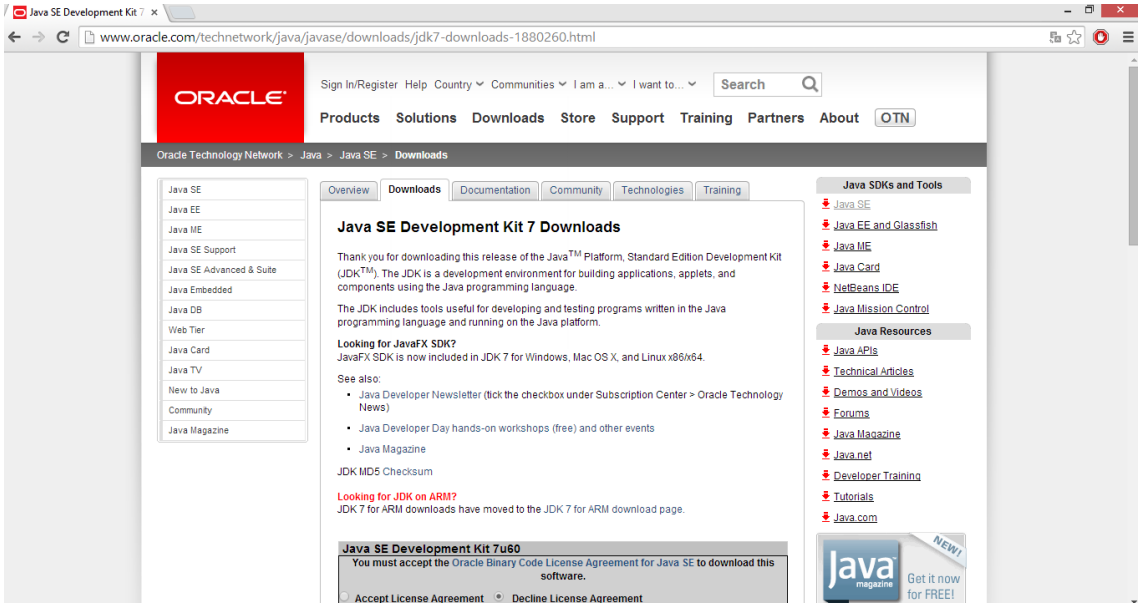
Contenido

Instalación del Java Development Kit (JDK)

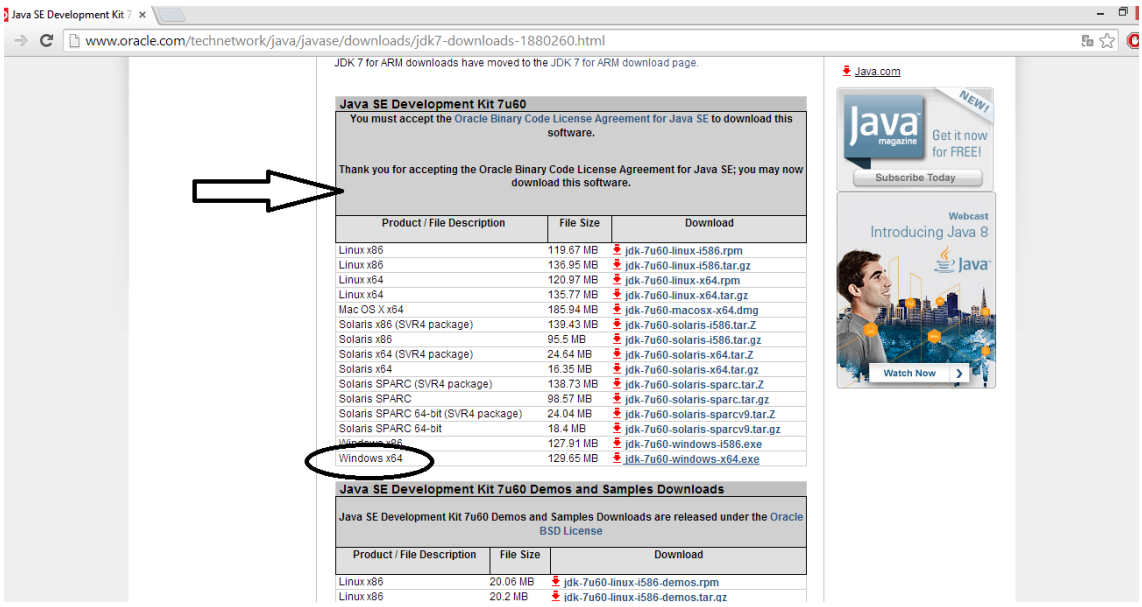
De gran importancia es que tengamos instalado el JDK, este debiera ser el primer paso a realizar. Puede sernos de gran ayuda el siguiente link http://www.luiskano.net/blog/2010/04/28/tutorial-instalar-java-jdk-en-windows-7/

Que nos muestra la instalación para Windows 7, además que tiene otros enlaces para la instalación en otros sistemas de Windows.
La instalación que se mostrará aquí es de la versión jdk 7, la más actual al día 7 de julio del 2014.

1.- Nos dirigimos a la página de Oracle a la sección de descargas del jdk, eventualmente encontraremos la versión más actual a la fecha.

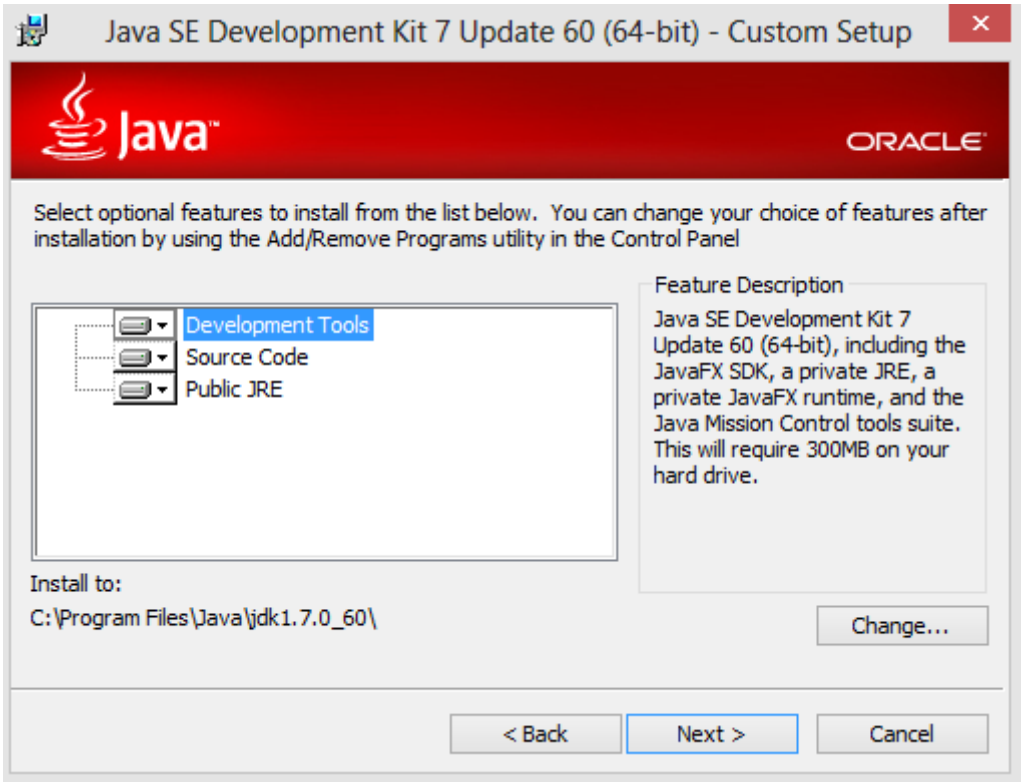


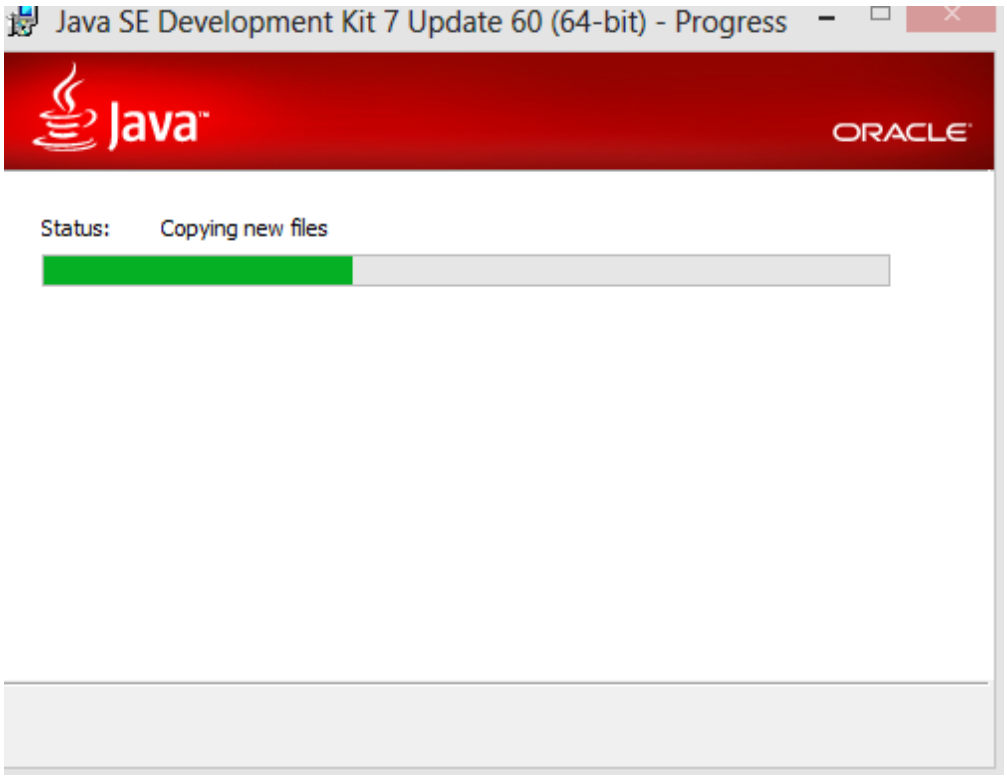
2.-Aceptamos las condiciones de la licencia para el JDK y escogemos la versión que corresponde a nuestro sistema operativo.



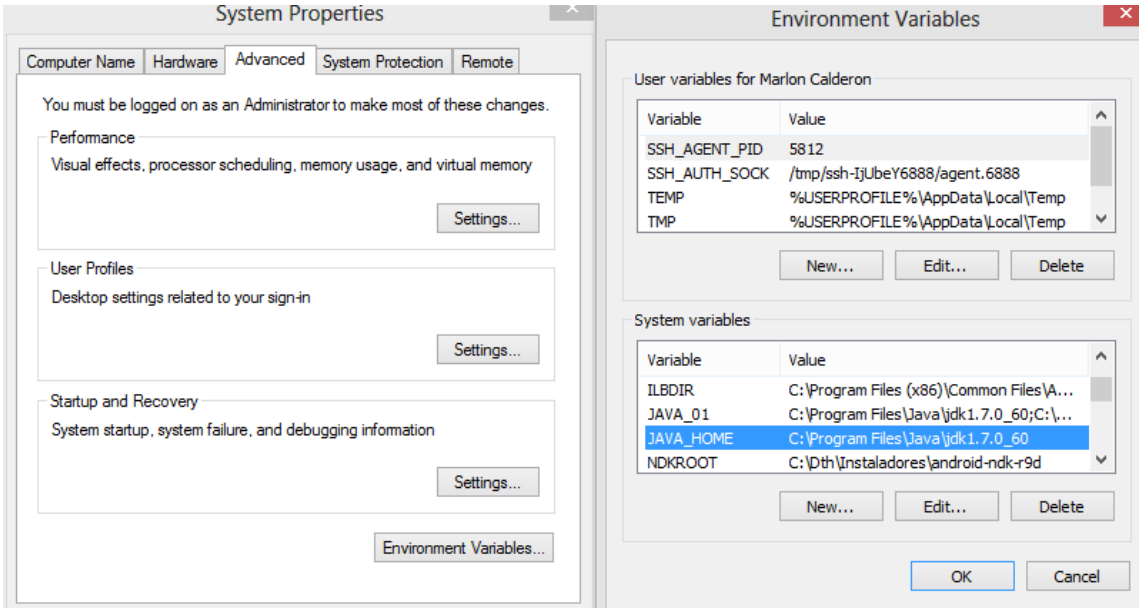
3.- La descarga inicia inmediatamente después, solo queda esperar a que termine.

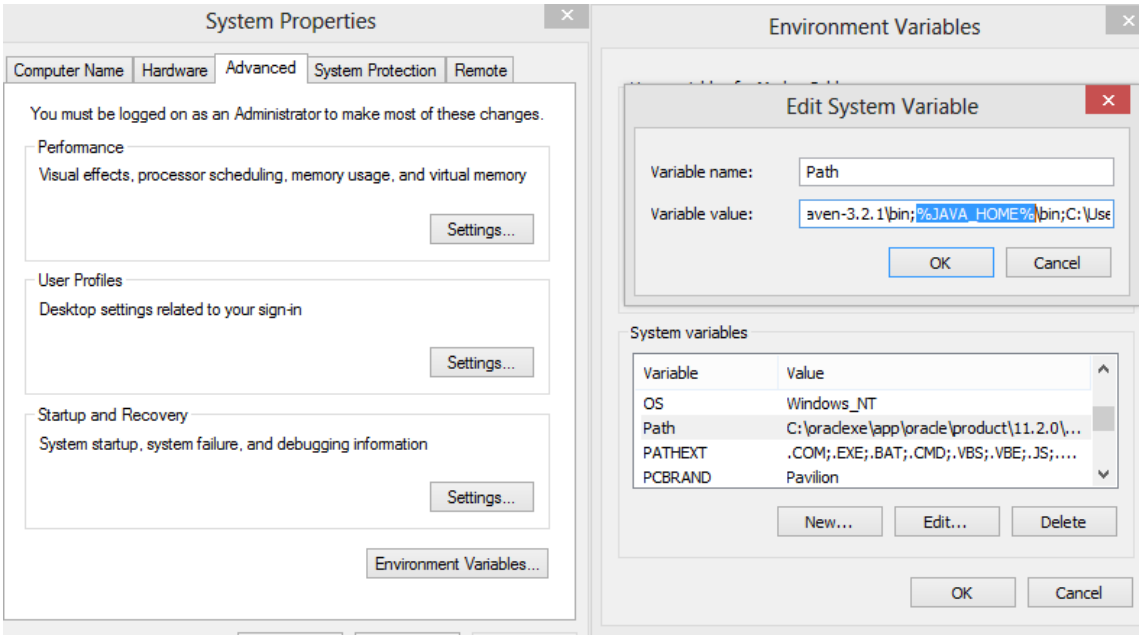
4.- Comenzamos la instalación. Básicamente se instala todo por defecto, es decir, escogiendo siguiente en cada pantalla, sin mayor tipo de configuración.


Java SE Development Kit 7 Update 60 (64-bit) - Custom Setup
Java
ORACLE
Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel
Development Tools
Source Code
Public JRE
Feature Description
Java SE Development Kit 7 Update 60 (64-bit), including the JavaFX SDK, a private JRE, a private JavaFX runtime, and the Java Mission Control tools suite. This will require 300MB on your hard drive.
Install to:
C:\Program Files\Java\jdk1.7.0_60\
Change...
< Back
Next >
Cancel

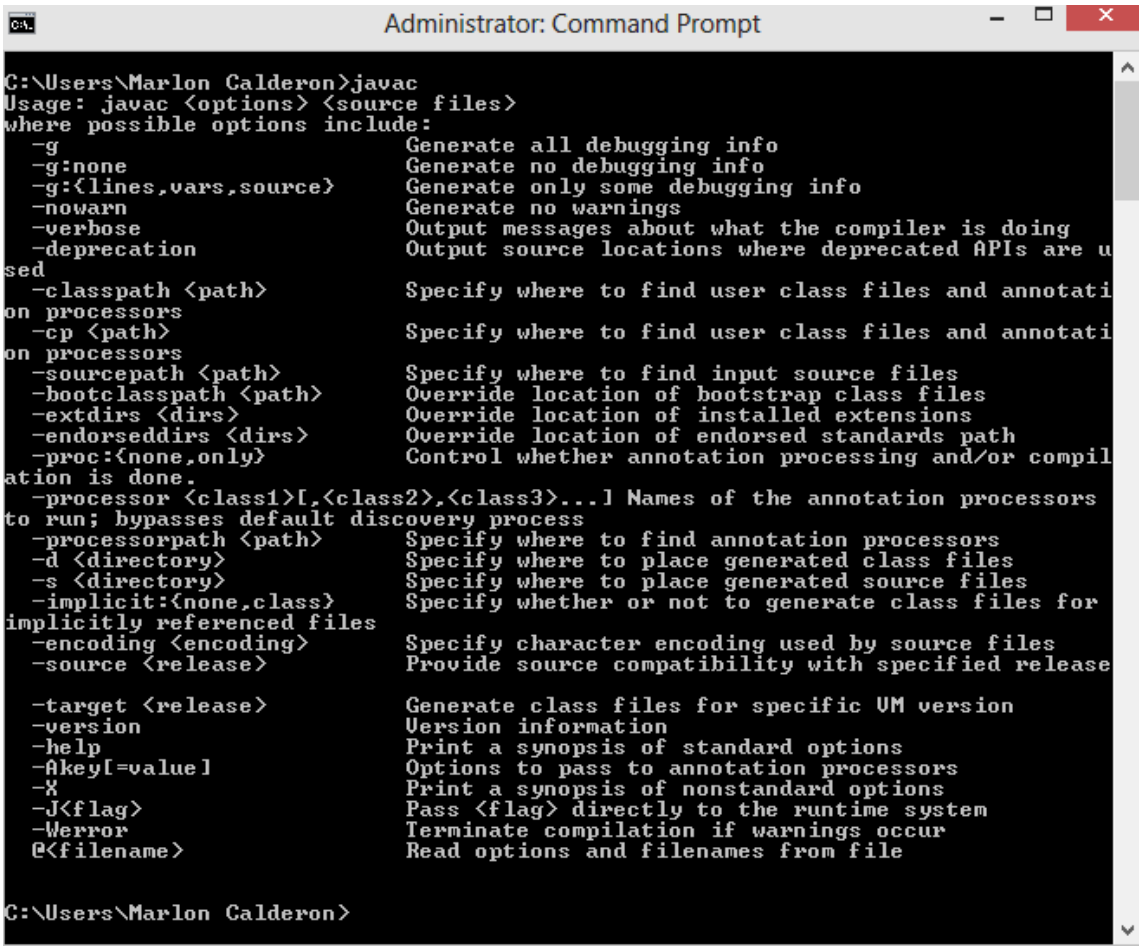

Java SE Development Kit 7 Update 60 (64-bit) - Progress
Java
ORACLE
Status:
Copying new files

5.- Se configura las variables de entorno





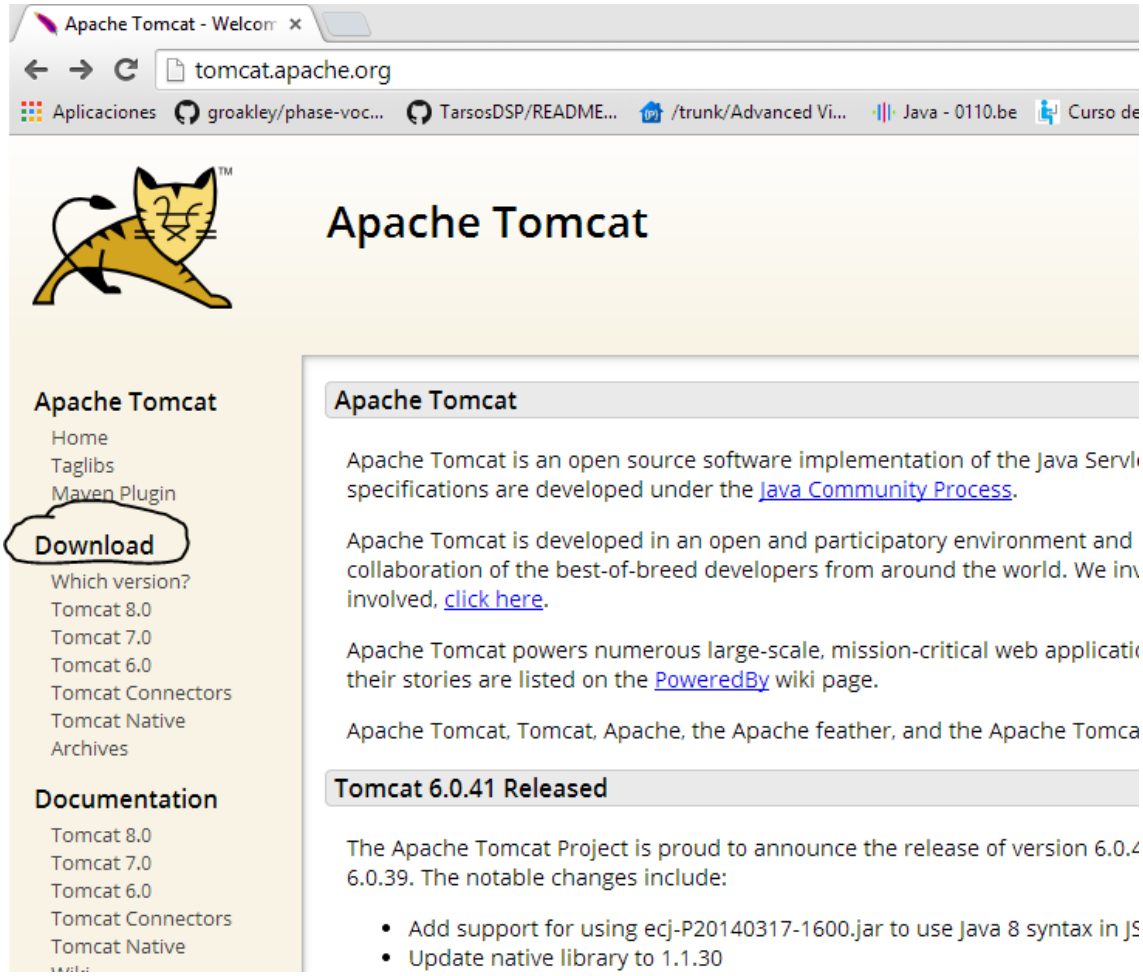
6.- Finalmente se verifica que está bien instalado



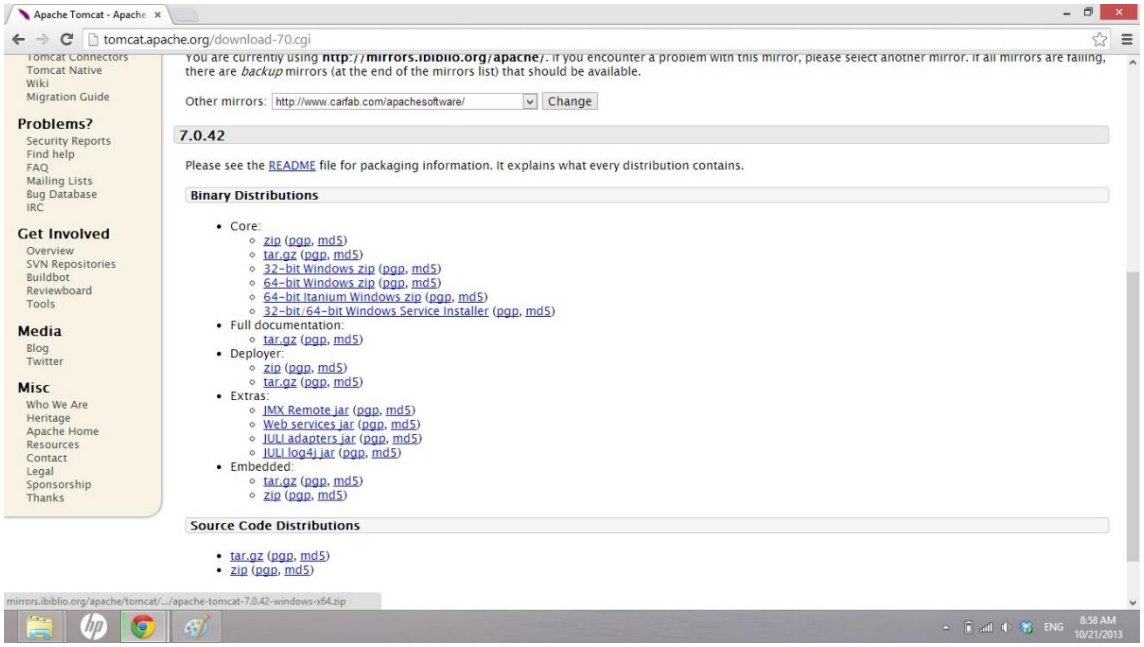
Instalación de Apache Tomcat

Instalación de Apache Tomcat 7.0.42

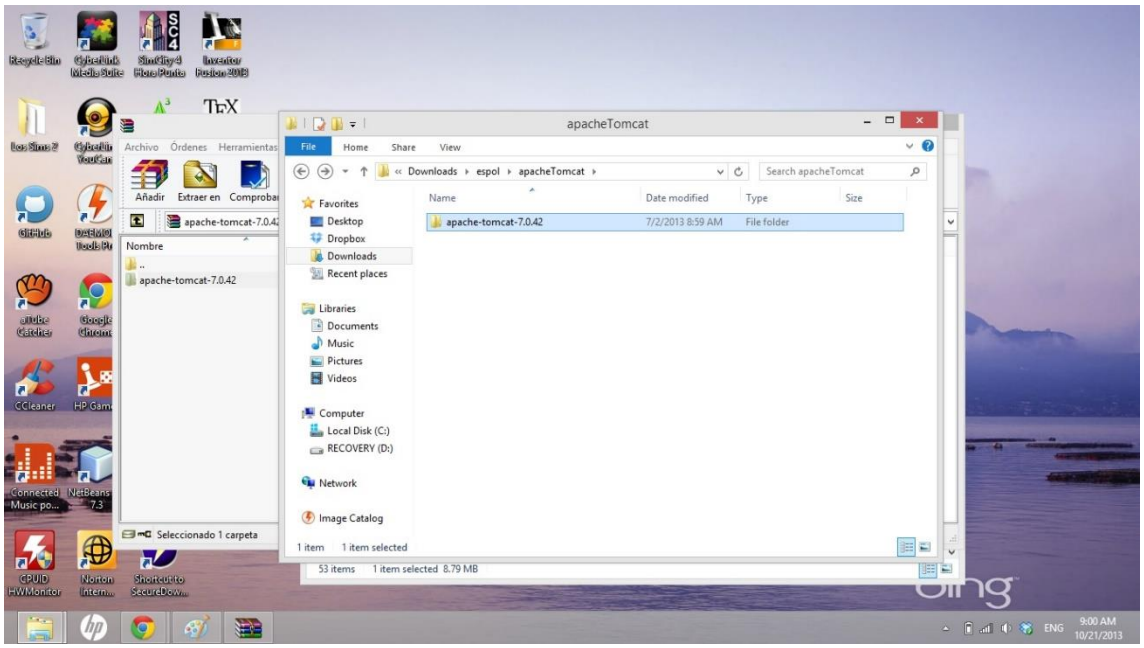
1.- Vamos directo a la página de apache a la sección de descargas



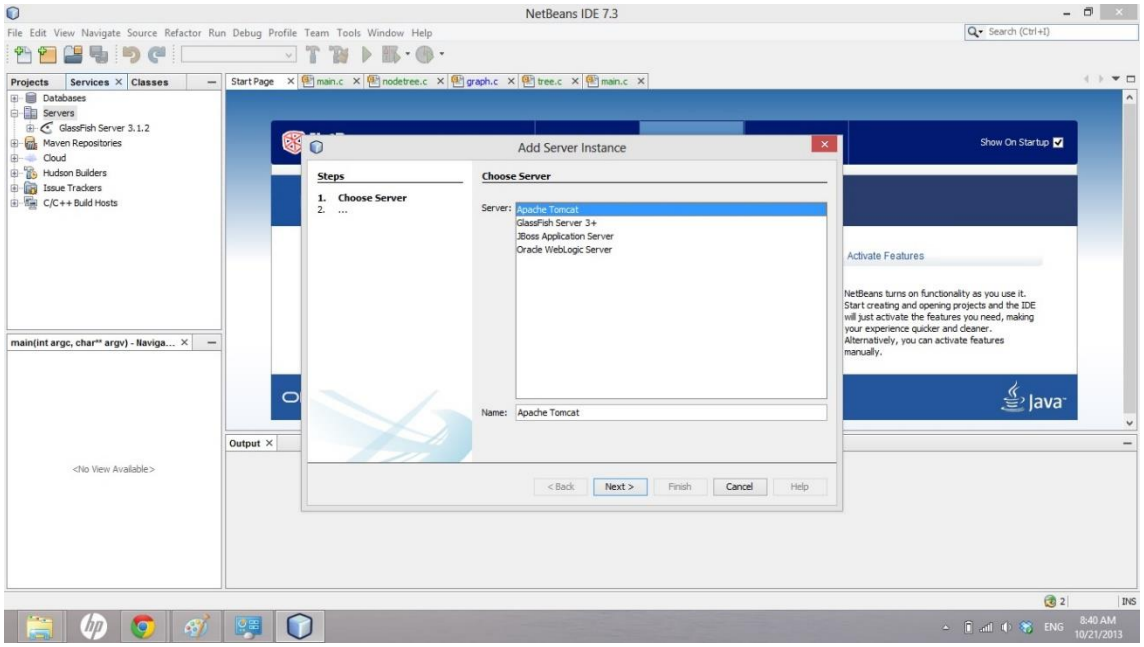
2.- Buscamos una versión para descargar



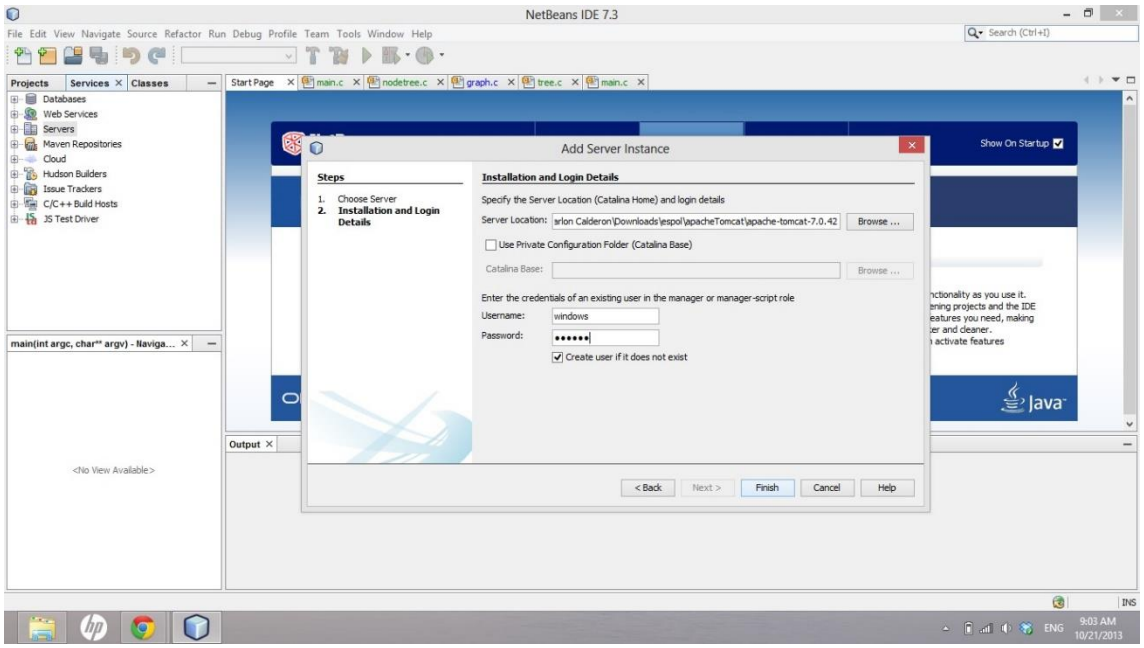
3.- Descargamos tanto el zip como el instalador



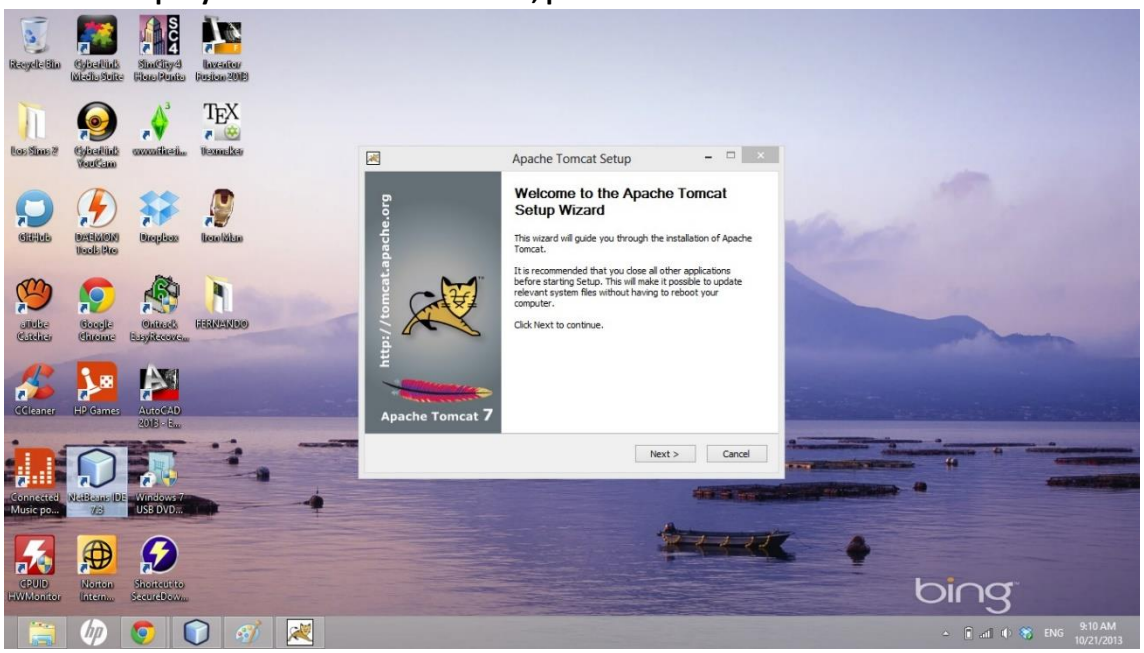
4.- Abrimos NetBeans y nos vamos a la sección services, en servers, escogemos agregar server (add server) escogemos Apache Tomcat



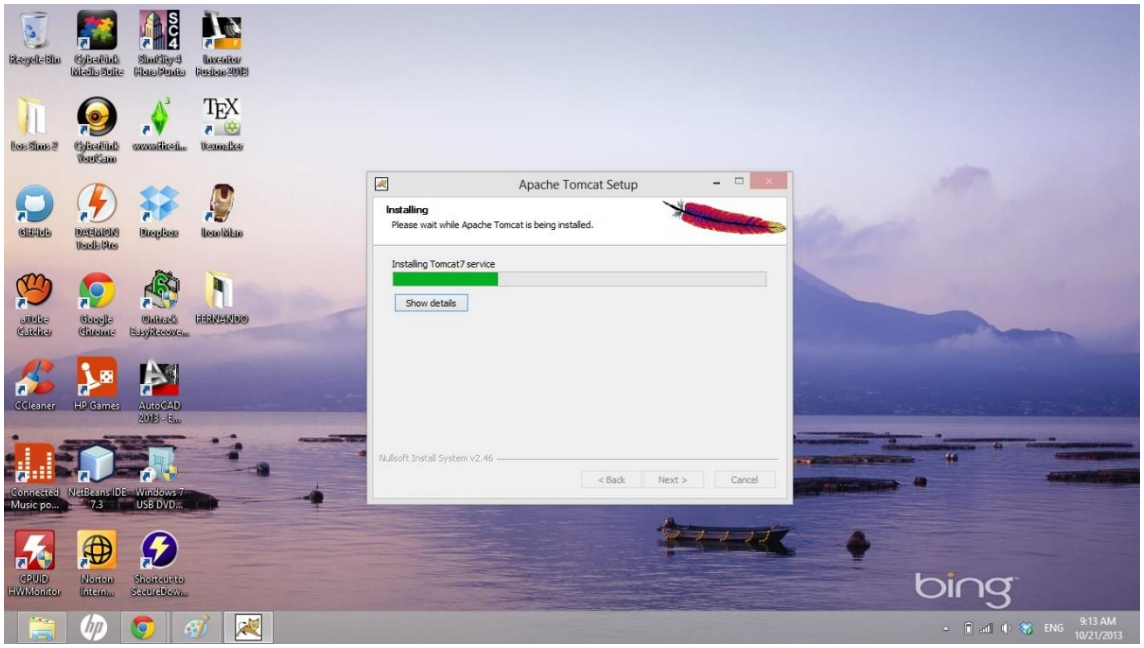
5.-Buscamos el directorio del apache tomcat y lo ingresamos, le damos un usuario y una clave.

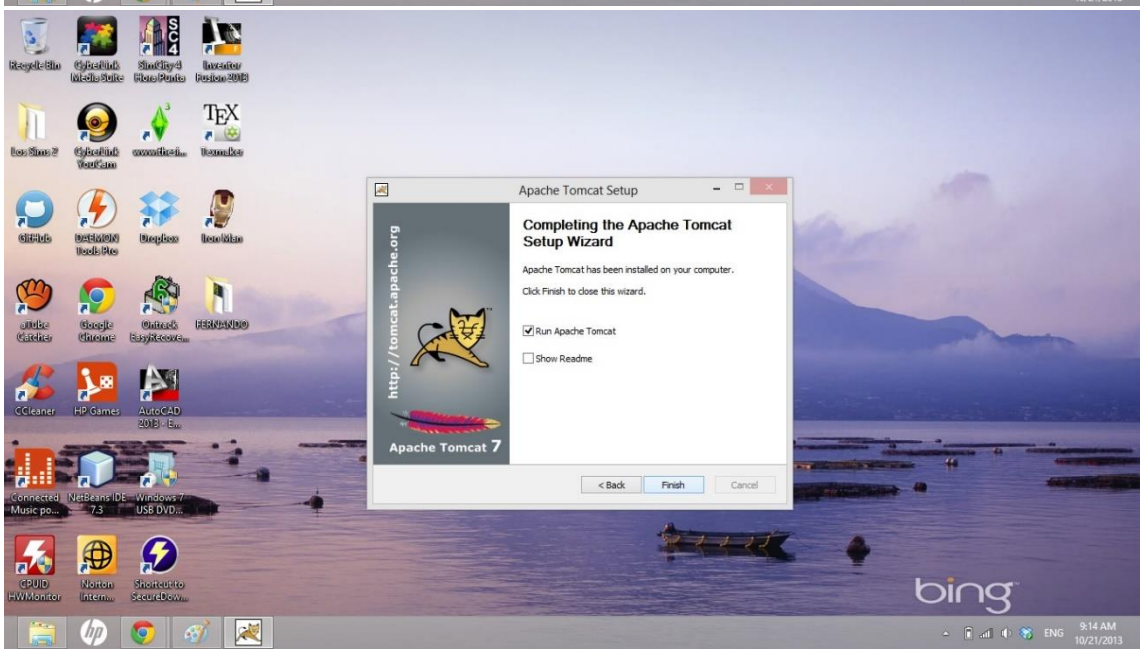


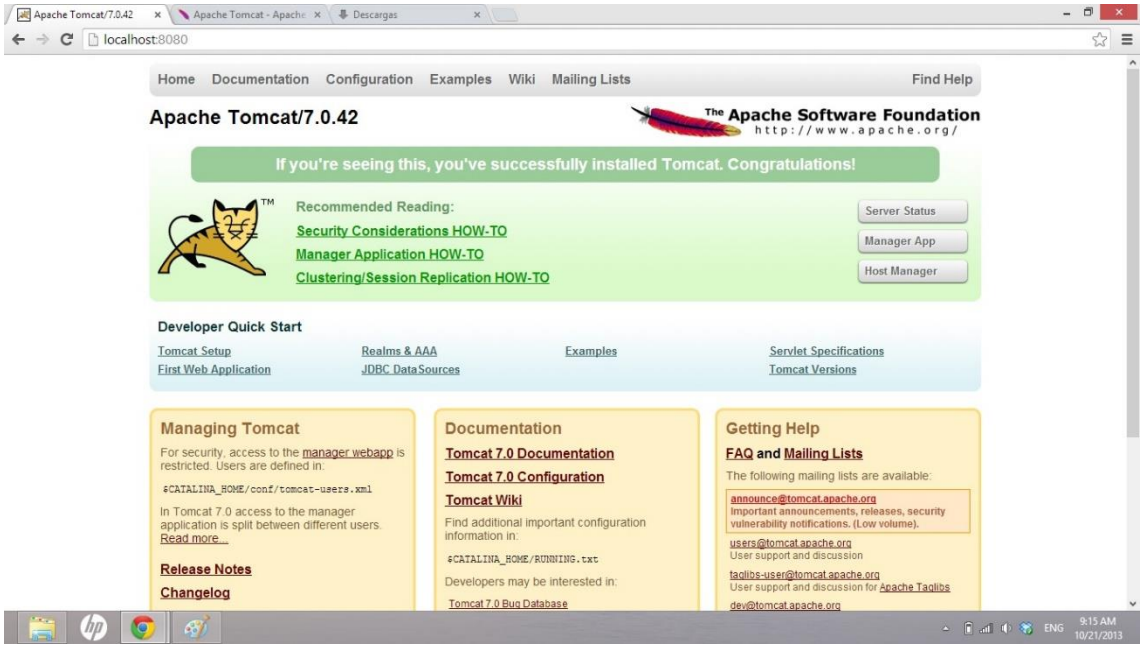
6.- Una vez que ya lo tenemos en NetBeans, procedemos a instalarlo utilizando el instalador.



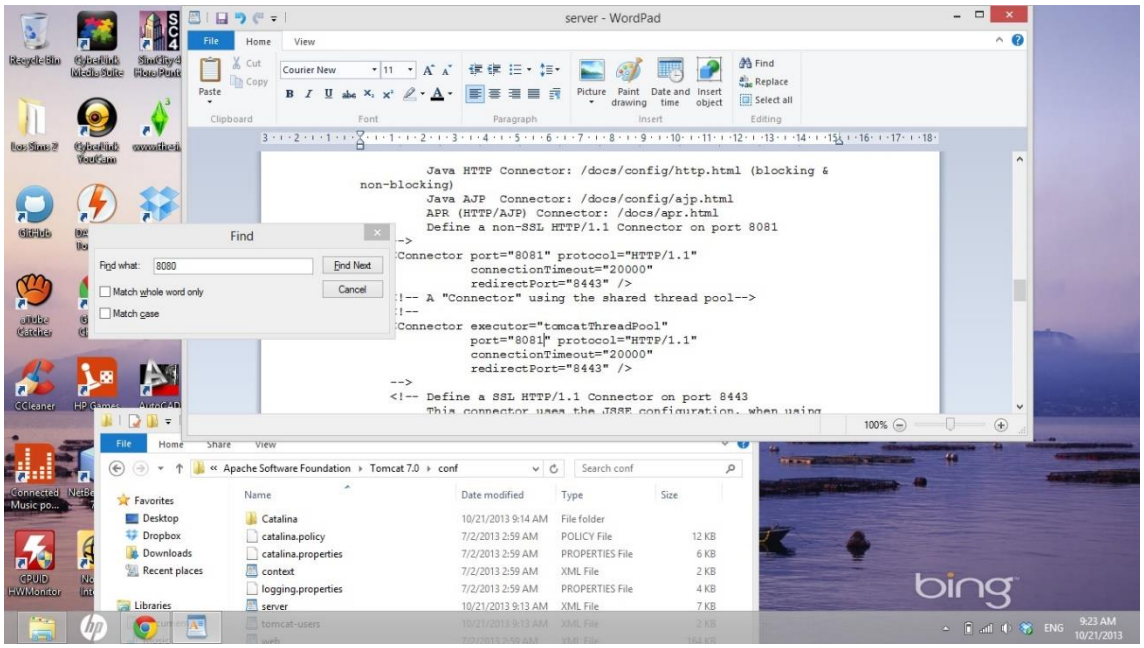
7.- Si Seguimos la instalación por defecto tendremos lo siguiente


Apache Tomcat Setup
Installing
Please wait while Apache Tomcat is being installed.
Installing Tomcat7 service
Show details
Nullsoft Install System v2.46
< Back
Next >
Cancel
bing
ENG
9:13 AM
10/21/2013

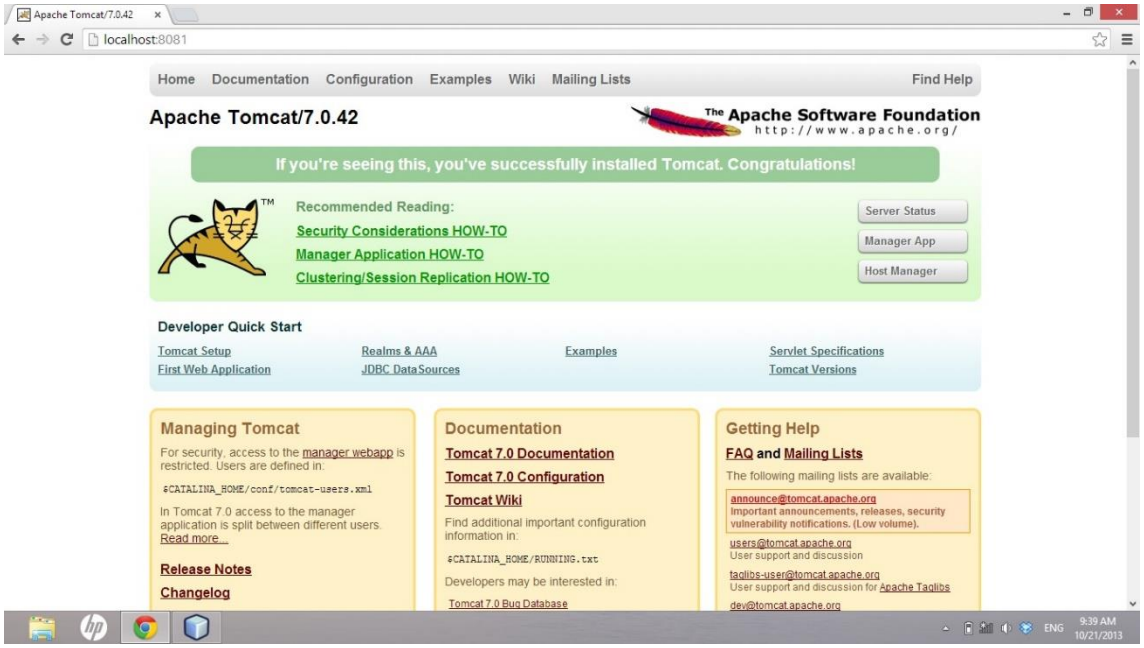

Apache Tomcat Setup
Completing the Apache Tomcat Setup Wizard
Apache Tomcat has been installed on your computer.
Click Finish to close this wizard.
Run Apache Tomcat
Show Readme
http://tomcat.apache.org
Apache Tomcat 7
< Back
Finish
Cancel
bing
ENG
9:14 AM
10/21/2013



8.- Procedemos a cambiar el puerto de nuestro apache tomcat, ya que estamos utilizando el 8081. Para esto basta con cambiar una línea en el documento server.xml



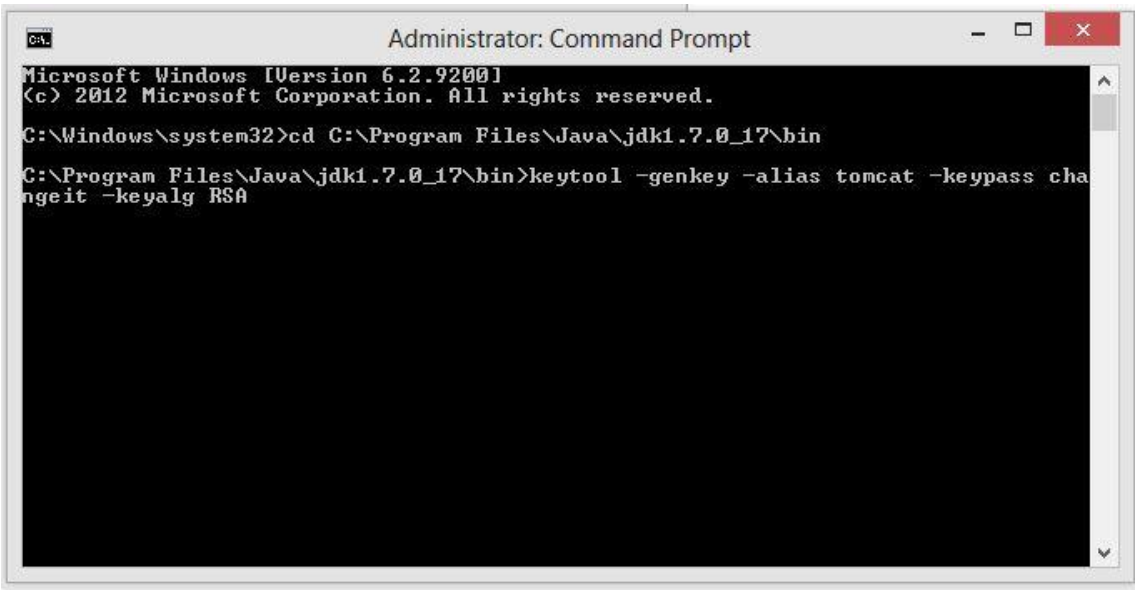
9.- Reiniciamos apache y lo volvemos a ver en el navegador, solo que usando el puerto 8081


Apache Tomcat/7.0.42
localhost:8081
Home Documentation Configuration Examples Wiki Mailing Lists Find Help
Apache Tomcat/7.0.42
The Apache Software Foundation
http://www.apache.org/
If you're seeing this, you've successfully installed Tomcat. Congratulations!
Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO
Server Status
Manager App
Host Manager
Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions
Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
$CATALINA_HOME/conf/tomcat-users.xml
In Tomcat 7.0 access to the manager application is split between different users.
Read more...
Release Notes
Changelog
Documentation
Tomcat 7.0 Documentation
Tomcat 7.0 Configuration
Tomcat Wiki
Find additional important configuration information in:
$CATALINA_HOME/RUNNING.txt
Developers may be interested in:
Tomcat 7.0 Bug Database
Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
announce@tomcat.apache.org
Important announcements, releases, security vulnerability notifications. (Low volume).
users@tomcat.apache.org
User support and discussion
taglibs-user@tomcat.apache.org
User support and discussion for Apache Taglibs
dev@tomcat.apache.org
ENG
9:39 AM
10/21/2013

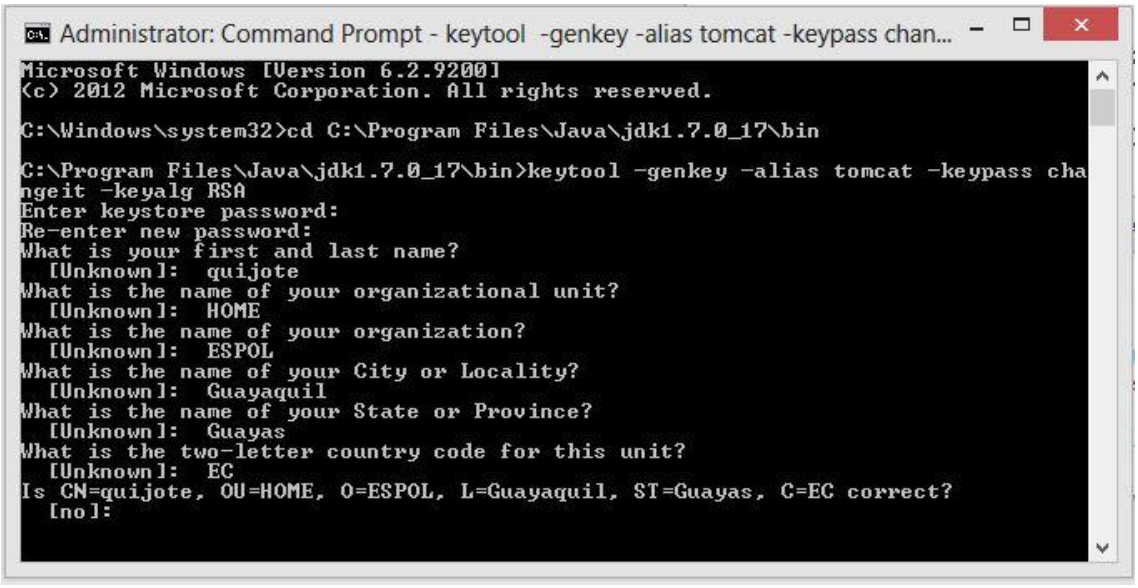
Creación del certificado

Creando certificado

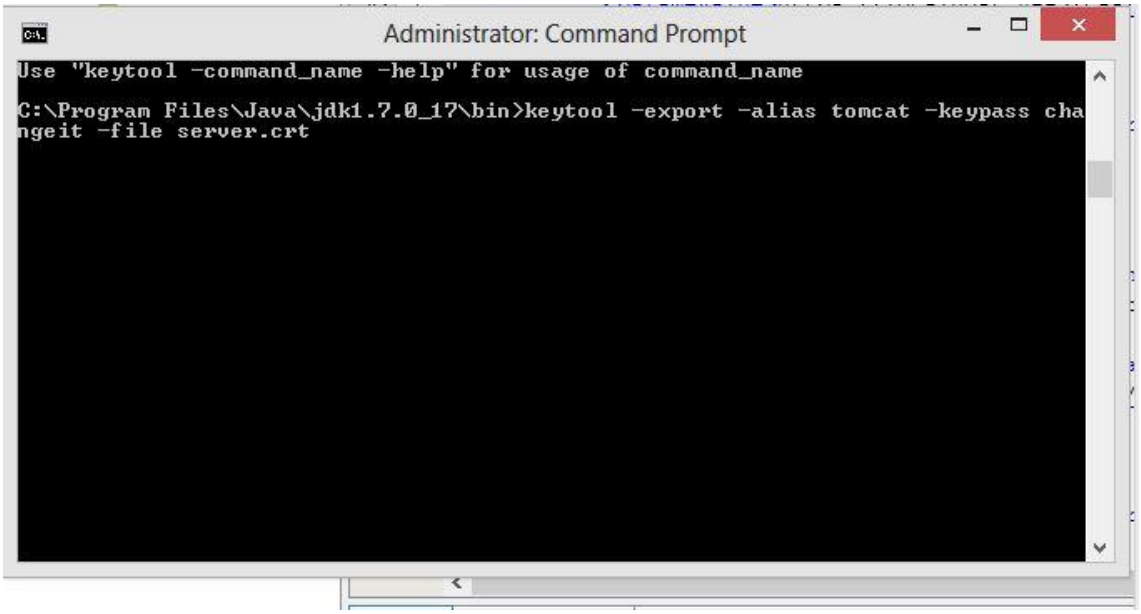
1. Trabajamos directamente con el CMD, es necesario abrirlo en modo de administrador. Ingresamos el comando para crear, el cual se ve en la captura de la imagen, es de gran importancia aclarar que se trabaja en la ruta donde se encuentra el jdk previamente instalado.



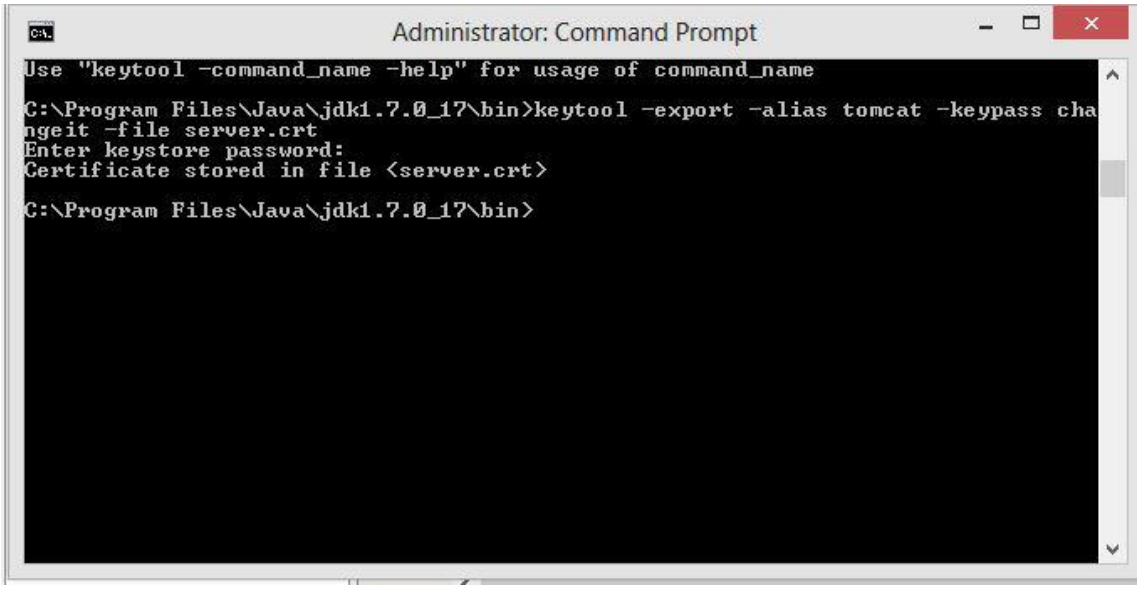
2.- Llenamos los datos que se nos van solicitando



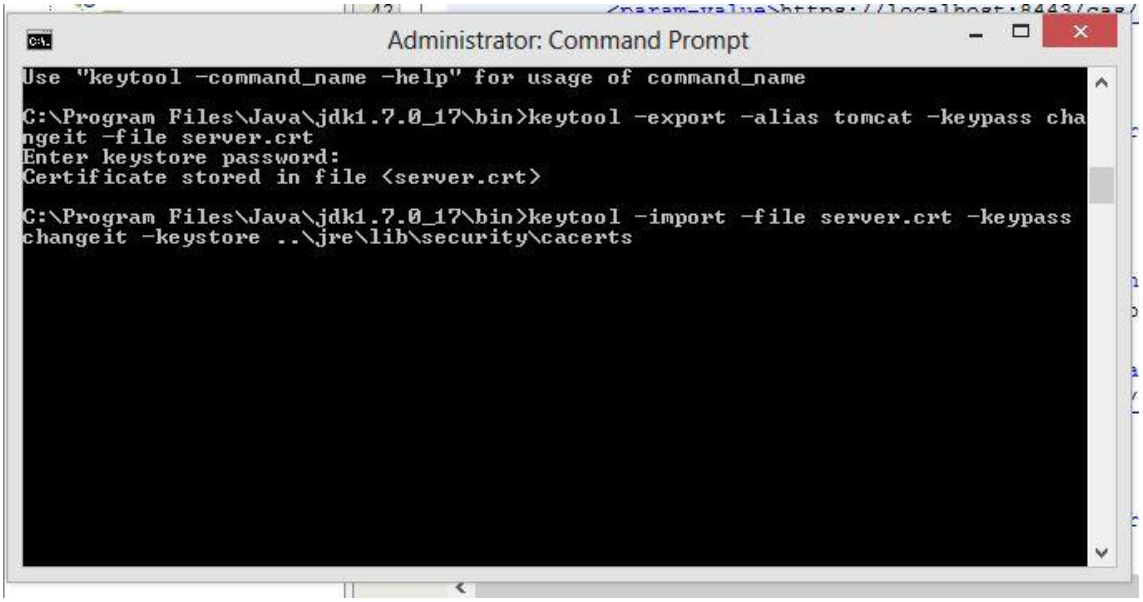
3.- Luego usamos el comando que se ve en la imagen, comando para exportar el certificado



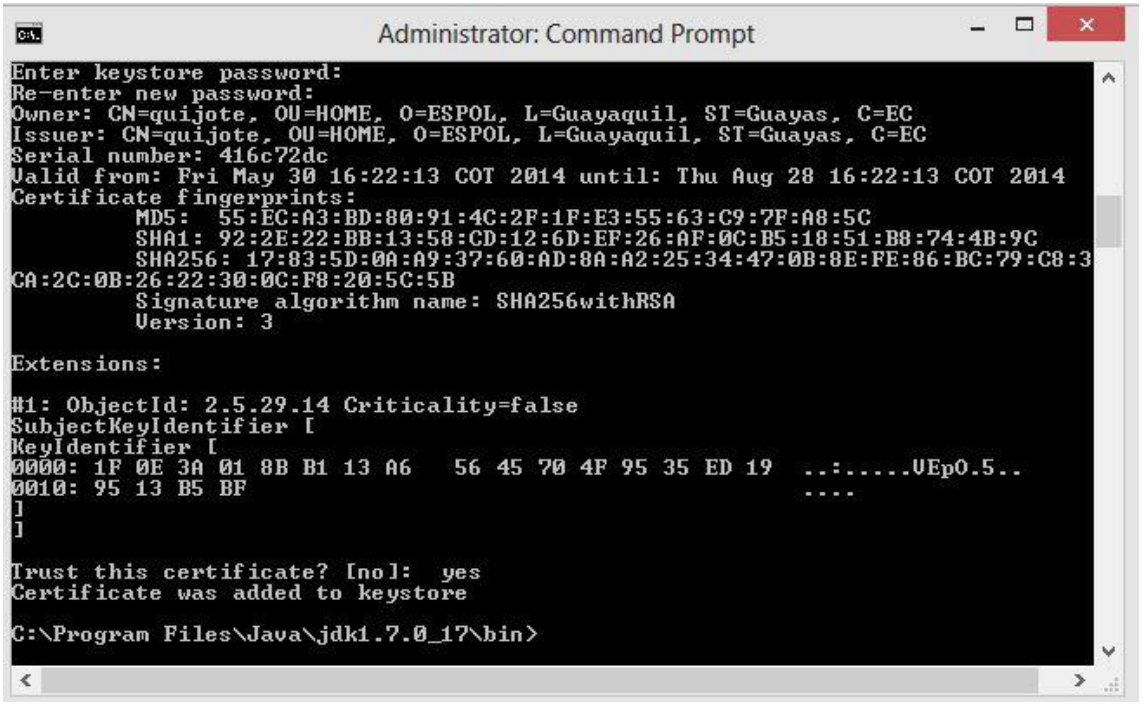
4.- El certificado creado, será guardado



5.- Ingresamos el comando que se ve en la imagen, comando para importar el certificado

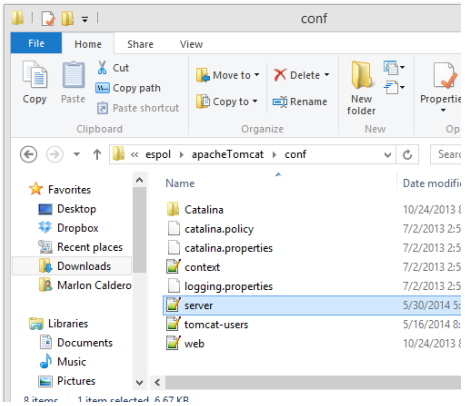


6.- Indicamos que el certificado es de confianza y termina la creación del certificado



Para finalizar la instalación del tomcat que usaremos con el certificado y así usar el puerto 8443 como https, solamente modificamos las líneas que se ven en la imagen, las cuales son las del documento server.xml que es parte de los archivos de configuración del tomcat.

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol" SSLEnabled="true"
           maxThreads="150" scheme="https" secure="true"
           enableLookups="false" maxHttpHeaderSize="8192"
           maxSpareThreads="75" minSpareThreads="25"
           clientAuth="false" sslProtocol="TLS"
           acceptcount="100" disableUploadTimeout="true"
           keystoreFile="C:/Users/Marlon Calderon/.keystore"
           keystorePass="changeit"
           truststoreFile="C:/Program Files/Java/jdk1.7.0_17/jre/lib/security/cacerts"/>
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443"/>


<!-- An Engine represents the entry point (within Catalina) that processes
     every request.  The Engine implementation for Tomcat stand alone
     analyzes the HTTP headers included with the request, and passes them
     on to the appropriate Host (virtual host).
     Documentation at /docs/config/engine.html -->

<!-- You should set jvmRoute to support load-balancing via AJP ie :
<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
-->
<Engine defaultHost="localhost" name="Catalina">

  <!--For clustering, please take a look at documentation at:
```

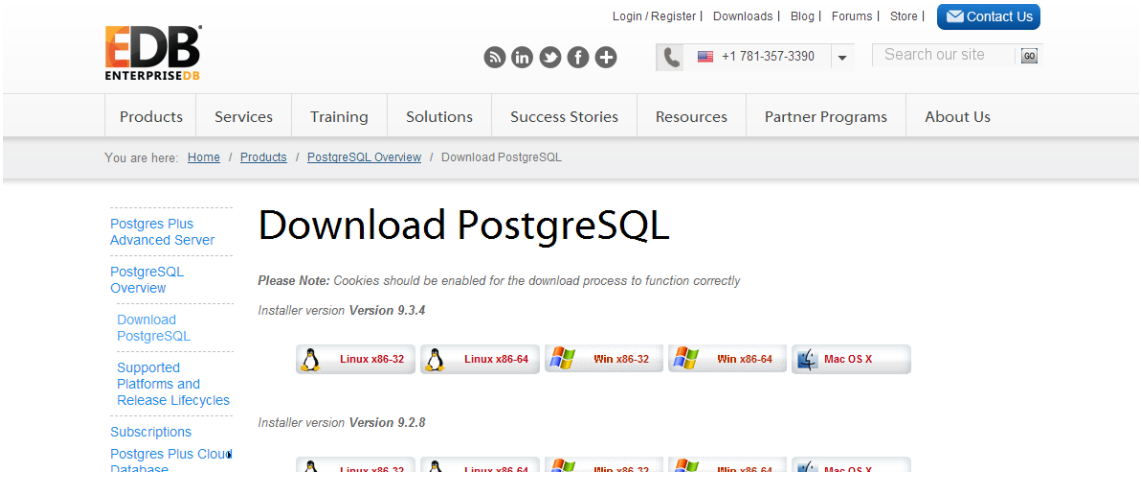


Instalación de PostgreSQL

Instalación PostgreSQL

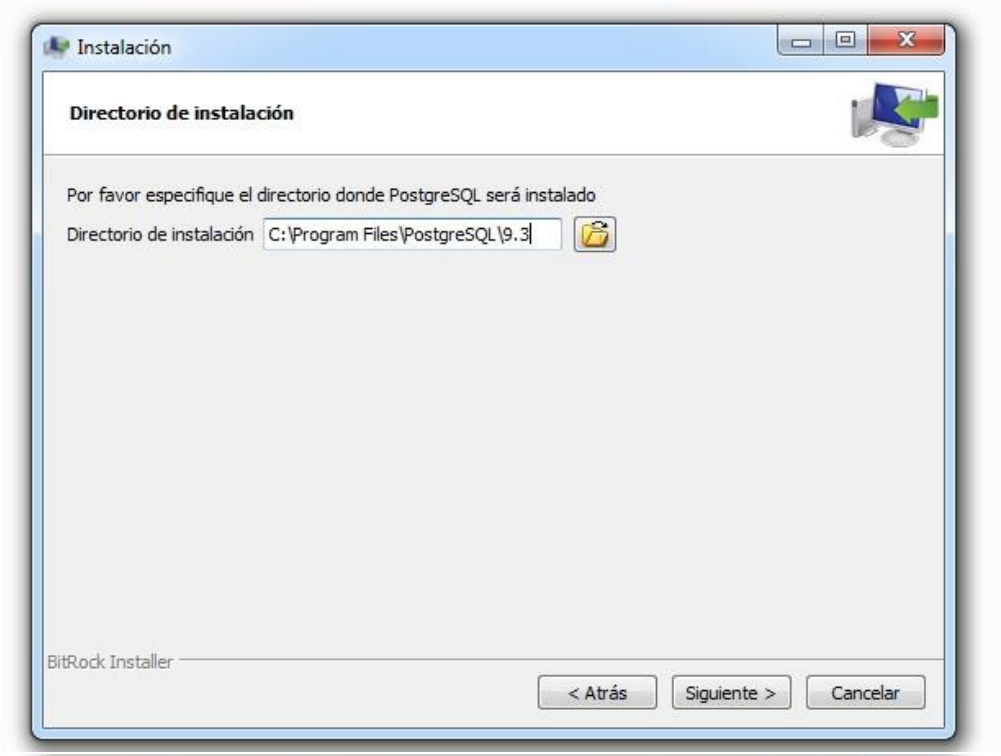
1.- En la siguiente dirección descargamos la versión de postgresql que instalaremos, se nos dan opciones a escoger por el sistema operativo, para este caso la descarga será para Windows.

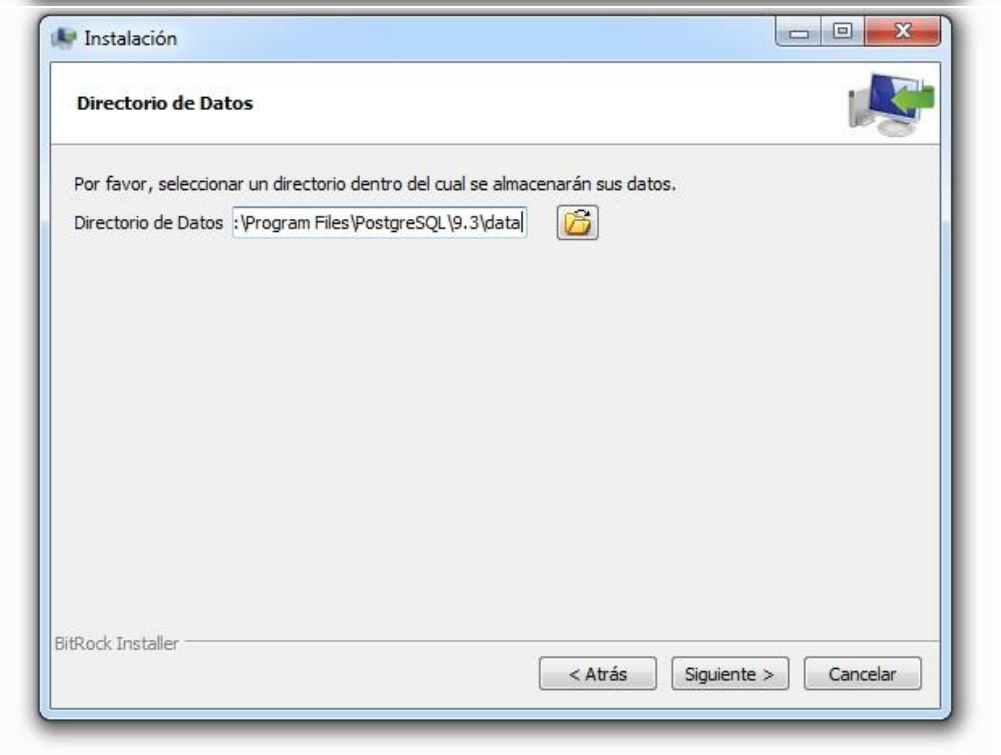
http://www.enterprisedb.com/products-services-training/pgdownload#windows



2.- Una vez descargado empezamos la instalación

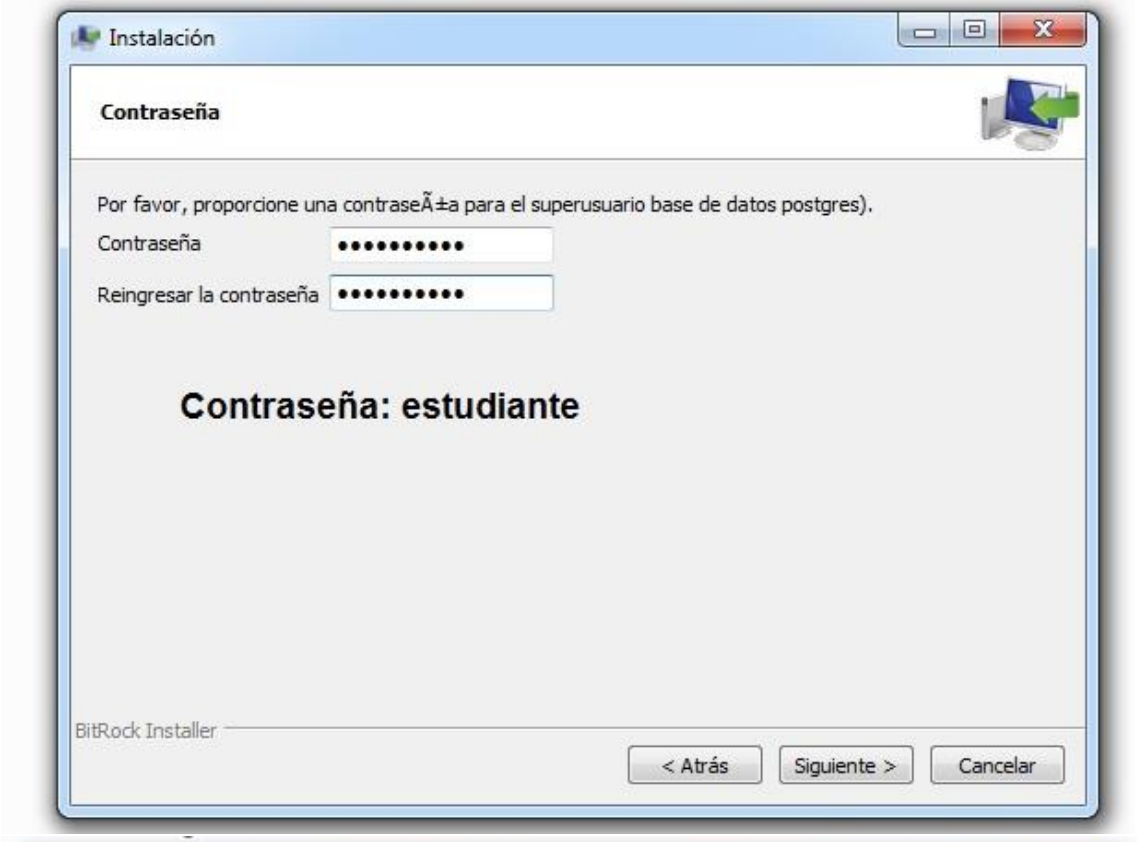
3.- Se selecciona siguiente, lo hicimos con las configuraciones por defecto, hasta llegar al punto donde nos pide la contraseña.

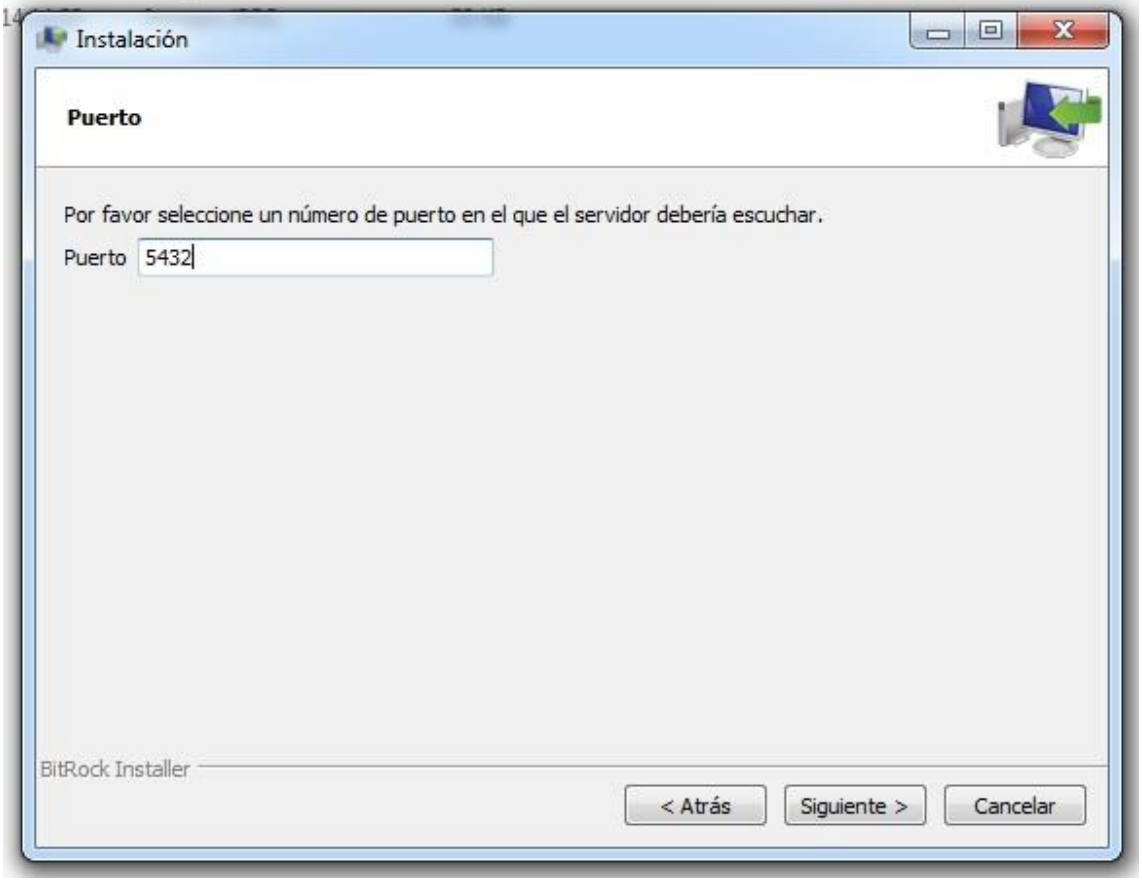


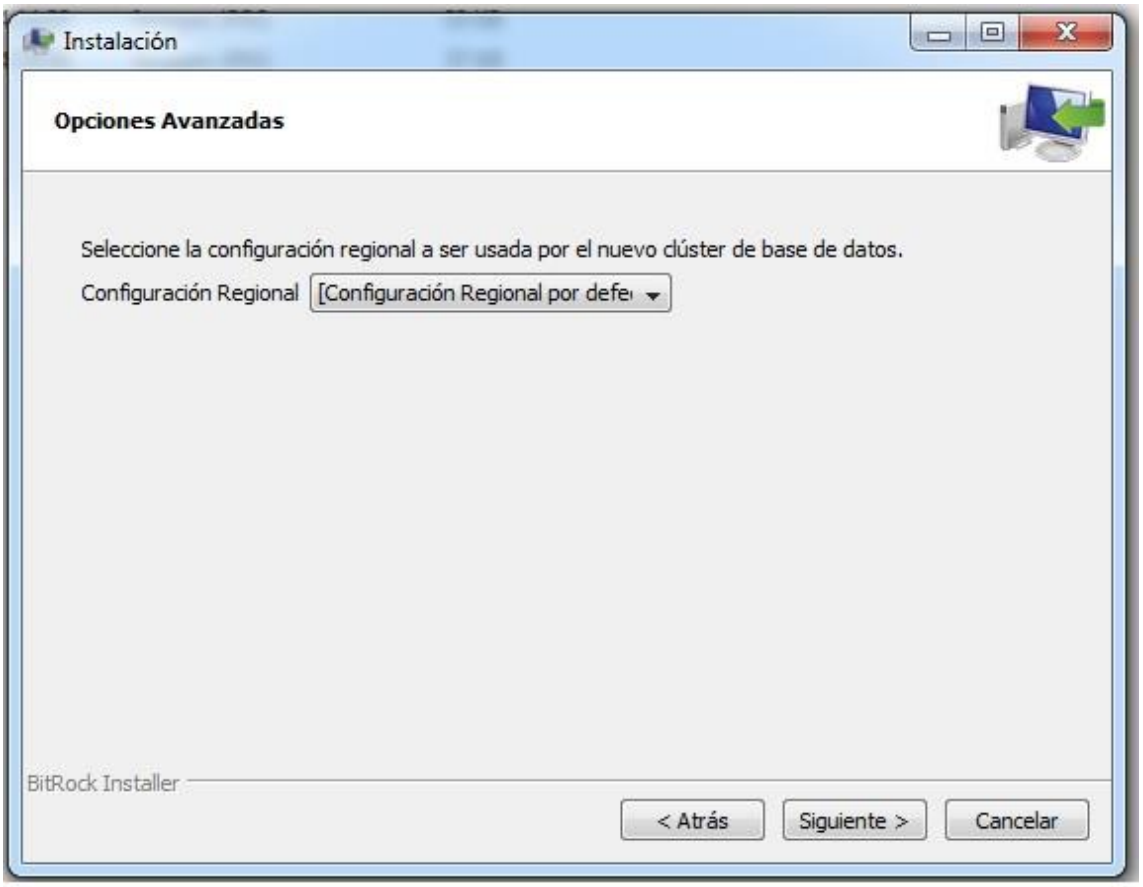


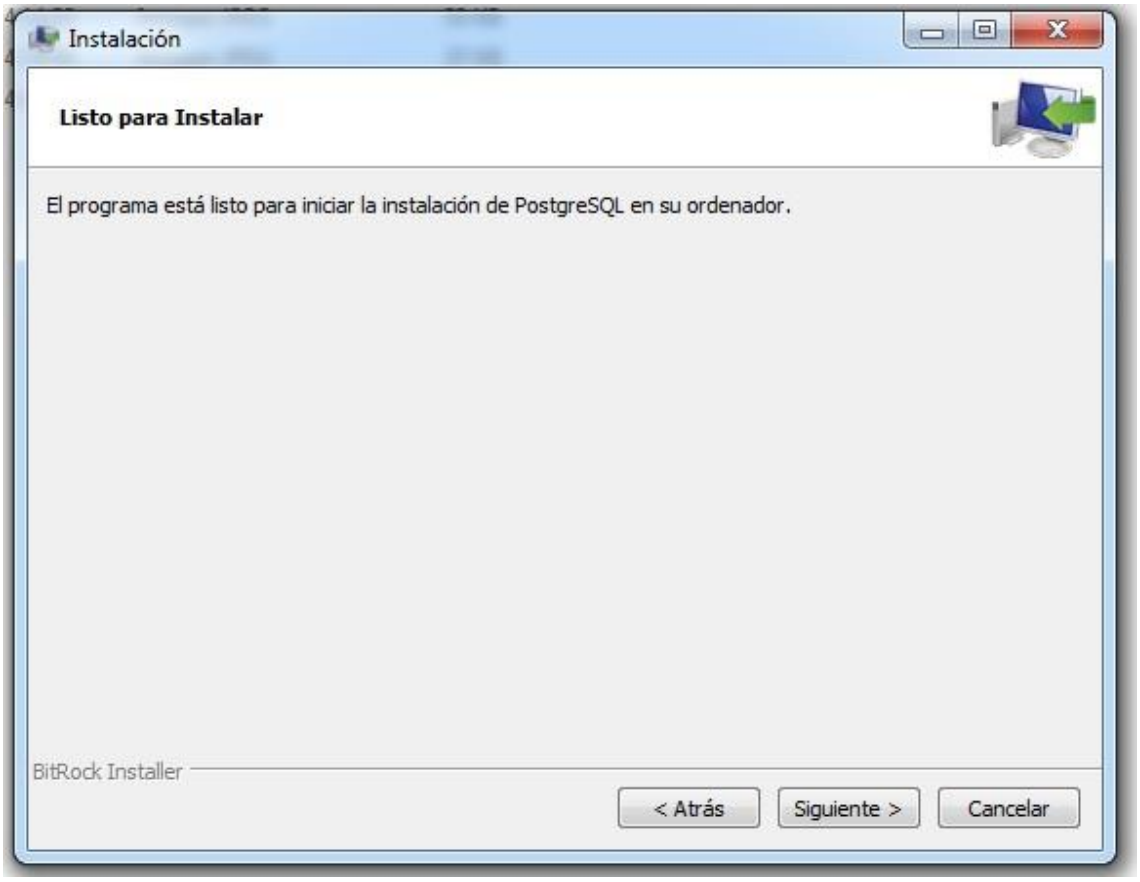
4.-Ponemos una contraseña, nosotros le pusimos de contraseña <u>estudiante</u> y así mismo el resto de configuraciones se dejaron por defecto, hasta que concluyó la instalación

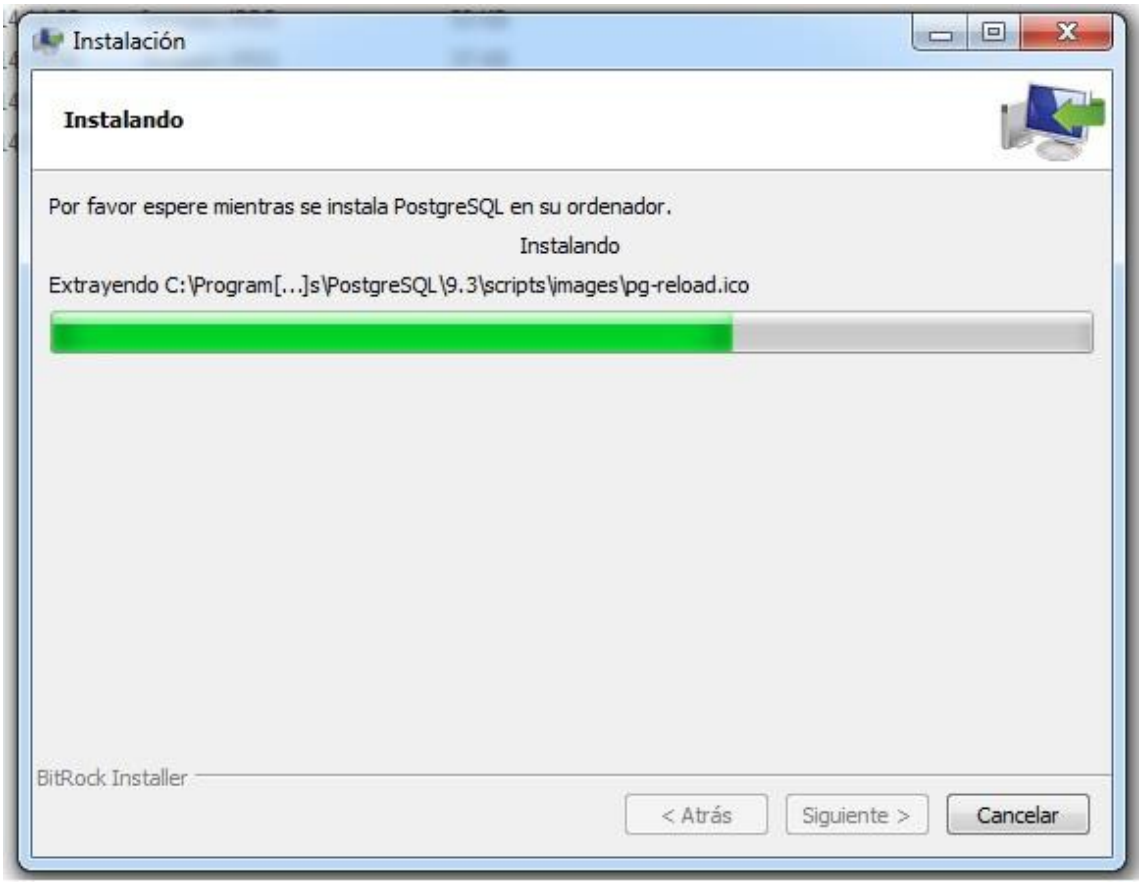
5/2014 14:54 Imagen JPEG 43 KB



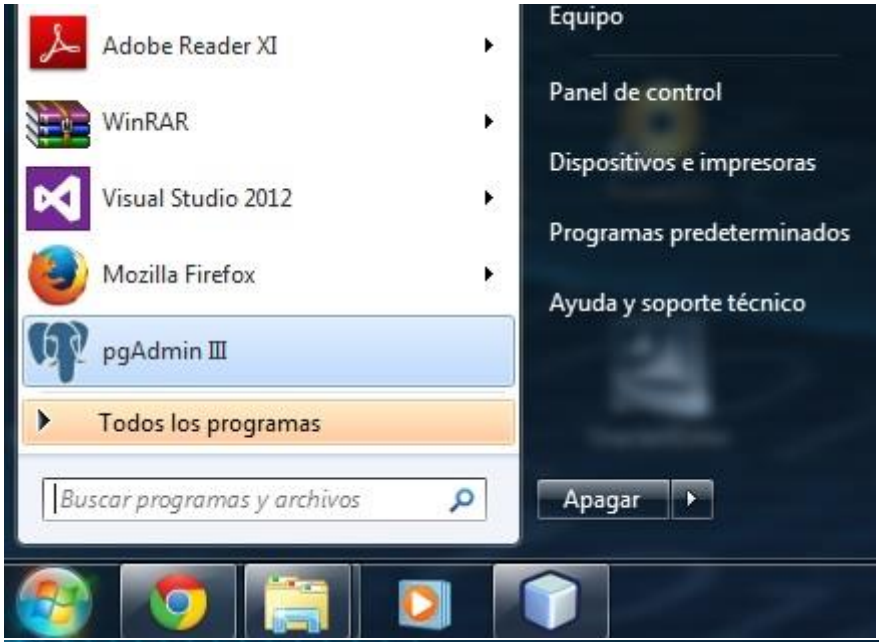


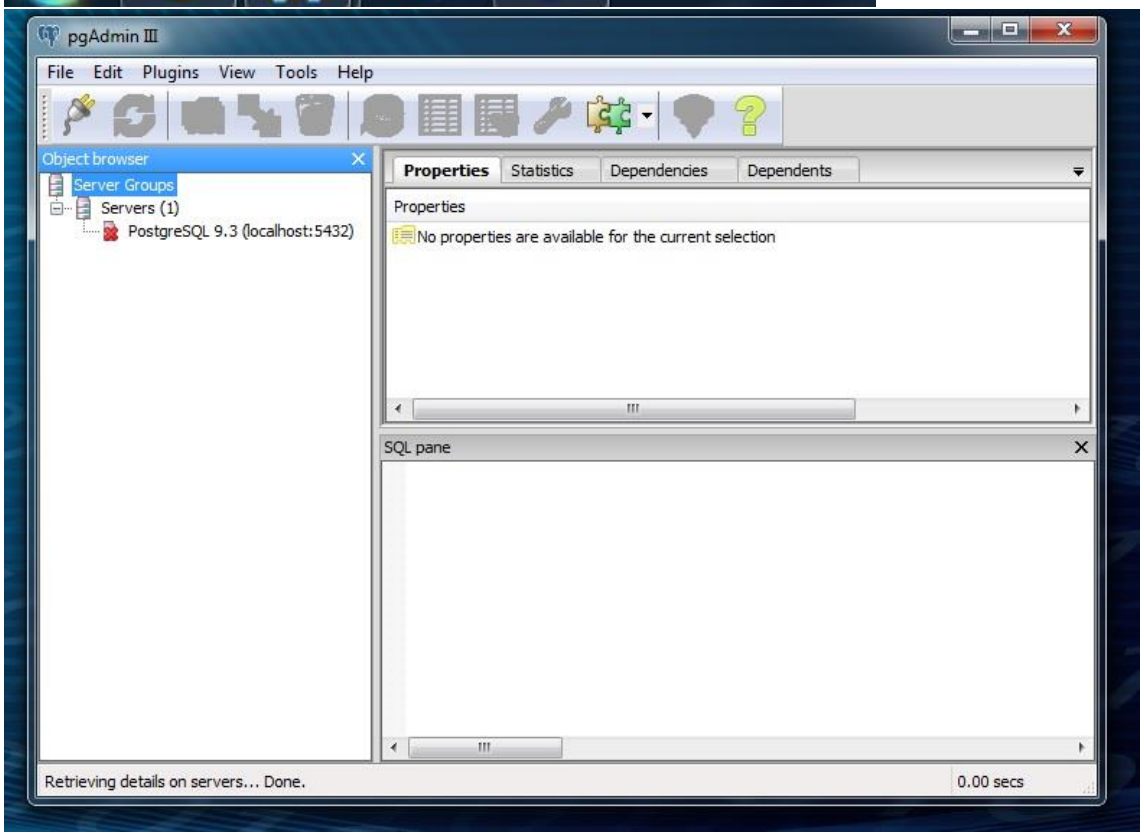

Instalación
Opciones Avanzadas
Seleccione la configuración regional a ser usada por el nuevo clúster de base de datos.
Configuración Regional
[Configuración Regional por defe
BitRock Installer
< Atrás
Siguiente >
Cancelar


Instalación
Listo para Instalar
El programa está listo para iniciar la instalación de PostgreSQL en su ordenador.
BitRock Installer
< Atrás
Siguiente >
Cancelar



5.- Finalmente ejecutamos el pgAdmin III para comprobar que todo esté bien instalado





Es necesaria la creación de una base de datos para la parte de la autenticación CAS

Se lo hará con el postgreSQL, utilizando el siguiente script

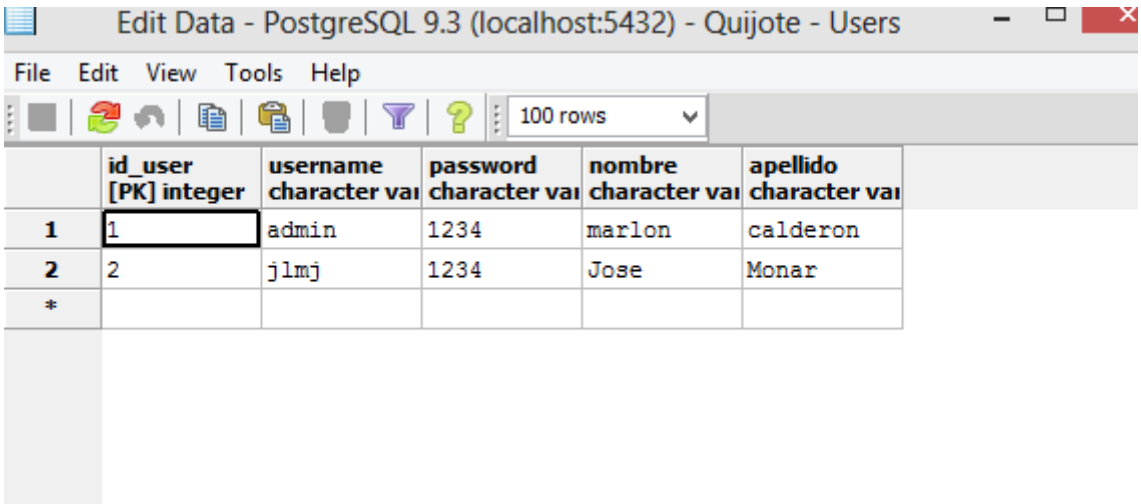
```
-- DROP DATABASE "Quijote";

CREATE DATABASE "Quijote"
  WITH OWNER = postgres
       ENCODING = 'UTF8'
       TABLESPACE = pg_default
       LC_COLLATE = 'English_United States.1252'
       LC_CTYPE = 'English_United States.1252'
       CONNECTION LIMIT = -1;
```

Luego será necesario tener una tabla que guarde los registros de los usuarios.

```
-- Table: "Users"

-- DROP TABLE "Users";

CREATE TABLE "Users"
(
  id_user serial NOT NULL,
  username character varying(100) NOT NULL,
  password character varying(100) NOT NULL,
  nombre character varying(100) NOT NULL,
  apellido character varying(100) NOT NULL,
  CONSTRAINT id_user PRIMARY KEY (id_user)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "Users"
  OWNER TO postgres;
```

Finalmente se hace el ingreso de los datos, de la siguiente manera

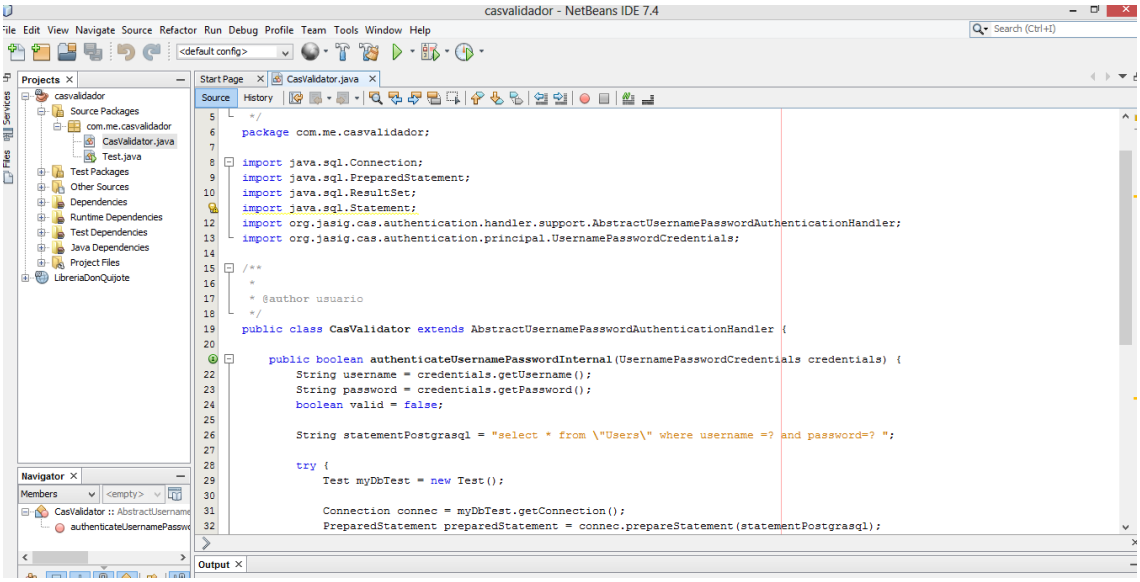
```
-- Table: "Users"

insert into "Users" (username, password, nombre, apellido)
values ('jlmj','1234','Jose','Monar');
```



	id_user [PK] integer	username character var	password character var	nombre character var	apellido character var
1	1	admin	1234	marlon	calderon
2	2	jlmj	1234	Jose	Monar
*					

Integración CAS-Spring-proyecto-Base PostgreSQL

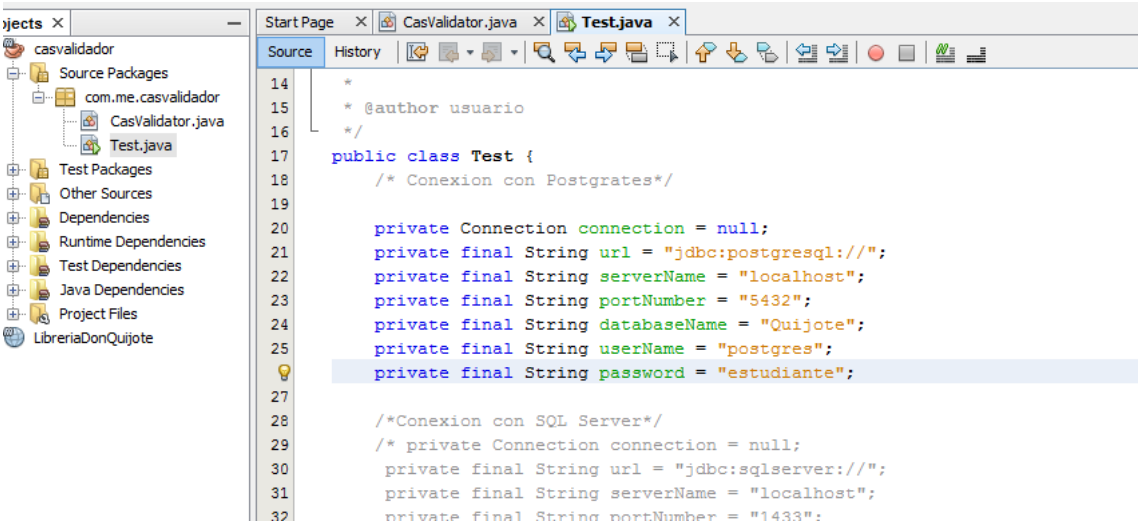
1.- Se necesita crear la clase casValidador, creando un proyecto de java bajo ese nombre.



2.- Se recomienda trabajar con los statements.

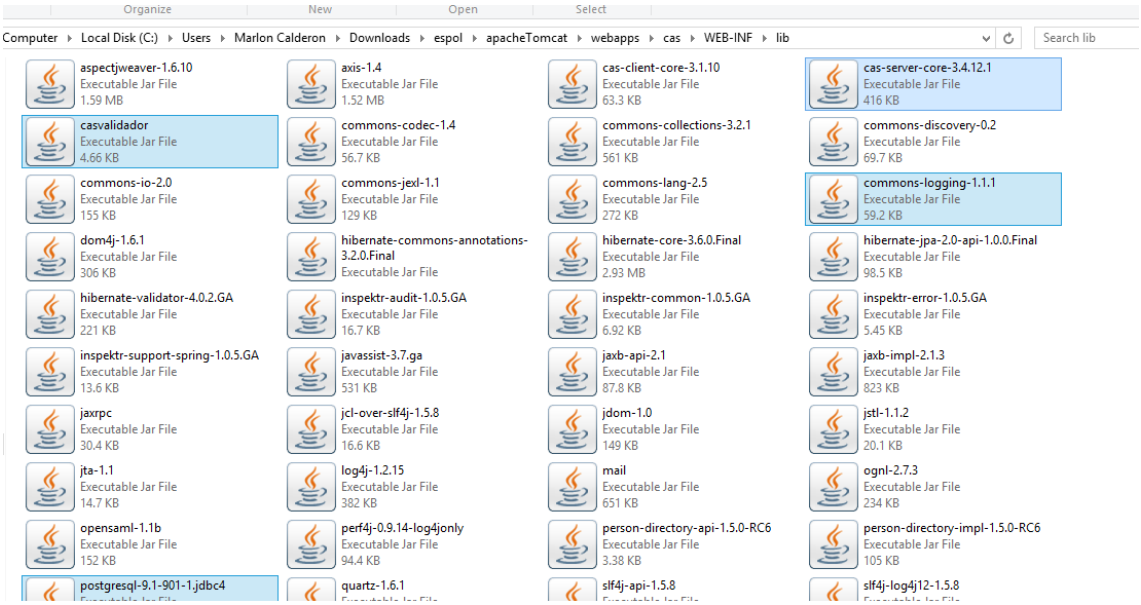
```
String statementPostgrasql = "select * from \"Users\" where username =? and password=? ";

try {
    Test myDbTest = new Test();

    Connection connec = myDbTest.getConnection();
    PreparedStatement preparedStatement = connec.prepareStatement(statementPostgrasql);
    preparedStatement.setString(1, username);
    preparedStatement.setString(2, password);

    ResultSet result = preparedStatement.executeQuery();

    while (result.next()) {
        valid = true;
```

3.- Se usa una clase de main, para realizar la conexión.



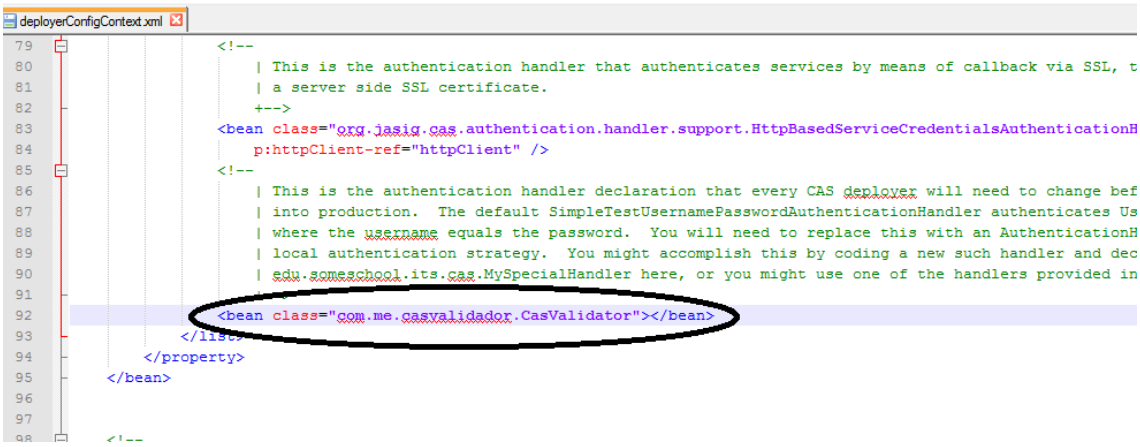
4.- Este proyecto se compilla y se genera de este modo un documento casvalidador.jar
Todas estas cosas están agregadas en el repositorio

5.- Finalmente este documento se copia junto con otros que son necesarios para CAS, estos son los siguientes: cas-server-core-3.4.12.1, casvalidador.jar, commons-loggind-1.1.1, postgresql-9.1.901-1.jdbc4

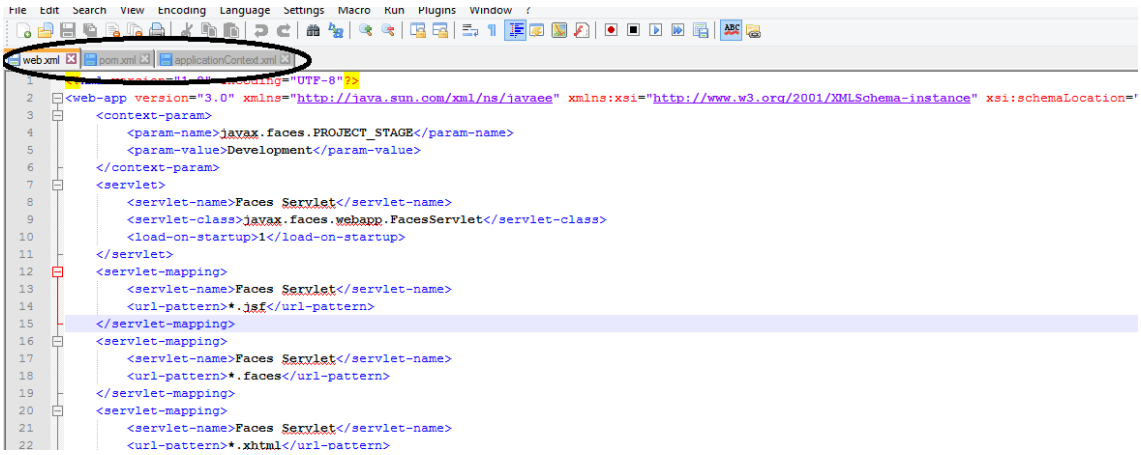


Esto va dentro de la carpeta de cas que está entre las webapps de Apache Tomcat, dentro de la parte de las librerías.

6.- Se debe cambiar el archivo deployerConfigContext.xml que se encuentra en la carpeta WEB-INF de la carpeta cas.
Se indica que se utilizará el casvalidator que hemos hecho en el paso anterior.



7.- Finalmente se realizará la configuración para el proyecto. Para esto se configuran tres archivos importantes del proyecto que son: web.xml, Pom.xml, applicationContext.xml.



En web.xml

Se agregan las siguientes líneas.

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/applicationContext.xml</param-value>
</context-param>
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>


<filter>
  <filter-name>springSecurityFilterChain</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>

<filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

En pom.xml

Se incluyen las siguientes dependencias.

```xml
<dependency>
    <groupId>commons-logging</groupId>
    <artifactId>commons-logging</artifactId>
    <version>1.1.1</version>
</dependency>
<dependency>
    <groupId>cas</groupId>
    <artifactId>casclient</artifactId>
    <version>2.1.1</version>
</dependency>
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>
<dependency>
    <groupId>org.jasig.cas.client</groupId>
    <artifactId>cas-client-core</artifactId>
    <version>3.3.1</version>
</dependency>

<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
    <version>3.0.0.RELEASE</version>
    <optional>false</optional>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>3.0.0.RELEASE</version>
    <optional>false</optional>
</dependency>
```

```
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-core</artifactId>
    <version>3.0.0.RELEASE</version>
    <optional>false</optional>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-config</artifactId>
    <version>3.0.0.RELEASE</version>
    <scope>compile</scope>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-cas-client</artifactId>
    <version>3.0.0.RELEASE</version>
    <optional>false</optional>
</dependency>
```

En applicationContext.xml

Se especifican las direcciones de las vistas, además de los roles, tipos de usuarios que tiene la aplicación y la configuración general de Spring.

```
<sec:http entry-point-ref="casAuthenticationEntryPoint" auto-config="true" path-type="ant">

    <sec:custom-filter before="CAS_FILTER" ref="casSingleSignOutFilter"/>
    <sec:custom-filter after="CAS_FILTER"  ref="casAuthenticationFilter"/>
    <sec:intercept-url pattern="/administrador/*" access="ROLE_ADMIN"/>
    <sec:intercept-url pattern="/*" access="ROLE_VENDEDOR"/>

    <sec:logout logout-success-url="https://quijote:8443/cas/logout" invalidate-session="true"/>
</sec:http>


<!--
    Required for the casProcessingFilter, so define it explicitly set and
    specify an Id Even though the authenticationManager is created by
    default when namespace based config is used.
-->
<sec:authentication-manager alias="authenticationManager">
    <sec:authentication-provider ref="casAuthenticationProvider" />
</sec:authentication-manager>
```

```
-->
<bean id="serviceProperties" class="org.springframework.security.cas.ServiceProperties">
    <property name="service" value="http://quijote:8084/pruebaCAS/j_spring_cas_security_check"/>
    <property name="sendRenew" value="false"/>
</bean>


<!--
    The CAS filter handles the redirect from the CAS server and starts the ticket validation.
-->
<bean id="casAuthenticationFilter" class="org.springframework.security.cas.web.CasAuthenticationFilter">
    <property name="authenticationManager" ref="authenticationManager"/>
</bean>


<!--
    Handles a logout request from the CAS server
-->
<bean id="casSingleSignOutFilter" class="org.jasig.cas.client.session.SingleSignOutFilter" />


<!--
    The entryPoint intercepts all the CAS authentication requests.
    It redirects to the CAS loginUrl for the CAS login page.
-->
<bean id="casAuthenticationEntryPoint" class="org.springframework.security.cas.web.CasAuthenticationEntryPoint">
    <property name="loginUrl" value="https://quijote:8443/cas/login"/>
    <property name="serviceProperties" ref="serviceProperties"/>
</bean>



<!--
   Handles the CAS ticket processing.
-->
<bean id="casAuthenticationProvider" class="org.springframework.security.cas.authentication.CasAuthenticationProvider">
    <property name="userDetailsService" ref="userServices"/>
    <property name="serviceProperties" ref="serviceProperties"/>
    <property name="ticketValidator">
        <bean class="org.jasig.cas.client.validation.Cas20ServiceTicketValidator">
            <constructor-arg index="0" value="https://quijote:8443/cas"/>
        </bean>
    </property>
    <property name="key" value="cas"/>
</bean>


<!--
    The users available for this application.
-->
<sec:user-service id="userServices">
    <sec:user name="admin" password="1234" authorities="ROLE_ADMIN, ROLE_VENDEDOR"></sec:user>
    <sec:user name="jlmi" password="1234" authorities="ROLE_VENDEDOR"></sec:user>
</sec:user-service>
```