



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

DESARROLLO APLICACION NOTIFICACIONES+

Por:
DAVID ENRIQUE PRIETO MELENDEZ

Realizado con la asesoría de:
Tutor Académico: PROF. RICARDO MONASCAL
Tutor Industrial: ING. ALEXANDER RAMIREZ

INFORME DE PASANTÍA LARGA
Presentado ante la Ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero en Computación

Sartenejas, SEPTIEMBRE de 2015



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

DESARROLLO APLICACION NOTIFICACIONES+

Por:
DAVID ENRIQUE PRIETO MELENDEZ

Realizado con la asesoría de:
Tutor Académico: PROF. RICARDO MONASCAL
Tutor Industrial: ING. ALEXANDER RAMIREZ

INFORME DE PASANTÍA LARGA
Presentado ante la Ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero en Computación

Sartenejas, SEPTIEMBRE de 2015

DESARROLLO APLICACION NOTIFICACIONES+

Por:

David Enrique Prieto Melendez

RESUMEN

El presente informe describe las actividades realizadas durante el proyecto de pasantía larga, el cual consistió en diseñar y desarrollar un prototipo funcional de la Aplicación Web de *Notificaciones+* y un prototipo funcional de la Aplicación Móvil del mismo. Este conjunto de aplicaciones están orientadas a los clientes de bancos, que tengan un dispositivo con la capacidad de desplegar un navegador web, o un dispositivo que funcione bajo las plataformas Android e iOS. Los prototipos funcionales de *Notificaciones+* permiten la recepción de mensajes clasificados de acuerdo a su tipo, que un banco desee mandar a sus clientes. También se permite a los clientes mandar una respuesta al banco según el tipo específico del mensaje, en caso de que el banco requiera una respuesta por parte de sus clientes para una operación.

Para lograr el conjunto de objetivos de *Notificaciones+* fue necesaria la implementación de la lógica de las aplicaciones para el manejo de la recepción y gestión de mensajes provenientes de la plataforma *Synergy Push*. Desde el punto de vista tecnológico se hizo uso de herramientas y tecnologías como JavaScript, AngularJS, Ionic Framework, Node.js. Los resultados de esta pasantía, realizados de acuerdo a la Metodología de Synergy-GB y documentados con artefactos de desarrollo de sistemas implantados en la empresa, incluyen el diseño y desarrollo de *Notificaciones+*, así como la integración adecuado con los servicios web y push de la plataforma *Synergy Push*.

*A mi mismo,
por enseñarme a luchar por lo que quiero, apoyarme
y darme la fortaleza para seguir adelante.*

Agradecimientos

Agradezco a Ricardo Monascal, por su disposición a ayudar y por guiarme a lo largo de la pasantía y de toda mi carrera, incluso cuando no tenía tiempo para sí mismo.

A Alejandra Facchin, por tenerme paciencia.

A Marielby Soares, por su constante ayuda y atención, sin tener ninguna razón para hacerlo.

A Miguel Fagundez, por estar siempre dispuesto a ayudarme en este proyecto.

A Ernesto Hernández-Novich, por enseñarme a ver la vida y mi profesión de otra manera.

A Synergy-GB por brindarme la oportunidad y la confianza para desarrollar este proyecto.

A mi perra Miku, por hacerme compañía cuando más lo necesitaba y darme calor en las noches.

Y a todas las personas que han contribuido en mi desarrollo profesional.

Índice General

Índice General	vii
Índice de Tablas	x
Índice de Figuras	xi
Lista de Símbolos y Abreviaturas	xii
Introducción	1
1. Entorno Empresarial	3
2. Marco Teórico	6
2.1. Notificación Push	6
2.2. Servicio web	6
2.3. Transferencia de Estado Representacional (REST)	7
2.4. Arquitectura de tres capas	7
2.5. Patrón MVC (Modelo-Vista-Controlador)	8
2.6. Software as a Service	8
2.7. WEB 2.0	9
2.8. Single Page Application (SPA)	9
3. Marco Tecnológico	10
3.1. HTML (HyperText Markup Language)	10
3.2. CSS (Cascading Style Sheets)	10
3.3. JavaScript	10
3.4. JSON	11
3.5. Node.js	11

3.6. Grunt	11
3.7. PhoneGap	11
3.8. Cordova	12
3.9. AngularJS	12
3.10. Ionic Framework	12
4. Marco Metodológico	13
4.1. Metodología de proyectos Synergy-GB	13
4.1.1. Concepción	13
4.1.2. Diseño	14
4.1.3. Construcción	14
4.1.4. Transición	14
4.2. Scrum	15
4.2.1. El equipo Scrum (Scrum team)	15
4.2.2. El Sprint	16
4.2.3. Artefactos	17
5. Desarrollo de la Aplicación	18
5.1. Concepción	18
5.1.1. Requerimientos	19
5.1.2. Riesgos	20
5.1.3. Tecnologías y plataformas de desarrollo	21
5.1.4. Casos de Uso	21
5.1.5. Planificación Inicial	22
5.2. Diseño	22
5.2.1. Diseño de la arquitectura	23
5.2.1.1. Vista de Escenarios	23
5.2.1.2. Vista Lógica	24
5.2.1.3. Vista de Desarrollo	24
5.2.1.4. Vista de Implementación	26
5.2.1.5. Vista de Procesos	27
5.2.1.6. Vista de Datos	27
5.2.2. Planificación del desarrollo	27
5.3. Construcción	27

5.3.1. Iteración 1	29
5.3.2. Iteración 2	30
5.3.3. Iteración 3	31
5.3.4. Iteración 4	32
5.3.5. Descripción general del producto	33
5.4. Transición	38
6. Retos Enfrentados y Logros Adicionales	39
6.1. Retos enfrentados	39
6.2. Logros adicionales	40
7. Conclusiones y Recomendaciones	42
Bibliografía	45
A. Especificación Funcional	47
B. Documento de Arquitectura de Software	69

Índice de Tablas

5.1.	Resumen de requerimientos funcionales de la aplicación web y de la aplicación móvil de Notificaciones+. Elaboración propia.	19
5.2.	Resumen de requerimientos no funcionales de la aplicación web y de la aplicación móvil de Notificaciones+. Elaboración propia.	20
5.3.	Iteraciones de desarrollo de Notificaciones+ y sus objetivos. Elaboración Propia.	28

Índice de Figuras

1.1. Estructura Organizacional de Synergy-GB, C.A.	4
4.1. Etapas de la metodología de proyectos Synergy-GB. (Synergy-GB Área de Calidad)	14
5.1. Diagrama de Casos de Uso inicial de Notificaciones+. Elaboración propia. . .	22
5.2. Diagrama de casos de uso definitivo de Notificaciones+. Elaboración propia.	24
5.3. Especificación del Caso de Uso “Recibir Notificación Push”. Elaboración propia.	25
5.4. Diagrama de Notificaciones+. Elaboración propia.	25
5.5. Diagrama de Componentes de Notificaciones+. Elaboración propia.	26
5.6. Diagrama de Despliegue de Notificaciones+. Elaboración propia.	26
5.7. Pantallas de Inicio de Sesión de Notificaciones. Elaboración propia.	34
5.8. Pantalla de Registro. Elaboración propia.	34
5.9. Pantalla de Menú Principal. Elaboración propia.	35
5.10. Pantalla Modificar Perfil. Elaboración propia.	35
5.11. Pantalla de Buzón con botones de filtrado versión Móvil. Elaboración propia.	36
5.12. Pantalla de Buzón con botones de filtrado versión Web. Elaboración propia.	37
5.13. Pantalla de Buzón con opciones de mensaje. Elaboración propia.	38

Lista de Símbolos y Abreviaturas

GB: Global Business

API: Application Programming Interface

SOAP: Simple Object Access Protocol

SOA: Service Oriented Architecture

REST: Representational State Transfer

HTTP: Hypertext Transfer Protocol

MVC: Modelo Vista Controlador (Model–View–Controller)

SPI: Service Provider Interface

POM: Project Object Model

JSON: JavaScript Object Notation

Introducción

Con el pasar de los años, las empresas evolucionan con el fin de ofrecer a sus clientes mejores productos. Synergy-GB es una compañía que desarrolla gran variedad de aplicaciones web y móviles, teniendo como gran parte de su clientela bancos; habiendo desarrollado, entre otras, las aplicaciones móviles bancarias del Banco Banesco y del Banco Mercantil. El mercado bancario es cada día más exigente y se encuentra en constante necesidad de innovaciones para ser más atractivo a sus clientes y poder ofrecer mejor calidad de servicios. En un inicio se ofrecían servicios de banca común (como la capacidad de consulta de saldos y balances de las cuentas de cada usuario del banco), sin la necesidad de una computadora de escritorio o portátil sino con el simple uso de un dispositivo móvil o Smartphone. Sin embargo, con el paso del tiempo y poco a poco, lo que fue bautizado como banca móvil fue expandiéndose, tomando en cuenta operaciones cada vez más complejas, como la disposición de saldos y movimientos de las tarjetas de crédito. De esta manera se hizo evidente que la mayoría de las operaciones bancarias, que en un principio debían seguir un proceso tedioso directamente en los bancos para poder realizarse, ahora pueden realizarse en el móvil de forma rápida y cómoda.

El proyecto de pasantía descrito en el siguiente informe tiene, entre sus propósitos, innovar sobre un tema en particular de la banca móvil: las notificaciones bancarias, dado que es un tema con el cual los bancos ya han experimentado explotar en diversas vías y se vuelve engoroso. Mediante el desarrollo de una aplicación móvil y una aplicación web se ofrece una solución de unificación de notificaciones. El siguiente informe mostrará no solo como se puede ofrecer un buzón digital de notificaciones bancarias, sino la expansión de la comunicación entre un banco y sus clientes, así sean por motivos de seguridad o informativos, que permita al usuario escoger opciones y comunicarse de vuelta con el banco de manera sencilla e inmediata. Todo esto en pro de que el usuario se vea beneficiado con toda esta información disponible de una forma sencilla y fácil, en un solo lugar. De esta manera Synergy-GB podrá ofrecer algo

innovador que facilite la vida de su clientela. El objetivo general del proyecto de pasantía, planteado por la empresa y logrado a lo largo de su desarrollo, fue: “Desarrollar la aplicación web y la aplicación móvil multiplataforma en PhoneGap, Ionic Framework y Angular.js para recibir notificaciones y alertas enriquecidas e interactivas”. Para alcanzarlo se plantearon los siguientes objetivos específicos:

- Diseñar la arquitectura de la aplicación basándose en la arquitectura de Synergy Push.
- Desarrollar la integración con Synergy Push.
- Desarrollar el módulo envío de notificaciones.
- Desarrollar el módulo de gestión de contenido y respuestas de los usuarios.
- Desarrollar la aplicación web tomando en cuenta las mejores prácticas del diseño web 2.0.
- Desarrollar la aplicación móvil multiplataforma, tomando en cuenta las mejores prácticas del desarrollo para dispositivos móviles y utilizando la plataforma PhoneGap, ionic, angularjs y nodejs.

Este informe presenta los resultados de la investigación, diseño e implementación de *Notificaciones+*. Se explicarán las diferentes fases del desarrollo del mismo y el proceso de transformación del concepto abstracto inicial en un prototipo funcional, junto con las decisiones de diseño tomadas a lo largo del desarrollo.

El informe está organizado de la siguiente manera: en el capítulo 1 se provee una visión general de Synergy-GB para familiarizar al lector con la misma. En el capítulo 2 se presentan algunos conceptos teóricos fundamentales. En el capítulo 3 se describen las herramientas tecnológicas utilizadas en el desarrollo del prototipo. En el capítulo 4 se describe la metodología de Synergy-GB, utilizada en la pasantía. El capítulo 5 presenta el proceso completo de diseño e implementación del canal Web de Banca. El capítulo 6 señala los retos enfrentados durante el desarrollo. Luego, se exponen las conclusiones y algunas recomendaciones derivadas del proceso de investigación y desarrollo, seguidas de las referencias bibliográficas y el glosario. Finalmente, en los Apéndices se presentan los artefactos que se realizaron a lo largo de este proyecto de pasantía.

Capítulo 1

Entorno Empresarial

En esta sección se describe el ambiente organizacional en el que se desarrolló el proyecto de pasantía de la Aplicación web y móvil de *Notificaciones+*. Se presenta la empresa Synergy-GB, sus valores, objetivos y la estructura organizativa.

Synergy-GB es una empresa perteneciente al grupo Corporativo SYNERGY-GB Corporation, dedicada al desarrollo y comercialización de productos bajo Tecnologías de Información. Estudia las tendencias a nivel de aplicaciones corporativas actuales a las empresas, a fin de ofrecer soluciones en sus mercados que estén en línea con las prioridades gerenciales y de negocio del mundo actual.

La cartera de aplicaciones va desde Soluciones Integrales Sistémicas que resuelven una problemática compleja en la empresa, hasta Soluciones Puntuales Departamentales que resuelven problemas específicos en procesos de negocio donde se ha perdido el control gerencial [1].

La misión de la empresa, según revela su portal web es “crear, desarrollar y apoyar modelos de negocio que mejoren la competitividad y productividad de sus clientes” [1]. La visión de Synergy-GB es “Convertir a Synergy-GB en una organización de alcance hispanoamericano, que combine en forma creativa, ideas, talento, tecnología, visión empresarial, innovación y excelencia en el servicio” [1]. Los valores que definen a Synergy-GB como empresa son [1]:

- Compromiso con la Calidad.
- Compromiso con la Satisfacción al Cliente.
- Compromiso con la Generación de un Legado.
- Sentido de Propiedad.
- Sentido Emprendedor.
- Integridad y Honestidad.

- Orientación a Resultados.
- Compromiso con la Innovación y el Desarrollo de nuevas ideas.
- Proactividad.
- Trabajo en Equipo.
- Socialmente Responsables.
- Comercialmente Astutos.
- Diversión como elemento catalizador.

En la Figura 1.1 se presenta la estructura organizativa de la empresa [2]. El desarrollador ocupó el puesto de pasante en el área de aplicaciones.



Figura 1.1: Estructura Organizacional de Synergy-GB, C.A.

Entre las áreas de la empresa, las siguientes son las más relevantes [3]:

- Oficina de proyectos:
Área dentro de la organización que se ocupa de centralizar y coordinar la dirección de proyectos.
- Diseño e Imagen:
Área encargada de realizar los diseños que proyecten la imagen Corporativa de la empresa y además de los diseños adecuados a la imagen de cada cliente de acuerdo a los objetivos de cada proyecto.
- Posventa:
Es el área que gestiona la relación con los clientes después que se implantan las soluciones, productos o servicios que ofrece la empresa, además se encarga de gestionar la atención de incidencias, errores, fallas, reclamos, mantenimiento, actualizaciones o requerimientos de nuestros clientes.
- Calidad e Implementación:

Área que se encarga de velar por la calidad de los servicios que ofrece la empresa de forma integral, desde los procesos, la metodología de trabajo y ejecución de proyectos hasta las encuestas de satisfacción después que se entrega una solución.

- Administración:

El área de Administración gerencia los procesos relacionados con la gestión de la información financiera y laboral de la empresa.

- Aplicaciones:

Esta área se encarga de gerenciar la fábrica de proyectos y productos en particular, la asignación del Talento a los diversos proyectos, productos o servicios de fábrica, implantación o posventa. El área desarrolla los Skills técnicos para el desarrollo de aplicaciones Móviles o Web así como la Integración de plataformas con foco continuo en la Innovación tecnológica y la entrega de soluciones de vanguardia.

- Unidad Comercial:

Esta Unidad es la encargada de la gestión de ventas y relacionamiento con clientes y prospectos, coordinando las actividades que permitan llevar una oportunidad desde su detección (proactiva o reactivamente), hasta la presentación y negociación del cierre de la misma en el marco de una metodología que garantice una alta probabilidad de éxito.

- Unidad Productos:

Esta Unidad es la máxima responsable de la gestión de productos de la organización. Su responsabilidad abarca desde la concepción de los productos hasta su maduración y consolidación. Gestiona los productos a lo largo de todo su ciclo de vida ayudando en cada momento a definir las estrategias comerciales y de mercadeo a seguir.

Capítulo 2

Marco Teórico

En este capítulo se describen los conceptos estudiados para elaborar el proyecto. Al mismo tiempo, proporciona un conocimiento de la teoría que le da significado a las soluciones encontradas durante el desarrollo del proyecto.

2.1. Notificación Push

Las Notificaciones Push son mensajes que se envían de forma directa a dispositivos móviles (Smartphones y/o tablets) con sistema operativo iOS, Android, Blackberry o Windows Phone. La recepción por parte de los dispositivos no está garantizada. [4].

2.2. Servicio web

Los servicios web son un conjunto de protocolos y estándares definidos que sirven para intercambiar datos por medio de comunicaciones remotas en redes como Internet, permitiendo la interoperabilidad entre aplicaciones desarrolladas en distintos lenguajes de programación o ejecutadas en distintas plataformas [5].

La definición de los servicios web se basa en una serie de especificaciones y estándares aceptados por la mayoría como SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) y UDDI (Universal Description, Discovery and Integration), basados en XML (Extensible Markup Language). Sin embargo también existen otros protocolos de servicios web como REST (Representational State Transfer, descrito en la sección 2.3) y

distintos formatos de intercambio de datos y mensajes adicionales a XML tales como JSON (JavaScript Object Notation) [5].

2.3. Transferencia de Estado Representacional (REST)

La Transferencia de Estado Representacional (REST, por sus siglas en inglés), es un estilo arquitectónico para sistemas distribuidos. Los sistemas REST normalmente se basan en el protocolo HTTP para definir todas las operaciones que se pueden realizar. Estas operaciones aplican sobre recursos, los cuales son entidades semánticas que representan un objeto o concepto bien definido dentro del sistema [5].

Entre los fundamentos de REST se encuentran [5]:

- Está basado en un protocolo cliente/servidor sin estado.
- Posee un conjunto de operaciones bien definidas que se aplican a todos los recursos de información.
- Ofrece una sintaxis uniforme para identificar los recursos.

2.4. Arquitectura de tres capas

La arquitectura de tres capas es una arquitectura cliente/servidor compuesta por la capas de datos, aplicación y presentación [6]. La capa de datos es responsable de almacenar, en un servidor de datos dedicado, información de la configuración y datos del negocio. La capa de aplicación, también denominada capa lógica, de negocio o servidor de aplicación, controla toda la funcionalidad, y provee procesos lógicos de negocio y acceso a los datos. La capa de presentación ofrece a los usuarios la lógica de presentación; es decir, el acceso a la información a través de una interfaz de usuario [7]. La utilización de una arquitectura de tres capas en el desarrollo de sistemas, en base a las características previamente expuestas, ofrece las siguientes ventajas:

- Brinda capacidad de reutilización de código y funcionalidades, especialmente en la capa intermedia o de negocios.
- Provee independencia entre capas y un mínimo impacto a la hora de realizar cambios.
- Facilita del mantenimiento y mejora del sistema. La arquitectura de tres capas tiene su origen a principio de los años noventa en vista de la necesidad de superar las limitaciones inherentes en una arquitectura de dos capas. La utilización de tres capas establece un diseño efectivo cuando se necesita un esquema de tipo cliente/servidor o de datos

distribuidos mediante la provisión de un mejor desempeño, flexibilidad, mantenimiento facilitado, reutilización y escalabilidad [8].

2.5. Patrón MVC (Modelo-Vista-Controlador)

Antes de proporcionar una definición de Modelo-Vista-Controlador, hay que definir un patrón de diseño. “Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software” [9]. MVC es un patrón de diseño de software que separa el código fuente en tres capas o componentes con funciones muy especiales [10]:

- **Modelo:** administra el comportamiento y los datos del dominio de la aplicación (es decir, maneja la lógica del negocio), responde las solicitudes de información acerca de su estado (provenientes, generalmente, de la vista), mientras que manipula los datos a petición del controlador.
- **Controlador:** Es la capa que interpreta eventos (usualmente acciones del usuario), informando a la vista y al modelo sobre los cambios necesarios para generar una respuesta. De manera más precisa, el controlador informa al modelo sobre cambios en la información, y a la vista sobre la manera en la que deben presentarse los datos provenientes del modelo.
- **Vista:** Es la sección encargada de “presentar” el modelo (lógica del negocio). Esta capa se centra en la interacción directa con el usuario del sistema o aplicación.

2.6. Software as a Service

SaaS es un modelo de entrega de software en el que el fabricante es responsable de la operación técnica diaria del software proporcionado a los clientes (incluyendo el mantenimiento y soporte), mientras los clientes disfrutan de los beneficios del software desde ubicaciones remotas [11]. La aparición de SaaS como un mecanismo efectivo de entrega de software crea una oportunidad para los departamentos de Tecnología de la Información (IT por sus siglas en inglés de Information Technology) de cambiar su enfoque de despliegue y soporte de aplicaciones para la gestión de los servicios que ofrecen éstas aplicaciones [12].

2.7. WEB 2.0

La web 2.0 es la revolución de los negocios en la industria de la computación causada por el movimiento de Internet como una plataforma, y un intento por entender las reglas para el éxito de esa plataforma. La principal de esas reglas es la siguiente: construir aplicaciones que aprovechen los efectos de la red para que mejoren a medida que la gente las utiliza [11]. La web 2.0 generalmente se refiere a un conjunto de patrones sociales, arquitectónicos y de diseño que resultan en la migración masiva de los negocios al Internet como plataforma. Estos patrones se enfocan en los modelos de interacción entre comunidades, personas, computadoras y software. En esencia, las filosofías y tecnologías de la web 2.0, permiten que la interacción, a niveles sin precedentes, ayude a promover la innovación, la velocidad y la sencillez; por ejemplo: aprovechar la inteligencia colectiva de la gente, descubrir y aprovechar las comunidades de interés y conectar a las personas entre si y a la información relevante de manera más eficiente [11].

2.8. Single Page Application (SPA)

Una *Single-Page-Application* (o aplicación de una sola página), es una aplicación o sitio web que maneja una sola página web, con el objetivo de proveer una experiencia de uso más fluida al usuario (similar a las aplicaciones de escritorio). En una SPA se recupera todo el código necesario para cargar una sola página, o las fuentes necesarias se cargan dinámicamente y se agregan a la página conforme se necesiten. La página no es refrescada nunca, aunque ciertas tecnologías dan la ilusión de que se está navegando en más de una página. Muchas veces interactuar con una SPA involucra comunicarse con servicios web [13].

Capítulo 3

Marco Tecnológico

En este capítulo se presentan las características principales de las herramientas tecnológicas seleccionadas para el desarrollo del proyecto de pasantía.

3.1. HTML (HyperText Markup Language)

HTML (HyperText Markup Language) es un lenguaje de marcado para describir documentos web mediante etiquetas. Consiste en una serie de etiquetas para definir la estructura de la página y mediante ella se pueden definir, textos imágenes y otros [14].

3.2. CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets), define cómo mostrar los elementos HTML, permite separar la estructura del documento de su estilo y los detalles propios del formato del mismo. Se pueden definir estos estilos y formatos en el mismo documento HTML o en un documento de formato .css aparte [15].

3.3. JavaScript

JavaScript es un lenguaje de programación interpretado que permite darle lógica al marcado (HTML) y al estilo (CSS) de las páginas web. Forma parte de las herramientas que tiene un navegador web para poder mostrar una página web dinámica [16].

3.4. JSON

JSON, acrónimo de JavaScript Object Notation, es formato de intercambio de datos liviano basado en un subconjunto del lenguaje de programación JavaScript. Este formato posee dos estructuras básicas: pares nombre/valor y listas ordenadas de valores, por lo cual es entendible con facilidad por los humanos y simple de interpretar o generar para las máquinas. Representa una de las principales alternativas al XML [17].

3.5. Node.js

Node.js es una plataforma construida sobre V8 JavaScript Runtime, el cuál es el motor de JavaScript de código abierto de Google (V8 JavaScript Engine, 2008), para construir aplicaciones de red escalables. Node.js utiliza un manejador de eventos, un modelo de I/O no bloqueante que lo hace ligero y eficiente, ideal para aplicaciones de tiempo real que manejan gran cantidad de datos y que se ejecutan a través de dispositivos distribuidos [18].

3.6. Grunt

Es un intérprete (y ejecutor) de tareas de JavaScript que se usa para la automatización de tareas repetitivas en el desarrollo de aplicaciones web para que el programador pueda concentrarse en las áreas relevantes del desarrollo. Entre estas tareas se encuentran compilación, minificación y pruebas unitarias [19].

3.7. PhoneGap

Phonegap es una solución (software libre) para construir aplicaciones móviles multiplataformas usando tecnologías web estándar, como HTML, CSS y JavaScript [20]. Además permite el acceso a los widgets de los móviles, tales como el acceso a los sensores del dispositivo, acceso a la cámara, acelerómetro y otros a través de su API. Entre los beneficios de Phonegap están:

- Incrementa la velocidad de desarrollo de aplicaciones móviles.
- Permite que desarrollando una vez las aplicaciones sean multiplataforma.
- No es necesario saber desarrollar en cada uno de los lenguajes (de cada plataforma)

móvil) para desarrollar la aplicación. Sólo es necesario tener conocimientos de programación web y tener en cuenta el API.

3.8. Cordova

Es la plataforma en la cual está basada PhoneGap. Es la que permite el uso de las funciones de los dispositivos móviles, sin importar a que plataforma pertenecen (Android, iOS, Windows Phone, BlackBerryOS). Por esa razón, la mayoría de las funciones y comandos son idénticas, y muchas documentaciones se refieren a PhoneGap y Cordova como la misma plataforma [21].

3.9. AngularJS

Es un framework (software libre) mantenido por Google y la comunidad que ayuda a la creación de aplicaciones de página única (SPA) usando HTML, CSS y JavaScript. Logra simplificar el desarrollo ya que está basado en el modelo MVC que permite estructurar el código de manera adecuada para darle cierto poder de desarrollo del lado del cliente. Es extensible y su integración es sencilla; cada característica puede ser sobrescrita o modificada según se requiera [22].

3.10. Ionic Framework

Ionic es un framework de desarrollo de aplicaciones usando HTML5 que contribuye a diseñar aplicaciones móviles que permitan dar la sensación de ser nativas en cuanto a su desempeño [23]. Se apoya en Phonegap para poder crear la aplicación y potencia así la velocidad de desarrollo de la misma. Además, viene integrado con AngularJS.

Capítulo 4

Marco Metodológico

En este capítulo se describe la metodología utilizada para el desarrollo del proyecto, la cual es una versión “ágil” de OpenUp y se denomina Metodología de Proyectos Synergy-GB; además, se describe brevemente el marco de trabajo utilizado para la gestión del desarrollo del prototipo funcional tanto de la aplicación web como de la aplicación móvil de Notificaciones+.

Para efectos de este proyecto de pasantía, son muchos los artefactos que se describen en la Metodología de Proyectos Synergy-GB que no son aplicables al mismo ya que consisten en entregables directos a algún cliente.

4.1. Metodología de proyectos Synergy-GB

Describe las herramientas y artefactos que facilitan la comunicación y alineación tanto interna (entre los integrantes del equipo de trabajo) como externa (de Synergy-GB hacia el cliente) en cuanto al seguimiento y progreso de los proyectos [24].

En la Figura 4.1 se muestran las etapas en las cuales se divide la metodología de proyectos Synergy-GB [24].

4.1.1. Concepción

En esta etapa se definen los documentos de requerimientos y el acuerdo con el cliente, acerca de los entregables que se deben realizar en el proceso de ejecución del proyecto. Los involucrados en esta etapa son el líder de proyecto (Synergy-GB) y el cliente, y la persona responsable de cada una de las actividades es el líder de proyecto [24].



Figura 4.1: Etapas de la metodología de proyectos Synergy-GB. (Synergy-GB Área de Calidad).

4.1.2. Diseño

En esta etapa se construyen los diseños de funcionalidad y servicios que se requieren para la etapa de construcción del producto. Las personas involucradas en esta etapa de la metodología son: líder de proyecto, arquitecto, diseñador, integración y calidad [24].

4.1.3. Construcción

En esta etapa del proyecto, se realiza el desarrollo de la aplicación por parte del equipo del proyecto (Integración, Ux, Móvil, Web) encargado en el área correspondiente; en este caso, el encargado designado por la empresa es el pasante. Para su gestión, se utilizó el marco de trabajo Scrum. La fase se dividió en tres (3) iteraciones, de un mes de duración cada una. En el capítulo 5 se describirá con más detalle las actividades realizadas [24].

4.1.4. Transición

La etapa de transición representa un punto de validación y cierre para el proyecto. Es en este momento cuando se realizan todas las pruebas sobre los distintos componentes de la solución, determinando que su funcionamiento es el correcto y su presentación cumple con los diseños elaborados en tempranas etapas. Por lo tanto, se llevan a cabo las pruebas de estrés por parte del equipo de Integración y Ux, y luego las pruebas de los servicios e integrales por parte del equipo de calidad involucrado en el proyecto [24].

4.2. Scrum

Es un marco de trabajo dentro del cual las personas pueden afrontar complejos problemas adaptativos, a la vez que entregan productos del máximo valor posible de forma productiva y creativa. No es una técnica de construcción de productos, sino simplemente un marco de trabajo flexible en donde se pueden emplear diversas técnicas o procesos [25].

Scrum se fundamenta en la teoría empírica de procesos, en donde se asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea una aproximación iterativa e incremental para optimizar la predictibilidad y controlar el riesgo [25].

Al fundamentarse en la teoría empírica de procesos, Scrum debe soportarse sobre los siguientes tres pilares: transparencia, inspección y adaptación [25].

4.2.1. El equipo Scrum (Scrum team)

El Scrum team está conformado por el dueño del producto, el equipo de desarrollo y el Scrum master (jefe del Scrum o facilitador). A continuación, se especifican las responsabilidades de cada uno de los integrantes del equipo [25]:

- Dueño del producto (product owner):

Es el encargado de maximizar el valor del producto y del trabajo del equipo de desarrollo; a su vez, es la única persona que puede gestionar el product backlog (pila del producto) (más tarde se especificará qué es exactamente). Es el responsable del producto desde un punto de vista comercial o de negocio.

- Equipo de desarrollo:

Es el equipo de trabajo que participa en un Sprint dentro de la práctica Scrum. Realiza las tareas de diseño y desarrollo de software. Desde un punto de vista más general, son los profesionales que realizan los incrementos de un producto, posiblemente funcional, al final de cada sprint. Sólo los integrantes del equipo de desarrollo están autorizados para realizar los incrementos del producto.

- Scrum master:

Tiene como responsabilidad asegurar que Scrum, como marco de trabajo, es entendido y llevado a cabo correctamente. Es un líder servil que se encuentra a la orden de todo el Scrum team. Por otra parte, el Scrum master es aquel puente entre agentes externos al Scrum team y este último.

4.2.2. El Sprint

El núcleo de Scrum es el Sprint, un período de tiempo de a lo sumo un mes durante el cual se crea un incremento de producto utilizable y potencialmente entregable. Cada nuevo sprint comienza inmediatamente después de la finalización del Sprint anterior [25].

Para asegurar el correcto funcionamiento del sprint, deben seguirse las siguientes normas [25]:

- No se realizan cambios que afectarían al objetivo del sprint.
- La composición del Equipo de Desarrollo se mantiene constante.
- Los objetivos de calidad no disminuyen.
- El alcance puede ser clarificado y renegociado entre el Dueño de Producto y el Equipo de Desarrollo a medida que se va aprendiendo más.

Los Sprints contienen y consisten en la reunión de planificación del sprint (Sprint Planning Meeting), los Scrums diarios (Daily Scrums), el trabajo de desarrollo, la revisión del sprint (Sprint review), y la retrospectiva del sprint (Sprint Retrospective) [25]:

- Reunión de Planificación del Sprint (Sprint Planning Meeting):

El trabajo a realizar durante el Sprint es planificado en la Reunión de Planificación de Sprint. Este plan es creado mediante el trabajo colaborativo del Equipo Scrum al completo. La Reunión de Planificación de Sprint consta de dos partes, cada una de ellas da respuesta a las siguientes preguntas, respectivamente: ¿Qué será entregado en el Incremento resultante del Sprint que comienza? ¿Cómo se conseguirá hacer el trabajo necesario para conseguir el Incremento?

- Scrum Diarios (Daily Scrums):

Es una reunión de máximo 15 minutos, para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una predicción acerca del trabajo que podría ser completado antes del siguiente.

- Revisión del Sprint (Sprint Review):

Al final del Sprint se lleva a cabo una Revisión de Sprint, para inspeccionar el Incremento y adaptar la Pila de Producto si fuese necesario. Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se ha hecho durante el Sprint.

- Retrospectiva del Sprint (Sprint Retrospective):

La Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse

a sí mismo, y crear un plan de mejoras que sean abordadas durante el siguiente Sprint.

4.2.3. Artefactos

Los artefactos de Scrum representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum, están específicamente diseñados para maximizar la transparencia de la información clave, que es necesaria para asegurar que los Equipos Scrum tengan éxito al entregar un Incremento [25]:

- Pila de Producto (Product Backlog):

La Pila de Producto es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única fuente de requerimientos para cualquier cambio a realizarse en el producto. El Dueño de Producto (Product Owner) es el responsable de la Pila de Producto, incluyendo su contenido, disponibilidad y ordenación.

- Pila de Sprint (Sprint Backlog):

La Pila de Sprint es el conjunto de elementos de la Pila de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint.

Para los efectos de esta pasantía, el dueño del producto fue Alexander Ramírez. El scrum master fue Ana Dávila y el desarrollador fue el pasante designado por la empresa, supervisado por el líder del área de movilidad Gabriel Vega.

Se desarrollaron los siguientes documentos entregables desde el punto de vista de un proyecto de software, los cuales a su vez, representan la documentación del sistema obtenido como resultado de la pasantía: Lista de Riesgos, Documento de Arquitectura de Software y Documento de Especificación Funcional.

Capítulo 5

Desarrollo de la Aplicación

Este capítulo describe las actividades realizadas en la concepción, diseño, construcción y transición del prototipo de la aplicación web y de la aplicación móvil de Notificaciones+. Está dividido en secciones, cada una de ellas representando una fase de la metodología adoptada, descrita en el capítulo anterior.

5.1. Concepción

El objetivo de esta fase consistió en recolectar la información necesaria para levantar los requerimientos del cliente para las aplicaciones a desarrollar. Por este motivo se realizaron reuniones periódicas con el mismo para determinar detalladamente dichos requerimientos y definir las características que tanto la aplicación web, como la aplicación móvil de Notificaciones+ debería poseer. Esta fase tuvo una duración de 2 semanas, y se realizaron las siguientes actividades:

- Familiarización con la empresa y el entorno laboral.
- Investigación sobre las prácticas de la empresa.
- Levantamiento de requerimientos funcionales y no funcionales.
- Identificación y mitigación de riesgos que puedan afectar el desarrollo del sistema.
- Investigación general de herramientas disponibles para ambas aplicaciones (Web y Móvil)
- Investigación de la plataforma Synergy Push.

En esta etapa del desarrollo, se inició el documento de especificación funcional y el documento de arquitectura de software de Notificaciones+ para Synergy-GB, el cual se refinó en

repetidas oportunidades de acuerdo a las solicitudes que el cliente realizó (ver Apéndices A y B, el cual incluye los requerimientos).

5.1.1. Requerimientos

La lista de requerimientos se realizó de acuerdo al plan de trabajo y a varias reuniones con el cliente para aclarar todo el objetivo de ambas aplicaciones. En la Tabla 5.1 se presenta un resumen de los requerimientos funcionales de la aplicación web y de la aplicación móvil de Notificaciones+ (véase el Apéndice B para más información).

Requerimiento	Descripción
Gestionar sesión	Un usuario debe poder iniciar sesión luego de autenticar sus credenciales y cerrar una sesión previamente iniciada.
Permitir registro	Un usuario debe poder registrarse desde su dispositivo móvil a la plataforma Synergy Push Services para poder luego iniciar sesión y usar la plataforma.
Recibir Notificación Push	El dispositivo del usuario debe poder recibir automáticamente notificaciones Push provenientes de la plataforma Synergy Push Services, una vez el usuario haya iniciado sesión.
Recibir y Abrir Mensajes Completos	Un usuario debe poder seleccionar una notificación de su buzón y abrirla para ver todo su contenido.
Devolver respuesta de una o dos opciones a mensajes que lo requieren	Un usuario debe poder seleccionar una opción de respuesta a mensajes que requieran esta acción por parte del usuario, como alertas de seguridad interactiva y notificaciones de operaciones bancarias.
Compartir Mensaje	Un usuario debe poder compartir desde el buzón y la vista de detalles, un mensaje a través de otras aplicaciones instaladas en su dispositivo móvil.
Eliminar Mensajes	Un usuario debe poder eliminar desde el buzón y la vista de detalles, cualquier mensaje que se encuentre allí.

Tabla 5.1: Resumen de requerimientos funcionales de la aplicación web y de la aplicación móvil de Notificaciones+. Elaboración propia.

Un resumen de los requerimientos no funcionales pueden verse en la Tabla 5.2 (pueden consultarse a detalle en el Apéndice B).

Requerimiento	Descripción
Funcionalidad	El sistema debe cumplir con todas las funcionalidades especificadas en los casos de uso.
Usabilidad	El sistema debe ser fácil e intuitivo de usar.
Eficiencia	El sistema debe llevar a cabo todos sus procesos de manera rápida y eficiente.

Tabla 5.2: Resumen de requerimientos no funcionales de la aplicación web y de la aplicación móvil de Notificaciones+. Elaboración propia.

5.1.2. Riesgos

El propósito de la lista de riesgos es, en primer lugar, identificar aquellos elementos que pueden afectar negativamente la ejecución del proyecto. Además, permite medir el impacto de cada riesgo y planificar estrategias de mitigación y acciones de contingencia en caso de materializarse alguno de ellos.

Los principales riesgos detectados fueron, en su mayoría, desacuerdos que se originaban por problemas de comunicación entre el desarrollador y el cliente, además de decisiones de diseño que no estaban tomadas y quedaban a improvisación por parte del desarrollador. No obstante, como en este caso el rol de cliente es asumido por la misma empresa Synergy-GB, las acciones tomadas para mitigar estos riesgos consisten en la realización de reuniones periódicas para mantener los objetivos y requerimientos claros y actualizados.

Probablemente el riesgo principal sea que la funcionalidad principal de la aplicación, que son la recepción de notificaciones Push, dependían de software de terceros, experimental, ajeno a la plataforma Ionic, del cual la compañía nunca había utilizado y no había ninguna garantía que funcionara.

Otros riesgos que influían en el desarrollo de la aplicación fueron aquellos asociados al cálculo erróneo del tiempo de desarrollo, la pendiente de la curva de aprendizaje de determinadas herramientas utilizadas para el desarrollo del proyecto, en especial AngularJS y su interacción con Cordova, y la subestimación del monto de esfuerzo a ser empleado. Todos estos riesgos fueron mitigados con estrategias de planificación que contemplaran la importancia de los componentes críticos de la aplicación y la complejidad de ciertas herramientas a utilizar (para mayor detalle de los riesgos consultar el Apéndice A).

5.1.3. Tecnologías y plataformas de desarrollo

El proyecto tenía que ser desarrollado bajo ambientes adecuados y hacer uso de tecnologías innovadoras que otorgaran el máximo de beneficios. Para la implementación de la aplicación móvil de Notificaciones+ se utilizaron las herramientas implícitas (frameworks, lenguajes y plataformas) en el desarrollo de una aplicación multiplataforma con Ionic Framework. En orden piramidal:

- Ionic Framework
- AngularJS
- JavaScript
- HTML5/CSS3
- ngCordova
- Cordova/Phonegap
- iOS/Android

Se usó Sublime como editor de texto. Toda la programación se hizo en una PC corriendo Ubuntu 14.10. Se usó un computador MAC OSX para compilar y correr bajo la plataforma iOS debido a las políticas de exclusividad de esa herramienta. También se hizo uso limitado de los respectivos emuladores Android y iOS. Los dispositivos de prueba fueron:

- iPhone 5 con iOS 7
- Samsung Galaxy Nexus con Android 4.3
- HTC One S con Android 4.1.1
- Huawei con Android 4.4

5.1.4. Casos de Uso

Los Casos de Uso proporcionan uno o más escenarios, los cuales definen la secuencia acciones que debe llevar a cabo el usuario para activar una funcionalidad de las aplicaciones.

El Modelo de Casos de Uso de la aplicación es utilizado con el fin de ilustrar tanto las funcionalidades como la relación entre ellas y los actores que las usan. En la Figura 5.1 se puede observar el Diagrama de Casos de Uso inicial, que posteriormente se refinó ya que se ampliaron las funcionalidades y se aclararon ciertos requerimientos de software.

En el mismo se observa que se identificaron dos actores. En primer lugar, el Usuario, que se encarga de hacer uso de Notificaciones+. Luego, el otro actor es Synergy Push que es la plataforma a la cual se conecta la aplicación para obtener mensajes.

5.1.5. Planificación Inicial

Esta actividad consistió en estimar una planificación para el desarrollo del proyecto: el orden en que serán implementados los Casos de Uso (CU) y el tiempo que debía tomar cada uno de ellos. La Figura 5.1 muestra el diagrama de Casos de Uso Inicial que permitió establecer la planificación.

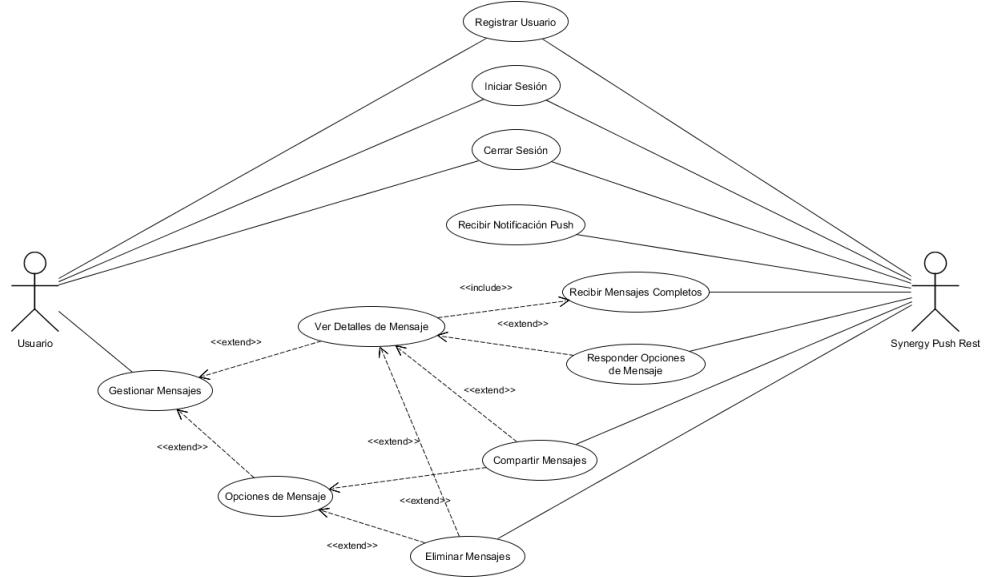


Figura 5.1: Diagrama de Casos de Uso inicial de Notificaciones+. Elaboración propia.

Luego, se estableció un plan de cuatro iteraciones de 4 semanas cada una aproximadamente. En la primera iteración se planificó hacer todas las vistas tanto su diseño como su implementación en Ionic con HTML5/CSS3, en la segunda iteración se planificó llevar a cabo los CU relacionados con “Gestionar Mensajes”, en la tercera iteración los CU relacionados con “Recibir Notificaciones Push”, “Registrarse” y “Recibir Mensajes Completos” que son las involucradas directamente con plataforma Synergy Plus. Finalmente, en la última iteración los CU que permitan iniciar y cerrar sesión y todos los demás detalles que falten.

5.2. Diseño

En esta fase, inicialmente se estudiaron los requerimientos del dueño del producto y el API REST de Synergy Plus, para así elaborar las maquetas visuales de la aplicación siguiendo los lineamientos establecidos para el desarrollo de software de la empresa. Luego se pasó a

hacer una elaboración de las vistas basadas en las maquetas mientras estas eran aprobadas por el dueño del producto y fuera elaborado el diseño definitivo por parte del departamento de diseño.

5.2.1. Diseño de la arquitectura

A la hora de diseñar la arquitectura de este sistema, se tuvo en consideración las características importantes como las siguientes:

- Escalabilidad: Debe poder instanciarse de manera fácil a cualquier banco (u otro cliente que necesite un producto como Notificaciones+), de manera que el producto sea rentable con una fácil personalización de acuerdo a lo que se solicite.
- Modularidad: Debe ser lo más modular y de bajo acoplamiento posible para evitar inconvenientes cuando se necesiten hacer cambios a la aplicación.
- Flexibilidad: Debe poder cambiar incluso su funcionalidad de la manera menos traumática posible.

5.2.1.1. Vista de Escenarios

En la Figura 5.2 se presenta el Diagrama de Casos Uso definitivo del proyecto de pasantía Notificaciones+ para Synergy-GB. Se puede ver como en el transcurrir de las iteraciones cambió el diagrama de acuerdo a nuevos requerimientos del cliente. Lo que es más pertinente de acotar es que los Casos de Uso son iguales tanto para la aplicación móvil como la aplicación web, exceptuando Recibir Notificaciones Push” que es exclusivo de la aplicación móvil.



Figura 5.2: Diagrama de casos de uso definitivo de Notificaciones+. Elaboración propia.

En la Figura 5.3 se presenta la especificación del CU “Recibir Notificación Push”. En el Apéndice B se pueden consultar las especificaciones de todos los CU del diagrama de la Figura 5.2.

5.2.1.2. Vista Lógica

Las aplicaciones (web y móvil) estarán conformadas por clases que se encargarán de definir los objetos que se manejan en el sistema y su relación. A continuación, en la Figura 5.4 se muestra el Diagrama de Clases para Notificaciones+, el cual representa la Vista Lógica de la misma.

En el diagrama de clases se pueden observar las estructuras manejadas por la plataforma Synergy Push y por lo tanto por la aplicación.

5.2.1.3. Vista de Desarrollo

La vista de desarrollo de la arquitectura de Notificaciones+ se ve reflejada en el Diagrama de Componentes que se muestra en la Figura 5.5

NP_4: Recibir Notificación Push	
Descripción: Permite al dispositivo del usuario recibir notificaciones Push de Synergy Push Services de manera automática.	
Requerimiento: NPRF_3	
Precondición: Synergy Push Services debe estar disponible.	
FLUJO BASICO:	
ACTOR	SISTEMA
1 Synergy Push Services envía una notificación Push al dispositivo móvil del cliente a través del sistema Push del OS en particular que utilice el dispositivo en cuestión.	2 Recibe la notificación y la guarda, para ser mostrado en el buzón cuando el usuario lo abra.
FLUJOS ALTERNOS: Ninguno.	
Post condición: El usuario queda registrado en Synergy Push Services.	
Requerimientos especiales: Ninguno.	
Puntos de extensión: Ninguno.	

Figura 5.3: Especificación del Caso de Uso “Recibir Notificación Push”. Elaboración propia.

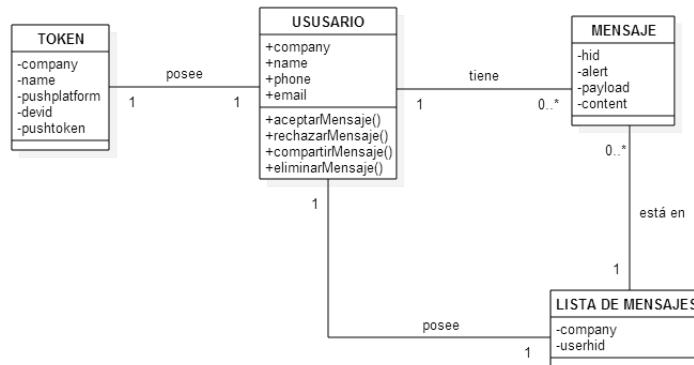


Figura 5.4: Diagrama de Notificaciones+. Elaboración propia.

En este diagrama se puede observar el patrón de diseño MVC (Model-View-Controller) ya que cada capa representa un elemento de los anteriores, se aprecia como Ionic App y AngularJS Web App representan la capa de las vistas (View). Para el caso de la Sesión y cada uno de los siguientes dos componentes: Mensaje y Buzón, representan, en conjunto, el controlador (Controller). Y para finalizar, la capa de servicios de Notificaciones+ representa los datos o el modelo como tal (Model).

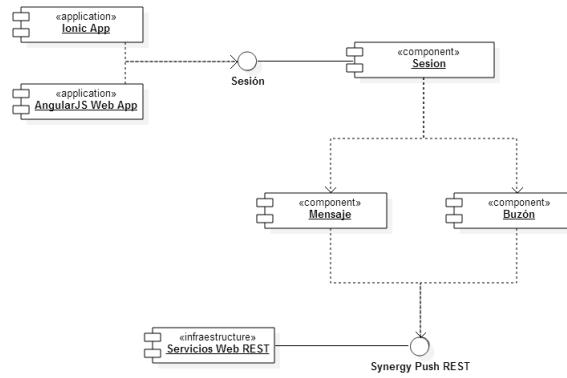


Figura 5.5: Diagrama de Componentes de Notificaciones+. Elaboración propia.

5.2.1.4. Vista de Implementación

La vista de implantación de la arquitectura de Notificaciones+ se ve reflejada en el Diagrama de Despliegue que se muestra en la Figura 5.6

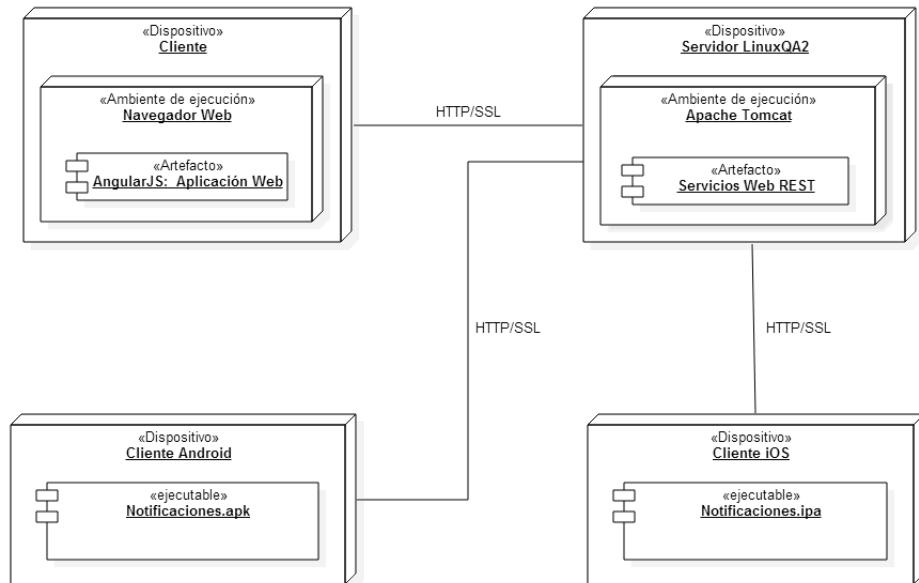


Figura 5.6: Diagrama de Despliegue de Notificaciones+. Elaboración propia.

Se puede ver como co-existen las aplicaciones móviles (para Android y iOS) junto con la aplicación web, e interactúan con los servicios web Rest cuando las aplicaciones se encuentran todas desplegadas. Es importante aclarar que la comunicación se realiza a través de HTTP debido a que la plataforma SYnergy Push todavía es un prototipo, posteriormente cada una

de estas se comunicará a través de HTTPS/SSL. La información que se comparte entre los nodos se encuentra en formato JSON.

5.2.1.5. Vista de Procesos

La Vista de Procesos no fue desarrollada, dado que la aplicación no resuelve aspectos de concurrencia de usuarios y de datos. Los mismos son resueltos por la arquitectura Synergy Push y el sistema bancario al que se instancie la aplicación.

5.2.1.6. Vista de Datos

La Vista de Datos no fue desarrollada dado que la aplicación como tal no cuenta con una base de datos propia. Esta se alimenta de datos provistos por un core bancario debido al marco de trabajo de Synergy-GB.

Es importante mencionar que cuando la aplicación sea instanciada para algún banco particular, se alimentará de los datos expuestos por dicho banco. Esos datos corresponden a los atributos de las clases del diagrama de clases mostrado en la Figura 5.4.

5.2.2. Planificación del desarrollo

Se armó un plan inicial de desarrollo basado en los objetivos del proyecto descritos en la introducción del presente informe, lo que se planteaba en el plan de trabajo y los requerimientos que se levantaron junto con el cliente. Este plan consta de cuatro iteraciones, de unas 4 semanas de duración cada una aproximadamente, el mismo sufrió varios cambios a medida que se implementaban las funcionalidades para incluir actividades que no habían sido previstas, reestablecer prioridades en el desarrollo, y otras cosas que permitieron gestionar los cambios en el proceso de construcción. En la Tabla 5.3 se describen los objetivos de cada iteración:

5.3. Construcción

Esta fase se centró en el desarrollo de las funcionalidades y Casos de Uso previamente definidos, siguiendo la planificación especificada en la Tabla 5.3. El mismo se llevó a cabo de acuerdo a los estándares de Synergy-GB promoviendo la modularidad, reutilización y gerencia de cambio (cambios de requisitos de software no planificados).

Iteración	Objetivos finales
Iteración 1	<ul style="list-style-type: none"> ■ Diseño detallado e implementación del módulo de notificaciones y alertas e integración con Synergy Push: <ul style="list-style-type: none"> ● Envío de notificaciones interactivas y recepción de respuestas ● Envío de información promocional con contenido Web ● Envío de recibos de transacciones realizadas en otros canales
Iteración 2	<ul style="list-style-type: none"> ■ Diseño detallado e implementación de la aplicación móvil y web de notificaciones y alertas: <ul style="list-style-type: none"> ● Gestión de notificaciones y respuestas ■ Diseño detallado e implementación de la aplicación móvil en phonegap, para las funcionalidades de notificación y alertas
Iteración 3	<ul style="list-style-type: none"> ■ Diseño detallado e implementación de la aplicación móvil en phonegap incluyendo todas sus funcionalidades.
Iteración 4	<ul style="list-style-type: none"> ■ Prueba de la aplicación móvil en las distintas plataformas móviles, y desarrollo de ajustes para problemas específicos de plataformas. ■ Finalización de los casos pendientes de las iteraciones anteriores.

Tabla 5.3: Iteraciones de desarrollo de Notificaciones+ y sus objetivos. Elaboración Propia.

Al finalizar cada iteración, los componentes desarrollados fueron probados para permitir la verificación de la correctitud de las funcionalidades implementadas. Al finalizar esta etapa, dichos componentes quedaron listos para pasar a las pruebas de sistemas y de aceptación, las cuales, sobre la base de la metodología que se está utilizando (Metodología de Proyectos de Synergy-GB), se hacen en la fase de transición.

5.3.1. Iteración 1

En esta iteración se realizaron las tareas de los Sprint 1, 2, 3 y 4 (pueden consultarse a detalle en el Apéndice F). Para cumplir los objetivos propuestos para esta etapa, en primer lugar, se realizó un estudio del API de la plataforma Synergy Push y de los requerimientos detallados del dueño del producto para realizar un diseño preliminar de la aplicación móvil, incluyendo las vistas. De la plataforma Synergy Push se dio a entender 2 distinciones esenciales del funcionamiento de la aplicación y de qué es lo que va a recibir y manejar:

- Notificaciones que alertan al usuario que posee un mensaje. Una notificación posee toda la información del mensaje excepto su contenido. El usuario puede recibir notificaciones de dos maneras distintas:
 - Vía Push: exclusivo para dispositivos móviles. Son recibidos siempre y cuando el usuario haya hecho login, aún así el dispositivo esté en descanso y bloqueado.
 - Vía HTTP: se reciben dentro de la aplicación, tanto móvil como web, a solicitud del usuario. Siempre se puede recibir la notificación de un mensaje mientras este no haya sido abierto.
- Mensajes: versión completa de la notificación, que posee un contenido de texto o dirección web. Una vez abierto se elimina de la plataforma Synergy Push y quedará contenido en el dispositivo móvil o en el navegador dónde el usuario lo haya abierto con la aplicación web.

En segundo lugar, una vez entregada una versión inicial de los diseños de vistas y funcionalidades, se procedió a empezar a implementar las vistas preliminares a la espera de las vistas oficiales del departamento de diseño. Una vez hechas las vistas preliminares se pasó a implementar las funcionalidades básicas de transición.

Al ser recibido el diseño de la aplicación del departamento de diseño de la empresa se pasó a adaptar las vistas ya hechas. Las vistas no eran convencionales y no se podían hacer con las utilidades que trae Ionic por defecto, así que se tuvo que hacer mucha investigación y experimentación para manipular el código CSS para lograr lo exigido.

Luego, se empezó la integración de lo ya programado a la plataforma Android y así empezar a implementar la recepción de las notificaciones Push, fundamentales para esta aplicación. Debido a la naturaleza experimental de esta funcionalidad, nueva en el framework Ionic, requirió gran cantidad de tiempo y pruebas, exclusivamente en Android. La creación de la versión móvil iOS se dejó para después de que la versión Android tuviera todas sus funcionalidades fundamentales implementadas, a petición del líder de movilidad de la empresa encargado de supervisar el proyecto.

Al finalizar esta iteración se hicieron pruebas manuales por parte del desarrollador usando directamente la plataforma Synergy Push para probar la recepción y el manejo de las notificaciones Push. Haciendo esto se descubrieron varios de los errores superados en la sección de Retos y Logros.

5.3.2. Iteración 2

En esta iteración se realizaron las tareas de los Sprint 5 y 6 (pueden consultarse a detalle en el Apéndice F), que incluyen las funcionalidades: login, apertura y recepción de mensajes, eliminar mensajes, registro y modificar perfil. Se tuvo que hacer también las vistas de registro y modificar perfil ya que no habían sido tomadas en consideración por el departamento de diseño ni el dueño del producto, para esto fue necesario que el desarrollador y el líder de departamento de diseño las hicieran en conjunto.

En esta iteración estaba planificado la implementación del grueso de las funcionalidades de la aplicación móvil multiplataforma y el comienzo de la aplicación web, con la intención de empezar la aplicación web desde cero, pero después de hecha la estructura principal se tomó la decisión de usar una plantilla web propiedad de la empresa, que no estaba lista todavía, así que todo lo que era desarrollo web fue postergado para la última iteración.

La aplicación está diseñada para trabajar en conjunto con la plataforma Synergy Push y con algún sistema del cliente que compre la aplicación, y como dicho sistema no existe todavía hay algunas cosas que tuvieron que resolverse:

- La aplicación debe iniciar sesión con la página del cliente (un banco u otro) así que se creo un mini servidor que aceptara inicio de sesión con el específico único de implementar esta funcionalidad.
- La aplicación debe registrarse con Synergy Push y el sistema del cliente al mismo tiempo así que se hizo uso nuevamente del mini servidor provisional para que aceptara registro de la aplicación tanto web como móvil.

- Los mensajes deberían ser enviados por el sistema del cliente a la plataforma Synergy Push y es esta la que lo gestiona y envía a la aplicación de los usuarios particulares. Para implementar y hacer pruebas de esto simplemente se hizo uso del modo desarrollador de Synergy Push y se enviaron los mensajes manualmente.
- Las respuestas de los mensajes interactivos son recibidas por el mismo sistema del cliente que envió dichos mensajes en primer lugar, para eso nuevamente se hizo uso del mini servidor provisional que recibiera las respuestas.

Luego se pasó a agregar guardado de mensajes persistente en la aplicación móvil, de manera que los mensajes abiertos no se perdieran hasta que el usuario los borrara la aplicación recordara un usuario.

Luego se exportó todo lo hecho tanto a la plataforma Android como a la plataforma iOS, descubriendo el error de Synergy Push para enviar notificaciones Push para iOS. La habilidad de recibir notificaciones Push por parte de la aplicación móvil fue comprobada por el desarrollador por medio de otros programas auxiliares. Todas las demás funcionalidades implementadas funcionan en ambas plataformas.

Al finalizar la iteración, se realizaron pruebas unitarias para todas las funcionalidades implementadas. Al igual que en la iteración anterior, estas pruebas fueron llevadas a cabo de forma manual utilizando la plaraforma Synergy Push y el mini servidor provisional.

5.3.3. Iteración 3

En esta iteración se realizaron las tareas de los Sprint 7 y 8 (pueden consultarse a detalle en el Apéndice F) que incluyen las funcionalidades: *pull refresh*, autenticación, compartir mensajes y botones de respuesta, además del arreglo de un bug de la recepción Push para Android detectado por el desarrollador. Esta iteración consiste básicamente en terminar los últimos detalles y resolver los errores de la aplicación móvil.

Primero se implementó la funcionalidad *pull refresh* simplemente consiste en darle poder al usuario para hacer que la aplicación buscara notificaciones nuevas desde el buzón vía HTTP sin tener que esperar que llegara la notificación vía Push. Se tuvo que hacer uso de un algoritmo muy eficiente para mezclar la lista de notificaciones entrante con la que ya estaba almacenada en el teléfono para ahorrar recursos computacionales del dispositivo.

Luego se hizo la funcionalidad autenticar, que simplemente consistió en una mejora del login ya implementado. Se hizo que la aplicación mandara una solicitud en particular al

mini servidor y que este devolviera a su vez una respuesta particular que le permitiera a la aplicación avanzar a la siguiente vista.

Para la funcionalidad de compartir solo se hizo uso del plugin de Cordova diseñado para eso, que funciona adecuadamente para ambas plataformas móviles y comparte los mensajes de Notificaciones+ a través de cualquier aplicación de mensajería o de red social que el usuario tenga instalada en su dispositivo.

Los botones de respuesta se implementaron haciendo uso del mini servidor provisional para que recibiera dichas respuestas. Se tuvo una reunión con el dueño del producto y este indicó que los mensajes se podían responder solamente una vez y así fue implementado.

La última tarea del Sprint consistía en resolver un error de recepción Push en la versión Android de Notificaciones+, dónde las notificaciones Push no son recibidas por la aplicación si esta estaba en segundo plano. Se hizo investigación y varias pruebas hasta localizar el error en el código fuente del plugin de recepción Push. Para resolver esto se tuvo que modificar dicho código fuente para adaptarlo a las necesidades de la aplicación. Se desconoce si este error se presenta también para iOS ya que como se ha mencionado antes, la plataforma Synergy Push no es capaz de mandar notificaciones Push a iOS todavía.

5.3.4. Iteración 4

En esta iteración se realizaron las tareas de los Sprint 9 y 10 (pueden consultarse a detalle en el Apéndice F) que abarca todo el desarrollo de la aplicación Notificaciones+ Web. Una vez obtenido acceso a la plantilla web de la empresa, se partió a adaptar las partes útiles las necesidades de la aplicación tanto visual como en funcionalidad. Debido a que no era necesario hacer una vista desde cero sino adaptar una, los aspectos visuales fueron sugeridos directamente por el líder del departamento de diseño sin necesidad de realizar un diseño completo.

Primero se adaptó una vista de manejador de correo electrónico perteneciente a la plantilla para hacer de vista buzón de Notificaciones+. Luego se pasó a hacer las funcionalidades. Después de toda la experiencia adquirida en el desarrollo de la aplicación en Ionic, el desarrollo web fue relativamente sencillo. Una vez completadas todas las funcionalidades de la vista buzón, incluyendo filtro por tipo de mensajes y por texto introducido por el usuario, se pasó a desarrollar las siguientes vistas.

Las vistas y funcionalidades de login, registro y modificar perfil se adaptaron igualmente de las páginas prediseñadas de la plantilla. La pantalla de modificar perfil fue una adaptación

de la de registro ya que no existía un equivalente en la plantilla. Sus funcionamientos fueron análogos a la versión móvil de Ionic, haciendo uso de las mismas técnicas de programación y del mini servidor provisional.

El último requisito que se atacó fue la vista de detalles de mensajes, y sus funcionalidades inherentes como compartir, eliminar y responder mensajes. La respuesta de mensajes es análoga a la de la aplicación móvil. La funcionalidad de eliminar mensajes no lo fue tanto ya que en la aplicación web se puede enviar los mensajes a una papelera desde el buzón antes de ser borrados y luego en la papelera se pueden eliminar varios o todos los mensajes al mismo tiempo seleccionándolos antes. Desde la vista de detalles de mensaje también se pueden borrar los mensajes pero solo definitivamente sin pasar por la papelera primero. La funcionalidad compartir requirió aplicar links directos a las redes sociales correspondientes excepto para Facebook, cuyas políticas de desarrollo obligan a registrar la aplicación Notificaciones+ para poder compartir algo.

Finalmente se hicieron pruebas manuales para verificar el correcto desempeño de todas las funcionalidades implementadas.

5.3.5. Descripción general del producto

Al finalizar la fase de construcción, se cuenta con un prototipo funcional estable de Notificaciones+ tanto web como móvil, validado por el cliente. La aplicación permite: iniciar y cerrar sesión; recibir mensajes vía push o Internet, guardar los mensajes, compartir y eliminar mensajes.

Al abrir las aplicaciones, se muestra la pantalla de inicio de sesión, en la cual deben escribirse los datos del usuario para poder iniciar su sesión y entrar a su buzón. En la Figura 5.7 se muestra las capturas de pantalla de dicha vista (tanto para móvil como para web).

Si el usuario no está registrado, se le da la opción al usuario de registrarse a la plataforma en la pantalla login. En la Figura 5.8 se muestra la captura de pantalla de esta página.

Una vez iniciada la sesión del usuario se visualiza el menú principal, donde el usuario puede acceder directamente al buzón con o sin un filtro de mensajes seleccionado. Además, se le ofrece la opción de modificar su perfil. En la Figura 5.9 se muestra la captura de pantalla de esta página.

Si el usuario selecciona la opción de modificar su perfil se le muestra dicha pantalla, con el campo nombre en gris indicando que es lo único que no puede modificar. En la Figura 5.10 se muestra la captura de pantalla de esta página, se puede acceder desde el Menú Principal

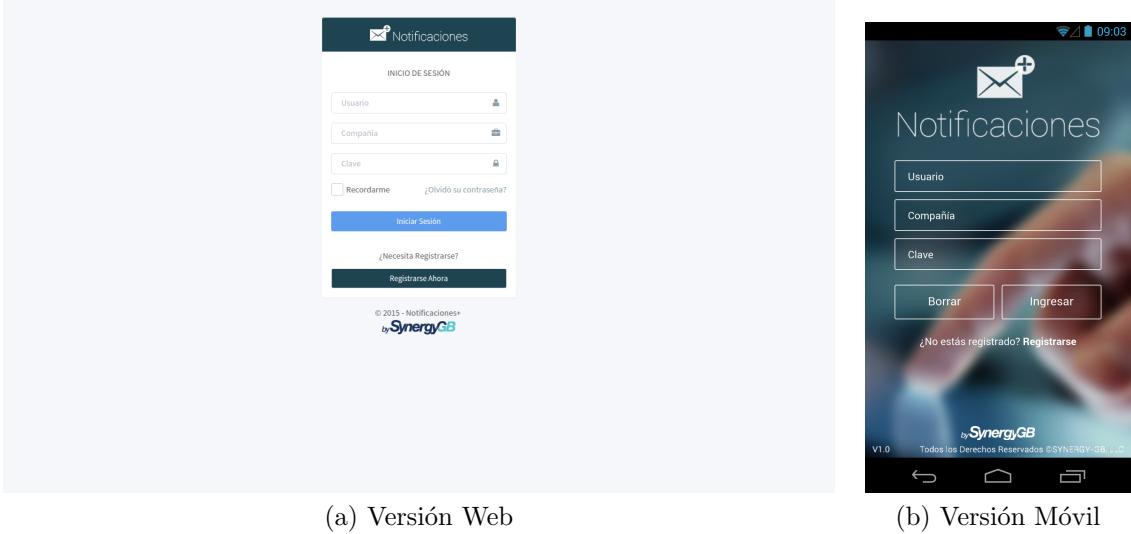


Figura 5.7: Pantallas de Inicio de Sesión de Notificaciones. Elaboración propia.

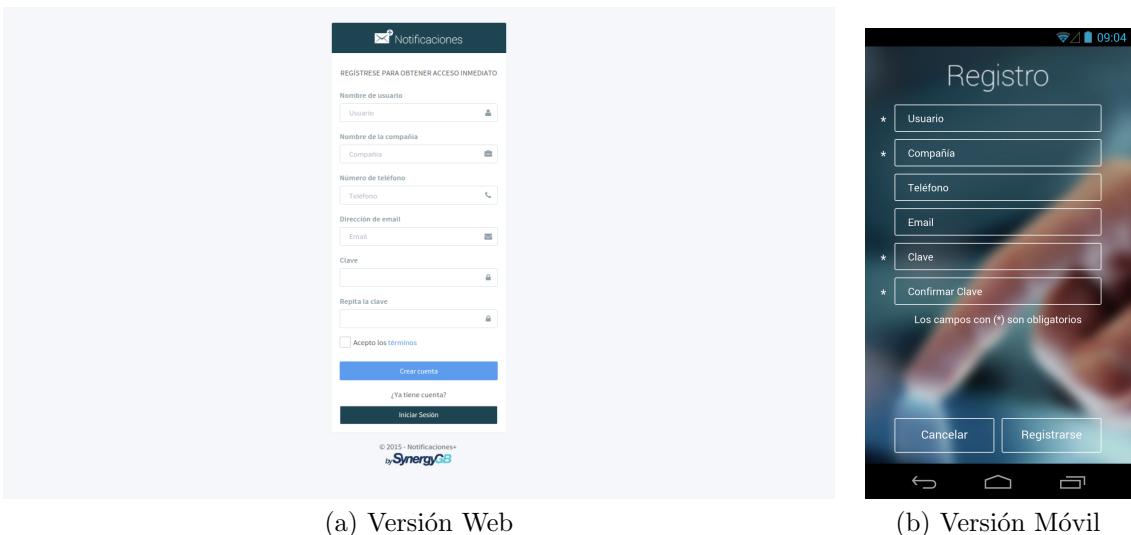


Figura 5.8: Pantalla de Registro. Elaboración propia.



Figura 5.9: Pantalla de Menú Principal. Elaboración propia.

en la aplicación móvil o desde el Buzón en la versión web.

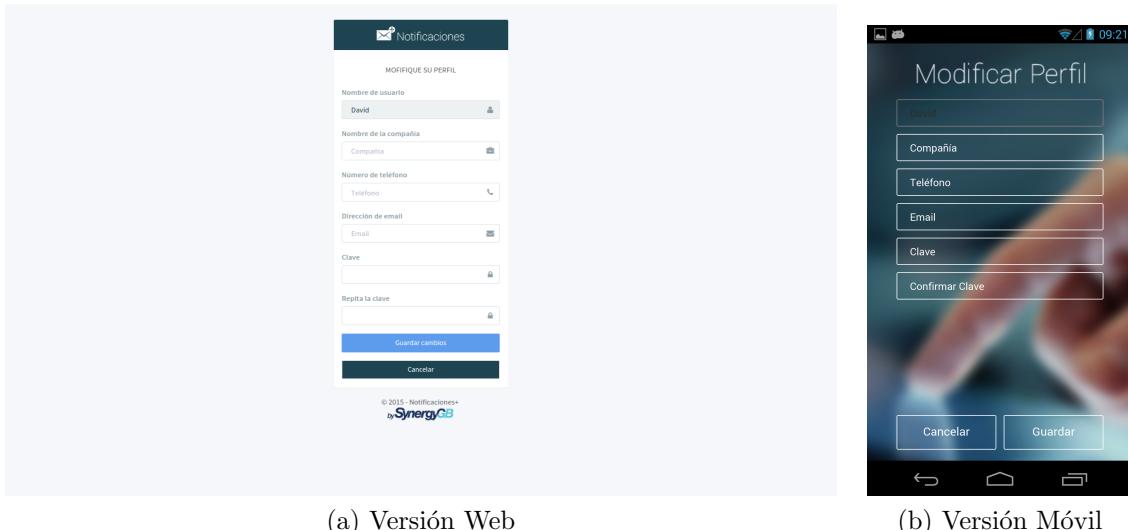


Figura 5.10: Pantalla Modificar Perfil. Elaboración propia.

En la pantalla buzón, se muestran todos los mensajes recibidos, los cuales pueden ser filtrados por tipo al presionar uno de los botones mostrados, o por el texto que contienen en el cuadro de búsqueda. En la Figura 5.11 se puede observar la pantalla de la versión móvil sin alterar y bajo la acción de los botones. En la Figura 5.12 se puede observar la pantalla de la versión web sin alterar y bajo la acción de los botones. Cabe acotar que para que aparezca el cuadro de búsqueda en la versión web hay que solicitarlo mediante el botón indicado.

Al deslizar un mensaje hacia la izquierda se muestra los botones de opción de compartir y eliminar. De seleccionarse eliminar se mostrará un menú de confirmación. El aspecto de

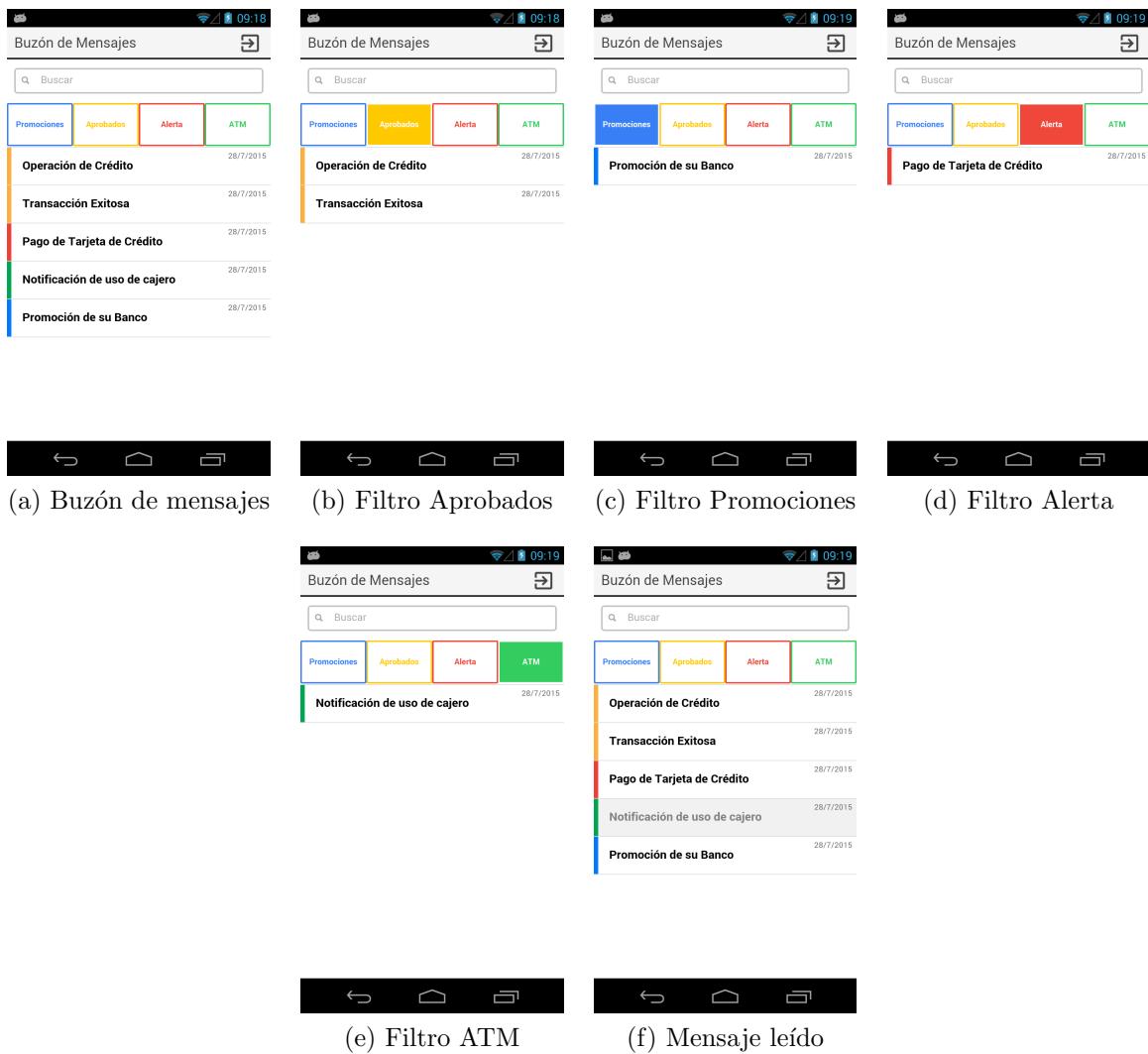
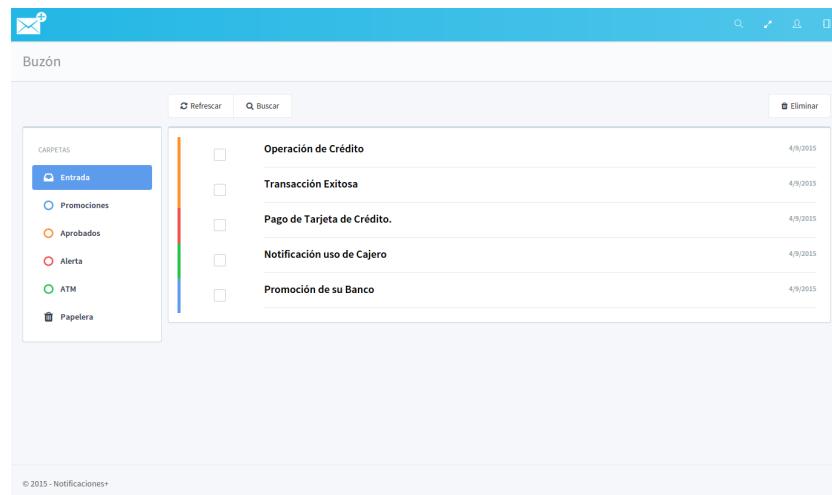
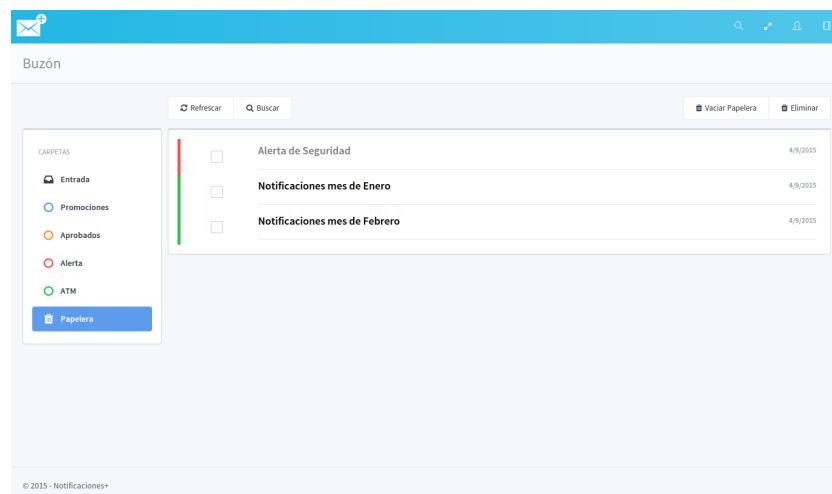


Figura 5.11: Pantalla de Buzón con botones de filtrado versión Móvil. Elaboración propia.



(a) Buzón de mensajes



(b) Papelera

Figura 5.12: Pantalla de Buzón con botones de filtrado versión Web. Elaboración propia.

los menús de confirmación y de las opciones para compartir dependen de la plataforma de la aplicación (iOS o Android). Ver la Figura 5.13.



Figura 5.13: Pantalla de Buzón con opciones de mensaje. Elaboración propia.

5.4. Transición

La fase de transición no fue llevada a cabo debido a que el objetivo del proyecto fue, desde su concepción, la elaboración de un prototipo inicial 100 % funcional. Es por esto que tanto la aplicación web como la aplicación móvil de Notificaciones+ aún no ha entrado en una fase formal de transición a producción. Sin embargo, la plataforma Synergy Push ya se encuentra en producción y ya está siendo utilizada por bancos clientes de la empresa, así que la puesta en producción de Notificaciones+ dependería del momento en que la empresa quiera hacerlo.

Capítulo 6

Retos Enfrentados y Logros Adicionales

Este capítulo muestra aquellos retos que se presentaron a lo largo de la pasantía; así como la descripción de algunos logros adicionales que no estaban en el plan inicial.

6.1. Retos enfrentados

Durante el desarrollo del proyecto de pasantía surgieron una serie de retos considerables. El primer reto en presentarse fue el trabajar con una herramienta aún nueva, con la que solo un pasante había trabajado anteriormente, y que todavía está en su infancia en su desarrollo, lo que significó defectos y errores en la plataforma que dificultaran el aprendizaje y desarrollo además de poco soporte y documentación comparado con las plataformas nativas móviles.

Luego, la interfaz proporcionada por el departamento de diseño para la aplicación móvil constituyó un gran reto ya que, aunque la aplicación es simple, la interfaz de diseño minimalista contiene muchas particularidades que no son provistas por la plataforma Ionic, así que se tuvieron que modificar estas y hacer trucos de programación para lograr los requerimientos visuales y funcionales de la interfaz. Hay dos casos en particular que cabe resaltar: los botones múltiples del menú principal y los botones filtros de la página buzón.

Los botones del menú principal fueron un reto en el sentido en que se requería una forma particular de los botones, no el estándar rectangular que provee Ionic, así que se tuvo que improvisar botones invisibles, parcialmente solapados entre si, que estuvieran libre del contexto del resto de la página para colocarlos debajo de la imagen de botones provistas

por el departamento de diseño para que funcionaran como se requería. Los botones filtros significaron tener que modificar la funcionalidad normal las cabeceras de página y de los botones provista por Ionic. Se tuvo que modificar la acción de los botones para que estos se quedaran presionados mientras ejercían su función como filtros, pero que a la vez solamente se permitiera uno presionado a la vez. Estos dos retos fueron superados gracias a la manipulación de las funcionalidades de AngularJS sobre HTML y CSS de la aplicación, con ideas privistas por la comunidad de StackOverflow.

Otro reto importante fue el mantener un buen desempeño de la aplicación, ya que posiblemente tenga que manejar grandes cantidades de mensajes. Por lo tanto se hizo uso de las herramientas que AngularJS ofrece para que la manipulación de mensajes (apertura, eliminación y guardado) sea lo más eficiente posible.

Possiblemente el reto más importante, no por su dificultad sino por lo crítico que era para la funcionalidad de la aplicación, era el uso del plugin para la recepción de notificaciones push en Cordova. Primero no se tenía mucho soporte ya que era un plugin nuevo, hecho por terceros ajenos a la plataforma Ionic, y se requirió mucho tiempo y ensayo y error para hacerlo funcionar. Después se descubrió un fallo: las notificaciones push provenientes de la plataforma Synergy Push no eran recibidas cuando la aplicación estaba en segundo plano. Se detectó el origen del problema en un fallo del diseño del código fuente del plugin de Cordova, por lo que se tuvo que modificar el código fuente para solventarlo.

6.2. Logros adicionales

Durante el desarrollo del proyecto de pasantía se alcanzaron una serie de logros adicionales que no se encontraban estipulados en el plan de trabajo inicial. Entre ellos se encuentra la realización las maquetas iniciales que normalmente corresponderían al equipo de diseño o equipo encargado de planificar el producto, pero fue delegada al pasante que entregó al dueño del producto para su aprobación y luego al equipo de diseño para la fabricación de la interfaz definitiva. El dueño del producto especificó las funcionalidades que debía tener la aplicación y a partir de estos el pasante creó los casos de uso y luego las maquetas en base a esos casos.

Para que el sistema de tipos de mensaje (Promoción, Aprobados, Alerta y ATM) y su clasificación (Respuesta Simple, Respuesta de dos Opciones, o WebView) funcionara, se tuvo que crear un estándar de identificación para que cada mensaje proveniente de Synergy Push estuviera debidamente identificado y la aplicación supiera identificarlos y tratarlos, esto fue

un logro adicional ya que la empresa no había decidido todavía como tratar a los mensajes, solo estaba hecha la plataforma Synergy Push que trata mensajes genéricos y el concepto de la aplicación. Durante este trayecto se descubrió un error en la implementación de Synergy Push que la empresa no estaba al tanto, y fue informada para su futura corrección, y mientras estas se llevaban a cargo se implementó una solución provisional dentro de la aplicación que solventara el error contenido en los mensajes, haciendo uso de expresiones regulares.

Otro defecto encontrado de la plataforma Synergy Push es que actualmente envía los mensajes sin fecha, por lo que fue necesario agregarle a Notificaciones+ la funcionalidad de agregarle fecha a los mensajes cuando son recibidos.

La modificación del código fuente del plugin de recepción de notificaciones push de Cordova se cuenta como un logro adicional. El defecto del diseño de este plugin también fue reportado en su repositorio y en la comunidad de StackOverflow.

Finalmente, otro logro adicional fue la localización de otro error, pero esta vez es en la plataforma Ionic, que fue reportado a los creadores. La plataforma Ionic posee animaciones cuando manipula elementos en una lista, y uno de estas animaciones desencadena un error fatal cuando corre bajo la plataforma iOS 7. Para solventarlo se eliminó la animación exclusivamente en esa plataforma.

Capítulo 7

Conclusiones y Recomendaciones

En este proyecto de pasantía se desarrolló la aplicación web y móvil de Notificaciones+, donde sus objetivos es ofrecer a sus usuarios una manera de recibir y recopilar todas sus notificaciones bancarias como alertas de seguridad, promociones y transacciones de cajeros automáticos. El proceso completo incluyó el diseño y desarrollo de las aplicaciones, y su conexión con la plataforma de mensajería SYnergy Push ya existente en la compañía.

Cabe acotar que las opciones y tipos de mensajes demostradas en esta aplicación a lo largo de todo el informe no son fijas y se pueden adaptar a las necesidades y preferencias de cualquier banco al que se le instancie la aplicación, incluso el cliente al que se le ofrezca la aplicación podría ser otro tipo de empresa diferente a un banco.

Para el momento de concepción de este proyecto no existía un competidor en el mercado venezolano, ya que todos los bancos notifican a sus clientes a través de mensajería SMS o e-mail.

Para poder lograr el proyecto, se usaron distintas herramientas entre las que se encuentran PhoneGap/Cordova, Ionic Framework y AngularJS para la aplicación móvil, las cuales permitieron su desarrollo sin necesidad de tener conocimientos específicos de ninguna plataforma móvil sino de programación web, y con el uso de buenas prácticas de programación para así lograr que la aplicación lograra un desempeño lo más similar posible a una aplicación nativa Android o iOS. AngularJS fue también la herramienta utilizada para desarrollar la aplicación web, usando adicionalmente Gulp para la automatización de tareas y una plantilla de aplicación web de la empresa.

Luego de 20 semanas de su inicio, al final del proyecto de pasantía, se obtuvo un prototipo funcional para la aplicación móvil. Cumple con todos los requerimientos hechos por el cliente en el plan inicial y también con las funcionalidades extras que surgieron durante el desarrollo,

ideadas por el pasante como complemento a las requerimientos básicos originales con la intención de mejorar el producto final, guíándose del ejemplo de otras aplicaciones comunes que no están relacionadas a este proyecto.

A pesar de que el producto final de este proyecto cuenta con la aprobación del cliente, el pasante considera posibles futuras mejoras y ampliaciones que le darían a la empresa la posibilidad de hacer un producto más competitivo y atractivo a los posibles bancos y demás clientes que estén interesados en adquirir el producto. Estas ampliaciones son:

- Agregar una pantalla adicional para mensajes guardados. Ya que a pesar de que los filtros en el buzón de mensajes son muy útiles, y cualquier mensaje abierto no borrado es un mensaje guardado, podría convertirse en un volumen incómodo y desagradable para el usuario.
- Ampliar la información mostrada en el ítem de mensajes en la pantalla buzón.
- Agregar la habilidad de contener archivos PDF a los mensajes, como alternativa a los mensajes Web View, para así poder mandar información como recibos de cajero y de transacciones a los mensajes y tenerlos guardados sin necesidad de poseer siempre una conexión a Internet.
- A medida de que la plataforma Ionic crezca, instanciar la aplicación a Windows Phone además de Android y iOS.
- A pesar de no ser necesario actualmente, se podría cambiar el guardado de mensajes actual simple agregando una base de datos, para así poder guardar eficientemente información necesaria en futuras ampliaciones. En su estado actual la aplicación no necesita una base de datos ya que solo guarda el texto de los mensajes, pero si se agregan funcionalidades adicionales podría ser necesaria.

Se recomienda enormemente para la salida a producción de Notificaciones+ corregir los problemas de la plataforma Synergy Push, ya que evitaría posibles futuros conflictos dentro de la aplicación y además le quitaría carga de procesamiento que usa para corregir el defecto. También se le debe agregar a los mensajes provenientes de Synergy Push la *fecha de envío*, así la aplicación no tiene que agregarles fecha que sería realmente la *fecha de recibo*.

Por último se recomienda que se realicen pruebas exhaustivas que garanticen la calidad de la aplicación, incluyendo pruebas de capacidad de carga y desempeño para ver que tan capaz es una aplicación hecha en Ionic para manejar grandes cantidades de mensajes. Además de probar la aplicación en distintos dispositivos con diferentes tamaños y versiones de sistema operativo, sobre todo en la plataforma Android.

La experiencia de la pasantía permitió al pasante poner en práctica los conocimientos adquiridos a lo largo del periodo universitario para solventar un problema real, así como aprender nuevas herramientas y tecnologías como servicios web REST, de los cuales el pasante no sabía su existencia hasta entrar en contacto con Synergy, y aunque no se generaron servicios web si se trabajó con ellos. También se generó un crecimiento en cuando a la gestión del tiempo, gestión de los cambios en los requerimientos y estimación del tiempo que llevaría realizar alguna actividad. Adicionalmente, el pasante tuvo la oportunidad de trabajar sobre una arquitectura con muchos conceptos e innovaciones. Ionic y AngularJS son herramientas que tienen muchas aplicaciones y mucho futuro, esta experiencia resultó muy beneficiosa para el pasante ya que no solo aumenta sus destrezas y habilidades en el ámbito laboral, sino que lo introdujo al mundo de aplicaciones y programación web y móvil, lo cual era su objetivo principal al aceptar la pasantía en Synergy.

Finalmente, a los futuros pasantes se les recomienda definir claramente el alcance y los requerimientos al comenzar la pasantía y mantener reuniones periódicas con el cliente o responsable del proyecto en la empresa para que el mismo apruebe lo desarrollado hasta el momento. Así mismo, se recomienda estudiar las buenas prácticas utilizadas por la empresa y tomarlas en cuenta a lo largo del desarrollo, utilizar patrones de diseño de software en caso de ser posible y consultar la documentación de las herramientas para sacar todo el provecho posible de las mismas. Tener siempre en cuenta las buenas prácticas de programación aprendidas en la carrera universitaria; tener paciencia al tratar con una plataforma nueva, ya que es algo que un ingeniero de computación o software va a tener que hacer durante toda su vida; ser ordenado y competente, siempre apuntando a la mejor manera de hacer las cosas.

Bibliografía

- [1] Synergy-GB, “Synergy-GB.” <http://synergy-gb.com>, consultado el 21 de julio de 2015, 2013.
- [2] Synergy-GB, “Organigrama de Synergy-GB.” consultado el 21 de julio de 2015, 2015.
- [3] Synergy-GB, “Áreas de la empresa.” consultado el 21 de julio de 2015, 2014.
- [4] M. Baumber, “What are push notifications?,” 2014.
- [5] L. Richardson, *RESTful Web Services for the Real World*. O'Reilly Media, Inc, 1 ed., 2007.
- [6] M. P. . P. Team, “Microsoft application architecture guide, 2nd edition.” <https://msdn.microsoft.com/en-us/library/ee658109.aspx>, consultado el 21 de abril de 2015.
- [7] A. O. Ramirez, “Three-tier architecture.” <http://www.linuxjournal.com/article/3508>, consultado el 21 de abril de 2015.
- [8] D. G. . M. Shaw, *An Introduction to Software Architecture*. Carnegie Mellon University, 1994.
- [9] N. Tedeschi, “¿Qué es un Patrón de Diseño.” <http://msdn.microsoft.com/es-es/library/bb972240.aspx>, consultado el 27 de julio de 2015.
- [10] J. G. P. H. B. W. K. S. Allen, *Professional ASP.NET MVC 3*. Apress, 2011.
- [11] J. Governor, D. Hinchcliffe, and D. Nickull, *Web 2.0 Architectures*. O'Reilly Media, Inc, 1 ed., 2009.
- [12] G. Carraro and F. Chong, “Software as a Service (SaaS): An Enterprise Perspective,” 2006.

- [13] I. Evans, “Single page application (spa) una tendencia creciente.” <http://tecnimedios.com/blog/programacion/jquery/single-page-application-spa-una-tendencia-creciente/>, consultado el 27 de julio de 2015.
- [14] “What is HTML?.” <http://www.w3.org/MarkUp/>, consultado el 12 de agosto de 2014, 2013.
- [15] “Cascading Style Sheets (CSS) Snapshot 2010.” <http://www.w3.org/TR/CSS/>, consultado el 12 de agosto de 2014, 2011.
- [16] “JavaScript.” <https://developer.mozilla.org/es/docs/Web/JavaScript>, consultado el 12 de agosto de 2014, 2014.
- [17] “Introducing JSON.” <http://www.json.org/>, consultado el 12 de agosto de 2014.
- [18] “Node.” <http://nodejs.org/>, consultado el 27 de julio de 2015.
- [19] “Grunt.” <http://gruntjs.com/>, consultado el 27 de julio de 2015.
- [20] “Phonegap.” <http://phonegap.com/about/faq/>, consultado el 27 de julio de 2015.
- [21] “Cordova.” <https://cordova.apache.org/#about>, consultado el 27 de julio de 2015.
- [22] “Angularjs.” <https://angularjs.org/>, consultado el 27 de julio de 2015.
- [23] “Ionic framework.” <http://ionicframework.com/docs/overview/>, consultado el 27 de julio de 2015.
- [24] “Metodología de Poyectos Synergy-GB [Enfoque en Seguimiento].” consultado el 27 de Julio de 2015.
- [25] J. S. Ken Schwaber, “The Scrum Guide.” <https://www.scrum.org/scrum-guide>, consultado el 27 de Julio de 2015, 2013.

Apéndice A

Especificación Funcional



SYNERGY-GB

Especificación Funcional

Notificaciones Enriquecidas e Interactivas

Tabla de contenidos

Introducción	2
Alcance	2
Glosario	2
Descripción de los Casos de Uso.....	2
Tabla de Actores/Interesados.....	3
Tabla de Casos de Uso	3
Diagrama de Casos de Uso.....	4
Especificaciones de los Casos de Uso	5
Requerimientos Funcionales	11
Requerimientos No Funcionales.....	19
Funcionalidad	19
Usabilidad	19
Eficiencia.....	19
Mantenibilidad	20
Aspectos sobre la Organización	20
Aspectos Legales	20

Introducción

Alcance

El presente documento contiene las especificaciones de los requerimientos de software de Notificaciones Enriquecidas e Interactivas, tanto funcionales como no funcionales, las cuales se encuentran asociadas a todos Casos de Uso definidos para el mismo.

Glosario

Descripción de los Casos de Uso

En esta sección describen los casos de uso del sistema.

Número	Nombre del Caso de Uso	Descripción
NP_1	Iniciar sesión	Permite a cualquier persona registrada, entrar en sesión personalizada para realizar las operaciones deseadas.
NP_2	Cerrar sesión	Permite al usuario que posee una sesión activa, cerrarla una vez terminadas las operaciones que estuviera realizando o en cualquier momento que desee, posterior a haber iniciado sesión.
NP_3	Registrar Usuario	Permite al usuario registrarse en el sistema.
NP_4	Recibir Notificación Push	Permite al dispositivo del usuario recibir notificaciones Push de Synergy Push Services de manera automática.
NP_5	Gestionar Mensajes	Permite al usuario decidir qué hacer con las notificaciones recibidas.
NP_6	Opciones de Mensaje	Permite al usuario ver las opciones de compartir y eliminar mensaje para escoger una.
NP_7	Ver Detalles de Mensaje	Permite al usuario abrir un mensaje y ver su contenido completo.
NP_8	Recibir Mensaje Completo	Permite al usuario recibir todo el contenido de una notificación que se le mande a través de Synergy Push. Se hace de manera automática al abrir el mensaje.
NP_9	Responder Opciones de Mensaje	Permite al usuario responder a la o las opciones de notificaciones que así lo requieran.
NP_10	Compartir Mensaje	Permite al usuario compartir cualquier notificación.
NP_11	Eliminar Mensaje	Permite al usuario eliminar cualquier notificación.

Tabla de Actores/Interesados

En esta sección se describen los stakeholders.

Nombre	Descripción	Responsabilidades
Pasante de Proyecto	Es la persona designada por la empresa para realizar el diseño y desarrollo de la aplicación.	Diseñar e implementar la aplicación.
Líder de Investigación y desarrollo	Es quien supervisa el desarrollo del proyecto, propone y aprueba las decisiones acerca de la arquitectura del sistema.	Supervisar el desarrollo del proyecto. Verificar que la planificación se cumpla. Aprobar decisiones de alto nivel acerca de la arquitectura del sistema.
Cliente	Es la persona encargada de definir los requerimientos según los cuales se desarrollara la aplicación.	Contribuir al levantamiento de requerimientos y la confirmación de los mismos durante todo el desarrollo.
Cliente del Banco	Es el usuario final de la aplicación.	Recibir notificaciones del banco u otras entidades.

Tabla de Casos de Uso

A continuación se presenta la tabla de casos de uso:

Número	Nombre	Actor
NP_1	Iniciar sesión	Cliente del Banco, Synergy Push Services
NP_2	Cerrar sesión	Cliente del Banco, Synergy Push Services
NP_3	Registrar Usuario	Cliente del Banco, Synergy Push Services
NP_4	Recibir Notificación Push	Synergy Push Services
NP_5	Gestionar Mensajes	Cliente del Banco, Synergy Push Services
NP_6	Opciones de Mensaje	Cliente del Banco, Synergy Push Services

NP_7	Ver Detalles de Mensaje	Cliente del Banco, Synergy Push Services
NP_8	Recibir Mensaje Completo	Cliente del Banco, Synergy Push Services
NP_9	Responder Opciones de Mensaje	Cliente del Banco, Synergy Push Services
NP_10	Compartir Mensaje	Cliente del Banco
NP_11	Eliminar Mensaje	Cliente del Banco

Diagrama de Casos de Uso

A continuación se presenta el Diagrama de los Casos de uso del sistema:



Especificaciones de los Casos de Uso

NP_1: Iniciar sesión	
Descripción: Permite a cualquier persona registrada, entrar en sesión personalizada para realizar las operaciones deseadas.	
Requerimiento: NPRF_1	
Precondición: Synergy Push Services debe estar disponible.	
FLUJO BASICO:	
ACTOR	SISTEMA
1 El cliente del banco ingresa a la vista inicial de la aplicación móvil o web.	2 Muestra el formulario de inicio de sesión.
3 El cliente del banco introduce su usuario y contraseña y presiona el botón acceder.	4 Realiza una petición POST al servicio “login” de Synergy Push Services.
5 Synergy Push Services envía la respuesta a la petición en formato JSON.	6 Recibe y extrae la información del JSON y muestra el buzón del cliente.
FLUJOS ALTERNOS:	
ACTOR	SISTEMA
	6.1.A Existe un error en la comunicación con el servicio, se redirige a la página de inicio de sesión mostrando el mensaje “Se ha perdido comunicación con el servicio”.
	6.1.B Encuentra un error en la respuesta y muestra dicho mensaje de error.
	6.1.C Encuentra en la respuesta del servicio que el cliente no tiene datos asociados y muestra un mensaje indicando que el usuario no está registrado.
Post condición: Se levanta una sesión para el usuario luego de ser autenticado.	
Requerimientos especiales: Ninguno	
Puntos de extensión:	

NP_2: Cerrar Sesión	
Descripción: Permite a un cliente del banco que posee una sesión activa, cerrarla cuando lo desee.	
Requerimiento: NPRF_1	
Precondición: Synergy Push Services debe estar disponible.	
FLUJO BASICO:	
ACTOR	SISTEMA

1 El cliente del banco selecciona la opción Cerrar Sesión en la esquina superior de la vista de buzón.	2 Realiza una petición POST al servicio “logout” de Synergy Push Services.
3 Synergy Push Services envía la respuesta a la petición en formato JSON.	4 Recibe la respuesta y redirige a la página de inicio de sesión.
FLUJOS ALTERNOS:	
ACTOR	SISTEMA
	4.1.A Existe un error en la comunicación con el servicio, se redirige a la página de inicio de sesión mostrando el mensaje “Se ha perdido comunicación con el servicio”
	4.1.B Encuentra un error en la respuesta y muestra dicho mensaje de error en una ventana emergente.
Post condición: El sistema ha cerrado la sesión del usuario.	
Requerimientos especiales: Ninguno	
Puntos de extensión: Ninguno	

NP_3: Registrarse	
Descripción: Permite al usuario registrarse en el sistema.	
Requerimiento: NPRF_2	
Precondición: Synergy Push Services debe estar disponible.	
FLUJO BASICO:	
ACTOR	SISTEMA
1 El cliente del banco ingresa a la vista inicial de la aplicación móvil o web y selecciona la opción de registrarse.	2 El cliente es redirigido a la vista de registro.
3 El cliente del banco introduce el usuario y la contraseña que escoja dos veces.	4 Realiza una petición POST al servicio “register” de Synergy Push Services.
5 Synergy Push Services envía la respuesta a la petición en formato JSON.	6 Recibe la respuesta y redirige a la página de inicio de sesión mostrando el mensaje “Registro satisfactorio”
FLUJOS ALTERNOS:	
ACTOR	SISTEMA
	6.1.A Existe un error en la comunicación con el servicio, se redirige a la página de inicio de sesión mostrando el mensaje “Se ha perdido comunicación con el servicio”
	6.1.B Encuentra un error en la respuesta y muestra dicho mensaje de error en una ventana emergente.

	6.1.C Encuentra en la respuesta del servicio que el nombre de usuario ya está registrado y muestra un mensaje acorde.
Post condición:	El usuario queda registrado en Synergy Push Services.
Requerimientos especiales:	Ninguno.
Puntos de extensión:	Ninguno.

NP_4: Recibir Notificación Push

Descripción: Permite al dispositivo del usuario recibir notificaciones Push de Synergy Push Services de manera automática.

Requerimiento: NPRF_3

Precondición: Synergy Push Services debe estar disponible.

FLUJO BASICO:

ACTOR	SISTEMA
1 Synergy Push Services envía una notificación Push al dispositivo móvil del cliente a través del sistema Push del OS en particular que utilice el dispositivo en cuestión.	2 Recibe la notificación y la guarda, para ser mostrado en el buzón cuando el usuario lo abra.

FLUJOS ALTERNOS: Ninguno.

Post condición: El usuario queda registrado en Synergy Push Services.

Requerimientos especiales: Ninguno.

Puntos de extensión: Ninguno.

NP_5: Gestionar Mensajes

Descripción: Permite al usuario ver su buzón de mensajes y decidir qué hacer con ellos, abrir, compartir o eliminar.

Requerimiento: NPRF_4, NPRF_5, NPRF_6, NPRF_7.

Precondición: Synergy Push Services debe estar disponible.

FLUJO BASICO:

ACTOR	SISTEMA
1 El cliente del banco entra en su buzón.	2 Muestra la vista de buzón.

FLUJOS ALTERNOS: Ninguno.

Post condición: Se presenta al cliente su buzón.

Requerimientos especiales: Ninguno

Puntos de extensión: Opciones de Mensaje y Ver Detalles de Mensaje.

NP_6: Opciones de Mensaje

Descripción: Permite al usuario ver las opciones de compartir y eliminar mensaje para escoger una.

Requerimiento: NPRF_6, NPRF_7

Precondición: Debe haber al menos un mensaje en el buzón.

FLUJO BASICO:

ACTOR	SISTEMA
1 El cliente del banco deja presionado por un momento sobre el mensaje en la vista de buzón.	2 Despliega un pequeño menú con las opciones compartir y eliminar.

FLUJOS ALTERNOS: Ninguno.

Post condición: Se presenta al cliente del banco un sub menú en pantalla.

Requerimientos especiales: Ninguno

Puntos de extensión: Compartir Mensaje y Eliminar Mensaje.

NP_7: Ver Detalles de Mensaje

Descripción: Permite a un a un cliente del banco ver un mensaje con todo su contenido.

Requerimiento: NPRF_4

Precondición: Synergy Push Services debe estar disponible y debe haber al menos un mensaje en el buzón.

FLUJO BASICO:

ACTOR	SISTEMA
1 El cliente selecciona un mensaje de su buzón.	2 Muestra la vista de detalles de mensaje.

FLUJOS ALTERNOS: Ninguno.

Post condición: Muestra la información del mensaje y su contenido.

Requerimientos especiales: Responder Opciones de Mensaje.

Puntos de extensión: Responder Opciones de Mensaje, Compartir Mensaje, Eliminar Mensaje.

NP_8: Recibir Mensaje Completo

Descripción: Permite al usuario recibir todo el contenido de una notificación.

Requerimiento: NPRF_4

Precondición: Synergy Push Services debe estar disponible y debe haber al menos un mensaje en el buzón.

FLUJO BASICO:

ACTOR	SISTEMA
1 El cliente selecciona un mensaje de su buzón.	2 Redirige a la vista de detalles del mensaje y realiza una petición GET al servicio “message” de Synergy Push Services.

3 Synergy Push Services envía la respuesta a la petición en formato JSON.	4 Recibe y extrae la información del JSON y muestra la información completa contenida en el mensaje en la vista detalle del mensaje.
FLUJOS ALTERNOS:	
ACTOR	SISTEMA
	4.1.A Existe un error en la comunicación con el servicio, se redirige a la vista de buzón mostrando el mensaje "Se ha perdido comunicación con el servicio"
	4.1.B Encuentra un error en la respuesta y muestra dicho mensaje de error.
Post condición: Muestra la información del mensaje y su contenido.	
Requerimientos especiales: Responder Opciones de Mensaje.	
Puntos de extensión: Responder Opciones de Mensaje, Compartir Mensaje, Eliminar Mensaje.	

NP_9: Responder Opciones de Mensaje.

Descripción: Permite al usuario responder a la o las opciones de notificaciones que así lo requieran.

Requerimiento: NPRF_5

Precondición: El usuario debe estar en la vista de detalles de mensaje.

FLUJO BASICO:

ACTOR	SISTEMA
1 El usuario presiona el botón o uno de los botones (en caso de haber 2) que muestra una determinada opción.	2 Pide al usuario que introduzca su contraseña.
3 El usuario introduce su contraseña.	4 Autentica la contraseña, redirige a la vista de buzón y manda una petición POST al servicio donde se originó el mensaje.

FLUJOS ALTERNOS:

ACTOR	SISTEMA
	4.1.A Existe un error en la comunicación con el servicio, mostrando el mensaje "Se ha perdido comunicación con el servicio"
	4.1.B Hay fallo de autenticación, muestra un mensaje diciendo "contraseña incorrecta" y vuelve a pedir la contraseña al usuario.

Post condición: Se presenta al cliente del banco la vista de buzón.

Requerimientos especiales: Ninguno

Puntos de extensión: Ninguno.

NP_10: Compartir Mensajes

Descripción: Permite al usuario compartir cualquier mensaje.

Requerimiento: NPRF_6

Precondición: Debe haber al menos un mensaje en el buzón y el usuario debe haber desplegado el menú de opciones.

FLUJO BASICO:

ACTOR	SISTEMA
1 El usuario escoge la opción para compartir del menú desplegado.	2 Despliega un menú que contiene aplicaciones contenidas en el dispositivo para compartir el mensaje.
3 El usuario escoge una aplicación.	3 Envía a la aplicación escogida el texto, dirección URL y/o imagen contenida en el mensaje.

FLUJOS ALTERNOS: Ninguno.

Post condición: Se presenta al cliente del banco la vista de detalles de mensaje.

Requerimientos especiales: Ninguno

Puntos de extensión: Ninguno.

NP_11: Eliminar Mensaje

Descripción: Permite al usuario eliminar cualquier notificación que tenga en su buzón.

Requerimiento: NPRF_7

Precondición: Debe haber al menos un mensaje en el buzón y debe de haber sido recibido por el usuario.

FLUJO BASICO:

ACTOR	SISTEMA
1 El usuario escoge la opción para eliminar del menú desplegado.	2 Despliega una ventana emergente de verificación de eliminar mensaje.
3 El usuario selecciona la opción eliminar de la ventana emergente desplegada.	4 Borra el mensaje de la aplicación.

FLUJOS ALTERNOS: Ninguno.

Post condición: Se presenta al cliente del banco la vista de detalles de mensaje.

Requerimientos especiales: Ninguno

Puntos de extensión: Ninguno.

Requerimientos Funcionales

A continuación se presenta un cuadro con la leyenda de la estructura de datos a presentar en cada uno de los requerimientos:

Campo = Nombre del campo en la pantalla.	Descripción = Descripción del campo.
Tipo = (A) alfa, (N) numérico, (AN) alfanumérico, (F) Fecha, (B) Booleano	L: Longitud = Tamaño del campo.
RVP: Rango de Valores Posibles = Rangos aceptables.	VPD: Valor por Defecto = Valor preestablecido para el campo.
I/O = Si el campo es de entrada o salida.	R/O = Si el campo es requerido u opcional.
Validaciones = Referencia a validaciones al campo que no correspondan con reglas de negocio.	

En la especificación de los mismos, no se colocaran las interfaces de error que corresponden a una notificación de tipo alert, sino únicamente las comprendidas dentro de la interfaz de Notificaciones Enriquecidas e Interactivas como tal.

- < NPRF_1 – Gestionar sesión >

Característica Asociada: El usuario podrá iniciar y cerrar sesión.
Descripción del requerimiento funcional: Un usuario debe poder iniciar sesión luego de autenticar sus credenciales y cerrar una sesión previamente iniciada.
Interfaz del requerimiento:



Estructura de datos:

Campo	Descripción	T	L	RPV	VPD	I/O	R/O	Validaciones
Usuario	Nombre del cliente del banco	AN				I	R	
Contraseña	Contraseña del cliente del banco.	AN				I	R	

Prioridad:

(X) Alta () Media alta () Media () Media baja () Baja

- < NPRF_2 – Permitir Registro >

Característica Asociada:

El usuario podrá registrarse a la plataforma Synergy Push Services desde su dispositivo móvil.

Descripción del requerimiento funcional:

Un usuario debe poder registrarse desde su dispositivo móvil a la plataforma Synergy Push Services para poder luego iniciar sesión y usar la plataforma.

Estructura de datos:

Campo	Descripción	T	L	RPV	VPD	I/O	R/O	Validaciones
Usuario	Nombre del cliente del banco	AN				I	R	
Contraseña	Contraseña del cliente del banco.	AN				I	R	
Confirmar Contraseña	Contraseña del cliente del banco.	AN				I	R	

Prioridad:

(X) Alta () Media alta () Media () Media baja () Baja

- < NPRF_3 – Recibir Notificación Push >

Característica Asociada:

El usuario podrá recibir notificaciones Push automáticas.

Descripción del requerimiento funcional:

El dispositivo del usuario debe poder recibir automáticamente notificaciones Push provenientes de la plataforma Synergy Push Services, una vez el usuario haya iniciado sesión.

Estructura de datos:

Campo	Descripción	T	L	RPV	VPD	I/O	R/O	Validaciones
Título	Título de la notificación. Es el contenido del campo alert en la estructura de datos del mensaje.	AN				O	R	

Prioridad:

(X) Alta () Media alta () Media () Media baja () Baja

- < NPRF_4 – Recibir y Abrir Mensajes Completos >

Característica Asociada:

El usuario podrá abrir y ver en detalle el contenido de una notificación recibida.

Descripción del requerimiento funcional:

Un usuario debe poder seleccionar una notificación de su buzón y abrirla para ver todo su contenido.

Interfaz del requerimiento:



Estructura de datos de un mensaje:

Campo	Descripción	T	L	RPV	VPD	I/O	R/O	Validaciones
Título	Título de la notificación.	AN				O	R	
Contenido	Contenido del mensaje.	AN				O	O	

Prioridad:

(X) Alta () Media alta () Media () Media baja () Baja

- < NPRF_5 – Devolver respuesta de una o dos opciones a mensajes que lo requieran >

Característica Asociada:

El usuario podrá seleccionar una respuesta a los mensajes que lo soliciten.

Descripción del requerimiento funcional:

Un usuario debe poder seleccionar una opción de respuesta a mensajes que requieran esta acción por parte del usuario, como alertas de seguridad interactiva y notificaciones de operaciones bancarias.

Interfaces del requerimiento:



Estructura de datos:

Campo	Descripción	T	L	RPV	VPD	I/O	R/O	Validaciones
Título	Título de la notificación.	AN				O	R	
Contenido	Contenido del mensaje.	AN				O	O	
Aprobar	Botón verde para aprobar la solicitud.			true, false		O	R	
Rechazar	Botón rojo para rechazar la solicitud.			true, false		O	R	

Prioridad:

(X) Alta () Media alta () Media () Media baja () Baja

- < NPRF_6 – Compartir Mensaje >

Característica Asociada:

El usuario podrá compartir cualquier mensaje.

Descripción del requerimiento funcional:

Un usuario debe poder compartir desde el buzón y la vista de detalles, un mensaje a través de otras aplicaciones instaladas en su dispositivo móvil.

Interfaces del requerimiento:



Interfaces del requerimiento:

Prioridad:

Alta

Media alta

Media

Media baja

Baja

- < NPRF_7 – Eliminar Mensajes >

Característica Asociada:
El usuario podrá eliminar cualquier mensaje de su buzón.
Descripción del requerimiento funcional: Un usuario debe poder eliminar desde el buzón y la vista de detalles, cualquier mensaje que se encuentre allí.
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media alta <input type="checkbox"/> Media <input type="checkbox"/> Media baja <input type="checkbox"/> Baja

Requerimientos No Funcionales

Funcionalidad

ID Requerimiento	RNF-1			
Nombre del Requerimiento	Funcionalidad			
Características	EL sistema debe poder hacer lo que se especifica en los casos de uso.			
Descripción del Requerimiento	El sistema debe cumplir con todas las funcionalidades especificadas en los casos de uso.			
Prioridad:				
(X) Alta	() Media alta	() Media	() Media baja	() Baja

Usabilidad

ID Requerimiento	RNF-2			
Nombre del Requerimiento	Usabilidad			
Características	EL sistema debe ser usable.			
Descripción del Requerimiento	El sistema debe ser fácil e intuitivo de usar.			
Prioridad:				
(X) Alta	() Media alta	() Media	() Media baja	() Baja

Eficiencia

ID Requerimiento	RNF-3
Nombre del Requerimiento	Eficiencia
Características	El sistema debe llevar a cabo todos sus procesos de manera rápida y eficiente.
Descripción del Requerimiento	El sistema debe llevar a cabo todos sus procesos de manera rápida y eficiente.

Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media alta	<input type="checkbox"/> Media	<input type="checkbox"/> Media baja	<input type="checkbox"/> Baja
------------	--	-------------------------------------	--------------------------------	-------------------------------------	-------------------------------

Mantenibilidad

ID Requerimiento	RNF-3			
Nombre del Requerimiento	Eficiencia			
Características	El sistema debe tener una estructura mantenible a lo largo del tiempo.			
Descripción del Requerimiento	El sistema debe poseer todos los módulos, arquitecturas y estructuras necesarias para que el mismo sea mantenible al pasar el tiempo.			
Prioridad:				
<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media alta	<input type="checkbox"/> Media	<input type="checkbox"/> Media baja	<input type="checkbox"/> Baja

Aspectos sobre la Organización

No aplica.

Aspectos Legales

- Synergy-GB será dueña de todo lo que desarrolle durante la pasantía.
- No se podrán divulgar los detalles específicos de la implementación, el código, los datos de las bases de datos, la configuración de servidores ni cualquier otra información sensible que pueda ser utilizada por la competencia de Synergy-GB o agentes maliciosos para perjudicar al producto desarrollado durante la pasantía o la empresa.
- En el libro de pasantía se podrá utilizar el diseño de la solución, diagramas generales, de clases, arquitecturas y todo lo necesario para presentar y argumentar el proyecto ante la universidad.

Apéndice B

Documento de Arquitectura de Software



SYNERGY-GB

Documento de Arquitectura de Software

Notificaciones Push Enriquecidas

Table of Contents

Introducción	3
Representación Arquitectónica	3
Vista de Escenarios	4
Vista Lógica	6
Vista de Procesos	7
Vista de Desarrollo	7
Vista de Despliegue o Implantación	8
Tamaño y Desempeño	8
Calidad	9
Referencias	10

Introducción

Este documento provee una visión general de la arquitectura del sistema de notificaciones push enriquecidas móvil multiplataforma, usando distintas vistas arquitectónicas para representar distintos aspectos del sistema, brindando al lector una visión global y comprensible del diseño general del mismo.

El modelado de las vistas permitirá crear un modelo del sistema mucho más completo y constituye una base sólida para su desarrollo. En este documento se incluyen la vista de escenarios representada en el Diagrama de Casos de Uso, la vista lógica con el Diagrama de Clases, la vista de desarrollo ilustrada por el Diagrama de componentes y la vista de despliegue representada por el Diagrama de Despliegue.

Representación Arquitectónica

La Arquitectura de Software básicamente es considerada como una vista del sistema donde se incluyen sus componentes principales y la interacción y coordinación entre ellos, necesarios para cumplir con los requerimientos específicos del sistema.

En este documento se utiliza el Modelo 4+1 Vistas de Kruchten, el cual organiza la descripción de la arquitectura de software en cinco vistas (Kruchten, 1995), donde se exponen las decisiones de diseño y una quinta vista para ilustrar y validar dichas decisiones. Estas vistas se componen por la de escenarios, lógica, procesos, desarrollo e implantación. Es importante destacar que éstas son complementadas con elementos de UML que proporcionen más información a nivel gráfico. Las vistas son las siguientes:

- La vista de escenarios, también llamada casos de uso, contiene los casos de uso críticos del sistema, representados mediante el diagrama de casos de uso. Es de gran importancia ya que a través de esta vista se puede entender mejor la funcionalidad del sistema y funciona como un indicador que ayuda al diseñador a descubrir los elementos de la arquitectura durante su diseño (Kruchten, 1995).
- La vista lógica comprende las abstracciones fundamentales del sistema a partir del dominio del problema, trata principalmente los requerimientos funcionales del sistema y forma la estructura del mismo, incluye el diagrama de clases (Kruchten, 1995).
- La vista de procesos describe el diseño de concurrencia y aspectos de sincronización y muestra algunos de los requisitos no funcionales, como son desempeño, rendimiento, seguridad, fiabilidad, entre otros (Kruchten, 1995).
- La vista de desarrollo o implantación expone las características y requisitos que debe cumplir el sistema a nivel técnico (Kruchten, 1995).

- La vista de implementación engloba la distribución de paquetes de la aplicación y de manera general la distribución y funcionamiento de las distintas capas del sistema. Describe el mapeo del software en el hardware (Kruchten, 1995).

Vista de Escenarios

Para ilustrar la vista de escenarios, se incluye el diagrama de casos de uso definitivo.



Figura 1. Diagrama de Casos de Uso tentativo. Elaboración Propia.

A continuación se presenta, la Tabla 1, donde se describen los casos de uso del sistema.

Número	Nombre del Caso de Uso	Descripción
CA_1	Iniciar Sesión	Permite a cualquier persona registrada, entrar en sesión personalizada para realizar las operaciones deseadas.
CA_2	Cerrar Sesión	Permite al usuario que posee una sesión activa, cerrarla una vez terminadas las operaciones que estuviera realizando o en cualquier momento que desee, posterior a haber iniciado sesión.
CA_3	Registrarse	Permite al usuario registrarse en el sistema.
CA_4	Recibir Notificación Push	Permite al usuario recibir notificaciones Push de Synergy Push Services.
CA_5	Gestionar Mensajes	Permite al usuario decidir qué hacer con las notificaciones recibidas.
CA_6	Recibir Mensajes	Permite al usuario recibir todas las notificaciones que se le manden a través de Synergy Push. Se hace de manera automática.
CA_7	Aprobar Mensajes	Permite al usuario aprobar notificaciones que requieran aprobación del usuario como alertas de seguridad y notificación de operaciones.
CA_8	Rechazar Mensajes	Permite al usuario rechazar notificaciones que requieran aprobación del usuario como alertas de seguridad y notificación de operaciones.
CA_9	Compartir Mensajes	Permite al usuario compartir cualquier notificación.
CA_10	Eliminar Mensajes	Permite al usuario eliminar cualquier notificación que tenga en su buzón.

Tabla 1. Descripción de Casos de Uso. Elaboración Propia.

Vista Lógica

A continuación se presenta el Diagrama de Clases.

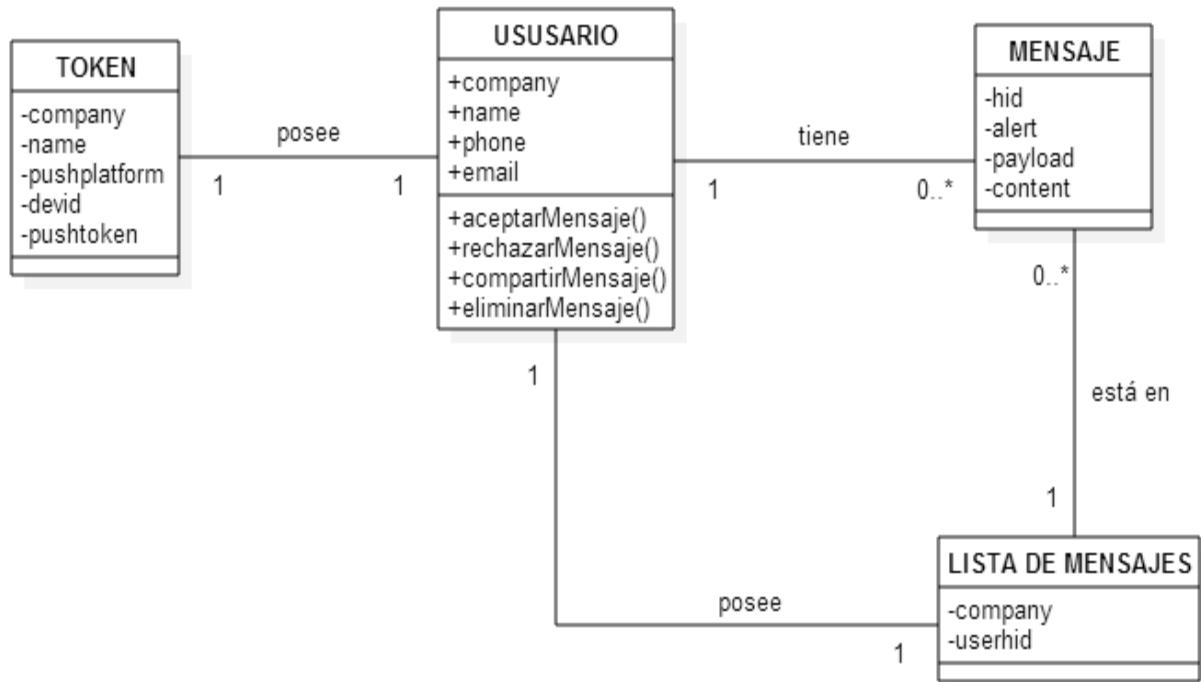


Figura 2. Diagrama de Clases tentativo. Elaboraci  n Propia.

Vista de Procesos

La Vista de Procesos no fue desarrollada, dado que la aplicación no resuelve aspectos de concurrencia de usuarios y de datos. Los mismos son resueltos por la arquitectura Synergy Push dentro de los módulos que ofrece.

Vista de Desarrollo

La vista de desarrollo se realizó a través de un diagrama de componentes, donde se muestra la relación entre los mismos. Cabe destacar que solo los componentes más importantes están presentes en el diagrama de modo que se facilite su comprensión.

A continuación, en la figura 4, se muestra el diagrama de componentes (resumido) del sistema.

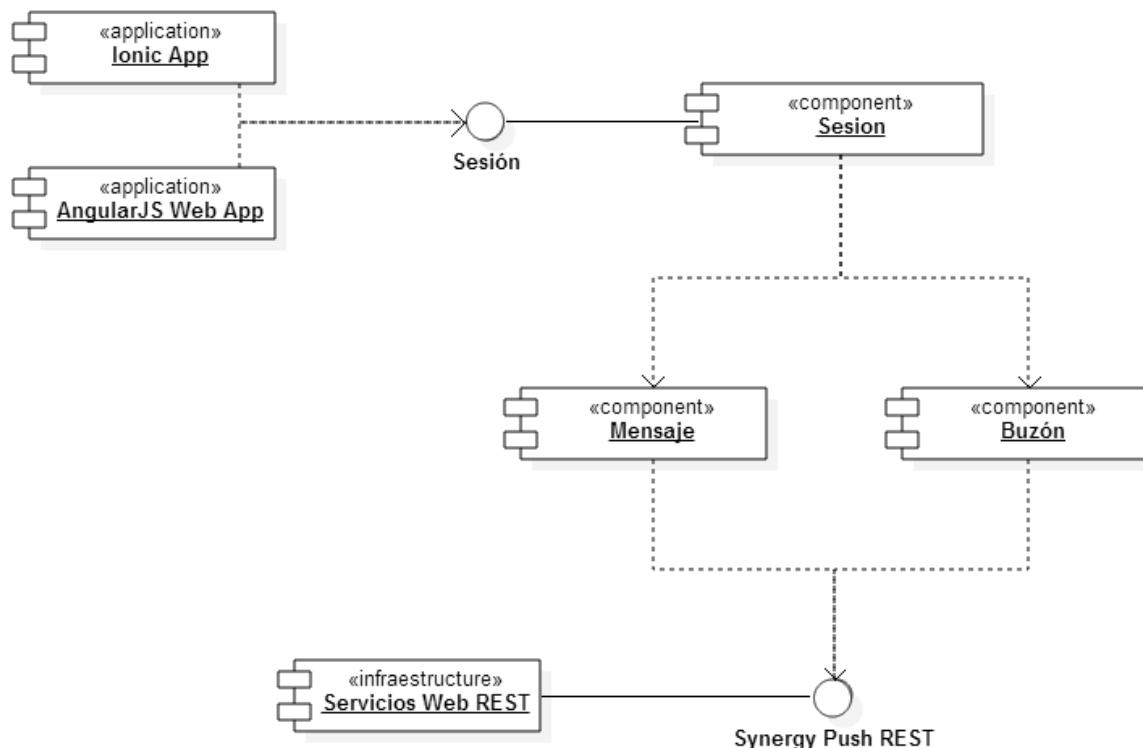


Figura 3. Diagrama de Componentes. Elaboración Propia.

Vista de Despliegue o Implantación

La vista de implantación se ve reflejada en la figura 5, mostrada a continuación.

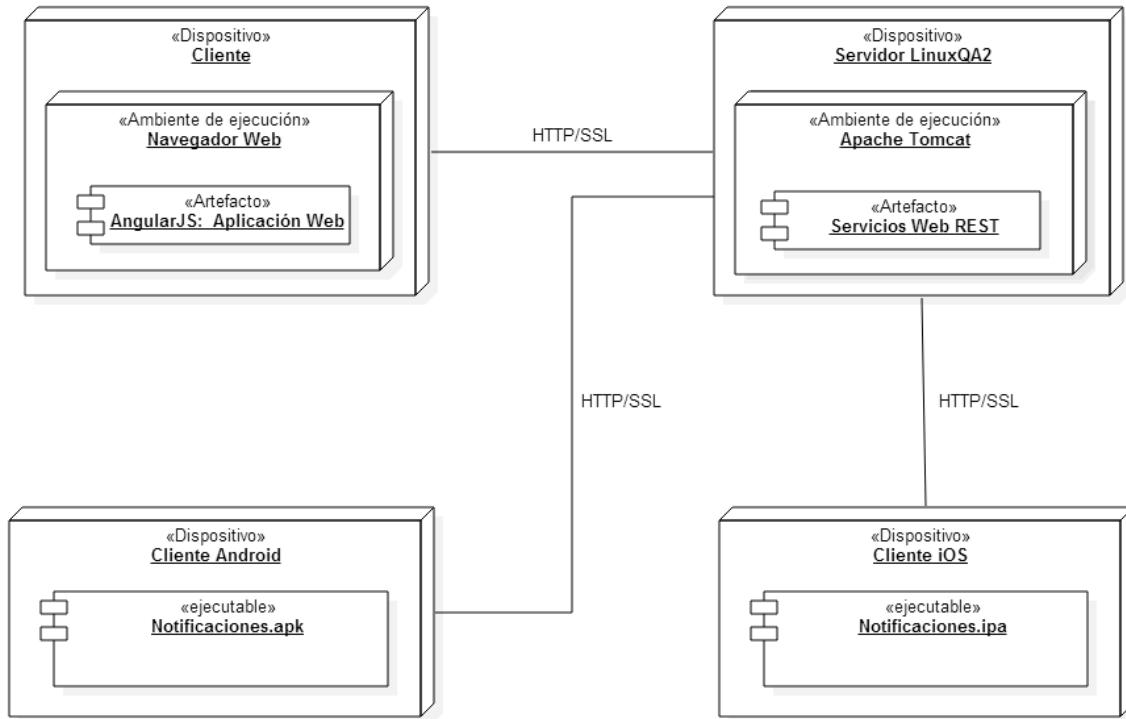


Figura 4. Diagrama de Despliegue. Elaboración Propia.

Es importante aclarar que la comunicación se realiza a través de HTTP debido a que es un prototipo, posteriormente cada una de estas se comunicará a través de HTTPS/SSL. La información que se comparte entre los nodos se encuentra en formato JSON.

Tamaño y Desempeño

- La aplicación móvil multiplataforma corre en dispositivos iOS y Android.
- Solo se permite un dispositivo por usuario y un usuario por dispositivo.
- El recurso necesario para la utilización del sistema web es un explorador o navegador de internet.

Calidad

- La arquitectura MVC (Modelo-Vista-Controlador) separa la lógica de negocio (el modelo) y la interfaz (la vista) por lo que se consigue un mantenimiento más sencillo del sistema, por ejemplo, si el sistema debe ejecutarse tanto en un navegador estándar como en un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva (y funcionará en general para cualquier dispositivo con un navegador web), sin necesidad de modificar el controlador y el modelo (Para el caso web). Incluso para el caso de la aplicación móvil aplica lo mismo, gracias a que AngularJS permite trabajar con el patrón MVC del lado del cliente en una aplicación diseñada con Ionic (En este caso con crear una sola vista, se podrá ejecutar tanto en el sistema operativo iOS como en el sistema operativo Android).
- El sistema es desarrollado como SaaS (Software as a Service), por lo que la aplicación podrá funcionar en cualquier plataforma, garantizando una alta portabilidad.
- El desarrollo de las aplicaciones está basado en módulos lo que permite ser escalable ya que permite la incorporación nuevas características para ampliar sus capacidades de manera fácil. El sistema está diseñado con esta característica para que se puedan agregar nuevos componentes sin tener que afectar lo ya existente ya que se siguieron las mejores prácticas de diseño.
- La aplicación móvil podrá desenvolverse (por ser multiplataforma) tanto en iOS como en Android y la aplicación web sólo requerirá de un navegador web.

Referencias

Kruchten, P. (Noviembre de 1995). Architectural Blueprints - The "4+1" View Model of Software Architecture. *IEEE Software Magazine*, 42-50. Recuperado el 23 de Septiembre de 2013, de <http://www.cs.ubc.ca/~gregor/teaching/papers/4+1viewarchitecture.pdf>

David Prieto. Abril de 2015. Investigación de la Arquitectura Notificaciones Push Enriquecidas.