Software to build network frames, with GUI, CLI and programming library

Software para construir marcos de red, con GUI, CLI y biblioteca de programación

Software para construir pacotes de rede com GUI, CLI e biblioteca de programação

Marta Lucia López Muñoz[1]
José Luis Alejandro Carrillo Valderrama[2]

Received: January 10th, 2019
Accepted: March 26th, 2019
Available: May 21th, 2019

How to cite this article:
J. L. A. Carrillo-Valderrama, M. L. López-Muñoz, "Software to build network frames, with GUI, CLI and programming library", *Revista Ingeniería Solidaria*, vol. 15, n.° 2, 2019.
DOI: https://doi.org/10.16925/2357-6014.2019.02.04

Research article. https://doi.org/10.16925/2357-6014.2019.02.04

[1] Corporación Politécnico de la Costa Atlántica, Barranquilla, Colombia.
ORCID: http://orcid.org/0000-0002-9221-2259

[2] Corporación Politécnico de la Costa Atlántica, Barranquilla, Colombia.
ORCID: https://orcid.org/0000-0001-8517-9334
Email: jcarrillov@pca.edu.co

Abstract

Introduction: This publication is the product of a research project named "Design and implementation of a software to build network frames with CLI, GUI interface and programming library" of the Corporación Politécnico de la Costa Atlántica in Barranquilla, Colombia between the year 2016 and 2018. A piece of software with multiplatform capability, GUI, CLI and programming library was developed. The software allows for the intuitive construction of network frames, speeding up construction times, thus facilitating practical learning of the concept of encapsulation networking.

Objective: The research aims to build a multiplatform computing solution, with a GUI, CLI and programming library, in order to build network frames intuitively.

Methodology: The scope of the project is descriptive, since the development includes the analysis and implementation of the software and its components; as well as functional tests.

Results: The software was built according to the phases proposed in the methodology. A high percentage of students obtained average scores in the evaluation carried out on frames, packages and segments.

Conclusion: The results demonstrate the need to apply a new teaching methodology based on the elaboration of frames, to improve the learning process in computer networks.

Originality: A software solution that complements existing limitations in portability, versatility and ease of use is proposed. This solution is applicable both in the workplace and in the educational field.

Limitations: The processes of sending and capturing frames are separated in the software, meaning that TCP communication cannot be maintained. It is necessary to build a GUI application to capture network frames and not depend on a tool like Wireshark for visualizing the frames.

Keywords: Frame, Encapsulation, Network computers, Software, Teaching.

Resumen

Introducción: Esta publicación es el producto de un proyecto de investigación titulado "Diseño e implementación de un software para construir paquetes de red, con interfaz CLI, GUI y librería de programación" de la Corporación Politécnico de la Costa Atlántica en Barranquilla, Colombia entre el año 2016 y 2018. Se ha desarrollado un software multiplataforma con GUI, CLI y biblioteca de programación. El software permite la construcción de tramas de red de manera intuitiva, acelerando los tiempos de construcción. De esta forma facilita el aprendizaje práctico del concepto de encapsulamiento de red.

Objetivo: La investigación tiene como objetivo construir una solución informática multiplataforma, que permita ser utilizado como GUI, CLI y librería de programación, para construir tramas de red de forma intuitiva.

Metodología: El alcance del proyecto es descriptivo, ya que el desarrollo incluye el análisis y la implementación del software y sus componentes; así como pruebas funcionales.

Resultados: Se construyó el software de acuerdo a las fases planteadas en la metodología. Se obtuvo un alto porcentaje de estudiantes con desempeño regular en la evaluación efectuada sobre tramas, paquetes y segmentos.

Conclusión: Los resultados demuestran la necesidad de aplicar una nueva metodología de enseñanza basada en la elaboración de tramas, para mejorar el proceso de aprendizaje en redes de computadores.

Originalidad: Se propone una solución informática que complementa las limitaciones actuales en portabilidad, versatilidad y facilidad de uso. Esta solución es aplicable tanto en el ámbito laboral como en el ámbito educativo**.**

Limitaciones: Los procesos de envío y escucha de tramas se encuentran separados en el software, por lo que no se puede mantener una comunicación TCP. Es necesario construir una aplicación GUI para escuchar las tramas de red y no depender de una herramienta como Wireshark para visualizar las tramas de forma intuitiva.

Palabras clave: tramas, encapsulamiento, redes de computadores, software, enseñanza.

Resumo

Introdução: esta publicação é produto de um projeto de pesquisa intitulado "Desenho e implantação de um *software* para construir pacotes de rede, com interface GUI, CLI e biblioteca de programação" da Corporación Politécnico de la Costa Atlántica, em Barranquilla, Colômbia, entre 2016 e 2018. Desenvolveu-se um *software* multiplataforma com GUI, CLI e biblioteca de programação. O *software* permite construir tramas de rede de maneira intuitiva, que acelera os tempos de construção. Dessa forma, facilita a aprendizagem prática do conceito de encapsulamento de rede.

Objetivo: esta pesquisa tem como objetivo construir uma solução informática multiplataforma que permita ser utilizada como GUI, CLI e biblioteca de programação, para construir tramas de rede de forma intuitiva.

Metodologia: trata-se de um projeto com abordagem descritiva, já que seu desenvolvimento inclui a análise e a implantação do *software* e de seus componentes, bem como testes funcionais.

Resultados: foi construído o *software* de acordo com as fases propostas na metodologia. Foi obtida alta porcentagem de estudantes com desempenho regular na avaliação realizada sobre tramas, pacotes e segmentos.

Conclusão: os resultados demonstram a necessidade de aplicar uma nova metodologia de ensino baseada na elaboração de tramas, para melhorar o processo de aprendizagem em redes de computadores.

Originalidade: propõe-se uma solução informática que complementa as limitações atuais em portabilidade, versatilidade e facilidade de uso. Essa solução é aplicável tanto no âmbito corporativo quanto no educativo.

Limitações: os processos de envio e escuta de tramas estão separados no *software*, portanto não pode ser mantida comunicação TCP. É necessário construir uma aplicação GUI para escutar as tramas de rede e não depender de uma ferramenta como Wireshark para visualizar as tramas de forma intuitiva.

Palavras-chave: tramas, encapsulamento, redes de computadores, *software*, ensino.

1. Introduction

In this research project we present the development of a multiplatform software with graphical interface (GUI), command lines (CLI) and programming library, which allows for the construction of network frames.

The developed software is a multifunctional tool, which can be used for specialized developments in education programs for the installation of networks and partners, as well as for penetration tests —pentesting—, in security, and as a teaching resource in professional training processes for students.

According to the above, S. Wang, D. Xu, and S. Yan [1], show the students' difficulties in understanding the protocol data unit (PDU) of the TCP / IP model, making it difficult to understand the principle and the function of each network layer, considering the content as abstract and boring during master classes. To solve this problem, the authors propose the use of the Wireshark tool as an active and dynamic, didactic resource to facilitate the learning of the protocols of the TCP / IP model and the monitoring of network traffic, improving training and interest in learning. Studies have been conducted at the methodological level in order to encourage students in thematic networks computers as shown by A. Zafra, E. Gibaja, M. Luque, and L. Toribio [2].

However, a need has been clearly stated within these studies because presently, the students cannot control the making and sending of network frames in their entirety, in a didactic way —based on visual learning— or through the use of block diagrams. The current solution is place requires the use of a personal identifier in the payload of the network frame, in order to guarantee that the data reaches its expected destination.

Through this proposed software, the learning of the concept of data encapsulation is facilitated through a network, promoting didactic, intuitive and versatile learning since the user will have access to the structure of the network frames with similar block diagrams to be used with graphs of the TCP / IP model used in diverse texts of networks. Likewise, this software was created with interpreted languages, therefore it can be executed in Windows, Linux and macOS (multiplatform), making this software highly compatible in different aspects and programming languages. The Libpcap library was chosen as the programming library, that provides functions for the capture of packages independent of the operating system, as established in [3] and [4].

All the above show the great advantages of the proposed solution because the current market tools —Ostinato, packETH, Scapy and Nmap—, do not provide this didactic and technological flexibility when working, as shown in Mejia et al. [5].

Currently, a great variety of research has been developed on the methodology of education suitable for the concepts of computer networks. Some authors present an approach called ascending, so called in the jargon of telecommunications, to start explaining the concepts from the lowest layer of the OSI or TCP / IP model until reaching the highest layer, as was described in [6] and [7]. Other authors propose the descending model that starts from the application layer and ends with the physical layer or network link, as proposed by Rodríguez et al. [8], who defends this approach in the teaching of computer networks. It is necessary to describe these approaches since the proposed software leans towards the bottom-up approach, allowing for the construction of network frames from the data link layer to the application layer of the TCP / IP model in that strict order.

2. Literature review

In the 1970s a set of protocols was developed. Transmission Control Protocol / Internet Protocol (TCP / IP) allowing for communication between computers. Currently, it is the basis of network interconnection and is made up of 4 layers, described in ascending order: Physical Layer, Network Layer, Transport Layer and Application Layer, as described by [7] and [9]. Other authors like A. S. Tanenbaum and D. J. Wetherall [6] explain the TCP/IP protocol with 5 layers, separating the physical layer of the data link layer proposed in the OSI model.

For a long time, there has been vulnerability to attacks against the TCP / IP protocols. Harris & Hunt [10], raised configuration methods to limit the effectiveness of common attack methods such as IP spoofing, TCP sequence number attacks and session hijacking. Thus, the authors proposed the use of patches or service packs to address new vulnerabilities.

In the educational field, the process of teaching the subject of computer networks has traditionally been performed using abstract concepts such as TCP / IP protocol, data encapsulation and network frames. Protocol analysis tools have been used to augment student motivation and understanding of the exposed theories. According to S. Wang, D. Xu, and S. Yan [1], the use of the Wireshark tool when teaching TCP / IP protocol can improve student learning by relating theoretical abstract knowledge with direct and active examples. Other authors, such as Jiao & Hao [11], state that network protocol analysis systems such as Sniffer, NetXray, Iris and network simulation systems —such as simulator 2 (NS2) and OPNET— allow students better understand abstract network concepts but generate disadvantages in terms of cost and a greater need for environment configuration. They propose the use of a demonstration system teaching laboratory of the network, based on the web with the following functions: analysis of protocol, demonstration of the process of encapsulation of data, transmission of packages in the communication of the network.

Finally, Benavices et al. [12], propose the use of embedded applications in Software Defined Networks (SDN) as Internet-based educational tools in computer networks. The authors present the difficulty of teaching control of computer networks due to the lack of physical devices and networks, thus the use of SDN when developing work, facilitate the teaching of computer networks.

3. Materials and methods

At present, the tools for manipulating network frames present limitations in portability and versatility. This project attempts to find a solution to these limitations —thereby positioning it as an experimental development according to the Frascati Manual—. The scope of the project is descriptive, as the development includes the analysis and implementation of the software and its components through the concept of the architecture and expected functionalities —considered the variables of interest in the project— as well as the tests carried out to establish the need to strengthen the teaching of the basic concepts of computer networks.

The stages of this project are based on I. Sommerville, V. Campos Olguín, and S. Fuenlabrada Velázquez [13], and are described sequentially below:

- Stage 1. Requirements analysis: The requirements of the solution are obtained and the use cases that establish the basic guidelines for the frame generator are constructed.
- Stage 2. Design and Architecture: The general functioning of the solution is determined by the functional and non-functional requirements, and the characteristics, resources and limitations of implementation, such as hardware, software, restrictions, usability, etc. At this stage, languages and platforms are also defined. The Java programming language (J2SE) and the Libpcap library were chosen.
- Stage 3. Implementation: The computer solution is built, according to the documentation generated by the previous stages, in such a way that it has multiplatform capacity and uses a GUI, CLI and programming library.
- Step 4. Software Validation: It develops and implements test cases where it is clear that the software adequately responds to basic, exceptional and alternate flows of each use case.
- Stage 5. Monitoring and evaluation: In order to support the validation of the software from the functional and didactic point of view, an evaluation is carried out on 25 students of the "Politécnico de la Costa Atlántica" Corporation belonging to the Computer Science program in the subject of networks of computers, on the topic of network encapsulation. The evaluated topics were frames, packages and network segments, with the highest achievable grade being 5.0 and the lowest being 0.0. Measures of central tendency were used to interpret the results.

4. Results

4.1. Software Requirements

The diverse functional and non-functional requirements are analyzed to establish the basic guidelines for the creation of the frame generator that the software supports.

4.1.1. Functional requirements:

R1. The software shall be able to create network frames using layers of the TCP / IP model, through a web application.

R2. The software shall be able to send raw network frames, using bytes in hexadecimal format.

R3. The software shall be able to capture network frames in raw format with hexadecimal notation, given a selected network interface.

R4. The software shall be able to capture network frames, given a selected network interface.

4.1.2. Non-functional requirements:

R5. The software shall be able to send network frames using the layers of the TCP / IP model, through a command line interface (CLI).

R6. The software shall be able to support orders from multiple sockets connected through the JSON-RPC v2.0 format, creating a multi-threaded socket server. The possible orders are:

a) The software shall be able to capture the network frames in raw format with bytes in hexadecimal format, given a selected network interface.

b) The software shall be able to capture the network frames in JSON format, given a selected network interface.

c) The software shall be able to send network frames in JSON format, given a selected network interface.

d) The software shall be able to send network frames in raw format with bytes in hexadecimal format, given a selected network interface.

e) The software shall be able to show a list of the network interfaces of the equipment that the socket server is running.

4.1.3. Use cases

The following use cases are elaborated using the information collected in the previous stage and the need established in the problem statement:

- CU 001. Show Network Interfaces – CLI Option.
- CU 002. Show Network Interfaces - WEB Option.
- CU 003. Capture frames in network interface - WEB Option.
- CU 004. Capture frames in network interface - CLI Option.
- CU 005. Send frames in network interface - CLI Option.
- CU 006. Send frames in network interface - WEB Option.

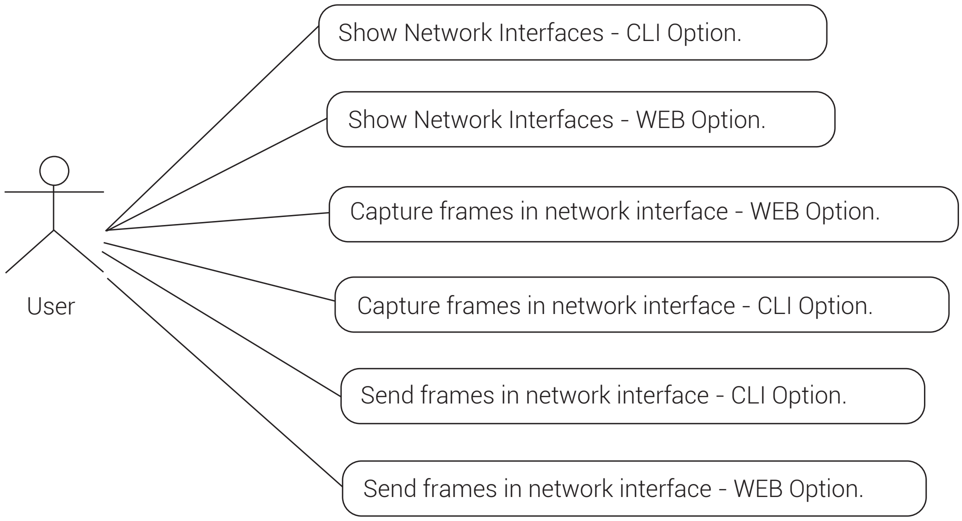

Show Network Interfaces - CLI Option.
Show Network Interfaces - WEB Option.
Capture frames in network interface - WEB Option.
User
Capture frames in network interface - CLI Option.
Send frames in network interface - CLI Option.
Send frames in network interface - WEB Option.

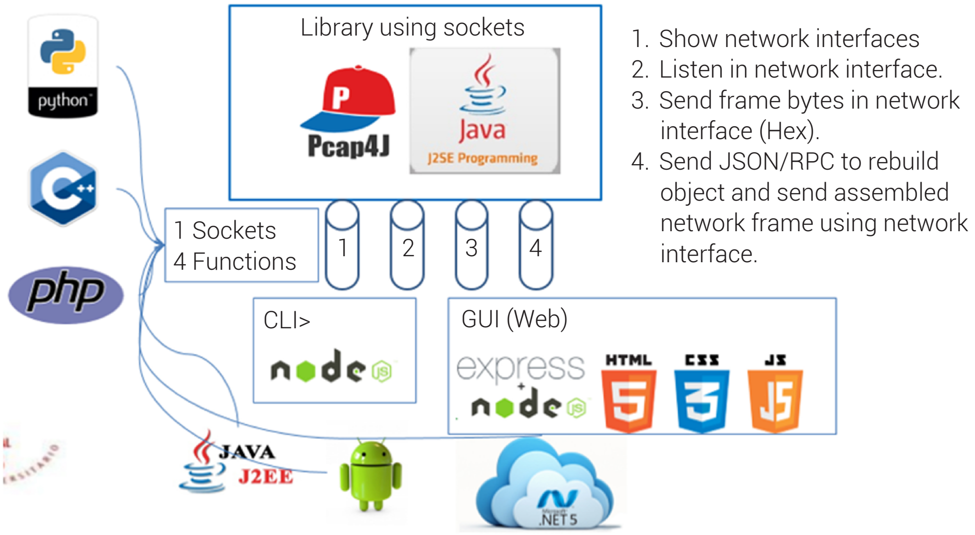
Figure 1. Use cases diagram.
Source: own work

4.2. Design and Architecture

After the software specification stage, based on the functional specifications document, it is necessary to establish the structure of the software that will be implemented. For this, the following sub-stages are necessary for each use case:

- Architecture diagrams
- Sequence diagrams

Next, the developed architecture diagram is presented, in order to explain the operation of the software:



1. Show network interfaces
2. Listen in network interface.
3. Send frame bytes in network interface (Hex).
4. Send JSON/RPC to rebuild object and send assembled network frame using network interface.

Figure 2. Proposed architecture diagram.
Source: own work

Now, the sequence diagram of all the use cases are presented:

CU001 - Show network interfaces - CLI Option.

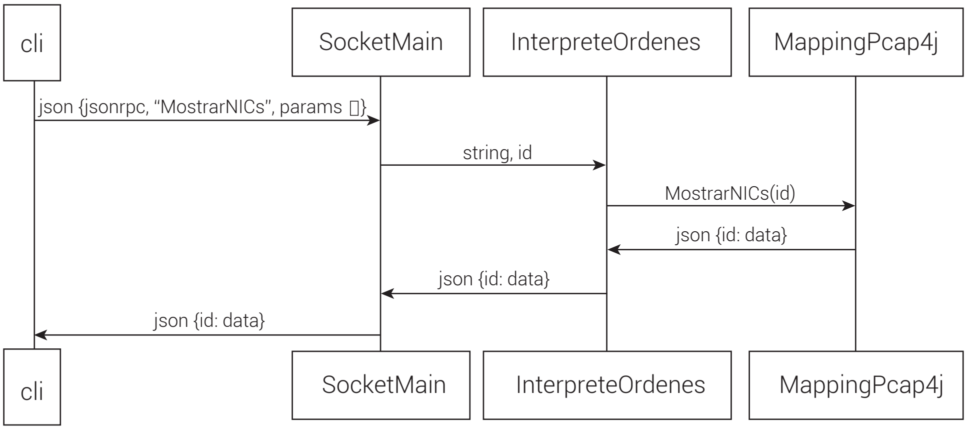


Figure 3. Sequence diagram CU 001.
Source: own work

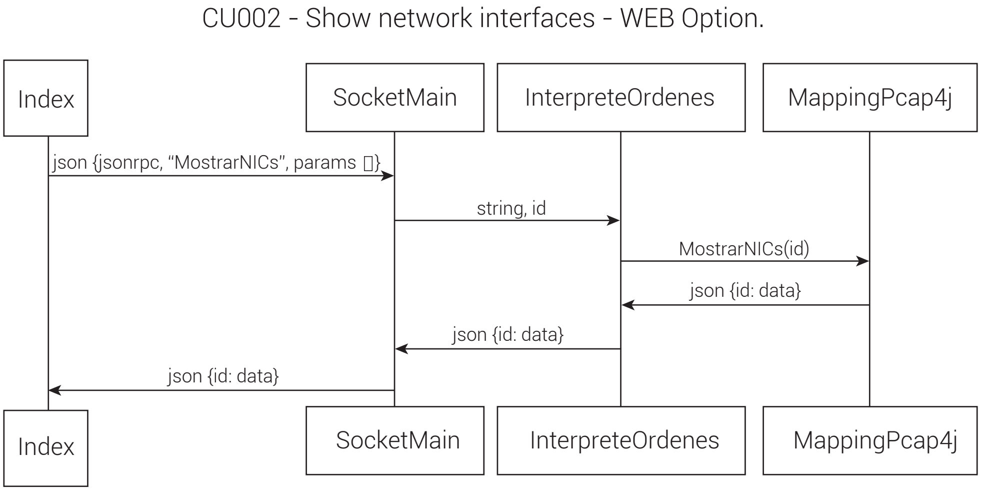


Figure 4. Sequence diagram CU 002.
Source: own work

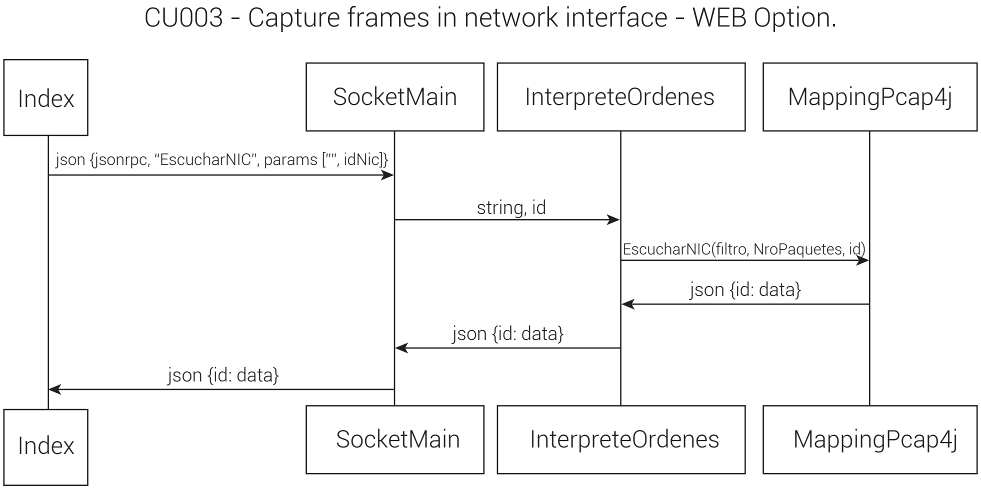


Figure 5. Sequence diagram CU 003.
Source: own work

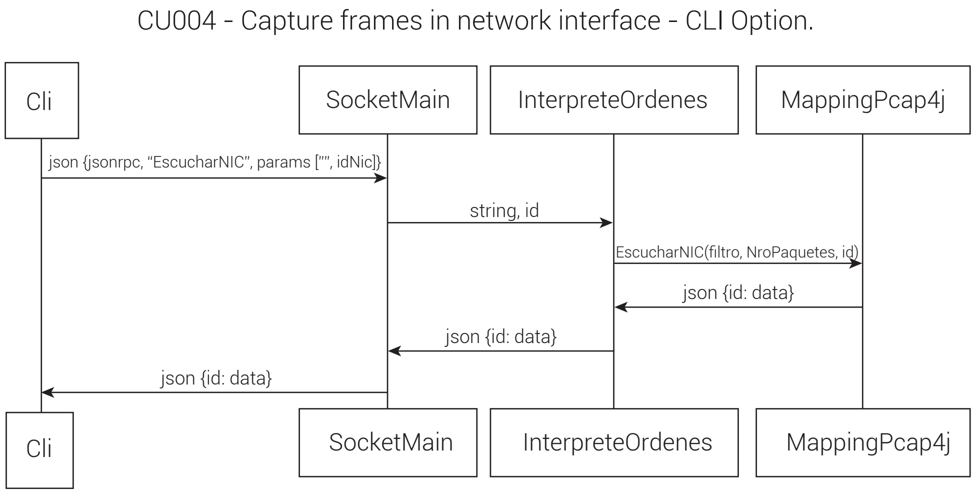

CU004 - Capture frames in network interface - CLI Option.
Cli
SocketMain
InterpreteOrdenes
MappingPcap4j
json {jsonrpc, "EscucharNIC", params ["", idNic]}
string, id
EscucharNIC(filtro, NroPaquetes, id)
json {id: data}
json {id: data}
json {id: data}
Cli
SocketMain
InterpreteOrdenes
MappingPcap4j

Figure 6. Sequence diagram CU 004.
Source: own work

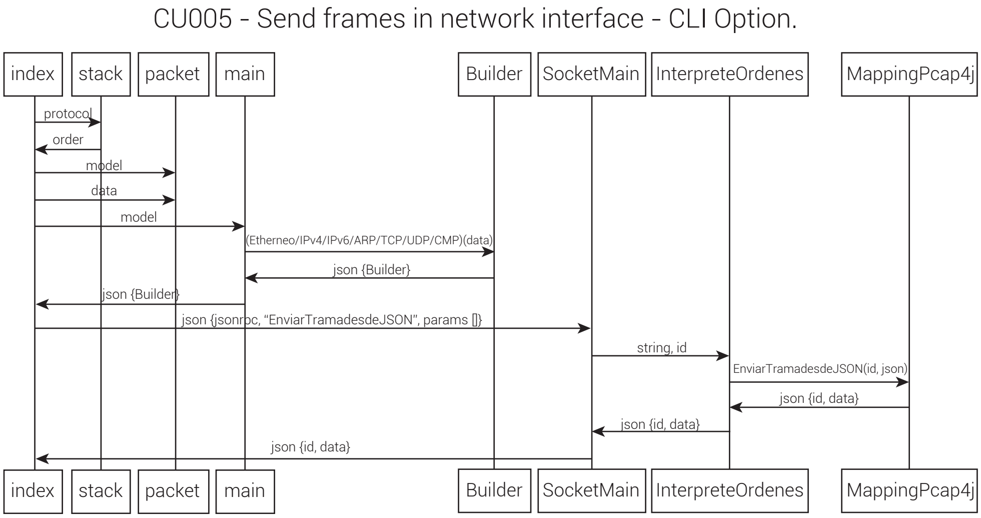


Figure 7. Sequence diagram CU 005.
Source: own work

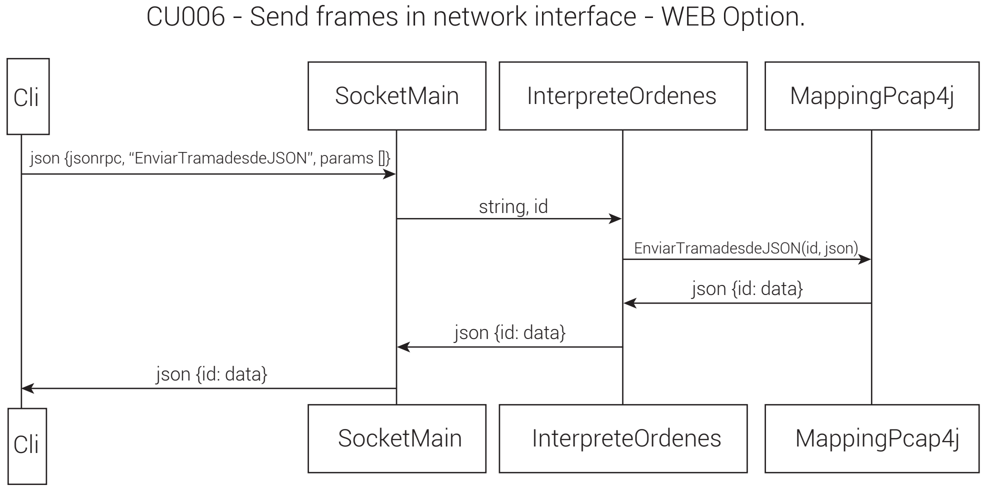


Figure 8. Sequence diagram CU 006.
Source: own work

4.3. Software Implementation

According to the design document, the software is developed using the methodologies and steps described in each use case.

The software module of the socket library is developed using the IDE Netbeans 8.0 (for greater ease of use, portability and speed). The software version used is JDK 1.8. The pcap4j-core-1.4.1 library is used [14], which allows for the use of the Libpcap / Winpcap library from Java independently from the platform in which it is being used. This allows for the fulfillment of a goal of the project (the development of a multiplatform solution). The socket library uses the JSON RPC version 2.0 format for sending and receiving data. The web page is created with the platform for JavaScript execution called NodeJS [15] version 6.3.1 on the server side. On the client side JQuery, jsonviewer, HTML5 and CSS3 were used to create a user-friendly and very intuitive Web interface, even for people who are just learning the basics of networking. We also built Javascript functions, such as Builders, for the creation of network frames in JSON format, in order to support their creation and sending of these to the socket library.
The CLI is included in the same folder in which the website was developed as they share the Client.js library to send messages to the socket library. Note that the CLI was also built in NodeJS version 6.3.1.

4.4. Software validation

After the software implementation, the corresponding tests are carried out, using as a basis the initial specification document taken into account for the design and implementation. The software tests must obey the following criteria: Functionality and Security. As a result of the software creation process, a technological tool is created that complies with the proposed functional and non-functional requirements, which allows the creation of network frames using the layers of the TCP / IP model.

In the graphic interface (see Figure 9), the options of sending network frames using bytes in hexadecimal format, capturing network frames in hexadecimal notation and downloading the network frame are presented, using the sockets programming library developed in Java.

In addition to the above, the software, through the command line interface (see Figure 10), allows for the option of using an existing network interface on the host where the socket library is running. It also supports the capture and sending of network frames, in JSON format, for their respective transformation, by the socket library.

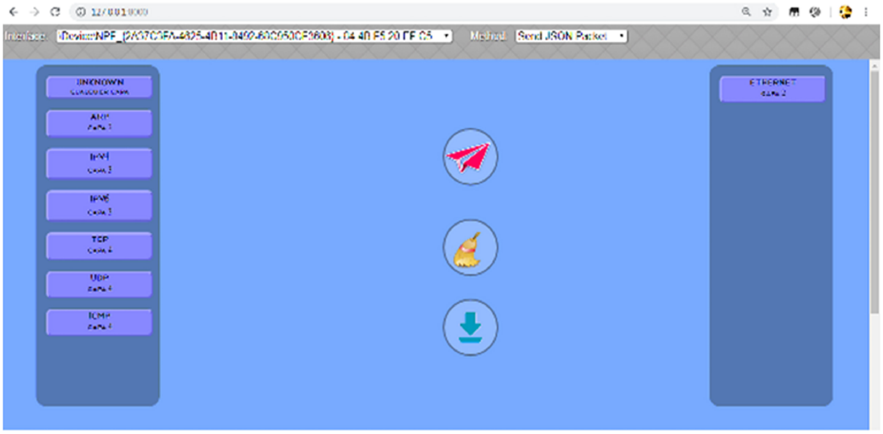
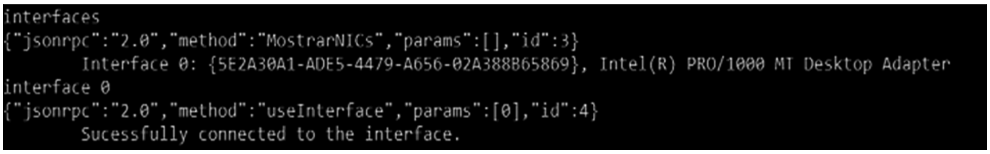

Interface
\Device\NPF_{2A07C0FA-4825-4D11-9492-6C0050CF3808} - 04 40 F5 20 FE C5
Method
Send JSON Packet
UNKNOWN
ARP
IPV4
IPV6
TCP
UDP
ICMP
ETHERNET

Figure 9. Graphical User Interface (GUI).
Source: own work



```
interfaces
{"jsonrpc":"2.0","method":"MostrarNICs","params":[],"id":3}
        Interface 0: {5E2A30A1-ADE5-4479-A656-02A388B65869}, Intel(R) PRO/1000 MT Desktop Adapter
interface 0
{"jsonrpc":"2.0","method":"useInterface","params":[0],"id":4}
        Sucessfully connected to the interface.
```

Figure 10. Command User Interface (CLI).
Source: own work

4.5. Monitoring and evaluation

In Table 1, the results of the evaluation, conducted by 25 students on the concept of network encapsulation, belonging to the academic module of "computer networks" of the Systems Engineering program at the "Politécnico Costa Atlántica" Corporation, are shown. The results provide an arithmetic average of 3.3, mode of 3.3, variance of 0.09 and a standard deviation of 0.3:

Table 1. Evaluation results carried out.

Lic-Lsc	f_i	x_i	$x_i f_i$	F_i	F'_i	h_i	H'_i	$[x_i - x]^2 f_i$
2,0 - 2,4	1	2,2	2,2	1	25	0,04	0,04	1,263376
2,5 - 2,9	2	2,7	5,4	3	24	0,08	0,12	0,778752
3,0 - 3,4	12	3,2	38,4	15	22	0,48	0,6	0,184512
3,5 - 3,9	8	3,7	29,6	23	10	0,32	0,92	1,131008
4,0 - 4,4	2	4,2	8,4	25	2	0,08	1	1,534752

Source: own work

When looking at Table 1, it is notable that the highest concentration of students lies within a range of 3.0 - 3.4 which indicates regular learning and the highest grades are between 4.0 - 4.4 (see Figure 11). Approximately 8% of students are in this interval (see Figure 12), but none achieve higher grades considering that the highest grade is 5. This demonstrates the coherence that exists between the academic level of students and the need to improve the teaching and learning process in the subjects evaluated.

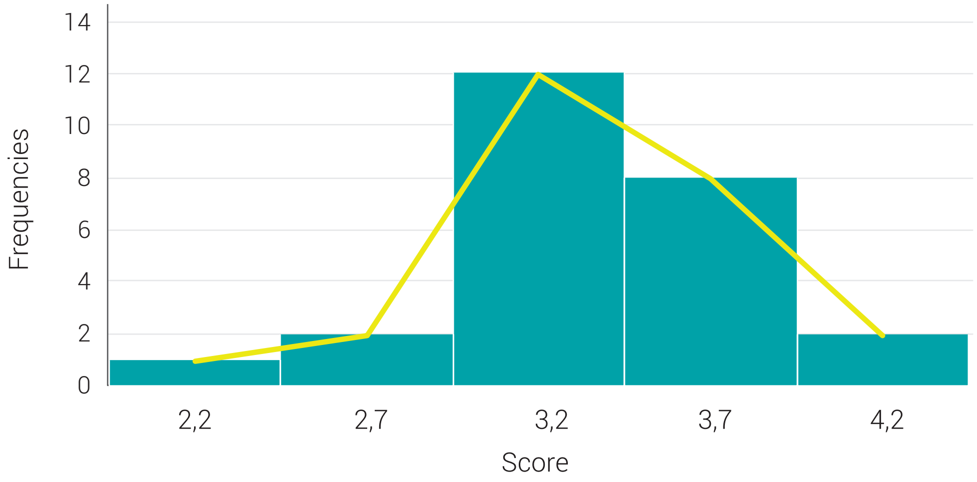


Figure 11. Number of students by score.
Source: own work

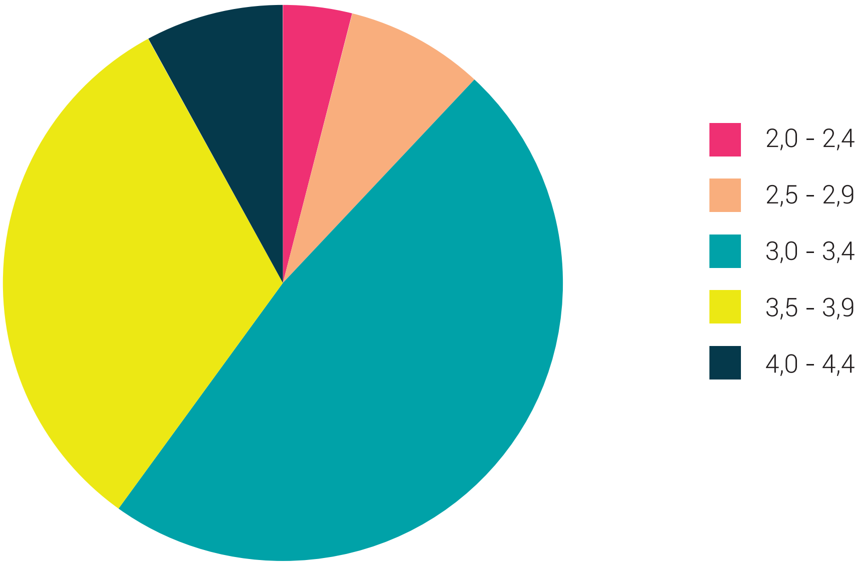


Figure 12. Percentage graphical representation by score.
Source: own work

5. Discussion and conclusions

A multiplatform software was developed, executable in Windows, Linux and macOS, which allows the sending of network frames using the layers of the TCP / IP model, through a graphical user interface (GUI), command line interface (CLI) and programming library; which can be used by different programming languages through the socket library, previously described.

The software allows, through the GUI, for an interaction with the different layers of the TCP / IP model by students and network professionals. Through the creation and sending of the network frames, they can better understand the construction, structure and encapsulation of these, allowing for their customization according to the needs and interests of the user.

According to the literature review, S. Wang, D. Xu, and S. Yan [1], have found the limitations in the current teaching methods related to the learning of network concepts but these solutions are based on using existing tools such as Wireshark, teaching methods related to the network concepts learning. They proposed the use of existing tools such as Wireshark, demonstrating a positive effect in the practical teaching of these concepts. However, for the construction of the source network frame, the authors of this research propose the use of the ping command with a fixed size of

3000 bytes of data as the identifying mark of the network frame or more exactly, an ICMP packet, which does not allow it to be identified with bytes created by students, preventing the sending of personalized messages on the LAN with MAC Source and MAC Destination without another upper layer protocol; a feature, easy to use, that is supported by the software created in our research. Therefore, the use of the Wireshark tool is a complement to the research results to capture the network frames created by the students through the GUI application.

Similar to the previous case, W. Jiao and X. Hao [11], analyzed the advantages and disadvantages of tools for TCP / IP protocol analysis and carried out a web-based teaching demonstration system. This system allows the understanding of the analysis of the structure of IPv4, IPv6 and ARP packets; as well as the teaching of network topologies and OSI - TCP / IP models. In this system, the edition of protocol data is done through the use of a learning module based on specific cases such as ICMP requests, the three-way handshake of TCP connection, disconnection of TCP connection and UDP transmission. However, there is no evidence of a tool for creating the network plot, that allows linking these concepts already explained in the system proposed by these authors, to send network frames; these are easily created and customized by students, to other computers in a network.

Furthermore, Benavices et al. [12], proposed the use of embedded applications in SDN as Internet-based educational tools in computer networks, facilitating teaching through the use of emulated network devices. However, these authors present the difficulty of making network frames in this way (when compared to other market tools such as Ostinato or Scapy). Learning the concepts of network frame encapsulation and PDU (Protocol Data Unit), requires the understanding of the communication details around computer networks.

In the computer security field, Harris & Hunt [10], propose the use of patches and service packs to address new vulnerabilities to the TCP / IP protocol. This is related to research because the developed software can be used to test security in firewalls and IDS/IPS rules, crafting packets easily, and verify if the patches and service packs are useful.

As a recommendation, it is proposed to implement research for the validation and comparison of the impact that the use of software would have on professional training processes for students and professors as opposed to just the knowledge, skills and abilities related to TCP / IP protocol. It is also recommended as future work, to further investigate the teaching methodology of the network encapsulation concept using packet crafting and its effects on the learning process according to the results obtained in point 3.5. Monitoring and evaluation [16-18].

6. References

[1] W. Shaoqiang, X. DongSheng, and Y. ShiLiang, "Analysis and application of Wireshark in TCP/IP protocol teaching," in *2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT)*, Shenzhen, China, 2010, pp. 269–272. [Online]. doi: 10.1109/EDT.2010.5496372.

[2] A. Zafra, E. Gibaja, M. Luque, and L. Toribio, "Propuesta metodológica para motivar a los estudiantes en el estudio de las Redes de Computadores," *Enseñanza y aprendizaje de ingeniería de computadores: Revista de Experiencias Docentes en Ingeniería de Computadores*, no. 8, pp. 37–48, 2018. [Online]. Available: https://dialnet.unirioja.es/servlet/articulo?codigo=6710896.

[3] A. Monge Lopez, *Aprendiendo a programar con Libpcap*,. [Online]. Available: http://e-ghost.deusto.es/docs/2005/conferencias/pcap.pdf.

[4] D. Zhou, Z. Yan, Y. Fu, and Z. Yao, "A survey on network data collection," *Journal of Network and Computer Applications*, vol. 116, pp. 9–23, Aug. 2018. [Online]. doi: 10.1016/j.jnca.2018.05.004.

[5] A. F. Castaño Mejía, L. M. Ocampo Pelaéz, and M. S. Orozco Piedrahita, *Fundamentación teórica y exploración de generadores de tráfico multiplataforma*, Jul. 2014, pp. 21-30. [Online]. Available: http://repositorio.ucp.edu.co/handle/10785/2089.

[6] A. S. Tanenbaum and D. J. Wetherall, "Modelos de referencia," in *Redes de computadoras*, México: Pearson Educación, 2012.

[7] W. R. Stevens and K. R. Fall, "TCP/IP Security Protocols and Layering," in *TCP/IP Illustrated, Volume 1: The Protocols*, pp. 832–833.

[8] N. R. Rodríguez *et al.*, "Aplicación del enfoque descendente para la enseñanza de Redes de Computadoras, un estudio comparativo," in XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)., 2016, pp. 1270–1279. [Online]. Available: http://hdl.handle.net/10915/56294.

[9] P. M. Miller, "The Internet Protocol Suite Model," in *TCP/IP: Complete 2 Volume Set*, Universal-Publishers, 2010, pp. 22–24.

[10] B. Harris and R. Hunt, "TCP/IP security threats and attack methods," *Computer Communications*, vol. 22, no. 10, pp. 885–897, Jun. 1999. [Online]. doi: 10.1016/S0140-3664(99)00064-X.

[11] W. Jiao and X. Hao, “Design of a web-based computer network experiment teaching demonstration system,” *International Journal of Information Technology and Management*, vol. 13, no. 1, p. 44, 2014. [Online]. doi: 10.1504/IJITM.2014.059144.

[12] C. Benavices, I. García, H. Alaiz, A. Alonso, and J. M. Alija, “Networking Control Education by the use of Software Defined Networking Tools and Techniques,” *IFAC-PapersOnLine*, vol. 48, no. 29, pp. 307–312, [Online]. doi: 10.1016/j.ifacol.2015.11.253.

[13] I. Sommerville, V. Campos Olguín, and S. Fuenlabrada Velázquez, “El Proceso Unificado Racional,” in *Ingeniería de software*, 9ª ed., México: PEARSON EDUCACIÓN, 2011.

[14] K. Yamada, *A Java library for capturing, crafting, and sending packets.: kaitoy/pcap4j*. 2015,[Online]. Available: https://github.com/kaitoy/pcap4j.

[15] Node.js JavaScript runtime. *Node.js Foundation,* 2019, p.1. [Online]. Available: https://github.com/nodejs/node.

[16] C. H. Caicedo y A. Smida, “Intensidad informacional para la longitudinalidad asistencial en sistemas de salud”, Visión electrónica, vol. 10, no. 1, pp. 83-95, jun. 2016. https://doi.org/10.14483/22484728.11612

[17] K. Duarte Barón y C. Borrás Pinilla, “Generalidades de robots paralelos”, Visión electrónica, vol. 10, no. 1, pp. 102-112, jun. 2016. https://doi.org/10.14483/22484728.11711

[18] J. F. Herrera-Cubides, P. A. Gaona-García, C. E. Montenegro-Marín, S. Sánchez-Alonso, y D. Martin-Moncunill, “Abstraction of linked data’s world”, Visión electrónica, vol. 13, no. 1, pp. 57-74, feb. 2019. https://doi.org/10.14483/22484728.14397