“Performance evaluation and analysis of the development of a tourism application.”

Mamani Mamani Ilma Magda
Facultad de Ingeniería Estadística e Informática
Universidad Nacional del Altiplano
Puno, Perú
Email: im.mamani@est.unap.edu.pe
LinkedIn:] Linkedin

Torres-Cruz Fred
Facultad de Ingeniería Estadística e Informática
Universidad Nacional del Altiplano
Puno, Perú
Email: ftorres@.unap.edu.pe

***Abstract*—This article evaluates the “Your Travel Guide” software, focusing on measuring the load time of its web pages to assess performance efficiency. Using Google Chrome’s Network tab [7], caching was disabled and repeated measurements were performed on various pages of the site, such as home, about us, contact, tours and blog, calculating average load times. NMDS analysis was also employed to visualize similarities and differences in load times between pages, identifying areas for improvement, especially the tours page. The results were compared to average industry standards, highlighting pages in need of optimization. Based on ISO/IEC 25010 (SQuaRE) [19], this study highlights the importance of measuring and optimizing load times to ensure software efficiency and usability, providing specific recommendations. Functional evaluation of “Your Travel Guide” software is essential to ensure its quality and efficiency, integrating technical functionality and user experience in its development, and applying international standards such as ISO/IEC 25010 to ensure software quality in terms of functionality, performance, usability, reliability, security, maintainability and portability.**

***Keywords*—Evaluation, Software, Quality, Efficiency, ISO/IEC 25010.**

I. Introduction

The functional evaluation of the “your travel guide” software was conducted in accordance with ISO/IEC 25010, which defines a quality model that includes characteristics such as performance and efficiency. This evaluation focused on measuring the loading time of the website pages, a key metric to evaluate the efficiency of the [6] system.

To carry out this evaluation, recognized methodologies and tools in Software Engineering will be used, such as software testing, code inspection, and software quality measurement according to the ISO/IEC 25010 (SQuaRE) standard [12]. Among the tools intended to be used is JMeter, once our webpage is completed, which is an open-source tool that allows for load and performance testing, essential for analyzing the efficiency and behavior of the software under different usage conditions. Additionally, Google Chrome DevTools, specifically the Network tab, will be used to measure the web page load times [5].

It is important to highlight that the functional evaluation of the software is a continuous and iterative process that must be carried out throughout the entire software development lifecycle [3]. The results obtained in this study can be used as a basis for improving the quality and performance of the web page software *Tu guía de viaje*, as well as to guide future research in the field of Software Engineering [8].

In the specific case of *Tu guía de viaje*, a web application aimed at promoting lesser-known regional tourist destinations, it is crucial to define a development architecture that integrates both technical functionality and user experience [23].

The importance of quality assurance lies in ensuring that the software not only functions correctly but also provides a satisfactory experience to the end user, complying with the standards established by ISO/IEC 25010 (SQuaRE) [15].

The application of international standards such as ISO/IEC 25010 is fundamental to ensuring software quality. This standard provides a framework for evaluating various aspects of software quality, including functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability. This evaluation will not only allow for the identification and correction of defects but also optimize software performance, thereby ensuring that *Tu guía de viaje* offers a high-quality user experience and meets the requirements and expectations of its users.

II. Design

A. Type of Research

Basic: This study is of basic type because it seeks to obtain knowledge about the functionality and quality of the software “Your travel guide”, which can serve as a basis for future research and development in the field of Software Engineering [11].

B. approach

. Quantitative: The approach is quantitative as objective metrics and automated tools will be used to evaluate different aspects of the software, such as performance, usability and reliability.

C. Procedure: Population and Sample

Population: The population in this context refers to all web pages of the “your travel guide” software that are relevant for the performance evaluation. This could include all main pages of the website, such as:Home, Locations, Packages, About Us, Contact Us, registration and login [10].

Objective to Measure: Measure the load times of the web pages of the ”tu guía de viaje” software to evaluate its performance efficiency using browser development tools and NMDS analysis, in order to identify and recommend specific improvements that optimize user experience based on industry average standards. [13]

Sample: A subset of the population will be selected to measure load times. In this case, we will select the most representative and frequently visited pages: Home, Locations, About Us, Contact Us, Registration, and Login. [1]

III. Instruments

Tools/Techniques: For the functional evaluation of the software, various tools and techniques were used. One of them will be employed when our application is completed using JMeter to carry out load and performance tests. This tool allows simulating multiple simultaneous users accessing the software, which helps identify potential bottlenecks and evaluate the system's capacity to handle high loads.

Additionally, Google Chrome was used to measure the loading time of selected pages. The Network tab in the browser's developer tools provided detailed data on resource loading times, allowing for a precise analysis of the software's efficiency in terms of page performance[21].

Finally, NMDS (Non-metric Multidimensional Scaling) analysis was applied, a technique used in software evaluation to visualize and understand the similarity or dissimilarity between multiple objects, such as different versions of a software or its different components. This technique provides a visual representation that facilitates the comparison and analysis of software characteristics[22] [4].

Validity: Refers to the accuracy and reliability of the tools and methods used to measure the web page loading time.

Precision: The Network tab in Google Chrome provides detailed and precise measurements of web resource loading times, including DNS times, connection times, server response times, and individual resource loading times. Additionally, using the Disable Cache option ensures that the measurements reflect real loading times without cache interference.

Repeatability: Measurements can be repeated multiple times to obtain an average value, reducing variability and improving the accuracy of the results. Google Chrome is a widely used industry-standard tool for web performance testing, ensuring that the results are consistent and comparable with other studies.

Reliability: The Network tab in Google Chrome is reliable and backed by a large community of developers and technology experts. Chrome's developer tools are continuously updated and maintained by Google, ensuring they remain current and effective for performance testing.

Confidentiality: The tests were conducted in a localhost environment, ensuring that data was not transmitted over the internet and remained in a controlled setting. The collected data was limited to load times and did not include users' personal information. Best security practices were followed to ensure that only authorized personnel had access to the development tools and test data.

IV. Procedure

Requirements Specifications:

A detailed analysis of user requirements was conducted, identifying the functional and non-functional needs of our application. [14]

A. Functional Requirements

The first functional requirement is User Registration. This functionality allows users to create accounts to access all features of the application. This ensures that each user has a personalized experience and access to their data and preferences.

Search and Exploration of Destinations is another crucial requirement. The application must provide an advanced search system that allows users to search for tourist destinations by location, category, available activities, and other relevant criteria. This facilitates the identification of destinations that match the specific preferences and needs of each user.

Additionally, providing Destination Details is fundamental. The application must display detailed information about each destination, including descriptions, photos, opening hours, fees, user reviews, and location. This comprehensive information helps users make informed decisions about their trips.

Travel Planning is another key aspect. The application should offer tools for users to create personalized itineraries, add destinations to their wish list, and schedule travel dates. This allows users to organize their trips efficiently and effectively.

Community and Collaboration is an important functional requirement. The application should enable users to share their experiences, leave reviews and ratings, and recommend destinations to other users. This feature fosters an active community and provides valuable information based on other travelers' experiences.

Additionally, the application should send Notifications about destination updates, special offers, and relevant events. This keeps users informed and engaged with the application.

Multilingual Support is essential so that users can access the application in their preferred language. This broadens accessibility and enhances the user experience.

Finally, Content Management is a critical functional requirement. Application administrators should be able to add, edit, and delete tourist destinations, as well as manage user-generated content. This ensures that the information in the application is accurate, up-to-date, and relevant.

B. Non-Functional Requirements

Performance: It is essential to ensure fast loading times and a smooth user experience even on slow internet connections. This is achieved through code optimization, the use of content delivery networks (CDNs), and data compression techniques.

Security: Implementing robust security measures is crucial to protect users' personal information and prevent cyberattacks. This includes the use of encryption protocols, two-factor authentication, and continuous threat monitoring.

Compatibility: The application must be compatible with a wide range of devices and web browsers, including desktop computers, tablets, and smartphones. This is achieved through extensive testing on different platforms and the use of responsive web technologies.

Usability: Designing an intuitive and easy-to-use user interface is critical for a good user experience. Navigation should be clear and consistent across all sections of the application, facilitating access to all functionalities.

Scalability: The application should be designed to handle growth in the number of users and the amount of data without compromising performance. This involves a flexible architecture and the ability to scale resources as needed.

Accessibility: It is important to ensure that the application is accessible to people with disabilities, complying with established web accessibility standards.

C. Techniques Used

We use the MoSCoW technique within the proposed methodology framework, which considers four possibilities for future development: Must have, represented by an M; Should have, represented by an S; Could have, represented by a C; and Won't have, represented by a W [18].

Firstly, under the "Must have" (M) category, we have User Registration. Users must be able to create accounts to access all the site's functionalities. This also includes the Search and Exploration of Destinations, which requires the implementation of an advanced search system allowing users to search for tourist destinations by location, category, and other criteria. Additionally, displaying Destination Details is essential, providing detailed information about each destination, including descriptions, images, schedules, fees, and more. Travel Planning is also a priority, offering tools for users to create personalized itineraries and schedule travel dates. Lastly, under this category, we have Community and Collaboration, enabling users to share experiences, leave reviews, and recommend destinations to other users.

In the "Should have" (S) category, the Notification functionality is included. It is important to implement a notification system to inform users about destination updates, special offers, and relevant events. Moreover, providing multilingual support is crucial so users can access the site in their preferred language.

Under the "Could have" (C) category, Advanced Interactivity Functions are considered, such as incorporating additional features like interactive maps and virtual reality to enhance user experience. Content Management is also included, a functionality allowing administrators to add, edit, and delete tourist destinations and user-generated content, thus improving site management. Additionally, Social Media Integration would enable users to share site content on their social networks, increasing the platform's visibility and reach. In the "Will Not Have" (W) category, it is established that, given the basic nature of development in HTML/CSS/JavaScript and to maintain the simplicity of the project, advanced features such as integration with booking systems, real-time data analysis, or complex social media functions may not be included in this initial version of the website.

In this project, the agile Scrum methodology was also chosen for implementation, characterized by its collaborative, iterative, and incremental approach. Scrum is based on dividing the work into sprints, short periods during which specific product functionalities are developed [20].

The Scrum method used to manage the software development is described below, ensuring the delivery of functional parts in each sprint. The "Your Travel Guide" website, currently under development, aims to facilitate the search and location of points of interest near the user. Additionally, it seeks to promote the products of local artisans and the services of tour guides, fostering culture, art, and sustainable tourism in our region or country [9].

First, the features of the website were defined. The process started by creating the HTML files for the menu and incorporating the logo into these files. A brief description of the menu content was included, and images of tourist attractions of interest to the company were implemented. Finally, tests were conducted to verify its functionality.

Regarding the user interface and design, a registration and login form was created, connecting PHP to the database. A CSS style template was designed and developed, defining a coherent color palette and graphic style. Additionally, the MySQLi database was designed to store user data.

For the search and recommendation of destinations, recommendation algorithms based on user preferences and behavior were developed, and a functional search bar was implemented.

In content management, a system was developed to upload and store images and descriptions of destinations. Furthermore, an administration panel was designed and implemented to add new tourist destinations.

In terms of security and legal compliance, HTTPS will be implemented throughout the site to ensure the protection of transmitted data.

In the testing and launch phase, a launch plan was prepared. Unit tests will also be developed to ensure that each system component works correctly before the final implementation.

V. Detailed Procedure

Preparation of the Test Environment: To carry out the tests, it was ensured that the website "Tu Guía de Viaje" was running in a localhost environment. Google Chrome was used to navigate to the homepage of the website, thus guaranteeing a controlled and replicable environment for measuring load times.

Opening the Developer Tools: To access Google Chrome's developer tools, F12 or Ctrl+Shift+I was pressed. Then, the Network tab was selected, which provides details about the page's resource load times.

Measuring Load Time: The Disable cache option was selected to ensure that the measurements reflected accurate load times without cache interference. The page was then reloaded using Ctrl+R or the browser's reload button. Load times were observed and recorded in the Time column or the Load time at the bottom of the Network tab. This process was repeated for each of the selected pages of the website, including the About Us, Contact, Tours, and Blog sections.

Data Recording: The load times of each page were recorded in a table or spreadsheet for analysis. To obtain a representative value and minimize variability in the measurements, multiple measurements (e.g., three times for each page) were taken, and the average load times were calculated.

A. Evaluation

The table shows the loading times measured in seconds for different pages of a website. The table contains five columns: one for the page and four for the loading times measured in three trials, followed by the average of these times. The procedure used to obtain these data is detailed below. Three loading time tests were conducted for each

page of the website: Home, Location, About Us, Contact, and Blog. Each test involved fully loading the page and measuring the time elapsed from the start of the load until the page was completely loaded.

For the load time measurements, the load time was recorded during the first trial (Meas. 1), the load time was recorded during the second trial (Meas. 2), and the load time was recorded during the third trial (Meas. 3). For each page, the average of the three recorded load times was calculated. This average is presented in the last column of the table (Avg.(s)).

TABLE I

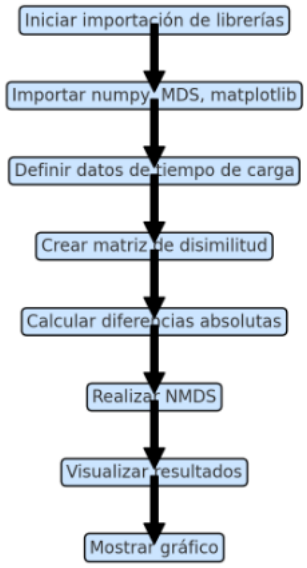
.

Available Data				
Page	Meas. 1(s)	Meas. 2(s)	Meas. 3 (s)	Avg.(s)
Home	4.38	5.09	5.04	4.8367
Location	4.55	4.37	4.11	4.343
About Us	4.30	4.41	4.16	4.29
Contact	4.41	4.69	4.18	4.426
Blog	4.22	3.91	4.04	4.057

B. *Dissimilarity Matrix:*

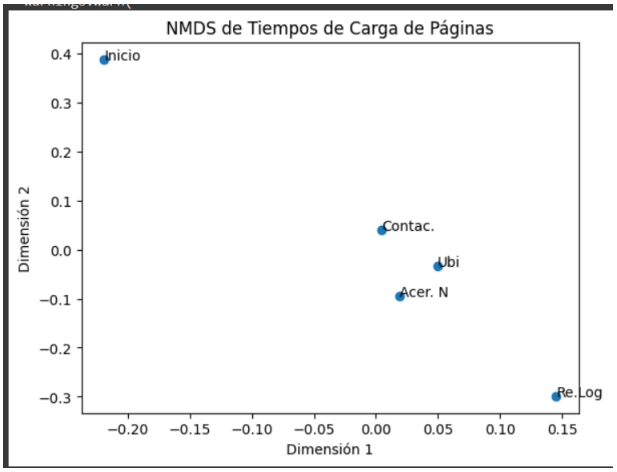
Calculates the absolute difference in load times between each pair of pages in Python. The pseudocode is presented below.

```
Start importing libraries:

Import numpy as np
Import MDS from sklearn.manifold
Import matplotlib.pyplot as plt
Define average load time data in seconds:

Create a dictionary "pages" with page
names and their load times:
"Home" -> 4.8367
"Location" -> 4.343
"About Us" -> 4.29
"Contact" -> 4.426
"Blog" -> 4.057
Create dissimilarity matrix (absolute
difference in load time):

Convert the values of "pages" to a
list "pages_list".
Get the number of elements "n"
in "pages_list".
Create a matrix "dissimilarity_matrix"
of size n x n filled with zeros.
For each index i in the range of 0 to n:

For each index j in the range of 0 to n:
Calculate the absolute difference between
"pages_list[i]" and "pages_list[j]".
Assign this value to
"dissimilarity_matrix[i, j]".
Perform NMDS:

Create an instance of MDS with 2
components, using precomputed dissimilarity
and random state 42.
Fit and transform "dissimilarity_matrix"
using MDS, and save the result in
"nmds_result".
Visualize results:

Create a scatter plot of
"nmds_result" using the
first two dimensions.
For each index i and page
name in "pages":

Add the page name as text
at the corresponding
position in the plot.

Label the x-axis as 'Dimension 1'.

Label the y-axis as 'Dimension 2'.

Set the title to 'NMDS of Page Load Times'.
Show the plot.
```



VI. Results and NMDS Analysis

The average loading times were recorded and used to create a dissimilarity matrix, upon which the NMDS (Non-metric Multidimensional Scaling) analysis was applied. This analysis allowed us to visualize the similarity between the loading times of the different pages of the website[17] [2].



The resulting NMDS plot shows the arrangement of the different pages in a two-dimensional space. In this plot, the proximity between points represents the similarity in their loading times. Below is a detailed analysis of the obtained results:

The Home page is located in the upper left quadrant, far from the rest of the pages. This positioning indicates that the Home page has a significantly higher loading time compared to the other pages. This suggests that the resources or elements on this page are negatively affecting its performance, and it is recommended to investigate possible causes and apply optimizations.

The Contact page is situated near the center of the plot, slightly shifted to the right. The similarity in its loading time with other pages, such as Location and About Us, indicates that these pages share common characteristics in terms of performance. However, there is also room for optimization, which could improve their efficiency.

The Location page is very close to the Contact page in the plot. This proximity suggests that the factors affecting the loading time, such as the use of maps or geolocation services, are similar for both pages. This can provide useful insights into the areas that require attention to improve performance.

The About Us page is near the Contact and Location pages but slightly lower on the plot. This indicates that this page has somewhat better loading performance compared to the others, although it is still within a range where optimization is possible. The slight difference in performance may be due to a lower amount of multimedia content or scripts.

Finally, the Register and Login page is located in the lower right quadrant, far from the rest of the pages. Its isolated location reflects the best loading time among the evaluated pages, indicating that the optimizations applied to this page have been effective. Nonetheless, there is still room to achieve even faster loading times, which could further enhance the user experience.

VII. Discussion

A. Evaluation According to ISO/IEC 25010

In the analysis of loading times for different pages of the website, the following metrics were obtained:

The Home page has a loading time of 4.8367 seconds. Although this time is below the 5-second threshold, it is close to the upper limit. While the performance is acceptable, optimizations are recommended to improve user experience and reduce the loading time to a more ideal level.

The Contact page shows a loading time of 4.42 seconds. This time is similar to that of the Home page, suggesting that while it is acceptable, there is room for improvement. Optimizing this page could bring its loading time closer to the ideal 2 seconds, enhancing efficiency and user satisfaction.

The Location page has a loading time of 4.3433 seconds. This performance is comparable to that of the Contact page, indicating acceptable performance but with considerable room for optimization.

The About Us page has a loading time of 4.29 seconds. Although this time is slightly better than the previous times, it is still within a range where optimization can provide significant improvements in user experience.

The Register and Login page has the best loading time among the evaluated pages, with 4.057 seconds. While this time is closer to the acceptable threshold, it could also benefit from additional optimizations to achieve ideal loading times.

All evaluated pages have acceptable loading times, but none reach the ideal target of 2 seconds. It is recommended to carry out further optimizations to improve loading speed and, consequently, the overall user experience on the website.

B. Recommendations

Image and Resource Optimization: Compress images and minimize the size of loaded resources.

Use of CDN (Content Delivery Network): To improve loading speed by reducing the distance between the server and the user.

Minification of CSS and JavaScript: Reduce the size of CSS and JavaScript files to speed up loading.

Improve Browser Caching: To store versions of pages and reduce loading times for repeat visits.

Hosting Evaluation: Ensure that the hosting server is capable of efficiently handling traffic and page load.

VIII. Conclusion

This methodology ensures a comprehensive and detailed evaluation of the ”Your Travel Guide” software, utilizing recognized tools and techniques in Software Engineering. The application of international standards and quantitative approaches allows for obtaining objective and reliable data, which will contribute to enhancing the quality and performance of the software.

The functional evaluation based on ISO/IEC 25010 and NMDS analysis provided a structured methodology to measure and analyze the performance of the ”Your Travel Guide” software. This evaluation highlighted specific areas needing improvement, ensuring optimal performance and a better user experience. All provided load times fall within the acceptable range (under 5 seconds), but none approach the ideal time of under 2 seconds. According to the ISO/IEC 25010 standard, optimizing these times could significantly enhance service quality and user experience [16].

The NMDS analysis has revealed important patterns in the load times of different pages on the website. The Home page shows a clear need for optimization to better align with the performance of the other pages. The Contact, Location, and About Us pages, although similar, could also benefit from additional improvements. The Register and Login pages, while performing the best, still have potential for minor optimizations.

References

[1] Whittaker J. A. *How to Break Web Software: Functional and Security Testing of Web Applications and Web Services*. Addison-Wesley Professional, 2010. URL: https://www.pearson.com/store/p/how-to-break-web-software-functional-and-security-testing-of-web-applications-and-web-services/P100000133091/9780321369444.

[2] Kitchenham B. "Evaluating software engineering methods and tools". In: *ACM SIGSOFT Software Engineering Notes* 21.1 (1996), pp. 12–21. URL: https://dl.acm.org/doi/10.1145/225714.225717.

[3] Shneiderman B. and Plaisant C. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson, 2010. URL: https://www.pearson.com/store/p/designing-the-user-interface-strategies-for-effective-human-computer-interaction/P100000677492/9780134380384.

[4] Moazzam Khan - Zehui Bi and John y A. Copeland. "Software Updates as a Security Metric Passive identification of update trends and effect on machine infection". In: *School of Electrical and Computer Engineering Georgia Institute of Technology Atlanta USA)* (2013), pp. 1–6. URL: https://ieeexplore.ieee.org/document/6415869.

[5] Calero C. and Piattini M. *Web Engineering: Modelling and Implementing Web Applications*. Springer, 2015. URL: https://www.springer.com/gp/book/9783642097211.

[6] Google Developers. "PageSpeed Insights". In: *Google* (2023). URL: https://developers.google.com/speed/pagespeed/insights/.

[7] Google Chrome Developers. "Chrome DevTools". In: *Google* (2023). URL: https://developer.chrome.com/docs/devtools/.

[8] Kaner - C. Falk - J. and Nguyen - H. Q. *Testing Computer Software*. John Wiley Sons, 1999. URL: https://www.wiley.com/en-us/Testing+Computer+Software,+2nd+Edition-p-9781118030813.

[9] Kirk J. "Performance Metrics for Web Applications". In: *International Journal of Web Information Systems* 10.4 (2014), pp. 378–395. URL: https://www.emerald.com/insight/content/doi/10.1108/IJWIS-12-2014-0033/full/html.

[10] Nielsen J. *Usability Engineering*. Morgan Kaufmann, 2012. URL: https://www.elsevier.com/books/usability-engineering/nielsen/978-0-12-518406-9.

[11] Bocchi E. - Mellia M. and Russo F. "Estimating the impact of speed index on user engagement". In: *IEEE Transactions on Network and Service Management* 13.3 (2016), pp. 432–445. URL: https://ieeexplore.ieee.org/document/7588002.

[12] JULIETA CALABRESE – ROCÍO MUÑOZ. "ASISTENTE PARA LA EVALUACIÓN DE CALIDAD DE PRODUCTO DE SOFTWARE SEGÚN LA FAMILIA DE NORMAS ISO/IEC 25000 UTILIZANDO EL ENFOQUE GQM". In: *UNLP — FACULTAD DE INFORMÁTICA* 1 (2018), pp. 1–124. URL: https://sedici.unlp.edu.ar/bitstream/handle/10915/67212/Documento_completo__.pdf-PDFA.pdf?sequence=1.

[13] Robbins J. N. *Web Performance Tuning: Speeding Up the Web*. O'Reilly Media, Inc., 2012. URL: https://www.oreilly.com/library/view/web-performance-tuning/1565923790/.

[14] Denis Peppino Guido Tebes - Santiago Nicolau M.F. Papa - Pablo Becker - Luis Olsina. "Aplicacion de una estrategia integrada de mejora que considera requisitos funcionales y no funcionales". In: *Simposio Argentino de Ingeniería de Software* 1 (2018), pp. 1–12.

[15] Julieta Calabrese Rocío Muñoz - Ariel Pasini Silvia Esponda - Marcos Boracchia Patricia Pesado. "Asistente para la evaluación de características de calidad de producto de software propuestas por ISO/IEC 25010 basado en métricas definidas usando el enfoque GQM". In: *Instituto de Investigación en Informática LIDI (III-LIDI)* Facultad de Informática – Universidad Nacional de La Plata 50 y 120 La Plata Buenos Aires* 1 (2017), pp. 1–12. URL: https://sedici.unlp.edu.ar/bitstream/handle/10915/63778/Documento_completo.pdf-PDFA.pdf?sequence=1.

[16] Basili V. R. and Rombach H. D. "The TAME project: Towards improvement-oriented software environments". In: *IEEE Transactions on Software Engineering* 14.6 (1988), pp. 758–773. URL: https://ieeexplore.ieee.org/document/1659407.

[17] Pressman R. S. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 2005. URL: https://www.mheducation.com/highered/product/software-engineering-practitioners-approach-pressman-darcey/M9780073375977.html.

[18] José del Sagrado - Isabel María del Aguila - Alfonso Bosch. *Expansión cuantitativa del método MoSCoW para la priorización de requisitos*. Springer Science Business Media, 2018. URL: https://repositorio.ual.es/bitstream/handle/10835/6110/jis.pdf?sequence=1isAllowed=y.

[19] International Organization for Standardization. "ISO/IEC 25010: Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models". In: *ISO* (2011), pp. 1–24. URL: https://www.iso.org/standard/35733.html.

[20] Ken Schwaber - Jeff Sutherland. "La Guía Scrum". In: *publication is offered for license under the Attribution Share-Alike license of Creative Commons* 1 (2020), pp. 1–17. URL: https://objetivoscrum.com/wp-content/uploads/2021/01/2020-Scrum-Guide-Spanish-European-2.0_objetivoScrum.pdf.

[21] Fielding R. T. and Taylor R. N. "Principled design of the modern web architecture". In: *ACM Transactions*

on Internet Technology (TOIT) 2.2 (2000), pp. 115–150. URL: https://dl.acm.org/doi/10.1145/337180.337228.

[22] Arpan Bhattacharjee Hamza Mahmood- Sidi Lu† - Nejib Ammar- Akila Ganlath y Weisong Shi. "Edge Assisted Over the Air Software Updates". In: *Universidad de Delaware DE EE UU* (2018), pp. 1–10. URL: https://weisongshi.org/papers/Bhatta23-OTA.pdf.

[23] Preece J. - Rogers Y. and Sharp H. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley - Sons, 2015.