

**ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA DE SISTEMAS**

**APLICACIÓN MÓVIL PARAMETRIZABLE Y MULTIPLATAFORMA PARA EL CORE  
BANCARIO FIT-BANK**

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

**DANIELA PAULINA RAMOS ACARO**

**daniela.ramos@epn.edu.ec**

**JEFFERSON VICENTE PAREDES VÁSQUEZ**

**Jefferson.paredes@epn.edu.ec**

**DIRECTOR: MSc, MARITZOL TENEMAZA.**

**maritzol.tenemaza@epn.edu.ec**

**CODIRECTOR: MSc, ADRIAN EGUEZ.**

**adrian.eguez@epn.edu.ec**

**Quito, Octubre 2019**

## **CERTIFICACIÓN**

Certificamos que el presente trabajo fue desarrollado por Daniela Paulina Ramos Acaro y Jefferson Vicente Paredes Vasquez bajo nuestra supervisión.

---

**MSc, Maritzol Tenemaza**

**DIRECTORA DE PROYECTO**

---

**MSc, Adrian Eguez**

**CODIRECTOR DE PROYECTO**

## **DECLARACIÓN DE AUTORÍA**

Nosotros, Daniela Paulina Ramos Acaro y Jefferson Vicente Paredes Vasquez, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

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

---

**Daniela Paulina Ramos Acaro**

---

**Jefferson Vicente Paredes Vasquez**

## DEDICATORIA

## AGRADECIMIENTO

# CONTENIDO

<b>1. INTRODUCCIÓN</b>	<b>10</b>
1.1. Planteamiento del problema	10
1.2. Ambiente de la Solución	11
1.2.1. Objetivos	11
1.2.2. Alcance	11
1.3. Estado del arte	12
1.3.1. Herramientas de análisis	12
1.3.2. Herramientas de desarrollo	12
1.3.3. Propuesta de metodología	17
<b>2. METODOLOGÍA</b>	<b>21</b>
2.1. Arquitectura de la aplicación	22
2.2. Arquitectura de comunicación de la aplicación	23
2.3. Definición de requerimientos	24
2.4. Desarrollo de la metodología	24
2.4.1. Historias épicas de usuario	24
2.4.2. Product Backlog	25
2.4.3. Planificación del Release	26
2.4.4. Sprint 0	27
2.4.5. Sprint 1	29
2.4.6. Sprint 2	40
2.4.7. Sprint 3	49
<b>3. RESULTADOS Y DISCUSIÓN</b>	<b>59</b>
3.1. Producto Final	59
3.1.1. Pantalla de ingreso a la aplicación	59
3.1.2. Pantallas Informativas de la Entidad Financiera	62
3.1.3. Pantalla de información consolidada	66
3.1.4. Pantalla detalle de movimientos	68
3.1.5. Pantalla de información detallada de cuentas	70
3.1.6. Pantalla de tabla de pagos	72

3.1.7.	Pantalla de menú lateral .....	74
3.1.8.	Pantalla de transferencias de dinero.....	76
3.2.	Pruebas con usuarios finales .....	80
3.2.1.	Utilidad percibida .....	81
3.2.2.	Facilidad de uso .....	82
4.	<b>CONCLUSIONES Y RECOMENDACIONES</b> .....	84
4.1	Conclusiones .....	84
4.2	Recomendaciones .....	85
5.	<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	86
6.	<b>ANEXOS</b> .....	89
6.1	Anexo I: Archivo de Configuración Auto Devops Gitlab .....	89
6.2	Anexo II: Detalle de historias de usuario para el Sprint 1 .....	90
6.3	Anexo III: Detalle Sprint 1 Backlog.....	92
6.4	Anexo IV: Revisión de criterios de aceptación del Sprint 1 .....	95
6.5	Anexo V: Detalle de historias de usuario para el Sprint 2.....	98
6.6	Anexo VI: Detalle Sprint 2 Backlog .....	99
6.7	Anexo VII: Revisión de criterios de aceptación del Sprint 2 .....	101
6.8	Anexo VIII: Detalle de historias de usuario para el Sprint 3.....	105
6.9	Anexo IX: Detalle Sprint 3 Backlog .....	107
6.10	Anexo X: Revisión de criterios de aceptación del Sprint 3 .....	110

## RESUMEN

El presente trabajo ofrece una aplicación móvil para el Core bancario FITBANK, disponible para dispositivos Android y IOS, donde su desarrollo se basó en la filosofía DevOps automatizando el proceso de construcción, realización de pruebas unitarias y calidad de código mediante el manejo de “Ramas” y “Pasos”, también se tomó como guía la metodología de desarrollo de software ágil Scrum, aplicando eventos y artefactos. Dando como resultado una aplicación que entrega a sus usuarios información importante de la entidad financiera como son: estados de cuenta, detalle de movimientos e información detallada de pagos realizados de préstamos y pólizas. También, la posibilidad de realizar transferencias entre distintos tipos de cuentas. Y ofrecer información sobre la entidad financiera referente a agencias, cajeros y productos. Finalmente se comprobó su utilidad y facilidad de uso bajo una encuesta realizada a varios usuarios, mismos que pudieron verificar su funcionamiento.

**Palabras clave:** DevOps, Scrum, Aplicación móvil, Core bancario, FITBANK.



## ABSTRACT

The present text offers a mobile application for the FITBANK Banking Core, available for Android and IOS devices, where its development was based on DevOps philosophy, automating the build process, unit tests and code quality through branch management and “ Stages ”, the Scrum agile software development methodology was also taken as a guide, applying events and artifacts. Resulting in an application that provides its users with important information from the financial institution such as: account statements, details of movements and detailed information on payments made of loans and policies. Also, the possibility of making transfers between different types of accounts. And offer information about the financial entity regarding agencies, ATMs and products. Finally, its usefulness and ease of use was verified under a survey of several users, who were able to verify its operation.

**Key words:** DevOps, Scrum, Mobile application, Banking Core, FITBANK.

# 1. INTRODUCCIÓN

## 1.1. Planteamiento del problema

Para la Comisión Económica para América Latina y el Caribe (CEPAL) en la región, el número de suscripciones a operadoras móviles creció 802,05% entre 2010 y 2015, esto implicó tener cerca de 50 suscripciones a operadoras móviles más por cada 100 personas [1]. Esta fuerte difusión de la alternativa móvil es posible por la gran cobertura de redes móviles y la diversidad y asequibilidad de los dispositivos. Este estudio concuerda con el Informe Global de Tecnologías de la Información 2016, donde Ecuador en el indicador de cobertura de red móvil por porcentaje de población ocupa el puesto 99 con un porcentaje de 96.9% [2], lo que evidencia un enorme y creciente uso del dispositivo móvil por gran parte de la población ecuatoriana, y de acuerdo al Instituto Nacional de Estadística y Censos (INEC) el 86,4% de los hogares posee al menos un teléfono celular [3] y según el Banco Central un 44,1% de la población del Ecuador usa servicios bancarios en línea [4], lo que evidencia la importancia del uso de esta tecnología para el crecimiento económico del país.

También el Banco Central del Ecuador informó que entre el año 2016 y 2017, la tasa de crecimiento de uso de los canales electrónicos fue del 16%. Y según la Superintendencia de Bancos, el 73% de las transacciones que se realizaron en instituciones financieras el 2017 fueron mediante canales electrónicos [5].

Por tal razón, la banca ecuatoriana ha volcado su esfuerzo hacia la masificación y modernización del uso de los productos y servicios bancarios a través de la banca móvil (*m-banking*), los bancos consideran más rentable este tipo de tecnologías ya que reducen costos y operan de forma más eficiente automatizando sus procesos [5]. Además, los servicios automatizados ofrecen al usuario “comodidad” por dar mayor flexibilidad, menores gastos, transacciones más efectivas, seguras y en menos tiempo.

En base a estos datos y como respuesta a la innovación y modernización, la empresa de desarrollo de Software Soft Warehouse que cuenta con más de 20 años de experiencia en el mercado tecnológico y que está presente en más de 30 entidades financieras a nivel nacional [6], propone el desarrollo de un aplicativo móvil multiplataforma para el Core Bancario FITBANK, aplicativo móvil que ofrezca a sus usuarios información importante acerca de las cuentas que posea en la entidad financiera como son: estados de cuenta, detalle de movimientos e información detallada de pagos realizados de préstamos. Además, la posibilidad de realizar

transferencias entre distintos tipos de cuentas. Y Finalmente ver información sobre la entidad financiera referente a agencias, cajeros y productos.

## **1.2. Ambiente de la Solución**

### **1.2.1. Objetivos**

#### **Objetivo General**

Desarrollar una aplicación móvil parametrizable para el Core bancario FITBANK disponible para dispositivos *Android* y *IOS*.

#### **Objetivos Específicos**

- Levantar los requerimientos funcionales mediante una reunión con la persona a cargo del proyecto de parte de la empresa Soft Warehouse, con el fin de especificar el alcance del desarrollo a realizar.
- Definir en conjunto con la persona a cargo del proyecto la arquitectura del sistema para el desarrollo de la aplicación.
- Aplicar la metodología Scrum en todas las etapas del desarrollo de la aplicación móvil usando como principal herramienta el marco de trabajo multiplataforma *NativeScript*.
- Realizar pruebas unitarias y de aceptación a cada una de las funcionalidades que sean implementadas en la aplicación móvil, para asegurar la calidad de esta.

### **1.2.2. Alcance**

Como prerrequisito principal para el uso de la aplicación móvil, el usuario y sus respectivos beneficiarios deben estar registrados previamente desde la página web de la institución financiera, el aplicativo no contempla dicha funcionalidad. Las características que se integran son:

- Aplicación nativa disponible para dispositivos *Android* y *IOS*.
- Integración del desarrollo y operación bajo el enfoque "*DevOps*".
- Autenticación mediante usuario, validación de imagen de usuario y contraseña.
- Consulta de estados de cuentas.
- Consulta de información de pólizas y préstamos.
- Transferencias entre cuentas propias del usuario, a terceros en la misma institución y a otras instituciones.

- Consulta de tablas de pagos de préstamos y pólizas.
- Consulta de información de cajeros, oficinas, productos y servicios de la institución financiera.

### 1.3. Estado del arte

#### 1.3.1. Herramientas de análisis

##### Banca móvil

La banca móvil (*m-banking*) es un nicho de la banca electrónica en la que los clientes acceden a un amplio abanico de servicios bancarios, tales como instrumentos de ahorro y crédito, vía canales electrónicos. La banca móvil requiere que el cliente tenga una cuenta donde pueda enviar transferencias y desde la que se puedan realizar pagos [7].

#### 1.3.2. Herramientas de desarrollo

Para el desarrollo del proyecto integrador se utilizaron las siguientes herramientas:

- Servicios *RESTful* del *Core* bancario para comunicación con la aplicación móvil.
- Marco de trabajo *Nativescript* para el desarrollo de la aplicación móvil en las plataformas de *Android* y *IOS*.
- Herramienta para la compilación y depuración de aplicaciones basadas en *NativeScript* (*SideKick*).
- Editor de texto para la escritura del código de programación (*Visual Studio Code*).
- Servicio web para control de versiones con herramientas *DevOps* (*Gitlab* con su suite *Auto DevOps*).
- El *framework* *Jasmine* para realizar las pruebas de unitarias.
- Plantilla en Excel para gestionar los artefactos de Scrum.

##### Servicios web *RESTful*

La comunicación desde la aplicación hacia el *Core* bancario se hará mediante el uso de servicios REST, estos servicios han ganado amplia adopción en toda la web como una alternativa más simple a los *Simple Object Access Protocol* (SOAP) y a los servicios web basados en el Lenguaje de Descripción de Servicios Web (*Web Services Description Language* - WSDL) [8].

*REST* define un conjunto de principios arquitectónicos por los que se pueden diseñar servicios Web que se centran en los recursos de un sistema, lo que incluye la forma en que los estados de los recursos se dirigen y transfieren a través de *Hypertext Transfer Protocol* (HTTP) por un amplio rango de clientes que están escritos en diferentes lenguajes. Los principios básicos de diseño de servicios Web de *REST* son [8]:

- Utilizar métodos HTTP de forma explícita.
- No mantener estados.
- Exponer los *Uniform Resource Locator* (URLS) como estructuras de directorios.
- Transferir *Extensible Markup Language* (XML), *JavaScript Object Notation* (JSON), o ambos.

Los servicios *REST* son el principal y único medio de comunicación entre la aplicación móvil con el *Core* bancario, a través del protocolo seguro *Hypertext Transfer Protocol Secure* (HTTPS).

### ***NativeScript***

El aplicativo móvil debía ser desarrollado tanto para las plataformas de *Android* como *IOS*, por tal razón se buscó un marco de trabajo que permita construir una aplicación que ofrezca una experiencia nativa en un lenguaje de programación conocido por los desarrolladores para no dedicar tiempo a capacitaciones antes de empezar el desarrollo.

La tecnología de *NativeScript* puede ser representada como una construcción de grandes partes así como lo describe la Figura 1. *NativeScript* es un marco de trabajo libre y de código abierto para construir aplicaciones móviles nativas multiplataforma. El mismo puede usar *TypeScript* con *Angular 2* y está compuesto de una máquina virtual *JavaScript*, un ejecutor, y un módulo de puente. La máquina virtual *JavaScript* interpreta y ejecuta código *JavaScript*. Luego, el módulo de puente traduce las llamadas a las API específicas de la plataforma y retorna los resultados a la función que los solicitó [9].

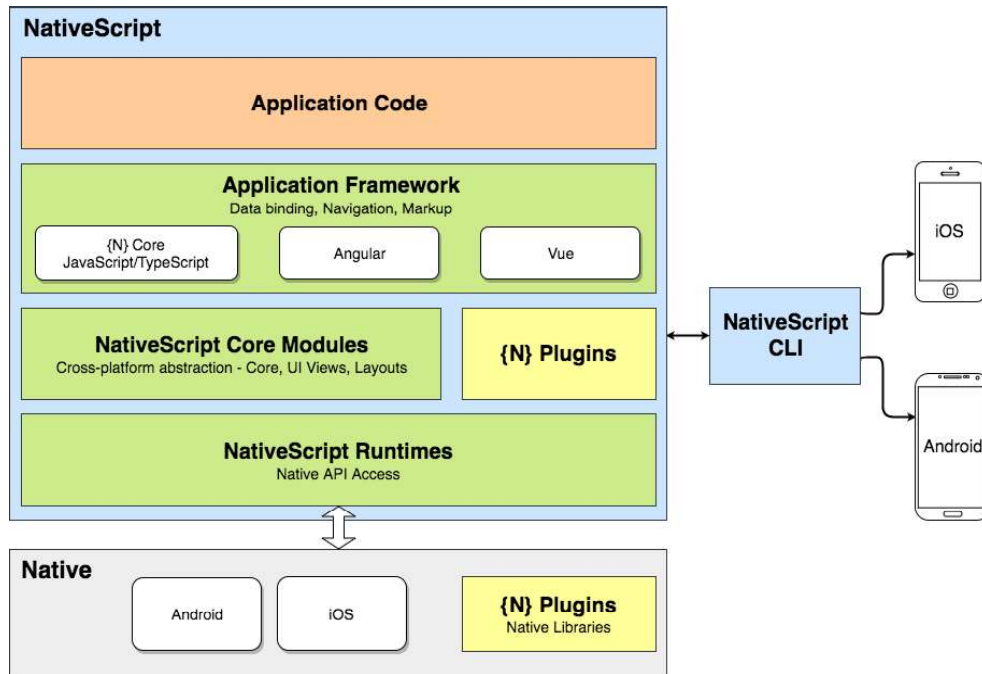


Figura 1: Arquitectura de NativeScript [10]

- **Tiempo de ejecución**

El tiempo de ejecución permite llamar a la interfaz de programación de aplicaciones (API) en los *frameworks* de *Android* y *IOS* utilizando el código JavaScript. Para hacerlo, utiliza las máquinas virtuales de JavaScript: V8 de Google para *Android* y la implementación de JavaScript de WebKit distribuida con *IOS* 7.0+ [10].

- **Módulos principales (*Core modules*)**

Los *Core* modules están ahí para proporcionar las abstracciones necesarias para acceder a las plataformas nativas subyacentes. Para cualquier módulo como el de gestos, por ejemplo. Define una API JS común que traduce el código de la aplicación *TypeScript* / *JavaScript* en llamadas de API de gestos nativos.

Los *Core* modules proporcionan una forma básica basada en XML para definir la interfaz de usuario, el enlace de datos y la navegación. Junto con Angular y Vue.js pueden usarse como tecnologías de *Application Framework* [10].

- **Interfaz de línea de comandos (CLI) de *NativeScript***

Es la interfaz que le permite crear y ejecutar aplicaciones utilizando *NativeScript*. Se ejecuta en sistemas operativos Windows, macOS o Linux [10].

## ***Plugins de NativeScript***

Los complementos de *NativeScript* son componentes básicos que encapsulan algunas funciones y ayudan a los desarrolladores a crear aplicaciones más rápidas. La mayoría son construidos por la comunidad, escritos en *TypeScript* / JavaScript. Algunos pueden incluir bibliotecas nativas, que se llaman desde el código TS / JS [10].

Resumiendo, las ventajas del desarrollo en *NativeScript* son [9]:

- Es libre y de código abierto. Se puede contribuir al código y usarlo como se quiera, en tanto no se viole la licencia Apache 2.0.
- Permite construir aplicaciones nativas para dispositivos *Android* y *IOS*. Cada componente de la UI que *NativeScript* expone es traducida en sus componentes UI nativos correspondientes.
- Da acceso a las API de plataformas nativas a través de código JavaScript. No se necesita tener conocimiento de Java u Objective-C para poder usar las API de las plataformas nativas porque se puede escribir todo en JavaScript. Esto significa que, si se necesita acceso a una característica específica del dispositivo, sólo se debe aprender cómo acceder a las API nativas con JavaScript.
- Ofrece a los usuarios una experiencia más cercana a la nativa que los *frameworks* de aplicaciones móviles híbridas como Córdoba.
- Permite a los desarrolladores construir fácilmente, desplegar y gestionar sus aplicaciones *NativeScript* a través de la plataforma *Telerik*.
- Dispone de soporte de día cero para nuevas plataformas nativas. Esto significa que se puede usar inmediatamente las últimas API nativas y los últimos componentes UI cuando Google o Apple actualicen sus plataformas.
- La documentación proporciona mucha información sobre cómo empezar, conceptos principales, y la interfaz de usuario. También hay ejemplos, tutoriales, un foro, preguntas en *Stack Overflow*, y aplicaciones de ejemplo que pueden ayudar a los principiantes a empezar con *NativeScript*.
- Escribir las aplicaciones en *NativeScript* con *TypeScript*. *TypeScript* transpila a JavaScript y agrega capacidades de programación orientada a objetos a JavaScript.
- Con *NativeScript CLI* se puede crear un nuevo proyecto, agregar una plataforma, ejecutar en un dispositivo y desplegar en una plataforma específica. Aparte, se puede instalar *plugins*, depurar la aplicación, y subir a la tienda de aplicaciones.

### ***NativeScript Sidekick***

Se seleccionó esta herramienta por su facilidad en la compilación y depuración de las aplicaciones en dispositivos de prueba (*Android* o *IOS*). También porque permite encontrar de manera más rápida diversos complementos que ayudan a la construcción de las aplicaciones. Además, ofrece los mismos servicios anteriormente mencionados en la nube, funcionalidad que fue muy útil en el caso de no contar con un dispositivo para compilar para *IOS* al momento del desarrollo y la ejecución de pruebas. Es un programa desarrollado por *Telerik* diseñado para ayudar a la creación de aplicaciones de *NativeScript* en lenguajes como Angular, JavaScript o *TypeScript* [11].

### ***Visual Studio Code***

Se trabajo con *Visual Studio Code* porque es un editor de código fuente desarrollado por *Microsoft* para *Windows*, *Linux* y *macOS*. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias [12].

### ***Filosofía DevOps***

Realizar manualmente los procesos de integración, pruebas y despliegue de incrementos de software ocasiona en la mayoría de los casos retrasos en la planificación de los proyectos y es propenso a errores con un resultado no confiable.

*DevOps* por su parte es un conjunto de principios y prácticas que optimizan el tiempo de entrega de software, gestionan la infraestructura como código y mejoran la experiencia del usuario en base a la retroalimentación de sus comentarios [13].

### ***GitLab***

*Gitlab* es un servicio web de control de versiones y desarrollo de software colaborativo basado en Git que tiene en su suite la opción de “*Auto DevOps*”.

*Auto DevOps* proporciona una configuración que permite detectar, construir, probar, implementar y monitorear automáticamente aplicaciones. Simplificando la configuración y ejecución de un ciclo de vida de desarrollo de software moderno y maduro [14].



En esta aplicación se hará uso de las características de auto construcción en cada versión (*Auto Build*), calidad de código (*Auto Code Quality*) y ejecución automática de pruebas unitarias (*Auto Test*).

### **Construcción automática (*Auto Build*)**

La compilación automática crea una compilación de la aplicación utilizando un archivo de texto (*Dockerfile*) el que contiene todos los comandos que un usuario podría llamar en la línea de comandos para ensamblar una imagen [15].

La imagen de Docker resultante se inserta automáticamente en el Registro de contenedores y se etiqueta con el SHA o la etiqueta de confirmación [16].

### **Auto Calidad del Código (*Auto Code Quality*)**

Utiliza la imagen de *Code Quality* para ejecutar análisis estáticos y otras comprobaciones de código en el código actual. El informe se crea y se carga como un artefacto que luego se puede descargar y consultar [17].

### **Auto Prueba (*Auto Test*)**

Auto Test ejecuta automáticamente las pruebas apropiadas para la aplicación utilizando los paquetes de compilación *Herokuish* y *Heroku* al analizar el proyecto y para detectar el lenguaje y el *framework* [18].

### ***Jasmine***

Framework de desarrollo basado en el comportamiento para probar el código JavaScript. No depende de ningún otro marco de JavaScript. No requiere un Modelo de Objetos del Documento (DOM). Tiene una sintaxis clara y obvia para que se pueda escribir fácilmente las pruebas [19].

### **1.3.3. Propuesta de metodología**

Para el desarrollo de la aplicación se utilizó eventos y artefactos de la metodología de desarrollo de software ágil iterativa “Scrum”. Esta metodología permitió añadir la funcionalidad de manera incremental basándose en el valor del negocio. Fue escogida entre otras como la más adecuada debido a que en el transcurso del desarrollo podría darse cambios en los requerimientos he integración de estos. Así Scrum deja fijos los requerimientos durante el *Sprint*, cualquier cambio a los requerimientos es tratado como un nuevo requerimiento para *Sprints* posteriores [20]. La

aplicación se construyó de forma iterativa entregando funcionalidad al final de cada sprint. A continuación, se detallan los roles, eventos y artefactos de la metodología que se hicieron uso.

## **Roles en Scrum**

- ***Scrum Master***

El Scrum Master es un facilitador que asegura que el Equipo Scrum cuente con un ambiente propicio para completar el proyecto con éxito. El Scrum Master guía, facilita y enseña las prácticas de Scrum a todos los involucrados en el proyecto; elimina los impedimentos que pueda tener el equipo y se asegura de que se estén siguiendo los procesos de Scrum [21].

El rol Scrum Master es representado por la directora del proyecto integrador, la cual ha guiado al Equipo Scrum a llevar con éxito el desarrollo del presente proyecto.

- ***Product Owner***

El Product Owner representa la voz del cliente, es la persona responsable de maximizar el valor del negocio para el proyecto. Este rol es responsable de articular los requisitos del cliente y de mantener la justificación del negocio del proyecto [22].

El rol Product Owner es representado por el arquitecto de TI de la empresa Soft Warehouse el cual proporcionó toda la información necesaria para llevar con éxito el desarrollo del aplicativo móvil.

- ***Equipo Scrum***

El Equipo Scrum es un grupo o equipo de personas responsables de entender los requerimientos del negocio especificados por el *Product Owner*, de estimar las historias de usuarios y de la creación final de los entregables del proyecto [23].

El desarrollo y pruebas del aplicativo fueron realizados por los dos integrantes del proyecto integrador con una disponibilidad de 3 horas diarias.

## Eventos Scrum

Los eventos que se utilizaron en el desarrollo del aplicativo son:

- **Reunion de planificación del Sprint (*Sprint Planning Meeting*)**

Cada sprint empieza con una reunión de planificación del Sprint (*Sprint Planning Meeting*) durante la cual se consideran las historias de usuario de alta prioridad para su inclusión en cada Sprint [24].

En el desarrollo del aplicativo, el equipo considera priorizar aquellas historias de usuario que le permiten a los interesados ver el valor del negocio y al tratarse de una aplicación bancaria se tomó en cuenta empezar por la funcionalidad de autenticación segura, siguiendo con las transferencias bancarias y finalizando con el despliegue de la información del usuario.

- **El Sprint**

Un *sprint* generalmente tiene una duración de una a seis semanas durante las cuales el Equipo Scrum trabaja en la creación de entregables en incrementos del producto [24].

Para el caso de estudio cada *sprint* tuvo una duración 2 semanas con disponibilidad del grupo de 3 horas diarias.

- **Reunión diaria de sincronización del equipo (*Scrum daily meeting*)**

Durante el *sprint*, se llevan cabo *Daily Standups* muy breves y concretos, donde los miembros del equipo discuten el progreso diario [24].

Para el caso de estudio las reuniones diarias fueron 0.25 h todos los días.

- **Reunión de Revisión del Sprint (*Sprint Review Meeting*)**

Hacia el final del sprint, se lleva a cabo una Reunión de Revisión del Sprint (*Sprint Review Meeting*) en la cual se proporciona una demostración de los entregables al *Product Owner* y a los *stakeholders* relevantes. El *Product Owner* acepta los entregables sólo si cumplen con los criterios de aceptación predefinidos [24].

Para el caso de estudio las reuniones de revisión se llevaron 2 h luego de finalizar el sprint.

- **Reunión de Retrospectiva del Sprint (*Retrospect Sprint Meeting*)**

El ciclo del *sprint* termina con una Reunión de Retrospectiva del *Sprint* (*Retrospect Sprint Meeting*), donde el equipo analiza las formas de mejorar los procesos y el rendimiento a medida que avanzan al siguiente *sprint* [24].

Para el caso de estudio las reuniones de revisión se llevaron a cabo 1 h 30 min luego de la Reunión de Revisión del *Sprint*.

## **Artefactos de Scrum**

Los artefactos que se utilizaron en el desarrollo del aplicativo son:

- **Lista de Producto (*Product Backlog*)**

Una vez que el *Product Owner* ha recibido los requerimientos del negocio del cliente y los ha escrito en forma de historias de usuario viables, este trabaja con el cliente y *stateholders* para entender los requerimientos del negocio que proporcionan el máximo valor empresarial [25].

El equipo ha considerado dimensionar el esfuerzo basados en puntos de historia mediante la técnica de *Planning Poker*.

El método de puntos de historia permite estimar y relacionar la complejidad de las historias de usuario con respecto a otra que es tomada como pivote, es decir cuántas historias de usuario se podrán implementar en función de la productividad del equipo, de la carga de trabajo y del resto de factores ambientales. Con ello las tareas de planificación se simplifican y se logran valores bastantes más cercanos a la realidad.

La técnica *Planning Poker* reúne al equipo de desarrollo con una baraja de *Poker* modificada con una pseudo-secuencia de Fibonacci. Así cada participante tiene las siguientes cartas: 0, 3, 5, 8? e infinito. El cero significa que la historia ya está hecha o no requiere ningún esfuerzo, el interrogante significa que falta información para estimar esa historia o tarea y el infinito que es demasiado grande y hay que trocearla [26].

El *Product Owner* refina constantemente el *Product Backlog* para garantizar que los nuevos requerimientos de negocio se añadan, y que los requisitos existentes estén debidamente

documentados y priorizados. Esto asegura que a los requisitos más valiosos en el cumplimiento de los objetivos del programa se les esté dando alta prioridad y que los restantes reciban una prioridad más baja [27]

- **Lista de Pendientes del Sprint (Sprint Backlog)**

Durante las reuniones de planificación del sprint, el Equipo Scrum se compromete con las historias de usuario para un sprint e identifica y estima las tareas. Cada miembro del Equipo Scrum selecciona las tareas en las que planean trabajar en el sprint con base en sus habilidades y experiencia. El Equipo Scrum elabora el Sprint Backlog y el Sprint Burndown Chart utilizando las historias de usuario y la lista antes mencionada durante las reuniones de planificación del sprint [28].

- **Sprint Burndown Chart**

El Sprint Burndown Chart es una gráfica que muestra la cantidad de trabajo pendiente en el actual sprint. Debe actualizarse al final de cada día conforme se concluye el trabajo. Dicha gráfica muestra el progreso que ha realizado el Equipo Scrum y permite también la detección de estimaciones que pudieron haberse hecho incorrectamente [29].

## 2. METODOLOGÍA

La metodología de desarrollo exige definir los roles de responsabilidad a los involucrados, en el presente proyecto se han identificado los roles que se muestran la Tabla 1.

Tabla 1: Roles en Scrum

<b>Roles de Scrum definidos para el proyecto</b>	
<b>Scrum Master</b>	Maritzol Tenemaza.
<b>Product Owner</b>	Arquitecto de TI empresa Soft Warehouse Heccer Benavides
<b>Scrum Team</b>	Daniela Ramos
	Jefferson Paredes

## 2.1. Arquitectura de la aplicación

*Nativescript* permite el uso de *frameworks* de *Javascript* para su desarrollo, en el presente proyecto integrador se va a utilizar Angular ya que los integrantes del equipo tienen experiencia utilizando este tipo de *framework*.

Basándose en la arquitectura que ofrece Angular la aplicación se dividirá en componentes y utilizará el patrón de arquitectura modelo vista controlador (MVC).

Cada componente tendrá una clase en *Typescript* en la que se mapeará el modelo de datos que viene del API Rest, otra clase igualmente en *Typescript* en la que se manejará los servicios y la lógica, finalmente una plantilla en HTML (template) donde se definirá la vista del usuario.

La comunicación entre la lógica y la vista se hará mediante enlace de datos (data bindings) y eventos. La figura 2 ofrece una vista de la aplicación enlistando todos sus componentes y como ejemplo la estructura de la comunicación interna del componente que gestiona la información de cajeros.

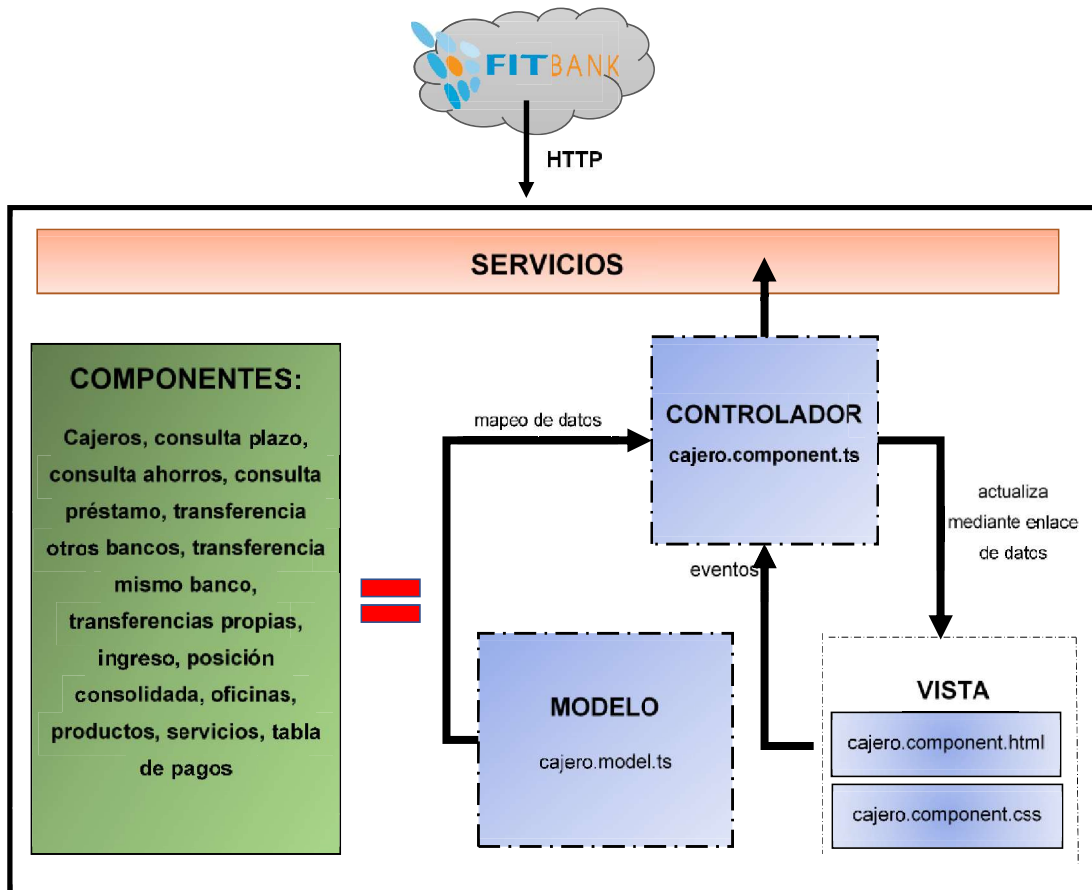


Figura 2: Arquitectura de la Aplicación

## 2.2. Arquitectura de comunicación de la aplicación

**Teléfonos *Android/IOS*:** Representa a la aplicación móvil desarrollada que será instalada en sistemas operativos *Android* y *IOS*.

**Servicios *REST*:** Se encarga de recibir las peticiones realizadas por la aplicación para interactuar con la BDD en caso de consultas, o interactuar con el UCI en el caso de transacciones.

**UCI (*Universal Channel Interface*):** Se encarga de gestionar la comunicación entre los medios externos y el *Core* del sistema financiero.

**Core:** Se encarga de procesar y realizar las transacciones provenientes de las aplicaciones externas, además de asegurar la transaccionalidad del Requerimientos de la aplicación.

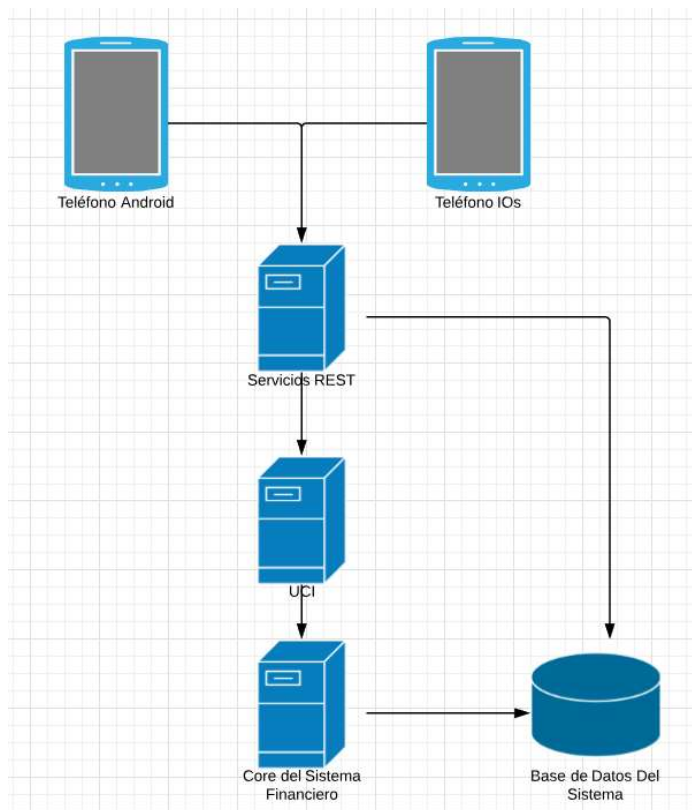


Figura 3: Arquitectura de comunicación del sistema

## 2.3. Definición de requerimientos

La banca móvil se define como un servicio por el que los usuarios pueden acceder a información transaccional de su entidad bancaria. En ocasiones, esto incluye operativa transaccional [30].

Basándose en esta definición y teniendo en cuenta lo conversado con el *Product Owner*, la aplicación debería cubrir como mínimo los siguientes requerimientos funcionales, como lo muestra la Tabla 2.

Tabla 2: Requerimientos funcionales

ID	REQUERIMIENTO
RF01	La aplicación permitirá realizar transferencias entre distintos tipos de cuentas en distintos tipos de entidades financieras.
RF02	La aplicación tendrá autenticación mediante usuario, validación de imagen de usuario y contraseña.
RF03	La aplicación permitirá visualizar la información y estado de cada una de las cuentas, préstamos y pólizas que el usuario tenga en la institución financiera.
RF04	La aplicación permitirá visualizar la información de las oficinas, productos, cajeros y servicios que tiene la institución financiera.
RF05	La aplicación permitirá la internacionalización de los textos mostrados.

## 2.4. Desarrollo de la metodología

### 2.4.1. Historias épicas de usuario

A partir de los requerimientos mencionados en la Tabla 2, se generaron las siguientes historias épicas de usuario como lo muestra la Tabla 3.



Tabla 3: Historias épicas de usuario

ID	HISTORIAS ÉPICAS DE USUARIO
HUE01	Como cliente requiero ejecutar transferencias entre distintos tipos de cuentas, en distintos tipos de entidades financieras para efectuar envío de dinero según mis necesidades.
HUE02	Como cliente requiero autenticarme en la aplicación mediante usuario, validación de imagen de usuario y contraseña. Para conservar la integridad y confidencialidad de mi dinero.
HUE03	Como cliente requiero poder ver información de cada una de mis cuentas, préstamos y pólizas en la institución financiera para conocer el saldo de estas.
HUE04	Como cliente requiero ver información de las oficinas, productos, cajeros y servicios que tiene la institución financiera para ahorrar tiempo de búsqueda de dicha información.
HUE05	Como cliente requiero poder utilizar la aplicación con el idioma predeterminado de mi dispositivo.

## 2.4.2. Product Backlog

Las historias de usuario épicas se dividen en historias de usuario, la dimensión/esfuerzo de cada una se definió de acuerdo con la técnica de *Planning Poker* donde la medida de estimación está basada en puntos de historia, se ha tomado como pivote la historia de usuario HUE04-01 asignándole 4 puntos y en base a la misma se ha estimado el resto. En cambio, la prioridad está basada en el valor que cada historia representa para el negocio, así como lo muestra la Tabla 4.

Tabla 4: Tabla de valoración de esfuerzo y prioridad

VALORACIÓN	ESFUERZO	PRIORIDAD DEL NEGOCIO
1	Bajo	Baja
2	Mediano	Mediana
3	Alto	Alta
4	Muy alto	Muy alta

A continuación, en la Tabla 5, se presenta el *Product backlog* inicial, donde se expone la descripción de la funcionalidad, el esfuerzo basado en puntos de historia y la prioridad basada en el valor del negocio.

Tabla 5: Product backlog

<b>PRODUCT BACKLOG</b>				
<b>HISTORIA ÉPICA DE USUARIO</b>	<b>ID</b>	<b>HISTORIA DE USUARIO</b>	<b>DIMENSIÓN / ESFUERZO (Puntos de historia)</b>	<b>PRIORIDAD</b>
HUE02	HUE02-01	Validación de nombre de usuario, imagen y contraseña	9	4
HUE01	HUE01-01	Transferencias entre cuentas propias	6	4
	HUE01-02	Transferencias a terceros en la misma institución	6	4
	HUE01-03	Transferencias a otras instituciones	6	4

Tabla 5: Product backlog

<b>PRODUCT BACKLOG</b>				
<b>HISTORIA ÉPICA DE USUARIO</b>	<b>ID</b>	<b>HISTORIA DE USUARIO</b>	<b>DIMENSIÓN / ESFUERZO</b>	<b>PRIORIDAD</b>
HUE03	HUE03-01	Posición consolidada de cuentas, préstamos y pólizas	10	3
	HUE03-02	Detalle de movimientos de cada cuenta	5	3
	HUE03-03	Tabla de amortización de préstamos y pólizas	5	3
HUE04	HUE04-01	Información de cajeros	4	3
	HUE04-02	Dirección de oficinas	4	3
	HUE04-03	Productos de la institución financiera	4	3
	HUE04-04	Servicios de la institución financiera	4	3

### 2.4.3. Planificación del Release

Teniendo en cuenta que para el cumplimiento de todos los requerimientos se estimó un tiempo de 6 semanas, en las cuales están incluidas las actividades de reuniones y desarrollo. Definimos que un Sprint durará 2 semanas, teniendo un total de 3 *Sprints* estos deberían ser suficientes para terminar todas las historias de usuario las cuales sumadas tienen 63 puntos de esfuerzo por 3 *Sprints*, así cada uno debería tener un promedio de 21 puntos de esfuerzo, para esto tomamos los 21 primeros puntos de esfuerzo para el primer Sprint, los siguientes 21 puntos para el segundo y así sucesivamente, con esto garantizamos un equilibrio de carga de esfuerzo en cada Sprint.

Finalmente, el equipo agrega el Sprint 0 con las tareas necesarias para que se lleve con éxito el desarrollo de los demás *Sprints*. La Tabla 6 presenta el *Release Planning* Inicial del proyecto.

Tabla 6: Release planning inicial

SPRINT 1		SPRINT 2		SPRINT 3	
ID	ESFUERZO	ID	ESFUERZO	ID	ESFUERZO
HUE02-01	9	HUE01-03	6	HUE03-03	5
HUE01-01	6	HUE03-01	10	HUE04-01	4
HUE01-02	6	HUE03-02	5	HUE04-02	4
				HUE04-03	4
				HUE04-04	4
TOTAL	21	TOTAL	21	TOTAL	21

### 2.4.4. Sprint 0

#### Objetivo:

Preparar el ambiente de desarrollo:

- Instalar el *framework NativeScript* con los requerimientos para construir para la plataforma de Android.
- Instalar el editor de texto *Visual Studio Code*.
- Instalar la herramienta *Sidekick*.
- Crear el repositorio para el control de versiones con herramientas *DevOps*.
- Verificar el correcto funcionamiento de los servicios *RESTful* del *Core* bancario.

## Ejecución del Sprint 0

Durante la ejecución de este Sprint se puede mencionar los siguientes detalles importantes:

### DevOps

Teniendo en cuenta las distintas etapas del ciclo de desarrollo del software se han implementado algunas prácticas de integración continua *DevOps* de acuerdo con los distintos ambientes por los que pasó la aplicación como lo muestra la Figura 4, el detalle del archivo se encuentra como Anexo I.

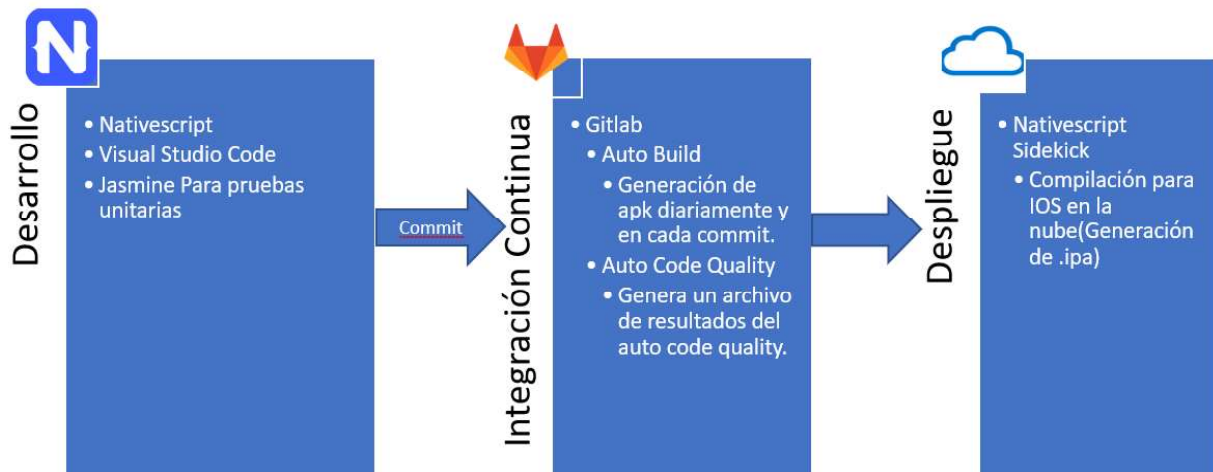


Figura 4: Integración continua

En la rama de desarrollo se consideró la construcción automática del software mediante la opción de *Auto Build* generando el archivo ejecutable de aplicaciones para *Android* (APK) en cada *Commit*.

En rama de pruebas del repositorio se consideró inicialmente realizar la opción de *Auto Test*, para realizar la ejecución automática de las pruebas unitarias desarrolladas en Jasmine, véase la referencia en el capítulo 1 Introducción, en el subcapítulo 1.3 estado del arte [19]. El inconveniente encontrado fue que, para la ejecución de estas, se necesita tener un dispositivo móvil conectado al computador en dónde se están ejecutando las mismas, debido a esta configuración el servidor donde está creado el repositorio no podía cumplir con esta necesidad. Por tal razón se decidió que la ejecución de las pruebas unitarias se va a realizar manualmente conforme se desarrollen las mismas.

También en la rama de pruebas se consideró verificar las reglas de código software mediante la opción de *Auto Code Quality* usando la librería de *tslint* más la librería *codelyzer* que verifica

estilo de código para proyectos de angular con *typescript*, esta verificación se ejecuta cada que se realiza un *commit*.

### Revisión del Sprint

Finalmente, se puede concluir que, si se alcanzó el objetivo del Sprint 0 que consistió en la preparación del ambiente de desarrollo, teniendo en cuenta que se hicieron modificaciones a las opciones de *Auto DevOps* descritas anteriormente.

### 2.4.5. Sprint 1

#### Objetivo:

- Autenticación mediante usuario, validación de imagen de usuario y contraseña en la aplicación.
- Realizar transferencias entre cuentas propias y a terceros en la misma institución.

#### Historias de usuario

El Sprint 1 satisface las historias de usuario HUE02-01, HUE01-01 y HUE01-02, las mismas que suman un total de 21 puntos de esfuerzo. Como se ha definido cada Sprint tendrá una duración de 2 semanas en las que hay que terminar los requerimientos contemplados como lo muestra la Tabla 7, el detalle completo de las historias de usuario para el Sprint 1 se presentan en el Anexo II.

Tabla 7: Historias de usuario Sprint 1

ID	TÍTULO	ESFUERZO	ESTADO
HUE02-01	Validación de nombre de usuario, imagen y contraseña	9	Por implementar
HUE01-01	Transferencias entre cuentas propias	6	Por implementar
HUE01-02	Transferencias a terceros en la misma institución.	6	Por implementar

## Sprint 1 Backlog

Una vez definidas las historias del Sprint, es necesario definir las tareas para su cumplimiento, el resumen del Backlog del Sprint 1 se muestra en la Tabla 8, el detalle completo del Sprint 1 Backlog se encuentra en el Anexo III.

Tabla 8: Sprint 1 Backlog

ID	ID TAREA	TAREAS
HUE02-01	TR-01	<p>Crear el modelo de datos para manejo de la información recibida del servidor por medio de los servicios <i>RESTful</i>.</p> <p>Crear el componente para el ingreso con:</p> <ul style="list-style-type: none"> <li>-Archivo de lógica del negocio en <i>Typescript</i>.</li> <li>-Vista en Angular.</li> </ul> <p>Crear el diccionario de palabras en inglés y español para la vista creada.</p> <p>Crear y ejecutar las pruebas unitarias del componente.</p>
	TR-02	<p>Crear el método para que consuma el servicio <i>REST</i> para la validación de la existencia del usuario.</p>
	TR-03	<p>Crear una matriz que se llene de acuerdo con un método de ordenamiento para que las imágenes se presenten aleatoriamente.</p>
	TR-04	<p>Crear el método que valide la imagen seleccionada en la matriz consumiendo el servicio <i>REST</i> de validación de imagen del usuario.</p>
	TR-05	<p>Crear un campo para ingresar la contraseña de manera oculta.</p>
	TR-06	<p>Crear el método que consuma el servicio <i>REST</i> de validación de la contraseña de usuario.</p>
	TR-07	<p>Crear el método para el cifrado de la contraseña.</p>

Tabla 8: Sprint 1 Backlog

ID	ID TAREA	TAREAS
HUE01-01	TR-08	Crear el modelo de datos para transferencias entre cuentas propias. Crear servicio para el consumo del <i>API REST</i> . Crear el componente para las transferencias entre cuentas propias. -Archivo de lógica del negocio en <i>Typescript</i> . -Vista en <i>Angular</i> Crear y ejecutar las pruebas unitarias del componente.
	TR-09	Crear diccionario de palabras en inglés y español para la vista
	TR-10	Crear el método para validación de las cuentas en los combos de cuenta origen y destino.
	TR-11	En la vista crear el input que solo permita el ingreso de números.
	TR-12	Crear el botón "Solicitar" con el método para enviar una solicitud de token al servidor del <i>Core</i> .
	TR-13	Crear el método que consuma el servicio <i>REST</i> que valida la temporalidad del Token que se va a usar en el envío de la transferencia.
	TR-14	Crear el botón "x" para limpiar todos los campos que contiene la pantalla.
	TR-15	Crear el método que consuma el servicio <i>REST</i> que envía la transferencia entre cuentas propias. Crear el mensaje que devuelve el servidor al intentar realizar la transferencia.
	TR-16	

Tabla 8: Sprint 1 Backlog

ID	ID TAREA	TAREAS
HUE01-02	TR-17	<p>Crear el modelo de datos para transferencias a terceros en la misma institución.                      Crear el servicio para el consumo del <i>API REST</i>.                      Crear el componente para las transferencias a terceros en la misma institución.                      -Archivo de lógica del negocio en <i>Typescript</i>.                      -Vista en Angular creada.                      Crear y ejecutar las pruebas unitarias del componente.</p>
	TR-18	Crear diccionario de palabras en inglés y español para la vista
	TR-19	En la vista crear el input que solo permita el ingreso de números.
	TR-20	Crear el botón "Solicitar" con el método para enviar una solicitud de token al servidor del <i>Core</i> .
	TR-21	Crear el método que consuma el servicio <i>REST</i> que valida la temporalidad del Token que se va a usar en el envío de la transferencia.
	TR-22	Crear el botón "x" para limpiar todos los campos que contiene la pantalla.
	TR-23	Crear el método que consuma el servicio <i>REST</i> que envía la transferencia a terceros en la misma institución. Crear el mensaje que devuelve el servidor al intentar realizar la transferencia.

### Ejecución del Sprint

Durante el primer Sprint se puede mencionar los siguientes puntos:

- **Splash**

Al iniciar la aplicación se necesitaba que el usuario espere mientras esta se cargaba por eso se decidió implementar un *Splash* con el logotipo del *Core* bancario. Un *splash screen* se define como la pantalla o imagen de bienvenida que aparece cuando se carga la app en el móvil del usuario, dura mientras se carga la app y mantiene ocupado al usuario [31], la Figura 5 muestra el resultado.





Figura 5: Splash de la aplicación

- **Transferencias**

Para realizar transferencias, se necesita enviar un token de validación, a través del consumo del *API REST* se solicita esta clave temporal con los datos de sesión del usuario, la Figura 6 muestra el servicio.

```
obtainKeyTemporary() {  
    return this.http.post(Config.apiUrl + "clavetemporal", "",  
        {  
            params: Config.getCommonParams(),  
            headers: Config.getCommonHeaders()  
        });  
}
```

Figura 6: servicio para solicitar la clave temporal

La clave tiene un tiempo de caducidad de 5 minutos, la Figura 7 muestra un ejemplo del correo recibido por el usuario.



Figura 7: Correo de solicitud de la clave temporal

Luego de ingresar todos los datos para la transferencia, se consume el servicio que envía la petición al Core, este valida las cuentas de origen y destino para los fondos, el monto y la clave temporal, finalmente el sistema envía un mensaje de éxito en la transferencia o el error cometido, la Figura 8 muestra el servicio para enviar la transferencia.

```

ownTransfer(cuenta: string, cuentaPropia: string, clave: string, monto: string) {
    return this.http.post(Config.apiUrl + "tercerosmismainstitucion", "",
        {
            params: Config.getCommonParams()
                .append("cuenta", cuenta)
                .append("cuentaBe", cuentaPropia)
                .append("clave", clave)
                .append("mon", (method) Config.getCommonHeaders(): any
            headers: Config.getCommonHeaders()
        }
    );
}

```

Figura 8: Servicio para enviar una transferencia

Se utilizó la librería de Jasmine para crear un ambiente de pruebas que permita comprobar la funcionalidad del código. En la figura 9 se puede observar un ejemplo de prueba unitaria al componente de ingreso en el método para seleccionar la imagen del usuario.

```

it("Verificando el metodo que comprueba la imagen seleccionada del usuario",function(){
    loginComponent.imageSelected=new Icon("18.png",50018,4,3,true);
    loginComponent.imageFromServer=50018;
    loginComponent.submitImage();
    expect(loginComponent.isImageSelected).toEqual(true)
    && expect(loginComponent.isContinued).toEqual(true);
});

```

Figura 9: Ejemplo de prueba unitaria

## Revisión del Sprint 1

Finalmente, se puede concluir que, si se alcanzó el objetivo del Sprint 1 que consistió en realizar el ingreso a la aplicación y las transferencias entre cuentas propias y a terceros en la misma institución.

## Pruebas de aceptación

En la Tabla 9, se puede observar los criterios de aceptación que fueron cumplidos, en este caso hay detalles de algunos criterios que se deben completar de acuerdo con los comentarios como son cambios mínimos no debería representar una carga de trabajo significativa, el detalle de los criterios de aceptación están en el Anexo IV.

Tabla 9: Revisión de criterios de aceptación del Sprint 1

ID	CRITERIOS DE ACEPTACIÓN	CH EC KL IST	COMENTARIO
HUE02-01	La aplicación tendrá una pantalla inicial para el acceso de los usuarios, donde se podrá ingresar el usuario, escoger una imagen e ingresar la contraseña.	SI	
	La aplicación validará el ingreso de usuarios previamente creados en la página web.	SI	
	La aplicación contará con una matriz aleatoria de imágenes para validar la imagen del usuario.	SI	
	La aplicación validará la selección correcta de la imagen de usuario.	SI	
	La aplicación permitirá el ingreso oculto de la contraseña de usuario.	SI	
	La aplicación validará el ingreso de la contraseña del usuario.	SI	
	La aplicación protegerá el envío de la contraseña.	SI	

Tabla 9: Revisión de criterios de aceptación del Sprint 1

ID	CRITERIOS DE ACEPTACIÓN	CHECKLIST	COMENTARIO
HUE01-01	La aplicación tendrá una pantalla para realizar transferencias entre cuentas propias, donde se pueda escoger la cuenta propia desde la que se va a realizar la transferencia y otra opción en donde se seleccione la cuenta propia a la que debe ir la transferencia. El ingreso del monto, botones para solicitud del token, el envío de la transferencia o cancelación de esta.	SI	
	Los textos en la pantalla de transferencias propias se deberán visualizar con el idioma del dispositivo.	SI	
	Para realizar transferencias propias la aplicación debe validar que la cuenta de origen de fondos no aparezca en la lista de selección para el destino.	SI	
	Para realizar transferencias propias la aplicación debe permitir ingresar el monto solo en números.	NO	El teclado presenta caracteres alfanuméricos como los signos básicos de operaciones, se debe validar que no se puedan ingresar.
	Para realizar transferencias propias la aplicación debe permitir pedir un Token, mismo que se envía a un correo del usuario.	SI	
	Para realizar transferencias propias la aplicación debe permitir validar el Token ingresado.	SI	
	Para realizar transferencias propias la aplicación debe permitir limpiar los campos.	SI	
	En el caso que la transferencia entre cuentas propias no pueda ser ejecutada la aplicación debe mostrar el mensaje que envía el sistema.	SI	
	En el caso que la transferencia entre cuentas propias se ejecute satisfactoriamente la aplicación debe mostrar el mensaje que envía el sistema.	SI	

Tabla 9: Revisión de criterios de aceptación del Sprint 1

HUE01-02	La aplicación tendrá una pantalla para realizar transferencias a terceros en la misma institución, donde se pueda escoger la cuenta propia desde la que se va a realizar la transferencia, el ingreso del monto, botones para solicitud del token, el envío de la transferencia o cancelación de esta.	SI	
	Los textos en la pantalla de transferencias a terceros se deberán visualizarse con el idioma del dispositivo.	SI	
	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir ingresar el monto solo en números.	NO	El teclado presenta caracteres alfanuméricos como los signos básicos de operaciones, se debe validar que no se puedan ingresar.
	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir pedir un Token, mismo que se envía a un correo del usuario.	SI	
	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir validar el Token ingresado.	SI	
	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir limpiar los campos.	SI	
	En el caso que la transferencia a terceros en la misma institución no pueda ser ejecutada la aplicación debe mostrar el mensaje que envía el sistema.	SI	
	En el caso que la transferencia a terceros en la misma institución se ejecute satisfactoriamente la aplicación debe mostrar el mensaje que envía el sistema.	SI	

### **Adaptación del *Product Backlog***

Después de realizar un análisis entre los miembros del equipo, se concluyó que es necesario contar con un menú lateral en el que se cuente con las funciones principales de la aplicación, ya que la pantalla principal solo se desea que se centre en la posición consolidada del usuario, por tal razón se añade como tarea adicional para la historia HUE01-03 en el siguiente Sprint, la creación de este menú con las siguientes características:

Se debe contar con un menú lateral, el mismo deberá ocultarse/mostrarse al deslizarse de izquierda a derecha. Mostrará el nombre del usuario y contará con las opciones de:

1. Cuentas propias
2. Terceros misma institución
3. Otra institución.
4. Posición consolidada.

Cada opción llevará a su respectiva pantalla.

Finalmente, faltó por concluir un par de tareas para las historias HUE01-01 Y HUE01-02 que no representan un esfuerzo significativo. Por ende, se procedió a adaptar al *Product backlog* como se enseña en la Tabla 10.

Tabla 10: Product Backlog adaptado tras finalizar el Sprint 1

<b>Historia Épica de usuario</b>	<b>H U</b>	<b>Historia de usuario</b>	<b>Dimensión / Esfuerzo (Puntos de historia)</b>	<b>Prioridad</b>	<b>Estado</b>
HUE02	HUE02-01	Validación de nombre de usuario, imagen y contraseña	9	4	Terminado
HUE01	HUE01-01	Transferencias entre cuentas propias	0.5	4	En desarrollo
	HUE01-02	Transferencias a terceros en la misma institución	0.5	4	En desarrollo
	HUE01-03	Transferencias a otras instituciones	7	4	Por implementar

Tabla 10: Product Backlog adaptado tras finalizar el Sprint 1

Historia Épica de usuario	H U	Historia de usuario	Dimensión / Esfuerzo	Prioridad	Estado
HUE03	HUE03-01	Posición consolidada de cuentas, préstamos y pólizas	10	3	Por implementar
	HUE03-02	Detalle de movimientos de cada cuenta	5	3	Por implementar
	HUE03-03	Tabla de amortización de préstamos y pólizas	5	3	Por implementar
HUE04	HUE04-01	Información de cajeros	4	3	Por implementar
	HUE04-02	Dirección de oficinas	4	3	Por implementar
	HUE04-03	Productos de la institución financiera	4	3	Por implementar
	HUE04-04	Servicios de la institución financiera	4	3	Por implementar

### Retrospectiva del Sprint 1

La elaboración de la primera historia de usuario acerca del ingreso a la aplicación requirió más tiempo del estimado ya que se tuvo que trabajar con un usuario de prueba y se tuvo que gestionar la creación en la base de datos. Además, se necesitaba el envío del correo electrónico con el token de validación a uno de los integrantes del equipo, la solicitud para la creación de la notificación al correo retraso en ser aprobada en la empresa. Sin embargo, fue posible recuperar este tiempo en los días posteriores. El detalle del avance se observa en Figura 10.

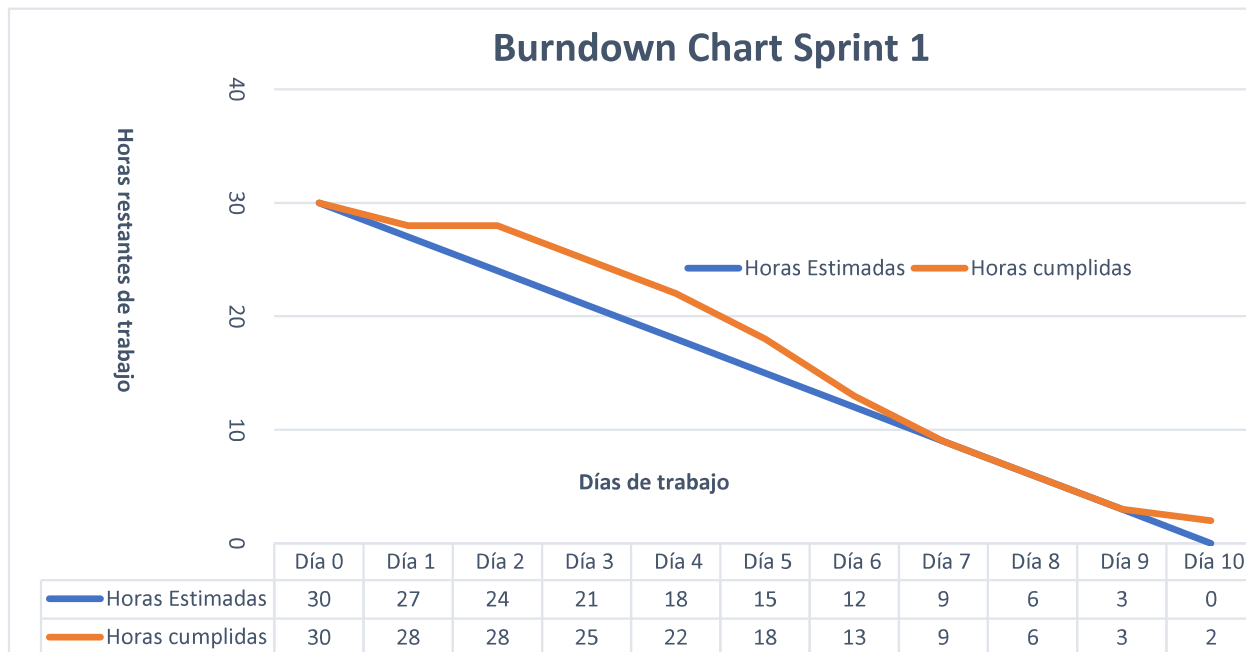


Figura 10: Burndown chart Sprint 1

## 2.4.6. Sprint 2

### Objetivo:

- Realizar transferencias a otras instituciones
- Mostrar la posición consolidada de cuentas, préstamos y pólizas.
- Mostrar el detalle de movimientos de cada cuenta.

### Historias de usuario

El Sprint 2 satisface las historias de usuario HUE01-03, HUE03-01 y HUE03-02, las mismas que suman un total de 22 puntos de esfuerzo a esto se añade un punto más por el esfuerzo de las tareas pendientes del Sprint pasado. Como se ha definido cada Sprint tendrá una duración de 2 semanas en las que hay que terminar los requerimientos contemplados como lo muestra la Tabla 11, el detalle completo de las historias de usuario para el Sprint 2 se presentan en el Anexo V.



Tabla 11: Historias de usuario Sprint 2

<b>ID</b>	<b>TÍTULO</b>	<b>ESFUERZO</b>	<b>ESTADO</b>
HUE01-01	Transferencias entre cuentas propias	0.5	Por implementar
HUE01-02	Transferencias a terceros en la misma institución	0.5	Por implementar
HUE01-03	Transferencias a otras instituciones	7	Por implementar
HUE03-01	Posición consolidada de cuentas, préstamos y pólizas	10	Por implementar
HUE03-02	Detalle de movimientos de cada cuenta	5	Por implementar

### **Sprint 2 Backlog**

Una vez definidas las historias del Sprint, es necesario definir las tareas para su cumplimiento, el resumen del Backlog del Sprint 2 se muestra en la Tabla 12, el detalle completo del Sprint 2 Backlog se encuentra en el Anexo VI.

Tabla 12: Sprint 2 Backlog

<b>H U</b>	<b>ID TAREA</b>	<b>TAREAS</b>
HUE01-01	TR1-11	Crear método para controlar el ingreso de los caracteres alfanuméricos
HUE01-02	TR1-19	Crear método para controlar el ingreso de los caracteres alfanuméricos

Tabla 12: Sprint 2 Backlog

H U	ID TAREA	TAREAS
HUE01-03	TR2-01	Crear el modelo de datos para transferencias a otras instituciones. Crear el servicio para el consumo del <i>API REST</i> . Crear el componente para las transferencias a otras instituciones. -Archivo de lógica del negocio en <i>Typescript</i> . -Vista en Angular creada. Crear y ejecutar las pruebas unitarias del componente.
	TR2-02	Crear diccionario de palabras en inglés y español para la vista
	TR2-03	En la vista crear el input que solo permita el ingreso de números.
	TR2-04	Crear el botón "Solicitar" con el método para enviar una solicitud de token al servidor del <i>Core</i> .
	TR2-05	Crear el método que consuma el servicio <i>REST</i> que valida la temporalidad del Token que se va a usar en el envío de la transferencia.
	TR2-06	Crear el botón "x" para limpiar todos los campos que contiene la pantalla.
	TR2-07	Crear el método que consuma el servicio <i>REST</i> que envía la transferencia a otras instituciones. Crear el mensaje que devuelve el servidor al intentar realizar la transferencia.
	TR2-08	
	TR2-09	Crear el componente para el menú lateral -Archivo de lógica del negocio en <i>Typescript</i> . -Vista en Angular creada.
HUE03-02	TR2-16	Crear el modelo de datos para el detalle de movimientos. Crear el servicio para el consumo del <i>API REST</i> . Crear el componente para el detalle de movimientos. -Archivo de lógica del negocio en <i>Typescript</i> . -Vista en Angular creada. Crear y ejecutar las pruebas unitarias del componente.
	TR2-17	Crear diccionario de palabras en inglés y español para la vista

Tabla 12: Sprint 2 Backlog

HUE03-01	TR2-10	<p>Crear el modelo de datos para la posición consolidada.                  Crear el servicio para el consumo del <i>API REST</i>.                  Crear el componente para la posición consolidada.                  -Archivo de lógica del negocio en <i>Typescript</i>.                  -Vista en Angular creada.                  Crear y ejecutar las pruebas unitarias del componente.</p>
	TR2-11	<p>Crear diccionario de palabras en inglés y español para la vista</p>
	TR2-12	<p>Crear el método que consuma el servicio <i>REST</i> que recibe la información de cada cuenta</p>
	TR2-13	<p>Crear el método que consuma el servicio <i>REST</i> que recibe la información de cada póliza</p>
	TR2-14	<p>Crear el método que consuma el servicio <i>REST</i> que recibe la información de cada préstamo</p>
	TR2-15	<p>Crear el método que controla el CSS para pintar el préstamo de acuerdo con su parametrización.</p>

## Ejecución del Sprint

### Menú lateral

Esta desarrollado para ocultarse/mostrarse al deslizarse de izquierda a derecha. Toma el nombre del usuario de acuerdo con los datos de sesión y cuenta con las opciones de transferencias, la Figura 11 muestra el menú desarrollado. Al seleccionar una opción se abre su respectiva pantalla.

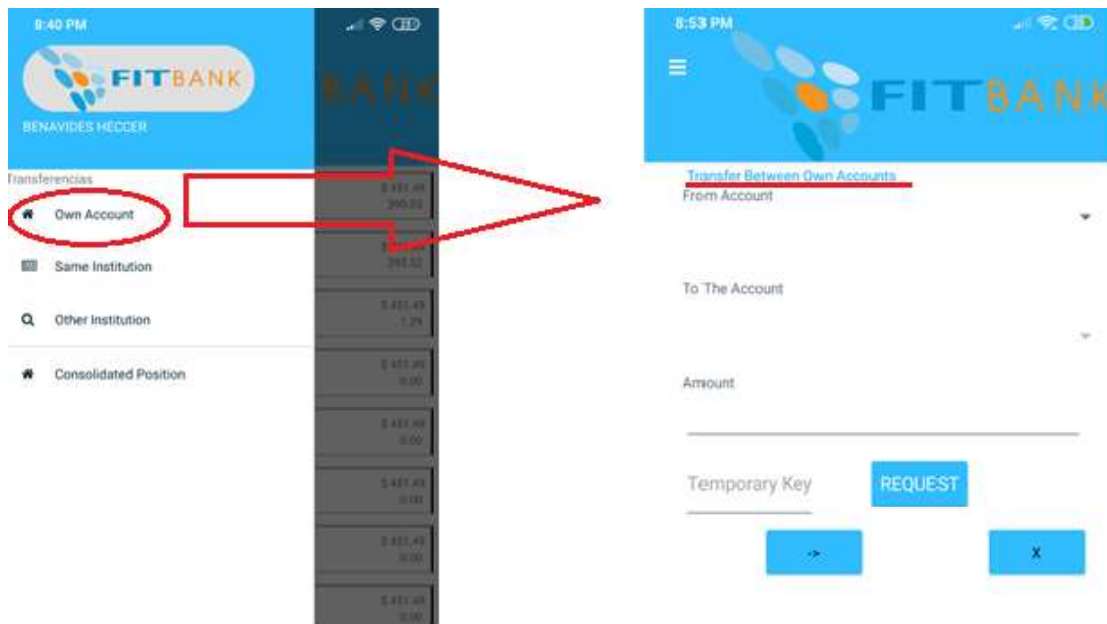


Figura 11: Menú lateral

### Posición consolidada

Muestra la información básica de cada cuenta, préstamo y póliza. En la Figura 12 se puede ver el método que consume el servicio que trae esta información al componente.

```

ngOnInit(): void {
  this.token = Config.token;
  this.homeService.obtainData(Config.user).subscribe(
    (resp: any) => {
      this.cuentasGeneral = resp; // .filter(value=>value.tipo==="ahorros");
      this.cuentasVista = this.cuentasGeneral.filter((value) => value.tipo === "ahorros");
      this.cuentasPlazo = this.cuentasGeneral.filter((value) => value.tipo === "póliza");
      this.cuentasPrestamo = this.cuentasGeneral.filter((value) => value.tipo === "préstamo");
      this.isBusy = false;
    }, (error) => Config.mostrarError(error)
  );
}

```

Figura 12: Método para posición consolidada

En el caso de los préstamos se muestra en color rojo aquellos que se encuentren en estado vencido, la Figura 13 muestra un ejemplo.

Consolidated Position	
Account: *****0031	Withheld Balance: \$ 0.00
	Total Balance: \$ 1,429.43
Term Accounts	
Product: DEPOSITOS A PLAZO FIJO	Account: *****0529
Capital: \$ 3,150.00	Rate: %
	Expiration: 2018/12/21
Product: DEPOSITOS A PLAZO FIJO	Account: *****0551
Capital: \$ 5,000.00	Rate: %
	Expiration: 2019/03/09
Loan Accounts	
Product: CONSUMO PRIORITARIO	Account: *****0115
Balance: \$ 4,561.37	Balance Amount: \$ 5,000.00
	Rate: %
	Expiration: 2019/03/02
Product: CONSUMO PRIORITARIO	Account: *****0116
Balance: \$ 5,000.00	Balance Amount: \$ 5,000.00
	Rate: %
	Expiration: 2019/03/02
Product: CONSUMO PRIORITARIO	Account: *****0122
Balance: \$ 4,216.56	Balance Amount: \$ 5,000.00
	Rate: %
	Expiration: 2019/03/04
Product: CONSUMO PRIORITARIO	Account: *****0125
Balance: \$ 10,000.00	Balance Amount: \$ 10,000.00
	Rate: %
	Expiration: 2020/02/27
Product: CONSUMO PRIORITARIO	Account: *****0126
Balance: \$ 5,000.00	Balance Amount: \$ 5,000.00
	Rate: %
	Expiration: 2019/03/02

Figura 13: Pantalla posición consolidada

## Detalle de movimientos

La figura 14 muestra el método que recibe la información de cada cuenta desde el consumo del servicio.

```

ngOnInit() {
  this.cuentaVista = CuenvistaService.cuentaVistaSeleccionada;
  this.cuentaService.obtainData(this.cuentaVista.id).subscribe(
    (result: any) => {
      this.detalleCuenta = result;
      this.isBusy = false;
    }, (error) => Config.mostrarError(error)
  );
}

```

Figura 14: Método de consulta de información

## Revisión del Sprint 2

Finalmente, se puede concluir que, si se alcanzó el objetivo del Sprint 2 que consistió en realizar las transferencias a otras instituciones, la posición consolidada del cliente y el detalle de movimientos de cada cuenta.

## Pruebas de aceptación

En la Tabla 13, se puede observar los criterios de aceptación que fueron cumplidos, el detalle de los criterios de aceptación está en el Anexo VII.

Tabla 13: Criterios de aceptación

ID	CRITERIOS DE ACEPTACIÓN	CHECK LIST
HUE01-01	Para realizar transferencias propias la aplicación debe permitir ingresar el monto solo en números.	SI
HUE01-02	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir ingresar el monto solo en números.	SI
HUE01-03	La aplicación tendrá una pantalla para realizar transferencias a otras instituciones, donde se pueda escoger la cuenta propia desde la que se va a realizar la transferencia, el ingreso del monto, botones para solicitud del token, el envío de la transferencia o cancelación de esta.	SI
	Los textos en la pantalla de transferencias a otras instituciones se deberán visualizar con el idioma del dispositivo	SI
	Para realizar transferencias a otras instituciones la aplicación debe permitir ingresar el monto solo en números.	SI
	Para realizar transferencias a otras instituciones la aplicación debe permitir pedir un Token, mismo que se envía a un correo del usuario.	SI
	Para realizar transferencias a otras instituciones la aplicación debe permitir validar el Token ingresado.	SI
	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir limpiar los campos.	SI
	En el caso que la transferencia a otras instituciones no pueda ser ejecutada la aplicación debe mostrar el mensaje que envía el sistema.	SI
	En el caso que la transferencia a otras instituciones se ejecute satisfactoriamente la aplicación debe mostrar el mensaje que envía el sistema.	
	La aplicación debe contar con un menú lateral, el mismo deberá ocultarse/mostrarse al deslizarse de izquierda a derecha. Mostrará el nombre del usuario y contará con las opciones de: 1. Cuentas propias, 2. Terceros misma institución, 3. Otra institución y 4. Posición consolidada. Cada opción llevará a su respectiva pantalla.	SI

Tabla 13: Criterios de aceptación

ID	CRITERIOS DE ACEPTACIÓN	CHECK LIST
HUE03-01	La aplicación contará con la pantalla principal de posición consolidada en la que se debe mostrar cada cuenta, préstamo y póliza	SI
	Los textos en la pantalla de posición consolidada se deberán visualizar con el idioma del dispositivo	SI
	En el caso de la posición consolidada se debe mostrar la información por cada cuenta del cliente	SI
	En el caso de la posición consolidada se debe mostrar la información de cada póliza	SI
	En el caso de la posición consolidada se debe mostrar la información de cada préstamo	SI
	En el caso de la posición consolidada se debe mostrar en rojo los préstamos que se encuentren en estado vencido.	SI
HUE03-02	La aplicación contará con la pantalla de detalle de movimientos en la que se debe mostrar cada movimiento de una cuenta.	SI
	Los textos en la pantalla de detalles de movimientos se deberán visualizar con el idioma del dispositivo	SI

### Adaptación del *Product Backlog*

La aplicación estaba desarrollándose bajo la versión de *Nativescript* 5.0 pero al final de este Sprint, la comunidad de desarrollo liberó la versión *Nativescript* 6.0. Se decidió esperar a completar con todas las historias que se encuentran mencionadas en el *Release Planning* para luego proceder con la migración. Para dicho proceso se crea una nueva historia de usuario denominada “Migración a *Nativescript* 6.0”. Según se investigó en la documentación de *Nativescript* esta migración se puede automatizar mediante un comando para la consola de *Nativescript CLI*, por tal razón su tiempo de estimación es de máximo 1 hora, lo cual no debería tener cambios significativos en el Sprint. Por ende, se procedió a adaptar al *Product backlog* como se enseña en la Tabla 14.

Tabla 14: Product Backlog

PRODUCT BACKLOG					
HISTORIA ÉPICA DE USUARIO	H U	HISTORIA DE USUARIO	DIMENSIÓN / ESFUERZO (Puntos de historia)	PRIORIDAD	ESTADO
HUE02	HUE02-01	Validación de nombre de usuario, imagen y contraseña	9	4	Terminado
HUE01	HUE01-01	Transferencias entre cuentas propias	6	4	Terminado
	HUE01-02	Transferencias a terceros en la misma institución	6	4	Terminado
	HUE01-03	Transferencias a otras instituciones	6	4	Terminado
HUE03	HUE03-01	Posición consolidada de cuentas, préstamos y pólizas	10	3	Terminado
	HUE03-02	Detalle de movimientos de cada cuenta	5	3	Terminado
	HUE03-03	Tabla de amortización de préstamos y pólizas	5	3	Por implementar
HUE04	HUE04-01	Información de cajeros	4	3	Por implementar
	HUE04-02	Dirección de oficinas	4	3	Por implementar
	HUE04-03	Productos de la institución financiera	4	3	Por implementar
	HUE04-04	Servicios de la institución financiera	4	3	Por implementar
HUE05	HUE05-01	Migración a <i>Nativecsript</i> 6.0	1	3	Por implementar



## Retrospectiva del Sprint 2

La tarea de realizar el menú lateral llevo más tiempo del estimado debido a que no realizaba el ruteo, se tuvo que emplear un SPIKE [32] para leer la documentación de *Nativescript*, también en la elaboración de la posición consolidada se necesitaba un elemento que muestre la información en modo de carta, pero el *framework* no tiene desarrollado un elemento con estas características, así que se tuvo que adaptar los estilos para que tome la apariencia deseada. Pese a estos alargues si se pudo completar con el tiempo estimado. El detalle del avance se observa en la Figura 15.

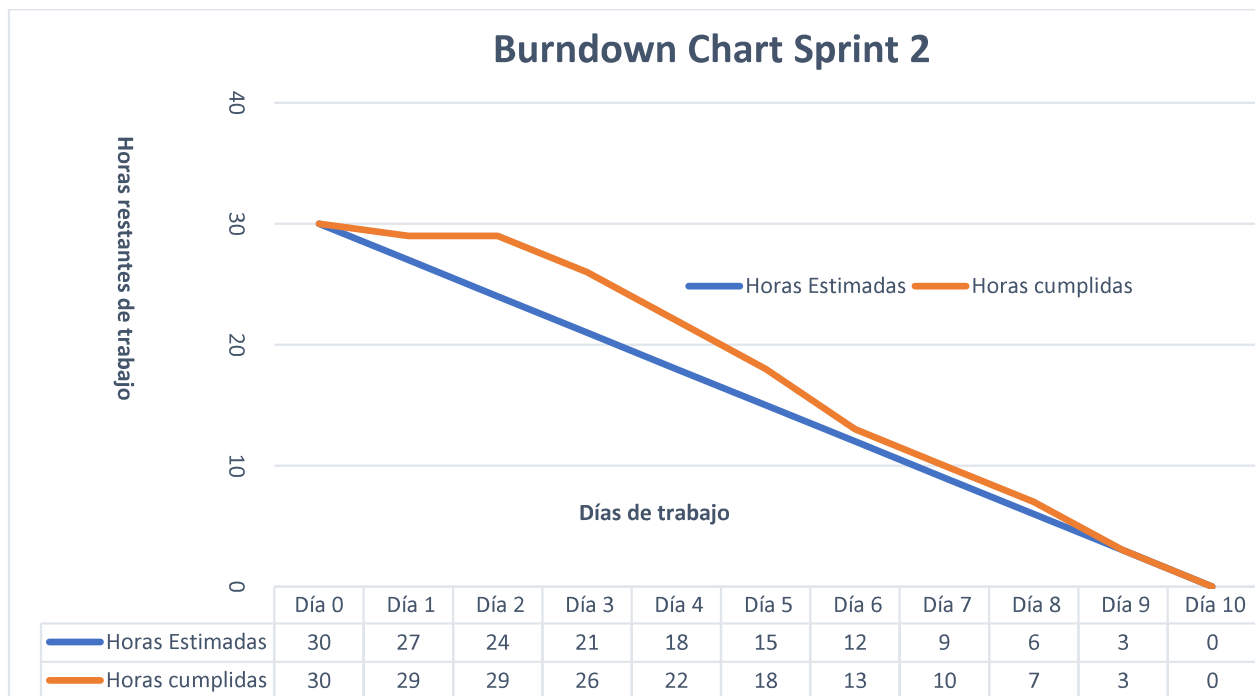


Figura 15: Burndown chart Sprint 2

### 2.4.7. Sprint 3

#### Objetivo:

- Mostrar la tabla de amortización de préstamos y pólizas
- Mostrar la información de cajeros, oficinas, productos y servicios de la institución financiera.
- Migrar a *Nativescript* 6.0

## Historias de usuario

El Sprint 3 satisface las historias de usuario HUE03-03, HUE04-01, HUE04-02, HUE04-03, HUE04-04 y HUE05-01, las mismas que suman un total de 22 puntos de esfuerzo. Como se ha definido cada Sprint tendrá una duración de 2 semanas en las que hay que terminar los requerimientos contemplados como lo muestra la Tabla 15 el detalle completo de las historias de usuario para el Sprint 3 se presentan en el Anexo VIII.

Tabla 15: Historias de usuario Sprint 3

ID	TÍTULO	ESFUERZO	ESTADO
HUE03-03	Tabla de amortización de préstamos y pólizas	5	Por implementar
HUE04-01	Información de cajeros	4	Por implementar
HUE04-02	Dirección de oficinas	4	Por implementar
HUE04-03	Productos de la institución financiera	4	Por implementar
HUE04-04	Servicios de la institución financiera	4	Por implementar
HUE05-01	Migración a <i>Nativescript</i> 6.0	1	Por implementar

## Sprint 2 Backlog

Una vez definidas las historias del Sprint, es necesario definir las tareas para su cumplimiento, el resumen del Backlog del Sprint 3 se muestra en la Tabla 16, el detalle completo del Sprint 3 Backlog se encuentra en el Anexo IX.

Tabla 16: Sprint 3 Backlog

H U	ID TAREA	TAREAS
HUE03-03	TR3-01	Crear el modelo de datos para el detalle de pagos de un préstamo. Crear el servicio para el consumo del <i>API REST</i> . Crear el componente para el detalle de pagos de un préstamo. -Archivo de lógica del negocio en <i>Typescript</i> . -Vista en Angular creada. Crear y ejecutar las pruebas unitarias del componente.
	TR3-02	Crear diccionario de palabras en inglés y español para la vista
	TR3-03	Crear el modelo de datos para el detalle de pagos de una póliza. Crear el servicio para el consumo del <i>API REST</i> . Crear el componente para el detalle de pagos de una póliza. -Archivo de lógica del negocio en <i>Typescript</i> . -Vista en Angular creada. Crear y ejecutar las pruebas unitarias del componente.
	TR3-04	Crear diccionario de palabras en inglés y español para la vista
HUE04-01	TR3-05	Crear los botones con las propiedades que permitan mostrarse y ocultarse sobre la pantalla de ingreso a la aplicación
	TR3-06	Crear el método que dirija a la pantalla de cajeros luego de presionar el botón en el menú flotante.
		Crear el modelo de datos para la información de cajeros. Crear el servicio para el consumo del <i>API REST</i> . Crear el componente para la información de cajeros. -Archivo de lógica del negocio en <i>Typescript</i> . -Vista en Angular creada. Crear y ejecutar las pruebas unitarias del componente.
TR3-07	Crear el método que consuma el servicio que muestra la información de los cajeros de acuerdo con la ciudad.	

Tabla 16: Sprint 3 Backlog

H U	ID TAREA	TAREAS
HUE04-02		Crear el método que dirija a la pantalla de oficinas luego de presionar el botón en el menú flotante.
	TR3-08	Crear el modelo de datos para la información de oficinas. Crear el servicio para el consumo del <i>API REST</i> . Crear el componente para la información de oficinas. -Archivo de lógica del negocio en <i>Typescript</i> . -Vista en Angular creada. Crear y ejecutar las pruebas unitarias del componente.
	TR3-09	Crear el método que consuma el servicio que muestra la información de las oficinas de acuerdo con la ciudad.
HUE04-03		Crear el método que dirija a la pantalla de Productos luego de presionar el botón en el menú flotante.
	TR3-10	Crear el modelo de datos para la información de productos. Crear el servicio para el consumo del <i>API REST</i> . Crear el componente para la información de productos. -Archivo de lógica del negocio en <i>Typescript</i> . -Vista en Angular creada. Crear y ejecutar las pruebas unitarias del componente.
	TR3-11	Crear el método que consuma el servicio que muestra la información de cada tipo de producto.
HUE04-04		Crear el método que dirija a la pantalla de Servicios luego de presionar el botón en el menú flotante.
	TR3-12	Crear el modelo de datos para la información de servicios. Crear el servicio para el consumo del <i>API REST</i> . Crear el componente para la información de servicios. -Archivo de lógica del negocio en <i>Typescript</i> . -Vista en Angular creada. Crear y ejecutar las pruebas unitarias del componente.
HUE05-01	TR3-13	Ejecutar el comando que propone la documentación de <i>Nativescript</i> y verificar el funcionamiento de la aplicación.

## Ejecución del Sprint

### Tabla de préstamos y pólizas

Para ambos casos se cuenta con una descripción general y el monto, mediante un enlace al final de esta pantalla se puede acceder a la tabla de amortización en el que se puede ver a detalle cada cuota como se muestra en la Figura 16.

Detalle Loan		
Producto:	CONSUMO PRIORITY	
Cuenta:	*****0115	
Balance Actual:	\$ 456,37	
Balance Original:	\$ 400,00	
Tasa:	14,75 %	
Expiración:	2019/03/02	
Date Concesion:		2018-03-07
Original Amount:		\$ 5,000.00
Term:		360
Rate:		14,75 %
Quotas:		12
Due:		2019-03-02
Method:		FRANCES
Quota Pending:		2
Overdue installments:		10
Maximum Payment Date:		2018-05-06
Value Pay:		\$ 451,49
<a href="#">Payment Table</a>		

Payment Table			
Date:	2018/04/06	Value Fees:	\$ 451,49
Fee:	1	Paid Value:	390,03
Date:	2018/05/16	Value Fees:	\$ 451,49
Fee:	2	Paid Value:	50,00
Date:	2018/06/05	Value Fees:	\$ 451,49
Fee:	3	Paid Value:	0,00
Date:	2018/07/05	Value Fees:	\$ 451,49
Fee:	4	Paid Value:	0,00
Date:	2018/08/04	Value Fees:	\$ 451,49
Fee:	5	Paid Value:	0,00
Date:	2018/09/03	Value Fees:	\$ 451,49
Fee:	6	Paid Value:	0,00
Date:	2018/10/03	Value Fees:	\$ 451,49
Fee:	7	Paid Value:	0,00
Date:	2018/11/02	Value Fees:	\$ 451,49
Fee:	8	Paid Value:	0,00
Date:	2018/12/02	Value Fees:	\$ 451,49
Fee:	9	Paid Value:	0,00
Date:	2019/01/01	Value Fees:	\$ 451,49
Fee:	10	Paid Value:	0,00
Date:	2019/01/31	Value Fees:	\$ 451,49
Fee:	11	Paid Value:	0,00
Date:	2019/03/02	Value Fees:	\$ 451,55
Fee:	12	Paid Value:	0,00

Figura 16: Pantalla de datos y de tabla de pagos

### Menú flotante

La aplicación también cuenta con un menú flotante que simula estar sobrepuesto en la pantalla de ingreso, en el mismo se encuentra información de la institución financiera, como es la dirección de los cajeros por ciudad, la dirección de las oficinas que posee la institución financiera por ciudad, el detalle de los productos que se ofrecen y finalmente, los servicios que están disponibles para su uso, como se muestra en la Figura 17.

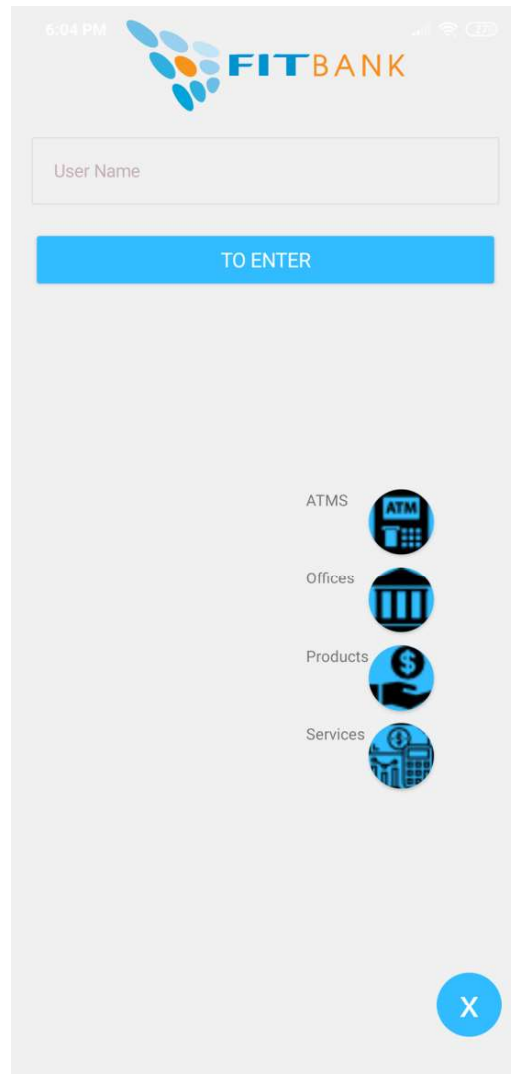


Figura 17: Menú Flotante

La figura 18 muestra el método que consume el servicio para cargar las oficinas de la institución financiera por ciudad.

```
extractDataOffices() {  
  this.infoOficinasService.getDataOffices()  
    .subscribe((result: any) => {  
      this.arrOffices = result;  
  
      console.log(result);  
  
    }, (error) => Config.mostrarError(error));  
}
```

Figura 18: Carga de información de oficinas

La figura 19 muestra el método que consume el servicio para saber la dirección de la oficina seleccionada.

```
extractDirectionOffice(office: Office) {  
  
  this.infoOficinasService.getDataDirectionOffices(office)  
    .subscribe((result: any) => {  
      this.arrDirectionOffices = result;  
      console.log("resultado: ");  
  
      console.log(result);  
  
    }, (error) => Config.mostrarError(error));  
}
```

Figura 19: Carga de información de dirección de oficinas

Finalmente, se tiene cada una de las pantallas de información, la figura 20 muestra la información de las oficinas por ciudad y el detalle de direcciones de cada una.

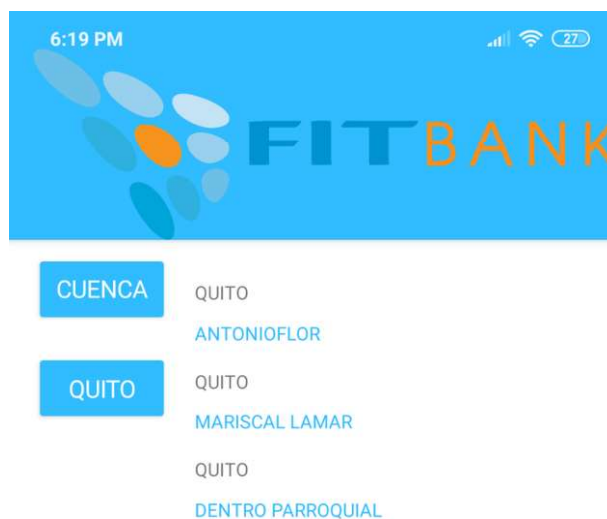


Figura 20: Pantalla de información de las oficinas

### Revisión del Sprint 3

Finalmente, se puede concluir que, si se alcanzó el objetivo del Sprint 3.

### Pruebas de aceptación

En la Tabla 17, se puede observar los criterios de aceptación que fueron cumplidos, el detalle de los criterios de aceptación está en el Anexo X.

Tabla 17: Revisión de criterios de aceptación del Sprint 3

H U	CRITERIOS DE ACEPTACIÓN	CHECKLIST
HUE03-03	La aplicación contará con la pantalla de tabla de pagos de préstamos, para llegar a la misma se accederá desde la posición consolidada al ver el detalle de un préstamo.	SI
	Los textos en la pantalla de tabla de pagos de préstamos se deberán visualizar con el idioma del dispositivo	SI
	La aplicación contará con la pantalla de tabla de pagos de pólizas, para llegar a la misma se accederá desde la posición consolidada al ver el detalle de una póliza.	SI
	Los textos en la pantalla de tabla de pagos de pólizas se deberán visualizar con el idioma del dispositivo	SI
HUE04-01	La aplicación tendrá un menú flotante en la parte inferior derecha, en la página de inicio, con cuatro opciones: <i>ATMs</i> , oficinas, productos y servicios. El menú se mostrará y ocultará al presionar.	SI
	El menú flotante tendrá la opción de <i>ATMs</i> , al presionar en este se mostrará una pantalla con botones correspondientes a las ciudades donde existen cajeros de la institución financiera.	SI
	Al presionar en una ciudad de la pantalla de <i>ATMs</i> se muestran las direcciones de los cajeros disponibles	SI
HUE04-02	El menú flotante tendrá la opción de Oficinas, al presionar en este se mostrará una pantalla con botones correspondientes a las ciudades donde existen oficinas de la institución financiera.	SI
	Al presionar en una ciudad de la pantalla de Oficinas se muestra las direcciones de las oficinas existentes	SI



Tabla 17: Revisión de criterios de aceptación del Sprint 3

H U	CRITERIOS DE ACEPTACIÓN	CHECKLIST
HUE04-03	El menú flotante tendrá la opción de Productos, al presionar en este se mostrará una pantalla con botones correspondientes a los tipos de productos que existen en la institución financiera.	SI
	Al presionar en un Producto de la pantalla de Productos se muestre los diferentes productos que ofrece la institución financiera	SI
HUE04-04	El menú flotante tendrá la opción de Servicios, al presionar en este se mostrará una pantalla con botones correspondientes a los tipos de servicios que existen en la institución financiera.	SI
HUE05-01	La aplicación contará con la última versión del <i>framework</i>	SI

### Adaptación del *Product Backlog*

En la Tabla 18 se puede observar cómo se cumplieron todas las tareas planteadas de acuerdo con cada uno de los *Sprints* planeados.

Tabla 18: Product Backlog tras la finalización del proyecto

PRODUCT BACKLOG					
HISTORIA ÉPICA DE USUARIO	H U	HISTORIA DE USUARIO	DIMENSIÓN / ESFUERZO (Puntos de historia)	PRIORIDAD	ESTADO
HUE02	HUE02-01	Validación de nombre de usuario, imagen y contraseña	9	4	Terminado
HUE05	HUE05-01	Migración a <i>Nativecsript</i> 6.0	1	3	Terminado

Tabla 18: Product Backlog tras la finalización del proyecto

HISTORIA ÉPICA DE USUARIO	H U	HISTORIA DE USUARIO	DIMENSIÓN / ESFUERZO	PRIORIDAD	ESTADO
HUE01	HUE01-01	Transferencias entre cuentas propias	6	4	Terminado
	HUE01-02	Transferencias a terceros en la misma institución	6	4	Terminado
	HUE01-03	Transferencias a otras instituciones	6	4	Terminado
HUE03	HUE03-01	Posición consolidada de cuentas, préstamos y pólizas	10	3	Terminado
	HUE03-02	Detalle de movimientos de cada cuenta	5	3	Terminado
	HUE03-03	Tabla de amortización de préstamos y pólizas	5	3	Terminado
HUE04	HUE04-01	Información de cajeros	4	3	Terminado
	HUE04-02	Dirección de oficinas	4	3	Terminado
	HUE04-03	Productos de la institución financiera	4	3	Terminado
	HUE04-04	Servicios de la institución financiera	4	3	Terminado

### Retrospectiva del Sprint 3

Debido a que la funcionalidad que se debía implementar en este Sprint era muy parecida en estructura a otros componentes, el tiempo de estimación por historia de usuario fue más que suficiente para su desarrollo. Finalmente, la migración no causó inconvenientes y la prueba del correcto funcionamiento de cada componente se llevó con éxito. En la última reunión de retrospectiva, no se tuvo ninguna observación ya que se logró cumplir con todo el trabajo planificado dentro del tiempo establecido. El detalle del avance se observa en la Figura 21.

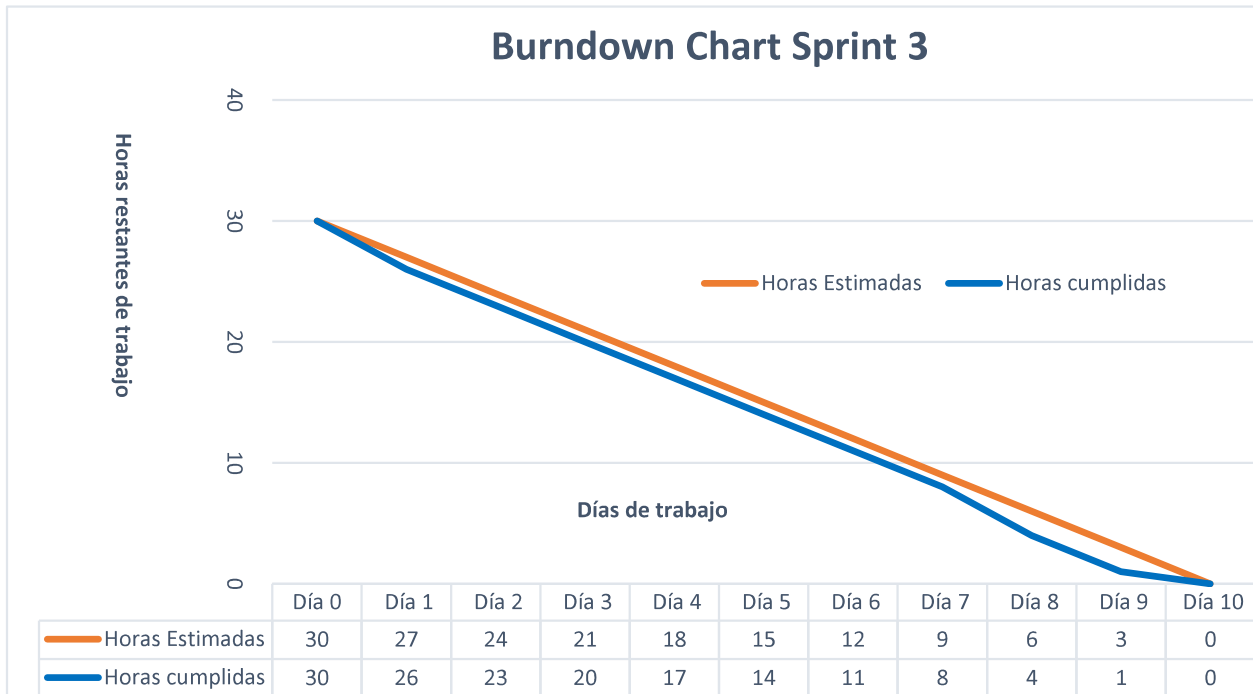


Figura 21: Burndown Chart Sprint 3

### 3. RESULTADOS Y DISCUSIÓN

#### 3.1. Producto Final

El producto final de este proyecto consta de ocho secciones principales que se detallarán a continuación.

##### 3.1.1. Pantalla de ingreso a la aplicación

- Prototipo de pantalla de ingreso a la aplicación

La pantalla de ingreso a la aplicación deberá contener los elementos que se encuentran detallados en la Figura 22:

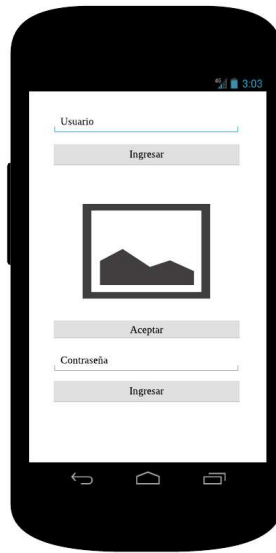


Figura 22 Pantalla Inicial

La pantalla inicial deberá contener un ingreso de texto para el usuario, además de una sección de imágenes en la que se selecciona la registrada por el usuario en la aplicación web, podrá también ingresar su contraseña de manera oculta para que no se vea la información ingresada.

- **Resultado Pantalla Inicial**

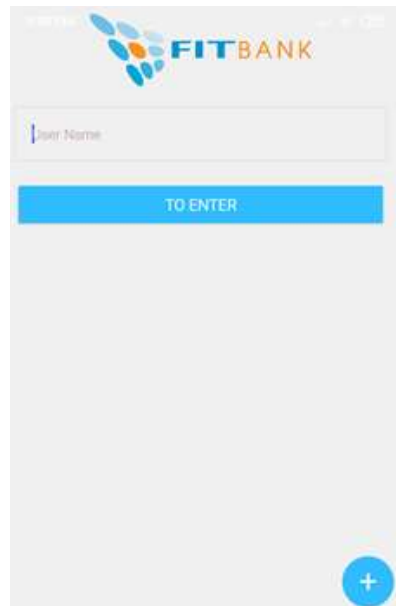


Figura 23 Pantalla Inicial Usuario

En la Figura 23 se puede verificar que se tiene un campo para que se ingrese el usuario, Además de un botón con el cual se realizara una validación de que el usuario exista registrado en el sistema.

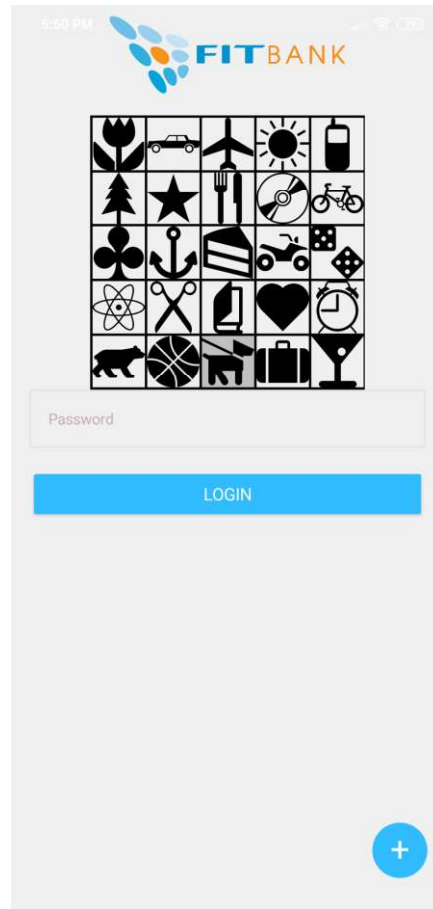


Figura 24 Pantalla Inicial con Imágenes

En la Figura 24 se puede verificar que se despliega una galería de imágenes de las cuales el usuario puede escoger la que él había registrado anteriormente en la página web. El usuario también dispone de un campo para ingresar su contraseña, el cual oculta la información que ingresa el usuario. Existe un botón para verificar que la información que ingreso sea correcta y poder ingresar a la información del usuario.

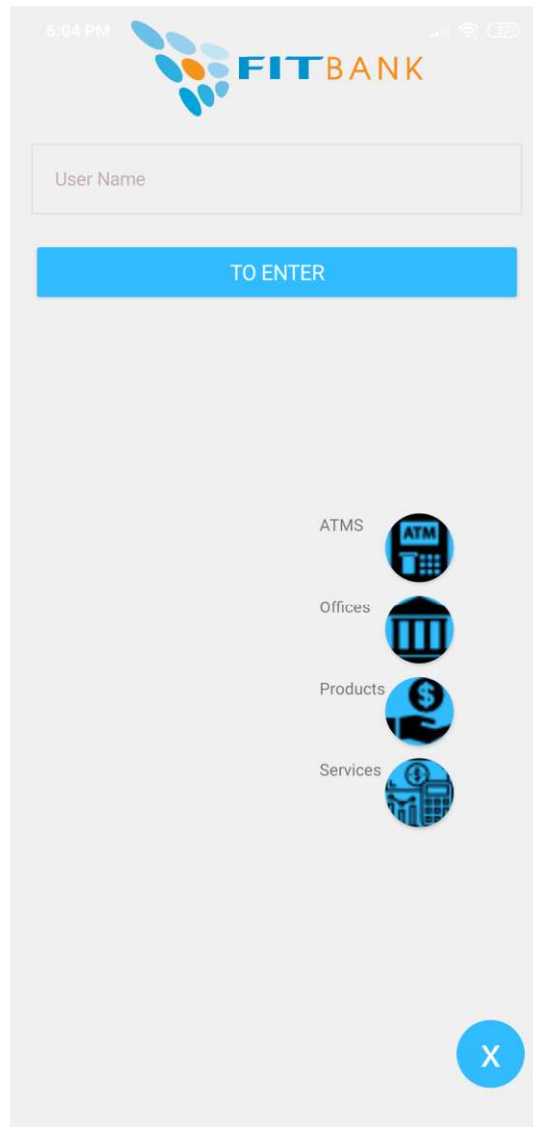


Figura 25 Pantalla Inicial Botones Información

En la Figura 25 se puede verificar la existencia de cuatro botones. Los botones se despliegan después de seleccionar el botón flotante ubicado en la parte inferior derecha de la pantalla inicial. Los botones nos permiten ingresar a las pantallas que mostraran información de la entidad financiera.

### 3.1.2. Pantallas Informativas de la Entidad Financiera

- **Prototipo de pantallas informativas de la entidad financiera**

Las pantallas informativas de la entidad financiera deberán contener los elementos detallados en la Figura 26:

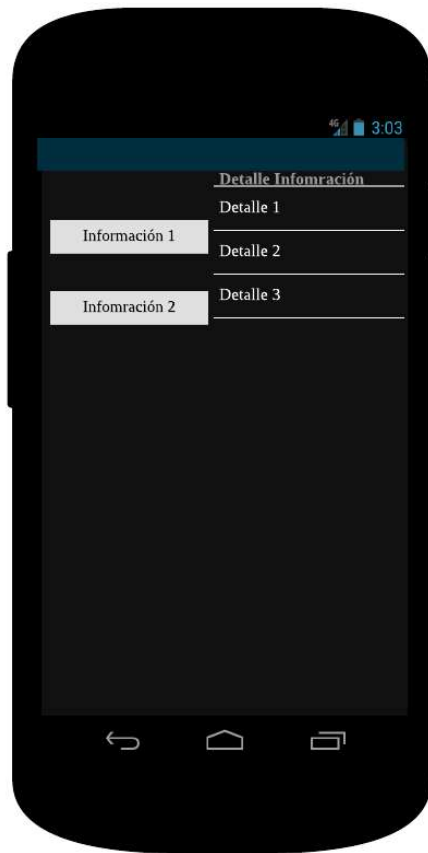


Figura 26 Pantallas Informativas

Las pantallas informativas mostraran un listado de los ítems que dispone la entidad financiera. Cada ítem de información de ser necesario podrá desplegar uno o varios detalles de este.

- **Resultado pantallas informativas**
  - **Pantalla Cajeros**

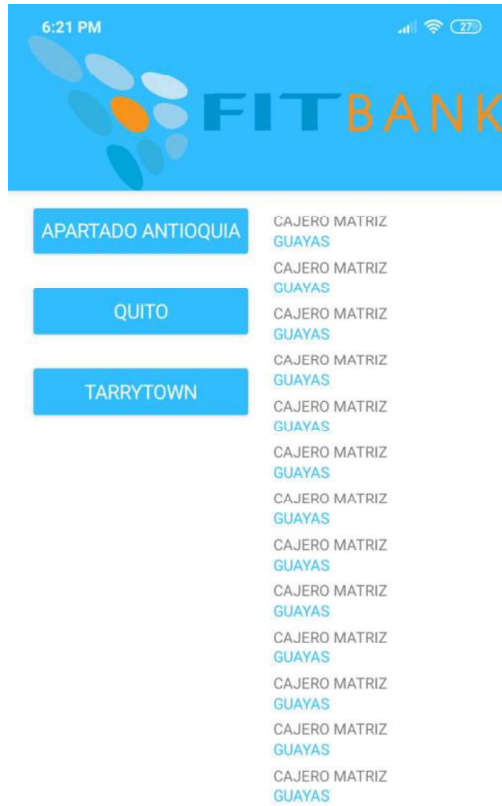


Figura 27 Pantalla Informativa Cajeros

En la Figura 27 se muestra la pantalla informativa de los cajeros, la cual mostrará las ciudades en donde la entidad financiera tiene disponibilidad de ellos. Al seleccionar una ciudad desplegará las direcciones de los cajeros que se tiene registrados en el sistema.

- **Pantalla Oficinas**

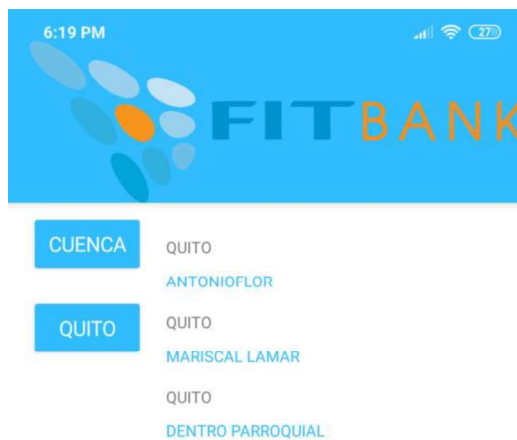


Figura 28 Pantalla Informativa Oficinas



En la Figura 28 se muestra la pantalla informativa de oficinas, la cual desplegará las ciudades en donde dispone de estas la entidad financiera. Al seleccionar la ciudad nos desplegará a detalle las oficinas disponibles registradas en el sistema.

- **Pantalla Productos**



Figura 29 Pantalla Informativa Productos

En la Figura 29 se muestra la pantalla informativa de productos, la cual mostrará los módulos que ofrece la entidad financiera. Al seleccionar un módulo se desplegarán los productos que están registrados en el sistema.

○ **Pantalla Servicios**

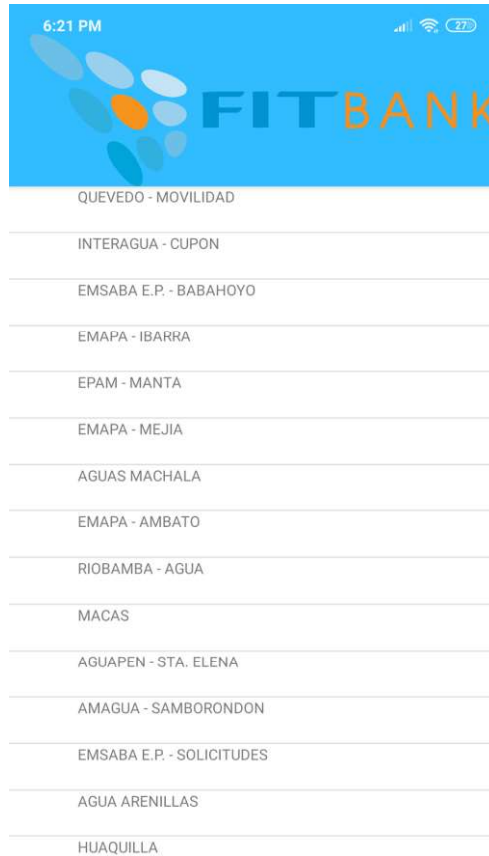


Figura 30 Pantalla Informativa Servicios

En la Figura 30 se muestra la pantalla informativa de los servicios, la cual mostrará una lista de estos que ofrece la entidad financiera.

**3.1.3. Pantalla de información consolidada**

- **Prototipo pantalla de información consolidada**

La pantalla de información consolidada del usuario deberá contener los elementos detallados en la Figura 31:

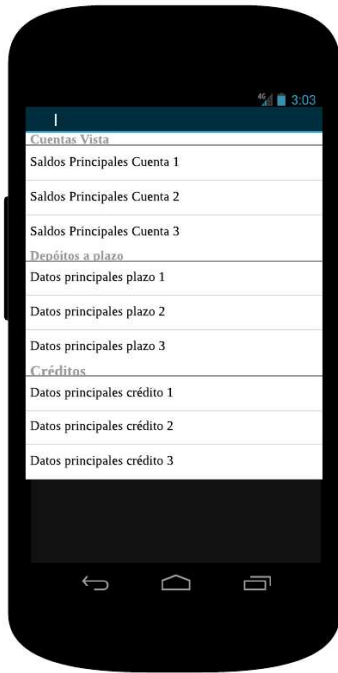


Figura 31 Pantalla de Información Consolidada

La pantalla de información consolidada del usuario deberá presentar cada una de las cuentas que dispone. Las cuentas se presentarán agrupadas por cada módulo. En cada cuenta se presentará la información más relevante de la misma.

- **Resultado pantalla de información consolidada**

7:00 PM

FITBANK

Consolidated Position

View Accounts

Product:	AHORROS CORRIENTE		
Account:	****1111	Available Balance:	\$ 30,551.66
		Withheld Balance:	\$ 1,200.00
		Total Balance:	\$ 31,973.66

Product:	CERTIFICADOS APORTACION		
Account:	****1511	Available Balance:	\$ 2,500.00
		Withheld Balance:	\$ 0.00
		Total Balance:	\$ 2,500.00

Product:	CUENTA BASICA		
Account:	*****0072	Available Balance:	\$ 90.00
		Withheld Balance:	\$ 0.00
		Total Balance:	\$ 90.00

Product:	CUENTA BASICA		
Account:	*****0074	Available Balance:	\$ 130.00
		Withheld Balance:	\$ 0.00
		Total Balance:	\$ 130.00

Product:	AHORRO PROGRAMADO		
Account:	*****0030	Available Balance:	\$ 904.06
		Withheld Balance:	\$ 0.00
		Total Balance:	\$ 904.06

Product:	AHORRO PROGRAMADO		
Account:	*****0031	Available Balance:	\$ 1,429.43
		Withheld Balance:	\$ 0.00
		Total Balance:	\$ 1,429.43

Term Accounts

Product:	DEPOSITOS A PLAZO FIJO		
Account:	*****0529	Capital:	\$ 3,150.00
Rate:	%	Expiration:	2018/12/21

Product:	DEPOSITOS A PLAZO FIJO		
Account:	*****0551	Capital:	\$ 5,000.00
Rate:	%	Expiration:	2019/03/09

Loan Accounts

Figura 32 Pantalla de Información Consolidada Resultado

La Figura 32 muestra la pantalla de información consolidada, la cual muestra en grupos las cuentas que tiene el usuario en la entidad financiera. Para cada cuenta se despliega la información más relevante de la misma. Para acceder al detalle de cada cuenta se debe seleccionar sobre la misma.

### 3.1.4. Pantalla detalle de movimientos

- Prototipo de pantalla de detalle de movimientos

La pantalla de detalle de movimientos de cuenta deberá contener los elementos detallados en la Figura 33:

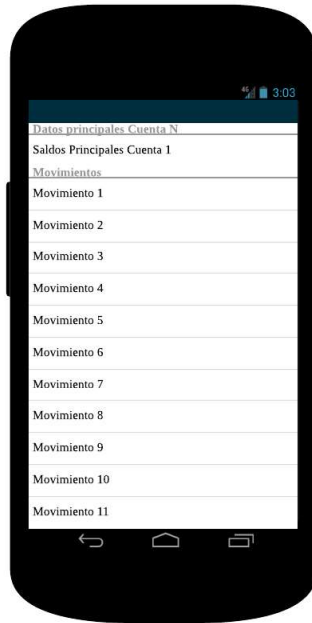


Figura 33 Pantalla Detalle de Movimientos de una Cuenta

La pantalla de detalle de movimientos presentará una cabecera con los datos principales de la cuenta. La pantalla desplegará todos los últimos movimientos de la cuenta, mostrando información de cada uno.

- **Resultado pantalla de detalle de movimientos**

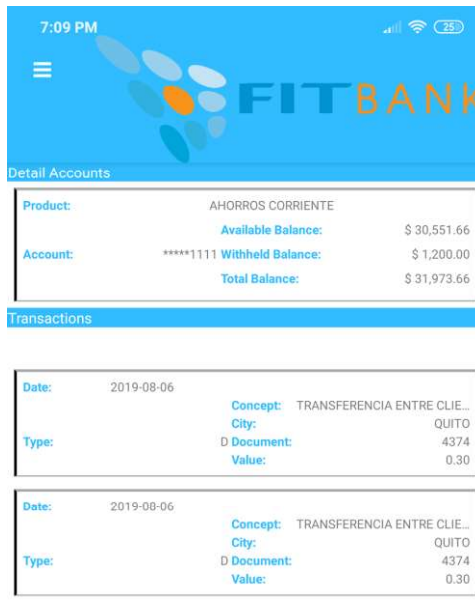


Figura 34 Pantalla de detalle de movimientos de una cuenta resultado

La Figura 34 muestra la pantalla de movimientos, la cual contiene una cabecera en la que muestra los datos principales de la cuenta. La pantalla muestra la lista de movimientos que dispone esta cuenta, cada uno con la información más relevante para conocimiento del usuario.

### 3.1.5. Pantalla de información detallada de cuentas

- **Prototipo de pantalla de información detallada de cuentas**

La pantalla de información detallada de cuentas se utilizará para mostrar a detalle datos de las cuentas de depósitos a plazo fijo y créditos, la cual deberá contener los elementos detallados en la Figura 35:

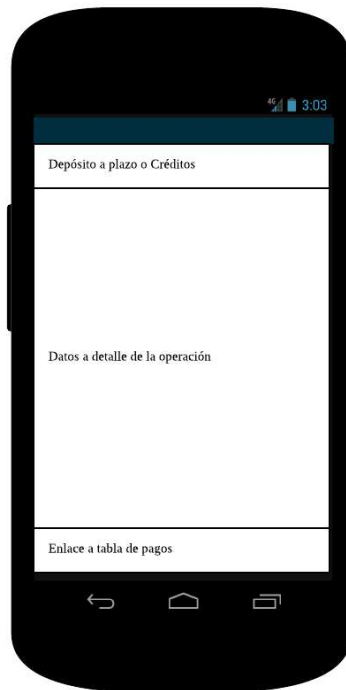


Figura 35 Pantalla de información detallada de cuentas

La pantalla contará con una cabecera en la cual se muestra la información más relevante de la cuenta. La pantalla presentara los datos a detalle de la cuenta sea esta un depósito a plazo o un crédito. La pantalla dispondrá de un enlace a la tabla de pagos de la cuenta.

- Resultado pantalla de información detallada de cuentas
  - Pantalla de información de créditos



Figura 36 Pantalla de detalle de información de créditos

La Figura 36 muestra pantalla de información detallada de un crédito., la cual muestra una cabecera con la información relevante del mismo. A continuación, muestra a detalle las condiciones del crédito. Dispone de un enlace en el borde inferior izquierdo para acceder a la pantalla de tabla de pagos del crédito.

- **Pantalla de información de depósitos a plazo**

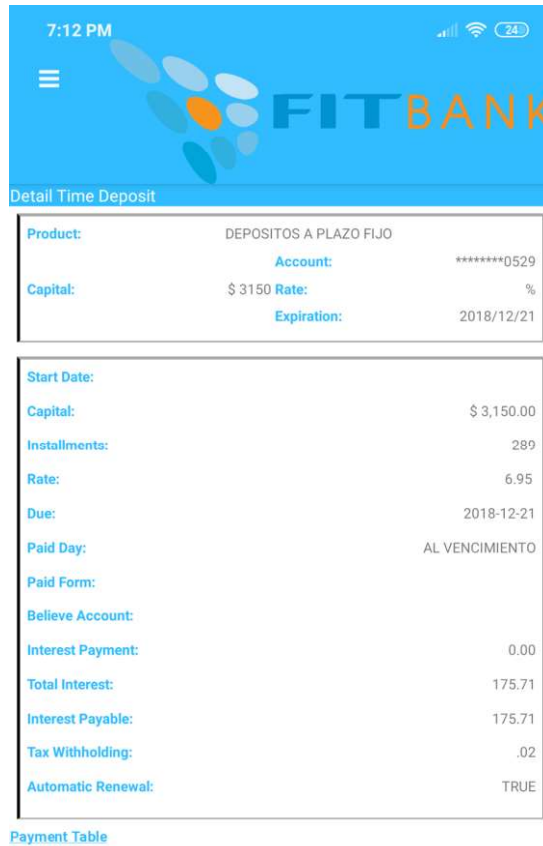


Figura 37 Pantalla detalle de información de depósitos a plazo

La Figura 37 muestra la pantalla de información detallada de un depósito a plazo, la cual muestra una cabecera con la información relevante del mismo. A continuación, muestra a detalle las condiciones del depósito plazo. Dispone de un enlace en el borde inferior izquierdo para acceder a la pantalla de tabla de pagos del depósito a plazo.

### 3.1.6. Pantalla de tabla de pagos

- **Prototipo de pantalla de tabla de pagos**

La pantalla de tabla de pagos deberá contener los elementos detallados en la Figura 38:



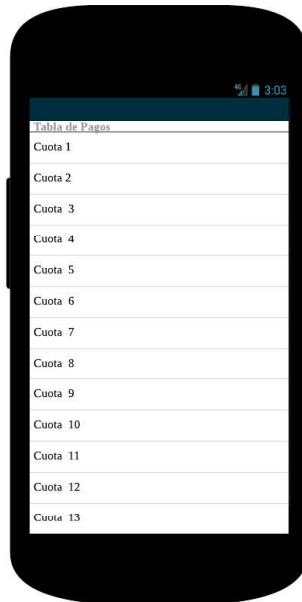


Figura 38 Pantalla de tabla de pagos

La pantalla de tabla de pagos desplegará cada una de las cuotas que disponga la cuenta, sea está un depósito a plazo o un crédito. En cada cuota se mostrará la información más relevante de la misma.

- **Resultados de la pantalla de tabla de pagos**
  - **Pantalla de tabla de pagos de un crédito**

Payments Table			
Date:	2018/04/08	Value Fees:	\$ 451.49
Fee:	1	Paid Value:	390.03
Date:	2018/05/08	Value Fees:	\$ 451.49
Fee:	2	Paid Value:	393.52
Date:	2018/06/07	Value Fees:	\$ 451.49
Fee:	3	Paid Value:	1.29
Date:	2018/07/07	Value Fees:	\$ 451.49
Fee:	4	Paid Value:	0.00
Date:	2018/08/06	Value Fees:	\$ 451.49
Fee:	5	Paid Value:	0.00

Figura 39 Pantalla de tabla de pagos de un crédito

La Figura 39 muestra la pantalla de tabla de pagos de un crédito, la cual muestra todas las cuotas que tenga esta cuenta. La pantalla muestra para cada cuota la fecha de vencimiento, el valor de la cuota, el número de cuota y el valor pagado de la misma.

- **Pantalla de tabla de pagos de un depósito a plazo**

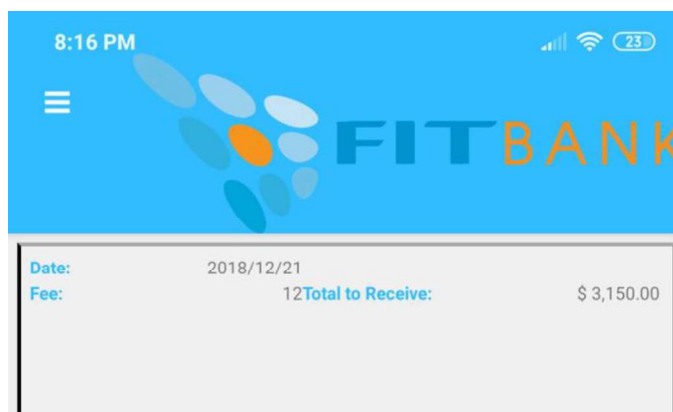


Figura 40 Pantalla de tabla de pagos de un depósito a plazo

La Figura 40 muestra la pantalla de tabla de pagos de un depósito a plazo, la cual muestra todas las cuotas que tenga esta cuenta. La pantalla muestra para cada cuota la fecha de vencimiento, el valor a recibir, el número de cuota.

### 3.1.7. Pantalla de menú lateral

- **Prototipo de pantalla de menú lateral**

La pantalla de menú lateral el cual se puede acceder desde cualquier pantalla excepto la pantalla inicial, deberá contener los elementos detallados en la Figura 41:

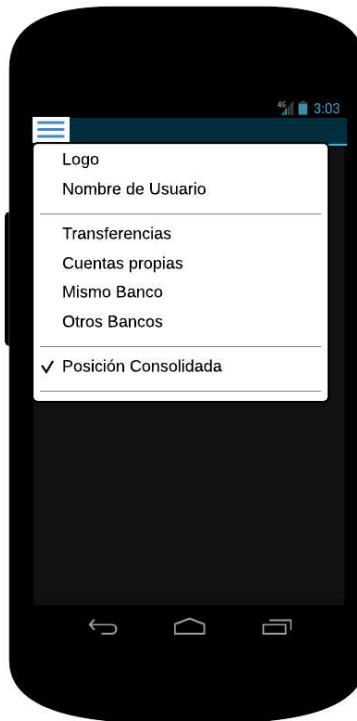


Figura 41 Pantalla de menú lateral

El menú lateral deberá mostrar el logo de la entidad financiera y el nombre del usuario que ingreso al sistema en la parte superior. La pantalla mostrará una opción para cada tipo de trasferencia que se puede realizar. La pantalla mostrara una opción para ingresar a la pantalla de información consolidada.

**Resultado pantalla de menú lateral**

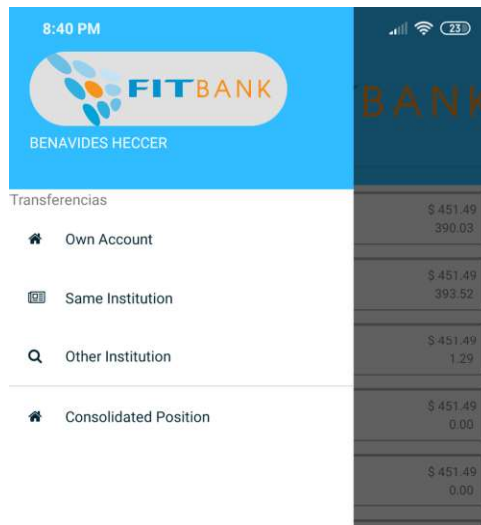


Figura 42 Pantalla de menú lateral resultado

La Figura 42 muestra la pantalla del menú lateral, la cual muestra el logo de la entidad y el nombre del usuario en la parte superior. La pantalla muestra las opciones para el ingreso a los tipos de transferencias y para el ingreso a la pantalla de información consolidada.

### 3.1.8. Pantalla de transferencias de dinero

- Prototipo de pantalla de transferencias de dinero

La pantalla de transferencias de dinero entre cuentas de cualquier tipo deberá contener los elementos detallados en la Figura 43:

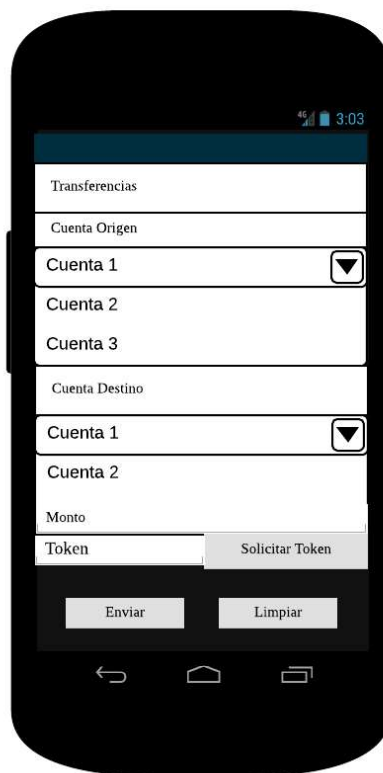


Figura 43 Pantalla de transferencias de dinero

La pantalla de transferencias de dinero deberá presentar un listado de cuentas del usuario para seleccionarla como cuenta de origen. La pantalla presentará un listado de las cuentas registradas como beneficiarios en la página web para seleccionarla como cuenta de destino. La pantalla tendrá un campo para ingresar el monto de la transferencia. La pantalla tendrá un botón para solicitar una clave temporal para realizar la transferencia y tendrá un campo para ingresar esa

clave por parte del usuario. La pantalla tendrá un botón para enviar la transferencia detallada y otro para borrar los datos ingresados.

- **Resultado pantalla de transferencias de dinero**

- **Pantalla de transferencia en cuentas propias**

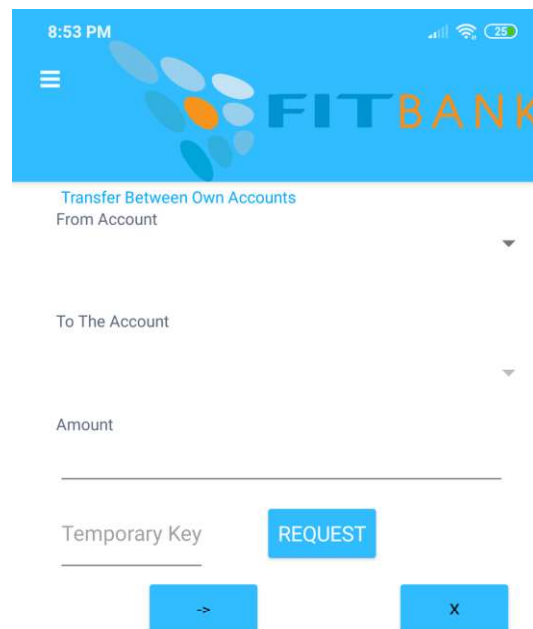


Figura 44 Pantalla de transferencias en cuentas propias

La Figura 44 muestra la pantalla de transferencias en cuentas propias, la cual muestra un listado de cuentas del usuario tanto en las de origen como en las de destino. La pantalla tiene un campo en el cual puede ser ingresado el monto de la transferencia. La pantalla tiene un botón para solicitar la clave temporal para poder realizar la transferencia, además de un campo en donde

puede ingresarla. La pantalla tiene dos botones, uno para realizar el envío de la transferencia y otro para limpiar todos los campos anteriores.

- **Pantalla de transferencias a terceros en cuentas del mismo banco**

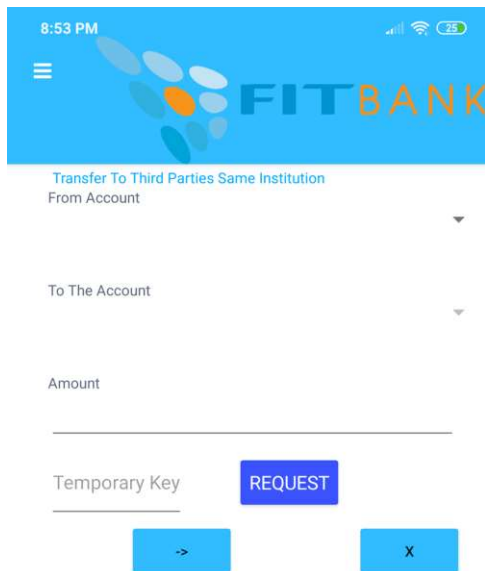


Figura 45 Pantalla de transferencias a terceros en la misma institución

La Figura 45 muestra la pantalla de transferencias a terceros, la cual muestra un listado de cuentas del usuario para que seleccione la cuenta de origen. La pantalla muestra un listado de cuentas de la misma entidad financiera que hayan sido registrados anteriormente en la página web. La pantalla tiene un botón para solicitar la clave temporal para poder realizar la transferencia, además de un campo en donde puede ingresarla. La pantalla tiene dos botones, uno para realizar el envío de la transferencia y otro para limpiar todos los campos anteriores.

- **Pantalla de transferencias a terceros en otras instituciones**

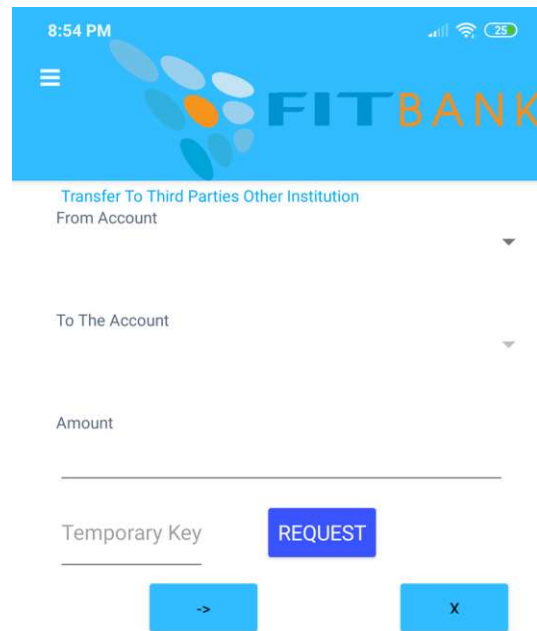


Figura 46 Pantalla de transferencias a terceros en otras instituciones

La Figura 46 muestra la pantalla de transferencias a terceros, la cual muestra un listado de cuentas del usuario para que seleccione la cuenta de origen. La pantalla muestra un listado de cuentas de otras entidades financieras que hayan sido registrados anteriormente en la página web. La pantalla tiene un botón para solicitar la clave temporal para poder realizar la transferencia, además de un campo en donde puede ingresarla. La pantalla tiene dos botones, uno para realizar el envío de la transferencia y otro para limpiar todos los campos anteriores.

### 3.2. Pruebas con usuarios finales

Se realizó una evaluación del producto final sobre la utilidad y facilidad de uso percibidas para la aplicación móvil del Core bancario FITBANK de acuerdo con el modelo *Technology Acceptance Model* (TAM) [33]. Se efectuaron un total de 6 preguntas por cada usuario, 3 para la utilidad percibida y 3 para la facilidad de uso; la respuesta de cada pregunta se ajustó a una escala del 1 al 7 (donde 1 es la peor calificación y 7 es la mejor). Se realizó pruebas con un total de 20 personas que trabajan en la empresa Soft Warehouse en la sede de Quito. El detalle de la encuesta elaborada se encuentra en las Figuras 47 y sus resultados se encuentran en el Anexo XI.

#### Cuestionario de utilidad percibida

The image shows a digital questionnaire titled "Utilidad percibida" (Perceived Utility). It contains three questions, each followed by a 7-point Likert scale. The scale is represented by seven radio buttons, with the first one selected. The text "De ninguna manera" (In no way) is on the left and "Absolutamente" (Absolutely) is on the right. The questions are:

- ¿Cree que se han facilitado sus trámites al conocer la información que se puede consultar en la aplicación móvil? \*
- ¿Cree que el uso del aplicativo le permite ahorrar tiempo y evitar el uso otro canal de atención para sus necesidades en la entidad financiera? \*
- ¿Utilizaría esta aplicación para manejar su dinero, buscar información y efectuar transferencias? \*

Figura 47: Formulario para la utilidad percibida



## Cuestionario facilidad de uso percibida

**Facilidad de uso**

¿La información que ofrece la aplicación es clara y entendible? \*

1 2 3 4 5 6 7

De ninguna manera        Absolutamente

¿Es intuitivo para usted usar la aplicación? \*

1 2 3 4 5 6 7

De ninguna manera        Absolutamente

¿Le resultó sencillo consultar la información y realizar transferencias entre distintas cuentas? \*

1 2 3 4 5 6 7

De ninguna manera        Absolutamente

Figura 48: Formulario para facilidad de uso percibida

### 3.2.1. Utilidad percibida

La Figura 49 representa un promedio de las respuestas a la encuesta de utilidad percibida.

La pregunta 1 que se refiere a la facilidad que brinda la aplicación para resolver trámites obtuvo el mejor promedio (6.6/7) debido a que la información que brinda la aplicación le ayuda al usuario a simplificar tareas ya que le proporciona gran cantidad de información no solo personal sino también de toda la entidad financiera.

La pregunta 2 que se refiere al ahorro de tiempo y la búsqueda de otro canal de atención obtuvo un promedio alto de (6.1/7) puesto que, los usuarios si creen que la aplicación es útil y representa un ahorro de tiempo, además si cumple con sus expectativas para no decidir buscar otro canal para hacer sus transacciones en la entidad financiera.

Finalmente, la pregunta 3 que se refiere a la utilidad que le darían a la aplicación para el manejo de dinero, información y el uso de transferencias obtuvo un promedio aceptable de (6/7), lo que indica que los usuarios si creen que la aplicación es beneficiosa para estos fines y la usarían sin problemas.

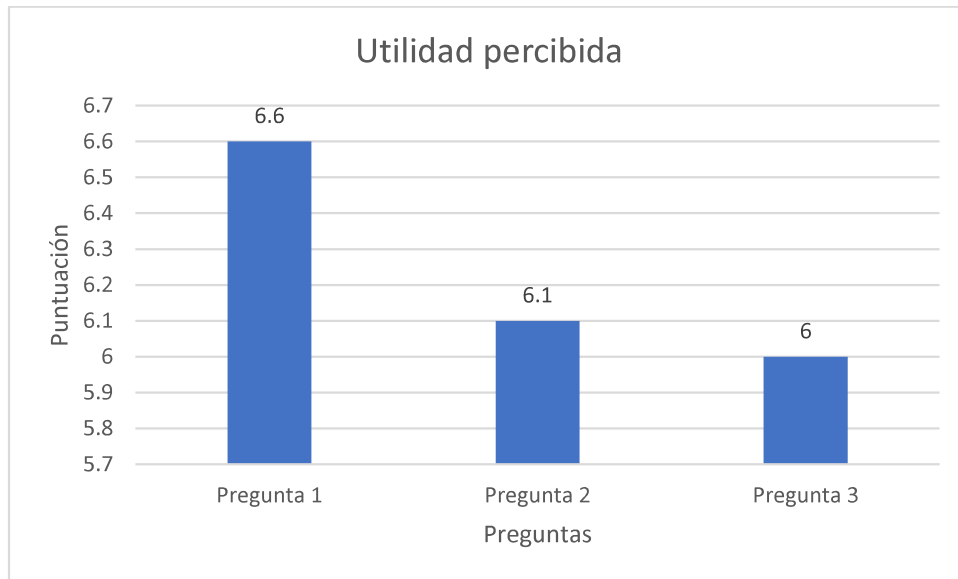


Figura 49: Promedio por pregunta de la utilidad percibida en el producto final

### 3.2.2. Facilidad de uso

La Figura 50 representa un promedio de las respuestas a la encuesta de facilidad de uso.

La pregunta 1 que consultó sobre la claridad y entendimiento de la información en la aplicación obtuvo un promedio aceptable de (6.2/7). Lo que refleja que la aplicación si es comprensible para los usuarios.

La pregunta 2 que se refiere a la intuición que percibe el usuario al usar la aplicación obtuvo el mejor promedio (6.3/7). Puesto que, para el diseño de cada pantalla se trató de organizar la información en los elementos más comunes que manejan los usuarios en aplicaciones móviles.

La pregunta 3 que se refiere a la sencillez en la búsqueda de información y realización de transferencias se obtuvo un puntaje de (6.1/7) lo que evidencia que, existe falta de familiaridad de la aplicación con los usuarios al momento de ubicar el contenido, además que existen términos usados en el negocio que son pocos conocidos por el usuario común, por ejemplo, la diferencia entre transferencias con cuentas propias en el mismo banco y a otros bancos.

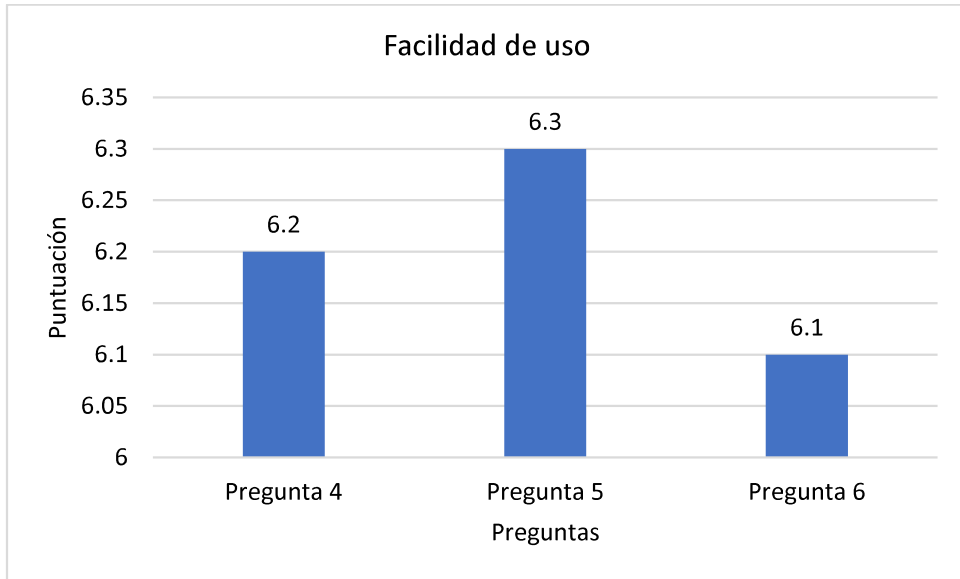


Figura 50: Promedio por pregunta de la facilidad de uso percibida en el producto final

## 4. CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

- La definición de una arquitectura para el sistema antes de empezar el desarrollo permitió a todos los involucrados entender la estructura del proyecto, de esta manera, se facilitó la comprensión para el equipo de desarrollo creando código mantenible, testeable y de calidad.
- Mediante una revisión sistemática y conversaciones con la empresa, se encontraron las características importantes para definir una aplicación móvil que pueda calzar al giro del negocio y cumpla con las necesidades de los usuarios finales que tiene el *Core* bancario.
- La utilización de herramientas libres para el desarrollo de la aplicación facilitó la adquisición, instalación y la interoperabilidad entre cada una. De manera que no se invirtió en la compra de licencias, causando un ahorro de tiempo y dinero para el desarrollo.
- La integración continua permitió a los desarrolladores poner sus fragmentos de código en el repositorio principal en cuestión de minutos, así mismo el proceso de compilación y la localización de errores fue más fácil y su tiempo de corrección más corto.
- El manejo de versiones en las distintas ramas del repositorio permitió automatizar las pruebas unitarias y probar la calidad de código existente, entregándose un código de mejor calidad y que cumple con estándares internacionales.
- Realizar automáticamente los procesos de construcción, pruebas unitarias y verificación de la calidad del código optimizó la calidad y el tiempo de entrega de los incrementos en cada iteración evitando así el retraso de la planificación de cada Sprint y en sí el desarrollo del aplicativo.
- El uso de Scrum como metodología permitió desarrollar una aplicación basada en el valor del negocio, la participación del *Product Owner* permitió obtener incrementos que respondan a las necesidades del cliente de manera planificada y organizada. Este rol retroalimentó al desarrollo del aplicativo en cada iteración de manera efectiva y oportuna.

- Las pruebas de aceptación definieron criterios de aceptación basados en las historias de usuario que se generaron antes del desarrollo del aplicativo móvil. Estos casos fueron presentados al Product Owner en cada revisión para su evaluación, y todos ellos fueron aprobados. Por tanto, se concluye que la aplicación pudo satisfacer las expectativas del usuario.
- *Nativescript* permite traducir código a los componentes nativos de cada plataforma, si se necesita acceder a una característica específica del dispositivo solo se debe aprender cómo acceder a las *APIs* nativas con Javascript, lo que se traduce en ahorro de tiempo de desarrollo ya que se está creando una aplicación que se puede desplegar en las dos plataformas de manera eficiente.

## 4.2 Recomendaciones

Para trabajos futuros se recomienda:

- Ofrecer la gestión de beneficiarios desde la aplicación móvil para evitar la dependencia con la aplicación web.
- Incluir la opción de despliegue automático en la Play store dentro de las opciones de *DevOps* para ahorrar tiempo en el proceso de actualización de esta.
- Incluir el uso de mapas para la geolocalización de oficinas y cajeros para mejorar la presentación de la información al usuario.
- Incluir el uso de la huella dactilar para el ingreso en la aplicación para aprovechar las características de los últimos modelos de dispositivos en el mercado.

## 5. REFERENCIAS BIBLIOGRÁFICAS

- [1] CEPAL, «Estado de la banda ancha en América Latina y el Caribe,» Octubre 2016. [En línea]. Available: <https://www.cepal.org/es/publicaciones/40528-estado-la-banda-ancha-america-latina-caribe-2016>. [Último acceso: 23 Diciembre 2018].
- [2] F. E. Mundial, «Informe Global de Tecnologías de la Información,» 6 Julio 2016. [En línea]. Available: <http://www.cdi.org.pe/InformeGlobaldeInformacion/GITR2016.html>. [Último acceso: 25 Noviembre 2018].
- [3] INEC. (2013). Ecuador en cifras. [En línea]. Available: [http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas\\_Sociales/TIC/Resultados\\_principales\\_140515.Tic.pdf](http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/Resultados_principales_140515.Tic.pdf). [Último acceso: 21 Mayo 2018]
- [4] T. Evelyn. (2018, 14 Febrero). El uso de la banca digital creció un 30% en el Ecuador el año pasado. [En línea]. Available: <http://www.revistalideres.ec/lideres/banca-digital-ecuador-tecnologia-informe.html>. [Último acceso: 21 Mayo 2018].
- [5] M. Joaquín, N. Carlos, M. Carla y B. Silvia, “Análisis de los factores determinantes de la lealtad hacia los servicios bancarios online”, Cuadernos de Economía y Dirección de la Empresa, vol. 14, nº 1, pp. 26-39, Enero 2011.
- [6] I. Villavicencio. (2013, Julio). SoftWare House. [En línea]. Available: <https://fitbank.com/fitbank.html>. [Último acceso: 18 Febrero 2019].
- [7] F. Prior y J. Santomá. (2008, Marzo). “La banca móvil como catalizadora de la bancarización de los pobres: Modelos de Negocio y Desafíos regulatorios”. [En línea]. Available: <https://media.iese.edu/research/pdfs/DI-0738.pdf>, pp 2. [Último acceso: 2 Junio 2019].
- [8] A Rodríguez. (2019, Noviembre). Servicios Web de RESTful: Los aspectos básicos. [En línea]. Available: <https://www.ibm.com/developerworks/ssa/library/ws-restful/index.html>
- [9] A. Wern. (2016, Julio). Una introducción a NativeScript. [En línea]. Available: <https://code.tutsplus.com/es/articles/an-introduction-to-nativescript--cms-26771>. [Último acceso: 2 Junio 2018].

- [10] Progress. (2019). How NativeScript Works. [En línea]. Available: <https://docs.nativescript.org/angular/core-concepts/technical-overview>
- [11] Nativescript. Welcome to NativeScript Sidekick. [En línea]. Available: <https://docs.nativescript.org/sidekick/intro/introduction>. [Último acceso: 31 Julio 2019].
- [12] F. Lardinois. (2015). Techcrunch. [En línea]. Available: <https://techcrunch.com/2015/04/29/87onstrucc-shocks-the-world-with-visual-studio-code-a-free-code-editor-for-os-x-linux-and-windows/>
- [13] Sandobalín, Julio & Zuñiga-Prieto, Miguel & Insfran, Emilio & Abrahão, Silvia & Cano Genoves, Carlos. (2016). Una aproximación DevOps para el Desarrollo Dirigido por Modelos de Servicios Cloud.
- [14] GitLab. Auto DevOps. [En línea]. Available: <https://docs.gitlab.com/ee/topics/autodevops/>. [Último acceso: 2 Junio 2019].
- [15] Docker. (2019). Dockerfile reference. [En línea]. Available: <https://docs.docker.com/engine/reference/builder/>
- [16] GitLab. Auto DevOps-Auto Build. [En línea]. Available: <https://docs.gitlab.com/ee/topics/autodevops/#auto-build>
- [17] GitLab. Auto DevOps-Auto Code Quality. [En línea]. Available: <https://docs.gitlab.com/ee/topics/autodevops/#auto-code-quality-starter>
- [18] GitLab. Auto DevOps-Auto Test. [En línea]. Available: <https://docs.gitlab.com/ee/topics/autodevops/#auto-test>
- [19] Jasmine. Getting started. [En línea]. Available: <https://jasmine.github.io>
- [20] S. Ambler. (2005). Agile Requirements Change Management. [En línea]. Available: <http://agilemodeling.com/essays/changeManagement.htm>. [Último acceso: 4 diciembre 2017].
- [21] T. Satpathy. Una guía para el Cuerpo de Conocimiento de Scrum (Guía SBOK™). 3rd ed. Avondale, Arizona: SCRUMstudy™, 2017, pp.11.
- [22] T. Satpathy. Una guía para el Cuerpo de Conocimiento de Scrum (Guía SBOK™). 3rd ed. Avondale, Arizona: SCRUMstudy™, 2017, pp.45.
- [23] T. Satpathy. Una guía para el Cuerpo de Conocimiento de Scrum (Guía SBOK™). 3rd ed. Avondale, Arizona: SCRUMstudy™, 2017, pp.46.

[24] T. Satpathy. Una guía para el Cuerpo de Conocimiento de Scrum (Guía SBOK™). 3rd ed. Avondale, Arizona: SCRUMstudy™, 2017, pp.2-3.

[25] T. Satpathy. Una guía para el Cuerpo de Conocimiento de Scrum (Guía SBOK™). 3rd ed. Avondale, Arizona: SCRUMstudy™, 2017, pp.34.

[26] S. Casanova. (2019). Estimación ágil con la técnica Planning Poker.

[En línea]. Available: <https://samuelcasanova.com/2016/01/88onstrucci-agil-con-la-tecnica-planning-poker/>

[27] T. Satpathy. Una guía para el Cuerpo de Conocimiento de Scrum (Guía SBOK™). 3rd ed. Avondale, Arizona: SCRUMstudy™, 2017, pp.148.

[28] T. Satpathy. Una guía para el Cuerpo de Conocimiento de Scrum (Guía SBOK™). 3rd ed. Avondale, Arizona: SCRUMstudy™, 2017, pp.222.

[29] T. Satpathy. Una guía para el Cuerpo de Conocimiento de Scrum (Guía SBOK™). 3rd ed. Avondale, Arizona: SCRUMstudy™, 2017, pp.224.

[30] A. Enríquez. (2010, Julio). Definiciones de Dinero Móvil. Mobile Money for the Unbanked [En línea]. Available: [https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2012/03/Definiciones\\_de\\_Dinero\\_Movil.pdf](https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2012/03/Definiciones_de_Dinero_Movil.pdf)

[31] App Marker. ¿Qué es la pantalla Splash (imagen de bienvenida)?. [En línea]. Available:<https://theappmaker8.zendesk.com/hc/es/articles/218130997--Qué-es-la-pantalla-Splash-imagen-de-bienvenida->

[32] T. Satpathy. Una guía para el Cuerpo de Conocimiento de Scrum (Guía SBOK™). 3rd ed. Avondale, Arizona: SCRUMstudy™, 2017, pp.113.

[33] F. D. Davis, «Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology,» MIS Quarterly, vol. 13, pp. 319-340, 1989.



## 6. ANEXOS

### 6.1 Anexo I: Archivo de Configuración Auto Devops Gitlab

image: fittns

stages:

- build
- test

build\_after\_push:

only:

- pushes

stage: build

script:

- cd BancaMovilV2
- npm install --unsafe-perm=true --allow-root
- tns build android
- cd ..
- mkdir apk/
- cp BancaMovilV2/platforms/android/app/build/outputs/apk/debug/app-debug.apk apk/app-debug.apk

artifacts:

name: "\$CI\_PROJECT\_NAME-\$CI\_COMMIT\_SHA"

expire\_in: 6 hrs

paths:

- "apk/\*.apk"

daily\_build:

only:

- schedules
- web

stage: build

script:

- cd BancaMovilV2
- npm install --unsafe-perm=true --allow-root
- tns build android
- cd ..

- mkdir apk/
- cp BancaMovilV2/platforms/android/app/build/outputs/apk/debug/app-debug.apk apk/app-debug.apk

artifacts:

name: "\$CI\_PROJECT\_NAME-\$CI\_COMMIT\_SHA"

expire\_in: 12 hrs

paths:

- "apk/\*.apk"

code\_quality:

only:

- nativescript5

stage: test

allow\_failure: true

script:

- cd BancaMovilV2
- npm install --unsafe-perm=true --allow-root
- tslint -v
- tslint -p tsconfig.json > result\_code.txt || mkdir apk

artifacts:

paths:

- "BancaMovilV2/result\_code.txt"

## 6.2 Anexo II: Detalle de historias de usuario para el Sprint 1

Tabla 19: Historia de usuario HUE02-01

HISTORIA DE USUARIO	HUE02-01
<b>TITULO:</b> VALIDACIÓN DE NOMBRE DE USUARIO, IMAGEN Y CONTRASEÑA	<b>USUARIO:</b> CLIENTE
<b>DESCRIPCIÓN:</b> Como cliente requiero poder autenticarme en la aplicación haciendo uso de un nombre de usuario, contraseña y validar mi imagen de usuario de una matriz de imágenes aleatorias.	
PRIORIDAD:4	ESFUERZO: 9
<b>PROGRAMADOR RESPONSABLE:</b> JEFFERSON PAREDES	

Tabla 20: Historia de usuario HUE01-01

<b>HISTORIA DE USUARIO</b>		<b>HUE01-01</b>
<b>TITULO:</b> TRANSFERENCIAS ENTRE CUENTAS PROPIAS		<b>USUARIO:</b> CLIENTE
<b>DESCRIPCIÓN:</b> Como cliente requiero poder realizar transferencias entre cuentas propias para poder organizar mi dinero.		
PRIORIDAD:4		ESFUERZO: 6
<b>PROGRAMADOR RESPONSABLE:</b> DANIELA RAMOS		

Tabla 21: Historia de usuario HUE01-02

<b>HISTORIA DE USUARIO</b>		<b>HUE01-02</b>
<b>TITULO:</b> TRANSFERENCIAS A TERCEROS EN LA MISMA INSTITUCIÓN.		<b>USUARIO:</b> CLIENTE
<b>DESCRIPCIÓN:</b> Como cliente requiero poder realizar transferencias a terceros en la misma institución para poder realizar pagos varios		
PRIORIDAD:4		ESFUERZO: 6
<b>PROGRAMADOR RESPONSABLE:</b> DANIELA RAMOS		

### 6.3 Anexo III: Detalle Sprint 1 Backlog

Tabla 22: Detalle Sprint 1 Backlog

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	TAREAS
HUE02-01	TR1-01	La aplicación tendrá una pantalla inicial para el acceso de los usuarios, donde se podrá ingresar el usuario, escoger una imagen e ingresar la contraseña	<p>Crear el modelo de datos para manejo de la información recibida del servidor por medio de los servicios <i>RESTful</i>.</p> <p>Crear el componente para el ingreso con:</p> <ul style="list-style-type: none"> <li>-Archivo de lógica del negocio en <i>Typescript</i>.</li> <li>-Vista en Angular.</li> </ul> <p>Crear el diccionario de palabras en inglés y español para la vista creada.</p> <p>Crear y ejecutar las pruebas unitarias del componente.</p>
	TR1-02	La aplicación validará el ingreso de usuarios previamente creados en la página web.	Crear el método para que consuma el servicio <i>REST</i> para la validación de la existencia del usuario.
	TR1-03	La aplicación contará con una matriz aleatoria de imágenes para validar la imagen del usuario.	Crear una matriz que se llene de acuerdo con un método de ordenamiento para que las imágenes se presenten aleatoriamente.
	TR1-04	La aplicación validará la selección correcta de la imagen de usuario.	Crear el método que valide la imagen seleccionada en la matriz consumiendo el servicio <i>REST</i> de validación de imagen del usuario.
	TR1-05	La aplicación permitirá el ingreso oculto de la contraseña de usuario	Crear un campo para ingresar la contraseña de manera oculta.
	TR1-06	La aplicación validará el ingreso de la contraseña del usuario.	Crear el método que consuma el servicio <i>REST</i> de validación de la contraseña de usuario.
	TR1-07	La aplicación protegerá el envío de la contraseña.	Crear el método para el cifrado de la contraseña.

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	TAREAS
HUE01-01	TR1-08	La aplicación tendrá una pantalla para realizar transferencias entre cuentas propias, donde se pueda escoger la cuenta propia desde la que se va a realizar la transferencia y otra opción en donde se seleccione la cuenta propia a la que debe ir la transferencia. El ingreso del monto, botones para solicitud del token, el envío de la transferencia o cancelación de esta.	<p>Crear el modelo de datos para transferencias entre cuentas propias.</p> <p>Crear servicio para el consumo del <i>API REST</i>.</p> <p>Crear el componente para las transferencias entre cuentas propias.</p> <p>-Archivo de lógica del negocio en <i>Typescript</i>.</p> <p>-Vista en Angular.</p> <p>Crear y ejecutar las pruebas unitarias del componente.</p>
	TR1-09	Los textos en la pantalla de transferencias propias se visualizarán con el idioma del dispositivo.	Crear diccionario de palabras en inglés y español para la vista
	TR1-10	Para realizar transferencias propias la aplicación debe validar que la cuenta de origen de fondos no aparezca en la lista de selección para el destino.	Crear el método para validación de las cuentas en los combos de cuenta origen y destino.
	TR1-11	Para realizar transferencias propias la aplicación debe permitir ingresar el monto solo en números.	En la vista crear el input que solo permita el ingreso de números.
	TR1-12	Para realizar transferencias propias la aplicación debe permitir pedir un Token, mismo que se envía a un correo del usuario.	Crear el botón "Solicitar" con el método para enviar una solicitud de token al servidor del <i>Core</i> .
	TR1-13	Para realizar transferencias propias la aplicación debe permitir validar el Token ingresado.	Crear el método que consuma el servicio <i>REST</i> que valida la temporalidad del Token que se va a usar en la transferencia.
	TR1-14	Para realizar transferencias propias la aplicación debe permitir limpiar los campos.	Crear el método para el botón "x" que permita limpiar todos los campos que contiene la pantalla.
	TR1-15	En el caso que la transferencia entre cuentas propias no pueda ser ejecutada la aplicación debe mostrar el mensaje que envía el sistema.	<p>Crear el método que consuma el servicio <i>REST</i> que envía la transferencia entre cuentas propias. Crear el mensaje que devuelve el servidor al intentar realizar la transferencia.</p>
	TR1-16	En el caso que la transferencia entre cuentas propias se ejecute satisfactoriamente la aplicación debe mostrar el mensaje que envía el sistema.	

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	TAREAS
HUE01-02	TR1-17	La aplicación tendrá una pantalla para realizar transferencias a terceros en la misma institución, donde se pueda escoger la cuenta propia desde la que se va a realizar la transferencia, el ingreso del monto, botones para solicitud del token, el envío de la transferencia o cancelación de esta.	<p>Crear el modelo de datos para transferencias a terceros en la misma institución.</p> <p>Crear el servicio para el consumo del <i>API REST</i>.</p> <p>Crear el componente para las transferencias a terceros en la misma institución.</p> <ul style="list-style-type: none"> <li>-Archivo de lógica del negocio en <i>Typescript</i>.</li> <li>-Vista en Angular creada.</li> </ul> <p>Crear y ejecutar las pruebas unitarias del componente.</p>
	TR1-18	Los textos en la pantalla de transferencias a terceros se deberán visualizar con el idioma del dispositivo	Crear diccionario de palabras en inglés y español para la vista
	TR1-19	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir ingresar el monto solo en números.	En la vista crear el input que solo permita el ingreso de números.
	TR1-20	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir pedir un Token, mismo que se envía a un correo del usuario.	Crear el botón "Solicitar" con el método para enviar una solicitud de token al servidor del <i>Core</i> .
	TR1-21	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir validar el Token ingresado.	Crear el método que consuma el servicio <i>REST</i> que valida la temporalidad del Token que se va a usar en el envío de la transferencia.
	TR1-22	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir limpiar los campos.	Crear el botón "x" para limpiar todos los campos que contiene la pantalla.
	TR1-23	En el caso que la transferencia a terceros en la misma institución no pueda ser ejecutada la aplicación debe mostrar el mensaje que envía el sistema.	<p>Crear el método que consuma el servicio <i>REST</i> que envía la transferencia a terceros en la misma institución.</p> <p>Crear el mensaje que devuelve el servidor al intentar realizar la transferencia.</p>
	TR1-24	En el caso que la transferencia a terceros en la misma institución se ejecute satisfactoriamente la aplicación debe mostrar el mensaje que envía el sistema.	

## 6.4 Anexo IV: Revisión de criterios de aceptación del Sprint 1

Tabla 23: Revisión de criterios de aceptación del Sprint 1

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	CHECKLIST	COMENTARIO
HUE02-01	TR-01	La aplicación tendrá una pantalla inicial para el acceso de los usuarios, donde se podrá ingresar el usuario, escoger una imagen e ingresar la contraseña	<b>SI</b>	
	TR-02	La aplicación validará el ingreso de usuarios previamente creados en la página web.	<b>SI</b>	
	TR-03	La aplicación contará con una matriz aleatoria de imágenes para validar la imagen del usuario.	<b>SI</b>	
	TR-04	La aplicación validará la selección correcta de la imagen de usuario.	<b>SI</b>	
	TR-05	La aplicación permitirá el ingreso oculto de la contraseña de usuario	<b>SI</b>	
	TR-06	La aplicación validará el ingreso de la contraseña del usuario.	<b>SI</b>	
	TR-07	La aplicación protegerá el envío de la contraseña.	<b>SI</b>	

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	CHEC KLIST	COMENTARIO
HUE01-01	TR-08	La aplicación tendrá una pantalla para realizar transferencias entre cuentas propias, donde se pueda escoger la cuenta propia desde la que se va a realizar la transferencia y otra opción en donde se seleccione la cuenta propia a la que debe ir la transferencia. El ingreso del monto, botones para solicitud del token, el envío de la transferencia o cancelación de esta.	SI	
	TR-09	Los textos en la pantalla de transferencias propias se deberán visualizar con el idioma del dispositivo	SI	
	TR-10	Para realizar transferencias propias la aplicación debe validar que la cuenta de origen de fondos no aparezca en la lista de selección para el destino.	SI	
	TR-11	Para realizar transferencias propias la aplicación debe permitir ingresar el monto solo en números.	NO	En el campo Monto el teclado solo permite el ingreso de números. Se debe controlar el ingreso de los caracteres alfanuméricos que aún aparecen.
	TR-12	Para realizar transferencias propias la aplicación debe permitir pedir un Token, mismo que se envía a un correo del usuario.	SI	
	TR-13	Para realizar transferencias propias la aplicación debe permitir validar el Token ingresado.	SI	
	TR-14	Para realizar transferencias propias la aplicación debe permitir limpiar los campos.	SI	
	TR-15	En el caso que la transferencia entre cuentas propias no pueda ser ejecutada la aplicación debe mostrar el mensaje que envía el sistema.	SI	
	TR-16	En el caso que la transferencia entre cuentas propias se ejecute satisfactoriamente la aplicación debe mostrar el mensaje que envía el sistema.	SI	



H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	CHECKLIST	COMENTARIO
HUE01-02	TR-17	La aplicación tendrá una pantalla para realizar transferencias a terceros en la misma institución, donde se pueda escoger la cuenta propia desde la que se va a realizar la transferencia, el ingreso del monto, botones para solicitud del token, el envío de la transferencia o cancelación de esta.	<b>SI</b>	
	TR-18	Los textos en la pantalla de transferencias a terceros se deberán visualizar con el idioma del dispositivo	<b>SI</b>	
	TR-19	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir ingresar el monto solo en números.	<b>NO</b>	En el campo Monto el teclado solo permite el ingreso de números. Se debe controlar el ingreso de los caracteres alfanuméricos que aún aparecen.
	TR-20	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir pedir un Token, mismo que se envía a un correo del usuario.	<b>SI</b>	
	TR-21	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir validar el Token ingresado.	<b>SI</b>	
	TR-22	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir limpiar los campos.	<b>SI</b>	
	TR-23	En el caso que la transferencia a terceros en la misma institución no pueda ser ejecutada la aplicación debe mostrar el mensaje que envía el sistema.	<b>SI</b>	
	TR-24	En el caso que la transferencia a terceros en la misma institución se ejecute satisfactoriamente la aplicación debe mostrar el mensaje que envía el sistema.	<b>SI</b>	

## 6.5 Anexo V: Detalle de historias de usuario para el Sprint 2.

Tabla 24: Historia de usuario HUE01-03

<b>HISTORIA DE USUARIO</b>		<b>HUE01-03</b>
<b>TITULO:</b> Transferencias a otras instituciones	<b>USUARIO:</b> CLIENTE	
<b>DESCRIPCIÓN:</b> Como cliente requiero poder realizar transferencias a otras instituciones para poder realizar pagos varios.		
PRIORIDAD:4	ESFUERZO: 6	
<b>PROGRAMADOR RESPONSABLE:</b> JEFFERSON PAREDES		

Tabla 25: Historia de usuario HUE03-01

<b>HISTORIA DE USUARIO</b>		<b>HUE03-01</b>
<b>TITULO:</b> Posición consolidada de cuentas, préstamos y pólizas	<b>USUARIO:</b> CLIENTE	
<b>DESCRIPCIÓN:</b> Como cliente requiero ver la posición consolidada de mis cuentas, préstamos y pólizas en la institución financiera para saber el saldo de estas.		
PRIORIDAD:3	ESFUERZO: 10	
<b>PROGRAMADOR RESPONSABLE:</b> DANIELA RAMOS		

Tabla 26: Historia de usuario HUE03-02

<b>HISTORIA DE USUARIO</b>		<b>HUE03-02</b>
<b>TITULO:</b> Detalle de movimientos de cada cuenta	<b>USUARIO:</b> CLIENTE	
<b>DESCRIPCIÓN:</b> Como cliente requiero ver el detalle de movimientos de cada cuenta para saber el estado de cada una.		
PRIORIDAD:3	ESFUERZO: 5	
<b>PROGRAMADOR RESPONSABLE:</b> JEFFERSON PAREDES		

## 6.6 Anexo VI: Detalle Sprint 2 Backlog

Tabla 27: Detalle Sprint 2 Backlog

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	TAREAS
HUE01-01	TR1-11	Para realizar transferencias propias la aplicación debe permitir ingresar el monto solo en números.	Crear método para controlar el ingreso de los caracteres alfanuméricos
HUE01-02	TR1-19	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir ingresar el monto solo en números.	Crear método para controlar el ingreso de los caracteres alfanuméricos

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	TAREAS
HUE01-03	TR2-01	La aplicación tendrá una pantalla para realizar transferencias a otras instituciones, donde se pueda escoger la cuenta propia desde la que se va a realizar la transferencia, el ingreso del monto, botones para solicitud del token, el envío de la transferencia o cancelación de esta.	<p>Crear el modelo de datos para transferencias a otras instituciones.</p> <p>Crear el servicio para el consumo del <i>API REST</i>.</p> <p>Crear el componente para las transferencias a otras instituciones.</p> <p>-Archivo de lógica del negocio en <i>Typescript</i>.</p> <p>-Vista en Angular creada.</p> <p>Crear y ejecutar las pruebas unitarias del componente.</p>
	TR2-02	Los textos en la pantalla de transferencias a otras instituciones se deberán visualizar con el idioma del dispositivo	Crear diccionario de palabras en inglés y español para la vista
	TR2-03	Para realizar transferencias a otras instituciones la aplicación debe permitir ingresar el monto solo en números.	En la vista crear el input que solo permita el ingreso de números.
	TR2-04	Para realizar transferencias a otras instituciones la aplicación debe permitir pedir un Token, mismo que se envía a un correo del usuario.	Crear el botón "Solicitar" con el método para enviar una solicitud de token al servidor del <i>Core</i> .
	TR2-05	Para realizar transferencias a otras instituciones la aplicación debe permitir validar el Token ingresado.	Crear el método que consume el servicio <i>REST</i> que valida la temporalidad del Token que se va a usar en el envío de la transferencia.
	TR2-06	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir limpiar los campos.	Crear el botón "x" para limpiar todos los campos que contiene la pantalla.
	TR2-07	En el caso que la transferencia a otras instituciones no pueda ser ejecutada la aplicación debe mostrar el mensaje que envía el sistema.	Crear el método que consume el servicio <i>REST</i> que envía la transferencia a otras instituciones.
	TR2-08	En el caso que la transferencia a otras instituciones se ejecute satisfactoriamente la aplicación debe mostrar el mensaje que envía el sistema.	Crear el mensaje que devuelve el servidor al intentar realizar la transferencia.
	TR2-09	La aplicación debe contar con un menú lateral, el mismo deberá ocultarse/mostrarse al deslizarse de izquierda a derecha. Mostrará el nombre del usuario y contará con las opciones de: 1. Cuentas propias, 2. Terceros misma institución, 3. Otra institución y 4. Posición consolidada. Cada opción llevará a su respectiva pantalla.	<p>Crear el componente para el menú lateral</p> <p>-Archivo de lógica del negocio en <i>Typescript</i>.</p> <p>-Vista en Angular creada.</p>

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	TAREAS
HUE03-01	TR2-10	La aplicación contará con la pantalla principal de posición consolidada en la que se debe mostrar cada cuenta, préstamo y póliza	<p>Crear el modelo de datos para la posición consolidada.            Crear el servicio para el consumo del <i>API REST</i>.            Crear el componente para la posición consolidada.            -Archivo de lógica del negocio en <i>Typescript</i>.            -Vista en Angular creada.            Crear y ejecutar las pruebas unitarias del componente.</p>
	TR2-11	Los textos en la pantalla de posición consolidada se deberán visualizar con el idioma del dispositivo	Crear diccionario de palabras en inglés y español para la vista
	TR2-12	En el caso de la posición consolidada se debe mostrar la información por cada cuenta del cliente	Crear el método que consuma el servicio <i>REST</i> que recibe la información de cada cuenta
	TR2-13	En el caso de la posición consolidada se debe mostrar la información de cada póliza	Crear el método que consuma el servicio <i>REST</i> que recibe la información de cada póliza
	TR2-14	En el caso de la posición consolidada se debe mostrar la información de cada préstamo	Crear el método que consuma el servicio <i>REST</i> que recibe la información de cada préstamo
	TR2-15	En el caso de la posición consolidada se debe mostrar en rojo los préstamos que se encuentren en estado vencido.	Crear el método que controla el CSS para pintar el préstamo de acuerdo con su parametrización.
HUE03-02	TR2-16	La aplicación contará con la pantalla de detalle de movimientos en la que se debe mostrar cada movimiento de una cuenta.	<p>Crear el modelo de datos para el detalle de movimientos.            Crear el servicio para el consumo del <i>API REST</i>.            Crear el componente para el detalle de movimientos.            -Archivo de lógica del negocio en <i>Typescript</i>.            -Vista en Angular creada.            Crear y ejecutar las pruebas unitarias del componente.</p>
	TR2-17	Los textos en la pantalla de detalles de movimientos se deberán visualizar con el idioma del dispositivo	Crear diccionario de palabras en inglés y español para la vista

## 6.7 Anexo VII: Revisión de criterios de aceptación del Sprint 2

Tabla 28: Revisión de criterios de aceptación del Sprint 2.

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	TAREAS	CHECKLIST
HUE01-01	TR1-11	Para realizar transferencias propias la aplicación debe permitir ingresar el monto solo en números.	Crear método para controlar el ingreso de los caracteres alfanuméricos	SI
HUE01-02	TR1-19	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir ingresar el monto solo en números.	Crear método para controlar el ingreso de los caracteres alfanuméricos	SI
HUE03-01	TR2-10	La aplicación contará con la pantalla principal de posición consolidada en la que se debe mostrar cada cuenta, préstamo y póliza	Crear el modelo de datos para la posición consolidada. Crear el servicio para el consumo del API REST. Crear el componente para la posición consolidada. -Archivo de lógica del negocio en Typescript. -Vista en Angular creada. Crear y ejecutar las pruebas unitarias del componente.	SI
	TR2-11	Los textos en la pantalla de posición consolidada se deberán visualizar con el idioma del dispositivo	Crear diccionario de palabras en inglés y español para la vista	SI
	TR2-12	En el caso de la posición consolidada se debe mostrar la información por cada cuenta del cliente	Crear el método que consume el servicio REST que recibe la información de cada cuenta	SI
	TR2-13	En el caso de la posición consolidada se debe mostrar la información de cada póliza	Crear el método que consume el servicio REST que recibe la información de cada póliza	SI
	TR2-14	En el caso de la posición consolidada se debe mostrar la información de cada préstamo	Crear el método que consume el servicio REST que recibe la información de cada préstamo	SI
	TR2-15	En el caso de la posición consolidada se debe mostrar en rojo los préstamos que se encuentren en estado vencido.	Crear el método que controla el CSS para pintar el préstamo de acuerdo con su parametrización.	SI

<b>H U</b>	<b>ID TAREA</b>	<b>CRITERIOS DE ACEPTACIÓN</b>	<b>TAREAS</b>	<b>CHECKLIST</b>
HUE01-03	TR2-01	La aplicación tendrá una pantalla para realizar transferencias a otras instituciones, donde se pueda escoger la cuenta propia desde la que se va a realizar la transferencia, el ingreso del monto, botones para solicitud del token, el envío de la transferencia o cancelación de esta.	<p>Crear el modelo de datos para transferencias a otras instituciones.</p> <p>Crear el servicio para el consumo del API REST.</p> <p>Crear el componente para las transferencias a otras instituciones.</p> <p>-Archivo de lógica del negocio en Typescript.</p> <p>-Vista en Angular creada.</p> <p>Crear y ejecutar las pruebas unitarias del componente.</p>	<b>SI</b>
	TR2-02	Los textos en la pantalla de transferencias a otras instituciones se deberán visualizar con el idioma del dispositivo	Crear diccionario de palabras en inglés y español para la vista	<b>SI</b>
	TR2-03	Para realizar transferencias a otras instituciones la aplicación debe permitir ingresar el monto solo en números.	En la vista crear el input que solo permita el ingreso de números.	<b>SI</b>
	TR2-04	Para realizar transferencias a otras instituciones la aplicación debe permitir pedir un Token, mismo que se envía a un correo del usuario.	Crear el botón "Solicitar" con el método para enviar una solicitud de token al servidor del Core.	<b>SI</b>
	TR2-05	Para realizar transferencias a otras instituciones la aplicación debe permitir validar el Token ingresado.	Crear el método que consume el servicio REST que valida la temporalidad del Token que se va a usar en el envío de la transferencia.	<b>SI</b>
	TR2-06	Para realizar transferencias a terceros en la misma institución la aplicación debe permitir limpiar los campos.	Crear el botón "x" para limpiar todos los campos que contiene la pantalla.	<b>SI</b>
	TR2-07	En el caso que la transferencia a otras instituciones no pueda ser ejecutada la aplicación debe mostrar el mensaje que envía el sistema.	Crear el método que consume el servicio REST que envía la transferencia a otras instituciones. Crear el mensaje que devuelve el servidor al intentar realizar la transferencia.	<b>SI</b>
	TR2-08	En el caso que la transferencia a otras instituciones se ejecute satisfactoriamente la aplicación debe mostrar el mensaje que envía el sistema.		<b>SI</b>

<b>H U</b>	<b>ID TAREA</b>	<b>CRITERIOS DE ACEPTACIÓN</b>	<b>TAREAS</b>	<b>CHECKLIST</b>
HUE01-03	TR2-09	La aplicación debe contar con un menú lateral, el mismo deberá ocultarse/mostrarse al deslizarse de izquierda a derecha. Mostrará el nombre del usuario y contará con las opciones de: 1. Cuentas propias, 2. Terceros misma institución, 3. Otra institución y 4. Posición consolidada. Cada opción llevará a su respectiva pantalla.	<p>Crear el componente para el menú lateral</p> <ul style="list-style-type: none"> <li>-Archivo de lógica del negocio en Typescript.</li> <li>-Vista en Angular creada.</li> </ul>	<b>SI</b>
HUE03-02	TR2-16	La aplicación contará con la pantalla de detalle de movimientos en la que se debe mostrar cada movimiento de una cuenta.	<p>Crear el modelo de datos para el detalle de movimientos.</p> <p>Crear el servicio para el consumo del API REST.</p> <p>Crear el componente para el detalle de movimientos.</p> <ul style="list-style-type: none"> <li>-Archivo de lógica del negocio en Typescript.</li> <li>-Vista en Angular creada.</li> </ul> <p>Crear y ejecutar las pruebas unitarias del componente.</p>	<b>SI</b>
	TR2-17	Los textos en la pantalla de detalles de movimientos se deberán visualizar con el idioma del dispositivo	Crear diccionario de palabras en inglés y español para la vista	<b>SI</b>



## 6.8 Anexo VIII: Detalle de historias de usuario para el Sprint 3.

Tabla 29: Historia de usuario HUE03-03

<b>HISTORIA DE USUARIO</b>		<b>HUE03-03</b>
<b>TÍTULO:</b> Tabla de amortización de préstamos y pólizas	<b>USUARIO:</b> CLIENTE	
<b>DESCRIPCIÓN:</b> Como cliente requiero ver la tabla de amortización de cada préstamo y póliza para saber el estado de estos.		
<b>PRIORIDAD:</b> 3	<b>ESFUERZO:</b> 5	
<b>PROGRAMADOR RESPONSABLE:</b> JEFFERSON PAREDES		

Tabla 30: Historia de usuario HUE04-01

<b>HISTORIA DE USUARIO</b>		<b>HUE04-01</b>
<b>TÍTULO:</b> Información de cajeros	<b>USUARIO:</b> CLIENTE	
<b>DESCRIPCIÓN:</b> Como cliente requiero ver la información de los cajeros que tiene la institución financiera para saber su ubicación.		
<b>PRIORIDAD:</b> 3	<b>ESFUERZO:</b> 4	
<b>PROGRAMADOR RESPONSABLE:</b> JEFFERSON PAREDES		

Tabla 31: Historia de usuario HUE04-02

<b>HISTORIA DE USUARIO</b>		<b>HUE04-02</b>
<b>TÍTULO:</b> Dirección de oficinas	<b>USUARIO:</b> CLIENTE	
<b>DESCRIPCIÓN:</b> Como cliente requiero ver la dirección de las oficinas ubicadas en cada ciudad para saber cómo llegar.		
PRIORIDAD:3	ESFUERZO: 4	
<b>PROGRAMADOR RESPONSABLE:</b> DANIELA RAMOS		

Tabla 32: Historia de usuario HUE04-03

<b>HISTORIA DE USUARIO</b>		<b>HUE04-03</b>
<b>TÍTULO:</b> Productos de la institución financiera	<b>USUARIO:</b> CLIENTE	
<b>DESCRIPCIÓN:</b> Como cliente requiero ver los productos que ofrece la institución financiera para saber qué productos puedo solicitar.		
PRIORIDAD:3	ESFUERZO: 4	
<b>PROGRAMADOR RESPONSABLE:</b> DANIELA RAMOS		

Tabla 33: Historia de usuario HUE04-04

<b>HISTORIA DE USUARIO</b>		<b>HUE04-04</b>
<b>TÍTULO:</b> Servicios de la institución financiera	<b>USUARIO:</b> CLIENTE	
<b>DESCRIPCIÓN:</b> Como cliente requiero ver los servicios que ofrece la institución financiera para saber qué servicios puedo solicitar.		
PRIORIDAD:3	ESFUERZO: 4	
<b>PROGRAMADOR RESPONSABLE:</b> DANIELA RAMOS		

Tabla 34: Historia de usuario HUE05-01

<b>HISTORIA DE USUARIO</b>		<b>HUE05-01</b>
<b>TÍTULO:</b> Migración a <i>Nativescript</i> 6.0	<b>USUARIO:</b> CLIENTE	
<b>DESCRIPCIÓN:</b> Como cliente requiero tener la última versión del <i>framework</i> en mi aplicación móvil.		
PRIORIDAD:3	ESFUERZO: 1	
<b>PROGRAMADOR RESPONSABLE:</b> JEFFERSON PAREDES		

## 6.9 Anexo IX: Detalle Sprint 3 Backlog

Tabla 35: Detalle Sprint 3 Backlog

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	TAREAS
HUE03-03	TR3-01	La aplicación contará con la pantalla de tabla de pagos de préstamos, para llegar a la misma se accederá desde la posición consolidada al ver el detalle de un préstamo.	<p>Crear el modelo de datos para el detalle de pagos de un préstamo.            Crear el servicio para el consumo del <i>API REST</i>.            Crear el componente para el detalle de pagos de un préstamo.            -Archivo de lógica del negocio en <i>Typescript</i>.            -Vista en Angular creada.            Crear y ejecutar las pruebas unitarias del componente.</p>
	TR3-02	Los textos en la pantalla de tabla de pagos de préstamos se deberán visualizar con el idioma del dispositivo	Crear diccionario de palabras en inglés y español para la vista
	TR3-03	La aplicación contará con la pantalla de tabla de pagos de pólizas, para llegar a la misma se accederá desde la posición consolidada al ver el detalle de una póliza.	<p>Crear el modelo de datos para el detalle de pagos de una póliza.            Crear el servicio para el consumo del <i>API REST</i>.            Crear el componente para el detalle de pagos de una póliza.            -Archivo de lógica del negocio en <i>Typescript</i>.            -Vista en Angular creada.            Crear y ejecutar las pruebas unitarias del componente.</p>
	TR3-04	Los textos en la pantalla de tabla de pagos de pólizas se deberán visualizar con el idioma del dispositivo	Crear diccionario de palabras en inglés y español para la vista
HUE04-01	TR3-05	La aplicación tendrá un menú flotante en la parte inferior derecha, en la página de inicio, con cuatro opciones: ATMs, oficinas, productos y servicios. El menú se mostrará y ocultará al presionar.	Crear los botones con las propiedades que permitan mostrarse y ocultarse sobre la pantalla de ingreso a la aplicación
	TR3-06	El menú flotante tendrá la opción de ATMs, al presionar en este se mostrará una pantalla con botones correspondientes a las ciudades donde existen cajeros de la institución financiera.	<p>Crear el método que dirija a la pantalla de cajeros luego de presionar el botón en el menú flotante.</p>
			<p>Crear el modelo de datos para la información de cajeros.            Crear el servicio para el consumo del <i>API REST</i>.            Crear el componente para la información de cajeros.            -Archivo de lógica del negocio en <i>Typescript</i>.            -Vista en Angular creada.            Crear y ejecutar las pruebas unitarias del componente.</p>
TR3-07	Al presionar en una ciudad de la pantalla de ATMs se muestran las direcciones de los cajeros disponibles	Crear el método que consuma el servicio que muestra la información de los cajeros de acuerdo con la ciudad.	

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	TAREAS
HUE04-02	TR3-08	El menú flotante tendrá la opción de Oficinas, al presionar en este se mostrará una pantalla con botones correspondientes a las ciudades donde existen oficinas de la institución financiera.	<p>Crear el método que dirija a la pantalla de oficinas luego de presionar el botón en el menú flotante.</p> <p>Crear el modelo de datos para la información de oficinas.</p> <p>Crear el servicio para el consumo del <i>API REST</i>.</p> <p>Crear el componente para la información de oficinas.</p> <p>-Archivo de lógica del negocio en <i>Typescript</i>.</p> <p>-Vista en Angular creada.</p> <p>Crear y ejecutar las pruebas unitarias del componente.</p>
	TR3-09	Al presionar en una ciudad de la pantalla de Oficinas se muestra las direcciones de las oficinas existentes	<p>Crear el método que consuma el servicio que muestra la información de las oficinas de acuerdo con la ciudad.</p>
HUE04-03	TR3-10	El menú flotante tendrá la opción de Productos, al presionar en este se mostrará una pantalla con botones correspondientes a los tipos de productos que existen en la institución financiera.	<p>Crear el método que dirija a la pantalla de Productos luego de presionar el botón en el menú flotante.</p> <p>Crear el modelo de datos para la información de productos.</p> <p>Crear el servicio para el consumo del <i>API REST</i>.</p> <p>Crear el componente para la información de productos.</p> <p>-Archivo de lógica del negocio en <i>Typescript</i>.</p> <p>-Vista en Angular creada.</p> <p>Crear y ejecutar las pruebas unitarias del componente.</p>
	TR3-11	Al presionar en un Producto de la pantalla de Productos se muestre los diferentes productos que ofrece la institución financiera	<p>Crear el método que consuma el servicio que muestra la información de cada tipo de producto.</p>
HUE04-04	TR3-12	El menú flotante tendrá la opción de Servicios, al presionar en este se mostrará una pantalla con botones correspondientes a los tipos de servicios que existen en la institución financiera.	<p>Crear el método que dirija a la pantalla de Servicios luego de presionar el botón en el menú flotante.</p> <p>Crear el modelo de datos para la información de servicios.</p> <p>Crear el servicio para el consumo del <i>API REST</i>.</p> <p>Crear el componente para la información de servicios.</p> <p>-Archivo de lógica del negocio en <i>Typescript</i>.</p> <p>-Vista en Angular creada.</p> <p>Crear y ejecutar las pruebas unitarias del componente.</p>
HUE05-01	TR3-13	La aplicación contará con la última versión del <i>framework</i>	<p>Ejecutar el comando que propone la documentación de <i>Nativescript</i> y verificar el funcionamiento de la aplicación.</p>

## 6.10 Anexo X: Revisión de criterios de aceptación del Sprint 3

Tabla 36: Revisión de criterios de aceptación del Sprint 3

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	TAREAS	CHE CKLIST
HUE03-03	TR3-01	La aplicación contará con la pantalla de tabla de pagos de préstamos, para llegar a la misma se accederá desde la posición consolidada al ver el detalle de un préstamo.	<p>Crear el modelo de datos para el detalle de pagos de un préstamo.            Crear el servicio para el consumo del <i>API REST</i>.            Crear el componente para el detalle de pagos de un préstamo.            -Archivo de lógica del negocio en <i>Typescript</i>.            -Vista en Angular creada.            Crear y ejecutar las pruebas unitarias del componente.</p>	SI
	TR3-02	Los textos en la pantalla de tabla de pagos de préstamos se deberán visualizar con el idioma del dispositivo	Crear diccionario de palabras en inglés y español para la vista	SI
	TR3-03	La aplicación contará con la pantalla de tabla de pagos de pólizas, para llegar a la misma se accederá desde la posición consolidada al ver el detalle de una póliza.	<p>Crear el modelo de datos para el detalle de pagos de una póliza.            Crear el servicio para el consumo del <i>API REST</i>.            Crear el componente para el detalle de pagos de una póliza.            -Archivo de lógica del negocio en <i>Typescript</i>.            -Vista en Angular creada.            Crear y ejecutar las pruebas unitarias del componente.</p>	SI
	TR3-04	Los textos en la pantalla de tabla de pagos de pólizas se deberán visualizar con el idioma del dispositivo	Crear diccionario de palabras en inglés y español para la vista	SI

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	TAREAS	CHE CKLIST
HUE04-01	TR3-05	La aplicación tendrá un menú flotante en la parte inferior derecha, en la página de inicio, con cuatro opciones: <i>ATMs</i> , oficinas, productos y servicios. El menú se mostrará y ocultará al presionar.	Crear los botones con las propiedades que permitan mostrarse y ocultarse sobre la pantalla de ingreso a la aplicación	SI
	TR3-06	El menú flotante tendrá la opción de <i>ATMs</i> , al presionar en este se mostrará una pantalla con botones correspondientes a las ciudades donde existen cajeros de la institución financiera.	Crear el método que dirija a la pantalla de cajeros luego de presionar el botón en el menú flotante. Crear el modelo de datos para la información de cajeros. Crear el servicio para el consumo del API REST. Crear el componente para la información de cajeros. -Archivo de lógica del negocio en Typescript. -Vista en Angular creada. Crear y ejecutar las pruebas unitarias del componente.	SI
	TR3-07	Al presionar en una ciudad de la pantalla de <i>ATMs</i> se muestran las direcciones de los cajeros disponibles	Crear el método que consuma el servicio que muestra la información de los cajeros de acuerdo con la ciudad.	SI
HUE04-02	TR3-08	El menú flotante tendrá la opción de Oficinas, al presionar en este se mostrará una pantalla con botones correspondientes a las ciudades donde existen oficinas de la institución financiera.	Crear el método que dirija a la pantalla de oficinas luego de presionar el botón en el menú flotante. Crear el modelo de datos para la información de oficinas. Crear el servicio para el consumo del API REST. Crear el componente para la información de oficinas. -Archivo de lógica del negocio en Typescript. -Vista en Angular creada. Crear y ejecutar las pruebas unitarias del componente.	SI
	TR3-09	Al presionar en una ciudad de la pantalla de Oficinas se muestra las direcciones de las oficinas existentes	Crear el método que consuma el servicio que muestra la información de las oficinas de acuerdo con la ciudad.	SI

H U	ID TAREA	CRITERIOS DE ACEPTACIÓN	TAREAS	CHE CKLIST
HUE04-03	TR3-10	El menú flotante tendrá la opción de Productos, al presionar en este se mostrará una pantalla con botones correspondientes a los tipos de productos que existen en la institución financiera.	<p>Crear el método que dirija a la pantalla de Productos luego de presionar el botón en el menú flotante.</p> <p>Crear el modelo de datos para la información de productos.</p> <p>Crear el servicio para el consumo del <i>API REST</i>.</p> <p>Crear el componente para la información de productos.</p> <p>-Archivo de lógica del negocio en <i>Typescript</i>.</p> <p>-Vista en Angular creada.</p> <p>Crear y ejecutar las pruebas unitarias del componente.</p>	SI
	TR3-11	Al presionar en un Producto de la pantalla de Productos se muestre los diferentes productos que ofrece la institución financiera	<p>Crear el método que consuma el servicio que muestra la información de cada tipo de producto.</p>	SI

## 6.11 Anexo XI: Encuestas Realizadas

### Resultados de las encuestas

¿Cree que se han facilitado sus trámites al conocer la información que se puede consultar en la aplicación móvil?

37 respuestas

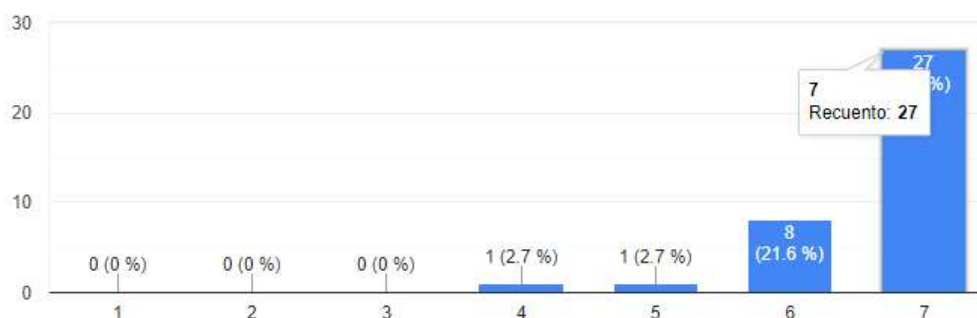


Figura 51: Respuestas de la pregunta 1 para la utilidad percibida



¿Cree que el uso del aplicativo le permite ahorrar tiempo y evitar el uso otro canal de atención para sus necesidades en la entidad financiera?



37 respuestas

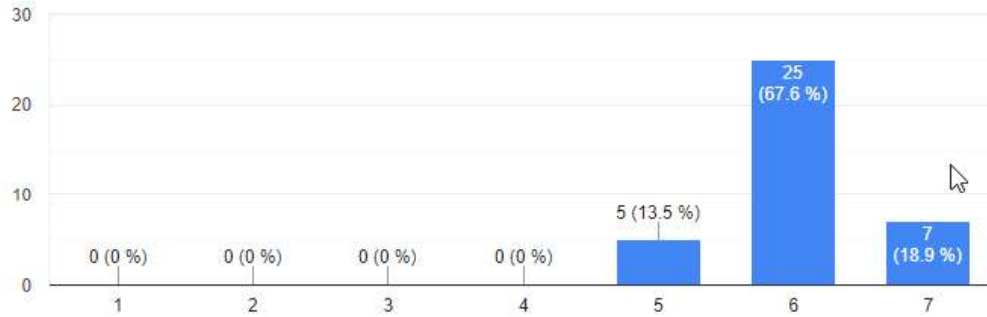


Figura 52: Respuestas de la pregunta 2 para la utilidad percibida

¿Utilizaría esta aplicación para manejar su dinero, buscar información y efectuar transferencias?

37 respuestas

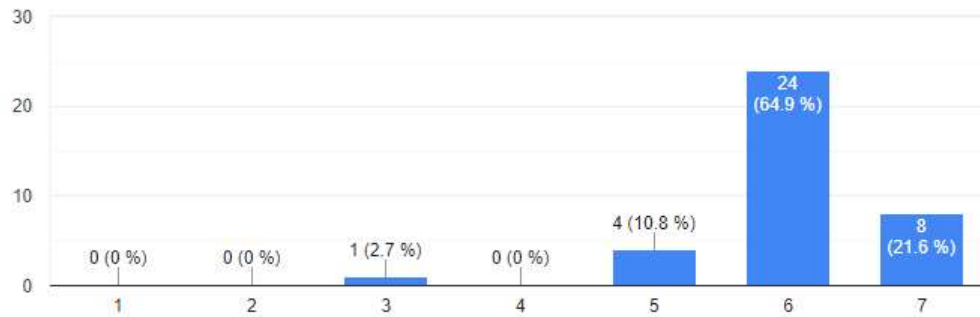


Figura 53: Respuestas de la pregunta 3 para la utilidad percibida

¿La información que ofrece la aplicación es clara y entendible?

37 respuestas

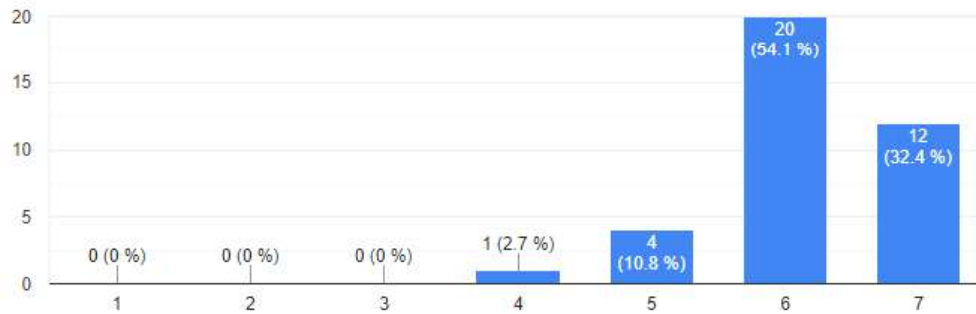


Figura 54. Respuestas de la pregunta 1 para la facilidad de uso percibida

¿Es intuitivo para usted usar la aplicación?



37 respuestas

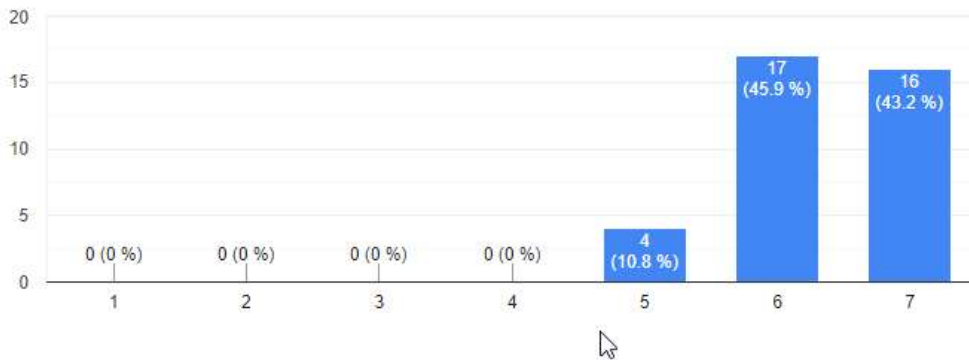


Figura 55: Respuestas de la pregunta 2 para la facilidad de uso percibida

¿Le resultó sencillo consultar la información y realizar transferencias entre distintas cuentas?



37 respuestas

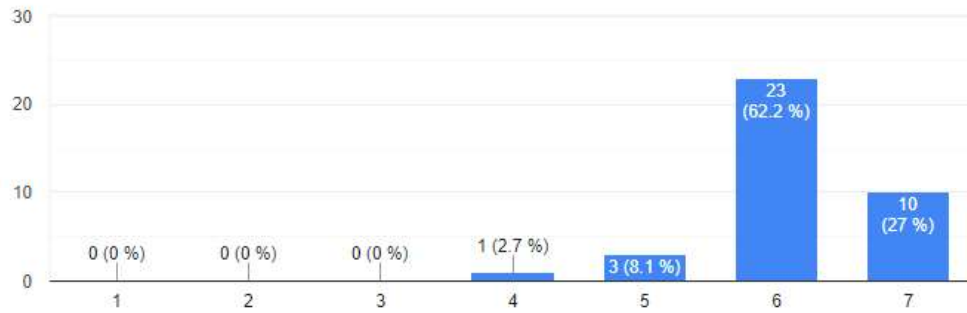


Figura 56: Respuestas de la pregunta 3 para la facilidad de uso percibida