INTEGRATED MANAGEMENT PLATFORM WITH FLUTTER AND FIREBASE TECHNOLOGIES AS AN ENHANCEMENT IN ACADEMIC PROCESSES

INTEGRATED MANAGEMENT PLATFORM WITH FLUTTER AND FIREBASE TECHNOLOGIES AS AN ENHANCEMENT IN ACADEMIC PROCESSES

Velázquez Santana Eugenio César[1], Velázquez Velazco Marco Aurelio[2], Tiburcio Zúñiga Mario Humberto[3], Muñiz Rascado Luis José[4], Aragón Hernández José Pedro[5]https://doi.org/10.61117/ipsumtec.v8i2.376

[1] National Technological Institute of Mexico/Zacatepec Technological Institute,eugenio.vs@zacatepec.tecnm.mx , https://orcid.org/0000-0002-7683-7224.
[2] National Technological Institute of Mexico/Zacatepec Technological Institute,marco.vv@zacatepec.tecnm.mx , https://orcid.org/0009-0008-8235-8909.
[3] National Technological Institute of Mexico/Zacatepec Technological Institute,mario.tz@zacatepec.tecnm.mx , https://orcid.org/0009-0008-8614-4465.
[4] National Technological Institute of Mexico/Zacatepec Technological Institute,luis.mr@zacatepec.tecnm.mx , https://orcid.org/0000-0003-2941-2482.
[5] National Technological Institute of Mexico/Zacatepec Technological Institute,jose.ah@zacatepec.tecnm.mx , https://orcid.org/0000-0002-9858-0159.

Abstract -- The growing complexity of academic and administrative processes in higher education institutions (HEIs) means that they need to optimize their processes to be more efficient and effective. This research presents the development of a state-of-the-art Integrated Management System (IMS), which was designed to meet the needs of the Zacatepec Institute of Technology. The IMS was built using the Flutter Framework and Google's Firebase development platform. The construction of this IMS began with the collection of requirements through interviews with stakeholders, as well as the creation of prototypes, followed by the design and development phases. In addition, agile development principles such as Scrum were used to ensure flexibility throughout the project. The results include improvements to academic processes, as well as improved response times between various actors and departments within the institution.

Keywords: Firebase, Flutter, Google Cloud, Dart.

Abstract -- The growing complexity of academic and administrative processes in Higher Education Institutions (HEI) makes them see the need to optimize their processes to be more efficient and effective. This research presents the development of an Integral Management System (IMS), which was designed to meet the needs of the Instituto Tecnológico de Zacatepec. The IMS was built using the Flutter and Firebase framework. The construction of this IMS began with requirements gathering through stakeholder interviews, as well as prototyping, followed by the design and development phases. Additionally, agile development principles such Scrum were used to ensure flexibility throughout the project. The results include improvements made to academic processes, in addition to improving response time between various actors and departments of the Institution.

Key words – Firebase, Flutter, Google Cloud, Dart.

INTRODUCTION

SGIs have become increasingly important in recent years as organizations strive to optimize their processes and improve efficiency [5]. HEIs are no exception, as they face the challenge of managing complex workflows related to student management, course management, faculty and administrative management, as well as student academic trajectories, among many other tasks. Comprehensive management systems must often be adapted to the unique requirements of each HEI, which is why they require significant adjustments such as integration with existing critical processes, as well as complex approval processes [6].

As a result, there is a need for a comprehensive management system that can meet the unique requirements of HEIs and improve the overall efficiency of their operations [1].

This research was based on a comprehensive analysis of existing work processes within the Zacatepec Institute of Technology, where

Received: June 8, 2025 | Accepted: September 9, 2025 | Published: October 4, 2025

contributions from stakeholders, including administrators, teachers, and students, were considered.

The creation of the comprehensive management system aims to digitize critical processes, ranging from admission, academic performance, and teacher and student evaluation to teacher and course schedule management, among other processes.

The development of this system with the Flutter framework integrates a web and mobile interface that can be accessed from any device [16]. It will incorporate automatic notifications, real-time status updates, and customizable approval processes.

In addition to the above, the comprehensive management system seeks to improve the efficiency of student and teacher operations, primarily by enhancing the overall experience of the Tecnológico de Zacatepec community. Furthermore, the incorporation of automated notifications and customizable approval processes will speed up response times, ensuring that critical tasks, such as teacher schedule management or course changes, are processed efficiently [6].

As such, this research was conducted to contribute to the growing body of knowledge on integrated management systems supported by the use of new information system development technologies, establishing a framework that can be adopted by other higher education institutions within Mexico.

By understanding the complexities of existing academic processes and collaborating with stakeholders, this research created a transformative tool that positively impacts all operations at Tecnológico de Zacatepec and ultimately benefits the technological community. Through innovation and continuous improvement, this specialized system has the potential to set new standards in the field of academic management in higher education in our country, allowing HEIs to thrive in an ever-evolving academic landscape.

The technological tools for building the SGI are described below. First, Flutter is described, which is a software development kit (SDK) created by Google under an open-source license [8]. The latest version of

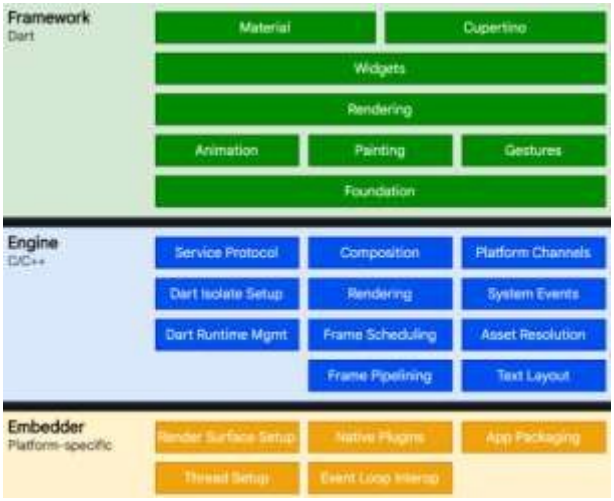
Flutter has three main components: 1.- Embedder (specific to the iOS or Android platform), 2.- Rendering engine, and 3.- Basic library and widgets.

As mentioned in [10 and 16], applications developed in Flutter are written using the object-oriented programming language known as Dart. This programming language runs on a virtual machine written in C/C++. One feature

The unique feature of Flutter technology is that it draws each pixel independently.

In other words, it has collections of internal widgets. Figure 1 describes the components of Flutter.

Figure 1. *Flutter components.*



Source: ItDo, 2025.

Widgets form a hierarchy based on composition. Each widget is placed inside another, from which it inherits properties. There is no separate "application" object; this function is assumed by the root widget.

In Flutter, user interaction occurs by updating the widget hierarchy. This is done by comparing the new widget with the old one, modifying only the different elements. Complex widgets can also be created by combining several simple widgets. The widgets used are not native to the platforms on which the application runs (Android or iOS), but are specific representations.

Widgets are divided into two categories: 1.- Stateless widgets are those whose state does not change (such as text or an icon) and do not depend on other interface components. They also have the function of describing part of the user interface through the recursive creation of other widgets that form the user interface. 2.- Stateful widgets are those whose state changes through user interaction with the application, but also when other application parameters change. That is why each stateful widget has an initial state that contains information about the widget [8].

As a complement to Flutter, Dart has emerged**,** which is an optimized programming language that allows for the rapid development of a solution that can be implemented on multiple platforms.

Received: June 8, 2025 | Accepted: September 9, 2025 | Published: October 4, 2025

As a type-safe language, it uses a combination of static type checking and runtime checks to ensure that the value of a variable always matches the static type of the variable, ensuring that a variable is checked and validated at runtime when the code is executed [4].

Using this paradigm at runtime, Dart performs static code analysis, filtering and eliminating the possibility of generating a null exception. The technologies used by Dart for compilation offer two possibilities for executing code: 1. Native platform, for applications intended to run on mobile and desktop platforms. 2. Web platform, for applications that run in a web browser [2].

For web applications, both compilation modes translate to JavaScript code. Regardless of the method used for compilation and execution, code execution requires the Dart runtime. Multithreaded programming in Dart is achieved through the use of asynchronous wait isolation constructs, as well as dedicated classes.

As mentioned in [13 and 18], code execution within an application is performed within a hybrid execution thread. This abstraction method is called isolation. In Dart, each isolation has a single encapsulated execution thread, i.e., a secure thread.

On the other hand, Firebase is a cloud-based platform that offers numerous back-end services for web and mobile applications. Firebase creates reliable and scalable applications with the goal of allowing users to not worry about the infrastructure [13 and 7].

For the development, testing, and deployment of mobile and online applications, in addition to the above, it provides a wide range of tools and services, including a real-time database, authentication, cloud storage, hosting, cloud functions, and messaging, among other notable features.

Without writing any server-side code, developers can quickly set up a complete back-end for their applications using Firebase [12].

Firebase Database is a cloud-based NoSQL database that synchronizes data from all clients in real time [8]. All clients of cross-platform applications created with JavaScript.

SDKs for Android, iOS, and Android share a real-time database instance and immediately obtain the latest data changes.

When Firebase is disconnected, it saves user data and updates it when the connection is established, thus remaining active even when not connected to the servers. Since no application server is required, Firebase can be accessed through browsers and mobile devices [13].

DEVELOPMENT

The research design used in this study is qualitative in nature, meaning that interviews were conducted with stakeholders and focus groups. Its objective is to collect data on the existing academic processes at the Zacatepec Institute of Technology and identify the key characteristics and requirements for a comprehensive management system.

In addition to the above, the methodology selected for the development of the system was Scrum, which allows for frequent incremental deliveries to customers [13]. Some PMBOK reference framework project management artifacts were also used. With the help of these artifacts, the relevant project documentation was managed, notably the Project Charter, Scope, and Work Breakdown Structure, as well as the Work Schedule for each member of the project team [17].

Based on Scrum and its focus on agile development, it was possible to control customer requirements, as well as the problems that arose during the development of the system [12]. Furthermore, this methodology made it possible to better identify the processes, roles, tasks, and responsibilities of each user, generating a high level of trust among stakeholders, who became active participants during the development cycle, given their proximity to the work team.

On the other hand, Visual Studio Code was selected as the Integrated Development Environment (IDE) because it provides several tools for working directly with Google Cloud, given its simplicity, code reuse, and communication capabilities, among other features that will facilitate the implementation process of the proposed solution [9].

During the project analysis stage, the main set of requirements, called the Product Backlog, was defined. It contains all the functional and non-functional requirements that the system to be built must meet.

Functional requirements:

- Communication. The system must allow communication between team members and stakeholders in the workflow process.
- Workflow automation. The system must automate repetitive and routine tasks to improve efficiency and reduce errors.

Received: June 8, 2025 | Accepted: September 9, 2025 | Published: October 4, 2025

Non-functional requirements:

- Security. The system must have robust security measures in place to protect confidential data and prevent unauthorized access.
- Scalability. The system should be able to handle a growing number of users and workflows as the university department expands.
- Usability. The system should be easy to use and navigate, with clear instructions and an intuitive interface.
- Performance. The system must be fast and responsive, with minimal delay or downtime.

The functional requirements were specified through use cases, user stories, and some relevant data flow diagrams. That is why the Product Backlog was defined in work meetings with stakeholders. From there, iterations or Sprints were defined, in which I would work incrementally until the system was fully built.

Each iteration created its own Sprint Backlog, which contained the various features to be built with their manager and their estimated time. Additionally, it was agreed that the maximum duration of a Sprint would be four weeks and that, within each Sprint, the project leader would call a daily meeting, called a Scrum Daily Meeting, to review tasks completed, tasks to be done, and any problems that had arisen, with the aim of obtaining feedback from the work team. At the end of each Sprint, a Sprint Review was held to evaluate the components built and discuss the planning for the next Sprint.

Google Cloud, as the cloud service provider, provided all the necessary configuration for the development of this system, such as the operating system, data security, hosting, and network capabilities.

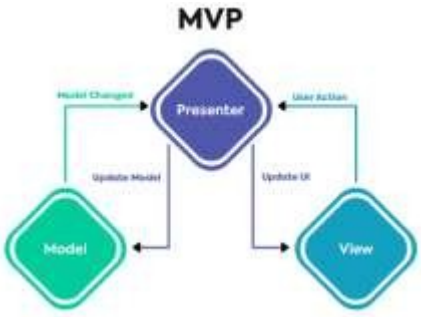
With this platform, the programmers and project leader were able to adopt an iterative way of working, where requirements were concentrated according to the stage of development. With this, the SGI components were delivered as an advanced prototype for customer approval [12].

As a result, the Google Cloud platform supports agile system development in order to improve communication and customer service, thereby reducing the system's delivery time.

In conclusion, it can be seen that the Google Cloud platform provides a collaborative environment when agile methodologies and cloud computing are combined for projects such as this one.

The architecture used for this development was the Model-View-Presenter (MVP) model, which is shown in Figure 2:

Figure 2. *Model-View-Presenter of the SGI.*



Source: MVP, 2025.

In this type of model, it was necessary to identify and separate the functionality between the model, which represents the data layer and business logic of the application, as well as managing access to the database and web services. [14].

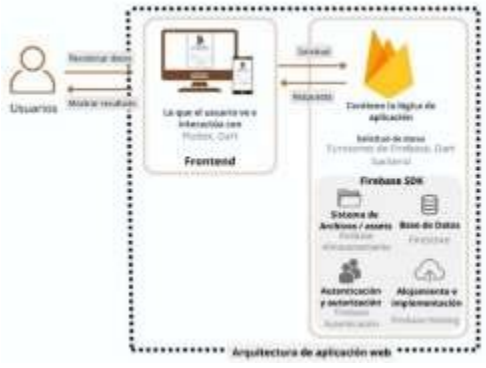
On the view side, there is the presentation layer, which is responsible for displaying the user interface, receiving information from the presenter, and updating the user interface as necessary.

Finally, the presenter layer acts as an intermediary between the model and the view, where it retrieves data from the model and presents it to the view in an appropriate manner.

This model has proven effective in controlling the behavior of the graphical interface. In conclusion, the MVP architecture used by Flutter is definitely the best technology available, as it is open source, saves time, is cost-effective, and integrates seamlessly with Firebase.

Based on the above, the architecture used to develop the application between Flutter and Firebase is shown in Figure 3.

Figure 3. *Web application architecture*.



Source: Own elaboration.

The proposed SGI will follow a client-server architecture, in which the system can be accessed through a web-based interface (client) and will be hosted on a server. The web-based approach ensures broad accessibility and compatibility with various devices, allowing students, teachers, and staff to access the system from their laptops, tablets, or smartphones.

Based on the above, the server side is responsible for processing requests, managing data, and ensuring secure access, while the client side provides an interface for users to interact with the system. Finally, the development environment for this project includes six personal computers with at least 8 GB of RAM, a 64 Core i5 processor, and a 1 TB solid-state drive.

The software components include Flutter SDK, Android Studio, and Visual Studio Code. The development environment is also configured to support continuous integration and deployment using tools such as GitHub and Jenkins.

User authentication was implemented using JSON Web Tokens (JWT). When users log in, the system verifies their credentials and issues a unique token. This token is then used for subsequent API calls to authenticate and authorize user actions. By adopting JWT, the system improves security by preventing unauthorized access to sensitive resources.

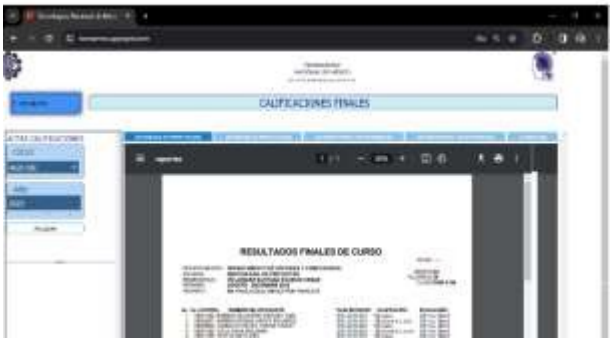
Figure 4 shows the login interface for SGI users. In this case, the figure shows access for a user with the role of teacher.

Figure 4. *Login*.

Source: Own elaboration.

Figure 5 shows one of the teacher's views, where they can review the grade reports recorded in previous semesters. This type of query can be carried out much more efficiently and easily, given that this process was previously done manually and, on occasions, had to be requested in writing from the school services department, which was very time-consuming.

Figure 5. *Transcript*.



Source: Own elaboration.

Figure 6 shows one of the views available to the Human Resources (HR) department, as well as to the teacher, where they can review their current and previous semester work schedules.

It should be noted that this process greatly helps to streamline the validation of schedules between the HR department and each of the academic heads, making this process faster. In addition, this system reduces paper consumption between both departments, since only the sets that are necessary for physical storage are printed, thus avoiding unnecessary printing.

Figure 6. *Faculty schedule*.

Source: Own elaboration.

Figure 7 shows one of the views available to the Academic Development (AD) department, as well as to the teacher themselves, where they can review the evaluations made by their students in the current and previous periods. It should be noted that all this information is generated automatically at the end of the period corresponding to that evaluation. This helps the AD department, together with the academic director, to schedule training courses for the teaching staff based on the results and needs that have emerged from the evaluation.

Figure 7. *Faculty evaluation*.

Received: June 8, 2025 | Accepted: September 9, 2025 | Published: October 4, 2025

Source: Prepared internally.

The SGI used the Flutter framework and the Firebase non-relational database manager. During this work, there were some problems, such as designing the application structure and the appropriate database schema. However, these were overcome with a basic design that was ready to be iterated as many times as necessary. In addition, Firebase's capabilities were used to change the schema without significant interruptions.

Another important challenge was ensuring real-time data synchronization between multiple users and devices. This also posed a challenge, as potential data inconsistencies and conflicts had to be handled, which were resolved using Firestore timestamps and transactions.

In addition to handling user authentication and security to protect sensitive data and ensure that only authorized users could access the system.

In addition to the above, Firebase Authentication was used for user login, and access control mechanisms were implemented using Firebase security rules to define and enforce access restrictions based on user roles and permissions.

Offline support was another essential aspect that had to be resolved, as users needed to access and modify data while offline and synchronize changes once they reconnected. To do this, Firebase's offline capabilities were used to store data locally and automatically synchronize changes when the device came back online.

Testing and debugging were essential to ensure a comprehensive management system with minimal failures. To this end, unit, integration, and system tests were performed at each stage of development. In accordance with Scrum principles, the work was divided into manageable tasks and a milestone schedule was created. Essential functions were prioritized first, and additional functions were implemented when time permitted.

DISCUSSION AND ANALYSIS OF RESULTS

The results obtained after the implementation of the Comprehensive Management System for the Zacatepec Institute of Technology are detailed below, bearing in mind that the main objective of this project is to optimize academic processes in various areas, seeking to reduce response times for various procedures, as well as some costs related to stationery and supplies necessary for the generation of printed documentation.

Quantitative results:

- Reduction in stationery expenses: There was a 15% decrease in purchases of bond paper boxes during the first half of 2024 due to the use of the system, compared to the same period last year. This is attributed to the fact that the SGI generates documents digitally, which facilitates their reading and use.
- Improved accuracy in the issuance of digital documents: The accuracy of digital documentation increased by 20%, avoiding errors in academic processes and in the documents themselves, which translates into a reduction in errors in forms or official documents issued.
- Reduction in the time required for academic processes: The average time required to prepare official documents was reduced by 20% thanks to the automation of academic formats.

Qualitative results:

- Greater visibility of academic processes: The system provides a real-time view of the status of administrative procedures related to student and faculty documentation.
- Improved collaboration between departments: The SGI facilitates communication and information exchange between the different departments involved in managing the academic process for students and teachers, which translates into greater efficiency and coordination between areas.
- Greater satisfaction among the student community: Improved response times and accuracy in documents for student procedures have contributed to increased student satisfaction.

The results obtained show that the Comprehensive Management System has met its objectives, highlighting the increase in the efficiency of academic processes involving students and teachers. It also led to lower stationery costs, more accurate document issuance, and faster information processing, which are clear signs of how the system has helped the tech community as a whole.

Received: June 8, 2025 | Accepted: September 9, 2025 | Published: October 4, 2025

This system has also achieved greater visibility of academic processes, improved collaboration between departments, and increased student satisfaction, highlighting the value of the Comprehensive Management System as a management tool.

Discussion

As mentioned in [1], the new tools provided by Google Cloud have opened up new paradigms for working in the cloud and, as a result, new growth opportunities for various institutions, such as the Zacatepec Institute of Technology.

These new technologies bring with them innovation in the way applications are developed, which, in the case of this work, is hybrid, meaning it works for both the web and mobile devices, whether Android or iOS. Likewise, [6] indicates that, for developers, cloud computing provides a range of services and infrastructure at affordable prices. Higher education institutions now have the opportunity to reduce and even eliminate unnecessary infrastructure costs, which would be reflected in economic savings.

In agreement with [3], who developed the research *Accelerating multi-platform development with Flutter*, and with [15] *Implementation of the Flutter framework for the development of an e-commerce application*. They mention, respectively, that this framework has evolved rapidly, becoming the preferred method for developing multi-platform software. In this research, we agree with what the authors mention, given that the development of the software project was quite fast due to the technology itself and the software development methodology used. In addition, this framework allows applications to be compatible with macOS, Linux, and Windows operating systems.

In addition to the above, the authors of the research [11] *Design and construction of a web-based and mobile device-based final project management system for the University of Tabanan (Simantab) using the Flutter framework*, point out that the system requirements specifications must be modeled with the support of data flow diagrams, which will be used to create the interfaces. This is done in this research through user stories (which are modeled in small diagrams) for each of the Sprints executed during the project.

CONCLUSIONS

For the Zacatepec Institute of Technology, the Integrated Management System incorporates fluid communication between departments, teachers, administrators, and students.

All this is achieved through robust security via user authentication and access control and optimized communication within the academic community.

By implementing this system, the efficiency of its academic processes was improved, in addition to improving communication and maintaining data security, benefiting staff and students.

The development of a comprehensive management system using Flutter and Firebase has demonstrated its potential to revolutionize operations in higher education institutions in our country by providing an easy-to-use interface, automation capabilities, and efficient management of academic processes. In addition, the system offers a valuable solution for improving overall productivity and efficiency.

ACKNOWLEDGMENTS

The authors would like to thank the Zacatepec Institute of Technology for the facilities provided for the preparation of this research paper. In addition, special mention goes to Alma Flores Ramírez, a student in the Computer Systems Engineering program with control number 22090633, who contributed to the development of the Comprehensive Management System.

BIBLIOGRAPHY

[1] Adedapo O.A., Bello J., Ogunrinde M.A., Oyediran M.O., Adio M.O., and Oyekola O.A. Development of a Workflow Management System for Higher Education: Case study of a Department in a Nigerian University. Journal of Engineering and Technology. 2023 December; vol

[2] Arb, G.I. and Al-Majdi, K. A freights status management system based on Dart and Flutter programming language. In Journal of Physics: Conference Series, IOP Publishing. 2020 May; vol 1530(1):012-020.

[3] Arif, S., Preetam, S., Mritunjay, R., Amar, K., Chandrahas, S., Shilpi, S., Prafulla, C. Accelerating Cross-platform Development with Flutter Framework. Journal of Open Source Developments. 2023; vol. 10(2): 1–11p.

[4] Google I/O. Dart Programming. 2025. Accessed February 20, 2025. Available at: https://www.dart.dev

[5] Mohamed, A., Mohamed H., Issam T., Robin M. Higher education strategy in digital transformation. 2021 September; vol. 27(3): 3171-3195. Available at doi: 10.1007/s10639-021-10739-1

[6] Vicenzo, M. Digital Transformation Challenges for Universities: Ensuring Information Consistency Across Digital Services. Cataloging & Classification

Quarterly. January 2019; vol. 56(7): 592–606. Available at: https://doi.org/10.1080/01639374.2018.1504847

[7] Google Firebase. Firebase. Accessed February 15, 2025. Available at: https://www.firebase.google.com

[8] Google Flutter. Flutter. Accessed February 23, 2025. Available at: https://www.flutter.dev

[9] Google Cloud. Accessed March 10, 2025. Available at: https://www.cloud.google.com.

[10] Khedkar, S., Thube, S., Estate, W.I. and Naka, C. Real time databases for applications. International Research Journal of Engineering and Technology (IRJET). June 2017; vol. 4(6): 2078-2082. Available at: https://www.irjet.net/archives/V4/i6/IRJET-V4I6401.pdf

[11] Krisna, D., Oka, G., Sandi, W. Design and Development of a Web- and Mobile-Based Final Project Management Information System (Simantab) for Tabanan University Using the Flutter Framework. Journal of Social Sciences and Humanities Technology. June 2024; vol. 2(2): 114-122. Available at: https://s.id/jurnalsutaso

[12] Michal, Hron. Why and how is Scrum being adapted in practice: A systematic review. Journal of Systems of Software. January 2022; vol. 183. https://doi.org/10.1016/j.jss.2021.111110

[13] Moroney, L. An Introduction to Firebase. The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform. USA. Apresspp Publishing; 2017. Accessed February 20, 2025. 271p. Available at: http:// Definitive-Guide-Firebase-Android-Platform/dp/1484229428

[14] MVP. Model View Presenter. Modified April 25, 2024. Accessed January 10, 2025. Available at: https://keepcoding.io/blog/que-es-el-modelo-vista-presentador/

[15] Nursobah, S., Muhammad, I., & Kansil, J. Implementation of the Flutter Framework for Developing an E-Commerce Application. Journal TEPIAN. 2024. vol. 5(4). 127-135. https://doi.org/10.51967/tepian.v5i4.3110

[16] Payne, R. Developing in Flutter. In: Beginning App Development with Flutter. Berkeley, CA. USA. Apress Publishing. 2019; 309p. Available at: https://doi.org/10.1007/978-1-4842-5181-2_2.

[17] Scrum. Scrum org. What is Scrum? Accessed on March 20, 2025. Available at: https://www.scrum.org.

[18] Tashildar, A., Shah, N., Gala, R., Giri, T., and Chavhan, P. Application development using flutter. International Research Journal of Modernization in Engineering Technology and Science. August 2020; vol. 2 (8):1262-1266. Available at: https://www.irjmets.com/uploadedfiles/paper/volume2/issue_8_august_2020/3180/1628083124.pdf

[19] ItDo. Accessed March 22, 2025. Available at: https://www.itdo.com/blog/flutter-el-nuevo-framework-de-google/

Contribution Role Table

Role	Author(s)
Conceptualization	Marco Aurelio Velázquez Velazco
Data curation	José Pedro Aragón Hernández
Methodology	Mario Humberto Tiburcio Zúñiga
Administration of the project	Luis José Muñiz Rascado
Software	Marco Aurelio Velázquez Velazco
Supervision	Luis José Muñiz Rascado
Visualization	Mario Humberto Tiburcio Zúñiga
Writing (Original draft)	Eugenio César Velázquez Santana
Writing, revision, and editing.	Eugenio César Velázquez Santana

Received: June 8, 2025 | Accepted: September 9, 2025 | Published: October 4, 2025