

SERVIDOR CHAT EN TARJETA NODEMCU

SECCION INVITADA SEGURIDAD DE LA INFORMACION

Erick Quezada Mosqueda, Fernando Rojas Ramos, Bárbara Emma Sánchez Rinza
Benemérita Universidad Autónoma de Puebla
Facultad de ciencias de la computación
14 sur y avenida San Claudio
Tel.,52 222 2295500
brinza@hotmail.com

RESUMEN.

En el siguiente trabajo se implementará un servidor de chat en una tarjeta NodeMCU, de esta manera dos usuarios conectados a la misma red podrán intercambiar mensajes conociendo la dirección IP y el puerto asignados al servidor.

Primero se presentará una breve introducción sobre los conceptos de servidor chat e internet de las cosas, siendo este último un concepto que ha estado ganando popularidad en los últimos años y que es parte importante de la llamada cuarta revolución industrial o Industria 4.0.

Además, se describirá la tarjeta de desarrollo que se está utilizando en este trabajo, la tarjeta NodeMCU. Esta tarjeta es de fácil adquisición, bajo costo y cuenta con un módulo WiFi (ESP8266) lo que la hace una herramienta muy útil para proyectos de internet de las cosas.

Después se presentará el desarrollo del proyecto iniciando con la instalación de los controladores necesarios para conectar la tarjeta nodeMCU a una computadora, después se mostrará la IDE a utilizar y se presentará un diagrama de flujo describiendo el funcionamiento del sistema.

Para finalizar se presentarán los resultados obtenidos, así como las conclusiones que se obtuvieron de la realización del proyecto.

Palabras Clave: NodeMCU, WiFi (ESP8266), Java, COBOL, Python.

ABSTRACT.

In the following work a chat server will be implemented in a NodeMCU card, in this way two users connected to the same network will be able to exchange messages knowing the IP address and the port assigned to the server.

First a brief introduction will be presented on the concepts of chat server and internet of things, the latter being a concept that has been gaining popularity in recent years and is an important part of the so-called fourth industrial revolution or Industry 4.0.

In addition, the development card that is being used in this work, the NodeMCU card, will be described. This card is easy to acquire, low cost and has a WiFi module (ESP8266) which makes it a very useful tool for internet projects of things.

Then the project development will be presented, starting with the installation of the necessary drivers to connect the nodeMCU card to a computer, then the IDE to be used will be shown and a flow diagram will be presented describing the functioning of the system.

Finally, the results obtained will be presented, as well as the conclusions that were obtained from the realization of the project.

Keywords: NodeMCU, WiFi (ESP8266), Java, COBOL, Python.

1. INTRODUCCIÓN

Un servidor de chat se refiere a una computadora que está dedicada a brindar poder de procesamiento para conectar y mantener la comunicación entre sus clientes. Algunos ejemplos de servidores de chat actuales son: IRCd, UnrealIRCd, InspIRCd y Openfire [1].

La tarjeta NodeMCU (ver Figura. 1) es una plataforma IoT (Internet de las cosas) de código abierto. Incluye el firmware que se ejecuta en el SoC Wi-Fi ESP8266 de Espressif Systems y el hardware que se basa en el módulo ESP-12. El término "NodeMCU" se refiere al firmware en lugar de a los kits de desarrollo. El firmware utiliza el lenguaje Lua. Se basa en el proyecto eLua y se basa en el SDK no operativo de Espressif para el ESP8266. Utiliza muchos proyectos de código abierto, como lua-cjson, y spiffs.

El internet de las cosas (en inglés, Internet of Things, abreviado IoT) es un concepto que se refiere a una interconexión digital de objetos cotidianos con internet. El concepto de internet de las cosas fue propuesto en 1999, por Kevin Ashton, en el Auto-ID Center del MIT, en donde se realizaban investigaciones en el campo de la identificación por radiofrecuencia en red (RFID) y tecnologías de sensores. El internet de las cosas debería codificar, en teoría, de 50 a 100 000 millones de objetos y seguir el movimiento de estos. Se calcula que todo ser humano está rodeado, al menos, por un total de aproximadamente 1000 a 5000 objetos. Con la próxima generación de aplicaciones de internet (protocolo IPv6) se podrían identificar todos los objetos, algo que no se podía hacer con IPv4. Este sistema sería capaz de identificar instantáneamente por medio de un código a cualquier tipo de objeto.

En este trabajo se aplica el concepto de IoT al implementar un servidor de chat en una plataforma pequeña, de bajo costo y con conexión a Internet. Es importante señalar la utilidad de tener un sistema con estas características, nos permite mantener

comunicación con distintos equipos de cómputo si así lo deseamos.



Figura 1. Tarjeta nodeMCU

2. DESARROLLO

Lo primero que se realizó fue la instalación de los controladores de la tarjeta NodeMCU en una computadora ver figura 2. Esto es necesario ya que desde ahí se programa la tarjeta. Para ello lo primero que se debe realizar es descargar la IDE de Arduino. Arduino es una tarjeta de adquisición de datos similar a la NodeMCU por lo que su interfaz de desarrollo sirve para programar ambas tarjetas.

Una vez descargada la IDE de Arduino se deben instalar los controladores para la NodeMCU, para ello basta solo con instalar los controladores de la tarjeta de red de la tarjeta: ESP8266. Esto se realizó desde la interfaz de Arduino [4].

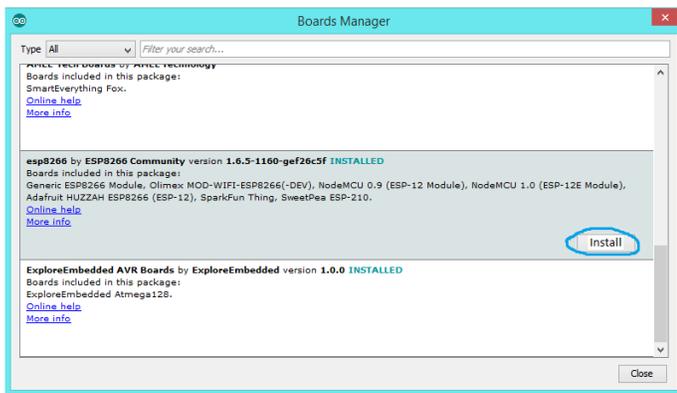


Figura 2. Instalación de controladores del módulo ESP8266

A continuación, se escribió el código que permitirá la implantación del servidor chat en la NodeMCU. Es importante señalar que la programación tanto de la tarjeta NodeMCU como de Arduino está basada en C, además la IDE contiene varias librerías precargadas que facilitan el desarrollo de proyectos. En este caso utilizaremos la librería <ESP8266WiFi.h>.

A continuación, se muestra un diagrama de flujo indicando el funcionamiento del sistema [2], ver figura 3.

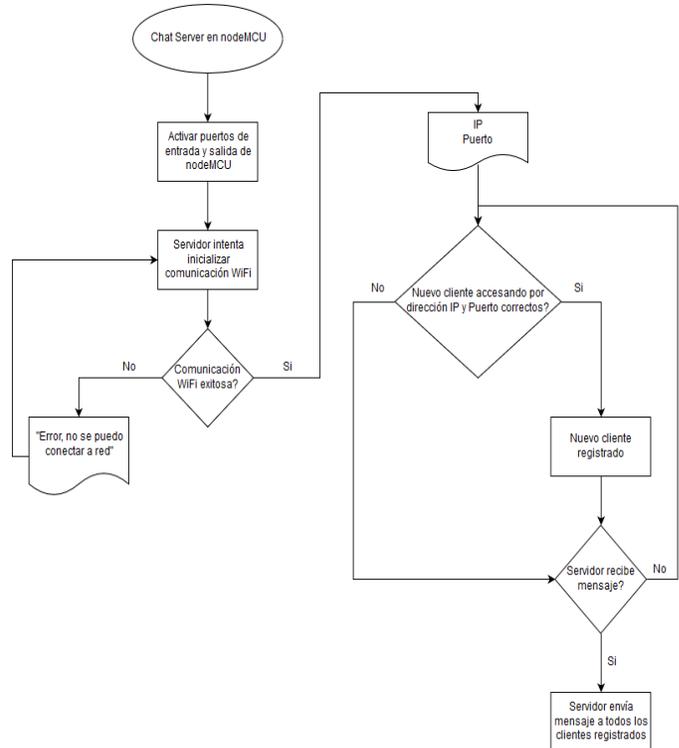


Figura 3. Diagrama de flujo del sistema

Una vez escrito el código se carga el programa en tarjeta NodeMCU en donde podemos observar en el monitor serial la dirección IP y el puerto asignados a la tarjeta, ver figura 4. Esta información es necesaria para conectarnos al servidor. También es necesario señalar que en esta implementación de servidor chat tanto el servidor como los clientes deben de estar conectados a la misma red.

```
Connected to INFINITUM3567
IP address: 192.168.1.113
Port: 5000
```

Figura. 4. Red, Dirección IP y puerto asignados a la tarjeta NodeMCU.

Para conectarnos al servidor es necesario bajar en los equipos cliente una aplicación llamada PuTTY [3], ver figura 5.

PuTTY es un cliente SSH, Telnet, rlogin, y TCP raw con licencia libre. Algunas características que son:

- El almacenamiento de hosts y preferencias para uso posterior.
- Control sobre la clave de cifrado SSH y la versión de protocolo.
- Clientes de línea de comandos SCP y SFTP, llamados "pscp" y "psftp" respectivamente.
- Control sobre el redireccionamiento de puertos con SSH, incluyendo manejo empotrado de reenvío X11.
- Completos emuladores de terminal xterm, VT102, y ECMA-48.
- Soporte IPv6.
- Soporte 3DES, AES, RC4, Blowfish, DES.
- Soporte de autenticación de clave pública.
- Soporte para conexiones de puerto serie local.

El nombre PuTTY proviene de las siglas Pu: Port unique TTY: terminal type. Su traducción al castellano sería: Puerto único de tipo terminal.

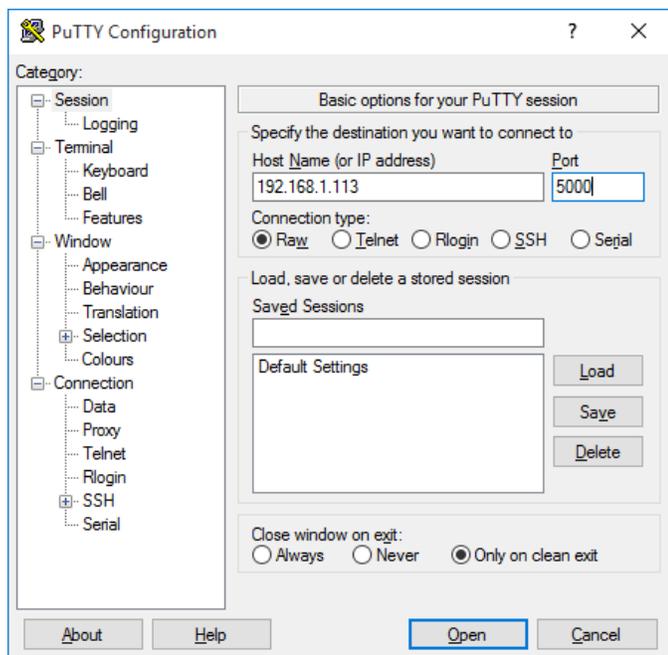


Figura 5. Aplicación PuTTY con la información de la tarjeta NodeMCU

Dando click en open se abrirá una interfaz estilo ventana de comandos en donde se visualizarán los mensajes enviados por todos los clientes del servidor.

3. RESULTADOS

A continuación, se muestra la interfaz de PuTTY con los mensajes emitidos por dos computadoras cliente conectadas al servidor chat en la NodeMCU. La ventana luce igual para ambos clientes, ver figura 6.

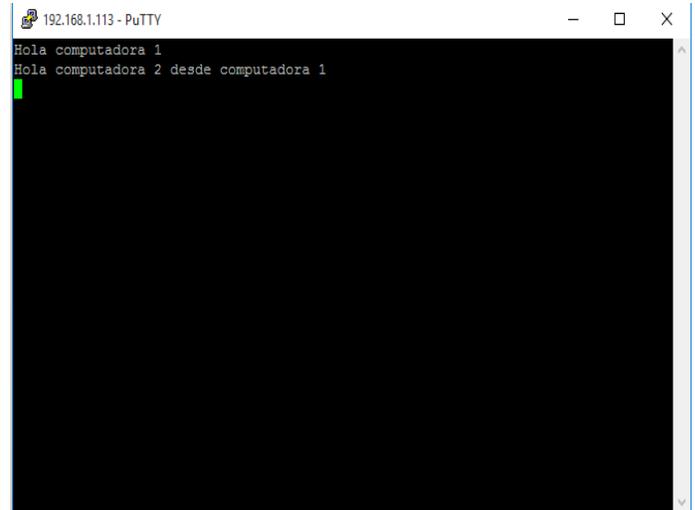


Fig. 6. Interfaz PuTTY con mensajes enviados por los clientes conectados

4. CONCLUSIONES

La implementación del servidor de chat fue exitosa ya que se logró el intercambio de mensajes entre dos computadoras conectadas al servidor montado en la tarjeta nodeMCU.

La gran cantidad de librerías y aplicaciones disponibles en internet permiten una fácil implementación de conceptos como la comunicación por sockets, en este trabajo se pudo demostrar con la ayuda del módulo WiFi ESP8266 integrado en la tarjeta nodeMCU y con ayuda de aplicaciones externas como PuTTY es posible implementar un servidor que maneje el envío y recepción de mensajes en un dispositivo pequeño y de bajo costo. El concepto de Internet of Things está cobrando más relevancia cada día debido a su importancia y utilidad tanto en la industria como en la sociedad. Sistema de bajo costo como la tarjeta nodeMCU permiten que este tipo de tecnología sea accesible y a su vez permiten a los desarrolladores implementar una gran variedad de aplicaciones, como son por ejemplo sistema de monitoreo con sensores, servidores y clientes de aplicaciones web, proyectos de robótica entre otros.

En cuanto a seguridad, actualmente el sistema tiene tres restricciones para que un usuario pueda acceder al servidor:

1. El usuario debe estar conectado en la misma red que el servidor
2. El usuario debe conocer la dirección IP del servidor
3. El usuario debe conocer el puerto por el cual se está llevando la comunicación con el servidor

Como mejora en la seguridad del sistema se puede implementar una contraseña para que, además de las tres condiciones previamente mencionadas el usuario necesite ingresar dicha contraseña para acceder al servidor e interactuar con los demás usuarios.

REFERENCIAS

- 1] Calvert, K.; Donahoo, M. (2008). TCP/IP Sockets in Java: Practical Guide for Programmers. Second Edition. MK
- [2] Harold, E. (2013). Java Network Programming. 4th Edition. O'Reilly Media, Inc.
- [3] Usuario: Japh (2018). Create a Local Chat Server using NodeMCU devkit v1.0 board via Telnet. Obtenido de <https://steemit.com/utopian-io/@japh/create-a-local-chat-server-using-nodemcu-devkit-v1-0-board-via-telnet>
- [4] Página web de PuTTY. Obtenido de <https://www.putty.org/>
- [5] Página web de Arduino. Obtenido de <https://www.arduino.cc/>
- [6] Java Documentation. <https://docs.oracle.com/javase/tutorial/networking/sockets/>
- [1] Brookshier, D..”JavaBeans Developer's reference.” New Riders Publishing, 2017
- [2] Campione, M.; Walrath, K. “The Java Tutorial Continued: the Rest of the JDK” Addison-Wesley, 2008
- [3] Cornell, G.; Horstmann, Cay S. “Core Java” (2ª ed.). Prentice Hall, 2012
- [4] Flanagan, D. “JAVA in a Nutshell” (2ª ed.): O'Reilly&Associates, Inc, 2013
- [5] Geary, David M. “Graphic JAVA. Mastering the AWT” (2ª ed.). Prentice Hall. 2015
- [6] Jenkins, Michael S. “Abstract Data Types in Java “(1ª ed.). McGraw-Hill. 2011
- [7] Horstmann, Cay S.; Cornell, G “Core Java Volume II-Advanced Features.” Prentice Hall: 2016
- [8] Orfali, R.; Harkey, D.; Edwards, J."Instant CORBA". John Wiley & Sons, inc.2015
- [9] Orfali, R.; Harkey, D.Client/Server Programming with JAVA and CORBA. John Wiley & Sons, inc, 2017.
- [10] Sridharan, P. “Advanced Java Networking.”: Prentice Hall 2014
- [11] G. Coulouris, J. Dollimore, T. Kindberg. “Sistemas Distribuidos – Conceptos y Diseño” (3ed.) Addison Wesley, 2011