

PLATAFORMA INTEGRAL DE GESTIÓN CON TECNOLOGÍAS FLUTTER Y FIREBASE COMO MEJORA EN LOS PROCESOS ACADÉMICOS

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

Velázquez Santana Eugenio César¹, Velázquez Velazco Marco Aurelio², Tiburcio Zúñiga Mario Humberto³,
Muñiz Rascado Luis José⁴, Aragón Hernández José Pedro⁵

<https://doi.org/10.61117/ipsumtec.v8i2.376>

¹Tecnológico Nacional de México/Instituto Tecnológico de Zacatepec, eugenio.vs@zacatepec.tecnm.mx, <https://orcid.org/0000-0002-7683-7224>.

²Tecnológico Nacional de México/Instituto Tecnológico de Zacatepec, marco.vv@zacatepec.tecnm.mx, <https://orcid.org/0009-0008-8235-8909>.

³Tecnológico Nacional de México/Instituto Tecnológico de Zacatepec, mario.tz@zacatepec.tecnm.mx, <https://orcid.org/0009-0008-8614-4465>.

⁴Tecnológico Nacional de México/Instituto Tecnológico de Zacatepec, luis.mr@zacatepec.tecnm.mx, <https://orcid.org/0000-0003-2941-2482>.

⁵Tecnológico Nacional de México/Instituto Tecnológico de Zacatepec, jose.ah@zacatepec.tecnm.mx, <https://orcid.org/0000-0002-9858-0159>.

Resumen -- La creciente complejidad de los procesos académicos y administrativos en las Instituciones de Educación Superior (IES), hacen que se vean en la necesidad de optimizar sus procesos para ser más eficientes y eficaces. La presente investigación presenta el desarrollo de un Sistema de Gestión Integral (SGI) de última generación, el cual fue diseñado para atender las necesidades del Instituto Tecnológico de Zacatepec. El SGI se construyó mediante el Framework Flutter y la plataforma de desarrollo Firebase de Google. La construcción de este SGI comenzó con la recopilación de requisitos mediante entrevistas a los interesados, así como la creación de prototipos, seguido de las fases de diseño y desarrollo. Además, se emplearon principios de desarrollo ágil como Scrum, para garantizar la flexibilidad durante todo el proyecto. Como resultados, se incluyen las mejoras realizadas a los procesos académicos, además de mejorar el tiempo de respuesta entre diversos actores y departamentos de la Institución.

Palabras Clave: 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 a 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

as 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.

INTRODUCCIÓN

Los SGI se han vuelto cada vez más importantes en los últimos años a medida que las organizaciones se esfuerzan por optimizar sus procesos y mejorar la eficiencia [5]. Las IES no son una excepción, ya que enfrentan el desafío de gestionar flujos de trabajo complejos relacionados con la gestión de estudiantes, la gestión de cursos, la gestión de docentes y administrativos, así como la trayectoria escolar de los estudiantes, entre muchas otras tareas más. Los sistemas de gestión integral deben adaptarse a menudo a los requisitos únicos de cada IES, es por ello que necesiten adecuaciones importantes como la integración con los procesos críticos existentes, así como con procesos de aprobación complejos [6].

Como resultado, existe la necesidad de un sistema de gestión integral que pueda satisfacer los requisitos únicos de las IES y mejorar la eficiencia general de sus operaciones [1].

La presente investigación se basó en un análisis exhaustivo de los procesos de trabajo existentes dentro del Instituto Tecnológico de Zacatepec, en donde se

consideraron aportes de las partes interesadas, entre ellos los administrativos, los profesores y los estudiantes.

La creación del sistema de gestión integral tiene por objetivo, digitalizar procesos críticos, que van desde la admisión, la trayectoria escolar y, así como la evaluación de docentes a los estudiantes, la gestión de horarios de docentes y de materias, entre otros procesos más.

El desarrollo de este sistema con el framework Flutter, integra una interfaz web y móvil, con la que se podrá acceder desde cualquier dispositivo [16]. Incorporará notificaciones automáticas, actualizaciones de estado en tiempo real y procesos de aprobación personalizables.

Aunado a lo anterior, el sistema de gestión integral busca mejorar la eficiencia de las operaciones de los estudiantes y docentes principalmente, mejorando la experiencia general de la comunidad del Tecnológico de Zacatepec. Además, la incorporación de notificaciones automatizadas y procesos de aprobación personalizables, acelerar los tiempos de respuesta, garantizando que las tareas críticas, como la gestión de horarios docentes o cambios de materias, se procesen de manera eficiente [6].

Como tal, esta investigación se realizó para contribuir al creciente conjunto de conocimientos sobre los sistemas de gestión integral apoyados en el uso de las nuevas tecnologías de desarrollo de sistemas de información, estableciendo un marco que pueda ser adoptado por otras instituciones de educación superior dentro de México [1].

Al comprender las complejidades de los procesos académicos existentes y colaborar con las partes interesadas, esta investigación creó una herramienta transformadora que impacta positivamente todas las operaciones del Tecnológico de Zacatepec, y en última instancia, beneficia a la comunidad del tecnológico. A través de la innovación y la mejora continua, este sistema especializado tiene el potencial de establecer nuevos estándares en el ámbito de la gestión académica en la educación superior de nuestro país, permitiendo a las IES prosperar en un panorama académico en constante evolución.

A continuación, se describen las herramientas tecnológicas para la construcción del SGI. En primer lugar, se describe Flutter, el cual es un kit de desarrollo de software (SDK) creado por Google bajo una licencia de código abierto [8]. La última versión de

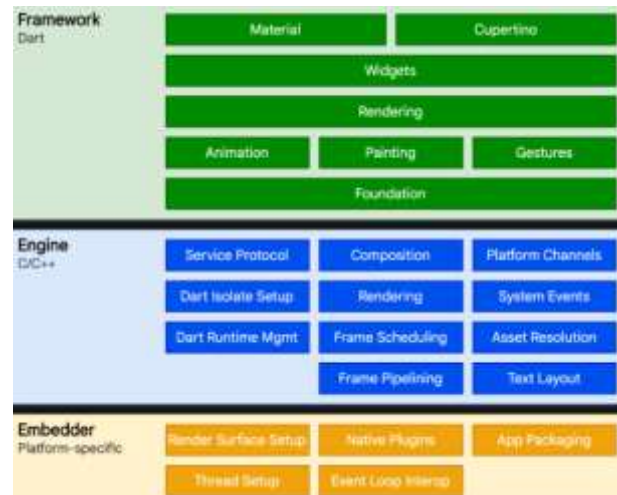
Flutter tiene tres componentes principales: 1.- Embedder (específico para la plataforma iOS o Android), 2.- Motor de renderizado y 3.- Biblioteca básica y los widgets.

Como lo menciona [10 y 16], las aplicaciones desarrolladas en Flutter, se escriben utilizando el lenguaje de programación orientado a objetos conocido como Dart. Este lenguaje de programación se ejecuta en una máquina virtual escrita en C/C++. Una característica

única de la tecnología Flutter, es que dibuja cada píxel de forma independiente.

Es decir, tiene colecciones de widgets internos. En la figura 1, se describen los componentes de Flutter.

Figura 1. Componentes de Flutter.



Fuente. ItDo, 2025.

Los widgets forman una jerarquía basada en la composición. Cada widget se coloca dentro de otro, del cual hereda propiedades. No existe un objeto de "aplicación" separado; esta función la asume el widget raíz.

En Flutter, la interacción con el usuario se da actualizando la jerarquía del widget, esto se hace comparando el nuevo widget con el antiguo, modificándose solo los diferentes elementos. También se pueden crear widgets complejos combinando varios widgets simples. Los widgets utilizados, no son los nativos de las plataformas en las que se ejecuta la aplicación (Android o iOS), sino que son representaciones específicas.

Los widgets se dividen en dos categorías: 1.- widgets sin estado, son aquellos cuyo estado no cambia (como un texto o un ícono) y no dependen de otros componentes de la interfaz, además tiene la función de describir parte de la interfaz de usuario mediante la creación recursiva de otros widgets que forman la interfaz de usuario. 2.- widgets con estado, son aquellos cuyo estado cambia mediante la interacción del usuario con la aplicación, pero también al cambiar otros parámetros de la aplicación, es por ello que cada widget con estado, tiene un estado inicial que contiene información sobre el widget [8].

Como complemento de Flutter, surge Dart, que es un lenguaje de programación optimizado que permite el rápido desarrollo de una solución que puede implementarse en varias plataformas.

Al ser un lenguaje con seguridad de tipos, utiliza una combinación de verificación de tipos estáticos y comprobaciones de tiempo de ejecución para garantizar que el valor de una variable siempre coincida con el tipo estático de la variable, lo que garantiza que una variable se verifique y valide en tiempo de ejecución, cuando se ejecuta el código [4].

Usando este paradigma en tiempo de ejecución, Dart realiza análisis de código estático, filtrando y eliminando así la posibilidad de generar una excepción nula. Las tecnologías utilizadas por Dart para la compilación ofrecen dos posibilidades para ejecutar el código: 1.- Plataforma nativa, para aplicaciones destinadas a ejecutarse en plataformas móviles y de escritorio. 2.- Plataforma web, para aplicaciones que se ejecutan en un navegador web [2].

Para aplicaciones web, ambos modos de compilación se traducen a código JavaScript. Independientemente del método utilizado para la compilación y ejecución, la ejecución del código necesita el tiempo de ejecución de Dart. La programación multiproceso en Dart se realiza mediante el uso de construcciones aisladas de espera asíncrona, así como clases dedicadas.

Como lo menciona [13 y 18], la ejecución del código dentro de una aplicación se realiza dentro de un hilo de ejecución híbrido. Este método de abstracción se llama aislar. En Dart, cada aislamiento tiene un único subproceso de ejecución encapsulado, es decir, un subproceso seguro.

Por otro parte, Firebase es una plataforma basada en la nube que ofrece numerosos servicios back-end para aplicaciones web y móviles. Firebase crea aplicaciones confiables y escalables con el objetivo de que los usuarios no se preocupen por la infraestructura [13 y 7].

Para el desarrollo, prueba e implementación de aplicaciones móviles y en línea, además de lo anterior, proporciona una amplia gama de herramientas y servicios, base de datos en tiempo real, autenticación, almacenamiento en la nube, alojamiento, funciones en la nube y mensajería, son algunas de sus características destacadas.

Sin escribir ningún código del lado del servidor, los desarrolladores pueden configurar rápidamente un back-end completo para sus aplicaciones utilizando Firebase [12].

Firebase Database, es una base de datos NoSQL basada en la nube, que sincroniza datos de todos los clientes en tiempo real [8]. Todos los clientes de aplicaciones multiplataforma creadas con JavaScript.

SDK para Android, IOS y Android comparten una instancia de base de datos en tiempo real y obtienen inmediatamente los cambios de datos más recientes.

Cuando Firebase se desconecta, guarda los datos del usuario y los actualiza cuando se establece la conexión, estando así activo incluso cuando no está conectado a los servidores. Dado que no se requiere un servidor de aplicaciones, se puede acceder a Firebase a través de navegadores y dispositivos portátiles [13].

DESARROLLO

El diseño de investigación utilizado en este trabajo, es de enfoque cualitativo, es decir, se realizaron entrevistas a partes interesadas y grupos focales, y tiene como objetivo recopilar datos sobre los procesos académicos existentes en el Instituto Tecnológico de Zacatepec e identificar las características y requisitos clave para un sistema de gestión integral.

Aunado a lo anterior, la metodología seleccionada para el desarrollo del sistema, fue Scrum la cual permite hacer entregas frecuentes de manera incremental a los clientes [13]. Así también, se utilizaron algunos artefactos para la gestión de proyectos de marco de referencia PMBOK, con ayuda de estos artefactos, se gestionó la documentación relevante del proyecto, destacando el Project Charter, Scope, y Work Breakdown Structure, así como el Work Schedule de cada miembro del equipo de trabajo del proyecto [17].

Basándose en Scrum y en su orientación hacia el desarrollo ágil, se lograron controlar los requisitos del cliente, además de los inconvenientes que se presentaban durante el desarrollo del sistema [12]; además, con esta metodología fue posible identificar de mejor manera los procesos, los roles, las tareas, y las responsabilidades de cada usuario, generando una buena confianza por parte de los interesados, quienes se convirtieron en actores durante el ciclo de desarrollo, dado su cercanía con el equipo de trabajo.

Por otra parte, el Entorno de Desarrollo Integrado (IDE) seleccionado fue Visual Studio Code, dado que provee varias herramientas para trabajar directamente con Google Cloud, dada su simplicidad, su reutilización de código y a su capacidad de comunicación, entre otras características más que facilitarán llevar a cabo el proceso de implementación de la solución propuesta [9].

En la etapa de análisis del proyecto, se definió la pila de requerimientos principal denominada Product Backlog, que contiene todos los requerimientos funcionales y no funcionales que deberá satisfacer el sistema a construir.

Requerimientos funcionales:

- Comunicación. El sistema debe permitir la comunicación entre los miembros del equipo y las partes interesadas en el proceso de flujo de trabajo.
- Automatización del flujo de trabajo. El sistema debe automatizar tareas repetitivas y rutinarias para mejorar la eficiencia y reducir errores.

Requerimientos no funcionales:

- Seguridad. El sistema debe contar con medidas de seguridad sólidas para proteger los datos confidenciales y evitar el acceso no autorizado.
- Escalabilidad. El sistema debería poder manejar un número creciente de usuarios y flujos de trabajo a medida que el departamento universitario se expande.
- Usabilidad. El sistema debe ser fácil de usar y navegar, con instrucciones claras y una interfaz intuitiva.
- Rendimiento. El sistema debe ser rápido y receptivo, con un mínimo de retraso o tiempo de inactividad.

Los requerimientos funcionales, fueron especificados a través de casos de uso, historias de usuario, así como algunos diagramas de flujo de datos relevantes. Es por ello que el Product Backlog, se fue definiendo en las reuniones de trabajo con los interesados. Por lo que, a partir de ahí se definieron las iteraciones o Sprints, en las que iría trabajando de forma incremental, hasta lograr que el sistema fuera construido en su totalidad.

Cada Iteración, creaba su propio Sprint Backlog, el cual contenía las diversas funcionalidades a construir con su responsable, y su tiempo estimado. Adicionalmente, se acordó que el tiempo máximo de un Sprint fuese de cuatro semanas y que, dentro de cada Sprint, el líder de proyecto convocara a una reunión diaria, llamada Scrum Daily Meeting para conocer tareas realizadas, tareas por hacer y problemas que hayan surgido, esto con la finalidad de tener retroalimentación del equipo de trabajo. Al final de cada Sprint, se realizaba un Sprint Review para evaluar los componentes construidos y comentar la planeación del próximo Sprint.

Google Cloud como proveedor del servicio en la nube, proporcionó toda la configuración necesaria para el desarrollo de este sistema, como fue el sistema operativo, la seguridad de datos, las capacidades de alojamiento y de la red.

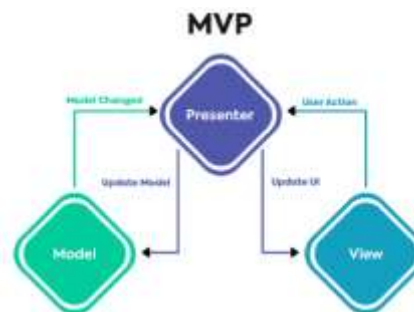
Con esta plataforma, los programadores y el líder de proyecto lograron adoptar una forma de trabajo iterativa, donde los requerimientos fueron concentrados según la etapa de desarrollo, con esto, los componentes del SGI, fueron entregados como un prototipo avanzado para la aprobación del cliente [12].

Como resultado, la plataforma de Google Cloud afianza el desarrollo de sistemas de manera ágil, a fin de mejorar la comunicación y atención dada al cliente, y con ello reducir el tiempo de entrega del Sistema.

A manera de conclusión, se aprecia que la plataforma de Google Cloud proporciona un ambiente de colaboración, cuando se combina el uso de metodologías ágiles y el cómputo en la nube para proyectos como este.

La arquitectura utilizada para el presente desarrollo fue el Modelo Vista Presentador (MVP), la cual se muestra en la figura 2:

Figura 2. Modelo Vista Presentador del SGI.



Fuente. MVP, 2025.

En este tipo de modelo, hubo que identificar y separar la funcionalidad entre el modelo, que representa la capa de datos y lógica del negocio de la aplicación, además de gestionar el acceso a la base de datos y servicios web. [14].

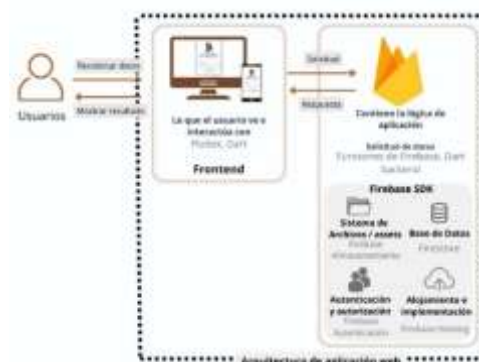
Por el lado de la vista, se tiene la capa de presentación y se encarga de mostrar la interfaz de usuario, la cual recibe información del presentador y actualiza la interfaz del usuario, según sea necesario.

Finalmente, la capa de presentador actúa como intermediario entre el modelo y la vista, en donde recupera datos del modelo y los presenta a la vista de manera adecuada.

Este modelo ha demostrado su efectividad en el control del comportamiento de la interfaz gráfica. A manera de conclusión, la arquitectura MVP que utiliza Flutter, es definitivamente la mejor tecnología que existe, ya que es de código abierto, ahorra tiempo, es rentable, y se integra perfectamente con Firebase.

Con base en lo explicado, la arquitectura que se utilizó para el desarrollo de la aplicación entre Flutter y Firebase se muestra en la Figura 3.

Figura 3. Arquitectura de la aplicación web.



Fuente. Elaboración propia.

El SGI propuesto seguirá una arquitectura cliente-servidor, en la que se podrá acceder al sistema a través de una interfaz basada en web (cliente) y estará alojado en un servidor. El enfoque basado en la web garantiza una amplia accesibilidad y compatibilidad con varios dispositivos, lo que permite a los estudiantes, profesores y personal acceder al sistema desde sus computadoras portátiles, tabletas o teléfonos inteligentes.

Con base en lo anterior, el lado del servidor es responsable de procesar solicitudes, administrar datos y garantizar el acceso seguro, mientras que el lado del cliente proporciona una interfaz para que los usuarios interactúen con el sistema. Finalmente, el entorno de desarrollo de este proyecto incluye 6 computadoras personales con al menos 8 GB de Memoria RAM, un procesador de 64 Core i5 con disco sólido de 1TB.

Los componentes de software incluyen Flutter SDK, Android Studio, Visual Studio Code. El entorno de desarrollo también está configurado para admitir la integración y la implementación continua utilizando herramientas como GitHub y Jenkins.

La autenticación de usuarios se implementó mediante JSON Web Tokens (JWT). Cuando los usuarios inician sesión, el sistema verifica sus credenciales y emite un token único. Luego, este token se utiliza para llamadas API posteriores para autenticar y autorizar acciones del usuario. Al adoptar JWT, el sistema mejora la seguridad al evitar el acceso no autorizado a recursos confidenciales.

En la figura 4, se puede apreciar la interfaz de inicio de sesión para los usuarios del SGI, para este caso, la figura que se muestra es el de acceso a un usuario con el rol de docente.

Figura 4. Inicio de sesión.



Fuente. Elaboración propia.

En la figura 5, se muestra una de las vistas del docente, en donde puede revisar las actas de calificaciones asentadas en semestres pasados. Este tipo de consultas, se puede realizar de una manera mucho más eficiente y sencilla, dado que este proceso, anteriormente se hacía de forma manual y en ocasiones, se tenía que solicitar al departamento de servicios escolares por escrito, lo cual consumía demasiado tiempo este proceso.

Figura 5. Acta de calificaciones.



Fuente. Elaboración propia.

En la figura 6, se muestra una de las vistas que tiene el departamento del Recursos Humanos (RH), así como el propio docente, en donde puede revisar sus horarios laborales tanto vigentes como de semestres anteriores.

Cabe señalar que este proceso, ayuda en gran medida a eficientar la validación de horarios entre el área de RH con cada uno de los jefes académicos, haciendo más rápido este proceso, además, con este sistema, se reduce el consumo de papel entre ambos departamentos, dado que solo se imprimen los juegos que sean necesarios para su conservación de manera física, evitando así, las impresiones innecesarias.

Figura 6. Horario del docente.



Fuente. Elaboración propia.

En la figura 7, se muestra una de las vistas que tiene el departamento del Desarrollo Académico (DA), así como el propio docente, en donde puede revisar las evaluaciones que han realizado sus estudiantes en el periodo vigente y anteriores, destacando que toda esta información, se genera de manera automática al cerrar el periodo correspondiente a dicha evaluación. Esto ayuda a que el departamento de DA junto con el jefe académico, puedan programar cursos de capacitación para la planta docente con base en los resultados y necesidades que hayan surgido de dicha evaluación.

Figura 7. Evaluación docente.



Fuente. Elaboración propia.

El SGI, utilizó el framework Flutter y el gestor de base de datos no relacional Firebase, durante este trabajo, se tuvieron algunos problemas como el diseño de la estructura de la aplicación, así como el esquema de base de datos adecuados. Sin embargo, se lograron superar con un diseño básico y preparado para iterar cuantas veces fuese necesario. Además, se utilizaron las capacidades de Firebase para cambiar el esquema sin interrupciones significativas.

Otro desafío importante fue garantizar la sincronización de datos en tiempo real entre múltiples usuarios y dispositivos. Lo anterior, también, significó un reto, dado que se tuvieron que manejar posibles inconsistencias y conflictos de datos, los cuales se resolvieron mediante las marcas de tiempo y transacciones de Firestore.

Además de manejar la autenticación y la seguridad del usuario para proteger los datos confidenciales y garantizar que sólo los usuarios autorizados pudieran acceder al sistema.

Aunado a lo anterior, se utilizó Firebase Authentication para el inicio de sesión de los usuarios y se implementaron mecanismos de control de acceso, utilizando reglas de seguridad de Firebase para definir y hacer cumplir restricciones de acceso basadas en roles y permisos de los usuarios.

El soporte sin conexión fue otro aspecto esencial que se tuvo que resolver, ya que los usuarios necesitaban acceder y modificar datos mientras estaban sin conexión, y sincronizar los cambios una vez que se reconectaban, para ello, se utilizaron las capacidades sin conexión de Firebase, para almacenar datos localmente y sincronizar los cambios automáticamente cuando el dispositivo volvía a estar en línea.

Las pruebas y la depuración fueron fundamentales para garantizar un sistema de gestión integral con las mínimas fallas. Para ello, se ejecutaron pruebas unitarias, de integración y de sistema en cada etapa de desarrollo. Con apego a lo que señala Scrum, el trabajo se dividió en tareas manejables y se creó un cronograma de hitos. Las funciones esenciales se priorizaron primero y se implementaron funciones adicionales cuando el tiempo lo permitió.

DISCUSIÓN Y ANÁLISIS DE RESULTADOS

A continuación, se detallan los resultados obtenidos tras la implementación del Sistema de Gestión Integral para el Instituto Tecnológico de Zacatepec, recordando que el objetivo principal de este proyecto, es optimizar los procesos académicos de diversas áreas, buscando reducir los tiempos de respuesta para diversos trámites, así como algunos costos relacionados con la papelería e insumos necesarios para la generación de documentación impresa.

Resultados cuantitativos:

- Reducción de gasto en papelería: Se observó una disminución del 15% en las compras de cajas de papel bond, durante el primer semestre del año 2024 debido al uso del sistema, en comparación con el mismo período del año anterior. Esto se atribuye a que el SGI, genera los documentos en digital, lo que facilita su lectura y uso.
- Mejora en la precisión de la emisión de documentos digitales: La precisión de la documentación digital aumentó en un 20%, evitando errores en los procesos académicos y en los mismos documentos, lo que se traduce en una reducción de errores en formatos u oficios emitidos.
- Disminución del tiempo del tiempo para procesos académicos: El tiempo promedio de elaboración de documentos oficiales, se redujo en un 20%, gracias a la automatización de formatos académicos.

Resultados cualitativos:

- Mayor visibilidad de los procesos académicos: El sistema proporciona una visión en tiempo real del estado que guardan los trámites administrativos que están relacionados con documentación de estudiantes y docentes.
- Mejora en la colaboración entre departamentos: El SGI, facilita la comunicación y el intercambio de información entre los diferentes departamentos involucrados en la gestión del proceso académico de los estudiantes y docentes, lo que se traduce en una mayor eficiencia y coordinación entre áreas.
- Mayor satisfacción de la comunidad estudiantil: La mejora en los tiempos de respuesta y la precisión en los documentos para trámites de los estudiantes, han contribuido a aumentar la satisfacción de los mismos.

Los resultados obtenidos demuestran que el Sistema de Gestión Integral, ha cumplido con los objetivos planteados, destacando el aumento en la eficiencia de los procesos académicos en donde están involucrados, estudiantes y docentes. Así también, se logró la reducción de costos en papelería, la mejora en la precisión de emisión de documentos y la disminución del tiempo de procesamiento de información, los cuales son indicadores claros del impacto positivo del sistema para la comunidad tecnológica en general.

Así también, con este sistema, se logró una mayor visibilidad de los procesos académicos, la mejora en la colaboración entre departamentos y el aumento de la satisfacción de los estudiantes, lo que resalta el valor del Sistema de Gestión Integral como herramienta de gestión.

Discusión

Como lo menciona [1], las nuevas herramientas que proporciona Google Cloud han abierto nuevos paradigmas de trabajo en la nube, y por consecuencia nuevas oportunidades de crecimiento para diversas instituciones, como es el caso del Instituto Tecnológico de Zacatepec.

Estas nuevas tecnologías, traen consigo, una innovación en la manera de desarrollar aplicaciones, que, para el caso del presente trabajo, es híbrida, es decir, funciona para web como para dispositivos móviles, ya sea Android o iOS. Así mismo, [6], indican que, para los desarrolladores, el cómputo en la nube otorga una gama de servicios e infraestructura a precios accesibles. Las IES tienen hoy en día la oportunidad de reducir e incluso eliminar gastos innecesarios en infraestructura, lo cual se vería reflejado en ahorros económicos.

En concordancia con [3], quienes desarrollaron la investigación *Acelerando el desarrollo multiplataforma con Flutter*, y con [15] *Implementación del framework Flutter para el desarrollo de una aplicación de comercio electrónico*. Respectivamente mencionan que este framework ha evolucionado rápidamente, por lo que se ha convertido en el método predilecto para desarrollar software multiplataforma. En la presente investigación, se está de acuerdo con lo que mencionan los autores, dado que el desarrollo del proyecto de software, fue bastante rápido por la misma tecnología y por la metodología de desarrollo de software utilizada, además, este framework permite que las aplicaciones sean compatibles con los sistemas operativos macOS, Linux y Windows.

Aunado a lo anterior, los autores de la investigación [11] *Diseño y construcción de un sistema de gestión de proyectos finales basado en la web y dispositivos móviles para la Universidad de Tabanan (Simantab) utilizando el framework Flutter*, señalan que las especificaciones de requisitos del sistema, se deben modelar con el apoyo de diagramas de flujo de datos, los cuales servirán para la creación de las interfaces. Esto mismo, se realiza en esta investigación, mediante las historias de usuario (que se modelan en pequeños diagramas) de cada uno de los Sprint ejecutados durante el proyecto.

CONCLUSIONES

Para el Instituto Tecnológico de Zacatepec, el Sistema de Gestión Integral incorpora una comunicación fluida entre departamento, docentes, administrativos y estudiantes.

Todo esto a través de una seguridad sólida mediante autenticación y control de acceso de usuarios y una comunicación optimizada entre la comunidad académica.

Al implementar este sistema, se mejoró la eficiencia de sus procesos académicos, además de mejorar la comunicación y mantener la seguridad de los datos, beneficiando al personal y a los estudiantes.

El desarrollo de un sistema de gestión integral utilizando Flutter y Firebase, ha demostrado su potencial para revolucionar las operaciones en las IES de nuestro país, al proporcionar una interfaz fácil de usar, capacidades de automatización y una gestión eficiente de los procesos académicos, además, el sistema ofrece una solución valiosa para mejorar la productividad y la eficiencia general.

AGRADECIMIENTOS

Los autores agradecen al Instituto Tecnológico de Zacatepec, por las facilidades otorgadas para la elaboración del presente trabajo de investigación; Además, como mención especial, a la estudiante Alma Flores Ramírez de la carrera de Ingeniería en Sistemas Computacionales con número de control 22090633, quien contribuyó al desarrollo del Sistema de Gestión Integral.

BIBLIOGRAFÍA

- [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 8(4):456-462. <https://doi.org/10.46792/fuoyejt.v8i4.1126>.
- [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 Publishin. 2020 May; vol 1530(1):012-020. doi: 10.1088/1742-6596/1530/1/012020
- [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. Consultado el 20 de febrero del 2025. Disponible en: <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. Disponible en doi: 10.1007/s10639-021-10739-1
- [6] Vincenzo, M. Digital Transformation Challenges for Universities: Ensuring Information Consistency Across Digital Services. *Cataloging & Classification*

- Quarterly. 2019 January; vol. 56(7): 592–606. Disponible en: <https://doi.org/10.1080/01639374.2018.1504847>
- [7] Google Firebase. Firebase. Consultado 15 de febrero del 2025. Disponible en: <https://www.firebase.google.com>
- [8] Google Flutter. Flutter. Consultado el 23 de febrero del 2025. Disponible en: <https://www.flutter.dev>
- [9] Google Cloud. Consultado el 10 marzo del 2025. Disponible en: <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). 2017 june; vol. 4(6): 2078-2082. Disponible en: <https://www.irjet.net/archives/V4/i6/IRJET-V4I6401.pdf>
- [11] Krisna, D., Oka, G., Sandi, W. Rancang Bangun Sistem Informasi Manajemen Tugas Akhir Universitas Tabanan (Simantab) Berbasis Web Dan Mobile Menggunakan Framework Flutter. Journal Science Teknologi Sosial Humaniora. 2024 Juni; vol. 2(2): 114-122. Disponible en: <https://s.id/jurnalsutaso>
- [12] Michal, Hron. Why and how is Scrum being adapted in practice: A systematic review. Journal of Systems of Software. 2022 January; 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. EE.UU. Editorial Apress; 2017. Consultado el 20 de febrero del 2025. 271p. Disponible en: <http://Definitive-Guide-Firebase-Android-Platform/dp/1484229428>
- [14] MVP. Modelo Vista Presentador. Modificado 25 de abril 2024. Consultado 10 de enero del 2025. Disponible en: <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/tepiian.v5i4.3110>
- [16] Payne, R. Developing in Flutter. In: Beginning App Development with Flutter. Berkeley, CA. EE.UU. Editorial Apress. 2019; 309p. Disponible en: https://doi.org/10.1007/978-1-4842-5181-2_2.
- [17] Scrum. Scrum org. ¿Qué es Scrum?. Consultado el 20 marzo del 2025. Disponible en: <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. 2020 August; vol. 2 (8):1262-1266. Disponible en: https://www.irjmets.com/uploadedfiles/paper/volume2/issue_8_august_2020/3180/1628083124.pdf
- [19] ItDo. Consultado el 22 de marzo de 2025. Disponible en: <https://www.itdo.com/blog/flutter-el-nuevo-framework-de-google/>

Tabla de Roles de Contribución

Rol	Autor(es)
Conceptualización	Marco Aurelio Velázquez Velazco
Curación de datos	José Pedro Aragón Hernández
Metodología	Mario Humberto Tiburcio Zúñiga
Administración del proyecto	Luis José Muñiz Rascado
Software	Marco Aurelio Velázquez Velazco
Supervisión	Luis José Muñiz Rascado
Visualización	Mario Humberto Tiburcio Zúñiga
Redacción (Borrador original)	Eugenio César Velázquez Santana
Redacción Revisión y edición.	Eugenio César Velázquez Santana



Esta obra está bajo una licencia internacional Creative Commons Atribución 4.0.