Mensajería cliente-servidor aplicando sockets en las herramientas GEANY IDE 1.31, PHYTON 3.7 y POSTGRESQL 9.5 en el sistema operativo CENTOS 7

Loja Mora Nancy Magaly[1], Molina Ríos Jimmy Rolando[1] , Morocho Román Rodrigo Fernando[1] , Porras Suriaga Celmira Marcela [1]

*Universidad Técnica de Machala, Facultad de Ingeniería Civil, Machala, Ecuador
e-mail: nmloja, jmolina, rmorocho, cmporras_est {@utmachala.edu.ec}

RESUMEN:

En el presente trabajo se muestra un caso práctico de mensajería instantánea cliente-servidor mediante la instalación y configuración de las herramientas open-source Geany 1.31, Python 3.7 y PostgreSQL 9.5 para Linux, desarrollando una aplicación sencilla pero funcional entre dos sistemas operativos mediante sockets, donde se evidencia un protocolo de comunicación, destacando las características más relevantes de esta arquitectura, funcionalidad y características más relevantes.

Con los resultados del desarrollo de esta aplicación se pretende evidenciar una posible solución ante la deficiencia de una conexión a internet en la transferencia de diferentes archivos ya sean de tipo textual, imágenes, videos o documentos, que quedará como un trabajo futuro que s epoda implementar con interfaz gráfica y transferencia de archivos, pero para este caso dem0ostrativo será solo por la con sola de los dos sistemas operativos.

Palabras clave: Cliente-Servidor, Software Libre, Transferencia, Aplicación, Mensajería

ABSTRACT:

This work shows a practical case of client-server instant messaging through the installation and configuration of the open source tools Geany 1.31, Python 3.7 and PostgreSQL 9.5 for Linux, a simple but functional application between operating systems using sockets, where evidence of a communication protocol, highlighting the most relevant features of this architecture, functionality and most relevant features.

With the results of the development of this application, it is sought to show a possible solution to the deficiency of an internet connection in the transfer of different files, whether textual, images, videos or documents, which will remain as a future work that can be implemented with graphical interface and file transfer, but for this case it will only be the only one of the two operating systems.

Keywords: Client-Server, Free Software, Transfer, Application, Messaging

Introducción

En la actualidad, las redes de comunicación han evolucionado en todos los ámbitos afectando la forma de comunicarnos, sobretodo porque nos permite estar permanentemente conectados e informados de los últimos acontecimientos, cabe destacar el conjunto de protocolos de comunicación de computadoras se destaca TCP/IP que soporta aplicaciones de red, servidores web, correo electrónico, transferencia de archivos, entre otros (Rodríguez Erazo, 2014). La arquitectura TCP/IP está formada por cuatro capas: capa de red, de acceso a la red, de transporte y aplicación (Zafra, Gibaja, Luque, & Ventura, 2014).

Uno de los problemas latentes que se generan por la deficiencia de internet es la transferencia de archivos, lo que nos permite pensar en una aplicación que permita el incremento de la transferencia, basados en antecedentes con el uso del Modelo Cliente/Servidor estableciendo la comunicación entre dos máquinas y/o sistemas operativos se logra utilizando diferentes técnicas (Acosta Gonzaga, Álvarez Cedillo , & Gordillo Mejia, 2015), una de ellas son los sockets, que consiste en configurar una red cliente-servidor para establecer flujo de información entre transmisor y receptor (Murillo Morera & Caamaño Polini, 2015).

¿Qué es un Modelo Cliente-Servidor?

“La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes” (Sepulveda Ibañez, 2014)

En el modelo cliente servidor, un cliente envía un mensaje solicitando un determinado servicio a un servidor y este envía uno o varios mensajes con la respuesta, en un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el de cliente para otras.(Ver Fig.1) (Jijón , 2017)

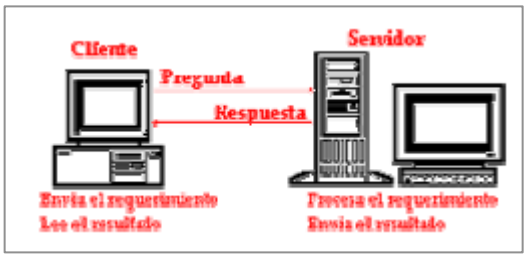


Fig. 1: Modelo Cliente/Servidor

Entonces, los procesos de cliente interactúan con los procesos de servidor individuales en equipos anfitriones que se encuentran separados con el objetivo de que se pueda acceder a los recursos que estos administran.

Cliente

“Es el que permite al usuario formular los requerimientos y pasarlos al servidor” (López Fuentes, 2015) ,donde el cliente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, desarrollados en plataformas que permiten construir interfaces graficas de usuario, accediendo a los servicios en cualquier parte de una red.

Las funciones que destacan son la administración de interfaz de usuario, interactuar con el usuario, procesar la lógica de la aplicación y hacer validaciones locales, generar requerimientos de base de datos y formatear resultados (Diez de la Puente, Valdivia Martínez, & Zhu, 2017).

Servidor

“Es el encargado de atender múltiples clientes que hacen peticiones de algún recurso administrado por él, normalmente maneja todas las funciones

relacionadas con la mayoría de las reglas de negocio y los recursos de los datos" (Luján Mora, 2013)

Las funciones que lleva a cabo el proceso servidor son aceptar los requerimientos de bases de datos que hacen los clientes, procesar requerimientos de bases de datos, formatear datos para trasmitirlos a los clientes y procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos (Combita Niño & Ariza Colpas, 2015).

Características de la Arquitectura Cliente/Servidor

Las características básicas de una arquitectura Cliente/Servidor son (Batini & Scannapieca, 2018):

- Combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.
- Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco e input-output devices.
- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- Existe una clara distinción de funciones basada en el concepto de "servicio", que se establece entre clientes y servidores.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.
- Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.
- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio (Ardilla Funeque, 2014).

Transferencia de Archivos

"La transferencia de archivos en un entorno de red implica un conjunto de reglas y procedimientos para que se entiendan las partes implicadas en la transferencia y

se pueda realizar el envío de manera satisfactoria" (Murillo Morera & Caamaño Polini, 2015), en un modelo cliente-servidor resulta necesario que un protocolo sea el encargado de regular la transferencia del archivo desde un punto a otro.

Sockets

"Es una interfaz de entrada-salida de datos que permite la intercomunicación entre procesos ejecutándose en el mismo o en distintos sistemas, unidos mediante una red" (Pereira, 2015).

Es decir, un socket es un mecanismo con que se comunican procesos para intercambiar información de forma bidireccional entre dos dispositivos o sistemas.

Mensajería

"Es un punto intermedio entre los sistemas de chat y correo electrónico, basado en texto. El texto es enviado a través de dispositivos conectados ya sea a una red o a Internet sin importar la distancia que exista (Universidad de Salamanca, 2015)".

Materiales y Métodos

Los materiales que se emplean para realizar la mensajería cliente/servidor como caso son:

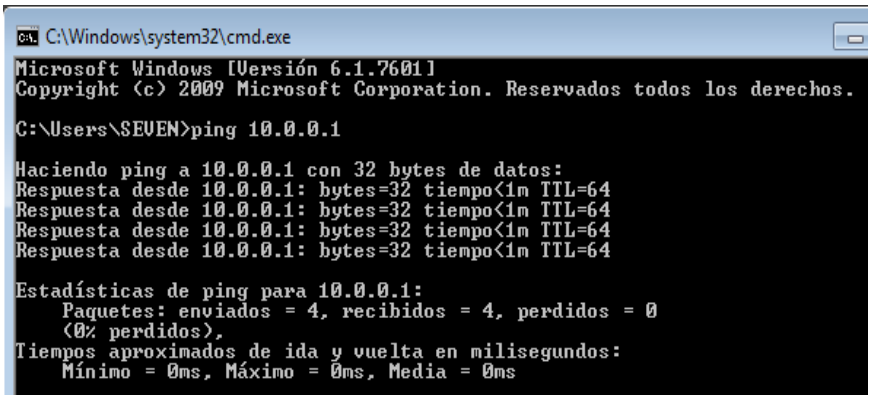
- Una computadora con dos sistemas operativos Windows y Centos 7.
- Editor de Texto Geany IDE 1.31
- Lenguaje de Programación Phyton3.7
- Base de Datos PostgreSQL 9.5

La Metodología empleada para la realización de la mensajería cliente/servidor entre los dos sistemas operativos es la metodología RAD que es un proceso de desarrollo de software, que comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE, también engloba la usabilidad, utilidad y rapidez de la ejecución (Egas & Jativa, 2016).

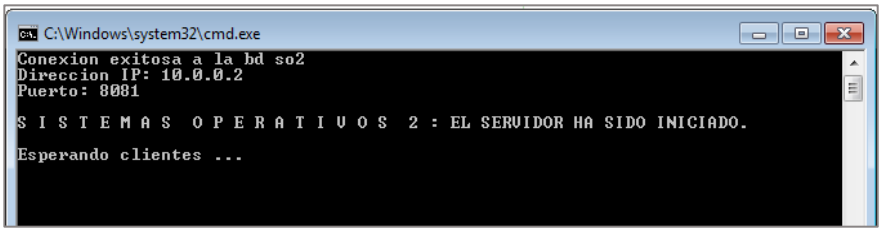
- Fase de Planificación de los Requisitos: identificación de la solución para el caso de estudio, además se instala y configura los programas necesarios.
- Fase de Diseño: se establece y comprueba la conexión TCP/IP entre el cliente y servidor que en este caso son los dos sistemas operativos.
- Fase de Construcción: se desarrolla el código de la aplicación de mensajería en el IDE de Geany.
- Fase de Implementación: se establece una conexión con la base de datos y se realizan pruebas de funcionamiento.

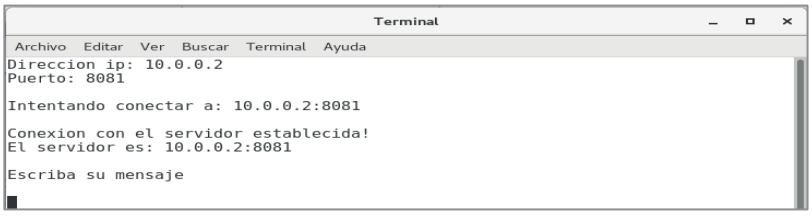
Discusión de Resultados

Una vez desarrollado el código para el cliente y el servidor se deben realizar las respectivas pruebas de funcionamiento del mismo, configurando en primera instancia la red entre el cliente y el servidor en los dos sistemas operativos y comprobar la existencia de conexión entre las mismas.

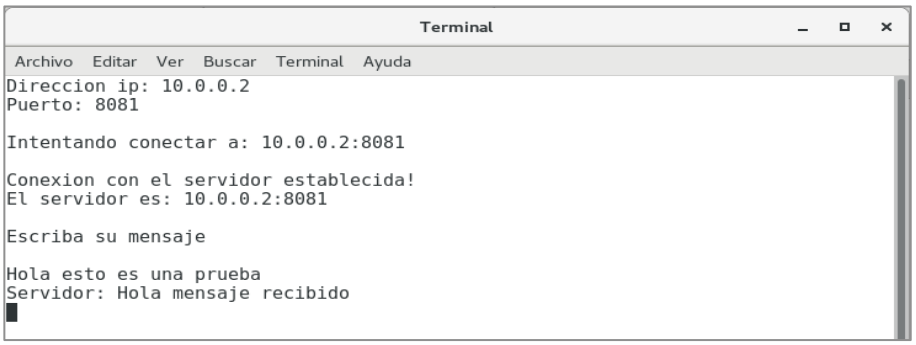


Se compila el código de cliente y servidor y mediante la dirección y puerto se establece la conexión para iniciar con la mensajería instantánea entre los sistemas operativos.

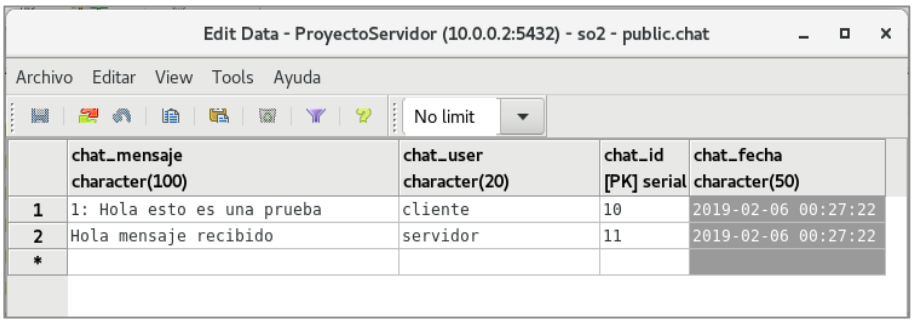




Se escribe un mensaje desde el sistema cliente y automáticamente llega el mensaje al sistema servidor a través de la consola de los mismos, donde puede responder instantáneamente.



Como un valor agregado a este caso práctico, todos los datos de esta mensajería se guardan en una base de datos.



	chat_mensaje character(100)	chat_user character(20)	chat_id [PK] serial	chat_fecha character(50)
1	1: Hola esto es una prueba	cliente	10	2019-02-06 00:27:22
2	Hola mensaje recibido	servidor	11	2019-02-06 00:27:22
*				

Conclusiones

- El desarrollo de una aplicación de mensajería cliente-servidor entre dos sistemas operativos nos permitió la experimentación con tecnologías básicas como los sockets, también se adquirieron nuevos conocimientos en el diseño y desarrollo de aplicaciones que pueden utilizarse en Internet siguiendo el modelo cliente-servidor.

- Además cabe recalcar que se logró el aprendizaje de la configuración y administración de herramientas open-source en el sistema operático Centos7.

- Al realizar las pruebas funcionamiento de la aplicación por consola, se evidencia la funcionalidad del protocolo y desarrollo mediante el lenguaje de programación Python y los mensajes almacenados en la base de datos postgresql.

Referencias Bibliográficas

Acosta Gonzaga, E., Álvarez Cedillo , J., & Gordillo Mejia, A. (2015). Arquitecturas en n-Capas: Un Sistema Adaptivo. *Polibits*.

Ardilla Funeque, H. J. (2014). Implementación Ciente-Servidor mediante Sockets. *Vinculos*, 48-59.

Batini, C., & Scannapieca, M. (2018). *Data-Centric Systems and Applications.* Milano: Board.

Combita Niño, H., & Ariza Colpas, P. P. (2015). Análisis y Desarrollo de un software web para la gestión y fomento de la investigación en instituciones de educación superior en Colombia. *Ingenium*.

Diez de la Puente, P., Valdivia Martínez, D., & Zhu, S. (2017). *Desarrollo de una Aplicación Cliente-Servidor para analisis de esperiencias de usuario en realidad virtual con seguimiento de movimiento del visor y reconocimiento gestual de las manos.* Madrid: Matritensis.

Egas, L., & Jativa, J. (2016). Evolución de la Metodologías de Desarrollo de la Ingeniería de Software en el proceso la Ingeniería de Sistemas Software y Determinación de una metodología adaptable orientada a una organización pequeña. *EcuCiencia*.

Jijón , N. (2017). *Modelados de Sistemas.* Lima.

López Fuentes, F. d. (2015). *Sistemas Distribuidos.* Cuajimalpa: Metropolitana.

Luján Mora, S. (2013). *Programación en Internet.* España: Club Universitario.

Murillo Morera, J., & Caamaño Polini, S. (2015). Implementación de un Servidor FTP utilizando el modelo Cliente/Servidor mediante el uso de Sockets en Lenguaje C Unix con el fin de mejorar los tiempos de respuesta en la Red. *UNICIENCIA*, 83-89.

Pereira, A. (2015). *Programación de Sockets.*

Rodríguez Erazo, C. (2014). Diseño, Desarrollo e Implementación del Portal Web de la empresa da auto ventas "Auto Fácil", aplicando la herramienta de desarrollo web Open Source Drupal. *Universidad Técnica del Norte*.

Sepulveda Ibañez, D. (2014). *Arquitectura Cliente/Servidor.* Costa Rica: Uniciencia.

Universidad de Salamanca. (2015). *El Factor Humano como elemento clave en el Elearning: El Tutor Online.* Salamanca: Grupo de Investigación e Iteración y Learning.

Zafra, A., Gibaja, E., Luque, M., & Ventura, S. (2014). Diseño de Aplicaciones Cliente/Servidor para el aprendizaje de las tenologias de comunicación. *Iniciación a la Investigación-Revista Electrónica*, 12.