

## Contenido

Instalación del Java Development Kit (JDK) .....	1
Instalación de Apache Tomcat .....	7
Creación del certificado .....	13
Instalación de PostgreSQL.....	16
Integración CAS-Spring-proyecto-Base PostgreSQL.....	23
Desplegando en Tomcat.....	29

## Instalación del Java Development Kit (JDK)

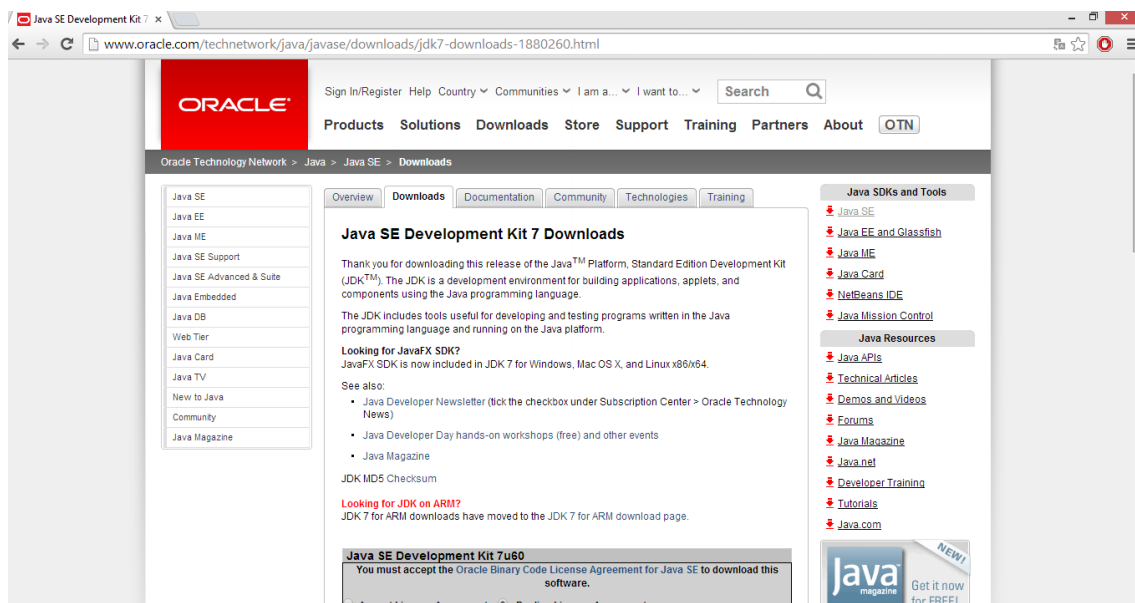
De gran importancia es que tengamos instalado el JDK, este debiera ser el primer paso a realizar. Puede sernos de gran ayuda el siguiente link

<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.**

Java SE Development Kit 7 x

www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

JDK 7 for ARM downloads have moved to the JDK 7 for ARM download page.

**Java SE Development Kit 7u60**

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	119.57 MB	<a href="#">jdk-7u60-linux-i586.rpm</a>
Linux x86	136.95 MB	<a href="#">jdk-7u60-linux-i586.tar.gz</a>
Linux x64	120.97 MB	<a href="#">jdk-7u60-linux-x64.rpm</a>
Linux x64	135.77 MB	<a href="#">jdk-7u60-linux-x64.tar.gz</a>
Mac OS X x64	185.94 MB	<a href="#">jdk-7u60-macosx-x64.dmg</a>
Solaris x86 (SVR4 package)	139.43 MB	<a href="#">jdk-7u60-solaris-i586.tar.Z</a>
Solaris x86	95.5 MB	<a href="#">jdk-7u60-solaris-i586.tar.gz</a>
Solaris x64 (SVR4 package)	24.64 MB	<a href="#">jdk-7u60-solaris-x64.tar.Z</a>
Solaris x64	16.35 MB	<a href="#">jdk-7u60-solaris-x64.tar.gz</a>
Solaris SPARC (SVR4 package)	138.73 MB	<a href="#">jdk-7u60-solaris-sparc.tar.Z</a>
Solaris SPARC	98.57 MB	<a href="#">jdk-7u60-solaris-sparc.tar.gz</a>
Solaris SPARC 64-bit (SVR4 package)	24.04 MB	<a href="#">jdk-7u60-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	18.4 MB	<a href="#">jdk-7u60-solaris-sparcv9.tar.gz</a>
Windows x86	127.91 MB	<a href="#">jdk-7u60-windows-i586.exe</a>
Windows x64	129.65 MB	<a href="#">jdk-7u60-windows-x64.exe</a>

**Java SE Development Kit 7u60 Demos and Samples Downloads**

Java SE Development Kit 7u60 Demos and Samples Downloads are released under the Oracle BSD License

Product / File Description	File Size	Download
Linux x86	20.06 MB	<a href="#">jdk-7u60-linux-i586-demos.rpm</a>
Linux x86	20.2 MB	<a href="#">jdk-7u60-linux-i586-demos.tar.gz</a>

Java.com

Java magazine

Get it now for FREE!

Subscribe Today

Webcast

Introducing Java 8

Watch Now

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

Solaris SPARC	98.57 MB	<a href="#">jdk-7u60</a>
Solaris SPARC 64-bit (SVR4 package)	24.04 MB	<a href="#">jdk-7u60</a>
Solaris SPARC 64-bit	18.4 MB	<a href="#">jdk-7u60</a>
Windows x86	127.91 MB	<a href="#">jdk-7u60</a>
Windows x64	129.65 MB	<a href="#">jdk-7u60</a>

**Java SE Development Kit 7u60 Demos and Samples Downloads**

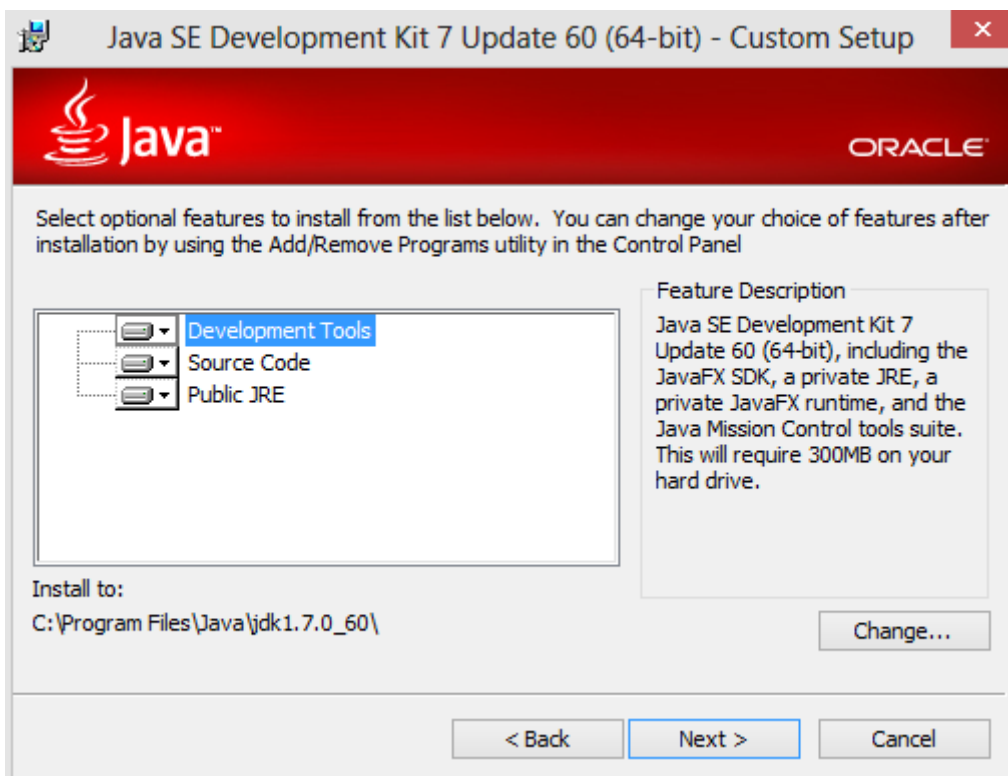
Java SE Development Kit 7u60 Demos and Samples Downloads are released under the Oracle BSD License

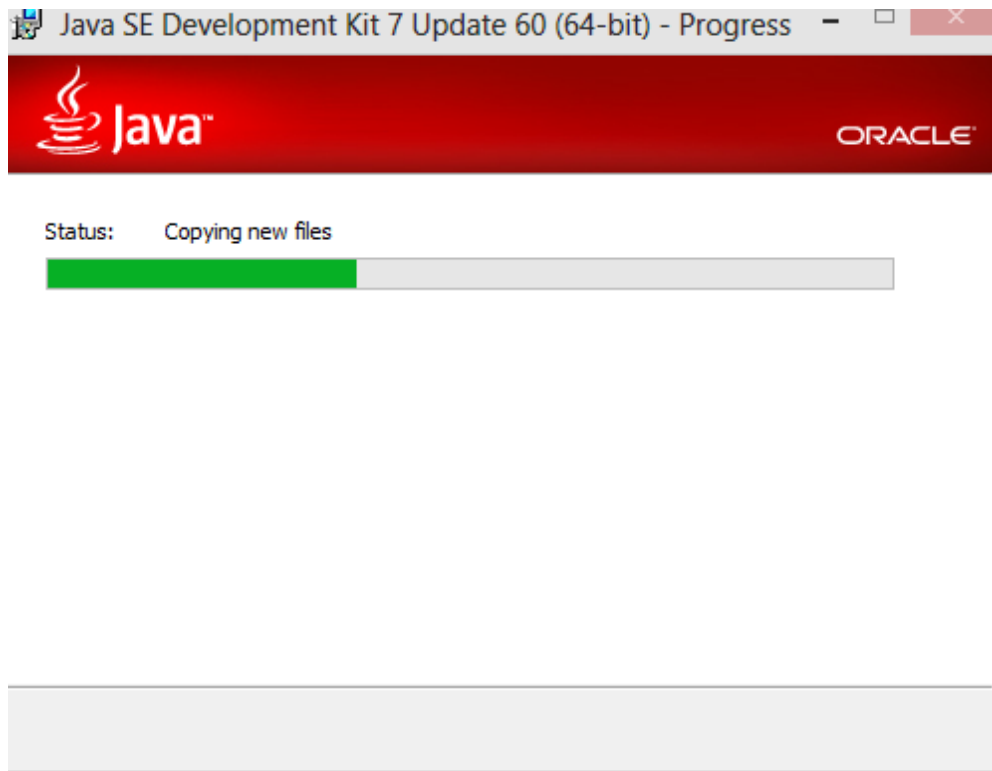
Product / File Description	File Size	Download
Linux x86	20.06 MB	<a href="#">jdk-7u60-linux-i586-demos.rpm</a>
Linux x86	20.2 MB	<a href="#">jdk-7u60-linux-i586-demos.tar.gz</a>

jdk-7u60-windows-x64.exe

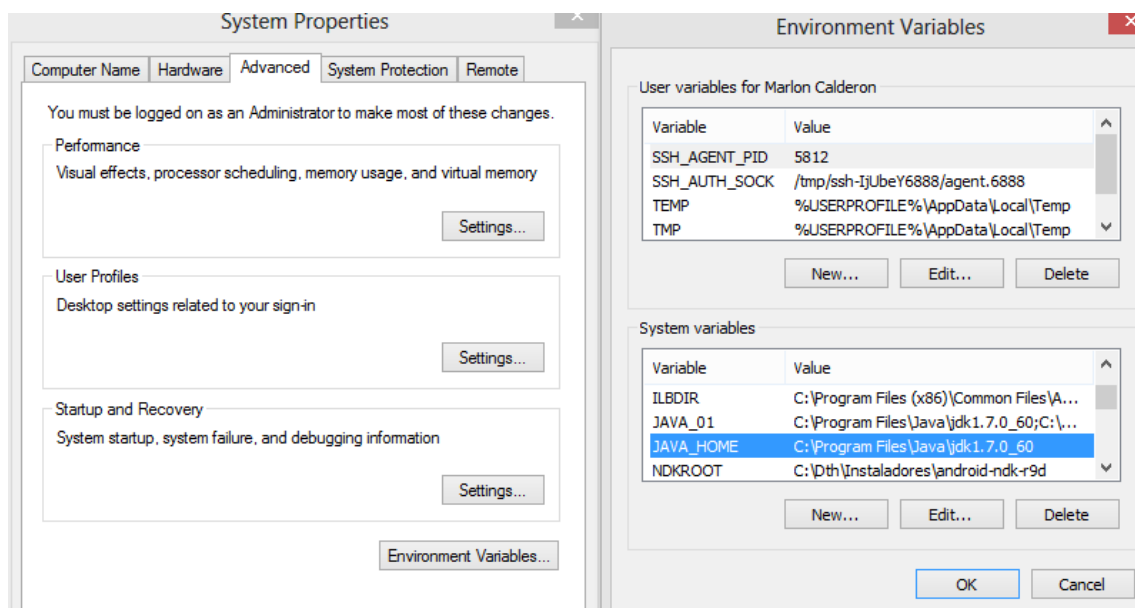
26,8/130 MB, 7 minutos r...

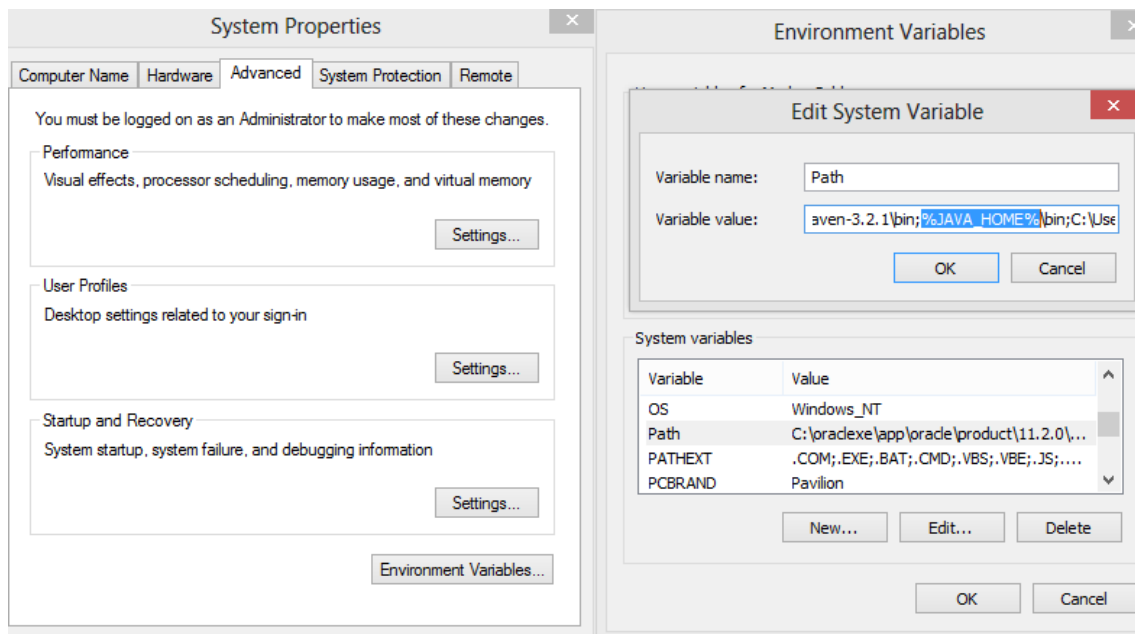
4.- Comenzamos la instalación. Básicamente se instala todo por defecto, es decir, escogiendo siguiente en cada pantalla, sin mayor tipo de configuración.





## 5.- Se configura las variables de entorno





6.- Finalmente se verifica que está bien instalado

```
Administrator: Command Prompt

C:\Users\Marlon Calderon>javac
Usage: javac <options> <source files>
where possible options include:
  -g               Generate all debugging info
  -g:none          Generate no debugging info
  -g:{lines,vars,source}  Generate only some debugging info
  -nowarn          Generate no warnings
  -verbose         Output messages about what the compiler is doing
  -deprecation     Output source locations where deprecated APIs are used
  -classpath <path> Specify where to find user class files and annotations processors
  -cp <path>       Specify where to find user class files and annotations processors
  -sourcepath <path> Specify where to find input source files
  -bootclasspath <path> Override location of bootstrap class files
  -extdirs <dirs>   Override location of installed extensions
  -endorseddirs <dirs> Override location of endorsed standards path
  -proc:{none,only} Control whether annotation processing and/or compilation is done.
  -processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run; bypasses default discovery process
  -processorpath <path> Specify where to find annotation processors
  -d <directory>   Specify where to place generated class files
  -s <directory>   Specify where to place generated source files
  -implicit:{none,class} Specify whether or not to generate class files for implicitly referenced files
  -encoding <encoding> Specify character encoding used by source files
  -source <release> Provide source compatibility with specified release

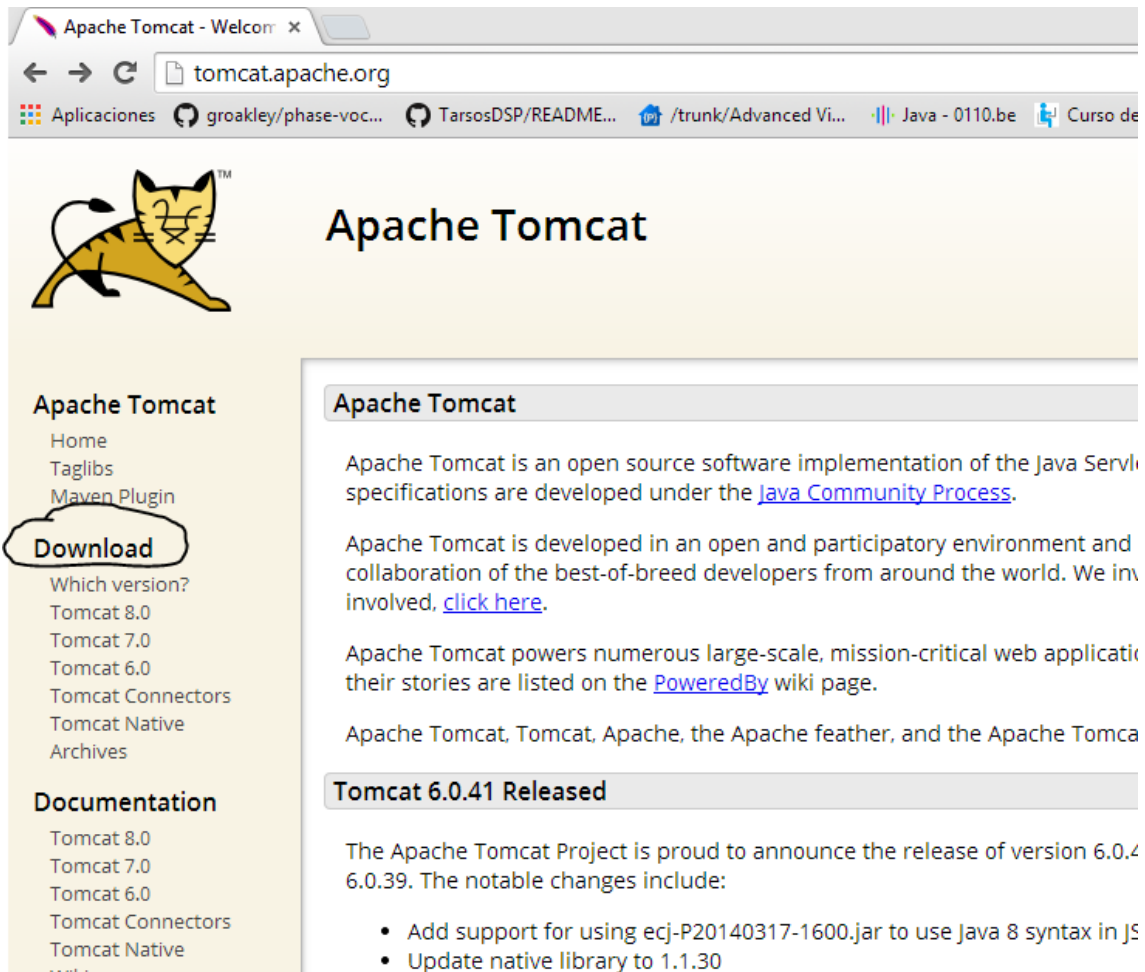
  -target <release>  Generate class files for specific VM version
  -version           Version information
  -help             Print a synopsis of standard options
  -Akey[=value]     Options to pass to annotation processors
  -X               Print a synopsis of nonstandard options
  -J<flag>          Pass <flag> directly to the runtime system
  -Werror           Terminate compilation if warnings occur
  @<filename>       Read options and filenames from file

C:\Users\Marlon Calderon>
```

# Instalación de Apache Tomcat

## Instalación de Apache Tomcat 7.0.42 (Cualquier versión superior funciona también)

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



The screenshot shows the Apache Tomcat homepage in a web browser. The browser's address bar displays 'tomcat.apache.org'. On the left sidebar, the 'Download' link is circled. The main content area features the Apache Tomcat logo, a description of the software, and a section titled 'Tomcat 6.0.41 Released'.

**Apache Tomcat**

Home  
Taglibs  
Maven Plugin  
**Download**  
Which version?  
Tomcat 8.0  
Tomcat 7.0  
Tomcat 6.0  
Tomcat Connectors  
Tomcat Native  
Archives

**Documentation**

Tomcat 8.0  
Tomcat 7.0  
Tomcat 6.0  
Tomcat Connectors  
Tomcat Native

**Apache Tomcat**

Apache Tomcat is an open source software implementation of the Java Servlet specifications are developed under the [Java Community Process](#).

Apache Tomcat is developed in an open and participatory environment and collaboration of the best-of-breed developers from around the world. We invite you to get involved, [click here](#).

Apache Tomcat powers numerous large-scale, mission-critical web applications. Their stories are listed on the [PoweredBy](#) wiki page.

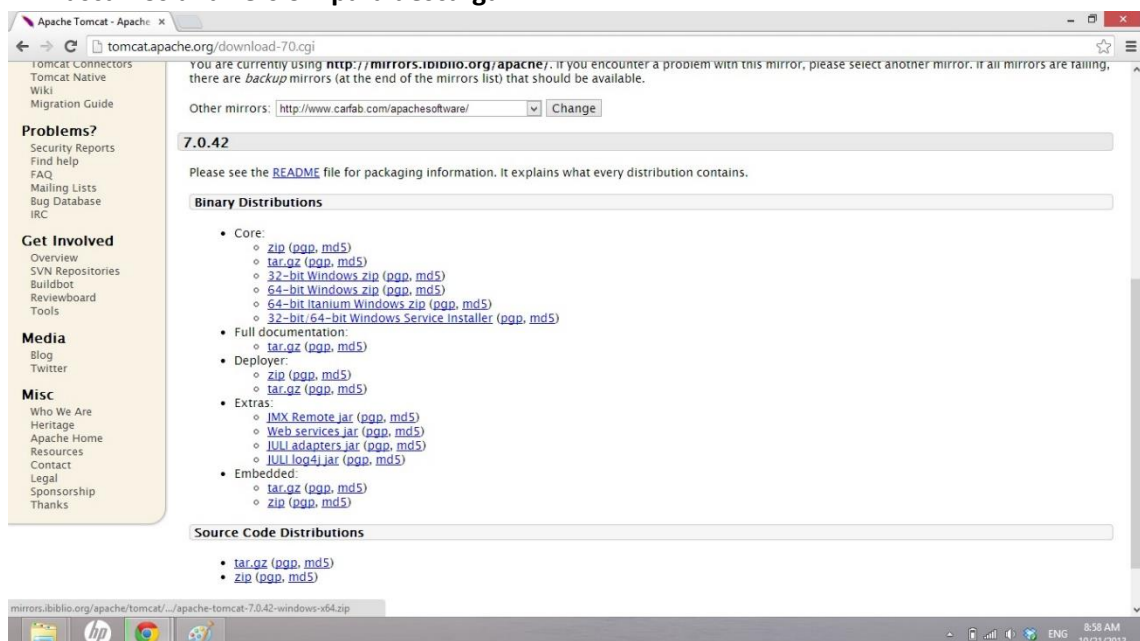
Apache Tomcat, Tomcat, Apache, the Apache feather, and the Apache Tomcat

**Tomcat 6.0.41 Released**

The Apache Tomcat Project is proud to announce the release of version 6.0.41. The notable changes include:

- Add support for using ecj-P20140317-1600.jar to use Java 8 syntax in JS
- Update native library to 1.1.30

### 2.- Buscamos una versión para descargar



The screenshot shows the Apache Tomcat download page for version 7.0.42. The browser's address bar displays 'tomcat.apache.org/download-70.cgi'. The page lists various download options for the 7.0.42 version, including core, full, and embedded distributions.

Apache Tomcat - Apache x

tomcat.apache.org/download-70.cgi

You are currently using <http://mirrors.ibiblio.org/apache/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are [backup mirrors](#) (at the end of the mirrors list) that should be available.

Other mirrors: <http://www.cafab.com/apachsoftware/> [Change](#)

**7.0.42**

Please see the [README](#) file for packaging information. It explains what every distribution contains.

**Binary Distributions**

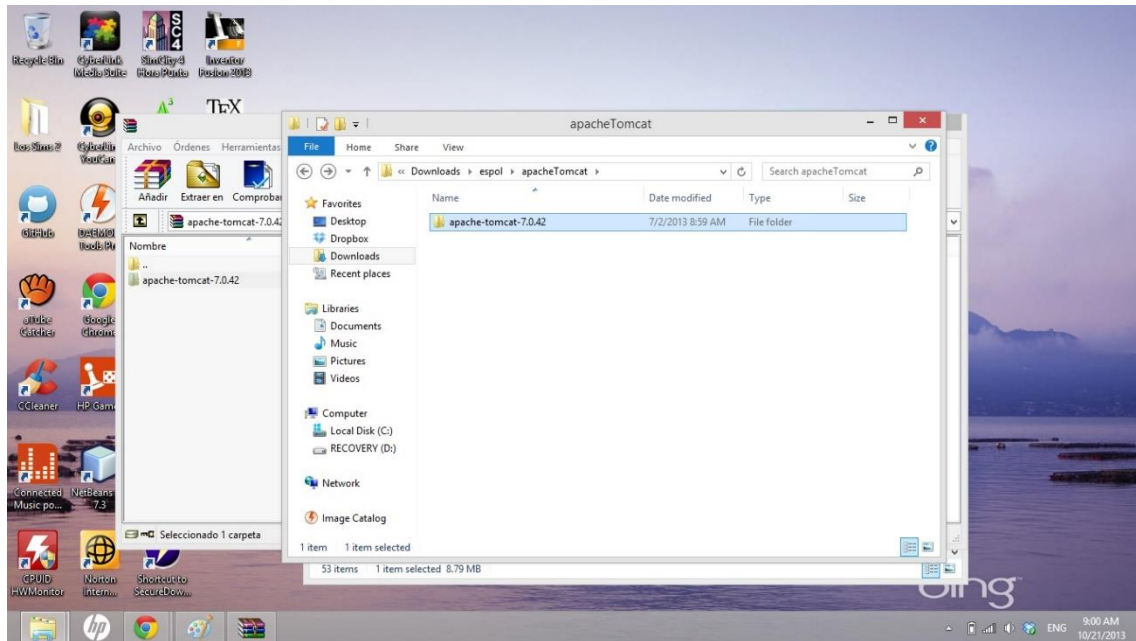
- Core:
  - [zip \(pgp, md5\)](#)
  - [tar.gz \(pgp, md5\)](#)
  - [32-bit Windows zip \(pgp, md5\)](#)
  - [64-bit Windows zip \(pgp, md5\)](#)
  - [64-bit Itanium Windows zip \(pgp, md5\)](#)
  - [32-bit, 64-bit Windows Service Installer \(pgp, md5\)](#)
- Full documentation:
  - [tar.gz \(pgp, md5\)](#)
- Deployer:
  - [zip \(pgp, md5\)](#)
  - [tar.gz \(pgp, md5\)](#)
- Extras:
  - [IMX Remote jar \(pgp, md5\)](#)
  - [Web services jar \(pgp, md5\)](#)
  - [JULI adapters jar \(pgp, md5\)](#)
  - [JULI log4j jar \(pgp, md5\)](#)
- Embedded:
  - [tar.gz \(pgp, md5\)](#)
  - [zip \(pgp, md5\)](#)

**Source Code Distributions**

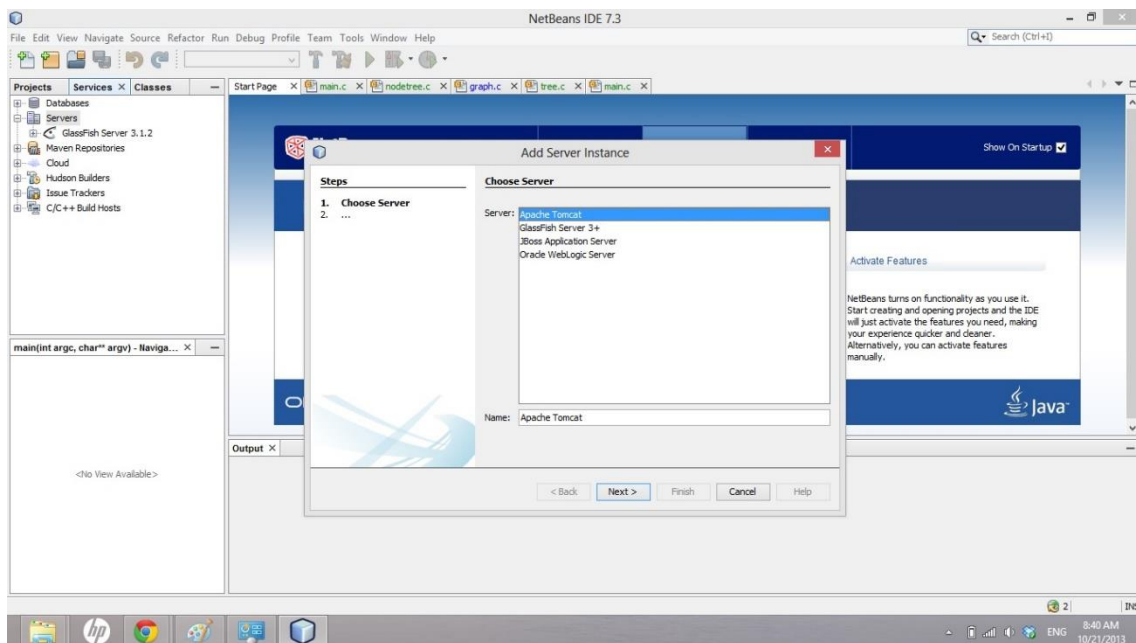
- [tar.gz \(pgp, md5\)](#)
- [zip \(pgp, md5\)](#)

mirrors.ibiblio.org/apache/tomcat/.../apache-tomcat-7.0.42-windows-x64.zip

### 3.- Descargamos tanto el zip como el instalador

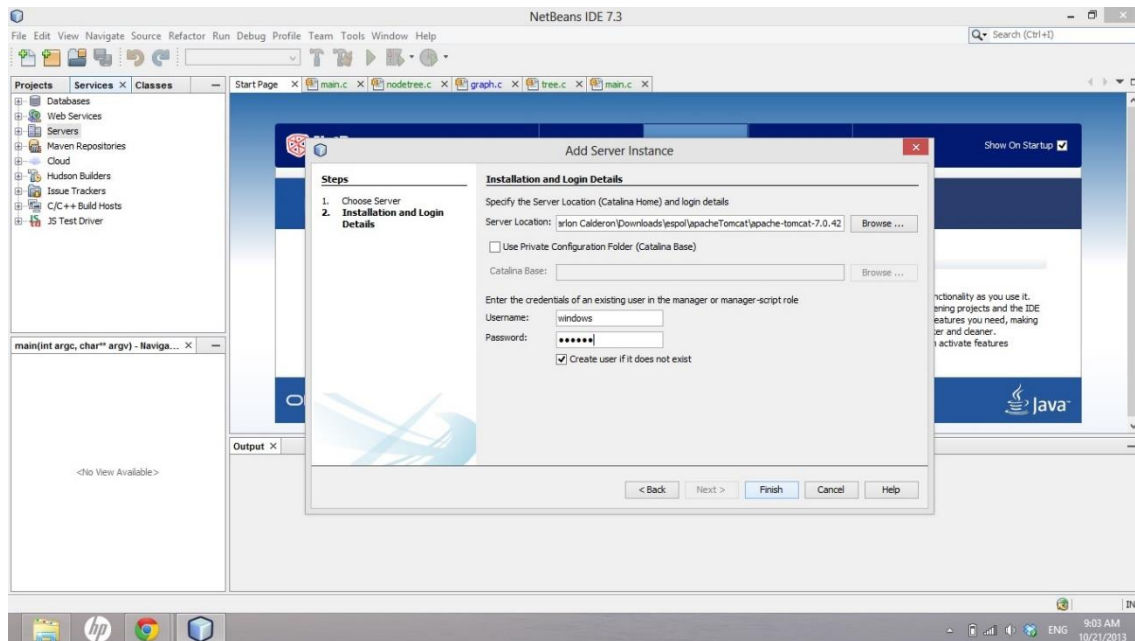


### 4.- Abrimos NetBeans y nos vamos a la sección services, en servers, escogemos agregar server (add server) escogemos Apache Tomcat

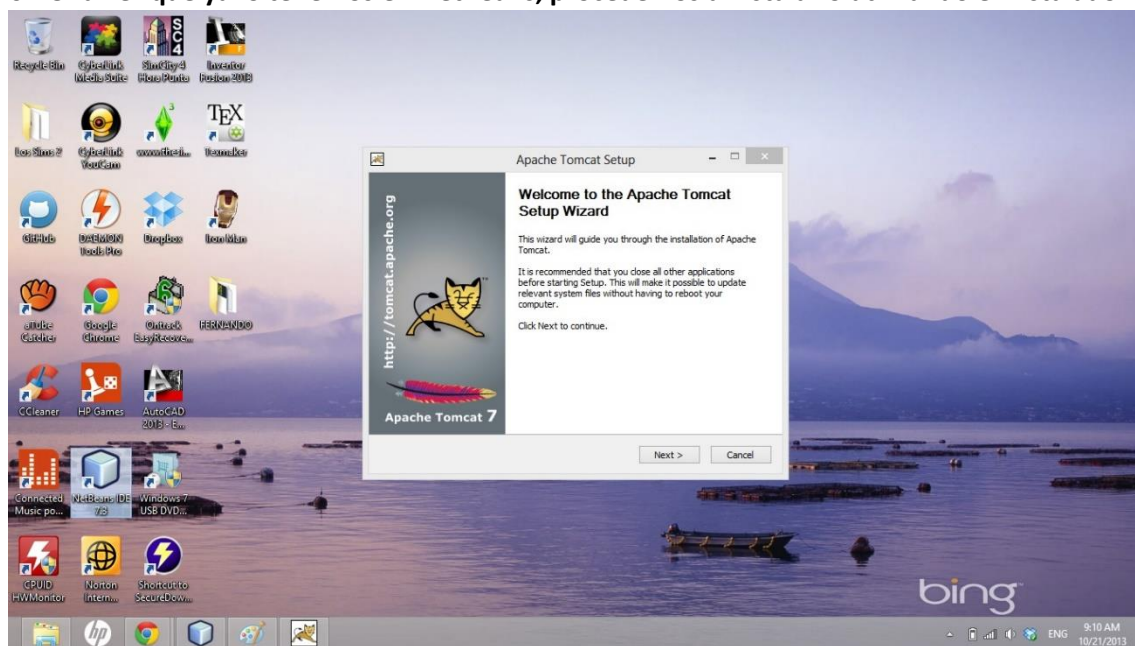




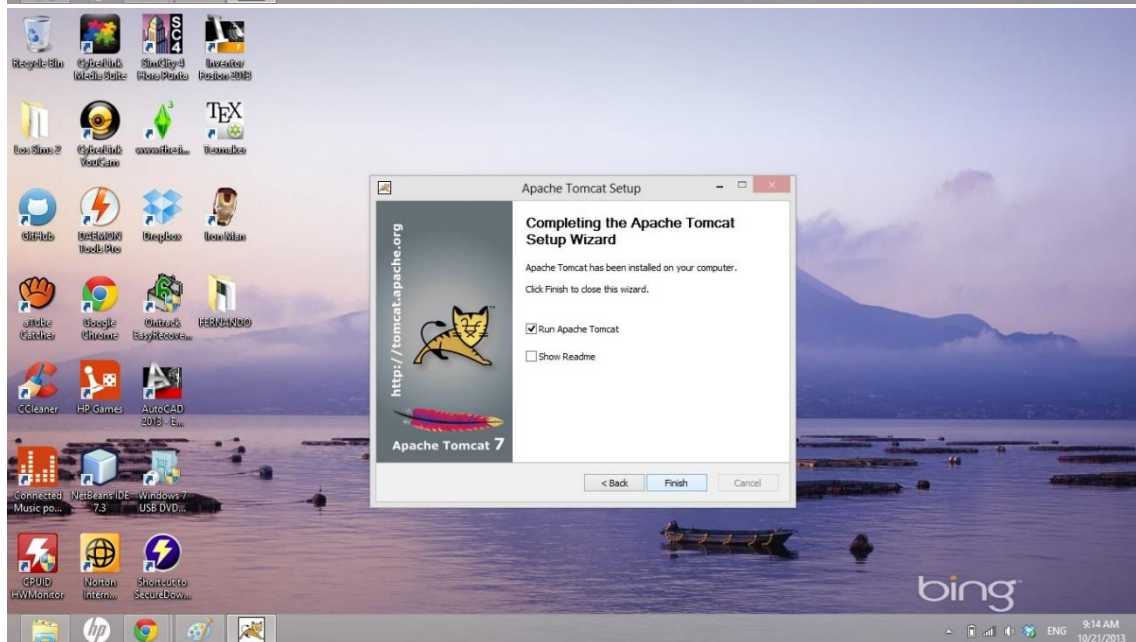
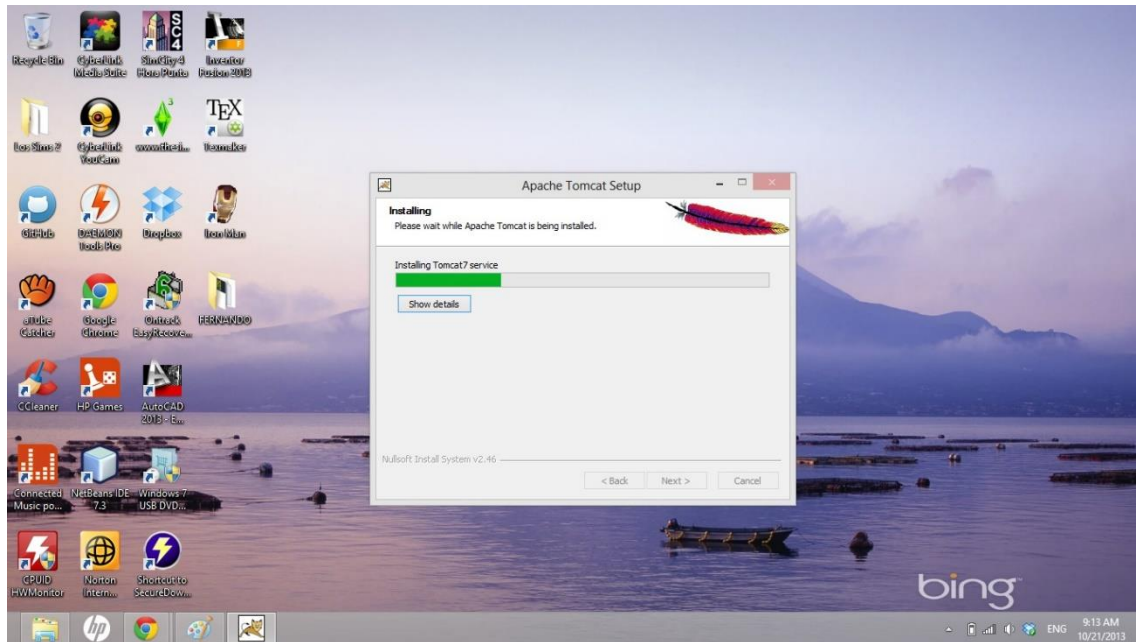
**5.-Buscamos el directorio del apache tomcat y lo ingresamos, le damos un usuario y una clave.**

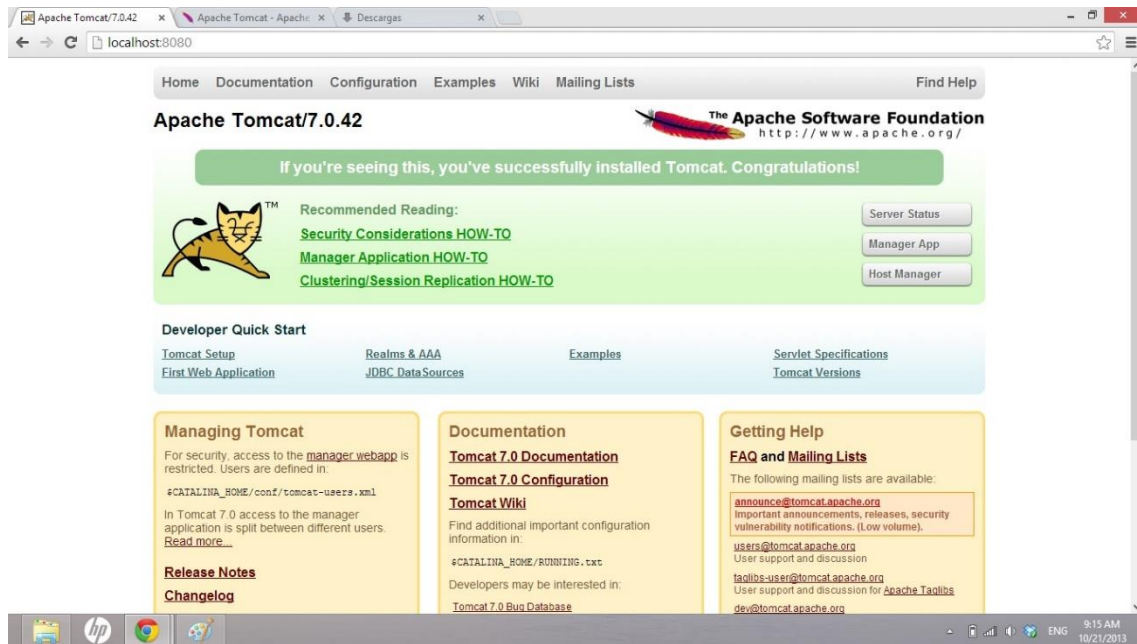


**6.- Una vez que ya lo tenemos en NetBeans, procedemos a instalarlo utilizando el instalador.**

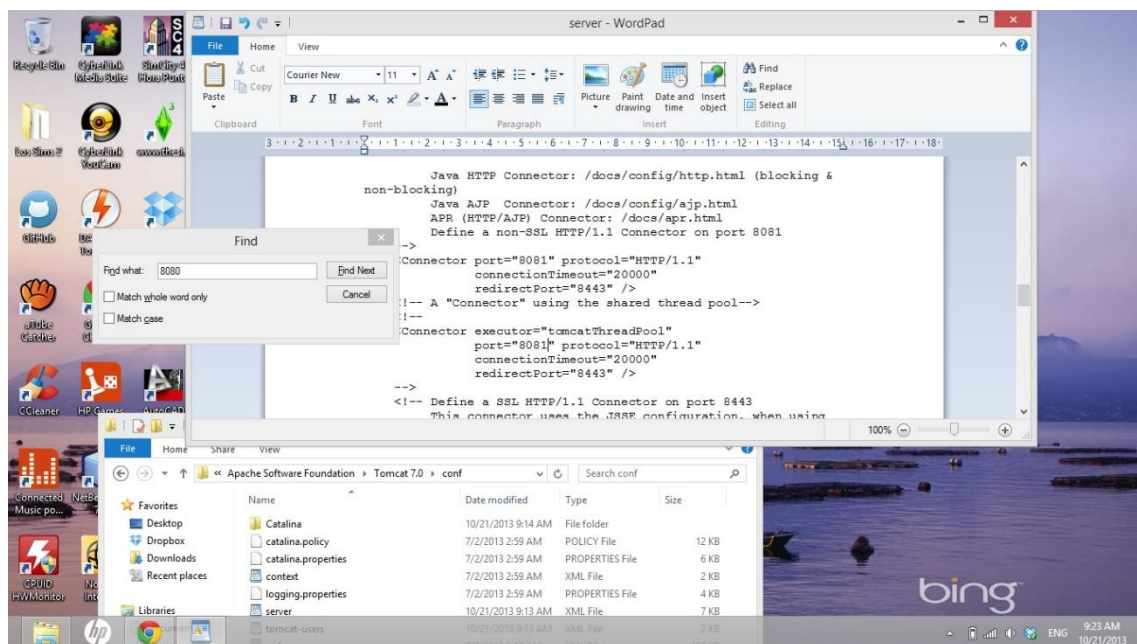


**7.- Si Seguimos la instalación por defecto tendremos lo siguiente**





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

HomeDocumentationConfigurationExamplesWikiMailing ListsFind 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 Data Sources

[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](mailto:announce@tomcat.apache.org)  
Important announcements, releases, security vulnerability notifications. (Low volume).

[users@tomcat.apache.org](mailto:users@tomcat.apache.org)  
User support and discussion

[taglibs-user@tomcat.apache.org](mailto:taglibs-user@tomcat.apache.org)  
User support and discussion for [Apache Taglibs](#)

[dev@tomcat.apache.org](mailto:dev@tomcat.apache.org)



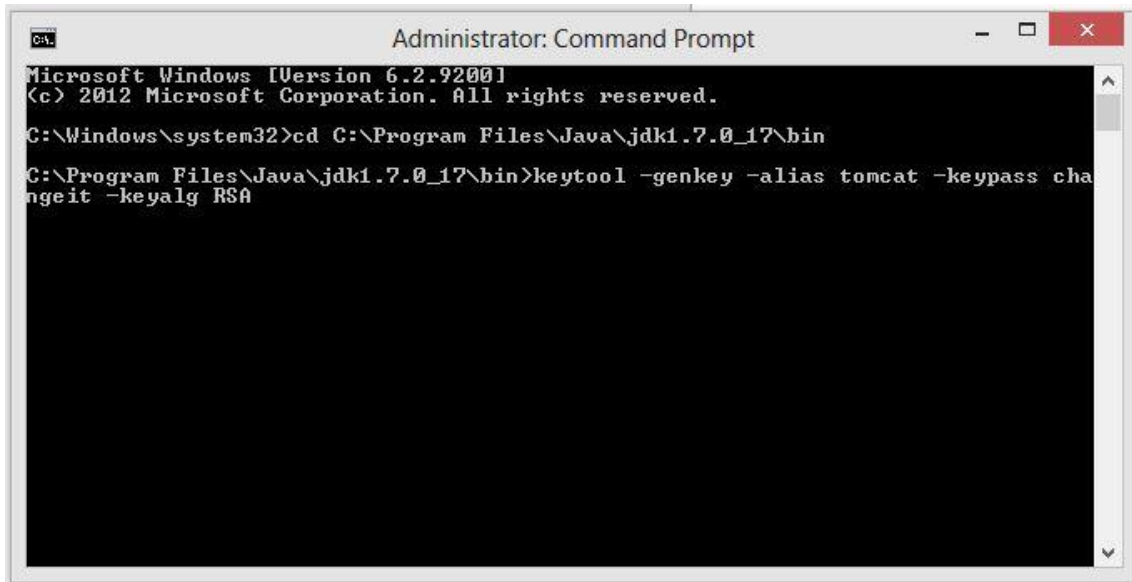
ENG9:39 AM10/21/2013



## Creación del certificado

### Creando certificado

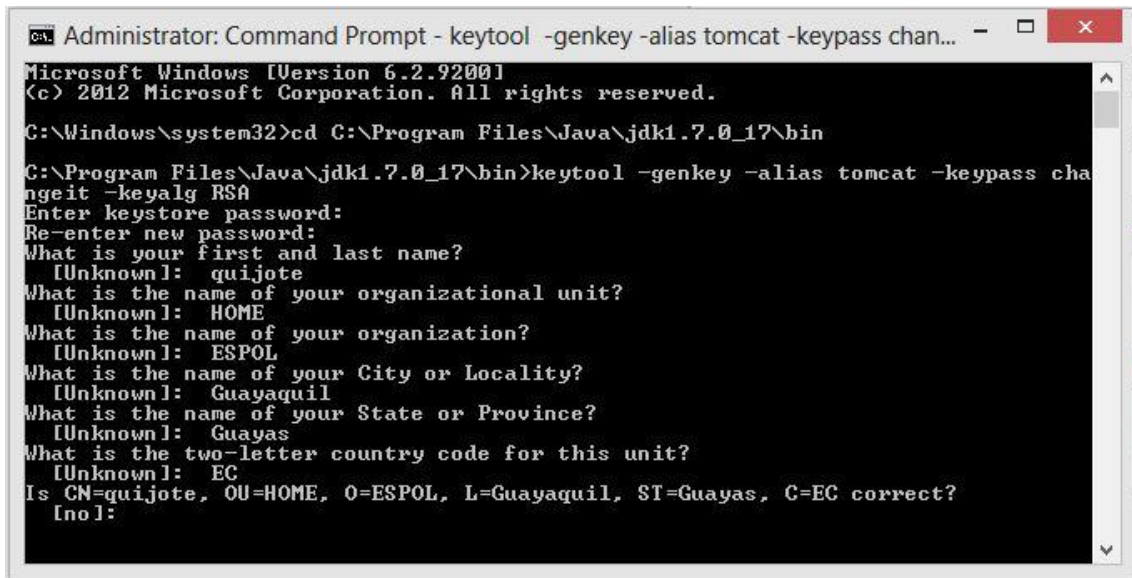
1. Trabajamos directamente con el CMD, es necesario abrirlo en modo de administrador. Ingresamos el comando para crear, el cual se ve en la captura de la imagen, es de gran importancia aclarar que se trabaja en la ruta donde se encuentra el jdk previamente instalado.



```
Administrator: Command Prompt
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\Java\jdk1.7.0_17\bin
C:\Program Files\Java\jdk1.7.0_17\bin>keytool -genkey -alias tomcat -keypass cha
ngeit -keyalg RSA
```

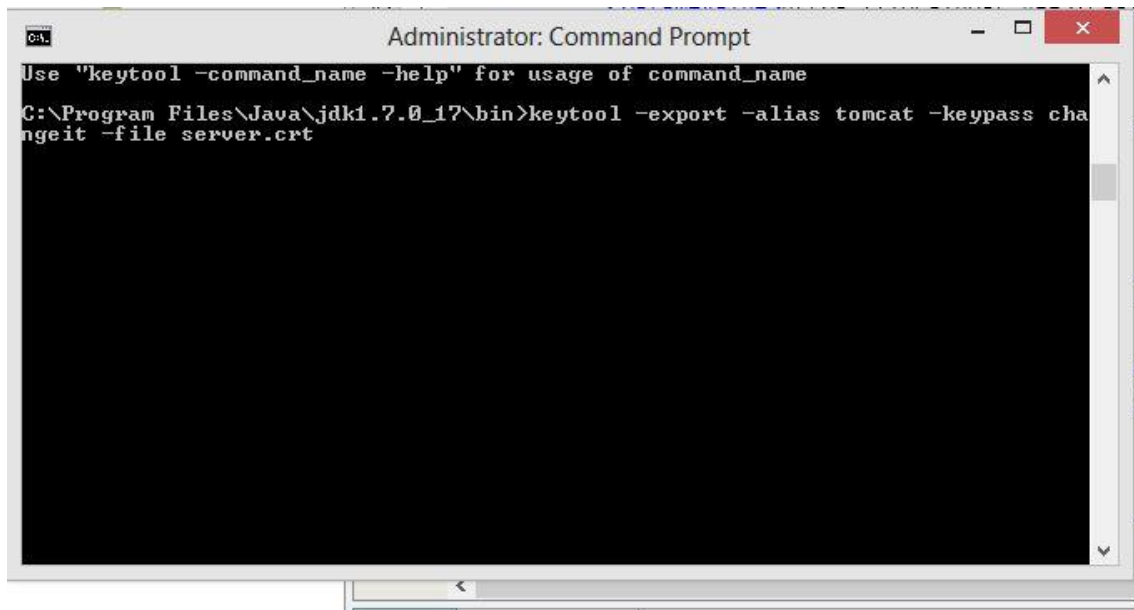
2.- Llenamos los datos que se nos van solicitando



```
Administrator: Command Prompt - keytool -genkey -alias tomcat -keypass chan...
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

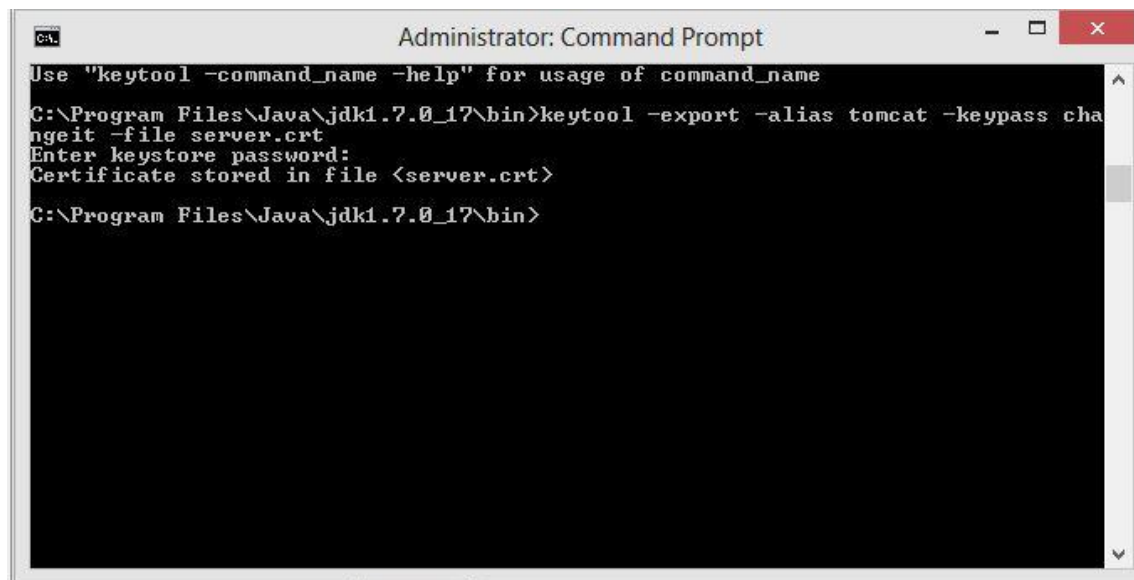
C:\Windows\system32>cd C:\Program Files\Java\jdk1.7.0_17\bin
C:\Program Files\Java\jdk1.7.0_17\bin>keytool -genkey -alias tomcat -keypass cha
ngeit -keyalg RSA
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: quijote
What is the name of your organizational unit?
[Unknown]: HOME
What is the name of your organization?
[Unknown]: ESPOL
What is the name of your City or Locality?
[Unknown]: Guayaquil
What is the name of your State or Province?
[Unknown]: Guayas
What is the two-letter country code for this unit?
[Unknown]: EC
Is CN=quijote, OU=HOME, O=ESPOL, L=Guayaquil, ST=Guayas, C=EC correct?
[no]:
```

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



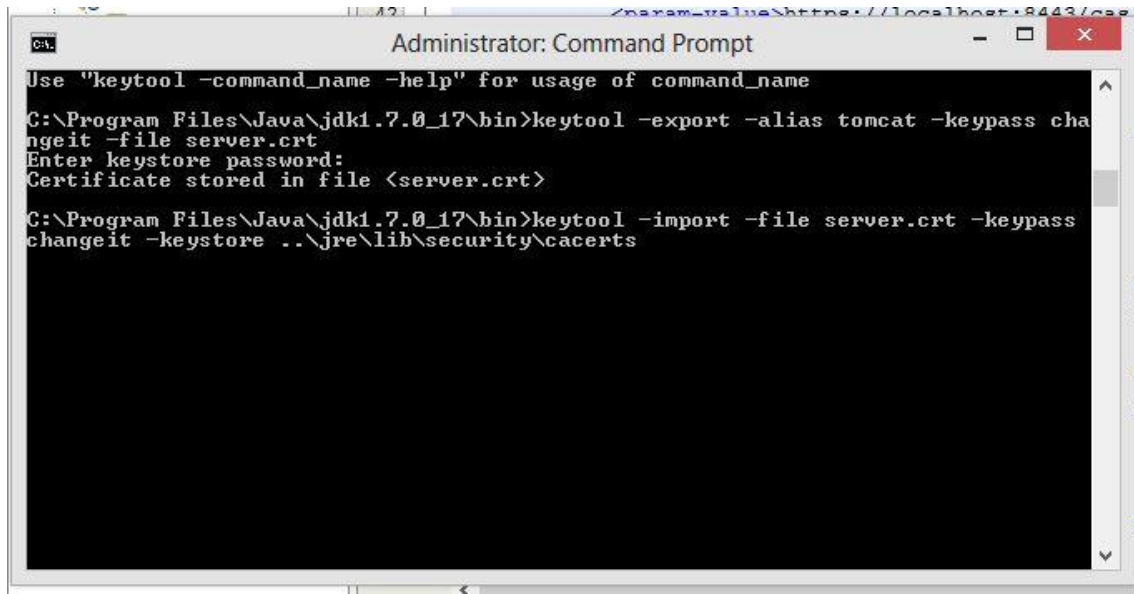
```
Administrator: Command Prompt
Use "keytool -command_name -help" for usage of command_name
C:\Program Files\Java\jdk1.7.0_17\bin>keytool -export -alias tomcat -keypass changeit -file server.crt
```

4.- El certificado creado, será guardado



```
Administrator: Command Prompt
Use "keytool -command_name -help" for usage of command_name
C:\Program Files\Java\jdk1.7.0_17\bin>keytool -export -alias tomcat -keypass changeit -file server.crt
Enter keystore password:
Certificate stored in file <server.crt>
C:\Program Files\Java\jdk1.7.0_17\bin>
```

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



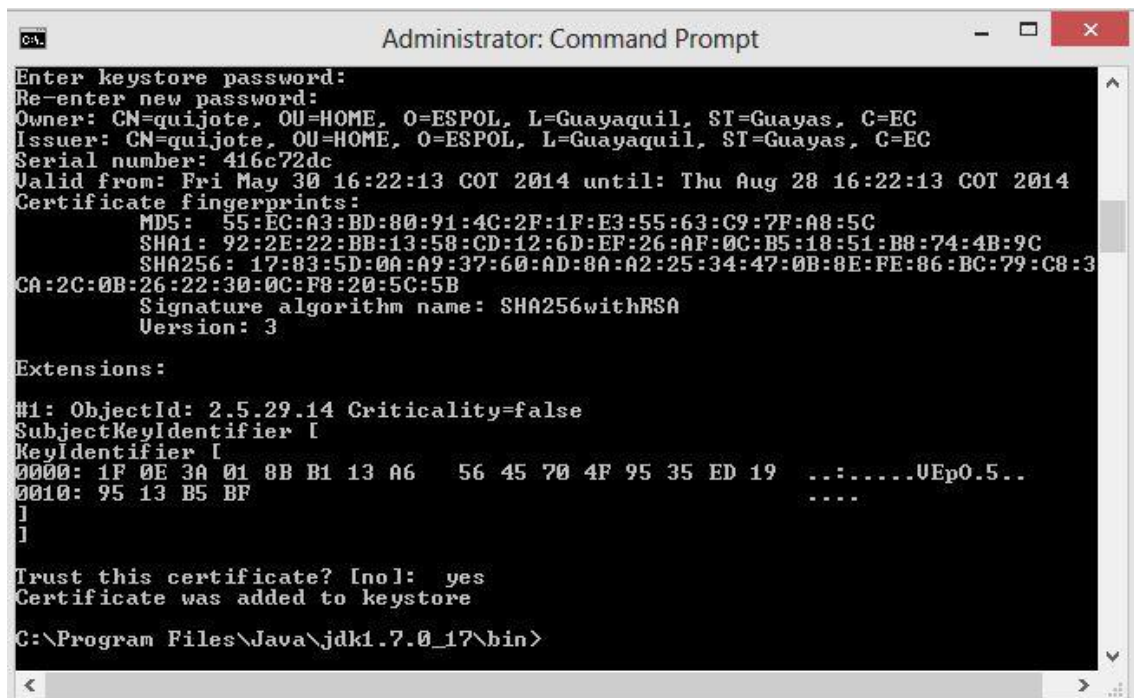
```
Administrator: Command Prompt

Use "keytool -command_name -help" for usage of command_name

C:\Program Files\Java\jdk1.7.0_17\bin>keytool -export -alias tomcat -keypass changeit -file server.crt
Enter keystore password:
Certificate stored in file <server.crt>

C:\Program Files\Java\jdk1.7.0_17\bin>keytool -import -file server.crt -keypass changeit -keystore ..\jre\lib\security\cacerts
```

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



```
Administrator: Command Prompt

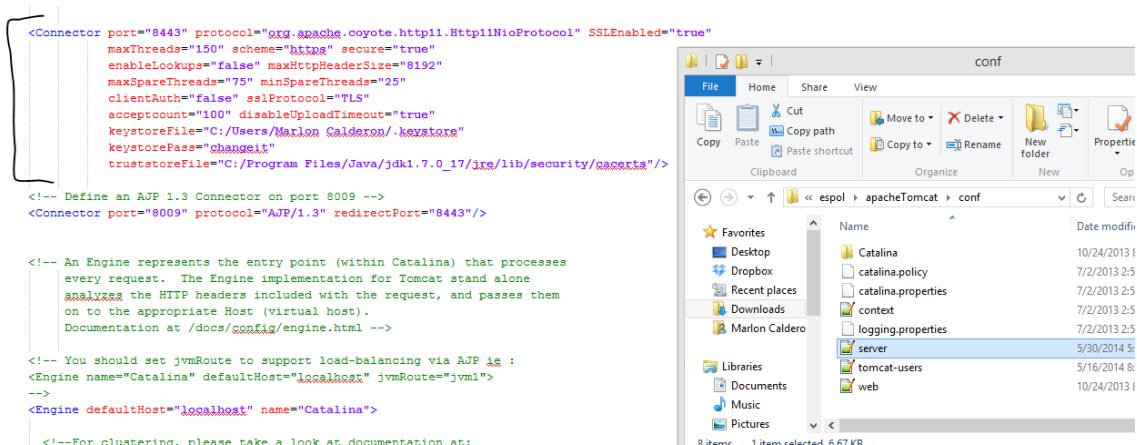
Enter keystore password:
Re-enter new password:
Owner: CN=quijote, OU=HOME, O=ESPOL, L=Guayaquil, ST=Guayas, C=EC
Issuer: CN=quijote, OU=HOME, O=ESPOL, L=Guayaquil, ST=Guayas, C=EC
Serial number: 416c72dc
Valid from: Fri May 30 16:22:13 COT 2014 until: Thu Aug 28 16:22:13 COT 2014
Certificate fingerprints:
    MD5: 55:EC:A3:BD:80:91:4C:2F:1F:E3:55:63:C9:7F:A8:5C
    SHA1: 92:2E:22:BB:13:58:CD:12:6D:EF:26:AF:0C:B5:18:51:B8:74:4B:9C
    SHA256: 17:83:5D:0A:A9:37:60:AD:8A:A2:25:34:47:0B:8E:FE:86:BC:79:C8:3
CA:2C:0B:26:22:30:0C:F8:20:5C:5B
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 1F 0E 3A 01 8B B1 13 A6    56 45 70 4F 95 35 ED 19    ...:.....UEp0.5..
0010: 95 13 B5 BF                                     ....
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore

C:\Program Files\Java\jdk1.7.0_17\bin>
```

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.



Esto es lo que exactamente esta.

```
<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.

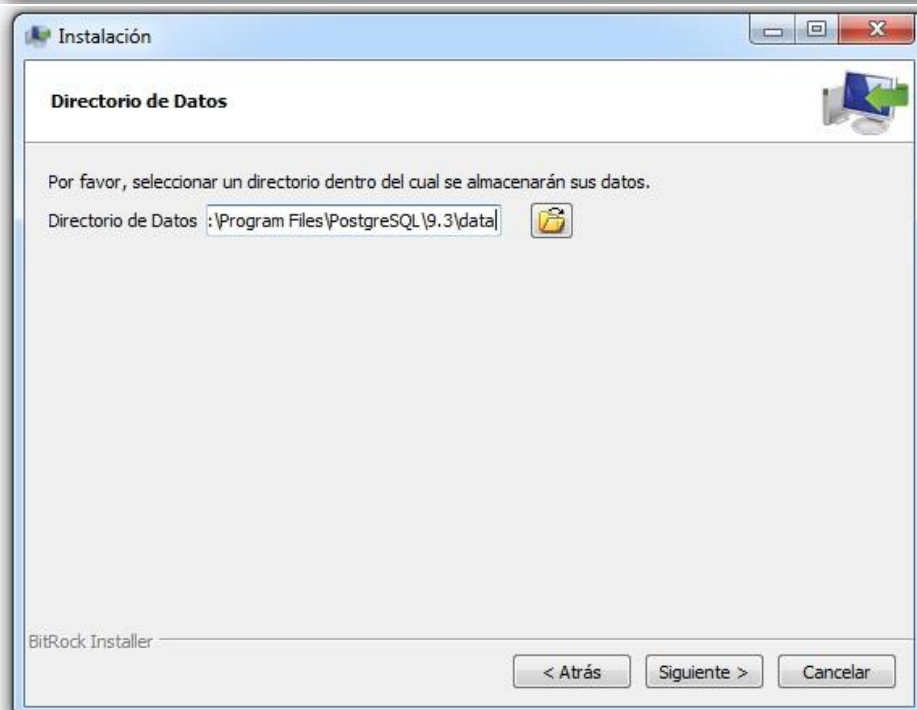
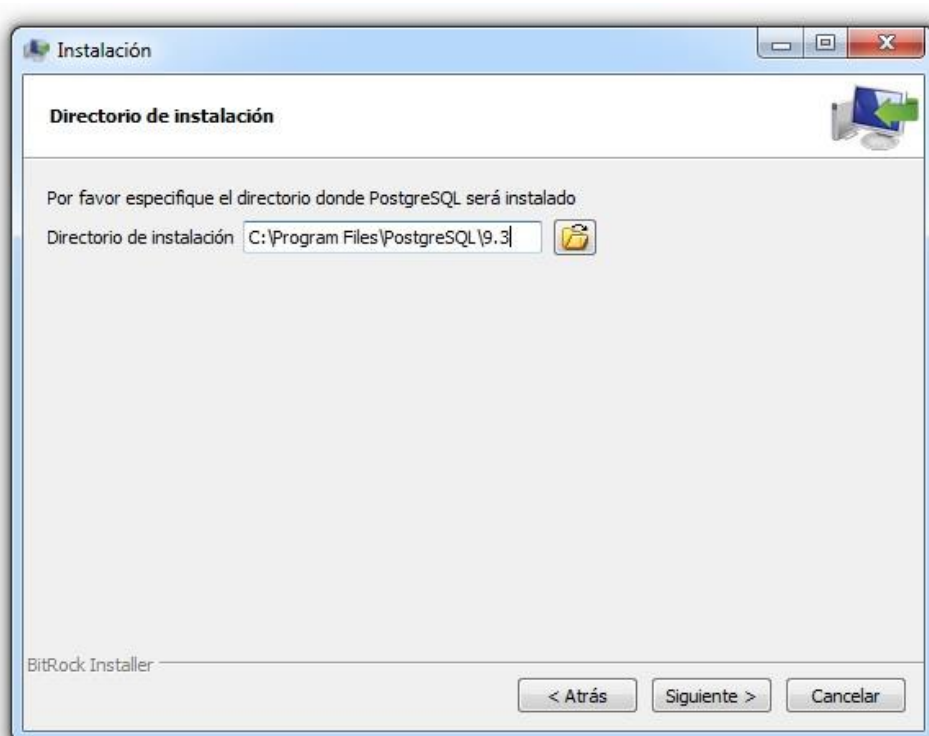
<http://www.enterprisedb.com/products-services-training/pgdownload#windows>

2.- Una vez descargado empezamos la instalación

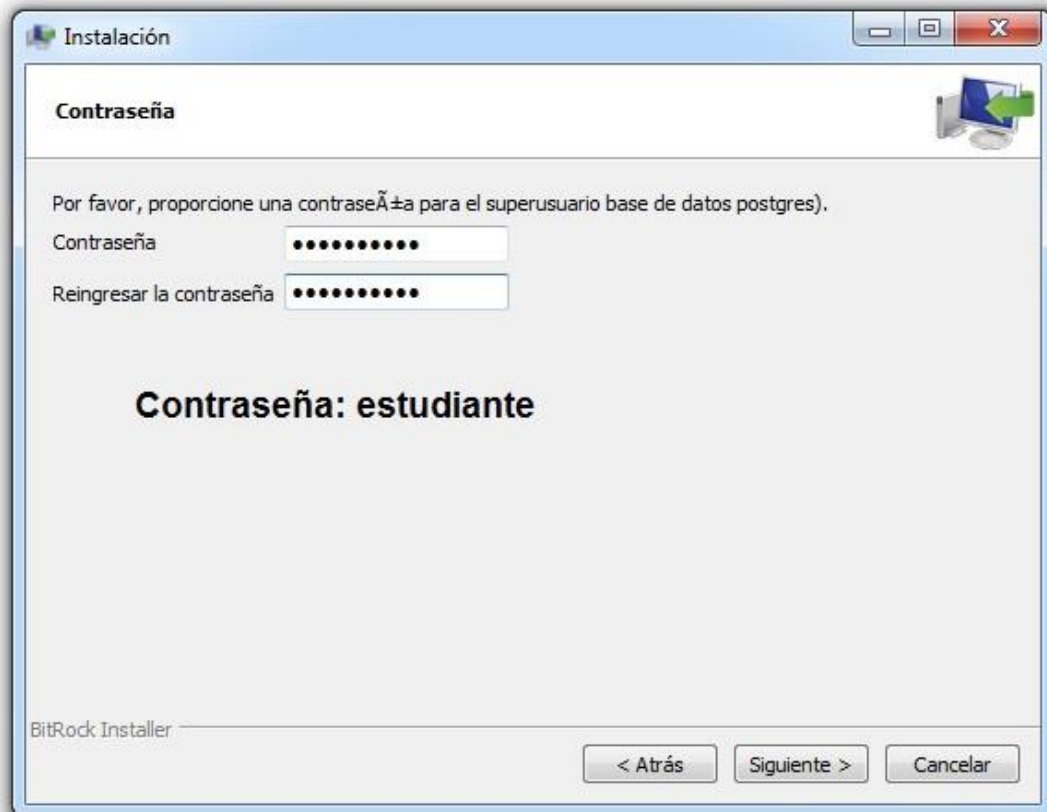




3.- Se selecciona siguiente, lo hicimos con las configuraciones por defecto, hasta llegar al punto donde nos pide la contraseña.



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



**Instalación**

**Contraseña**

Por favor, proporcione una contraseña para el superusuario base de datos postgres).

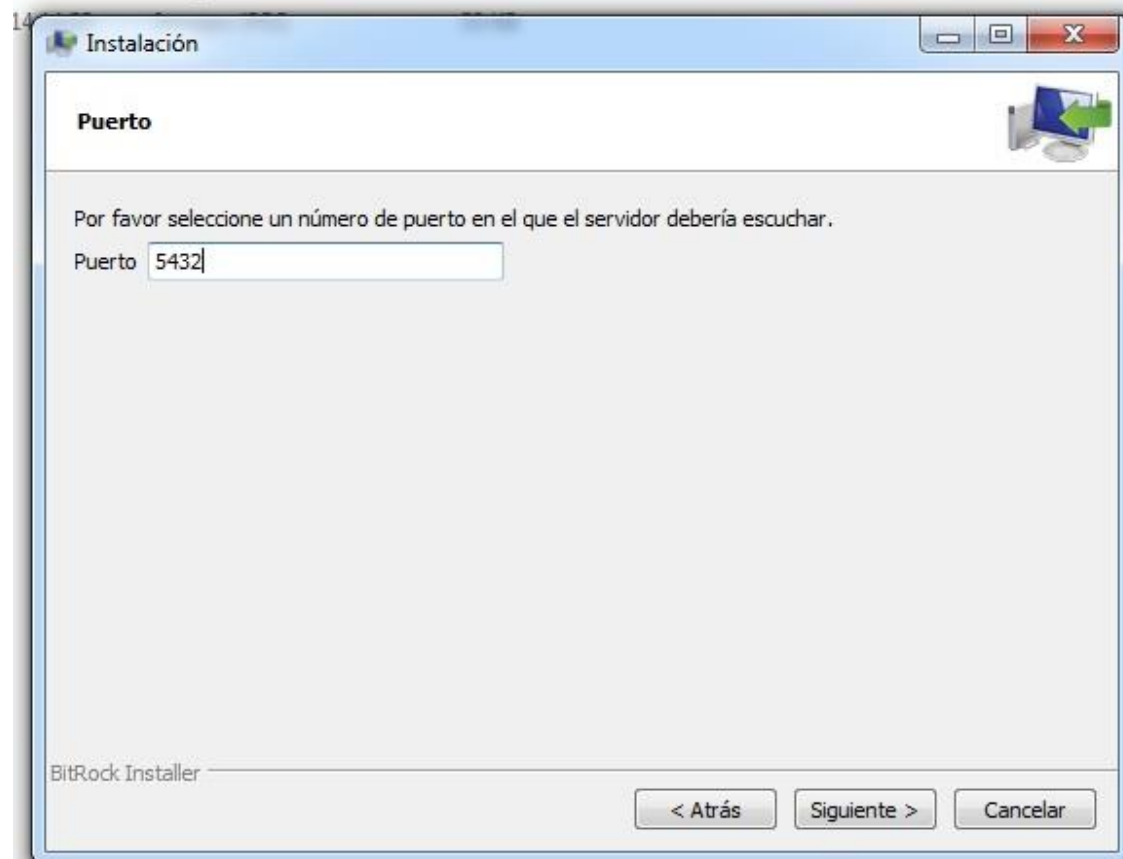
Contraseña

Reingresar la contraseña

**Contraseña: estudiante**

BitRock Installer

< Atrás    Siguiendo >    Cancelar



**Instalación**

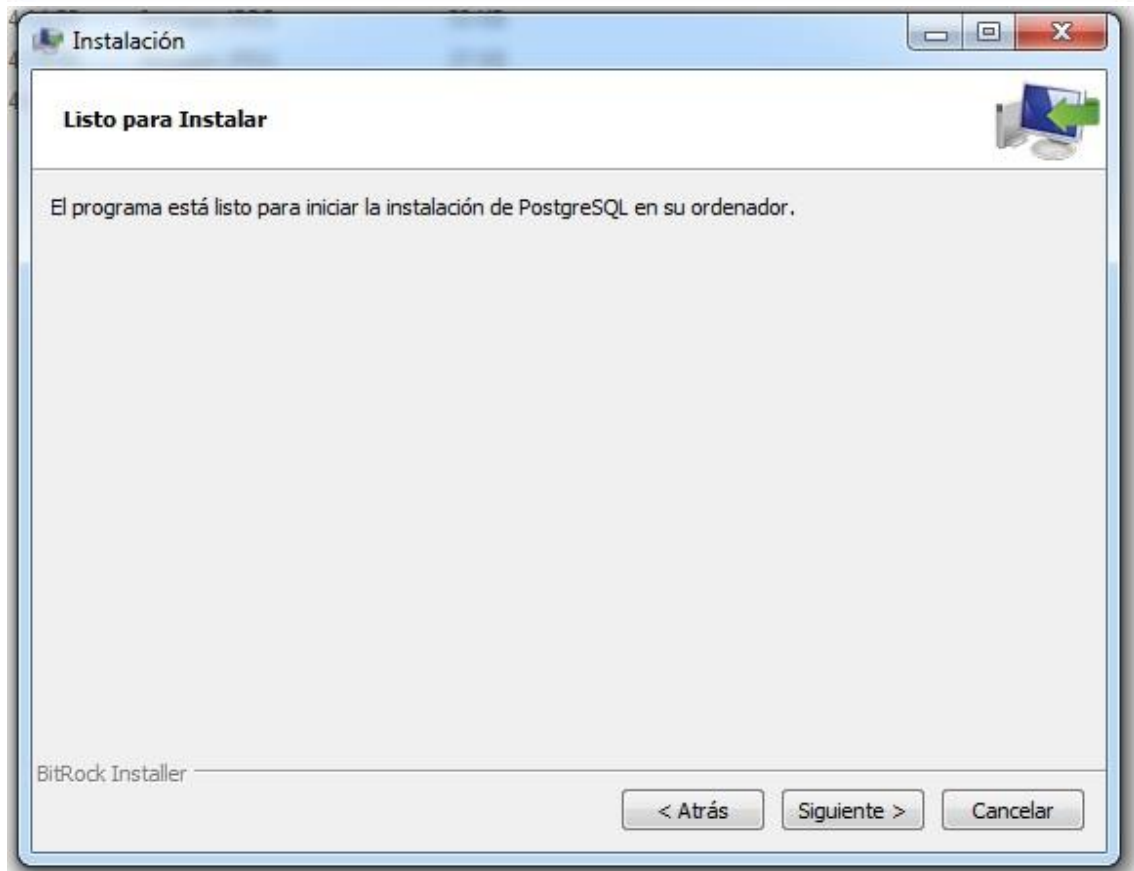
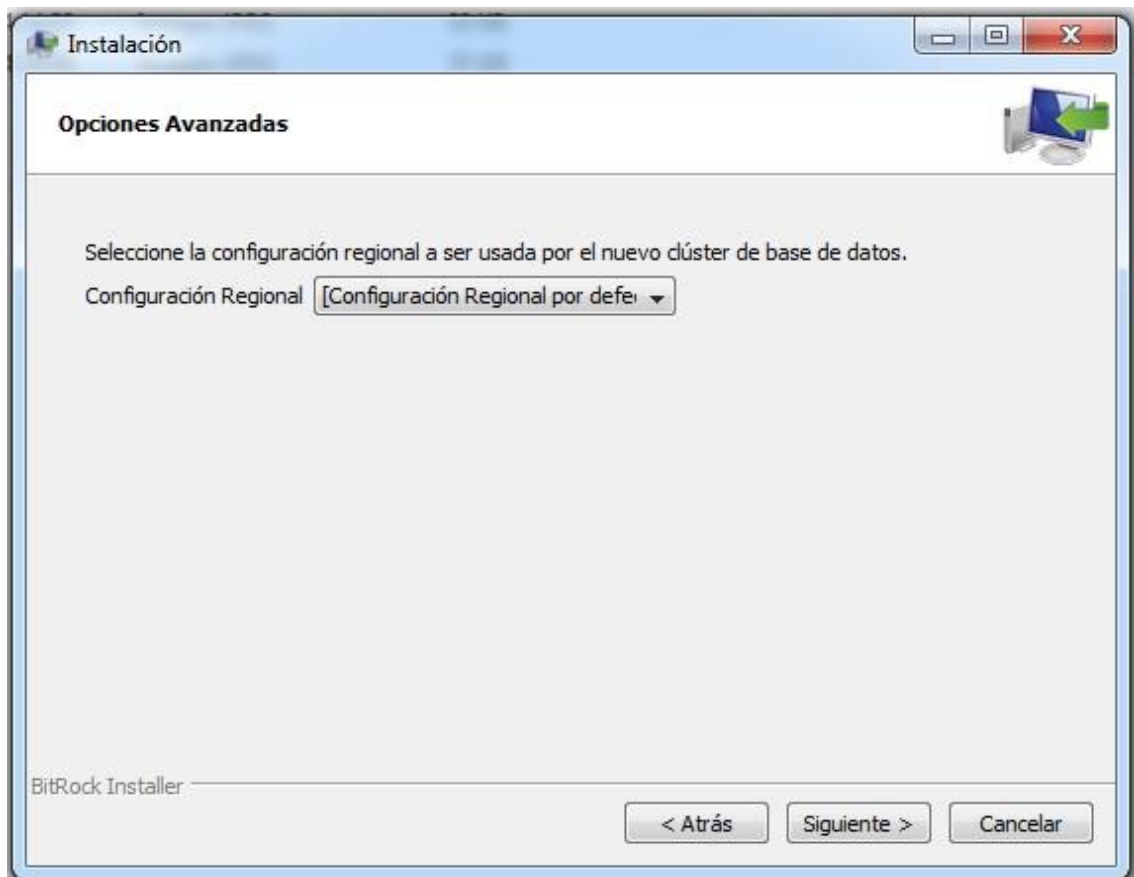
**Puerto**

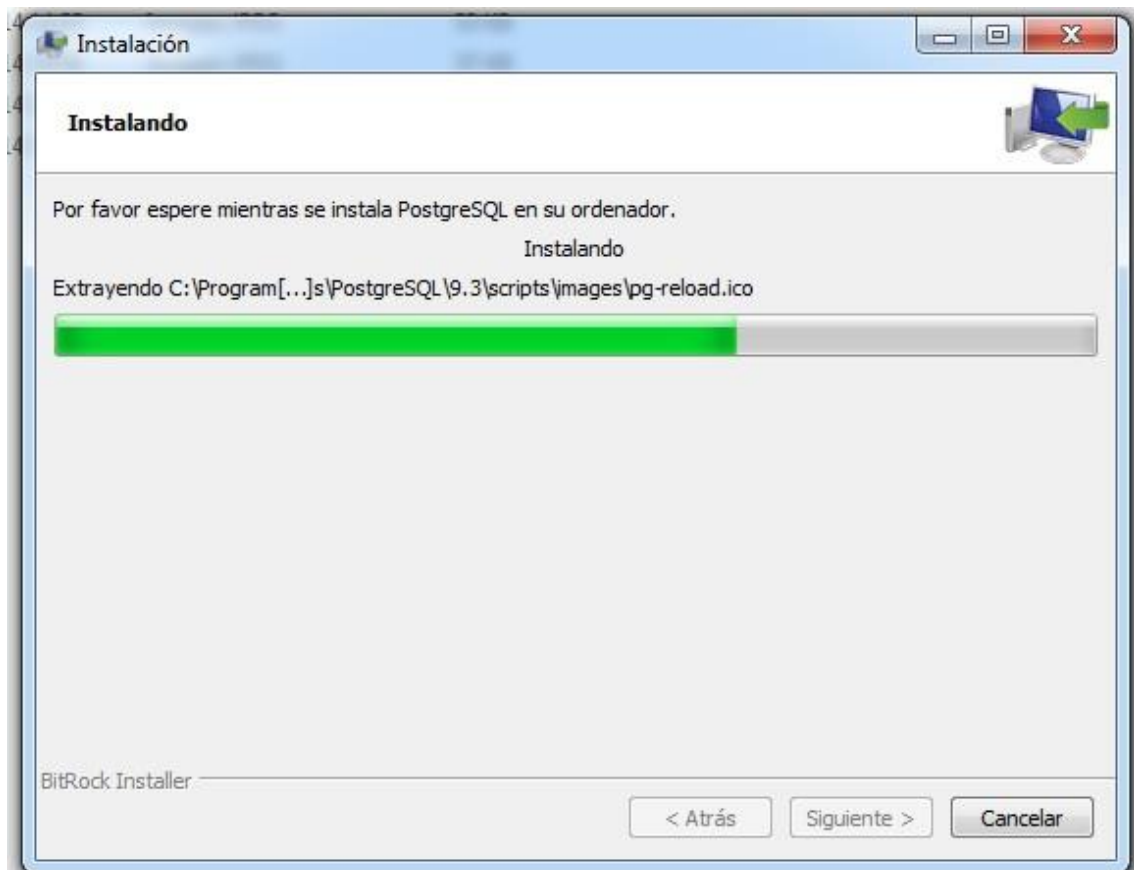
Por favor seleccione un número de puerto en el que el servidor debería escuchar.

Puerto

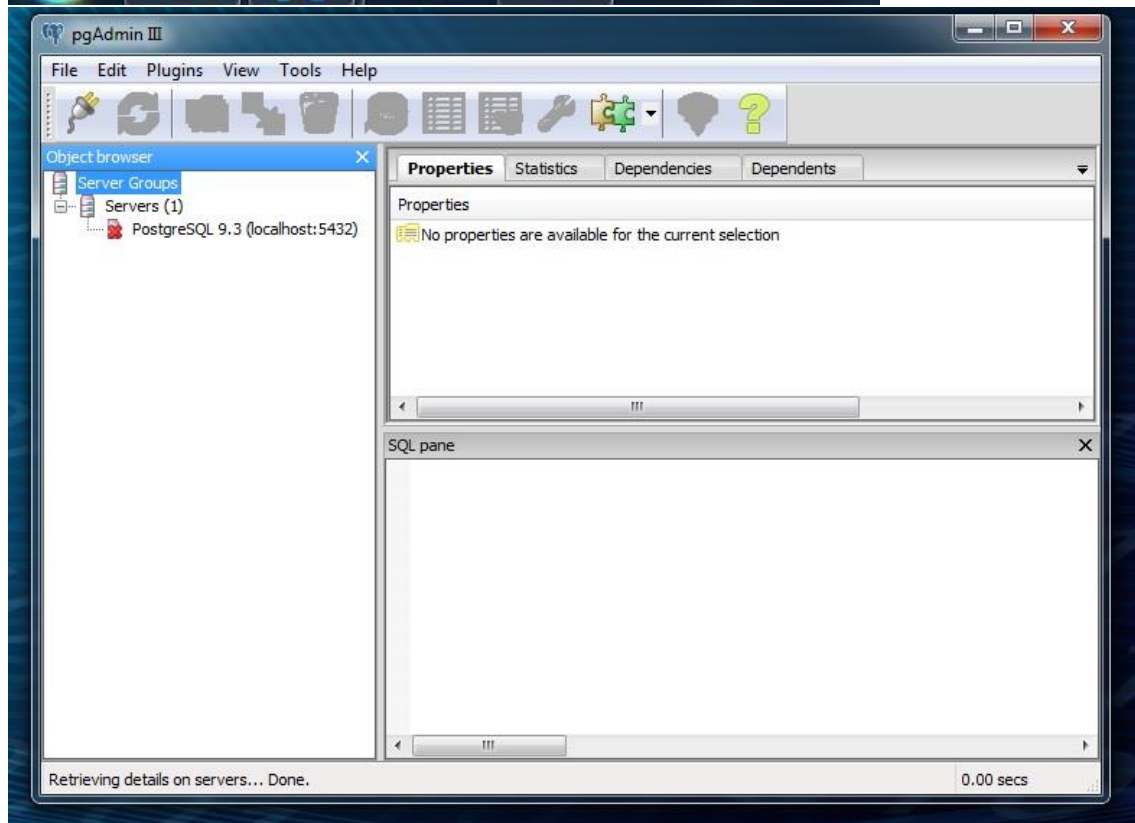
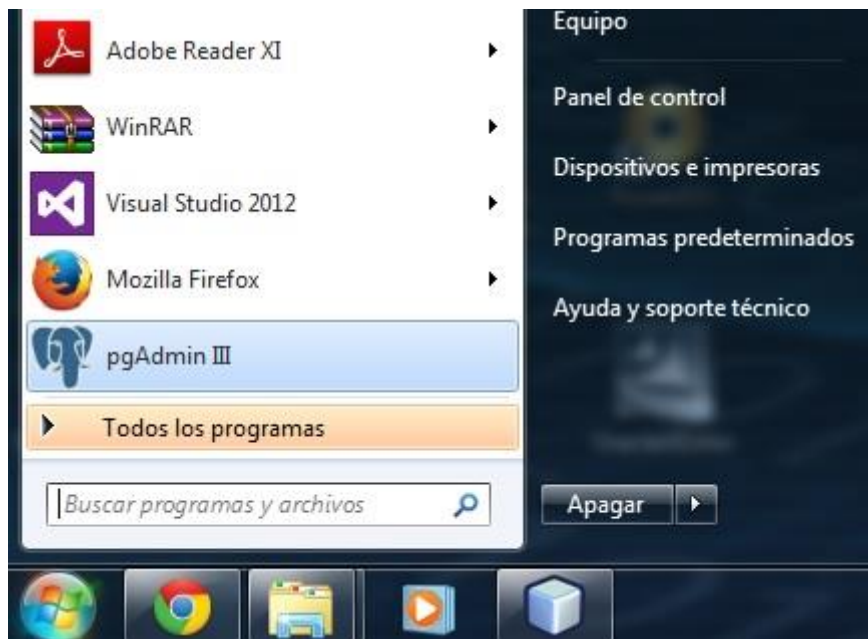
BitRock Installer

< Atrás    Siguiendo >    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: usuario

-- DROP TABLE usuario;

CREATE TABLE usuario
(
    idusuario serial NOT NULL,
    idrol serial NOT NULL,
    username character varying(50) NOT NULL,
    password character varying(50) NOT NULL,
    nombre character varying(50) NOT NULL,
    apellido character varying(50) NOT NULL,
    cedula character varying(15) NOT NULL,
    fechanacimiento date NOT NULL,
    salario double precision NOT NULL,
    sexo boolean NOT NULL,
    estadoborrado boolean NOT NULL,
    enabled boolean NOT NULL,
    CONSTRAINT usuario_pkey PRIMARY KEY (idusuario),
    CONSTRAINT usuario_idrol_fkey FOREIGN KEY (idrol)
        REFERENCES rol (idrol) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT usuario_cedula_key UNIQUE (cedula),
    CONSTRAINT usuario_username_key UNIQUE (username)
);
```

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

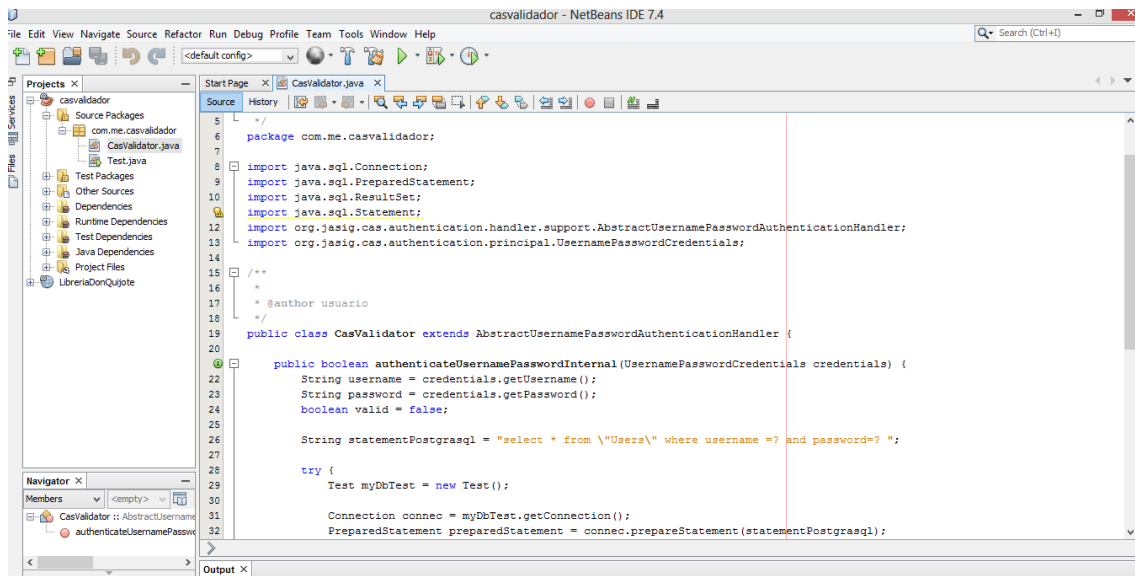
```
--Usuario--
insert into usuario (idrol, username, password, nombre, apellido, cedula, fechanacimiento, salario, sexo, estadoborrado, enabled)
values (1, 'Administrador', 'ACM1234', 'Quijote', 'ACM', '0000000000', '2014-08-29', 0.00, False, False, True); --date se inserta asi
insert into usuario (idrol, username, password, nombre, apellido, cedula, fechanacimiento, salario, sexo, estadoborrado, enabled)
values (1, 'Vendedor', '12345', 'Quijote', 'ACM', '9999999999', '2014-08-29', 0.00, False, False, True); --date se inserta asi año-
```

	idrol serial	username character varying(50)	password character varying(50)	nombre character varying(50)	apellido character varying(50)	cedula character varying(15)	fechanacimiento date	salario double precision	sexo boolean	estadoborrado boolean	enabled boolean
1	1	Administrador	ACM1234	Quijote	ACM	0000000000	2014-08-29	0	FALSE	FALSE	TRUE
2	2	Vendedor	12345	Quijote	ACM	9999999999	2014-08-29	0	FALSE	FALSE	TRUE
*											

El script para creación de tablas y la inserción de datos se adjunta como documento entregable.

## 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=? ";

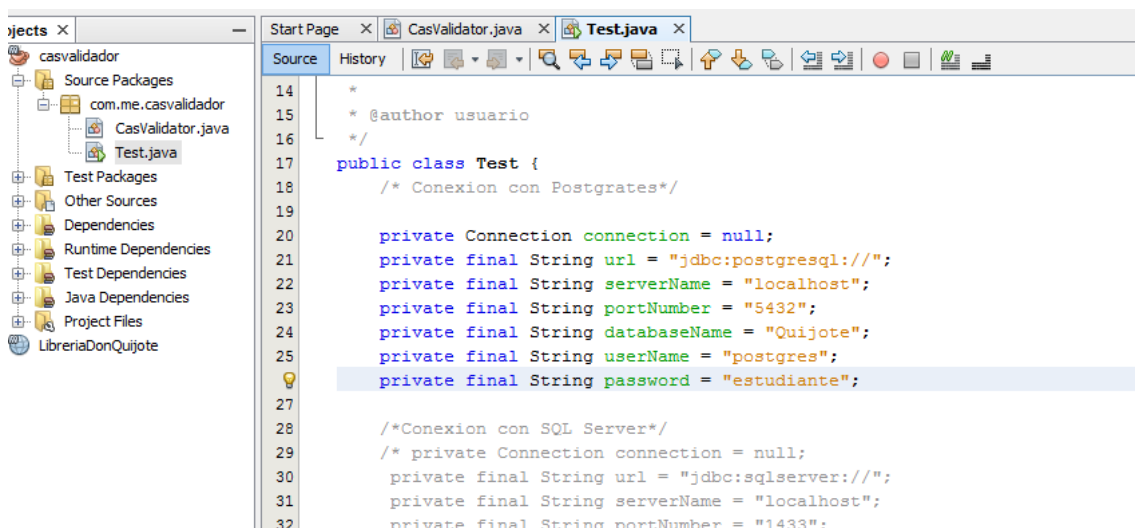
try {
    Test myDbTest = new Test();

    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;
    }
}
```

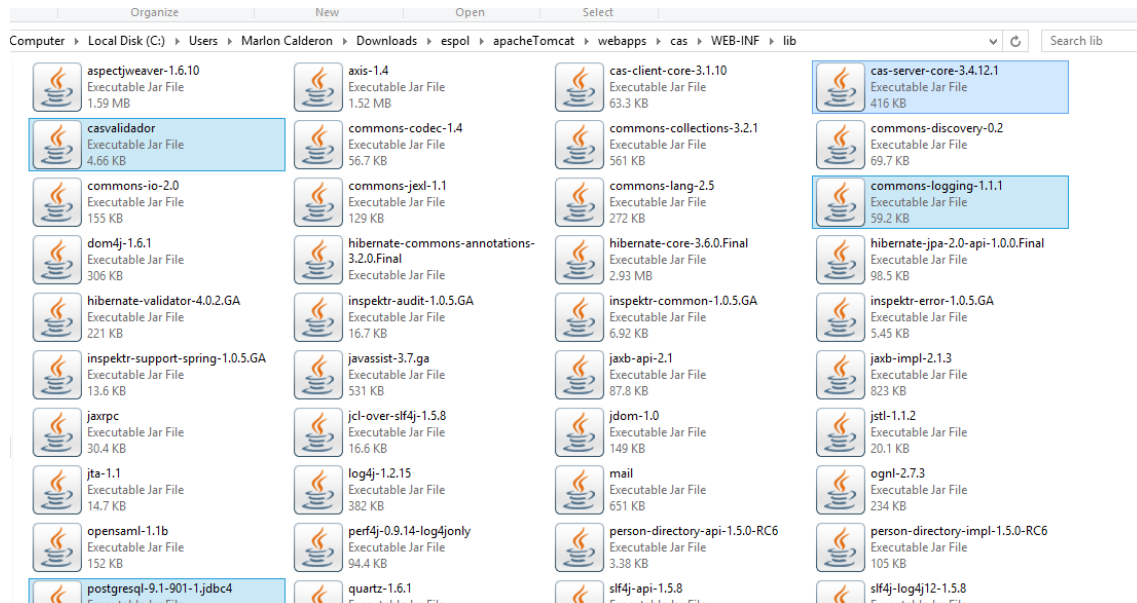
## 3.- Se usa una clase de main, para realizar la conexión.



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

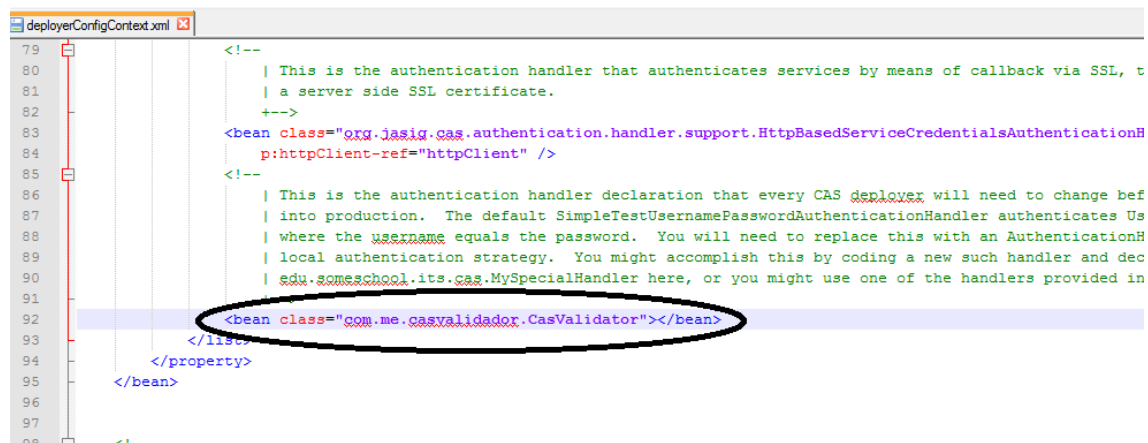


5.- Finalmente este documento se copia junto con otros que son necesarios para CAS, estos son los siguientes: cas-server-core-3.4.12.1, casvalidador.jar, commons-logind-1.1.1, postgresql-9.1.901-1.jdbc4

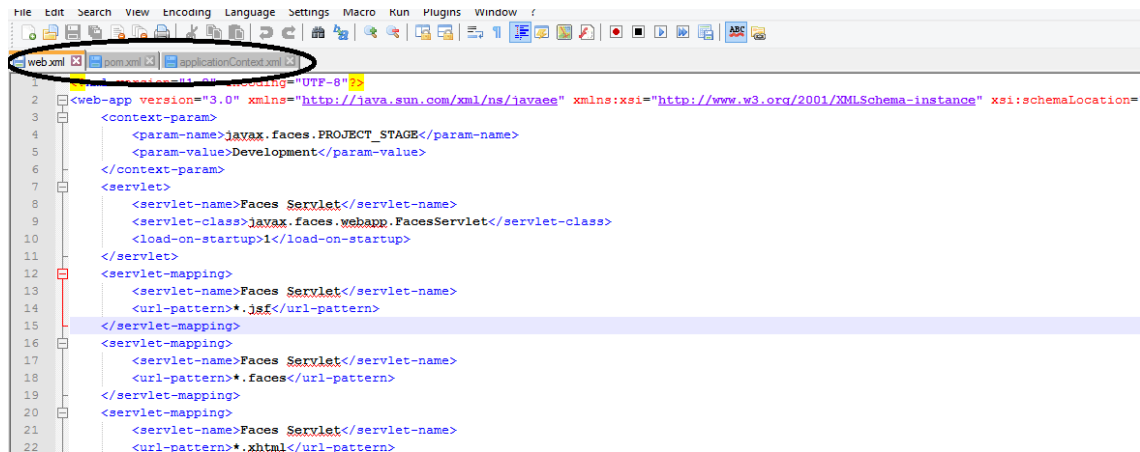


Esto va dentro de la carpeta de cas que está entre las webapps de Apache Tomcat, dentro de la parte de las librerías.

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 casvalidador 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.

```
<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://guijote: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://guijote:8084/pruebaCAS/_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://guijote: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://guijote: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 name="jlmj" password="1234" authorities="ROLE_VENDEDOR"/>
</sec:user-service>

```



## Desplegando en Tomcat

Finalmente se debe modificar un documento que está en la siguiente dirección  
C:\Windows\System32\Drivers\etc\

Se escoge el archivo hosts y se modifica, agregando las líneas:

```

17 #          38.25.63.10      x.acme.com
18
19 # localhost name resolution is handled wit
20 # 127.0.0.1      localhost
21 # ::1           localhost
22
23 127.0.0.1      localhost
24 127.0.0.2      guijote

```

Con esto se permite el uso del certificado guijote.

Entre los entregables se encuentra el cas. Esta carpeta de cas se la copia en la carpeta webapps del tomcat.

Luego se despliega el war utilizando el manager app del tomcat y lo visualizamos así.  
(Para todo esto utilizamos el certificado quijote, es decir, no entramos al tomcat por localhost)

Trayectoria	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/LibreríaDonQuijote	Ninguno especificado		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/cas	Ninguno especificado	Central Authentication System (CAS) 3.4.12.1	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 5 minutos
/docs	Ninguno especificado	Tomcat Documentation	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar > 30 minutos

Finalmente damos click en LibreríaDonQuijote y visualizamos la aplicación.

LIBRERIA DON QUIJOTE

Introduzca su Usuario y Contraseña.

Usuario: Administrador

Contraseña:

☐ Avisarme antes de abrir sesión en otros sitios.

INICIAR SESIÓN limpiar

Por razones de seguridad, por favor cierre su sesión y su navegador web cuando haya terminado de acceder a los servicios que requieren autenticación.

Copyright © 2009 - 2010 JASIG, Inc. All rights reserved.  
Powered by [JASIG Central Authentication Service 3.4.12.1](#)

JASIG

El despliegue se realizó en una computadora que no tenía instalaciones previas, en otras palabras casi nueva.

Con características muy parecidas a la del siguiente enlace:

<http://h10025.www1.hp.com/ewrf/wc/document?cc=es&lc=es&dlc=es&docname=c03517942>

Se siguió el manual omitiendo ciertos pasos propios del proyecto, pues estos vienen incluidos en los archivos entregables que solo se copian en la carpeta webapps, pero de todos modos se detalla cómo se hicieron.