UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO

FACULTAD DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES



Práctica No 0-SA. Consulta a una Base de Datos vía Correo Electrónico

**GRUPO 05-SA:**

CADIMA BARRERO JHAFETH 216008131

TORREZ ARAMAYO JORGE RODRIGO 217051367

MENDOZA ALLERDING JOSE CARLOS 216031850

**SIGLA:** INF513-SA **MATERIA:** TECNOLOGIA WEB

**SEMESTRE:** 1-2020

**DOCENTE:** M.sc. Ing. EVANS BALCÁZAR VEIZAGA

Santa Cruz-Bolivia

**Realizar un programa en java que permita conectar a una Base de Datos ubicada en:**

**Servidor**: virtual.uagrm.edu.bo (PostgreSQL)

**Usuario**: agenda

**Contraseña**: agendaagenda

**Base de Datos:** db\_agenda

**Tabla:** amigo

Mediante un **Patrón** enviado en el **SUBJECT** de un **Correo Electrónico**, permita realizar el **SELECT** de la tabla amigo en los campos de tipo String, y devuelva un correo con los registros de la tabla que **coinciden** con el **patrón**.

SUBJECT: en => select \* from amigo where .... \*en\*

El programa consta de 3 clases:

* SMTP.java
* POP3.java
* DB.java

La clase SMTP contiene todos los atributos y métodos para conectarnos al servidor de correo SMTP y poder enviarlos, su código es el siguiente:

package Datos;

import java.io.\*;

import java.net.\*;

public class SMTP {

private static final String HOST = "virtual.fcet.uagrm.edu.bo";

private static final int PORT = 25;

private static final String FROM = "jhcadima@gmail.com";

private static final String RCPT = "grupo05sa@virtual.fcet.uagrm.edu.bo";

private String COMAND = "";

private static final String SALTO = "\n";

private Socket Conexion;

private BufferedReader Entrada;

private DataOutputStream Salida;

public SMTP() {

this.Conexion = null;

this.Entrada = null;

this.Salida = null;

}

public void connect() throws IOException {

this.Conexion = new Socket(HOST, PORT);

this.Entrada = new BufferedReader(new InputStreamReader(Conexion.getInputStream()));

this.Salida = new DataOutputStream(Conexion.getOutputStream());

System.out.println("S:" + this.Entrada.readLine());

}

public void close() throws IOException {

this.Conexion.close();

this.Entrada.close();

this.Salida.close();

}

public void logIn() throws IOException {

COMAND = "HELO " + HOST + SALTO;

this.Salida.writeBytes(COMAND);

System.out.println("S:" + this.Entrada.readLine());

}

public void logOut() throws IOException {

COMAND = "quit" + SALTO;

this.Salida.writeBytes(COMAND);

System.out.println("S:" + this.Entrada.readLine());

}

public void sendMail(String SUBJECT, String DATA) throws IOException {

COMAND = "MAIL FROM: " + FROM + SALTO;

this.Salida.writeBytes(COMAND);

System.out.println("S:" + this.Entrada.readLine());

COMAND = "RCPT TO: " + RCPT + SALTO;

this.Salida.writeBytes(COMAND);

System.out.println("S:" + this.Entrada.readLine());

COMAND = "DATA" + SALTO;

this.Salida.writeBytes(COMAND);

System.out.println("S:" + this.Entrada.readLine());

COMAND = "Subject: " + SUBJECT + SALTO;

this.Salida.writeBytes(COMAND);

COMAND = DATA + SALTO;

this.Salida.writeBytes(COMAND);

COMAND = "." + SALTO;

this.Salida.writeBytes(COMAND);

System.out.println("S:" + Entrada.readLine());

}

}

La clase POP3 contiene todos los atributos y métodos para conectarnos al servidor y así poder leer los correos de la bandeja de entrada para analizar los SUBJECT, su código es el siguiente:

package Datos;

import java.io.\*;

import java.net.\*;

public class POP3 {

private static final String HOST = "virtual.fcet.uagrm.edu.bo";

private static final int PORT = 110;

private static final String USER = "grupo05sa";

private static final String PASS = "grupo05grupo05";

private static final String SALTO = "\n";

private String COMAND = "";

private Socket Conexion;

private BufferedReader Entrada;

private DataOutputStream Salida;

public POP3() {

this.Conexion = null;

this.Entrada = null;

this.Salida = null;

}

public void connect() throws IOException {

this.Conexion = new Socket(HOST, PORT);

this.Entrada = new BufferedReader(new InputStreamReader(Conexion.getInputStream()));

this.Salida = new DataOutputStream(Conexion.getOutputStream());

System.out.println("S:" + this.Entrada.readLine());

}

public void close() throws IOException {

this.Conexion.close();

this.Entrada.close();

this.Salida.close();

}

public void logIn() throws IOException {

COMAND = "user " + USER + SALTO;

this.Salida.writeBytes(COMAND);

this.Entrada.readLine();

COMAND = "pass " + PASS + SALTO;

this.Salida.writeBytes(COMAND);

System.out.println("S:" + this.Entrada.readLine());

}

public void logOut() throws IOException {

COMAND = "quit" + SALTO;

this.Salida.writeBytes(COMAND);

System.out.println("S:" + this.Entrada.readLine());

}

public String getList() throws IOException {

COMAND = "list" + SALTO;

this.Salida.writeBytes(COMAND);

return getData(this.Entrada);

}

public void delete(String j) throws IOException {

this.Salida.writeBytes("dele " + j + SALTO);

System.out.println("S:" + this.Entrada.readLine());

}

public String getStat() {

String line = "";

try {

this.Salida.writeBytes("STAT" + SALTO);

line = this.Entrada.readLine();

} catch (IOException e) {

System.out.println(e.getMessage());

}

if (line != "") {

line = line.substring(4);

int i = 1;

while (line.charAt(i) != ' ') {

i++;

}

line = line.substring(0, i);

}

return line;

}

public String getSubject(String Mail) throws IOException {

String Line = "";

COMAND = "retr " + Mail + SALTO;

this.Salida.writeBytes(COMAND);

while ((Line = this.Entrada.readLine()) != null) {

//LINEA DEL SUBJECT

if (Line.contains("Subject:")) {

break;

}

}

Line = Line.substring(9);

getData(Entrada);

return Line;

}

public String getMail(String mail) throws IOException {

COMAND = "retr " + mail + SALTO;

this.Salida.writeBytes(COMAND);

return getData(this.Entrada);

}

static protected String getData(BufferedReader buffer) throws IOException {

String Data = "";

String Line = "";

while ((Line = buffer.readLine()) != null) {

//ULTIMA LINEA

if (Line.equals(".")) {

break;

}

//LINEA COMIENZA CON .

if ((Line.length() > 0) && (Line.charAt(0) == '.')) {

Line = Line.substring(1);

}

Data = Data + Line + SALTO;

}

return Data;

}

}

La clase DB contiene todos los atributos y métodos para conectarnos al servidor de base de datos PostgreSQL y realizar las consultas con el patrón especificado en el SUBJECT de un MAIL, su código es el siguiente:

package Datos;

import java.sql.\*;

public class DB {

private Connection connection;

private Statement statement;

private static final String HOST = "jdbc:postgresql://virtual.fcet.uagrm.edu.bo:5432/db\_agenda";

private static final String USER = "agenda";

private static final String PASS = "agendaagenda";

public DB() throws ClassNotFoundException, SQLException {

connection = null;

Class.forName("org.postgresql.Driver");

}

public void connect() throws SQLException {

connection = DriverManager.getConnection(HOST, USER, PASS);

}

public void close() throws SQLException {

connection.close();

}

public String selectFromTable(String condition) throws SQLException {

String tabla = "amig\_ci\tamig\_nombre\t\tamig\_appm\t\tamig\_telf\tamig\_dir\t\t amig\_fnac";

statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery("SELECT \* FROM amigo where "

+ "amig\_nombre like '%" + condition + "%'\n"

+ "or amig\_appm like '%" + condition + "%'\n"

+ "or amig\_telf like '%" + condition + "%'\n"

+ "or amig\_cel like '%" + condition + "%'\n"

+ "or amig\_dir like '%" + condition + "%'\n"

+ "or amig\_fnac like '%" + condition + "%'"

+ ";");

while (resultSet.next()) {

int ci = resultSet.getInt("amig\_ci");

String nombre = resultSet.getString("amig\_nombre");

String appm = resultSet.getString("amig\_appm");

String telf = resultSet.getString("amig\_telf");

String dir = resultSet.getString("amig\_dir");

String fnac = resultSet.getString("amig\_fnac");

tabla = tabla + "\n" + ci + "\t" + nombre + "\t" + appm + "\t" + telf + "\t" + dir + "\t" + fnac;

}

statement.close();

resultSet.close();

return tabla;

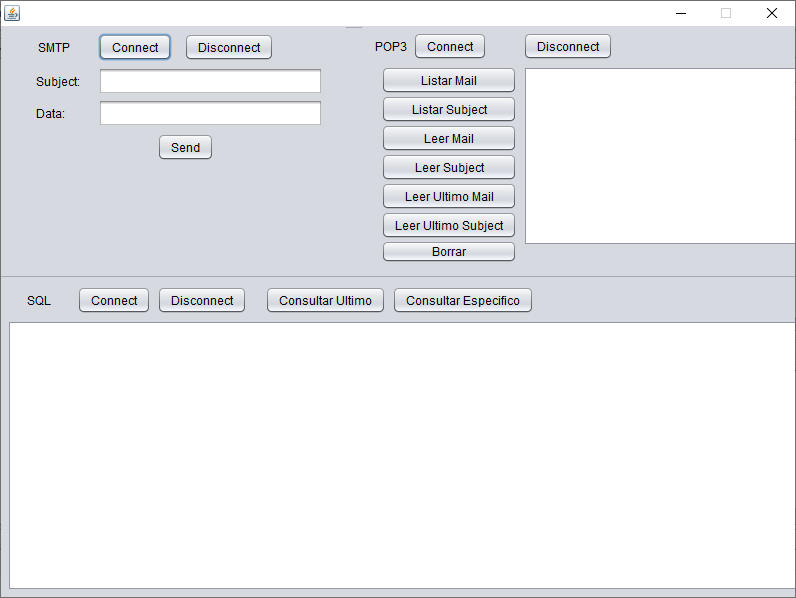
}

}

**IMPLEMENTACION**

Las clases anteriormente mencionadas fueron implementadas en un formulario JFrame, para tener una interfaz grafica más cómoda para usar.

El formulario tiene el siguiente aspecto:



Se tienen tres grupos de botones y áreas de texto, cada una para las clases mencionadas, explicaremos los botones y los métodos que ejecutan:

Los botones de Connect son en todos los casos para iniciar la conexión con los servidores, en el caso de SMTP y POP3 se utilizo la clase java.net.Socket; y para SQL java.SQL.\*; En SMTP y POP3 el botón Connect también ejecuta el logIn();

Los botones de Disconnect son en todos los casos para terminar la conexión con los servidores. En SMTP y POP3 el botón Disconnect también ejecuta el logOut();

El botón Send envía un correo electrónico a nuestro correo asignado: [grupo05sa@virtual.fcet.uagrm.edu.bo](mailto:grupo05sa@virtual.fcet.uagrm.edu.bo) con Asunto y Cuerpo que se rellenen en los cuadros de textos que están posicionados arriba de dicho botón.

Listar Mail genera una lista de los correos disponibles en nuestra bandeja, utiliza la función getList().

Listar Subject genera una lista de los subjects de los correos de nuestra bandeja, utiliza la función getStat() y getSubject() repetidas veces.

Leer Mail nos muestra el mail solicitado según numero, utiliza la función getMail().

Leer Subject nos muestra el subject del mail solicitado según numero, utiliza la función getSubject()

Leer Ultimo Mail nos muestra el ultimo mail recibido en nuestra bandeja, utiliza la función getMail() y getStat().

Leer Ultimo Subject nos muestra el subject del ultimo mail recibido en nuestra bandeja, utiliza la función getSubject() y getStat().

Borrar nos permite borrar un correo de nuestra bandeja indicando el numero del correo, utiliza la función delete();

Consultar Ultimo genera una consulta a la base de datos con el patrón indicado en el subject del ultimo mail recibido en la bandeja. Utiliza la función selectFromTable().

Consultar Especifico genera una consulta a la base de datos con el patrón indicado en el subject del número de mail indicado. Utiliza la función selectFromTable().

Código de la implementación en formulario:

package Presentacion;

import Datos.DB;

import Datos.POP3;

import Datos.SMTP;

import java.awt.HeadlessException;

import java.io.\*;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class Main extends javax.swing.JFrame {

SMTP smail;

POP3 rmail;

DB base;

boolean b, s, r;

public Main() {

initComponents();

smail = new SMTP();

rmail = new POP3();

b = s = r = false;

try {

base = new DB();

} catch (ClassNotFoundException | SQLException e) {

System.out.println(e.getMessage());

}

}

@SuppressWarnings("unchecked")

private void initComponents() {…}

private void SConnectActionPerformed(java.awt.event.ActionEvent evt) {

try {

smail.connect();

smail.logIn();

s = true;

} catch (IOException e) {

System.out.println(e.getMessage());

}

}

private void SDisconnectActionPerformed(java.awt.event.ActionEvent evt) {

try {

smail.logOut();

smail.close();

s = false;

} catch (IOException e) {

System.out.println(e.getMessage());

}

}

private void SSendActionPerformed(java.awt.event.ActionEvent evt) {

if (s) {

String Subject = Text1.getText();

String Data = Text2.getText();

try {

smail.sendMail(Subject, Data);

JOptionPane.showMessageDialog(rootPane, "Enviado!");

if (r) {

rmail.close();

r = false;

}

} catch (IOException e) {

System.out.println(e.getMessage());

}

} else {

JOptionPane.showMessageDialog(rootPane, "Conecte con Servidor SMTP primero");

}

}

private void PConnectActionPerformed(java.awt.event.ActionEvent evt) {

try {

rmail.connect();

rmail.logIn();

r = true;

} catch (IOException e) {

System.out.println(e.getMessage());

}

}

private void PDisconnectActionPerformed(java.awt.event.ActionEvent evt) {

try {

rmail.logOut();

rmail.close();

r = false;

} catch (IOException e) {

System.out.println(e.getMessage());

}

}

private void PListarMActionPerformed(java.awt.event.ActionEvent evt) {

if (r) {

try {

Text3.setText(rmail.getList());

} catch (IOException e) {

System.out.println(e.getMessage());

}

} else {

JOptionPane.showMessageDialog(rootPane, "Conecte con Servidor POP3 primero");

}

}

private void PLeerMActionPerformed(java.awt.event.ActionEvent evt) {

if (r) {

try {

String Mail = JOptionPane.showInputDialog("Numero de Mail:");

Text3.setText(rmail.getMail(Mail));

} catch (IOException e) {

System.out.println(e.getMessage());

}

} else {

JOptionPane.showMessageDialog(rootPane, "Conecte con Servidor POP3 primero");

}

}

private void PLeerSActionPerformed(java.awt.event.ActionEvent evt) {

if (r) {

try {

String Mail = JOptionPane.showInputDialog("Numero de Mail:");

Text3.setText(rmail.getSubject(Mail));

} catch (IOException e) {

System.out.println(e.getMessage());

}

} else {

JOptionPane.showMessageDialog(rootPane, "Conecte con Servidor POP3 primero");

}

}

private void PLeerUMActionPerformed(java.awt.event.ActionEvent evt) {

if (r) {

try {

String Mail = rmail.getStat();

Text3.setText(rmail.getMail(Mail));

} catch (IOException e) {

System.out.println(e.getMessage());

}

} else {

JOptionPane.showMessageDialog(rootPane, "Conecte con Servidor POP3 primero");

}

}

private void PLeerUSActionPerformed(java.awt.event.ActionEvent evt) {

if (r) {

try {

String Mail = rmail.getStat();

Text3.setText(rmail.getSubject(Mail));

} catch (IOException e) {

System.out.println(e.getMessage());

}

} else {

JOptionPane.showMessageDialog(rootPane, "Conecte con Servidor POP3 primero");

}

}

private void BConnectActionPerformed(java.awt.event.ActionEvent evt) {

try {

base.connect();

b = true;

} catch (SQLException ex) {

System.out.println(ex.getMessage());

}

}

private void BDisconnectActionPerformed(java.awt.event.ActionEvent evt) {

try {

base.close();

b = false;

} catch (SQLException ex) {

Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void BConsultarUActionPerformed(java.awt.event.ActionEvent evt) {

if (b && r) {

try {

String Subject = rmail.getSubject(rmail.getStat());

Text4.setText(base.selectFromTable(Subject));

} catch (IOException | SQLException e) {

System.out.println(e.getMessage());

}

} else if (b) {

JOptionPane.showMessageDialog(rootPane, "Conecte con Servidor POP3 primero");

} else {

JOptionPane.showMessageDialog(rootPane, "Conecte con Servidor SQL primero");

}

}

private void BConsultarEActionPerformed(java.awt.event.ActionEvent evt) {

if (b && r) {

try {

String Mail = JOptionPane.showInputDialog("Numero de Mail:");

String Subject = rmail.getSubject(Mail);

Text4.setText(base.selectFromTable(Subject));

} catch (HeadlessException | IOException | SQLException e) {

System.out.println(e.getMessage());

}

} else if (b) {

JOptionPane.showMessageDialog(rootPane, "Conecte con Servidor POP3 primero");

} else {

JOptionPane.showMessageDialog(rootPane, "Conecte con Servidor SQL primero");

}

}

private void PBorrarActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if (r) {

try {

String Mail = JOptionPane.showInputDialog("Numero de Mail:");

rmail.delete(Mail);

rmail.logOut();

rmail.close();

r=false;

JOptionPane.showMessageDialog(rootPane, "Exito, Conecte de nuevo");

} catch (IOException e) {

System.out.println(e.getMessage());

}

} else {

JOptionPane.showMessageDialog(rootPane, "Conecte con Servidor POP3 primero");

}

}

private void PListarSActionPerformed(java.awt.event.ActionEvent evt) {

if (r) {

try {

String s="Mail\tSubject";

int k = Integer.parseInt(rmail.getStat());

for (int i = 1; i <=k; i++) {

s=s+"\n"+i+"\t"+rmail.getSubject(String.valueOf(i));

}

Text3.setText(s);

} catch (IOException e) {

System.out.println(e.getMessage());

}

} else {

JOptionPane.showMessageDialog(rootPane, "Conecte con Servidor POP3 primero");

}

}

public static void main(String args[]) {…}

//Variables declaration - do not modify

….

// End of variables declaration

}