

# SWII\_Grupo\_A

## Proyecto de Ingenieria de Software II - Grupo A

### GUIA DE DESPLIEGUE

---

Para que la aplicacion SysMedic pueda ser desplegada se necesita realizar algunos pasos previos considerando la configuracion de la base de datos a utilizar, CAS y la aplicacion misma, primero describiremos la estructura del repositorio en GIT ubicado en [https://github.com/pittersdb/SWII\\_Grupo\\_A](https://github.com/pittersdb/SWII_Grupo_A).

### Organizacion del Repositorio en GIT

El repositorio esta organizado en cuatro carpetas principales:

#### Components

Aqui se guardaran los componentes a utilizar en el proyecto ya sean compilados o en codigo fuente.

#### Databases

Informacion acerca de las bases de datos a utilizar en el proyecto, generalmente scripts para construirlas, queries y datos de prueba.

#### Project

Aqui reside el proyecto SysMedic que esta siendo trabajado en Netbeans y tiene una estructura Maven.

#### Tests

Aqui se encuentran proyectos de prueba en donde se realizan mitigaciones de tecnologias o estrategias que se utilizaran en el proyecto y de las cuales anteriormente no se tenia conocimiento.

### Intrucciones de despliegue

Las herramientas de desarrollo que se estan utilizando son:

- JDK 1.7

- Netbeans 8.0
- Glassfish 4.0 (asegurarse que en la instalacion se incluya el driver JDBC para postgres 9.3)
- PostgreSQL 9.3
- Maven 1.0.1 (incluido en Netbeans 8.0)

**\*\*Importante: \*\*** Con JDK 1.8 existen errores en CAS cuando se construye (build), es por ello que se opto por 1.7

1. Crear la base de datos, pra lo cual se tendran que seguir los siguientes pasos, de preferencia utilizar Pg Admin para generar la base:

1.1 En Pg Admin crear un usuario con todos los privilegios llamado "sysmedic" con password "sysmedic2014".

1.2 Crear una base de datos llamada "SysMedic" (debe llevar las mayusculas como se muestran).

1.3 Ir a la ubicacion "C:\Program Files\PostgreSQL\9.3\bin" en su carpeta de instalacion de postgres y abra alli una ventana de comandos con shift sostenido + click derecho y escogiendo "Abrir ventana de comandos aqui".

1.4 En la ventana de comandos escribimos el siguiente comando y lo ejecutamos:

```
psql -U sysmedic -d SysMedic < "D:\Ingenieria de Software II
Repo\SWII_Grupo_A\Databases\sysmedic_database.sql"
```

Considere que el path de ejemplo es en donde reside el archivo sysmedic\_database.sql que contiene los comandos necesarios y los datos para crear la base.

1.5 Desde Pg Admin ahora se refleja el nuevo schema en la base de datos SysMedic y ya puede ser utilizada por la aplicacion.

2. Crear el JDBC para la aplicacion, para ello se debe crear primero un pool de conexiones JDB y luego un recurso de JDBC. Dentro de la carpeta "Databases" en este repositorio hay otra carpeta llamada "Creacion del JDBC", las imagenes Pool1.png y Pool2.png indican los datos basicos para crear el pool, tomar en cuenta donde reside el servicio de postgres en su maquina (puertos, url, etc). Luego como se ve en la imagen CreaciondelJDBC.png enlazar el pool con un nuevo recurso JDBC llamado jdbc/SysMedic, es importante que el pool y el recurso tengan los nombres que la imagen indican porque en el proyecto se las referencia utilizandolos.

Para probar la conexión se puede presionar Ping en el pool creado, si todo está bien podemos continuar al siguiente paso.

3. Abrir CAS desde netbeans, este se encuentra en Components y es llamado cas-server-webapp, si la DB postgres a utilizar no tiene las configuraciones by default cuando se instala, entonces abrir el archivo `deployerConfigContext.xml` que se encuentra en "Web Pages/WEB-INF", ubicarse en el tag bean con `id = "DataSource"` y cambiar el url con el que corresponde a su maquina (el url por default contiene `localhost:5432`).
4. Desplegar CAS ejecutandolo desde Netbeans, se abra el explorador con la aplicacion si todo fue exitoso. Tomar en cuenta que CAS debe ser desplegado primero antes que la aplicacion sysmedic de la cual iniciaremos en el siguiente paso.
5. Abrir el proyecto SysMedic en Netbeans que se encuentra en la carpeta "Project".
6. Si es la primera vez que se baja el repositorio hay que agregar un archivo de configuracion que es ignorado por git con el objetivo de que funcione para varias estaciones, entonces, crear una carpeta llamada "classes" dentro de la carpeta "WEB-INF" del proyecto Sysmedic.
7. Dentro de la ya creada carpeta classes crear un archivo `.properties` llamado "config.properties", este se puede crear utilizando las funcionalidades de Netbeans.
8. Agregar al archivo las siguientes lineas:

```
server.https.name = https://localhost:8181
server.database.name = localhost:5432
server.config.file =
C:/Users/LUCAS/AppData/Roaming/NetBeans/8.0/config/GF_4.0/domain1/config/domain.xml
```

Sin embargo, considerar que este URL de "server.https.name" debe ser donde reside el servicio https en su servidor Glassfish ya que puede tener otro puerto asociado distinto al mostrado en este ejemplo. Lo mismo para "server.database.name" donde reside el servicio de postgres de donde se va a acceder a la base de datos, debería ser el mismo que se cambio en el paso 3.

El parametro "server.config.file" es el lugar donde reside el archivo domain.xml que usa Netbeans para la configuracion de Glassfish y sirve para ser especificado en los tests de EJBs, es IMPORTANTE que este path este con slash (/) y NO con backslash (\) para que sea tomado correctamente en los tests, ademas este path no necesariamente es el mismo archivo que se encuentra en la carpeta de instalacion de Glassfish y eso se puede comprobar si vamos en Netbeans al menu Tools -> Servers y existe un parametro llamado "Domains Folder", si ese es el caso es alli donde esta el domain.xml que se esta usando, caso contrario si este parametro no se encuentra y solo esta la referencia a la instalacion podemos estar seguros que el archivo de configuracion que se esta usando es el de la carpeta de instalacion de Glassfish.

Las primeras dos propiedades son utilizadas en los archivos de configuracion de Spring Security, por lo tanto al cambiarlas se esta configurando indirectamente a Spring que ya tiene integrada su comunicacion con CAS. El siguiente paso es importante.

9. NO ejecutar inmediatamente, primero hay que construir el proyecto, para este caso click derecho en el proyecto SysMedic -> Build esto es para que server.https.name y server.database.name sean tomados en cuenta en los archivos de configuraciones respectivos ya que un feature de Maven permite esto.
10. Ahora el proyecto ya puede ser desplegado y ejecutado, se abra el explorador mostrando el servicio de autentificacion, ahora se puede ingresar a la aplicacion con los datos de prueba de la base de datos. Puede ingresar con usuario "admin\_user" y contraseña "admin\_password" para probar la aplicacion como administrador.