

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.

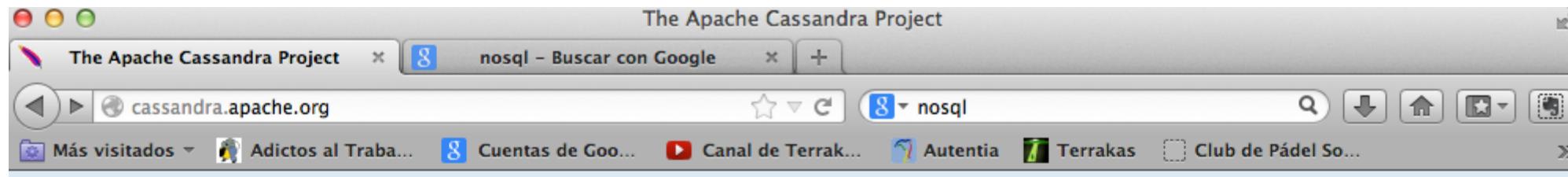
Me gusta saber las cosas de primera mano y "oler" la madurez de los productos, entornos y tecnologías y por ello vamos a dar los primeros pasos con Cassandra que es una base de datos "orientada a columnas".

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/

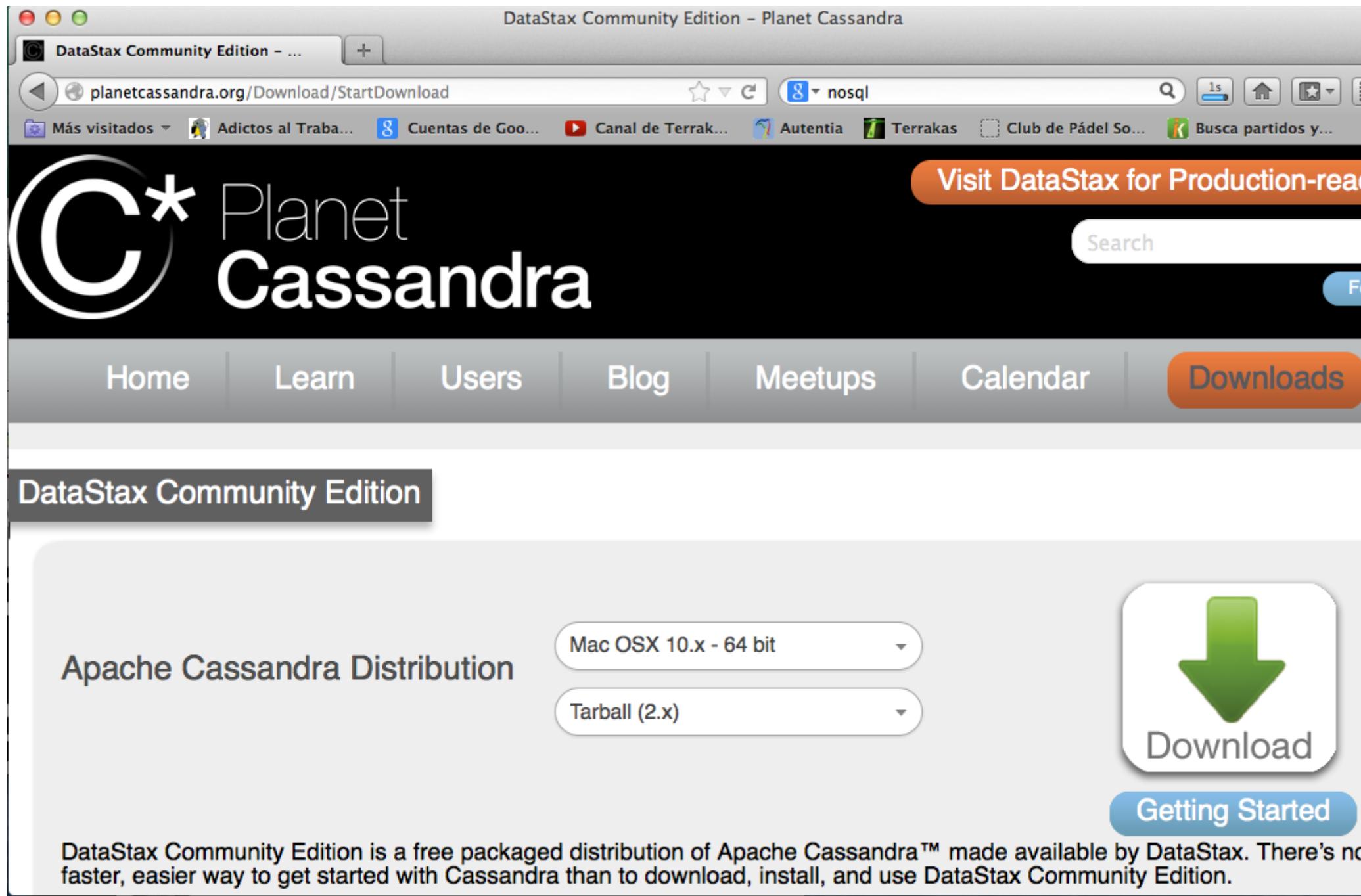


The Apache Cassandra database is the right choice when you need scalability and high availability without compromising performance. [Linear scalability](#) and proven fault-tolerance on commodity hardware or cloud infrastructure make it the perfect platform for mission-critical data. Cassandra's support 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.

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





4. Preparación del entorno.

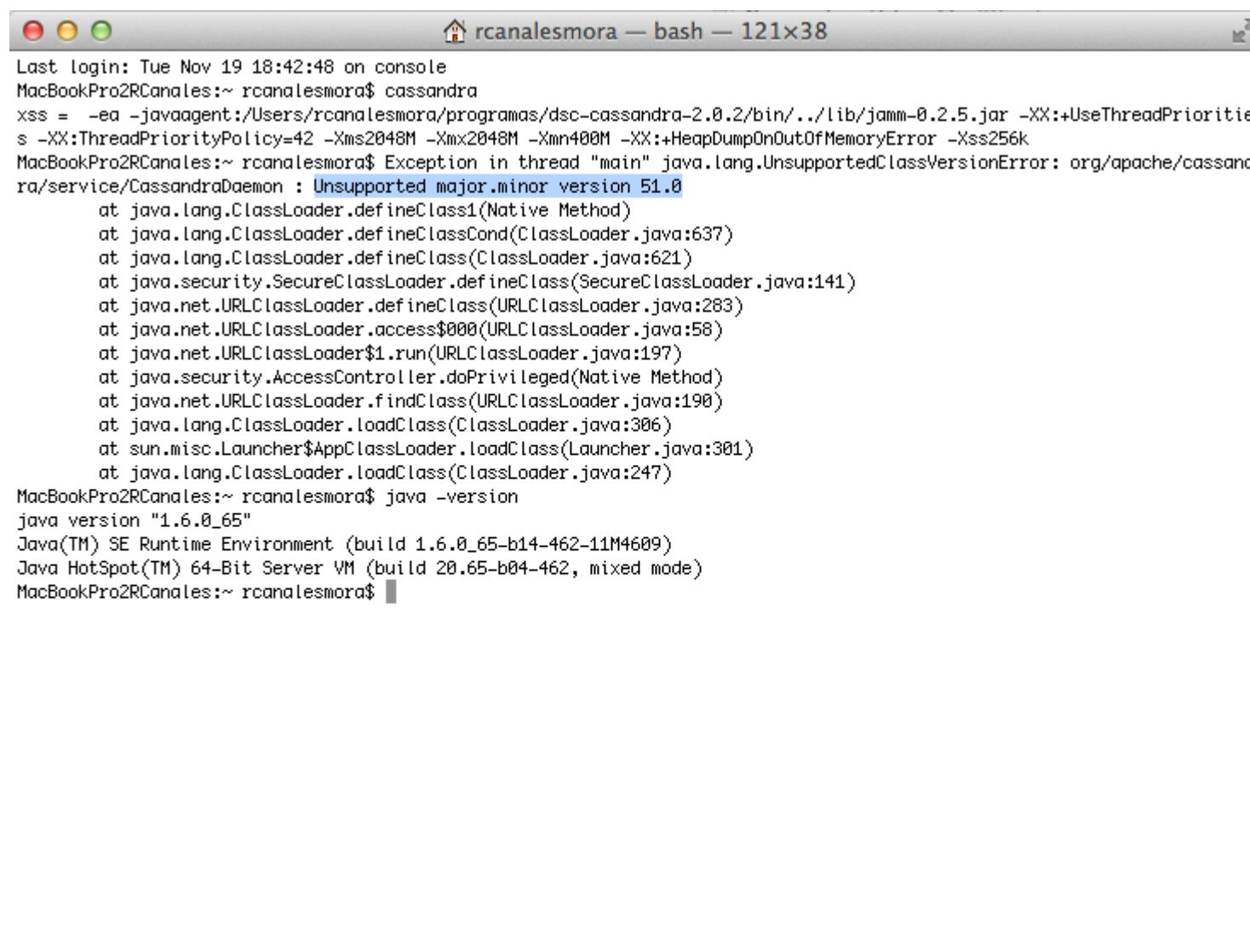
Ejecutamos los Scripts para instalar. Simplemente seguimos las instrucciones para crear un directorio de trabajo, desplegar los binarios, añadir el path a la variable \$PATH del entorno, etc.

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(paths.d/  
sh-3.2# echo "/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin" > cassandra  
sh-3.2# ls  
cassandra  mongodb  
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.



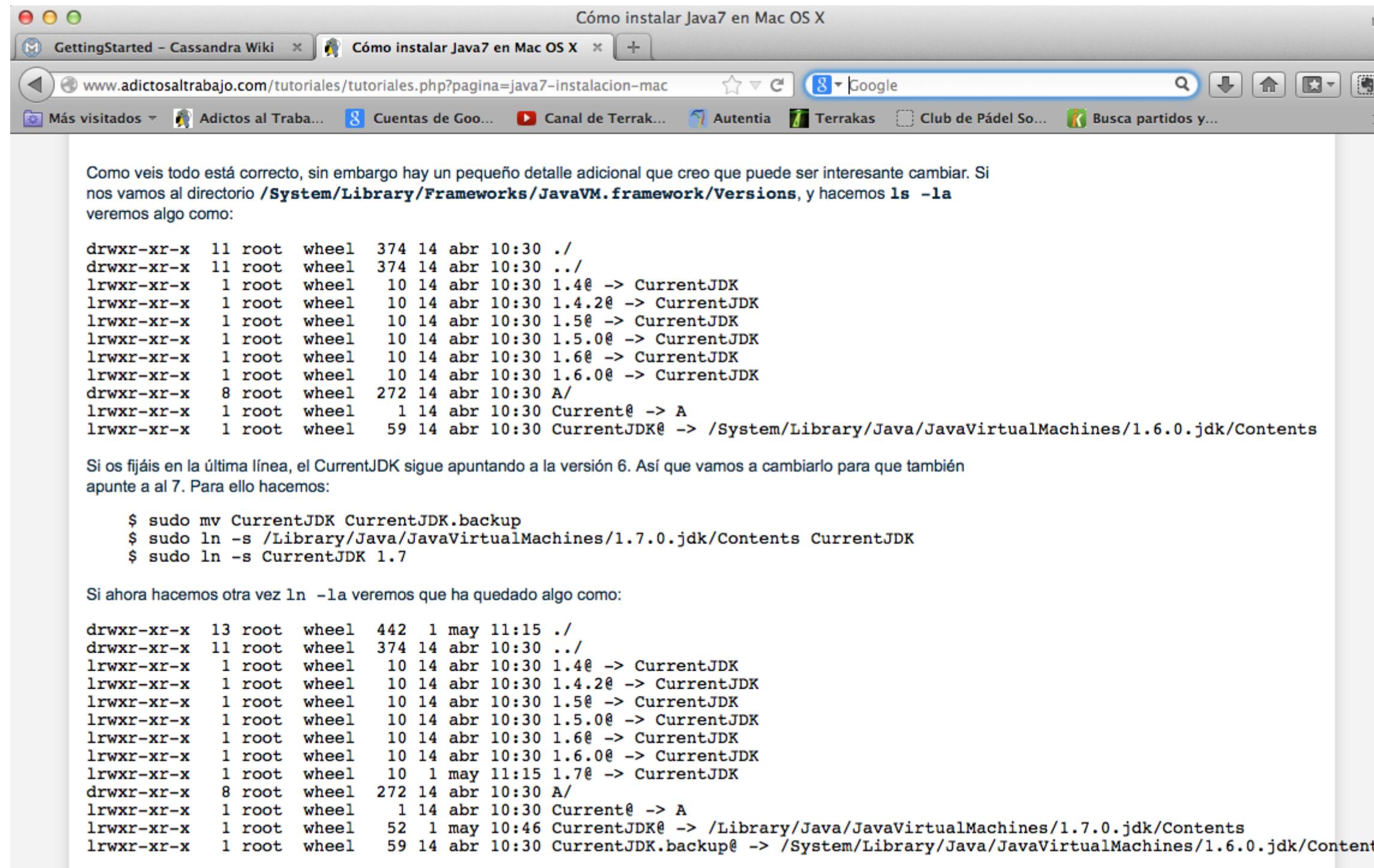
```
Last login: Tue Nov 19 18:42:48 on console  
MacBookPro2RCanales:~ rcanalesmora$ cassandra  
xss = -ea -javaagent:/Users/rcanalesmora/programas/dsc-cassandra-2.0.2/bin/../lib/jamm-0.2.5.jar -XX:+UseThreadPriorities -XX:ThreadPriorityPolicy=42 -Xms2048M -Xmx2048M -Xmn400M -XX:+HeapDumpOnOutOfMemoryError -Xss256k  
MacBookPro2RCanales:~ rcanalesmora$ Exception in thread "main" java.lang.UnsupportedClassVersionError: org/apache/cassandra/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.



The screenshot shows a Mac OS X desktop with a browser window open. The title bar says "Cómo instalar Java7 en Mac OS X". The address bar shows the URL "www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=java7-instalacion-mac". The page content discusses the steps to change the Java version from 6 to 7 by modifying the "/System/Library/Frameworks/JavaVM.framework/Versions" directory. It includes a terminal command to move the "CurrentJDK" symbolic link and replace it with "1.7". The terminal output shows the file listing for both versions 6 and 7.

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:

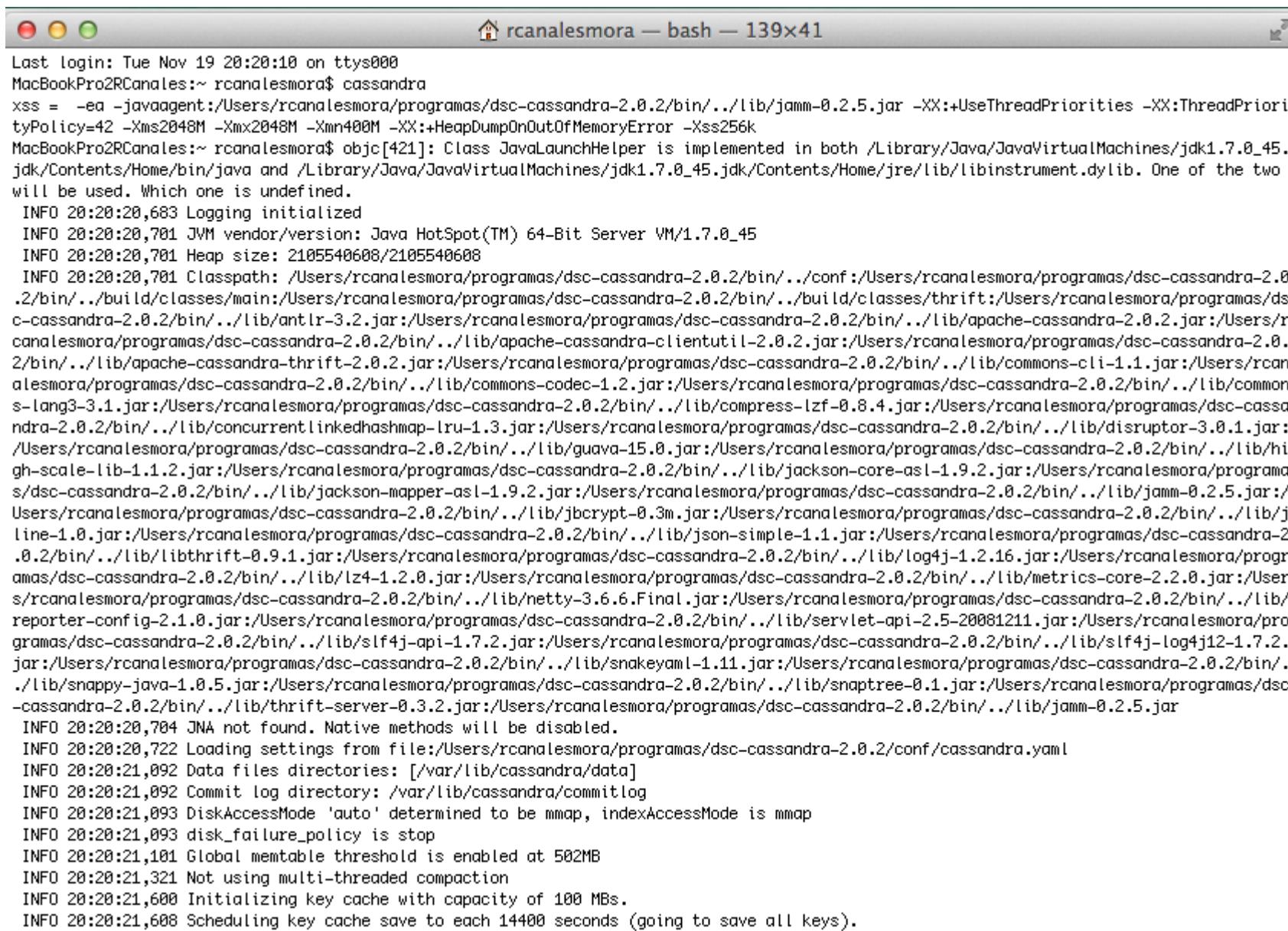
```
$ sudo mv CurrentJDK CurrentJDK.backup
$ sudo ln -s /Library/Java/JavaVirtualMachines/1.7.0.jdk/Contents CurrentJDK
$ sudo ln -s CurrentJDK 1.7
```

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.

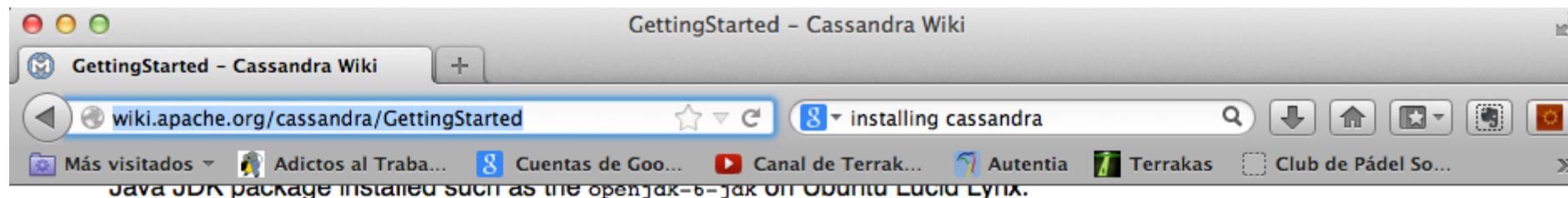


A screenshot of a Mac OS X terminal window titled "r canalesmora — bash — 139x41". The window displays the startup log for the Cassandra database. The log starts with the last login information and proceeds through various Java and Cassandra initialization messages, including classpath details and configuration settings. The log ends with the start of the key cache scheduling process.

```
Last login: Tue Nov 19 20:20:10 on ttys000
MacBookPro2RCanales:~ rcanalesmora$ cassandra
xss = -ea -javaagent:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/jamm-0.2.5.jar -XX:+UseThreadPriorities -XX:ThreadPriorityPolicy=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/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../conf:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../build/classes/main:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../build/classes/thrift:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/antlr-3.2.jar:/Users/r canalesmora/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/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/apache-cassandra-thrift-2.0.2.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/commons-cli-1.1.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/commons-codec-1.2.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/commons-lang3-3.1.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/compress-lzf-0.8.4.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/concurrentlinkedhashmap-lru-1.3.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/disruptor-3.0.1.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/guava-15.0.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/igh-scale-lib-1.1.2.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/jackson-core-asl-1.9.2.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/jamm-0.2.5.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/jackson-mapper-asl-1.9.2.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/jamm-0.2.5.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/jbcrypt-0.3m.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/json-simple-1.1.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/log4j-1.2.16.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/libthrift-0.9.1.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/metrics-core-2.2.0.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/netty-3.6.6.Final.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/reporter-config-2.1.0.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/servlet-api-2.5-20081211.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/slf4j-api-1.7.2.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/slf4j-log4j12-1.7.2.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/snakeyaml-1.11.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/snappy-java-1.0.5.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/snptree-0.1.jar:/Users/r canalesmora/programas/dsc-cassandra-2.0.2/bin/..../lib/thrift-server-0.3.2.jar:/Users/r canalesmora/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/r canalesmora/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 chorradas: 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 juguetear 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 usuarios;
Bad Request: unconfigured columnfamily usuarios
cqlsh:miespacio> select * from usuarios;

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

(1 rows)

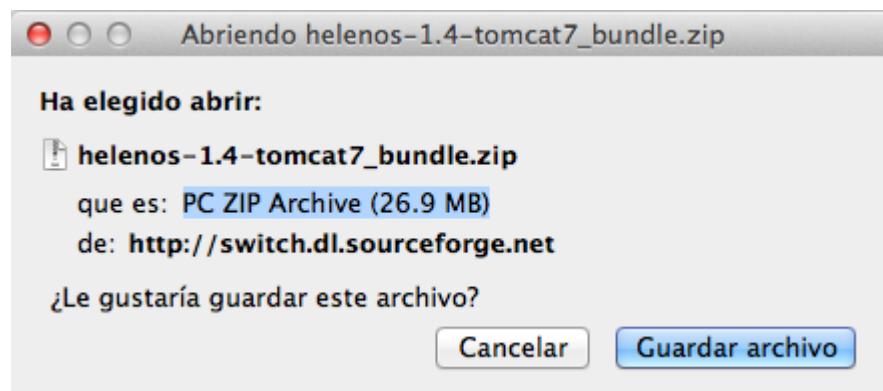
cqlsh:miespacio>

```

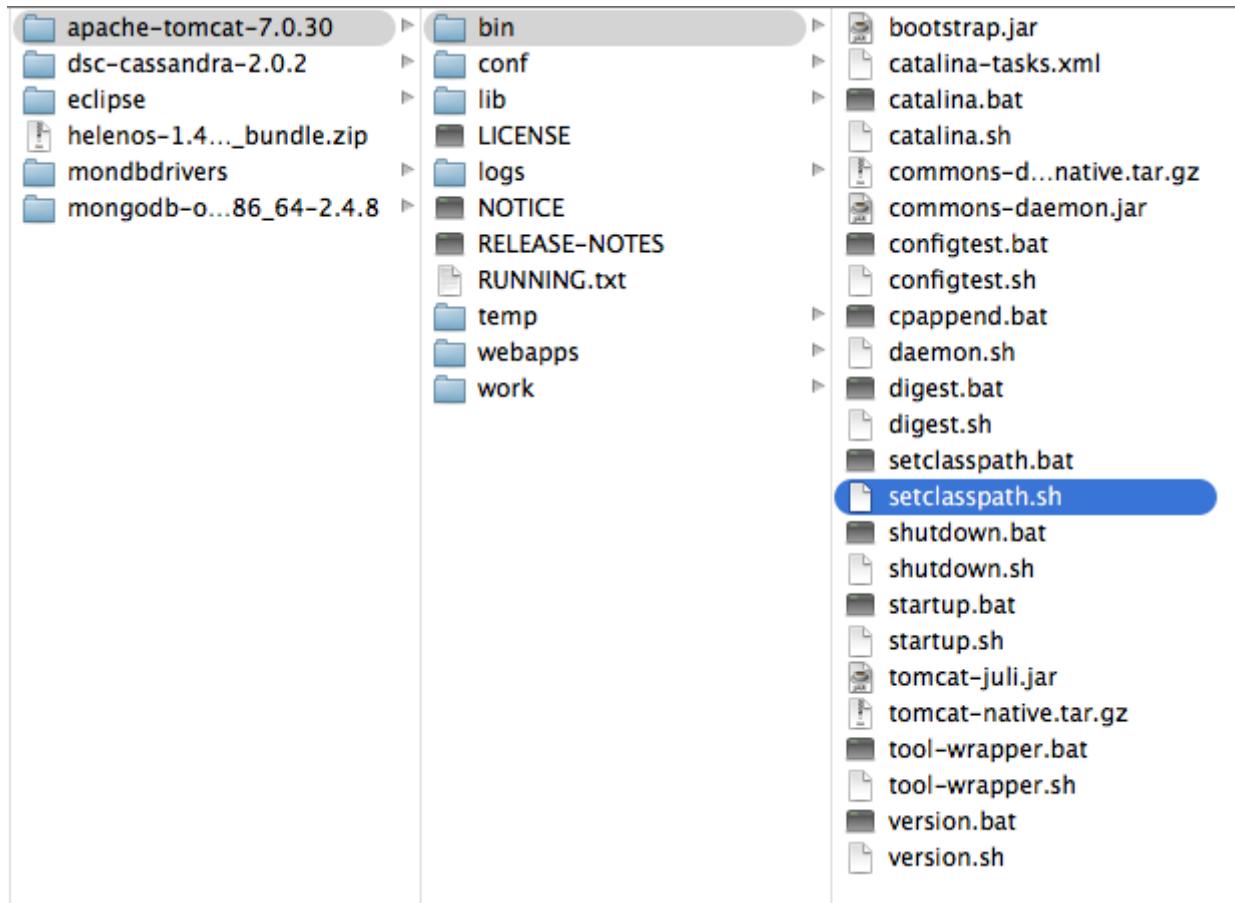
7. Instalación de la consola Helenos para Cassandra.

Como ya hice con MongoDB, voy a instalar me 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 descomprimimos.



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

```
bin — bash — 80x24
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**.

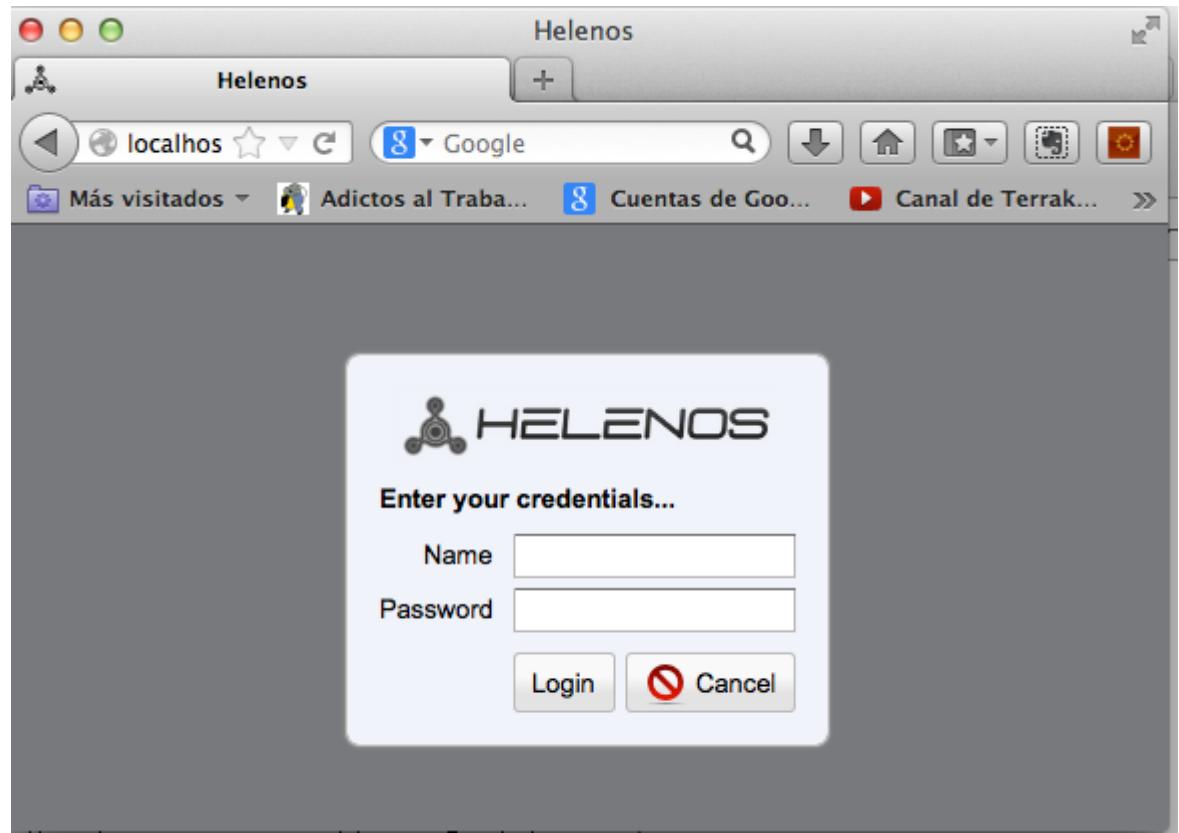
Editamos el fichero **.bash_profile**. Aquí tenéis una captura de la secuencia de comandos.

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

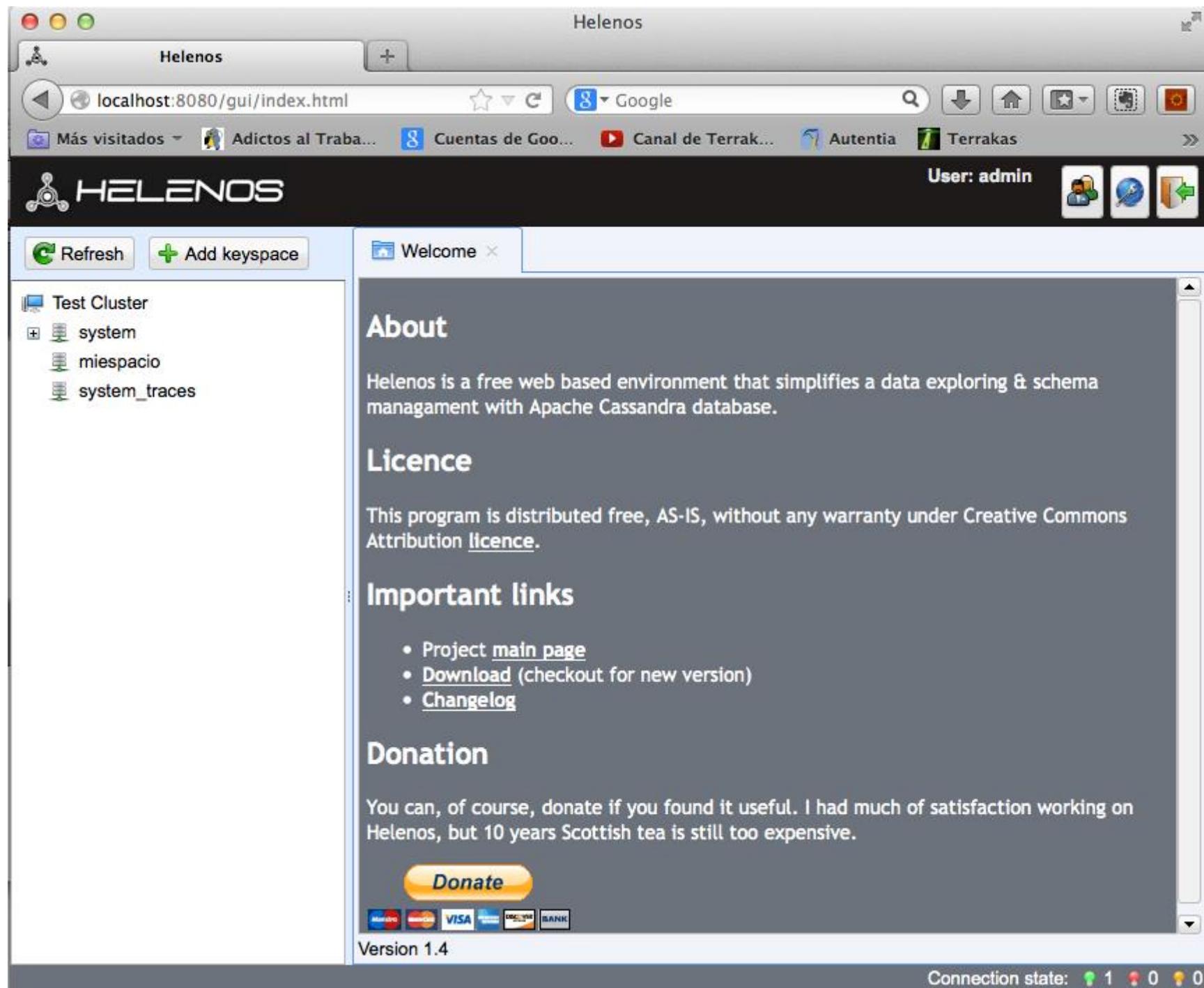
Arrancamos el servidor de nuevo y ya funciona correctamente.

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

Atacamos con un navegador por http a localhost:8080.

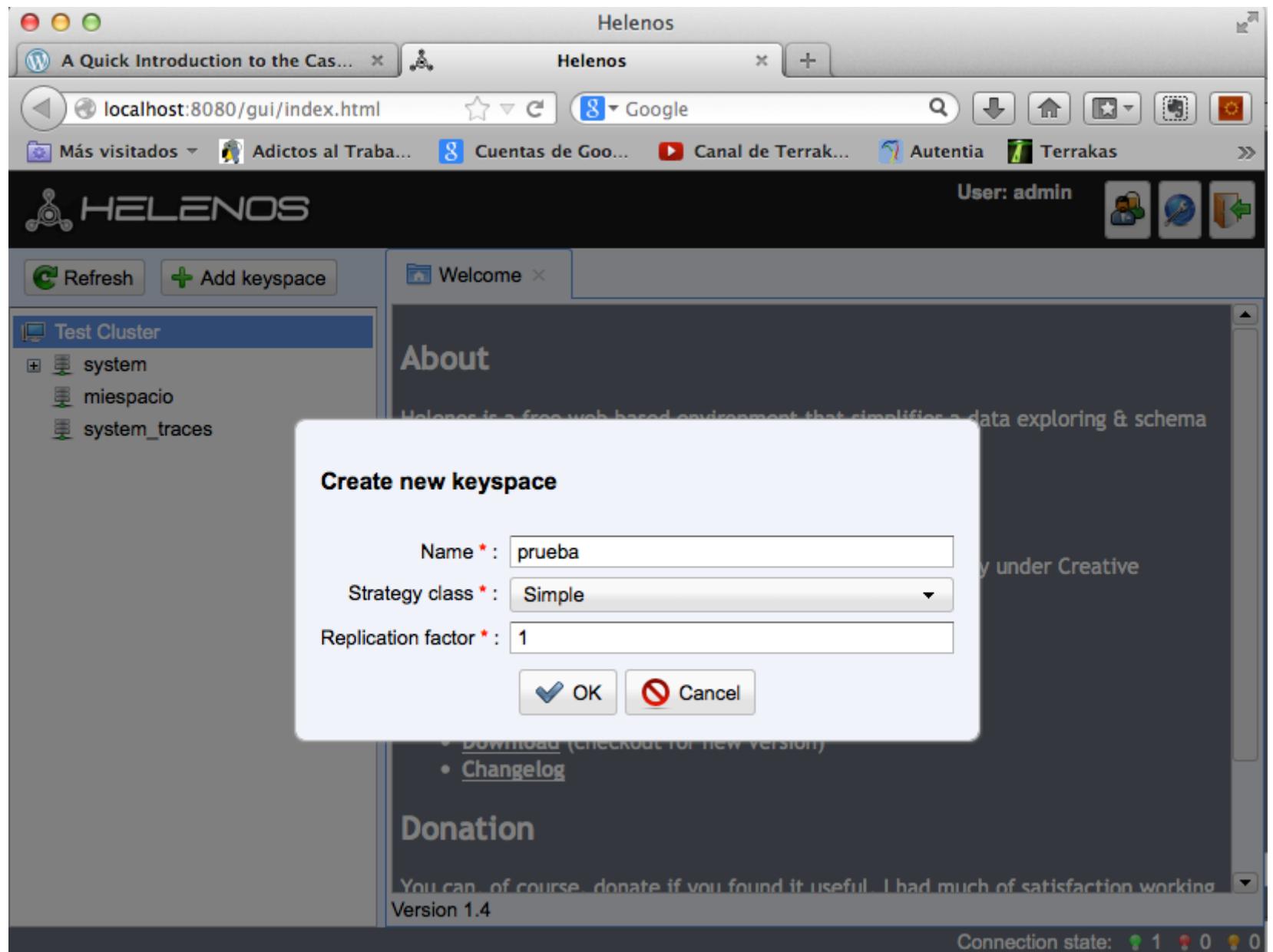


Metemos la contraseña por defecto: admin admin

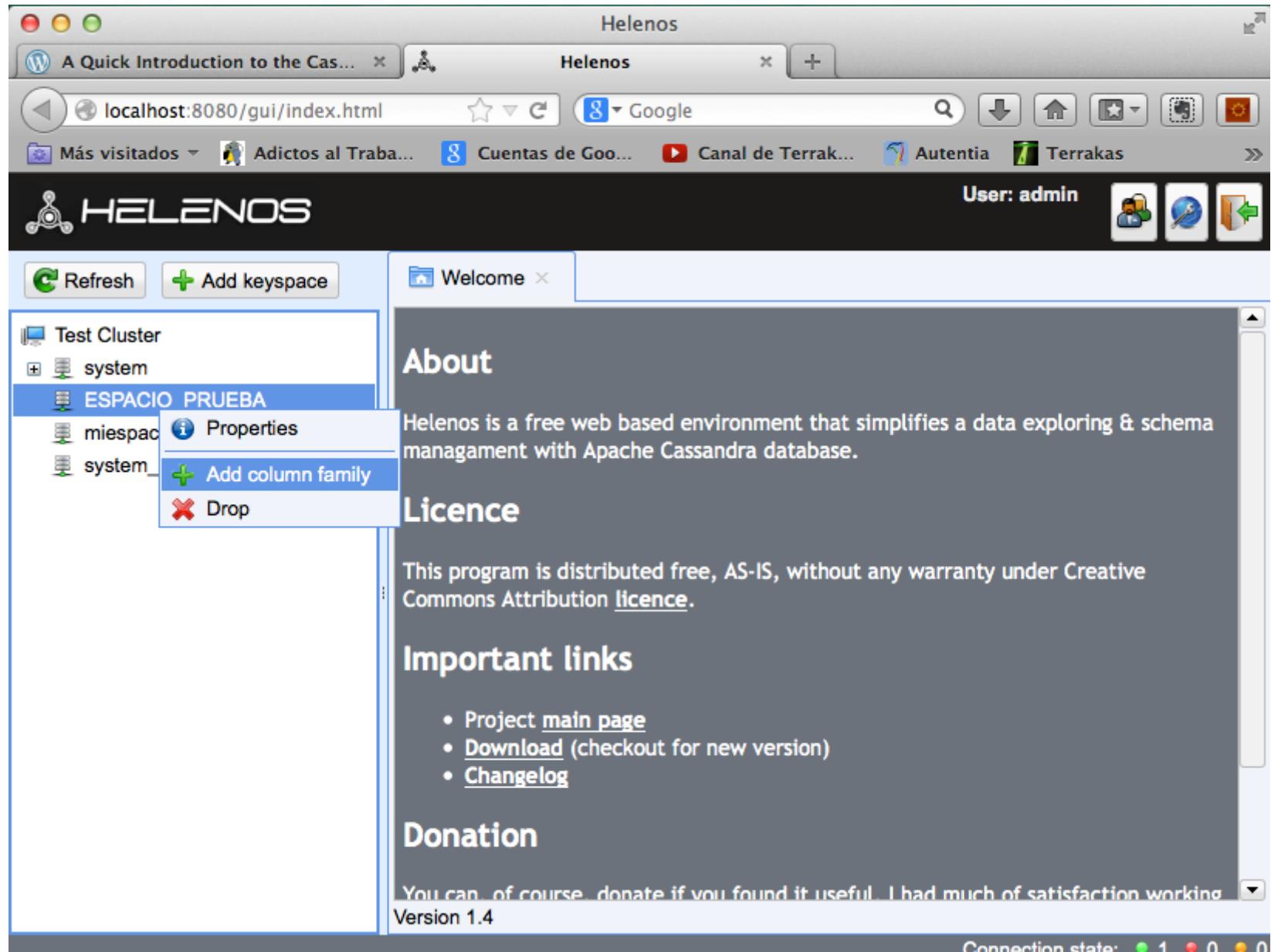


Bueno, ya tenemos otra herramienta con la que jugar.

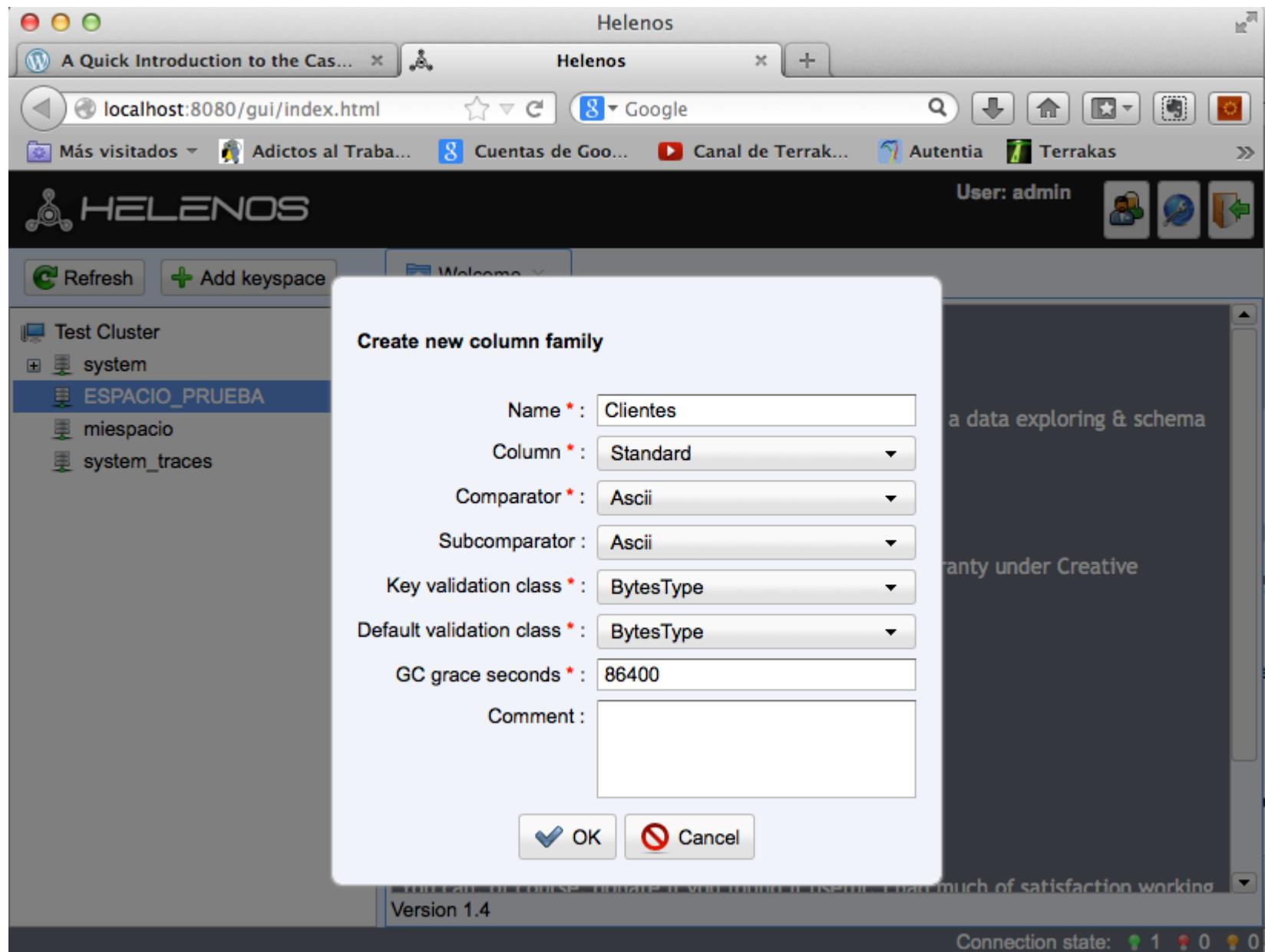
Es interesante sobre todo para ver los parámetros de inicialización.



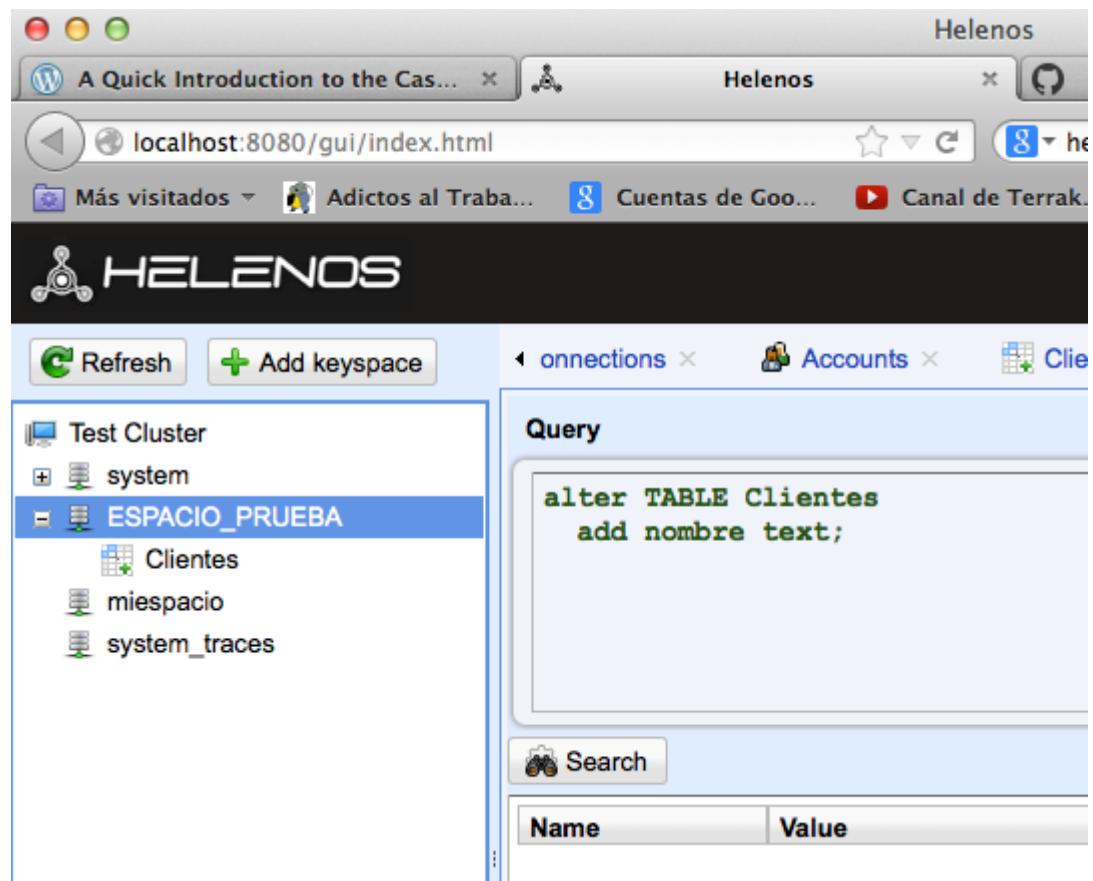
Añadimos familia de columnas (lo que vendría a ser conceptualmente una tabla pero con sus matices de que cada columna se almacena o puede almacenar por separado).



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.

The screenshot shows the Helenos web interface for managing a Cassandra cluster. The title bar says "Helenos". The left sidebar lists "Test Cluster", "system", and "ESPACIO_PRUEBA". Under "ESPACIO_PRUEBA", "Clientes" is selected, showing options: "Properties", "Browse data", "CQL query", "Drop", and "Truncate". The main content area displays the "Keyspace: ESPACIO_PRUEBA" page. It shows the "Basic" section with "Strategy class: org.apache.cassandra.locator.SimpleStrategy" and "Replication factor: 1". Below this is the "Column families" section, which contains a table:

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).

The screenshot shows a web browser window titled "Helenos" with the URL "localhost:8080/gui/index.html". The browser has several tabs open, including "A Quick Introduction to the Cas...", "Helenos", "tomekkup/helenos · GitHub", and "helenos cassandra". The main content area is titled "HELENOS" and shows a sidebar with a tree view of a Cassandra cluster structure. The tree includes "Test Cluster", "system", "ESPACIO_PRUEBA" (which is expanded), "Cuentas de Goo...", "Canal de Terrak...", "Autentia", "Terrakas", and "Club de Pádel So...". Under "ESPACIO_PRUEBA", "Cuentas de Goo..." is expanded, showing "Cuentas de Goo..." and "Canal de Terrak...". The "Cuentas de Goo..." node is selected. The main panel contains a "Query" section with the following code:

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

Below the query is a "Consistency level:" dropdown set to "A". At the bottom is a table with three columns: "Name", "Value", and "Clock".

Vemos el resultado.

The screenshot shows the HELENOS Cassandra management interface. The browser address bar displays 'localhost:8080/gui/index.html' and the title 'helenos cassandra'. The main window has a header 'HELENOS' with a logo. On the left, a sidebar shows a tree structure with 'Test Cluster', 'system', and 'ESPACIO_PRUEBA' selected. Under 'ESPACIO_PRUEBA', 'Clientes' is selected, and its properties are listed: 'miespacio', 'Properties', 'Browse data', 'CQL query' (which is currently selected), 'Drop', and 'Truncate'. The central area has tabs for 'ESPACIO_PRUEBA : Clientes' and 'ESPACIO_PRUEBA : Clientes'. A 'Query' section contains the CQL query: 'SELECT * FROM Clientes LIMIT 100;'. Below it is a 'Search' section. A table displays the results of the query:

Name	Value	Clock
nombre	Roberto Canales	December 2, 45882 00:49:33

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 las tablas e índices (por llamarlos así, con sus matices ;-)

GettingStarted - Cassandra Wiki DataStax DevCenter : DataStax
www.datastax.com/what-we-offer/products-services/devce gui enviromente cassandra
Más visitados Adictos al Traba... Cuentas de Goo... Canal de Terrak... Autentia Terrakas Club de Pádel So...
Support Blog Resources Careers Contact Us 
DATASTAX BIG DATA CHALLENGES CUSTOMERS WHAT WE OFFER DEVELOPER CENTER
COMPANY DOWNLOAD
Overview DataStax Enterprise **DataStax DevCenter** DataStax OpsCenter Support Training Consulting
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.

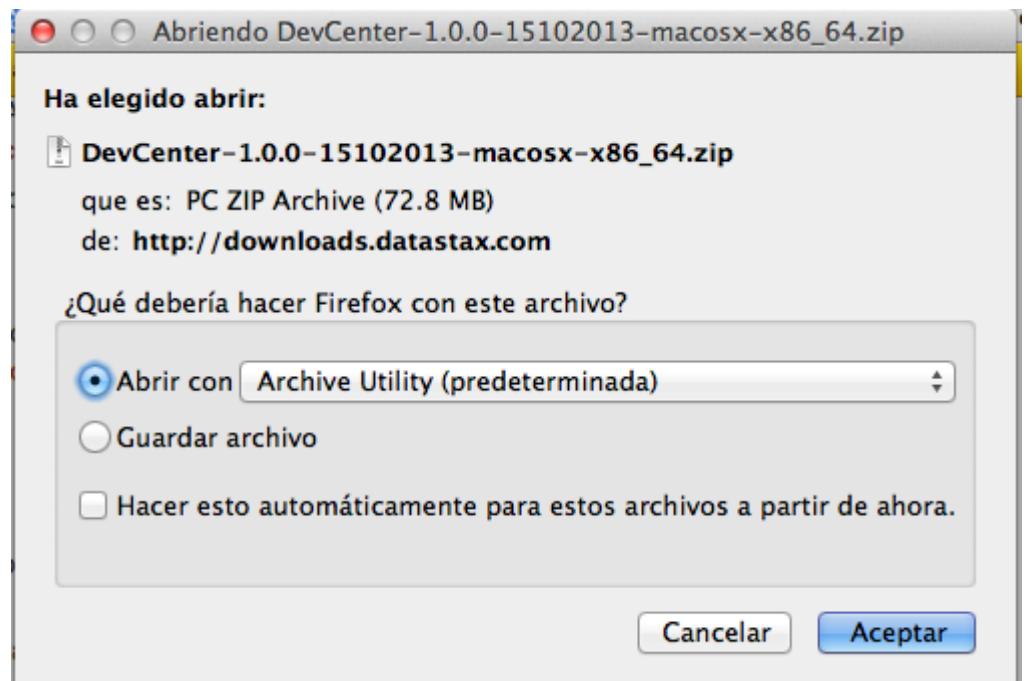
Free Virtual Training

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

[Start Learning Now >](#)

[DEVELOPER CENTER](#)

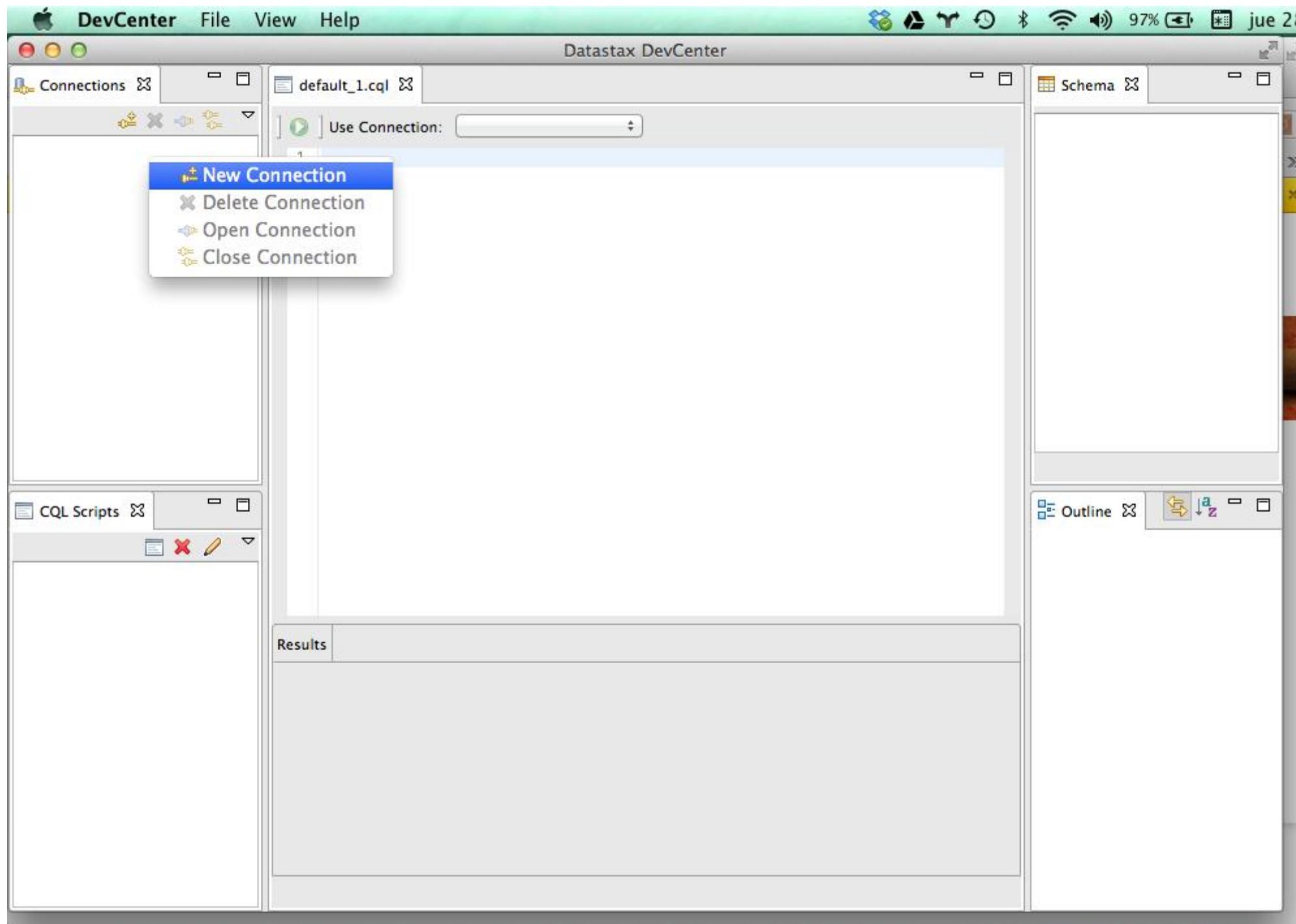
Descargamos el entorno:



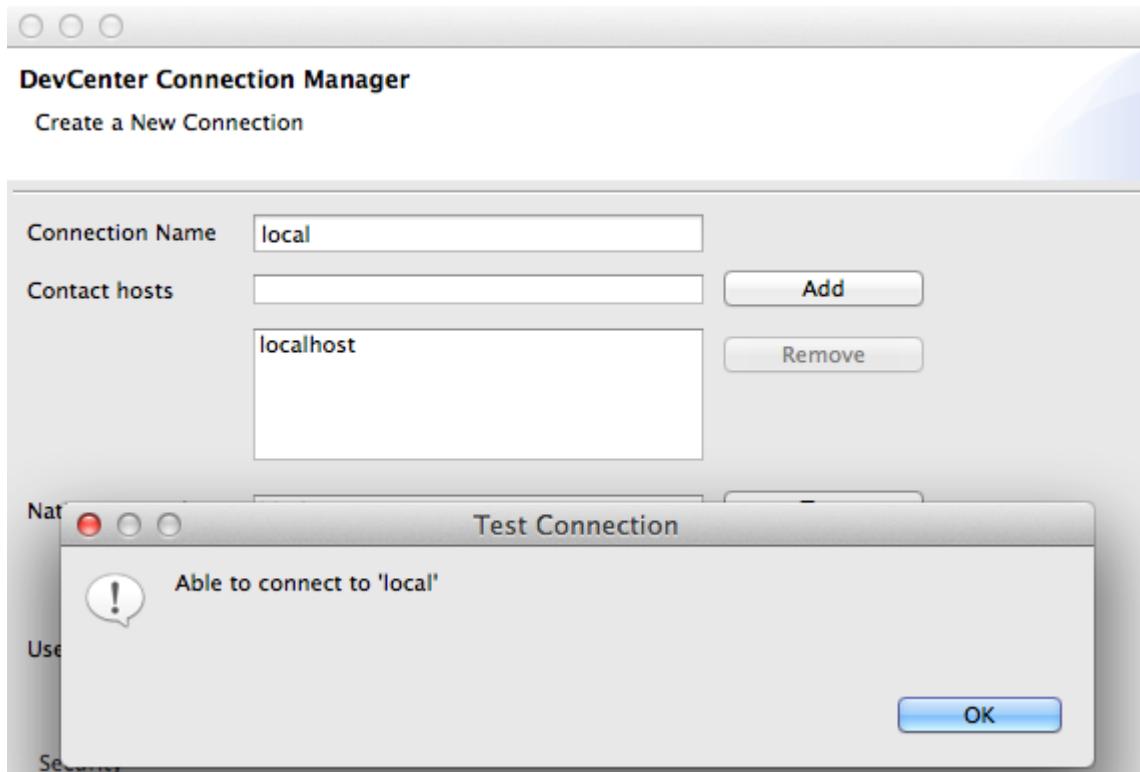
En Mac es una aplicación nativa



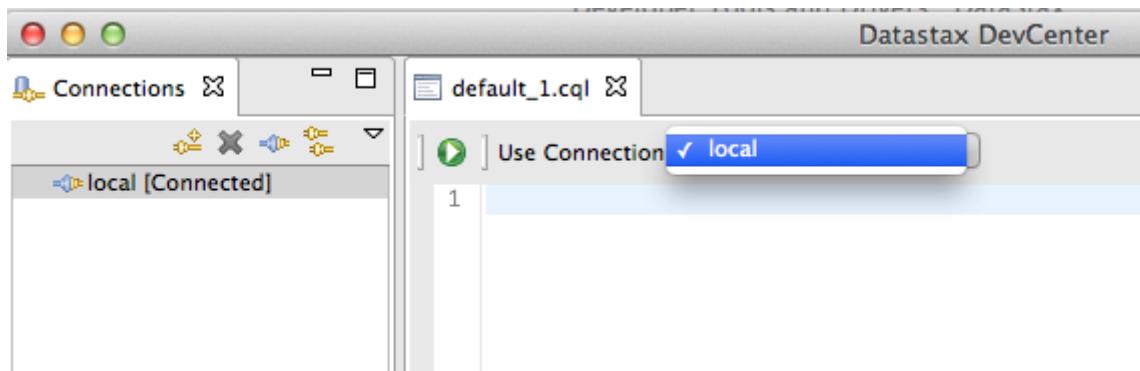
Si estamos acostumbrados a Eclipse es muy predecible. Creamos una nueva conexión.



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, ape1)
VALUES (1, 'Roberto', 'Canales');
```

```
select * from usuarios;
```

The screenshot shows the DataStax DevCenter interface. On the left, there's a 'CQL Scripts' panel with a toolbar. In the center, a main window displays a CQL script named 'default_1.cql'. The script includes creating a keyspace, using it, creating a table with columns usrid, nombre, and ape1, inserting a row, and selecting all from the table. Below the script is a 'Results' table showing the inserted data. On the right, a 'Schema: local' browser shows the keyspace 'miespacio' containing the 'usuarios' table with its columns. At the bottom, an 'Outline' panel lists the statements from the script.

```
*default_1.cql
```

```
Use Connection: local
```

```
1 CREATE KEYSPACE miespacio WITH REPLICATION =
2 { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
3
4 USE miespacio;
5
6 CREATE TABLE usuarios (
7   usrid int PRIMARY KEY,
8   nombre text,
9   ape1 text
10 );
11
12 INSERT INTO usuarios (usrid,nombre, ape1)
13   VALUES (1, 'Roberto', 'Canales');
14
15 select * from usuarios;
```

usrid	ape1	nombre
1	Canales	Roberto

Retrieved 1 row(s) in 662 ms

```
Schema: local
```

```
miespacio
```

```
usuarios
```

```
Columns
```

```
usrid (int)
```

```
ape1 (text)
```

```
nombre (text)
```

```
Partitioning Key
```

```
Clustering Key
```

```
Secondary Indexes
```

```
Outline
```

```
default_1.cql
```

- CREATE KEYSPACE miespacio
- USE miespacio
- CREATE TABLE usuarios
- INSERT INTO usuarios
- SELECT FROM usuarios

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_card.pdf

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

Más visitados Adictos al Traba... Google Apps

Este documento PDF podría no mostrarse correctamente.

Página: 1 de 1 Tamaño automático Abrir con un visor diferente

Apache Cassandra Query Language (CQL)

REFERENCE GUIDE - P.1

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

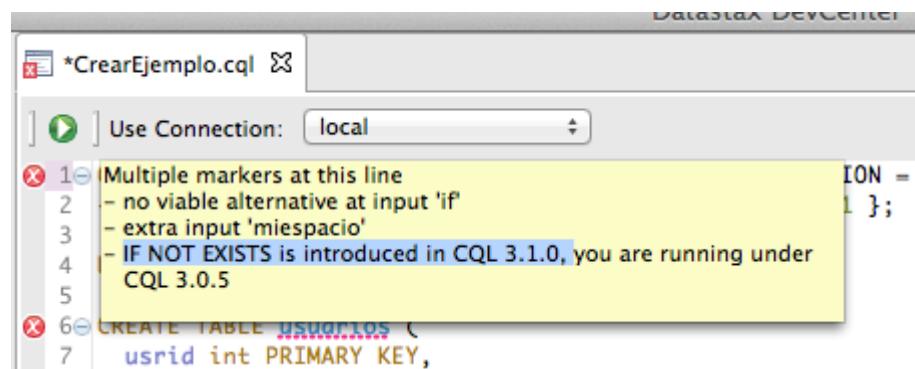
```
CREATE TRIGGER trigger_name ON table_name  
USING 'java_class'
```

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 reentrantantes ;-) que solo están disponibles en las últimas versiones.



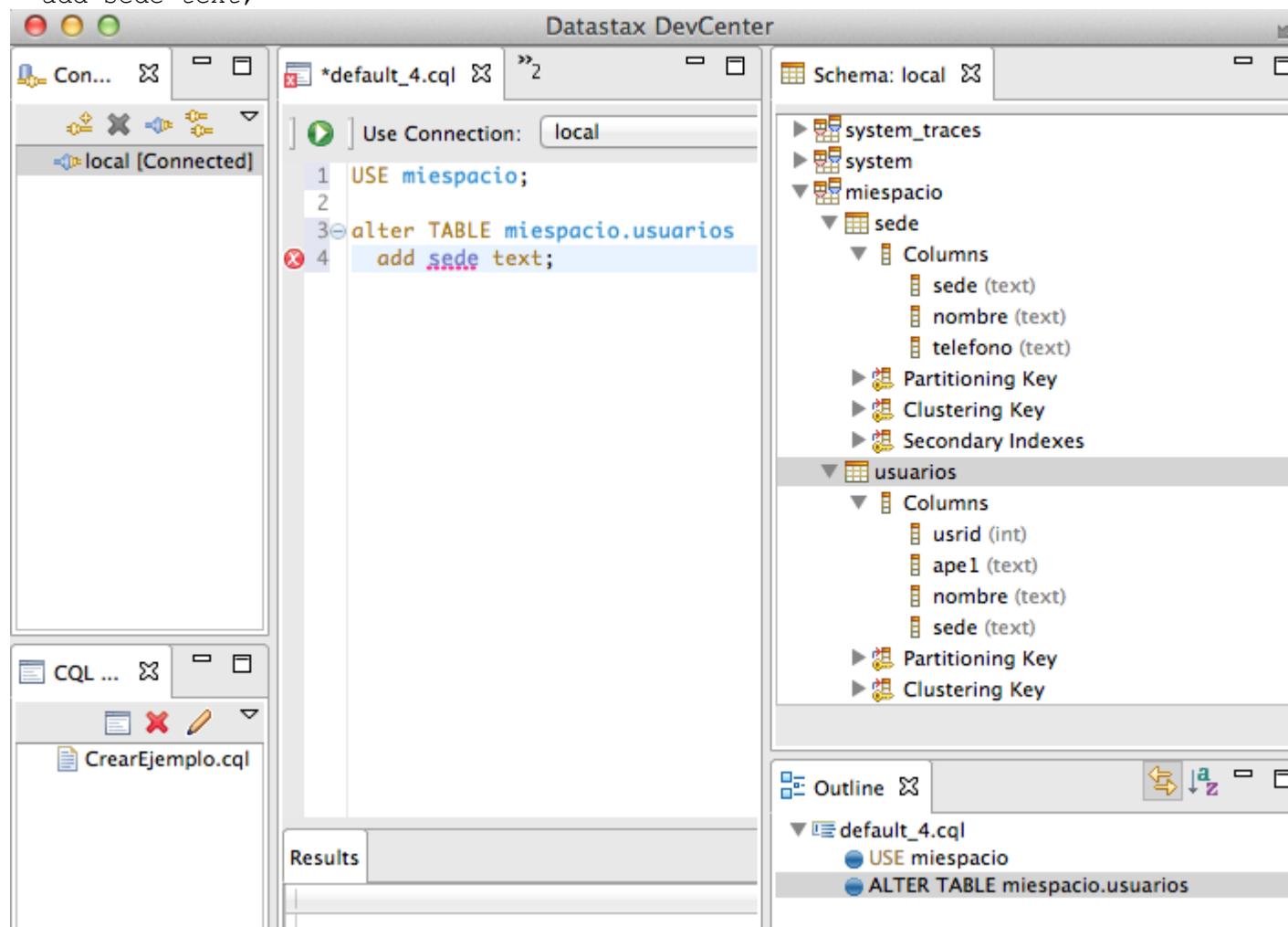
The screenshot shows a Datastax DevCenter interface with a code editor window titled "CrearEjemplo.cql". The connection is set to "local". The code contains several errors:

```
1 Multiple markers at this line
2 - no viable alternative at input 'if'
3 - extra input 'miespacio'
4 - IF NOT EXISTS is introduced in CQL 3.1.0, you are running under
5   CQL 3.0.5
6 CREATE TABLE usuarios (
7   usrid int PRIMARY KEY,
```

A yellow tooltip box highlights the first three error messages: "Multiple markers at this line", "- no viable alternative at input 'if'", and "- extra input 'miespacio'".

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;
```



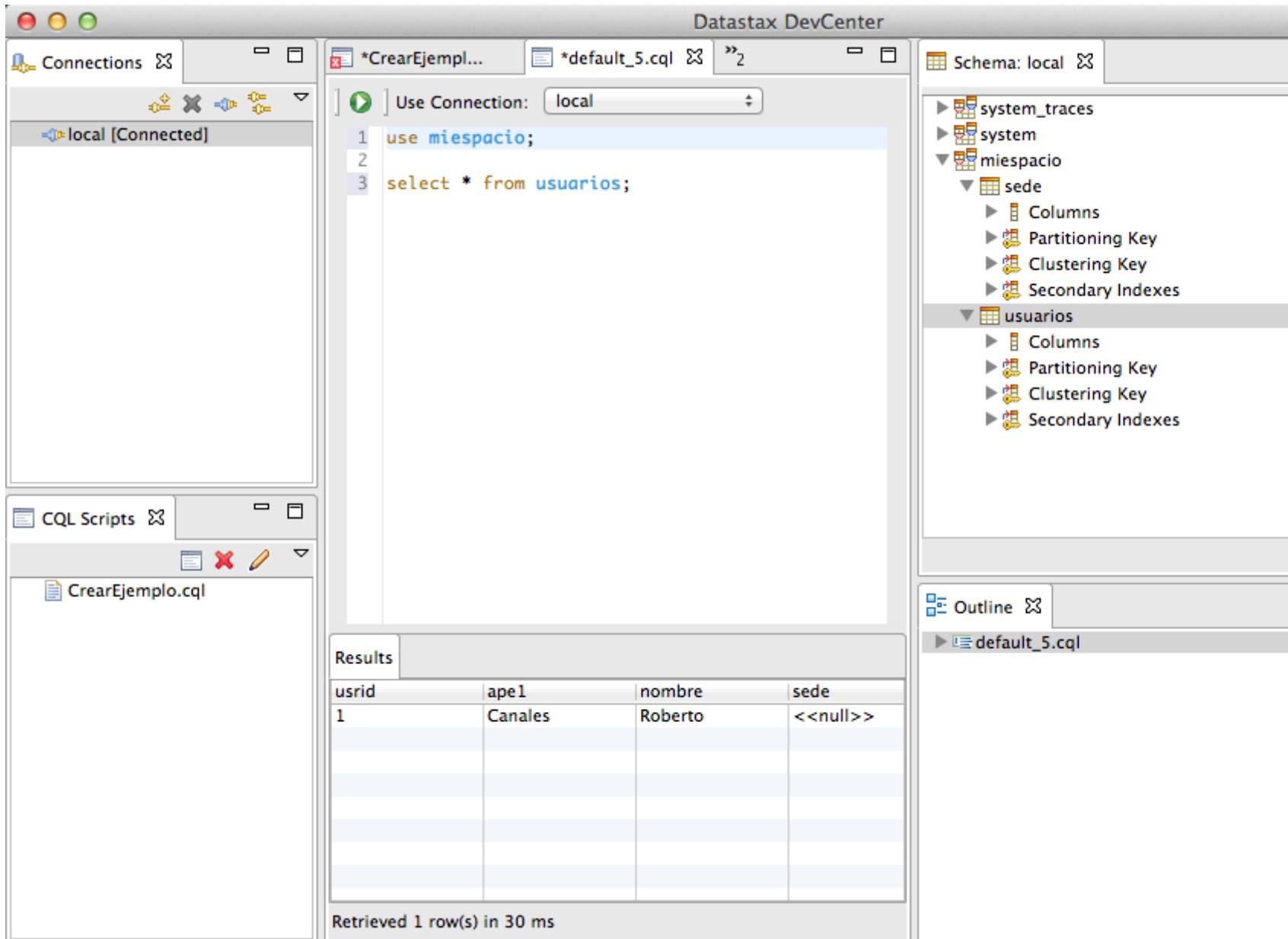
The screenshot shows the Datastax DevCenter interface with multiple windows:

- Connections:** Shows a single connection named "local [Connected]".
- CQL Editor:** Contains the CQL script:

```
1 USE miespacio;
2
3 alter TABLE miespacio.usuarios
4   add sede text;
```
- Schema Browser:** Shows the database schema for "Schema: local". It includes the "miespacio" keyspace, which contains the "sede" table and the "usuarios" table. The "sede" table has columns "sede" (text), "nombre" (text), and "telefono" (text). The "usuarios" table has columns "usrid" (int), "ape1" (text), "nombre" (text), and "sede" (text).
- Outline:** Shows the current file "default_4.cql" with the following content:

```
1 USE miespacio
2
3 ALTER TABLE miespacio.usuarios
```

Comprobamos que se ha creado



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

Tenemos que especificar la clave.

```

use miespacio;

update usuarios set sede = 'Madrid' where usrid = 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), hacia 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/intro_c.html#concept_ds_tgl_q2y_zj

CQL for Cassandra 2.0 | CQL for Cassandra 2.0 Version CQL 3.1

www.datastax.com/documentation/cql/3.1/webhelp/index.html#cql/ddl/ddl_intro_c.html#concept_d

Más visitados Adictos al Traba... Google Apps

CQL for Cassandra 2.0

DataStax Enterprise Docs | Support | Blog | Resources | Contact us | Mobile version | Print page

Planet Cassandra Community

DOWNLOAD

Content Search

Introduction

CQL data modeling

- ▼ Data modeling example
 - Example of a music service
 - Compound keys and clustering
 - Collection columns
 - Adding a collection to a table
 - Updating a collection
 - Querying a collection
- ▶ Indexing
 - Building and maintaining indexes
- ▶ Working with legacy applications
- Using CQL
- ▶ Starting cqlsh
- Using tab completion
- ▶ Creating and updating a keyspace
- ▶ Creating a table
- ▶ Querying a system table
- Retrieving and sorting results
- Using the keyspace qualifier
- Epiring columns
- Determining time-to-live for a column
- Determining the date/time of a write
- Adding columns to a table
- Altering the data type of a column
- ▶ Removing data
- ▶ Using collections
- Indexing a column
- Using lightweight transactions
- Paging through unordered partitioner results
- Using a counter
- ▶ Tracing consistency changes tutorial
- ▶ CQL reference
- Glossary

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 lotería',
  '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.

usrid	ope1	gastos	nombre	sede
1	Canales	{'2013-11-28 00:00:00+0100': 'Compra de lotería', '2013-11-29 00:00:00+0100': 'Cervezas con amigos'}	Roberto	Madrid

(1 rows)

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.

b