc-cassandra-2.0.2 rcanalesmora$ sudo chown -R `whoami` /var/log/cassandra
MacBookPro2RCanales:dsc-cassandra-2.0.2 rcanalesmora$ sudo mkdir -p /var/lib/cassandra
MacBookPro2RCanales:dsc-cassandra-2.0.2 rcanalesmora$ sudo chown -R `whoami` /var/lib/cassandra
MacBookPro2RCanales:dsc-cassandra-2.0.2 rcanalesmora$ sudo su
sh-3.2# cd /etc/pathsh.d/
sh-3.2# echo "/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin" > cassandra
sh-3.2# ls
cassandra mongod
sh-3.2# cat cassandra
/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin
sh-3.2#

```

Cuando lo tenemos instalado, sólo tenemos que ejecutar el comando **cassandra** para arrancar.



```

Last login: Tue Nov 19 18:42:48 on console
MacBookPro2RCanales:~ rcanalesmora$ cassandra
xss = -ea -javaagent:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/jamm-0.2.5.jar -XX:+UseThreadPrioritie
s -XX:ThreadPriorityPolicy=42 -Xms2048M -Xmx2048M -Xmn400M -XX:+HeapDumpOnOutOfMemoryError -Xss256k
MacBookPro2RCanales:~ rcanalesmora$ Exception in thread "main" java.lang.UnsupportedClassVersionError: org/apache/cassand
ra/service/CassandraDaemon : Unsupported major.minor version 51.0
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClassCond(ClassLoader.java:637)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:621)
    at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:141)
    at java.net.URLClassLoader.defineClass(URLClassLoader.java:283)
    at java.net.URLClassLoader.access$000(URLClassLoader.java:58)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:197)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:190)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:306)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:301)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:247)
MacBookPro2RCanales:~ rcanalesmora$ java -version
java version "1.6.0_65"
Java(TM) SE Runtime Environment (build 1.6.0_65-b14-462-11M4609)
Java HotSpot(TM) 64-Bit Server VM (build 20.65-b04-462, mixed mode)
MacBookPro2RCanales:~ rcanalesmora$

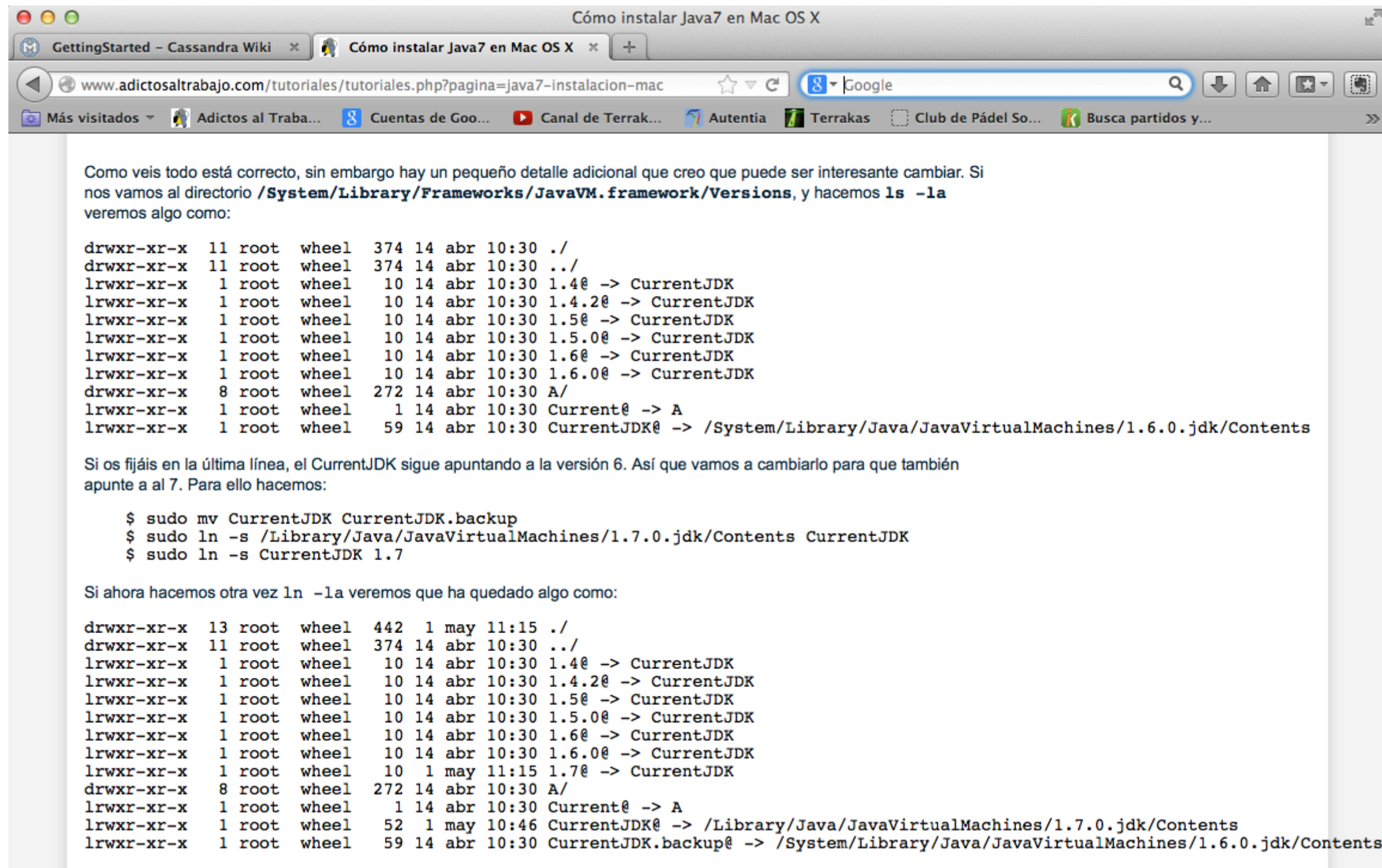
```

Podemos observar que en mi caso se produce un error. Como está construida en Java tiene que haber una coincidencia entre la versión en la que está compilada y la que tengo instalada (por lo menos tiene que ser esta última igual o superior).

5. Instalando la versión correcta de Java para Cassandra.

El problema es la versión de Java instalada en mi Mac. Para solucionarlo podéis consultar un tutorial de @alejandropgarci que cuenta los pasos para [instalar Java7 en Mac OS X](#).

Le hacemos caso descargándola, instalándola y haciendo unos pequeños cambios manuales en los ficheros del sistema para que encuentre la versión que necesitamos.



Como veis todo está correcto, sin embargo hay un pequeño detalle adicional que creo que puede ser interesante cambiar. Si nos vamos al directorio `/System/Library/Frameworks/JavaVM.framework/Versions`, y hacemos `ls -la` veremos algo como:

```
drwxr-xr-x  11 root  wheel  374 14 abr 10:30 ./
drwxr-xr-x  11 root  wheel  374 14 abr 10:30 ../
lrwxr-xr-x   1 root  wheel   10 14 abr 10:30 1.4@ -> CurrentJDK
lrwxr-xr-x   1 root  wheel   10 14 abr 10:30 1.4.2@ -> CurrentJDK
lrwxr-xr-x   1 root  wheel   10 14 abr 10:30 1.5@ -> CurrentJDK
lrwxr-xr-x   1 root  wheel   10 14 abr 10:30 1.5.0@ -> CurrentJDK
lrwxr-xr-x   1 root  wheel   10 14 abr 10:30 1.6@ -> CurrentJDK
lrwxr-xr-x   1 root  wheel   10 14 abr 10:30 1.6.0@ -> CurrentJDK
drwxr-xr-x   8 root  wheel  272 14 abr 10:30 A/
lrwxr-xr-x   1 root  wheel    1 14 abr 10:30 Current@ -> A
lrwxr-xr-x   1 root  wheel   59 14 abr 10:30 CurrentJDK@ -> /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents
```

Si os fijáis en la última línea, el CurrentJDK sigue apuntando a la versión 6. Así que vamos a cambiarlo para que también apunte a al 7. Para ello hacemos:

```
$ sudo mv CurrentJDK CurrentJDK.backup
$ sudo ln -s /Library/Java/JavaVirtualMachines/1.7.0.jdk/Contents CurrentJDK
$ sudo ln -s CurrentJDK 1.7
```

Si ahora hacemos otra vez `ln -la` veremos que ha quedado algo como:

```
drwxr-xr-x  13 root  wheel  442  1 may 11:15 ./
drwxr-xr-x  11 root  wheel  374 14 abr 10:30 ../
lrwxr-xr-x   1 root  wheel   10 14 abr 10:30 1.4@ -> CurrentJDK
lrwxr-xr-x   1 root  wheel   10 14 abr 10:30 1.4.2@ -> CurrentJDK
lrwxr-xr-x   1 root  wheel   10 14 abr 10:30 1.5@ -> CurrentJDK
lrwxr-xr-x   1 root  wheel   10 14 abr 10:30 1.5.0@ -> CurrentJDK
lrwxr-xr-x   1 root  wheel   10 14 abr 10:30 1.6@ -> CurrentJDK
lrwxr-xr-x   1 root  wheel   10 14 abr 10:30 1.6.0@ -> CurrentJDK
lrwxr-xr-x   1 root  wheel   10  1 may 11:15 1.7@ -> CurrentJDK
drwxr-xr-x   8 root  wheel  272 14 abr 10:30 A/
lrwxr-xr-x   1 root  wheel    1 14 abr 10:30 Current@ -> A
lrwxr-xr-x   1 root  wheel   52  1 may 10:46 CurrentJDK@ -> /Library/Java/JavaVirtualMachines/1.7.0.jdk/Contents
lrwxr-xr-x   1 root  wheel   59 14 abr 10:30 CurrentJDK.backup@ -> /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents
```

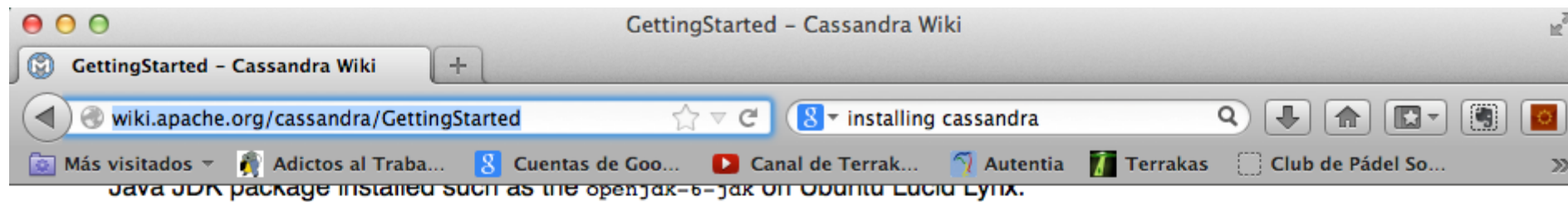
Como tip adicional (que he puesto como comentario en el tutorial), "me daba un error porque en `.bash_profile` estaba puesta la variable de entorno `JAVA_HOME` a la versión 6, que conviene ponerla a `JAVA_HOME=$(/usr/libexec/java_home)`"

Una vez corregida la variable de entorno JAVA_HOME ya arranca correctamente.

```
MacBookPro2RCanales:~ rcanalesmora — bash — 139x41
Last login: Tue Nov 19 20:20:10 on ttys000
MacBookPro2RCanales:~ rcanalesmora$ cassandra
xss = -ea -javaagent:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/jamm-0.2.5.jar -XX:+UseThreadPriorities -XX:ThreadPrior
ityPolicy=42 -Xms2048M -Xmx2048M -Xmn400M -XX:+HeapDumpOnOutOfMemoryError -Xss256k
MacBookPro2RCanales:~ rcanalesmora$ objc[421]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.7.0_45.
jdk/Contents/Home/bin/java and /Library/Java/JavaVirtualMachines/jdk1.7.0_45.jdk/Contents/Home/jre/lib/libinstrument.dylib. One of the two
will be used. Which one is undefined.
INFO 20:20:20,683 Logging initialized
INFO 20:20:20,701 JVM vendor/version: Java HotSpot(TM) 64-Bit Server VM/1.7.0_45
INFO 20:20:20,701 Heap size: 2105540608/2105540608
INFO 20:20:20,701 Classpath: /Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../conf:/Users/rcanalesmora/programas/dsc-cassandra-2.0
.2/bin/../build/classes/main:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../build/classes/thrift:/Users/rcanalesmora/programas/ds
c-cassandra-2.0.2/bin/../lib/antlr-3.2.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/apache-cassandra-2.0.2.jar:/Users/r
canalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/apache-cassandra-clientutil-2.0.2.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.
2/bin/../lib/apache-cassandra-thrift-2.0.2.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/commons-cli-1.1.jar:/Users/rcan
alesmora/programas/dsc-cassandra-2.0.2/bin/../lib/commons-codec-1.2.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/common
s-lang3-3.1.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/compress-lzf-0.8.4.jar:/Users/rcanalesmora/programas/dsc-cassa
ndra-2.0.2/bin/../lib/concurrentlinkedhashmap-lru-1.3.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/disruptor-3.0.1.jar:/
Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/guava-15.0.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/hi
gh-scale-lib-1.1.2.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/jackson-core-asl-1.9.2.jar:/Users/rcanalesmora/programa
s/dsc-cassandra-2.0.2/bin/../lib/jackson-mapper-asl-1.9.2.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/jamm-0.2.5.jar:/
Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/jbcrypt-0.3m.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/j
line-1.0.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/json-simple-1.1.jar:/Users/rcanalesmora/programas/dsc-cassandra-2
.0.2/bin/../lib/libthrift-0.9.1.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/log4j-1.2.16.jar:/Users/rcanalesmora/progr
amas/dsc-cassandra-2.0.2/bin/../lib/lz4-1.2.0.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/metrics-core-2.2.0.jar:/User
s/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/netty-3.6.6.Final.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/
reporter-config-2.1.0.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/servlet-api-2.5-20081211.jar:/Users/rcanalesmora/pro
gramas/dsc-cassandra-2.0.2/bin/../lib/slf4j-api-1.7.2.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/slf4j-log4j12-1.7.2.
jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/snakeyaml-1.11.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/.
../lib/snappy-java-1.0.5.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/snaptree-0.1.jar:/Users/rcanalesmora/programas/dsc
-cassandra-2.0.2/bin/../lib/thrift-server-0.3.2.jar:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/jamm-0.2.5.jar
INFO 20:20:20,704 JNA not found. Native methods will be disabled.
INFO 20:20:20,722 Loading settings from file:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/conf/cassandra.yaml
INFO 20:20:21,092 Data files directories: [/var/lib/cassandra/data]
INFO 20:20:21,092 Commit log directory: /var/lib/cassandra/commitlog
INFO 20:20:21,093 DiskAccessMode 'auto' determined to be mmap, indexAccessMode is mmap
INFO 20:20:21,093 disk_failure_policy is stop
INFO 20:20:21,101 Global memtable threshold is enabled at 502MB
INFO 20:20:21,321 Not using multi-threaded compaction
INFO 20:20:21,600 Initializing key cache with capacity of 100 MBs.
INFO 20:20:21,608 Scheduling key cache save to each 14400 seconds (going to save all keys).
```

6. Interactuamos desde consola con la nueva base de datos.

Para atacar la base de datos y asegurarnos que funciona seguimos el manual:



Step 4: Using cqlsh

`bin/cqlsh` is an interactive command line interface for Cassandra. You can define the schema and interact with data using it. Run the following command to connect to your local Cassandra instance:

```
$ bin/cqlsh
```

You should see the following prompt, if successful:

```
Connected to Test Cluster at localhost:9160.  
[cqlsh 2.3.0 | Cassandra 1.2.2 | CQL spec 3.0.0 | Thrift protocol 19.35.0]  
Use HELP for help.
```

For clarity, we will omit the `cqlsh` prompt in the following examples.

You can access the online help with 'help;' command. Commands are terminated with a semicolon (;) in `cqlsh`.

First, create a keyspace -- a namespace of tables.

```
CREATE KEYSPACE mykeyspace  
WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
```

Second, authenticate to the new keyspace:

```
USE mykeyspace;
```

Third, create a `users` table:

Y ejecutamos `cqlsh` (que tenemos en el path), para arrancar la aplicación interfaz de línea de comando para interactuar con el sistema.

```
CREATE KEYSPACE miespacio WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
```

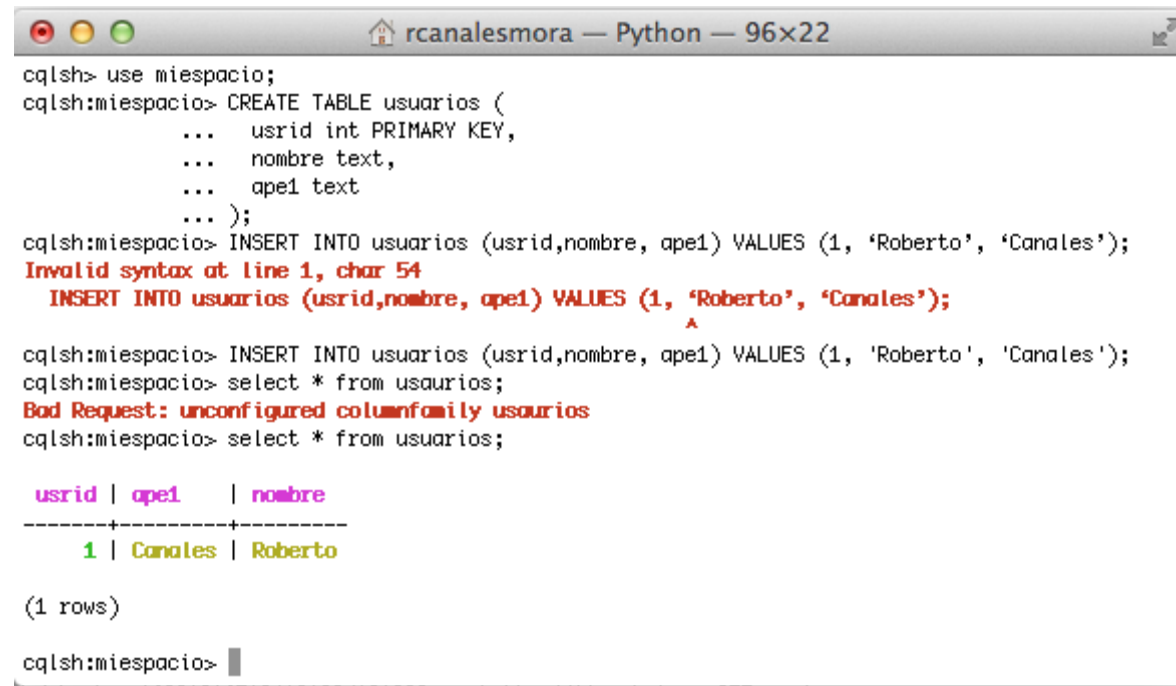
```
USE miespacio
```

```
CREATE TABLE usuarios (  
  usrid int PRIMARY KEY,  
  nombre text,  
  apel text  
);
```

```
INSERT INTO usuarios (usrid,nombre, ape1) VALUES (1, 'Roberto', 'Canales');

select * from usuarios;
```

Como podemos ver en la captura siguiente, nos pueden fallar las cosas por chorradillas: no especificar el espacio a usar, al copiar y pegar las comillas no son las correctas, etc. Bueno, pero estamos en órbita y ya podemos empezar a jugar un poquito más.



```
cqlsh> use miespacio;
cqlsh:miespacio> CREATE TABLE usuarios (
...   usrid int PRIMARY KEY,
...   nombre text,
...   ape1 text
... );
cqlsh:miespacio> INSERT INTO usuarios (usrid,nombre, ape1) VALUES (1, 'Roberto', 'Canales');
Invalid syntax at line 1, char 54
INSERT INTO usuarios (usrid,nombre, ape1) VALUES (1, 'Roberto', 'Canales');
cqlsh:miespacio> INSERT INTO usuarios (usrid,nombre, ape1) VALUES (1, 'Roberto', 'Canales');
cqlsh:miespacio> select * from usaurios;
Bad Request: unconfigured columnfamily usaurios
cqlsh:miespacio> select * from usuarios;

  usrid | ape1 | nombre
-----+-----+-----
      1 | Canales | Roberto

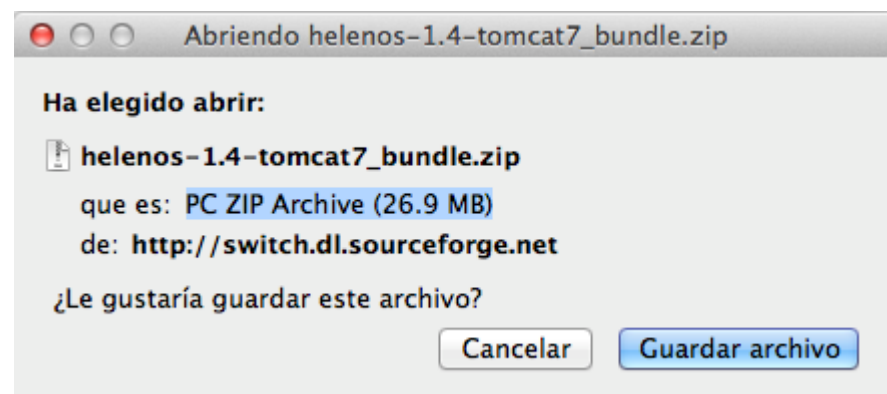
(1 rows)

cqlsh:miespacio>
```

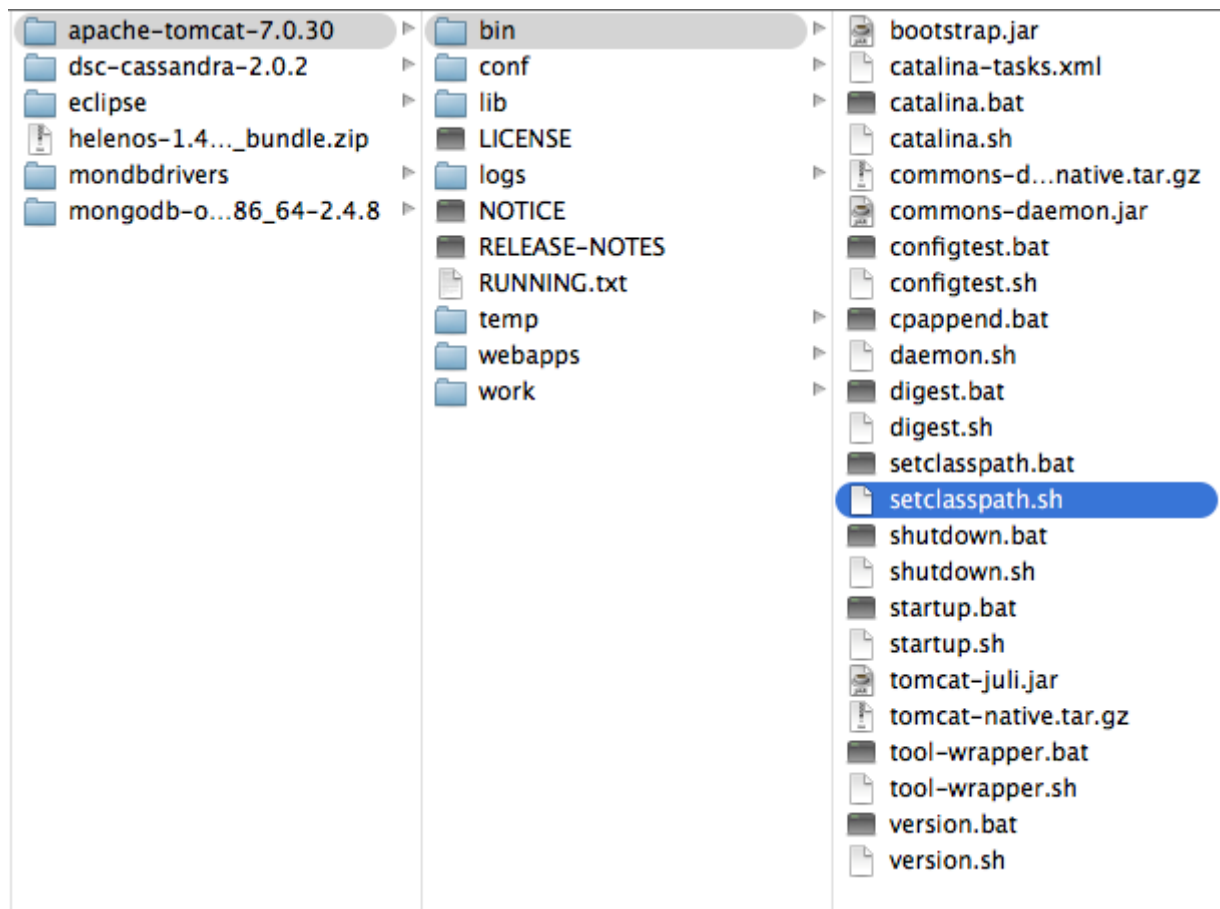
7. Instalación de la consola Helenos para Cassandra.

Como ya hice con MongoDB, voy a instalarme un entorno un pelín más amigable para jugar con Cassandra que la línea de comando básica tan susceptible de errores :-)

La primera que vamos a probar se llama [Helenos](#)



He descargado el bundle que incluye Apache Tomcat. Lo descomprimos.



Para arrancar apache ejecutamos **catalina.sh start**

```
MacBookPro2RCanales:~ rcanalesmora$ cd programas/
MacBookPro2RCanales:programas rcanalesmora$ cd apache-tomcat-7.0.30/
MacBookPro2RCanales:apache-tomcat-7.0.30 rcanalesmora$ cd bin
MacBookPro2RCanales:bin rcanalesmora$ ./catalina.sh
The BASEDIR environment variable is not defined
This environment variable is needed to run this program
MacBookPro2RCanales:bin rcanalesmora$
```

Nos da un error porque no están correctamente establecidas las variables de entorno, en concreto **CATALINA_HOME**.

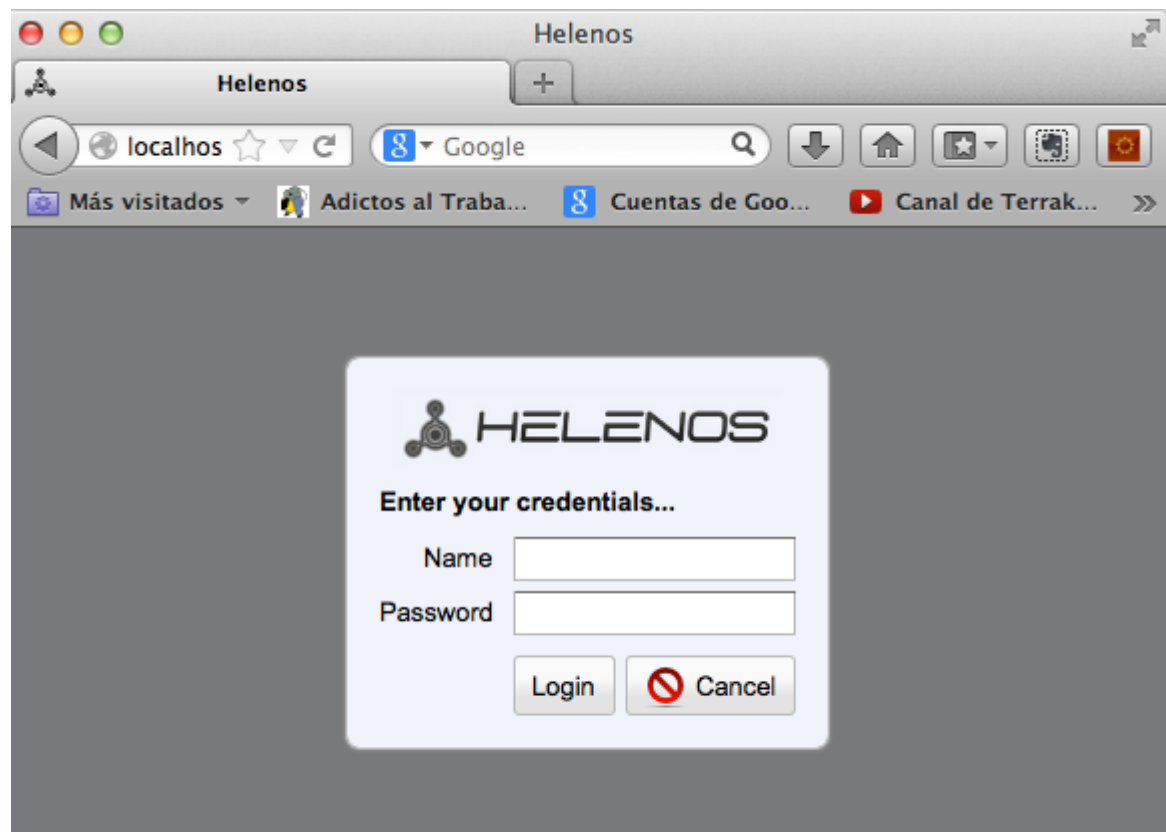
Editamos el fichero **.bash_profile**. Aquí tenéis una captura de la secuencia de comandos.

```
MacBookPro2RCanales:~ rcanalesmora$ cd programas/
MacBookPro2RCanales:programas rcanalesmora$ cd apache-tomcat-7.0.30/
MacBookPro2RCanales:apache-tomcat-7.0.30 rcanalesmora$ cd bin
MacBookPro2RCanales:bin rcanalesmora$ ./catalina.sh
The BASEDIR environment variable is not defined
This environment variable is needed to run this program
MacBookPro2RCanales:bin rcanalesmora$ cd ../../..
MacBookPro2RCanales:~ rcanalesmora$ vi .bash_profile
MacBookPro2RCanales:~ rcanalesmora$ cat .bash_profile
export JAVA_HOME=$(/usr/libexec/java_home)
export PATH=${PATH}:/Developer/android-sdk-mac_86/tools
export CATALINA_HOME=/Users/rcanalesmora/programas/apache-tomcat-7.0.30
export CATALINA_BASE=${CATALINA_HOME}
MacBookPro2RCanales:~ rcanalesmora$
```

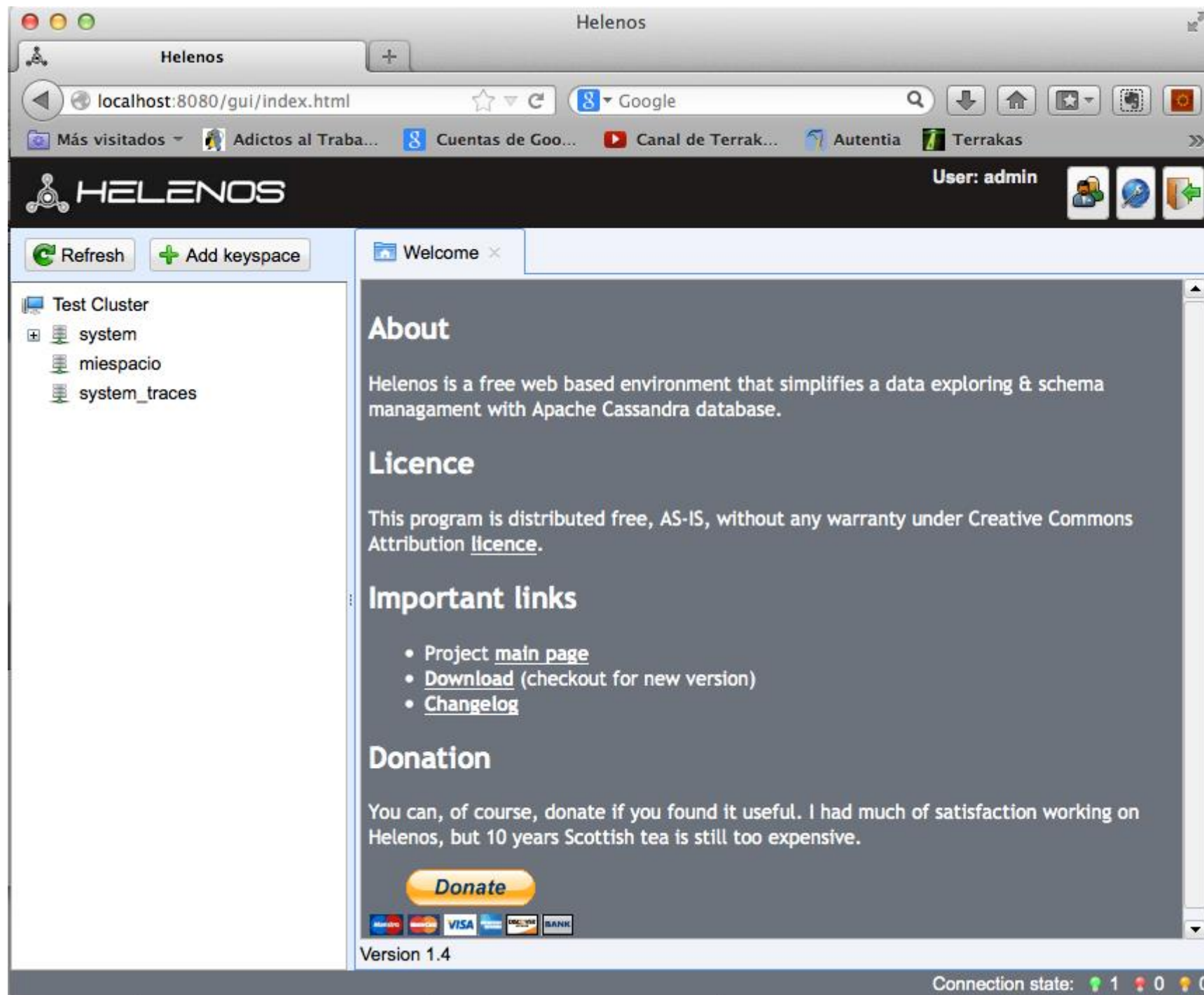
Arrancamos el servidor de nuevo y ya funciona correctamente.

```
MacBookPro2RCanales:bin rcanalesmora$ ./catalina.sh start
Using CATALINA_BASE:   /Users/rcanalesmora/programas/apache-tomcat-7.0.30
Using CATALINA_HOME:   /Users/rcanalesmora/programas/apache-tomcat-7.0.30
Using CATALINA_TMPDIR: /Users/rcanalesmora/programas/apache-tomcat-7.0.30/temp
Using JRE_HOME:        /Library/Java/JavaVirtualMachines/jdk1.7.0_45.jdk/Contents/Home
Using CLASSPATH:       /Users/rcanalesmora/programas/apache-tomcat-7.0.30/bin/bootstrap.jar:/Users/rcanalesmora/programas/apache-tomcat-7.0.30/bin/tomcat-juli.jar
MacBookPro2RCanales:bin rcanalesmora$
```

Atacamos con un navegador por http a localhost:8080.

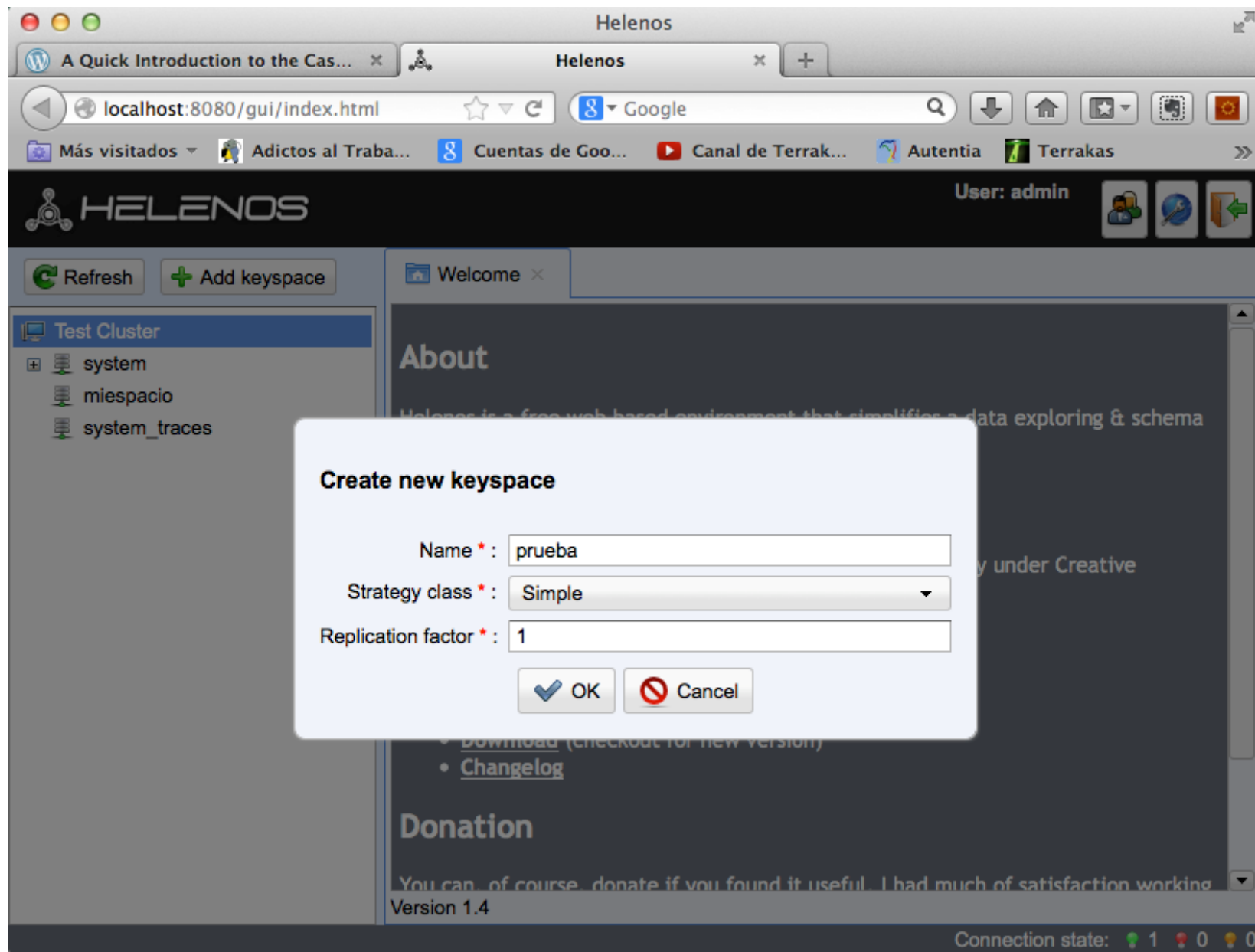


Metemos la contraseña por defecto: admin admin

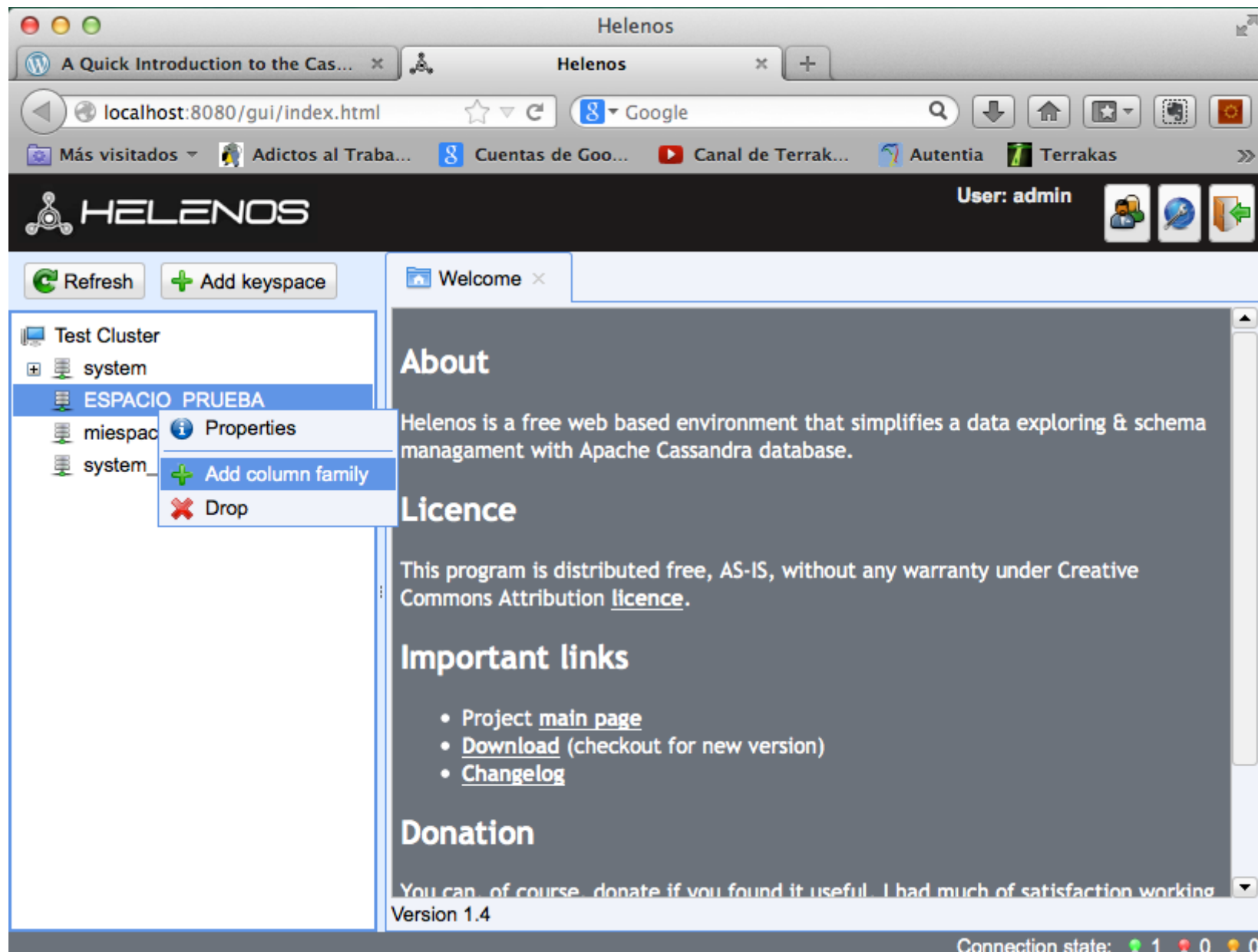


Bueno, ya tenemos otra herramienta con la que jugar.

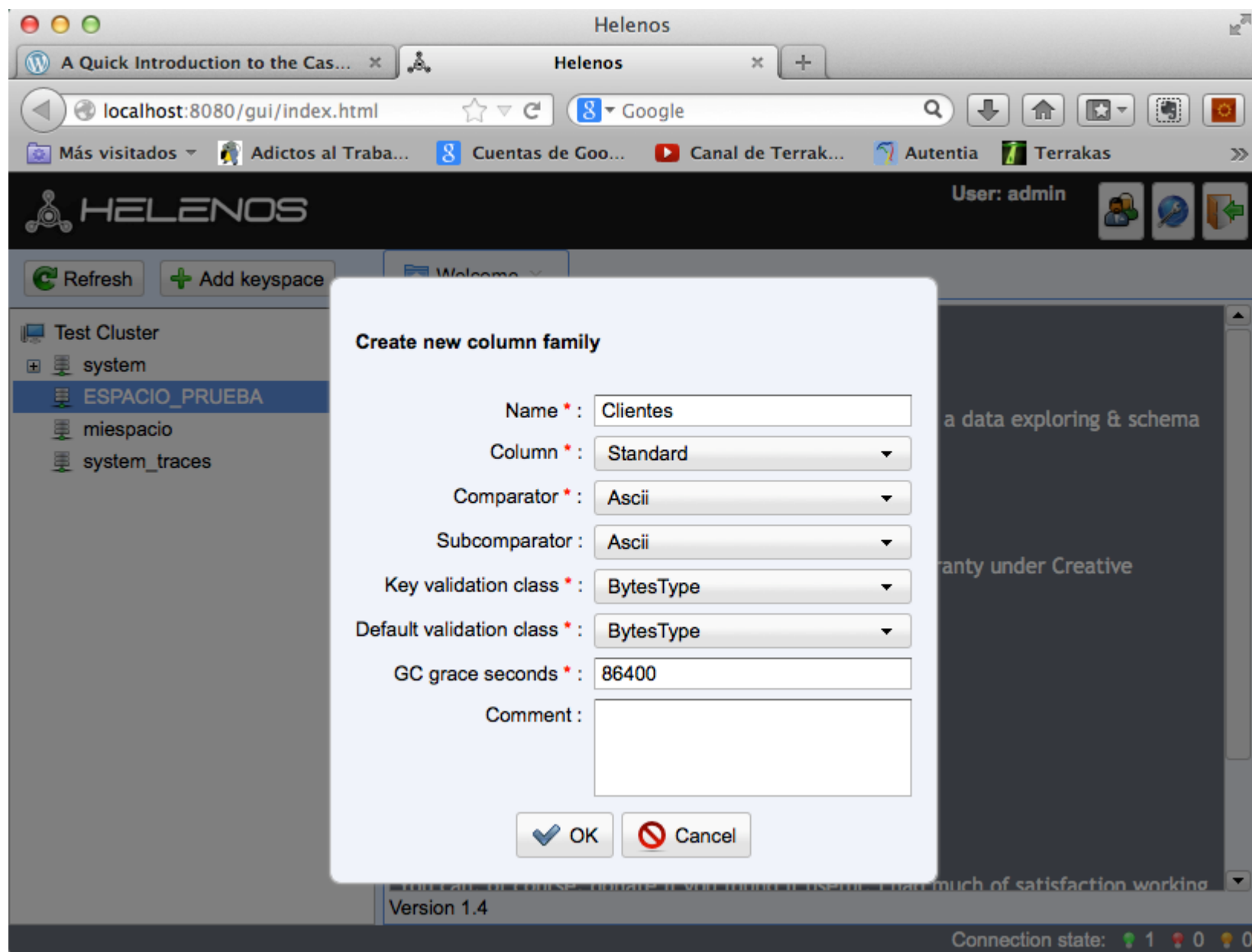
Es interesante sobre todo para ver los parámetros de inicialización.



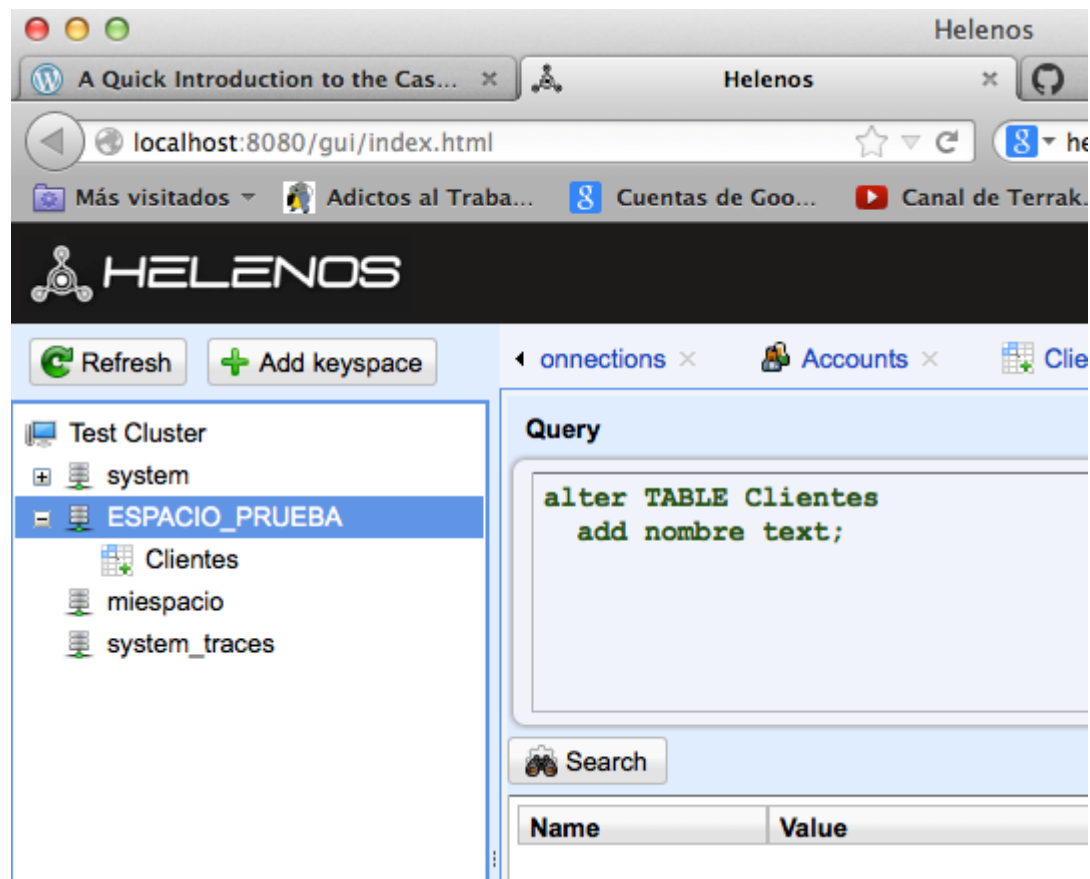
Añadimos familia de columnas (lo que vendría a ser conceptualmente una tabla pero con sus matices de que cada columna se almacena o puede almacenar por separado).



Añadimos los atributos (ojito que luego nos arrepentiremos de no estudiar un poquito mejor los tipos de datos que indicamos).



Desde la consulta CQL vamos a introducir un comando para añadir columnas (no he sido capaz de encontrar donde hacerlo visualmente).



Vemos que se ha añadido dando a propiedades.

HELENOS User: admin

Refresh Add keyspace

Test Cluster
system
ESPACIO_PRUEBA
Clientes
miespa
system

Properties
Browse data
CQL query
Drop
Truncate

Keyspace: ESPACIO_PRUEBA

Basic

Strategy class: **org.apache.cassandra.locator.SimpleStrategy**
Replication factor: 1

Column families

Id	Name	Type	Comp...	Sub c...	Key va...	Defaul...	Row c...	Key ca...	GC gr...
0	Clientes	Stand...	AsciiT...		Bytes...	Bytes...	0	0	86,400

Añadimos un registro (estoy haciendo un poco de trampa con la clave).

Browser tabs: A Quick Introduction to the Cas..., Helenos, tomekkup/helenos · GitHub

Address bar: localhost:8080/gui/index.html

Search bar: helenos cassandra

Navigation bar: Más visitados, Adictos al Traba..., Cuentas de Goo..., Canal de Terrak..., Autentia, Terrakas, Club de Pádel So...

Header: HELENOS User: admin

Buttons: Refresh, Add keyspace

Left sidebar:

- Test Cluster
 - system
 - ESPACIO_PRUEBA
 - Cientes
 - miespacio
 - system_traces

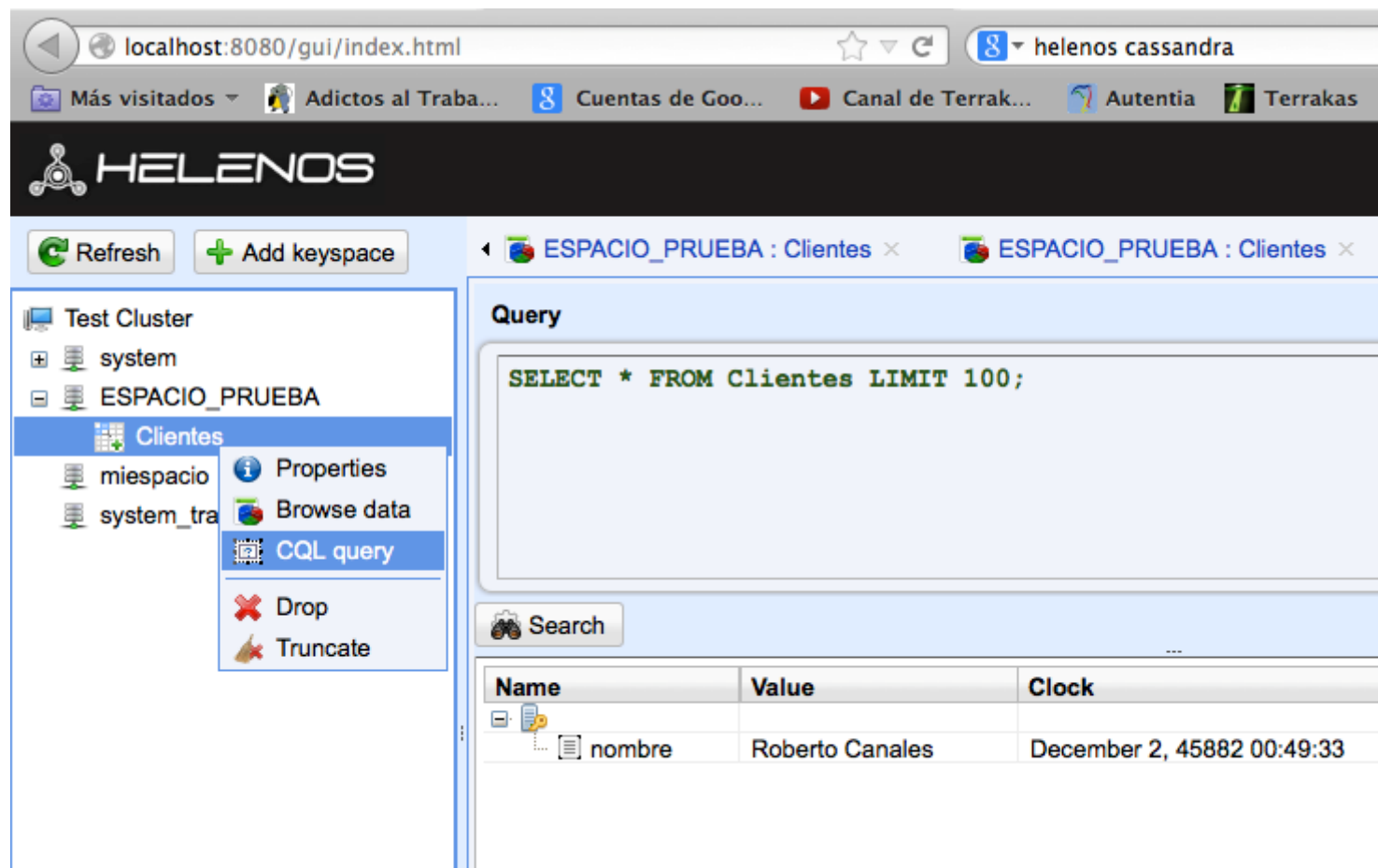
Query editor:

```
INSERT INTO Cientes (key,nombre) VALUES (0000, 'Roberto Canales');
```

Search: Consistency level: A

Name	Value	Clock
------	-------	-------

Vemos el resultado.



La verdad es que me tiene despistado esta herramienta porque lo que hago dentro de ella no lo veo fuera y lo que hago fuera no lo veo dentro (me refiero a una sesión de Terminal de cqlsh)... tendré que seguir investigando porque seguro que es una chorrada (relativo a permisos).

De todos modos, aunque no perderemos de vista esta herramienta por sus evoluciones futuras, voy a descargarme otra de DataStax.


8. Instalación de DataStax Development Center.

DataStax pone a disposición de los desarrolladores un entorno Eclipse adaptado para trabajar con Cassandra. Está bien porque puedes guardar las conexiones, ejecutar sentencias CQL y guardarlas como Scripts, te va diciendo los errores en las sentencias antes de ejecutarlas, se colorea el código, ves las estructuras de los espacios, tablas e índices (por llamarlos así, con sus matices ;-)

Browser window showing the DataStax DevCenter page. The address bar displays `www.datastax.com/what-we-offer/products-services/devcenter`. The page features the DataStax logo and navigation links: Support, Blog, Resources, Careers, Contact Us, and Planet Cassandra Community. The main navigation bar includes: BIG DATA CHALLENGES, CUSTOMERS, WHAT WE OFFER, DEVELOPER CENTER, COMPANY, and a DOWNLOAD button. The breadcrumb trail shows: DataStax > / What We Offer > / DataStax DevCenter.

DataStax DevCenter


Free Visual Query Tool for Cassandra Developers and Administrators



The developer tool of choice for Cassandra and DataStax Enterprise.

DataStax DevCenter is a free visual query tool that allows developers and administrators to easily create and run Cassandra Query Language (CQL) queries and commands against Apache Cassandra™ and DataStax Enterprise.

Free Virtual Training

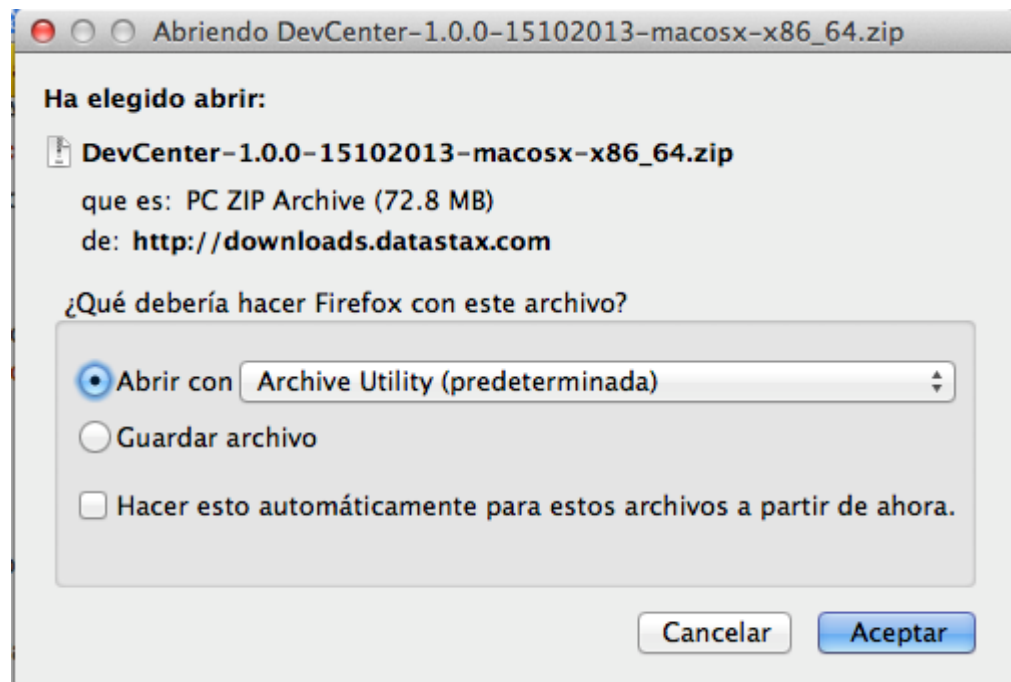


Learn Apache Cassandra with DataStax's free comprehensive online training.

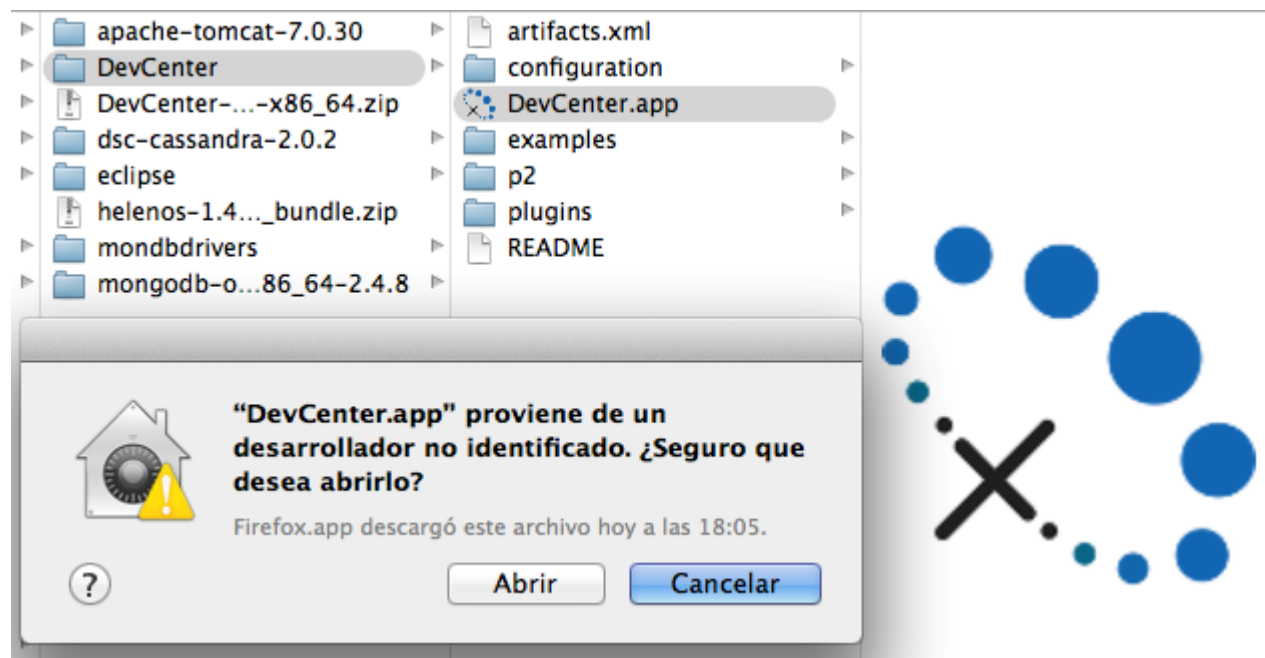
[Start Learning Now >](#)

DEVELOPER CENTER

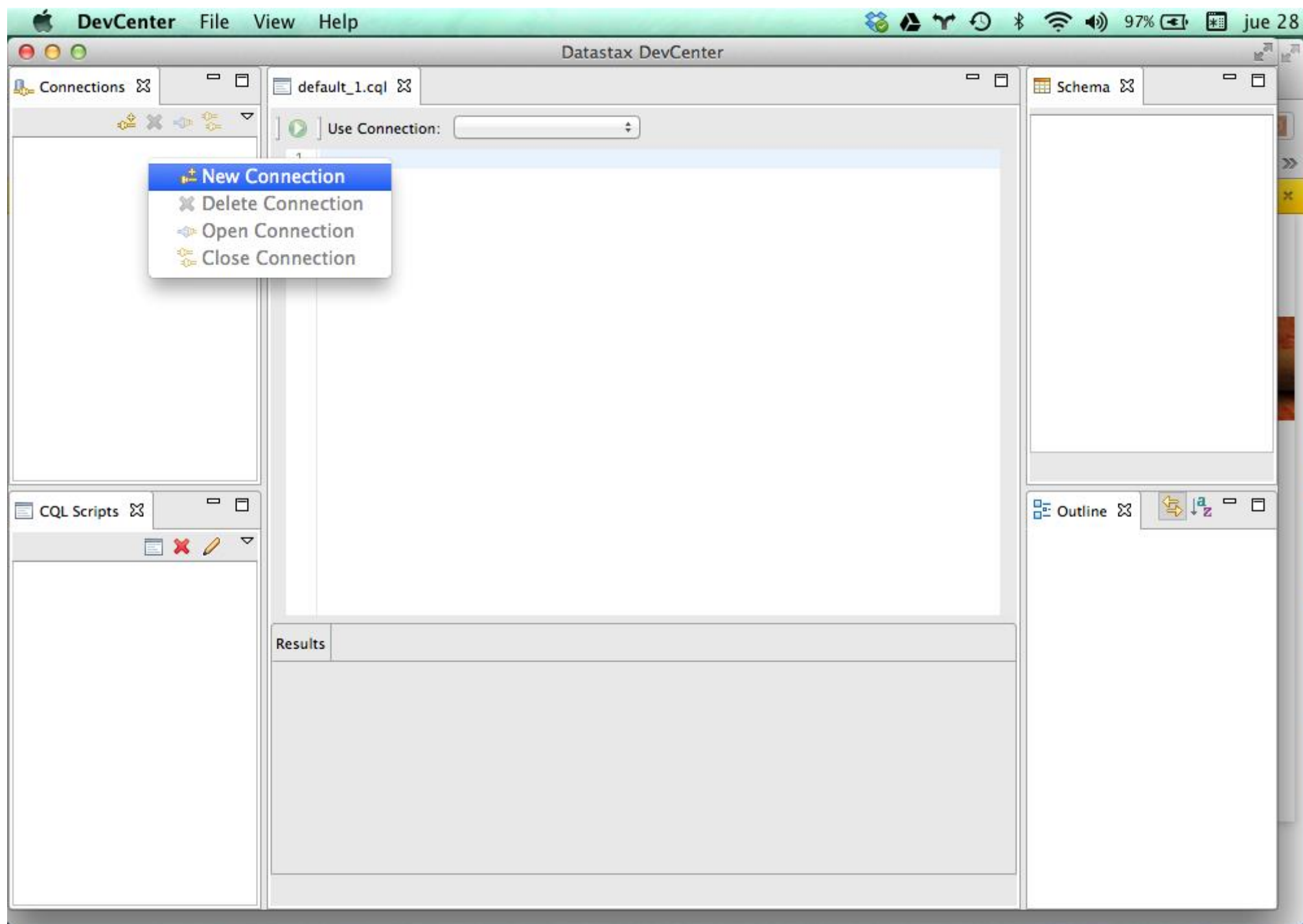
Descargamos el entorno:



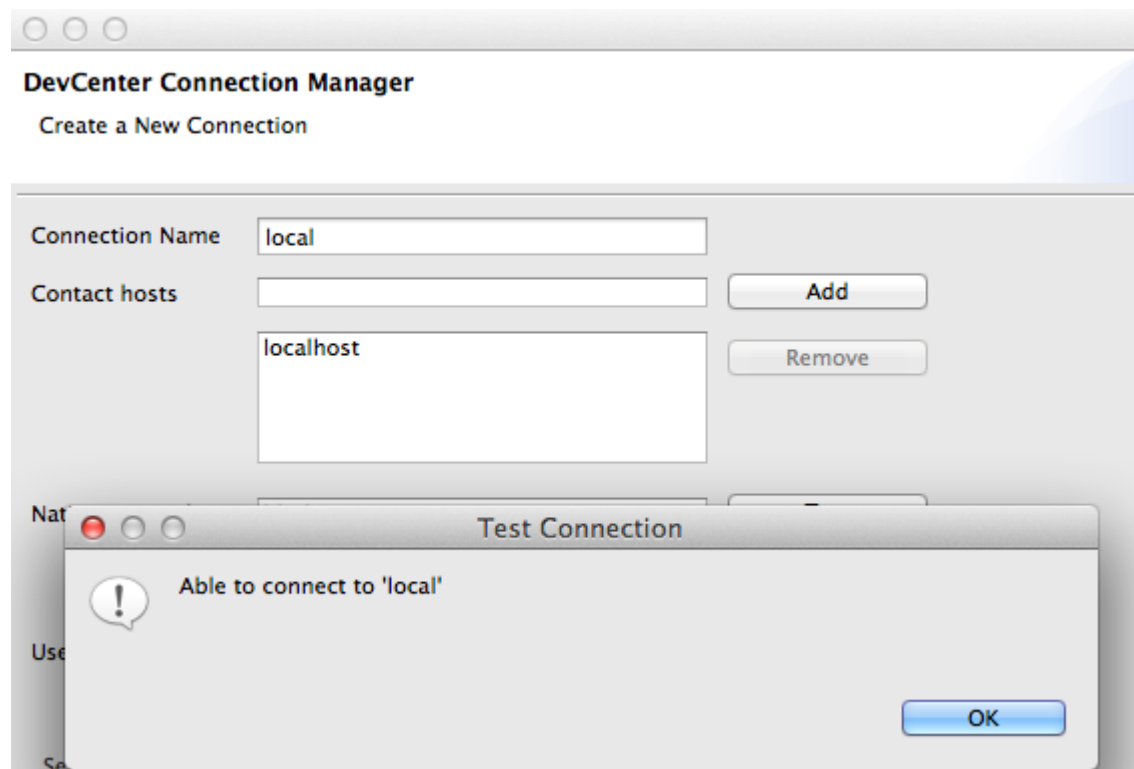
En Mac es una aplicación nativa



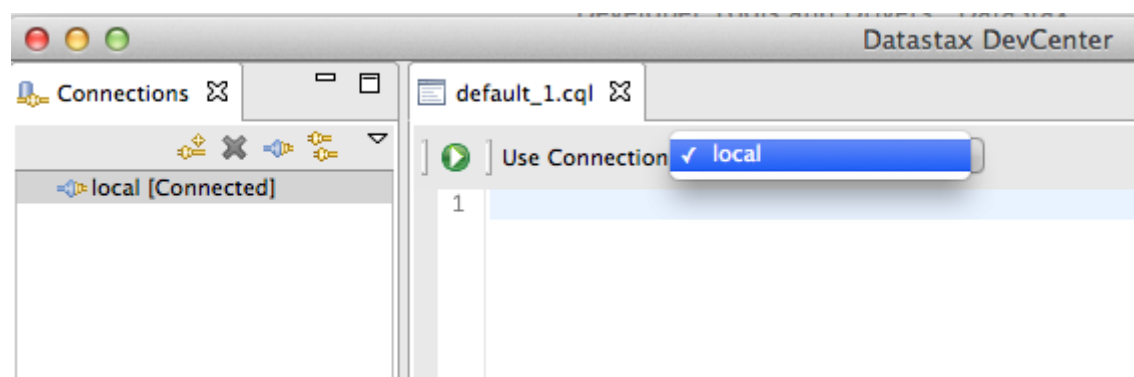
Si estamos acostumbrados a Eclipse es muy predecible. Creamos una nueva conexión.



Demos al botón de test a ver si todo va bien.



Elegimos en la pantalla principal la conexión a Cassandra a usar.



Y ya estamos operativos.

Podemos lanzar nuestros Scripts y ver inmediatamente los resultados. Si la última sentencia que pones es un **select** ves los cambios reflejados abajo.

```
CREATE KEYSPACE if not exists miespacio WITH REPLICATION =  
{ 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
```

```
USE miespacio;
```

```
CREATE TABLE usuarios (  
  usrid int PRIMARY KEY,  
  nombre text,  
  apel text  
);
```



```
INSERT INTO usuarios (usrid,nombre, ap1)  
VALUES (1, 'Roberto', 'Canales');
```

```
select * from usuarios;
```

The screenshot shows the Databricks DevCenter interface. The main editor displays a CQL script with the following steps:

- 1 CREATE KEYSPACE miespacio WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
- 2
- 3
- 4 USE miespacio;
- 5
- 6 CREATE TABLE usuarios (
7 usrid int PRIMARY KEY,
8 nombre text,
9 ap1 text
10);
- 11
- 12 INSERT INTO usuarios (usrid,nombre, ap1)
13 VALUES (1, 'Roberto', 'Canales');
- 14
- 15 select * from usuarios;

The 'Results' tab shows the output of the last query:

usrid	ap1	nombre
1	Canales	Roberto

Retrieved 1 row(s) in 662 ms

The right sidebar shows the 'Schema: local' view, which includes the 'usuarios' table with columns: usrid (int), ap1 (text), and nombre (text). The 'Outline' view shows the execution plan for the script, highlighting the 'INSERT INTO usuarios' step.

Para seguir jugando sería conveniente tener a mano un [resumen del lenguaje CQL](#)

CQL Reference Card - cql_3_ref_card.pdf

Documentation : DataStax

CQL Reference Card - cql_3_ref...

www.datastax.com/wp-content/uploads/2013/03/cql_3_ref_card.pdf

Google

Más visitados

Adictos al Traba...

Google Apps

Este documento PDF podría no mostrarse correctamente.

Abrir con un visor diferente

Página: 1 de 1

Tamaño automático

Apache Cassandra Query Language (CQL)

SPECIFICATION 3.1.0

DATASTAX

ALTER KEYSPACE

ALTER (KEYSPACE | SCHEMA) keyspace_name
WITH REPLICATION = map
| (WITH DURABLE_WRITES = (true | false))
AND (DURABLE_WRITES = (true | false))

map is a map collection, a JSON-style array of literals:
{ literal : literal, literal : literal ... }

ALTER TABLE

ALTER TABLE keyspace_name.table_name instruction

instruction is:
ALTER column_name TYPE cql_type
| (ADD column_name cql_type)
| (DROP column_name)
| (RENAME column_name TO column_name)
| (WITH property AND property ...)

cql_type is a CQL type, other than a collection or counter.
Exceptions: ADD supports a collection type and also, if the table
is a counter, a counter type.

property is a CQL 3 table storage property (p. 3) and value,
such as caching = 'all'.

ALTER USER

ALTER USER user_name WITH PASSWORD 'password'
(NOSUPERUSER | SUPERUSER)

BATCH

BEGIN BATCH (UNLOGGED | COUNTER)
USING TIMESTAMP timestamp
dml_statement;
dml_statement;
...
APPLY BATCH;

dml_statement is:
INSERT
UPDATE
DELETE

CREATE INDEX

CREATE CUSTOM INDEX IF NOT EXISTS index_name
ON keyspace_name.table_name (column_name)
USING class_name

Restrictions:
USING class_name is only allowed if CUSTOM is used and
class_name is a string literal containing a java class name.

index_name is an identifier, enclosed or not enclosed in double
quotation marks, excluding reserved words.

CREATE KEYSPACE

CREATE (KEYSPACE | SCHEMA) IF NOT EXISTS keyspace_name
WITH REPLICATION = map
AND DURABLE_WRITES = (true | false)

map is described in ALTER KEYSPACE.

CREATE TABLE

CREATE TABLE IF NOT EXISTS keyspace_name.table_name
(column_definition, column_definition, ...)
WITH property AND property ...

column_definition is:
column_name cql_type PRIMARY KEY
| (PRIMARY KEY (partition_key))

Restrictions:
• There should always be exactly one primary key definition.
• cql_type of the primary key must be a custom type or a CQL type (see last page),
other than a collection type.
• cql_type of a collection uses this syntax:

LIST<cql_type>
| SET<cql_type>
| MAP<cql_type, cql_type>

PRIMARY KEY is:
column_name
| (column_name1, column_name2, column_name3 ...)
| ((column_name4, column_name5), column_name6,
column_name7 ...)

column_name1 is the partition key.
column_name2, column_name3 ... are clustering keys.
column_name4, column_name5 are partitioning keys.
column_name6, column_name7 ... are clustering keys.

property is a CQL table storage property or one of these directives:
COMPACT STORAGE
| (CLUSTERING ORDER BY (clustering_key (ASC | DESC), ...))

CREATE TRIGGER

CREATE TRIGGER trigger_name ON table_name
USING 'java_class'

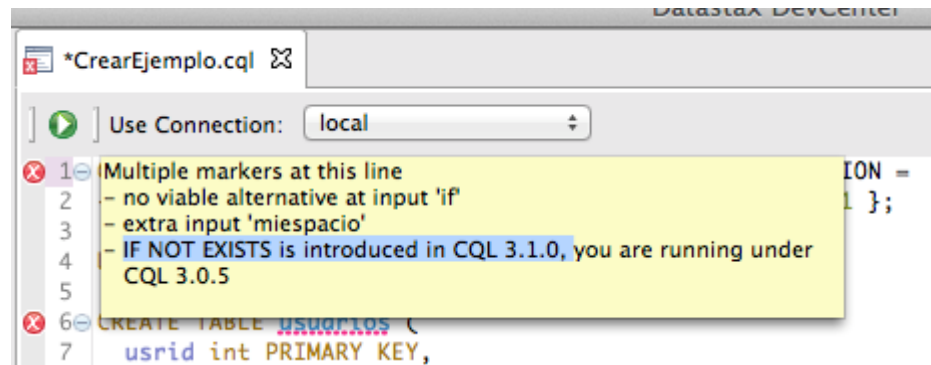
CREATE USER

CREATE USER user_name WITH PASSWORD 'password'
NOSUPERUSER | SUPERUSER

LEGEND:

• Uppercase means literal • Lowercase means not literal • Italics mean optional • The pipe (|) symbol means OR or AND/OR • Ellipsis (...) means repeatable • orange () indicate scope, not literal

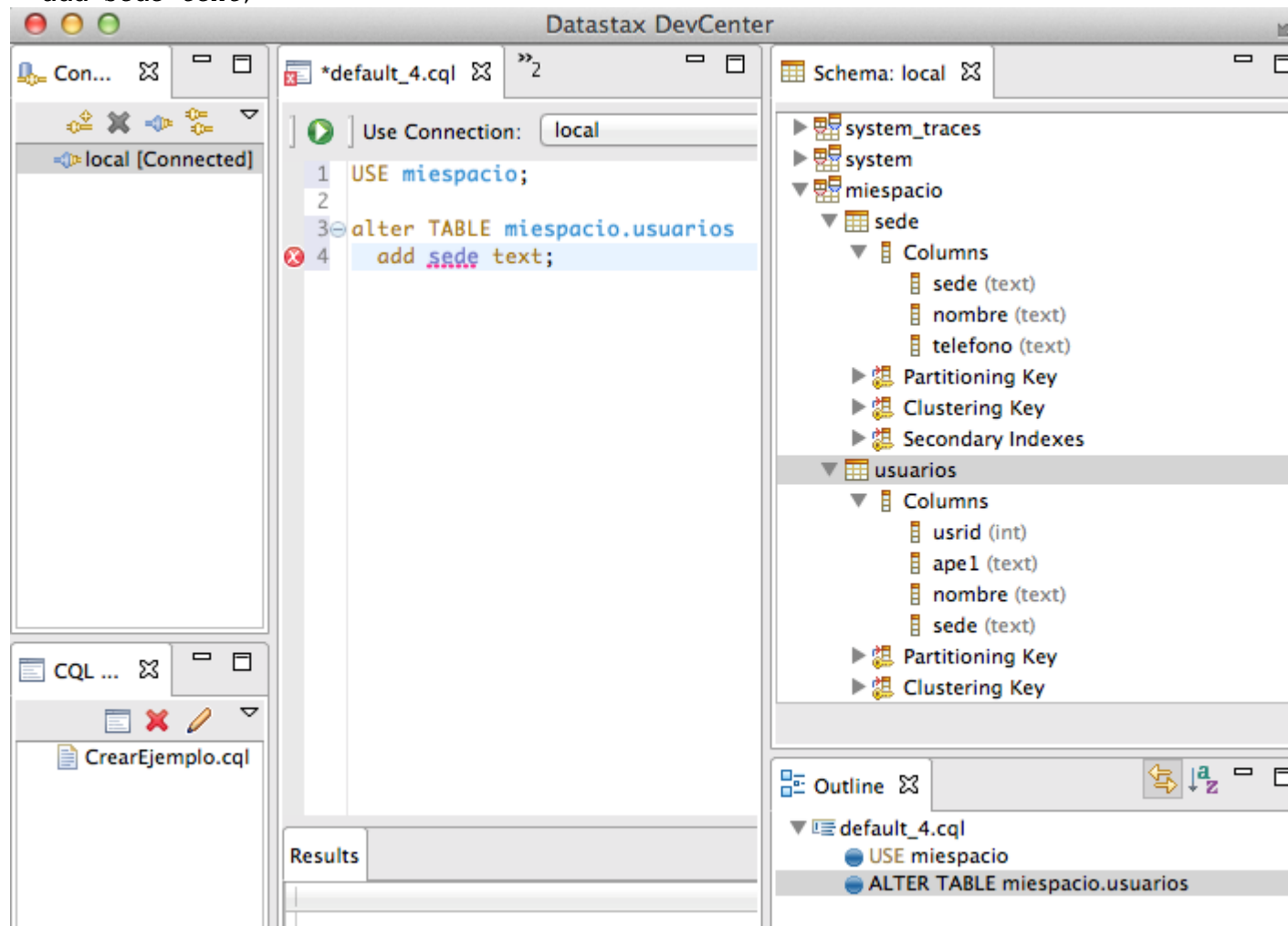
Aunque cuidado con la versión que estamos usando de CQL. Hay cositas interesantes, como sentencias condicionadas (tipo MySQL), para hacer los scripts reentretantes ;-) que solo están disponibles en las últimas versiones.



Probamos a hacer cosas básicas, como añadir columnas. Ésta es una de las gracia de estos tipos de base de datos, que las puedes añadir en cualquier momento porque las columnas se almacenan en sitios distintos.

```
USE miespacio;
```

```
alter TABLE miespacio.usuarios  
add sede text;
```



Comprobamos que se ha creado

The screenshot shows the Databricks DevCenter interface. On the left, there's a 'Connections' panel showing 'local [Connected]'. Below it is a 'CQL Scripts' panel with a file named 'CrearEjemplo.cql'. The main editor shows a CQL script with three lines: 'use miespacio;', 'select * from usuarios;'. Below the editor is a 'Results' table with columns 'usr_id', 'ape1', 'nombre', and 'sede'. The first row contains the values '1', 'Canales', 'Roberto', and '<<null>>'. At the bottom, it says 'Retrieved 1 row(s) in 30 ms'. On the right, there's a 'Schema: local' panel showing a tree view of the database schema, including 'system_traces', 'system', 'miespacio' (with sub-entities 'sede' and 'usuarios'), and an 'Outline' panel showing 'default_5.cql'.

usr_id	ape1	nombre	sede
1	Canales	Roberto	<<null>>

Los comandos de actualización son primos (aunque lejanos :-), hermanos de SQL.

Tenemos que especificar la clave.

```
use miespacio;
```

```
update usuarios set sede = 'Madrid' where usr_id = 1;
```

Estando investigando sobre estructuras más complejas de almacenamiento, durante un rato, tirando de un libro antiguo de Cassandra (del 2010.. ya viejo para esto), hacía referencia a super-columnas que pueden tener a su vez pares clave valor. Esto ha quedado obsoleto por otras variantes en versiones más modernas de CQL. Os recomiendo un pasito por este enlace:

http://www.datastax.com/documentation/cql/3.1/webhelp/index.html#cql/ddl/ddl_intro_c.html#concept_ds_tgl_q2y_zj

62c36092...	3	Roll Away	Back Door Slam	2b09185b...	Outside Woman Blues
62c36092...	2	He Must Obey	Fu Manchu	8a172618...	Moving in Stereo
62c36092...	1	Tres Hombres	ZZ Top	a3e64f8f...	La Grange

Cassandra stores an entire row of data on a node by partition key. If you have too much data in a row and want to spread the row of data over multiple nodes, use a **composite partition key**.

Collection columns

CQL introduces these collection types:

- set
- list
- map

In a relational database, to allow users to have multiple email addresses, you create an email_addresses table having a many-to-one (joined) relationship to a users table. CQL handles the classic multiple email addresses use case, and other use cases, by defining columns as collections. Using the set collection type to solve the multiple email addresses problem is convenient and intuitive.

Another use of a collection type can be demonstrated using the music service example.

Adding a collection to a table

The music service example includes the capability to tag the songs. From a relational standpoint, you can think of storage engine rows as partitions, within which (object) rows are clustered. To tag songs, use a collection set. Declare the collection set using the CREATE TABLE or ALTER TABLE statements. Because the songs table already exists from the earlier example, just alter that table to add a collection set, tags:

```
ALTER TABLE songs ADD tags set<text>;
```

Updating a collection

Update the songs table to insert the tags data:

```
UPDATE songs SET tags = tags + {'2007'}  
WHERE id = 8a172618-b121-4136-bb10-f665cfc469eb;  
UPDATE songs SET tags = tags + {'covers'}  
WHERE id = 8a172618-b121-4136-bb10-f665cfc469eb;  
UPDATE songs SET tags = tags + {'1973'}  
WHERE id = a3e64f8f-bd44-4f28-b8d9-6938726e34d4;  
UPDATE songs SET tags = tags + {'blues'}  
WHERE id = a3e64f8f-bd44-4f28-b8d9-6938726e34d4;  
UPDATE songs SET tags = tags + {'rock'}  
WHERE id = 7db1a490-5878-11e2-bcfd-0800200c9a66;
```

A music reviews list and a schedule (map collection) of live appearances can be added to the table:

```
ALTER TABLE songs ADD reviews list<text>;  
ALTER TABLE songs ADD venue map<timestamp, text>;
```


Vamos a modificar un poquito la estructura para crear un nivel superior de anidamiento de la información. Sobre todo tiene mucho sentido cuando esa información es única. Vamos a añadir los gastos asociados a un individuo.

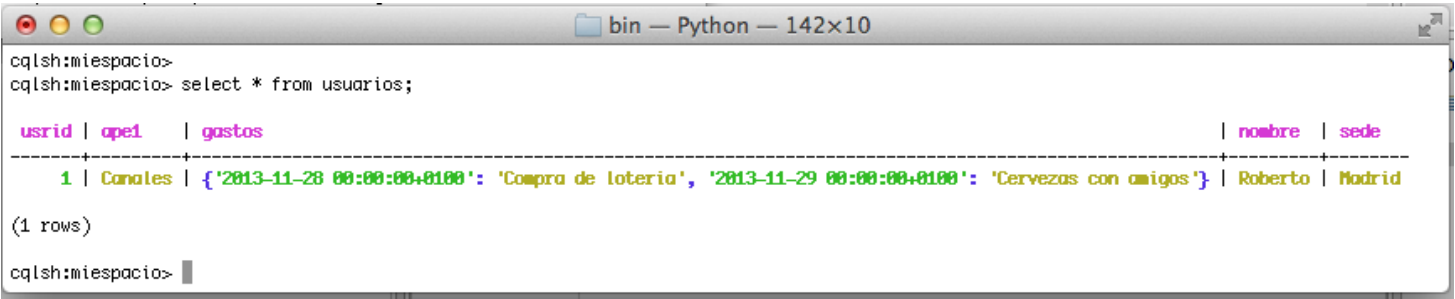
```
use miespacio;

ALTER TABLE usuarios ADD gastos map;

update usuarios set gastos =
{ '2013-11-28'  : 'Compra de loteria',
  '2013-11-29'  : 'Cervezas con amigos'}
where usrid = 1;

// select * from usuarios;
select gastos from usuarios where usrid = 1;
```

Podemos ver el resultado desde la consola cql desde la consola.



Bueno, para un primer tutorial y tomar el contacto creo que ya es suficiente. Con un poquito más de esfuerzo podemos partir de este punto para atacar tópicos más avanzados de Apache Cassandra.

9. Conclusiones.

A mi me gusta trabajar con tecnologías que ya van siendo demandadas en los entornos en los que trabajo: investigar pronto pero usarlas/proponerlas cuando están estables y hay referencias en el mercado cercano. Las tendencias están muy bien para captar la atención en charlas y conferencias pero dan poco de comer ;-)

Del BigData y bases de datos NoSql se lleva hablando durante meses (digo esto porque los años entran en una mano), pero siempre hay dudas sobre la madurez de las tecnologías.

Parece que ese momento para invertir recursos formativos ya ha llegado y que va a ser imparable el uso más habitual en clientes de estas tecnologías como complementarias para las existentes: en ningún caso sustitutivas porque se usan para cosas diferentes que las bases de datos relacionales.