

Análisis y Diseño Orientado a Objetos (FIM151) - UFASTA Trabajo Práctico Cursada 2014

Objetivo

En el marco del reglamento de cursada 2014 y como exigencia para la aprobación de la misma, el objetivo del práctico consiste en desarrollar un servidor de autenticación de usuarios para que distintas aplicaciones clientes puedan hacer uso del servicio y validar a sus usuarios.

Requisitos del TP

- Conformar grupos de no más de 3 personas
- Entregar la aplicación completamente funcional
- Entregar los fuentes y la respectiva documentación
- Cumplir con las fechas de entrega pautadas por la cátedra
- La presentación consistirá en probar la aplicación con una base de datos externa que la cátedra proveerá. Todos los integrantes deberán estar presentes

Descripción del proyecto

El primer modo de uso de la aplicación a diseñar e implementar (servidor de autenticación), debe responder en forma sincrónica a mensajes XML enviados por las aplicaciones cliente. A cada mensaje recibido se le enviará una respuesta que también será del tipo XML. El servidor deberá soportar mensajes con diferentes funciones.

Las operaciones de alta, baja, listado de usuarios y autenticaciones solo podrán ser realizadas por el administrador.

Toda la persistencia debe ser realizada en un base de datos relacional (mysql). El modelo de datos de la DB lo entrega la cátedra

La información de configuración del servidor debe estar en un archivo del tipo "properties" y debe contener: url de conexión a la base de datos, contraseña del administrador, puerto que utiliza el servidor de autenticación, etc.

El otro modo de uso, debe soportar la posibilidad de embeber el servidor de autenticación en cualquier otra aplicación Java (agregando el .jar como librería externa de la aplicación cliente).

Tipos de mensajes:

ADD: agrega un nuevo usuario .

REMOVE: quita un usuario.

MODIFY: permite modificar la contraseña de un usuario.

AUTHENTICATE: autentica un usuario con su contraseña.

LIST-USERS: lista todos los usuarios.

LIST-AUT: lista todas las autenticaciones que tuvo el usuario (AUT=Authentication).

Restricciones de diseño

Se deben considerar las siguientes restricciones:

- Programación en lenguaje JAVA.
- Definir y documentar la arquitectura
- OR mapping
- Se utilizará MySQL con el driver nativo JDBC y la base de datos definida por la cátedra
- Multihilo para atender clientes en simultáneo.
- Protocolo TCP-IP entre el cliente y el servidor implementado.

Recomendaciones de diseño

- Se valorará la utilización de Patrones de diseño en la aplicación.

Entregables

- Diagrama de clases.*
- Diagrama de paquetes.*
- Fuentes.*
- Manual de instalación.*
- Javadoc*

Presentación

La presentación del trabajo práctico se realizará en el laboratorio 2 con el sistema funcionando y con todos los integrantes presentes. El sistema deberá tener implementada toda la funcionalidad anteriormente descrita.

Estructura de mensajes que debe soportar el servidor de autenticación

Mensaje para agregar un usuario:

```
<MESSAGE TYPE="ADD">
  <USERNAME>usuario</USERNAME>
  <PASSWORD>contraseña</PASSWORD>
  <ADM-PASS>contraseña administrador</ADM-PASS>
</MESSAGE>
```

Mensaje para dar de baja un usuario:

```
<MESSAGE TYPE="REMOVE">
  <USERNAME>usuairo</USERNAME>
  <ADM-PASS>contraseña</ADM-PASS>
</MESSAGE>
```

Mensaje para modificar la contraseña de un usuario:

```
<MESSAGE TYPE="MODIFY">
  <USERNAME>usuario</USERNAME>
  <PASSWORD>contraseña</PASSWORD>
  <NEW-PASS>nueva contraseña</NEW-PASS>
</MESSAGE>
```

Mensaje para autenticar un usuario:

```
<MESSAGE TYPE="AUTHENTICATE">
  <USERNAME>usuario</USERNAME>
  <PASSWORD>contraseña</PASSWORD>
</MESSAGE>
```

Respuesta de los mensajes ADD,REMOVE,MODIFY y AUTHENTICATE:

```
<ACK STATUS="OK|ERROR">
  <DESC>motivo del error</DESC>
</ACK>
```

El atributo STATUS puede tener OK o ERROR:

- OK: se pudo procesar el mensaje.
- ERROR: no se pudo procesar el mensaje.
En este caso el tag "DESC" indica el posible motivo del error.

Mensaje para listar todos los usuarios actualmente dados de alta en el servidor

```
<MESSAGE TYPE="LIST-USERS" />
  <ADM-PASS>contraseña administrador</ADM-PASS>
</MESSAGE>
```

Respuesta

```

<LIST-USERS>
  <USER>
    <USERNAME>usuario</USERNAME>
    <TIMESTAMP>dd/mm/yyyy hh:mm:ss</TIMESTAMP>
  </USER>
  <USER>
    <USERNAME>usuario</USERNAME>
    <TIMESTAMP>dd/mm/yyyy hh:mm:ss</TIMESTAMP>
  </USER>
  .....
</LIST-USERS>

```

- El tag `TIMESTAMP` corresponde a la fecha y hora del alta.

Mensaje para listar todas las autenticaciones que realizó un usuario:

```

<MESSAGE TYPE="LIST-AUT" />
  <USERNAME>usuario</USERNAME>
  <ADM-PASS>contraseña administrador</ADM-PASS>
</MESSAGE>

```

Respuesta

```

<LIST-AUT>
  <AUT>
    <HOST>###.###.###.###</HOST>
    <TIMESTAMP>dd/mm/yyyy hh:mm:ss</TIMESTAMP>
  </AUT>
  <AUT>
    <HOST>###.###.###.###</HOST>
    <TIMESTAMP>dd/mm/yyyy hh:mm:ss</TIMESTAMP>
  </AUT>
  .....
</LIST-AUT>

```

- El tag `HOST` indica la IP desde la cual se solicitó la autenticación
- El tag `TIMESTAMP` corresponde a la fecha y hora de la autenticación.

Modelo de datos MYSQL

base de datos: **autenticacion**

tabla **usuarios**

```

username: char(20)
password: char(20)
timestamp: datetime

```

tabla **autenticaciones**

```

username: char(20)
host: char(15)
timestamp: datetime

```