



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

TRANSFORMACIÓN TECNOLÓGICA DEL CANAL DE APLICACIÓN MÓVIL DE UNA COMPAÑÍA DE RETAIL

**TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN
TECNOLOGÍAS DE LA INFORMACIÓN**

ESTEBAN JOSIAS CANDIA BURGOS

PROFESOR GUÍA:
DANIEL PEROVICH GEROSA

MIEMBROS DE LA COMISIÓN:
CECILIA BASTARRICA PIÑEYRO
JOCELYN SIMMONDS WAGEMANN
JAIME NAVON COHEN

SANTIAGO DE CHILE
2020

Resumen

Diariamente las personas utilizan aplicaciones móviles para comunicarse, buscar inspiración, encontrar asistencia o comprar productos entre otros servicios. Ser *Mobile First* [1], es decir, pensar primero en teléfonos móviles y luego computadores de escritorio al momento de desarrollar productos, se ha transformado en un deber dentro de la industria del comercio electrónico minorista, y para Sodimac no ha sido la excepción.

La compañía tiene una aplicación móvil orientada a la asistencia y venta de productos y servicios. Esta aplicación presenta importantes falencias en la calidad de los datos que expone, en el valor que entrega a los usuarios a través de sus funciones y las constantes caídas o errores presentes en la plataforma. Por esto Sodimac decide realizar la reingeniería de su aplicación móvil.

Los problemas evidenciados a través de las valoraciones negativas de los usuarios en las tiendas de aplicaciones tienen su raíz en dos aspectos: el proceso de desarrollo de software y la plataforma tecnológica. Para resolver esta situación, por un lado, se implementa un nuevo proceso de desarrollo que organiza el trabajo permitiendo dar transparencia a la organización y acelerar la entrega de valor a los usuarios. Por otro lado, se desarrolla una nueva arquitectura, constituida por una capa de back-end que permite acceder frecuente y correctamente a los datos de la compañía y una capa de front-end unificada, que permite desarrollar en corto tiempo y con un bajo esfuerzo.

Para la validación de la solución se implementó un piloto que primeramente recogió y validó las necesidades de los usuarios, y las plasmó a una nueva versión de la aplicación móvil Sodimac. Esta nueva versión está basada en un nuevo lenguaje de front-end, una nueva capa de back-end y fue construida bajo un marco de trabajo centrado en la agilidad. Luego de la puesta en marcha del piloto, se validó cuantitativa y cualitativamente junto a stakeholders y a usuarios finales la efectividad del piloto, a través de indicadores como la cantidad de descargas, ventas, la evaluación de los usuarios en las tiendas de aplicaciones, la estabilidad de la plataforma frente a errores y el rendimiento del equipo de desarrollo.

Los logros de este proyecto de tesis fueron duplicar la puesta en producción de nuevas versiones mensuales de la aplicación móvil a través del nuevo proceso de desarrollo, la implementación de una nueva plataforma tecnológica que mejoró considerablemente la estabilidad de la aplicación y finalmente la valoración positiva de los usuarios de la aplicación móvil a través del incremento en las visitas, descargas, transacciones y valoraciones positivas en las tiendas de aplicaciones.

Dedicatoria

A mi hijo Pedro quien nació durante el proceso de documentación de la presente tesis, por ser mi mayor razón para continuar aprendiendo y mejorando como profesional y persona día a día.

A mi esposa y compañera Paulina, por su paciencia, compañía y amor, la cual me ha acompañado en todo el proceso de mi Magíster, dándome tiempo para estudiar y el apoyo para finalizar este programa.

A mis padres por el apoyo económico, espiritual, su guía durante toda mi vida como estudiante y por estar siempre presente en sus oraciones.

Tabla de contenido

1. Introducción	1
1.1 Contexto	1
1.2 Problema	1
1.2.1 Problemas de Procesos	2
1.2.2 Problemas Tecnológicos	2
1.3 Solución	2
1.3.1 Proceso	3
1.3.2 Tecnología	3
1.4 Objetivos	4
1.5 Metodología	4
1.5.1 Fase de descubrimiento	4
1.5.2 Fase estratégica	5
1.5.3 Fase de procesos	5
1.5.4 Fase de construcción	5
1.5.5 Fase de validación	6
1.6 Estructura del informe	6
2. Marco teórico	7
2.1 UX/UI	7
2.2 Desarrollo de aplicaciones móviles	8
2.3 Metodologías ágiles	9
2.3.1 Scrum como marco de trabajo	9
2.3.2 El equipo Scrum	10
2.3.3 Eventos	10
2.3.4 Artefactos	11
2.4 APIs versus Microservicios	12
2.5 CI/CD	13
2.5.1 ¿Cuál es la diferencia entre CI y CD (y la otra CD)?	13
2.5.2 Integración Continua	15
2.5.3 Distribución continua	16
2.5.5 Implementación continua	16
2.6 Product Vision Board (PVB)	16
2.7 Vistas y puntos de vista	18
2.7.1 Vistas arquitectónicas	19
2.7.2 Puntos de vista	20

2.7.3 Catálogo de puntos de vista	20
2.8 Principios arquitectónicos	22
2.9 ETL: <i>Extract, Transform and Load</i>	23
2.9.2 ELT: <i>Extract, Load and Transform</i>	24
2.10 El método KPT: <i>Keep, Problem, Try</i>.....	25
2.11 The Go Product Roadmap.....	26
2.11 Resumen	27
3. <i>Situación actual: problema</i>.....	28
3.1 Contexto de negocio	28
3.2 Aplicación actual	28
3.3 Proceso de desarrollo actual	30
3.3.1 Descripción	30
3.3.2 Evaluación cualitativa	34
3.3.3 Evaluación cuantitativa.....	36
3.3.4 Evaluación y desafíos del proceso	40
3.4 Arquitectura de la aplicación móvil	40
3.4.1 Descripción de la capa de back-end.....	40
3.4.2 Descripción de la capa de front-end	42
3.4.3 Evaluación y desafíos de la arquitectura	43
3.5 Análisis de la situación actual.....	51
3.5.1 Resultados: retrospectiva del proceso de desarrollo	53
3.5.2 Resultados: retrospectiva de la arquitectura.....	55
3.6 Resumen	58
4. <i>Solución</i>	59
4.1 Estrategia de negocio	59
4.2 Proceso de desarrollo.....	61
4.2.1 Definición del nuevo proceso	62
4.2.2 Una nueva metodología y enfoque	64
4.3 Arquitectura móvil de la compañía.....	65
4.3.2 Alcance de la nueva arquitectura	66
4.3.1 Objetivos de negocio y motivadores	68
4.3.3 Preocupaciones arquitectónicas.....	69
4.3.4 Principios para demostrar trazabilidad.....	69
4.3.5 Restricciones.....	71
4.3.6 Vista funcional	72
4.3.7 Vista de información.....	75
4.3.8 Vista de despliegue	79

4.3.9 Vista de desarrollo	82
4.4 Resumen	88
5. Validación	89
5.1 Piloto	89
5.1.1 Proto-personas	89
5.1.2 Viaje del cliente	90
5.1.3 Hoja de ruta	91
5.1.4 Desarrollo del MVP en Scrum	93
5.2 Retrospectiva del MVP	100
5.2.1 Resultados de la retrospectiva del MVP	101
5.2.2 Conclusiones	102
5.3 Validación por percepción de mejora.....	103
5.3.1 Protocolo	104
5.3.2 Ejecución.....	105
5.3.3 Resultados	105
5.3.4 Análisis de los Resultados	106
5.4 Validación cuantitativa	107
5.4.1 Descargas, audiencia y transacciones.....	107
5.4.2 Velocidad y Sprints	110
5.4.3 Valoración de los usuarios	113
5.4.4 Análisis de los resultados.....	115
5.5 Resumen	116
6. Conclusiones	117
6.1 Trabajo realizado	117
6.2 Impacto de la solución	118
6.3 Lecciones aprendidas	119
6.4 Trabajo futuro.....	119
Bibliografía	120
Anexos	127
Anexo A - Solución	127
Base para una estrategia: estado del comercio electrónico móvil global	127
Definición de la estrategia de negocio móvil	130
Definición de objetivos de negocio y motivadores	131
Descubrimiento de preocupaciones arquitectónicas.....	132
Estableciendo trazabilidad.....	136
Ejemplo de la estructura de la API Catálogo de productos.....	136
Anexo B - Validación	138

Backlog del MVP	138
Diseño de la PDP	139
Ejemplo de historia de usuario	140
Encuesta A, personas transversales	141
Encuesta B, personas nuevas.....	149
Resultados cualitativos del nuevo proceso de desarrollo y arquitectura	157

Índice de figuras

Figura 1 - Flujo de trabajo de Scrum [14]	12
Figura 2 - Aplicación monolítica versus aplicación basada en microservicios [16].....	13
Figura 3 - Continuous Integration, Continuous Delivery y Continuous Deployment [17]	14
Figura 4 - Ejemplo de un Product Vision Board [18].....	17
Figura 5 - Investigación y validación con el Product Vision Board [18]	18
Figura 6 - Puntos de vista [20]	20
Figura 7 - Relaciones entre vistas [20].....	22
Figura 8 - Usando principios para demostrar trazabilidad [21]	23
Figura 9 - Proceso ETL [22]	24
Figura 10 - Proceso ELT [22]	25
Figura 11 - Estructura gráfica del método de retrospectiva KPT (Keep, Problem, Try) [23].....	26
Figura 12 - Esquema de navegación y funciones de la actual aplicación Sodimac	29
Figura 13 - Equipo de desarrollo actual de la aplicación móvil	32
Figura 14 - Flujo del proceso actual de desarrollo de la aplicación móvil	34
Figura 15 - Hoja de ruta de desarrollo para la aplicación Sodimac, segundo semestre 2016	38
Figura 16 - Cantidad de funciones comprometidas versus terminadas durante junio a diciembre de 2016	39
Figura 17 - Arquitectura actual de la aplicación móvil	41
Figura 18 - Promedio de evaluación mensual junio a diciembre de 2016, Android [30]	43
Figura 19 - Promedio de evaluación mensual junio a diciembre de 2016, iOS [31]	43
Figura 20 - Evaluación iOS: "no está en línea con la web" [31]	44
Figura 21 - Evaluación iOS: "Insuficiente" [31]	45
Figura 22 - Evaluación iOS: "Se cierra la app" [31]	45
Figura 23 - Evaluación iOS: "Pésimo diseño y funcionalidad" [31]	46
Figura 24 - Evaluación Android: "Al principio estaba bien..." [30]	46
Figura 25 - Evaluación Android: "Error en el ordenado de la lista" [30].....	47
Figura 26 - Evaluación Android: "No sirve" [30]	47
Figura 27 - Evaluación Android: "Problemas con códigos terminados en x" [30]	47
Figura 28 - Evaluación Android: "Inestable e inexacta" [30]	48

Figura 29 - Errores (bloqueos y excepciones) en la Aplicación Sodimac, iOS, 2016 [31].....	50
Figura 30 - Errores en App Sodimac, iOS de junio a diciembre de 2016 [33]	50
Figura 31 - Pronóstico de gasto mundial total en aplicaciones móviles [37]	60
Figura 32 - PVB de la aplicación móvil.....	61
Figura 33 - Organización de equipos Scrum en la compañía.....	64
Figura 34 - Canal CI/CD del desarrollo de la aplicación móvil	65
Figura 35 - Relaciones entre vistas	66
Figura 35 - Diagrama de contexto arquitectónico.....	67
Figura 36 - Vista funcional del sistema de la aplicación móvil	73
Figura 37 - Vista de información del sistema de la aplicación móvil.....	76
Figura 38 - Vista de despliegue del sistema de la aplicación móvil	80
Figura 39 - Tipos de proyectos disponibles en Ionic V3 [44]	84
Figura 40 - Estructura de un proyecto por defecto de Ionic V3	84
Figura 41 - Estructura del directorio app/ de un proyecto por defecto de Ionic V3	85
Figura 42 - Proceso de publicación en App Store (iOS) [45].....	86
Figura 43 - Resumen de Proto personas.....	90
Figura 44 - Viaje del cliente	91
Figura 45 - Hoja de ruta 2017 de la aplicación móvil	93
Figura 46 - Dinámica de trabajo Scrum	94
Figura 47 - Flujo de desarrollo de Back-end y Front-end	95
Figura 48 - parte del Sprint 4 dedicado a desarrollar la PDP.....	96
Figura 49 - Controlador Products de la API Catálogo de productos	97
Figura 50 - Respuesta del microservicio GET Products.....	97
Figura 51 - Respuesta del servicio de origen: getProductDetails de SL2.....	98
Figura 52 - Flujo de información desde back-end externo a front-end la aplicación móvil	99
Figura 53 - Fases en equipos de desarrollo [50]	103
Figura 54 - Conjunto de entrevistados para validación cualitativa	105
Figura 55 - Instalación por dispositivos Android semanal, 1 enero al 31 de diciembre de 2017 [30]	107
Figura 56 - Instalación por dispositivos iOS semanal, 1 enero al 31 de diciembre de 2017 [31]	108
Figura 57 - Usuarios por sistema operativo [33]	108

Figura 58 - Indicadores claves de transacciones, venta y tasa de conversión [33]	109
Figura 59 - Evolutivo de visitas a la aplicación móvil desde mayo a diciembre de 2017 [33]	110
Figura 60 - Flujo de trabajo desde el sprint hasta la publicación de una aplicación móvil.....	111
Figura 61 - Gráfico evolutivo de versiones publicadas de la aplicación móvil.....	112
Figura 62 - Gráfico Burndown del MVP	113
Figura 63 - Calificaciones a lo largo del tiempo en Play Store (Android) [30]	113
Figura 64 - Valoraciones en Play Store (Android) sobre la nueva versión de la Aplicación Móvil [30]	114
Figura 65 - La oportunidad de las aplicaciones móviles: share de transacciones [56]	128
Figura 66 - La oportunidad de las aplicaciones móviles: conversión [56]	129
Figura 67 - Backlog de la aplicación móvil	138
Figura 68 - Vista de front-end de la página de detalle de producto.....	139
Figura 69 - Historia de usuario MVP-85: Diseñar PDP	140

Índice de tablas

Tabla 1 - Protocolo para la evaluación FODA	35
Tabla 2 - Análisis FODA del proceso actual de desarrollo de la aplicación móvil	36
Tabla 3 - Tipologías de errores sobre comentarios de usuarios en las tiendas de aplicaciones	49
Tabla 4 - Retrospectiva KPT – Proceso de desarrollo y arquitectura actual	52
Tabla 6 - Ítems de acción (Try) del proceso de desarrollo.....	62
Tabla 8 - Objetivos y motivadores de negocio para el diseño de la nueva arquitectura.....	68
Tabla 14 - Restricciones arquitectónicas.....	72
Tabla 15 - Responsabilidad de elementos, vista funcional.....	74
Tabla 17 - Trazabilidad de decisiones arquitectónicas para la vista funcional	75
Tabla 18 - Detalle del flujo de datos en la vista de información	78
Tabla 19 - Trazabilidad de decisiones arquitectónicas para la vista de información.....	79
Tabla 20 - Nodos de la vista de despliegue	81
Tabla 21 - Trazabilidad de decisiones arquitectónicas para la vista de despliegue.....	82
Tabla 22 - Trazabilidad de decisiones arquitectónicas para la vista de desarrollo.....	87
Tabla 23 - Definición de la hoja de ruta del MVP	92
Tabla 24 - Funciones de la aplicación móvil definidas para el MVP	92
Tabla 25 - Protocolo para la retrospectiva KPT del MVP	100
Tabla 26 - Protocolo para las encuestas de validación del MVP al equipo de desarrollo y stakeholders	104
Tabla 27 - Resumen de encuesta sobre el logro de los objetivos.....	106
Tabla 28 - Resumen de encuesta sobre la adopción del nuevo proceso y arquitectura	106
Tabla 5 - Protocolo para la definición de la estrategia móvil	130
Tabla 7 - Protocolo para la definición de objetivos de negocio y motivadores.....	131
Tabla 9 - Protocolo para el descubrimiento de preocupaciones arquitectónicas	132
Tabla 10 - Principio arquitectónico P1.....	133
Tabla 11 - Principio arquitectónico P2.....	134
Tabla 12 - Principio arquitectónico P3.....	135
Tabla 13 - Estableciendo trazabilidad.....	136
Tabla 16 - Método de disponibilidad de productos en la API catálogo de producto	137

Tabla 29- Resumen de encuesta a usuarios transversales sobre el nuevo proceso de desarrollo.....	159
Tabla 30 - Resumen de encuesta a personas nuevas sobre el nuevo proceso de desarrollo.....	162
Tabla 31 - Resumen de encuesta a personas transversales sobre la nueva arquitectura	164
Tabla 32 - Resumen de encuesta a personas nuevas sobre la nueva arquitectura.....	167

1. Introducción

1.1 Contexto

Homecenter Sodimac es parte del holding Falabella y está enfocado en el mejoramiento del hogar y la construcción. Tiene más de 30 años en Chile y actualmente tiene presencia en 7 países de América Latina. La compañía tiene en su haber numerosos sistemas, procesos y herramientas que soportan la operación diaria de ventas y asistencia, no solo en un contexto físico sino también digital, donde hoy es uno de los líderes en el posicionamiento de contenidos digitales en términos de mejoramiento del hogar en Chile.

Entre los distintos canales de ventas y asistencia de Sodimac, existe un gran contraste en cuanto a la tecnología y a los procesos implementados en cada uno de ellos. Hay canales físicos como las tiendas (Sodimac Homecenter, Sodimac Constructor y Homy) y también virtuales como Sodimac.com (en sus formatos web desktop y web móvil), Homy.cl, VAT (venta asistida en tienda), venta a empresas y fono ventas. Sin embargo, hay un canal digital que nunca ha logrado establecerse vendiendo o prestando asistencia a clientes: el canal de la aplicación móvil Sodimac.

Hoy en día no existe la venta de productos o servicios a través de la aplicación Sodimac, las funcionalidades de asistencia a clientes carecen de coherencia respecto a los datos y lógicas de negocio de la compañía, y son evaluadas negativamente por sus usuarios en las tiendas de aplicaciones móviles. Muchos de estos problemas tienen su origen en el desconocimiento que tiene la compañía sobre los usuarios de la aplicación móvil y este tipo de tecnología, lo que, sumado a la inexistencia de una estrategia de negocio, plasmada en una hoja de ruta que guíe el desarrollo de la aplicación móvil, ha causado el bajo uso y las evaluaciones negativas.

Por años, el contar con una aplicación móvil que facilite la experiencia de los clientes frente a la asistencia y venta de productos ha sido una deuda que la compañía tiene consigo misma, con la tecnología y con sus clientes.

1.2 Problema

La compañía nunca ha logrado posicionarse positivamente en el mercado de las aplicaciones móviles en términos de uso, venta y la valoración positiva de los usuarios. La causa de esto radica en varios ámbitos, tanto en los procesos como en la tecnología involucrada, tal como se detallan a continuación.

1.2.1 Problemas de Procesos

Los principales problemas asociados al proceso de desarrollo de la aplicación móvil son:

- En la compañía existe un responsable del área de negocio para la aplicación móvil, quien define las funcionalidades y su prioridad de desarrollo, y un equipo de TI a cargo del desarrollo del software. Sin embargo, no existe un proceso formal de desarrollo con etapas y roles claramente definidos para la ejecución de los desarrollos y su mejora continua. Al no tener un proceso definido, éste se desarrolla basado en ensayo y error, lo cual ha generado pérdida de tiempo, dinero, duplicidad de roles, indefiniciones y funcionalidades mal evaluadas por los usuarios.
- El tiempo de puesta en producción hoy es lento, toma casi un mes el desarrollar una versión de la aplicación móvil por cada sistema operativo: Android (Google) e iOS (Apple) a nivel de front-end. Por otra parte, no existe la automatización de procesos como compilación, pruebas o tareas para la publicación de nuevas versiones en las tiendas de aplicaciones.

1.2.2 Problemas Tecnológicos

A continuación, se describen los principales problemas asociados a la tecnología que da soporte a la aplicación móvil:

- No existe una capa de back-end robusta y de alta disponibilidad que se comunique con los distintos sistemas de la compañía y que dé soporte a la aplicación móvil. Esto ha generado grandes problemas, como diferencias de información debido a procesos manuales de sincronización entre la aplicación móvil y los sistemas de la compañía.
- Hoy en día, el código fuente de la capa de front-end de la aplicación móvil ha sido modificado por al menos cinco programadores distintos en las últimas 10 versiones, lo que ha provocado variadas inconsistencias, múltiples caídas (cierres inesperados en la aplicación) y problemas de rendimiento, como demora excesiva en tiempos de carga de imágenes, textos y contenido en general (más de 30 segundos).

1.3 Solución

Para abordar los problemas que enfrenta la compañía frente a su aplicación móvil, se implementa un nuevo proceso de desarrollo de productos centrado en la agilidad, dada la experiencia de los miembros del equipo en este tipo de metodologías y además la necesidad de contar con un proceso de desarrollo de software que permita rápidamente liberar versiones para obtener retroalimentación de los usuarios. Por otro lado, se construye una nueva plataforma tecnológica

que entrega soporte al desarrollo, operación y mejora continua de la aplicación móvil.

El autor del presente trabajo de tesis es el encargado de la aplicación móvil dentro de la compañía. Su responsabilidad principal refiere a resolver problemas de negocio, procesos y tecnología. Para ello, se realizarán las siguientes actividades ejecutadas por equipos bajo su dirección.

1.3.1 Proceso

A continuación, se describen iniciativas de solución referente al proceso de desarrollo de la aplicación móvil:

- Dada la experiencia anterior de los miembros del equipo, se establece un proceso de desarrollo ágil que formaliza la forma de trabajo, creando equipos con roles definidos y experiencia en el desarrollo de proyectos de software en aplicaciones móviles.
- Se implementa la automatización de tareas repetitivas o manuales en el desarrollo, como la ejecución de pruebas, publicación en repositorios, compilación y puesta en producción.

1.3.2 Tecnología

A continuación, se describen las iniciativas de solución tecnológica para las capas de front-end y back-end de la aplicación móvil:

- Se desarrolla una arquitectura móvil en la nube, mediante APIs y microservicios que soportan la comunicación con los sistemas de la compañía, enfocándose en el rendimiento, estabilidad, la elasticidad necesaria para escalar recursos dinámicamente y robustez frente a distintos escenarios. A su vez, la nueva arquitectura asegura la coherencia de la información entre los sistemas de la compañía y la aplicación móvil.
- Los equipos de desarrollo realizan la reingeniería de la capa de front-end de la aplicación móvil, consistente en unificar el desarrollo en un solo código fuente, que luego pueda ser llevado a iOS y Android sin particularidades. Este desarrollo será híbrido, lo cual significa una mezcla entre una solución web y una solución nativa [2], donde el centro de las aplicaciones es construido usando tecnología web: HTML, CSS y JavaScript, los cuales son encapsulados dentro de una aplicación nativa y donde a través del uso de plugins, estas aplicaciones pueden acceder completamente a las características de los teléfonos móviles. El desarrollo híbrido genera además aplicaciones móviles flexibles, que no necesitan una puesta en producción nueva para publicar funcionalidades o contenidos adicionales, acelerando el desarrollo, los controles de calidad y disminuyendo la cantidad de errores.

1.4 Objetivos

El objetivo general del presente proyecto de tesis es crear en la compañía un canal de aplicación móvil valorado positivamente por sus usuarios, donde puedan comprar y ser asistidos en sus proyectos hogar y construcción.

Los objetivos específicos que permiten alcanzar el objetivo general son:

- Definir e implementar el proceso de desarrollo de software que acelere el tiempo de puesta en producción y permita entregar valor a los usuarios en cada versión de la aplicación.
- Implementar la plataforma tecnológica que dé soporte al desarrollo y a la operación de la aplicación móvil.
- Utilizando el nuevo proceso de desarrollo y la nueva plataforma tecnológica, desarrollar una aplicación móvil como producto mínimo viable que logre ser valorada positivamente por sus usuarios.

1.5 Metodología

La metodología que se utiliza para abordar el presente proyecto de tesis consta de cinco fases, las cuales se detallan a continuación y están centradas en desarrollar una nueva versión la aplicación móvil a través de un nuevo proceso de desarrollo y una nueva plataforma tecnológica.

1.5.1 Fase de descubrimiento

En esta primera fase se realiza un levantamiento de los usuarios de la aplicación móvil, entendiendo cómo se comportan frente a la aplicación actual y cómo se relacionan con la compañía en otros canales y puntos de contacto. Se abordan sus preocupaciones, necesidades, metas y tipos de persona. Para ello, se realizan las siguientes actividades:

- **Análisis y comprensión de los arquetipos de usuarios y clientes.** Se revisa un estudio realizado por el equipo de experiencia de usuario de la Sodimac durante 2016 sobre los tipos de personas que actualmente navegan en la aplicación móvil y los potenciales nuevos usuarios que podrían llegar a utilizarla. Esto da claridad de quiénes son el público objetivo.
- **Levantamiento de necesidades y oportunidades.** Se detectan cuáles son los puntos dentro del viaje del cliente en donde la aplicación móvil puede resolver una necesidad o enfrentar una oportunidad.

1.5.2 Fase estratégica

En esta fase se establecen las metas que la compañía quiere alcanzar con la aplicación móvil desde el punto de vista de negocio, considerando los recursos internos, el contexto externo a la compañía y las acciones que se estiman necesarias para cumplir las metas definidas. Para ello, se realizan las siguientes actividades:

- **Desarrollo de estrategia de negocio.** Se define la misión, visión, objetivos e indicadores de logro de objetivos (KPIs) del producto.
- **Definición de funciones y hoja de ruta.** Se detallan las funciones básicas que debe tener la nueva aplicación móvil con foco en la venta y asistencia, se priorizan en base a las necesidades y oportunidades levantadas desde los usuarios y se define la hoja de ruta para su desarrollo.

1.5.3 Fase de procesos

Esta fase se centra en los cambios culturales de la organización, como la adopción de un nuevo proceso de desarrollo de software para la aplicación móvil centrado en la agilidad, realizando las siguientes actividades:

- **Adopción de metodologías ágiles.** Se establece un proceso de desarrollo ágil de software el cual incorpora una definición de roles, artefactos y eventos.
- **Automatización.** Se desarrolla un canal de integración, despliegue e implementación continua para acelerar los tiempos de desarrollo, pruebas, integración, distribución e implementación en producción.

1.5.4 Fase de construcción

La fase de construcción está enfocada en desarrollar la capa de front-end de la nueva aplicación móvil y la capa de back-end que dará soporte a la operación de la aplicación.

- **Desarrollo de la infraestructura para la aplicación móvil.** Se desarrolla una nueva capa de back-end la cual da soporte a la aplicación móvil. Para ello, se adquiere el servicio de Microsoft Azure como solución de servidores virtuales en la nube. Además, se desarrollan APIs que expondrán a la aplicación móvil los datos que llegarán desde los microservicios que a su vez extraerán estos datos directamente desde los distintos sistemas legados de la compañía, ya sea en línea o a través de cargas de datos.

- **Desarrollo de front-end unificado.** Se desarrolla una nueva capa de front-end utilizando Ionic V3, un SDK híbrido de desarrollo de aplicaciones móviles basado en Angular, que reemplaza los desarrollos nativos actuales, con la idea de unificar el código fuente de la aplicación móvil en uno solo.

1.5.5 Fase de validación

Para la fase de validación se prueba el proceso y la tecnología desarrollada, apoyada por la estrategia de negocio. Esta fase se evalúa con la idea de mejorar continuamente para entregar un producto valorado positivamente por sus usuarios.

- **Desarrollo de un MVP.** Para validar el nuevo proceso de desarrollo y arquitectura, se construyen las funcionalidades básicas de una aplicación móvil para la compra de productos y servicios, además de funciones que entreguen asistencia en tiendas sobre los productos, obteniendo así un producto mínimo viable (MVP por sus siglas en inglés: *Minimum Viable Product*).
- **Evaluación.** Se revisan indicadores asociados al uso de la aplicación móvil desarrollada en el MVP como valoración, visitas, descargas, transacciones e indicadores cuantitativos y cualitativos asociados al nuevo proceso de desarrollo y a la nueva plataforma tecnológica implementada.

1.6 Estructura del informe

Este trabajo de tesis está estructurado de la siguiente forma. En el Capítulo 2 se explican los conceptos sobre los cuales se basa el trabajo de tesis, entregando una base de conocimiento para entender cada capítulo posterior. El Capítulo 3 plantea los problemas y limitantes del proceso de desarrollo y arquitectura inicial de la aplicación móvil. En el Capítulo 4 se describe la solución a los problemas planteados, la cual radica en implementar un nuevo proceso de desarrollo y una nueva arquitectura, ambos guiados por una estrategia de negocio. El Capítulo 5 documenta la validación de la solución a través de tres aspectos, el primero es el desarrollo un piloto, una nueva versión de la aplicación móvil, la cual utiliza el nuevo proceso de desarrollo y arquitectura. El segundo es la validación por parte de los stakeholders, basadas en su percepción. El tercero es una validación cuantitativa de los principales indicadores claves por parte de los usuarios de la aplicación. El Capítulo 6 presenta las conclusiones y la propuesta de trabajo a futuro.

2. Marco teórico

Para llevar a cabo la solución propuesta, se aplican métodos y técnicas de diferentes áreas de conocimiento de las Tecnologías de la Información. Se presenta a continuación una reseña de los conceptos fundamentales a aplicar, como a la experiencia que enfrenta el usuario en la aplicación móvil a través del diseño gráfico, diseño de la experiencia, flujo y transiciones de contenido y pantallas, lo cual se plasma a través de UX/UI. Se discutirá además sobre el desarrollo de las aplicaciones móviles como concepto de producto y sus particularidades respecto a otras plataformas. Por otra parte, se abordará el desarrollo de software basado en agilidad, en particular sobre el enfoque Scrum, el cual es utilizado por el equipo de Sodimac. Se relatará también el significado y diferencias entre APIs y Microservicios como elementos de una arquitectura de software.

Otros conceptos importantes como CI/CD, ETL, PVB también serán revisados en este capítulo, los cuales constituyen la base para la automatización de tareas (como CI/CD), la gestión de la información (ETL) y como abordar la estrategia de negocio de productos (PVB).

También este capítulo incluirá aspectos de la arquitectura de software como las vistas y puntos de vista y además como dar trazabilidad a las decisiones arquitecturas que se tomen desde los objetivos y motivadores de negocio.

2.1 UX/UI

UX corresponde al acrónimo de *User Experience* o experiencia de usuario en donde el foco es el usuario y la experiencia que se quiere lograr con él a través de un producto o servicio [3]. Antes de diseñar una aplicación móvil, es necesario comprender en primer lugar a los posibles usuarios y sus verdaderas motivaciones o necesidades, para luego considerar desde ese lugar, qué interfaz, qué contenidos y qué interacciones lograrán los resultados buscados y definidos en la estrategia de la compañía. Finalmente, es necesario validar con usuarios finales los resultados que produce la interfaz propuesta.

Una aproximación a la identificación de quiénes son los usuarios es realizar un estudio que defina arquetipos de persona [4], los cuales podrían eventualmente utilizar el producto o servicio. Para ello, se definen personas ficticias con nombres, gustos, necesidades, comportamientos y objetivos. Esto ayuda a tomar decisiones de diseño y acercarse a las personas reales que utilizarán la aplicación.

UI significa *User Interface* o interfaz de usuario y refiere al diseño de la interfaz que el sistema expone al usuario [5]. No refiere simplemente al diseño gráfico, sino al trabajo de disponer de manera estratégica todos los elementos que conforman el contenido visual al que se enfrenta el usuario para que la aplicación sea atractiva y coherente en la comunicación y experiencia que se desea lograr.

2.2 Desarrollo de aplicaciones móviles

El mercado de las aplicaciones móviles ha evolucionado mucho durante la última década [6]. Es por ello por lo que las compañías que tienen aplicaciones móviles deben aprender rápidamente a adaptarse a los cambios de expectativas y condiciones del entorno, con el objetivo de mantenerse competitivos. Por otra parte, un cambio de paradigma de una compañía tradicional a una centrada en aplicaciones móviles es lento [7], reinventarse y ser parte de un nuevo terreno toma tiempo y debe abordarse de forma planificada. Un ejemplo de esto es Facebook. Cuando lanzó su primera versión en formato aplicación móvil el fracaso fue inmediato, debido a que como compañía se enfocaron en la tecnología, aprendiendo rápidamente lenguajes de programación de aplicaciones móviles para liberar pronto una versión al mercado, sin detenerse en detalles particulares de este tipo de plataforma. En esto Facebook falló, porque el cambio de paradigma no es meramente tecnológico sino también cultural [7].

Las aplicaciones móviles están sujetas a reglas de publicación por parte de las tiendas de aplicaciones móviles, y puede tomar incluso semanas implementar un cambio o corregir un defecto. A Facebook el fracaso le significó traer nuevas personas a sus equipos (con conocimientos y experiencia en aplicaciones móviles), cambiar las metodologías de desarrollo y adquirir un portafolio de compañías con experiencia en aplicaciones móviles como Instagram y WhatsApp para adquirir conocimiento.

En términos de desarrollo, hoy existen 2 sistemas operativos líderes: Android de Google e iOS de Apple. En términos técnicos, ambos son complejos y distintos entre sí. Existen diferencias desde lo meramente gráfico hasta lo funcional. Un simple, pero no trivial ejemplo es la acción “volver atrás”. En Android puede un botón físico, parte del teléfono, por lo cual, no es obligatorio incluir en una aplicación un enlace para regresar a la pantalla anterior. En el caso de iOS, este botón físico no existe, por lo que siempre debe haber un enlace para ir atrás (por lo general ubicado en la parte superior izquierda de una aplicación).

La complejidad de que existan dos sistemas operativos (Android e iOS) radica en que para una misma aplicación móvil se requiera de dos desarrollos particulares, sin mencionar la posibilidad de introducir errores en dos códigos fuente distintos. Para mitigar todo lo anterior es que existen la alternativa del desarrollo híbrido de aplicaciones móviles [8], el cual permite desarrollar un único código fuente, con la opción de integrar características particulares de cada sistema operativo, con las cuales los usuarios están acostumbrados a operar en el día a día frente otras aplicaciones (como el ejemplo del botón o enlace volver atrás), para finalmente exportar una versión Android o una versión iOS.

Por otro lado, existen también las *Progressive Web Apps (PWA)*, las cuales a través de modernas capacidades web, entregan una experiencia tipo aplicación móvil nativa a los usuarios [9],

permitiendo el envío de notificaciones push, cargar contenido sin necesitar Internet, acceder al hardware del teléfono como GPS, cámara, micrófono, altavoz y otros.

Un caso de éxito respecto a las PWA es Alibaba, la compañía asiática deseaba entregar una gran experiencia a sus dos segmentos de usuarios móviles: la aplicación nativa (usuarios leales a la marca) y sitio web móvil (visitantes por primera vez, los cuales están conociendo a la marca). Sobre estos últimos es que había una pérdida del interés en volver a visitar el sitio, por lo que optaron por desarrollar una PWA para poder generar usuarios leales a la marca a través de las funciones mejoradas y personalizadas, como las notificaciones push, poder agregar la aplicación al escritorio del teléfono y derivar tráfico desde el sitio hacia la aplicación nativa con banners promocionales con links directos a descargar o abrir la aplicación nativa. Este desarrollo generó un 76% más de conversión en todo el sitio, un 14% más de usuarios mensuales en iOS y un 30% más en Android [10].

2.3 Metodologías ágiles

Construir aplicaciones flexibles, de calidad y que se adapten rápidamente a las demandas del mercado es importante. Sin embargo, también es importante tener un proceso y una organización que lo habilite. Es por eso por lo que las metodologías de desarrollo de productos (no sólo y exclusivamente software) basadas en la agilidad están diseñadas para iterar con rapidez, liberar frecuentemente versiones de un producto o característica a los usuarios y demostrar empíricamente su valor [11].

Contar con una base como Scrum [12], permite construir y validar funciones con usuarios finales en un par de semanas, lo cual da a la organización cierto control sobre el cambio, lo que es difícil de lograr con otros tipos de desarrollo tradicionales como los basados en el ciclo de vida en cascada [13].

2.3.1 Scrum como marco de trabajo

Scrum es un marco de trabajo para desarrollar, entregar y mantener productos complejos, en donde las personas pueden abordar problemas complejos adaptativos a la vez que entregan productos del máximo valor posible productiva y creativamente [12].

Son 3 los pilares que soportan la implementación del control de procesos empírico:

- **Transparencia:** los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. Los observadores deben compartir un entendimiento común de aquello que ven.
- **Inspección:** Los usuarios de Scrum deben constantemente inspeccionar los aspectos del proceso y su progreso hacia el objetivo para detectar tempranamente variaciones indeseadas.

- **Adaptación:** Si se detectan aspectos del proceso que se desvían de los límites aceptables y hacen que el producto resultante sea inaceptable, el proceso o material que está siendo procesado debe adaptarse.

Este enfoque cuenta con distintos aspectos para su adopción. A continuación se detalla cada uno, los que serán aplicados íntegramente por el equipo de la aplicación móvil:

2.3.2 El equipo Scrum

Está compuesto por los siguientes roles:

EQ1. Product Owner: o dueño del producto, es el responsable de maximizar el valor del producto resultante del trabajo del equipo de desarrollo, entendiendo las necesidades de usuarios, stakeholders.

EQ2. Equipo de desarrollo: profesionales que realizan el trabajo de entregar un incremento terminado del producto. Son auto-organizados y multifuncionales.

EQ3. Scrum Master: es responsable de promover y apoyar Scrum en términos de ayudar a toda la organización a entender la teoría, prácticas, reglas y valores del marco de trabajo.

2.3.3 Eventos

Estos eventos predefinidos existen con la finalidad de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son bloques de tiempo definidos bajo una duración máxima.

Cada uno de los eventos del Scrum constituye una oportunidad de inspección y adaptación según algún aspecto, siendo creados para habilitar los pilares de transparencia e inspección.

EV1. Sprint: el principal evento del Scrum es un bloque de tiempo de un mes o menos (en el caso del equipo de la aplicación móvil, se definieron 2 semanas), en donde se crea un incremento del producto terminado utilizable y potencialmente desplegable a los usuarios finales.

EV2. Sprint planning: el trabajo realizado durante el Sprint se planifica durante este evento mediante el trabajo colaborativo del equipo Scrum completo. Durante este evento se define lo que se podría entregar como un incremento resultante del Sprint que está por comenzar y además se define como se conseguirá hacer este trabajo.

EV3. Daily Scrum: es una reunión de una duración máxima de 15 minutos para el equipo de

desarrollo, en donde cada miembro responde 3 preguntas: ¿qué hice ayer?, ¿qué haré hoy? y ¿hay algo que me impide avanzar?

EV4. Sprint Review: al final del sprint se lleva a cabo una revisión del Sprint para inspeccionar el incremento y adaptar la lista del producto (Product Backlog) si fuese necesario. A este evento pueden ser invitados los stakeholders. Su finalidad es facilitar la retroalimentación y fomentar la colaboración.

EV5. Sprint Retrospective: es una oportunidad para que el equipo Scrum se inspeccione a sí mismo y evalúe los aspectos positivos, aquellos que no se hicieron bien y aquellas acciones que podrían realizar en el siguiente o futuros Sprints.

2.3.4 Artefactos

Los artefactos del Scrum representan el trabajo o valor generado en diversas formas, las cuales son útiles para demostrar transparencia y oportunidades para la inspección y adaptación.

AF1. Product Backlog: es una lista ordenada de todo lo que es necesario en el producto.

AF2. Sprint Backlog: corresponde al listado seleccionado de elementos del Product Backlog para el Sprint, más un plan para entregar el incremento del producto y conseguir el objetivo del Sprint.

AF3. Incremento: es la suma de todos los elementos del Product Backlog completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores.

El funcionamiento de Scrum durante un Sprint es bastante sencillo. Durante el Sprint Planning se revisan las historias existentes y priorizadas en el Product Backlog por el equipo de desarrollo para evaluar cuáles (según su capacidad), pueden tomar en la próxima iteración o Sprint. Una vez que se tienen todas las historias a tomar, estas pasan a constituir el Sprint Backlog. Luego, el equipo define tareas a partir de estas historias, para a continuación dar inicio al Sprint. Durante este periodo de tiempo, el equipo de desarrollo se reúne diariamente a revisar el estado del Sprint (Daily Sprint).

Una vez que el tiempo se acaba, el equipo Scrum se reúne a revisar el incremento desarrollado en una reunión llamada Sprint Review. Ahí se decide si el incremento es puesto en producción o no y se muestra a los stakeholders los avances del producto. Luego de esta reunión, tiene lugar el Sprint Retrospective en el que participa solamente el equipo Scrum para evaluar el proceso y el equipo para mejorar en la siguiente iteración. La Figura 1 resume el flujo de trabajo en Scrum.

SCRUM FRAMEWORK

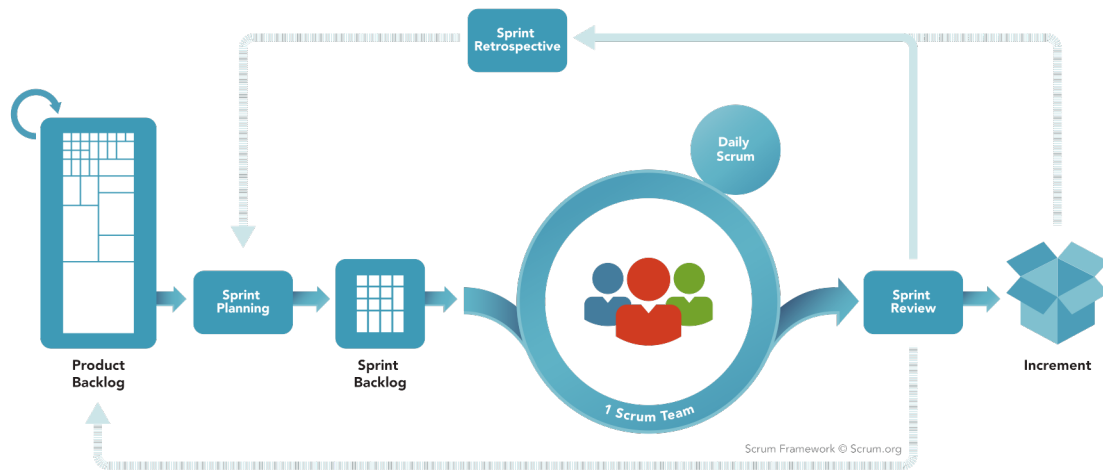


Figura 1 - Flujo de trabajo de Scrum [14]

2.4 APIs versus Microservicios

En frecuente leer en foros de tecnología o blogs ideas equivocadas frente a lo que significan los microservicios o APIs, por ejemplo: “los microservicios son servicios web, pero de alta granularidad”, “las APIs son microservicios” o “los microservicios son la implementación de una API” [15]. Pero las verdaderas diferencias entre estos dos conceptos son mucho más simples.

Una API (*Application Programming Interface*, por sus siglas en inglés) es simplemente una interfaz, un mecanismo por el cual se pueden obtener datos desde una aplicación o generar cambios en estos datos. Las APIs web modernas como las API REST, usan protocolos HTTP con JSON u XML para la carga o exposición de datos y por definición, desacoplan la llamada de la implementación, esto quiere decir que quien se comunica con la API no debiera tener idea de la implementación que hay debajo de ésta.

Por otro lado, la arquitectura basada en microservicios, es un enfoque alternativo para estructurar aplicaciones, dividiéndolas en componentes más pequeños y completamente independientes, lo que les permite tener mayor agilidad, escalabilidad y disponibilidad

En resumen, los microservicios y las APIs están relacionadas, pero no son lo mismo. Las APIs son

interfaces mientras que los microservicios son un enfoque de arquitectura de aplicaciones [16]. Esto puede verse ejemplificado en la Figura 2, donde a la izquierda de la imagen se muestra un único gran componente que es parte de una aplicación monolítica (x1) y expone 4 APIs (x4). Luego, a la derecha, se puede ver como múltiples componentes independientes y pequeños (x3), que son parte de una aplicación orientada a microservicios, exponen cada uno una o varias APIs por sí mismos (x4), en el fondo ambos enfoques de arquitectura pueden exponer APIs.

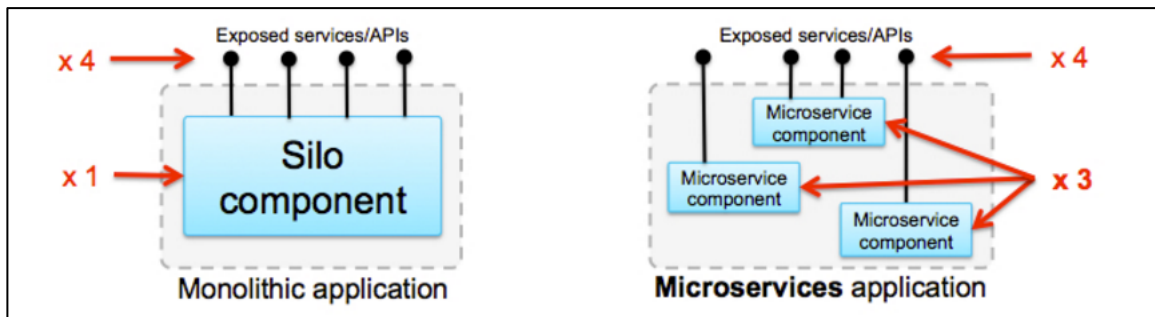


Figura 2 - Aplicación monolítica versus aplicación basada en microservicios [16]

2.5 CI/CD

La integración continua/distribución continua CI/CD (por sus siglas en inglés *Continuous Integration/Continuous Delivery*) es un método para distribuir frecuentemente software a los usuarios mediante el uso de la automatización en etapas de desarrollo del software [17]. sus principales conceptos son: la *integración continua*, la *distribución continua* y la *implementación continua*.

CI/CD incorpora la automatización continua y un control permanente en todo el ciclo de vida del software, desde las etapas de integración y pruebas, hasta las de distribución e implementación. Este conjunto de prácticas se conoce como “canal CI/CD”.

2.5.1 ¿Cuál es la diferencia entre CI y CD (y la otra CD)?

La sigla CI/CD tiene diferentes significados. La “CI” en CI/CD siempre se refiere a la *integración continua*, un proceso de automatización para los programadores, donde si los cambios realizados en el código tienen éxito en términos de pruebas, luego pueden ser combinados en un repositorio compartido con otros programadores, solucionando el problema de contar con muchas versiones de un software al mismo tiempo, creando conflictos entre sí.

“CD” en CI/CD se puede referir a la *distribución* o *implementación continua*, ambos son conceptos

relacionados que suelen utilizarse indistintamente, ambos se refieren a la automatización de las etapas posteriores del canal, pero a veces se usan por separado para explicar la cantidad de automatización que se está incorporando.

La *distribución continua* se refiere a los cambios que implementa un programador en un software, a los que se le realizan pruebas automatizadas y que luego son cargados en un repositorio (como GitLab, GitHub o Bitbucket) para luego ser implementado en un ambiente de producción. Esta es una solución al problema de la poca visibilidad y comunicación entre los equipos de negocio (comerciales) y de desarrollo, y su fin último es implementar código nuevo con el mínimo esfuerzo posible.

La *implementación continua* (la otra “CD”) hace referencia a la liberación automática de los cambios implementados por los programadores desde el repositorio hacia el ambiente de producción para que los usuarios puedan usarlos. Esto resuelve el problema de la sobrecarga del equipo con procesos manuales que retrasan la distribución del software.

En la Figura 3, se muestra un resumen de CI/CD como un proceso que suele representarse como un canal y que implica incorporar un alto nivel de automatización continua y supervisión constante al desarrollo del software.

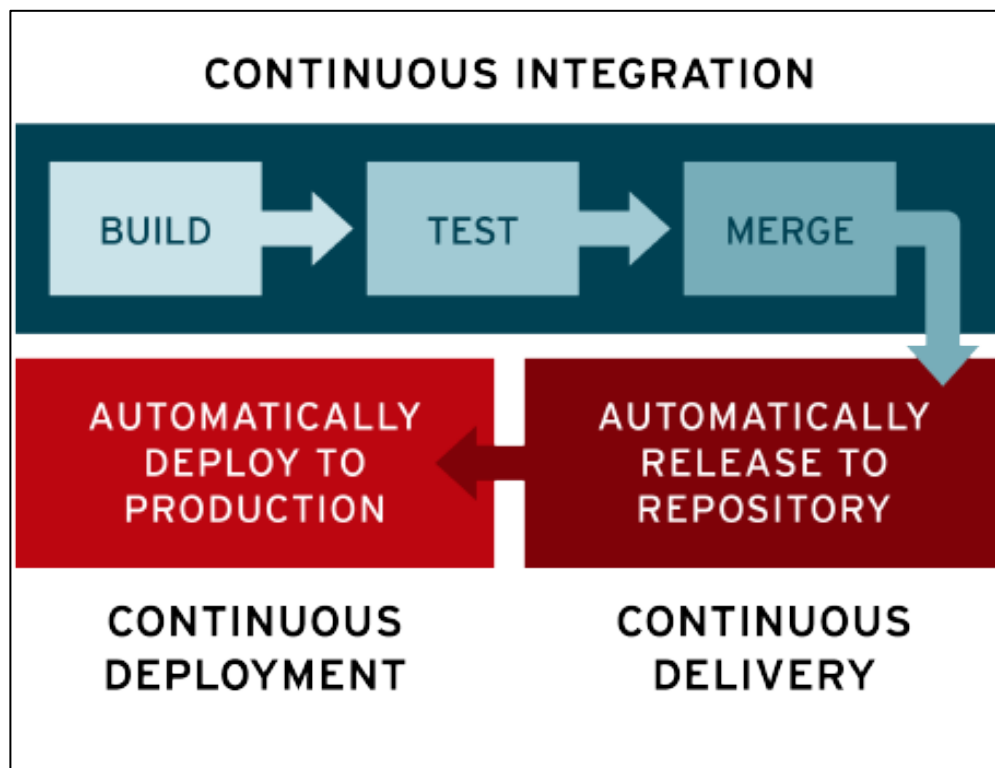


Figura 3 - Continuous Integration, Continuous Delivery y Continuous Deployment [17]

A continuación, un resumen de CI/CD punto por punto:

2.5.2 Integración Continua

El poder contar con múltiples programadores que trabajen de forma simultánea en distintas funciones de un mismo software es posible gracias a la integración continua, la cual ayuda a fusionar los cambios que se introducen en el código para incorporarlos frecuentemente a una versión compartida. Una vez que los cambios son fusionados, estos son validados con la ejecución automática de pruebas unitarias e integración para verificar que no se haya dañado la aplicación. En caso de encontrar conflictos entre el código nuevo y el actual, la CI facilita la resolución de errores rápidamente.

Las principales prácticas de la integración continua son:

Mantener un repositorio fuente único, en aplicaciones complejas los desarrolladores crean ramas para trabajar en distintas funcionalidades, creando complejidad, por lo que es importante mantener un control de las versiones del código fuente y publicar cada cambio en la rama principal.

Automatizar la compilación, no solo implica automatizar un proceso programado de compilación, sino validar la compilación.

Hacer compilaciones auto-testeables, el primer paso del proceso de validación es saber que una compilación con problemas realmente falló. El siguiente paso es determinar si el producto de la compilación funciona como es esperado. Esta última prueba debe incluirse como parte del proceso de compilación. Esto consiste en pruebas rápidas funcionales y no funcionales.

Todos confirman sus cambios (*commit*) en la línea principal todos los días, un buen ejercicio entre programadores es hacer uso de la comunicación para conocer qué cambios están haciendo, esto nace a raíz de integrar su código con el de otros cada día en una línea principal. Esto permite rápidamente detectar conflictos.

Corregir compilaciones con error inmediatamente, en CI siempre se está desarrollando sobre una base estable, pero puede pasar que los programadores rompan la línea principal, lo importante es ser cuidadoso de que, si sucede, debe ser reparado cuanto antes y siempre desarrollar y probar localmente antes de realizar un commit.

Mantener una compilación rápida, la idea completa de la integración continua es obtener retroalimentación rápidamente, XP (*Extreme Programming*) recomienda mantener un tiempo bajo de compilación de 10 minutos.

Probar en un clon de producción, esto permite que bajo condiciones controladas se pueda saber si existe algún error o riesgo que afecte a los usuarios finales.

Haga que sea fácil para cualquier persona obtener el último ejecutable, esto permite que cualquier pueda probar, hacer demostraciones o simplemente ver que ha cambiado últimamente, permitiendo obtener retroalimentación al interior de la organización.

Todos pueden ver lo que pasa, la integración continua tiene que ver con la comunicación y el que cualquier pueda ver fácilmente el estado del sistema y sus cambios hacen esto posible.

Implementación automatizada, para hacer integración continua son necesarios múltiples ambientes, como pruebas, integración o producción. Por lo que el mover los ejecutables entre ambientes varias veces al día debe ser algo fácil y automático.

2.5.3 Distribución continua

El objetivo de la distribución continua es tener una base de código que pueda ser implementado en un entorno de producción en cualquier momento. Luego de la automatización de las pruebas unitarias e integración de la CI, la *distribución continua* automatiza la liberación de este código validado hacia un repositorio.

2.5.5 Implementación continua

La última etapa de un canal CI/CD es la *implementación continua* la cual es una extensión de la *distribución continua*, permitiendo automatizar la liberación de software a un ambiente de producción. La idea de esta implementación es poner rápidamente en producción cualquier cambio en el código realizado por un programador luego de haberlo escrito y luego de haber pasado todas las pruebas automatizadas.

En conjunto, todas estas prácticas de CI/CD relacionadas hacen que la implementación de software se lleve a cabo con menos riesgos y en una menor cantidad de tiempo, ya que es más fácil liberar cambios en el software en fragmentos pequeños en vez de hacerlo todo de una vez.

2.6 Product Vision Board (PVB)

La visión juega un papel importante para dar vida a un nuevo producto, actuando como el objetivo que guía a todos los involucrados en su desarrollo. Igualmente, importante es la estrategia del producto, la cual representa el camino elegido para alcanzar la visión. Sin una visión compartida y una estrategia efectiva, es probable que los miembros del equipo de desarrollo del producto tomen distintas direcciones y el crear un producto exitoso se vuelva difícil.

Si bien es cierto, ambos conceptos parecen ser la clave del éxito, el escribirlos puede llegar a ser un desafío. Para ello nace el Product Vision Board, como una herramienta que ayuda a capturar las suposiciones sobre los usuarios y los clientes del producto, las necesidades que este debe abordar, sus características clave y el valor que el producto debe crear para el negocio [18].

El product vision board utiliza cinco secciones como se muestra en la Figura 4, las cuales son:

Visión: establece el objetivo general, la razón principal para crear un producto, el cambio positivo que se desea lograr. Lo ideal es establecer una visión grande e inspiradora, que contenga una breve declaración o lema.

Grupo objetivo (target group): describe el mercado o segmento de mercado que se desea abordar. Debe indicar a quien beneficiará el producto, quienes son sus usuarios y clientes. Es recomendado elegir un grupo homogéneo y claro y por sobretodo crear un producto (nuevo) para unos pocos, no para muchos [19].

Necesidades (needs): describen la propuesta de valor: el problema principal que aborda el producto o el beneficio principal que ofrece. Esta sección debe dejar en claro por qué las personas querrán usar o pagar el producto. Se debe describir cómo se ve el éxito para los usuarios y clientes. En caso de identificar varias necesidades, estas se deben priorizar.

Producto (product): el producto resume las tres o cinco características del producto, las cuales lo hacen destacar y que son fundamentales para su éxito.

Objetivos de negocio (business goals): esta sección explica por qué vale la pena que la compañía invierta en el producto. Establece además los beneficios comerciales deseados, por ejemplo, aumentar los ingresos, ingresar a un nuevo mercado, reducir costos, desarrollar la marca o adquirir nuevos conocimientos.






 Vision Help teams create great, UX-rich products			
 Target Group Users: Product managers and product owners Customers: Mid-size to large enterprises	 Needs Easily integrate UX artefacts into a product backlog	 Product Tablet app; data is held in GreenHopper Looks like a physical canvas; intuitive to use Provides guidance and templates	 Business Goals Open up a new revenue stream Develop our main brand

Figura 4 - Ejemplo de un Product Vision Board [18]

El product vision board no es solo una herramienta para comunicar la estrategia de un producto, también permite probar suposiciones y capturar los conocimientos recién adquiridos. Por ejemplo, resulta de gran utilidad identificar el mayor riesgo o la mayor incertidumbre en el tablero. Esto crea un enfoque y refuerza el fallar rápido, entendiendo rápidamente que es lo que funciona y que no, o que suposiciones son ciertas y cuales son falsas.

En la Figura 5 se describe el proceso, en donde: 1. se identifica el mayor riesgo, 2. se decide cómo abordarlo, 3. se recolectan datos y finalmente 4. se analizan los datos y se realizan cambios.

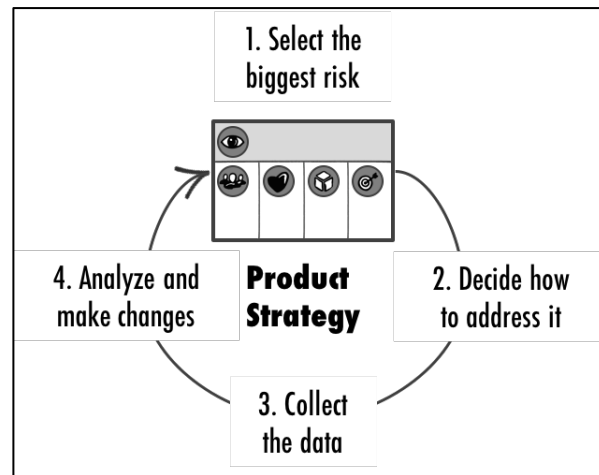


Figura 5 - Investigación y validación con el Product Vision Board [18]

2.7 Vistas y puntos de vista

Cuando se comienza a diseñar la arquitectura de un sistema, existen ciertas preguntas que podrían ser complejas de responder, como: ¿cuáles son los elementos funcionales principales de la arquitectura?, ¿cómo estos elementos interactúan unos con otros y con el mundo exterior?, ¿qué información será manejada, almacenada y presentada?, ¿qué hardware físico o virtual y software será requerido para soportar los elementos funcionales y de información? o ¿qué ambientes deben ser provistos?

Una tentación común es intentar responder a todas estas preguntas a través de un único modelo, el cual probablemente utilice una mezcla de notaciones formales e informales para describir los aspectos del sistema en un gran diagrama: estructura funcional, capa de software, concurrencia, comunicación entre componentes, implementación física o virtual y otros. Al pensar en un diagrama de este tipo, resulta imposible describir una arquitectura de software utilizando un único modelo, el cual será difícil de entender y poco claro al intentar identificar las características más importantes de la arquitectura. Por otra parte, no será útil para todos los stakeholders, quienes intentaran

entender solo los aspectos que les son de particular interés para cada uno.

Es necesario poder representar un sistema complejo en una forma que sea manejable y comprensible por stakeholders técnicos y de negocio. Un enfoque es atacar el problema desde diferentes direcciones simultáneamente. En este enfoque, el diseño arquitectónico es particionado en *vistas*, separadas pero interrelacionadas, donde cada una describe un aspecto de la arquitectura del sistema, pero colectivamente estas vistas describen el sistema completo [20].

2.7.1 Vistas arquitectónicas

Una vista es una representación de los aspectos estructurales de una arquitectura, la cual ilustra cómo la arquitectura aborda las preocupaciones de sus stakeholders.

Cuando se está decidiendo que incluir en una vista, se deben responder las siguientes preguntas:

- ¿A qué clase de stakeholders está dirigida la vista? Una vista puede estar enfocada en una clase particular de stakeholder, a un individuo específico o a un gran número de personas con distintos intereses y niveles de pericias o habilidades.
- ¿Cuánto conocimiento técnico tienen los stakeholders? Por ejemplo, los clientes o usuarios de un sistema podrían ser expertos en ciertas áreas, pero es poco probable que conozcan sobre el hardware o software del sistema, en ese caso, la vista podría aplicarse a los programadores, al equipo de operación o de soporte.
- ¿Qué preocupaciones de los stakeholders intenta abordar la vista?, ¿cuánto saben los stakeholders sobre el contexto arquitectónico?
- ¿Cuánto necesitan saber los stakeholders sobre este aspecto de la arquitectura?

Uno de los mayores desafíos a la hora de diseñar la arquitectura de un sistema, es obtener el nivel correcto de detalle en cada vista. Proveer mucho detalle podría hacer que los stakeholders se sientan abrumados con tanta información y no comprendan del todo el diseño, por otro lado, el proveer muy poco detalle trae consigo el riesgo de que los stakeholders hagan suposiciones no válidas.

Una estrategia es incluir en las vistas solo los detalles que estén relacionados a los objetivos del diseño de la arquitectura, esto es, todos los detalles que ayuden a explicar la arquitectura a los stakeholders o demuestren que las preocupaciones de los stakeholders están siendo consideradas en el diseño.

2.7.2 Puntos de vista

Un punto de vista es una colección de patrones, plantillas y convenciones para construir un tipo de vista. Los puntos de vista arquitectónicos proveen un marco de trabajo para la captura de conocimiento arquitectónico reutilizable, el cual puede ser usado para guiar la creación de un tipo particular de diseño arquitectónico.

Cuando se desarrolla una vista, ya sea que se use o no un punto de vista formalmente definido, se debe tener en claro, qué tipo de preocupaciones abordará la vista, qué tipo de elementos arquitectónicos presenta y a quién está dirigido el punto de vista. Es necesario que los stakeholders puedan entender esto también.

2.7.3 Catálogo de puntos de vista

Existen seis puntos de vista principales, los cuales se detallan en la Figura 6: funcional, de información, concurrencia, desarrollo, despliegue y operacional. Por supuesto que no todos los puntos de vista pueden aplicar a todas las arquitecturas y algunas podrían ser más importantes que otras en el diseño arquitectónico de un sistema.

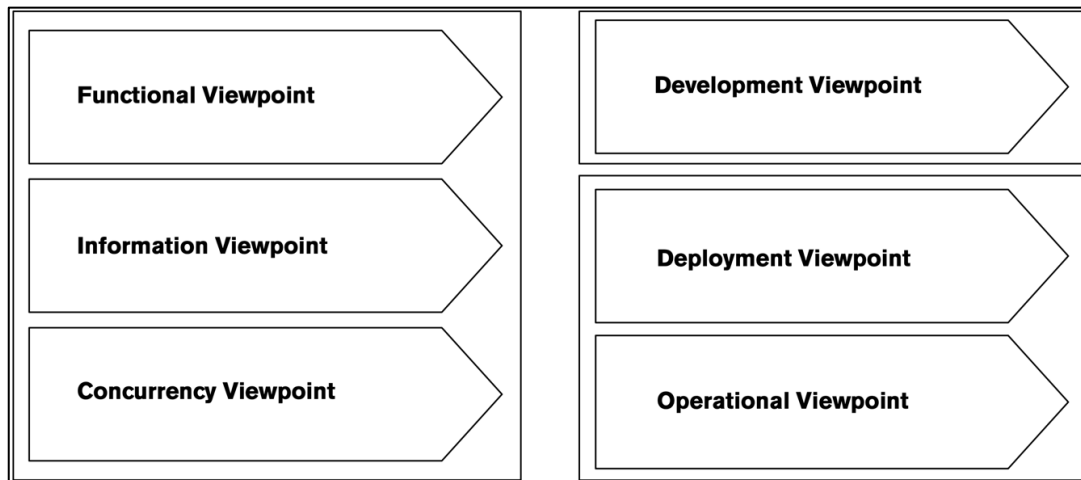


Figura 6 - Puntos de vista [20]

A continuación, se define cada uno de los principales puntos de vista:

PV1. Funcional: describe los elementos funcionales del sistema, sus responsabilidades, interfaces e interacciones primarias, y es lo que primero intentan leer los stakeholders. Este punto de vista lleva a dar forma a otros puntos de vista y además tiene un impacto significativo en las cualidades de un

sistema como su habilidad de cambiar, seguridad y rendimiento.

Esta vista esta está constituida por los siguientes elementos:

- **Elementos funcionales:** corresponde a una parte del sistema que cuenta con responsabilidades particulares y expone interfaces que permiten conectarse a otros elementos. En palabras simples podría significar un módulo de código, pero, por otro lado, podría ser también un paquete de aplicación, almacenamiento de datos o incluso un sistema completo.
- **Interfaz:** mecanismo por el cual se puede acceder a las funciones de un elemento a través de otro.
- **Conectores:** son las piezas de la arquitectura que conectan elementos entre sí para interactuar.
- **Entidades externas:** pueden representar otros sistemas, programas, piezas de hardware o cualquier otra entidad a la cual el sistema se comunica.

PV2. Información: describe la forma en que la arquitectura almacena, manipula, maneja y distribuye la información. El objetivo de este punto de vista es responder a preguntas como: contenido, estructura, propiedad, latencia, referencias y migración de los datos.

PV3. Concurrencia: describe la estructura de concurrencia del sistema y asigna elementos funcionales a unidades de concurrencia para identificar las partes del sistema que pueden ejecutarse simultáneamente, como esto es coordinado y controlado.

PV4. Desarrollo: describe la arquitectura que soporta el proceso de desarrollo de software.

PV5. Despliegue: describe el ambiente en el cual el sistema será desplegado, capturando el ambiente de hardware necesario para el sistema (nodos de procesamiento, redes y almacenamiento), el ambiente técnico requerido para cada elemento y la conexión de los elementos de software con el entorno donde serán ejecutados.

PV6. Operacional: describe como el sistema será operado, administrado y soportado cuando se encuentre operando en su ambiente de producción.

La Figura 7 muestra las relaciones entre las vistas creadas con los puntos de vista mencionados anteriormente.

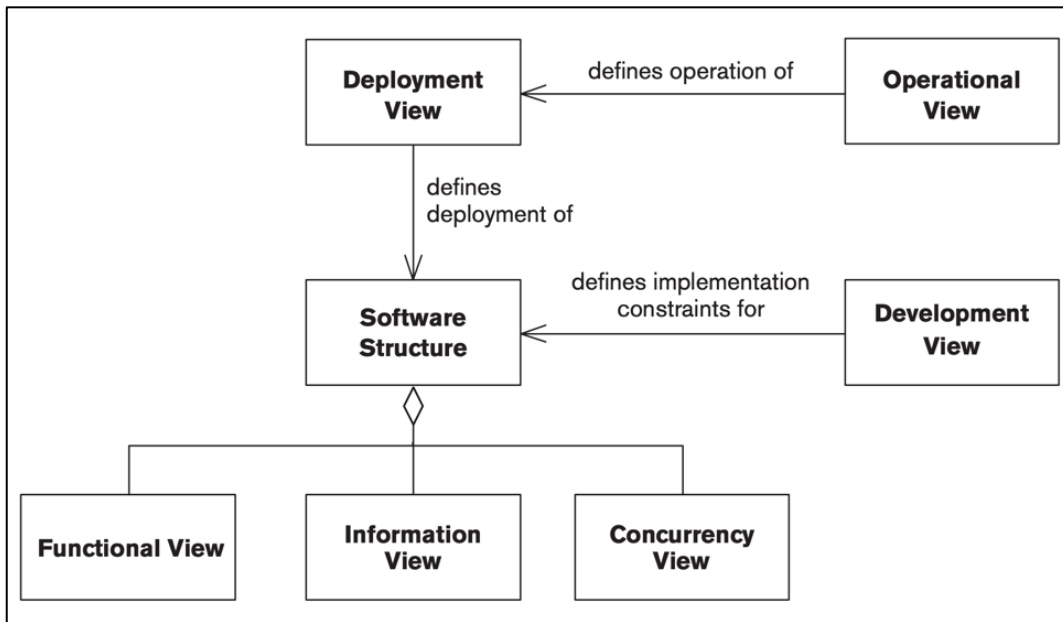


Figura 7 - Relaciones entre vistas [20]

2.8 Principios arquitectónicos

Un principio arquitectónico permite definir el marco de trabajo con el cual se diseñará la arquitectura de un sistema, y representa una creencia o un enfoque que da claridad al diseño, transformando lo implícito en explícito, con la finalidad de diseñar un sistema que mantenga a la vista todas las suposiciones de los stakeholders.

Uno de los usos más poderosos de los principios arquitectónicos es que permiten dar trazabilidad y justificación a las decisiones que serán tomadas para diseñar el nuevo sistema. La trazabilidad se produce al vincular principios mediante razones (por qué es un principio valioso y apropiado para la arquitectura del sistema) y consecuencias (que necesita ocurrir para que el principio se cumpla):

Objetivos y motivadores de negocio: se deben obtener de la definición del alcance de la arquitectura y sus preocupaciones, es importante involucrar a los stakeholders en su formulación.

Principios de negocio: para desarrollarlos es necesario usar los motivadores de negocio en donde reside su razón de ser.

Principios tecnológicos: para desarrollarlos, se necesita usar los principios de negocio en donde puede residir la razón de uno o varios principios.

Decisiones arquitectónicas: se deben usar principios tecnológicos para desarrollarlas, teniendo su razón en uno o varios de estos principios.

Una buena forma de graficar su aplicación es la Figura 8, donde se expresa cómo cada decisión tomada en el diseño de la arquitectura del nuevo sistema está alineada con los objetivos y motivadores del negocio [21], permitiendo entender la importancia de la arquitectura en el crecimiento y operación del sistema.

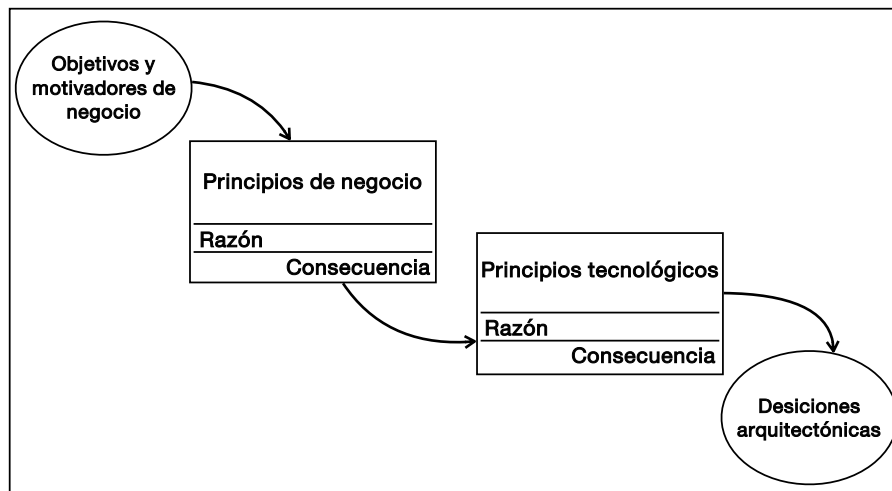


Figura 8 - Usando principios para demostrar trazabilidad [21]

2.9 ETL: *Extract, Transform and Load*

Un problema común que enfrentan las organizaciones es como reunir los datos provenientes de múltiples fuentes, en múltiples formatos y moverlos a uno o más lugares donde almacenarlos. Es posible además que el destino no sea el mismo tipo de almacén de datos que el del origen y a menudo también el formato es diferente o los datos deben limpiarse antes de ser cargados en su destino final.

Muchas herramientas, servicios y procesos han sido desarrollados a lo largo de los años para ayudar a abordar estos desafíos, pero no importa que proceso sea usado, es una necesidad común el coordinar el trabajo y aplicar cierto nivel de transformación de los datos en el canal de datos.

Extraer, transformar y cargar (ETL por sus siglas en inglés: *extract, transform and load*) es un canal de datos usado para recolectar datos de varias fuentes, transformar los datos de acuerdo a las reglas de negocio establecidas y cargarlos en un almacén de datos definido como destino [22].

La transformación de los datos generalmente involucra varias operaciones como filtrar, ordenar, agregar, unir datos, des-duplicar y validar datos. Este proceso se representa en la Figura 9.

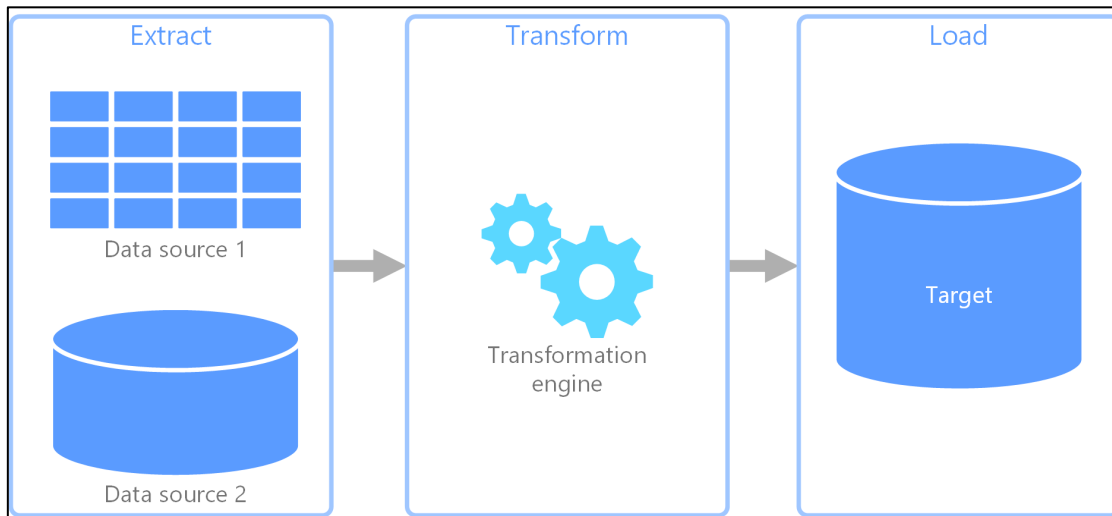


Figura 9 - Proceso ETL [22]

A menudo, las tres fases del proceso ETL corren en paralelo para ahorrar tiempo. Por ejemplo, mientras los datos son extraídos, un proceso de transformación puede estar trabajando en los datos ya recibidos y preparándolos para ser cargados. Por otro lado, el proceso de carga puede comenzar a preparar los datos más que esperar a que el proceso de extracción termine por completo.

2.9.2 ELT: *Extract, Load and Transform*

Otro proceso es: extraer, cargar y transformar (ELT) el cual difiere de ETL solamente en donde la transformación de los datos toma lugar. En el canal ELT la transformación ocurre en el almacenamiento de destino de los datos [22]. En vez de usar un motor separado de transformación, las capacidades de procesamiento del almacenamiento de destino son usadas para transformar los datos. Esto simplifica la arquitectura, removiendo el motor de transformación del canal ETL. Otro beneficio de este enfoque es que al escalar el almacenamiento de datos de destino, también escala la performance del canal ELT.

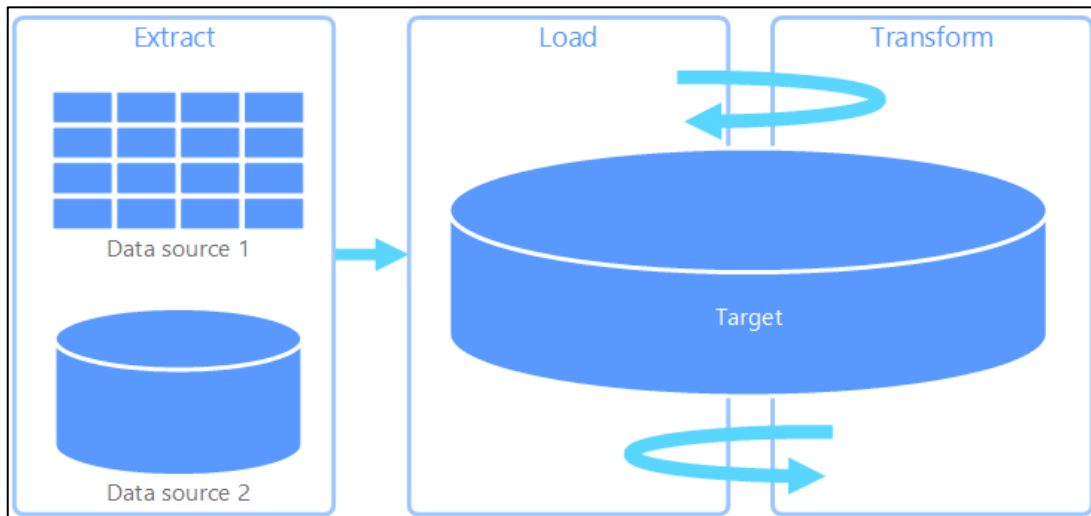


Figura 10 - Proceso ELT [22]

2.10 El método KPT: *Keep, Problem, Try*

KPT es un método que pretende formalizar el evento ágil de Scrum: Sprint Retrospective [23]. La utilización de este método es bastante simple y pretende que, sobre la revisión de un proceso previamente realizado, se listan ideas en una pizarra, las cuales están divididas en:

Keep: las cosas buenas que se deben mantener como parte del proceso o cultura del equipo.

Problem: los desafíos que se presentaron en el proceso y las mejoras que se deben realizar.

Try: las acciones correctivas para resolver los problemas o las mejoras para obtener mejores resultados en la siguiente iteración del proceso.

En la Figura 11 se muestra gráficamente cómo se estructura el método KPT, generalmente se utiliza una pizarra dividida en tres secciones, donde se colocan tarjetas adhesivas con ideas.

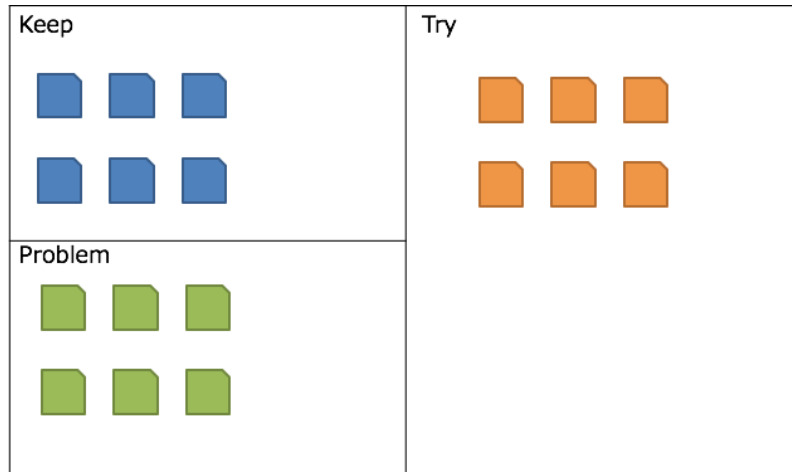


Figura 11 - Estructura gráfica del método de retrospectiva KPT (Keep, Problem, Try) [23]

La idea de este método es utilizarlo incrementalmente al final de cada Sprint, dar seguimiento a los ítems y validar la evolución de los problemas y soluciones planteadas.

Por cada acción escrita en la sección *Try*, algún miembro del equipo será designado para estar a cargo de ella (investigar, implementar un nuevo flujo, escribir un nuevo trozo de código, etc.). Además, esta persona estará encargada de presentar la solución o hallazgos en la próxima sesión de retrospectiva, en donde el equipo podrá decidir si la solución propuesta resuelve el problema o no.

Este método ayuda mejorar la cohesión del equipo, puesto que permite que sus miembros se preocupen por los problemas de otros y en conjunto trabajen en una solución. Cualquier miembro del equipo puede presentar una propuesta de solución a un problema y cada uno puede asumir la responsabilidad de intentar implementar la solución, la cual posteriormente será evaluada por el equipo.

2.11 The Go Product Roadmap

La herramienta llamada: *The Go Product Roadmap* [24] es un plan estratégico de alto nivel que menciona las cosas que le gustaría a la organización desarrollar en los próximos meses sobre el producto. Esta herramienta permite mantener alineados a los stakeholders y facilitar la priorización dada su simpleza.

Es importante mencionar que la idea no es que contenga demasiadas funcionalidades o detalles, porque no debe convertirse en una herramienta táctica como lo es el backlog del producto.

2.11 Resumen

Cada aspecto revisado en este capítulo, constituye un elemento esencial en el desarrollo de una aplicación móvil que enfrenta grandes desafíos en el terreno del comercio electrónico, en donde la competencia por ganar participación de mercado es importante en cuanto a la calidad del software y los procesos comerciales. Es por ello por lo que la base para poder entrar a la arena del comercio electrónico es contar con un equipo organizado, con un proceso de desarrollo flexible que permita entregar valor a los usuarios en un breve periodo

de tiempo y además contar con las plataformas tecnológicas preparadas para una alta demanda de peticiones y la flexibilidad de desplegar nuevas versiones del producto cuando sea necesario, entregando información de calidad y valor para los usuarios finales. Parte del desafío de construir un producto que sea exitoso en términos de uso radica en fundar la base para lograr calidad en la información, estabilidad en el uso de la plataforma, la experiencia del usuario, la automatización de tareas manuales y contar con una arquitectura robusta frente a escenarios complejos, como la alta concurrencia de usuarios.

3. Situación actual: problema

En el siguiente capítulo se comentará el escenario actual de América Latina respecto al comercio electrónico y como las grandes compañías como Amazon o Alibaba se suman a la industria local en cuanto a participación de mercado. Luego, se revisará la situación actual del proceso de desarrollo y la arquitectura de la aplicación móvil Sodimac, para finalmente en el análisis de la situación actual, concluir sobre los problemas organizacionales y tecnológicos que han afectado al desarrollo de la aplicación móvil.

3.1 Contexto de negocio

En el último tiempo el comercio electrónico local en América Latina ha visto a grandes compañías extranjeras como Amazon [25] o Alibaba [26] llegar a tener participación comercial en el medio local gracias al uso de buenas prácticas en el desarrollo de productos, logística y estrategias comerciales. Esto ha llevado a que muchas compañías hayan querido reformular sus procesos, tecnología y estrategias de negocio en vista del desarrollo que estas compañías lideran.

Al ser Sodimac una compañía diversa en cuanto a áreas y equipos es diversa también en términos de procesos y tecnología. Puede ocurrir que en una misma área existan equipos que ocupan metodologías e incluso tecnologías distintas para resolver problemas similares. Estas diferencias tienen un origen complejo que puede radicar en experiencias negativas bajo la definición de estándares corporativos de desarrollo y tecnología que no se adaptan al modelo de negocio actual de la compañía. Esto es lo que ha motivado la búsqueda de alternativas para abordar los desafíos propios de cada equipo.

Trabajar en silos es común dentro de la compañía, haciendo alusión a la falta de comunicación entre equipos y esfuerzos locales que no son compartidos ni como tecnología ni como experiencia a nuevos equipos y productos. Esto se refleja en parte en los sistemas monolíticos que cada área ha desarrollado, los que más allá de su definición, son un reflejo de que trabajar sin comunicación, entorpece la visión a largo plazo. Esto ha impactado en el crecimiento tecnológico de la compañía en los últimos años, pese a la estrategia e inversión declarada, llevando con ello al lento crecimiento de nuevos productos, retraso en la adopción de procesos y tecnología.

3.2 Aplicación actual

La aplicación móvil de Sodimac está enfocada en entregar información básica a los clientes del segmento hogar (quienes están en un proceso de remodelación) y construcción (profesionales de la construcción) para ayudarlos a en su proceso de compra de productos en las tiendas o el sitio web. Las funciones que posee son: buscador de productos, escáner para códigos de producto, ficha de

producto, seguimiento del estado de compras e información de tiendas como su ubicación.

En la Figura 12 se muestra el esquema de navegación y funciones de la aplicación móvil Sodimac.

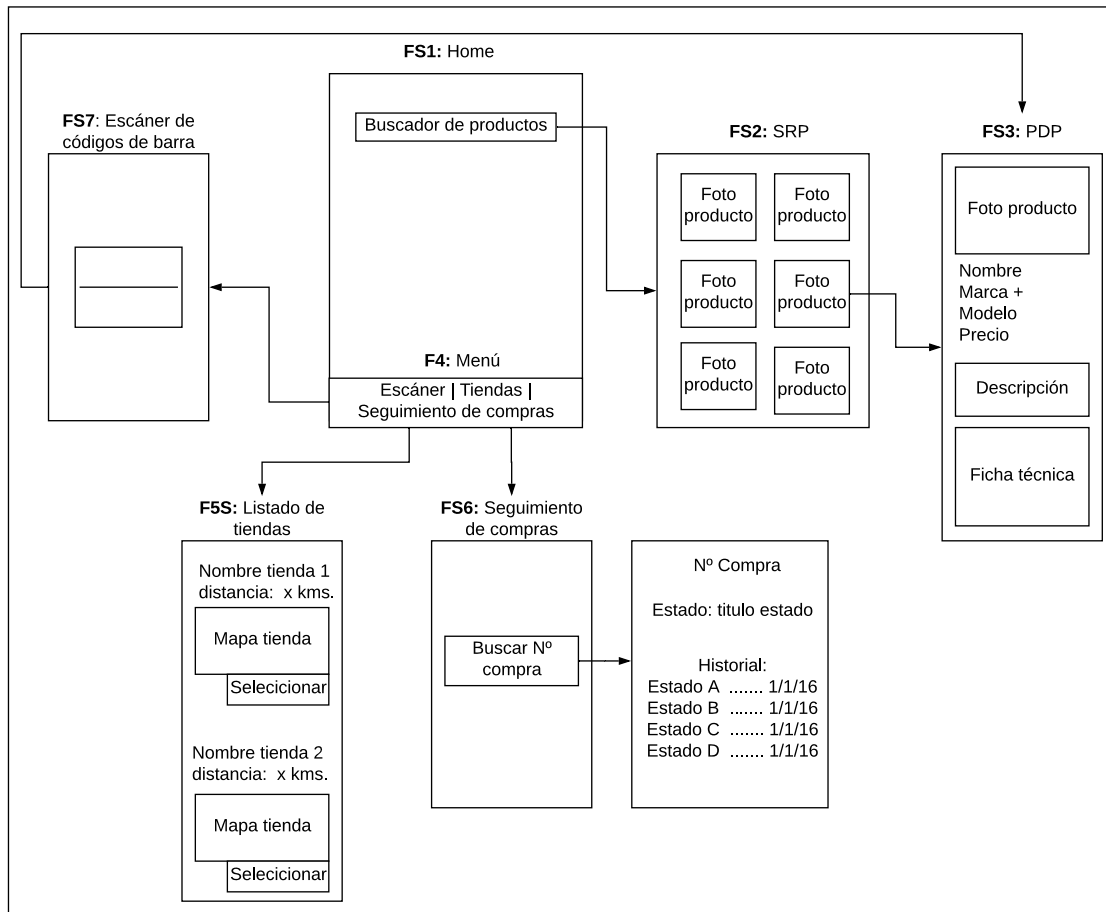


Figura 12 - Esquema de navegación y funciones de la actual aplicación Sodimac

Como se ve en la Figura 12, el flujo de navegación del usuario parte en el Home, que es la pantalla inicial. Desde aquí el usuario puede buscar productos por texto o escanear el código de barra de los productos. Además, puede revisar en el menú información de tiendas o ver el estado de una compra.

FS1. Home: es la pantalla inicial de la aplicación, donde se ubica el buscador de productos, el menú principal y el botón que permite acceder al escáner que permite leer los códigos de barra de los productos en tienda para ver su información detallada.

FS2. SRP: del acrónimo en inglés *Search Results Page*, es la página de resultados que es desplegada a partir de la búsqueda de productos vía texto. En ella se muestra un listado de productos junto a su foto, precios, nombre y marca. Al pulsar un producto como resultado, el usuario es llevado a la página de detalle del producto.

FS3. PDP: del acrónimo en inglés *Product Detail Page*, es la página de detalle del producto, la cual contiene información como fotos, precios, marca, descripción, ficha técnica, productos relacionados o complementarios.

FS4. Menú: acá están las funcionalidades primarias de uso como el escáner, tiendas y seguimiento de compras. Al pulsar cualquiera de estas opciones, el usuario es llevado a la pantalla de destino.

FS5. Listado de tiendas: acá se encuentra la información básica de todas las tiendas de Chile, como su dirección y horarios de apertura y cierre. El usuario puede seleccionar una tienda como favorita para que en base a esto se configure la disponibilidad y precios de los productos de esa tienda particular.

FS6. Seguimiento de compras: el usuario puede ingresar el número de compra obtenido al comprar por el sitio web y ver en qué estado se encuentra su entrega, los estados pueden ser: en ruta a domicilio, listo para ser retirado en una tienda, en proceso o entregado.

FS7. Escáner de códigos de barra: al pulsar esta función se abre la cámara del teléfono y permite escanear un código de barras de producto. Si el escáner encuentra el producto se muestra su ficha PDP, en caso contrario se alerta al usuario que no hay información disponible sobre ese producto.

3.3 Proceso de desarrollo actual

El proceso de desarrollo no está formalizado existe la percepción por parte de los stakeholders y el equipo de desarrollo, que este puede ser un punto crítico para el correcto desarrollo de la aplicación móvil. A continuación, se describirá el proceso, su organización, flujo de trabajo y además se mostrarán evaluaciones cualitativas y cuantitativas sobre el proceso.

3.3.1 Descripción

Desde el inicio del desarrollo de la aplicación móvil en la compañía, se ha utilizado un proceso exploratorio de desarrollo de software, donde existen algunas etapas definidas como las pruebas funcionales y la publicación en las tiendas de aplicaciones, pero otras como la definición de requerimientos o diseño, pueden mutar entre ciclos y permanecer como no declaradas formalmente entre el equipo de desarrollo y los stakeholders. Esto ha causado que la estimación de esfuerzo en el desarrollo e incluso el éxito de las versiones sea incierto y dependa de la capacidad propia y personal del equipo de desarrollo, lo cual no provee bases sólidas para una productividad a largo plazo y la mejora en la calidad del proceso al interior de la organización.

El equipo de desarrollo de la aplicación móvil está conformado por negocio, TI y proveedores de

desarrollo. A continuación, se describen los roles que existen en cada área presente en el desarrollo de la aplicación móvil.

Negocio

Es donde están los roles encargados de la definición y priorización de requerimientos que son implementados en la aplicación móvil, a su vez son quienes financian su desarrollo.

R1. Stakeholders: provenientes de distintos equipos de negocio de la compañía como: operaciones, logística, marketing, inteligencia de negocio y venta online. Si bien es cierto ellos no son parte del equipo de desarrollo, son quienes constantemente comparten necesidades del negocio con el encargado de la aplicación móvil.

R2. Encargado de experiencia usuaria: responsable de entregar propuestas de diseño de funcionalidades basadas en las mejores prácticas de usabilidad en aplicaciones móviles.

R3. Encargado de la aplicación móvil: tiene a su cargo el impulsar el desarrollo de la aplicación Sodimac, recogiendo necesidades de usuarios y stakeholders, definiendo las funcionalidades y su prioridad de desarrollo.

Existen gerentes de área para la experiencia usuaria y venta online, que tienen bajo su jefatura a los roles R2 y R3 respectivamente. Los roles de tipo gerente intervienen en las definiciones generales para el desarrollo de las aplicaciones como presupuesto y estrategia.

TI

Esta área gestiona el desarrollo de las aplicaciones en términos contractuales con los proveedores y además facilita la relación entre el negocio y los programadores en aspectos técnicos.

R4. Jefe de proyectos: este rol aporta con su experiencia en la definición técnica del desarrollo de las aplicaciones, además de facilitar el proceso de desarrollo entre el área de negocio, los proveedores de desarrollo y la integración con sistemas legados de la compañía.

Proveedor de desarrollo

Son los programadores de las aplicaciones en base a las definiciones funcionales y técnicas provistas por las áreas de negocio y TI.

R5. Programador Android: Encargado de la programación de la aplicación nativa para teléfonos con sistema operativo Android.

R6. Programador iOS: Encargado de la programación nativa de las aplicaciones para teléfonos iPhone de Apple.

La Figura 13 ilustra los roles que conforman el equipo de desarrollo actual respecto a: negocio, TI y proveedores de desarrollo, quienes juntos forman un solo gran equipo.

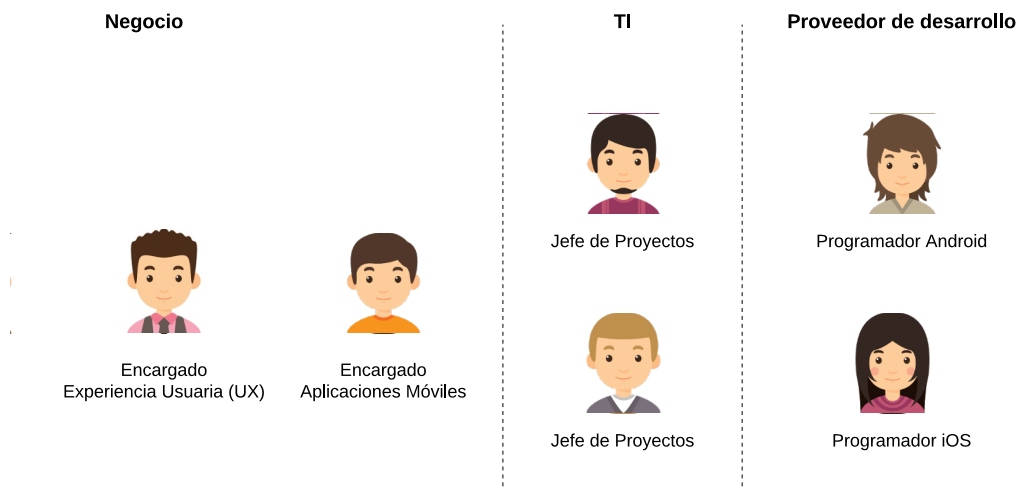


Figura 13 - Equipo de desarrollo actual de la aplicación móvil

En la Figura 14 se detalla el flujo del proceso actual de desarrollo de la aplicación móvil, junto a las etapas donde interactúan las áreas y roles previamente comentados:

E1. Recepción de necesidades u oportunidades: éstas vienen principalmente de las experiencias de los stakeholders de la compañía y son entregadas a través de mails o de palabra al encargado de desarrollo de la aplicación móvil para su evaluación.

E2. Análisis de necesidades: el encargado de la aplicación móvil define según juicio experto qué necesidades pueden ser resueltas a través de funcionalidades.

E3. Diseño de solución: las funcionalidades que nacen como respuesta a una necesidad son diseñadas según juicio experto y en base a las mejores prácticas de usabilidad por el encargado de experiencia usuaria.

E4. Definición de requerimientos: los diseños son descritos funcionalmente en un documento de requerimientos que es entregado a los programadores para ser implementadas según un calendario definido.

E5. Desarrollo funcional: una vez que parte el ciclo de desarrollo, el cual puede tomar hasta cuatro semanas, se realizan reuniones de seguimiento semanal y cada cierto tiempo se acuerda la entrega de versiones de las aplicaciones

E6. Pruebas de aceptación usuaria (UAT): con una versión entregada por los programadores, se realiza las UAT antes de que el ciclo de desarrollo termine. En caso de haber observaciones, éstas son comentadas a los programadores para su revisión y corrección.

E7. Evaluación en tienda de aplicaciones: una vez finalizado el ciclo de desarrollo y las versiones de las aplicaciones operan, se procede a acordar con los programadores la carga de la nueva versión en las tiendas de aplicaciones de Apple y Google. En el caso de Apple el tiempo de revisión previo a la publicación puede tomar hasta cinco días hábiles sin considerar la existencia de observaciones, donde se deben realizar las correcciones necesarias y enviar una nueva versión para ser evaluada y posteriormente publicada. En el caso de Android el proceso de publicación dura entre cuatro a seis.

E8. Publicación de versión productiva: de no existir observaciones que impidan la publicación, esta puede ser ejecutada automáticamente o de forma manual previa notificación, también se puede publicar a un segmento particular de la audiencia (ejemplo, liberar la nueva versión solo al 50% de los usuarios Android de Chile). Tanto en Android como en iOS existe un lapso (3 horas en promedio) en que demora en visualizarse la nueva versión en todos los segmentos definidos para publicación.

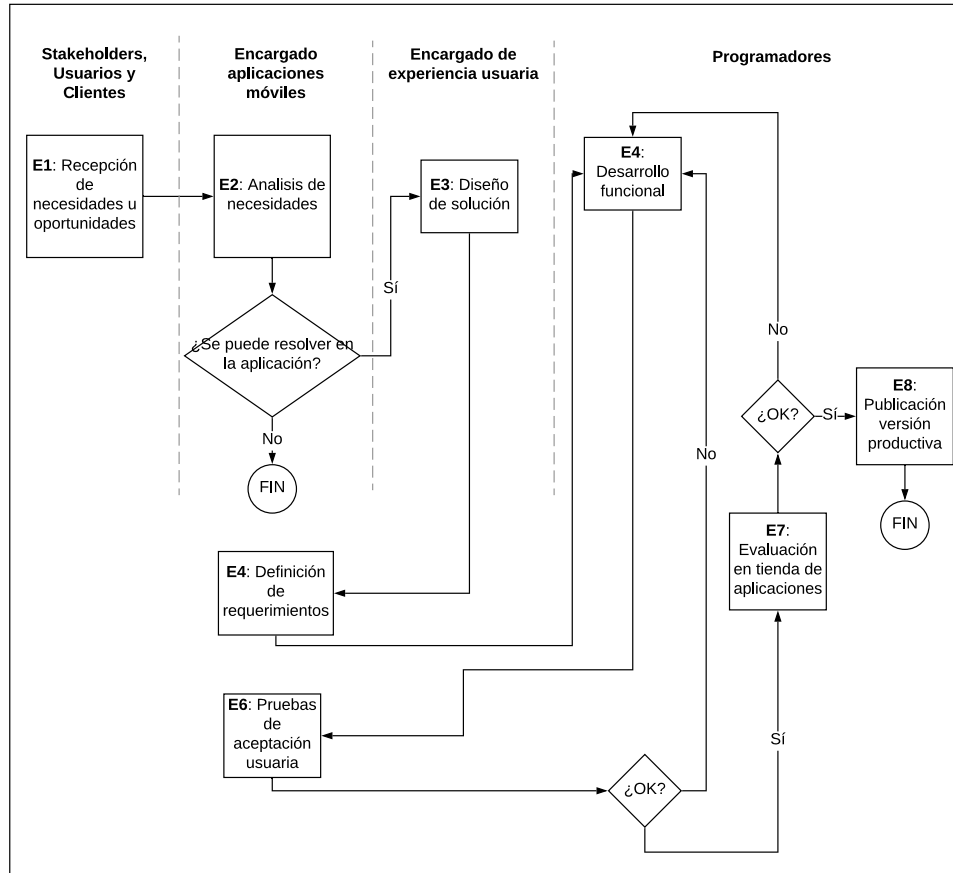


Figura 14 - Flujo del proceso actual de desarrollo de la aplicación móvil

3.3.2 Evaluación cualitativa

Para entender el estado actual del proceso de desarrollo de software de la compañía, se realizaron dos actividades de evaluación, una cualitativa y otra cuantitativa, con el fin de realizar un análisis que permita tomar decisiones para alcanzar las metas y objetivos planteados por la compañía.

En la primera evaluación se realizó un análisis cualitativo FODA, una herramienta de planificación estratégica diseñada para realizar un análisis interno (fortalezas y debilidades) y externo (oportunidades y amenazas) en una organización [27]. En este análisis participaron quienes conforman el equipo de desarrollo de la aplicación móvil y algunos stakeholders. La Tabla 1 describe el protocolo de la evaluación FODA.

Evaluación FODA	
Objetivo	Analizar el proceso de desarrollo de software de la aplicación móvil.
Procedimiento	<p>Asistentes: encargado de la aplicación móvil, encargado experiencia usuaria, jefes de proyecto de TI, gerente de TI, gerente desarrollo de venta online, subgerente de experiencia usuaria y gerente comercial.</p> <p>Duración: 4 horas.</p> <p>Equipo y material: pizarra y tarjetas adhesivas para desarrollar las ideas sobre el análisis FODA.</p> <p>Dinámica: en primer lugar, se define el análisis FODA como tal, luego se entregan tarjetas adhesivas a cada persona para que escriba una idea asociada a los cuatro aspectos del análisis FODA.</p>
Instrumento	<ol style="list-style-type: none"> 1. Definición de análisis FODA. 2. El moderador (encargado de la aplicación móvil) explica la dinámica a realizar con tarjetas adhesivas. 3. Cada persona escribe en una tarjeta la idea que tiene sobre los cuatro aspectos del análisis. 4. Se comienza con las fortalezas y luego que todos escriben ideas sobre ella. Estas se pegan en la pizarra y se continua con el siguiente aspecto. 5. Se discute sobre las ideas en las tarjetas y se define una visión de equipo sobre cada aspecto, agrupando las ideas similares y priorizando las más importantes.

Tabla 1 - Protocolo para la evaluación FODA

En la Tabla 2 se pueden ver las ideas concluyentes del análisis realizado, en donde se destaca como fortaleza la simplicidad y facilidad de adopción del proceso actual, dado que no tiene una documentación detallada, extensa planificación y fases complejas.

Respecto a las debilidades, claramente se identifica la falta de adaptación del proceso frente a situaciones complejas, como también la poca de transparencia de los avances, la falta de roles expertos y la inexacta estimación de esfuerzo para las tareas de desarrollo.

Quizás la gran oportunidad es que hoy no existen compañías en Chile que tengan aplicaciones móviles de comercio electrónico en el rubro del mejoramiento del hogar, generando una ventaja momentánea.

Finalmente, la gran amenaza pasa por el acelerado cambio en los gustos, necesidades y expectativas de los clientes. Hoy la compañía no está poniendo al cliente en el centro, obteniendo retroalimentación de sus usuarios o entregando valor constantemente.

Análisis FODA del proceso de desarrollo actual	
Fortalezas	<p>FR1. Puede ser fácilmente adoptado por personas sin conocimientos técnicos.</p> <p>FR2. No se necesita extensa planificación.</p> <p>FR3. Funciona bien para desarrollos con requerimientos fijos y claros.</p>
Debilidades	<p>DE1. El cliente no ve funcionalidades hasta que están totalmente terminadas.</p> <p>DE2. No funciona con proyectos largos y complejos en donde los requerimientos no son claros y pueden variar.</p> <p>DE3. Falsas impresiones del progreso de avance del desarrollo (visibilidad a stakeholders).</p> <p>DE4. La traducción de necesidades/oportunidades a requerimientos y funcionalidades es a juicio experto, sin datos como base.</p> <p>DE5. Compleja forma de estimación de tiempo y esfuerzo de desarrollo.</p>
Oportunidades	<p>OP1. No hay otras empresas en Chile con aplicaciones móviles que compitan directamente en el rubro del mejoramiento del hogar.</p>
Amenazas	<p>AM1. Necesidades cambiantes y demandantes de los usuarios.</p>

Tabla 2 - Análisis FODA del proceso actual de desarrollo de la aplicación móvil

3.3.3 Evaluación cuantitativa

Para evaluar el proceso de desarrollo actual en términos cuantitativos se consideró la cantidad de trabajo realizado en cada ciclo de desarrollo durante el tercer y cuarto trimestre del año 2016 en la aplicación móvil Sodimac. Esta revisión ayudo a determinar la velocidad del equipo, sabiendo aproximadamente cuanto puede hacer el equipo en futuros ciclos de desarrollo.

A continuación, se exponen las nuevas funcionalidades definidas por el encargado de la aplicación móvil para ser parte de entregables mensuales de la aplicación Sodimac durante la segunda mitad del año 2016. La Figura 15 ilustra el calendario de entregas.

F1. Homologación de información de productos: mostrar en la aplicación móvil la misma información sobre productos que el sitio web, manteniendo la coherencia y sincronización de los datos entre canales.

F2. Información detallada de tiendas: números de teléfono, servicios de todas las tiendas de Chile y además poder abrir Google Maps o Waze con la ruta desde la posición del usuario (utilizando el GPS) a la tienda seleccionada.

F3. Categorías de productos: navegación ordenada según categorías, bajo la misma taxonomía definida en el sitio web.

F4. Mejoras en el lector de códigos: incorporar un campo para introducir los códigos de producto (SKU) manualmente en caso de no poder ser identificados por la cámara del teléfono, además habilitar el uso del flash de la cámara y poder leer contenido de códigos QR.

F5. Carro de compras: listado de productos con precios y servicios asociados para la compra.

F6. Mapas de tienda: mapa de cada tienda de Chile que muestra la ubicación de los productos según el pasillo de la tienda en donde se encuentran.

F7. Listas de compra: poder crear listas por el usuario para agregar productos que luego pueden ser comprados.

F8. Redes sociales: función que permite compartir un producto en redes sociales.

F9. Chat de asistencia: chat de mensajería conectado con los operadores de servicio al cliente para la resolución de dudas.

F10. Login y registro: poder identificarse como usuario en la aplicación con la misma cuenta utilizada en el sitio web y además poder registrar una nueva cuenta.

F11. Click to call: botón que al ser pulsado devuelve una llamada al usuario para resolver dudas por parte del equipo de servicio al cliente.

F12. Cubicadores: calculadoras de cantidad de productos, como por ejemplo pisos, que al introducir un valor en metros (de la obra a realizar) entrega la cantidad exacta de cajas de pisos a utilizar.

F13. Videos de asistencia: tutoriales breves sobre cómo reparar una gotera, cambiar un neumático, preparar una mezcla de cemento o instalar una puerta.

F14. Promociones: reglas de negocio existentes en tiendas y el sitio web que no existen hoy en la aplicación, como comprar 3 productos y pagar 2 u obtener de regalo algún producto por la compra de otro.

F15. Notificaciones push: canal de mensajería instantánea que envía texto, imágenes, links externos o internos en la aplicación.

F16. Checkout: funcionalidad para comprar productos y servicios con despacho a domicilio o retiro en tienda.

F17. Información de despacho: simulador de entrega para despacho a domicilio y retiro en tienda (disponibilidades y costos asociados).

F18: Comparador de productos: comparación de características de productos similares.

C1: Junio	C2: Agosto	C3: Octubre	C4: Noviembre	C5: Diciembre
F1. Homologación de información de producto	F5. Carro de compras	F10. Login y registro	F14. Promociones	F17. Información de despacho
F2. Información detallada de tiendas	F6. Mapas de tienda	F11. Click to call	F15. Notificaciones push	F18. Comparador de productos
F3. Categorías de productos	F7. Listas de compra	F12. Cubicadores	F16. Checkout	
F4. Mejoras en lector de códigos	F8. Redes sociales	F13. Videos de asistencia		
	F9. Chat de asistencia			

Figura 15 - Hoja de ruta de desarrollo para la aplicación Sodimac, segundo semestre 2016

En la Figura 16 se muestra el trabajo realizado durante los meses de junio a diciembre de 2016 en base a la hoja de ruta comprometida en la Figura 15. El eje Y representa el número de funcionalidades (comprometidas versus terminadas al final de cada ciclo) y el eje X los ciclos como nuevas versiones de la aplicación. Como se puede notar, en ningún ciclo se pudo cumplir con las funciones comprometidas en la hoja de ruta, muchas de las que fueron consideradas inicialmente para un entregable finalmente no se realizaron por dependencias de terceros, no contar con los recursos para finalizar los desarrollos o la dificultad de estos. Esto llevo a que finalmente el equipo

no finalizara las funciones comprometidas dejándolas a medias o sin siquiera haber comenzado a trabajar en ellas. Para el equipo de desarrollo, esto podría significar inexperiencia al subestimar el esfuerzo de las tareas, falta de una metodología formal para la estimación de esfuerzo, capacidades técnicas o comunicación entre el equipo y los stakeholders.

Más preocupante aún son los ciclos C4 y C5 en donde no se entregó ninguna funcionalidad a los usuarios durante dos meses. Esto se agrava al considerar que parte de la estrategia de la compañía es ser competitivos en el comercio electrónico y el rubro del mejoramiento del hogar. El hecho de no haber entregado funcionalidades durante dos meses se traduce como falta de competencia por parte de la compañía hacia el segmento de usuarios y clientes que utilizan la aplicación móvil.

Por otra parte, existen muchas necesidades de los usuarios y clientes que día a día se ven reflejados en los comentarios que realizan en las tiendas de aplicaciones, muchos de ellos aluden a problemas como diferencias de precio entre la aplicación y las tiendas o el sitio web, o la ausencia de un medio para poder comprar productos en la aplicación. Estas cosas son importantes de resolver para la compañía, porque para los usuarios representan características y funciones básicas, esperadas por defecto en la aplicación móvi, dado que por ejemplo el sitio web móvil tiene la opción de comprar productos o su información sobre los productos es coherente con la información de las tiendas.

En la Figura 16, se puede ver que no se está entregando valor a los usuarios. Liberar solo tres versiones de un producto en seis meses durante la segunda mitad de 2016 siendo parte de un mercado altamente competitivo es impensado si se compara con las grandes compañías que liberan software múltiples veces durante un día [28], por lo que es necesario replantear el proceso de desarrollo para mejorar la productividad del equipo y también la calidad de las aplicaciones.

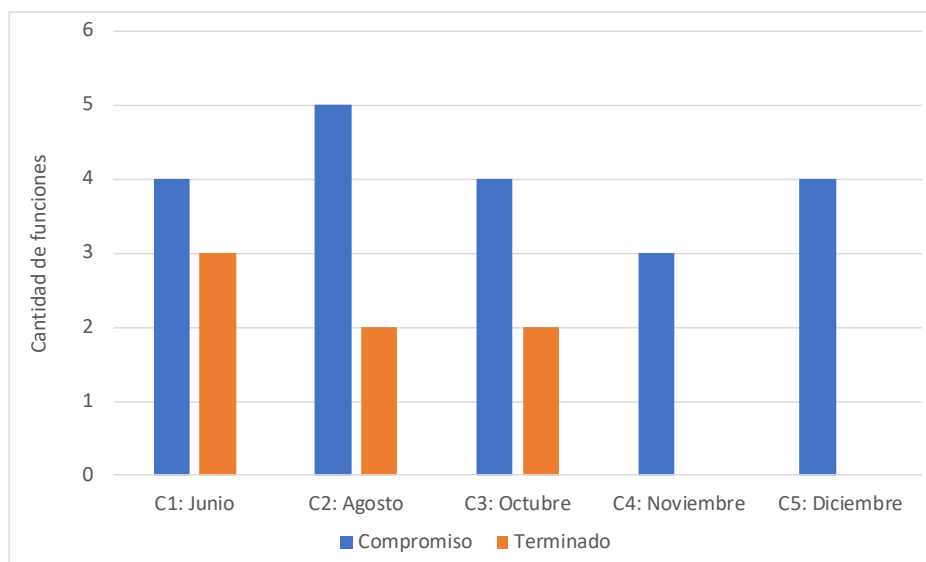


Figura 16 - Cantidad de funciones comprometidas versus terminadas durante junio a diciembre de 2016

3.3.4 Evaluación y desafíos del proceso

A pesar de que el proceso de desarrollo de la aplicación móvil se ha utilizado desde 2012, este no está exento de problemas y constantemente recibe críticas por parte de los stakeholders y del equipo de desarrollo a raíz de:

- Muchas tareas son abordadas desde el criterio y gusto personal de quien las toma, no contando con herramientas para su validación, datos cuantitativos, estudios y buenas prácticas de la industria que las sustenten y que incluyan a los usuarios y stakeholders.
- Al ser un proceso que muta entre ciclos, es complejo estimar los esfuerzos en el desarrollo y además el compromiso de hitos en fechas definidas para el negocio.
- Las funcionalidades son desarrolladas en su totalidad antes de ser puestas en manos de los usuarios finales, muchas veces esto ha llevado a que no sean aceptadas y finalmente son desechadas, perdiendo tiempo y dinero en el proceso.

El proceso utilizado por la compañía está basado en la planificación de gran parte del desarrollo de la aplicación móvil y podría ser efectivo en un ambiente en donde los requerimientos sean estables. En el caso del comercio electrónico, el ritmo de desarrollo de productos es rápido, el gusto de los usuarios cambia constantemente y existe mucha incertidumbre [29], por lo que se necesita un proceso que, tal como el producto, sea flexible y pueda adaptarse constantemente a las demandas de los usuarios, entregando a su vez transparencia a los miembros de la organización.

3.4 Arquitectura de la aplicación móvil

Esta se puede separar en dos capas: front-end y back-end. La primera refiere a la interfaz con la interactuar el usuario, como pantallas gráficas, flujos de navegación, comportamientos, botones y menús. La capa de back-end es responsable de la lógica de negocio que expone los datos de los sistemas de la compañía a los usuarios, como la información sobre productos y servicios.

3.4.1 Descripción de la capa de back-end

Como se puede ver en la Figura 17, actualmente la arquitectura de la aplicación móvil se basa en la extracción de datos desde los sistemas legados de la compañía, los cuales son exportados a hojas de cálculo y luego son cargados a una base de datos local que concentra los datos.

Uno de los problemas que ocurren con la arquitectura actual es la sincronización de la información y se puede ejemplificar con el proceso P3 de la Figura 17, el cual corresponde a una extracción y

carga de datos proveniente del sistema de promociones de precios para productos (SL4: Promociones en la Figura 17). Estos datos luego de ser cargados una vez por semana, no se mantienen sincronizados en tiempo real respecto a los cambios que puedan ocurrir sobre los precios, disponibilidad finita de los productos o la fecha de validez definida para una promoción. Esto hace que muchos productos se muestren desactualizados en las aplicaciones respecto a la información disponible en las tiendas o el sitio web, dado que el stock, validez de precios y promociones pueden cambiar varias veces en una semana (dependiendo de la estrategia comercial de la compañía), provocando que los usuarios presenten quejas por la diferencia de información.

Este modelo de extracción de información plantea problemas como inconsistencias basadas en la temporalidad de los datos (como la puntualidad, latencia y edad), afectando la experiencia de los usuarios de la aplicación móvil y en el fondo a la marca tras la compañía, sin mencionar el disgusto de los stakeholders. Es necesario entonces considerar que el negocio siempre generará cambios en la información de los productos o servicios bajo una frecuencia en parte no definida y la arquitectura que da soporte a la aplicación móvil debe estar preparada para asumir estos cambios.

En el aspecto técnico, los datos que son extraídos desde los sistemas legados son luego cargados en una base de datos SQL (BD1: Concentrador de información en la Figura 17), montada en un computador de escritorio regular que hace el rol de servidor a donde se conecta la aplicación móvil para extraer los datos a través de un servicio web (WS1: Webservice en Figura 17).

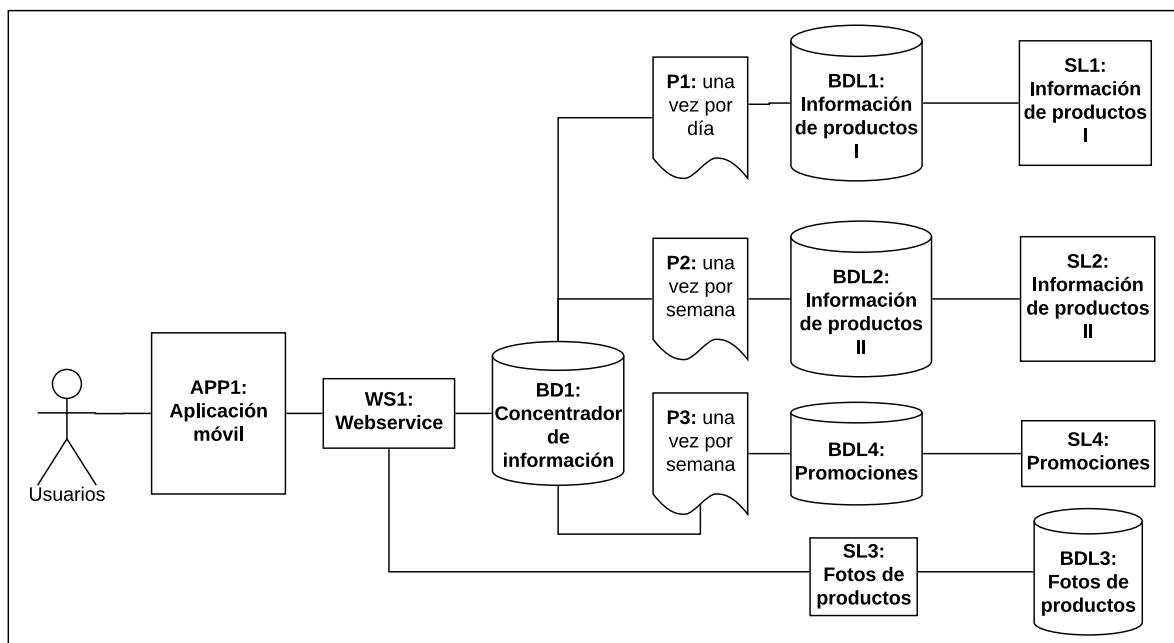


Figura 17 - Arquitectura actual de la aplicación móvil

A continuación, se describen las principales entidades de la arquitectura actual del sistema de la aplicación móvil:

APP1: Aplicación móvil, cliente ejecutado en un teléfono móvil.

WS1: Webservice, interfaz que reside en un computador de escritorio que cumple el rol de servidor, al cual se conecta la aplicación móvil para comunicarse con la base de datos local que contiene los datos desplegados en la aplicación móvil.

BD1: Concentrador de información, base de datos SQL que consolida la información que es cargada de forma manual a través de hojas de cálculo exportadas desde los sistemas legados.

SL1. Información de productos I, sistema que administra la información de tiendas físicas como horarios, direcciones, planos de las tiendas, productos exclusivos de tiendas físicas, etc. Toda esta información está almacenada en una base de datos BDL1: Información de productos I y es extraída una vez al día mediante el proceso P1 para ser cargada en BD1: Concentrador de información.

SL2. Información de productos II, información de productos pertenecientes al catálogo online de productos (no se incluyen los productos exclusivos de tienda), contiene en su base de datos BDL2: Información de productos II la meta data de productos como precios, marca, descripción, ficha técnica y productos relacionados. Estos datos son extraídos mediante un proceso P2 una vez por semana para ser cargados en BD1.

SL3. Fotos de productos, sistema que gestiona el almacenamiento de las fotos de los productos en sus distintas dimensiones y formatos en una base de datos BDL3: Fotos de productos. Este servicio se puede consultar en línea a través de WS1: Webservice.

SL4. Promociones, sistema que almacena los precios promocionales asociados a los productos. De este sistema, se extrae de forma programada (una vez por semana) y mediante un proceso P3, una hoja de cálculo con datos que luego es cargada en BD1 una vez por semana.

3.4.2 Descripción de la capa de front-end

Respecto a la capa de front-end, es complejo dar un diagnóstico general dado que la aplicación móvil está estructurada con patrones de diseño distintos entre versiones y lógicas de negocio que han sido manipuladas por distintos proveedores de desarrollo en los últimos años, perdiendo la integridad conceptual del código planteado en las versiones de Android e iOS (este último basado en Objective-C).

3.4.3 Evaluación y desafíos de la arquitectura

Para la evaluación tecnológica de la aplicación móvil, se consideraron como muestra los datos provenientes de las calificaciones existentes en las tiendas de aplicaciones de Apple y Google para la aplicación Sodimac. En las Figura 18 y Figura 19 a continuación, se ven las calificaciones correspondientes a los meses de junio a diciembre de 2016, las cuales rondan entre la nota 1 y 3 de un máximo de 5.

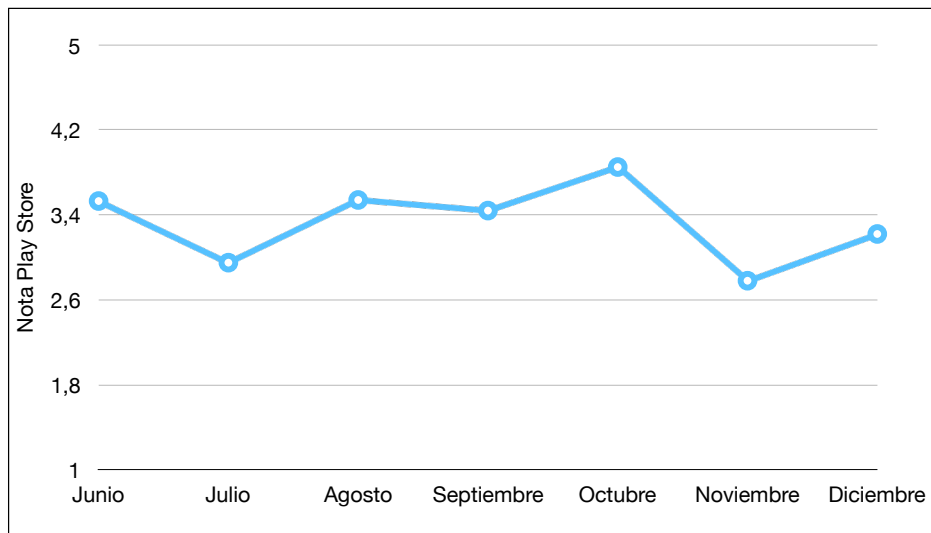


Figura 18 - Promedio de evaluación mensual junio a diciembre de 2016, Android [30]

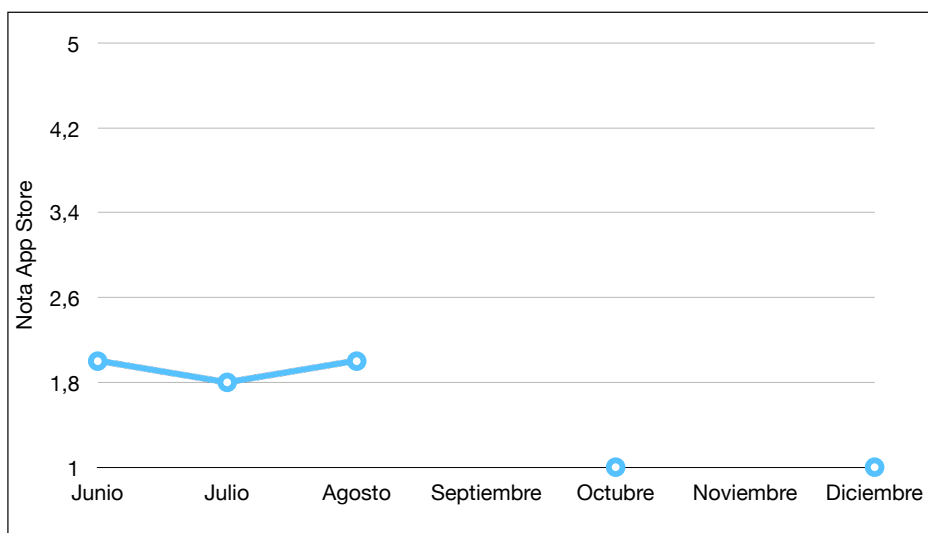


Figura 19 - Promedio de evaluación mensual junio a diciembre de 2016, iOS [31]

En la Figura 19 se puede ver como en los meses de septiembre y noviembre de 2016 no hubo valoraciones para la versión iOS de la aplicación móvil.

Según el informe *App Store Ranking, Top Charts* [32] de App Annie, durante el mes de diciembre de 2016, 8 aplicaciones móviles dentro del top 10 de las mejores aplicaciones móviles de la categoría compras gratuitas en Chile tanto en Play store (Android) como App Store (iOS), tienen una evaluación con nota 4 o superior, esto refleja que una calificación esperada para una aplicación destacada dentro de la categoría compras gratuitas es en su mayoría una nota 4 o superior. En el caso de la aplicación Sodimac su nota se ha mantenido históricamente entre 2,0 y 3,9.

Si se hace una revisión en detalle de los comentarios asociados a las calificaciones, destacan las malas evaluaciones en las tiendas de aplicaciones. Estos comentarios son generados por variados motivos, como inconformidad frente a los precios de los productos, disponibilidad de marcas, disponibilidad de productos, etc. Sin embargo, hay muchos otros comentarios que se destacan por tener un origen técnico y el equipo de desarrollo de la aplicación móvil los ha podido identificar como parte de los defectos del sistema actual (front-end y back-end) de la aplicación Sodimac. Por otro lado, Android tiene mejores evaluaciones que iOS, esto puede estar dado por que distintos proveedores de desarrollo han trabajado en cada sistema operativo, notándose la falta de coherencia o integridad conceptual en el diseño del código y diferencias en la calidad de la codificación entre versiones.

A continuación, algunos comentarios extraídos de las tiendas de aplicaciones de iOS [30] y Android [31] que hacen referencia directa a la calidad de la información y de manera indirecta al software y su arquitectura:

iOS

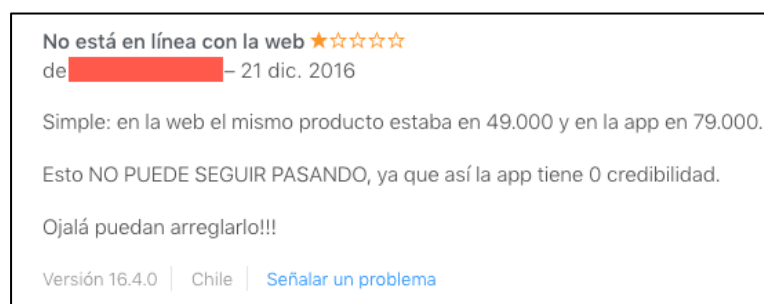


Figura 20 - Evaluación iOS: "no está en línea con la web" [31]

Diagnóstico al comentario de la Figura 20: las diferencias de precio vienen dadas por la frecuencia de carga de los datos de precios provenientes de SL1: Sistema de Información de Productos I. Esta

carga se hace una vez al día dado que no existe un servicio en alta disponibilidad que exponga datos como los precios en tiempo real.



Figura 21 - Evaluación iOS: "Insuficiente" [31]

Diagnóstico al comentario de la Figura 21: error detectado en una de las versiones de la aplicación donde no se estaba filtrando el mostrar productos sin foto. Esta lógica existía en versiones previas de la aplicación, pero dado que no hay pruebas de regresión que validen que las modificaciones en el código no generarán bugs, no existe control respecto a los cambios en cada versión de la aplicación.

Respecto a la rapidez en el despliegue de contenidos, se debe en general a utilizar una arquitectura no pensada en la alta disponibilidad, además de que no existe ninguna optimización respecto a la carga asíncrona de contenidos en la capa de front-end.

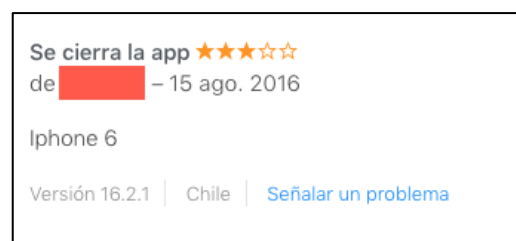


Figura 22 - Evaluación iOS: "Se cierra la app" [31]

Diagnóstico al comentario de la Figura 22: los usuarios de sistema operativo iOS que poseen la versión 16.2.1 de la aplicación móvil y que utilizan un iPhone 6, experimentan el cierre de la aplicación al abrirla. Al no tener este tipo de teléfono para pruebas (físico), el equipo de desarrollo no pudo replicar este error para encontrar una solución.

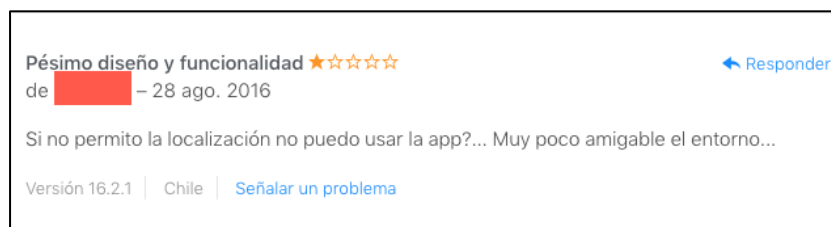


Figura 23 - Evaluación iOS: "Pésimo diseño y funcionalidad" [31]

Diagnóstico al comentario de la Figura 23: cuando los usuarios no aceptan compartir su ubicación con la aplicación al momento de usarla por primera vez (GPS), ésta no deja que el usuario avance en el flujo de navegación, bloqueando el uso de la aplicación. Este tipo de lógica en el código está errada al no dar una alternativa a los usuarios no desean compartir información de uso con la aplicación.

Android

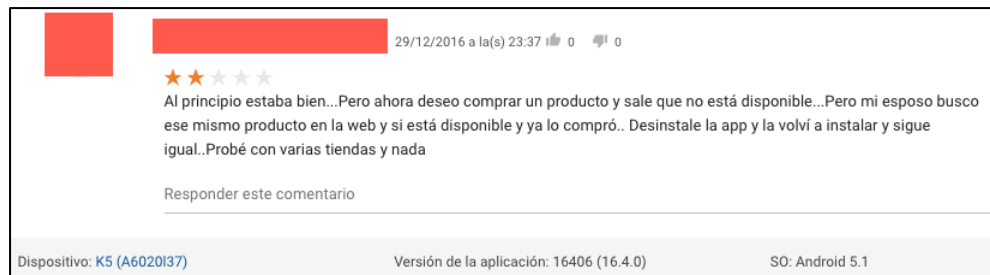


Figura 24 - Evaluación Android: "Al principio estaba bien..." [30]

Diagnóstico al comentario de la Figura 24: en la gran mayoría, los problemas asociados a diferencias en la información entre canales (sitios web y tiendas versus la aplicación móvil) tiene que ver con la frecuencia de actualización de los datos proveniente de los sistemas legados, dado que la arquitectura de back-end de la aplicación móvil hace que la sincronización de datos en tiempo real no sea posible dada la escasa escalabilidad y nula alta disponibilidad de la arquitectura actual.

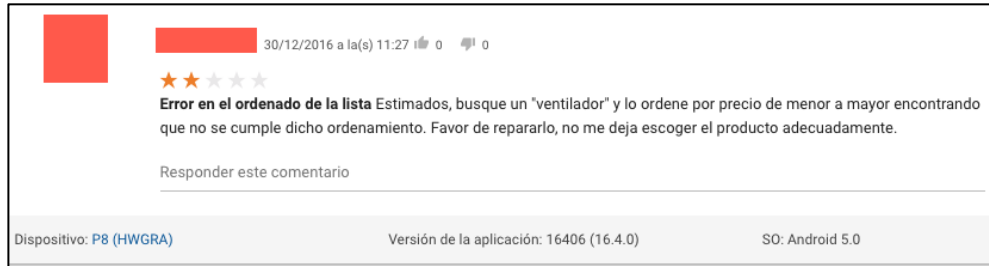


Figura 25 - Evaluación Android: "Error en el ordenado de la lista" [30]

Diagnóstico al comentario de la Figura 25: un error en la lógica de los métodos de orden para las páginas de resultados de búsqueda fue introducido en la versión 16.4.0 de la aplicación. El no contar con pruebas unitarias y de regresión hizo que este tipo de error no fuera evidenciado previamente. Por otra parte, las pruebas de aceptación usuaria también fallaron.



Figura 26 - Evaluación Android: "No sirve" [30]

Diagnóstico al comentario de la Figura 26: este error solo está presente en Android en una versión particular de la aplicación, provocando que al ejecutar la selección de una opción al inicio de la aplicación y posterior a la instalación por primera vez en un teléfono, la aplicación se congelara.

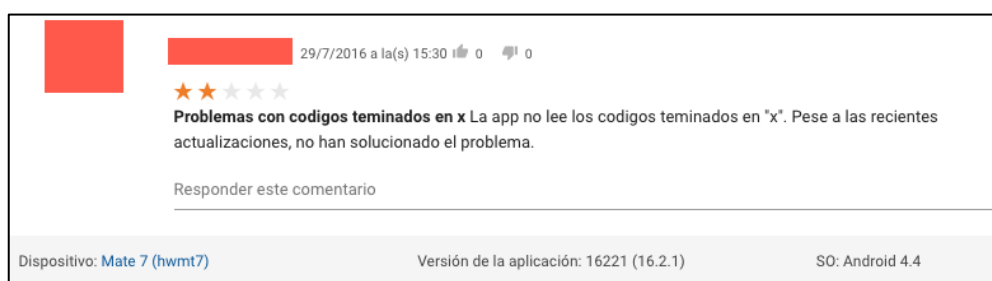


Figura 27 - Evaluación Android: "Problemas con códigos terminados en x" [30]

Diagnóstico al comentario de la Figura 27: los códigos de producto terminados en “X” no son leídos por el escáner de códigos de barra, siendo ésta una de las principales funciones dispuestas para la asistencia en tienda de los usuarios y clientes. Este problema ocurre debido a que el servicio utilizado para la búsqueda de productos fue desarrollado pensando solo en que los códigos serían siempre números, entonces el tipo de dato que incluye letras es asumido como inválido.

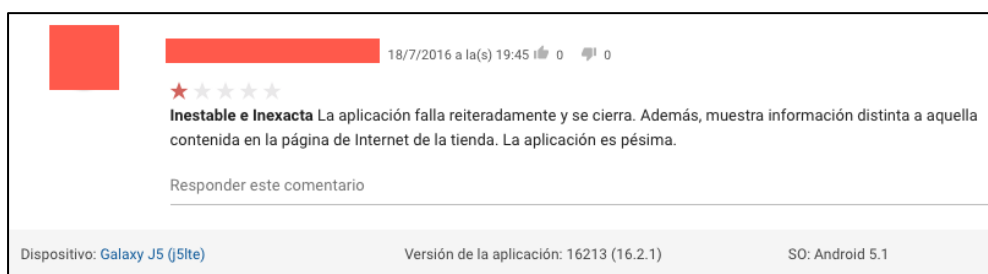


Figura 28 - Evaluación Android: "Inestable e inexacta" [30]

Diagnóstico al comentario de la Figura 28: nuevamente se evidencia la diferencia en la información respecto a otros canales, en este caso el sitio web. Respecto a la robustez de la aplicación, sucede que muchos escenarios no están considerados por la aplicación y no hay una reacción definida, ocasionando caídas (cierre) o congelamientos, lo cual puede suceder al buscar un código no existente de producto o pulsar reiteradas veces un botón.

En resumen, los comentarios son numerosos y variados en las tiendas de aplicaciones. Solo en Play Store (Android) hay 80 comentarios con una calificación 2 en el periodo de junio a diciembre de 2016 para la aplicación Sodimac, por lo que un buen método para comprender sus orígenes sería clasificarlos según un tipo definido para poder abordar luego su resolución de forma organizada, tal como se hace en la Tabla 3, en donde se agruparon los problemas bajo una tipología asociada a la arquitectura de la información, la estabilidad de la plataforma o lógicos.

Identificador	Tipo	Descripción
T1	Arquitectura de la información	Diferencias de información entre distintos canales de la compañía, considerando que la actual arquitectura que une a la aplicación móvil con los sistemas legados presenta problemas en su escalabilidad, estabilidad, seguridad y sincronización de datos.
T2	Estabilidad	Reiteradas caídas y pantallas congeladas frente a errores no controlados o falta de respuesta por parte de los servicios debido a la saturación por múltiples consultas a una arquitectura no pensada en la alta disponibilidad de sus servicios.
T3	Lógicos	Errores presentes que nacen de las interacciones del usuario con la interfaz de la aplicación, como la introducción de datos, selección de opciones o gestos táctiles en la pantalla (tap, swipe, pinch, etc).

Tabla 3 - Tipologías de errores sobre comentarios de usuarios en las tiendas de aplicaciones

Otra forma de evidenciar problemas en la aplicación móvil es el reporte de error que entrega Apple. Lamentablemente Google en 2016 no entregaba información sobre bloqueos o excepciones en la aplicación móvil, por lo que no se pudo considerar, aun cuando Android hoy representa en términos de uso casi un 80% del total de aplicaciones instaladas [33].

En la Figura 29 y Figura 30, se puede ver como a lo largo del tiempo y de forma sostenida se han mantenido errores de dos tipos en iOS: bloqueos y excepciones. Estos se manifiestan a través del cierre de la aplicación o el congelamiento de la pantalla sin poder realizar acción alguna más que cerrar la aplicación y volverla a abrir para volverla a utilizar. En el caso de la Figura 29, se muestran los errores de sistema (bloqueos y excepciones) entregados por Apple en App Store Connect (portal de gestión de aplicaciones móviles de Apple).

Otra fuente considerada fue Google Analytics, herramienta de analítica que se utiliza para dar seguimiento al comportamiento de los usuarios en la aplicación Sodimac, y en donde se registraron errores no controlados presentados al usuario a través de mensajes técnicos en iOS, lo cual se evidencia en la Figura 30.

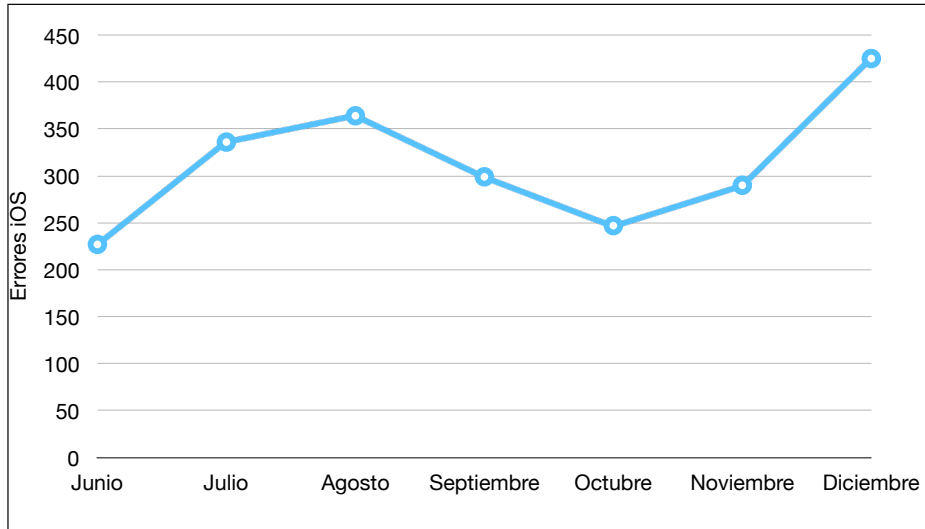
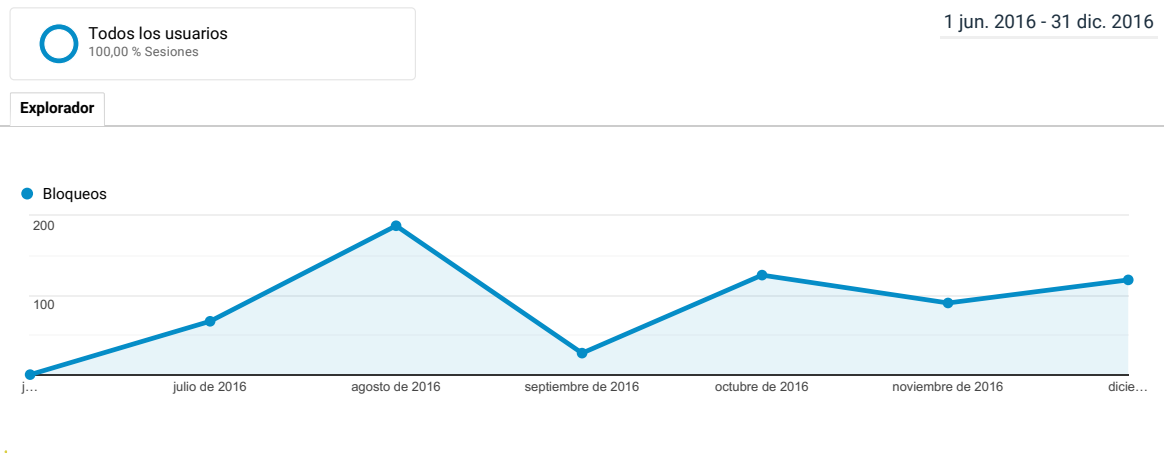


Figura 29 - Errores (bloqueos y excepciones) en la Aplicación Sodimac, iOS, 2016 [31]

Bloqueos y excepciones



Versión de la aplicación	Bloqueos	Excepciones
	615 % del total: 100,00 % (615)	615 % del total: 100,00 % (615)
1. 16.1.3.8	421 (68,46 %)	421 (68,46 %)
2. 16.3.2	77 (12,52 %)	77 (12,52 %)
3. 16.2.1	56 (9,11 %)	56 (9,11 %)
4. 16.4.1	42 (6,83 %)	42 (6,83 %)
5. 16.3.1	12 (1,95 %)	12 (1,95 %)
6. 16.3.0	6 (0,98 %)	6 (0,98 %)
7. 16.4.0	1 (0,16 %)	1 (0,16 %)

Figura 30 - Errores en App Sodimac, iOS de junio a diciembre de 2016 [33]

Más allá de estos reportes que entrega App Store Connect y Google Analytics sobre iOS, no hay un catastro de cuál es el origen de estos errores, quedando sin revisión y a la espera de ser evidenciados de forma masiva por los usuarios o en su defecto afectando algún aspecto importante en el uso la aplicación móvil.

El hecho de que no haya control sobre los errores inducidos en el código de la aplicación móvil afecta directamente a la calidad del producto, lo cual se ve en las evaluaciones y comentarios de los usuarios en las tiendas de aplicaciones debido a su deficiente arquitectura, pobre estabilidad e inesperado comportamiento lógico.

Si se mira al software y su arquitectura como una característica del producto, la cual da soporte al valor que se desea entregar a los usuarios, entonces el contar con deficiencias técnicas responde a una deuda que debe ser abordada antes de continuar desarrollando otros aspectos del producto, como su difusión o el desarrollo de nuevas funciones para los usuarios, dado que se podrían estar perdiendo recursos en la inversión de un producto con defectos en sus cimientos y acumular problemas técnicos que luego se deberán resolver, lo que podría llevar a perder la confianza de los usuarios.

El costo de que un usuario elimine una aplicación es grande si se considera que, para volver a instalar la misma aplicación, el usuario debe buscar a conciencia nuevamente la aplicación en la tienda de aplicaciones móviles, tener espacio suficiente en el teléfono móvil, aceptar los permisos de configuración requeridos, registrarse o identificarse para finalmente poder utilizar la aplicación. En general es un proceso de varias fases que fácilmente puede hacer perder el interés a cualquier usuario si la experiencia no es agradable. Por otra parte, la performance de la aplicación móvil puede hacer que los usuarios la conserven o la desinstalen [34].

3.5 Análisis de la situación actual

Para consolidar las características que definen el proceso actual de desarrollo de la aplicación móvil y la arquitectura sobre la cual está desarrollada, se utilizó el método KPT, tal como se describe en la Tabla 4.

Para los dos aspectos de la situación actual del desarrollo de la aplicación móvil (proceso de desarrollo y arquitectura) se realizó una retrospectiva particular. Cada retrospectiva se realizó en un tiempo total de 2 horas y cada sección tuvo 15 minutos asignados para su desarrollo.

Retrospectiva KPT – Proceso de desarrollo y arquitectura actual	
Objetivo	Analizar y concluir ideas sobre las evaluaciones cualitativas y cuantitativas del proceso de desarrollo de la aplicación móvil y su arquitectura actual.
Procedimiento	<p>Asistentes: 8 personas, todas juntas en cada sesión, encargado de la aplicación móvil, encargado experiencia usuaria, jefes de proyecto, gerente de TI, gerente desarrollo de venta online, subgerente de experiencia usuaria y subgerente comercial de venta online.</p> <p>Duración: 2 sesiones (una por cada tema, una cada día), 2 horas cada una, con 15 minutos para la revisión de cada aspecto de KPT.</p> <p>Equipo y material: pizarra con matriz KPT y tarjetas adhesivas para desarrollar las ideas sobre cada aspecto de la retrospectiva.</p> <p>Dinámica: el moderador de la reunión (agile coach) define lo que es una retrospectiva y comenta las particularidades del método KPT. Posteriormente se entregan tarjetas a cada asistente para desarrollar las ideas necesarias para el análisis.</p>
Instrumento	<ol style="list-style-type: none"> 1. Durante 15 minutos cada asistente escribe en una tarjeta una idea referente a las cosas buenas a mantener en el equipo (<i>Keep</i>). 2. Al terminar los 15 minutos el moderador recoge las tarjetas y los pega en la pizarra en la cuadrícula <i>Keep</i>, luego lee cada tarjeta y los asistentes definen cuáles son similares para agruparlos. 3. Se repite el proceso con el aspecto <i>Problem</i>. 4. Una vez que están los problemas agrupados en la sección <i>Problem</i>, se solicita que, se definan acciones a implementar para resolver cada uno, esto constituye <i>Try</i>. 5. Para este último aspecto (<i>Try</i>) solo se consultó al equipo de desarrollo, ya que parte de su rol es ser responsables del producto y proceso, además del desarrollo e implementación de la plataforma tecnológica. 6. Terminado el tiempo, el moderador agrupa las tarjetas con ideas propuestas para abordar los problemas y comenta que por cada idea debiera existir un responsable que esté a cargo de buscar una solución o mejora. 7. Una vez asignados los responsables se discute sobre el tema en general, para consolidar las visiones de cada asistente respecto a los problemas a afrontar y las acciones que se llevaran a cabo para resolverlos.

Tabla 4 - Retrospectiva KPT – Proceso de desarrollo y arquitectura actual

3.5.1 Resultados: retrospectiva del proceso de desarrollo

A continuación, se detallan las ideas por cada aspecto del método KPT para el proceso de desarrollo actual de la aplicación móvil, en donde las cosas positivas que el equipo debe mantener son mantener un proceso simple, poca documentación y un nivel de adopción fácil pensando en el traspaso de conocimiento.

Respecto a los problemas o aspectos a mejorar, estos apuntan en general a la rigidez del proceso y a la poca o nula organización existente en él, lo cual lleva a contar con iteraciones que no permiten lograr una productividad sostenida en el tiempo y una mejora en las actividades realizadas por el equipo para los stakeholders y usuarios finales.

De cara a la mejora de estos aspectos definidas en el cuadro *Try*, se encuentra la necesidad de contar con un proceso flexible, que se adapte a los cambios de requerimientos, escenarios comerciales y gustos de los usuarios, además de que el proceso cuente con roles, hitos, ceremonias y actividades claras, bajo una organización establecida que permita que en cada iteración el proceso sea el mismo pero que pueda mejorar, adaptarse y evolucionar tal como las aplicaciones.

Detalles de los aspectos a conservar, *Keep*:

KD1. Contar con un proceso de fácil adopción, los stakeholders y miembros del equipo de desarrollo entienden el proceso de desarrollo actual. El proceso es simple de comprender y se puede adoptar rápidamente.

KD2. Poca documentación, si bien existe documentación, ésta es escasa, pero contiene al menos ciertas líneas comentadas en el código y algunos esquemas de arquitectura que permiten tener una noción clara de cómo funciona la aplicación móvil, su flujo de datos y lógica de negocio.

KD3. Simplicidad en las etapas del desarrollo, el proceso de desarrollo es simple, con pocas etapas y casi nula dependencia de otras áreas.

KD4. Auto gestionable, la forma de trabajo, horarios, priorización y muchos otros aspectos del proceso de software actual son definidos por el mismo equipo, siendo responsables por completo del proceso y su aplicación en el día a día.

Detalle de los problemas, *Problem*:

PD1. No hay visibilidad de avance, a medida que el desarrollo de software avanza, no hay una exposición de los objetivos alcanzados, metas cumplidas o hitos logrados. Esto hace que los

stakeholders desconfíen del proceso y de la aplicación móvil como producto al no ver avances que fundamenten la inversión.

PD2. Estimación de esfuerzo inexacta, muchas tareas se sub o sobre-estiman, perdiendo en ocasiones recursos como tiempo y dinero.

PD3. Priorización en base a juicio experto, el orden en el cual los requerimientos son abordados no está dado por el beneficio hacia los usuarios finales. Esto hace que muchas sean de nula utilidad para los usuarios, lo cual es evidenciado en las evaluaciones que ellos realizan en las tiendas de aplicaciones.

PD4. Los requerimientos son fijos, el desarrollo actual sigue un formato estricto en términos de requerimientos, en que estos son tomados y luego difícilmente son modificados hasta ver finalizada una función en su totalidad.

PD5. No se ven funciones hasta el final del desarrollo, el proceso actual no permite probar funcionalidades antes de ser desarrolladas y completadas en su totalidad.

PD6. No hay roles definidos, las personas que son parte del equipo cumplen distintos roles, los cuales no están especificados ni detalladas las tareas que cumple cada uno.

PD7. Resistencia al cambio, frente a los cambios en los requerimientos, tecnología, procesos e incluso el mercado, hay una gran resistencia, quizás dada por el desconocimiento de estos mismos, la falta de habilidades técnicas en el desarrollo de software y los prejuicios sobre los usuarios y su comportamiento frente a la tecnología móvil.

Detalle de los aspectos a intentar, Try:

TD1. Visibilidad de avance a stakeholders, el nuevo proceso debe ser transparente para todos los interesados en el desarrollo de la aplicación móvil, y debe permitir ver de forma clara el estado de avance o trabajo pendiente para la entrega de software.

TD2. Utilizar un método riguroso de estimación de esfuerzo, contar con un método para la estimación del esfuerzo que sea acordada y consensuada por el equipo, entregando una proyección de la capacidad real del equipo, de forma que los stakeholders puedan conocer también el ritmo de avance del equipo y su capacidad.

TD3. Fallar rápido y barato, la aplicación de prácticas de desarrollo incremental y evolutivo permiten que se puedan validar funciones rápidamente sin contar con versiones finales y completamente terminadas.

TD4. Producir entregables constantes y en breve tiempo, el entregar software constantemente a los usuarios para la obtención de retroalimentación hace que el producto evolucione, se adapte a las cambiantes necesidades de los usuarios y entregue el valor necesario para mantenerlas instaladas y en uso.

TD5. Usar un enfoque de desarrollo adaptativo, adaptarse a los cambios del mercado, avances en la tecnología y necesidades de los usuarios.

TD6. Priorizar en base a las necesidades de los usuarios, todas las funciones implementadas en las aplicaciones deben nacer de las necesidades de los usuarios y clientes de la compañía.

TD7. Saber quiénes son nuestros clientes y usuarios, cada versión de la aplicación móvil debe incluir la retroalimentación obtenida en la validación del entregable. Esto debe venir de la mano de los usuarios finales.

TD8. Equipos y roles, el nuevo proceso debe contar con roles claros y definidos, con responsabilidades para cada miembro del equipo.

TD9. Instancias de seguimiento de desarrollo en equipo, el nuevo proceso debe contar con prácticas que permitan revisar el avance en el día a día y dar seguimiento al plan para evitar desviaciones que retrasen la entrega de software.

TD10. Incluir a los stakeholders en el proceso de desarrollo, este aspecto es fundamental para obtener el punto de vista de negocio de los stakeholders. Por otro lado, la retroalimentación de los stakeholders puede agregar información relevante para el desarrollo del producto.

3.5.2 Resultados: retrospectiva de la arquitectura

Respecto a la retrospectiva enfocada en la arquitectura actual de la aplicación móvil, las cosas que se deben mantener son pocas, como la utilización de una capa intermedia que permita dar robustez a la aplicación frente a posibles caídas de los sistemas legados, el utilizar una lógica simple en términos funcionales de cara a los usuarios y finalmente es muy valorado el conocimiento adquirido y detallado que existe de cada sistema legado de la compañía.

Dentro de los aspectos negativos está el contar con una arquitectura física que no permite la escalabilidad frente a la alta demanda de consultas, la inexistencia de pruebas unitarias y de regresión que permitan evitar nuevos y recurrentes errores introducidos en cada versión, las cargas de datos no sincronizadas con otros canales de la compañía y finalmente la baja calidad del código de la aplicación móvil. A continuación, el detalle de cada aspecto.

Detalles de los aspectos a conservar, *Keep*:

KA1. Capa intermedia, el contar con una capa que replique datos y permita funcionar de forma autónoma frente a la caída de un sistema legado es algo que permite la resiliencia.

KA2. Funciones de baja complejidad, las lógicas de programación implementadas hoy en la aplicación móvil son simples, lo cual permite su comprensión y modificación.

KA3. Conocimiento detallado de cada sistema, la arquitectura actual de la aplicación móvil convive en un entorno conocido de sistemas legados, consumiendo desde los orígenes los datos desplegados en las aplicaciones.

Detalle de los problemas, *Problem*:

PA1. Frecuencia de sincronización de datos, la frecuencia hoy establecida y dada por las capacidades físicas de la arquitectura hacen que los datos desplegados en las aplicaciones sean antiguos y no son coherentes con la información que se muestra a los clientes.

PA2. Arquitectura de baja disponibilidad, unos pocos usuarios concurrentes extras (aproximadamente un 10% más) hacen que inmediatamente las respuestas a peticiones sean más lentas o simplemente generen un error de tipo *time out*.

PA3. No hay pruebas del código, en ambas capas: front-end y back-end, no existen pruebas de tipo alguno para la validación de funciones, esto causa que entre versiones existan errores introducidos que luego son evidenciados por los usuarios finales.

PA4. Lenguaje de front-end desactualizado, en particular las versiones iOS de las aplicaciones siguen utilizando el mismo lenguaje con el cual se construyeron en 2012, Objective-C. Hoy existe Swift, el cual trae considerables ventajas por sobre su antecesor y para las aplicaciones iOS hoy es el estándar.

PA5. Lentitud en pasos a producción, el proceso de paso a producción de front-end y back-end involucran procesos manuales de compilación y publicación. A eso se debe sumar que el desarrollo de las aplicaciones es particular para cada sistema operativo (Android e iOS), por lo que las UAT realizadas por el encargado de la aplicación móvil es doble. Todo esto hace que el proceso de publicar una nueva versión de las aplicaciones sea lento y complejo.

PA6. Falta de integridad conceptual, el diseño del software para front-end y back-end ha sido manipulado por varios programadores en los últimos años, lo que ha generado una pérdida en la integridad del código, afrontando de formas diversas el desarrollo de funciones y lógica. Esto ha hecho que las aplicaciones se comporten de maneras inesperadas, lentas y el mantenimiento de

código antiguo también es complejo dado que muchas veces no está comentado y difiere en su lógica entre ambos sistemas operativos.

Detalle de los aspectos a intentar, Try:

TA1. Arquitectura distribuida, contar con una arquitectura no monolítica para asegurar la baja dependencia entre componentes.

TA2. Arquitectura de alta disponibilidad, es necesario contar con escalamiento automático de recursos y servicios para los entornos productivos de back-end.

TA3. Código de front-end unificado, combinar el código de front-end de Android e iOS en uno solo, hará que la integridad conceptual del diseño sea mayor, y la velocidad de la codificación y pruebas también aumenten.

TA4. Umbrales, los datos que inevitablemente deberán continuar siendo cargados sin una conexión en línea debido a la baja disponibilidad del sistema legado de origen, no debieran ser transportados vía cargas masivas en cada ciclo, sino implementar una estrategia que luego de una carga masiva inicial, permita cargar solo la diferencia de información modificada en el sistema de origen para evitar problemas de rendimiento.

TA5. Pruebas automatizadas, para evitar errores introducidos en el código entre versiones lo mejor es implementar pruebas unitarias y de regresión automatizadas.

TA6. Bases de datos NoSQL, el manejo de grandes volúmenes de datos requiere de un modelo de base de datos que escale horizontalmente y que permita alcanzar un mayor rendimiento en cuanto a la carga y manipulación de los datos.

TA7. CI/CD, frente a constantes puestas en producción para la rápida obtención de retroalimentación de la aplicación móvil, es importante que cada programador pueda integrar continuamente sus cambios en el código con el resto, a través de prácticas como CI y disponer rápidamente versiones a producción mediante procesos automáticos de compilación, pruebas y publicación como CD.

3.6 Resumen

Parte de la motivación de realizar un análisis interno en la organización que cuestione los actuales procesos de desarrollo y arquitectura, es la necesidad de sobrevivir en términos comerciales en el altamente competitivo escenario del comercio electrónico. Es casi imposible que hoy una compañía dedicada al mercado minorista con una importante presencia en Internet no pueda validar las necesidades de sus clientes a través de funcionalidades en su aplicación móvil.

Hoy el diseño de plataformas web y de aplicaciones ha cambiado su estrategia hacia un concepto llamado *Mobile First* [1] en donde el diseño está enfocado primeramente en desarrollar los formatos móviles y luego pensar en el resto de las plataformas como versiones de escritorio o tablet de los sitios web.

La situación actual de la compañía es crítica respecto a desarrollo y arquitectura de software móvil. No existen procesos formales, roles definidos, hardware o servicios robustos que permitan que la aplicación móvil crezca en términos de usuarios, por lo que hoy es imposible pensar si quiera en publicidad, porque el solo hecho de tener más visitas generaría el colapso del sistema actual, sin contar la mala experiencia que vivirían los usuarios al enfrentarse a una aplicación móvil con datos desactualizados e incorrectos, funciones con errores y cierres inesperados. Es por esto por lo que urge un cambio en ambos aspectos (proceso y arquitectura), reformulando completamente la tecnología y la forma en que se aborda el desarrollo de productos guiado por una mirada de negocio orientada a la entrega de valor constante, poniendo al cliente en el centro y manteniendo la omnicanalidad del servicio en todos los canales de la compañía.

4. Solución

Tal como el escenario actual en que se desenvuelve la compañía presenta múltiples y complejos problemas asociados a distintas áreas del desarrollo de la aplicación móvil, la solución a ellos también se encuentra diversificada en varios aspectos, los cuales serán revisados en el presente capítulo. Estos pueden ser clasificados en: procesos y tecnología, ambos guiados por una estrategia de negocio. Para cada uno de los problemas presentados en el capítulo anterior, se desarrollaron propuestas de solución elaboradas con stakeholders, clientes y usuarios de la aplicación móvil.

Se revisará la estrategia de negocio de la aplicación Sodimac, la cual fue desarrollada junto a los stakeholders de la compañía y fue plasmada en un Product Vision Board [18]. Luego, respecto al proceso de desarrollo de software, se comentará el proceso elegido por el equipo de la aplicación móvil y como este nuevo enfoque se alinea con el contexto comercial de la compañía y sus cambiantes necesidades. Finalmente, se revisará en detalle cada aspecto de la nueva arquitectura desacoplada que permitirá el escalamiento y flexibilidad de recursos para soportar una alta demanda de usuarios.

4.1 Estrategia de negocio

Desde hace unos años se ha posicionado el termino *Mobile First* [1], el cual habla de pensar primero en los usuarios de dispositivos móviles y luego en otros formatos como tablet o escritorio para el diseño de software. Esto último viene dado por el creciente aumento y uso de este tipo de dispositivos a nivel mundial, donde muchos usuarios solo poseen un teléfono móvil para navegar por Internet y no un computador de escritorio. Según Statista, en 2016 un 62.9% de la población mundial ya poseía un teléfono móvil y para el 2019 el pronóstico es que el número de teléfonos móviles supere los 5 billones [35]. En Chile, la cantidad de teléfonos móviles personales (no empresas) llega a los 26 millones durante 2019 [36].

Desde el lanzamiento del iPhone hace más de 10 años, y el lanzamiento de las tiendas de aplicaciones de Apple y Google, se han visto nuevos casos de uso, modelos de negocio e innovación tecnológica que ha alimentado este dinámico mercado que en una década ha presentado un crecimiento enorme. En la Figura 31 se resume un estudio de App Annie [37], el cual comenta que el valor de la economía global de las aplicaciones móviles crecerá desde los \$1.3 trillones registrados en 2016 a los \$6.3 trillones en 2021, esto gracias al creciente del uso de dispositivos móviles y la relación entre usuarios y aplicaciones. El uso de las aplicaciones móviles también crecerá, pasando de 1.6 trillones de horas de uso en 2016 a 3.5 trillones en 2021 según el mismo estudio.

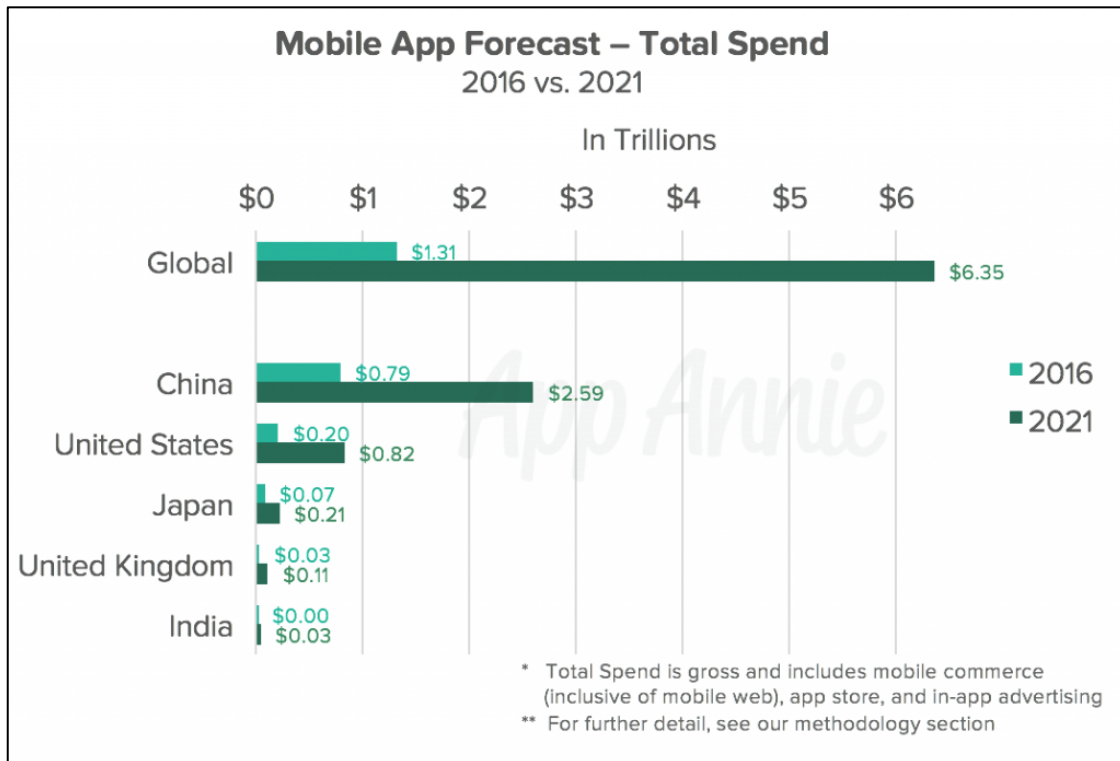


Figura 31 - Pronóstico de gasto mundial total en aplicaciones móviles [37]

A nivel mundial el uso de dispositivos móviles respecto al tráfico web ha superado al uso de equipos de escritorio desde 2017 [38], lo cual ha tenido un impacto directo en el desarrollo de software en rubros como el comercio minorista. Un caso de éxito que ejemplifica aquello es AliExpress, el marketplace online parte del grupo Alibaba, el cual considera el desarrollo de software móvil como un catalizador para su expansión internacional [39].

Para dar forma a la estrategia de la aplicación móvil de la compañía, se organizó una reunión para describir, visualizar y validar la visión de la aplicación Sodimac (ver Anexo 0). Como resultado, se obtuvo el Product Vision Board de la Figura 32, el cual resume la estrategia de la aplicación móvil de la compañía a través de una visión centrada en los usuarios y clientes, en donde la compañía parte del compromiso de entregar una experiencia memorable en el viaje de los clientes tanto en la asesoría que pueden obtener las tiendas, como en el contexto de compra de productos y servicios de forma online. El grupo objetivo está centrado en las personas que utilizan dispositivos móviles para navegar por internet y que además buscan productos para proyectos de transformación de su hogar. Sus necesidades son el acceso simplificado a la asistencia en la tienda, como información de productos y servicios y además una navegación clara y breve en cuanto a pasos y proceso para la compra de productos y servicios.

El producto que la compañía está apostando en desarrollar con esta estrategia es una aplicación móvil, con funciones de asistencia en tienda y venta de productos, y para medir el éxito del valor que puede generar el producto, es necesario medir su impacto respecto a objetivos de negocio, como su posicionamiento en las tiendas de aplicaciones, la asistencia en tienda vista desde la perspectiva de la satisfacción de los clientes, el aumento de los clientes recurrentes que compran productos y servicios y finalmente aumentar las transacciones realizadas en la aplicación móvil.

VISIÓN			
<p>Queremos que al ofrecer una experiencia memorable, guiando y acompañando a nuestros clientes para encontrar productos, proyectos, asesoría y servicios en su experiencia digital y en tiendas, lograremos ser líderes en el mejoramiento del hogar.</p> <p>Esta iniciativa será exitosa si logramos generar venta incremental, aumentar el compromiso con neutros clientes y mejorar su nivel de satisfacción.</p>			
GRUPO OBJETIVO	NECESIDADES	PRODUCTO	OBJETIVOS DE NEGOCIO
<p>Usuarios: personas que usan dispositivos móviles para navegar por internet.</p> <p>Cientes: segmento de clientes que buscan productos para proyectos de transformación de su hogar.</p>	<p>Acceso simple y rápido a asistencia en tienda, como información de productos y servicios.</p> <p>Navegación simple y compra expedita de productos y servicios.</p>	<p>Aplicación móvil con funciones de asistencia en tiendas y venta online de productos y servicios.</p>	<p>Posicionarse como un referente en las tiendas de aplicaciones móviles dentro de la categoría "Compras".</p> <p>Entregar asistencia a los clientes en las tiendas, sin necesidad de contar con un vendedor.</p> <p>Aumentar el número de clientes recurrentes que compran en la compañía.</p> <p>Aumentar el número de transacciones realizadas por dispositivos móviles.</p>
KPIS DE LOS OBJETIVOS DE NEGOCIO			
<ul style="list-style-type: none"> • Lograr posicionarse dentro del top-10 de aplicaciones móviles de compras en Chile en las tiendas de aplicaciones móviles. • Aumentar el indicador de satisfacción de "obtengo la información necesaria para hacer mi compra" en 10 puntos. • Lograr un incremento de un 10% en los usuarios recurrentes que realizan compras. • Incrementar las transacciones realizadas por dispositivos móviles en un 10%. 			

Figura 32 - PVB de la aplicación móvil

4.2 Proceso de desarrollo

Actualmente la compañía no posee un proceso formal de desarrollo de software y cada ciclo de desarrollo es exploratorio. Un aspecto negativo de este enfoque es que trae retrasos y extensos tiempos de pruebas manuales y puestas en producción (un mes en el mejor de los casos). Sin embargo, es simple y fácil de adoptar por los miembros del equipo.

4.2.1 Definición del nuevo proceso

En el Capítulo 3, se realizó un análisis sobre el proceso actual de desarrollo de la aplicación móvil, donde se detectaron: problemas, elementos a mantener e ítems a accionar para lograr una mejora en los distintos aspectos del proceso, tales como los tiempos de puesta en producción, la transparencia del proceso hacia los stakeholders y la flexibilidad del proceso frente al cambio en los requerimientos.

Considerando los aspectos positivos a mantener y los problemas, se definieron en la retrospectiva del proceso actual (ver Sección 3.5.1) ciertas acciones que se pueden intentar para lograr una mejora en el proceso. Estas acciones se listan en la Tabla 5.

Acción (Try)	Descripción
T1	Dar visibilidad de avance a stakeholders
T2	Utilizar un método formal de estimación de esfuerzo
T3	Fallar rápido y barato
T4	Producir entregables constantes y en breve tiempo
T5	Usar un enfoque de desarrollo adaptativo
T6	Priorizar en base a las necesidades de los usuarios
T7	Saber quiénes son nuestros clientes y usuarios
T8	Definir equipos y roles especializados
T9	Instancias de seguimiento de desarrollo en equipo
T10	Incluir a los stakeholders en el proceso de desarrollo

Tabla 5 - Ítems de acción (Try) del proceso de desarrollo

Todos los aspectos presentados anteriormente hablan de elementos comunes frente a un nuevo proceso: simplicidad, transparencia y adaptabilidad. Estos conceptos deben ser parte del nuevo

proceso de desarrollo. Para llevar a cabo esto, fue necesario revisar las metodologías de desarrollo de la industria que se enfocan en mejorar la productividad, simples de adoptar y que permitan validar funciones rápidamente con usuarios finales.

En una discusión en donde el equipo de la aplicación móvil se reunió para evaluar la definición del nuevo proceso, se expuso la existencia de dos corrientes: las metodologías tradicionales, usualmente orientadas a la planificación e inspiradas en disciplinas tradicionales de la ingeniería, y las metodologías ágiles, usadas ampliamente hoy por equipos de desarrollo de productos que requieren de constante adaptación. Las principales diferencias entre ambas metodologías son las siguientes:

- **Las metodologías ágiles son adaptativas en lugar de predictivas [13].**

Las metodologías asociadas a ingenierías tradicionales tienden a trazar planes detallados a largo plazo como parte del proceso de desarrollo de software. Esto podría funcionar bien en escenarios predecibles y bien conocidos. Se puede decir entonces que su naturaleza es ser resistentes al cambio. Las metodologías ágiles por otro lado aceptan el cambio, se adaptan a él.

- **Las metodologías ágiles están orientadas a las personas en lugar de los procesos [13].**

El objetivo de las metodologías asociadas a las ingenierías tradicionales es definir un proceso que funcione correctamente sea quien sea que las utilice. Las metodologías ágiles en cambio hablan de que ningún proceso compensa las habilidades del equipo de desarrollo, siendo entonces el proceso una herramienta de apoyo al equipo en el desarrollo de software.

Considerando estos dos aspectos claves frente al desarrollo de software y el contexto bajo el cual opera la compañía, es imperativo contar con un proceso de desarrollo de productos que opere bajo la misma dinámica: se adapte y evolucione frente a sus usuarios y mercado. Es por ello que el equipo de desarrollo decide utilizar metodologías ágiles. Además de esto, la decisión está alineada a los aspectos desprendidos de la retrospectiva, en donde para cada problema, aspecto mantener e intentar, las metodologías ágiles parecen tener una respuesta, brindando simplicidad en el proceso, roles claros y definidos, poca documentación, autogestión, adaptabilidad frente al cambio y transparencia como uno de sus pilares fundamentales.

Si se observa el desarrollo de productos altamente utilizados como Netflix, Amazon o Google, el factor común entre ellos es la aplicación de metodologías ágiles en el desarrollo de sus productos de software [40]. Algunas compañías como Spotify incluso han comentado en detalle su proceso de desarrollo y como este ha apoyado el desarrollo de su producto en términos de adaptación [41].

4.2.2 Una nueva metodología y enfoque

El termino *Agile* refiere a una filosofía de desarrollo de software y bajo ella, existen muchos enfoques específicos, tales como Extreme Programming, Scrum, Lean Development, etc. Cada uno de estos enfoques particulares tienen sus propias ideas, comunidades y líderes, pero aun así mantienen los mismos principios ágiles, e incluso a veces comparten ideas, prácticas y herramientas entre sí.

El equipo de la aplicación móvil utiliza Scrum el cual tendrá su propio Backlog. Además, existe un equipo adicional llamado Squad Integración tal como se muestra en la Figura 33, el cual está a cargo de apoyar al Squad App Sodimac con los desarrollos de back-end.

El rol UX Research está encargado de realizar la investigación centrada en los clientes y usuarios. El Scrum Master por otro lado, vela por la metodología en todos los equipos.

El rol de Business Analyst apoya la labor de levantar las necesidades de los usuarios para el producto. QA está a cargo de implementar las pruebas de calidad del producto. UX/UI está encargado de realizar los diseños gráficos y flujos de navegación de las aplicaciones que luego son tomados por los programadores, quienes desarrollan las funciones de front y back-end. Finalmente, el arquitecto está encargado del diseño del sistema de back-end que da soporte a la aplicación móvil.



Figura 33 - Organización de equipos Scrum en la compañía

El desarrollo de la aplicación móvil guiado por Scrum tiene un canal CI/CD, el cual se describe en la Figura 33, junto a los objetivos de las etapas que conforman los conceptos de Integración continua, distribución continua e implementación continua.

	Integración Continua			Distribución Continua	Implementación Continua
Objetivos	Crear una función de la aplicación	Pruebas funcionales y de aceptación de usuario	Integrar exitosamente el código al de otros programadores	Publicación de versión unificada en repositorio	Publicación de un incremento terminado para usuarios finales
Entrada	Product backlog	Función terminada	Función aprobada	Código fuente de front-end y back-end	Front-end: APK o IPA Back-end: scripts de instalación
Salida	Función terminada	Aprobación o rechazo de las versiones de las aplicaciones	Versión unificada	Versión unificada instaladas en repositorio	Versiones publicadas en producción
Ambientes	Desarrollo	Pruebas	Integración	Integración	Producción
Herramientas	Jira Gestor de requerimientos e incidencias NodeJS Framework back-end Ionic V3 Framework front-end Microsoft VS Code Editor de código	Selenium Pruebas automatizadas Test Flight Ambiente iOS UAT Play Store beta Ambiente Android UAT	GitLab Control de versiones Y almacenamiento de proyecto	GitLab Control de versiones Y almacenamiento de proyecto	App Store Play Store Jenkins Compilación automatizada (back-end) GitLab Compilación automatizada (front-end)

Figura 34 - Canal CI/CD del desarrollo de la aplicación móvil

4.3 Arquitectura móvil de la compañía

Para la definición de la nueva arquitectura, se establecieron los objetivos de negocio y motivadores, el alcance de la arquitectura, preocupaciones arquitectónicas, principios (arquitectónicos, tecnológicos y de negocio) y finalmente las restricciones que podrían aplicar a la nueva arquitectura.

Luego, la representación del diseño de la nueva arquitectura es presentado a través de vistas, las cuales facilitan la lectura a un rango amplio de stakeholders técnicos y de negocio. Cada vista por separada (pero interrelacionada), detalla un aspecto del sistema y colectivamente describen el sistema completo.

A continuación, en la Figura 35, se presenta un diagrama que muestra las relaciones entre las vistas creadas en las secciones finales del presente capítulo.

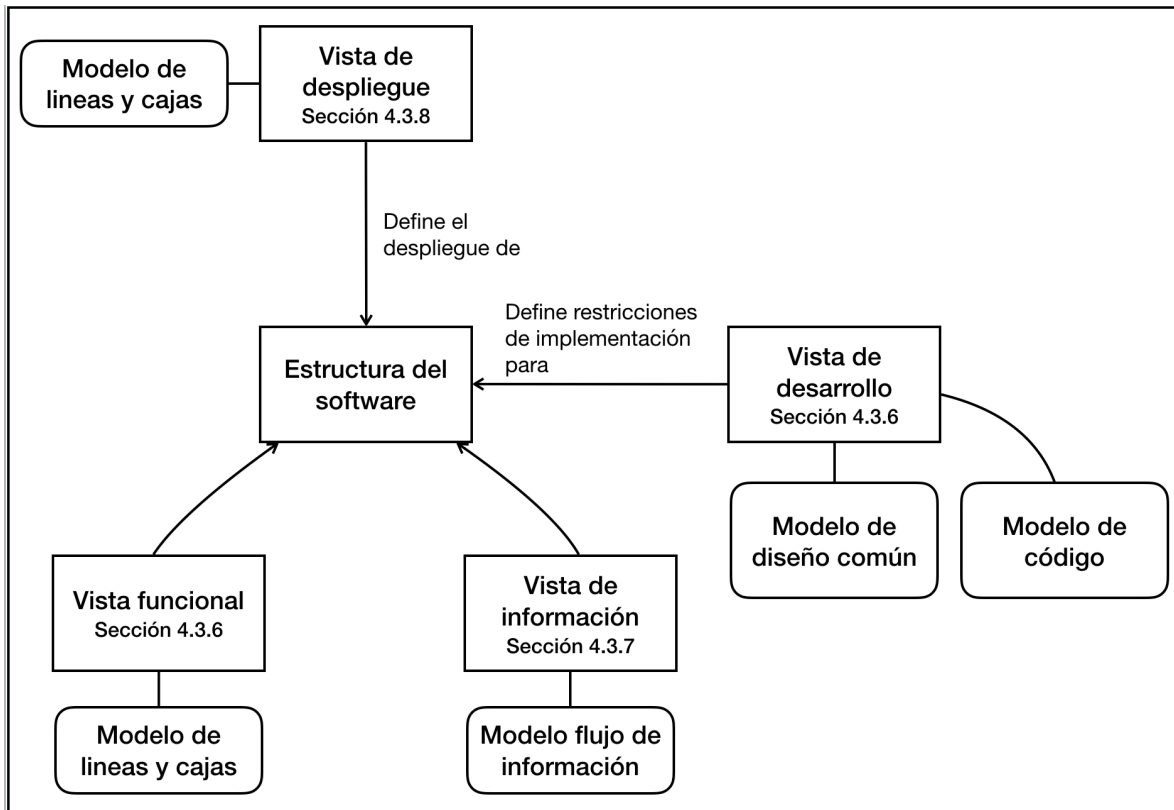


Figura 35 - Relaciones entre vistas

4.3.2 Alcance de la nueva arquitectura

La compañía tiene la necesidad de contar con una aplicación móvil que entregue valor en el contexto online (compra de productos y servicios) y en la tienda física. Para ello, es necesario definir un diseño arquitectónico que refleje la interacción de la aplicación móvil con los distintos sistemas de la compañía, considerando las áreas funcionales que proveerá el sistema, las interfaces externas al sistema que se comunican hacia y desde él, los sistemas que eventualmente son dados de baja con la implementación de la nueva arquitectura y cualquier dato que es migrado en el nuevo sistema.

Para una mayor claridad sobre el diseño arquitectónico, en la Figura 36 se presenta un diagrama de contexto de alto nivel respecto a los límites del sistema y sus entidades adyacentes. Este diagrama fue elaborado con los stakeholders.

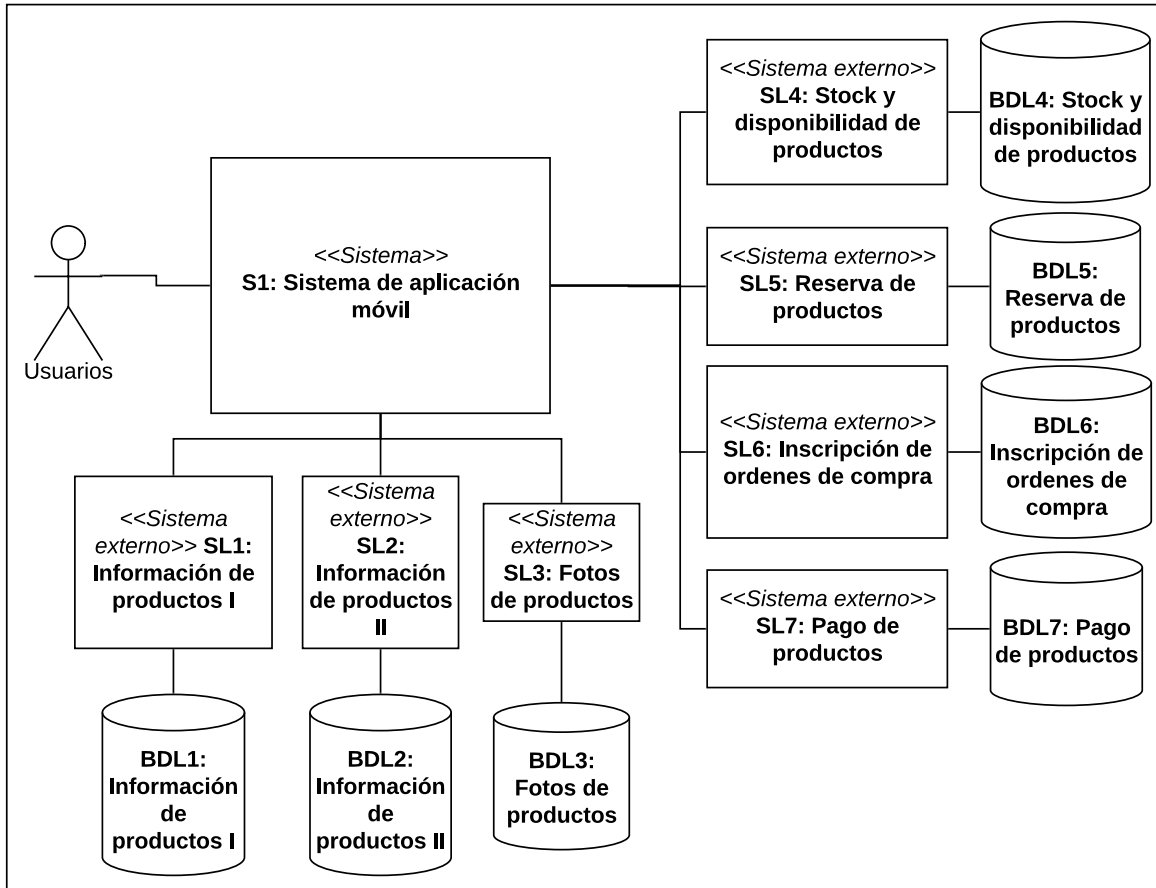


Figura 36 - Diagrama de contexto arquitectónico

Para este diagrama se detallan los siguientes sistemas:

S1. Sistema de aplicación móvil: contiene el ecosistema de la aplicación que se relaciona con los sistemas externos.

SL1. Información de productos I: administra los datos asociados a las tiendas físicas. Estos residen en la base de datos BDL1.

SL2. Información de productos II: datos de productos del catálogo online. Contiene en su base de datos BDL2 los datos de los productos y datos de clientes.

SL3. Fotos de productos: almacena las fotos de los productos en sus distintas dimensiones y formatos en la base de datos BDL3.

SL4. Stock y disponibilidad de productos: verifica la disponibilidad de entrega de productos respecto a: fecha, horario y ubicación. Estos datos se guardan en la base de datos BDL4.

SL5. Reserva de productos: en este sistema se ingresan las reservas de inventario para las transacciones, quedando registradas en la base de datos BDL5.

SL6. Inscripción de órdenes de compra: en este sistema se inscriben las órdenes de compra junto a los datos del cliente, entrega y pago. Todo queda guardado en la base de datos BDL6.

SL7. Pago de productos: el pago de productos y servicios es controlado por un sistema intermediario que enmascara la implementación de las pasarelas de pago.

Algunos sistemas serán dados de baja, como la base de datos BD1 que concentra los datos provenientes de cargas extraídas de los sistemas legados y los procesos de extracción de datos en hojas de cálculo P1, P2 y P3. Ver Figura 17.

Las bases de datos BDL1 y BDL2 de los sistemas legados de información de productos SL1 y SL2 además del sistema de fotos de productos SL3 y su base de datos BD3 se continuarán utilizando. Sin embargo, los sistemas de promociones SL4 ya no se consultará, dado que los datos están en SL2.

Sobre a la migración de datos para los sistemas que serán dados de baja o que ya no serán utilizados, se llevarán al nuevo sistema el registro de todos los productos y la información asociada a las tiendas físicas. Todo lo que no será migrado será revisado y modelado como parte del nuevo sistema.

4.3.1 Objetivos de negocio y motivadores

Los objetivos y motivadores del negocio fijan el contexto del proyecto y fundamentan su existencia. Relatan a su vez problemas específicos o potenciales oportunidades dentro de la organización. Un objetivo significa un fin particular y un motivador tiene que ver con una fuerza que hace actuar a la organización en un sentido u otro para proteger y hacer crecer su negocio. Para identificarlos, se realizó una sesión de trabajo (ver Anexo 0) el resultado se detalla en la Tabla 6:

Identificador	Descripción
O1	Contar con información coherente y siempre disponible.
O2	La aplicación móvil debe estar evaluada por sus usuarios, con una nota mínima de 4 estrellas en las tiendas de aplicaciones móviles.
M1	Unir la experiencia online con la experiencia offline (tiendas).

Tabla 6 - Objetivos y motivadores de negocio para el diseño de la nueva arquitectura

4.3.3 Preocupaciones arquitectónicas

Para dar forma a la nueva arquitectura, es necesario contar con la mayor cantidad posible de información proveniente de los stakeholders, ya que son ellos quienes tienen preocupaciones sobre la aplicación móvil. Muchas veces, estas tienen relación directa sobre la arquitectura, y como tal, pueden ser ambiguas e imposibles de medir [21]. Considerando esto, se deben explorar en profundidad, ayudando a los stakeholders a refinarlas y plantearlas como requerimientos, objetivos, intenciones o aspiraciones.

Junto a los stakeholders se realizó una votación donde aparecen las siguientes tres preocupaciones, las cuales fueron entendidas y consensuadas por todos los stakeholders entrevistados (ver Anexo 0):

P1. La aplicación debe ser rápida y fácil de usar.

P2. La información en la aplicación debe ser la misma que aparece en el sitio web o las tiendas.

P3. La aplicación móvil debe ser estable y continuar operando frente a la alta demanda.

Aunque estas tres principales preocupaciones son imprecisas y parecieran no poder medirse o cuantificarse, reflejan una fuerte orientación de negocio, por lo que no es posible ignorarlas. Es necesario traducirlas en requerimientos y objetivos que permitan definir la nueva arquitectura de la aplicación móvil. Para hacer esto, se clasifican las preocupaciones en: objetivos arquitectónicos, y requerimientos de arquitectura, tal como se muestra en la Tabla 24, Tabla 25 y Tabla 26 del Anexo 0.

4.3.4 Principios para demostrar trazabilidad

Para guiar la trazabilidad desde los objetivos y motivadores hasta las decisiones arquitectónicas que se implementarán en el sistema, se describirán a continuación los principios de negocio asociados a los objetivos y motivadores descritos previamente en este capítulo. Luego se detallarán los principios tecnológicos que nacen a consecuencia de los principios de negocio y finalmente las decisiones arquitectónicas como consecuencia de los principios tecnológicos.

Principios de negocio

PN1. Toda la información de productos y tiendas debe operar bajo el principio de omnicanalidad, el cual significa mantener una misma experiencia (de ventas, asistencia, operación, postventa, marca, lenguaje, etc.) entre canales de una compañía para sus clientes. Esto está dado fuertemente por el objetivo de negocio O1.

PN2. Los usuarios deben poder acceder a la aplicación y sitio web con su misma cuenta de usuario. Este principio viene dado por el objetivo de negocio O1.

PN3. La aplicación debe ser un producto llamativo y eficiente en términos de diseño, simple y fácil de usar por cualquier tipo de usuario (experto e inexperto). La razón de este principio recae en el objetivo de negocio O2.

PN4. La aplicación debe constantemente adaptarse, aprendiendo del comportamiento de sus usuarios y liberando rápidamente versiones con la retroalimentación obtenida. Esto viene dado por O2.

PN5. La aplicación móvil debe operar en las tiendas entregando información particular de este tipo de experiencia, como la ubicación de los productos o los horarios de apertura de las tiendas. Este objetivo de negocio viene dado por el motivador M1.

PN6. La aplicación siempre debe entregar información coherente a los usuarios, por lo que se debe considerar periodos de alta demanda en donde muchos usuarios ingresan a las aplicaciones y los sistemas pueden verse saturados. Resuelve O1.

PN7. En caso de haber algún problema o error grave en una versión, la aplicación debe ser capaz de volver a la versión anterior, almacenando y registrando los cambios realizados en cada publicación. Resuelve O2.

Principios tecnológicos

PT1. Cualquier aplicación móvil que necesite un dato, debe ser leído desde una única base de datos donde se ingresen y modifiquen estos datos. Resuelve el principio PN1.

PT2. Los datos deben ser replicados entre los sistemas en tiempo real, si hay un cambio en uno, debe reflejarse inmediatamente en todos los otros sistemas. Este principio soluciona lo referido en PN1.

PT3. Debe existir un único sistema concentrador de cuentas de usuario que maneje todos los datos relacionados a él, permitiendo la sincronización referente a cambios entre el sitio web y la aplicación móvil. Este principio soluciona PN2.

PT4. El diseño de la capa de front-end de las aplicaciones debe mantener las particularidades de Material Design para Android y Flat Design para iOS para facilitar la adopción de usuarios nativos de ambas plataformas. Este principio responde a PN3.

PT5. Implementar herramientas de analítica que permitan comprender el comportamiento de los usuarios en las aplicaciones para luego implementar mejoras en flujos de navegación y validación de funciones. Resuelve PN4.

PT6. Las aplicaciones deben entregar datos provenientes de los sistemas legados que hacen referencia a las tiendas. Resuelve PN5.

PT7. El back-end de las aplicaciones debe estar basado en la virtualización de servidores para permitir la escalabilidad dinámica de recursos frente a la alta demanda. Resuelve PN6.

PT8. Se debe aplicar el control de versiones para el código fuente de las aplicaciones y servicios desplegados en la nube para asegurar el control de los cambios. Resuelve PN7.

PT8. Se debe integrar un canal CI/CD para la integración y despliegue continuo de código y versiones de la aplicación móvil. Resuelve PN4.

4.3.5 Restricciones

Por restricciones sobre el diseño de la nueva arquitectura se pueden definir estándares, lineamientos estratégicos y políticas que pueden limitar las opciones arquitectónicas disponibles para su diseño. A continuación, en la Tabla 7 se listan y clasifican las restricciones.

Restricción	Tipo	Descripción
RS1	Estándares	Las bases de datos SQL solo pueden ser implementadas bajo licencias de Oracle o Microsoft.
RS2	Estándares	Los desarrollos en la nube deben estar basados en Microsoft Azure.
RS3	Estándares	JIRA y Confluence como herramientas de gestión de proyectos.
RS4	Estándares	No consultar en línea ciertos servicios de sistemas legados que existen sobre instancias físicas y que no poseen alta disponibilidad.
RS5	Lineamientos estratégicos	Usar control de versiones para el código fuente.
RS6	Lineamientos estratégicos	El uso de pruebas unitarias con un mínimo de un 85% de cobertura.
RS7	Lineamientos estratégicos	Utilizar Ionic V3 como Framework de aplicaciones móviles.
RS8	Lineamientos estratégicos	Las aplicaciones deben ser probadas en iOS Testflight y Android Beta.
RS9	Lineamientos estratégicos	Programar en ambiente de desarrollo, probar en ambiente de prueba y publicar en producción las versiones de las aplicaciones.
RS10	Políticas	Los usuarios deben ser encriptados para la seguridad de su información.

Tabla 7 - Restricciones arquitectónicas.

4.3.6 Vista funcional

A continuación, en la Figura 37 se expone un diagrama de estructura funcional de tipo líneas y cajas, que muestra los elementos funcionales representados como rectángulos de puntas cuadradas; los sistemas de back-end externos como rectángulos de puntas redondeada; los conectores que permiten comunicar elementos están representados como flechas que indican la dirección del flujo de la información; las interfaces funcionales como Middleware, quien se encarga de gestionar los accesos a los sistemas legados y el proceso ETL que permite cargar datos desde sistemas legados hacia un servidor de almacenamiento están representadas por un icono de nube; las líneas

horizontales gruesas representan las redes de datos; la entidad externa está representada como un teléfono móvil de cliente y finalmente el alcance de cada sistema (interno: Azure, externo: zona desmilitarizada o DMZ) está representado por los elementos que están dentro del rectángulo punteado.

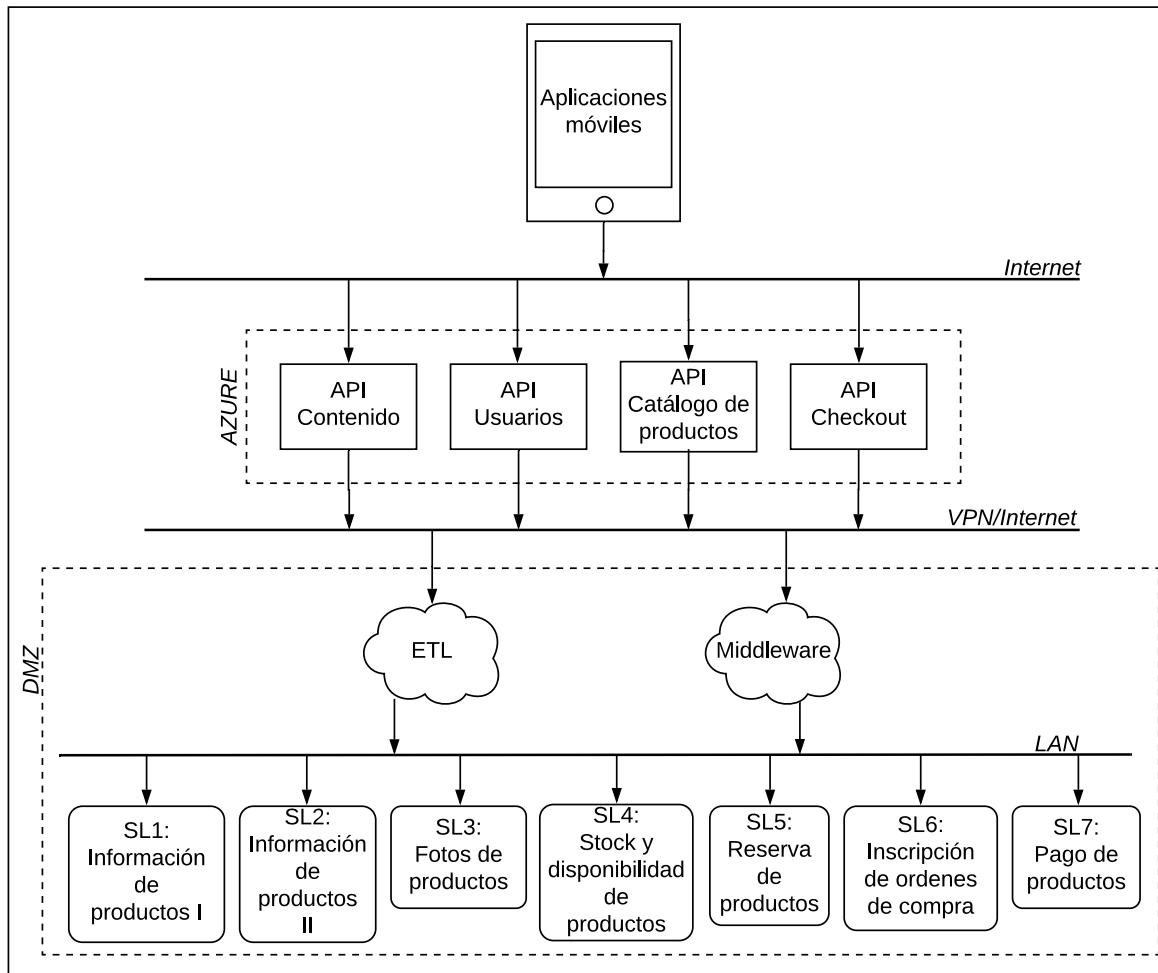


Figura 37 - Vista funcional del sistema de la aplicación móvil

La Tabla 8 describe las responsabilidades de cada elemento funcional del sistema:

Elemento	Responsabilidades
API Contenido	<ul style="list-style-type: none"> • Gestión de la carga de contenido dinámico en la aplicación.
API Usuarios	<ul style="list-style-type: none"> • Gestión de las cuentas de usuarios de la aplicación móvil. • Establecer roles para los usuarios. • Entregar seguridad de acceso a los datos de los usuarios.
API Catálogo de productos	<ul style="list-style-type: none"> • Gestión de la información asociada a productos. • Gestión de la información relacionada a las tiendas. • Regionalización de los datos (por país).
API Checkout	<ul style="list-style-type: none"> • Gestión de la compra de productos a través de la reserva de stock. • Gestión de la entrega de productos. • Inscripción de la orden de compra. • Proveer el control a la pasarela de pago de la compra. • Proveer información de seguimiento de compra.

Tabla 8 - Responsabilidad de elementos, vista funcional

En el Anexo 0 se puede encontrar un ejemplo de la estructura de la API Catálogo de productos.

Decisiones Arquitectónicas

Sobre esta vista se exponen las siguientes decisiones basadas en los principios tecnológicos:

DA1. Una API Catálogo de productos expone a la aplicación móvil los datos leídos desde los servicios, microservicios y bases de datos de los sistemas legados de productos y tiendas. Resuelve PT1.

DA2. La API Usuarios conecta a la aplicación móvil con el sistema legado de usuarios en donde están todos sus datos personales. Resuelve PT3

DA3. Se establece una conexión a través de cargas de datos vía proceso ETL desde el sistema legado que poseen información de productos y tiendas (SL1). Resuelve PT6.

Como resumen, se puede ver en la Tabla 9, la trazabilidad de las decisiones arquitectónicas hasta los objetivos y motivadores.

Decisión arquitectónica	Principio TI	Principio Negocio	Objetivo/Motivador
DA1	PT1	PN1	O1
DA2	PT3	PN2	O1
DA3	PT6	PN5	M1

Tabla 9 - Trazabilidad de decisiones arquitectónicas para la vista funcional

4.3.7 Vista de información

La información sobre productos, la logística y los clientes son elementos fundamentales dentro de la estrategia de la compañía, la cual debe ser soportada por procesos y tecnología que permitan su almacenamiento, mantención y disponibilidad. Ahora bien, cuando la estrategia de negocio de la compañía se basa en la omnicanalidad, es importante considerar que para diseñar la arquitectura del sistema, se debe mantener este principio representado a través de entidades que aseguran la disponibilidad y coherencia en los datos, para informar correctamente a los clientes sobre los distintos aspectos del servicio que presta la aplicación móvil, tal como lo experimentarían en otros canales de la compañía.

A continuación, en la Figura 38 está representada la vista de información, para ello se utilizó un modelo basado en el flujo de la información que va desde los sistemas legados, hasta las estructuras de datos o entidades dentro del nuevo sistema de la aplicación móvil. En ella, además, los rectángulos de puntas cuadradas representan las entidades internas, propias del sistema de la aplicación móvil; las flechas numeradas representan flujo de datos; los cilindros representan las bases de datos internas y los rectángulos de puntas redondeadas representan a las entidades externas al sistema de la aplicación móvil.

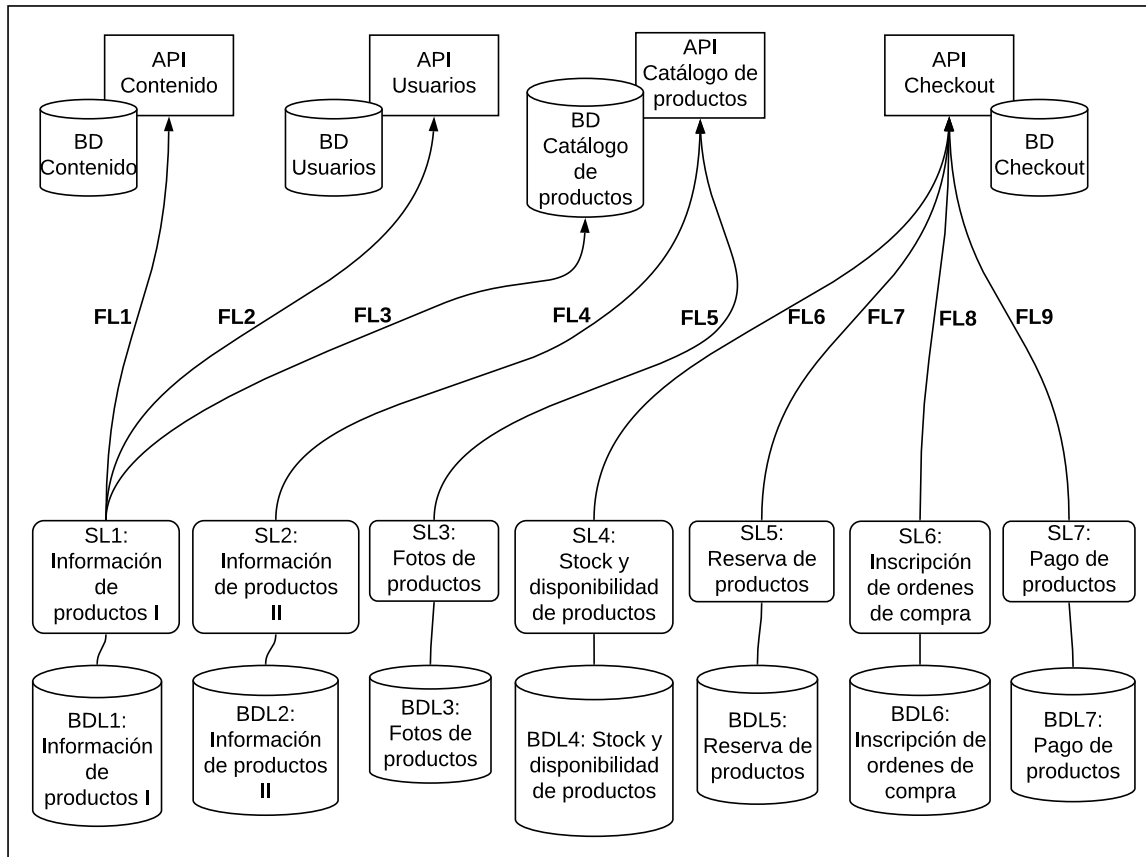


Figura 38 - Vista de información del sistema de la aplicación móvil

A continuación, en la Tabla 10 se detalla el flujo de los datos en la vista de información, considerando el tipo de conexión entre entidades y la temporalidad de los datos que fluyen entre ellas.

Flujo de datos	Descripción	Tipo	Temporalidad
FL1	<p>Origen: SL1, Información de productos I.</p> <p>Destino: API Contenidos.</p> <p>Microservicios que exponen datos a través del middleware.</p>	REST	On-call (tiempo real), necesario para exponer datos sensibles como precios.

<p>FL2</p>	<p>Origen: SL1, Información de productos I.</p> <p>Destino: API usuarios.</p> <p>Microservicios de login, registro, seguimiento de compras y libreta de direcciones de usuario.</p>	<p>REST</p>	<p>On-call, necesario para validar a los usuarios al momento de identificarse en la aplicación, además retorna los datos personales de compra.</p>
<p>FL3</p>	<p>Origen: SL1, Información de productos I.</p> <p>Destino: BD Catálogo de productos.</p> <p>ETL que carga una réplica de los datos de productos en la base de datos de productos.</p>	<p>ETL</p>	<p>La edad de la información es de 24 hrs. Dado que el proceso ETL carga un lote de información todos los días a las 08.00 hrs. en la base de datos.</p>
<p>FL4</p>	<p>Origen: SL2, Información de productos II.</p> <p>Destino: API Catálogo de productos.</p> <p>microservicios como: buscador, categorías, filtros, precios, variaciones, promociones, nombre del producto, marca, modelo, stock, etc.</p>	<p>REST</p>	<p>On-call, necesariamente consultados en línea para que la aplicación móvil esté sincronizada respecto a los cambios realizados sobre los datos sensibles: precios, stock y promociones.</p>
<p>FL5</p>	<p>Origen: SL3, Fotos de productos.</p> <p>Destino: API catálogo de productos.</p> <p>Servicio que retorna las fotos de un producto.</p>	<p>REST</p>	<p>On-call, dado que las fotos cambian regularmente y es necesario mantenerlas actualizadas.</p>
<p>FL6</p>	<p>Origen: SL4, Sstock y disponibilidad de productos.</p> <p>Destino: API Checkout.</p> <p>Microservicio que retorna la</p>	<p>REST</p>	<p>On-Call, debe ser en línea para la venta de productos, por ello estos datos deben ser coherentes con los demás</p>

	disponibilidad de despacho y retiro de productos.		sistemas de venta.
FL7	<p>Origen: API Checkout</p> <p>Destino: SL5, Reserva de productos</p> <p>Microservicio que ingresa una reserva de productos dada una intención de compra.</p>	REST	On-Call, para ingresar inmediatamente la reserva y no perder el stock o la disponibilidad de los productos previamente validada.
FL8	<p>Origen: API Checkout.</p> <p>Destino: SL6: Inscripción de órdenes de compra.</p> <p>Microservicio que recibe inscripciones de órdenes de compra provenientes de todos los canales de venta.</p>	REST	On-Call, necesario para ingresar en tiempo real una orden de compra dada una reserva previamente generada.
FL9	<p>Origen: API Checkout.</p> <p>Destino: SL7: Pago de productos.</p> <p>Dada una orden generada, este microservicio permite pagarla en las pasarelas de pago</p>	REST	On-Call, constituye una transacción que en caso de éxito o fallo retorna una alerta a la API Checkout.

Tabla 10 - Detalle del flujo de datos en la vista de información

Decisiones Arquitectónicas

DA4. Existe un proceso que escucha los cambios de datos sensibles, los cuales deben mantenerse actualizados en tiempo real dada su importancia para los usuarios, tales como, precios, promociones y disponibilidad de productos en tiempo real para evitar diferencias con otros sistemas. Resuelve PT2.

DA5. La API usuarios contiene microservicios que permiten sincronizar los datos y actualizaciones en línea de identificación de usuarios, registro, recuperación de clave, libreta de direcciones, carro de compra, favoritos y compras. Resuelve PT3.

La Tabla 11 presenta un resumen de la trazabilidad de decisiones arquitectónicas para la vista de información.

Decisión arquitectónica	Principio TI	Principio Negocio	Objetivo/Motivador
DA4	PT2	PN1	O1
DA5	PT3	PN2	O1

Tabla 11 - Trazabilidad de decisiones arquitectónicas para la vista de información

4.3.8 Vista de despliegue

Una vez que el sistema ha sido desarrollado, probado y se encuentra listo para ser puesto en producción, se debe desplegar sobre un entorno físico en donde se encuentran los ambientes de hardware que darán soporte a la operación [20]. En la vista de despliegue es donde se detalla esto, además de las dependencias que tiene el sistema en su entorno de ejecución.

En la Figura 39 se muestra un modelo de líneas y cajas en donde el rectángulo de puntas cuadradas representa a los nodos internos de procesamiento (propios del sistema de la aplicación móvil); los rectángulos de puntas redondeadas representan los nodos externos de procesamiento; el teléfono móvil representa al cliente; los rombos a hardware de seguridad como firewalls; las líneas delgadas son las conexiones entre nodos de hardware; las líneas gruesas representan las redes de datos públicas (Internet) y privadas (LAN); los asteriscos representan la existencia de determinados nodos de esa clase (por ejemplo, en el caso del Core Server, por cada sistema de back-end descrito previamente en la Figura 37 existe uno o varios servidores) y finalmente las áreas puntuadas representan los entornos internos (Azure) y externos (DMZ).

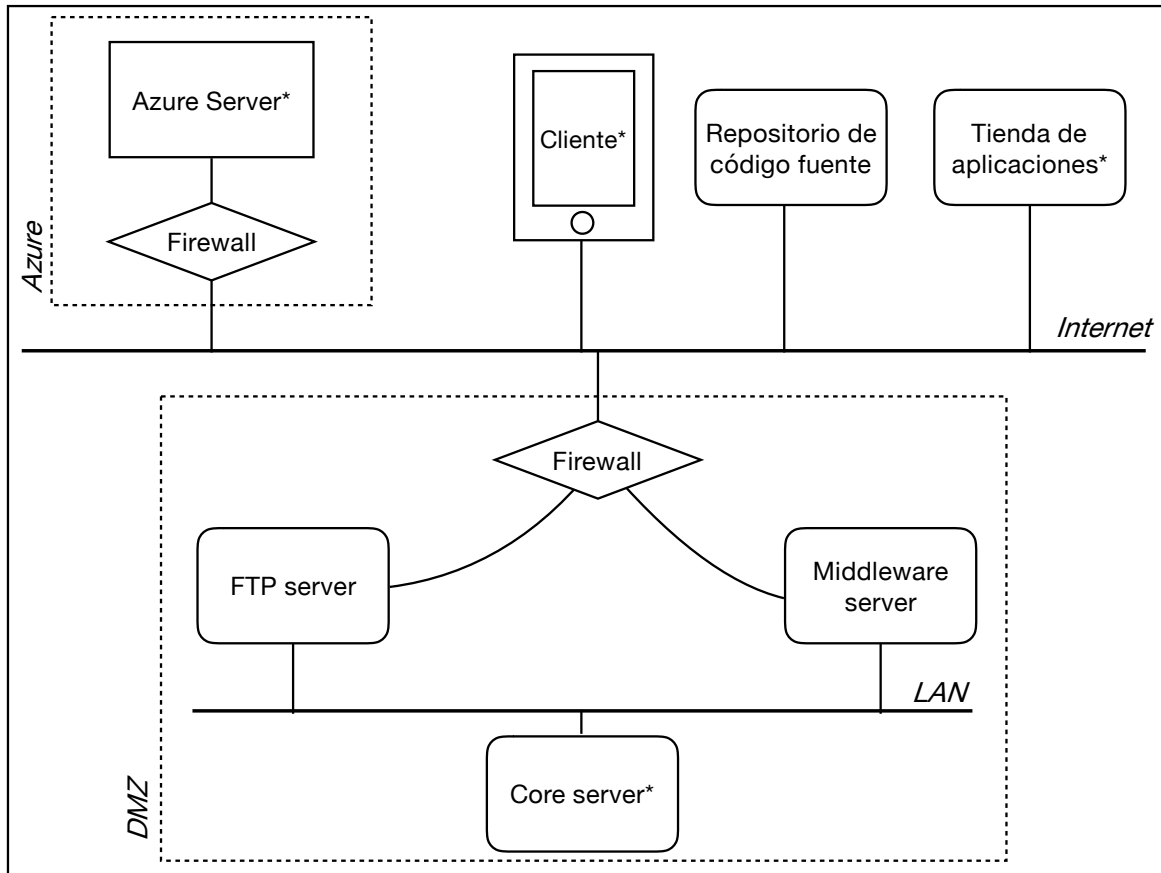


Figura 39 - Vista de despliegue del sistema de la aplicación móvil

La Tabla 12 detalla cada uno de los nodos presentes en la vista de despliegue:

Nodo	Descripción
Azure	<p>Cluster en la nube donde residen varias instancias de servidor de Microsoft Azure para distintos propósitos, como:</p> <ul style="list-style-type: none"> • ASE: App Service, en donde están desplegadas las APIs. • Blob Storage: almacenamiento de archivos. • SQL Server: motor de base de datos relacional. • Cosmos DB: motor de base de datos NoSQL. <p>La disponibilidad de instancias es dinámica.</p>
Cliente	<p>Teléfonos móviles Android o iOS, compatibles con la aplicación móvil desde las versiones 6 y 10 de sus sistemas operativos respectivamente.</p>

Repositorio de código fuente	GitLab, en donde se respalda el código fuente de front-end y back-end, para luego ser desplegado a los entornos de desarrollo, pruebas, integración y producción.
Tiendas de aplicaciones	Repositorios de aplicaciones para entornos de pruebas y producción para sistemas operativos Android e iOS.
Servidor FTP	Servidor IBM Sterling en donde los servidores de back-end ponen a disposición los datos para ser tomados por los procesos de ETL que los llevarán hacia las bases de datos en Azure.
Middleware	Servidor que permite la comunicación entre nodos internos (sistema de la aplicación móvil) y externos (back-end legado de la compañía).
Core Server	Bajo esta simplificación existen varias instancias de servidor por cada sistema de back-end legado descrito en la vista funcional (SL1 a SL7 junto a sus respectivas bases de datos, ver Figura 37).

Tabla 12 - Nodos de la vista de despliegue

Decisiones Arquitectónicas

DA6. A pesar de que el back-end de la aplicación móvil está en la nube, esta se conecta a través de la DMZ a los sistemas legados de la compañía para obtener los datos omnicanales (productos, servicios, tiendas, compras, clientes, usuarios, etc.). Resuelve PT1.

DA7. Se implementa Microsoft Azure como solución de servicios en la nube, ahí se definen APIs y microservicios que permiten automáticamente escalar los recursos de instancias y servicios frente a la alta demanda de consultas proveniente desde la aplicación móvil. Resuelve PT7.

DA8. Se implementa un repositorio en GitLab donde se envía y almacena cada cambio hecho en el código de las capas de front-end y back-end mediante un proceso automatizado de despliegue. Resuelve PT8.

La Tabla 13 resume la trazabilidad de las decisiones arquitectónicas para la vista de despliegue.

Decisión arquitectónica	Principio TI	Principio Negocio	Objetivo/Motivador
DA6	PT1	PN1	O1
DA7	PT7	PN6	O1
DA8	PT8	PN7	O2

Tabla 13 - Trazabilidad de decisiones arquitectónicas para la vista de despliegue

4.3.9 Vista de desarrollo

Para soportar el diseño y desarrollo de sistemas complejos es necesario realizar una exhaustiva planificación no solo del software en sí, sino también del ambiente de desarrollo en donde se diseñará, construirá y probará el sistema. Aspectos como el manejo de dependencias, gestión de la configuración de las entregas, estructura del código y el uso de estándares son necesarios para asegurar la integridad del sistema, ahorrar esfuerzo de desarrollo, facilitar su comprensión y operación [20]. La idea de esta vista no es detallar exhaustivamente la agenda de los programadores, sino que mantener organizada la codificación general del sistema, definiendo aspectos claves en la estructura del desarrollo.

Modelo de diseño común

Para maximizar los elementos en común entre implementaciones es bueno definir un conjunto de restricciones al diseño de los elementos del sistema. Los beneficios de estas restricciones son en primer lugar reducir el riesgo y la duplicidad de esfuerzo frente a la resolución de problemas identificados, y en segundo lugar, el tener elementos comunes incrementa la coherencia técnica de todo el sistema y lo hace mucho más fácil de entender, operar y mantener.

MD1. Estandarización del diseño:

- La capa de front-end de la aplicación móvil está compuesta por el Framework Ionic V3 basado en Angular y todos sus plugins deben ser integrados desde los repositorios oficiales de Ionic Native (wrapper para Cordova/PhoneGap que permite usar funciones nativas). Esto permite mantener un mismo entorno para todos los programadores, los que deben estar coordinados en las versiones a utilizar de los plugins y el framework.

- Las APIs web desarrolladas en la capa de back-end estarán basadas en el estilo arquitectónico REST.

MD2. Componentes de software estándar:

- Todas las funciones que requieran de autenticación, tanto para software de front-end o back-end como usuarios, deben interactuar con la API usuarios que posee un componente llamado SSO (*Single Sign On* por sus siglas en inglés) que entrega seguridad vía JWT (*JSON Web Token* por sus siglas en inglés), un estándar que permite entregar seguridad entre sistemas a través de llaves de acceso, autorizando la identidad de los usuarios y sus privilegios [42].
- La aplicación móvil debe basarse en el proyecto en blanco por defecto que entrega Ionic V3 para el desarrollo de la capa de front-end.
- Todas las aplicaciones deben utilizar Material Design o Flat Design como inspiración de diseño para los componentes gráficos de la interfaz de front-end. Dado que estas guías de estilo nativas de Android e iOS permiten que los usuarios accedan a una aplicación móvil que tenga elementos ya conocidos por ellos en el uso de cada sistema operativo [43].

Modelo de código

MC1. Estructura

Para el desarrollo de front-end, la estructura del proyecto es la que viene por defecto al instalar un proyecto en blanco de Ionic V3. A partir de ello se comienza el desarrollo e integración de plugins y componentes. Los pasos para la instalación son los siguientes:

1. Descargar e instalar la versión estable de Node.js, desde el sitio web oficial: <https://nodejs.org/en/>
2. Abrir cualquier herramienta de línea de comandos que se tenga configurada y escribir el comando: `npm install -g ionic`
3. Ejecutar en la línea de comandos: `ionic start [nombre-app] [tipo]`, donde el nombre puede ser cualquiera definido por el usuario y el tipo puede ser cualquiera de los descritos en la Figura 40. **Error! No se encuentra el origen de la referencia.**, los cuales corresponden a plantillas gráficas disponibles para comenzar a desarrollar una aplicación móvil.

- Una vez que la aplicación fue creada, se debe ejecutar los siguientes comandos los cuales abrirán un navegador web en donde se desplegara la aplicación móvil:

```
cd [nombre-app]
```

```
ionic serve
```

- Una vez que se está listo para desplegar la aplicación en un dispositivo real (Android o iOS) se deben configurar ambos entornos siguiendo la documentación oficial de Ionic:

<https://ionicframework.com/docs/intro/deploying/>

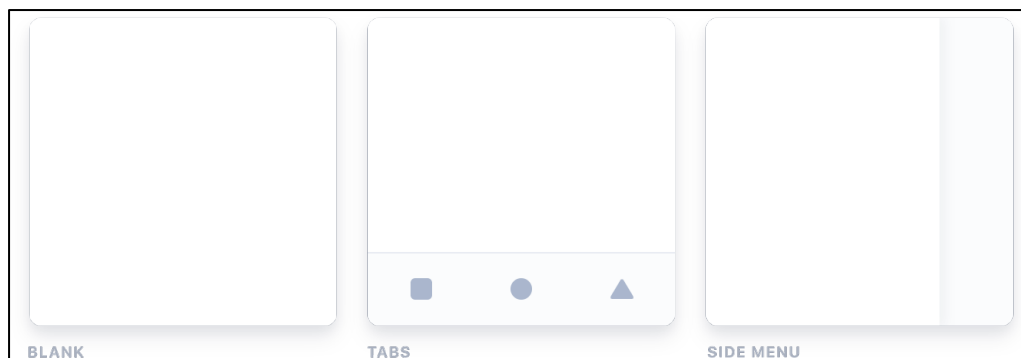


Figura 40 - Tipos de proyectos disponibles en Ionic V3 [44]

Todo proyecto en blanco de Ionic V3, inicialmente contendrá la estructura detallada en la Figura 41, en donde `src/directory` tiene items como el archivo `index.html`, archivos de configuración para test, una carpeta de recursos para imágenes y un directorio principal `app/` para el código de la aplicación.

```
> src/  
  > app/  
  > assets/  
  > environments/  
  > theme/  
  global.scss  
  index.html  
  karma.conf.js  
  main.ts  
  polyfills.ts  
  test.ts  
  tsconfig.app.json  
  tsconfig.spec.json
```

Figura 41 - Estructura de un proyecto por defecto de Ionic V3

En la Figura 42, el directorio `src/pp/` contiene la raíz de los componentes y módulos, adicionalmente contiene los directorios de funciones de la app como las páginas, componentes y servicios.

```
> src/  
  > app/  
      app-routing.module.ts  
      app.component.html  
      app.component.spec.ts  
      app.component.ts  
      app.module.ts
```

Figura 42 - Estructura del directorio `app/` de un proyecto por defecto de Ionic V3

MC2. Enfoque de construcción

- Para el desarrollo de los builds de front-end (Android e iOS), los programadores utilizarán Microsoft Visual Studio Code como editor de código fuente, GitLab (GIT) para el control de versiones, Node.js como entorno de ejecución de JavaScript, Ionic V3 como SDK de desarrollo basado en AngularJS y Apache Cordova, Android Studio para la compilación de versiones Android y finalmente Xcode para la compilación de versiones iOS.
- Para el desarrollo de back-end los programadores utilizarán Microsoft Visual Studio Code como IDE, GIT para el control de versiones, SQL Server y Cosmos DB como motores de bases de datos SQL y NoSQL respectivamente.

MC3. Proceso de puesta en producción (front-end)

- cuando los programadores terminan una funcionalidad, se ejecutan las pruebas automatizadas utilizando el software Selenium sobre los archivos de instalación de Android e iOS. En caso de no haber errores, estas podrán pasar a las UAT. En caso contrario, retornan a desarrollo.
- Los programadores de la aplicación integran unos con otros sus progresos mediante la herramienta de integración continua de GitLab

- Una versión unificada de la aplicación es desplegada al repositorio de código GitLab mediante la función de distribución continua de esta plataforma.
- Una vez que la versión unificada es correctamente respaldada en GitLab, se procede a implementar de forma automatizada una versión candidata a producción en las tiendas de aplicaciones de Google y Apple. Esto se realiza automáticamente a través de GitLab.

Cada tienda realiza una revisión de los archivos de instalación para el caso de Apple esta revisión es manual y minuciosa (puede tomar de uno a cinco días hábiles en recibir una respuesta). En caso de no haber observaciones, la aplicación es publicada, en caso contrario, se detallan comentarios sobre los distintos tipos de infracciones a las políticas de desarrollo de Apple y se deben corregir para luego subir otra versión para ser evaluada nuevamente. La Figura 43 muestra el proceso de publicación.

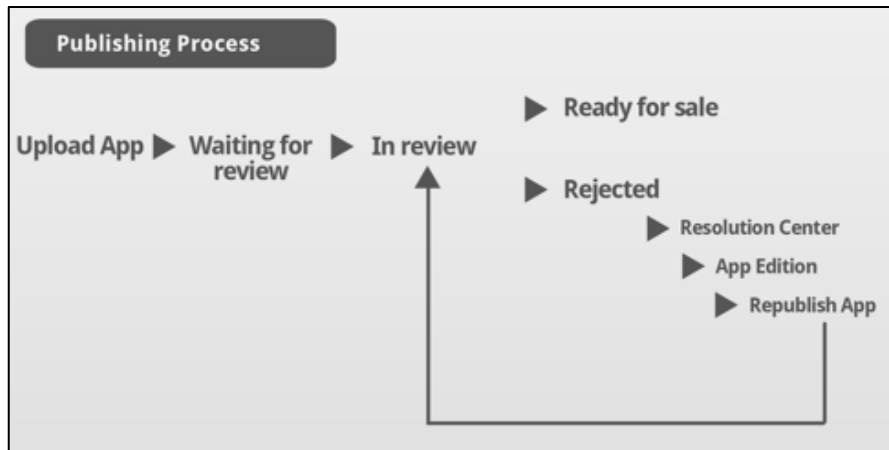


Figura 43 - Proceso de publicación en App Store (iOS) [45]

En Android el proceso de revisión es automático y la publicación puede tomar un par de horas.

MC4. Proceso de puesta en producción (back-end)

- Los programadores integran sus cambios locales a la rama de pruebas back-end.
- Se ejecutan las pruebas automatizadas unitarias y de integración a través de Jenkins.
- Si las pruebas resultan bien, el código es integrado automáticamente al de los otros programadores.
- Luego, el código unificado es distribuido al repositorio común de back-end en GitLab.
- Finalmente, el código es implementado en la rama de producción de back-end.

MC5. Gestión de la configuración

- Los programadores de front-end y de back-end desarrollan las funcionalidades de las aplicaciones en su entorno de desarrollo local y luego publicarán estos cambios en un repositorio de desarrollo en GitLab separado por proyecto, desde donde se podrá publicar luego en ambiente de pruebas o producción en Azure.
- Toda confirmación de cambios será versionada, dejando un comentario sobre la razón de la edición y el nombre del desarrollador responsable en caso de necesitar revisar un cambio o reparar alguna incidencia.

Decisiones arquitectónicas

DA9. La interacción y diseño de pantallas de la capa de front-end de las aplicaciones debe mantener las particularidades de Material Design para Android y Flat Design para iOS. Resuelve PT4.

DA10. Se incorporan herramientas para el seguimiento del comportamiento del usuario, rendimiento de la aplicación y recopilación de datos de negocio (comerciales). Resuelve PT5.

DA11.

La Tabla 14 resume de la trazabilidad de decisiones arquitectónicas para la vista de desarrollo.

Decisión arquitectónica	Principio TI	Principio Negocio	Objetivo/Motivador
DA9	PT4	PN3	O2
DA10	PT5	PN4	O2
DA11	PT8	PN4	O2

Tabla 14 - Trazabilidad de decisiones arquitectónicas para la vista de desarrollo

4.4 Resumen

Sin duda, hoy el comercio electrónico a nivel mundial ha adoptado tecnología y recursos que le han permitido crecer a escalas sin precedentes. Solo basta mirar los resultados de Alibaba durante el último evento llamado el día del soltero, realizado cada 11 de noviembre. Este año lograron recaudar cerca de 38.4 billones de dólares en 24 horas, lo cual significó un crecimiento del 24% respecto al evento anterior de 2018 [46], por otra parte, la participación de ingresos respecto a la aplicación móvil en el evento hoy es cerca de un 90% [47] Para enfrentar un escenario tan competitivo Sodimac implementa un proceso de desarrollo ágil, que permite publicar rápidamente nuevas versiones de la aplicación móvil para obtener retroalimentación de sus usuarios y mejorar o cambiar de ser necesario. Por otro lado, se construye una plataforma tecnológica basada en la nube, junto a una arquitectura distribuida y flexible, que permite ser fácilmente modificada sin una dependencia de otros componentes y que además permite escalar recursos rápidamente frente a la alta demanda de usuarios.

5. Validación

En el presente capítulo se detallará el desarrollo de una nueva versión de la aplicación móvil de Sodimac como MVP para validar el nuevo proceso de desarrollo y arquitectura. Se revisarán los indicadores cualitativos por parte del equipo de la aplicación móvil, quienes evaluarán el nuevo proceso y arquitectura. Finalmente se evaluará cuantitativamente la aplicación del nuevo proceso y arquitectura basándose en indicadores de rendimiento del equipo y sus entregables, también serán considerados algunos indicadores claves de uso de la aplicación móvil por parte de los usuarios.

5.1 Piloto

Para validar la solución propuesta a los problemas evidenciados frente al desarrollo de la aplicación móvil, se realizó un MVP durante los meses de enero a mayo de 2017, lo que implicó desarrollar una nueva versión de la aplicación Sodimac bajo un nuevo proceso de desarrollo y utilizando una nueva arquitectura.

En primer lugar, se identificaron las necesidades de los usuarios a través de un estudio realizado por el equipo de experiencia de usuario de la compañía, el cual se realizó previo al MVP, durante diciembre de 2016 para luego en enero de 2017 tener claridad de quienes serían los usuarios de la aplicación móvil. En dicho estudio se encuestó a clientes y se les hizo seguimiento en su relación con Sodimac. Como resultado se detectaron tres tipos de persona que utilizarían la aplicación móvil.

5.1.1 Proto-personas

Los tipos de personas que se relacionan con la compañía a través de la aplicación móvil enfocada en la mejora del hogar son tres y están descritas en la Figura 44 a través de un modelo de proto personas [48].



Figura 44 - Resumen de Proto personas

5.1.2 Viaje del cliente

Una vez que se han identificado los tipos de persona, es necesario conocer su comportamiento y relacionamiento frente a la compañía. Para ello se elaboró el viaje del cliente, el cual describe las cinco etapas por las que pasa cualquiera de estos tres tipos de personas y sus principales preguntas en cada una de ellas. Muchas de estas preguntas y necesidades ya eran conocidas por el equipo e incluso algunas estaban abordadas de algún modo en la versión actual de la aplicación móvil, como, por ejemplo, conocer el precio de los productos. En la Figura 45, están las cinco etapas del viaje del cliente las cuales son el resultado del estudio realizado por el equipo de experiencia de usuario: inspiración, planificación del proyecto, aprendizaje y toma de decisión, compra, post venta y administración.

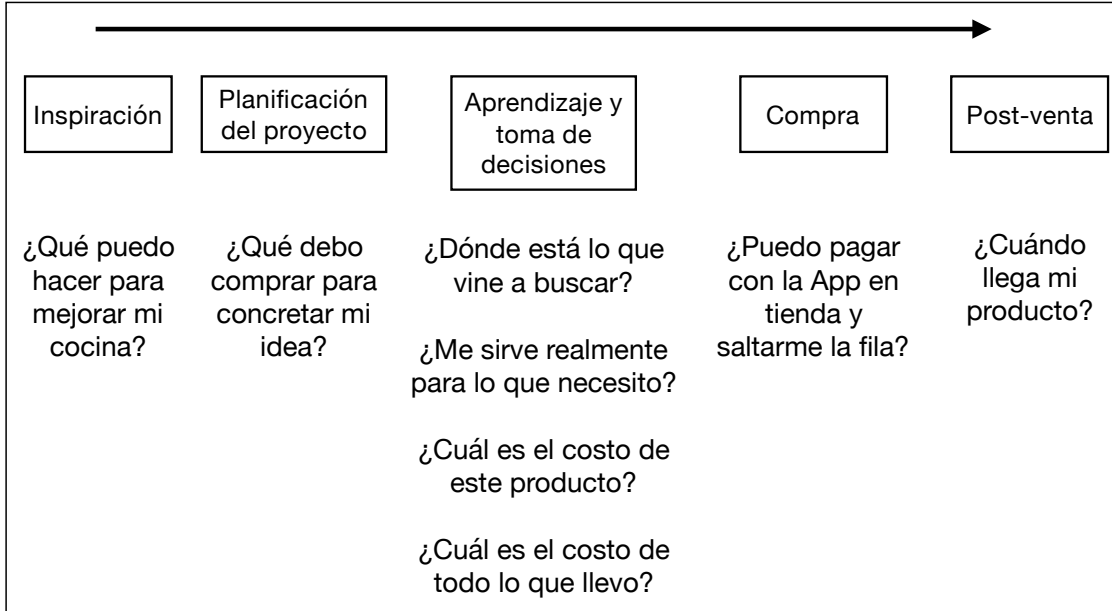


Figura 45 - Viaje del cliente

5.1.3 Hoja de ruta

Durante una sesión de trabajo, detallada en la Tabla 15, en donde se reunió el equipo de desarrollo de la aplicación móvil y stakeholders, se definieron las iniciativas para resolver las necesidades de los clientes presentes en su viaje. Estas se detallan en la Tabla 16.

Definición de la hoja de ruta del MVP	
Objetivo	Definir las funciones que serían parte del MVP
Procedimiento	<p>Asistentes: equipo de desarrollo de la aplicación móvil y stakeholders de TI, marketing, operaciones y venta online.</p> <p>Duración: 4 horas.</p> <p>Equipo y material: pizarra y tarjetas adhesivas.</p> <p>Dinámica: considerando los tipos de persona y su, se elaboró un listado de funciones que podrían responder a solucionar sus necesidades.</p>

Instrumento	Se analizan los tipos de cliente, su viaje y necesidades para elaborar un listado de funciones que serían parte del MVP. En la pizarra se escribió el viaje del cliente y sus preguntas. Para cada una los asistentes escribieron una idea de solución en una tarjeta adhesiva, luego se discutieron cada una, agrupando las similares y votando por las más relevantes, de las cuales se elaboró la hoja de ruta para el MVP. También se consideraron todos los elementos a intentar (Try) de las sesiones de retrospectiva sobre la arquitectura de la aplicación y proceso para ser parte de este MVP.
--------------------	---

Tabla 15 - Definición de la hoja de ruta del MVP

ID	Funciones	Etapas del viaje del cliente
1	Home, con ofertas y promociones para los usuarios.	Inspiración
2	Selección de tienda favorita para personalizar precios asociados a productos y despacho.	Planificación del proyecto
3	Navegación por categorías de productos.	Planificación del proyecto
4	Búsqueda de productos por texto y código.	Planificación del proyecto
5	Crear listas de productos favoritos.	Planificación del proyecto
6	Ficha de producto con: nombre, marca, fotos, disponibilidad, ubicación en tienda, productos relacionados, etc.	Aprendizaje y toma de decisión
7	Escáner de códigos de barra de productos.	Aprendizaje y toma de decisión
8	Login/registro, el mismo que utiliza el sitio web de la compañía.	Compra
9	Carro de compras.	Compra
10	Checkout (despacho y pago).	Compra
11	Seguimiento de compras.	Post venta y administración

Tabla 16 - Funciones de la aplicación móvil definidas para el MVP

Las funciones fueron puestas en una hoja de ruta que se detalla en la Figura 46.

Durante los meses que tomó el desarrollo del MVP, hubo versiones de la aplicación que fueron probadas y utilizadas solo por el equipo de desarrollo y los stakeholders, dado que el producto mínimo es el que podrán usar los usuarios finales considerando todas las funciones definidas.

Fecha	Enero	Febrero	Marzo	Abril	Mayo
Nombre	Versión 0.1	Versión 0.2	Versión 0.3	Versión 0.4	Versión 1.0
Objetivo	Una aplicación funcional para la búsqueda de productos	Ver información detallada de los productos	Funciones para facilitar la compra de productos	Comprar productos	Liberar MVP para usuarios finales
Característica	<ul style="list-style-type: none"> • Home • Selector de tienda • Buscador por texto • Página de resultados de búsqueda 	<ul style="list-style-type: none"> • Escáner de productos • Página de producto • Árbol de categorías • Favoritos 	<ul style="list-style-type: none"> • Login/ Registro • Carro de compras 	<ul style="list-style-type: none"> • Checkout 	<ul style="list-style-type: none"> • Seguimiento de compras
Métrica principal	% búsquedas que dieron resultados de productos	% productos escaneados y con información disponible	Cantidad de usuarios registrados. % usuarios que agregaron productos al carro	Número de de transacciones	% usuarios que revisan el seguimiento de sus compras

Figura 46 - Hoja de ruta 2017 de la aplicación móvil

5.1.4 Desarrollo del MVP en Scrum

El desarrollo de las funciones de front-end mencionadas en la hoja de ruta requirió del desarrollo de una nueva capa de back-end. Para ello, el proceso de desarrollo de ambas capas fue trabajado en Scrum, el marco de trabajo de metodologías ágiles que fue propuesto como solución en la Sección 4.2.1.

Cada Sprint duró 2 semanas y cada día hubo una Daily Scrum para controlar el avance del Sprint. Esto puede verse expresado gráficamente en la Figura 47.

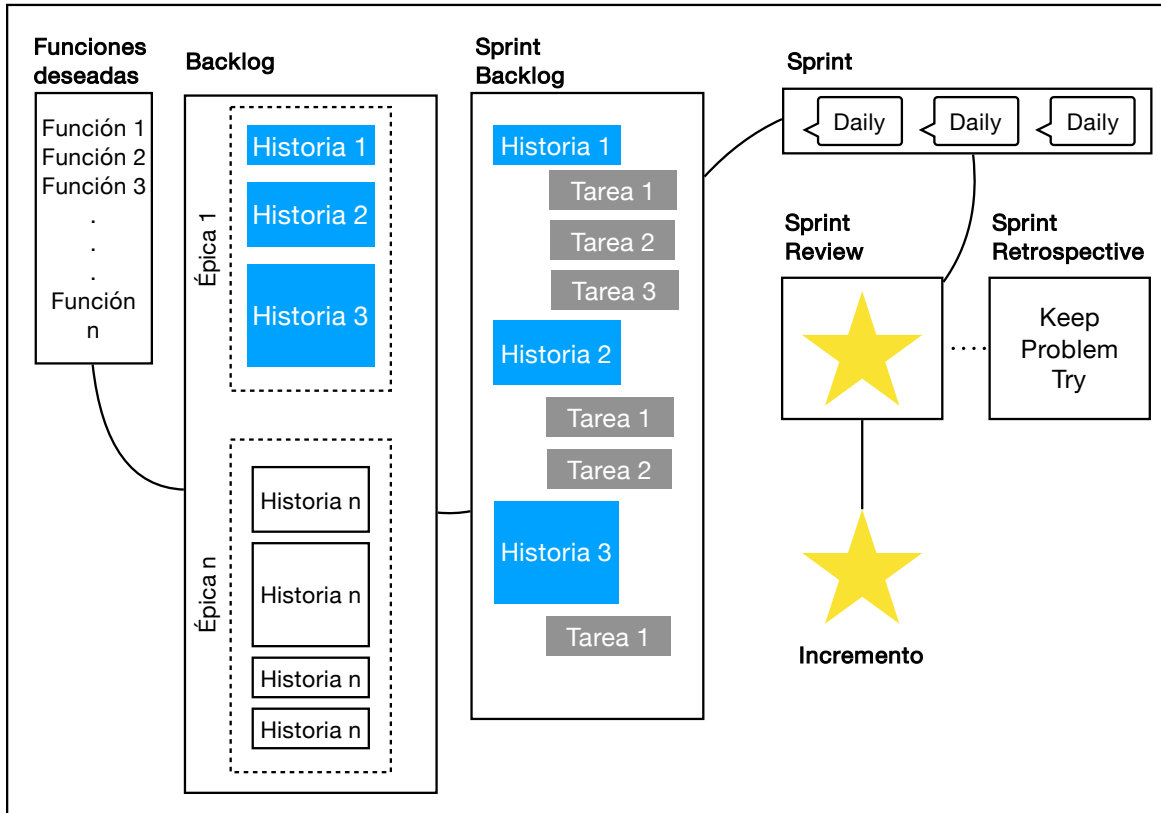


Figura 47 - Dinámica de trabajo Scrum

Como se explicó en la Sección 4.2.3, existe un equipo Scrum para la aplicación móvil Sodimac y un equipo de integración que realiza los desarrollos y conexiones entre los sistemas legados y la aplicación móvil. Ambos trabajaron en el desarrollo del MVP.

El autor del presente proyecto de tesis asumió el rol de Product Owner de la aplicación móvil y escribió las épicas e historias de usuario como se puede ver en la Figura 68 en el Anexo **¡Error! No se encuentra el origen de la referencia..**

Las épicas creadas agrupan funciones por temáticas:

EP1. Configuración Inicial: contiene todas las funciones básicas para la apertura y uso de la aplicación por primera vez por parte del usuario.

EP2. Home: acá se encuentran todas las funciones que contiene el home de la aplicación con la finalidad de ser una vitrina de promociones, productos, categorías y contenido relevante para el usuario.

EP3. Búsqueda y navegación: esta épica engloba a todas las funciones que permiten la navegación a través de la aplicación para encontrar un producto.

EP4. Página de producto: contiene toda la información del producto; dentro de ella hay varias secciones y funciones que evolucionarán durante el tiempo.

EP5: Cuenta de usuario: contiene todas las funciones personales del usuario que están relacionadas con el sitio web.

EP6. Checkout: dentro de esta épica están todas las funciones para la compra de productos y servicios en la aplicación.

EP7: Métricas y analítica: acá se agrupan todas las historias de usuario enfocadas en determinar datos de uso de las aplicaciones como las visitas, usuarios activos, accesos a páginas, uso de botones y menús, detección de errores, etc.

Una vista al Backlog del MVP en Jira está detallada en la Figura 68 en el Anexo **¡Error! No se encuentra el origen de la referencia..**

El flujo en el cual fluyó el trabajo dentro de los Sprints está ilustrado en la Figura 48. En primer lugar, se desarrollan aquellas tareas que tienen que ver con construir los microservicios que van desde los sistemas legados hacia el sistema de la aplicación móvil en la Azure donde expuestos a través de APIs y adicionalmente aquellos servicios que permiten extraer datos para disponibilizarlos en bases de dato en la nube. En la capa de front-end se diseñan e implementan las pantallas y funciones con datos de prueba. Una vez que los datos de back-end están expuesto en las APIs en la nube, las funciones son conectadas con los microservicios finales para mostrar datos reales.

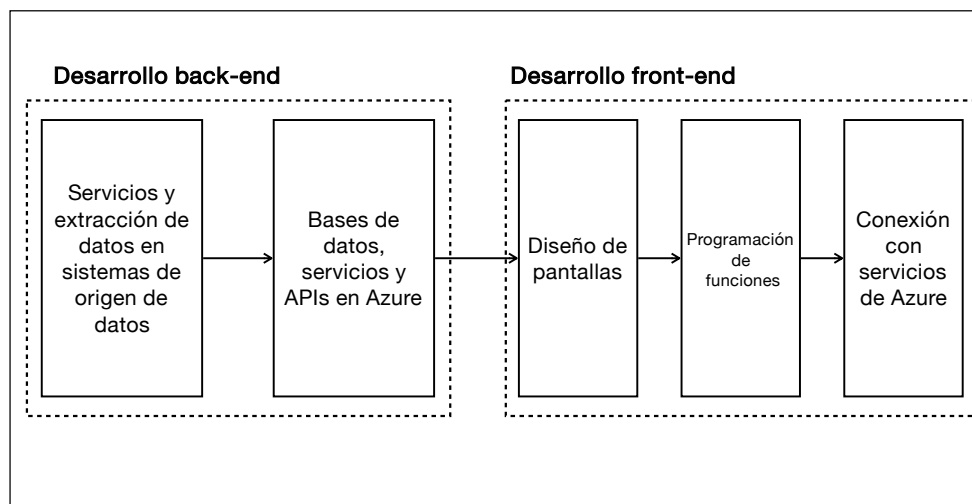


Figura 48 - Flujo de desarrollo de Back-end y Front-end

Historia: Ficha de producto (PDP)

Uno de los atributos más valorados por los usuarios de la aplicación móvil es la información de los productos. A continuación, se detallará el desarrollo de la ficha del producto o PDP durante el Sprint 4 abordado por los equipos Scrum, utilizando esfuerzos conjuntos de front-end y back-end. La Figura 49 muestra el detalle del Sprint donde se incluyeron las historias necesarias para desarrollar la PDP. Dentro de cada historia existieron sub-tareas que cada programador escribió y asignó como elementos a hacer para completar los criterios de aceptación de la historia.

HISTORIA	ID	PUNTOS
[UX/UI] Diseñar PDP	MVP 85	5
[DEV FRONT] Implementar PDP	MVP 86	8
[DEV BACK] Crear servicio GET Products	MVP 87	8
[DEV BACK] Crear API catálogo de productos	MVP 88	13
[DEV BACK] Crear BD Productos	MVP 89	8

Figura 49 - parte del Sprint 4 dedicado a desarrollar la PDP

Las etiquetas de tipo [UX/UI] corresponde al diseño gráfico y de experiencia de uso; la etiqueta [DEV FRONT] corresponde a desarrollos en front-end; las historias con la etiqueta [DEV BACK] corresponden a desarrollo en back-end, como construir la API Catálogo de productos (la cual incluye todos los controladores y métodos necesarios para desplegar datos en la PDP), desarrollar la base de datos BD Catálogo de productos y crear el microservicio que desde el origen (SL2: Sistema de productos II) retorne información que pueda ser luego consultada por la API Catálogo de productos. Ver sección 4.3.3.

Para la implementación de la PDP, se construyó la API Catálogo de productos a través de la historia MVP-88, la cual contiene varios controladores, uno de ellos llamado Products, el que cuenta a su vez con varios métodos para la exposición de datos de productos desde los sistemas legados SL1 y SL2 (Información de productos I y II respectivamente). El método GET Products es uno de ellos y

como se muestra en la Figura 50, al enviar como parámetros el país, el número de la tienda y el código del producto (SKU) hacia el servicio, este retorna toda la información a ser desplegada en la pantalla de la ficha del producto.

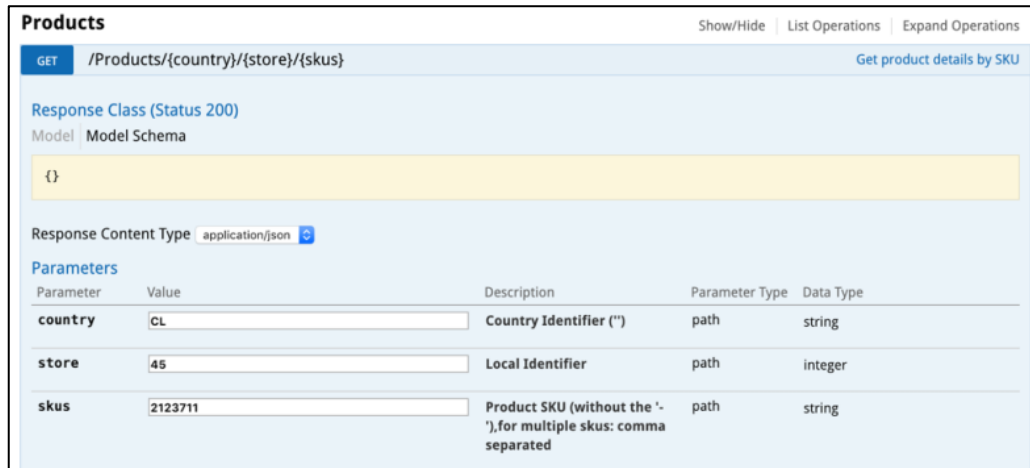


Figura 50 - Controlador Products de la API Catálogo de productos

En la Figura 51 se muestra un ejemplo de la respuesta de GET Products, donde se puede ver los datos que son retornados y luego son desplegados en la PDP. En rojo se destaca parte de estos datos, los que provienen del servicio origen getProductDetails de SL2: Información de productos II desarrollado en la historia MVP-87. En la respuesta del servicio de origen se pueden encontrar los mismos datos de SKU, nombre del producto y marca expuestos en la API en la Figura 52.



Figura 51 - Respuesta del microservicio GET Products

```
1 {
2   "productDetailsJson": [
3     {
4       "status": "OK",
5       "productId": "2123711",
6       "name": "Taladro percutor eléctrico 10 mm 550 W",
7       "brand": "Bauker",
8       "combo": false,
9       "published": true,
10      "webCategoryName": "Set de herramientas eléctricas e inalámbricas",
11      "webCategory": "cat750203",
12      "backEndCategoryName": "KIT TALADRO 10MM",
13      "backEndCategory": "0209020103",
14      "PriceFormat": "C/U",
15      "savings": 8000.0,
16      "COMPANY": 34990.0,
17      "INTERNET": 34990.0,
18      "CMR": 26990.0,
19      "NORMAL": 34990.0,
20      "CMRFrom": "2019-10-28T00:00:00GMT-03:00",
21      "CMRValidity": "2020-01-06T23:59:59GMT-03:00",
22      "stockLevel": "{2123711=302}",
23      "pickupInStore": "{2123711=true}",
24      "bvRating": 5.0,
25      "bvReview": 85
26    }
27  ]
28 }
```

Figura 52 - Respuesta del servicio de origen: getProductDetails de SL2

Algunos datos en la PDP como la descripción o la ubicación del producto en la tienda provienen del sistema SL1: Información de productos I, ver sección 4.3.3. Para ello se construyó una base de datos en la historia MVP-89, en donde se almacenan datos que son extraídos primeramente de SL1 mediante un proceso que luego los deja en una hoja de cálculo en un servidor FTP. Luego, un proceso ETL toma estos datos y los lleva hacia la base de datos en la nube a donde accede la API Catálogo de productos que homologa los datos y los une a los provenientes de SL1 para conformar los métodos que retornan toda la información asociada a un producto.

Gráficamente en la Figura 53 se muestra el flujo de la información desde los sistemas legados hasta la capa de front-end que ve el usuario para la API Catálogo de productos:

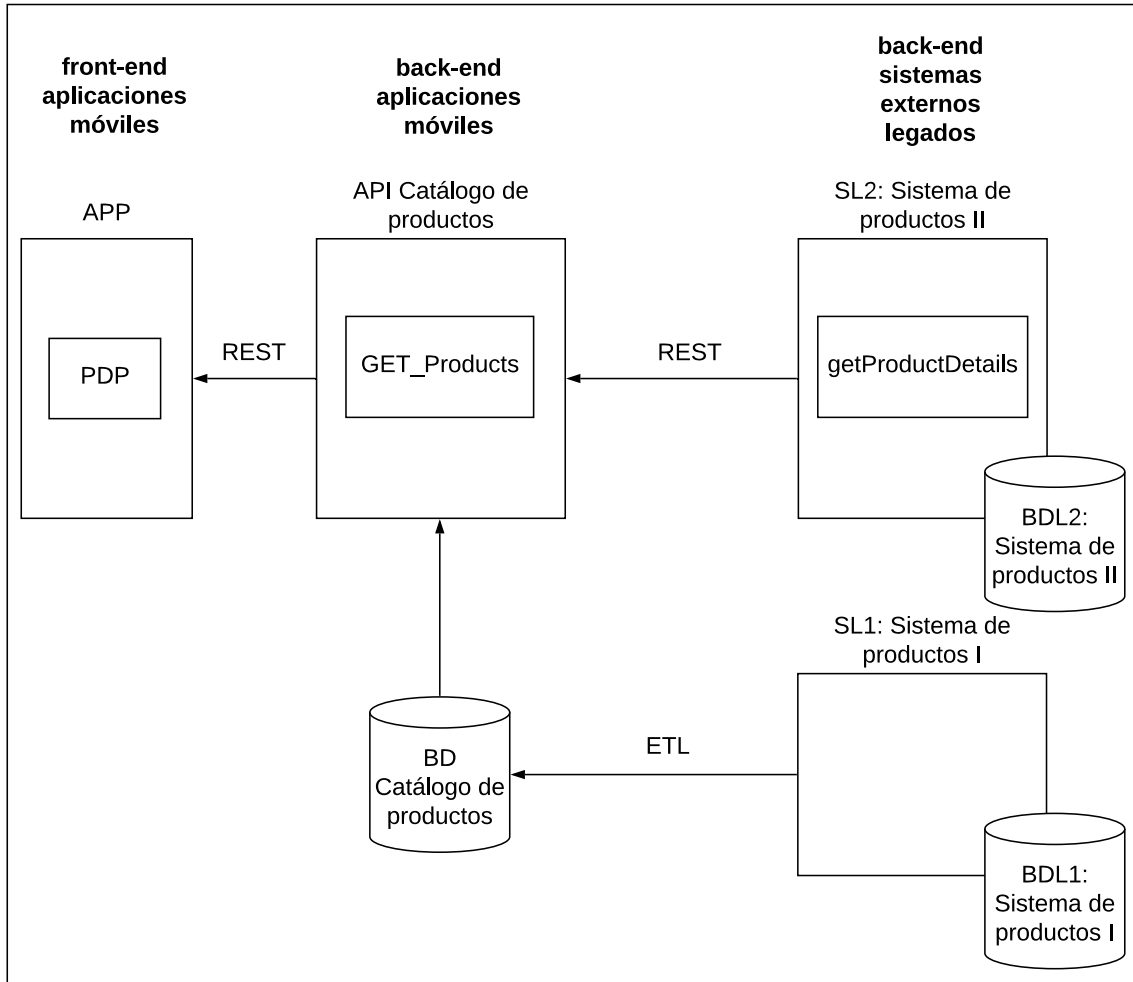


Figura 53 - Flujo de información desde back-end externo a front-end la aplicación móvil

Finalmente, el diseño de la PDP a nivel de pantalla, interacciones y vistas es desarrollado en la historia MVP-85. Una vez listo el diseño que está en HTML por parte del diseñador UX/UI, este es implementado en Ionic V3 por el programador de front-end, integrando las funciones y servicios de la API Catálogo de productos.

Un ejemplo de historia de usuario esta detallado en la Figura 70 en el Anexo 8.4, donde se puede ver el formato definido por el equipo de desarrollo Scrum, sus criterios de aceptación y tareas asociadas.

5.2 Retrospectiva del MVP

Una vez finalizado el MVP, se realizó una retrospectiva del tipo KPT asociado al desarrollo de la nueva aplicación, constituyendo un espacio de inspección para el equipo. La Tabla 17 muestra su protocolo.

Retrospectiva KPT – MVP	
Objetivo	Inspeccionar el último Sprint para identificar las cosas que salieron bien y las posibles mejoras para el siguiente Sprint.
Procedimiento	<p>Asistentes: equipos Scrum (App Sodimac e integración).</p> <p>Duración: 3 horas.</p> <p>Equipo y material: pizarra con una matriz KPT y tarjetas adhesivas para desarrollar las ideas sobre cada aspecto de la retrospectiva.</p> <p>Dinámica de la reunión: el moderador de la reunión define lo que es una retrospectiva y comenta las particularidades del método KPT. Posteriormente, se entregan tarjetas adhesivas a cada asistente para desarrollar las ideas necesarias para el análisis.</p>
Instrumento	<ol style="list-style-type: none"> 1. Durante 15 minutos cada asistente escribe en tarjetas adhesivas las ideas referentes a las cosas buenas a mantener en el equipo (<i>Keep</i>). 2. Al terminar los 15 minutos iniciales, el moderador recoge las tarjetas y las pega en la pizarra en la cuadrícula <i>Keep</i>. Luego, lee cada tarjeta y en conjunto a los asistentes se define cuáles son similares para agruparlas. 3. Se repite el proceso con el aspecto <i>Problem</i>. 4. Una vez que están los problemas agrupados en la sección <i>Problem</i>, se solicita que, considerando estos problemas, se definan acciones a implementar para resolver cada uno de ellos (<i>Try</i>). 5. Una vez terminado el tiempo, el moderador agrupa las tarjetas y pide asignar responsables por cada una para velar que la acción se cumpla. 6. Una vez asignados los responsables se discute sobre el tema en general, para consolidar las visiones de cada asistente respecto a los problemas a afrontar y las acciones que se llevarán a cabo para resolverlos.

Tabla 17 - Protocolo para la retrospectiva KPT del MVP

5.2.1 Resultados de la retrospectiva del MVP

Detalles de los aspectos a conservar, *Keep*:

KM1. Compromiso de los miembros de los equipos con el objetivo del MVP: todos aportaron a que el MVP pudiera ser puesto en manos de los usuarios finales trabajando bajo un mismo objetivo común.

KM2. Uso de microservicios: permitió acceder a los datos de una forma mucho más ordenada y simple desde la capa de front-end, facilitando además la integración con distintos sistemas y en el futuro con otros sistemas o aplicaciones móviles.

KM3. Diseñar funciones pensando en los tipos de usuario: Poner al usuario en el centro de las decisiones de diseño, el equipo tuvo confianza en desarrollar algo que puede ser aceptado de buena forma por los usuarios finales.

KM4. Liberar versiones tempranas para obtener validación de los usuarios: previo a la liberación del MVP a producción, múltiples versiones fueron liberadas solo a usuarios internos de la organización, lo que permitió obtener retroalimentación temprana de la aplicación.

KM5. Usar Ionic como marco de desarrollo, sin duda que Ionic V3 ha sido un gran avance en términos de velocidad y facilidad de adopción para los programadores, permitiendo probar rápidamente las funciones desarrolladas, ejecutar un mismo código fuente para Android e iOS y además integrar modularmente funciones ya desarrolladas por terceros.

KM6. Usar Scrum, el utilizar metodologías ágiles y principalmente Scrum ha ayudado a los equipos a avanzar con claridad en el desarrollo, ya que este marco entrega las herramientas para progresar en cada iteración de forma ordenada y transparente hacia la organización.

Detalle de los problemas, *Problem*:

PM1. Mala comunicación con stakeholders: algunos interesados declararon no estar enterados de algunas decisiones o ser parte del proceso de creación del MVP.

PM2. Estimación no acertada sobre el peso de las tareas: se sobre- y sub-estimó esfuerzo respecto al desarrollo de tareas en cada uno de los Sprints.

PM3. Requerimientos poco definidos: muchos aspectos técnicos y de negocio no estaban totalmente claros en las historias de usuario y se agregaron durante el Sprint.

PM4. No respetar las Dailys: muchas veces los miembros del equipo llegaban tarde o simplemente no se juntaban a realizar la reunión diaria de control de estado del Sprint.

PM5. Subir requerimientos al Sprint en medio de este: con la finalidad de adelantar trabajo y avanzar más rápido, el equipo añadía más trabajo a los Sprints, subiendo requerimientos a mitad de cada ciclo.

Detalle de los aspectos a intentar, *Try*:

TM1. Incluir a los stakeholders necesarios en los Sprint Reviews: esto facilita la transparencia hacia la organización de los avances, decisiones y del desarrollo de la aplicación móvil como producto a nivel general.

TM2. Incluir a los stakeholders en los refinamientos: cuando sea necesario tener mayor claridad sobre un requerimiento se deben invitar a las personas expertas en cada área.

T3. Acordar un DOR (*Definition of Ready*): permite tener una definición de qué características debe tener un requerimiento para poder ingresar al Sprint Backlog [48].

T4. El Scrum Master debe velar por la metodología: esto es parte del rol que debe cumplir dentro del equipo y hacia la organización, haciendo que se respeten y la metodología sea conocida y adoptada por todo el equipo Scrum.

5.2.2 Conclusiones

El desarrollo de un MVP resultó beneficioso para el equipo de desarrollo dado que incorporó varias herramientas que permitieron ordenar el trabajo y dar visibilidad hacia las áreas interesadas en el proyecto, como el uso de Jira para gestionar el trabajo, los Sprint Reviews para mostrar el incremento logrado en un Sprint o los gráficos *Burndown* [49] para mostrar el avance en un Sprint. Por otra parte, el utilizar nuevos enfoques como servicios en la nube y un nuevo paradigma en el desarrollo de front-end, tuvo una recepción positiva dadas las características que permiten ahorrar tiempo de desarrollo y puesta en producción.

Los aspectos negativos pasaron en gran medida por la inexperiencia del equipo al trabajar juntos por primera vez bajo una nueva metodología y tecnología, lo cual es parte del proceso de crecimiento y madurez de los equipos de desarrollo. Este fenómeno puede caracterizarse en cinco fases: *Forming*, *Storming*, *Norming*, *Performing* y *Adjourning* [50]. Tal como lo describe el gráfico en la Figura 54, los equipos van evolucionando desde un rendimiento inicial bajo, pero creciente, pasando por una caída y luego un aumento y estabilización. Esto tiene que ver con las relaciones personales en los equipos, sus interacciones, la resistencia al cambio y la resiliencia frente a escenarios adversos. Es natural entonces que los equipos comiencen en una fase de formación (*Forming*) con algo de incertidumbre frente al proceso, pero con foco en el propósito del equipo.

Luego viene la tormenta (*Storming*) donde aparecen los conflictos internos. Le sigue la reforma (*Norming*) donde hay una reconciliación en el equipo y se forma cohesión entre sus miembros. Luego el rendimiento (*Performance*) demostrará que el equipo puede ser independiente y eficiente, y finalmente llegar a la fase de levantamiento o conclusión (*Adjourning*) en donde el equipo finalmente se consolida y es reconocido por su trabajo.

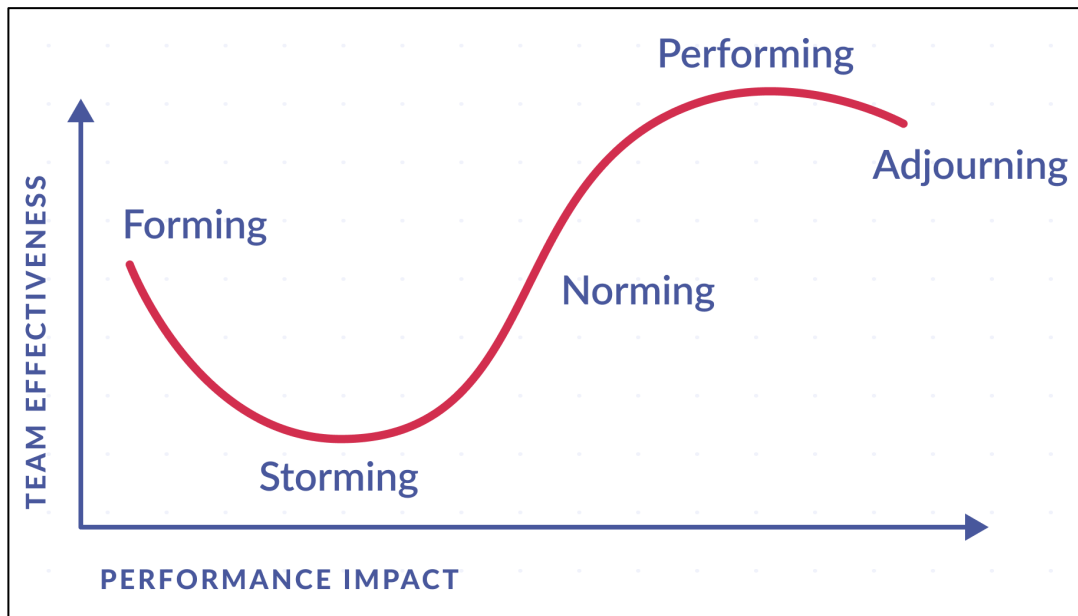


Figura 54 - Fases en equipos de desarrollo [50]

5.3 Validación por percepción de mejora

Esta evaluación posee un carácter cualitativo y se utilizó como herramienta de medición una encuesta a los stakeholders ligados al desarrollo de la aplicación móvil. Dado que durante el proceso de desarrollo del proyecto de tesis hubo varios cambios de personas en el equipo de desarrollo de la aplicación móvil (Squad App Sodimac e Integración) y stakeholders, se consideró un universo particular de encuestados y orden para las entrevistas, realizando preguntas con un enfoque diferente dado el contexto en el que participaron los encuestados. Los stakeholders se clasificaron en quienes han estado presentes desde el proceso y arquitectura previa y las nuevas personas que se incorporaron durante y posterior a la implementación del proceso y arquitectura nueva:

A. Transversales: miembros del equipo de desarrollo y stakeholders que han estado presentes durante el antiguo proceso y arquitectura y también durante el nuevo proceso y arquitectura. Cantidad: 3 personas.

B. Nuevos: personas del equipo de desarrollo y stakeholders que se incorporaron posterior a la implementación de la nueva arquitectura y proceso. Cantidad: 4 personas.

5.3.1 Protocolo

En la Tabla 18 se detalla el protocolo para la encuesta que se realizó a los miembros del equipo de desarrollo y stakeholders para validar el MVP desde un punto de vista cualitativo. En la Figura 55 se detalla el grupo de encuestados.

Encuestas al equipo de desarrollo y stakeholders	
Objetivo	Obtener información para determinar el impacto del nuevo proceso de desarrollo y arquitectura implementados durante el MVP de la nueva aplicación móvil.
Procedimiento	<p>Encuestados: equipos Scrum de desarrollo de la aplicación móvil: App Sodimac e Integración y stakeholders. Dentro hay personas de tipo Transversales y además Nuevos.</p> <p>Cantidad de encuestados: 7 personas.</p> <p>Duración de la encuesta: 20 minutos.</p> <p>Equipo y material: formulario online de Google Drive.</p> <p>Dinámica de la encuesta: se envía un correo electrónico a los encuestados comentando el objetivo y un enlace que abre un formulario de Google Drive con la encuesta. Se separan los encuestados en Transversales y Nuevos, cada segmento tiene una encuesta distinta (A y B respectivamente), enfocada en un aspecto del proceso o arquitectura según el contexto en el que participaron como se puede ver en la Figura 55.</p>
Instrumento	<ol style="list-style-type: none"> 1. Cada persona realiza la encuesta en la URL entregada. 2. Una vez pasado el plazo dado (1 semana) se revisan los resultados.

Tabla 18 - Protocolo para las encuestas de validación del MVP al equipo de desarrollo y stakeholders

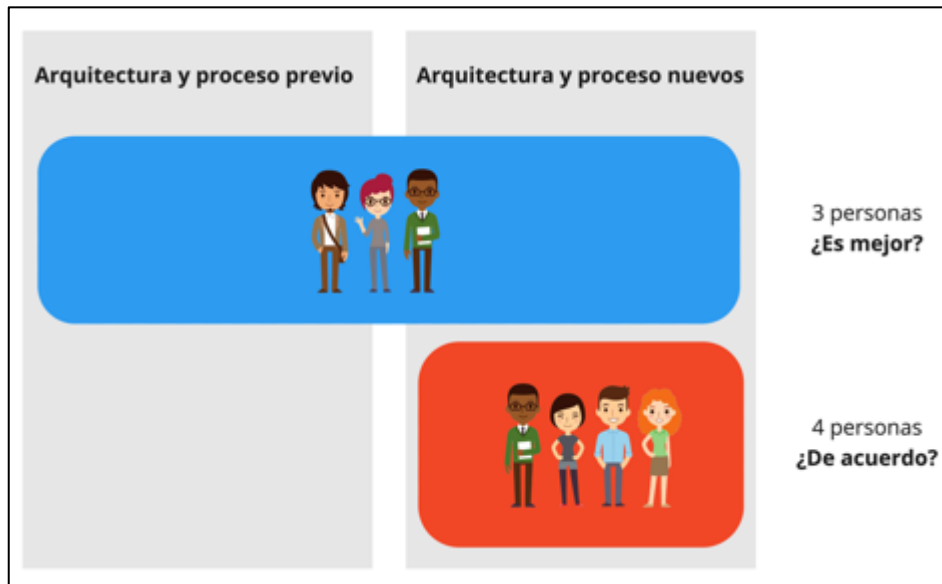


Figura 55 - Conjunto de entrevistados para validación cualitativa

En la sección 8.2.4 y 8.2.5 del Anexo se encuentra el detalle de las encuestas realizadas a personas transversales y nuevas en la organización.

5.3.2 Ejecución

La encuesta es enviada por correo electrónico durante el mes de diciembre de 2018 a los miembros del equipo de desarrollo y stakeholders que conocieron y experimentaron las 2 realidades (antiguo versus nuevo) proceso de desarrollo y arquitectura.

5.3.3 Resultados

Los resultados **RE1** a **RE4** (detallados en la Sección 8.7 del anexo) muestran que en general las personas encuestadas coinciden en que la situación posterior a la implementación de la nueva arquitectura y proceso de desarrollo de software para la aplicación móvil es mucho mejor que la situación anterior. Resumen de las encuestas **RE1** a **RE4**:

- **RE1:** encuesta a personas transversales el proceso de desarrollo.
- **RE2:** encuesta a personas nuevas sobre el proceso de desarrollo.
- **RE3:** encuesta a personas transversales sobre la arquitectura.
- **RE4:** encuesta a personas nuevas sobre la arquitectura.

RE5. La Tabla 19 presenta los resultados de la encuesta a personas transversales y nuevas respecto al logro de los objetivos, en total fueron 7 encuestados (P1 a P7) y dieron su opinión frente a al proceso de desarrollo, arquitectura y MVP. Alineados con los objetivos específicos del proyecto de tesis.

	TRANSVERSALES			NUEVOS			
	P1	P2	P3	P4	P5	P6	P7
Nuevo proceso: lograr versiones en producción con mayor velocidad y frecuencia	Muy de acuerdo	Muy de acuerdo	Muy de acuerdo	Muy de acuerdo	De acuerdo	De acuerdo	De acuerdo
Nueva arquitectura: mayor escalabilidad y coherencia de la información	De acuerdo	De acuerdo	De acuerdo	Muy de acuerdo	Muy de acuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo
Las aplicaciones móviles entregan valor y significado a usuarios, reflejado esto en su calificación	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	Ni de acuerdo ni en desacuerdo

Tabla 19 - Resumen de encuesta sobre el logro de los objetivos

RE6. La Tabla 20 detalla los resultados de la encuesta a personas transversales y nuevas respecto a la adopción del nuevo proceso de desarrollo y arquitectura, considerando como el cambio impactó en términos de su rol y para la compañía:

		TRANSVERSALES			NUEVOS			
		P1	P2	P3	P4	P5	P6	P7
Proceso de desarrollo	Su rol	6	7	7	7	7	7	6
	La compañía	3	3	4	1	1	2	3
Arquitectura	Su rol	6	7	7	5	7	7	7
	La compañía	1	2	2	1	1	1	1

Tabla 20 - Resumen de encuesta sobre la adopción del nuevo proceso y arquitectura

5.3.4 Análisis de los Resultados

La aceptación por parte de las personas transversales y nuevas en el equipo ha sido positiva respecto al nuevo proceso de desarrollo y la nueva arquitectura implementada. En algunos aspectos, como poner al cliente al centro de los requerimientos, existe aún la negativa por algunos miembros del equipo al ver que no siempre se priorizan las necesidades de los usuarios finales.

Respecto al cumplimiento de los objetivos, el equipo está de acuerdo con que se ha logrado una mayor frecuencia y velocidad en la puesta en producción de versiones de la aplicación, logrando por otro lado una escalabilidad a nivel de arquitectura y calidad en los datos que son desplegados. Finalmente, la aplicación está generando un impacto positivo en los usuarios quienes valoran sus funciones y contenido.

Se concluye entonces que el nuevo proceso y arquitectura validados con el MVP resultó exitoso para la compañía si se compara con la situación anterior, y hay posibilidades de mejora en cuanto a arquitectura, desarrollo de software y metodologías ágiles. Puntos importantes son el adoptar herramientas o instancias que permitan una mayor interacción entre stakeholders, incluir pruebas unitarias en front-end y back-end, adoptar un enfoque *Test-Driven* para agilizar aún más el proceso de pruebas y calidad del código fuente y finalmente madurar en cuanto al uso de Scrum.

5.4 Validación cuantitativa

Para medir el éxito cuantitativo del MVP, se revisaron indicadores alineados con los objetivos del presente proyecto de tesis, como la variación positiva en las descargas, el crecimiento en transacciones, el aumento en la frecuencia de versiones liberadas a producción, la disminución del tiempo de puesta en producción y el impacto en las evaluaciones de aplicaciones móviles.

5.4.1 Descargas, audiencia y transacciones

Descargas

Como se puede ver en la Figura 56, previo al lanzamiento del MVP en el mes de mayo de 2017, las instalaciones desde enero mantenían una constante en cuanto a número de instalaciones semanales. Luego, cuando es lanzado el MVP, se puede ver un alza destacada en el cuadro A, seguido de una normalización natural en B para luego retomar en C el aumento de las instalaciones, las cuales fueron apoyadas por el desarrollo de nuevas versiones y eventos como Cyber Day y Black Friday que aumentan la exposición de las plataformas de comercio electrónico en general.



Figura 56 - Instalación por dispositivos Android semanal, 1 enero al 31 de diciembre de 2017 [30]

Respecto a las instalaciones en iOS, como se puede ver en la Figura 57, no hay una variación importante en las descargas más allá de lo evidenciado en el cuadro A durante los primeros días de junio. Sin embargo, en B se puede ver como en noviembre gracias a Cyber Day sí hay un aumento en las descargas.

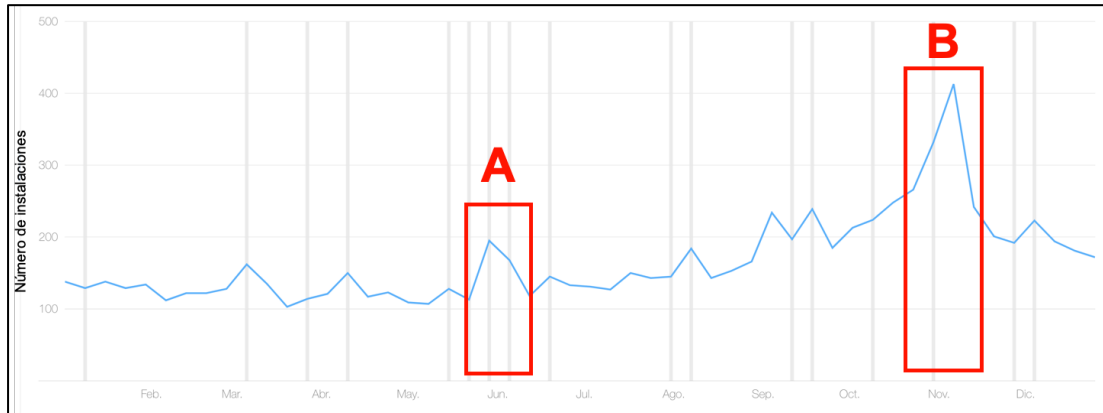


Figura 57 - Instalación por dispositivos iOS semanal, 1 enero al 31 de diciembre de 2017 [31]

Vale la pena mencionar que la participación en el uso de la aplicación móvil por parte de dispositivos Android es muy superior a iOS, siendo un 81% versus un 18% respectivamente [33], lo cual se muestra en la Figura 58 y por lo que también puede haber un bajo impacto en las instalaciones de iOS cuando fue publicado el MVP.

Sistema operativo	Usuarios	% Usuarios
1. Android	100.107	81,00 %
2. iOS	23.427	18,96 %
3. Windows	39	0,03 %
4. Macintosh	10	0,01 %
5. BlackBerry	6	0,00 %
6. (not set)	1	0,00 %

Figura 58 - Usuarios por sistema operativo [33]

Audiencia e indicadores comerciales

En la Figura 59 se puede ver la evolución desde mayo de 2017 hasta diciembre de ese año en cuanto a ventas y número de transacciones (el MVP se lanzó en mayo de 2017). Por otra parte, la evolución de visitas a la aplicación móvil puede verse en la Figura 59, en donde a largo de 2017 muestra un incremento de casi un 2.000% a finales de diciembre de ese año [33].

Como se puede notar, en ambos gráficos existe un incremento destacado en noviembre de 2017 lo cual se atribuye al evento Cyber Day que siempre atrae muchas visitas y transacciones en el rubro del comercio electrónico.

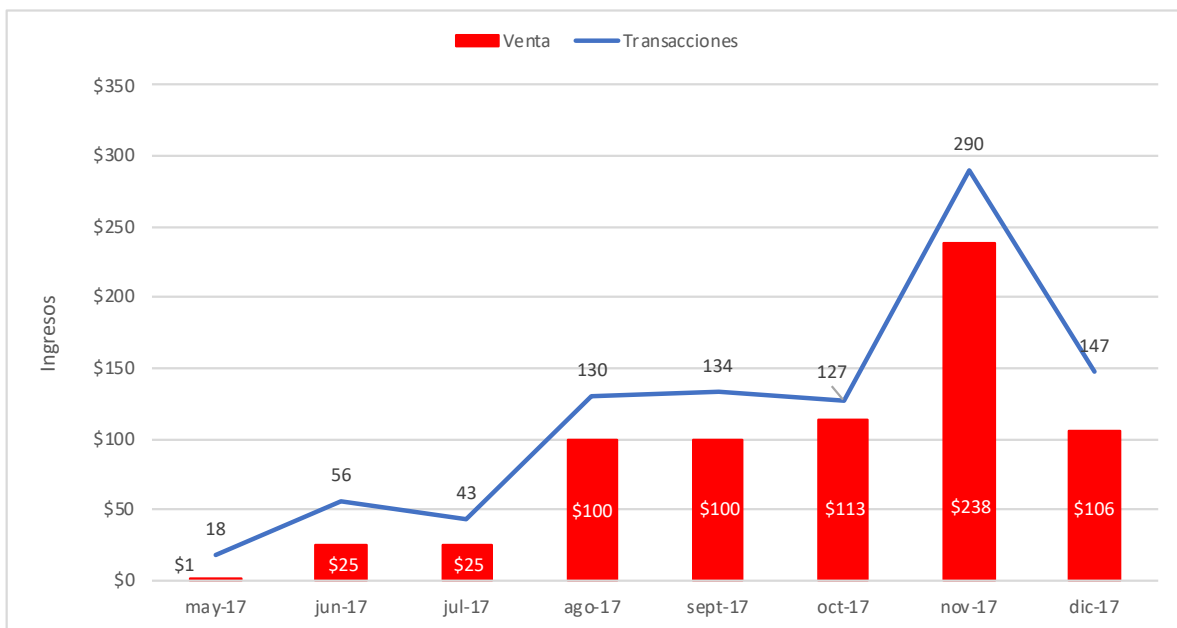


Figura 59 - Indicadores claves de transacciones, venta y tasa de conversión [33]

Las ventas están expresadas en unidades ficticias dada la sensibilidad y confidencialidad de la información para la compañía, pero guardan la proporción de la realidad.

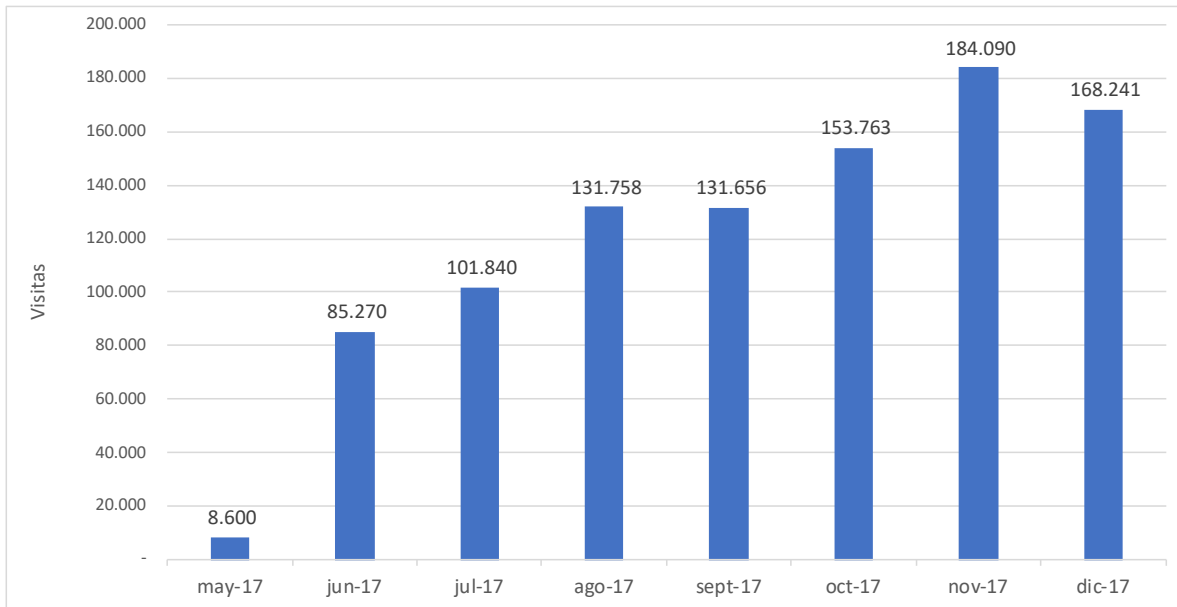


Figura 60 - Evolutivo de visitas a la aplicación móvil desde mayo a diciembre de 2017 [33]

5.4.2 Velocidad y Sprints

El equipo Scrum mantuvo un tiempo constante de desarrollo en Sprints de 2 semanas, donde al final de cada uno (y luego de la evaluación del equipo) se determinó si se publicaba el incremento para los usuarios (internos) o no, lo mismo ocurrió con el ultimo Sprint previo a la publicación para usuarios finales.

El flujo completo de desarrollo (Sprint) y publicación toma algo más de 2 semanas en verse disponible en las manos de los usuarios dado los tiempos necesarios para pruebas funcionales, UAT, compilación, integración e implementación. Por otro lado, existe la dependencia de publicación por parte de las tiendas de aplicaciones móviles, en especial en Apple (iOS) donde hay un periodo estimado de 3 a 5 días hábiles de revisión manual sin considerar que puede haber observaciones que rechacen la versión candidata, la cual se debe corregir y volver a enviar para ser evaluada nuevamente. Gráficamente el flujo de desarrollo desde el Sprint hasta la publicación se detalla en la Figura 61.

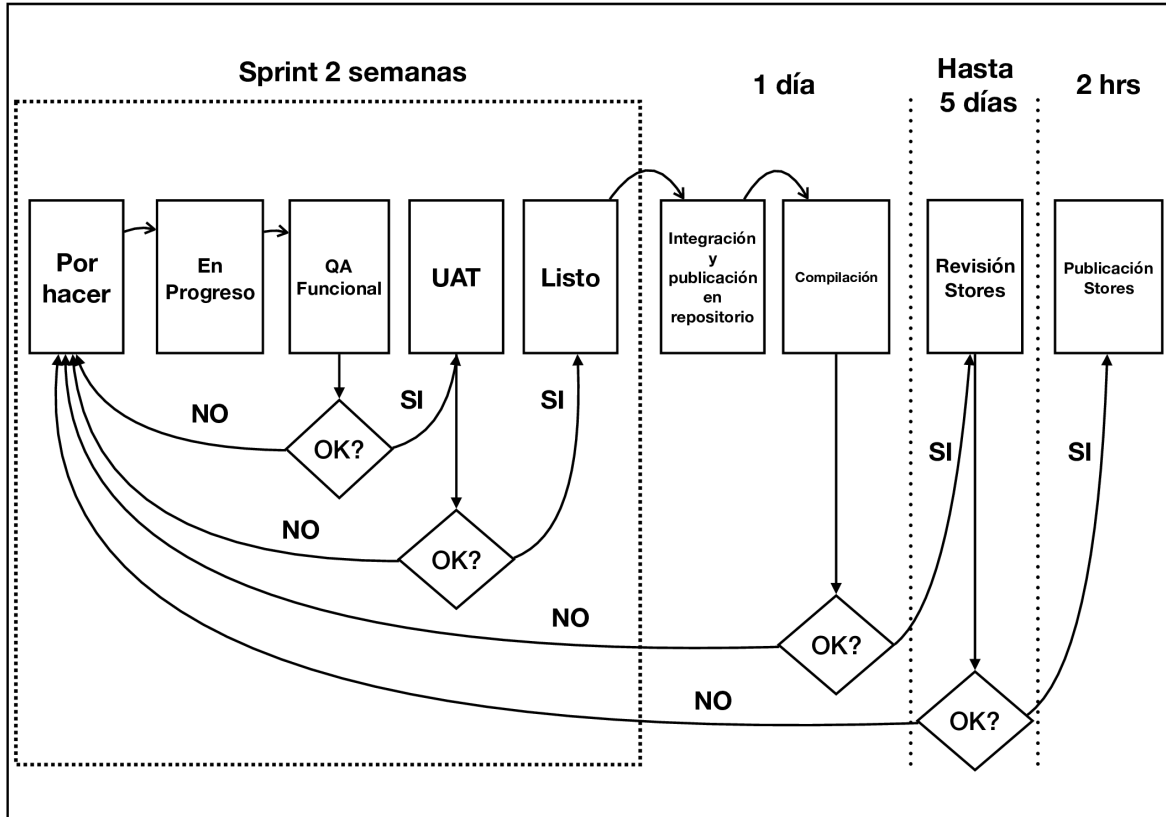


Figura 61 - Flujo de trabajo desde el sprint hasta la publicación de una aplicación móvil

Más allá de este pronóstico (que en el peor de los casos podría tomar incluso más de un mes en publicar una nueva versión de la aplicación para iOS), la experiencia del MVP demostró que el tiempo promedio que toma publicar una nueva versión de la aplicación para iOS es menor a 3 semanas, lo que es sin duda un avance importante en cuanto al tiempo de puesta en producción de nuevas versiones en el anterior proceso, donde tomaba un mes o incluso más.

Si se dejara de lado el tiempo inevitable de 3 a 5 días hábiles de Apple, estamos hablando de casi 2 semanas para poner en producción código de front-end. El caso de back-end es distinto, dado que no necesita de la revisión de terceros más que el propio proceso de QA del equipo para pasar a producción.

En la Figura 62, se muestra la tendencia evolutiva de las versiones puestas en producción y publicadas en las tiendas de aplicaciones desde septiembre de 2016 (previo a eso no hay datos en Play store o App Store Connect) a diciembre de 2017, en donde claramente hay un aumento en la frecuencia de publicación luego del MVP, el cual fue publicado en mayo de 2017 apareciendo en rojo en la Figura 62. Luego de ello, se puede ver como en general se publican 2 versiones por mes, lo cual calza con las 2 semanas por Sprint. En mayo se lanzaron 4 versiones dado que se incluyeron rápidamente en producción algunas correcciones a errores de la nueva versión. Las versiones

previas al MVP (mientras este se construía) corresponden a versiones correctoras de pequeños errores en la antigua aplicación móvil.

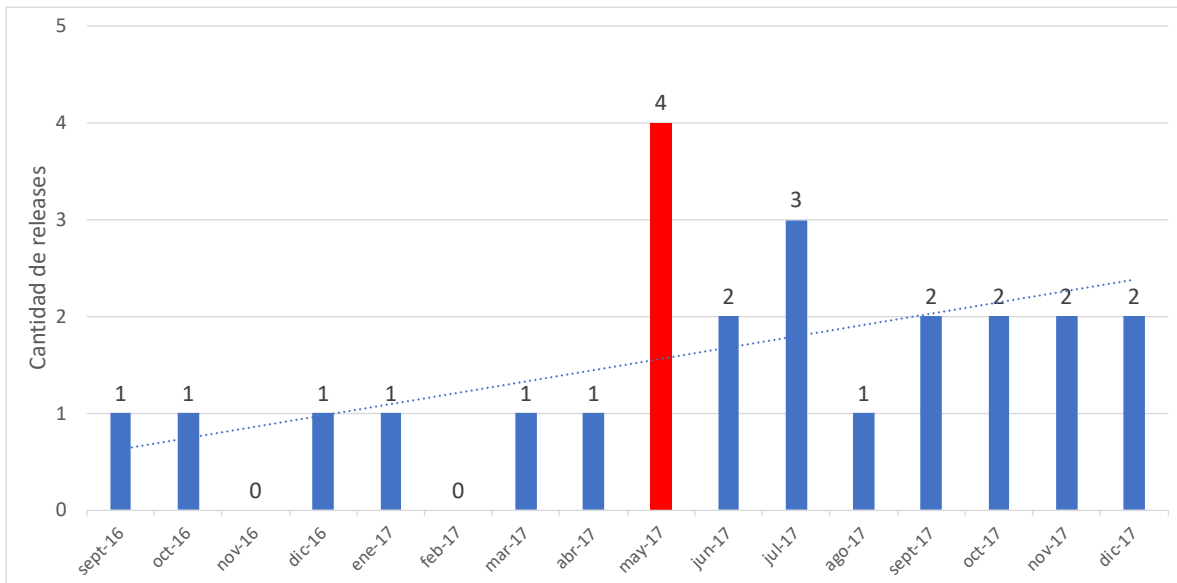


Figura 62 - Gráfico evolutivo de versiones publicadas de la aplicación móvil

Para estimar el esfuerzo de realizar las funciones de cada sprint, el equipo proyectó su capacidad a través de puntos de historia, los cuales fueron asignados a cada una de las 47 historias al final de cada refinamiento. Las 47 historias fueron abordadas en 10 Sprints durante los 5 meses que duro el desarrollo del MVP. La forma de puntuar fue basada en *Planning Poker*, una técnica ágil para estimar y planificar basada en el consenso del equipo de desarrollo [51]. El equipo de la aplicación móvil utilizó los valores: 2, 3, 5, 8 y 13 como puntos de historia.

A través de un gráfico tipo *Burndown* podemos revisar la evolución del trabajo restante sprint por sprint [52], tal como aparece en la figura Figura 63 en donde el promedio por Sprint fueron 43 puntos, sin embargo al comienzo, el equipo comenzó entregando muchos menos, pero a medida que había más confianza entre los miembros del equipo, claridad en los requerimientos y conocimiento de la plataforma el trabajo fluyó más rápido y si bien es cierto hubieron Sprints como el número 5 en donde hubo una caída en la cantidad de puntos entregados, esto fue dado por historias bloqueadas en el Backlog por bugs o incidencias que tomaron un segundo sprint en ser corregidas.

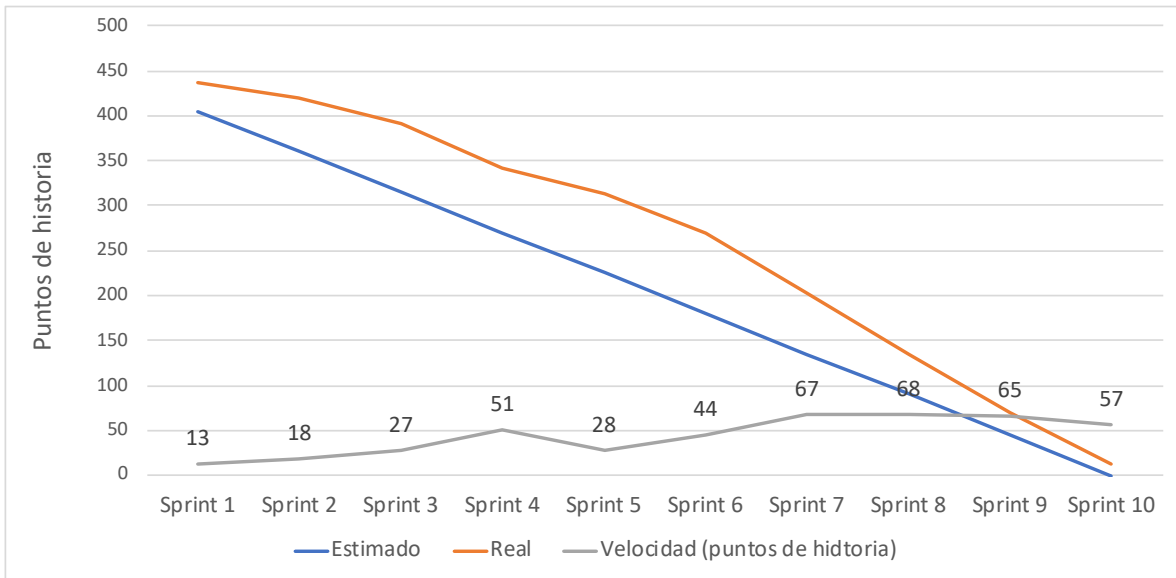


Figura 63 - Gráfico Burndown del MVP

5.4.3 Valoración de los usuarios

La valoración general de los usuarios una vez puesta en producción la nueva versión del MVP resulto positiva, logrando un gran impacto en términos del número de calificaciones y estrellas asignadas en Android. Lamentablemente en iOS no se registraron comentarios suficientes para obtener una medición comparativa con Android.

En la Figura 64 se observa que desde mayo de 2017 ha habido un incremento en la valoración promedio acumulada de la aplicación móvil [30]. Si bien es cierto, esta aun no logra un incremento mayor, el impacto de muchos años recibiendo calificaciones negativas entre 1 y 3 estrellas, han afectado al promedio acumulado que el usuario ve al momento de descargar la aplicación.

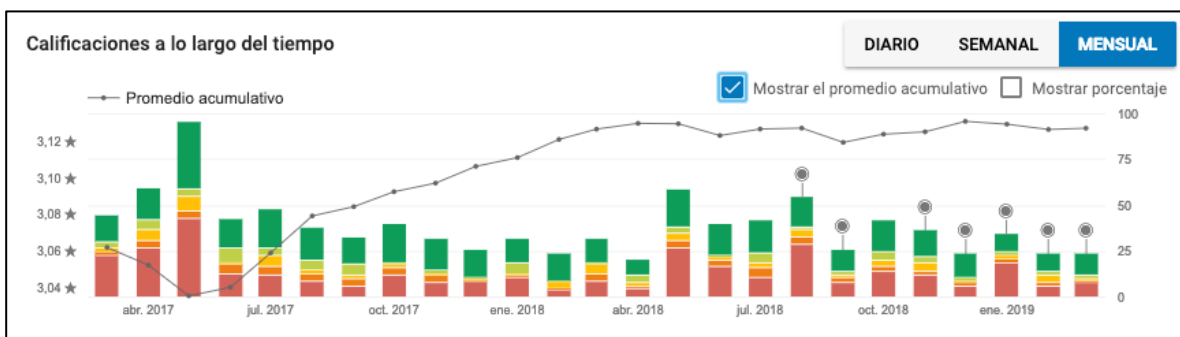


Figura 64 - Calificaciones a lo largo del tiempo en Play Store (Android) [30]

Algunos de los comentarios en Play Store fueron los siguientes, resumidos la Figura 65.

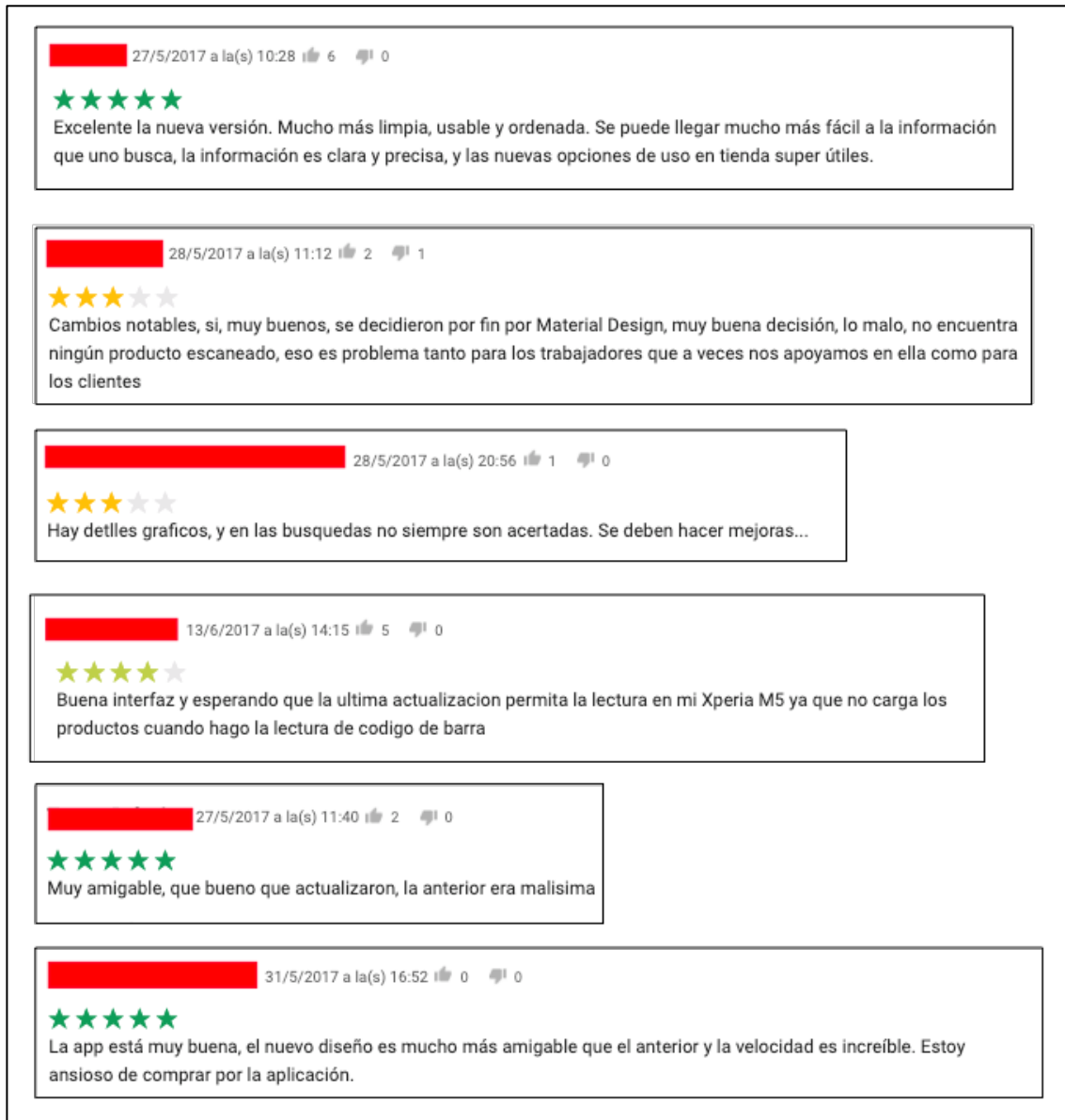


Figura 65 - Valoraciones en Play Store (Android) sobre la nueva versión de la Aplicación Móvil [30]

El mejorar la calificación es complejo dado que se necesita un gran número de usuarios que califiquen positivamente con 5 estrellas. En la actualidad, se está trabajando con el equipo de marketing de la compañía para atraer nuevos usuarios a la aplicación móvil según distintas estrategias de comunicación, tanto en tiendas físicas como en medios digitales.

5.4.4 Análisis de los resultados

En todos los aspectos revisados, existe una mejora evidente respecto a la situación anterior. Cualitativamente, el nuevo proceso de desarrollo es visto como mejor que el anterior en términos de adopción, facilidad de desarrollo, transparencia y trabajo colaborativo. Sin embargo, queda mucho por madurar en aspectos la adopción del proceso y el ejercicio Scrum, un ejemplo simple de aquello es el bajo compromiso con llegar a la hora a las Daily Scrum, respetar los 15 minutos acordados por el equipo o a veces faltar a ellas sin avisar.

Respecto a la nueva arquitectura, esta es vista como mejor que la anterior en términos de robustez, actualización en cuanto a estándares de desarrollo (framework de front-end, virtualización de servidores en la nube, orientación a microservicios, automatización de etapas en el canal CI/CD y APIs), lo que permite lograr una mayor escalabilidad para dar operación a la aplicación móvil. Por otro lado, queda trabajar en afinar aún más la automatización de procesos como las pruebas unitarias, lo que permitirá acelerar aún más la puesta en producción y asegurar la calidad del código fuente.

En términos cuantitativos se evidenció un incremento en el número de versiones publicadas, pasando de 1 mensual a 2 mensuales en promedio. Además, se aceleró la puesta en producción pasando de 4 semanas a 2 bajo un nuevo proceso de desarrollo. Por otro lado, se lograron cambios que impactaron positivamente en los usuarios, como el aumento en las descargas y visitas, además del incremento en las transacciones mensuales (venta y unidades vendidas), que, si bien es cierto, antes del MVP estas eran cero, luego de ser publicada la nueva versión de la aplicación móvil junto a la función de compra de productos, estos indicadores se han mantenido al alza mes a mes de forma constante.

Finalmente, la opinión de los usuarios de la aplicación móvil ha variado positivamente, sobre todo luego del MVP, donde fue notoria la mejora en cuanto a experiencia de uso, funciones y estabilidad general de la aplicación. Sobre este último aspecto queda bastante por seguir mejorando, dado que las malas calificaciones históricas (desde el año 2012) han empañado el promedio global de la aplicación.

5.5 Resumen

El desarrollar una nueva versión de la aplicación móvil de Sodimac, permitió validar positivamente el uso de un nuevo proceso de desarrollo de software basado en la agilidad y una nueva arquitectura basada en la nube. Los miembros del equipo reconocieron las ventajas de trabajar con roles, eventos y entregables definidos, lo cual permitió formalizar ágilmente el desarrollo del MVP. La nueva arquitectura presentó grandes mejoras respecto a la anterior en cuanto a modularidad y flexibilidad, pero quizás lo más importante fue la valoración positiva de los usuarios de la aplicación móvil en las tiendas de aplicaciones, además del incremento en descargas, visitas y ahora en también transacciones comerciales dentro de la aplicación. Finalmente, el equipo de desarrollo y stakeholders creen que, el que nuevos usuarios utilicen la aplicación y valoren sus funciones en su experiencia con Sodimac, es una consecuencia de la nueva organización y soporte tecnológico para el desarrollo la aplicación móvil en la compañía.

6. Conclusiones

En este capítulo se detalla el trabajo realizado, comparándolo con el objetivo general del presente proyecto de tesis. A continuación, se revisa el impacto causado por la solución implementada. Después se exponen las lecciones aprendidas durante el desarrollo del proyecto de tesis y finalmente se proponen mejoras a modo de trabajo futuro.

6.1 Trabajo realizado

El objetivo general del presente proyecto de tesis es: “es crear en la compañía un canal de aplicación móvil valorado positivamente por sus usuarios, donde puedan comprar y ser asistidos en sus proyectos hogar y construcción”.

Para cumplir el objetivo planteado, se realizó un análisis cualitativo y cuantitativo respecto a los problemas principales que afectaban a la compañía en los aspectos de desarrollo de software, como: proceso y arquitectura.

Como solución se propuso la adopción de un nuevo proceso de desarrollo de software basado en agilidad y el desarrollo de una nueva plataforma tecnológica a través de la virtualización de servidores, permitiendo escalar recursos de forma dinámica y a demanda en caso de experimentar alta demanda, APIs que permitieron homologar la información de distintas fuentes, normalizando datos y exponerlos de forma segura y ordenada bajo protocolos a la aplicación móvil. También se implementaron microservicios que se hicieron cargo del registro de usuarios o la ficha de los productos, componentes que anteriormente eran parte de un monolito en uno de los sistemas legados de la compañía. Hoy estos componentes modulares pueden ser llamados y reutilizados en distintas partes del código en una o varias aplicaciones y por cualquier programador con acceso a los repositorios fuente. Finalmente, la automatización de tareas como smoke test, integración a ramas de desarrollo, control de versiones, compilación de versiones y publicación en las tiendas de aplicaciones aceleraron los procesos manuales que antes tomaban incluso horas de configuración.

Para validar la solución propuesta, se realizó un MVP dirigido por una visión de negocio centrada en el usuario final y la entrega constante de valor y obtención de retroalimentación, a través de un piloto basado en la aplicación móvil Sodimac, la cual fue rediseñada en términos de front-end y back-end a través del uso de Scrum como marco de trabajo. Esta nueva versión de la aplicación móvil además constituyó un cambio de paradigma pasando de un desarrollo de software nativo para Android e iOS a un desarrollo unificado, agilizando el tiempo de desarrollo y optimizando recursos, puesto que ahora todos los programadores trabajan sobre un mismo código fuente.

La validación del MVP fue dada por los usuarios finales, quienes aumentaron en términos de uso. Además, se evidenció una mejora en las valoraciones en las tiendas de aplicaciones. Internamente

la compañía también pudo validar que el nuevo proceso y arquitectura aceleraron los tiempos de puesta en producción, pusieron al usuario al centro del desarrollo a través de la entrega de valor en cada nueva versión de la aplicación y el soporte tecnológico, tanto de front-end como de back-end, permite ahora recibir múltiples usuarios sin sufrir baja disponibilidad o errores no controlados.

Dado los resultados obtenidos en la validación del nuevo proceso de desarrollo y la nueva arquitectura, se concluye que el objetivo general del presente proyecto de tesis fue logrado, proveyendo a los usuarios de la aplicación y a la compañía, un producto de software construido a través de un proceso ágil, tecnología escalable y robusta y finalmente valorado positivamente por sus usuarios.

6.2 Impacto de la solución

El impacto logrado por la implementación de un nuevo proceso de desarrollo es grande dado que al interior de la compañía no había otros equipos que desarrollaran productos de forma ágil, permitiendo liberar versiones continuamente, obteniendo retroalimentación y mejorando la entrega de valor en cada iteración. Es por ello que el implementar este tipo de metodologías fue un ejemplo para otros equipos, quienes han adoptado herramientas similares para el desarrollo de sus productos. Por otro lado, es un cambio significativo en la forma de pensar y hacer las cosas, dado que las necesidades ya no salen solo de la compañía, sino que directamente de los clientes, quienes prácticamente son uno más al interior del equipo de desarrollo.

Respecto a la nueva arquitectura implementada, el impacto también fue importante, considerando que hasta ese minuto no había tecnología basada en la nube que se estuviese ocupando en la compañía, ya que todos los servidores e instancias eran físicas. Hoy gracias al uso de la virtualización de servidores, la compañía está migrando su estrategia de TI a este tipo de arquitectura, desacoplando además sus aplicaciones monolíticas en microservicios y exponiéndolos a través de APIs para que otros sistemas de la compañía puedan conectarse a las mismas fuentes de datos y evitar problemas como los vividos en el pasado por la aplicación móvil.

El cambio de paradigma de tener un código fuente por cada sistema operativo a solo también logro un gran impacto en cuanto a la filosofía del equipo y la compañía, en donde muchos prejuicios apuntaban a que no se iba a contar con una gran integración con el hardware del teléfono o los plugins nativos de Android o iOS, pero finalmente sin problemas se pudo desarrollar una aplicación que siendo en parte web (por estar basada en Angular, utilizar HTML y CSS), pudo lograr usar componentes nativos de Android o iOS como los gestores de archivos nativos de iOS o los servicios de autorización por huella dactilar de Android, los cuales si bien es cierto no fueron integrados en la versión MVP de la aplicación, están siendo considerado para nuevas entregas.

6.3 Lecciones aprendidas

Parte de las lecciones aprendidas tuvieron que ver con el trabajo en Scrum, dado que es una forma distinta de abordar el desarrollo de software.

Hay muchas malas costumbres e ideas mal concebidas sobre lo que es la agilidad [53], y por lo mismo, el adoptar la metodología según la Guía del Scrum y las buenas practicas discutidas en las comunidades, constituyen una instancia importante de aprendizaje para el equipo y la organización.

En términos tecnológicos, el desarrollo de una aplicación móvil tiene cualidades particulares a considerar, como el hardware de los teléfonos, la publicación en tiendas de aplicaciones y el comportamiento de los usuarios, por lo que el obtener retroalimentación constante, revisar métricas de comportamiento y continuamente estudiar el comportamiento de los usuarios de las aplicaciones, es necesario y obligatorio para evolucionar en el desarrollo de este tipo de tecnología.

6.4 Trabajo futuro

Como trabajo a futuro, quedan muchas cosas por incorporar en cuanto al proceso de desarrollo de software, el cual en los meses que tomó el MVP alcanzó un nivel de madurez básico y el equipo llegó a estar en una etapa *Forming* [50] en cuanto a la agilidad. Es necesario seguir aplicando prácticas ágiles dentro del equipo y también ver cómo estas pueden ser extendidas a otros equipos dentro de la compañía. De seguro que con los crecientes desafíos del comercio electrónico para la compañía, será necesario, y así como aumentan los usuarios y crece la aplicación en cada indicador clave definido, se necesitarán nuevas personas y equipos para dar soporte a al desarrollo y operación, por lo que un marco de trabajo de agilidad escalada [54] sería apropiado e interesante de aplicar a esta clase de escenario, en donde podrían confluir múltiples equipos Scrum trabajando sobre un mismo producto o varios que estén relacionados. También, como parte del proceso de desarrollo, un aspecto interesante a considerar en un futuro sería incorporar TDD, una técnica que guía el desarrollo de software primeramente pensando en las pruebas [55].

Respecto a la arquitectura, sería ideal continuar convirtiendo los grandes monolitos en pequeños servicios modulares, que hoy restan flexibilidad a otros canales de la compañía como el sitio web, en donde el alto acoplamiento impide realizar mejoras en corto plazo y el hecho de tocar una pieza de código muchas veces significa un alto riesgo para múltiples funciones. Por otra parte, añadir mayor control de recursos y seguridad sobre las APIs como un API management.

Sobre la aplicación móvil, queda Implementar más funciones centradas en la experiencia del cliente a través del viaje completo (como la asistencia a través de tutoriales o guías que pudiera recibir al realizar un proyecto de transformación de su hogar) e incluir más métricas detalladas del comportamiento de los usuarios, con las cuales tomar decisiones al momento de desarrollar.

Bibliografía

- [1] Initcoms, «Mobile First Design: Why It's Great and Why It Sucks,» [En línea]. Available: <https://designshack.net/articles/css/mobilefirst/>. [Último acceso: Diciembre 2019].
- [2] Ionic, «What is Hybrid App Development?,» [En línea]. Available: <https://ionicframework.com/resources/articles/what-is-hybrid-app-development>. [Último acceso: Diciembre 2019].
- [3] D. Normal y J. Nielsen, «The definition of User Experience (UX),» [En línea]. Available: <https://www.nngroup.com/articles/definition-user-experience/>. [Último acceso: Agosto 2019].
- [4] Kambrica, «Personas (arquetipos): qué son, para qué sirven, y quiénes las usan,» [En línea]. Available: <https://www.kambrica.com/personas-arquetipos-que-son-para-que-sirven-y-quienes-las-usan/>. [Último acceso: Diciembre 2019].
- [5] S. Lanoue, «UI vs. UX: What's the difference between user interface and user experience?,» [En línea]. Available: <https://www.usertesting.com/blog/2016/04/27/ui-vs-ux/>. [Último acceso: Agosto 2019].
- [6] Knowband, «10 years of growth of Mobile App Market,» [En línea]. Available: <https://www.knowband.com/blog/mobile-app/growth-of-mobile-app-market/>. [Último acceso: Diciembre 2019].
- [7] S.-C. Moatti, *Mobilized: An insider's guide to the business and future of connected technology*, Berrett-Koehler Publishers, 2016.
- [8] I. Corporation, «Native, web or hybrid mobile-app development,» [En línea]. Available: <ftp://public.dhe.ibm.com/software/pdf/mobile-enterprise/WSW14182USEN.pdf>. [Último acceso: Agosto 2019].
- [9] Google, «Progressive Web Apps,» [En línea]. Available: <https://developers.google.com/web/progressive-web-apps?hl=en>. [Último acceso: Diciembre

2019].

- [10] Google, «Alibaba.com increases conversions on the mobile web with PWA,» [En línea]. Available: <https://developers.google.com/web/showcase/2016/pdfs/alibaba.pdf>. [Último acceso: Diciembre 2019].
- [11] M. Fowler, «Agile software development,» [En línea]. Available: <https://martinfowler.com/agile.html>. [Último acceso: Agosto 2019].
- [12] K. Schwaber y J. Sutherland, «The Scrum Guide,» [En línea]. Available: <http://www.scrumguides.org/scrum-guide.html>. [Último acceso: Agosto 2019].
- [13] M. Fowler, «The New Methodology,» [En línea]. Available: <https://martinfowler.com/articles/newMethodology.html>. [Último acceso: Agosto 2019].
- [14] Scrum.org, «What is Scrum?,» [En línea]. Available: <https://www.scrum.org/resources/what-is-scrum>. [Último acceso: Diciembre 2019].
- [15] IBM, «Common misconceptions around microservices and APIs,» [En línea]. Available: <https://developer.ibm.com/integration/blog/2017/01/11/common-misconceptions-around-microservices-apis/>. [Último acceso: Diciembre 2019].
- [16] IBM, «Microservices, SOA, and APIs: Friends or enemies?,» [En línea]. Available: https://developer.ibm.com/tutorials/1601_clark-trs/. [Último acceso: Agosto 2019].
- [17] Red Hat, «¿Qué es la integración continua/distribución continua (CI/CD)?,» [En línea]. Available: <https://www.redhat.com/es/topics/devops/what-is-ci-cd>. [Último acceso: Diciembre 2019].
- [18] R. Pichler, «The product vision board,» [En línea]. Available: <https://www.romanpichler.com/blog/the-product-vision-board/>. [Último acceso: Diciembre 2019].
- [19] S. Blank, «The Customer Development Manifesto: Reasons for the Revolution (part 1),» [En línea]. Available: <https://steveblank.com/2009/08/31/the-customer-development-manifesto-reasons-for-the-revolution-part-1/>. [Último acceso: Diciembre 2019].

- [20] E. Woods Eoin y N. Rozanski, «Viewpoints and Views,» de *Software System Architecture*, Addison-Wesley, 2005, pp. 27-37.
- [21] E. Woods Eoin y N. Rozanski, «Architectural Principles,» de *Software System Architecture*, Addison-Wesley, 2005, pp. 100-104.
- [22] Microsoft, «Extract, transform, and load (ETL),» [En línea]. Available: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/etl>. [Último acceso: Diciembre 2019].
- [23] Code Artisan, «Retrospective method KPT,» [En línea]. Available: <http://code-artisan.io/retrospective-method-kpt/>. [Último acceso: Diciembre 2019].
- [24] R. Pichler, «The go product roadmap,» [En línea]. Available: <https://www.romanpichler.com/blog/goal-oriented-agile-product-roadmap/>. [Último acceso: Agosto 2019].
- [25] América Retail, «Ecommerce: Amazon y su producción en Latam,» [En línea]. Available: <https://www.america-retail.com/ecommerce/ecommerce-amazon-y-su-produccion-en-latam/>. [Último acceso: Diciembre 2019].
- [26] LABS, «Why the Nine-Year Rise of AliExpress Means Big Plans for Latin American Ecommerce,» [En línea]. Available: <https://labs.ebanx.com/en/articles/business/why-the-nine-year-rise-of-aliexpress-means-big-plans-for-latin-american-ecommerce/>. [Último acceso: Diciembre 2019].
- [27] Foda-Dafo.com, «Análisis FODA o DAFO,» [En línea]. Available: https://www.analisisfoda.com/#Que_es_el_analisis_FODA. [Último acceso: Diciembre 2019].
- [28] Net Solutions, «Why it's Vital to Release Software to Production Multiple Times a Day,» [En línea]. Available: <https://www.netsolutions.com/insights/why-do-great-product-companies-release-software-to-production-multiple-times-a-day/>. [Último acceso: Diciembre 2019].
- [29] Smart Insights , «Consumer behaviour and retail trends 2018,» [En línea]. Available: <https://www.smartinsights.com/ecommerce/multichannel-retail-strategy/consumer-behaviour-retail-trends/>. [Último acceso: Diciembre 2019].

- [30] Google, «Google Play Console - App Sodimac,» [En línea]. Available: <https://play.google.com/apps/publish/>. [Último acceso: Agosto 2019].
- [31] Apple, «App Store Connect - App Sodimac,» [En línea]. Available: <https://itunesconnect.apple.com>. [Último acceso: Agosto 2019].
- [32] App Annie, «App Annie,» [En línea]. Available: <https://www.appannie.com/apps/>. [Último acceso: Diciembre 2016].
- [33] Google, «Google Analytics - App Sodimac,» [En línea]. Available: <https://analytics.google.com>. [Último acceso: Agosto 2019].
- [34] Google, «Engage more users: 3 tips for optimizing mobile app performance,» [En línea]. Available: <https://www.thinkwithgoogle.com/marketing-resources/experience-design/app-performance/>. [Último acceso: Diciembre 2019].
- [35] Statista, «Mobile phone users worldwide 2015-2020,» [En línea]. Available: <https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/>. [Último acceso: Agosto 2019].
- [36] INE, «Telefonia fija, larga distancia, móvil e internet,» [En línea]. Available: <https://www.ine.cl/estadisticas/economia/transporte-y-comunicaciones/transporte-y-comunicaciones>. [Último acceso: Diciembre 2019].
- [37] App Annie, «The App Economy Forecast: \$6 Trillion in New Value,» 2017.
- [38] BroadBandSearch, «Mobile vs. Desktop usage (Latest 2019 data),» [En línea]. Available: <https://www.broadbandsearch.net/blog/mobile-desktop-internet-usage-statistics>. [Último acceso: Diciembre 2019].
- [39] Google, «AliExpress: E-Commerce on the Go,» [En línea]. Available: <https://www.thinkwithgoogle.com/intl/en-154/insights-inspiration/case-studies/aliexpress-e-commerce-go/>. [Último acceso: Diciembre 2019].
- [40] Harvard Business Review, «Agile at Scale,» [En línea]. Available: <https://hbr.org/2018/05/agile-at-scale>. [Último acceso: Diciembre 2019].

- [41] Spotify, «Spotify engineering culture,» [En línea]. Available: <https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/>. [Último acceso: Agosto 2019].
- [42] JWT, «Introduction to JSON Web Tokens,» [En línea]. Available: <https://jwt.io/introduction/>. [Último acceso: Diciembre 2019].
- [43] Ionic, «Core Concepts, adaptative style,» [En línea]. Available: <https://ionicframework.com/docs/intro/concepts>. [Último acceso: Diciembre 2019].
- [44] Ionic, «Installing Ionic,» [En línea]. Available: <https://ionicframework.com/docs/installation/cli>. [Último acceso: Agosto 2019].
- [45] Mobincube, «What if I decide to publish my application on App Store?,» [En línea]. Available: <https://mobincube.com/help/en/228-What-if-I-decide-to-publish-my-application-on-App-Store/resources/images/publishingProcess.png>. [Último acceso: Agosto 2019].
- [46] Forbes, «Forbes.com,» [En línea]. Available: <https://www.forbes.com/sites/sergeiklebnikov/2019/11/12/alibabas-1111-singles-day-heres-which-brands-profited-the-most/#5ea58e8b4863>. [Último acceso: Diciembre 2019].
- [47] Statista, «Mobile share of Alibaba's gross merchandise volume on Singles Day from 2013 to 2017,» [En línea]. Available: <https://www.statista.com/statistics/364715/alibaba-singles-day-1111-mobile-share/>. [Último acceso: Diciembre 2019].
- [48] Medium, «¿Qué es un user persona, una proto persona y un arquetipo?,» [En línea]. Available: <https://medium.com/@andrewdjandr/qu%C3%A9-es-un-user-persona-una-proto-persona-y-un-arquetipo-c4498c7f8836>. [Último acceso: Diciembre 2019].
- [49] Scrum.org, «Walking Through a Definition of Ready,» [En línea]. Available: <https://www.scrum.org/resources/blog/walking-through-definition-ready>. [Último acceso: Diciembre 2019].
- [50] Scrum Institute, «Scrum Burndown Chart - International Scrum Institute,» [En línea]. Available: https://www.scrum-institute.org/Burndown_Chart.php. [Último acceso: Diciembre 2019].

- [51] B. W. Tuckman, Stages of Small-Group Development Revisited, 1977.
- [52] Mountain goat software, «Planning Poker,» [En línea]. Available: <https://www.mountaingoatsoftware.com/agile/planning-poker>. [Último acceso: Diciembre 2019].
- [53] Scrum Institute, «Scrum Burndown Chart,» [En línea]. Available: https://www.scrum-institute.org/Burndown_Chart.php. [Último acceso: Diciembre 2019].
- [54] Agile Connection, «Top Twelve Myths of Agile Development,» [En línea]. Available: <https://www.agileconnection.com/article/top-twelve-myths-agile-development>. [Último acceso: Diciembre 2019].
- [55] Scrum.org, «Scaling Scrum with Nexus,» [En línea]. Available: <https://www.scrum.org/resources/scaling-scrum>. [Último acceso: Diciembre 2019].
- [56] K. Beck, Test-Driven Development, Addison-Wesley, 2003.
- [57] Criteo, «Global Commerce Review,» [En línea]. Available: http://www.criteo.com/es/wp-content/uploads/sites/8/2018/05/18_GCR_Q1_Report_Spanish_ES-1.pdf. [Último acceso: Agosto 2019].
- [58] B. Wake, «Scrum Manager,» [En línea]. Available: <https://www.scrummanager.net/bok/index.php?title=INVEST>. [Último acceso: Agosto 2019].
- [59] Criteo, «Retail and Travel Apps See 4x More Conversions than Mobile Web,» [En línea]. Available: <https://www.criteo.com/insights/retail-travel-apps-higher-conversions-mobile/>. [Último acceso: Agosto 2019].
- [60] M. Fowler, «Early Pain,» [En línea]. Available: <https://martinfowler.com/bliki/EarlyPain.html>. [Último acceso: Agosto 2019].
- [61] M. Fowler, «Microservices Guide,» [En línea]. Available: <https://martinfowler.com/microservices/>. [Último acceso: Agosto 2019].

[62] S. Moatti, *Mobilized*, 2016.

[63] Harvard Business Review, «The Rise of the Mobile-Only User,» [En línea]. Available: <https://hbr.org/2013/05/the-rise-of-the-mobile-only-us>. [Último acceso: Diciembre 2019].

[64] AliExpress, «AliExpress: E-Commerce on the Go,» [En línea]. Available: <https://www.youtube.com/watch?v=c8sfVr5Rysw>. [Último acceso: Diciembre 2019].

[65] Scrum.org, «What is a Sprint Retrospective?,» [En línea]. Available: <https://www.scrum.org/resources/what-is-a-sprint-retrospective>. [Último acceso: Diciembre 2019].

[66] E. Woods y N. Rozanski, «Scope, Concerns, Principles and Constraints,» de *Software System Architecture*, Addison-Wesley, 2005, pp. 105-106.

Anexos

En el presente capítulo se detallan elementos como contexto, ejemplos, protocolos o las herramientas que permitieron desarrollar la solución y validación del proyecto de tesis.

Anexo A - Solución

Parte de la solución fue desarrollar una estrategia de negocio que guiara el desarrollo de la aplicación móvil, para ello se revisó el estado interno y externo a la compañía, ambos sirvieron de base para el desarrollo de una estrategia propia. Otros elementos que ayudaron a definir la solución se organizaron primeramente en reuniones con los stakeholders. Cada una de esas reuniones están presentes en el presente capítulo mostrando el protocolo de su realización. Finalmente se incluye un ejemplo de la estructura de la API Catálogo de productos.

Base para una estrategia: estado del comercio electrónico móvil global

En el reciente informe *Global Commerce Review Q1 2018* [56], entregado por la consultora de marketing digital Criteo, aparecen indicadores interesantes respecto a la participación en transacciones de las aplicaciones móviles versus otros canales digitales como sitios web en su versión escritorio y versión móvil. En la Figura 66, se pueden ver dos gráficos por cada continente, el del lado izquierdo está basado en compañías que enfocan su estrategia comercial solo en sitios web de escritorio y versión móvil y en la derecha aquellos comercios que además tienen aplicaciones móviles. Cuando se usa esta última estrategia, la participación de las aplicaciones móviles es superior al 50% en la mayoría de los casos (en color amarillo), pero ocurre que, en escenarios como Latinoamérica, la participación de las aplicaciones móviles es menor al resto. Esto puede estar dado por muchos factores como la experiencia de las compañías locales en el desarrollo del comercio electrónico, el acceso masivo a la tecnología y medios de pago, seguridad y confianza en las transacciones online o experiencia en el desarrollo de canales logísticos no tradicionales. Un ejemplo de esto último es que no hace mucho tiempo, las grandes compañías nacionales de comercio minorista y electrónico no contaban con despacho en 24 horas, de hecho, hoy en Chile existe este tipo de despacho para la mayoría de los comercios solo en la Región Metropolitana, algo casi impensado en el panorama internacional).

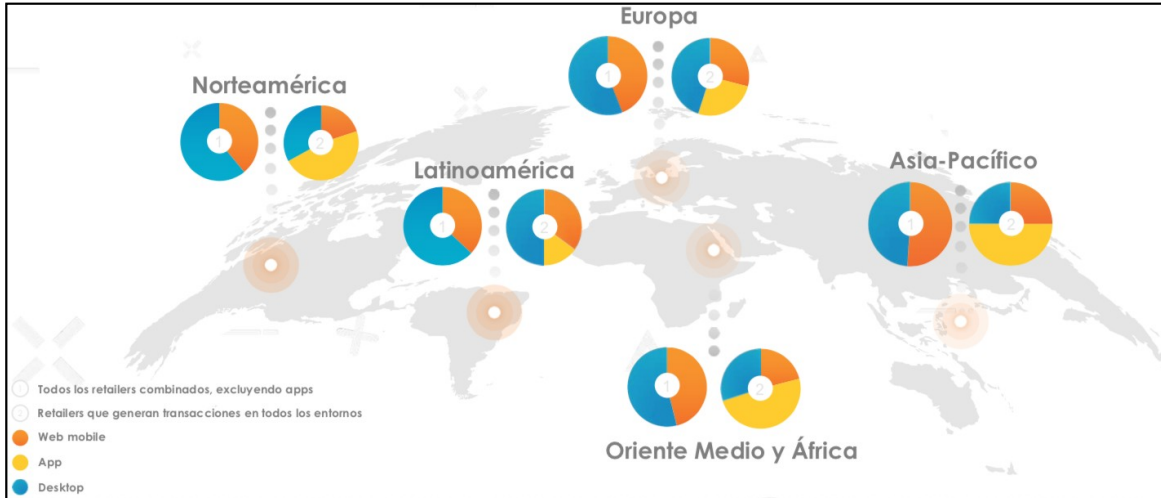


Figura 66 - La oportunidad de las aplicaciones móviles: share de transacciones [56]

Otro aspecto interesante revelado por el estudio realizado por Criteo es la tasa de conversión (CR, del inglés *Conversion Rate*) que logran las aplicaciones móviles frente a otros canales digitales como sitios web en su versión escritorio o móvil. La tasa de conversión por definición corresponde al porcentaje de usuarios que realizan una acción específica como una compra, descarga, registro de usuario o una reserva.

En el estudio realizado por esta compañía europea se evaluó la compra de productos en comercios minoristas que poseían una aplicación móvil y sitio web en versión escritorio y móvil, donde la tasa de conversión está dada por:

$$CR = \left(\frac{\text{Ventas realizadas}}{\text{Total visitas}} \right) \times 100$$

Tal como lo expresa la *Figura 67*, el resultado del estudio fue que la tasa de conversión en las aplicaciones móviles de comercios minoristas es más de 3 veces superior a la de la web móvil. Este resultado puede tener varias razones asociadas, algunas radican en que el usuario de aplicaciones móviles posee un grado de fidelización superior frente a la marca versus un usuario de sitio móvil. Esto viene dado por las características esenciales de las aplicaciones móviles, como que son un tipo de software que permanece las 24 horas del día prácticamente en el bolsillo de los usuarios, estableciendo un nexo de dependencia que dista de otros formatos como los sitios web, en donde el usuario accede bajo un contexto particular y de forma unidireccional (es el usuario quien decide acceder a la información, escribiendo la URL en el navegador y luego navegando en el sitio web). En

las aplicaciones móviles, también existen notificaciones push, basadas en inteligencia artificial y *Machine Learning* que permiten segmentar a los usuarios y entregarles contenido basado en el valor particular de quien es el receptor. Eso hace que los usuarios se sientan comprendidos en contexto y se sientan identificados con una aplicación.

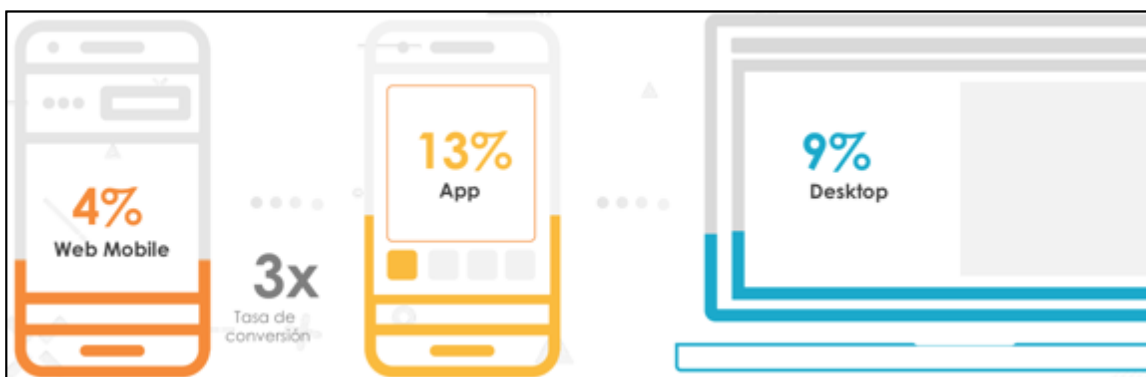


Figura 67 - La oportunidad de las aplicaciones móviles: conversión [56]

Al mirar la realidad internacional de mercados como Europa, Asia o Estados Unidos en donde las compras realizadas por aplicaciones móviles son líderes, hace pensar que este escenario está lejos de ocurrir en Latinoamérica, pero puede ser engañoso pensar de aquella forma, dado que hoy las redes sociales, el acceso a Internet, las mejoras en los procesos logísticos internacionales y leyes de colaboración comercial entre países, han hecho que comercios internacionales como Amazon o Aliaba no sean realidades lejanas, sino aplicaciones que funcionan localmente tal como cualquier comercio nacional. Hoy un comercio minorista local cuenta con plazos de despacho de 2 o 3 semanas durante eventos online como Cyber Day o Black Friday, versus los comercios internacionales los cuales muchos cuentan ya con despacho internacional directo a Chile y que según un monto determinado de dinero pueden entregar un producto en menos de una semana.

La competencia del comercio electrónico minorista de las grandes compañías no solo se está librando en los mercados internacionales, ya hay presencia en Chile. Un ejemplo claro fue la disposición de 50 millones de productos con envío directo y pago en moneda local para la navidad de 2018 por parte de Amazon, por lo que las compañías locales deben al menos ponerse al día en términos tecnológicos, incorporar las buenas prácticas de la industria en los procesos de desarrollos de productos e implementar estrategias de negocio competitivas.

Definición de la estrategia de negocio móvil

La Tabla 21 presenta el protocolo utilizado para la definición de la estrategia móvil de la compañía.

Definición de la estrategia móvil	
Objetivo	Considerando la situación actual de la aplicación móvil y el contexto externo, se define la meta a donde la compañía quiere posicionarse en el futuro próximo, generando un PVB.
Procedimiento	<p>Asistentes: equipo de desarrollo de la aplicación móvil, stakeholders de negocio y TI.</p> <p>Duración: 4 horas.</p> <p>Equipo y material: pizarra acrílica para explicar y discutir temas, y tarjetas adhesivas.</p> <p>Dinámica: una persona designada tomará notas sobre los acuerdos generados. La forma de abordar la reunión es presentar el objetivo de la reunión y luego presentar brevemente el significado de la herramienta PVB.</p>
Instrumento	<p>Exposición de temas (situación actual y contexto externo) y luego ronda de discusión:</p> <ol style="list-style-type: none">1. Revisión de la situación actual de la compañía: tecnología, proceso de desarrollo y posicionamiento comercial del producto.2. Contextualización del estado del arte del desarrollo de productos móviles. Revisión de cifras, buenas prácticas en tecnología, procesos de desarrollo y finalmente revisión de casos de éxito.3. ¿Qué es un PVB?, descripción de la herramienta, su objetivo, secciones y ejemplos.4. Desarrollo del PVB entre todos los asistentes mediante el uso de la pizarra y tarjetas adhesivas. Cada persona describe una idea que luego es comentada por un moderador y entre todos se acuerdan los puntos de cada sección del PVB.

Tabla 21 - Protocolo para la definición de la estrategia móvil

Definición de objetivos de negocio y motivadores

A continuación, la Tabla 22 describe el protocolo utilizado para la definición de los objetivos de negocio y motivadores, en donde cada pregunta resultó en una respuesta que era desarrollada un par de minutos y de la cual se extraía información para luego conformar una propuesta de objetivos y motivadores detectados. Finalmente se votó por las 6 más importantes según los stakeholders presentes.

Definición de objetivos de negocio y motivadores	
Objetivo	Rescatar y definir los objetivos del negocio y motivadores para la definición arquitectónica de la aplicación móvil.
Procedimiento	<p>Asistentes: encargado de negocio de la aplicación móvil y stakeholders de las áreas de venta online, marketing y operaciones.</p> <p>Duración: 2 horas.</p> <p>Equipo y material: grabadora y pauta de entrevista.</p> <p>Dinámica: se solicita a los asistentes que comenten su visión de negocio para la aplicación móvil con miras al presente año y también para el futuro (próximos 3 años).</p>
Instrumento	<p>La entrevista se hace bajo la siguiente pauta de preguntas:</p> <ol style="list-style-type: none"> 1. ¿Cuál considera usted que es el objetivo principal de la aplicación móvil? 2. ¿Cómo ve el panorama mundial y externo a la organización en cuanto a aplicaciones móviles? 3. ¿Cómo ve posicionadas en el ecosistema digital latinoamericano a las aplicaciones móviles de la compañía de aquí a diciembre del presente año y para finales de 2020? 4. En base a la pregunta anterior, ¿podría pensar en algún objetivo, meta o indicador concreto a lograr para finales de este año y finales de 2020?

Tabla 22 - Protocolo para la definición de objetivos de negocio y motivadores

Descubrimiento de preocupaciones arquitectónicas

Junto a los stakeholders se realizó una reunión para descubrir sus preocupaciones, las que podrían tener un fuerte impacto en las decisiones del diseño arquitectónico del nuevo sistema. La Tabla 23 presenta el protocolo para el descubrimiento de preocupaciones arquitectónicas.

Descubrimiento de preocupaciones arquitectónicas	
Objetivo	Explorar las preocupaciones de los stakeholders sobre la arquitectura de la aplicación móvil para luego llevarlas a objetivos y requerimientos cuantificables y medibles de arquitectura.
Procedimiento	<p>Asistentes: product owner, scrum master y stakeholders provenientes de las siguientes áreas: venta online, gerencia comercial, marketing, operaciones, TI, finanzas y legal.</p> <p>Duración: 4 horas.</p> <p>Equipo y material: pizarra y tarjetas adhesivas.</p> <p>Dinámica: considerando los objetivos de negocio y motivadores, se realizan preguntas a los entrevistados para definir las preocupaciones.</p>
Instrumento	<p>Se realiza la siguiente pregunta: considerando el nuevo sistema tecnológico de la aplicación móvil, ¿cuál considera usted que debiera ser un requerimiento, objetivo, intención o aspiración de este sistema?, nombre al menos tres.</p> <p>Luego, cada respuesta es escrita en una tarjeta adhesiva que es fijada en una pizarra.</p> <p>Así sucede con cada asistente, hasta la reunión grupal en donde se leen todas las respuestas, donde se agrupan aquellas similares, se refinan las que no son del todo claras para todos los asistentes y luego se realiza una votación por las seis más importantes.</p>

Tabla 23 - Protocolo para el descubrimiento de preocupaciones arquitectónicas

La Tabla 24 a continuación aborda la preocupación P1, resultado del descubrimiento de las preocupaciones arquitectónicas.

P1. La aplicación debe ser rápida y fácil de usar
Requerimiento de arquitectura
La respuesta de los servicios sobre productos desde los sistemas legados debe entregar un conjunto de 10 ítems para no saturarlos de peticiones.
La aplicación debe desplegar el contenido de forma asíncrona.
Objetivo arquitectónico
Incluir respuestas de fácil comprensión para el usuario frente a errores o excepciones como: sin conexión a Internet, no hay resultados para la búsqueda de productos, sin acceso a la señal GPS o no se obtuvo respuesta del servicio.
El uso de cualquier función no debe contemplar más de 3 pantallas para ser completada.
El tiempo de carga y despliegue de información en pantalla para los usuarios debe ser menor o igual a 3 segundos.
Las aplicaciones deben ser livianas, no pesar más allá de 15 Mb. para que su descarga sea rápida.

Tabla 24 - Principio arquitectónico P1

La Tabla 25 a continuación aborda P2.

P2. La información en la aplicación debe ser la misma que aparece en el sitio web o las tiendas
Requerimiento de arquitectura
Se deben desarrollar microservicios que se comuniquen con los sistemas legados de la compañía y luego exponerlos a través de interfaces hacia la aplicación móvil.
Se debe implementar una interfaz que exponga los mismos microservicios que utiliza el sitio web para el proceso transaccional, servicios de: consulta disponibilidad, reserva, pago e inscripción de orden de compra.
Objetivo arquitectónico
La sincronización de información de productos entre la aplicación móvil, sitios web y tiendas físicas debe ser en línea para datos sensibles como precios y disponibilidad de productos.
Los mismos atributos exhibidos en el sitio web y tiendas físicas sobre un producto deben estar disponibles en la aplicación móvil: fotos, marca, nombre producto, precio normal, precio promocional, descripción, ficha técnica y código SKU.
El proceso de compra tanto en las aplicaciones como en los sitios web de la compañía, deben tener los mismos 3 pasos: 1. agregar un producto al carro de compras, 2. seleccionar el método de entrega y 3. ejecutar el pago.
El árbol de navegación (categorías de productos) debe ser el mismo en las aplicaciones y sitios web.
Las aplicaciones deben utilizar el mismo manual de marca que refiere a los lineamientos gráficos a utilizar en el diseño de pantallas de sistemas digitales de la compañía.

Tabla 25 - Principio arquitectónico P2

La Tabla 26 a continuación aborda P3.

P3. La aplicación móvil debe ser estable y continuar operando frente a la alta demanda
Requerimiento de arquitectura
Como mínimo las aplicaciones deben permitir 100.000 usuarios concurrentes consultando información o realizando transacciones.
La arquitectura debe estar basada en un sistema elástico de instancias, que permitan disponibilidad recursos automáticamente en caso de ser necesario
Todos los servicios y microservicios que consumen las aplicaciones deben ser REST y basados en cifrado HTTPS.
Objetivo arquitectónico
Para la compra de productos y servicios, el usuario debe registrarse e identificarse con sus datos personales básicos de facturación según la ley en Chile, esto es: nombre completo, Rut, teléfono, dirección y forma de pago.

Tabla 26 - Principio arquitectónico P3

Estableciendo trazabilidad

En la Tabla 27 se establece el protocolo para definir los principios de negocio y tecnológicos que darán forma a la nueva arquitectura.

Estableciendo trazabilidad	
Objetivo	Basados en los motivadores y objetivos de negocio, definir los principios de negocio y principios tecnológicos que darán forma a la nueva arquitectura de la aplicación móvil.
Procedimiento	<p>Asistentes: equipo de desarrollo de la aplicación móvil y stakeholders.</p> <p>Duración: 1 hora para definir los principios de negocio y 1 hora para los principios tecnológicos.</p> <p>Equipo y material: pizarra y tarjetas adhesivas.</p> <p>Dinámica: considerando los objetivos de negocio y motivadores, se establecen los principios de negocio y los principios tecnológicos. Cada definición nace como consecuencia de la anterior.</p>
Instrumento	<p>Se analizan los objetivos de negocio y motivadores, a partir de ellos cada asistente escribe en una tarjeta un principio de negocio asociado. Luego, se leen en voz alta cada principio, se agrupan los que son similares y se discute sobre ellos definiendo una cantidad determinada de principios.</p> <p>Considerando ahora los principios de negocio, cada asistente escribe en una tarjeta un principio tecnológico que responde a uno o varios principios de negocio. El proceso transcurre tal como con la definición anterior.</p>

Tabla 27 - Estableciendo trazabilidad

Ejemplo de la estructura de la API Catálogo de productos

Esta API consolida la información relacionada a productos proveniente de los sistemas de información de productos I y II (SL1 y SL2 en la Figura 36) además del sistema de fotos de productos SL3.

Esta API está disponible en dos ambientes (considerar que son URLs ficticias dada la confidencialidad de la información para la compañía):

A1. Pruebas: <https://api-catalogo-test.foo.com>

A2. Producción: <https://api-catalogo.foo.com>

La API está constituida por los siguientes controladores: C1 Categorías, C2 Países, C3 Productos y C4 Tiendas.

Dentro del controlador C3 Productos se encuentran distintos métodos que exponen los datos provenientes de los sistemas de back-end de la compañía (SL1, SL2 y SL3). A continuación, en la Tabla 28 un ejemplo:

GET – Productos_Disponibilidades	
Descripción	Este método retorna la disponibilidad de entrega de uno o varios productos según su tipo: despacho a domicilio o retiro en tienda. Esta información proviene del microservicio Product de SL2: Sistema de información de productos II y es expuesto como directamente hacia la API Catálogo de productos.
URL (producción)	https://api-catalogo.foo.com/Productos/{pais}/{tienda}/{skus}/Disponibilidad
Parámetros de petición	<ul style="list-style-type: none"> País: código de país, ejemplo: CL, PE, AR, UR, BR, CO o MX. Tienda: código interno de tienda, ejemplo: 72. SKU: código identificador único de producto a consultar, ejemplo: 6300.
Respuesta	<pre>{ sku: 6300, country: CL, store: 72, homeDelivery: TRUE, // valor boolean para la disponibilidad de despacho a domicilio storePickup: TRUE // disponibilidad de retiro en tienda</pre>

Tabla 28 - Método de disponibilidad de productos en la API catálogo de producto

Anexo B - Validación

A continuación, se detallan algunos elementos como el backlog de MVP, un ejemplo de historia de usuario y diseño de una función de la aplicación móvil y finalmente las encuestas de validación por percepción de la nueva arquitectura y proceso.

Backlog del MVP

En la Figura 68 se muestra el backlog del MVP con las historias de usuario agrupadas por épicas.

Item	Epica	MVP ID	Acciones
Splash	Configuración inicial	MVP-14	↑ -
Tutorial Inicial	Configuración inicial	MVP-15	↑ -
Permisos de acceso a ubicación y cámara	Configuración inicial	MVP-56	↑ -
Selector de tienda favorita	Configuración inicial	MVP-30	↑ -
Carrusel de banners	Home	MVP-11	↑ -
Categorías destacadas	Home	MVP-12	↑ -
Banners simples	Home	MVP-13	↑ -
Menú principal	Home	MVP-16	↑ -
Tab bar	Home	MVP-17	↑ -
Input de buscador	Home	MVP-18	↑ -
Productos destacados	Home	MVP-19	↑ -
Header	Home	MVP-20	↑ -
Buscador por escáner	Búsqueda y naveg...	MVP-32	↑ -
Buscador por texto	Búsqueda y naveg...	MVP-31	↑ -
Página de resultados de búsqueda por texto - SRP	Búsqueda y naveg...	MVP-33	↑ -
Página de categorías	Búsqueda y naveg...	MVP-35	↑ -
Página de resultados de navegación por categorías - PLP	Búsqueda y naveg...	MVP-55	↑ -
Agregar a favoritos	Página de producto - ...	MVP-47	↑ -
Carrusel de imágenes	Página de producto - ...	MVP-48	↑ -
Modulo de precios	Página de producto - ...	MVP-49	↑ -
Modulo de disponibilidad de entrega	Página de producto - ...	MVP-50	↑ -
Agregar al carro	Página de producto - ...	MVP-51	↑ -
Ficha técnica	Página de producto - ...	MVP-52	↑ -
Descripción emocional	Página de producto - ...	MVP-53	↑ -
Productos sugeridos	Página de producto - ...	MVP-54	↑ -
Login	Cuenta de usuario	MVP-37	↑ -
Registro	Cuenta de usuario	MVP-38	↑ -
Recuperar contraseña	Cuenta de usuario	MVP-39	↑ -
Seguimiento de compras	Cuenta de usuario	MVP-41	↑ -
Carro de compras	Checkout	MVP-40	↑ -
Despacho	Checkout	MVP-42	↑ -
Pago	Checkout	MVP-43	↑ -
Confirmación de compra	Checkout	MVP-44	↑ -
Marcar home	Métricas y analítica	MVP-57	↑ -
Marcar PDP	Métricas y analítica	MVP-58	↑ -
Marcar PLP	Métricas y analítica	MVP-59	↑ -
Marcar carro de compras	Métricas y analítica	MVP-60	↑ -
Marcar favoritos	Métricas y analítica	MVP-61	↑ -
Marcar despacho	Métricas y analítica	MVP-62	↑ -
Marcar pago	Métricas y analítica	MVP-63	↑ -
Marcar confirmación de compra	Métricas y analítica	MVP-64	↑ -
Marcar buscador por texto	Métricas y analítica	MVP-65	↑ -
Marcar escaner	Métricas y analítica	MVP-66	↑ -

Figura 68 - Backlog de la aplicación móvil

Diseño de la PDP

En la Figura 69 se muestra la visualización de front-end de la ficha de productos del MVP.

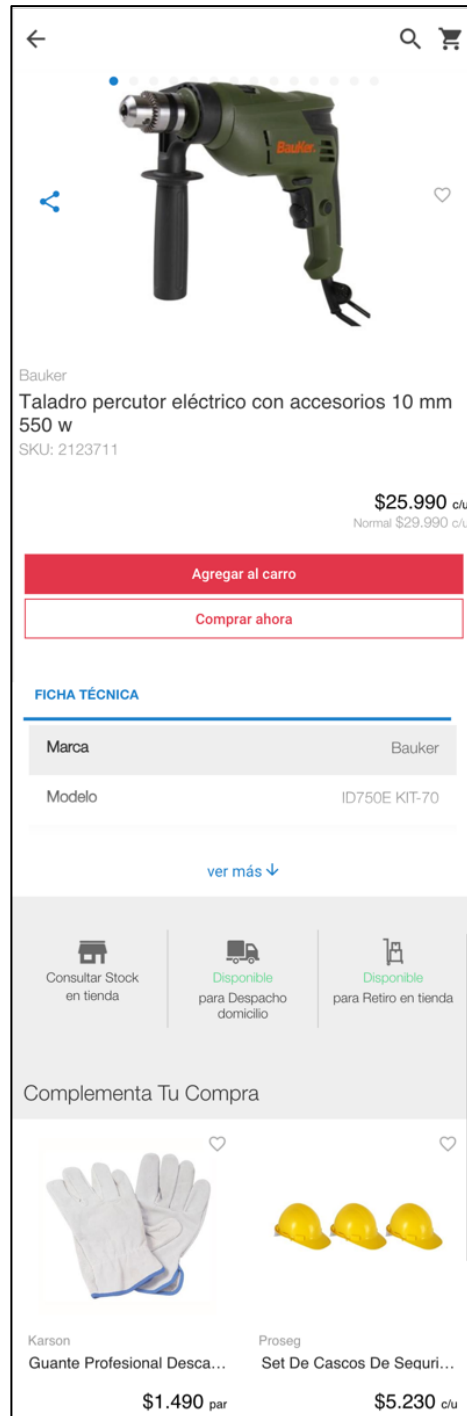


Figura 69 - Vista de front-end de la página de detalle de producto

Ejemplo de historia de usuario

En la Figura 70 se muestra un ejemplo de una historia de usuario, en este caso, el diseño de la ficha del producto (PDP) del MVP desarrollado.

MVP-85

[UX/UI] Diseñar PDP

Editar Comentar Asignar Por hacer En curso Listo Administración

Tipo: Historia Estado: POR HACER
Prioridad: Medium Resolución: (Ver flujo de trabajo)
Etiquetas: UX/UI Sin resolver
Sprint: Tablero Sprint 4

Descripción

COMO usuario de la aplicación móvil
QUIERO un diseño de ficha de producto que sea intuitivo y sencillo
PARA poder comprender la información de los productos y sentirme informado sin dudas

Notas

Las secciones de la ficha del producto son:

1. Imágenes del producto
2. Marca
3. Nombre
4. Código SKU
5. Precios
6. Ficha técnica
7. Disponibilidad de entrega
8. Productos complementarios

Criterios de aceptación

CA1:
DADO un diseño de PDP
CUANDO es visualizado por un usuario de la aplicación
ENTONCES este posee todas las secciones descritas en las notas

CA2:
DADO un diseño de PDP
CUANDO es visualizado por un usuario de la aplicación
ENTONCES este es comprendido y aceptado por el

Adjuntos

Suelta los archivos para adjuntarlos o [explorar](#).

Subtareas

1. [Hacer maqueta de PDP](#) POR HA... UX/UI
2. [Validar maqueta con usuarios](#) POR HA... UX/UI
3. [Diseñar pantallas en Invisión](#) POR HA... UX/UI
4. [Pasar diseño a HTML](#) POR HA... UX/UI

Figura 70 - Historia de usuario MVP-85: Diseñar PDP

Encuesta A, personas transversales

ET1. Evaluación de la situación: proceso de desarrollo.

A mantener (*Keep*)

A continuación se presentan los ítems positivos del antiguo proceso que se mantuvieron en el proceso al nuevo, y que fueron evaluados por stakeholders *transversales* y todos los stakeholders y miembros del equipo de desarrollo.

	Mucho mejor	Mejor	Ni mejor ni peor	Peor	Mucho peor	No sé
En términos de facilidad de adopción del proceso, el nuevo proceso es:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
En términos de documentación reducida, el proceso nuevo es:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
En términos de simplicidad y brevedad en etapas, el nuevo proceso es:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
En términos de autogestión, el nuevo proceso es:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Problemas (*Problem*)

A continuación se listan los ítems negativos que representaban un problema en el antiguo proceso y que se quitaron en el nuevo, y que fueron evaluados por stakeholders *transversales* y todos los stakeholders y miembros del equipo de desarrollo.

	Mucho mejor	Mejor	Ni mejor ni peor	Peor	Mucho peor	No sé
En términos de transparencia hacia los stakeholders, el nuevo proceso es:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
En cuanto a la priorización del nuevo proceso centrada en el usuario final, esta es:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
En términos de aceptación del cambio en cuanto a requerimientos, el nuevo proceso es:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Respecto a la puesta en producción de entregables en plazos breves (máx. 1 mes, min. 2 semanas) para usuarios finales, el nuevo proceso es:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Respecto a la definición de roles, bajo el nuevo proceso esto es:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Respecto a la resistencia al cambio del equipo, en el nuevo proceso:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

A intentar (*Try*)

A continuación se listan los ítems que en base a los problemas del antiguo proceso se podrían intentar en el nuevo proceso para su mejora, evaluados por stakeholders *transversales* y solo miembros del equipo de desarrollo.

	No considerado	Poco considerado	Neutral	Considerado	Mucho peor	No sé
Dar visibilidad de avance a los stakeholders, está:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usar método formal para la estimación de esfuerzo de desarrollo en el proceso:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fallar rápido y barato frente a la apuesta de hipótesis en términos de funciones para los usuarios finales.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Entregas constantes y en breve tiempo (mínimo 2 semanas, máximo 1 mes) a usuarios finales.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Utilizar un enfoque de desarrollo adaptativo de software que acepte el cambio	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Priorización de requerimientos sobre las necesidades de los usuarios y clientes.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Conocer a los usuarios y clientes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(arquetipos de personas).

Roles y equipos claros y definidos.

Dar seguimiento a las iteraciones de desarrollo.

Incluir a los stakeholders en el desarrollo del producto (aplicación móvil).

ET2. Evaluación de la situación: arquitectura

A mantener (*Keep*)

Sobre los items positivos de la antigua arquitectura y que son válidos de mantener en la nueva arquitectura. Grupo encuestado: Transversales y solo miembros del equipo de desarrollo.

	Mucho mejor	Mejor	Ni mejor ni peor	Peor	Mucho peor	No sé
La nueva arquitectura cuenta con una capa intermedia que permita ser resiliente frente a las caídas de los sistemas legados para datos no sensibles a actualizar en tiempo real.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Respecto a la baja complejidad de las funciones de las	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

aplicaciones, estas son:

En cuanto al conocimiento detallado de cada sistema perteneciente a la arquitectura de la aplicación móvil, este es:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

Problemas (*Problem*)

Sobre los ítems negativos de la antigua arquitectura que no se incluyeron en la nueva arquitectura.
Grupo encuestado: Transversales y solo miembros del equipo de desarrollo.

	Mucho mejor	Mejor	Ni mejor ni peor	Peor	Mucho peor	No sé
--	-------------	-------	------------------	------	------------	-------

Respecto a la frecuencia de actualización de datos, bajo la nueva arquitectura esta es:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

En cuanto a la alta disponibilidad, la nueva arquitectura es:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

En cuanto a la edad y actualización del lenguaje de programación, en la nueva arquitectura es:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

Considerando en la antigua arquitectura el uso de PCs de escritorio regulares usados como servidor, en la nueva arquitectura este tipo de uso es:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

En cuanto a la rapidez de ejecución de los pasos a producción, en la nueva arquitectura esto es:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

A intentar (*Try*)

En base a los problemas de la antigua arquitectura, ítems que pueden ser probados en la nueva arquitectura para evidenciar una mejora en ella. Grupo encuestado: Transversales y solo miembros del equipo de desarrollo.

	No considerado	Poco considerado	Neutral	Considerado	Mucho peor	No sé
Desarrollar Microservicios y funciones descentralizadas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar una arquitectura de alta disponibilidad.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Desarrollar solo un código fuente para ambos sistemas operativos.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar cargas por umbrales para los datos no necesarios en tiempo real desde los sistemas legados.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar pruebas automatizadas en las capas de front-end y backend.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Integrar bases de datos NoSQL para la carga de altos volúmenes de datos y gestión de estos.

Implementar canales de CI/CD en las capas de front-end y back-end.

ET3. Logro de los objetivos

Respecto a los objetivos planteados al comienzo del proyecto que plantea un nuevo proceso y arquitectura. Grupo encuestado: Transversales, todos los miembros del equipo y stakeholders.

Muy en desacuerdo En desacuerdo Ni de acuerdo ni en desacuerdo De acuerdo Muy de acuerdo

Con este nuevo proceso de desarrollo se logra poner en producción versiones con mayor frecuencia para los usuarios finales.

Con esta nueva arquitectura se logra la escalabilidad de recursos y coherencia en la información.

La aplicación móvil entrega valor y significado a sus usuarios, manifestado esto en el uso y ranking en las tiendas de aplicaciones.

ET7. Adopción del proceso de desarrollo y arquitectura

Grupo encuestado: Transversales, todos los miembros el del equipo y stakeholders.

En una escala de 1 a 7, donde 1 es muy malo o difícil y 7 es muy bueno o sencillo, cómo evalúa la adopción del nuevo proceso de desarrollo y nueva arquitectura en términos de:

	Su rol	Para la compañía
Proceso de desarrollo	<input type="text"/>	<input type="text"/>
Arquitectura móvil	<input type="text"/>	<input type="text"/>

Encuesta B, personas nuevas

EN1. Evaluación de la situación: proceso de desarrollo.

A mantener (*Keep*)

Sobre los ítems positivos del antiguo proceso que se mantuvieron en el proceso al nuevo. Grupo encuestado: nuevos, todos los stakeholders y miembros del equipo de desarrollo.

	Muy en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Muy de acuerdo	No sé
El proceso es de fácil adopción para los miembros del equipo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
El proceso requiere de poca documentación.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
En cuanto a las etapas y fases del proceso: es simple, posee pocas y breves etapas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
El proceso permite la autogestión al interior del equipo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Problemas (*Problem*)

Sobre los ítems negativos que representaban un problema en el antiguo proceso y que se quitaron en el nuevo. Grupo encuestado: nuevos, todos los stakeholders y miembros del equipo de desarrollo.

	Muy en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Muy de acuerdo	No sé
El proceso permite transparentar el avance del desarrollo a los stakeholders.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
La estimación del esfuerzo de desarrollo para la planificación de funciones es acertada.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
La priorización de las tareas se realiza en base a las necesidades del cliente.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Es posible modificar los requerimientos a lo largo del tiempo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Se realizan entregables a usuarios finales antes de contar con la versión final de una función o producto.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Existen roles definidos en el proceso.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Respecto al cambio en cuanto a requerimientos, tecnología o mercado, ¿existe resistencia para afrontarlo a nivel de proceso dentro del equipo?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

A intentar (*Try*)

Sobre los ítems que en base a los problemas del antiguo proceso se podrían intentar en el nuevo proceso para su mejora. Grupo encuestado: nuevos, todos los stakeholders y miembros del equipo de desarrollo.

	Muy en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Muy de acuerdo	No sé
Dar visibilidad de avance a los stakeholders, está:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usar método formal para la estimación de esfuerzo de desarrollo en el proceso:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fallar rápido y barato frente a la apuesta de hipótesis en términos de funciones para los usuarios finales.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Entregas constantes y en breve tiempo (mínimo 2 semanas, máximo 1 mes) a usuarios finales.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Utilizar un enfoque de desarrollo adaptativo de software que acepte el cambio.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Priorización de requerimientos sobre las necesidades de los usuarios y	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

clientes.

Conocer a los usuarios y clientes (arquetipos de personas).

Roles y equipos claros y definidos.

Dar seguimiento a las iteraciones de desarrollo.

Incluir a los stakeholders en el desarrollo del producto (aplicación móvil).

EN2. Evaluación de la situación: arquitectura

A mantener (*Keep*)

Sobre los ítems positivos de la antigua arquitectura y que son válidos de mantener dentro de la nueva arquitectura. Grupo encuestado: nuevos y solo miembros del equipo de desarrollo.

	Mucho mejor	Mejor	Ni mejor ni peor	Peor	Mucho peor	No aplica
La nueva arquitectura cuenta con una capa intermedia que permite ser resiliente frente a las caídas de los sistemas legados para datos no sensibles a actualizar en tiempo real.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Respecto a la baja complejidad de las funciones de las aplicaciones, estas son:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

En cuanto al conocimiento detallado de cada sistema perteneciente a la arquitectura de la aplicación móvil, este es:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

Problemas (*Problem*)

Sobre los ítems negativos de la antigua arquitectura que no se incluyeron en la nueva arquitectura.
Grupo encuestado: nuevos y solo miembros del equipo de desarrollo.

Mucho mejor	Mejor	Ni mejor ni peor	Peor	Mucho peor	No aplica
-------------	-------	------------------	------	------------	-----------

Respecto a la frecuencia de actualización de datos, bajo la nueva arquitectura esta es:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

En cuanto a la alta disponibilidad, la nueva arquitectura es:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

En cuanto a la edad y actualización del lenguaje de programación, en la nueva arquitectura es:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

Considerando en la antigua arquitectura el uso de PCs de

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

escritorio regulares usados como servidor, en la nueva arquitectura este tipo de uso es:

En cuanto a la rapidez de ejecución de los pasos a producción, en la nueva arquitectura esto es:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

A intentar (*Try*)

En base a los problemas de la antigua arquitectura, ítems que pueden ser probados en la nueva arquitectura para evidenciar una mejora en ella. Grupo encuestado: nuevos y solo miembros del equipo de desarrollo.

	Mucho mejor	Mejor	Ni mejor ni peor	Peor	Mucho peor	No aplica
Desarrollar Microservicios y funciones descentralizadas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar una arquitectura de alta disponibilidad.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Desarrollar solo un código fuente para ambos sistemas operativos.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar cargas por umbrales para los datos no necesarios en tiempo real	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

desde los sistemas legados

Implementar pruebas automatizadas en las capas de front-end y back-end

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

Integrar bases de datos NoSQL para la carga de altos volúmenes de datos y gestión de estos.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

Implementar canales de CI/CD en las capas de front-end y back-end-

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

EN3. Logro de los objetivos

Respecto a los objetivos planteados al comienzo del proyecto que plantea un nuevo proceso y arquitectura. Grupo encuestado: Nuevos, todos los miembros el del equipo y stakeholders.

Muy en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Muy de acuerdo
-------------------	---------------	--------------------------------	------------	----------------

Con este nuevo proceso de desarrollo se logra poner en producción versiones con mayor frecuencia para los usuarios finales.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

Con esta nueva arquitectura se logra la escalabilidad de recursos

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

y coherencia en la información.

La aplicación móvil entrega valor y significado a sus usuarios, manifestado esto en el uso y ranking en las tiendas de aplicaciones.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

EN7. Adopción del proceso de desarrollo y arquitectura

Grupo encuestado: nuevos, todos los miembros el del equipo y stakeholders.

En una escala de 1 a 7, donde 1 es muy malo o difícil y 7 es muy bueno o sencillo, cómo evalúa la adopción del nuevo proceso de desarrollo y nueva arquitectura en términos de:

	Su rol	Para la compañía
Proceso de desarrollo	<input type="checkbox"/>	<input type="checkbox"/>
Arquitectura móvil	<input type="checkbox"/>	<input type="checkbox"/>

Resultados cualitativos del nuevo proceso de desarrollo y arquitectura

RE1. La Tabla 29 presenta los resultados de la encuesta a personas transversales sobre el proceso de desarrollo:

		Transversales		
		P1	P2	P3
A mantener	En términos de facilidad de adopción del proceso, el nuevo proceso es:	Mucho Mejor	Mucho Mejor	Mucho Mejor
	En términos de documentación reducida, el proceso nuevo es:	Mucho Mejor	Mucho Mejor	Mucho Mejor
	En términos de simplicidad el nuevo proceso es:	Mucho Mejor	Mucho Mejor	Mucho Mejor
	En términos de autogestión, el nuevo proceso es:	Mucho Mejor	Mucho Mejor	Mucho Mejor
Problemas	En términos de transparencia hacia los stakeholders, el nuevo proceso es:	Mucho Mejor	Mejor	Mejor
	En términos de asertividad en cuanto a la estimación de esfuerzo de las tareas, está en el nuevo proceso es:	Mucho Mejor	Mucho Mejor	Mejor
	En cuanto a la priorización centrada en el usuario final, el apoyo del nuevo proceso es:	Mucho Mejor	Ni mejor ni peor	Ni mejor ni peor
	En términos de aceptación del cambios en cuanto a requerimientos, el nuevo proceso es:	Mucho Mejor	Ni mejor ni peor	Ni mejor ni peor

	Respecto a la puesta en producción de entregables en plazos breves (máximo 1 mes, mínimo 2 semanas), el nuevo proceso es:	Mucho Mejor	Mucho Mejor	Mejor
	Respecto a la definición de roles, bajo el nuevo proceso esto es:	Mucho Mejor	Mejor	Mejor
	Respecto a la resistencia al cambio por parte del equipo, en el nuevo proceso esto es:	Mucho Mejor	Mejor	Mejor
A intentar	Dar visibilidad de avance a los stakeholders	Considerado	Muy considerado	Muy considerado
	Usar método formal para la estimación de esfuerzo de desarrollo en el proceso:	Considerado	Considerado	Considerado
	Fallar rápido y barato	Muy considerado	Muy considerado	Considerado
	Entregas constantes y en breve tiempo (mínimo 2 semanas, máximo 1 mes) a usuarios finales	Muy considerado	Muy considerado	Muy considerado
	Utilizar un enfoque de desarrollo adaptativo de software que acepte el cambio	Considerado	Considerado	Muy considerado
	Priorización de requerimientos sobre las necesidades de los usuarios y clientes	Poco considerado	Neutral	Considerado
	Conocer a los usuarios y clientes (arquetipos de personas)	Poco considerado	Considerado	Considerado
	Roles y equipos definidos	Neutral	Neutral	Considerado
	Dar seguimiento a las iteraciones de desarrollo	Poco considerado	Considerado	Considerado

Incluir a los stakeholders en el desarrollo de la aplicación móvil	Considerado	Considerado	Considerado
--	-------------	-------------	-------------

Tabla 29- Resumen de encuesta a usuarios transversales sobre el nuevo proceso de desarrollo

RE2. En la Tabla 30 se presentan los resultados de la encuesta a personas nuevas sobre el proceso de desarrollo:

		Nuevos			
		P1	P2	P3	P4
A mantener	El proceso es de fácil adopción para los miembros del equipo	De acuerdo	Muy de acuerdo	Muy de acuerdo	Muy de acuerdo
	El proceso requiere de poca documentación	De acuerdo	De acuerdo	Muy de acuerdo	Muy de acuerdo
	En cuanto a las etapas y fases del proceso: es simple, posee pocas y breves etapas	De acuerdo	De acuerdo	De acuerdo	Muy de acuerdo
	El proceso permite la autogestión al interior del equipo	Muy de acuerdo	Muy de acuerdo	De acuerdo	Ni de acuerdo ni en desacuerdo
Problemas	El proceso permite transparentar el avance del desarrollo a los	De acuerdo	De acuerdo	De acuerdo	De acuerdo

	stakeholders				
	La estimación del esfuerzo de desarrollo para la planificación de funciones es acertada	De acuerdo	Ni de acuerdo ni en desacuerdo	En desacuerdo	Muy de acuerdo
	La priorización de las tareas se realiza en base a las necesidades del cliente	En desacuerdo	En desacuerdo	No sé	En desacuerdo
	Es posible modificar los requerimientos a lo largo del tiempo	Ni de acuerdo ni en desacuerdo	De acuerdo	De acuerdo	Muy de acuerdo
	Se realizan entregables a usuarios finales antes de contar con la versión final de una función o producto	De acuerdo	De acuerdo	De acuerdo	Muy de acuerdo
	Existen roles definidos en el proceso	Ni de acuerdo ni en desacuerdo	De acuerdo	De acuerdo	De acuerdo

	Respecto al cambio en cuanto a requerimientos, tecnología o mercado, ¿existe resistencia para afrontarlo a nivel de proceso dentro del equipo?	En desacuerdo	Ni de acuerdo ni en desacuerdo	Ni de acuerdo ni en desacuerdo	No sé
A intentar	Dar visibilidad de avance a los stakeholders, está:	Considerado	Considerado	Considerado	Considerado
	Usar método formal para la estimación de esfuerzo de desarrollo en el proceso:	Neutral	Considerado	Considerado	Considerado
	Fallar rápido y barato frente a la apuesta de hipótesis en términos de funciones para los usuarios finales	Neutral	Neutral	Considerado	Considerado
	Entregas constantes y en breve tiempo (mínimo 2 semanas, máximo 1 mes) a usuarios finales	Considerado	Considerado	Muy considerado	Muy considerado

Utilizar un enfoque de desarrollo adaptativo de software que acepte el cambio	Considerado	Considerado	Considerado	Considerado
Priorización de requerimientos sobre las necesidades de los usuarios y clientes está:	Poco considerado	Neutral	Neutral	Neutral
Conocer a los usuarios y clientes (arquetipos de personas) está:	Poco considerado	Neutral	Neutral	Neutral
Roles y equipos claros y definidos están:	Neutral	Neutral	Neutral	Considerado
Dar seguimiento a las iteraciones de desarrollo está:	Neutral	Considerado	Considerado	Considerado
Incluir a los stakeholders en el desarrollo del producto (aplicación móvil)	Neutral	Neutral	Considerado	Considerado

Tabla 30 - Resumen de encuesta a personas nuevas sobre el nuevo proceso de desarrollo

RE3. En la Tabla 31 se presentan resultados de la encuesta a personas transversales sobre la arquitectura:

		Transversales		
		P1	P2	P3
A mantener	La nueva arquitectura con una capa intermedia que permite ser resiliente frente a las caídas de los sistemas legados para datos no sensibles a actualizar en tiempo real es:	Mejor	Mejor	Mejor
	Respecto a la baja complejidad de las funciones de las aplicaciones, estas son:	Mucho mejor	Mejor	Ni mejor ni peor
	En cuanto al conocimiento detallado de cada sistema perteneciente a la arquitectura de la aplicación móvil, este es:	Mucho mejor	Ni mejor ni peor	Ni mejor ni peor
Problemas	Respecto a la frecuencia de actualización de datos, bajo la nueva arquitectura esta es:	Mejor	Mejor	Mejor
	En cuanto a la alta disponibilidad, la nueva arquitectura es:	Mucho mejor	Mucho mejor	Mucho mejor
	En cuanto a la edad y actualización del lenguaje de programación, en la nueva arquitectura es:	Mucho mejor	Mucho mejor	Mejor

	Considerando en la antigua arquitectura el uso de PCs de escritorio regulares usados como servidor, en la nueva arquitectura este tipo de uso es:	Mucho mejor	Mucho mejor	Mucho mejor
	En cuanto a la rapidez de ejecución de los pasos a producción, en la nueva arquitectura esto es:	Mucho mejor	Mucho mejor	Mejor
A intentar	Desarrollar Microservicios y funciones descentralizadas	Considerado	Considerado	Muy considerado
	Implementar una arquitectura de alta disponibilidad	Muy considerado	Muy considerado	Muy considerado
	Desarrollar solo un código fuente para ambos sistemas operativos	Muy considerado	Muy considerado	Muy considerado
	Implementar cargas por umbrales para los datos no necesarios en tiempo real desde los sistemas legados	Poco considerado	Considerado	Considerado
	Implementar pruebas automatizadas en las capas de Front-end y Back-end	Poco considerado	Considerado	Considerado
	Integrar bases de datos NoSQL para la carga de altos volúmenes de datos y gestión de estos	Neutral	Considerado	Muy considerado
	Implementar canales de CI/CD en las capas de Front-end y Back-end	Neutral	Neutral	Considerado

Tabla 31 - Resumen de encuesta a personas transversales sobre la nueva arquitectura

RE4. En la Tabla 32 se presentan los resultados de la encuesta a personas nuevas sobre la arquitectura:

		Nuevos			
		P1	P2	P3	P4
A mantener	La nueva arquitectura cuenta con una capa intermedia que permita ser resiliente frente a las caídas de los sistemas legados para datos no sensibles a actualizar en tiempo real	De acuerdo	Muy de acuerdo	Muy de acuerdo	Muy de acuerdo
	Las funciones de la aplicación móvil son de baja complejidad	De acuerdo	De acuerdo	Muy de acuerdo	Muy de acuerdo
	Existe un conocimiento detallado por parte del equipo sobre la arquitectura de los sistemas de la compañía	De acuerdo	Muy de acuerdo	De acuerdo	Muy de acuerdo
Problemas	La frecuencia de actualización de datos es alta	De acuerdo	De acuerdo	De acuerdo	En desacuerdo

	La nueva arquitectura posee alta disponibilidad en cuanto a servicios	De acuerdo	De acuerdo	De acuerdo	De acuerdo
	Los lenguajes de programación y arquitectura utilizada son actuales	De acuerdo	Muy de acuerdo	Muy de acuerdo	Muy de acuerdo
	Ha cambiado el uso de equipos PC como servidores de datos	De acuerdo	De acuerdo	De acuerdo	Muy de acuerdo
	Los pasos a producción son rápidos	Ni de acuerdo ni en desacuerdo	De acuerdo	De acuerdo	De acuerdo
A intentar	Desarrollar Microservicios y funciones descentralizadas	Considerado	Considerado	Considerado	Muy considerado
	Implementar una arquitectura de alta disponibilidad	Considerado	Muy considerado	Muy considerado	Muy considerado
	Desarrollar solo un código fuente para ambos sistemas operativos	Muy considerado	Muy considerado	Muy considerado	Muy considerado

	Implementar cargas por umbrales para los datos no necesarios en tiempo real desde los sistemas legados	Considerado	Considerado	Neutral	Neutral
	Implementar pruebas automatizadas en las capas de front-end y back-end	Poco considerado	Poco considerado	Poco considerado	Considerado
	Integrar bases de datos NoSQL para la carga de altos volúmenes de datos y gestión de estos	Considerado	Considerado	Considerado	Considerado
	Implementar canales de CI/CD en las capas de front-end y back-end	Poco considerado	Considerado	Considerado	Considerado

Tabla 32 - Resumen de encuesta a personas nuevas sobre la nueva arquitectura