

MANUAL DE CONFIGURACIÓN DEL SERVIDOR GLASSFISH

Programa de Doctorado en Ciencias de la Electrónica



**Universidad
del Cauca**

Integrantes:

Anderson Buitrón.

Santiago García.

Carolina Muñoz.

Danilo López.

Santiago López.

Santiago Pérez.

Paola Pino.

William Romero.

Fainy Rosero.

Facultad de Ingeniería Electrónica y Telecomunicaciones

Ingeniería de Sistemas

2017-2

Introducción

El presente manual, está diseñado con el propósito de brindarle al usuario una clara explicación mediante pasos sobre cómo debe realizar la configuración del servidor Glassfish, una vez configurado el servidor, el usuario podrá iniciar el Programa de Doctorados en Ciencias de la Electrónica.

NOTA: Antes de realizar la configuración del servidor Glassfish, primero debe leer detenidamente y realizar los pasos que se indican en el manual de configuración de la base de datos.

1. JDBC.

En primera instancia se debe iniciar el entorno de desarrollo Netbeans 8.1 (Figura 1).

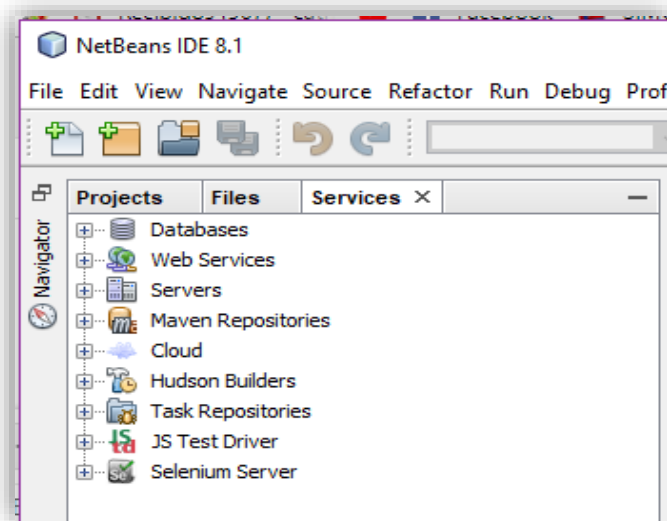


Figura 1. Netbeans 8.1.

Una vez se ha iniciado Netbeans 8.1, nos dirigimos a la pestaña **Services**, entre las opciones presentadas en la sección, se busca el ítem **Servers**, y se despliega para visualizar su contenido (Figura 2).

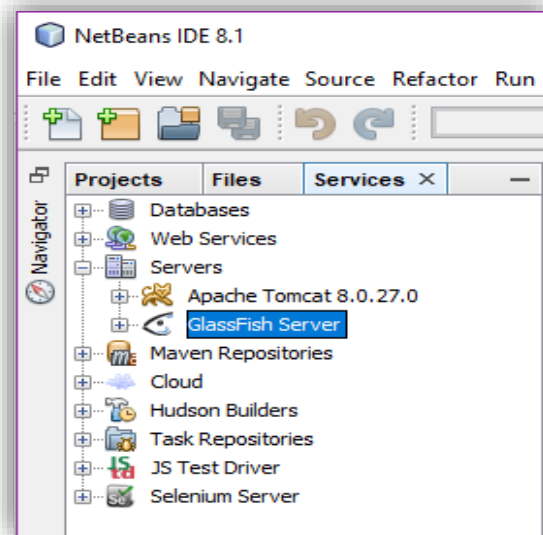


Figura 2. Servidores.

Seguidamente seleccionamos el servidor Glassfish 4.1 damos clic derecho y nuevamente clic en la opción **Start**.

Una vez iniciado el servidor, nos dirigimos a una pestaña del navegador Web e ingresamos la siguiente dirección **localhost:4848**. Al iniciar la interfaz de administración se procede a crear el pool de conexiones, para ello nos dirigimos a la opción **Resources** y damos clic en **JDBC**, que permite mostrar en la parte derecha de la interfaz dos opciones, **JDBC Resources** y **JDBC Connection Pools** (Figura 3).



Figura 3. Opciones JDBC.

Se escoge la opción **JDBC Connection Pools** y se da clic, esta acción nos dirigirá a otra interfaz, en la cual se presenta una tabla con nueva información (Figura 4).



Figura 4. Tabla JDBC Connection Pools.

Seleccionamos el botón **New** y al dar clic se visualizará un formulario en el cual ingresamos la siguiente información (Figura 5).

- **Pool Name:** doctoradoConnectionPool
- **Resource Type:** javax.sql.DataSource
- **Database Driver Vendor:** MySql

Figura 5. Formulario New JDBC Connection Pool.

Una vez se ha ingresado la información se procede a dar clic en el botón **Next** que se encuentra ubicado en la parte superior derecha de la interfaz.

Considerando que el anterior procedimiento se ha realizado de manera correcta, aparecerá una interfaz, en la cual debemos desplazarnos hasta la parte inferior, donde aparece otra tabla para adicionar propiedades (**Additional Properties**), pero antes de crear una propiedad debemos eliminar las que se encuentran en dicha tabla, por lo cual seleccionamos todas las propiedades y las eliminamos dando clic en el botón **Delete Properties** (Figura 6).

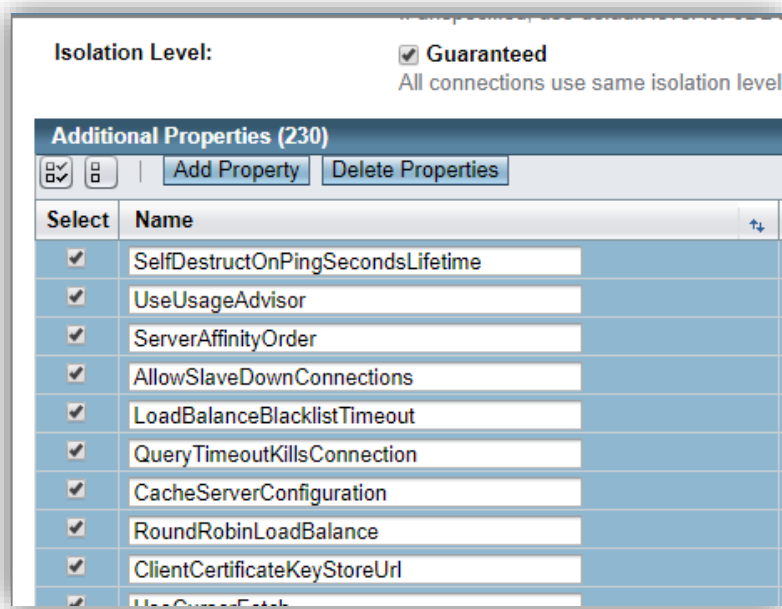


Figura 6. Additional Properties.

Dado que se han eliminado todas las propiedades procedemos a crear siete nuevas propiedades, para ello, daremos clic en el botón **Add Property** para cada propiedad, incluyendo la información de la siguiente manera (Figura 7).

- **DriverClass:** com.mysql.jdbc.Driver
- **DatabaseName:** doctorado
- **Password:** Doc2017_I
- **URL:** jdbc:mysql://localhost:3306/doctorado?zeroDateTimeBehavior=convertToNull
- **ServerName:** localhost
- **PortNumber:** 3306

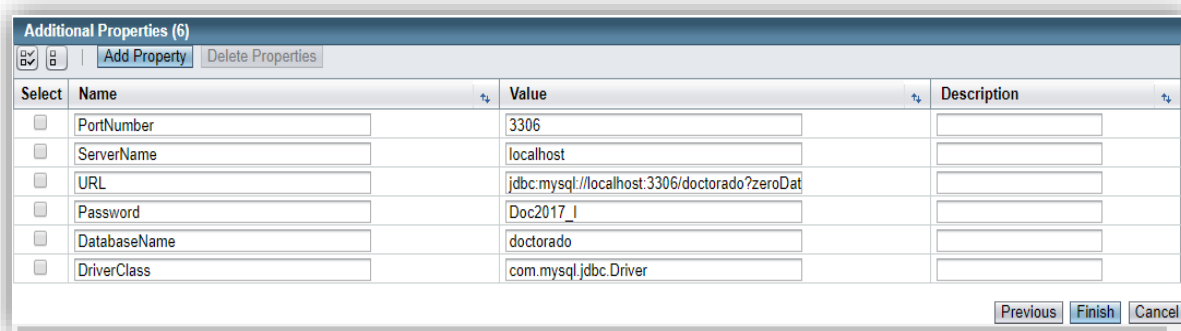
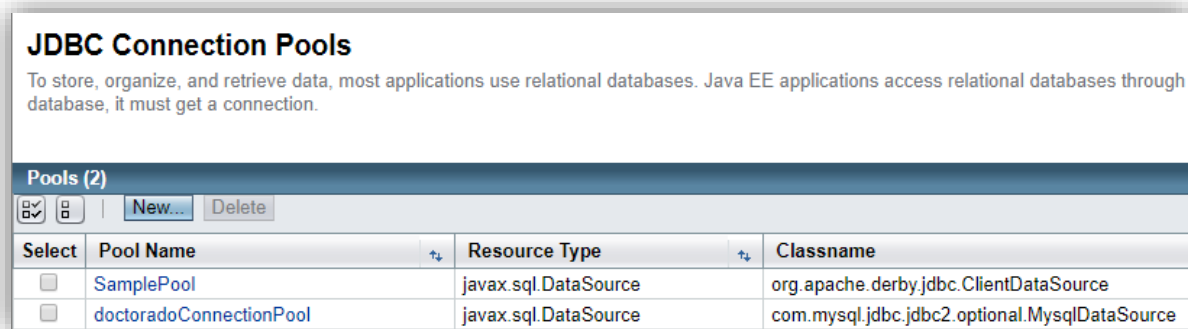


Figura 7. Añadir Propiedades.

Cuando se hayan ingresado las siete propiedades, damos clic en el botón **Finish**.

Después de realizar todo este primer paso, se nos mostrará una lista de pool de conexiones, donde encontraremos la conexión que hemos creado (**doctoradoConnectionPool**) (Figura 7).



JDBC Connection Pools

To store, organize, and retrieve data, most applications use relational databases. Java EE applications access relational databases through database, it must get a connection.

Pools (2)

☐ ☐ | [New...](#) [Delete](#)

| Select | Pool Name | Resource Type | Classname |
|--------------------------|-------------------------|----------------------|---|
| <input type="checkbox"/> | SamplePool | javax.sql.DataSource | org.apache.derby.jdbc.ClientDataSource |
| <input type="checkbox"/> | doctoradoConnectionPool | javax.sql.DataSource | com.mysql.jdbc.jdbc2.optional.MysqlDataSource |

Figura 7. JDBC Connection Pools.

Ahora como segundo paso, procedemos a crear el **JDBC Resources**, así que escogemos la otra opción que se presenta en la interfaz del JDBC (Figura 8).

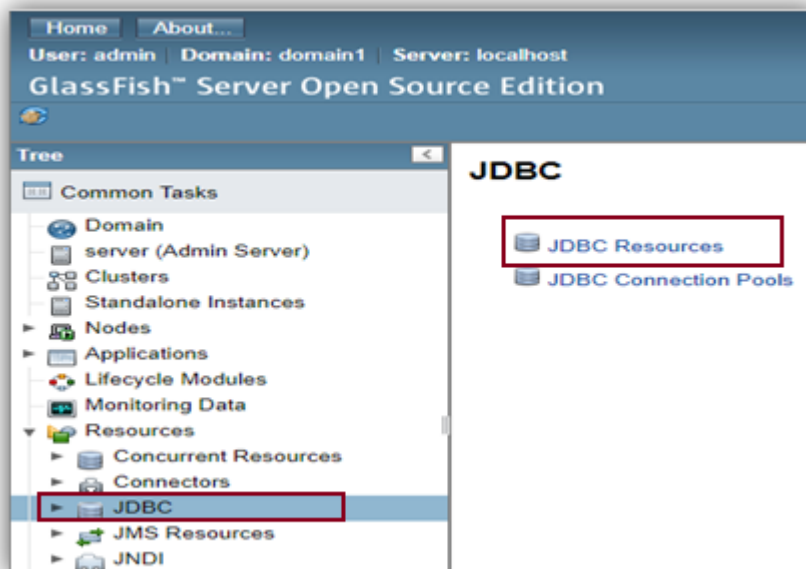


Figura 8. JDBC Resources.

Al dar clic en la opción JDBC Resources, se nos mostrará una lista con **JDBC Resources** existentes, damos clic en el botón **New** para crear el JDBC Resources (Figura 9).

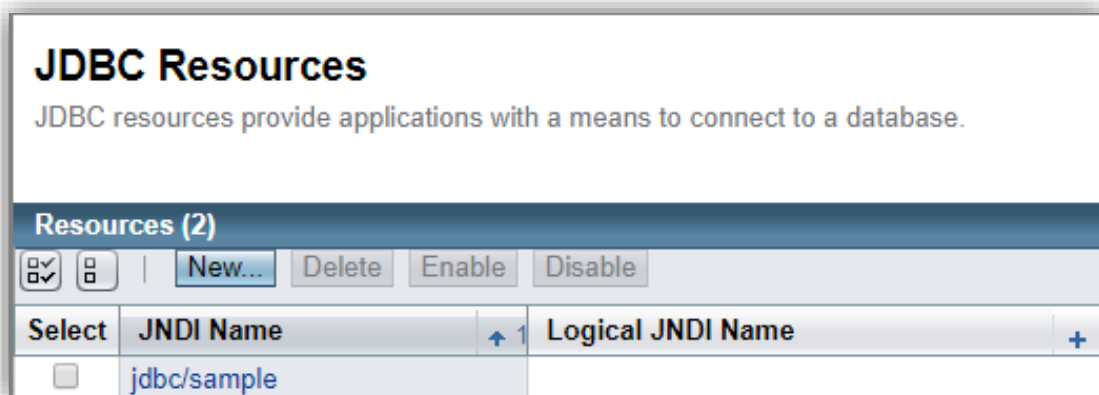


Figura 9. Lista de JDBC Resources.

Seguidamente se nos mostrará un formulario, en el cual se deben ingresar los siguientes datos.

- **JNDI Name:** jdbc/doctorado
- **Pool Name:** doctoradoConnectionPool (se escoge el pool de conexión que se creó anteriormente).

Posteriormente damos clic en el botón **OK**.

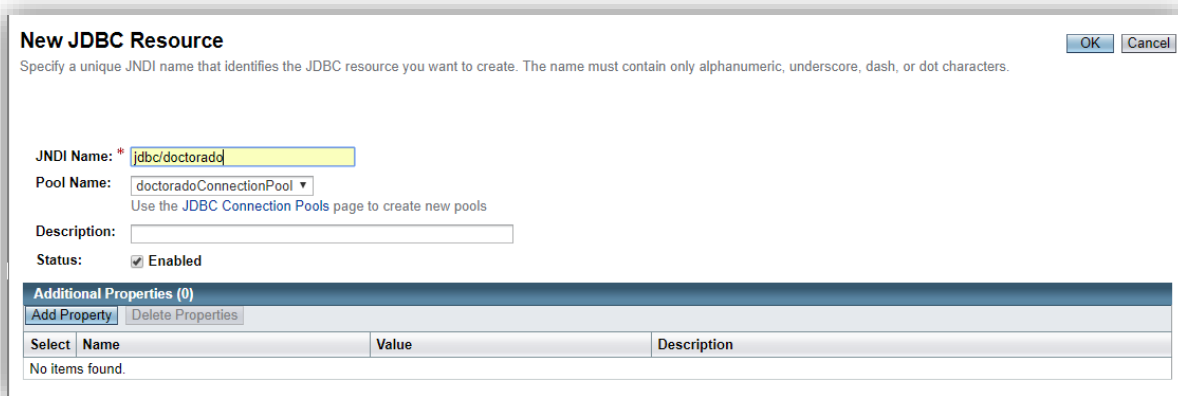


Figura 10. Información para crear el JDBC Resources.

Una vez realizada la acción correctamente, nos aparecerá el nuevo JDBC Resources en la lista (Figura 11).

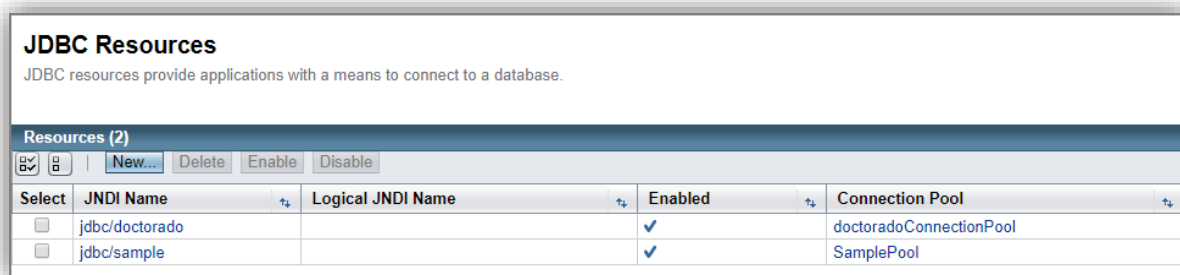


Figura 11. Jdbc/doctorado.

1. Dominio Seguridad.

Para poder realizar la configuración del dominio de seguridad, en el menú que presenta la interfaz del servidor Glassfish en la parte izquierda, buscamos el ítem **Configurations -> Server-config -> Security**, luego damos clic en la opción **Realms** y luego clic en el botón **New** presentado en la parte derecha de la interfaz (Figura 12).

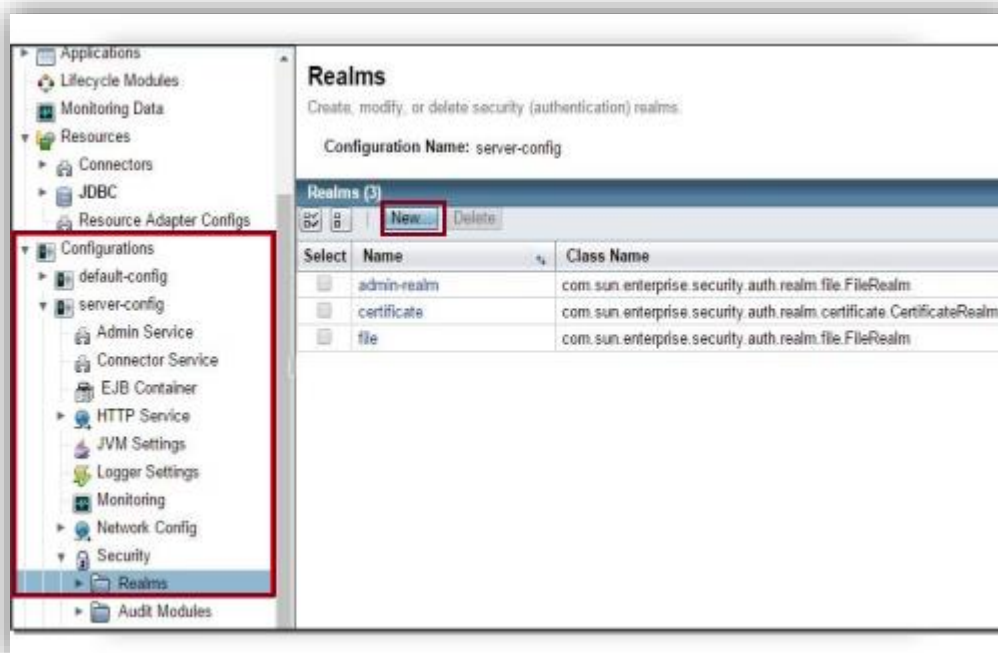
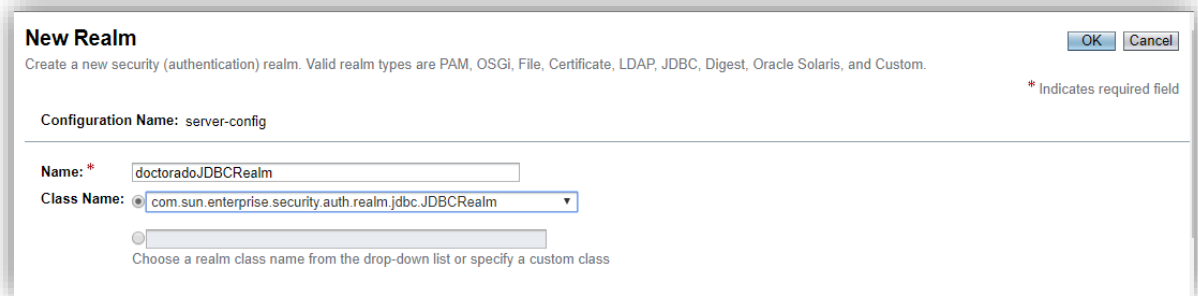


Figura 12. Creación del Realms.

Realizado lo anterior, se nos mostrará un formulario en el cual se deben ingresar la siguiente información (Figura 13).

- **Realm Name:** doctoradoJDBCRealm
- **Class Name:** com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm



The 'New Realm' dialog box is used to create a new security (authentication) realm. It includes a title bar, a subtitle, and a list of valid realm types. The 'Configuration Name' is set to 'server-config'. The 'Name' field is 'doctoradoJDBCRealm'. The 'Class Name' is 'com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm'. There is a radio button for 'Choose a realm class name from the drop-down list or specify a custom class'.

New Realm OK Cancel

Create a new security (authentication) realm. Valid realm types are PAM, OSGi, File, Certificate, LDAP, JDBC, Digest, Oracle Solaris, and Custom.

* Indicates required field

Configuration Name: server-config

Name: * doctoradoJDBCRealm

Class Name: com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm

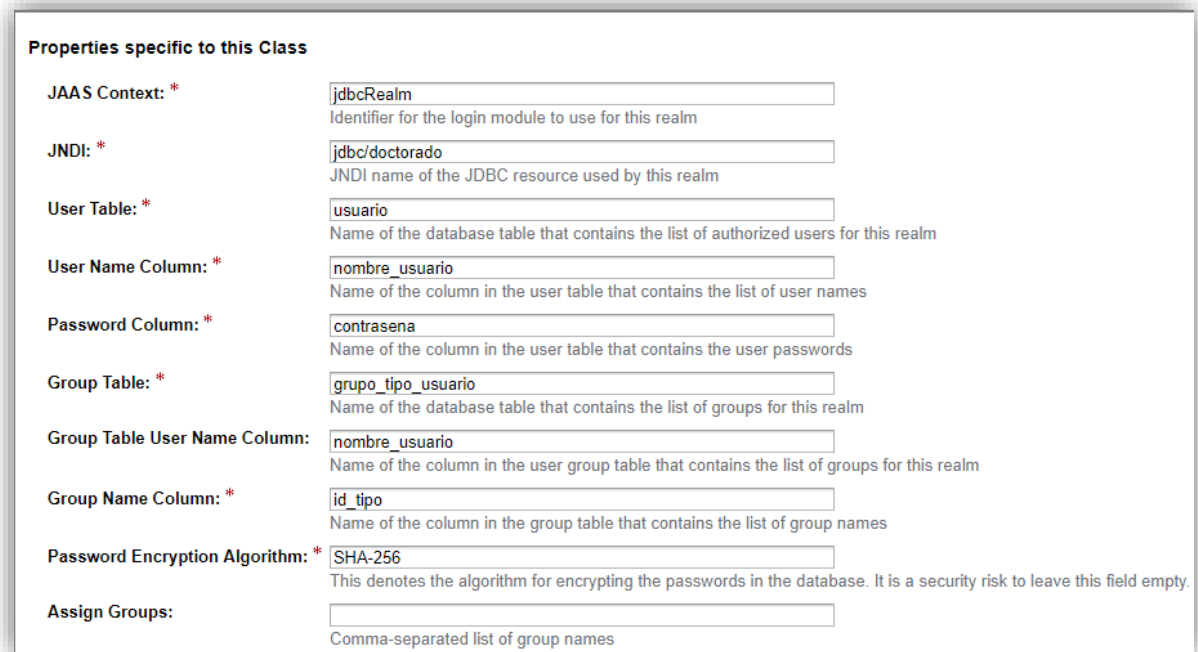
Choose a realm class name from the drop-down list or specify a custom class

Figura 13. Información del Realm.

Una vez se escoja el **Class Name** correspondiente, se desplegará un formulario (**Properties specific to this Class**), en el cual ingresamos los siguientes datos.

- **JAAS Context:** jdbcRealm
- **JNDI:** jdbc/doctorado
- **User Table:** usuario
- **User Name Column:** nombre_usuario
- **Password Column:** contrasena
- **Group Table:** grupo_tipo_usuario
- **Group Table User Name Column:** nombre_usuario
- **Group Name Column:** id_tipo
- **Password Encryption Algorithm:** SHA-256

El formulario quedará de la siguiente manera (Figura 14).



The 'Properties specific to this Class' dialog box is used to configure the properties for the selected realm. It includes a title bar and a list of properties. The 'JAAS Context' is 'jdbcRealm'. The 'JNDI' is 'jdbc/doctorado'. The 'User Table' is 'usuario'. The 'User Name Column' is 'nombre_usuario'. The 'Password Column' is 'contrasena'. The 'Group Table' is 'grupo_tipo_usuario'. The 'Group Table User Name Column' is 'nombre_usuario'. The 'Group Name Column' is 'id_tipo'. The 'Password Encryption Algorithm' is 'SHA-256'. The 'Assign Groups' field is empty.

Properties specific to this Class

JAAS Context: * jdbcRealm
Identifier for the login module to use for this realm

JNDI: * jdbc/doctorado
JNDI name of the JDBC resource used by this realm

User Table: * usuario
Name of the database table that contains the list of authorized users for this realm

User Name Column: * nombre_usuario
Name of the column in the user table that contains the list of user names

Password Column: * contrasena
Name of the column in the user table that contains the user passwords

Group Table: * grupo_tipo_usuario
Name of the database table that contains the list of groups for this realm

Group Table User Name Column: nombre_usuario
Name of the column in the user group table that contains the list of groups for this realm

Group Name Column: * id_tipo
Name of the column in the group table that contains the list of group names

Password Encryption Algorithm: * SHA-256
This denotes the algorithm for encrypting the passwords in the database. It is a security risk to leave this field empty.

Assign Groups:
Comma-separated list of group names

Figura 14. Propiedades específicas de esta clase.

Luego nos dirigimos a la parte inferior de la interfaz y damos clic en el botón **OK**.

Si la configuración se ha realizado de manera correcta, se nos mostrará el Realm creado en la lista de Realms (Figura 15).

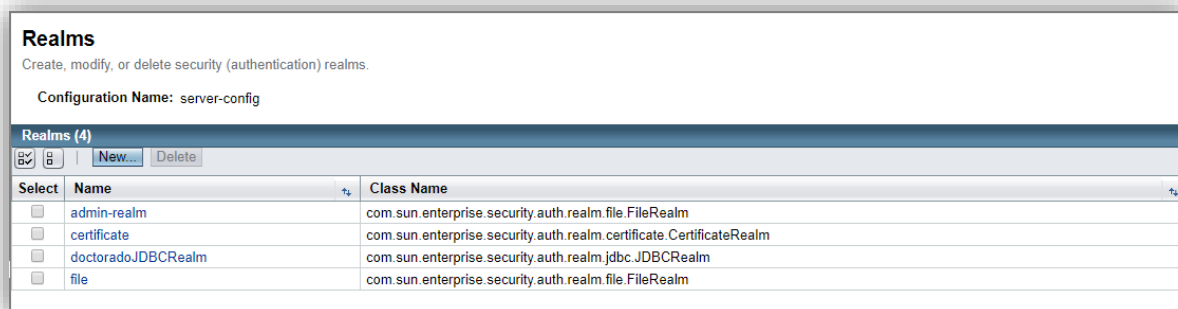


Figura 15. Listado de Realms.

2. Configuración del entorno de desarrollo.

Teniendo en cuenta las configuraciones anteriormente realizadas, debemos tener en cuenta que debemos configurar nuestro entorno de desarrollo Netbeans 8.1 para que la aplicación funcione correctamente.

Lo primero que se debe realizar es la creación del descriptor glassfish, para ello se selecciona el **Proyecto -> New -> Glassfish Descriptor** (Figura 16).

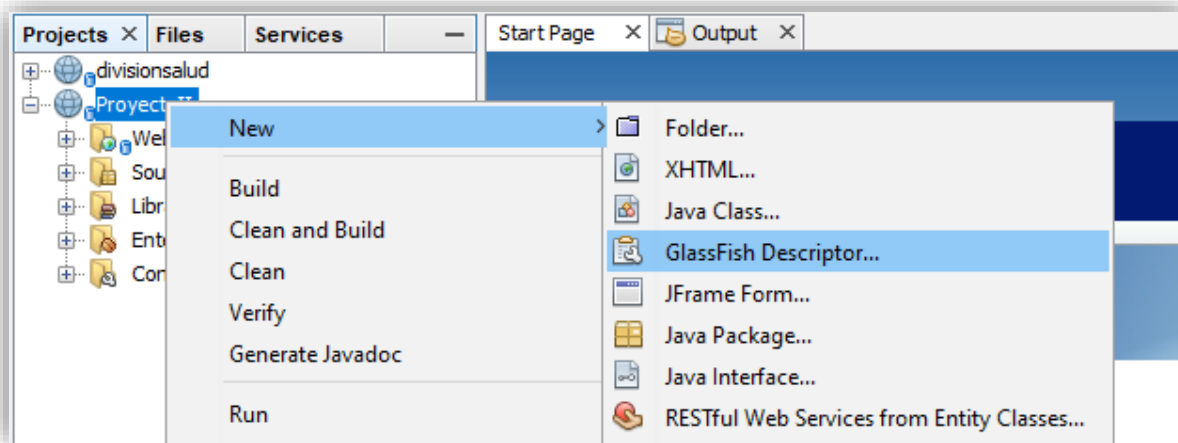


Figura 16. Creación del descriptor Glassfish.

Considerando que ya se ha creado el descriptor Glassfish llamado **glassfish-web.xml**, el cual estará presente en la carpeta **Configuration Files** y en **WEB-INF**. Lo que haremos a continuación es escoger el descriptor glassfish que se encuentra en la carpeta WEB-INF, como se puede observar al abrir el archivo va a presentar una ventana con varias pestañas, entonces nos situamos en la pestaña **General** y en **Context Root** ingresamos información de la siguiente manera (Figura 17).

Todos los cambios que se hagan en el descriptor glassfish de la carpeta WEB-INF, quedarán guardados en el descriptor glassfish de la carpeta Configuration Files.

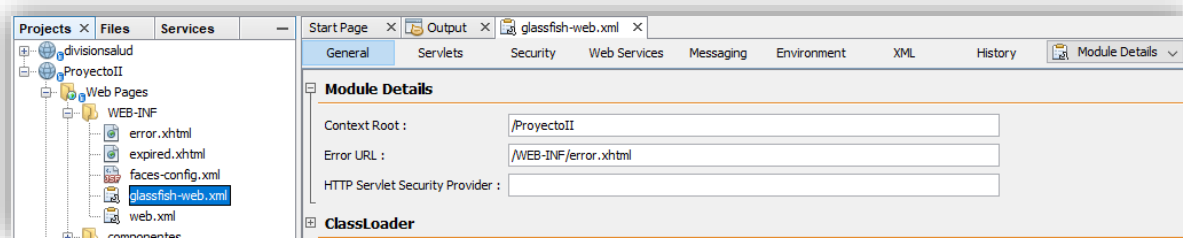
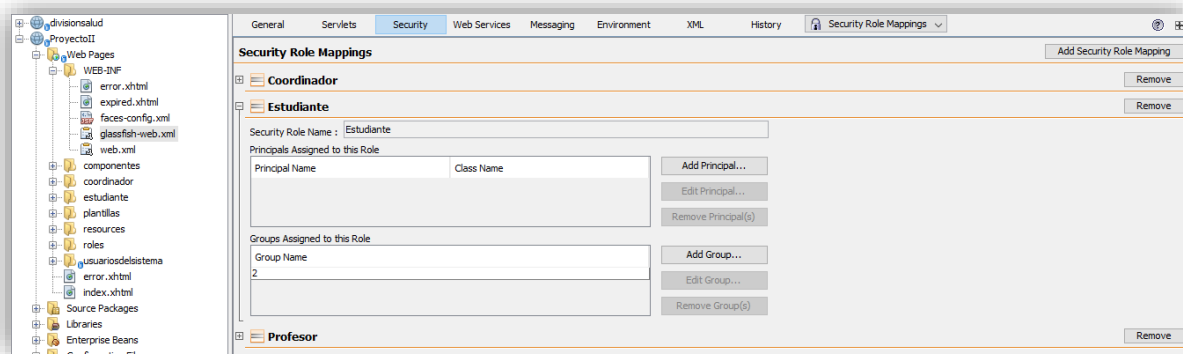
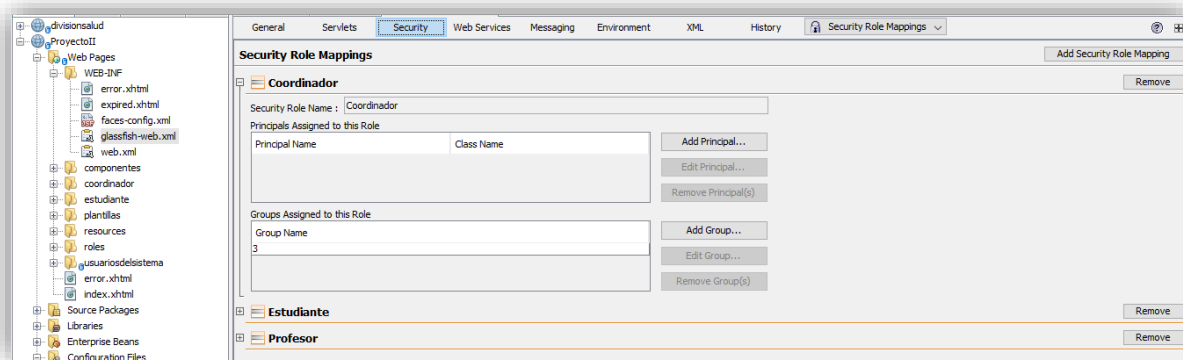


Figura 17. Configuración general glassfish-web.xml.

Seguidamente vamos a la pestaña **Security** y damos clic en el botón **Add Security Role Mapping**, que nos permite agregar los roles de estudiante, coordinador y profesor que son los que existen en la aplicación (Figura 18).



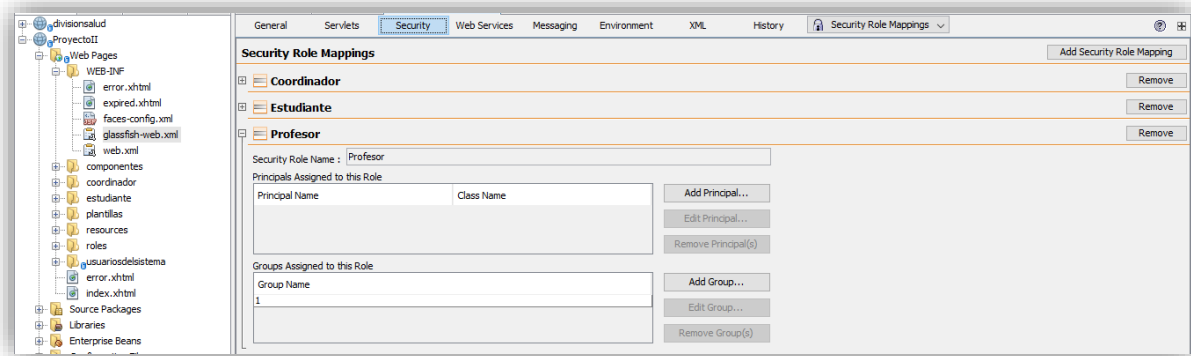


Figura 18. Roles de la aplicación.

Realizadas las modificaciones, damos por terminada la configuración del descriptor glassfish, los números que cada rol tiene asignado, son los identificadores que se le dieron a cada uno en la base de datos.

Ahora procedemos a modificar el archivo [web.xml](#), el cual se encuentra en las carpetas [Configuration Files](#) y en [WEB-INF](#), de la misma manera que con el descriptor glassfish, este archivo nos presenta una ventana con varias pestañas, de las cuales solo vamos a configurar [Pages](#) y [Security](#).

Por lo cual nos situamos en la pestaña [Pages](#) y en el campo [Welcome Files](#), verificamos que este la ruta principal de la aplicación, en este caso sería [faces/index.xhtml](#), también se debe incluir la(s) página(s) para mostrar errores en el campo [Error Page Location](#) e ingresamos [/WEB-INF/error.xhtml](#), de la siguiente manera (Figura 19).

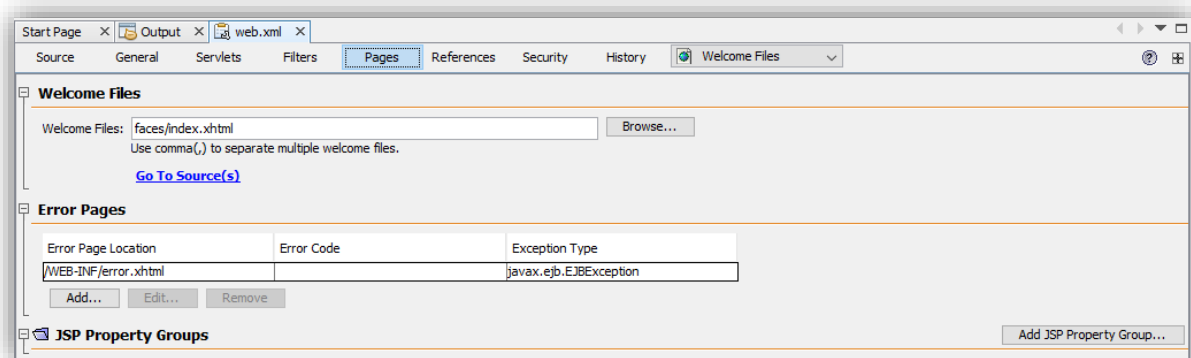


Figura 19. Información pestaña Pages.

Después de haber realizado este paso, vamos a la pestaña [Security](#), buscamos el ítem [Login Configuration](#) y seleccionamos la opción [Form](#), la cual nos permite seleccionar la página de las credenciales del usuario (Login, Contraseña), la página de error y el nombre del Realm.

Para la aplicación, la información es la siguiente.

- [Form Login Page](#): [/faces/error.xhtml](#)
- [Form Error Page](#): [/faces/error.xhtml](#)

- **Realm Name:** doctoradoJDBCRealm

Luego buscamos el ítem **Security Roles** y agregamos cada uno de los roles, con una pequeña descripción (Figura 20).

| Role Name | Description |
|-------------|---------------------|
| Profesor | Usuario Profesor |
| Estudiante | Usuario Estudiante |
| Coordinador | Usuario Coordinador |

Figura 20. Información pestaña Security.

Si nos desplazamos hacia abajo encontramos el ítem **Security Constraints**, que nos permite agregar las páginas correspondientes a cada rol, damos clic en el botón **Add Security Constraints** y se nos mostrará una interfaz que contiene un formulario, donde ingresaremos los correspondientes datos de la siguiente manera (Figura 21).

Hay que tener en cuenta que la opción **Enable Authentication Constraint**, debe estar habilitada, para poder agregar el rol en Role Name(s).

| Name | URL Pattern | HTTP Method | Description |
|--------------|---------------------------------------|-------------|--------------------|
| Estudiante - | /faces/usuariosdelistema/estudiante/* | | Estudiante - pages |

Figura 21. Información pestaña Security Constraints.

En este formulario debemos ingresar información en los siguientes campos, en **Display Name**, **Web Resource Collection** y **Role Name(s)**.

En el campo Display Name se ingresa un nombre que especifique a que rol pertenecen las vistas (estudiante pages).

En **Web Resources Collection** damos clic en el botón **Add** e inmediatamente nos aparece un formulario, el cual debe contener la información correspondiente a cada rol (Figura 22).

Resource Name: Estudiante -

Description: Estudiante - pages

URL Pattern(s): /faces/usuariosdelsistema/estudiante/*

Use comma (,) to separate multiple patterns.

HTTP Method(s): ☒ All HTTP Methods
☐ Selected HTTP Methods

☐ GET ☐ POST
☐ HEAD ☐ PUT
☐ OPTIONS ☐ TRACE
☐ DELETE

OK Cancel

Figura 22. Resource Collection de Estudiante.

De igual manera es así para cada rol. En **Resources Name** se ingresa un nombre que esté relacionado con el Rol, en **Description** se ingresa una pequeña descripción que haga referencia al rol y al tipo de recurso que se tiene y en **URL Pattern(s)** se ingresa la ruta de cada una de las vistas que manejará cada rol.

Teniendo en cuenta que en caso de que se requiera ingresar varias vistas, todas se separan por **“,” (coma)** y si es un paquete de vistas solo se coloca **“*” (asterisco)**.

En el caso del rol de Coordinador se ingresa la información de esta manera (Figura 23).

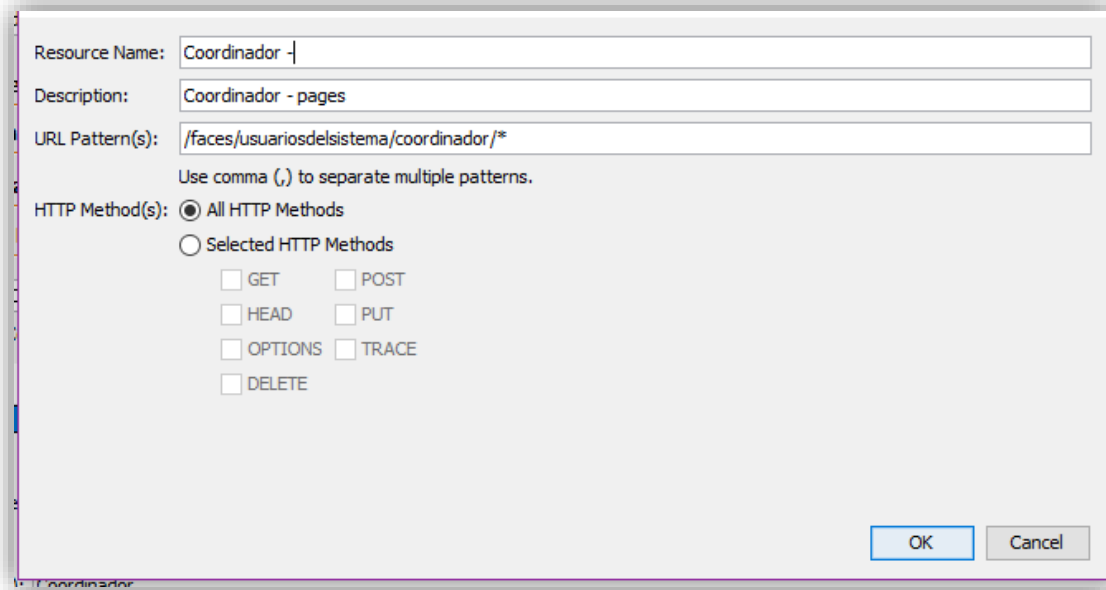


Figura 23. Resource Collection Coordinador.

Por último, si todos los pasos anteriores se han efectuado de manera correcta, lo que hacemos es abrir el archivo **Persistence.xml** ubicado en la carpeta **Configuration Files** y en la pestaña **Desing**, en el campo **Data Source** ingresamos el JDBC que se ha creado en pasos anteriores en este caso es **jdbc/doctorado** (Figura 24).

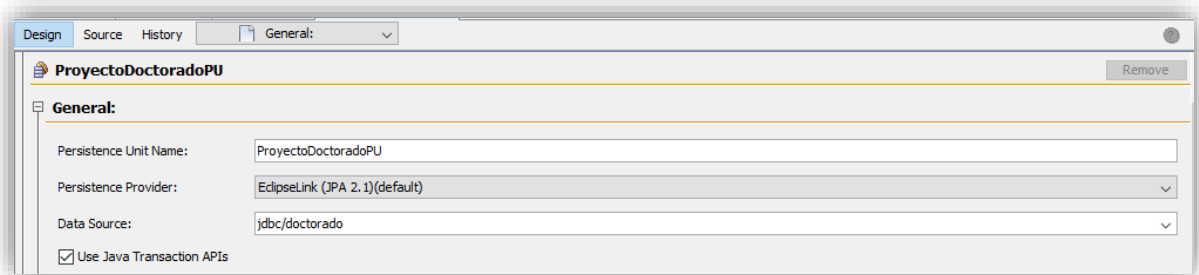


Figura 24. Configuración del archivo Persistence.xml.

NOTA: Abrir el servidor web de plataforma Xampp e iniciarlo.

Para asegurarnos de que la aplicación está funcionando correctamente, nos dirigimos al entorno de desarrollo Netbeans 8.1 y seleccionamos el proyecto en este caso ProyectoII damos clic derecho y nuevamente clic en la opción **Run** (Figura 25).

Esperamos mientras el servidor se ejecuta para que pueda iniciar la aplicación.

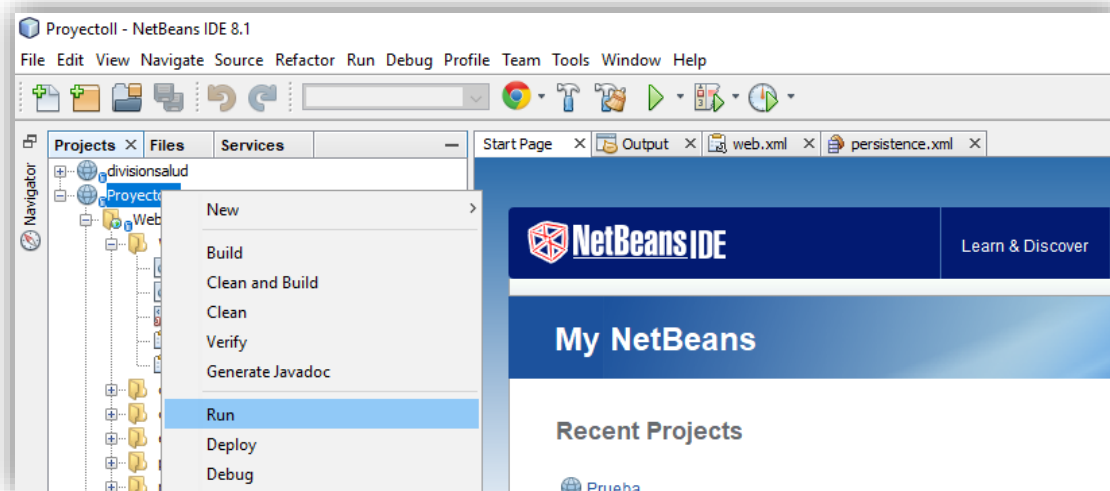


Figura 25. Iniciar Proyectoll.

Una vez inicie la aplicación, la interfaz será la siguiente (Figura 26).

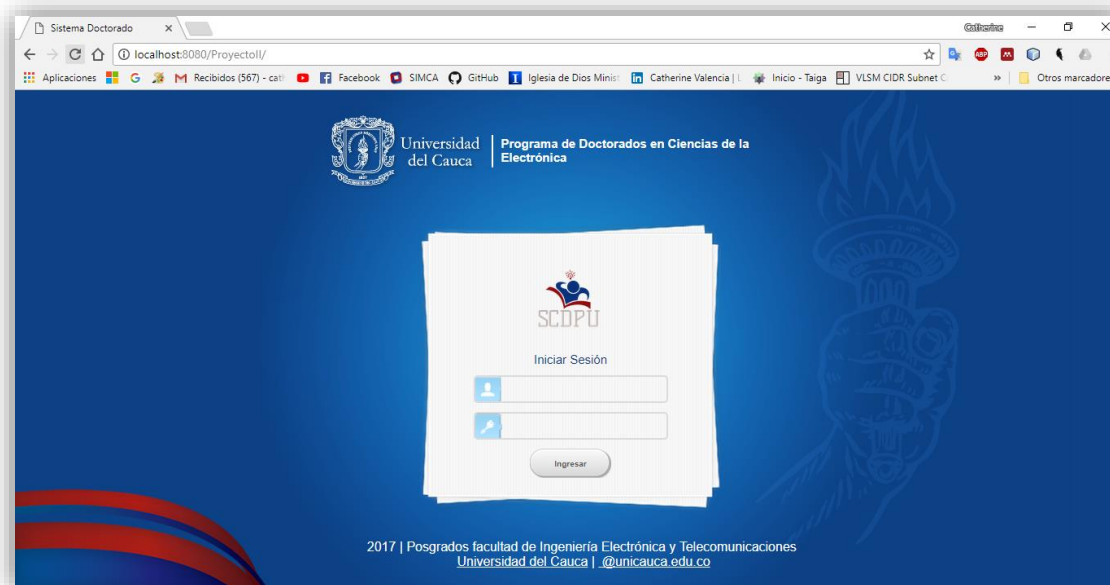


Figura 26. Aplicación Programa de Doctorado en Ciencias de la Electrónica.

NOTA: Después de realizar los pasos anteriores, por favor leer el manual de configuración de OpenKM.