

GUIA DE DESPLIEGUE



INTEGRANTES:

BANCHON Z. JIMMY

REVELO H. VANESSA

SAMANIEGO V. STEPHANY

Tabla de contenido

SERVPET	3
REQUISITOS.....	3
HISTORIAL DE REVISIONES	3
INSTALACION DEL GLASSFISH 4.0 + CAS JASIG + HTTPS	4
CAS JASIG CONFIGURACIÓN PARA POSTGRESQL.....	10
INSTALACION DEL POSTGRESQL	12
POSTGRESQL + SCRIPTS	1
DESPLIEGUE DE SERVPET	3

SERVPET

Es una aplicación integrada con CAS y SPRING para una veterinaria con el propósito de ofrecer un mejor servicio a los clientes.

REQUISITOS

- Application Server Glassfish 4.0
- JDK 7
- PostgreSQL 9.3

HISTORIAL DE REVISIONES

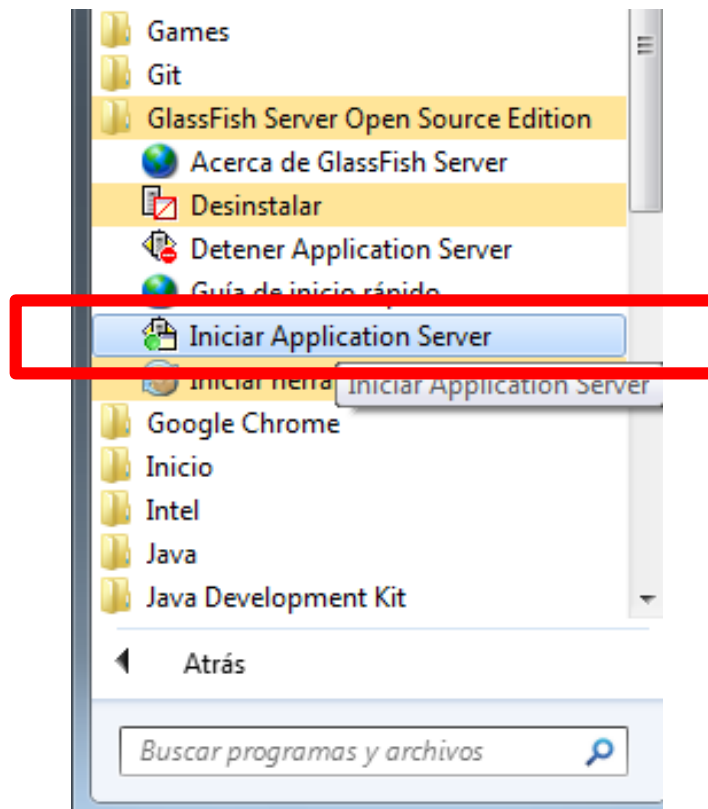
FECHA DE REVISION	TITULO	INTEGRANTES
JUN 1/ 2014	INTEGRACION DEL CAS	BANCHON Z. JIMMY REVELO H. VANESSA SAMANIEGO V. STEPHANY
JUN 9/ 2014	PAGINAS DE APP	
JUL 5/ 2014	INTEGRACION CON SPRING	
JUL 30/ 2014	MODELO ER- CAPA DE PRESENTACION	

INSTALACION DEL GLASSFISH 4.0 + CAS JASIG + HTTPS

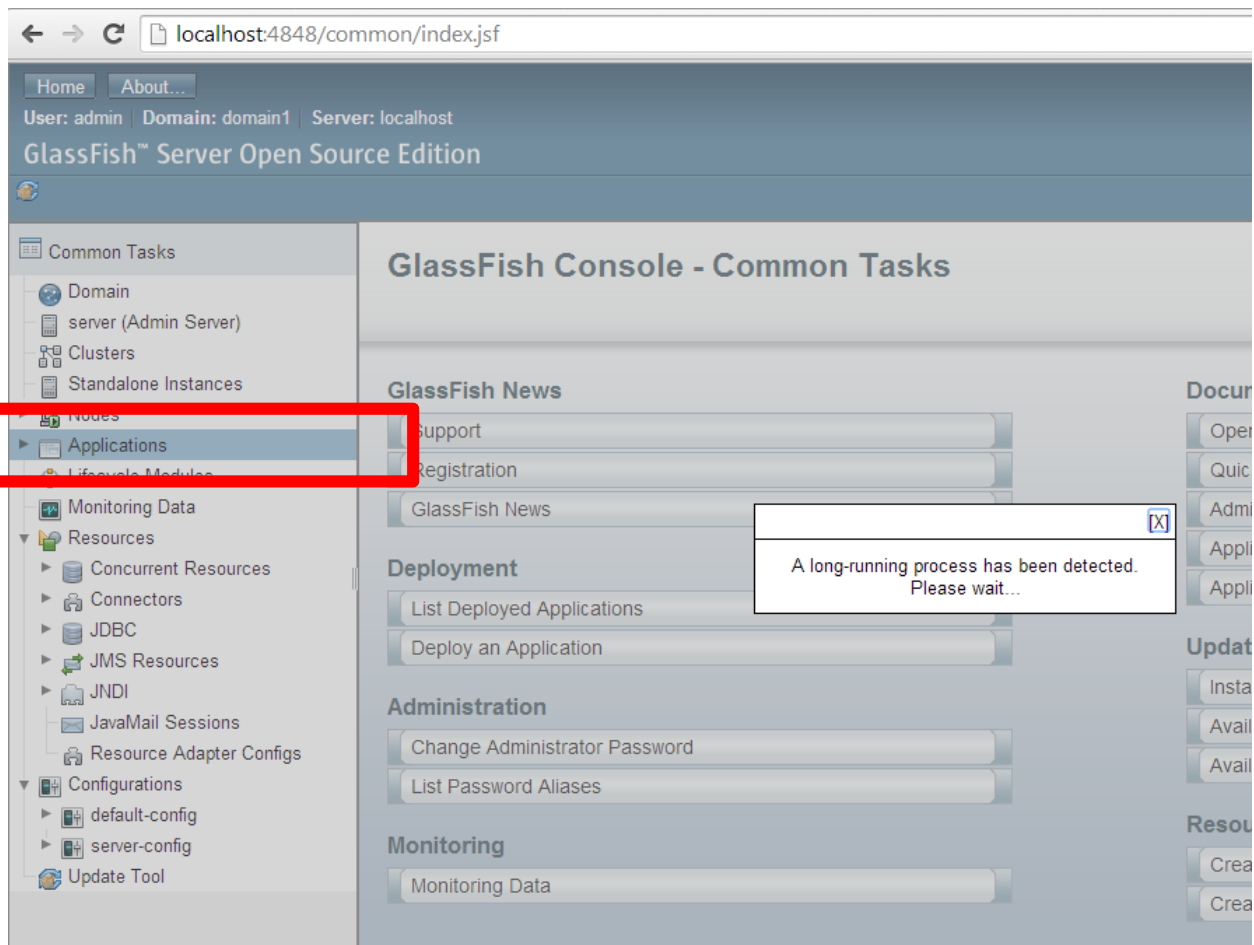
- Nos dirigimos a la página oficial del servidor descargas
 - Se escoge NATIVE INSTALLER y se inicia la descarga



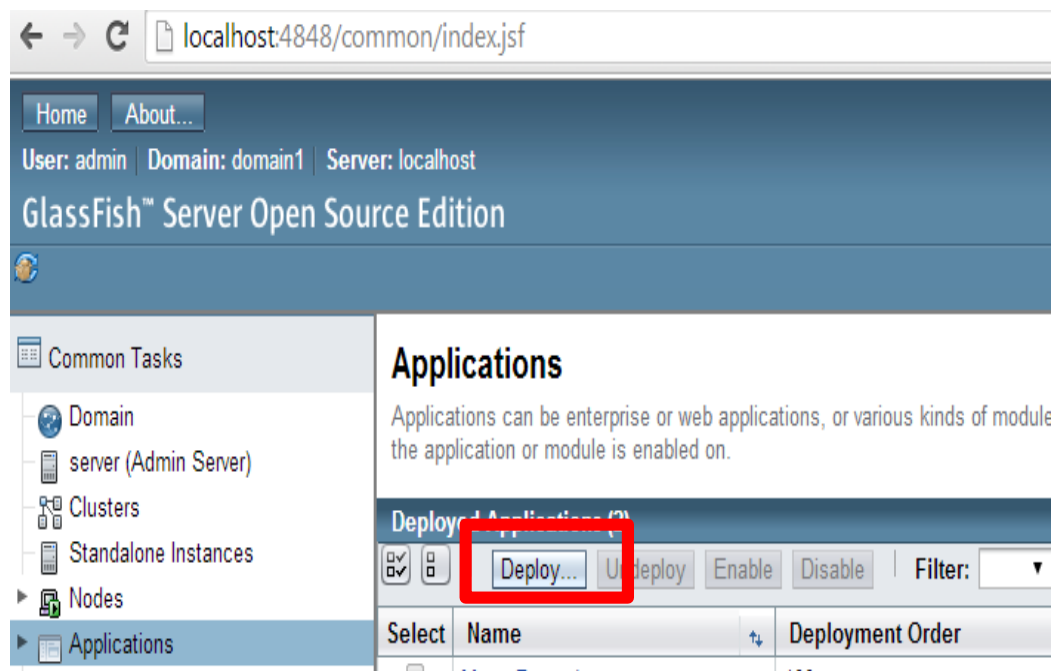
- Una vez que ya se tiene el CAS-SERVER-3.5.2 ya descargado se realizará lo siguiente:
 - Desplegar Cas Jasig en nuestro Servidor
 - Primero se irá a desplegar él .war del proyecto *Cas Jasig* en el *glassfish*
 - Para ello se inicia el servidor.



- Se puede ir a <http://localhost:8080/> o directamente a <http://localhost:4848/> -> se escoge la opción Applications

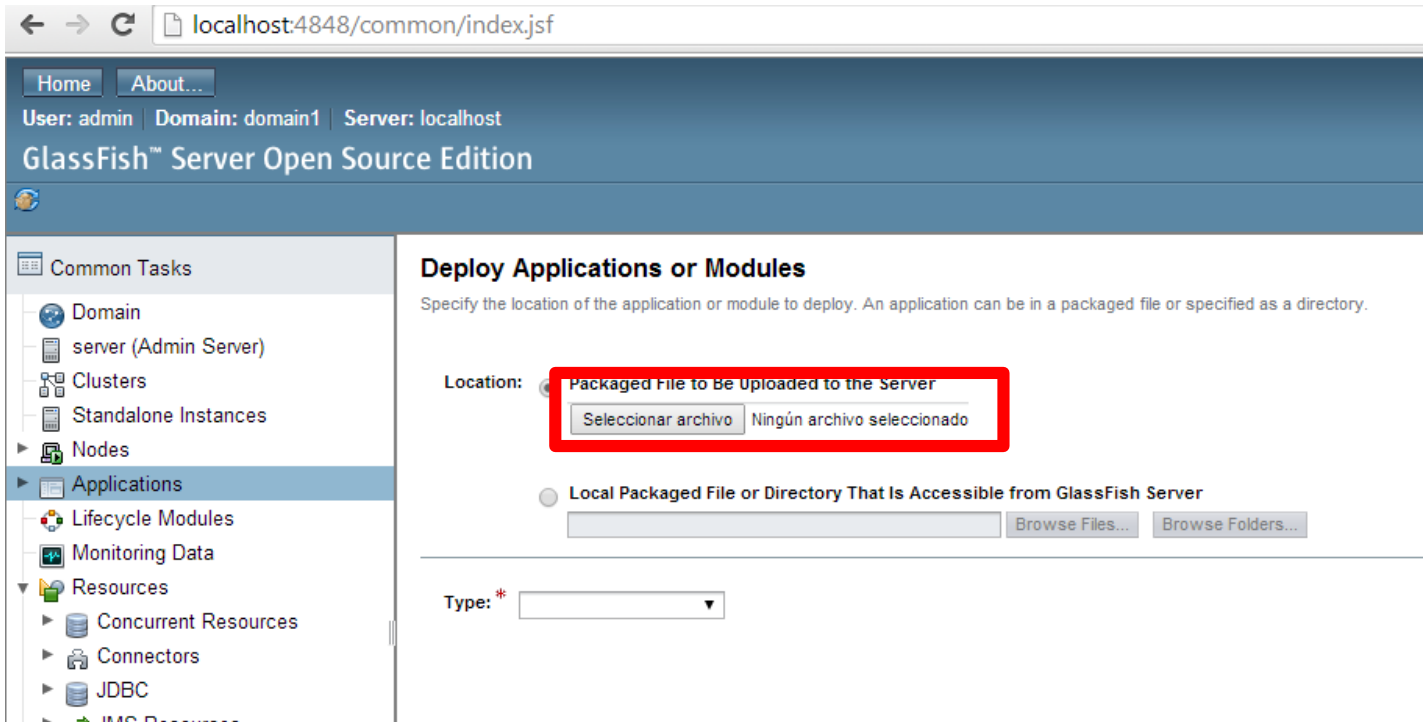


- Se escoge *deploy*



○ Buscas el .war de la aplicación de *Cas Jasig* generalmente se encuentra en la siguiente ruta :

- ...cas-server-3.5.2-release\cas-server-3.5.2\modules\cas-server- webapp-3.5.2.war **[puede cambiar el nombre del .war si lo desea]**



← → ↻ localhost:4848/common/index.jsf

Home About...

User: admin | Domain: domain1 | Server: localhost

GlassFish™ Server Open Source Edition

Common Tasks

- Domain
 - server (Admin Server)
 - Clusters
 - Standalone Instances
 - Nodes
 - Applications**
 - Lifecycle Modules
 - Monitoring Data
 - Resources
 - Concurrent Resources
 - Connectors
 - JDBC
 - IMS Resources

Deploy Applications or Modules

Specify the location of the application or module to deploy. An application can be in a packaged file or specified as a directory.

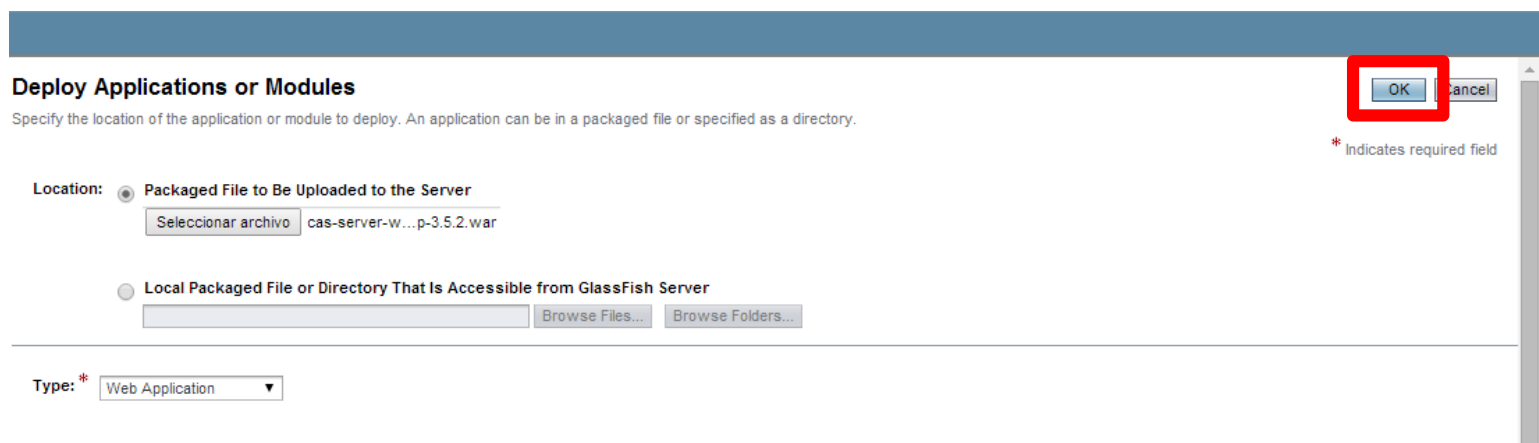
Location: ☒ **Packaged File to Be Uploaded to the Server**

Ningún archivo seleccionado

☐ Local Packaged File or Directory That Is Accessible from GlassFish Server

Type: *

- Luego da clic en OK



Deploy Applications or Modules

Specify the location of the application or module to deploy. An application can be in a packaged file or specified as a directory.

Location: ☒ **Packaged File to Be Uploaded to the Server**

cas-server-w...p-3.5.2.war

☐ Local Packaged File or Directory That Is Accessible from GlassFish Server

Type: *

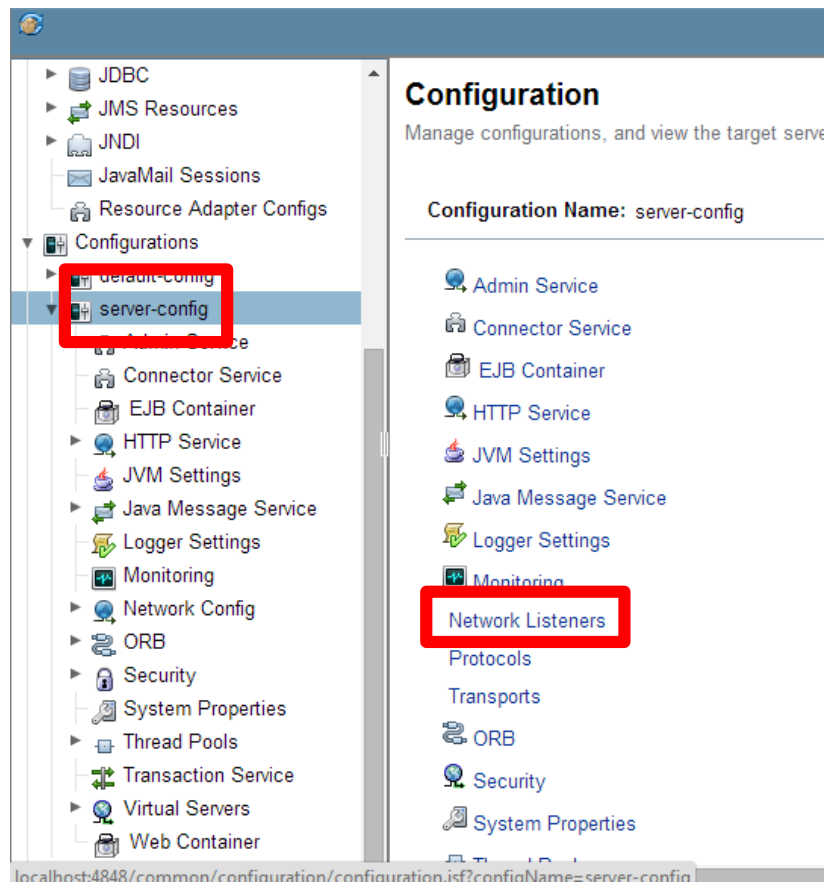
* Indicates required field

- Teniendo como resultado lo siguiente:

<input type="checkbox"/>	cas-server-webapp-3.5.2	100	<input checked="" type="checkbox"/>	web	Launch Redeploy Reload
--------------------------	-------------------------	-----	-------------------------------------	-----	--

- Ahora para realizar el certificado y dar una conexión segura se procede a realizar lo siguiente:

- Se elegirá *Server-Config*; Seescope *Network Listeners*



- Luego *http-listener-2*



- Se deberá colocar el puerto 8443

General	SSL	HTTP	File Cache
---------	-----	------	------------

Edit Network Listener

Modify an existing network listener.

[Load Defaults](#)

Configuration Name: server-config

Name: http-listener-2
Protocol: http-listener-2
Status: ☒ Enabled
Security: ☒ Enabled
JK Listener: ☐ Enabled
If selected, listener is an Apache mod_jk listener.

Port: *

- En la pestaña SSL se habilita la *casilla* SSL3

General	SSL	HTTP	File Cache
---------	-----	------	------------

SSL

Modify SSL settings.

Configuration Name: server-config

SSL: ☒ Enabled
TLS: ☒ Enabled
Client Authentication: ☐ Enabled
Requires the client to authenticate itself to the server.

Certificate NickName: *
Takes a single value, identifies the server's keypair and certificate.

Key Store:
Name of the keystore file (for example, keystore.jks)

Trust Algorithm:
Name of the trust management algorithm (for example, PKIX) to use

Max Certificate Length:
Maximum number of non-self-issued intermediate certificates that c

Luego de seguir los pasos anteriores, se dirige de nuevo a *Applications* y se da clic en launch; saldrán dos links, se elegirán el segundo link que tiene ya conexión segura que al cambiar el puerto glassfish genera un certificado por defecto con nuestro usuario de administrador



Web Application Links

If the server or listener is not running, the link may not work. In this event, check the status of the se

Application Name: cas-server-webapp-3.5.2

Links:
[server] <http://gianella09:8080/cas-server-webapp-3.5.2>
[server] <https://gianella09:8443/cas-server-webapp-3.5.2>

Obteniendo la siguiente página

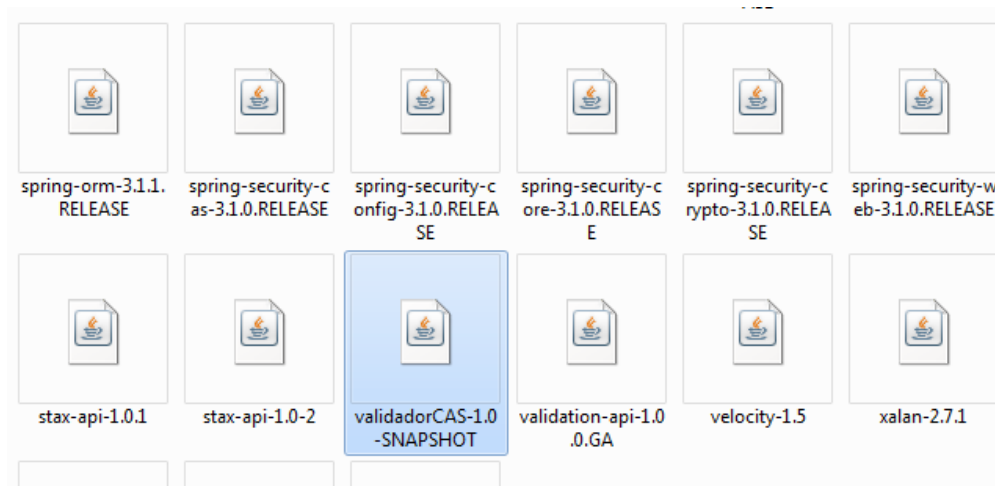
A screenshot of a web browser displaying the JASIG Central Authentication Service (CAS) login page. The browser's address bar shows a URL with a red 'x' icon and the text 'https://gianella09:8443/cas-server-webapp-3.5.2/login'. The page features the JASIG logo at the top left. Below it is a dark blue header with the text 'Central Authentication Service (CAS)'. The main content area has a light gray background and contains a login form. The form has the heading 'Introduzca su NetID y Contraseña.' and two input fields: 'NetID:' and 'Contraseña:'. Below the password field is a checkbox labeled 'Avisarme antes de abrir sesión en otros sitios.' At the bottom of the form are two buttons: 'INICIAR SESIÓN' and 'limpiar'. To the right of the form, there is a section titled 'Languages:' with a grid of links for English, Spanish, French, Russian, Ne, Croatian, Czech, Slovenian, and Catalan. At the bottom of the page, there is a footer with the text 'Copyright © 2005 - 2012 Jasig, Inc. All rights reserved. Powered by [Jasig Central Authentication Service 3.5.2](#)'.

CAS JASIG CONFIGURACIÓN PARA POSTGRESQL

- Dirigiendonos a la siguiente ruta:

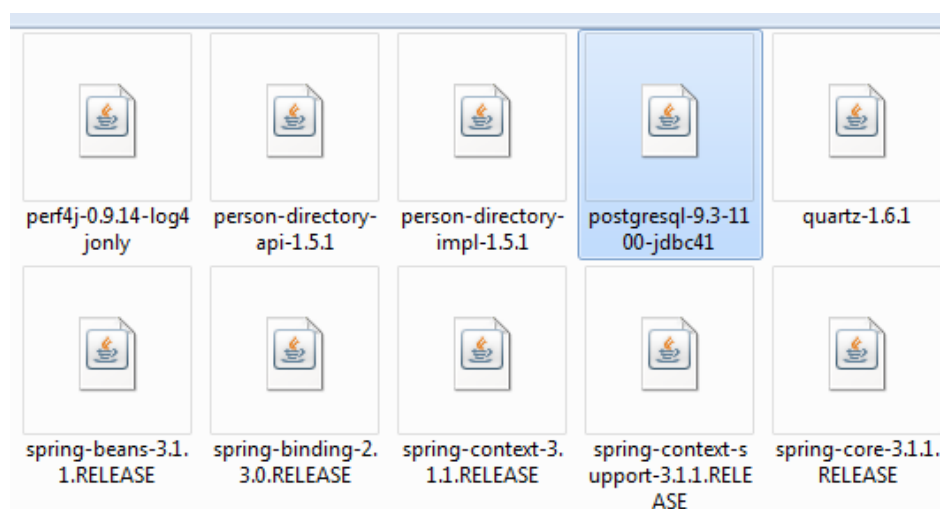
`C:\glassfish4\glassfish\domains\domain1\applications\cas-server-webapp-3.5.2\WEB-INF\lib`

❖ Se tendrá que colocar el jar de la misma aplicacion en la ruta indicada:



*Cuyo jar se encuentra en **Servpet\MavenPet\conexion\target**

❖ Además se deberá colocar el jar de postgresQL 9.3



- ❖ Otra configuración a realizar se debe ir al `deployerConfigContext.xml`, ubicado en la siguiente ruta:

C:\glassfish4\glassfish\domains\domain1\applications\cas-server-webapp-3.5.2\WEB-INF

Nombre	Fecha de modifica...	Tipo	Tamaño
classes	08/08/2014 11:14	Carpeta de archivos	
lib	09/08/2014 9:50	Carpeta de archivos	
spring-configuration	08/08/2014 11:14	Carpeta de archivos	
unused-spring-configuration	08/08/2014 11:14	Carpeta de archivos	
view	08/08/2014 11:14	Carpeta de archivos	
cas.properties	08/08/2014 11:14	Archivo PROPERTI...	5 KB
cas-servlet	08/08/2014 11:14	Documento XML	13 KB
deployerConfigContext	09/08/2014 9:58	Documento XML	12 KB
login-webflow	08/08/2014 11:14	Documento XML	10 KB
restlet-servlet	08/08/2014 11:14	Documento XML	3 KB
web	08/08/2014 11:14	Documento XML	8 KB

Por último se coloca la siguiente línea

```

<!--
<property name="authenticationHandlers">
  <list>
    <!--
      | This is the authentication handler that authenticates services by means of callback via SSL, thereby v
      | a server side SSL certificate.
    <!-->
    <bean class="org.jasig.cas.authentication.handler.support.HttpBasedServiceCredentialsAuthenticationHandler"
      p:httpClient-ref="httpClient" />
    <!--
      | This is the authentication handler declaration that every CAS deployer will need to change before depl
      | into production. The default SimpleTestUsernamePasswordAuthenticationHandler authenticates UsernamePa
      | where the username equals the password. You will need to replace this with an AuthenticationHandler t
      | local authentication strategy. You might accomplish this by coding a new such handler and declaring
      | edu.somewhere.its.cas.MySpecialHandler here, or you might use one of the handlers provided in the ada
    <!-->
    <bean class="com.software.validadorcas.conexion.Validador" />
  </list>
</property>

```

INSTALACION DEL POSTGRESQL

- En la página oficial de postgresQL nos descargamos el instalador para nuestro Sistema Operativo Windows.

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

Products	Services	Training	Solutions	Success Stories	Resources	Partner Programs	About Us
----------	----------	----------	-----------	-----------------	-----------	------------------	----------






You are here: [Home](#) / [Products](#) / [PostgreSQL Overview](#) / Download PostgreSQL

[Postgres Plus Advanced Server](#)
[PostgreSQL Overview](#)
[Download PostgreSQL](#)
[Supported Platforms and Database Interfaces](#)

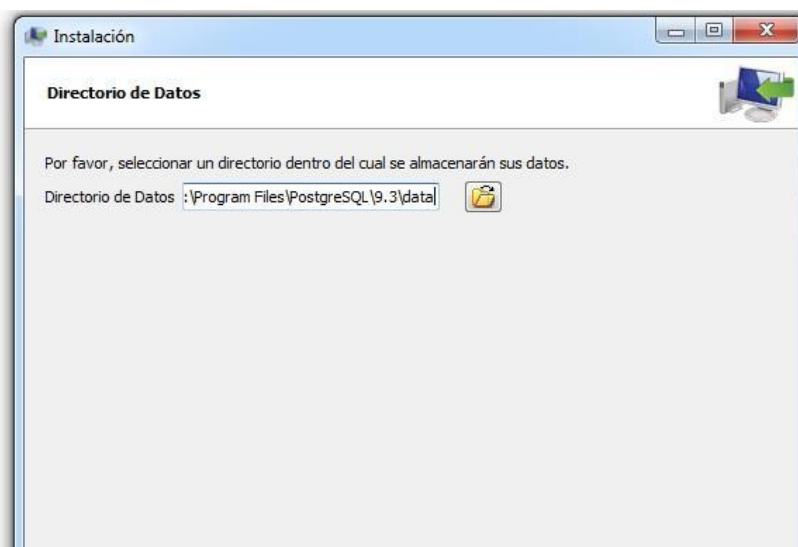
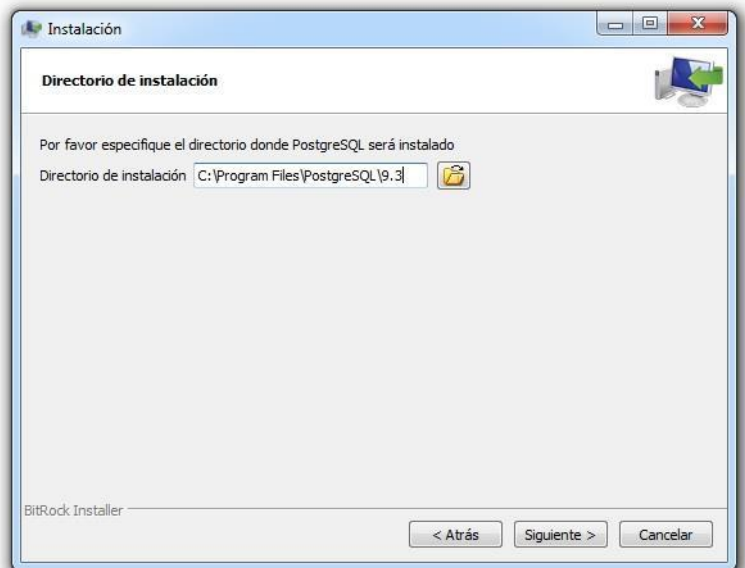
Download PostgreSQL

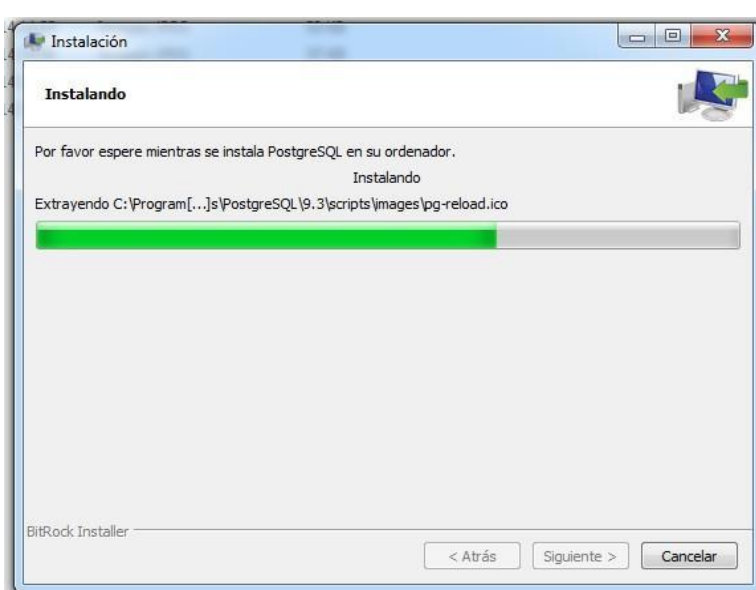
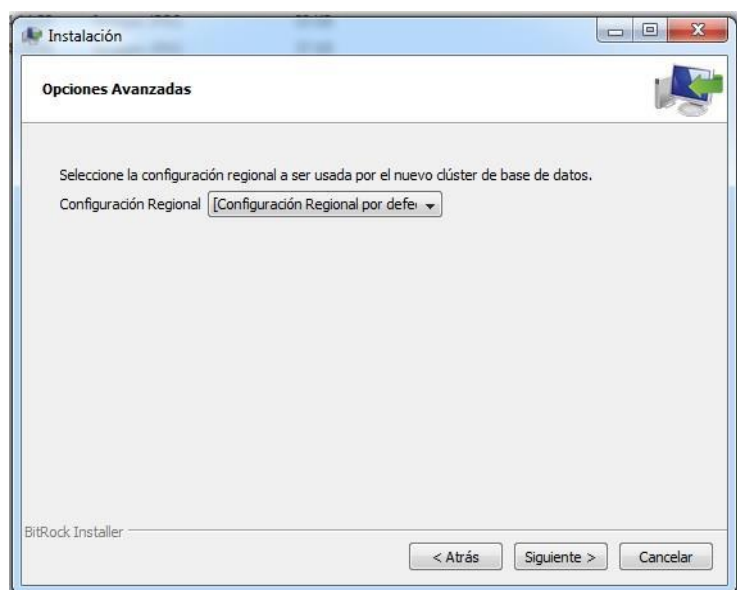
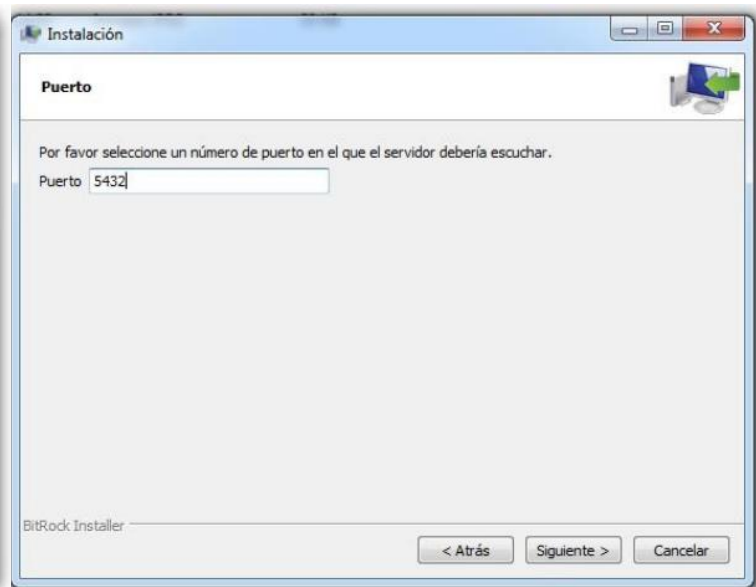
Please Note: Cookies should be enabled for the download process to function correctly

Installer version **Version 9.3.4**

 **Linux x86-32**  **Linux x86-64**  **Win x86-32**  **Win x86-64**  **Mac OS X**

- Una vez descargado empezaremos con la instalación





POSTGRESQL + SCRIPTS

- Se creara la Base con el siguiente nombre “veterinaria”

```
-- DROP DATABASE veterinaria;

CREATE DATABASE veterinaria
WITH OWNER = postgres
ENCODING = 'UTF8'
TABLESPACE = pg_default
LC_COLLATE = 'Spanish_Spain.1252'
LC_CTYPE = 'Spanish_Spain.1252'
CONNECTION LIMIT = -1;
```

- Ahora para la creación de cada tabla se procederá a coger los scripts iniciando con la tabla ROL

```
-- Table: rol

-- DROP TABLE rol;

CREATE TABLE rol
(
    idrol serial NOT NULL,
    rol text,
    CONSTRAINT rol_pkey PRIMARY KEY (idrol)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE rol
OWNER TO postgres;
```

- ❖ Luego dando los siguientes valores

```
--
-- TOC entry 2030 (class 0 OID 25360)
-- Dependencies: 180
-- Data for Name: rol; Type: TABLE DATA; Schema: public; Owner: postgres
--

COPY rol (idrol, rol) FROM stdin;
1  ROLE_ADMIN
2  ROLE_DOCTOR
3  ROLE_EMPLOYEE
4  ROLE_CLIENT
\.
```

- Ahora para la Tabla usuario que es la que necesitamos para iniciar sesión en nuestro APP

```
-- Table: usuario

-- DROP TABLE usuario;

CREATE TABLE usuario
(
    cedula text NOT NULL,
    nombre text,
    apellido text,
    email text,
    password text,
    direccion text,
    idrol integer,
    CONSTRAINT usuario_pkey PRIMARY KEY (cedula),
    CONSTRAINT usuario_idrol_fkey FOREIGN KEY (idrol)
        REFERENCES rol (idrol) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE usuario
    OWNER TO postgres;
```

❖ Teniendo los siguientes datos

```
--
-- TOC entry 2034 (class 0 OID 25376)
-- Dependencies: 184
-- Data for Name: usuario; Type: TABLE DATA; Schema: public; Owner: postgres
--

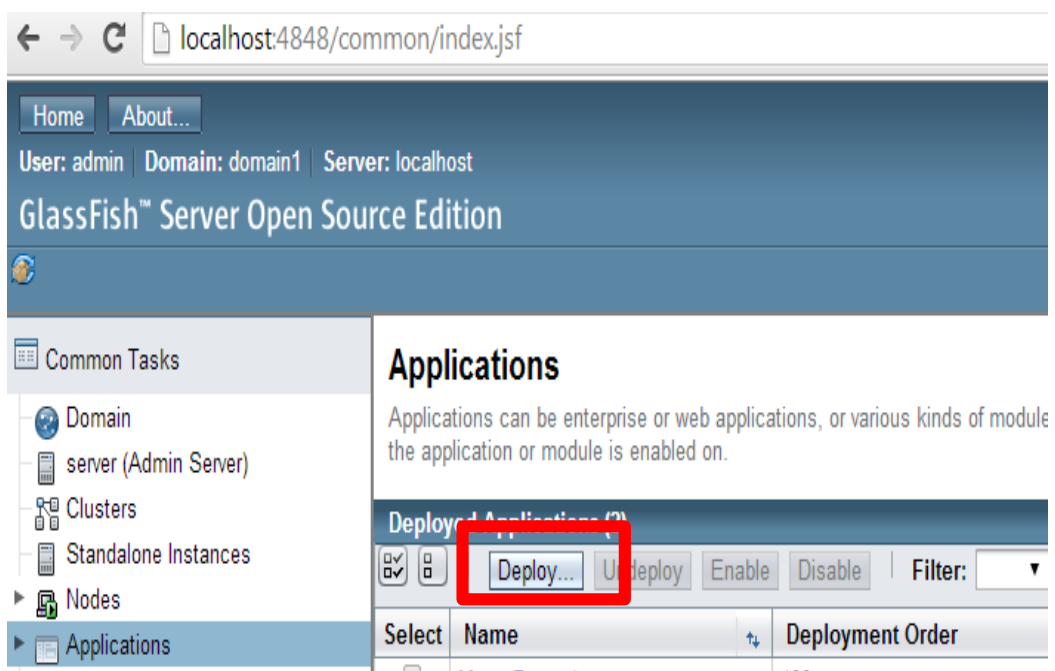
COPY usuario (cedula, nombre, apellido, email, password, direccion, idrol) FROM stdin;
920818184    cristhian    ponguillo    cjpongui@yahoo.com    cjpongui    Acacias    2
923468780    jimmy        banchon     jialbanc@hotmail.com    jial        Alborada    4
930561022    vanessa      revelo       vanessarevelo@ymail.com    vane        Cdls.Orquideas    1
950425677    stephany     samanieho    sksamani@gmail.com    samani      Villa Club    3
0912345689    carolina     morales     caro@morales.com    1239767337    ciudadela    orquideas    2
\
```

TENIENDO EN TOTAL LAS SIGUIENTES TABLAS

- CITAS
- SERVICIO
- USUARIO
- ROL
- CONTACTENOS
- DETALLE_CITA
- MASCOTAS
- NOTIFICACIONES
- USUARIO_HAS_CITAS

DESPLIEGUE DE SERVPET

- Ya habiendo hecho los pasos anteriores podrá desplegar nuestra aplicación sin ningún inconveniente
- Se inicia el servidor y se procede a subir el .war de ServPet y se sigue el mismo procedimiento que se indicó con el CAS



- Se escoge el archivo ubicado en la siguiente ruta:

Servpet\MavenPet\MavenPet-web\target

surefire	28/08/2014 15:13	Carpeta de archivos	
surefire-reports	27/08/2014 21:25	Carpeta de archivos	
test-classes	29/08/2014 13:28	Carpeta de archivos	
MavenPet-web-1.0-SNAPSHOT.war	28/08/2014 15:13	Archivo WAR	14.055 KB

- Teniendo como resultado ya subido el .war de SERVPET

[MavenPet-web](#)

[Launch](#) | [Redeploy](#) | [Reload](#)

- Escogiendo *Launch*; y se escoge el siguiente link

Web Application Links

If the server or listener is not running, the link may not work. In this event, check th

Application Name: MavenPet-web

Links:

[server] <http://gianella-PC:8080/MavenPet-web>

[server] <https://gianella-PC:8443/MavenPet-web>

- Y como resultado se obtiene la aplicación desplegada y podrá hacer uso de ella en este caso sale con un diseño propio de la app que fue modificada para su presentación dentro del CAS.



NOTA: UD PUEDE MODIFICAR EL CAS COLOCANDO SU PROPIO DISEÑO MODIFICANDO LOS CSS DEL PROPIO CAS INSTALADO EN NUESTRO APP SERVER