Software Architecture for Planning Educational Scenarios by Applying an Agile Methodology

Carlos Andrés Tavera Romero (✉)
Universidad Santiago de Cali, Cali, Colombia
carlos.tavera00@usc.edu.co

Jesús Hamilton Ortiz
Closemobile R&D Telecommunications LS, Fuenlabrada–España

Osamah Ibrahim Khalaf
Al-Nahrain University, Baghdad, Iraq

Wadys Montilla Ortega
Fundación Universitaria de Popayán, Popayán, Colombia

Abstract—This paper focuses on research related to the difficulties introduced by academic resources management in institutions of Higher Education and specifically in the Fundación Universitaria de Popayán, a private university, where the planning of academic scenarios is performed manually at the beginning of each academic period for an average of 1,776 subjects and suitable location for an average of 8,000 students, which leads to determine the importance of considering a software architecture that theoretically supports software development, physically and logically, allowing development in a successful and reliable way that reduces errors, costs, and time in the results, specifically for the case study at the Fundación Universitaria de Popayán. Therefore, the research has made it possible to find the need for an architectural vision and a base for the effective development of software, supported by the different views of authors with years of research on the subject. The research itself is supported by the agile design thinking methodology, the basis for meeting the needs of the end user, and where the quality attributes workshop method is included in the prototyping phase, which guarantees to involve stakeholders in advance for the control of quality attributes in the project. This article shows the flaws that arise in the academic scenario management process and the importance of establishing, from the architecture and agile methodology, a process that remedies the difficulties of the process performed.

Keywords—Educational setting, software architecture, resource planning, educational management

1 Introduction

Fundación Universitaria de Popayán (FUP) is a private Higher Education institution, since its creation in 1982. It has promoted the training of competent professionals committed to the development of the region. To achieve this, it has focused its efforts on the continuous improvement in its administrative and academic processes.

Since 2006, the FUP began to show a vertiginous growth of students, teachers, and administrative staff, starting from 531 students to more than 8,000 students until now, who belong to the undergraduate, postgraduate, and virtual programs, and starting from as less as 60 employees and to reaching to more than 800 teaching and administrative staff. Additionally, 14 years ago the institution had two offices, one in the historic sector of the city of Popayán Cauca and another situated 8 km south of the city in the rural area of Timbio Cauca, where the main campus is situated.

The increase in students, as shown in Figure 1, requires the institution's attention to the physical infrastructure to guarantee the attention of the student staff, as well as the teaching staff. As of today, FUP has nine locations, seven in the city of Popayán Cauca, one 8 km south, near the rural area of Timbio Cauca Municipality, and one in the Santander de Quilichao Cauca Municipality.
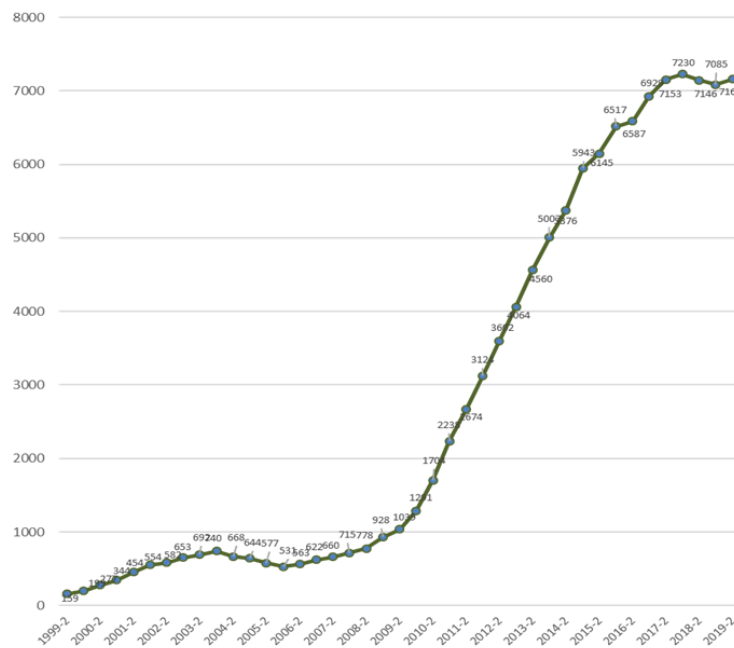

Fig. 1. History of students enrolled in the Fundación Universitaria de Popayán
Source: Own elaboration

Due to this, it became evident from the past that there is an urgent need to guarantee that the spaces for the normal development of academic activities are planned in an

efficient way; however, in each period, the timely distribution at the beginning of each semester becomes more complex. [1].

The Colombian Technical Standard NTC 4595, reads in one of its paragraphs: The regulations on facilities also include the most recent legal provisions on earthquake resistance and accessibility, which are related to the students' safety and the particular conditions of the people with limitations. [2].

The difference in the number of students per program demands the need of space and capacity for them, which is specific for each study plan; therefore, it is necessary to rotate groups of students by venues according to the capacity of the classrooms. Hence, the National Accreditation Council (CNA, by its acronym in Spanish) in its guidelines for the accreditation of undergraduate programs defines factor No. 2 in its characteristic No. 5, Admitted students and institutional capacity: The number of students entering the program is compatible with the capacities that the institution and the program have to ensure those admitted possess the necessary conditions to advance their studies until their completion, and this is how it evaluates the Institutional Policies for the definition of the number of Students admitted to the Program, in accordance with the available faculty members, physical resources and academic support. [3].

Parallel to institutional growth, the manual planning and the process for assigning spaces for the academic activities becomes more complex; initially performed on planner calendar-style paper sheets, as shown in Figure 2, where each subject is recorded every day in classrooms, with its respective teacher, according to the requirements of class schedules by program. There is an average of 55 subjects per academic program, for 13 academic programs, when a single group is planned; however, for large programs (that plan double or triple groups), the registration becomes more expensive, since the subjects are doubled or tripled.

Fig. 2. Sheet for manual registration of classrooms. File transcript

As of 2012, a planner-type format is designed in a spreadsheet (Figure 3), with the difference that it is completed directly on the computer, facilitating copying and pasting

in the case of the same subjects. For the rest, the process remains manual; a spreadsheet is filled in to automatically generate the information in another one.

In the spreadsheet filling process, one person per program intervenes, generally the program subprincipal, who requests the space taking into account the demand of students and their characteristics (students with limitations, right-handed students, left-handed students, practical class, and theoretical class, among others). If the space with the required characteristics is not available, he/she must change or shift subjects and/or teachers in the class schedule so as to match the suitable availability; a situation that delays and makes the process even more difficult.

It is clear that FUP presents a deficiency in the spaces assignment process that averts accelerate and makes the planning phase more efficient in terms of projection thereof, according to the requirement of the academic community; this without considering that the process is error prone.

What has been described shows that the process of planning spaces for classrooms demands an average of 15 days and sometimes more time by the staff involved, and it is becoming increasingly complex given the continuous growth of the institution.

Fig. 3. Template for space allocation
Source: Frame taken directly from Institutional Excel spreadsheet. 2019.

This article presents the stages in which the research was performed, starting with the software architecture [4], the design thinking [5] methodology phases used, and the conclusions of the research.

2 Materials and Methods

2.1 Software architecture

Since the emergence of terminology associated with software architecture (SA) in the 1960s, and its consolidation and dissemination in the 1990s, SA has been a key element in software development, because it understands the way in which applications or designs are built, both in their physical and logical aspects; therefore, it is considered a fundamental stage in the construction of the software and describes the way in which a system is structured and the satisfaction capacity based on the needs of the end user [6].

The importance of architecture lies in the fact that it is a look at the system as a whole with its main components, the behaviour of those components, and the way they interact and coordinate to achieve the mission of the system. The architectural view is an abstract view, providing the highest level of understanding and the suppression or deferral of detail, inherent in most abstractions.

The SA has several models, namely: Stages in engineering and object-oriented design [7], structural models, pattern-based architecture [8], process architecture, scenario-based architecture, dynamic models, process models, architectural patterns[9], event-driven architecture, and the data access object pattern.

The architecture oriented to customer services (service-oriented architecture SOA [10]) is a concept of SA where the software consists of a set of services, features, and rules, and it is the business requirements that dictate the way in which they interrelate. It is designed so that the system is highly scalable and flexible to new architecture requirements. Due to the dynamics of the academic scenario planning process, it adapts itself to the needs, as shown in Figure 4.

2.2 Design thinking methodology

The "thinking about design" methodology [11], as should in Figure 5, allows the actors engaged in the problem to be involved in the whole process, avoiding improvisation in the expected results and, thus, guaranteeing a higher percentage of success in the project result.
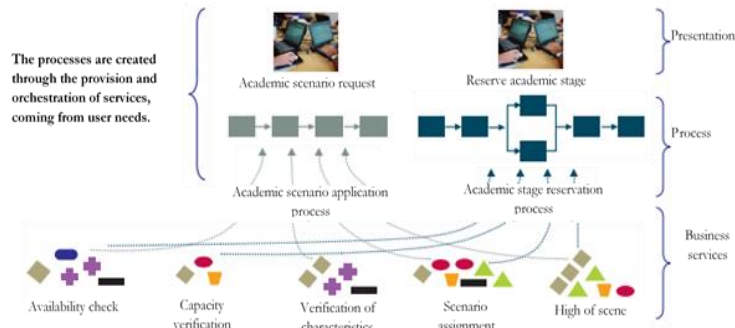
Fig. 4. Architecture-planning academic scenarios.
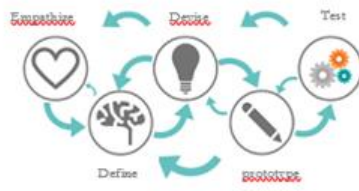
Fig. 5. Design thinking method

This methodology focuses on innovation, which generates greater value in products and services, and due to its application, solutions are generated that otherwise might not have been implemented, allowing developers to "put themselves in the shoes" of their customers or users, unlike the traditional process that focuses on the product to define the experience, a methodology in which everything revolves around the user experience, fundamental to achieve an efficient solution.

Considering the above, the methodology allowed to effectively learn/understand the users' reactions when interacting with the process, the basis for knowing more about the solution, which allowed creating new paradigms based on critical thinking and creativity.

A key point of the methodology is that it allows us to focus on reality, to understand the consumer's needs, to ask questions whose answers will be the solutions that the user requires, and exposing a work team to accelerated but consistent rhythms, leading to results in the short term and viable solutions.

The selected methodology allows achieving successful solutions, as it is a collective creation where the contribution of each professional is important for the construction of the final idea; this forces us to feel the needs of the consumer, hence, the generation of prototypes under the dynamics of trial and error allows each version to be better than the previous versions.

In Figure 6, the mental map extracted from the description of the process is displayed and a high number of variables are identified to be considered by the person in charge of guaranteeing an optimal management of resources.

2.3 Quality attributes workshop method

The quality attributes workshop (QAW) method is included in the prototyping phase of the agile methodology [12] (see Figure 7) to involve the stakeholders from the beginning of the development project and in a way identify the quality attributes. Its execution implies a coordinator and the stakeholders.
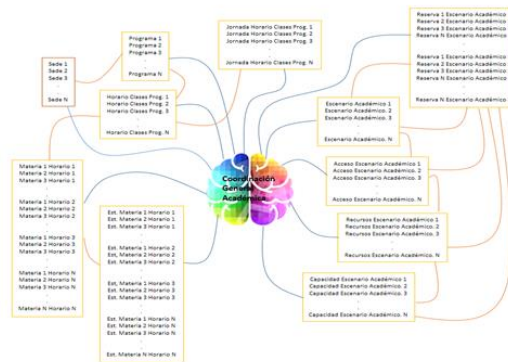
Fig. 6. Academic scenario management mental map
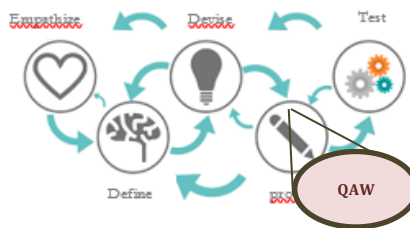
Fig. 7. QAW Method Linkage in design thinking methodology

3 Results and Discussion

3.1 Differences between architecture and design

The SA community states that architecture is significantly opposed to design, although some authors such as Taylor and Medvidovic [13] indicate that current philosophy preserves the articulation between these fields in a state of diverse interpretation, preserving the following different viewpoints:

1. The one that maintains the view that architecture and design are the same
2. Architecture at a level of abstraction that goes beyond design
3. Architecture as something new that to some extent differs from design.

These same authors consider that the architecture abstraction level that exceeds design is the closest to reality and that to some extent the two of them lead to the same purpose.

It is clear that organizations are already aware that the extra cost of software development requires a control intervention and that many of the benefits offered by the methodologies have not been evidenced. However, architecture is conceived as something more articulated than the sum of analysis and design separately. SA is the first step in the production of a software design in a sequence that distinguishes three steps:

1. Architecture: It relates the capabilities of the system expressed in the requirement with the components of the system that will implement it.
2. Code design: It contemplates the phase from algorithms and data structures and understands its components in this primitive from the programming language. It contemplates the phase from the algorithms and the data structures and understands its components in this primitive form from the programming language [14].
3. Executable design: A lower level of detail related to code design that addresses situations such as memory allocation and data formats.

3.2 Software architecture relevance

The SA has become important because it reduces costs, avoids errors, and has implemented mission critical systems. In summary and according to expert opinions, the virtues of SA are:

1. Mutual communication: Referring to the representation of a considerable level of common abstraction that, if not all participants, are able to use as a support to reach mutual understanding, teamwork, and effective communication with each other.
2. Early design decisions: SA as the virtue of the earliest final design determination on a system.
3. Constructive constraints: Representation of how the partial plans for development will be distributed, identifying the components and the dependencies between them.
4. Reuse or transferable abstraction of a system: SA represents a significantly small archetype from the point of view in which a system is organized and its components are articulated among themselves.
5. Evolution: The SA can measure the evolution of the system.
6. Analysis: From the point of view of verifying the consistency of the system, the architectural process generates new opportunities for analysis, highlighting compliance with quality attributes, compliance with the conditions required by a style, dependency analysis and domain, and business specific analysis.
7. Administration: Success stories show that project success lies in a viable architecture, which is key to the industrial development process.

The SA is, by all accounts, still in a formative stage. Its theorists are not yet in a position to assure (as the Three Amigos did).

Once the architectures have been analysed, based on the project where it will be applied, it is determined that the service-oriented architecture (SOA) adjusts to the needs of software development for classroom management [15]; the architecture is approached from the architecture method proposed by the SEI, based on QAW, from the point of view of the early involvement of stakeholders in the project, from which a higher quality is guaranteed in the process.

Regarding the design thinking methodology, it is of great importance to consider that the tool focuses on innovation, which generates greater value in products and services, and due to its application, solutions are generated that otherwise might not have been implemented, allowing developers to "put themselves in other's shoes" of their customers or users, unlike the traditional process that focuses on the product to define the experience. In design thinking, the methodology revolves around the user experience, which is essential to achieve an efficient solution.

This means that the methodology allows to efficiently learn/understand user, interpret their reactions when interacting with the prototype, and the basis for knowing more about the product or the solution, hence, creating new paradigms based on critical thinking and creativity. A key point of the methodology is that it allows to focus on reality, to feel the consumer's needs, to ask questions whose answers would be the same as those asked by the user, subjecting a team to accelerated rates, and leading to short-term results and viable solutions, which allowed to lead to success, as it is a collective creation where the contribution of each professional is important for the construction of the final idea. This led to understanding the needs of the user, which no other method did; hence, the generation of prototypes under trial/error dynamics allowed each version to be better than the previous version.

The first phase of the methodology was approached from the challenge of understanding the needs of students, teachers, some administrative staff, and specifically the person in charge of the space planning. Additionally, it allowed us to understand problems, such as the needs of the academic programs, the needs of adequate spaces to the demand of students, special population characteristics of the population, needs of the students and/or teachers, given the fact that the campus are located in old construction sites, they lack an elevator or escalator to access the second floor, and theoretical or practical classes, in general terms the context where the problem occurs. With participation in the process, it was evidenced that the dedication time for the management of academic scenarios in accordance with the requirements of the academic programs took longer than it should imply; the process has its beginning in contemplating the demand of students in the different subjects to later organize the academic schedules and activities that demand time and that are the basis for starting the process of assigning academic scenarios, which does not allow starting the allocation of resources further delaying the assignment; this without mentioning that the academic schedules consolidation culminated on the weekend prior to the academic activities start [16], which means that the person in charge of the space distribution process has only 2 days to plan the university resources, an amount that is described in Table 1.

The total number of subjects shown in Table 1 does not include twice as many subjects that must be programmed by dividing groups by number of students; similarly, the

opening of new postgraduate programs between periods increases the number of modules to be programmed.

Table 1. Number of programs/subjects per day

Working day	Number of Programs	Number of subjects
Morning	13	1083
Late	2	101
Night	10	693
Weekend	1 (undergraduate)	55
Weekend	4 (postgraduate)	47
Total	30	1979

Mathematical formulas for calculating the number of subjects:

Journeys =

Journey$_1$	$\overset{def}{=}$	Morning
Journey$_2$	$\overset{def}{=}$	Late
Journey$_3$	$\overset{def}{=}$	Night
Journey$_4$	$\overset{def}{=}$	Weekend

$$Subject{\times}Journey\left(Program_i,\ Journey_j\right) \tag{1}$$

$$TOTAL\_Subjects{\times}Journeys \quad = \quad \sum_{\substack{1\leq i \leq n \\ 1\leq j \leq 4}} Subject{\times}Journey\left(Program_i, Journey_j\right) \tag{2}$$

Considering the above table, and taking into account the chaos that is generated by the lack of adequate space planning for the academic activities, both for students and teachers, in addition to the fact that the classroom capacity is uncertain, some schedules intersect that time is wasted in trying to find available places according to the groups characteristics; it leads us to think that the process begins with the adequate class schedules organization for a correct spaces planning. The main problem is that teachers and students cannot meet their objective in each subject due to the lack of guarantee for the normal development of their activities. To overcome this issue, the idea of providing an adequate SA that supports the development of a program performing the process in an automated way arises and that each academic program has access and control of the reservation according to their needs.

Although the project seeks quick solutions, many ways of approaching the solution are proposed, replacing agile methodologies with traditional methodologies, concluding that agile methodologies allow organizing what is going to be performed, and troubleshooting possible failures and looking for ways to prevent them. The above without forgetting that factors such as time, scope, and cost must be taken into account; in this sense the tools related to agile and predictive methodologies are selected [12]. However, starting from the design to arrive at an efficient solution, it guarantees that the content is refined step by step until it ends in a solution that meets the team's objectives, and being ambitious, exceeds them [17].

Based on the interest of the users, a prototype is built as a tool to learn as much as possible from the users in the shortest possible time and, thus, adapt the final product to solve a real problem. This makes the purpose of the investigation tangible and provides learning in a construction phase. The result is based on the system design prototype that would guarantee communication with users and on which the success of the next phase of development to be executed in the short term would depend. Considering the need to show more clearly the quality of both the process and the result, the QAW method is involved in the prototyping phase, which guarantees to follow up on the design phase and greater effectiveness in the expected result.

The QAW method allowed higher quality in the attributes, since under the leadership of a development project coordinator the dynamics of the stakeholders is linked and understood, with whom the project phase begins and ends according to their needs.

Finally, the process is documented. The idea evolves and becomes the solution. A way of introducing the tools to enhance the productivity of the actors is devised. Because it is an operational prototype, considering the isolation generated by Covid-19, the evaluation or testing phase is performed from the perspective of three experts.

4   Conclusion

Based on the theorization of different authors about SA, it can be concluded that it is important to start with a successful SA to guarantee the resources and time saving in the development phase. This could be demonstrated empirically and theoretically through on-site experiments.

Thanks to the fact that the proposed model successfully faced modifications in the parameters and taking into account the different variables, for the case study of the FUP, it can be seen that the applicability of a dynamic model would guarantee the adaptation to the changes of the system.

Considering the business model of the case study, it is evident that the architecture that best fits is SOA and it becomes a highly recommended model for Higher Education institutions to migrate their applications to the cloud computing paradigm, because with the help of the service security, better managed and more efficient use of resources can be executed.

The research for this article shows that the result of a good application depends on a good development and an accurate modelling with a good SA.

The QAW method guarantees a quality cycle in the development of the software from the architecture; this configures an ideal environment to perform improvement of the software development process.

5   Acknowledgement

6   References

[1] D. Ifenthaler and J. Y.-K. Yau, "Reflections on Different Learning Analytics Indicators for Supporting Study Success," Int. J. Learn. Anal. Artif. Intell. Educ., vol. 2, no. 2, p. 4, 2020, https://doi.org/10.3991/ijai.v2i2.15639

[2] Ministerio Colombiano de Educación, "Planeamiento y diseño de Instala-ciones y ambientes escolares," Bogotá, 2020. [Online]. Available: https://www.mineducacion.gov.co/1759/articles-355996_recurso_10.pdf

[3] Consejo Nacional de Acreditación, "Lineamientos para la acreditación de programas de pregrado," Bogotá, 2013. [Online]. Available: https://www.cna.gov.co/1741/articles-186359_pregrado_2013.pdf. https://doi.org/10.18041/2382-3240/saber.2020v15n1.6311

[4] I. Ungurean and N. C. Gaitan, "A software architecture for the industrial internet of things—a conceptual model," Sensors (Switzerland), vol. 20, no. 19, pp. 1–19, 2020, https://doi.org/10.3390/s20195603

[5] H. F. Martins, A. C. de Oliveira, E. D. Canedo, R. A. D. Kosloski, R. Á. Paldês, and E. C. Oliveira, "Design thinking: Challenges for software requirements elicita-tion," Inf., vol. 10, no. 12, pp. 1–27, 2019, https://doi.org/10.3390/info10120371

[6] P. Lu, X. Cong, and D. Zhou, "E-learning-oriented software architecture de-sign and case study," Int. J. Emerg. Technol. Learn., vol. 10, no. 4, pp. 59–65, 2015, https://doi.org/10.3991/ijet.v10i4.4698

[7] J. Bräuer, R. Plösch, M. Saft, and C. Körner, "Measuring object-oriented de-sign principles: The results of focus group-based research," J. Syst. Softw., vol. 140, pp. 74–90, 2018, https://doi.org/10.1016/j.jss.2018.03.002

[8] C. Fehling, F. Leymann, J. Rütschlin, and D. Schumm, "Pattern-Based Development and Management of Cloud Applications," Futur. Internet, vol. 4, no. 1, pp. 110–141, 2012, https://doi.org/10.3390/fi4010110

[9] R. S. Kalawsky, D. Joannou, Y. Tian, and A. Fayoumi, "Using architecture patterns to architect and analyze systems of systems," Procedia Comput. Sci., vol. 16, pp. 283–292, 2013, https://doi.org/10.1016/j.procs.2013.01.030

[10] T. Mahmoud, "Integration of Lightweight Semantic Enterprise Service-Oriented Architecture within Corporate Environmental Management Information Systems," Int. J. Recent Contrib. from Eng. Sci. IT, vol. 1, no. 1, p. 58, 2013, https://doi.org/10.3991/ijes.v1i1.2933

[11] D. C. Chou, "Applying desing thinking method to social entrepreneurship project," Comput. Standars Interfaces, vol. 55, pp. 73–79, 2018, https://doi.org/10.1016/j.csi.2017.05.001

[12] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, "Agile software development: Methodologies and trends," Int. J. Interact. Mob. Technol., vol. 14, no. 11, pp. 246–270, 2020, https://doi.org/10.3991/ijim.v14i11.13269

[13] R. N. Taylor, N. Medvidovic, and E. M. Dashofy, Software Architecture: Foundations, Theory and Practice, 1st ed. Wiley, 2009.

[14] J. Besson and R. Foerch, "Large scale object-oriented finite element code de-sign," Comput. Methods Appl. Mech. Eng., vol. 142, no. 1–2, pp. 165–187, 1997, https://doi.org/10.1016/s0045-7825(96)01124-3

[15] M. Napitupulu, D. K. Walanda, D. Poba, and S. H. V. Pulukadang, "Ace chemistry classroom management with LMS schoology," Int. J. Interact. Mob. Technol., vol. 14, no. 12, pp. 179–185, 2020, https://doi.org/10.3991/ijim.v14i12.15585

[16] C. Zan, "A distributed distribution and scheduling algorithm of educational resources based on vector space model," Int. J. Emerg. Technol. Learn., vol. 14, no. 4, pp. 58–72, 2019, https://doi.org/10.3991/ijet.v14i04.10132

[17] A. R. Bayanova, V. Tugun, K. G. Erdyneeva, N. A. Mashkin, and Z. M. Sakhipova, "The Opinions of Technology Supported Education of University Stu-dents," Int. J. Emerg. Technol. Learn., vol. 15, pp. 4–14, 2020. https://doi.org/10.3991/ijet.v15i23.18779

7   Authors

Carlos Andrés Tavera Romero from Research Group COMBA I+D in Faculty of Engineering of the Universidad Santiago de Cali, Colombia.

Jesús Hamilton Ortiz is CEO of Closemobile R&D Telecommunications LS, Fuenlabrada, España.

Osamah Ibrahim Khalaf from Al-Nahrain University, Al–Nahrain Nanorenewable Energy Research Center, Baghdad, Iraq.

Wadys Montilla Ortega is a Professor and General Academic Coordinator in Fundación Universitaria de Popayán (FUP), Popayán, Colombia.