

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**DESARROLLO DE UNA APLICACIÓN MÓVIL SEGURA. ESTUDIO DE
CASO: INFORMACIÓN TECNOLÓGICA DEL ECUADOR S.A.**

**TRABAJO DE TITULACION PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

Carlos Arturo Aguirre Maldonado

carlos.aguirre@epn.edu.ec

DIRECTOR: Ing. Carlos Eduardo Anchundia Valencia

carlos.anchundia@epn.edu.ec

CODIRECTOR: Pamela Catherine Flores Naranjo, Ph.D

pamela.flores@epn.edu.ec

Quito, enero 2022

DECLARACIÓN

Yo, Carlos Arturo Aguirre Maldonado, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



CARLOS ARTURO AGUIRRE MALDONADO

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Carlos Arturo Aguirre Maldonado bajo nuestra supervisión.



ING. CARLOS ANCHUNDIA

DIRECTOR DE PROYECTO



PHD. PAMELA FLORES

CODIRECTOR DE PROYECTO

DEDICATORIA

Dedico este trabajo a mis padres Carlos Arturo Aguirre Maldonado y Gladys Alicia Maldonado Silva, quienes han sido mis guías en mi formación académica y humana. Y que sin su apoyo hasta el final no hubiese podido llegar hasta el punto en que estoy en este momento.

A mi hermana Gladys Beatriz quien me ha impulsado a seguir adelante y cuya compañía me ha ayudado para no desistir en mis estudios, esforzarme a ser mejor cada vez y conseguir cumplir mis objetivos estudiantiles y laborales.

A mi jefe y compañeros de trabajo quienes me han apoyado durante mis últimos años de carrera y han sido comprensivos conmigo respecto a mis obligaciones laborales y académicas, dándome la oportunidad de concluir mis estudios a la par de adquirir experiencia en el mundo laboral.

A mis amigos y demás personas que me han acompañado en este proceso académico y que han aportado a consolidar el cumplimiento de mis metas personales de diversas formas.

A mis profesores de la Escuela Politécnica Nacional, quienes han sido una guía a lo largo de mi carrera y a quienes debo el conocimiento que actualmente poseo y aplico en la vida laboral.

Finalmente, a mis compañeros de carrera con quienes he compartido experiencias y que de una u otra forma mantendré en mis recuerdos en los años venideros.

AGRADECIMIENTO

Agradezco a mis profesores de la Escuela Politécnica Nacional, quienes han sido una guía a lo largo de mi carrera y a quienes debo el conocimiento que actualmente poseo y aplico en la vida laboral y a mis compañeros de carrera con quienes he compartido experiencias y que de una u otra forma mantendré en mis recuerdos en los años venideros.

Contenido

DECLARACIÓN	ii
CERTIFICACIÓN	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
Índice de Figuras	3
Índice de Tablas	5
Resumen	6
Abstract	7
1. Introducción	8
1.1. Antecedentes.....	8
1.2. Objetivos	9
1.2.1. Objetivo General.....	9
1.2.2. Objetivos Específicos	9
2. Marco Teórico.....	11
2.1. Metodología de desarrollo Mobile-D	11
2.1.1. Historias de usuario	12
2.2. Automatización de pruebas	13
2.3. Estándar de verificación de seguridad de aplicaciones móviles (MASVS)	13
3. Desarrollo de la aplicación: Mobile-D.....	16
3.1. Fase de exploración	16
3.1.1. Establecimiento de <i>stakeholders</i>	16
3.1.2. Definición de alcance.....	17
3.1.3. Establecimiento del proyecto	20
3.2. Fase de inicialización.....	30
3.2.1. Configuración del proyecto	30
3.2.2. Día de prueba (iteración cero)	34
3.3. Fases de producción, estabilización y pruebas.....	38
3.3.1. Funcionalidad de aprobaciones	38
3.3.2. Funcionalidad de notificaciones	57
3.3.3. Funcionalidad de presentación de reportes gráficos	60

3.3.4.	Otras funcionalidades	65
3.3.5.	Requisitos de seguridad MASVS	70
3.3.6.	Resumen de producción	77
4.	Resultados.....	81
5.	Conclusiones y recomendaciones	84
5.1.	Conclusiones	84
5.2.	Recomendaciones	85
6.	Referencias	87

Índice de Figuras

Figura 1: Ciclo de desarrollo traducido de Mobile-D [4].....	11
Figura 2: Resumen de fases de Mobile-D.....	12
Figura 3: Requerimientos de MASVS v1.3.....	15
Figura 4: Muestra de la gestión de priorización de requerimientos con <i>Confluence</i>	21
Figura 5: Diagrama de implementación	22
Figura 6: Esquema de navegabilidad.....	23
Figura 7: Muestra de historia de usuario ACT-10 de fase inicial [4].....	26
Figura 8: Gestión de prioridad de historias de usuario con Jira.....	26
Figura 9: <i>DevExpress NuGet Package</i>	32
Figura 10: Repositorio Bitbucket e historias de usuario.....	33
Figura 11: Tablero Kanban con historias de usuario escogidas	34
Figura 12: Muestra de aplicación de SBE en historia ACT-10.....	35
Figura 13: Archivo CSV de datos de entrada de historia ACT-10.....	36
Figura 14: Métodos para automatización de pruebas.....	36
Figura 15: Ejecución pruebas unitarias.....	37
Figura 16: Publicación Servicio Compras.....	37
Figura 17: Muestra de método de prueba	39
Figura 18: Muestra de métodos del servicio WCF de compras	39
Figura 19: Muestra de pantalla del sistema CG/Web: Buzón de solicitudes.....	40
Figura 20: Mockup de pantalla: Buzón de solicitudes.....	40
Figura 21: Instalación Paquetes <i>NuGets</i>	41
Figura 22: Archivos para implementar <i>FontAwesome</i>	41
Figura 23: Configurar propiedades de archivos de <i>FontAwesome</i>	42
Figura 24: Uso de iconos <i>FontAwesome</i> por el nombre.....	42
Figura 25: Pantalla buzón de solicitudes versión inicial.....	44
Figura 26: Pantalla buzón de solicitudes versión media.....	45
Figura 27: Pantalla buzón de solicitudes versión final.....	45
Figura 28: Pantalla web de aprobación de solicitud	46
Figura 29: Mockup aprobación de solicitud de compra	47
Figura 30: Mockup pantalla de detalle de solicitud.....	47
Figura 31: Pantalla de aprobación y detalle de solicitud versión inicial	48
Figura 32: Pantalla de aprobación y detalle de solicitud versión 2	49
Figura 33: Pantalla de aprobación de solicitud de compra versión final	50
Figura 34: Entidad que hereda de solicitud de compra	51
Figura 35: Campos adicionales agregados a solicitudes OT y PS	52

Figura 36: Buzón de órdenes de compra	53
Figura 37: Pantalla de aprobación de orden de compra.....	53
Figura 38: Muestra de historia ACT-1 original.....	54
Figura 39: Muestra de historia ACT-1 modificada	55
Figura 40: Consumo de servicio web en lógica de negocio.....	55
Figura 41: Pantalla de aprobación de solicitud de vacaciones	56
Figura 42: Control de ingreso de observación para aprobación	56
Figura 43: Método de envío de notificación.....	58
Figura 44: Ejemplo de notificación de solicitud / orden pendiente de aprobación.....	59
Figura 45: Estructura de <i>trigger</i> de compras	60
Figura 46: Mockup de pantalla de análisis mediante gráfica	61
Figura 47: Pantalla de filtrado de datos del gráfico	62
Figura 48: Gráficas de resumen de ítems / proveedores.....	62
Figura 49: Gráfica para análisis de cantidad de órdenes de compra atendidas	63
Figura 50: Pantalla de filtro de datos para reporte gráfico de rubros	64
Figura 51: Muestra de reporte gráfico de resumen de rubros.....	64
Figura 52: Pantalla principal del sistema CG/Web	66
Figura 53: Mockup menú principal app	66
Figura 54: Pantalla de menú principal.....	67
Figura 55: Mockup de pantalla de inicio de sesión (Login).....	68
Figura 56: Pantalla de inicio de sesión (Login).....	68
Figura 57: Mockup de pantalla de menú de opciones de modulo.....	69
Figura 58: Pantalla de menú de opciones del módulo.....	70
Figura 59: Pantalla de Login - campo contraseña	73
Figura 60: Certificado SSL/TLS asociado a servidor.....	74
Figura 61: Implementación de SSL/TLS en aplicación	75
Figura 62: Configuración <i>release</i>	76
Figura 63: Manejo de excepciones - mensajes en controles.....	77
Figura 64: Manejo de excepciones - <i>popUps</i>	77
Figura 65: Cantidad de historias completadas por tipo de funcionalidad	78
Figura 66: Gráfico resumen historias creadas vs finalizadas.....	80
Figura 67: Resultados de <i>Checklist</i> de aplicación de MASVS v1.3	82
Figura 68: Nivel de cumplimiento de estándar de seguridad MASVS.....	83

Índice de Tablas

Tabla 1: Definición de responsables del proyecto	19
Tabla 2: Requerimientos	20
Tabla 3: Planificación de Fases	29

Resumen

La empresa Información Tecnológica del Ecuador S.A. (I.T. del Ecuador) ofrece el producto software instalable Control Gerencial / Web (CG/Web) para la gestión de procesos críticos de empresas. Desde el año 2019 se han producido eventuales retrasos en la ejecución de los procesos que involucran acciones de usuarios de nivel gerencial debido a sus necesidades de movilidad. Como solución se propuso el desarrollo CGApp, que es una aplicación móvil nativa que brinda acceso remoto a través de servicios WCF. Para asegurar la calidad del producto final, en este proyecto se aplicó la metodología de desarrollo de software Mobile-D en conjunto con prácticas orientadas a pruebas; así como *DevExpress* y la plataforma *XamarinForms* para la generación de interfaces de usuario semejantes al sistema CG/Web así como el estándar de seguridad móvil MASVS v1.3, para brindar un entorno seguro y confiable. CGApp incorpora los procesos de nivel gerencial más importantes como son la aprobación de solicitudes y órdenes de compras, así como la aprobación de solicitudes de vacaciones. Para la revisión y aprobación de estas solicitudes la aplicación integra las actividades de consulta de datos, la actualización de estados y la capacidad de emitir notificaciones. Esta última es una característica de valor ya que logra disminuir los “tiempos muertos” en la espera de aprobación por parte de los usuarios gerenciales en los procesos. La integración de esta aplicación con el sistema CG/Web permite que sea ofertado a manera de paquete. Esto otorga a I.T. del Ecuador la posibilidad de presentar un producto más competitivo y de mayor categoría tecnológica en el mercado nacional.

Palabras clave: aplicación móvil nativa, Mobile-D, *DevExpress*, *XamarinForms*, nivel gerencial, notificaciones.

Abstract

“Información Tecnológica del Ecuador S.A.” (“I.T. del Ecuador”) is an business that offers the installable software product “Control Gerencial / Web” (CG/Web) to support the critical processes management of businesses. Since 2019 it has been suffering of delays in the execution of processes that involve managerial level user actions due to their mobility needs. Like a solution, it was proposed develop CGApp, a native mobile application that provides remote access using WCF services. Mobile-D Software develop methodology along with practices focused in testing were applied for assuring the final product quality. *DevExpress* and the *XamarinForms* framework were applied to generate and maintain the look and feel of the CG/Web’s user interfaces and security standard MASVS v1.3 to offer a secure and trust environment. CGApp incorporates the most important manager level processes like purchase requests and order approval as well as vacation requests approval. For reviewing and approval process, the app integrates checks of data, update states and the functionality to notify events. This one is a valuable characteristic because it would influence on the decrement of “death times” in the managerial level user approval processes. The integration with CH/Web could be offered like a package allowing “I.T. del Ecuador” the possibility of presenting a more competitive and technologically categorized product in the national market.

Key words: native mobile application, Mobile-D, *DevExpress*, *XamarinForms*, manager level, notifications

1. Introducción

En esta sección se describen la situación inicial que llevó a la realización del presente proyecto y los objetivos que se buscan cumplir. Los antecedentes resumen lo investigado y discutido durante la propuesta del proyecto.

1.1. Antecedentes

La empresa Información Tecnológica del Ecuador S.A. (I.T. del Ecuador) es una empresa ecuatoriana cuyo giro de negocio se enfoca al desarrollo y distribución de software y hardware [1]. En la actualidad, su producto estrella es el sistema software ERP¹ denominado Control Gerencial / Web (CG/Web) [1]. Dicho sistema se orienta al control de los procesos operativos y administrativos de empresas públicas y privadas a nivel nacional.

CG/Web se oferta como un producto instalable a los clientes de I.T. del Ecuador y, por motivos de seguridad, su acceso se encuentra circunscrito a las intranets del cliente restringiendo el ingreso al sistema al interior de sus oficinas. Para los casos en donde un usuario necesita ingresar desde el exterior de las instalaciones, es necesario la configuración de una red VPN.

Sin embargo, la arquitectura inicial no contempla la capacidad de lidiar con los requerimientos actuales de movilidad de los usuarios de nivel gerencial. Por esta razón, los procesos gestionados en CG/Web han sufrido retrasos significativos en su ejecución según lo indica un levantamiento de información realizado por I.T. del Ecuador. Durante el año 2019 el 80% de los procesos fueron postergados [2] debido al retraso en el cumplimiento de actividades relacionadas con los usuarios gerenciales del sistema. Estas actividades incumplidas provocaron un estancamiento en el desarrollo de los procesos de la empresa cliente, lo cual les conllevó a multas y recargos. Este estancamiento evidencia la necesidad de contar con un medio de acceso inmediato al sistema, que no esté condicionado por la existencia de VPNs en la infraestructura tecnológica de los clientes.

En el mercado nacional, entre los sistemas de información ERP que compiten con CG/Web se encuentran opciones que cuentan con un acceso mediante dispositivos móviles. Sistemas como Odoo², por ejemplo, cuentan con integración de una aplicación móvil que permite el acceso remoto a información y procesos de sus clientes. Además,

¹ Acrónimo de Enterprise Resource Planning

² Sistema ERP de Leadsolutions, solución que se especializa para empresas privadas.
https://www.odoo.com/es_ES/

estos proporcionan notoriedad e interés en el mercado por sus interfaces de fácil manejo para los usuarios.

Debido a la competencia agresiva en el mercado y a la necesidad de modernización, se ha considerado que CG/Web incorpore una solución móvil que satisfaga las necesidades de sus clientes y permita mejorar la participación de los usuarios gerenciales en los procesos empresariales. De igual forma, I.T. del Ecuador busca dar un mayor valor competitivo al sistema frente a otras soluciones existentes a nivel nacional.

Por lo expuesto, el presente trabajo tiene como objetivo desarrollar una solución que permita agilizar la ejecución de los procesos que requieren intervención de usuarios gerenciales. Para esto se analizarán las funcionalidades del sistema web que formen parte de las necesidades de dichos usuarios y adaptarlas a un entorno móvil. Para la gestión del desarrollo de esta aplicación se utilizarán prácticas, técnicas y metodologías que permitan asegurar su calidad. Esta solución notificará información relevante sintetizada y priorizada al usuario de forma que componente reduzca los retrasos de ejecución de los principales procesos gestionados en el sistema CG/Web.

1.2. Objetivos

En esta sección se describen las metas que se buscan cumplir con el presente proyecto. Estos objetivos se basan en los requerimientos del cliente y en lo discutido durante la propuesta de tema de proyecto.

1.2.1. Objetivo General

Desarrollar una aplicación móvil segura para el sistema CG/Web a fin de agilizar la ejecución de los procesos que requieren intervención de usuarios gerenciales.

1.2.2. Objetivos Específicos

- Asegurar la calidad de la aplicación móvil mediante el uso de la metodología propuesta en el proceso de desarrollo de software.
- Agilizar la ejecución de los procesos críticos manejados por el sistema CG/Web mediante el análisis y comprensión de los procesos, datos, y experiencias de usuario para determinar los “tiempos muertos”, necesidades de diseño y arquitectura de la aplicación móvil.

- Integrar una aplicación móvil al sistema CG/WEB a través del estudio y uso de tecnologías de comunicación y seguridad que permitan potenciar las capacidades que ofrece la solución actual.

2. Marco Teórico

En esta sección se describen los contenidos teóricos investigados y aplicados durante el proyecto de desarrollo. Se presenta un resumen de la metodología empleada y temas complementarios a la ejecución automática de pruebas en C#.

2.1. Metodología de desarrollo Mobile-D

Según la documentación de Mobile-D, esta metodología es útil para la gestión del desarrollo del presente proyecto debido a cuatro características. Esta metodología se basa en el enfoque ágil, por lo cual provee versatilidad ante posibles cambios durante el desarrollo. Mobile-D sigue un proceso incremental que permite la presentación de productos funcionales al cliente, favoreciendo la retroalimentación. La metodología está diseñada para un grupo de trabajo pequeño, ideal de 10 personas [3], lo cual es beneficioso por la cantidad de responsables del proyecto. Esta es presentada con cinco fases de ciclo de vida que le hacen ideal para la gestión de desarrollo de aplicaciones móviles [4] como se puede ver en la Figura 1.



Figura 1: Ciclo de desarrollo traducido de Mobile-D [4]

En cada fase de Mobile-D se deben cumplir una cantidad de tareas y objetivos que permiten el avance en el proyecto. En la Figura 2 se muestra un resumen de los contenidos de cada fase que se indican en la documentación publicada en el sitio web Virtual.vtt.fi [4].

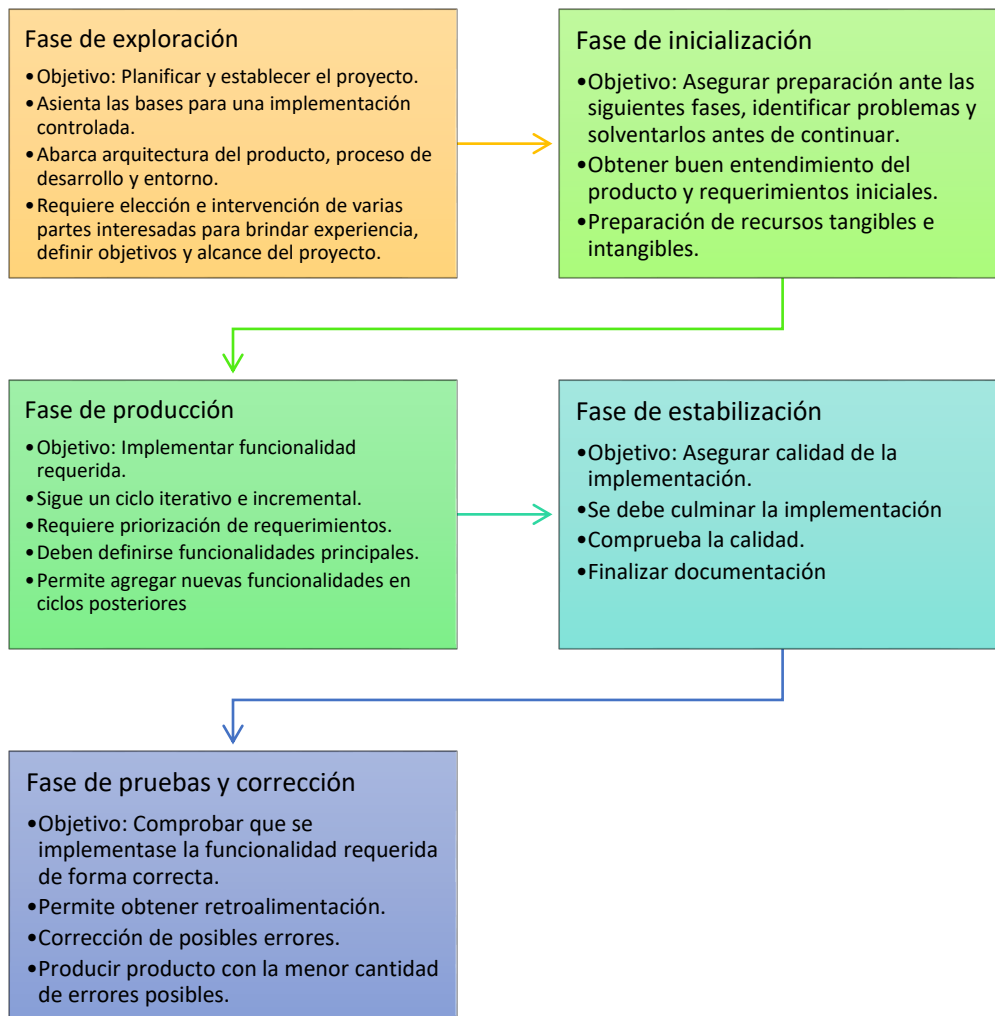


Figura 2: Resumen de fases de Mobile-D

2.1.1. Historias de usuario

Como parte de la fase de exploración en la metodología Mobile-D los requerimientos recolectados y definidos para el proyecto deben ser analizados en función de las experiencias propias de los usuarios. Para esto se recomienda el uso de las “historias de usuario”, la cual es un conjunto de información que contiene una explicación detallada sobre los procesos que componen un requerimiento.

Una historia de usuario debe describir el cumplimiento de un requerimiento desde el punto de vista del usuario final, es decir, de aquel que conoce el proceso y hará uso de la solución. Por ello es importante que la redacción de una historia de usuario sea apoyada por la presencia e intervención de los usuarios y siga una plantilla que incluye el tipo de usuario, el objetivo que busca cumplir y lo que requiere para cumplirlo. Esta estructura es común en el desarrollo ágil, como se menciona en el blog de la compañía Mountain Goat Software.

Como parte del proceso de desarrollo es importante que estas historias cuenten con datos que permitan al equipo identificar los tiempos aproximados de desarrollo, los responsables de cumplir con la implementación, el registro de trabajo y un registro de posibles cambios en la historia. Estos datos si bien no forman parte integral de la estructura de una historia de usuario, son relevantes para llevar un control en el progreso del trabajo del proyecto.

2.2. Automatización de pruebas

Un proyecto de pruebas unitarias del IDE Visual Studio, el cual utiliza el *framework* MS Test, proporciona un medio para declarar métodos que serán llamados por el ejecutor en tiempos específicos [5]. En este proyecto se utilizaron los métodos de tipo *ClassInitialize* y *TestCleanup*.

Los métodos de tipo *ClassInitialize* son ejecutados una sola vez antes de ejecutar los métodos de prueba de una clase [5]. Estos métodos permiten configurar pre condiciones que requieren las pruebas y que afectarán sus resultados. En caso de múltiples iteraciones de ejecución, estos métodos sólo se ejecutarán antes de la primera iteración.

Los métodos de tipo *TestCleanup* son ejecutados una vez cada que se han ejecutado las pruebas de una clase [5]. Estos métodos permiten realizar la limpieza de datos, archivos y configuraciones producidas por las pruebas. En caso de múltiples iteraciones de ejecución, estos métodos son ejecutados una vez al final de cada iteración.

2.3. Estándar de verificación de seguridad de aplicaciones móviles (MASVS)

El *Mobile Application Security Verification Standard* (MASVS) es un esfuerzo por establecer un marco de requisitos de seguridad necesarios para diseñar, desarrollar y aprobar la seguridad de aplicaciones móviles iOS y Android [6]. Parte de sus objetivos es proporcionar una guía durante las fases de desarrollo de aplicaciones móviles seguras y pruebas de seguridad. Para ello el estándar define 3 niveles de verificación de seguridad en base a su alcance en la protección de datos en la aplicación móvil.

El primer nivel (MASVS-L1) o seguridad estándar contiene requerimientos genéricos de seguridad que se recomiendan para todas las aplicaciones móviles. Este abarca las vulnerabilidades más comunes y permite obtener una aplicación móvil que se adhiere a las mejores prácticas de seguridad. Con este nivel se cumplen los requisitos básicos en calidad de código, manejo de datos sensibles e interacción con el entorno móvil.

El segundo nivel (MASVS-L2) o defensa en profundidad añade controles que permiten obtener una aplicación resistente a ataques más sofisticados. Es recomendado para

aplicaciones que manejan datos de sensibilidad alta y en las que el usuario final no sea considerado como potencial adversario. Para su aplicación requiere un modelo de amenazas y la seguridad debe formar parte fundamental de la arquitectura y diseño de la aplicación.

El tercer nivel (MASVS-R) se enfoca en la resistencia contra ingeniería inversa y la manipulación desde el lado del cliente. Este nivel es recomendado para aplicaciones que manejan datos de alta confidencialidad o que requieran la protección de propiedad intelectual del producto. Con este nivel se ayuda a la prevención de amenazas específicas cuando el usuario final es considerado malicioso o el sistema operativo móvil ha sido comprometido.

El estándar abarca un total de ocho requerimientos que cubren elementos de un producto de software móvil. Cada requerimiento busca cumplir un objetivo de control mediante la aplicación de requisitos de seguridad que se detallan en el documento. En la Figura 3 se muestra un resumen de los requerimientos y los objetivos de control que busca cumplir cada uno.

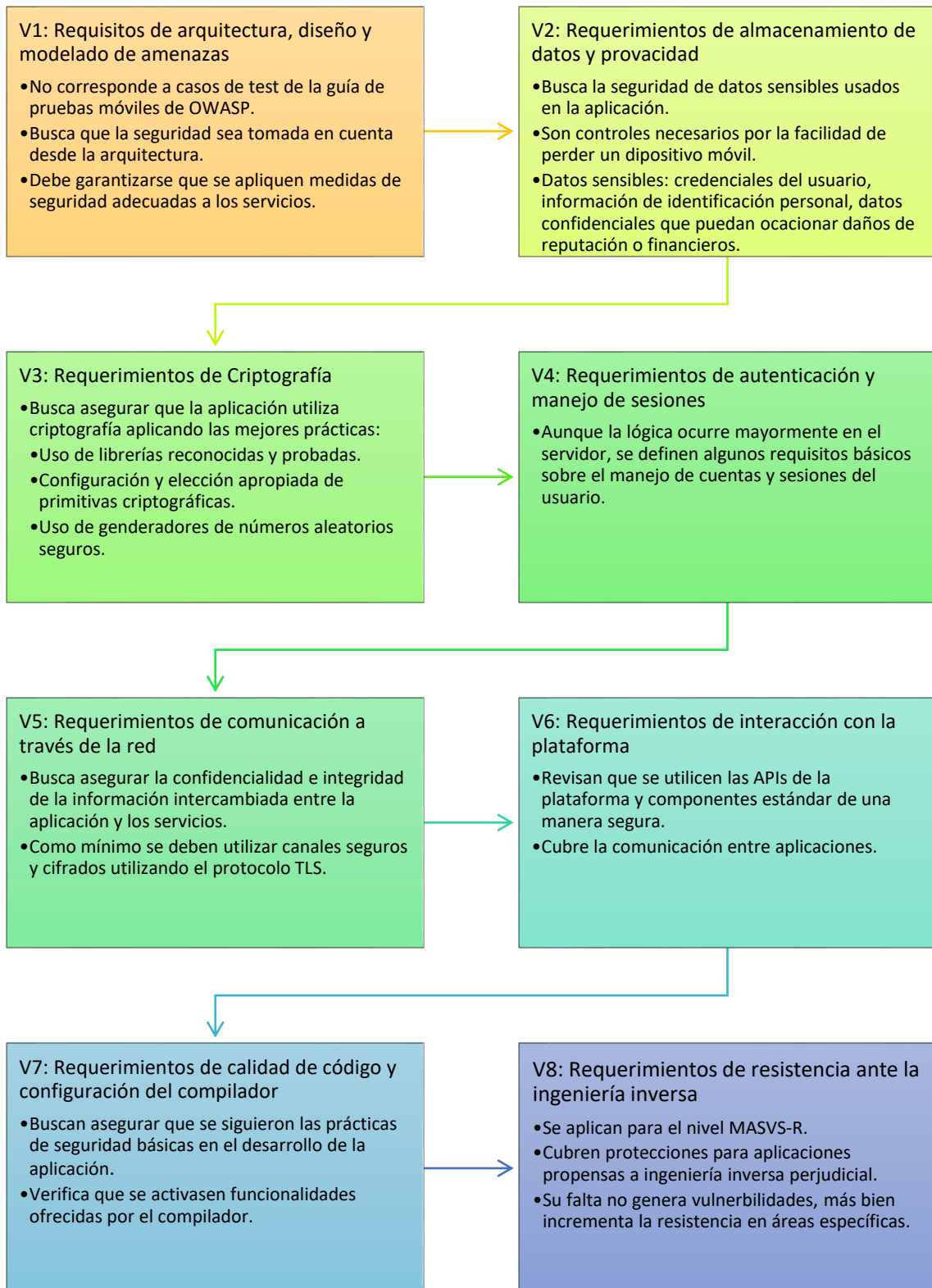


Figura 3: Requerimientos de MASVS v1.3

3. Desarrollo de la aplicación: Mobile-D

En esta sección se describen las fases que se siguieron durante el desarrollo del proyecto. Estas fases se corresponden con las fases de Mobile-D que se describen en la Sección 2.1 en la Figura 1.

3.1. Fase de exploración

Durante esta fase se buscó conocer lo mejor posible el sistema e identificar los recursos humanos y materiales para el proyecto, como se mencionan en la Sección 2.1 en la Figura 2. Las tareas correspondientes a esta fase se adaptaron para solventar las necesidades del proyecto.

3.1.1. Establecimiento de *stakeholders*

En esta fase se determinó a las personas responsables y sus labores en el presente proyecto. Se definieron responsabilidades para la gestión del proyecto, el desarrollo de la aplicación, el equipo de exploración y las pruebas. La definición de responsables y sus responsabilidades se resume en la Tabla 1.

Como responsable de la gestión y desarrollo del proyecto se designó al autor de la tesis como el responsable. Entre sus responsabilidades se encuentran la planificación del proyecto, la investigación de elementos técnicos necesarios para solventar los requerimientos, dirigir y efectuar el desarrollo de la aplicación móvil. Para complementar la investigación que tuvo que llevar a cabo se lo incluyó en el equipo de exploración a fin de que adquiriera un mayor entendimiento del sistema CG/Web.

El equipo de exploración es un grupo de expertos en el sistema que apoyarán al desarrollo con su conocimiento técnico y operativo. La responsabilidad principal de este grupo es el asesoramiento durante el desarrollo para el entendimiento de las funcionalidades de CG/Web. Para esto se eligieron miembros de I.T. del Ecuador especialistas en los módulos del sistema, así como en su arquitectura y código fuente. Entre los escogidos se encuentra el Ingeniero Marco Muzo quien es el Gerente General de la empresa y Product Owner de CG/Web. La Ingeniera Sandra Oñate quien la Jefa del grupo de desarrollo y por tanto quien mayor conocimiento técnico tiene del sistema web. A nivel operativo se eligió al Ingeniero Alexis Alarcón, encargado de la implementación del módulo de nómina y activos fijos. El Ingeniero Jorge Díaz quien funge como consultor senior de los módulos de bodegas, tesorería y seguridades. Finalmente el Ingeniero Klever Yáñez quien presta servicios de consultor senior operativo y de desarrollo de los módulos de nómina y compras. La interacción durante

el proyecto con los expertos mencionados fue prominente para conocer las directrices de los procesos y las necesidades de los usuarios de nivel gerencial.

3.1.2. Definición de alcance

En esta sección se describen las sesiones que se efectuaron durante el proyecto para la obtención de los requerimientos funcionales y no funcionales de la aplicación. Para la extracción de los requerimientos funcionales se realizaron 4 sesiones virtuales con los responsables especificados en la Tabla 1.

El objetivo de la primera sesión fue dialogar con los *stakeholders* acerca de los procesos que se abordarían en el proyecto y definir las actividades que se adaptarían a la aplicación móvil. Durante la reunión se analizaron los resultados del levantamiento de información acerca de los retrasos en la ejecución de los procesos gestionados en CG/Web de los clientes. Este retraso se evidenciaba en tiempos de espera para la aprobación en nivel gerencial de las solicitudes de compras, vacaciones y bodegas. Estos tiempos coincidían con épocas del año en que estos usuarios se encontraban en viajes laborales constantes. En base a esto se enfocó la reunión hacia las funcionalidades que permitan la aprobación de estas solicitudes y un medio para informar al usuario la existencia de solicitudes pendientes de aprobación. Se obtuvo esquematizaciones y notas de las características generales de la aplicación y las funcionalidades dialogadas.

Durante la segunda reunión el objetivo fue continuar y terminar el trabajo de la primera para culminar la recolección de ideas de funcionalidades. En esta ocasión se centró la reunión en funciones complementarias a las aprobaciones discutidas en la reunión anterior. Se definieron funcionalidades de consulta de información mediante reportes gráficos. Estos reportes deben indicar información que permita al usuario conocer el estado actual de la empresa y poder tomar decisiones en base a sus datos. Tras esta reunión se obtuvo una lista a implementar en la aplicación móvil que se utilizó como base para la definición de los requerimientos en las siguientes reuniones.

En la tercera reunión se buscó definir los requerimientos en base a las funcionalidades del sistema web obtenidas y dialogadas. Con esto se efectivizaron los requerimientos para el proyecto y se solicitó determinar la importancia de los módulos para el usuario, a fin de priorizar los requerimientos. Durante esta reunión se empezó a discutir características no funcionales de la aplicación.

En la cuarta reunión se analizó la información solicitada y se evidenció que el estado de los módulos en el sistema web no era homogéneo y su importancia para los clientes era diversa. Se llegó al consenso de que el módulo más utilizado y con mayor valor para los clientes era el Módulo de Compras, enfocado a los procesos de adquisición de bienes y servicios. El segundo en importancia es el Módulo de Nómina, en el cual se gestionan los ingresos y descuentos de los trabajadores de los clientes. Por último, el Módulo de Bodegas resultaba tener menor relevancia en comparación debido a que es el módulo más reciente en ser desarrollado y cuenta con varias limitaciones de diseño y operación. Con esta información se acordó priorizar los requerimientos para que se dé mayor relevancia a las funcionalidades más utilizadas en el sistema.

Para las características no funcionales del proyecto se coordinó una nueva reunión virtual con los *stakeholders* descritos en la Tabla 1. En esta se determinó la arquitectura general de la aplicación, se examinó de forma general el código fuente de la versión web del sistema y se identificaron recursos que se legarían para el desarrollo de la aplicación. Durante esta reunión se discutió la aplicación del estándar MASVS v1.3 para la aplicación. Se determinó que la información que manejará la aplicación no requiere de una protección elevada y por tanto se consensó que se aplicarían los requerimientos de controles de seguridad para el nivel MASVS-L1 descrito en la sección 2.3.

Los requerimientos se encuentran descritos en la Tabla 2, el código RF indica que se trata de un requerimiento funcional, mientras que el código NF los marca como requerimientos no funcionales. Los requerimientos de controles de seguridad considerados son los indicados en la documentación del estándar MASVS v1.3 [6] para el nivel MASVS-L1. Se concertó utilizar el Mobile App Security Checklist 1.1 [7], oficial de OWASP, para gestionar el cumplimiento de los requerimientos de seguridad y obtener una evaluación final de la calidad de la aplicación móvil en dicho apartado.

Tabla 1: Definición de responsables del proyecto

Nombre	Responsabilidades
Autor	
Carlos Aguirre (Desarrollador)	<ul style="list-style-type: none"> • Planificación del proyecto • Investigación sobre aspectos técnicos necesarios para el desarrollo de la aplicación. • Desarrollo de la aplicación móvil. • Corrección de errores.
Product owner	
Ing. Marco Muzo (Gerente general)	<ul style="list-style-type: none"> • Definición de requerimientos. • Asesoramiento sobre arquitectura y funcionalidades del sistema web CG/Web. • Pruebas de la aplicación.
Otros <i>stakeholders</i>	
Ing. Sandra Oñate (Jefa del grupo de desarrollo de software)	<ul style="list-style-type: none"> • Definición de requerimientos. • Asesoramiento en arquitectura, funcionalidades, desarrollo y mantenimiento de CG/Web • Pruebas de la aplicación
Ing. Alexis Alarcón (Director de implementación: módulos de Nómina y Activos Fijos)	<ul style="list-style-type: none"> • Definición de requerimientos. • Asesoramiento en funcionalidad de módulos de Nómina. • Pruebas de la aplicación.
Ing. Jorge Díaz (Consultor Senior de implementación: módulo de Bodegas, Tesorería y Seguridades)	<ul style="list-style-type: none"> • Definición de requerimientos. • Asesoramiento sobre funcionalidad de módulos de Bodegas. • Pruebas de la aplicación
Ing. Kléver Yáñez (Consultor Senior de implementación y desarrollo: módulo de Compras y Nómina)	<ul style="list-style-type: none"> • Definición de requerimientos. • Asesoramiento sobre funcionalidad de módulo Nómina y módulo Compras. • Pruebas de la aplicación

Tabla 2: Requerimientos

Código	Requerimiento
Requerimientos funcionales	
RF1	La aplicación debe permitir realizar la aprobación de una solicitud de vacaciones.
RF2	La aplicación debe permitir realizar un análisis comparativo de los grupos de rubros de nómina.
RF3	La aplicación debe permitir realizar un análisis de horas extras.
RF4	La aplicación debe permitir realizar la aprobación de una solicitud de bodega.
RF5	La aplicación debe permitir realizar un análisis de inventario.
RF6	La aplicación debe permitir realizar la aprobación de una solicitud de compra.
RF7	La aplicación debe permitir realizar la aprobación de una orden de compra.
RF8	La aplicación debe permitir realizar un análisis de las órdenes de compras.
RF9	La aplicación debe permitir recibir notificaciones cuando existan solicitudes pendientes de aprobación
Requerimientos no funcionales	
NF1	El desarrollo de la aplicación debe ser realizado para el sistema operativo Android, para la versión actual más estable y no se tomarán en cuenta versiones inferiores a la misma.
NF2	Los datos actualizados por la aplicación deben actualizarse para los usuarios de la versión web del sistema CG/Web.
NF3	Las notificaciones enviadas a la aplicación deben identificar la actividad que el usuario debe utilizar.
NF4	El desarrollo de la aplicación debe seguir la arquitectura que utiliza el sistema CG/Web.
NF5	El acceso a la aplicación será permitido con las credenciales que los usuarios tengan asignados a su perfil del sistema CG/Web.
NF6	Los usuarios deben tener acceso a los módulos de la aplicación en base al sistema de roles y permisos del sistema CG/Web.
NF7	La aplicación debe ser desarrollada utilizando la herramienta de desarrollo DevExpress.

3.1.3. Establecimiento del proyecto

En esta sección se describen las tareas realizadas en base a la documentación de Mobile-D que se encuentra resumida en la Sección 2.1 en la Figura 2. El establecimiento del proyecto consiste principalmente en la identificación de recursos necesarios para llevar a cabo el proyecto.

3.1.3.1. Análisis de requerimientos funcionales

Tras las sesiones con el grupo de *stakeholders* se realizó la priorización de los requerimientos funcionales. Se asignó el nivel de prioridad en función del módulo más

utilizado por los clientes del sistema web, el cual se indicó en la Sección 3.1.2. Además, se determinó su nivel de importancia en función a su relevancia en el proceso de aprobación de solicitudes. Para la gestión de estos requerimientos se utilizó la herramienta online *Confluence*³ del grupo *Atlassian* (Figura 4), la cual permite asignar la prioridad de los requerimientos mediante etiquetas de importancia.

Código	Requerimiento	Importancia
2	La aplicación debe permitir realizar un análisis comparativo de los grupos de rubros de nómina.	MEDIA
3	La aplicación debe permitir realizar un análisis de horas extras.	MEDIA
4	La aplicación debe permitir realizar la aprobación de una solicitud de bodega.	BAJA
5	La aplicación debe permitir realizar un análisis de inventario.	BAJA
6	La aplicación debe permitir realizar la aprobación de una solicitud de compra.	ALTA
7	La aplicación debe permitir realizar la aprobación de una orden de compra.	ALTA

Figura 4: Muestra de la gestión de priorización de requerimientos con *Confluence*

3.1.3.2. Análisis y diseño del sistema

Después del análisis de los requerimientos iniciales se realizó el análisis y diseño de los componentes de la aplicación móvil. Se definió el modelo arquitectónico del proyecto y su división. En esta sección se detallan los elementos que forman parte del diseño, el modelo arquitectónico resultante se muestra en la Figura 5.

El desarrollo de la aplicación móvil debía seguir un modelo arquitectónico similar al utilizado por el sistema CG/Web por lo que se utilizó un modelo conceptual de cliente/servidor. La aplicación móvil funge como cliente, que le permite al usuario conectarse mediante internet a un conjunto de servicios publicados por un servidor.

El proyecto cuenta con dos secciones: frontend y backend. La primera involucra la aplicación móvil y los componentes relacionados a los dispositivos que manejará el

³ Confluence es una herramienta software enfocada a la gestión de proyectos empresariales de la empresa Atlassian. Permite administrar los objetivos, alcance, equipos de trabajo y requerimientos del proyecto. <https://www.atlassian.com/software/confluence>

usuario en el cliente. Por otro lado, la segunda sección abarca la programación de los servicios y su interacción con los datos del sistema. Ambas secciones se comunican mediante la Internet para el envío y recepción de datos.

La sección frontend abarca las pantallas que permiten la consulta de información para la aprobación de solicitudes, la presentación de reportes gráficos, la recepción de notificaciones y el ingreso de datos. Esta sección es nueva dentro del esquema del sistema CG/Web como se marca en el diagrama de implementación presentado en la Figura 5. Para el desarrollo de las pantallas de la aplicación de esta sección se propuso utilizar de base las pantallas de CG/Web, pero adaptándolos al ambiente móvil y modernizándolos.

Para la sección backend se planteó legar la funcionalidad existente en el sistema web, el consumo de los métodos y entidades. El desarrollo de esta parte se enfocó principalmente en la transformación de los conjuntos de datos utilizados en CG/Web a un formato que permita su comunicación con las aplicaciones clientes, en este caso se optó por *JSON*. Para funcionalidades que no tenían un equivalente en el sistema web fue necesario adicionar nuevos métodos de acceso a datos y lógica de negocio. Por ejemplo, se agregó una lógica nueva para la generación de registros que servirían de base para el envío de notificaciones. Por otro lado, para la presentación de reportes gráficos fue indispensable crear nuevos métodos que permitan el acceder y filtrar los datos necesarios. Estas adiciones se muestran señalizadas en el diagrama presentado en la Figura 5.

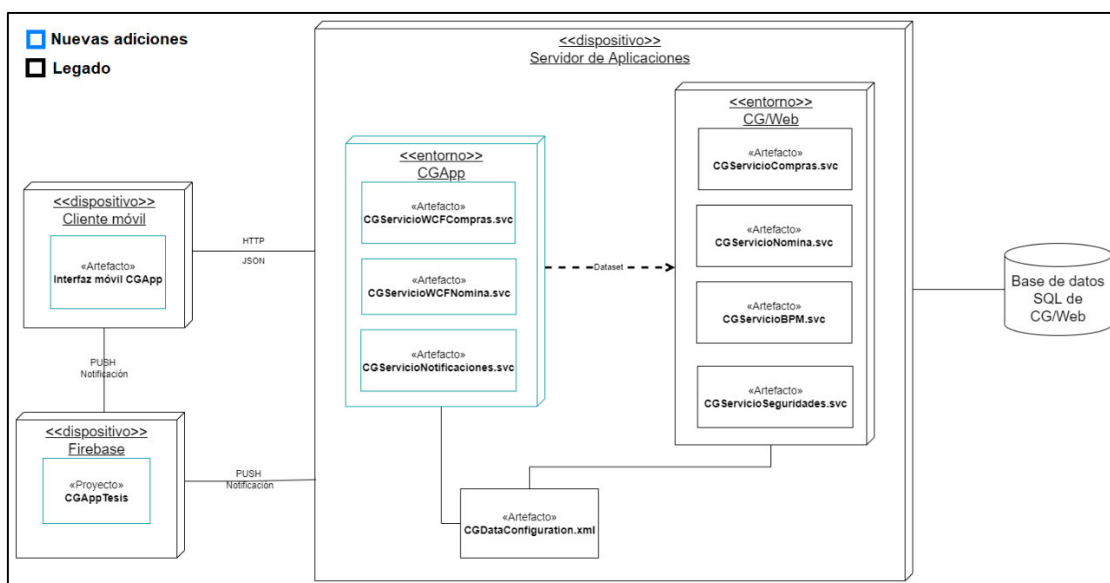


Figura 5: Diagrama de implementación

Para permitir el acceso desde los clientes mediante Internet se propuso el uso de servicios WCF que toman los datos generados por el método de la lógica de negocio. Estos utilizan un archivo web.config que contiene la información de la base de datos del módulo que el usuario desea utilizar, así como sus credenciales de acceso. A este nivel se maneja dependiendo de la configuración del sistema CG/Web del cliente (Ver Anexo 2).

Tras analizar los requerimientos iniciales se propuso que la aplicación móvil se divida en los siguientes módulos, basados en los módulos del sistema web, y sus funciones.

- 1) Módulo Compras
 - a. Aprobación solicitudes de compra.
 - b. Aprobación órdenes de compra.
 - c. Resumen compras.
- 2) Módulo Nómina
 - a. Aprobación de solicitud de vacaciones.
 - b. Análisis de horas extras
 - c. Análisis comparativo de rubros nómina
- 3) Módulo Bodegas
 - a. Aprobación de solicitud de bodega
 - b. Análisis de inventario

La aplicación APPCG/Web sigue el esquema de navegación de la Figura 6, el cual se diseñó en base a los requerimientos analizados en la Sección 3.1.3.2 y a la navegabilidad de CG/Web.

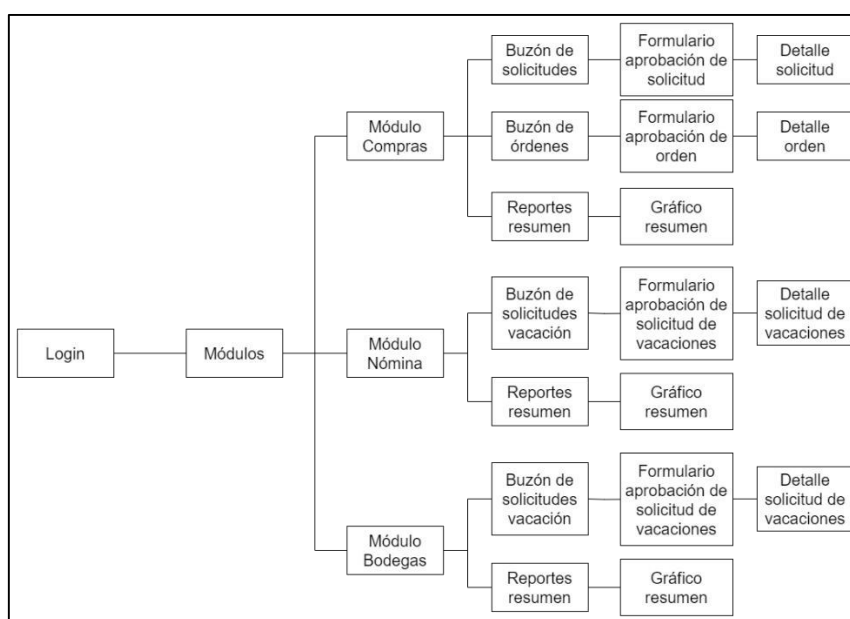


Figura 6: Esquema de navegabilidad

3.1.3.3. Diseño de la base de datos

Dado que la aplicación extiende al sistema CG/Web y depende de varias de sus funcionalidades, los datos fueron almacenados en la base de datos proporcionada por I.T. del Ecuador en su instalación. Cada módulo tiene un esquema de base de datos propio que se complementa con los demás. De forma general cada base tiene tablas para el almacenamiento de datos de entidades, operaciones y parámetros que se utilizan en cada módulo.

Se realizó un análisis estructural de la base de datos mediante ingeniería inversa para identificar las tablas necesarias en base a los requerimientos funcionales iniciales mencionados en la Sección 3.1.2 utilizando la herramienta *PowerDesigner*⁴. Para el presente proyecto se accedió a los esquemas de los módulos identificados en la sección 3.1.3.2 y al esquema del módulo financiero. Cabe recalcar que la mayoría de accesos a base se realizan mediante el consumo de métodos ya existentes en el sistema CG/Web; sin embargo se resumirá a continuación la interacción entre las tablas y los datos que intervienen en las funcionalidades de la aplicación.

La primera interacción destacable es la relación de los componentes de una solicitud y orden de compras. Para obtener los datos completos de una instancia de estas entidades se requiere la relación entre la tabla “cabecera de solicitud” (CabeceraSolicitud) y la tabla “detalle de solicitud”. Para identificar una solicitud en concreto es necesario obtener un listado de los datos de la tabla CabeceraSolicitud filtrado con la información del usuario aprobador. Esta interacción se repite con las tablas equivalentes en el módulo de bodegas.

La segunda interacción es la decodificación de parametrización. En CG/Web existen datos que se han parametrizado utilizando tablas de catálogos, guardando en las tablas únicamente los códigos que los empatan con un elemento en específico. En las solicitudes de compras esta información se encuentra en la cabecera y el detalle, por lo que es requerido el acceso a las tablas catálogos para el despliegue e identificación de información.

La tercera interacción es crítica para la aprobación de solicitudes de vacaciones. Los datos de cabecera y detalle de estas solicitudes se encuentran en tablas que pertenecen al módulo de nómina, sin embargo, requieren de tablas pertenecientes al módulo del

⁴ PowerDesigner es una de modelado de datos que permite el análisis, visualización y manipulación de metadatos. Facilita la generación de modelos de datos desde una base de datos existente. <https://www.powerdesigner.biz/ES/powerdesigner/powerdesigner-features.html>

BPM de CG/Web. El listado de las solicitudes de vacaciones depende de la tabla de tareas y la tabla de solicitudes del BPM. Mientras que el despliegue de la información de una solicitud relaciona a las tablas de cabecera, detalle y periodo de vacaciones del módulo de nómina. Al momento de realizar una aprobación se actualizan las cinco tablas mencionadas (2 del módulo BPM y 3 del módulo de nómina).

Como cuarta interacción importante para el proyecto es la existente entre las tablas de catálogos y las tablas de operaciones. Esta es necesaria para la obtención de datos de los reportes de nómina. Se requiere decodificar los datos de la tabla de operaciones de forma que sean entendibles para el usuario.

La última interacción destacable es del módulo financiero, pues es el eje de CG/Web al gestionar las tablas de seguridades. Esta interacción permite la autenticación de los usuarios, condicionar el acceso a opciones de menú mediante roles y el uso de los permisos que se asignan a los usuarios desde el módulo de seguridades del sistema web y que condicionan los accesos a la aplicación móvil.

3.1.3.4. Gestión de historias de usuario

En base a los requerimientos funcionales mencionados en la Sección 3.1.2 se construyeron las historias de usuario. Para su gestión se utilizó la herramienta online Jira Software⁵ que permite gestionar las historias de usuario siguiendo un formato similar al especificado en la documentación de la metodología, incluyendo apartados como los puntos asignados, tiempo estimado, tiempo trabajado, entre otros. En la Figura 7 se incluye una muestra de una historia de usuario de la fase inicial, indicando que está hecha conforme a la documentación de Mobile-D [4].

⁵ Jira Software es una herramienta software especializada para la gestión de proyectos de desarrollo de software de equipos ágiles de la empresa Atlassian. <https://www.atlassian.com/software/jira>



Figura 7: Muestra de historia de usuario ACT-10 de fase inicial [4]

Se asignaron niveles de prioridad desde Muy Baja hasta Muy Alta en función del nivel del requerimiento y la importancia de la actividad para completar el proceso de aprobación (Figura 8).

Tipo	Clave	Resumen	Persona asignada	Informador	Pr ↑
ACT-4	ACT-4	Como usuario gerencial del módulo de nómina se desea acceder a la información de la cantidad de los tipos de horas extras trabajadas en un año para realizar un análisis comparativo	Carlos Aguirre	Carlos Aguirre	▼
ACT-5	ACT-5	Como usuario gerencial del módulo de nómina se desea recibir una notificación para poder conocer la existencia de solicitudes de vacaciones pendientes de aprobación y poder aprobarlas	Carlos Aguirre	Carlos Aguirre	=
ACT-2	ACT-2	Como usuario gerencial del módulo de nómina se desea acceder a la información de los grupos de rubros de nómina para realizar un análisis global.	Carlos Aguirre	Carlos Aguirre	=
ACT-1	ACT-1	Como usuario gerencial del módulo de nómina se desea acceder a la información de una solicitud de vacaciones para poder aprobarla	Carlos Aguirre	Carlos Aguirre	=
ACT-17	ACT-17	Como usuario gerencial del módulo de compras se desea acceder a la información de las órdenes de compras efectuadas en un periodo de tiempo determinado para poder realizar un análisis	Carlos Aguirre	Carlos Aguirre	▲

Figura 8: Gestión de prioridad de historias de usuario con Jira

En la descripción de cada historia de usuario se incluyeron los posibles escenarios que se podían presentar para cada funcionalidad, como se describe en la sección 3.2.2. Estos escenarios fueron usados como base para la creación de pruebas unitarias que permitiesen comprobar cuando una historia de usuario fue solventada en el desarrollo.

Como agregado a la gestión se calificó el esfuerzo que se considera que será aplicado en cada historia de usuario con el valor de puntos de historia. Se utilizó una escala de 1 al 20 recomendada por la documentación de Jira Software [8], mientras más tiempo de implementación requiere la historia, más alto debe ser el puntaje. De esta forma se intentó precisar de mejor forma el tiempo estimado para completar la historia en cuestión.

3.1.3.5. Análisis de requerimientos y pruebas de seguridad

Para el diseño de las pruebas de seguridad en base al estándar MASVS v1.3 descrito en la sección 2.3 y el *Checklist* oficial de OWASP mencionado en la sección 3.1.2 se usó de base la guía de evaluación incluida en este último archivo. Cada requerimiento tiene un procedimiento de evaluación que describe las verificaciones que se deben efectuar para comprobar que dicho apartado ha sido cumplido por la aplicación.

Los procesos de evaluación dictaminan los diferentes aspectos que se deben considerar para evaluar el cumplimiento de uno o varios requerimientos de seguridad. Estos procesos pueden realizarse mediante pruebas estáticas, dinámicas o análisis de la aplicación. Este documento permite controlar el estado del requerimiento mediante tres opciones:

- **Pass:** Indica que la aplicación ha superado las evaluaciones y por tanto cumple con el requerimiento en cuestión.
- **Fail:** Significa que la aplicación no supera las pruebas y por ello no cumple con el requerimiento de seguridad.
- **N/A:** Indica que el requerimiento de seguridad no es aplicable en el aplicación actual y por tanto no se lo toma en consideración para la evaluación del nivel de cumplimiento del estándar MASVS.

Debido a la naturaleza del proyecto actual, se realizó un análisis de los requerimientos aplicables en función de la arquitectura del sistema CG/Web. De esta forma se depuró el *Checklist* señalando aquellos requisitos cuya implementación no era aplicable debido a no ser compatibles con las características del sistema CG/Web. Con esta depuración se configuró el archivo para que su evaluación de la seguridad de la aplicación sea lo más precisa posible tomando en cuenta las características del proyecto.

Para el presente proyecto se siguieron las pautas expuestas en los procesos de evaluación en los requerimientos para realizar el desarrollo y posteriores pruebas de seguridad. Para gestionar el cumplimiento de las pruebas se utilizó la plantilla del *Checklist* oficial depurada y adaptada al proyecto (Ver Anexo 1). Las pruebas asociadas a la verificación de los requerimientos de seguridad se realizaron en base a las recomendaciones proporcionadas por el procedimiento de evaluación.

3.1.3.6. Planificación de fases

Se realizó una planificación de las fases de la metodología en base a los requerimientos obtenidos y los módulos propuestos para la aplicación. Se especificaron las iteraciones que se consideraron necesarias en cada una, como se muestra en la Tabla 3.

Tabla 3: Planificación de Fases

Fase	Iteración	Descripción
Exploración	Iteración 0	Propuesta del proyecto, recolección de requerimientos iniciales.
Inicialización	Iteración 0	Establecimiento del proyecto. Análisis de requerimientos iniciales. Planificación de fases Generación historias de usuarios. Día de Prueba.
Producción	Iteración 1	Implementación de funcionalidad RF1. Actualización de historias de usuario. Generación de pruebas de Aceptación.
	Iteración 2	Implementación de funcionalidad RF2. Actualización de historias de usuario. Generación de pruebas de Aceptación.
	Iteración 3	Implementación de funcionalidad RF3. Actualización de historias de usuario. Generación de pruebas de Aceptación.
	Iteración 4	Implementación de funcionalidad RF4. Actualización de historias de usuario. Generación de pruebas de Aceptación.
	Iteración 5	Implementación de funcionalidad RF5. Actualización de historias de usuario. Generación de pruebas de Aceptación.
	Iteración 6	Implementación de funcionalidad RF6. Actualización de historias de usuario. Generación de pruebas de Aceptación.
	Iteración 7	Implementación de funcionalidad RF7. Actualización de historias de usuario. Generación de pruebas de Aceptación.
	Iteración 8	Implementación de funcionalidad RF8. Actualización de historias de usuario. Generación de pruebas de Aceptación.
	Iteración 9	Implementación de funcionalidad RF9. Actualización de historias de usuario. Generación de pruebas de Aceptación.
Estabilización	Iteración 13	Refactorización funcionalidades módulo compras. Aplicación pruebas de aceptación.
	Iteración 14	Refactorización funcionalidades módulo financiero. Aplicación pruebas de aceptación.
	Iteración 15	Refactorización funcionalidades módulo nómina. Aplicación pruebas de aceptación.
	Iteración 16	Refactorización funcionalidades módulo bodegas. Aplicación pruebas de aceptación.
Pruebas	Iteración 17	Evaluación de pruebas finales.

3.2. Fase de inicialización

En esta sección se describen las actividades realizadas durante la fase de inicialización siguiendo la metodología Mobile-D. Durante esta fase se empezó con la configuración del proyecto en la cual se debe asegurar la existencia y preparación de los recursos identificados en la fase pasada. También se inició con la implementación de funcionalidades como base para la producción.

3.2.1. Configuración del proyecto

Para el cumplimiento de esta etapa fue necesario definir los recursos necesarios para el desarrollo del proyecto. Esto incluye tecnología, hardware, software y conocimiento sobre el sistema web. Debido a la arquitectura del proyecto y su compatibilidad con los recursos del sistema CG/Web se utilizaron dos soluciones en el ambiente de desarrollo.

Herramientas y plataformas software

La selección de herramientas y plataformas fue condicionada por IT del Ecuador para mantener la compatibilidad de las arquitecturas y mantenimientos usadas o aplicadas para CG/Web, dado que es un sistema propietario. Por esto se utilizó de base el sistema operativo Android, las herramientas Visual Studio, DevExpress y la plataforma Xamarin.Forms.

Para el sistema operativo (SO) se tuvo una reunión con los directivos I.T. del Ecuador en la que se consultó sobre el SO más utilizado por los clientes de la compañía. Tras una consulta a los representantes de los clientes más recientes, por ejemplo Metro de Quito y ETAPA, se determinó que este SO es *Android*. La versión para la cual se decidió desarrollar fue la versión 10, publicada en el año 2019, pues en el momento del inicio del desarrollo del proyecto (2020) era la versión estable del SO, según lo indica *Cuthbertson*, Director Senior de Gestión de Producto de *Android* [9].

El desarrollo de la aplicación se lo realizó en el Entorno de Desarrollo Integrado (IDE) Visual Studio debido a que es la herramienta adoptada por I.T. del Ecuador para el desarrollo de sus proyectos. Este IDE permite la gestión de código para proyectos de aplicaciones para el SO *Android*, con el lenguaje de programación C# [10]. Esto es importante porque facilita el mantenimiento y gestión al utilizar el lenguaje base del sistema CG/Web.

Se utilizaron dos versiones de este IDE: para el desarrollo y publicación de servicios WCF se utilizó la versión 2015 que es la versión utilizada en la empresa para el

desarrollo y mantenimiento del código fuente del sistema web CG/Web. Por otro lado, para el desarrollo de la aplicación móvil se eligió la versión 2019, pues esta cuenta con compilación para la versión de *Android* utilizada en el desarrollo.

Para el desarrollo de las interfaces de usuario se utilizó la herramienta *DevExpress*⁶, misma que es utilizada para las interfaces gráficas del sistema CG/Web. Por esto fue seleccionada la plataforma de desarrollo *Xamarin* que es una plataforma de código abierto de Windows para la construcción de aplicaciones móviles con lenguaje C y compatible con .NET [11]. Con esto se buscó modernizar las interfaces de usuario, respetando los términos y funcionalidades del CG/Web.

Solución para desarrollo backend

Para el desarrollo comprendido desde el acceso a los datos hasta su presentación mediante servicios WCF se utilizó una solución de Visual Studio 2015. Esta se enfocó en la programación de: Acceso a datos, Lógica de negocio y Servicios. La versión del IDE se utilizó por ser la herramienta de desarrollo principal de I.T del Ecuador para gestionar el código fuente de CG/Web. Para el envío y recepción de información se configuraron proyectos de servicios WCF, uno para cada módulo a desarrollar. Se complementó mediante el uso de servicios web para el manejo de notificaciones utilizando la plataforma *Firebase*. Estos servicios toman métodos especiales de una lógica de negocio configurada con los *plugins* de *Google* y *FirebaseApp* para el envío de mensajes. Estos métodos cuentan con un archivo web.config que contienen información similar a la presente en el archivo de los servicios WCF. Para la ejecución de estos servicios se utilizó un Servicio de Windows que consume el servicio web cada vez que se cumple un intervalo de tiempo. Esto permite el envío de notificaciones según la información recuperada de la base de datos del sistema CG/Web, consiguiendo notificaciones a tiempo real.

Solución para desarrollo frontend

Para el desarrollo de la aplicación móvil se hizo uso de una solución de Visual Studio 2019. Debido a que permite la compilación de aplicaciones móviles multiplataforma con la versión 10 del sistema operativo *Android*. Para cumplir con el desarrollo usando la herramienta *DevExpress* [12] se agregó el *NuGet Package*, como se ve en la Figura 9.

⁶ La herramienta *DevExpress* proporciona un paquete con controles *Xamarin.Forms* para el desarrollo de aplicaciones móviles. <https://www.devexpress.com/xamarin/>

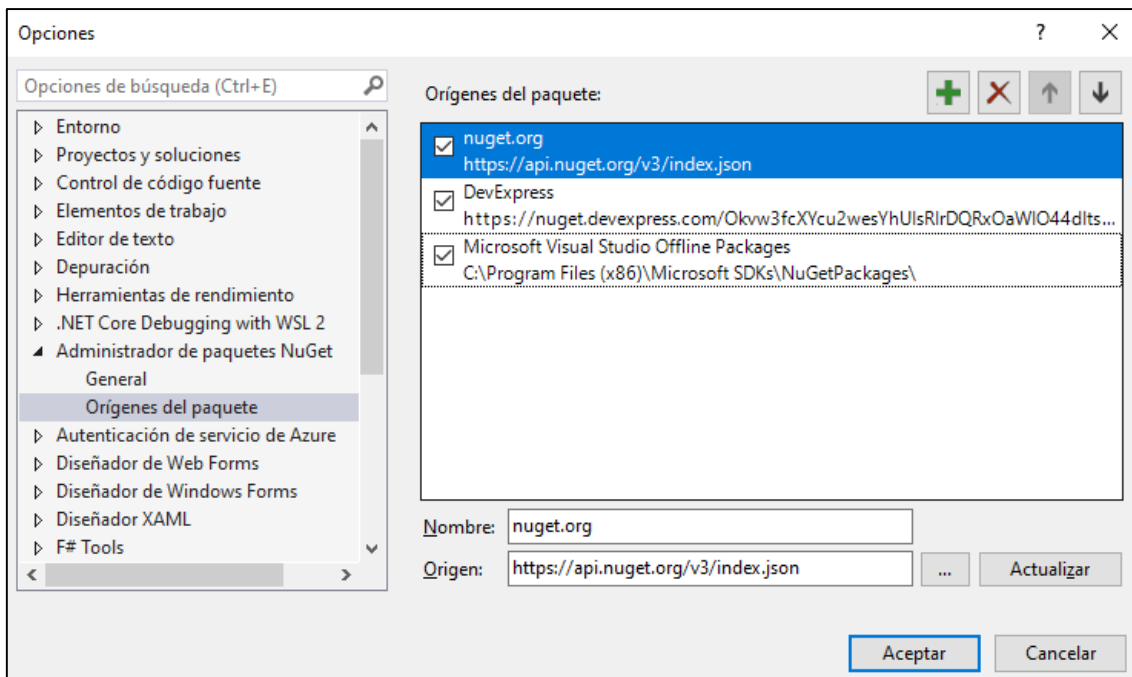


Figura 9: *DevExpress NuGet Package*

El producto resultante de este proyecto se comunica mediante internet utilizando la información presente en el archivo `App.config` de la aplicación. Este indica el módulo que el usuario utiliza, la dirección IP pública para consumir los servicios WCF y la información del sitio que tiene configurados los servicios. Esto se ve complementado con la plataforma en línea *Firebase*, que se encarga de enviar las notificaciones a los dispositivos de la capa *frontend* mediante *tokens* generados al instalar la aplicación. Para esto se agregó el *plugin* de *Firebase.PushNotification*, el cual permite la recepción de notificaciones de la aplicación. También se integró la librería `.Net`, necesaria para el manejo de *nuggets* que permitiesen la administración de elementos *JSON* (*Newtonsoft.json*), complementar el diseño de las pantallas (*Xamarin.Essentials*) y obtener información del dispositivo (*Xam.Plugin.DeviceInfo*).

Control de versiones

Para gestionar el control de cambios en el desarrollo del código de la aplicación se utilizó la herramienta *Bitbucket*⁷. Gracias a esto se cuenta con un control de versiones utilizando *git* y se tiene acceso a las historias de usuario gestionadas con la herramienta *Jira Software* (Figura 10). Con esto se puede llevar el control del estado de las historias y a su prioridad durante el desarrollo.

⁷ *Bitbucket* es una herramienta online desarrollado por la compañía *Atlassian*. Permite almacenar y gestionar el código fuente de un proyecto de desarrollo de software, incluyendo control de versiones mediante *git* y compatibilidad con *Jira Software*. <https://www.atlassian.com/es/software/bitbucket>



Figura 10: Repositorio Bitbucket e historias de usuario

Gestión de servicios

Para la publicación y posterior consumo de los servicios WCF se preparó un sitio web en el servidor de aplicaciones utilizado por el sistema CG/Web. Este sitio permite el acceso al servicio web utilizando la IP pública de la red y por tanto permite acceder utilizando la aplicación móvil mediante internet. El sitio se encuentra configurado con los siguientes datos:

Sitio CGApp

- Tipo de proyecto: ASP.NET
- Framework: Net Framework v4.0
- IIS v10.0

3.2.2. Día de prueba (iteración cero)

Como parte de la iteración 0 de la metodología Mobile-D se realizó el inicio de la implementación de una historia de usuario para identificar posibles problemas y desafíos durante la producción. Se eligió la historia de usuario ACT-10, teniendo en cuenta las historias de usuario relacionadas como se muestra en la Figura 11.



Figura 11: Tablero Kanban con historias de usuario escogidas

Para la implementación en código fuente se siguió la técnica Test-Driven Development (TDD) que consiste en desarrollar a partir de pruebas unitarias a nivel de la lógica de negocio [13], adaptada para ser aplicada a los procesos. Para gestionar estas pruebas se agregó un proyecto de pruebas unitarias de Visual Studio a la solución de servicios WCF.

Debido a la naturaleza de las historias de usuario fue necesario realizar la conexión de este proyecto con la base de datos para utilizar los datos reales durante las pruebas. Para esta conexión se utilizó el *Framework EnterpriseLibrary* que es empleado por I.T. del Ecuador para la conexión de la aplicación web. De esta forma se aseguró que el desarrollo satisfaga las necesidades del negocio manteniendo relación con la versión web original del sistema.

Sin embargo, durante la planificación y desarrollo se evidenció que la aplicación sola de TDD desembocaría en problemas de gestión de datos de entrada y en base. La ejecución continua de pruebas usando datos reales que debían ser afectados provocó que los mismos fuesen no válidos para pruebas posteriores e imperaba realizar cambios

a nivel de datos. Para solventar este problema se utilizó una práctica como complementación de TDD.

Para los datos de entrada se recurrió a la práctica de Especificación por Ejemplo (SBE por sus siglas en inglés). Esta práctica, también conocida como Desarrollo Guiado por Pruebas de Aceptación (ATDD) o Desarrollo Guiado por Ejemplos, es recomendada para proyectos donde el dominio del negocio es muy importante. Con su aplicación se listaron los posibles casos para cada escenario de las historias de usuario, con lo cual se concluyó que estos eran un mismo escenario con distintos finales.

Los datos de aceptación y entrada obtenidos se especificaron en las historias de usuario, como se puede ver en la Figura 12. Con estos se desarrollaron pruebas que describiesen la historia y comprobasen la ejecución, extracción y alteración de datos. Para ello fue necesario utilizar archivos de valores separados por comas (.csv) como fuentes de datos de las pruebas, permitiendo aplicar los ejemplos de SBE en el desarrollo como se muestra en la Figura 13.



Ejemplos del escenario

Ejemplos:

ROL_EMPLEADO	PERMISO_EM PLEADO	MODALIDAD_ SOLICITUD	ESTADO_SOL_ ORIGINAL	ESTADO_SOL_ NUEVO	ESTADO_ AUDITORI A	TIPO_A UDITORI IA
APROCOMPRAS (Aprobador de compras)	Apr (Aprobación SubGerencia)	OC (Orden de Compra)	V (Validada)	P (Aprobada)	V (Aprobar)	SOL (Solicitu d)
APROCOMPRAS (Aprobador de compras)	Aut (Aprobación Gerencia)	OC (Orden de Compra)	P (Aprobada)	D (Despachada)	P (Autorizar)	SOL (Solicitu d)

Escenario 2: Anular una solicitud
Dado que el usuario quiere **aprobar una solicitud**, pero se selecciona la opción "Anular", cuando selecciona no rehacerla y selecciona un [MOTIVO_ANULACION] entonces se comprueba que la Solicitud de Compra no se encuentre

Figura 12: Muestra de aplicación de SBE en historia ACT-10


```

ACT10_datosEntrada.csv
1 ACCION,USUARIO,TERMINAL,ROL_EMPLEADO,PERMISO_EMPLEADO,MODALIDAD_SOLICITUD,ESTADO_SOL_ORIGINAL,ESTADO_SOL_NUEVO,ESTADO_AUDITORIA,TIPO_AUDITORIA,CODIGO_I
2 A,CGIFS,TerminalSubG,APROCOMPRAS,Apr,OC,V,Z,Z,SOL,,,
3 A,CGIFS,TerminalG,APROCOMPRAS,Aut,OC,P,D,P,SOL,,,
4 Z,CGIFS,TerminalSubG,APROCOMPRAS,Apr,OC,V,Z,Z,SOL,INFINC,INFORMACIÓN INCORRECTA,
5 Z,CGIFS,TerminalSubG,APROCOMPRAS,Apr,OC,V,Z,Z,SOL,NOPROC,NO PROCEDE PROCESO DE COMPRA,
6 Z,CGIFS,TerminalSubG,APROCOMPRAS,Apr,OC,V,Z,Z,SOL,PARERR,PARTIDA PRESUPUESTARIA ERRADA,
7 Z,CGIFS,TerminalSubG,APROCOMPRAS,Apr,OC,V,Z,Z,SOL,VALSOB,VALOR SOBREPASA INFIMA CUANTÍA,
8 Z,CGIFS,TerminalG,APROCOMPRAS,Aut,OC,P,Z,Z,SOL,INFINC,INFORMACIÓN INCORRECTA,
9 Z,CGIFS,TerminalG,APROCOMPRAS,Aut,OC,P,Z,Z,SOL,NOPROC,NO PROCEDE PROCESO DE COMPRA,
10 Z,CGIFS,TerminalG,APROCOMPRAS,Aut,OC,P,Z,Z,SOL,PARERR,PARTIDA PRESUPUESTARIA ERRADA,
11 Z,CGIFS,TerminalG,APROCOMPRAS,Aut,OC,P,Z,Z,SOL,VALSOB,VALOR SOBREPASA INFIMA CUANTÍA,
12 R,CGIFS,TerminalSubG,APROCOMPRAS,Apr,OC,V,Z,Z,SOL,INFINC,INFORMACIÓN INCORRECTA,Solicitud Recreada de <NUMERO_SOLICITUD_ORIGINAL>
13 R,CGIFS,TerminalSubG,APROCOMPRAS,Apr,OC,V,Z,Z,SOL,NOPROC,NO PROCEDE PROCESO DE COMPRA,Solicitud Recreada de <NUMERO_SOLICITUD_ORIGINAL>
14 R,CGIFS,TerminalSubG,APROCOMPRAS,Apr,OC,V,Z,Z,SOL,PARERR,PARTIDA PRESUPUESTARIA ERRADA,Solicitud Recreada de <NUMERO_SOLICITUD_ORIGINAL>
15 R,CGIFS,TerminalSubG,APROCOMPRAS,Apr,OC,V,Z,Z,SOL,VALSOB,VALOR SOBREPASA INFIMA CUANTÍA,Solicitud Recreada de <NUMERO_SOLICITUD_ORIGINAL>
16 R,CGIFS,TerminalG,APROCOMPRAS,Aut,OC,P,Z,Z,SOL,INFINC,INFORMACIÓN INCORRECTA,Solicitud Recreada de <NUMERO_SOLICITUD_ORIGINAL>
17 R,CGIFS,TerminalG,APROCOMPRAS,Aut,OC,P,Z,Z,SOL,NOPROC,NO PROCEDE PROCESO DE COMPRA,Solicitud Recreada de <NUMERO_SOLICITUD_ORIGINAL>
18 R,CGIFS,TerminalG,APROCOMPRAS,Aut,OC,P,Z,Z,SOL,PARERR,PARTIDA PRESUPUESTARIA ERRADA,Solicitud Recreada de <NUMERO_SOLICITUD_ORIGINAL>
19 R,CGIFS,TerminalG,APROCOMPRAS,Aut,OC,P,Z,Z,SOL,VALSOB,VALOR SOBREPASA INFIMA CUANTÍA,Solicitud Recreada de <NUMERO_SOLICITUD_ORIGINAL>

```

Figura 13: Archivo CSV de datos de entrada de historia ACT-10

Usando los datos como base se creó un archivo para la historia elegida y se crearon pruebas para sus escenarios. Al aplicar estos valores se evidenció que la base de datos debía tener una gestión que permitiese la aplicación automatizada de las pruebas sin intervención del usuario. Para esto se hizo uso de los métodos de ClassInitialize y TestCleanup descritos en la Sección 2.1.1 como se muestran en la Figura 14.

```

[ClassInitialize] Primer método en ejecutarse
0 referencias | CarlosAguirreIT, Hace 20 días | 1 autor, 1 cambio
public static void copiarEstadoOriginalTablaAfectada(TestContext context)
{
    dsOriginalCompras = new AutomatizacionPruebas().getSolicitudCompras();
    dsOriginalAuditoria = new AuditoriaGeneralManejador().listarAuditoriaGeneral();
}

[TestCleanup] Método que se ejecuta después de cada iteración
0 referencias | CarlosAguirreIT, Hace 20 días | 1 autor, 1 cambio
public void devolverTablaAfectadaAEstadoOriginal()
{
    SolicitudDST dsNuevoCompras = new AutomatizacionPruebas().getSolicitudCompras();

    AuditoriaGeneralDST dsNuevoAuditoria = new AuditoriaGeneralManejador().listarAuditoriaGeneral();

    new AutomatizacionPruebas().retornarEstadoOriginal(dsOriginalCompras, dsNuevoCompras);
    new AutomatizacionPruebas().retornarEstadoOriginal(dsOriginalAuditoria, dsNuevoAuditoria);
}

```

Figura 14: Métodos para automatización de pruebas

En el método ClassInitialize se creó una copia de las tablas que serían afectadas al ejecutar la prueba. Por otro lado, en TestCleanup se realizó el retorno de la tabla a su estado original usando la copia creada. Se utilizó la ejecución de pruebas para comprobar la correcta implementación de la técnica TDD como se ve en la Figura 15. La ejecución de las pruebas automatizadas cumple con el algoritmo general en el cual se describen los posibles resultados al ejecutar una prueba (Ver Anexo 3).

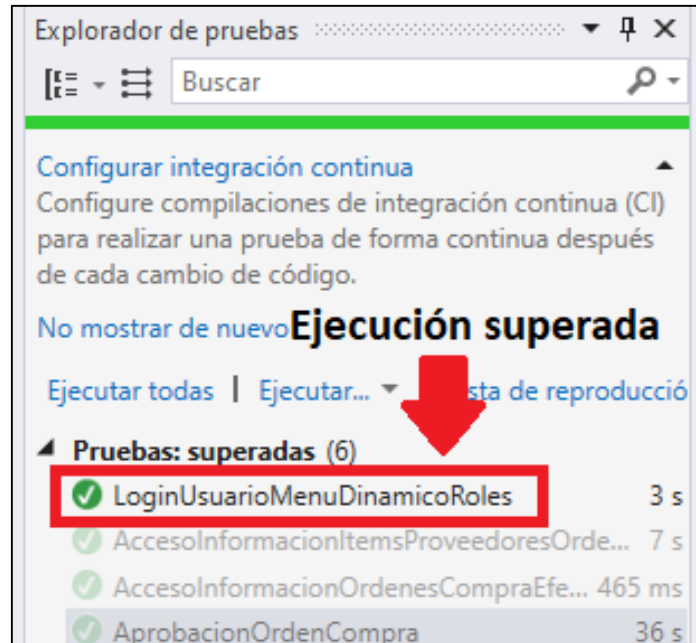


Figura 15: Ejecución pruebas unitarias

Una vez superadas las pruebas se implementó el servicio WCF para el módulo al que la historia de usuario pertenece y se publicó en el sitio antes configurado, como se muestra en la Figura 16. Tras comprobar que se reciben datos mediante el servicio se empezó la codificación de la aplicación móvil.

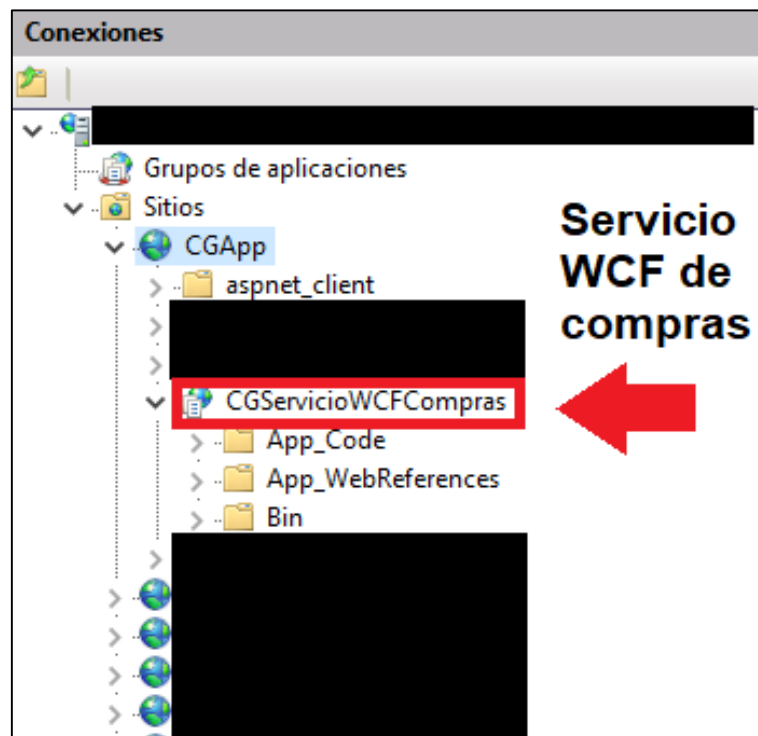


Figura 16: Publicación Servicio Compras

3.3. Fases de producción, estabilización y pruebas

En esta sección se describe el desarrollo de la aplicación móvil en base a las historias de usuario, su estabilización y correcciones tras su revisión. Para cada historia se aplicó el proceso indicado por la metodología Mobile-D, explicado en la Sección 2.1. Estas se han agrupado en base al tipo de funcionalidad con el fin de representar su evolución general.

Se siguió la prioridad asignada a cada historia y a los requerimientos para definir el orden de desarrollo. Se empezó por las historias del módulo de compras referentes a la aprobación de solicitudes y órdenes. Después con el desarrollo de las notificaciones y por último se trabajó en las consultas de información resumida. Se han detallado las historias referentes al módulo de compras como ejemplo del desarrollo de la aplicación; debido a que se manejó un proceso similar en la implementación de cada módulo.

Para complementar el desarrollo de estas fases y asegurar la calidad de la seguridad del proyecto, se siguió como guía el Estándar de Verificación de Seguridad de Aplicación Móvil Versión 1.3 (MASVS por sus siglas en inglés) [6] de OWASP⁸. Se aplicaron los controles compatibles con la aplicación móvil, la arquitectura y controles propios del sistema CG/Web y la aceptación de los *stakeholders* para el presente proyecto.

3.3.1. Funcionalidad de aprobaciones

En esta sección se describe el desarrollo e implementación de las historias de usuario que describen el proceso de aprobación de una solicitud u orden. Cada módulo descrito en la Sección 3.1.3.2 cuenta con al menos un proceso de aprobación de solicitud propio.

Uno de los principales retos de este proyecto fue conseguir migrar la información mostrada en formularios web a un entorno móvil. Para esto se realizó un análisis usando de base las pantallas del sistema CG/Web y las descripciones de cada historia de usuario. A continuación, se describe el proceso que fue utilizado para el desarrollo.

Cada historia de usuario fue utilizada de base para diseñar una prueba unitaria. La cual fue creada en una clase que hereda métodos de evaluación y ejecución. En las clases de prueba se incluye los métodos descritos en la Sección 3.2.2 y un método de prueba que describe el proceso especificado en la historia de usuario respectiva. En la Figura

⁸ Siglas en inglés de Proyecto abierto de seguridad de aplicaciones web

17 se visualiza una muestra de método de prueba en el que se señala las secciones que lo componen y que se corresponden a las secciones de la historia.

```

[TestMethod]
public void AprobacionSolicitudOrdenCompras() //ACT10 es código de la Historia de Usuario
{
    //Dado
    var usuario = creacionUsuario(TestContext);
    cuandoSeCreaUsuario_verificarAsignacionPermisosCorrespondientes(usuario, TestContext);
    List<SolicitudCompra> buzon = getBuzonUsuario(usuario);
    dependiendoTipoUsuario_cuandoConsultaBuzonSolicitudes_verificarEstadoCorrespondiente(usuario, buzon);
    SolicitudCompra solicitud_buzon = desplegarDatosSolicitud(buzon[0], TestContext);
    cuandoSolicitudDesplegada_verificarDatosObligatoriosNoNulos(solicitud_buzon);
    Terminal terminalUsuario = new Terminal(TestContext.DataRow["TERMINAL"].ToString());

    //Cuando
    MotivoAnulacion motivoAnulacion = new MotivoAnulacion(TestContext.DataRow["CODIGO_MOTIVO_ANULA"].ToString());
    cuandoSeleccionaMotivoAnulacion_verificarDescripcionMotivoEsperado(motivoAnulacion, TestContext);

    //Entonces
    new ComprasManejador().realizarAprobacionSolicitud(TestContext.DataRow["ACCION"].ToString(), usuario, solicitud_buzon, terminalUsuario, motivoAnulacion);
    dependiendoAccion_cuandoFinalizaProcesoAprobacionSolicitud_compararEstadoNuevoMotivoAnulacionCorrectos(solicitud_buzon, TestContext, dsOriginalCompras);
    dependiendoAccion_cuandoFinalizaProcesoAprobacion_verificarDatosAuditoriaNoNulos(solicitud_buzon, usuario, TestContext);
    dependiendoAccion_cuandoFinalizaProcesoAprobacionSolicitud_verificarDatosAuditoriaGeneralNoNulos(solicitud_buzon, usuario, TestContext, dsOriginalAuditoria);
}
    
```

Método de prueba

Sección de condiciones de prueba

Sección de acción del usuario

Sección ejecución y evaluación del resultado

Figura 17: Muestra de método de prueba

A nivel de lógica de negocio se utilizó una clase que actúa como manejador del módulo respectivo. En esta se definieron los métodos que permiten completar los procesos definidos en la historia de usuario. Se utilizaron clases de las entidades que participan en dichos procesos que sirven para convertir los *datasets* utilizados por el sistema CG/Web para su manejo en la aplicación móvil. Finalmente, se configuró el servicio WCF del módulo de compras para consumir los métodos de la clase manejador. En la Figura 18 se evidencia una muestra de esta configuración.

```

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json, UriTemplate = "getSolicitudOrdenCompraClase?modalidad={modalidad}&compania={compania}&numero={numero}")]
public SolicitudCompra getSolicitudOrdenCompraClase(string modalidad, string compania, decimal numero)
{
    SolicitudCompra solicitudOrdenCompra = new SolicitudCompra(modalidad, compania, numero);

    return solicitudOrdenCompra;
}
    
```

Configuración para consumo por URI

Figura 18: Muestra de métodos del servicio WCF de compras

Para las historias de usuario que tienen un equivalente funcional en el sistema CG/Web se utilizó la pantalla web asociada para su análisis y el diseño de la versión móvil. En base a esto se diseñaron *mockups* que permitiesen ejemplificar el formato de forma general. En la Figura 19 se puede ver como muestra de las pantallas web el buzón de solicitudes pendientes de aprobación, mientras que en la Figura 20 se evidencia el *mockup* obtenido tras su análisis.

Modalidad: Orden de Compra		APROBACION SOLICITUD SUBGERENCIA							
Compañía	Nro. Solicitud	F. Solicitud	Solicitante	Area Solicitante	Resp. Recepcion	Estado	Editar	Anular	Acciones
AGUA POTABLE	55	2020/05/22	CGWEB ADMINISTRADOR		WILSON MACIAS	VALIDADA			
AGUA POTABLE	58	2020/05/22	CGWEB ADMINISTRADOR		MARIO IVAN CÁRDENAS TORRES	VALIDADA			
AGUA POTABLE	59	2020/05/22	CGWEB ADMINISTRADOR		MARIO IVAN CÁRDENAS TORRES	VALIDADA			
AGUA POTABLE	124	2020/10/07	CGWEB ADMINISTRADOR	SUBGERENCIA DE MERCADERO	GERARDO MACIAS	VALIDADA			
AGUA POTABLE	125	2020/10/08	CGWEB ADMINISTRADOR	SUBGERENCIA DE MERCADERO	DR. MANUEL CAJIAO VELARDE	VALIDADA			

Figura 19: Muestra de pantalla del sistema CG/Web: Buzón de solicitudes

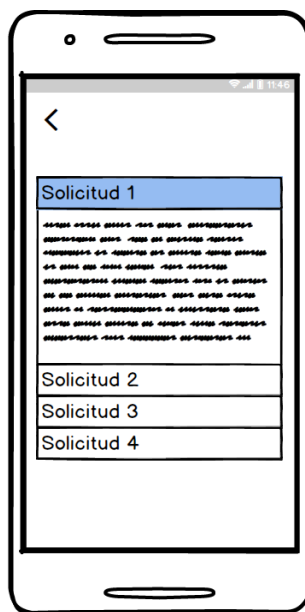


Figura 20: Mockup de pantalla: Buzón de solicitudes

Fue necesario instalar paquetes *NuGets* para el manejo de elementos de la Interfaz de Usuario y de los datos obtenidos del servicio, como se ve en la Figura 21. Entre los paquetes se incluyó *Xamarin.Forms* para los controladores básicos de las pantallas móviles, *Newtonsoft.Json* para manejar la información obtenida desde los servicios usando clases inicializadas desde un *JSON*. De la misma manera se incluyeron los paquetes *XamarinForms* de *DevExpress* para el manejo de controles específicos relacionados con la versión web.

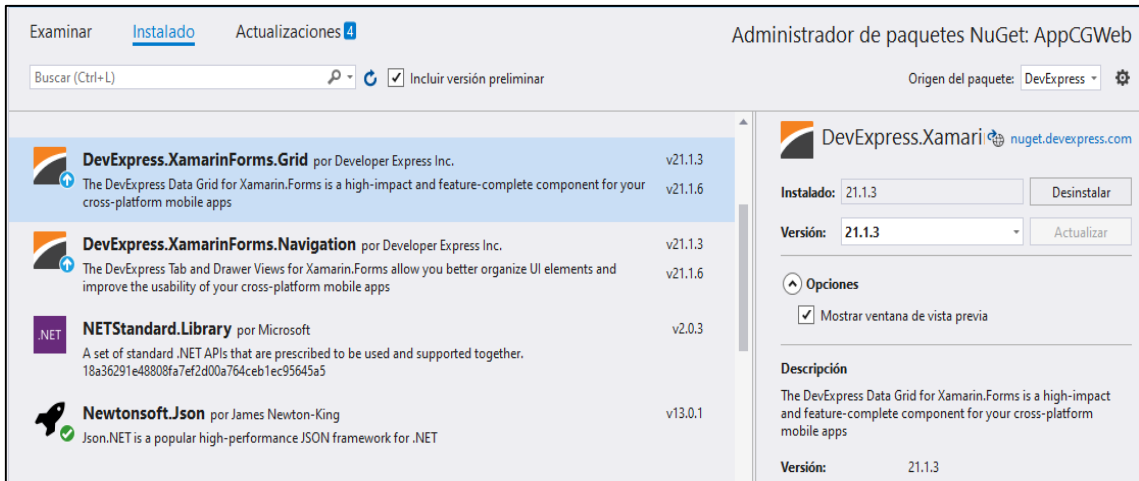


Figura 21: Instalación Paquetes NuGets

Para mejorar el diseño y su adaptación al ambiente móvil se incluyeron íconos que complementasen los datos mostrados. Para esto se implementó el *framework* *FontAwesome* al proyecto. Se descargó de su página oficial los archivos del paquete para desarrollo local. Se agregaron los archivos .otf al proyecto *Android* en la carpeta *Assets*, como se indica en el tutorial “*Use FontAwesome in a Xamarin.Forms app*” publicado por Sebastian Jensen [14] y se evidencia en la Figura 22.



Figura 22: Archivos para implementar *FontAwesome*

Complementando su tutorial, se requiere configurar las propiedades de los tres archivos indicando que, durante la compilación, se los trate como recursos incrustados, como se ve en la Figura 23. Esto es necesario para poder llamarlos mediante la propiedad “*FontFamily*” de los elementos compatibles en pantalla, como por ejemplo los *labels*.

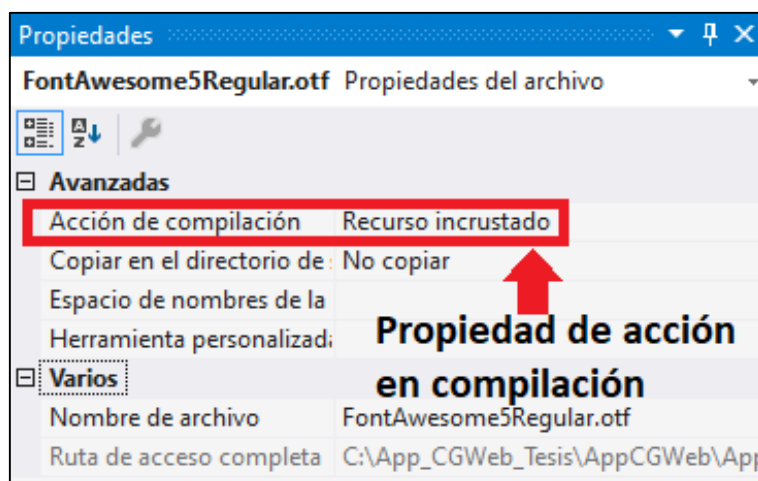


Figura 23: Configurar propiedades de archivos de FontAwesome

Por último, con el fin de facilitar la identificación de los iconos de FontAwesome se integró la clase FontAwesomeIcons.cs del usuario Matthew Robbins en su repositorio de GitHub **fa2cs**⁹. De esta forma se pudieron utilizar los íconos de esta librería mediante su nombre y no su código (Figura 24).

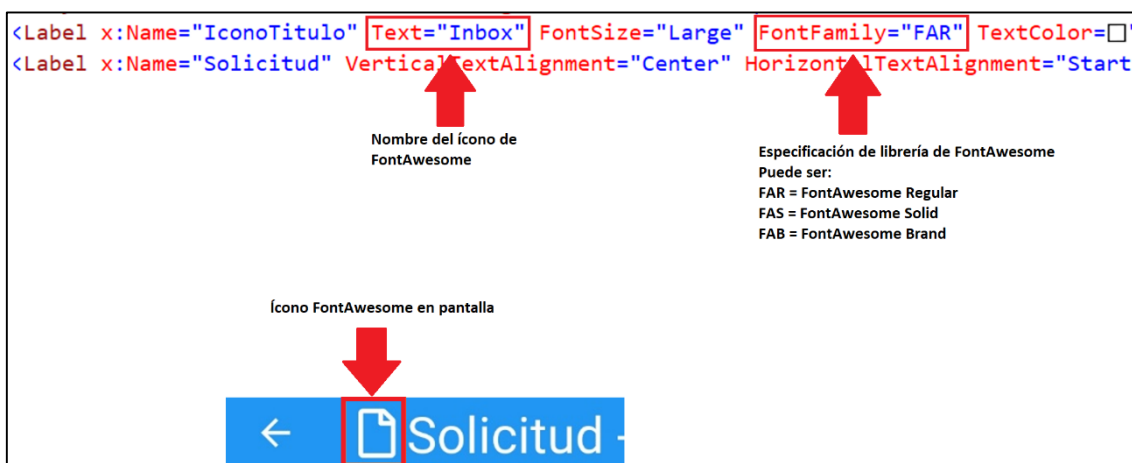


Figura 24: Uso de iconos FontAwesome por el nombre

A continuación, se detallarán las pantallas resultantes del proceso seguido en cada historia de usuario. Se incluyen los desafíos y soluciones utilizadas en cada historia y la relación que guardan entre sí.

⁹ Repositorio de Github del usuario Matthew Robbins que indica cómo consumir la librería FontAwesome mediante los nombres de los íconos. <https://github.com/matthewrdev/fa2cs>

3.3.1.1. ACT-10: Aprobación de solicitud de orden de compra

La historia de usuario ACT-10 describe la aprobación de una solicitud de orden de compra. Este proceso abarca la selección de una solicitud desde un buzón asignado al usuario, la revisión de su información y su aprobación.

Para el buzón de solicitudes se completó la prueba descrita en la Sección 3.2.2 con el método `AprobacionSolicitudOrdenCompras()`, el cual describe el proceso desde que el usuario ingresa al buzón de solicitudes de compras hasta que aprueba la solicitud. En la primera pantalla se muestran las solicitudes pendientes de aprobación asignadas a un usuario; se definió que debía tener las siguientes características.

- Se muestran las solicitudes a las cuáles el usuario tiene permisos de acceso.
- Muestra información general de las solicitudes.
- Permite acceder al Formulario de Aprobación de una solución.
- Permite regresar a la pantalla anterior.

Como versión inicial se implementó una pantalla con un *datagrid* que despliega los datos obtenidos mediante el consumo del servicio WCF de compras configurado como se indica en la Sección 3.2.1. Esta sirvió para demostrar la funcionalidad de la pantalla y la correcta ejecución de los métodos del servicio. En esta se accedía a la revisión de la solicitud seleccionando la fila respectiva. Sin embargo, como se ve en la Figura 25.a, la información no se muestra en su totalidad y requiere que el usuario se desplace o gire el dispositivo en sentido horizontal como se indica en la Figura 25.b

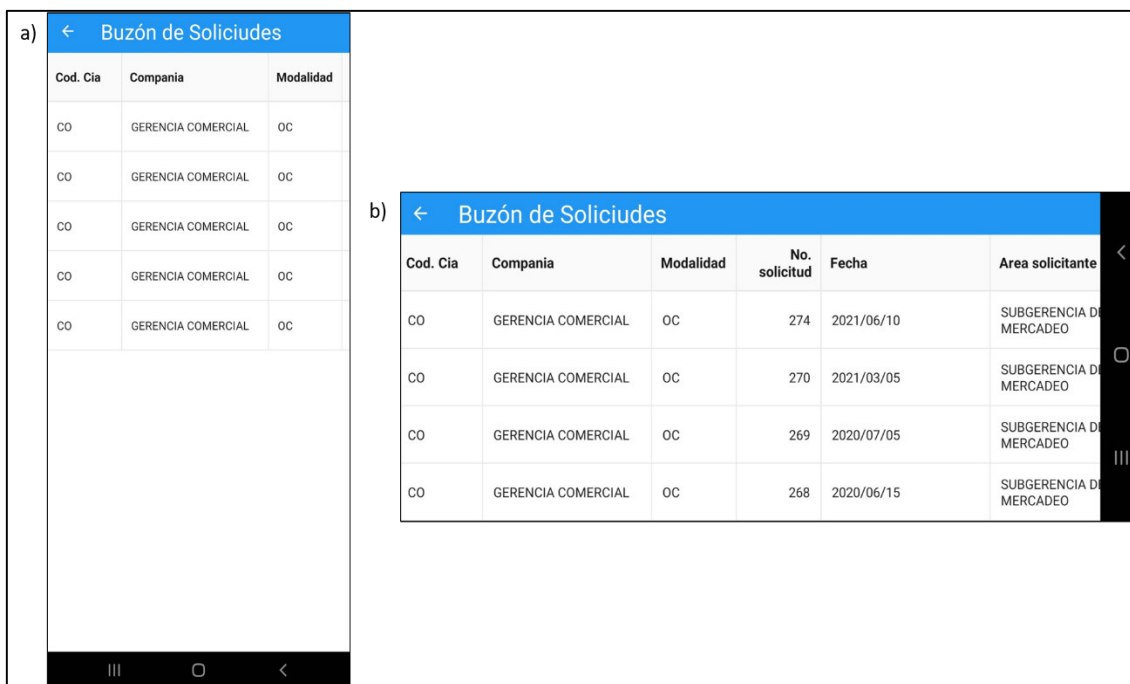


Figura 25: Pantalla buzón de solicitudes versión inicial

Se modificó la pantalla en base al *mockup* indicado en la Sección 3.3.1 para que presente la información completa de la solicitud sin necesidad de acciones extras del usuario. Esta versión aprovechó la modificación del *datagrid* para desplegar filas y columnas de forma personalizada utilizando una cuadrícula predefinida como contenedor (Figura 26).

Con toda la información disponible en una sola pantalla se identificaron datos que no eran relevantes para la identificación de las solicitudes mostradas, aparecían de forma redundante o se volvían innecesarios debido a que eran mostrados en el formulario de aprobación. Por lo expuesto se modificó nuevamente el diseño de la pantalla eliminando los datos que no aportaban valor a la aplicación.

Además, se cambió el *datagrid* por un *collectionview* para mejorar la distribución de los elementos y facilitar la integración de íconos. De esta forma se consiguió implementar que el acceso a la revisión de las solicitudes sea mediante una acción *swipe*. Esto fue escogido como control para evitar que el usuario acceda a la revisión de una solicitud por error. Esta pantalla y los controles de Xamarin.Forms utilizados se muestran en la Figura 27.



Figura 26: Pantalla buzón de solicitudes versión media

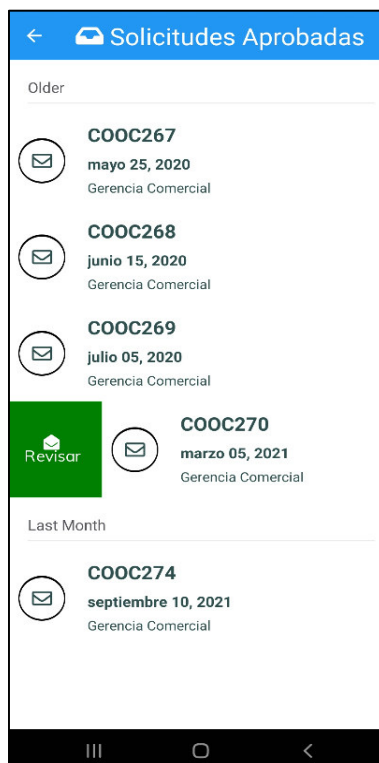


Figura 27: Pantalla buzón de solicitudes versión final

Una vez el usuario selecciona una solicitud pendiente de aprobación se despliega la pantalla con los datos de la misma. Se utilizó de base la pantalla de aprobación de solicitud del sistema CG/Web (Figura 28) para determinar la información, diseño y acciones que el usuario podrá realizar en la versión de la aplicación móvil.

Figura 28: Pantalla web de aprobación de solicitud

Se determinó con el *product owner* que se requerían dos pantallas para la versión móvil. Se graficó el *mockup* de la Figura 29 como diseño guía para la pantalla de aprobación; esta pantalla:

- Muestra un formulario que permite realizar la aprobación de una solicitud/orden.
- Permite regresar a la pantalla anterior.
- Permite revisar un justificativo y observación antes de anular o aprobar la solicitud/orden.
- Permite seleccionar el botón “Ver Detalle” permite pasar a una pantalla de Detalle de Solicitud/Orden.

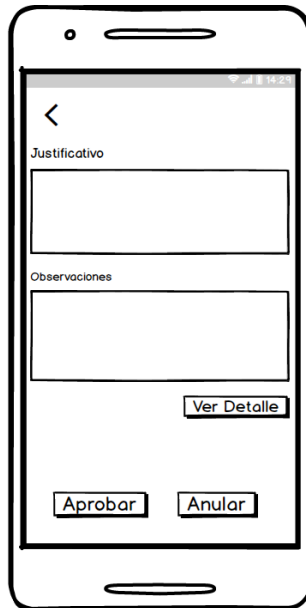


Figura 29: Mockup aprobación de solicitud de compra

Se graficó el *mockup* de la Figura 30 como diseño guía para la pantalla de aprobación; en esta pantalla:

- Se muestra información detallada de la solicitud/orden referente a ítems y presupuesto.
- El botón con el ícono de flecha apuntando a la izquierda permite regresar a la pantalla anterior

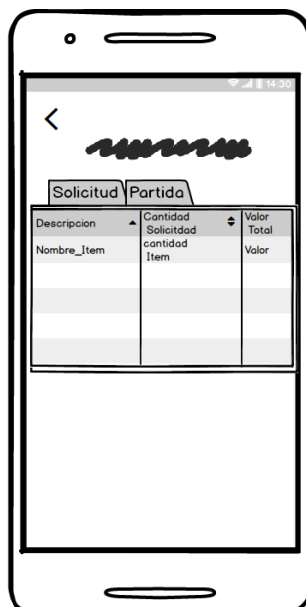


Figura 30: Mockup pantalla de detalle de solicitud

Como versión inicial se implementó una pantalla con un conjunto de controles *MultiLineEdit* para desplegar los datos obtenidos mediante el consumo del servicio WCF de compras (Figura 31.a). Al pulsar el botón “Detalles” se muestra la información de ítems y partidas presupuestarias de la solicitud usando pestañas y *datagrids* (Figura 31.b).

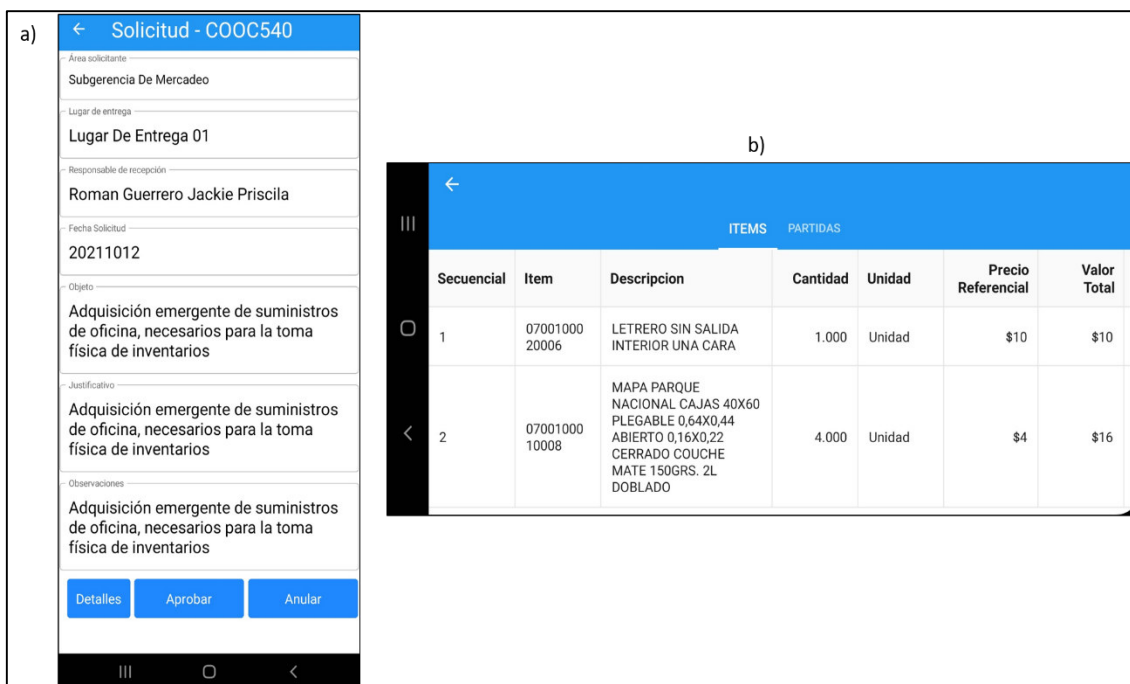


Figura 31: Pantalla de aprobación y detalle de solicitud versión inicial

En esta versión se evidenció que la información mostrada requería de un apoyo visual para mejorar su comprensión. También se identificó que la información de partidas presupuestarias mostradas era redundante al mostrar los mismos datos a excepción del monto por ítem. Por estos motivos se rediseñaron ambas pantallas. En la pantalla de aprobación (Figura 32.a) se agregaron íconos desde imágenes con formato png, esto debido a que el control *multiLineEdit* no es compatible con la librería *FontAwesome*. En la pantalla de detalles (Figura 32.b) se agruparon a nivel de lógica los registros de las partidas presupuestarias para mostrar una sola fila con el total sumado de monto utilizado.

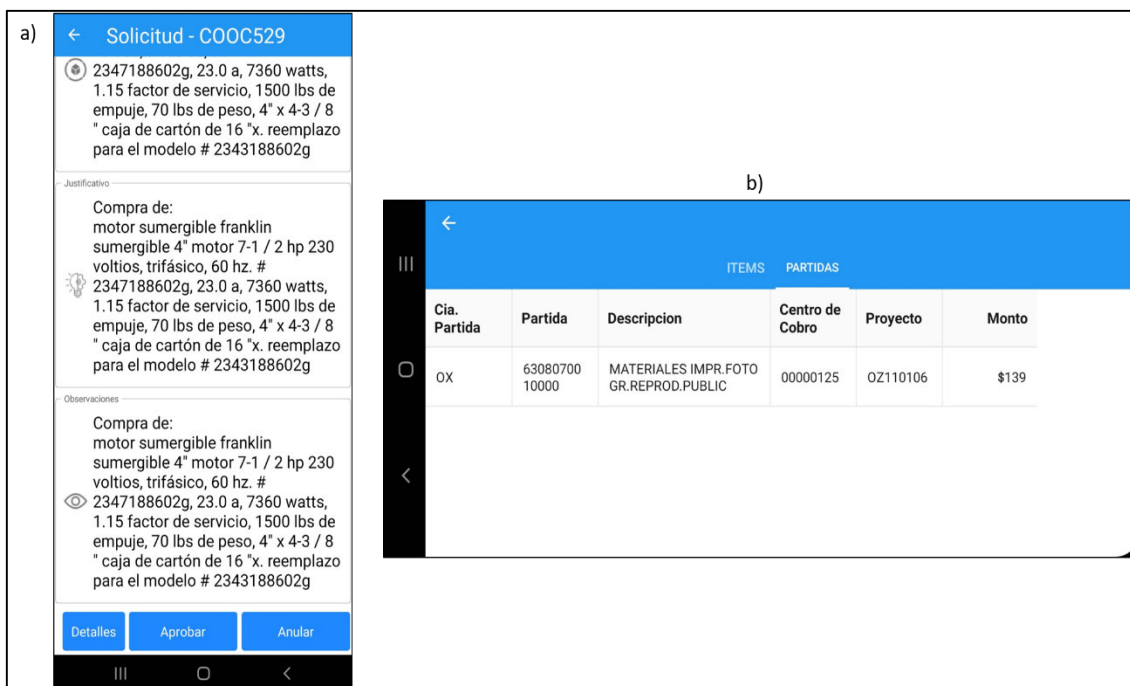


Figura 32: Pantalla de aprobación y detalle de solicitud versión 2

Con estas nuevas versiones se identificó un inconveniente con el diseño, pues el usuario debe revisar la información de ítems y partidas para decidir si aprueba o anula la solicitud. Sin embargo, al estar los datos en una pantalla distinta, existía la posibilidad que no sean tomados en cuenta durante la revisión. Además, si bien los íconos ayudaban en la comprensión, los nombres de los datos no eran visibles por el tamaño de fuente y el estar rodeados de bordes.

Se rediseñó una vez más la pantalla, buscando que la información de detalle sea incluida en la pantalla de aprobación. También se decidió cambiar los controles *multiLineEdit* por un diseño usando *grids* individuales para cada dato. Estos servirían de contenedor para un conjunto de *labels* que desplegasen la información de forma más clara y permitiesen aplicar la librería de *FontAwesome* para los íconos.

Para complementar el diseño se realizó una encuesta (Ver Anexo 4) entre los clientes y stakeholders exponiendo diseños preliminares. Los resultados (Ver Anexo 5) permitieron diseñar un nuevo formato que cumpliera con las demandas expuestas por los usuarios. Se buscaba un diseño que permitiera diferenciar claramente el inicio y final de una sección de datos, pero que no ocupe el protagonismo de la pantalla por su color o forma.

Por otro lado, al integrar la información de detalle se evidenció que muchos de los datos de los ítems no eran relevantes para la aprobación. Por esto, con asesoría de los

stakeholder, se decidió agrupar los ítems en función de los datos que sí eran importantes para decidir la aprobación. Además, se aplicó el formato de la segunda versión de la pantalla de buzón de solicitudes (Figura 26) para desplegar la información de ítems y partidas.

En la Figura 33.a se muestra el diseño final, los datos de la solicitud se despliegan diferenciándose por el resalte de color del título. Por su parte, los ítems del detalle se agruparon por su código de CPC, la sumatoria de estos valores y el valor total utilizado en los ítems de dicho grupo. El valor de ínfima cuantía, al ser un dato general, se despliega una única vez junto al título de la sección de ítems (Figura 33.b).

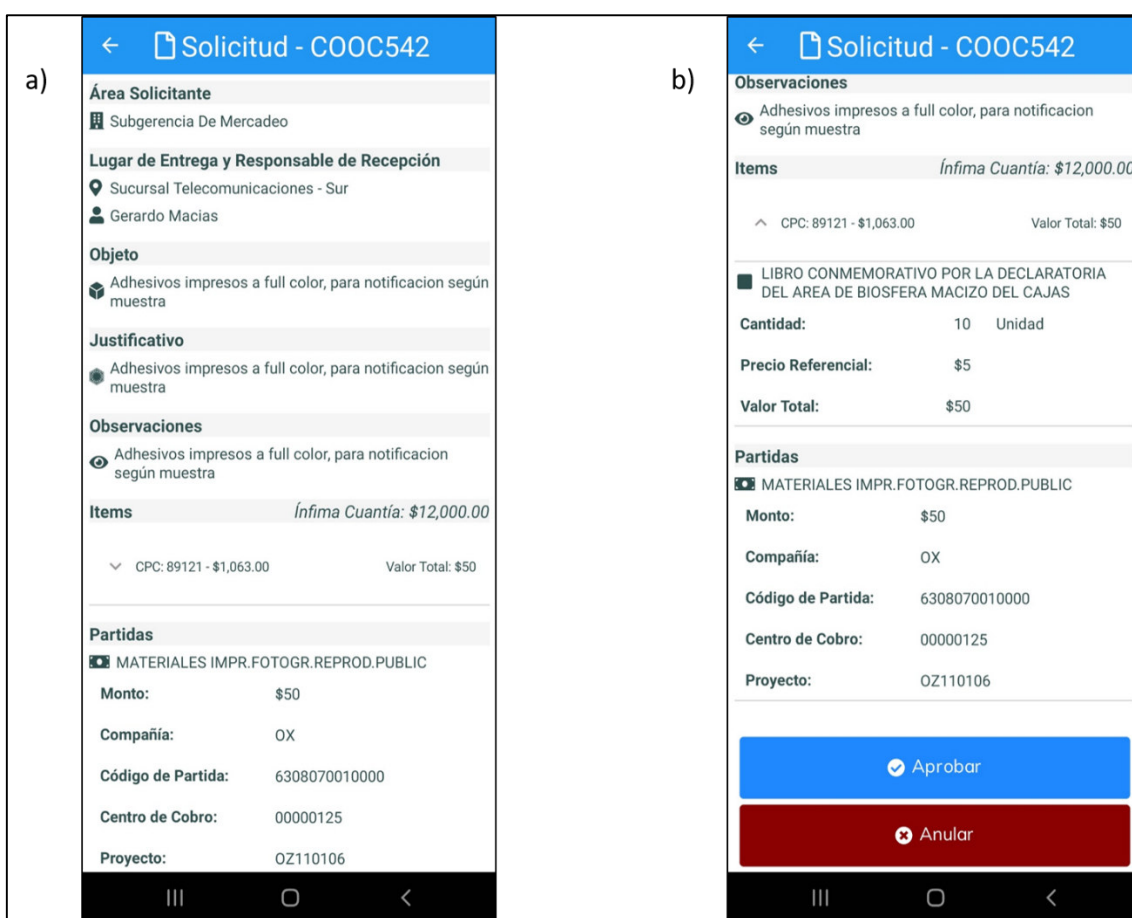


Figura 33: Pantalla de aprobación de solicitud de compra versión final

3.3.1.2. ACT-9: Aprobación de solicitud de orden de trabajo/prestación de servicios

Para esta historia de usuario se siguió el proceso descrito en la Sección 3.3.1. El proceso y datos son muy similares al descrito en la historia de usuario ACT-10 (sección 3.3.1.1). Sin embargo se agregan cinco datos adicionales a los desplegados en pantalla y se

altera el proceso en sus validaciones. A nivel de lógica se agregó una entidad que hereda de la solicitud de orden de compra y al obtener los datos se agregaron los siguientes:

- Términos de Pago
- Condiciones de Multa
- Administrador
- Tiempo de Entrega (Días)
- Miembro de Comisión

```
19 referencias | CarlosAguirreT, Hace 1 día | 1 autor, 2 cambios
public class SolicitudCompraOrdenTrabajo_PrestacionServicios : SolicitudCompra
{
    6 referencias | CarlosAguirreT, Hace 14 días | 1 autor, 1 cambio
    public string CodTerminosPago { get; set; }
    3 referencias | CarlosAguirreT, Hace 14 días | 1 autor, 1 cambio
    public string TerminosPago { get; set; }
    6 referencias | CarlosAguirreT, Hace 14 días | 1 autor, 1 cambio
    public string CodCondicionMulta { get; set; }
    3 referencias | CarlosAguirreT, Hace 14 días | 1 autor, 1 cambio
    public string CondicionMulta { get; set; }
    6 referencias | CarlosAguirreT, Hace 14 días | 1 autor, 1 cambio
    public string CodAdministrador { get; set; }
    3 referencias | CarlosAguirreT, Hace 14 días | 1 autor, 1 cambio
    public string Administrador { get; set; }
    6 referencias | CarlosAguirreT, Hace 14 días | 1 autor, 1 cambio
    public string CodMiembrosComision { get; set; }
    3 referencias | CarlosAguirreT, Hace 14 días | 1 autor, 1 cambio
    public string MiembrosComision { get; set; }
    6 referencias | CarlosAguirreT, Hace 14 días | 1 autor, 1 cambio
    public decimal TiempoEntrega { get; set; }
}
```

Herencia de solicitud de compra

Nuevos campos

Figura 34: Entidad que hereda de solicitud de compra

Se reutilizó el desarrollo de las pantallas descritas en la sección 3.3.1.1. La mayor diferencia corresponde al formulario de aprobación de solicitud. Fue agregado un *stacklayout* que contuviese los datos adicionales propios de las modalidades de prestación de servicios y órdenes de compras. De esta forma sólo se despliega esta nueva sección si la modalidad de la solicitud coincide con las dos mencionadas.



Figura 35: Campos adicionales agregados a solicitudes OT y PS

3.3.1.3. ACT-11: Aprobación de órdenes de compra

La historia de usuario ACT-8 describe el proceso para la aprobación de una orden de compra. El proceso y diseño es muy similar al utilizado en las Secciones 3.3.1.1 y 3.3.1.2. El mayor cambio se muestra en los datos que se despliegan en pantalla. La orden de compra cuenta con información del proveedor, el contacto con este y sus instrucciones especiales. Además, la información de ítems a este nivel no tiene datos sobre el código de Cuentas Por Cobrar (CPC) ni ínfima cuantía, por tanto en pantalla no se muestran agrupados.

Como se puede apreciar en la Figura 36, el buzón de órdenes es muy similar al buzón de solicitudes, con la diferencia de que presenta el código de la solicitud que originó la orden. En el formulario de aprobación se presentan cambios en los datos mostrados, incluyendo ahora información del proveedor asignado a la orden de compra seleccionada (Figura 37.a).

En la sección de ítems se muestra la información completa sin agrupación (Figura 37.b) pues el usuario debe revisar cada apartado para aprobar la orden. Además, en la sección de partidas presupuestarias, se incluye el ítem al cual afecta el valor de IVA y el monto utilizado de la partida.



Figura 36: Buzón de órdenes de compra

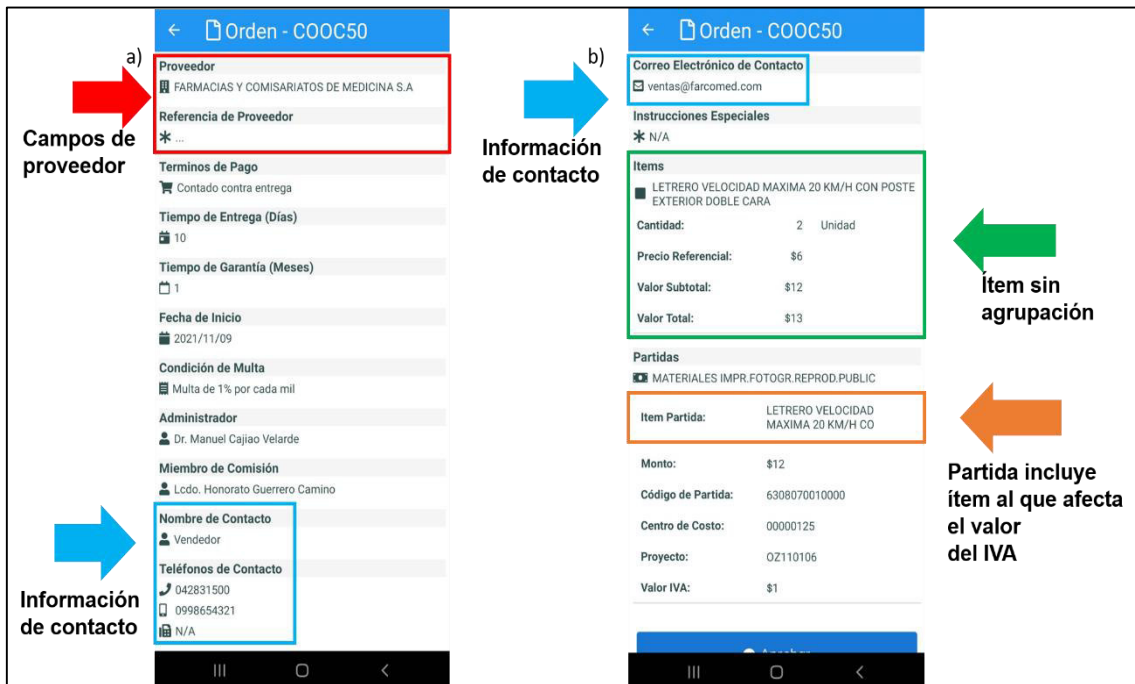


Figura 37: Pantalla de aprobación de orden de compra

3.3.1.4. ACT-1: Aprobación de solicitud de vacaciones

Esta historia de usuario describe el proceso de aprobación de una solicitud de vacaciones, de la cual se visualiza una muestra en la Figura 38. Durante el desarrollo del presente proyecto, varias funcionalidades del sistema CG/Web fueron modificadas, incluyendo la descrita en esta historia. La aprobación de una solicitud de vacaciones fue integrada en un nuevo módulo de BPM del sistema web, pero conservando su funcionalidad anterior en el módulo de nómina.

ACT-1

- Usuario
- Fecha
- Hora
- Nombre del terminal
- <ESTADO_SOLICITUD>
- Número

Ejemplos:

ROL_USUARIO	OPCIÓN_APROBAR	ID_PAS O	ESTADO_ORIGINAL_SOL	ESTADO_NUEVO_SOL
NOMINA	Verificar Solicitud	2	S	R
NOMINA	Aprobar Solicitud	3	R	A

Consejo de expertos: pulsa **M** para comentar

Figura 38: Muestra de historia ACT-1 original

En la Figura 39 se aprecia una muestra del nuevo proceso, que incluye la creación de una nueva entidad: “Solicitud de acción de personal”. Tras el cambio, la aprobación de la solicitud de vacación debe incluir la actualización de esta nueva entidad con información como el estado, número de paso, nombre de tarea, entre otros. Debido a esto fue necesario cambiar el consumo de los recursos legados del sistema web original.

ACT-1

y el usuario al ingresar una observación y seleccionar la opción <OPCIÓN_APROBAR>, entonces a la solicitud se le asigna el <ESTADO_NUEVO_SOLNOM>, <ESTADO_NUEVO_SOLAP> y se registran datos de auditoría:

- Usuario
- Fecha
- Hora
- Nombre del terminal
- <ESTADO_SOLICITUD>
- Número

Ejemplos:

ROL_USUARIO	OPCIÓN_APROBAR	ID_PASO	ESTADO_ORIGINAL_SOLNOM	ESTADO_ORIGINAL_SOLAP	ESTADO_NUEVO_SOLNOM	ESTADO_NUEVO_SOLAP
NOMINA	Verificar Solicitud	2	S	P	R	C
NOMINA	Aprobar Solicitud	3	R	P	A	C

Campos agregados y actualizados por el cambio del proceso

Se incluye un historial de las versiones corregidas de la historia

Versiones corregidas: Como usuario gerencial del...

Figura 39: Muestra de historia ACT-1 modificada

Para permitir la conexión con el módulo de nómina, se decidió realizar el consumo del servicio web publicado en la intranet del cliente desde la lógica de negocio (Figura 40). De esta forma se asegura que el usuario que ingresa a la aplicación tenga permisos en el módulo de BPM y el módulo de nómina. Además, permite controlar que los registros generados a partir de la aprobación sean almacenados en las bases de datos respectivas de cada módulo.

```
namespace Nomina.LogicaNegocio
{
    4 referencias | CarlosAguirreIT, Hace 10 días | 1 autor, 1 cambio
    public class NominaManejador
    {
        CGServicioNomina.Service proxyNomina = new CGServicioNomina.Service();
        2 referencias | CarlosAguirreIT, Hace 20 días | 1 autor, 1 cambio
        public NominaManejador()
        {
        }
    }
}
```

Instancia de servicio web de nómina

Referencia de servicio web de nómina en Lógica de Negocio

Figura 40: Consumo de servicio web en lógica de negocio

A nivel de la pantalla el diseño es similar a la pantalla de aprobación de solicitud de compra con los campos correspondientes a la solicitud de vacaciones (Figura 41). El detalle que se muestra al usuario describe los periodos que el solicitante tiene asignados, la cantidad de días disponibles y usados incluyendo los que se utilizarían de aprobarse la solicitud.

Para completar el proceso de aprobación de una solicitud de vacaciones es necesario que el usuario aprobador ingrese una observación. Para esto se incluyó una opción para el usuario ingrese la observación que desee antes de confirmar su elección de aprobación de la solicitud (Figura 42). Esta opción se despliega sin importar que la solicitud sea aprobada o rechazada.

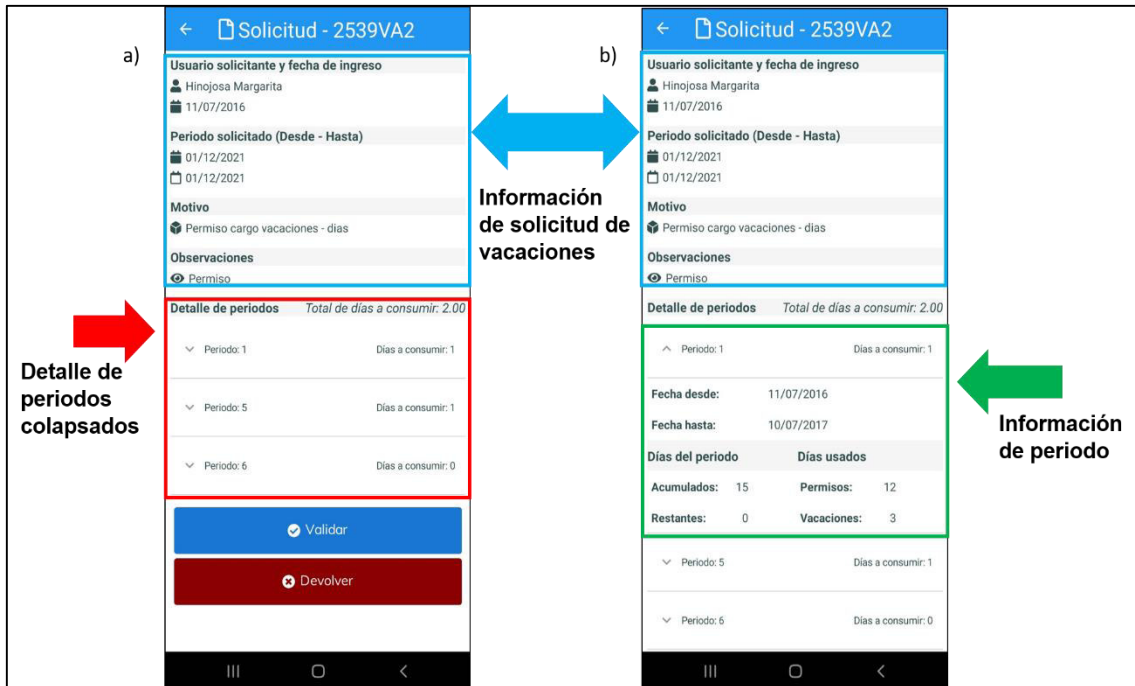


Figura 41: Pantalla de aprobación de solicitud de vacaciones



Figura 42: Control de ingreso de observación para aprobación

3.3.2. Funcionalidad de notificaciones

En esta sección se describe el desarrollo e implementación de las historias de usuario que describen el proceso de la recepción de una notificación cuando el usuario tiene solicitudes pendientes de aprobación. En cada módulo se tiene al menos un nivel de aprobación, el cuál es identificado por el estado que tiene la solicitud.

Cada usuario cuenta con un usuario aprobador designado, quien se encarga de aprobar o anular las órdenes o solicitudes enviadas. Considerando esto se utilizó el estado de la solicitud u orden para identificar los pasos del proceso anteriores a las aprobaciones. De este modo se aseguró que los usuarios que deben realizar una aprobación reciban una notificación al dispositivo móvil una vez que su usuario asignado envíe la solicitud u orden.

Para el envío de las notificaciones se utilizó la plataforma *Firebase* de *Google* para la gestión de notificaciones *PUSH* mediante *Firebase Cloud Messaging*¹⁰ (FCM). Esta solución de mensajería multiplataforma requirió la configuración de un proyecto *Firebase* en la nube para generar un token de identificación del dispositivo móvil. Esto es necesario para el envío de notificaciones como se indica en su documentación [15].

Mediante su integración al proyecto se encontró que la solución tiene problemas de compatibilidad con el lenguaje de programación *C#*. Para solventar esta limitación se siguió el tutorial de “*Implement Push Notifications with FCM and Xamarin.Forms*” del usuario de Youtube Gerald Versluis [16]. En base a su tutorial se realizó el desarrollo a nivel de lógica de negocio, a continuación, se describe el método utilizado para el envío (Figura 43).

Para enviar la notificación se requiere la creación de una aplicación de *Firebase* utilizando el archivo *JSON* de la *key* de *Google* que se genera en la plataforma al configurar el proyecto [15]. En el caso de que existan notificaciones pendientes de enviarse, se recupera el *token* del dispositivo móvil generado por *Firebase* y almacenado con los datos del usuario. Con este *token*, se configura el contenido de la notificación y es enviada al dispositivo. Luego, la notificación es marcada como enviada y termina el proceso eliminando la aplicación de *Firebase*. En caso de que falle el envío, se genera una excepción y se elimina la aplicación de *Firebase*.

¹⁰ *Firebase Cloud Messaging* es una solución de mensajería multiplataforma que permite el envío de mensajes a dispositivos móviles. <https://firebase.google.com/docs/cloud-messaging?hl=es>

```

public void enviarNotificaciones()
{
    FirebaseApp.Create(new AppOptions() { Credential = GoogleCredential.FromFile("private_key.json")});
    try
    {
        MensajeríaDST dsNotificaciones = ListarNotificacionesByEstado(EstadosNotificaciones.NUEVA);
        if (dsNotificaciones.MensajeríaCGWeb.Count > 0)
        {
            foreach (MensajeríaDST.MensajeríaCGWebRow notificación in dsNotificaciones.MensajeríaCGWeb)
            {
                var registrationToken = notificación.ID_DISPOSITIVO.Trim();
                var message = new Message()
                {
                    Token = registrationToken,
                    Notification = new Notification() {Title = notificación.TITULO,Body = notificación.MENSAJE}
                };
                string response = FirebaseMessaging.DefaultInstance.SendAsync(message).Result;
                MensajeríaDST dsNotificación = ListarNotificacionesById(notificación.SECUENCIAL);
                if (dsNotificación.MensajeríaCGWeb.Count > 0)
                {
                    dsNotificación.MensajeríaCGWeb[0].ESTADO = EstadosNotificaciones.ENVIADA;
                    GrabarMensajesCGWeb(dsNotificación);
                }
            }
        }
        FirebaseApp.DefaultInstance.Delete();
    }
    catch (Exception)
    {
        FirebaseApp.DefaultInstance.Delete();
    }
}

```

Creación de App de Firebase

Revisa si existen notificaciones pendientes de enviar

Recupera token del dispositivo del usuario

Configuración de la notificación

Envío de notificación

Marca notificación como enviada

Elimina la App de Firebase si falla o completa el envío

Figura 43: Método de envío de notificación

Este método es ejecutado mediante la publicación de un servicio web en el sitio configurado para un módulo del sistema CG/Web. Para consumir este servicio se desarrolló un servicio *Windows* que se ejecute en el servidor mencionado. Este servicio *Windows* se ejecuta en un intervalo de cinco segundos para que el envío de la notificación sea lo más pronto posible, simulando una ejecución mediante internet.

Las notificaciones deben generarse al momento que el usuario asociado al aprobador envía una solicitud u orden, para ello se creó un *trigger* asociado a la actualización de la tabla que contiene la información de la solicitud u orden (Figura 44).

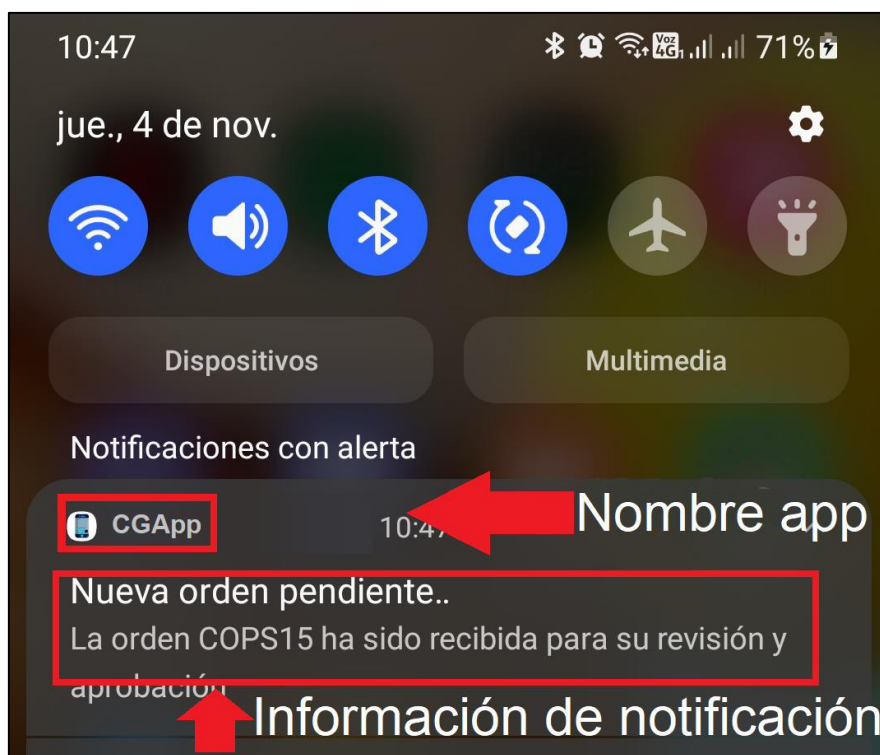


Figura 44: Ejemplo de notificación de solicitud / orden pendiente de aprobación

A continuación, se explican los detalles de los *triggers* desarrollados para cada historia de usuario y las particularidades de cada caso.

3.3.2.1. ACT-18 y ACT-5: Notificación de recepción de una solicitud pendiente de aprobación

En estas historias de usuario se describe la necesidad de recepción de una notificación cuando se ha enviado una solicitud pendiente de aprobación en el módulo de compras (ACT-18) y en el módulo de nómina (ACT-5). Para esto se diseñaron *triggers* que se activan al actualizarse un registro de la tabla asociada, con una estructura y objetivo similar. Estos *triggers* se ejecutan tras la actualización de un registro de la tabla a la que están asociados como se muestra en la Figura 45.

En el *trigger* se verifica que el estado de la solicitud sea el correspondiente al paso o nivel anterior a la aprobación. Si la actualización cumple la condición indicada, toma el valor del último usuario que ha realizado una acción con la solicitud y busca el nombre del usuario aprobador que tiene asignado. Con estos datos, realiza una inserción en la tabla de notificaciones de un nuevo registro que incluye el código de la solicitud recibida para aprobación.


```

CREATE TRIGGER UPDATE_NOTIFICACION_SOL
ON TIGSA_CSOLCA
AFTER UPDATE AS
BEGIN
    DECLARE @EstadoSolicitud VARCHAR(1)
    DECLARE @UsuarioAprobador VARCHAR (100)
    DECLARE @CodigoSolicitud VARCHAR (1000)

    SELECT
    @EstadoSolicitud = inserted.CSCSTS,
    @CodigoSolicitud = CONCAT(inserted.CSCCIA, inserted.CSCMCO, inserted.CSCNSO)
    FROM INSERTED

    IF (@EstadoSolicitud = 'V')
    BEGIN
        select @UsuarioAprobador = USUARIO_APROBADOR from ITBPM_RELACION_USUARIOS WHERE USUARIO IN (SELECT inserted.CSCUSUC FROM INSERTED)

        INSERT INTO TIGSA_MSG01 (TITULO, MENSAJE, ESTADO, USUARIO)
        VALUES('Nueva solicitud pendiente', CONCAT('La solicitud ',@CodigoSolicitud,' ha sido recibida para su revisión y aprobación'), 'N', @UsuarioAprobador)
    END

    IF (@EstadoSolicitud = 'A')
    BEGIN
        select @UsuarioAprobador = USUARIO_APROBADOR from ITBPM_RELACION_USUARIOS WHERE USUARIO IN (SELECT inserted.CSCUAP FROM INSERTED)

        INSERT INTO TIGSA_MSG01 (TITULO, MENSAJE, ESTADO, USUARIO)
        VALUES('Nueva orden pendiente', CONCAT('La orden ',@CodigoSolicitud,' ha sido recibida para su revisión y aprobación'), 'N', @UsuarioAprobador)
    END
END
GO

```

Figura 45: Estructura de *trigger* de compras

3.3.3. Funcionalidad de presentación de reportes gráficos

En esta sección se describe el desarrollo e implementación de las historias de usuario que describen la obtención de información específica de un módulo y su presentación en un gráfico. Para esto se diseñó un *mockup* de ejemplo (Figura 46) en donde se indica que la pantalla contendrá el gráfico con los datos consultados y una con información si resulta necesario para comprender la gráfica.

Se determinó que estas pantallas debían tener las siguientes características:

- Se muestran datos del módulo resumidos en un gráfico estadístico.
- Se muestra información acerca de los datos expresados en la gráfica y filtros para seleccionar los datos.

El tipo de gráfico estadístico utilizado fue consultado con los *stakeholders* en función de la información requerida y la capacidad de presentación en la pantalla de un dispositivo móvil. Los datos son obtenidos consumiendo un servicio WCF y se realiza una conversión de su presentación en función de los componentes del gráfico utilizado. A continuación se describe el desarrollo de las historias de usuario que describen el análisis de información mediante una gráfica de resumen.

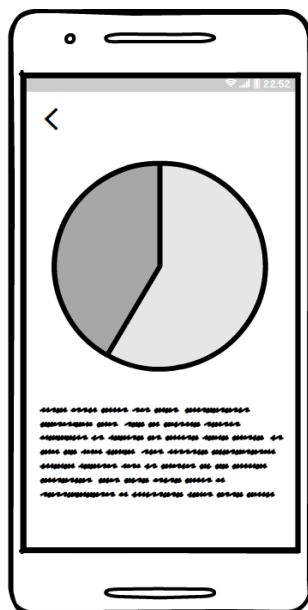


Figura 46: Mockup de pantalla de análisis mediante gráfica

3.3.3.1. ACT-12: Análisis de ítems / proveedores más utilizados en un periodo de tiempo.

La historia ACT-12 indica la necesidad de obtener la información de los ítems / proveedores más utilizados en un periodo de tiempo. Esto con el fin de llevar un control de los ítems más adquiridos, los proveedores más requeridos o el monto gastado en ítems o proveedores. Con esta información se busca complementar los datos presentados al usuario aprobador para que pueda decidir si anula o aprueba una orden o solicitud.

Para gestionar la generación de estas gráficas fue necesario desarrollar una pantalla que permita seleccionar filtros y escoger los datos que se desean visualizar (Figura 47). Esta se compone de dos secciones principales: filtros de datos y selección de datos. La sección de “filtros” permite seleccionar valores que limitaran los datos que se recuperan para el gráfico, en este caso se escoge el rango de fechas que se va a considerar. La sección de “selección” muestra que tipo de datos se pueden mostrar en el gráfico.

Al seleccionar la información que se desea visualizar se desplegará una pantalla con el gráfico estadístico generado. Para la presente historia se escogió utilizar gráficos de pastel con el formato de “dona” (Figura 48.a). Estos gráficos presentan los datos en el interior de las secciones de la “dona”, en caso de que los datos sean monetarios, se muestra el símbolo de la moneda (Figura 48.b). Esta información se complementa con las “leyendas”, la cual indica la notación de colores, indicando a qué ítem o proveedor

pertenece cada color. Por último, al seleccionar una sección, se muestra un texto “indicio” que muestra el ítem o proveedor que se ha seleccionado y el color.

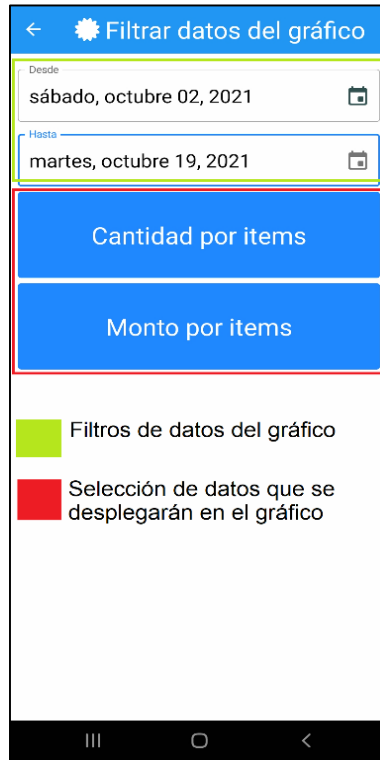


Figura 47: Pantalla de filtrado de datos del gráfico



Figura 48: Gráficas de resumen de ítems / proveedores

3.3.3.2. ACT-17: Análisis de cantidad de órdenes de compra pendientes / atendidas.

La historia ACT-11 describe la necesidad de obtener la información de las órdenes de compra, pendientes o atendidas en un periodo de tiempo. Esto con el fin de llevar un control de las órdenes atendidas y de esta forma evaluar el desempeño del personal. Utiliza una pantalla de filtro de datos similar a la descrita en la Sección 3.3.3.1; en el caso particular de esta historia de usuario, se da a elegir los datos en función del estado que estos tienen. El despliegue de los datos (Figura 49) es muy similar al indicado en la sección ya mencionada, la diferencia radica en que este gráfico se divide en función de las modalidades de compra y el estado de las órdenes. Tiene las mismas secciones indicadas en la Figura 48.



Figura 49: Gráfica para análisis de cantidad de órdenes de compra atendidas

3.3.3.3. ACT-2: Análisis global de grupos de rubros de nómina

La historia ACT-2 describe la necesidad de obtener la información del monto de los grupos de rubros de nómina y presentarlos de forma que se pueda analizar mediante comparación los valores en un periodo de tiempo. Esto con el fin de llevar un control del

estado general de los gastos de la empresa respecto a los ingresos y descuentos de sus trabajadores.

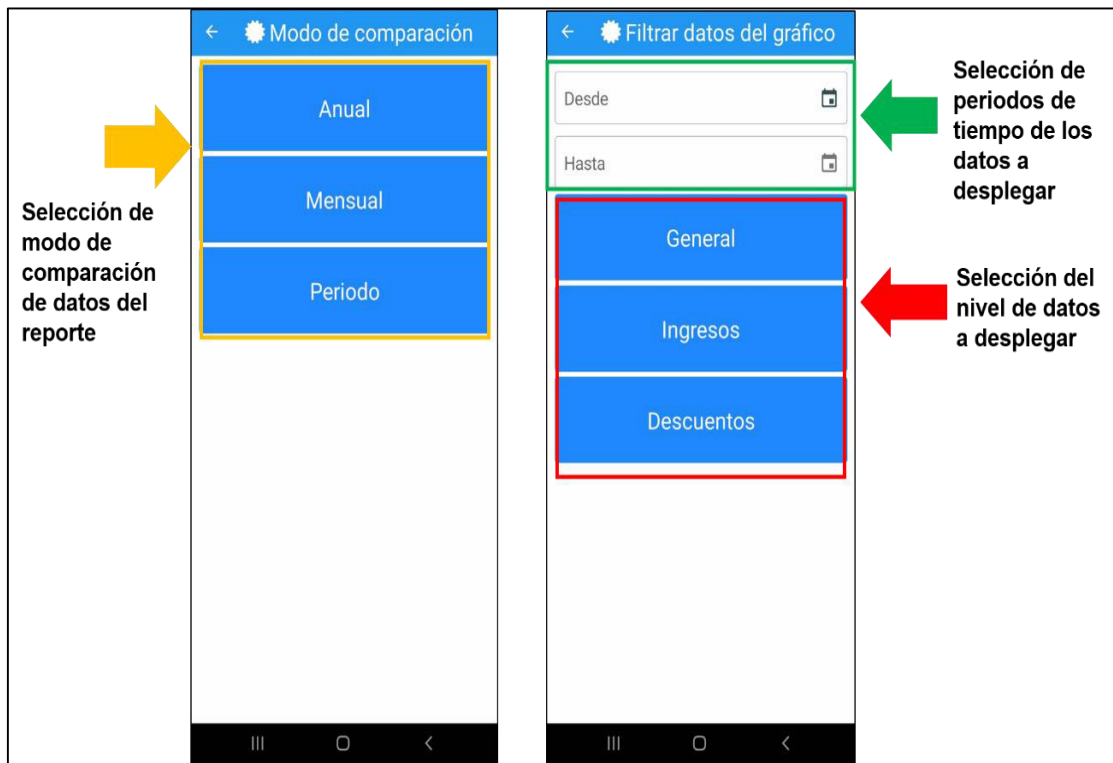


Figura 50: Pantalla de filtro de datos para reporte gráfico de rubros

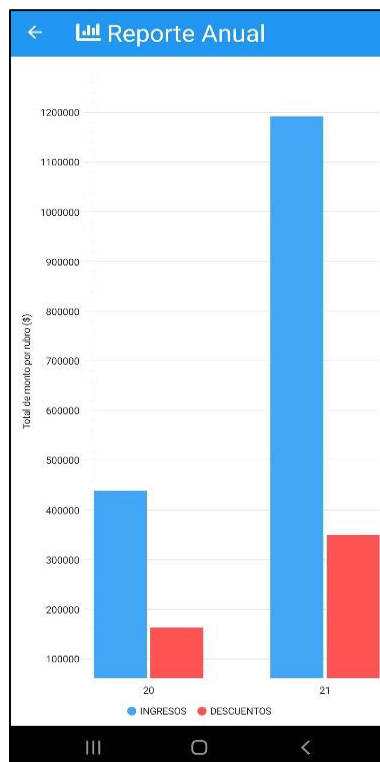


Figura 51: Muestra de reporte gráfico de resumen de rubros

Esta funcionalidad se desarrolló de forma similar las descritas en las Secciones 3.3.3.1 y Figura 47 mediante el despliegue de información en gráficos de resumen a modo de reporte. Para esta historia se optó por gráficos de barras para permitir la comparación entre dos estados o periodos. Utiliza una pantalla de filtro de que permite la selección de los periodos que se desea analizar, el tipo de datos que se quiere mostrar y el nivel de resumen del gráfico (Figura 50).

El despliegue de los datos (Figura 51) se realiza en los gráficos de barras de forma que se muestra la comparación entre los datos similares de los periodos seleccionados. En caso de ser un gráfico de un único periodo, la comparación se realiza entre los datos disponibles para dicho tiempo.

3.3.4. Otras funcionalidades

En esta sección se describe el desarrollo de las pantallas que no están contempladas en las historias de usuario anteriormente mencionadas. Sin embargo, cumplen funciones que complementan el uso de la aplicación y permiten acceder a las pantallas descritas en las Secciones 3.3.1 y 3.3.3. Entre estas se incluyen la selección de un módulo, la autenticación del usuario y el menú de opciones de un módulo seleccionado.

3.3.4.1. Funcionalidades de presentación del menú principal

Para esta pantalla se tomó de base la pantalla inicial del sistema CG/Web (Figura 52). En esta se muestran los módulos de los que se compone el sistema web. Permite acceder al módulo deseado mediante la selección de la opción respectiva representada por una imagen. Se diseñó un *mockup* (Figura 53) en el que se indican los módulos a los cuales se tendrá acceso mediante la solución del presente proyecto en función de la estructura descrita en la Sección 3.1.3.2. Se definieron las siguientes características para la pantalla:

- Primera pantalla que se muestra en la aplicación.
- Se muestran los módulos principales que componen la aplicación.
- Permite ingresar al menú de un módulo.
- Permite acceder a la pantalla de Login.



Figura 52: Pantalla principal del sistema CG/Web

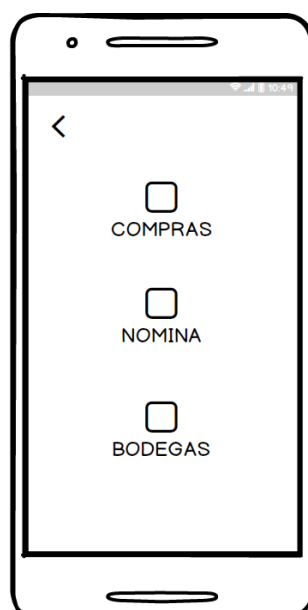


Figura 53: Mockup menú principal app

En los accesos a los módulos en la aplicación se utilizaron las imágenes asociadas en la pantalla del sistema CG/Web, como se ve en la Figura 54 . Para ello se utilizó el control *ImageButton* propio de *Xamarin.Forms*, el cual permite configurar una imagen en un botón para su selección.

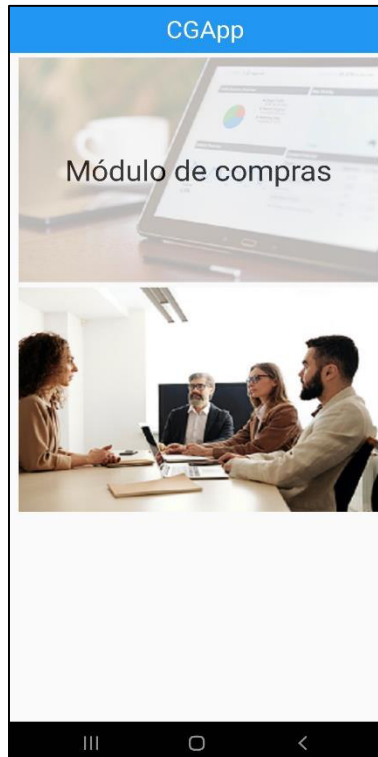


Figura 54: Pantalla de menú principal

3.3.4.2. Funcionalidad de inicio de sesión

En esta pantalla se realiza el ingreso de credenciales del usuario, su autenticación y el acceso al módulo seleccionado en la pantalla descrita en la Sección 3.3.4.1. Se diseñó un *mockup* (Figura 55) para indicar la estructura de la pantalla y se definieron las siguientes características:

- Segunda pantalla a la cual el usuario tiene acceso.
- Se pide el ingreso de las credenciales del usuario.
- Se comprueba que el usuario existe registrado en el sistema con las credenciales que utiliza en el sistema CG/Web.
- Si la autenticación se realiza con éxito, se actualiza la información del usuario con el *token* de *Firebase* generado y que identifica al dispositivo.
- Condiciona las opciones del menú que se presentarán en la siguiente pantalla.

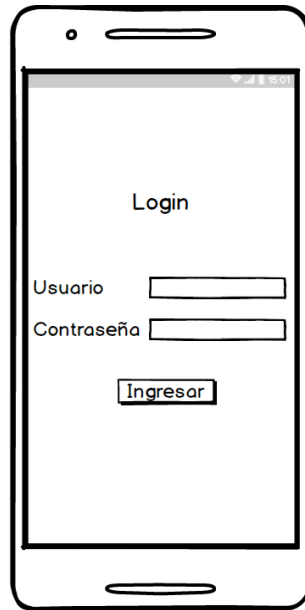


Figura 55: Mockup de pantalla de inicio de sesión (Login)

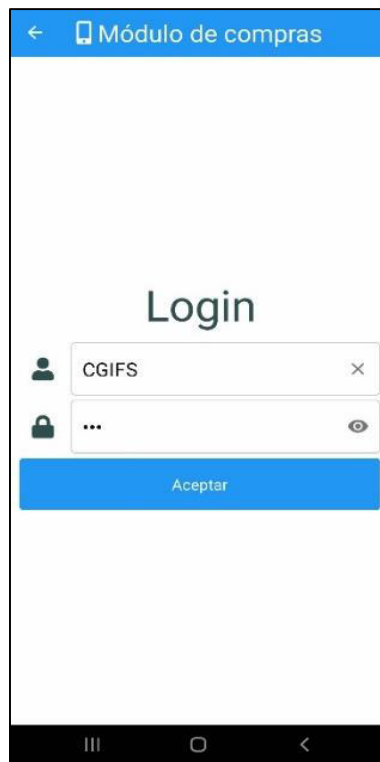


Figura 56: Pantalla de inicio de sesión (Login)

En el desarrollo se agregaron íconos de *FontAwesome* para complementar los datos y opciones de la pantalla, además del título del módulo al que se desea acceder (Figura 56).

3.3.4.3. Funcionalidad de presentación de menú de opciones de un módulo

Para el diseño de esta pantalla se realizó un *mockup* (Figura 57) usando de base la pantalla descrita en la Sección 3.3.4.1. Mediante esta pantalla se permite el acceso a las opciones que se describen en las historias de usuario desarrolladas. Además, debe cumplir las siguientes características:

- Permite ingresar a la acción que describe.
- Permite regresar a la pantalla anterior (Inicio de sesión).
- Sus opciones se muestran condicionadas por los permisos del usuario.

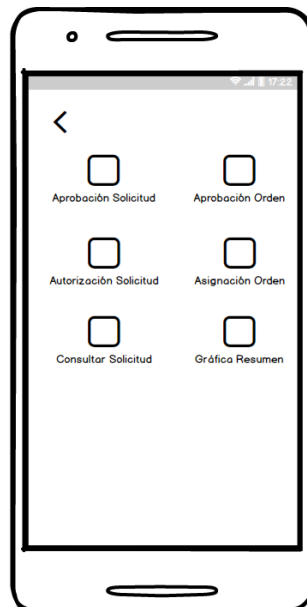


Figura 57: Mockup de pantalla de menú de opciones de modulo

Esta pantalla se manejó mediante un control *drawerView*, el cual se compone de una sección principal y un *drawer*. Esto debido a que las opciones del módulo se clasifican en sub opciones en función de datos; por ejemplo, la modalidad de compra que divide los buzones de aprobación. De esa forma el usuario selecciona la opción del menú a la que desea acceder (Figura 58.a) y se le despliega las sub opciones para que especifique a cual desea ingresar (Figura 58.b). Estas opciones están condicionadas a los permisos que tiene el usuario y que son comprobados en la pantalla de inicio de sesión (Sección 3.3.4.2).



Figura 58: Pantalla de menú de opciones del módulo

3.3.5. Requisitos de seguridad MASVS

En esta sección se describe la evaluación de la aplicación respecto a los requisitos del estándar MASVS por grupo de requerimientos. En cada sección se muestra el análisis y/o pruebas que se realizaron por requisito para verificar su correcta aplicación.

3.3.5.1. Arquitectura, diseño y modelado de Amenazas

Durante el desarrollo se identificaron todos los componentes que se requerían para que la aplicación cumpliera con los requerimientos del proyecto. Cada elemento fue identificado en las diferentes fases del desarrollo. A nivel de diseño se identificaron los componentes y su relación, como se ve en los modelos de arquitectura (Ver sección 3.1.3.2 y Anexo 2). Con esto se cumple el requisito de seguridad MSTG-ARCH-1 por el cual se deben identificar y asegurar la utilidad de los componentes.

El proyecto cuenta con controles de seguridad legados del sistema CG/Web como el inicio de usuario con credenciales, la asignación de roles al usuario para el acceso de funcionalidades y la validación de datos de entrada, entre otros. Además, cuenta con controles para controlar el despliegue de información en el entorno móvil. Estos se basan en los casos de ejemplo que, como se indica en la sección 3.2.2, se utilizaron en

las historias de usuario para complementar la información. Los controles de seguridad se aplican en su mayoría a nivel del servidor, con casos puntuales como la verificación de datos de entrada en pantalla siendo aplicados en la aplicación cliente. Con esto se cumple con el requerimiento MSTG-ARCH-2 del estándar que indica que los controles de seguridad deben ser aplicados en los servidores correspondientes, no solo en el cliente.

La aplicación maneja información del usuario que es considerada sensible dentro del sistema CG/Web y el contexto del proyecto: credenciales del usuario y el *token* único del dispositivo. Las credenciales, contraseña y nombre de usuario, es información requerida para el acceso a los servicios del sistema CG/Web. Además, el último es indispensable para la llamada de los servicios debido a los registros de auditoría en varias operaciones. Por otro lado, el *token* del dispositivo es el identificador que permite diferenciarlo y es vital para el envío de notificaciones. Estos datos son sensibles debido a que prestan oportunidad para identificar e interactuar con el dispositivo, ingresar con su sesión en el sistema y acceder a los servicios.

3.3.5.2. Almacenamiento de datos y la privacidad

Como se menciona en la sección 3.3.5.1, la aplicación maneja tres datos considerados como sensibles: las credenciales del usuario (nombre y contraseña) y el *token* del dispositivo. En el diseño de la aplicación se concertó con los *stakeholders* que no era relevante el almacenar información en la memoria del dispositivo móvil debido a que el usuario requiere conectarse a Internet para poder realizar las operaciones. Sin embargo, durante el desarrollo se identificó que el *token* del dispositivo debía almacenarse en la memoria debido a la plataforma utilizada para las notificaciones.

Para guardar esta información se consideraron las sugerencias de seguridad propuestas en la documentación de *Android Developers*, la cual indica tres posibles medios para almacenar los datos: Almacenamiento interno, almacenamiento externo y proveedores de contenido [17]. Se descartó el uso del almacenamiento externo debido a los problemas de seguridad que representa que pueda ser modificado por aplicaciones externas o ser extraviado al guardarse en una memoria extraíble. El proveedor de contenido es recomendado si la información deberá ser accedida o ser exportada hacia otras aplicaciones. Dado que este dato es requerido sólo por esta aplicación se decidió utilizar la opción del almacenamiento interno. De esta forma se cumple con el requisito

MSTG-STORAGE-2 al no guardar datos sensibles fuera del propio contenedor de la aplicación móvil.

En el proyecto fue necesario el uso de la plataforma *Firebase* para el envío de notificaciones *push* al dispositivo de un usuario. Esto se realiza guardando el *token* de identificación del dispositivo en la base de datos del sistema CG/Web y enviándolo junto al mensaje de notificación a la plataforma como se explica en la sección 3.3.2. Al ser un requisito del proyecto contemplado en la arquitectura de la aplicación no presenta conflicto con el cumplimiento del requisito de seguridad MSTG-STORAGE-4.

En cuanto a la presentación de información sensible mediante las pantallas, de los tres datos identificados sólo dos son utilizados de forma visible en alguna interfaz. El nombre de usuario y la contraseña requieren ser escritos y mostrados durante el inicio de sesión en la aplicación. En el caso del nombre de usuario, se concertó con los *stakeholders* que a pesar de considerarse información sensible, su nivel de protección no debía ser tan estricto. Para comodidad del usuario, se debe permitir visualizar este dato en la pantalla de inicio de sesión y utilizar la caché del teclado del dispositivo móvil para agilizar su escritura. Sin embargo, para la contraseña estos dos aspectos deben ceñirse a los requerimientos de seguridad al ser un dato de alta sensibilidad y su ingreso debe ser estrictamente personal. Para cumplir los requerimientos MSTG-STORAGE-7 y MSTG-STORAGE-8 se utilizó el control de *DevExpress PasswordEdit*, el cual de forma predeterminada reemplaza los caracteres que se despliegan en pantalla por símbolos de puntos. Sin embargo, para ayuda del usuario es posible desactivar de forma temporal esta función mediante el ícono ubicado al final del cuadro de entrada de texto. De igual forma, este control desactiva la caché del teclado, evitando que la contraseña pueda ser ingresada sin presencia del usuario. Ambos requerimiento cumplidos se aprecian en la Figura 59, la cual muestra la pantalla de *Login* a mitad del ingreso de la contraseña.

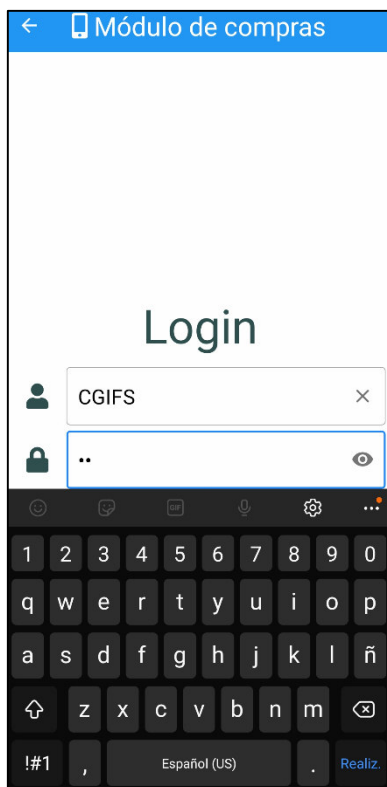


Figura 59: Pantalla de Login - campo contraseña

3.3.5.3. Criptografía

Para la encriptación de información sensible se requirió que se utilice el método de encriptación ocupado por el sistema CG/Web para la seguridad y privacidad de los datos sensibles y la autenticación del usuario. Dado que este método de encriptación es utilizado y no se ha reportado problemas en su uso en el sistema CG/Web, cumple con el requerimiento de seguridad MSTG-CRYPTO-2 que requiere el uso de implementaciones de criptografías probadas.

Dado que el uso de los métodos del sistema CG/Web legados condiciona la aplicación del método de encriptación del sistema web se utilizan las primitivas de seguridad apropiadas para el caso del proyecto. Esto cumple con el requerimiento MSTG-CRYPTO-3 del estándar MASVS, puesto que es necesario ocupar las herramientas legadas del sistema CG/Web por la condición de “sistema satélite” de la solución del presente proyecto.

Sin embargo, pese a estar condicionado a las herramientas del sistema web base, se cumplieron los requerimientos posibles, como es el caso del MSTG-CRYPTO-5 en el que se indica que no debe usarse la misma clave para varios propósitos. Esto se cumple

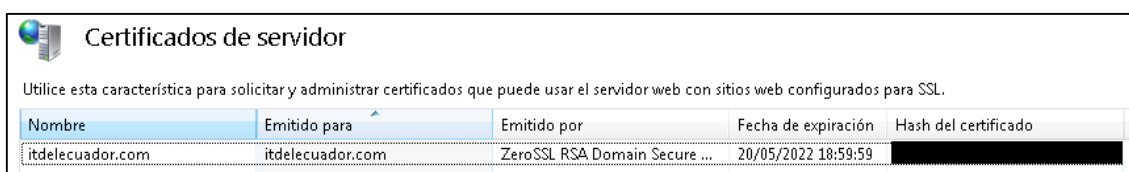
al utilizar la configuración del sistema CG/Web para la encriptación requerida en la autenticación al sistema, pero definiendo una clave distinta concertada con los *stakeholders* para realizar la encriptación de otros datos.

3.3.5.4. Autenticación y manejo de sesiones

Como se indica en la sección 3.2.1, la aplicación de este proyecto consume servicios WCF publicados en el servidor del cliente. Para acceder a estos servicios y como parte de la arquitectura heredada del sistema CG/Web, el usuario debe iniciar sesión autenticándose con usuario y contraseña. Esto se realiza mediante la llamada al servicio de *login* para lo cual las credenciales deben cumplir con todas las condiciones de un usuario del sistema CG/Web. Estos usuarios deben existir en la base de datos, y contar con los permisos respectivos del sistema para acceder a los módulos que les sean necesarios. Una vez la aplicación autentica al usuario mediante el servicio respectivo, este tiene acceso a las opciones asignadas mediante rol a su usuario, lo que condiciona el despliegue de menú dinámico que se explica en la sección 3.3.4.3. Este mecanismo de usuario/contraseña cumple con los requerimientos MSTG-AUTH-1 y MSTG-AUTH-5 del estándar de seguridad.

3.3.5.5. Comunicación a través de la red

Respecto a la comunicación por internet, el requerimiento MSTG-NETWORK-1 requiere que la información sea cifrada y enviada usando TLS. Para esto es necesario el uso de certificado SSL/TLS configurado en el servidor. Para el proyecto se verificó que el servidor utilizado tenga un certificado válido asociado, como se ve en la Figura 60.



Certificados de servidor

Utilice esta característica para solicitar y administrar certificados que puede usar el servidor web con sitios web configurados para SSL.

Nombre	Emitido para	Emitido por	Fecha de expiración	Hash del certificado
itdelecuador.com	itdelecuador.com	ZeroSSL RSA Domain Secure ...	20/05/2022 18:59:59	[Redacted]

Figura 60: Certificado SSL/TLS asociado a servidor

A nivel de la aplicación se configuró en las opciones avanzadas de Android que se implemente SSL/TLS 1.2, la versión más actual permitida por el compilador para el proyecto, como se muestra en la Figura 61. Con esta configuración se cumple con el requerimiento MSTG-NETWORK-2 al cumplir con la mejor configuración en la aplicación para la versión del sistema operativo.

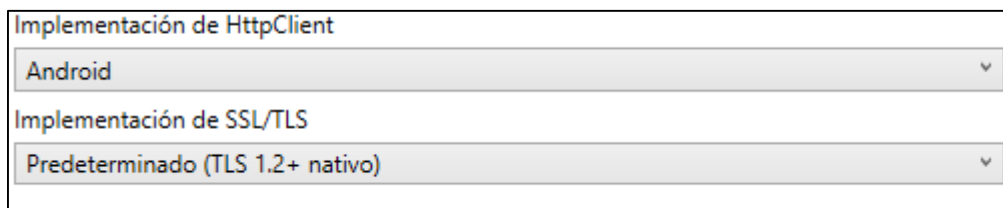


Figura 61: Implementación de SSL/TLS en aplicación

3.3.5.6. Interacción con la plataforma

La entrada de datos se realiza mediante el consumo de los servicios WCF como se explica en la sección 3.2.1. Estos datos de entrada son obtenidos desde la base de datos del sistema CG/Web y expuestos en pantalla para que el usuario pueda seleccionar las combinaciones que le permiten acceder a la información y funciones detalladas en la sección 3.3.1. Las excepciones son las observaciones ingresadas al realizar una anulación de solicitud/orden de compra y la aprobación de solicitudes de vacaciones.

Todos los datos de entrada pasan por una serie de verificaciones que comprueban su validez en el nivel de Lógica de Negocio de los componentes del servidor. Estas comprobaciones son legadas del sistema CG/Web y controlan que no se ingresen datos basura a la base de datos, que los datos sean del tipo que corresponde y que sean válidos en el contexto de la aplicación. Estos ingresos controlados se comprobaron durante el desarrollo mediante la aplicación de las pruebas automatizadas mencionadas en la sección 3.2.2 y cuyos resultados se describen en la sección 3.3.1. En el caso de las funcionalidades de notificaciones y consulta de información no se permite la inserción de datos a la base.

A nivel de Acceso de Datos se controla mediante sentencias SQL la consulta de datos de la base de datos del sistema CG/Web. En este nivel se legan los métodos utilizados por el sistema web y que son necesarios para las funcionalidades de la aplicación móvil, como se indica en el diagrama de arquitectura (Ver sección 3.1.3.2).

3.3.5.7. Calidad de código y configuración de compilador

Como parte del despliegue del APK se corroboró que la configuración *Release* sea correcta para la generación. Como se ve en la Figura 62, el IDE *Visual Studio* permite

inhabilitar las opciones de compilación del proyecto por configuración. Para el caso actual se aseguró que las características *Fast Deployment* propias del modo de depuración se encuentren deshabilitadas. Así también se comprobó que las opciones de depuración se encuentren desactivadas en el perfil en cuestión. Esto permite cumplir con los requerimientos MSTG-CODE-2 y MSTG-CODE-3 puesto que al iniciar la compilación, las configuraciones descritas se aplican y provocan que los símbolos de debug, en caso de existir, sean removidos de los binarios nativos.

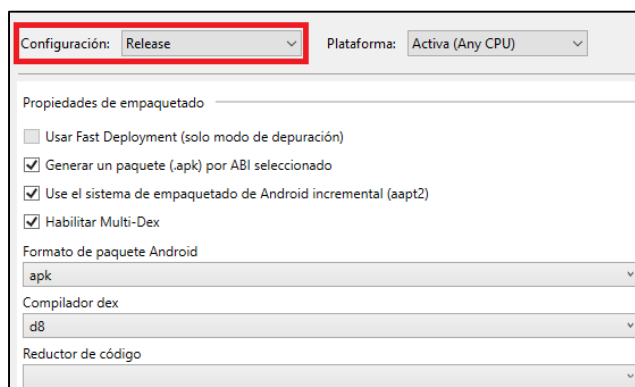


Figura 62: Configuración release

Durante el desarrollo de la aplicación fue necesario hacer uso de los códigos de prueba y logs de errores para identificar posibles falencias de la aplicación. Sin embargo, esto se manejó a nivel del servidor mediante la aplicación de pruebas como se explica en la sección 3.2.2. En cuanto al aplicativo cliente, estos mensajes propios de depuración fueron eliminados una vez se completaba cada historia de usuario. Sin embargo, como control para evitar las caídas debido a excepciones, a nivel de la app se utilizaron las prestaciones de cada control. Por ejemplo, se utilizó el control *Edit Text* de *Xamarin.Forms* de *DevExpress* con mensajes de error que son activados si el dato ingresado no cumple con los controles impuestos (Ver Figura 63). Además, se utilizaron *popUps* de mensaje para que el usuario reciba un aviso en caso de producirse excepciones en acciones en el lado del servidor como se muestra en la Figura 64. De esta forma se cumplen los requerimientos MSTG-CODE-4 y MSTG-CODE-6 que se refieren a los mensajes de depuración y a las excepciones en la aplicación respectivamente.

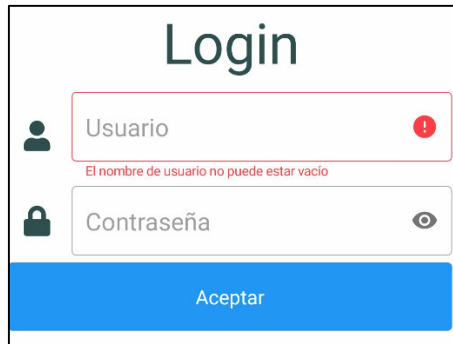


Figura 63: Manejo de excepciones - mensajes en controles



Figura 64: Manejo de excepciones - popUps

3.3.6. Resumen de producción

Durante la elaboración del presente proyecto se buscó desarrollar las funcionalidades identificadas tras el análisis de los requisitos funcionales, explicado en la Sección 3.1.3. En función de la prioridad asignada tras el análisis, se trabajó en aquellas que más importancia tenían para la empresa y para los objetivos propuestos en el presente proyecto. En esta sección se resumirá el proceso de desarrollo general y se expondrán las limitaciones detectadas que afectaron la ejecución.

Para el desarrollo se definió un total de 14 historias de usuario que comprendían el análisis de requerimientos funcionales detallado en la Sección 3.1.3.1. Se completó un total 8 historias, las cuales han sido explicadas en las Secciones 3.3.1 (Funcionalidad de aprobaciones), 3.3.2 (Funcionalidad de notificaciones) y 3.3.3 (Funcionalidad de presentación de reportes gráficos) según tu tipo de funcionalidad. En la Figura 65 se indica un gráfico resumen con el número de historias completadas por tipo de funcionalidad.

Fueron en total 6 historias de usuario que no lograron completarse durante el desarrollo, esto debido a cambios del sistema CG/Web ocurridos durante el proyecto. En su última actualización, el sistema web implementó un nuevo módulo que se integra a dos de los módulos contemplados en el alcance del presente proyecto; Nómina y Bodegas.

En el caso del módulo de Bodegas, las solicitudes ahora forman parte del buzón general del módulo BPM y genera nuevos registros en base de datos. Dado que a fecha de finalización del presente proyecto todavía se está definiendo la nueva lógica para su aprobación, el desarrollo la historia de usuario que describe dicho proceso (ACT-6) fue interrumpido. Esto en concordancia con los *stakeholders*, quienes han aprobado que se aplique el tiempo para mejoras en las pantallas de uso general de la aplicación (Sección 3.3.4) y en correcciones de formato de las demás pantallas.

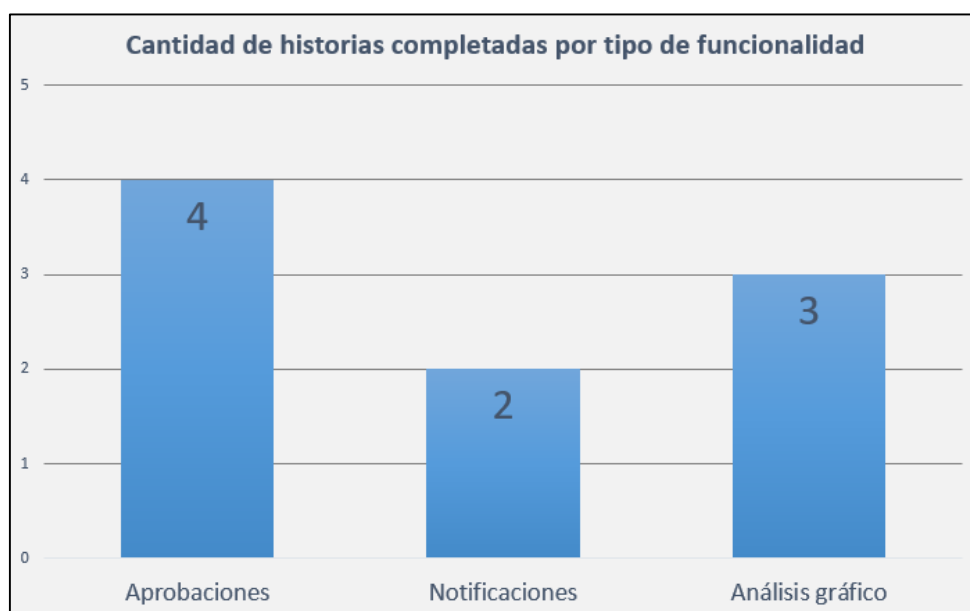


Figura 65: Cantidad de historias completadas por tipo de funcionalidad

De igual forma el desarrollo de la historia ACT-5, al estar ligado a la funcionalidad ya mencionada al ser la notificación de pendiente de aprobación, fue interrumpido para dar más tiempo a las demás funcionalidades. Esto fue acordado debido a que, si el usuario no tiene la oportunidad de aprobar la solicitud, recibir una notificación en la aplicación provocaría confusión al usuario.

Finalmente, debido al cambio en el sistema CG/Web antes mencionado, fue necesario realizar un análisis intensivo para redefinir el proceso de aprobación de solicitud de vacaciones. Por ello, se seleccionaron historias de usuario con menor relevancia para cumplir los objetivos propuestos en el proyecto:

- ACT-2: Como usuario gerencial del módulo de nómina se desea acceder a la información de los grupos de rubros de nómina para realizar un análisis global.
- ACT-3: Como usuario gerencial del módulo de nómina se desea acceder a los funcionarios que hayan trabajado más horas extras un periodo de nómina para poder realizar un análisis
- ACT-4: Como usuario gerencial del módulo de nómina se desea acceder a la información de la cantidad de los tipos de horas extras trabajadas en un año para realizar un análisis comparativo
- ACT-7: Como usuario gerencial del módulo de bodegas se desea acceder a los cinco bienes con mayor inventario de una bodega en un periodo de tiempo para poder realizar un análisis
- ACT-8: Como usuario gerencial del módulo de bodegas se desea recibir una notificación para poder conocer la existencia de solicitudes de bodega pendientes de aprobación y poder aprobarlas

El desarrollo de estas historias fue suspendido en consenso con los *stakeholders*, pues su importancia para el valor de la aplicación resultaba inferior al resto. Esto respaldado por el nivel de prioridad asignado durante el análisis de la Sección 3.1.3.1. A continuación, en la Figura 66 se muestra un gráfico de resumen de las historias definidas y las completadas durante el proyecto. Por su parte, debido a las características propias del proyecto, se detectaron requisitos de seguridad no compatibles con el desarrollo. Se llevó un control de estos gracias al documento *CheckList* oficial de OWASP en el cual se indican aquellos controles que no eran aplicables al proyecto en cuestión (Ver Anexo 1).

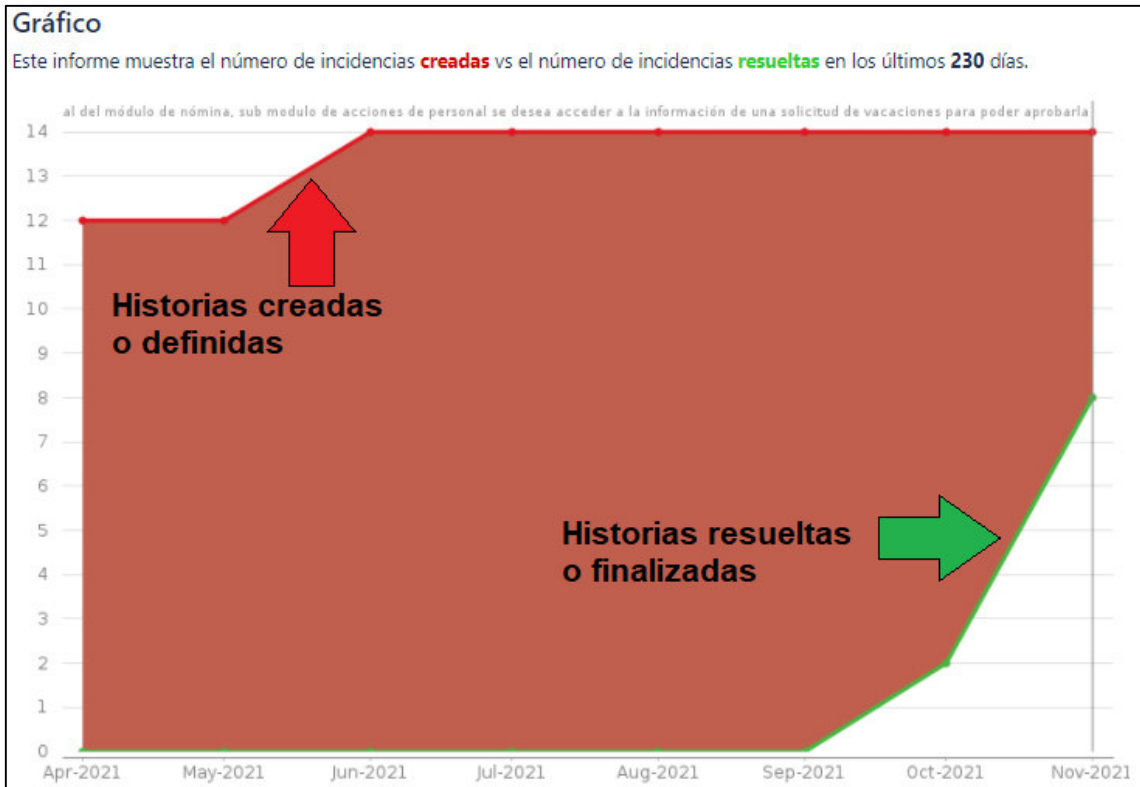


Figura 66: Gráfico resumen historias creadas vs finalizadas

4. Resultados

En esta sección se describen los resultados obtenidos tras el proceso de desarrollo del proyecto. Se incluye una descripción general de la solución en función de la metodología Mobile-D, la encuesta aplicada durante el desarrollo, los gráficos y puntuación final de la aplicación del estándar MASVS v1.3 mediante la aplicación del *CheckList*.

Para el desarrollo del presente proyecto se aplicó la metodología de desarrollo ágil Mobile-D, la cual como se explica en la sección 2.1 consta de cinco fases. Durante estas fases se deben cumplir un determinado número de actividades que requieren de entradas y generan salidas, las cuales se detallan en la documentación de la metodología. Como ejemplo se menciona la lista de requerimientos iniciales, la cual es un producto de la recolección de requerimientos de la fase de exploración pero que se vuelve entrada durante las actividades para la definición del alcance [4]. En función de esta documentación se siguieron las actividades descritas en secciones separadas. Sin embargo, las fases de producción, estabilización y pruebas se consideraron partes de una sola fase, esto por la utilización de la técnica SBE que se indica en la sección 3.2.2. Para el presente proyecto se agruparon las actividades de las 3 fases, provocando que se pueda llevar un mejor control en su avance y sea posible recibir mayor retroalimentación de los *stakeholders*.

Con el fin de implementar las funcionalidades descritas en los requerimientos de la sección 3.1.3.1, fue necesario trabajar junto a los *stakeholders* para comprender los procesos y datos involucrados. Estas experiencias de usuario fueron registradas en las historias de usuario, como se indican en la sección 3.1.3.4 y permitieron tener un mejor control en el avance del proyecto. Por esto fue posible identificar los componentes que se debían legar desde el sistema CG/Web y los que debían ser agregados durante el desarrollo del proyecto para completar la arquitectura de la solución (Ver sección 3.1.3.2 y Anexo 2). Como complemento al diseño de la solución e historias de usuario se participó a los *stakeholders* en una encuesta ya mencionada en sección 3.3.1.1. Los resultados de la misma que se pueden ver en el Anexo 5 permitieron tener una mayor comprensión sobre las necesidades de diseño de los usuarios. Por ejemplo, el color y contraste de las pantallas para aprobaciones y la forma de dividir las secciones en la interfaz sin afectar a la cantidad de información.

Finalmente, para implementar la seguridad como parte de la solución del proyecto se usó de base el estándar MASVS v1.3 para determinar requerimientos de seguridad

aplicables y determinar si la aplicación resultante del proyecto puede considerarse segura. Como se ve en la Figura 67, la aplicación cumple con un nivel de cumplimiento de requerimientos de seguridad aceptable. Los requerimientos V2 y V4 se muestran como los de mayor porcentaje de cumplimiento, siendo los apartados referentes al almacenamiento de datos y autenticación respectivamente. Los demás requerimientos muestran un porcentaje de cumplimiento más cercano a 70%, indicando que si bien la aplicación cumple con controles de seguridad para esos requisitos, no todos han sido aplicados con éxito. Por último, los requerimientos V8 que tienen un porcentaje de cumplimiento de 0% sólo son indicados como referencia, pues como se indicó en la sección 3.1.2, el nivel aspirado a conseguir con este proyecto es el nivel L1 de MASVS v1.3, en el cual los controles del requerimiento V8 no son aplicables. Esto no significa que el producto tenga vulnerabilidades, como se describe en la sección 2.3, esto es un indicador que la aplicación cuenta con la protección estándar básica.

Para la aplicación de la seguridad en la solución se tuvieron que depurar controles que no eran compatibles con la realidad del entorno de desarrollo. Esto debido a conflictos y condiciones presentadas por el sistema CG/Web, del cual la solución del proyecto es un sistema satélite. Una vez aplicados los controles compatibles descritos en la sección 3.3.5 se pudo obtener un resultado del documento de *CheckList*, el cual se puede ver en la Figura 68 y que indica que el cumplimiento del estándar MASVS v1.3 es más que razonable.

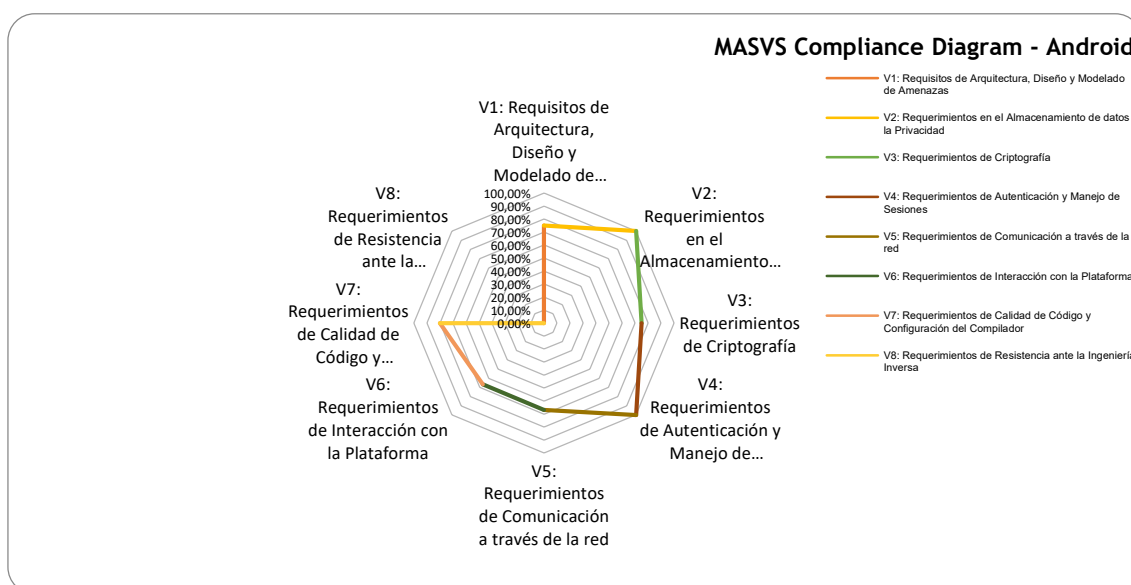


Figura 67: Resultados de *Checklist* de aplicación de MASVS v1.3

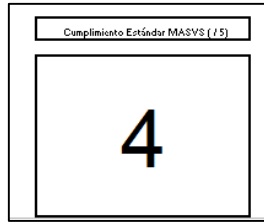


Figura 68: Nivel de cumplimiento de estándar de seguridad MASVS

5. Conclusiones y recomendaciones

En esta sección se describen las conclusiones obtenidas tras el desarrollo del proyecto incluyendo el estado de la solución respecto a las metas propuestas en la Sección Objetivos. También se incluyen las recomendaciones que se han formulado en base a la experiencia del desarrollo del proyecto.

5.1. Conclusiones

El producto CGApp es una aplicación móvil segura para el sistema CG/Web que agiliza la ejecución de los procesos de aprobación que requieren intervención de usuarios gerenciales tales como los directores de los proyectos de empresas clientes de I.T. del Ecuador. Como se explicó en la Sección 3.1.2, la funcionalidad de aprobación de solicitudes es la principal actividad de estos usuarios que provocaba el retraso en ejecución de procesos. La interfaz gráfica de esta funcionalidad fue modernizada como se indica en la Sección 3.3.1 para adaptarla al ambiente móvil, permitiendo una mejor interacción con el usuario. Esta solución ha efectuado las actividades de nivel gerencial con datos válidos de un piloto de empresas como Metro de Quito y ETAPA, quienes han contratado el sistema web CG/Web, comprobando su utilidad para dichos clientes.

La aplicación de la metodología Mobile-D en el proceso de desarrollo de software asegura la calidad del entorno de la aplicación móvil durante su ejecución. Su complementación con la práctica SBE permitió cubrir los posibles escenarios como se explica en la Sección 3.2.2, de forma que se puede asegurar la fiabilidad de la aplicación y su eficacia en la ejecución de las actividades. Esto, sumado a la automatización personalizada de pruebas unitarias para el sistema CG/Web, permite la planificación de nuevos proyectos de actualización del sistema manteniendo la calidad del presente trabajo.

Para agilizar la ejecución de los procesos críticos manejados por CG/Web, CGApp incluye la recepción de notificaciones de solicitudes pendientes de aprobación en tiempo real. Esto aporta a la reducción y/o eliminación teórica de “tiempos muertos” identificados en la Sección 3.1.2 y permite que los usuarios gerenciales gestionen de mejor forma su tiempo de respuesta. Así también, el acceso inmediato a información actualizada del estado de la empresa con relación a los módulos abarcados permite a los usuarios gerenciales el agilizar su toma de decisiones.

La integración de CGAPP con el sistema CG/Web se evidencia en la utilización de su base de datos, recursos preinstalados en los entornos de los clientes de I.T. del Ecuador

indicados en el diagrama arquitectónico en la Sección 3.1.3.2, lo que ha permitido un importante sincretismo con el software web original, así como la aplicación de tecnologías de comunicación y controles de seguridad descritos en el estándar MASVS y detallados en la sección 3.3.5 los que permiten potenciar las capacidades del sistema CG/Web, otorgando mayor valor a su contratación. La integración de ambos productos vuelve al producto de I.T. del Ecuador una opción más competitiva dentro del mercado nacional actual de los sistemas ERP.

5.2. Recomendaciones

Durante el desarrollo del producto CGApp se evidenciaron dificultades en el análisis del código fuente debido a una desmesurada cantidad de términos ambiguos o de nomenclatura similar pero significado distinto entre módulos. Por esto se recomienda que se realice un análisis para la definición de un estándar en las prácticas utilizadas en el equipo de desarrollo del sistema. Esto con el fin de facilitar el entendimiento de las funcionalidades y llevar un mejor control entre módulos.

El código fuente del sistema CG/Web cuenta con una preocupante cantidad de datos quemados como entrada de métodos. Se aconseja considerar el uso de recursos que otorga el lenguaje C# para la definición y uso de constantes y diccionarios. Con esto se busca evitar la quema de datos en distintas secciones, lo cual será un obstáculo en caso de actualizaciones. También se conseguiría mejorar el entendimiento del código al utilizar nombres de constantes en lugar de sus valores.

La estructura por capas del sistema CG/Web presenta varias ventajas para la organización del código y el encapsulamiento de sus funcionalidades. Sin embargo, se identificó que este potencial se ha desperdiciado en varias secciones de código, evidenciado en la lógica de negocio codificada en la capa de presentación de pantallas. Se aconseja realizar un análisis exhaustivo para identificar y migrar estas secciones a su capa respectiva para permitir un mejor consumo de recursos en cada módulo y prevenir fallas de seguridad en las pantallas del sistema.

En varias funcionalidades, el sistema CG/Web debe realizar inserciones en distintas tablas de la base de datos. Estos registros se manejan a nivel de lógica de negocio en el sistema, requiriendo el uso de transacciones. Sin embargo, esto provoca que el sistema ocupe mucho tiempo en caso de que sean varias inserciones en diversas tablas. Se recomienda que se apliquen *stored procedures* para optimizar el tiempo de ejecución de las inserciones, en especial aquellas que deben manejar un gran volumen de datos.

Se propone potenciar la capacidad de control y seguridad del producto CG/Web en base a los resultados de la aplicación del estándar MASVS en el proyecto actual. Puesto que se evidenciaron falencias en controles estandarizados de seguridad que en algunos casos limitaban la aplicación de controles específicos de dicho estándar. Esto con el fin de robustecer la seguridad del sistema web y consecuencia la del producto resultante de este proyecto en versiones futuras. El gestionar estos controles permitirá que los usuarios puedan interactuar de mejor forma con las opciones de la aplicación y otorgará una mejor recepción de notificaciones.

6. Referencias

- [1] 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>.
- [2] M. Muzo, «Levantamiento de procesos postergados en sistema CG/Web de ETAPA,» Información Tecnológica del Ecuador, Quito, 2019.
- [3] P. Blanco, J. Camarero, A. Fumero, A. Werterski y P. Rodríguez, «Metodología de desarrollo ágil para sistemas móviles. Introducción al desarrollo con Android y el iPhone,» 2009. [En línea]. Available: http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf. [Último acceso: 27 Noviembre 2020].
- [4] 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].
- [5] 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].
- [6] 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].
- [7] 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].
- [8] D. Radigan, «Puntos de historia y estimación,» Atlassian, 20 Agosto 2019. [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/estimation>. [Último acceso: 19 Septiembre 2021].

- [9] S. Cuthbertson, «Welcoming Android 10!,» 3 Septiembre 2019. [En línea]. Available: <https://android-developers.googleblog.com/2019/09/welcoming-android-10.html>. [Último acceso: 30 Noviembre 2020].
- [10] Microsoft, «Desarrollo móvil multiplataforma en Visual Studio,» Documentación Visual Studio, 17 Octubre 2019. [En línea]. Available: <https://docs.microsoft.com/es-es/visualstudio/cross-platform/cross-platform-mobile-development-in-visual-studio?view=vs-2019>. [Último acceso: 30 Noviembre 2020].
- [11] Microsoft, «¿Qué es Xamarin?,» Documentación de Xamarin, 28 Mayo 2020. [En línea]. Available: <https://docs.microsoft.com/es-es/xamarin/get-started/what-is-xamarin>. [Último acceso: 3 Diciembre 2020].
- [12] DevExpress, «Use NuGet Packages to Install Office File API Components,» DevExpress.com, 19 Marzo 2021. [En línea]. Available: <https://docs.devexpress.com/OfficeFileAPI/400744/install-nuget-packages>. [Último acceso: 2 Mayo 2021].
- [13] D. Rivera, «Cómo implementar Test-Driven Development (TDD) sin morir en el intento,» NURSOFT, 5 Junio 2019. [En línea]. Available: <https://medium.com/nursoft/implementar-tdd-facilmente-4d2cfaa9172>. [Último acceso: 20 Mayo 2020].
- [14] S. Jensen, «Use FontAwesome in a Xamarin.Forms app,» medium.com/, 15 Febrero 2019. [En línea]. Available: <https://medium.com/@tsjdevapps/use-fontawesome-in-a-xamarin-forms-app-2edf25311db4>. [Último acceso: 18 Septiembre 2021].
- [15] Firebase, «Configura una app cliente de Firebase Cloud Messaging en Android,» Google, 2020 Octubre 4. [En línea]. Available: <https://firebase.google.com/docs/cloud-messaging/android/client?hl=es>. [Último acceso: 2021 Octubre 15].
- [16] G. Versluis, «Implement Push Notifications with FCM and Xamarin.Forms,» Youtube, 08 Junio 2021. [En línea]. Available:

<https://www.youtube.com/watch?v=7w2q2D6mR7g&list=PLfbOp004UaYX5B2pKKp5j7QPniyLMPkSO>. [Último acceso: 15 Octubre 2021].

- [17] Android Developers, «Desarrolladores de Android: Sugerencias de seguridad,» 15 Junio 2020. [En línea]. Available: <https://developer.android.com/training/articles/security-tips#StoringData>. [Último acceso: 25 Agosto 2021].

Anexos

Anexo 1: Checklist de requerimientos MASVS de OWASP modificado

Disponible en:

<https://docs.google.com/spreadsheets/d/1pTC5We653HOIqdFM90Yv2LsIBGG-uCOk/edit?usp=sharing&oid=104872747150509045657&rtpof=true&sd=true>

Anexo 2: Diagrama de componentes

Disponible en:

https://drive.google.com/file/d/1LGyK1dha_UkwkDTlibAvWEeenhS06Q5W/view?usp=sharing

Anexo 3: Algoritmo general de ejecución de prueba automatizada

Disponible en:

<https://drive.google.com/file/d/1cvvrJYkfABJGop4g9wdZJwEvMMm-L5uv/view?usp=sharing>

Anexo 4: Formato de encuesta sobre apariencia de pantallas

Disponible en:

https://drive.google.com/file/d/1Tb_0lyImDlelwB-lZb-ezsoe9tvQZKOY/view?usp=sharing

Anexo 5: Resultados de encuesta sobre apariencia de pantallas

Disponible en:

<https://drive.google.com/file/d/1bZTeaBalHIMj-Vg2tGMMIXFU1FRUy9X8/view?usp=sharing>