ISBN 978-981-11-3671-9
Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering (WCSE 2017)
Beijing, 25-27 June, 2017, pp. 762-766
doi: 10.18178/wcse.2017.06.133

Instant Messaging Application (MI-UTPL)

Liliana Enciso [1+], Juan Pachar [2], Claudio Palacios [2], Elmer Zelaya[3] and Pablo Quezada-Sarmiento[4]

[1, 2, 3, 4] Universidad Técnica Particular de Loja, Departamento de Ciencias de la Computación y Electrónica,
[1] Grupo Investigación IoT Network Communications on Smart Environment,
[1, 4] Grupo Innovación Tecnológica Educativa INNOEDUCA

Abstract. The extra-class activities that teachers expose to their students are of vital importance for the technical-scientific development of themselves, starting from this context; in the present work will be developed a service for that students of the Technical University of Loja to make their group activities have of an instant messaging tool and of this way can communicate without having to use the social networks. To carry out the application, we used Client-Server architecture, which was built with free-use development tools such as Java using the Sockets services to communicate the different processes through the network. To fulfill the project, a search of similar applications was made, to see its feasibility, but its use was granted to certain places and countries so that, we decided that the application was feasible. For the fulfillment of the project was employed a methodology in cascade; this will allow to divide the project in phases and at the same time develop it of an orderly and concise manner. In the results, we got a tool of Instant Messaging where students who perform their works and tasks in areas of the university can communicate of fast, effective, safe way, and without necessity to find distractions like in typical and social applications.

Keywords: Network, Java, Chat, Socket, Client-Server.

1. Introduction

Academic assignments are the class events that provide opportunities for students can use their “multiple” cognitive and motivational resources to the service of the achieving of personal and educational goals [1]. Nowadays, the term social network is very used calling so to the different sites or websites that offer to register people and contact with many individuals in order to share contents, interact and create communities about similar interests [2][3][4], these influence in positive manner when they are used with modesty and moderation; the problem take root when the student has an abuse of this, where the whole day the learner is in front the computer [5]. Interpersonal relationships are in constant transformation in the daily life of the human being, and in this transformation have affected the new technological applications generating changes over interpersonal communication [6][7][8], digital communication has caused formal changes in the communicative genres and material among interpersonal relationships [9]. Was found that those subjects who made an excessive use of the Internet, presented associated symptomatology, obsessive compulsive, a greater interpersonal sensitivity, somatization, anxiety and hostility [10].

Information security, is another area of importance at the moment to communicate with other people, social networks, e-mails and chat rooms of instant messaging which have shown their vulnerabilities becoming visible to third parties, recognized media such as The Daily and RT revealed that only in 2014 More than five million users and passwords came to light associated to Gmail mail services [11]. Taking into account this background, which aim help the students who are trained in this prestigious educational institution by providing them with an application that allows fast, secure and effective communication between different individuals that develop the task entrusted. Instant messaging application (MI-UTPL) will

[+] Corresponding author. Tel.: +593-7-3701444; fax: +000-0-0000000.
E-mail address: lenciso@utpl.edu.ec

be developed in a robust language, multiplatform and compiled as it is JAVA, which will permit the use of Sockets to transmit the different processes via network. In addition, will be design a friendly interface.

2. Related Works

Actually the instant messaging applications are developed under a great variety of protocols, some of them are owners, and for this reason they do not offer documentation or give access to their sources, on the other hand, there are other free and are documented for the disposition of all users. These protocols are used by the clients such as MSN Messenger Touch[12]. IRC is a communication protocol in real-time based on text, which allows the conference between 2 or more people and is classified inside of the instant messaging [13][14]. According [15], the OSCAR protocol is a protocol of proprietary development and does not provide documentation or code. To know its way to act and adapt its programs there are developers who have employed inverse engineering. The clients do not connect directly to each other; they do it through servers who take responsible of delivering the messages to their receivers [15][16]. XMPP is an open and extensible protocol. This protocol in its topological operation is based on the classic architecture client-server, although it does not force it to do so and using TLS, allows encrypting messages using different algorithms such as RSA and DSS [17]. The most used instant messaging clients are Whatsapp, Facebook Messenger, Viber, Line, We-Chat, Telegram and Google Talk (using the XMPP open protocol). Lately have appeared some messaging clients that offer the possibility to connect to several networks at the same time privately and without any type of tracking [18].

3. Theoretical Framework

Chat: It is more comfortable ask information by Chat instead of call by telephone, in the chat, writing requires a fluency what make that in many occasions the speakers prefer leaving rules of grammar and orthography in search to get a greater efficiency communicative [19].

Client-Server Architecture: This technology allows the creation of distributed applications, the main advantage of this Architecture is that it facilitates the separation of functions according to their service[20]. The client-server architecture allows the creation of distributed applications, implies the existence of a relation among processes that request services (clients) and processes that respond to these services (servers)[21][22].

Java: It is a programming language of high level, in its great part Java syntax is influenced by the popular language C and by Visual Basic, although it take away some low-level features that has C as are memory management and the appointers (pointers) [23]. Java has all the characteristics of the paradigm of programming orientated to objects, inheritance, encapsulation, abstraction, polymorphism, modularity among another [24]. For the development of the application has used version 8 Java [25].

Sockets: Sockets are basically forms in which we can interconnect 2 or more programs through use of the Internet[25][26]. Apache HTTP Server: Known as Apache, it is a highly configurable server of modular design, authentication of different types, personalized answers in front of server, creation of dynamic content, high configurable in the creation and management of logs, great scalability and content negotiation [26].

4. Methodology

In order to carry out our project we have thought it convenient to use a Cascade methodology, which allows us to will develop our project in a set of phases of structured and orderly manner.

4.1. Design

In the design part will develop the application modeling, creation of the sockets as in the client-server connection that we are going to use (see Figure 1). In the design part will develop the application modeling, in the classes and the aspect of the application. The project will have two packages: Client and server with their corresponding classes according to the diagram above. In the diagram the methods that come with the same name of the class are the builders. Then, Size that implements and extends each class: Server: ClientThread implements Runnable, ListDaTaListener for Client: ClientController implements ActionListener. Runnable because required to be threads (Threads), ListDataListener to listen the events that chains the application when it receives a new message and so can show it later, and ActionListener to know when the user of the client wishes to send a message and thus can send it. In addition, we did decide to design a logo of the application with modern tones to create a visual impact in the users (see figure 2).

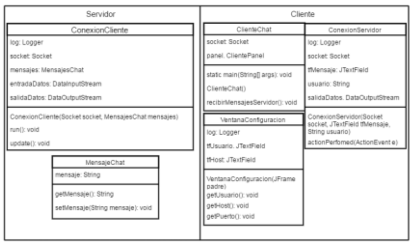


Fig. 1: Class diagram of the client-server connection through sockets.

Fig. 2: Logo Mi-UTPL.

5. Results

In this stage was done the modelling as in design of the interface as in the classes, that is to say are created the different methods of the classes and encode the application implementing the Client-Server architecture, in Figure 3 shows the relation of the application and its communication via sockets.

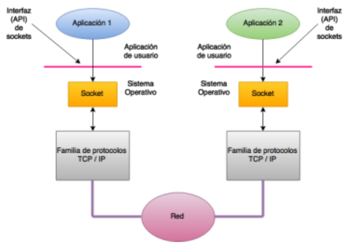
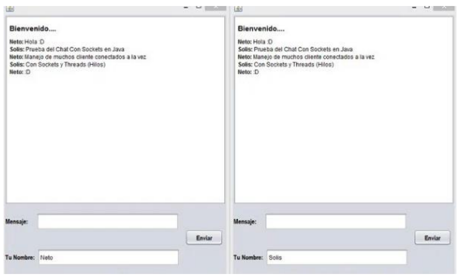


Fig. 3: Communication among sockets.


Bienvenido....
Neto: Hola :D
Solis: Prueba del Chat Con Sockets en Java
Neto: Manejo de muchos cliente conectados a la vez
Solis: Con Sockets y Threads (Hilos)
Neto: :D

Mensaje:
Tu Nombre: Neto

Bienvenido....
Neto: Hola :D
Solis: Prueba del Chat Con Sockets en Java
Neto: Manejo de muchos cliente conectados a la vez
Solis: Con Sockets y Threads (Hilos)
Neto: :D

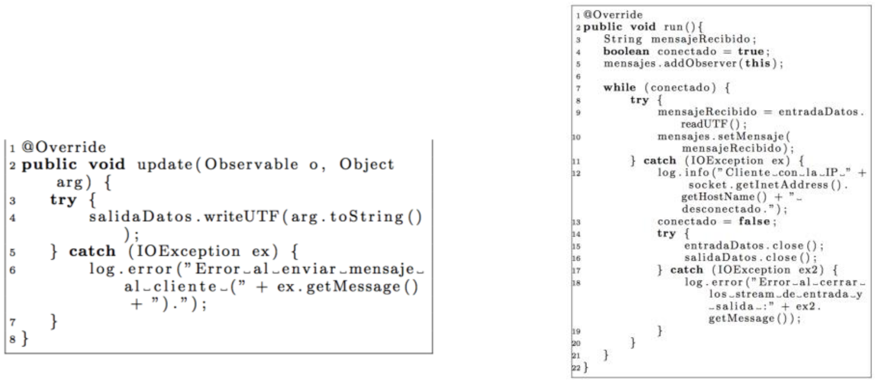
Mensaje:
Tu Nombre: Solis

Fig. 4: Communication via sockets including ports.

5.1. Development

The application will be structured in two parts: a client and a server. The server application will be written in the Java programming language, and will be in charge, among other things, of management of the access components to real data such as the following:

Code 1: Run method that inherits from Thread, this is responsible for receiving client messages, and update method is responsible for sending the message to the client. In line 2 the run method points to the list of observers messages so that later the messages contains a new message, it be notified and can send the message to the client thanks to the update method, regardless of whether the message the client has been sent, that is attended by the same thread or by another different (see figure 5).


```
1 @Override
2 public void update(Observable o, Object
      arg) {
3     try {
4         salidaDatos.writeUTF(arg.toString()
          );
5     } catch (IOException ex) {
6         log.error("Error␣al␣enviar␣mensaje␣
              al␣cliente␣(" + ex.getMessage()
               + ").");
7     }
8 }
```

```
1 @Override
2 public void run(){
3     String mensajeRecibido;
4     boolean conectado = true;
5     mensajes.addObserver(this);
6
7     while (conectado) {
8         try {
9             mensajeRecibido = entradaDatos.
                  readUTF();
10            mensajes.setMensaje(
                  mensajeRecibido);
11        } catch (IOException ex) {
12            log.info("Cliente␣con␣la␣IP␣" +
                   socket.getInetAddress().
                  getHostName() + "␣
                  desconectado.");
13            conectado = false;
14            try {
15                entradaDatos.close();
16                salidaDatos.close();
17            } catch (IOException ex2) {
18                log.error("Error␣al␣cerrar␣
                      los␣stream␣de␣entrada␣y
                      ␣salida␣:" + ex2.
                      getMessage());
19            }
20        }
21    }
22 }
```
Fig. 5: Code1 - Run and Main Method.

Code 2: Method that allows the connection with the server. In line 1, the Server Connection class that implements ActionListener and this class is calling so by pressing the button send, simply initialize a DataOutputStream in the constructor to can receive data in action Performed, send the text to the chat server preceded by the name of the user to then be shown in the chat and delete the content of the text field so that be possible write the following message without having to delete it manually, the lines 12 to 19 create and capture the different exceptions that might arise in the execution(see figure 6).

```
1 public class ConexionServidor implements
      ActionListener {
2     private Logger log = Logger.getLogger(
          ConexionServidor.class);
3     private Socket socket;
4     private JTextField tfMensaje;
5     private String usuario;
6     private DataOutputStream salidaDatos;
7
8     public ConexionServidor(Socket socket,
          JTextField tfMensaje, String
          usuario) {
9         this.socket = socket;
10        this.tfMensaje = tfMensaje;
11        this.usuario = usuario;
12        try {
13            this.salidaDatos = new
                  DataOutputStream(socket.
                  getOutputStream());
14        } catch (IOException ex) {
15            log.error("Error␣al␣crear␣el␣
                  stream␣de␣salida␣:␣" + ex.
                  getMessage());
16        } catch (NullPointerException ex) {
17            log.error("El␣socket␣no␣se␣creo
                  ␣correctamente.␣");
18        }
19    }
20
21    @Override
22    public void actionPerformed(ActionEvent
          e) {
23        try {
24            salidaDatos.writeUTF(usuario +
                  ":␣" + tfMensaje.getText()
                  );
25            tfMensaje.setText("");
26        } catch (IOException ex) {
27            log.error("Error␣al␣intentar␣
                  enviar␣un␣mensaje:␣" + ex.
                  getMessage());
28        }
29    }
30 }
```

Fig. 6: Code 2 - Method that allows the connection with the server.

5.2. Implementation

In the implementation the application is copied in a new computer and will begin to perform the respective functional samples, possible failures of the application. It was carried out the implementation of Client-Server Architecture and the lock libraries and feels integration in order to provide a user dynamic interface.

The application has the ability to extend or adapt to other areas. For better application performance, you should inactivate the firewall of the connected hosts. In the table 1 can show the results obtained performing the communication between the client and the server.

Table 1. Validation of the Application

	Distance	Time
Client-Server	0cm	2.345 ms
	1m	1s
	3m	3s
	5m	The message does not arrive

6. Conclusion

When analyzing, developing, coding and implementing the application, the team got the following conclusions:

- The application that was built helps improve communication among the different individuals of a study group that wishes to carry out their activities.
- It was possible to obtain all the information about the used tools.
- The tools with which the application was executed carried out the expectations for which they were chosen.
- The development team considers that knowledge acquired during this project is important and increase coding practice.

7. References

[1] P. Winne, R. Marx. A cognitive-processing analysis of motivation within classroom tasks. AcademicPress. 1989.

[2] J.D. Rivera. Revista de universidad y sociedad del conocimiento. Edicio Statys. 2007.

[3] L. Enciso, P. Quezada, J. Fernandez, B. Figueroa, V. Espinoza. Analysis of performance of the routing protocols ad hoc using random waypoint mobility model applied to an urban environment (2016) WEBIST 2016 - Proceedings of the 12th International Conference on Web Information Systems and Technologies, 1, pp. 208-213.

[4] L.E. Quispe, R.T. Tandazo, L.M. Galan. Analysis of Ad Hoc routing protocols for emergency and rescue scenarios (2012) WEBIST 2012 - Proceedings of the 8th International Conference on Web Information Systems and Technologies, pp. 781-786.

[5] J. Calzada-Morales. La evaluación educativa. Asociación Oaxaqueña de Psicología. Calzada Madero. 2011.

[6] M. Tapia. Redes sociales y relaciones interpersonales en Internet. Universidad Nacional de San Luis. 2012

[7] L.E. Quispe, L.M. Galan. Behavior of Ad Hoc routing protocols, analyzed for emergency and rescue scenarios, on a real urban area, Expert Systems with Applications, Volume 41, Issue 5, April 2014, Pages 2565-2573, ISSN 0957-4174.

[8] L.E. Quispe, L.M. Galan. Analysis of QoS parameter in AODV a DSR in mobile Ad Hoc networks (2012) WEBIST 2012 - Proceedings of the 8th International Conference on Web Information Systems and Technologies, pp. 276-279.

[9] G.R. Laborda. (2005). Tecnologías, redes y comunicación interpersonal. estudiantes universitarios de Santiago de Chile.

[10] CK.Yang. Sociopsychiatric characteristics of adolescents who use computers to excess. Psychiatrica Scandinavica. US National Library of Medicine. 2001.

[11] M. Loaysa. Hackers al asecho. Editorial revista ciencia. 2015.

[12] R. Nimmelapelli. Implementation of a sip message processor. Computer Engineering. 2006.

[13] Z. Wang, X. Jiang, W. Cui, X. Wang, M. Grace. Automatic reverse engineering of encrypted messages. 2009.

[14] J.C. Tourino. Communication protocols on high-speed. IEEE. 2011.

[15] J.M. De Torres Encinas. Protocolos para la mensajería instantánea. 2012. Descargado de: https://sisdtor.wordpress.com/2012/02/08/protocolos-para-la-mensajeria-instantanea/

[16] P. A. Quezada-Sarmiento, L. E. Enciso-Quispe, J. Garbajosa and H. Washizaki, "Curricular design based in bodies of knowledge: Engineering education for the innovation and the industry," 2016 SAI Computing Conference (SAI), London, 2016, pp. 843-849.

[17] J.C.D. García. Estudio del protocolo xmpp de mensajería instantánea, de sus antecedentes, y de sus aplicaciones civiles y militares. 2008.

[18] V. Palomo. Las aplicaciones de mensajería instantánea más famosas. 2015.

[19] D.S. Levis. El chat: El habla escrita de niños y jóvenes. Ponencia presentada en ALAIC. 2006.

[20] L. Enciso, P. Quezada, L. Barba-Guamán, L. Solano, P.-P. Alarcón. Open drugstores mobile app (2016a). Advances in Intelligent Systems and Computing, 444, pages 797-806. Springer International Publishing, Cham.

[21] P. Quezada, L. Enciso, J, Garbajosa. Using tools of cloud computing for defining, planning, monitoring innovation projects and knowledge generation [Uso de herramientas de computación en la nube para definir, planificar, controlar proyectos de innovación y generación de conocimiento] (2015) XI Jornadas Iberoamericanas de Ingenieria de Software e Ingenieria del Conocimiento, JIISIC 2015, pp. 121-131.

[22] L. E. Quispe, L. M. Galan, "Assessment of Throughput Performance under NS2 in Mobile Ad Hoc Networks (MANETs)," 2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks, Madrid, 2013, pp. 338-343.

[23] K. Sowizra. The java 3d api specification. editorial Inc Usa. 2010.

[24] P.D. Deitel. Como programar en java. Pearson Education. 2012.

[25] Java 8. 2017. Descargado de: https://www.java.com/es/download/faq/java8.xml.

[26] E. Q. Catalinas, A. G. Román, J.P. Rodríguez. Mantenimiento de portales de la información: explotación de sistemas informáticos. (2007). Editorial Paraninfo.