

Desarrollo de Middleware para Interconexión de una Aplicación Móvil con un Sistema Heredado



Middleware Development for Interconnecting a Mobile App with a Legacy System

Aguirre, Carlos A; Anchundia, Carlos E

 **Carlos A Aguirre**
caguirremal11@gmail.com
Escuela Politécnica Nacional, Ecuador

 **Carlos E Anchundia**
carlos.anchundia@epn.edu.ec
Escuela Politécnica Nacional, Ecuador

Latin-American Journal of Computing
Escuela Politécnica Nacional, Ecuador
ISSN: 1390-9266
ISSN-e: 1390-9134
Periodicidad: Semestral
vol. 10, núm. 1, 2023
lajc@epn.edu.ec

Recepción: 30 Septiembre 2022
Aprobación: 26 Octubre 2022

URL: <http://portal.amelica.org/ameli/journal/602/6023721002/>

DOI: <https://doi.org/10.5281/zenodo.7503927>

Resumen: El proceso de modernizar un sistema heredado puede llegar a ser complejo debido a limitaciones presentadas frente a las nuevas tendencias y tecnologías. La arquitectura de un sistema, su código fuente, la gestión de datos y la aplicación de malas prácticas son aspectos que los ingenieros deben considerar al implementar actualizaciones sobre estos sistemas. Sistemas como el Control Gerencial/Web (CG/Web) de la empresa ecuatoriana Información Tecnológica del Ecuador S.A. (I.T. del Ecuador) ejemplifican estos retos. Se propuso el desarrollo de la aplicación móvil CGApp como solución a las necesidades de movilidad de los usuarios gerenciales. Sin embargo, su desarrollo fue condicionado por la arquitectura monolítica del sistema heredado, siendo necesario el desarrollo e implementación de un middleware como medio para la interacción entre el componente móvil y los elementos del sistema CG/Web. Por esto, fue necesario aplicar un proceso de reingeniería con base en el sistema heredado, desarrollar métodos para la traducción de datos, aplicar controles de seguridad y rediseñar las pantallas para adaptarlas a un entorno móvil. De esta forma se consiguió integrar la aplicación móvil con el sistema heredado, agregando valor al proyecto.

Palabras clave: *sistemas heredados, middleware, aplicación móvil, seguridad de datos, reingeniería, pruebas automatizadas.*

Abstract: The modernization of legacy systems is a complex process due to the limitations that they can present when facing of new trends and technologies. Systems such as the “Control Gerencial/Web” (CG/Web) software of the Ecuadorian company “Información Tecnológica del Ecuador S.A.” (“I.T. del Ecuador”) exemplify these challenges. The system architecture, source code, data management and possible bad practices applied are some aspects that engineers must consider when implementing updates of these systems. We develop the CGApp mobile application as a solution to the mobility needs of managerial users of the CG/Web system. However, the development was conditioned by the monolithic architecture of the legacy system, requiring the design and implementation of a middleware as a means of interaction between the mobile component and the elements of the CG/Web. Therefore, we need to perform a legacy system reengineering process, developing methods for data translation, applying security controls and redesigning screens to adapt them to a mobile environment. As a result, it was possible to integrate the mobile application with the legacy system, adding value to the project.

Keywords: *legacy systems, middleware, mobile app, Data security, reengineering, automated tests.*

I. INTRODUCCIÓN

La arquitectura de CG/Web fue diseñada para el uso de usuarios operativos que no requerían movilidad laboral. Sin embargo, las exigencias para usuarios de nivel gerencial, que forman parte de los procesos, requieren otro tipo de operatividad que el sistema no puede proveer adecuadamente. Como resultado, la limitación de esta arquitectura genera los siguientes inconvenientes: (i) restringe el ingreso al sistema desde el interior de las oficinas e impacta a la ejecución oportuna de las tareas de los clientes de alto nivel; (ii) obliga al departamento de TICs, la implementación de redes VPN u otras estrategias cuya seguridad no puede ser garantizada o controlada por el sistema; (iii) genera multas y recargos administrativos derivados del incumplimiento de actividades [5]; y (iv) el abandono de los clientes que optan por sistemas más dinámicos y orientados a la movilidad ofertados por la competencia. La empresa I.T. del Ecuador tiene la urgencia de proveer una solución que pueda competir en un mercado cada vez más agresivo, incursionando tanto en estructuras tecnológicas modernas y en enfoques orientados al valor y experiencia de sus usuarios. Los requerimientos básicos que la nueva solución debe incorporar son:

Las tecnologías y tendencias emergentes constantemente ponen a prueba la capacidad de mejora de servicios ofertados por las empresas. Esta situación es aprovechada por aquellas entidades con visión innovadora y que, como resultado, buscan una ventaja competitiva que las destaque de sus competidores. Esta tarea se torna compleja cuando las adaptaciones o mejoras involucran la adaptación de nuevas tecnologías o arquitecturas a sistemas heredados [1]. Durante este tipo de adaptación, es mandatorio analizar la arquitectura, funcionalidad y proceso de mantenimiento de los sistemas para determinar las estrategias y riesgos de una modernización [2]. Esta modernización de sistemas heredados puede incluir actividades como migración de infraestructura, integración con nuevas plataformas, refactorización de código e implementación de interconexiones a otros sistemas que complementen funcionalidades y necesidades especializadas.

Un caso real del contexto antes descrito se evidencia en la empresa ecuatoriana Información Tecnológica del Ecuador S.A. (I.T. del Ecuador), lo cual enfoca al desarrollo y distribución de software y hardware [3]. Esta empresa cuenta con un producto ERP denominado Control Gerencial/Web (CG/Web); destinado al control de procesos operativos y administrativos de empresas públicas y privadas a nivel nacional [4]. CG/Web se oferta como un producto instalable para que, por motivos de seguridad, su acceso se encuentre circunscrito exclusivamente dentro de las intranets de cada cliente. Sin embargo, la empresa tiene la necesidad de incorporar nuevas tecnologías a este sistema.

La arquitectura de CG/Web fue diseñada para el uso de usuarios operativos que no requerían movilidad laboral. Sin embargo, las exigencias para usuarios de nivel gerencial, que forman parte de los procesos, requieren otro tipo de operatividad que el sistema no puede proveer adecuadamente. Como resultado, la limitación de esta arquitectura genera los siguientes inconvenientes: (i) restringe el ingreso al sistema desde el interior de las oficinas e impacta a la ejecución oportuna de las tareas de los clientes de alto nivel; (ii) obliga al departamento de TICs, la implementación de redes VPN u otras estrategias cuya seguridad no puede ser garantizada o controlada por el sistema; (iii) genera multas y recargos administrativos derivados del incumplimiento de actividades [5]; y (iv) el abandono de los clientes que optan por sistemas más dinámicos y orientados a la movilidad ofertados por la competencia. La empresa I.T. del Ecuador tiene la urgencia de proveer una solución que pueda competir en un mercado cada vez más agresivo, incursionando tanto

en estructuras tecnológicas modernas y en enfoques orientados al valor y experiencia de sus usuarios. Los requerimientos básicos que la nueva solución debe incorporar son:

- Incrementar la operatividad de los usuarios gerenciales en relación con las actividades de aprobación a través del uso de dispositivos móviles.
- Facilitar el proceso de aprobación tomando en cuenta el uso de pantallas pequeñas y el manejo orientado al uso de aplicaciones táctiles.
- Garantizar la seguridad y disponibilidad de los datos sensibles del cliente sin requerir la intervención de sus departamentos de tecnologías
- Interoperar la nueva solución con el sistema heredado que actualmente se encuentra en funcionamiento

Para el desarrollo de la solución se siguió el ciclo de vida de desarrollo de software bajo una perspectiva ágil, enfocando el análisis hacia la actualización de arquitectura y actividades de reingeniería. El proceso se describe en las secciones posteriores de la siguiente forma. En la Sección II, se describe el desarrollo de la aplicación móvil, el proceso de análisis de requisitos que involucra la especificación por ejemplos y automatización de pruebas basada en los ejemplos, cuestiones de diseño de interfaz de usuario de la aplicación y en análisis y actualización de la arquitectura del sistema heredado. En la Sección III, se detallan los resultados tanto de la arquitectura, factores de seguridad que fueron considerados y los elementos de usabilidad relevantes para crear una aplicación sencilla y no propensa a errores de uso. En la Sección IV, se presenta una discusión sobre las actividades de actualización de las arquitecturas como punto clave para la mejora de la competitividad. Finalmente, en la Sección V se presentan las conclusiones sobre los requerimientos de esta solución.

II. METODOLOGÍA

Para la construcción de la solución, se consideraron cuatro aspectos de calidad relevantes para asegurar el máximo valor del producto: (a) el desarrollo de la aplicación móvil; (b) la seguridad en el flujo de datos; (c) la mejora de la usabilidad del sistema; y (d) el middleware de interconexión con el sistema heredado. A continuación, se detallan los procesos metodológicos que se consideraron en cada aspecto.

A. Desarrollo de aplicación móvil

Para el desarrollo de la aplicación móvil, se optó por el empleo de un proceso ágil (desarrollo guiado por el comportamiento - BDD [11]) con la finalidad de ayudar al entendimiento de la problemática y descubrimiento de alternativas de solución. El entorno de trabajo fue condicionado por el sistema heredado, siendo mandatorio el uso del lenguaje C# para permitir mantener la misma tecnología en el sistema heredado y en la aplicación móvil. Finalmente, se prefirió incorporar pruebas automatizadas con criterios de aceptación y pruebas de unidad.

En el apartado metodológico de desarrollo se empleó un proceso denominado MOBILE-D [6]. Esta metodología se enfoca en el análisis de desarrollos orientados a aplicaciones móviles para grupos de trabajo pequeños. Su estructura por fases permitió dirigir el desarrollo de funcionalidades basado en pruebas dentro de ciclos iterativos. MOBILE-D hace hincapié en la validación del producto con la incorporación de una fase de integración, lo cual posibilita la identificación de potenciales problemas antes del despliegue final.

Durante el proceso de análisis de requerimientos y diseño se aplicó la práctica de especificación por ejemplos (SBE) [7]. Como resultado, se especificaron 14 historias de usuario [8] orientadas a los tipos usuarios de nivel gerencial. Esta práctica ayudó en a las actividades de verificación de software a través de la ejemplificación y discusión de escenarios reales. En promedio se manejaron tres escenarios y por cada uno

de ellos, se manejaron entre dos y diez ejemplos para asegurar la mayor cobertura de casos de ejecución. Acompañando a la gestión de las historias, se utilizó la plataforma Jira [9] para el seguimiento del desarrollo de las historias de usuario.

La automatización de pruebas es fundamental dentro de los procesos ágiles. Al utilizar SBE fue posible emplear prácticas como Desarrollo Guiado por Ejemplos (EDD) [10]; sin embargo, al no contar con un marco de trabajo en el entorno de desarrollo se optó por emular el proceso con métodos disponibles con la implementación `UnitTesting` como `ClassInitialize`, `TestCleanup` [12] y el uso de archivos CSV como contenedor de los ejemplos a ejecutar. Con estos tres elementos, se logró implementar un proceso automatizado del control de calidad del producto final que seguía el proceso especificado en la Fig. 1.

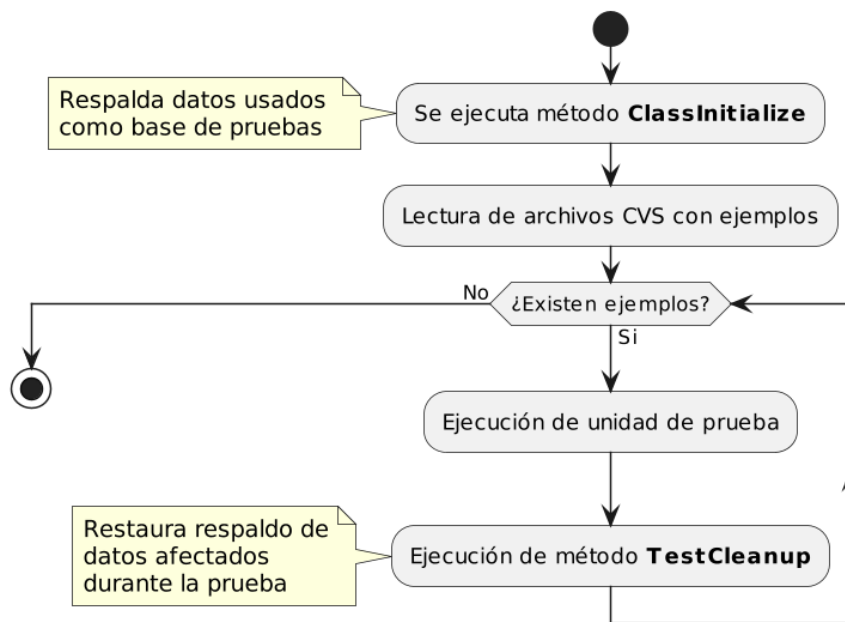


FIG. 1 .

Proceso de emulación de prueba automatizada en el entorno de desarrollo

B. Seguridad en el flujo de datos

La seguridad durante el flujo de datos es vital en las aplicaciones que utilizan la red pública como medio de transmisión. Como base para la implementación de controles de seguridad se utilizó el estándar de evaluación *Mobile Application Security Verification Standard* (MASVS) versión 1.3 [13], el cual considera un total de 8 grupos de requerimientos de seguridad. Este incorpora un check list que sirvió como guía durante el desarrollo del proyecto para la implementación de funciones de seguridad de la solución. No obstante, debido a las decisiones gerenciales en la realización del proyecto, se omitió el grupo de requerimientos de seguridad 8, llamado “Resistencia ante la ingeniería inversa”.

C. Usabilidad

El sistema heredado fue construido con la herramienta DevExpress versión 11.1.12 [14], cuyo diseño de interfaces se limita al despliegue de tablas estáticas. Esta estructura se ha mantenido en el sistema heredado desde el año 2012 pese a su actualización a la versión 19.2 del 2020 [15] (Fig. 2). Esto ha causado que el sistema no se alinee a buenas prácticas para diseños de interfaces que aconsejan un comportamiento dinámico e

interactivo, con elementos limpios, navegabilidad vertical y alineada a “gestos” estandarizados en los entornos móviles

Modalidad: Orden de Compra

Compañía	Nro. Solicitud	F. Solicitud	Solicitante
AGUA POTABLE	55	2020/05/22	CGWEB ADMINISTRADOR
AGUA POTABLE	58	2020/05/22	CGWEB ADMINISTRADOR
AGUA POTABLE	59	2020/05/22	CGWEB ADMINISTRADOR
AGUA POTABLE	124	2020/10/07	CGWEB ADMINISTRADOR
AGUA POTABLE	125	2020/10/08	CGWEB ADMINISTRADOR

FIG. 2 .

Ejemplo de pantalla sistema heredado

Consecuentemente, se estructuró el diseño de pantallas para que permita el dinamismo y la minimización de errores a través de la interacción de un solo botón. La aplicación móvil incorporó controles de la herramienta *DevExpress Mobile UI XamarinForms* [16]. Cada pantalla (Fig. 3) se compone de un control *Layout* que da el formato general de la aplicación, un *ScrollView* que permite desplazar la interfaz en dirección vertical y un *Grid* que sirve para contener y ubicar los controles de la pantalla. Dependiendo de la función a realizar en la pantalla, se tiene dos controles generales. En pantallas que despliegan los datos de detalle de entidades del sistema, como por ejemplo solicitudes, se utilizó la propiedad *Grid.Group* para ordenar los datos en grupos. A su vez, en las pantallas que despliegan listas, como en los buzones de los usuarios, se utilizó los controles *CollectionView*, para implementar el control *SwipeContainer*. Este control tiene la propiedad *StartSwipeItem* que permitió aplicar controles a las acciones de deslizamiento al acceder a los datos de detalle de las entidades del sistema.

D. Middleware CG/WebApp

Con el fin de que la aplicación móvil se conecte con los componentes del sistema heredado fue necesario implementar un medio para la conversión entre las interfaces de datos del entorno web a las entidades del entorno móvil y viceversa. Este *middleware* integró la aplicación CGApp con el sistema CG/Web, por lo que fue nombrado CG/WebApp.

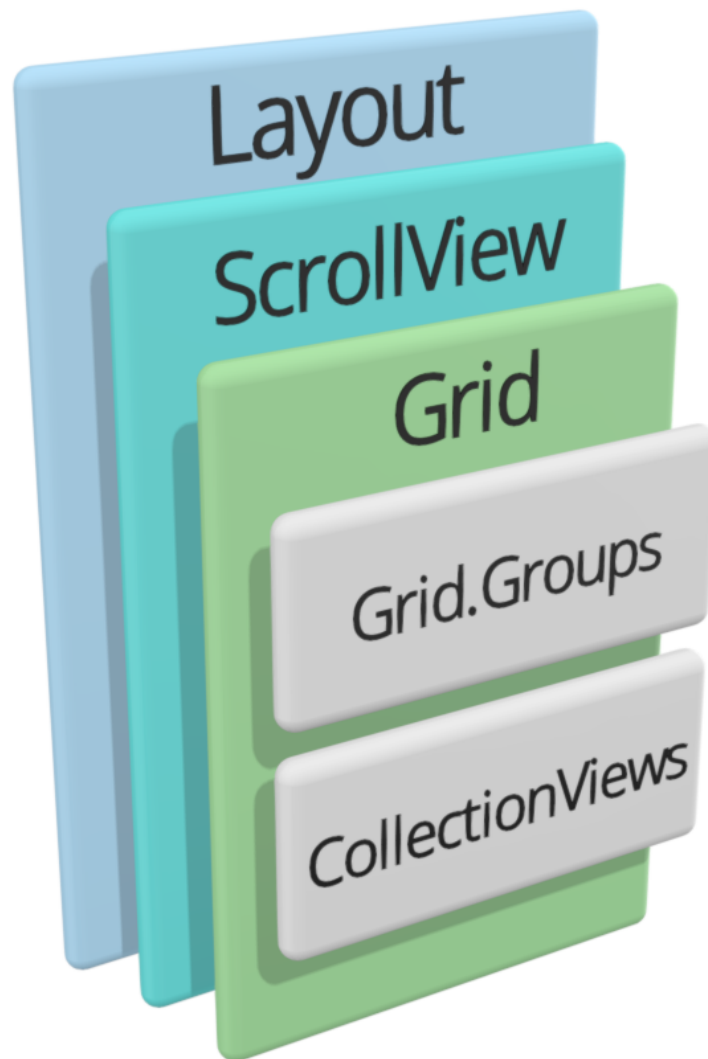


FIG. 3.
Estructura de la pantalla de la aplicación móvil

Dado que la arquitectura cliente/servidor del sistema heredado no podía ser modificada, se realizó un análisis de esta para determinar los componentes que pudieran servir como puntos de comunicación para un *middleware*. Se determinó que el sistema cuenta con una estructura interna, relacionada con las funcionalidades de los usuarios gerenciales, está conformada por cinco capas, destacándose: Servicios y Lógica de Negocio del sistema web. Igualmente, se identificó que el sistema cuenta con reglas del negocio que, en lugar de implementarse en la capa de lógica, se encontraban embebidas directamente en la capa de Presentación (pantallas del sistema web). Este problema imposibilitó reutilizar estos métodos y controles. De igual forma la gestión de los datos dentro del sistema heredado se realiza por conjuntos de datos, las cuáles son entidades incompatibles con el entorno móvil.

En cuanto a la estructura de la información, se encontró que la forma de gestión de los datos no era propicia para el uso de dispositivos móviles. El sistema legado maneja *DataSets* que se caracterizan por ser demasiado pesados para la transmisión, por lo que se requiere un proceso de conversión para reducir el tamaño de los datos. También se pudo identificar que algunos datos podrían limitar la información que se puede obtener de las consultas. Por ejemplo, el sistema manejaba fechas utilizando tipos de datos String, lo que impide acceder a prestaciones y formatos del tipo Date. Finalmente, se identificaron casos en los que los datos eran enviados

con identificadores de nivel de base datos, y se debió realizar una conversión entendible para el usuario a nivel de las pantallas del sitio web.

La solución requirió realizar un proceso de reingeniería para reestructurar adecuadamente las responsabilidades del sistema y poder realizar una correcta reutilización de código y reglas de negocio. Como resultado, se distribuyó y refactorizó en código de la capa Presentación en las capa de negocio tanto de la aplicación como del *middleware*.

Para el manejo de datos se implementaron los siguientes métodos. Los constructores servían como traductores de la información transmitida de *DataSet* a objetos ligeros. Métodos de conversión que validan y usan estructuras de datos optimizadas. Por ejemplo, para utilizar las fechas String en datos tipo *Date* se validó el formato y se inicializó objetos en la clase respectiva. Métodos de mapeo de los datos de códigos con los catálogos obtenidos de los componentes heredados, consiguiendo transmitir la información de la descripción de estos datos, lo que posibilitó reducir la cantidad de entidades utilizadas en la aplicación móvil en comparación con sus equivalentes del sistema heredado. Por último, métodos de conversión de clases a *DataSets*, para transmitir la información de la aplicación móvil a los métodos heredados y que puedan afectar a la base con la actualización de datos.

III. RESULTADOS

A. Arquitectura de la solución

Una vez identificadas las partes relevantes que componen la solución, se obtuvo la arquitectura de la solución. Esta arquitectura se compone de los elementos heredados del sistema CG/Web, la aplicación móvil CGApp, el *middleware* de interconexión entre ambos AppCG/Web y una aplicación de terceros para la gestión de notificaciones, como se aprecia en la Fig. 4.

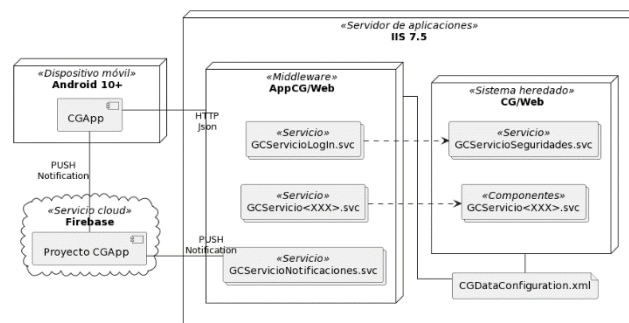


FIG. 4 .
Arquitectura de la solución: *middleware* y sistema heredado

En esta nueva arquitectura, los componentes del *middleware* interactúan con los elementos heredados del sistema web con el objetivo de acceder a los datos necesarios para las actividades de la aplicación móvil utilizando los conjuntos de datos personalizados del sistema CG/Web. La aplicación interactúa con esta sección de la solución, mediante clases, para acceder a los métodos mediante los servicios WCF. La implementación de terceros *Firebase* gestiona el envío de Notificaciones PUSH conectándose con el *middleware* y la aplicación CGApp. Estas toman un registro de la existencia de acciones pendientes del usuario, por ejemplo, solicitudes que no se han actualizado, y envía una notificación al servicio en la nube del proyecto *Firebase*. Este genera y envía un mensaje tipo PUSH al equipo móvil registrado por el usuario que ha iniciado sesión en la aplicación.

B. Seguridad

A fin de verificar el nivel de seguridad con el que cuenta la solución, se la evaluó utilizando el *check list* proporcionado por el estándar MASVS [17]. Este documento ejemplifica los controles de seguridad que deben cumplirse para que la aplicación móvil apruebe el estándar. Para esto se evalúan los ocho grupos de requerimientos de seguridad, permitiendo identificar el porcentaje de cumplimiento con el que cuenta cada grupo. Como se indicó en la Sección II.B, en el presente proyecto se omitió el grupo 8, y se obtuvieron los porcentajes de cumplimiento que se muestran en la Fig. 5.

Con el propósito de cumplir estos requerimientos, se implementaron controles de seguridad en base a los que se estipulan en el estándar MASVS. Estos controles fueron adaptados a la realidad de CGApp y se implementaron en la aplicación móvil, los servicios WCF y en el dominio utilizado para el acceso por la red. Sin embargo, debido a las limitaciones de compatibilidad con el estándar MASVS presentadas en el sistema CG/Web no fue posible aplicar la totalidad de controles de seguridad. Por esto motivo, algunos de los grupos de requerimientos de seguridad no pudieron ser cumplidos en su totalidad.

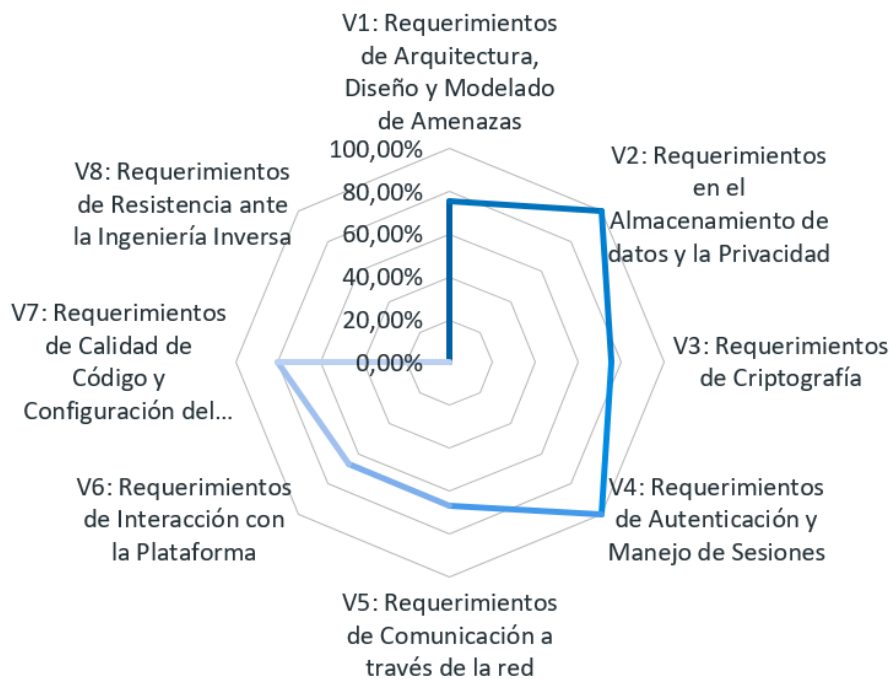


FIG. 5.

Porcentajes de cumplimiento de grupos de requerimientos de seguridad

Como resultado de esta evaluación, se obtuvo una valoración de 4 sobre 5, lo que indica el cumplimiento del estándar de seguridad, y asegura que la solución si cuenta como una aplicación móvil segura (Fig. 6).

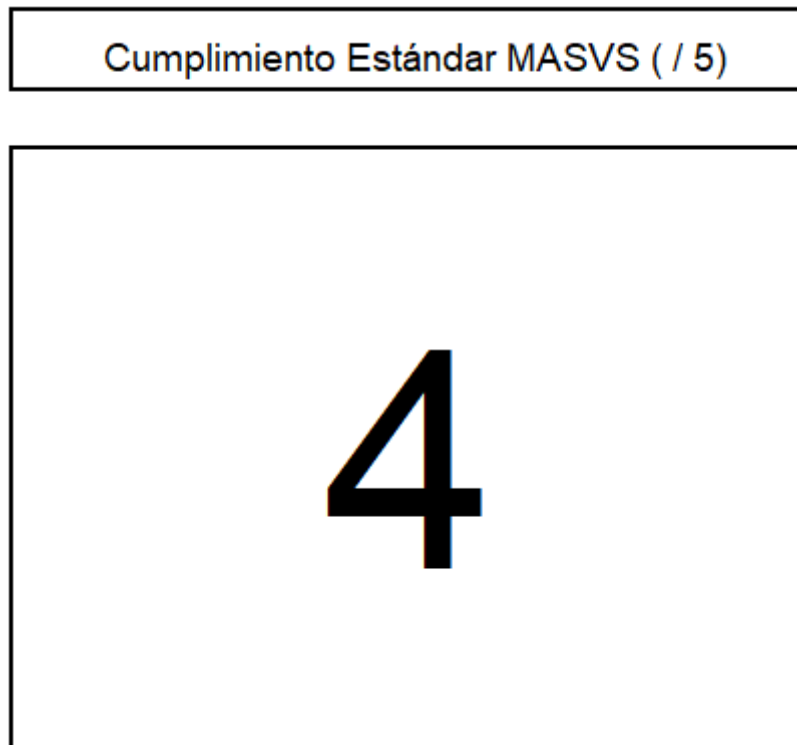


FIG. 6.

Extracto de resultado de evaluación MASVS del check list en Excel

C. Pruebas de funcionalidad

En el desarrollo de esta solución, se siguió de forma rigurosa la metodología Mobile-D y el conjunto de prácticas para un desarrollo guiado por el comportamiento. Con esto no solo se mantuvo un orden durante el desarrollo de la solución, sino que además se realizó un análisis de requisitos enfocados en el valor del negocio y reducción de fallas de sistema. Como resultado se generaron diversos artefactos como historias de usuario, archivos de características y criterios de aceptación (features), ejemplos en formato Gherkin, y UnitTests.

Con base en estos artefactos se pudo determinar los resultados esperados para el diseño de un conjunto de pruebas unitarias automatizadas para verificar el cumplimiento de las funcionalidades esperadas en la aplicación, como se mencionó en la sección II.A. El gestor de pruebas de C# proporcionado por el IDE Visual Studio permite identificar el resultado obtenido de cada prueba en un proyecto ejecutado, como se evidencia en la Fig. 7. Para esta solución se especificaron 13 pruebas unitarias que fueron ejecutadas y superadas. De esta forma, se aseguró no solo el funcionamiento de CGApp sino también los casos recopilados con el dueño de la solución y que corresponden a las características de valor solicitadas requeridas por el cliente.

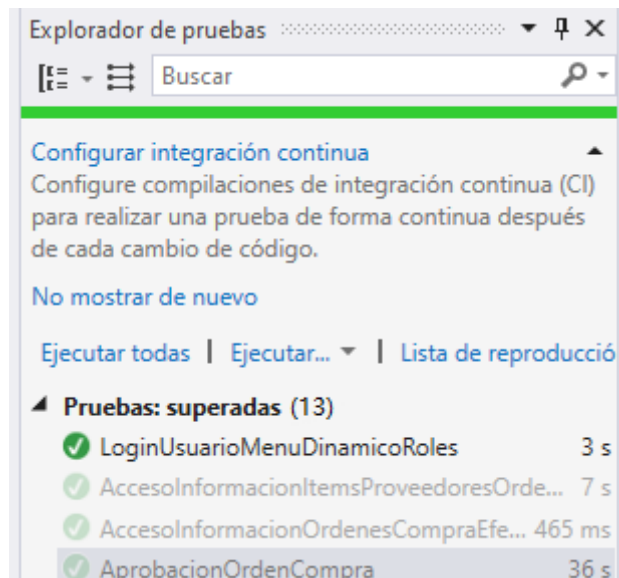


FIG. 7 .
Pruebas superadas identificadas en gestor de pruebas

D. Encuestas de usabilidad de aplicación móvil

Tras un análisis de la estructura de las pantallas del sistema heredado y discernir la información relevante para cada actividad realizada en CGApp, se obtuvo un diseño para las interfaces de usuario de la aplicación móvil basado en las reglas heurísticas de Nielsen [18]. Con el fin de asegurar que las mismas solventen las necesidades de los usuarios, se elaboró una encuesta en base a las adaptaciones aplicadas para la compatibilidad con el entorno móvil. Estas se aplicaron a cinco expertos de los módulos del sistema CG/Web incluidos en la aplicación móvil, quienes han tenido mayor contacto con las actividades de los usuarios finales, por ende, consideramos como informantes calificados y relevantes para el desarrollo de las interfaces del proyecto. Se utilizó la herramienta Formularios Google para realizar la encuesta usando un medio digital y obtener la tabulación de los datos de forma eficiente, como se ve en la Fig. 8.



FIG. 8 .
Ejemplo de encuesta sobre diseño de CGApp

Estos resultados permitieron corregir el diseño preliminar y enfocarlo para adaptarse a las preferencias de los usuarios. Esto se evidencia sobre todo en las pantallas de Buzón y Aprobación, de las cuáles se muestra un ejemplo en la Fig. 9. De esta forma se buscó facilitar la adaptación y el cumplimiento de las actividades a los usuarios.

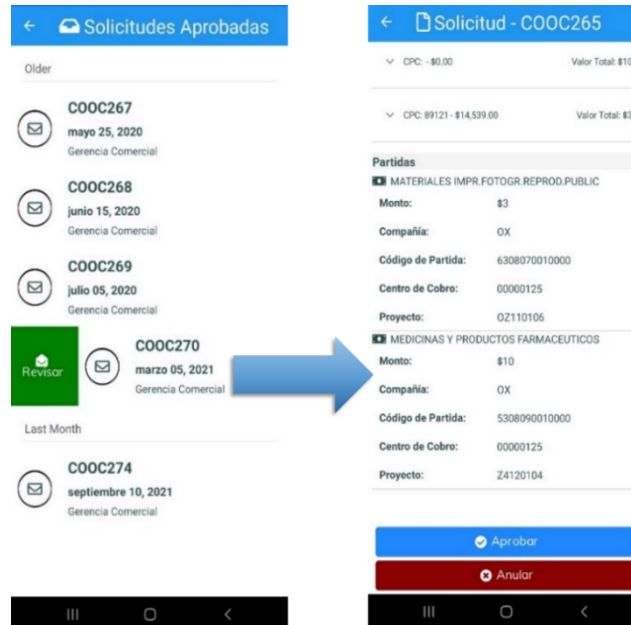


FIG. 9.

De esta forma se buscó facilitar la adaptación y el cumplimiento de las actividades a los usuarios.

IV. DISCUSIÓN

Cuando se requiere que un sistema sea mejorado o actualizado en las fases de mantenimiento, la arquitectura define cuan complejo será el proceso. Una arquitectura monolítica puede no ser la más adecuada para la escalabilidad del software. En el caso presentado, por ejemplo, la arquitectura del sistema debe ser desplegada en cada cliente y sus mantenimientos deben ser analizados y aplicados individualmente, generando problemas de integración entre distintas versiones y configuraciones locales únicas. En este escenario, pudo ser posible la integración con tecnologías enfocadas a la nube; pero, se condiciona el funcionamiento a una gestión de arquitectura local, y se elevan costos y tiempo. El análisis de la estructura existente permitió encontrar problemas de responsabilidad entre las distintas capas, diseño que imposibilitó la reutilización de funciones y afectó las etapas de codificación de reglas del negocio, que inicialmente se encontraban embebidas en las pantallas web. También se detectó un mal uso de tipos de datos que no permite sacar provecho de estructuras especializadas y dificultan la interconexión entre los sistemas. Esto evidenció una falta de organización en el proceso de diseño del sistema y la generación de caos durante la codificación y procesos de adaptación del sistema heredado. La reingeniería de la estructura del sistema heredado facilitaría futuras actualizaciones y aumentaría la reputación de este frente a los clientes al contar con capas desacopladas y cohesionadas.

En este proyecto, se aplicaron distintas prácticas para la verificación y validación de software; como lo son las pruebas automatizadas de la funcionalidad en base a las experiencias de usuario y la comprobación de los controles de seguridad mediante el *checklist* de MASVS. Desde las etapas tempranas de ciclo de vida del desarrollo, se emplearon ejemplos e historias de usuario para mejorar el entendimiento del sistema y propiciar una adecuada comunicación con el cliente. El resultado de este análisis permitió descubrir elementos importantes que facilitaron las actividades de diseño y codificación. La especificación de ejemplos ejecutables permitió asegurar la calidad y cumplimiento de los escenarios de prueba. Con esto se logró contar

con evidencia técnica documental sobre el cumplimiento de los criterios de aceptación del sistema. Sin embargo, esto exhibió la falta de cultura en la aplicación de técnicas y herramientas de gestión del sistema heredado al no contar con algún registro histórico de documentación sobre las funcionalidades del sistema, su codificación ni su ejecución. Promover el uso de herramientas de gestión permitirá mantener un registro de las actividades relacionadas al desarrollo del sistema heredado y ahorrará recursos al facilitar las actualizaciones en el mismo.

Como parte de la aplicación del enfoque ágil, se aplicaron pruebas automatizadas en el desarrollo del proyecto que permitieron verificar y garantizar el cumplimiento de las funcionalidades esperadas en la aplicación móvil. Su utilización resultó ser provechosa para el control ante los cambios realizados en el desarrollo del *middleware* y en actualizaciones de los componentes heredados. A modo de ejemplo, durante el tiempo de desarrollo se detectó un cambio en uno de los elementos del sistema heredado que afectaba directamente a la ejecución de métodos del *middleware*, lo que permitió aplicar los ajustes respectivos para solventar este cambio. Pese a esto, se evidenció que no se aplicaron este tipo de pruebas en el desarrollo del sistema heredado, pasando por alto algunas modificaciones en diversas secciones del sistema y afectó a sus dependencias. Esto puede causar que no se controlen errores en la ejecución de funcionalidades que podrían ser detectados a tiempo, lo cual repercute en las actividades de los usuarios finales y a la reputación del sistema ante el cliente. Promover la configuración y aplicación de estas pruebas puede evitar inconvenientes en actualizaciones futuras y agregando valor técnico al sistema heredado.

V. CONCLUSIONES

El proceso aplicado en el proyecto permitió el desarrollo de una aplicación móvil segura que otorga un medio de acceso inmediato al sistema CG/Web. Esta aplicación posibilita a los usuarios gerenciales a cumplir sus operaciones dentro de los procesos empresariales mediante sus dispositivos móviles, de forma que su movilidad no interfiera con el normal desarrollo de las actividades del sistema. De esta forma, CGApp mejora el rendimiento del cumplimiento de las asignaciones de los usuarios al permitir acceso remoto al sistema. La reingeniería realizada a las interfaces de usuario del sistema heredado permitió compactar el proceso de aprobación del sistema web para adaptarlo al entorno móvil. Esto incluyó el rediseño de las interfaces para que sean accesibles en pantallas pequeñas de forma táctil, lo cual facilita el proceso de aprobación y revisión de información a los usuarios finales en sus dispositivos móviles.

El uso del estándar MASVS versión 1.3 para el desarrollo y evaluación de la seguridad en el proyecto permitió garantizar que los datos sensibles del cliente se mantengan seguros en su flujo dentro de la red pública. De esta forma, se pudo confirmar la disponibilidad de los datos en el entorno móvil sin necesitar la intervención de los departamentos de tecnologías del cliente. Por último, el desarrollo del *middleware* mediante el exhaustivo análisis de la funcionalidad del sistema heredado permitió la interconexión de este con la aplicación móvil del proyecto. Gracias a esto, fue posible la interacción de los usuarios finales con los datos disponibles, permitiendo mantener la integridad entre las acciones realizadas por los usuarios en la aplicación y el estado de la información en el sistema web. De este modo, se pudo agregar un nuevo valor al sistema heredado ante el usuario y permitió que el sistema CG/Web pueda modernizarse a futuro para continuar siendo relevante en el mercado.

REFERENCES

T. Brehm, «What Is A Legacy System?», 9 Enero 2021. [En línea]. Available: <https://entranceconsulting.com/what-is-legacy-system-and-legacy-software/>. [Último acceso: 10 Septiembre 2022].

- V. Sawant, «A brief guide to legacy system modernization,» Rackspace technology, 28 Diciembre 2020. [En línea]. Available: <https://www.rackspace.com/blog/brief-guide-legacy-system-modernization>. [Último acceso: 16 Septiembre 2022].
- V. Alarcón, «Elaboración del plan estratégico para la empresa Información tecnológica del Ecuador S.A.,» Mayo 2008. [En línea]. Available: <https://bibdigital.epn.edu.ec/bitstream/15000/1029/1/CD-1473%282008-05-26-02-20-54%29.pdf>.
- IT del Ecuador, «SOLUCIÓN,» Aggity, 20 Agosto 2020. [En línea]. Available: <http://itdelecuador.com/>. [Último acceso: 12 Septiembre 2022].
- M. Muzo, «Levantamiento de procesos postergados en sistema CG/Web de ETAPA,» Información Tecnológica del Ecuador, Quito, 2019.
- agile.vtt.fi, «Mobile-D patterns,» virtual.vtt.fi, 9 Septiembre 2005. [En línea]. Available: http://virtual.vtt.fi/virtual/agile/mobile-d_docs/. [Último acceso: 27 Noviembre 2020].
- C. McKenzie, «specification by example (SBE),» TechTarget, 10 Septiembre 2014. [En línea]. Available: <https://www.techtarget.com/searchsoftwarequality/definition/Specification-by-example-SBE>. [Último acceso: 16 Septiembre 2022].
- M. Rehkopf, «Historias de usuario con ejemplos y plantilla,» Atlassian, 20 Octubre 2018. [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/user-stories>. [Último acceso: 10 Septiembre 2022].
- Atlassian, «Jira Software,» Atlassian, 9 Enero 2020. [En línea]. Available: <https://www.atlassian.com/es/software/jira>. [Último acceso: 10 Septiembre 2022].
- T. Girba, «An example of example-driven development,» Feenk, 14 Febrero 2019. [En línea]. Available: <https://medium.com/feenk/an-example-of-example-driven-development-4dea0d995920>. [Último acceso: 21 Septiembre 2022].
- M. Myint, «Comparative Study of Test-Driven Development (TDD), Behavior-Driven Development (BDD) and Acceptance Test-Driven Development (ATDD),» International Journal of Trend in Scientific Research and Development (IJTSRD), vol. 3, n° 4, pp. 231-234, 2019
- G. Barré, «MSTest v2: Test lifecycle attributes,» MEZIANTOU'S BLOG, 02 Diciembre 2018. [En línea]. Available: <https://www.meziantou.net/mstest-v2-test-lifecycle-attributes.htm>. [Último acceso: 25 Agosto 2021].
- J. Willemsen, «OWASP owasp-mstg Releases,» 11 Agosto 2019. [En línea]. Available: <https://github.com/OWASP/owasp-mstg/releases/tag/1.1.3-excel>. [Último acceso: 27 Agosto 2021].
- DevExpress, «Current Version/Build,» DevExpress, 8 Septiembre 2022. [En línea]. Available: <https://www.devexpress.com/support/versions.xml>. [Último acceso: 19 Septiembre 2022].
- DevExpress, «When Only the Best Will Do v19.2,» DevExpress, 14 Junio 2020. [En línea]. Available: <https://www.devexpress.com/subscriptions/new-2019-2.xml>. [Último acceso: 19 Septiembre 2022].
- DevExpress, «Free Xamarin.Forms UI Controls,» DevExpress, 20 Abril 2020. [En línea]. Available: <https://docs.devexpress.com/MobileControls/400545/xamarin-forms/index>. [Último acceso: 19 Septiembre 2022].
- C. Holguera, B. Müller, S. Schleier y J. Willemsen, «OWASP Mobile Security Testing Guide,» OWASP, 13 Mayo 2021. [En línea]. Available: <https://owasp.org/www-project-mobile-security-testing-guide/>. [Último acceso: 27 Agosto 2021].
- E. Schroeter, «11 Usability Heuristics Every Designer Should Know,» careerfoundry, 6 Agosto 2021. [En línea]. Available: <https://careerfoundry.com/en/blog/ux-design/usability-heuristics/>. [Último acceso: 30 Octubre 2022].