



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

**DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA PLATAFORMA
ANDROID PARA SISTEMA DE COMPRAS COLABORATIVAS**

Por

Genessis Karolina Sánchez Centeno

INFORME DE PASANTÍAS

Presentado ante la ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero en Computación

Sartenejas, Febrero de 2017



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

DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA PLATAFORMA ANDROID PARA SISTEMA DE COMPRAS COLABORATIVAS

Por

Genessis Karolina Sánchez Centeno

Realizado con la asesoría de:

Prof. Ascánder Suárez

Ing. Carlos Inguanzo

INFORME DE PASANTÍAS

Presentado ante la ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero en Computación

Sartenejas, Febrero de 2017



UNIVERSIDAD SIMÓN BOLÍVAR
VICERRECTORADO ACADÉMICO
DECANATO DE EXTENSIÓN
Coordinación de Cooperación Técnica

534e42929f

ACTA DE EVALUACIÓN

Período: Julio-Diciembre 2016 Fecha: 03/03/2017 Tipo Pasantía: Pasantía larga EP-3420

Título: Desarrollo de aplicación móvil para la plataforma Android para sistema de compras colaborativas

Nombres y Apellidos: Genessis Karolina Sanchez Centeno CI.: 21618186

Carnet: 11-10935 Carrera: Ingeniería de la Computación 0800

APROBADO: ☒ REPROBADO: ☐

| JURADO EXAMINADOR | |
|--|------------------------|
| TUTOR ACADÉMICO Nombre: Ascander Suarez | Firma: CI: 4435018 |
| TUTOR INDUSTRIAL Nombre: Carlos Miguel Inguanzo | Firma: CI: 17981253 |
| JURADO Nombre: David Coronado | Firma: CI: 11159277 |
| Art. 17. Reglamento C.4 Cursos en Cooperación: Cuando la evaluación global del tutor industrial de la pasantía sea igual o superior a 4,5 puntos en la escala de 5, el Jurado Examinador podrá reconocer el trabajo aprobado como SOBRESALIENTE , justificándolo en forma razonada en el Acta: | |
| Justificación: <i>Los resultados tienen un impacto positivo y significativo en las operaciones de la empresa</i> | |
| Firmas Jurado Examinador: | |

Sello del Departamento

Sello de la Coordinación de Carrera

Sello del Departamento

Validación de CCT
Firma y Sello

Nota: Este documento no puede llevar tachaduras ni enmiendas y todas las firmas del Jurado Examinador deben estar acompañadas de sus correspondientes sellos en húmedo. En caso de inasistencia del tutor industrial, se debe colocar **NO-PRESENTE** en el espacio reservado para su firma. La evaluación por parte del tutor industrial estará documentada mediante los respectivos procedimientos que establezca el Decanato de Extensión. El Jurado Examinador firmará nuevamente en la zona de justificación al otorgar la mención SOBRESALIENTE.

Puerta de Comunidad a Comunidad

Sartenejas, Baruta, Edif. Comunicaciones Telf: (0212) 906.3157 al 64 Apartado postal 89000, Zip Code 1080-A. www.usb.ve



UNIVERSIDAD SIMÓN BOLÍVAR

VICERRECTORADO ACADÉMICO
DECANATO DE ESTUDIOS PROFESIONALES
Coordinación de Ingeniería de la Computación

ACTA DE EVALUACION DEL INFORME DE PASANTIA LARGA

Código de la asignatura: EP3420

Fecha: 21-02-2017

Nombre del estudiante: Génessis Karolina Sánchez Centeno

Carnet: 11-10935

Título del proyecto: DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA PLATAFORMA ANDROID PARA SISTEMA DE COMPRAS COLABORATIVAS

Tutor: Prof. Ascánder Suárez

Jurados: Profesor David Coronado

Nosotros, miembros del Jurado designado por la Coordinación Docente de INGENIERÍA DE LA COMPUTACIÓN, en la fecha indicada para la evaluación del Proyecto de Pasantía Larga en cuestión, hemos evaluado el contenido de la misma y hacemos constar nuestra decisión de **aceptar el documento para su presentación en acto público**. Se convoca al estudiante, Genessis Karolina Sánchez Centeno, a la presentación pública del Proyecto de Pasantía Larga "DESARROLLO DE UNA APLICACIÓN MÓVIL PARA LA PLATAFORMA ANDROID PARA SISTEMA DE COMPRAS COLABORATIVAS" en el edificio "Matemáticas y Sistemas" sala "Sala de Reuniones del Departamento de Computación".

Fecha: 03/03 /2017

Hora: 10:00 am

Lugar: Sede de Sartenejas

Prof. David Coronado
Jurado (Presidente)
C.I. 11.159.277



Prof. Ascánder Suárez
Tutor Académico
C.I. 4.435.018

RESUMEN

En este documento se exponen y detallan todas y cada una de las actividades realizadas a través del desarrollo de una aplicación móvil del sistema de compras colaborativas Club Mercado para dispositivos de la plataforma Android.

Club Mercado permite a una comunidad de compradores, conformada por una red de grupos independientes, la compra de productos directamente a fabricantes, productores e importadores mediante tecnologías de punta y procesos de logística que garantizan, de forma cómoda y a menor costo, que los mismos sean más frescos y de mayor calidad.

Para el desarrollo de esta aplicación móvil nativa para dispositivos Android se empleó la metodología de desarrollo ágil e iterativo por fases SCRUM. Asimismo, se utilizó como principal herramienta el SDK que ofrece el sistema operativo Android, el cual trabaja bajo el patrón de arquitectura de software Modelo Vista Controlador, basándose en el popular lenguaje de programación Java.

Este proyecto de pasantía derivó finalmente en el desarrollo de la versión 1.0 de la aplicación móvil Club Mercado para dispositivos de la plataforma Android.

Palabras clave: aplicación, móvil, Android, desarrollo, ágil.

DEDICATORIA

*A mis padres y hermana,
por el apoyo incondicional que me dan.
Ustedes hicieron esto posible, eternamente gracias.*

AGRADECIMIENTOS

A mis padres, hermana, primos y prima por siempre apoyarme y creer en mí, incluso más de lo que yo lo hago en ocasiones. De más está decir que los amo.

A mis profesores, quienes, a pesar de las adversidades, siguen en la búsqueda de la excelencia gracias a su admirable vocación. En especial, reconocer la labor que desempeñan o desempeñaron Alejandro Teruel, Marlene Goncalves, Ricardo Monascal, Ernesto Hernández-Novich y María Esther Vidal. Gracias por hacerme una mejor profesional.

A mi tutor académico y profesor, Ascander Suárez, por aceptar mi proyecto y por sus enseñanzas. A mi tutor industrial, Carlos Inganzo, por confiar plenamente en mis capacidades y por ayudarme en todo lo que pudiera. A todas las personas que hacen o hicieron vida en Ávila Holding Group durante mi período de pasantías por haber hecho el ambiente de trabajo inmejorable y, en particular, a Domingo Arteaga, mi amigo y futuro colega.

A todos los amigos que hice en la universidad; entre ellos, Daniel Astorga, Delinés de Sousa y Luisana Bustamante: gracias por ser mis primeros amigos universitarios cuando era nueva. Asimismo, a todos los miembros del Laboratorio Docente de Aulas Computarizadas por ser mi familia y apoyarme en todo lo que necesité incondicionalmente, en especial, a Gustavo Gutiérrez, por ser mi amigo desde el bachillerato y pareja en los primeros proyectos de la carrera (así como por las incontables colas). ¡Gracias!

A mis muchachos de siempre, por siempre estar ahí desde primaria. Sin duda, ustedes fueron los que más animaron mi vida universitaria. Gracias por tantas risas, viajes y anécdotas memorables. Gracias por su amistad verdadera.

Por último, a mi amada Universidad Simón Bolívar, por brindarme (a pesar de las adversidades) todas las herramientas para formarme como profesional. Gracias por tanto conocimiento, emociones y momentos inolvidables.

ÍNDICE GENERAL

| | |
|--|-----|
| ÍNDICE GENERAL..... | VII |
| ÍNDICE DE FIGURAS | XI |
| LISTA DE ABREVIATURAS..... | XII |
| INTRODUCCIÓN..... | 1 |
| • ANTECEDENTES | 1 |
| • JUSTIFICACIÓN E IMPORTANCIA | 1 |
| • PLANTEAMIENTO DEL PROBLEMA | 2 |
| • OBJETIVOS..... | 2 |
| • <i>Objetivo general</i> | 2 |
| • <i>Objetivos específicos</i> | 3 |
| CAPÍTULO 1..... | 4 |
| 1.1 IDBC GROUP..... | 4 |
| 1.1.1 <i>Misión</i> | 4 |
| 1.1.2 <i>Visión</i> | 4 |
| 1.1.3 <i>Estructura organizacional</i> | 5 |
| 1.1.4 <i>Cargo ocupado por el pasante</i> | 5 |
| CAPÍTULO 2..... | 6 |
| 2.1 API..... | 6 |
| 2.2 ARQUITECTURA CLIENTE – SERVIDOR | 6 |
| 2.3 MODELO VISTA CONTROLADOR | 6 |
| 2.4 <i>REPRESENTATIONAL STATE TRANSFER</i> | 7 |
| 2.5 ESTRATEGIA DE DESARROLLO DE APLICACIONES MÓVILES EN ENTORNOS NATIVOS | 8 |
| CAPÍTULO 3..... | 9 |
| 3.1 <i>PYTHON</i> | 9 |
| 3.2 <i>DJANGO</i> | 9 |
| 3.3 <i>DJANGO REST FRAMEWORK</i> | 10 |
| 3.4 <i>SQLITE</i> | 10 |
| 3.5 XML | 10 |

| | | |
|-------------------|---|-----------|
| 3.6 | JAVA..... | 11 |
| 3.7 | ANDROID SOFTWARE DEVELOPMENT KIT | 11 |
| 3.8 | HERRAMIENTAS PARA EL CONTROL DE VERSIONES Y PLANIFICACIÓN..... | 11 |
| 3.8.1 | Git..... | 11 |
| 3.8.2 | GitHub..... | 11 |
| 3.8.3 | Taiga | 12 |
| CAPÍTULO 4 | | 13 |
| 4.1 | ROLES..... | 13 |
| 4.1.1 | Dueño del Producto o Product owner..... | 13 |
| 4.1.2 | Equipo o Team | 14 |
| 4.1.3 | Facilitador o Scrum master..... | 14 |
| 4.1.4 | Stakeholders | 14 |
| 4.2 | EVENTOS | 14 |
| 4.2.1 | Sprint..... | 15 |
| 4.2.2 | Sprint Planning | 15 |
| 4.2.3 | Daily Scrum..... | 15 |
| 4.2.4 | Sprint Review..... | 15 |
| 4.2.5 | Sprint Retrospective | 16 |
| 4.3 | ARTEFACTOS | 16 |
| 4.3.1 | Product Backlog..... | 16 |
| 4.3.2 | Sprint Backlog..... | 16 |
| CAPÍTULO 5 | | 17 |
| 5.1 | PRIMER SPRINT | 17 |
| 5.1.1 | Objetivos | 17 |
| 5.1.2 | Actividades | 17 |
| 5.1.2.1 | Familiarización con los lenguajes de programación y herramientas a utilizar | 17 |
| 5.1.2.2 | Asistencia a charla sobre la metodología y procesos de IDBC Group | 17 |
| 5.2 | SEGUNDO SPRINT..... | 18 |
| 5.2.1 | Objetivos | 18 |
| 5.2.2 | Actividades | 18 |
| 5.2.2.1 | Familiarización con el modelo de negocio de Club Mercado | 18 |
| 5.2.2.2 | Estudio de la plataforma web desarrollada para Club Mercado | 18 |
| 5.3 | TERCER SPRINT..... | 18 |
| 5.3.1 | Objetivos | 18 |
| 5.3.2 | Actividades | 19 |
| 5.3.2.1 | Familiarización con la creación de APIs en <i>Python</i> (comparación entre <i>API Frameworks</i>) | 19 |
| 5.3.2.2 | Diseño de los servicios web a ser un API para el sistema web Club Mercado | 19 |
| 5.3.2.3 | Creación del API para comunicación cliente-servidor de Club Mercado | 19 |

| | | |
|----------|--|----|
| 5.4 | CUARTO <i>SPRINT</i> | 19 |
| 5.4.1 | Objetivos | 19 |
| 5.4.2 | Actividades | 20 |
| 5.4.2.1 | Implementación del módulo de permisos sobre el API..... | 20 |
| 5.4.2.2 | Implementación de las funcionalidades para la muestra, creación y actualización de los datos correspondientes a los modelos de interés de la base de datos a través del API..... | 20 |
| 5.5 | QUINTO <i>SPRINT</i> | 20 |
| 5.5.1 | Objetivos | 20 |
| 5.5.2 | Actividades | 20 |
| 5.5.2.1 | Análisis, diseño e implementación la interfaz el tour de inicio que desemboque en el registro o inicio de sesión de usuarios | 20 |
| 5.6 | SEXTO <i>SPRINT</i> | 21 |
| 5.6.1 | Objetivos | 21 |
| 5.6.2 | Actividades | 21 |
| 5.6.2.1 | Análisis, diseño e implementación la interfaz de inicio y cierre de sesión..... | 21 |
| 5.6.2.2 | Análisis, diseño e implementación de la interfaz de recuperación de contraseñas | 22 |
| 5.7 | SÉPTIMO <i>SPRINT</i> | 22 |
| 5.7.1 | Objetivos | 22 |
| 5.7.2 | Actividades | 22 |
| 5.7.2.1 | Análisis, diseño e implementación de una base de datos local y motor de sincronización de datos para los modelos correspondientes al módulo de manejo de usuarios | 22 |
| 5.7.2.2 | Análisis, diseño e implementación de la interfaz de registro de usuario | 23 |
| 5.8 | OCTAVO <i>SPRINT</i> | 23 |
| 5.8.1 | Objetivos | 23 |
| 5.8.2 | Actividades | 24 |
| 5.8.2.1 | Análisis y diseño de la interfaz de un perfil de usuario que comprenda la visualización o edición de los datos personales, compras realizadas y estado de cuenta del usuario | 24 |
| 5.8.2.2 | Implementación de la interfaz de visualización y modificación de datos personales en el perfil de usuario | 24 |
| 5.9 | NOVENO <i>SPRINT</i> | 25 |
| 5.9.1 | Objetivos | 25 |
| 5.9.2 | Actividades | 25 |
| 5.9.2.1 | Diseño e implementación de la interfaz de visualización de las estadísticas de compra realizadas en el perfil de usuario | 25 |
| 5.10 | DÉCIMO <i>SPRINT</i> | 26 |
| 5.10.1 | Objetivos | 26 |
| 5.10.2 | Actividades | 26 |
| 5.10.2.1 | Diseño e implementación de la interfaz de visualización de saldo y movimientos realizados en el perfil de usuario | 26 |
| 5.11 | DÉCIMO PRIMER <i>SPRINT</i> | 26 |

| | | |
|--|---|-----------|
| 5.11.1 | Objetivos | 26 |
| 5.11.2 | Actividades | 27 |
| 5.11.2.1 | Análisis, diseño e implementación de la interfaz para la visualización de categorías que desemboque en el pasillo de sus productos asociados | 27 |
| 5.11.2.2 | Análisis, diseño e implementación de la interfaz para la visualización de productos dentro de categorías | 27 |
| 5.12 | DÉCIMO SEGUNDO SPRINT | 28 |
| 5.12.1 | Objetivos | 28 |
| 5.12.2 | Actividades | 28 |
| 5.12.2.1 | Implementación de la funcionalidad de visualizar y agregar al carrito de compras | 28 |
| 5.13 | DÉCIMO TERCER SPRINT | 29 |
| 5.13.1 | Objetivos | 29 |
| 5.13.2 | Actividades | 29 |
| 5.13.2.1 | Implementación de la funcionalidad de búsqueda de productos | 29 |
| 5.13.2.2 | Implementación del cierre de compra | 29 |
| 5.14 | DÉCIMO CUARTO SPRINT | 30 |
| 5.14.1 | Objetivos | 30 |
| 5.14.2 | Actividades | 30 |
| 5.14.2.1 | Análisis, diseño de la interfaz para el abono de saldo y visualización de abonos realizados | 30 |
| 5.14.2.2 | Implementación de la interfaz de visualización de abonos realizados | 30 |
| 5.15 | DÉCIMO QUINTO SPRINT | 31 |
| 5.15.1 | Objetivos | 31 |
| 5.15.2 | Actividades | 31 |
| 5.15.2.1 | Implementación de la interfaz del formulario de abono de saldo | 31 |
| 5.16 | DÉCIMO SEXTO SPRINT | 32 |
| 5.16.1 | Objetivos | 32 |
| 5.16.2 | Actividades | 32 |
| 5.16.2.1 | Realización del despliegue de la aplicación móvil en la plataforma <i>Play Store</i> de Google | 32 |
| 5.17 | DIFICULTADES GENERALES ENCONTRADAS | 32 |
| CONCLUSIONES Y RECOMENDACIONES | | 34 |
| REFERENCIAS BIBLIOGRÁFICAS | | 36 |
| APÉNDICE A. DIAGRAMA DE CASOS DE USO | | 38 |
| APÉNDICE B. DIAGRAMA DE CLASES | | 39 |
| APÉNDICE C. SCREENSHOTS DEL SISTEMA CONTENIDO | | 40 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1.1: Estructura organizacional de IDBC Group | 5 |
| Figura A.1: Diagrama de Casos de Usos de la aplicación móvil Club Mercado para Android..... | 38 |
| Figura B.1: Diagrama de Clases de la aplicación móvil Club Mercado para Android..... | 39 |
| Figura C.1: <i>Screenshot</i> de la vista de tour de inicio (1)..... | 40 |
| Figura C.2: <i>Screenshot</i> de la vista de tour de inicio (2)..... | 40 |
| Figura C.3: <i>Screenshot</i> de la vista de tour de inicio (3)..... | 41 |
| Figura C.4: <i>Screenshot</i> de la vista de tour de inicio (4)..... | 41 |
| Figura C.5: <i>Screenshot</i> de la vista de inicio de sesión..... | 42 |
| Figura C.6: <i>Screenshot</i> de la vista de reinicio de contraseña..... | 42 |
| Figura C.7: <i>Screenshot</i> de la vista de registro de usuario..... | 43 |
| Figura C.8: <i>Screenshot</i> de la vista de gestión de datos personales en el perfil de usuario | 43 |
| Figura C.9: <i>Screenshot</i> de la vista de visualización de estadísticas de compra en el perfil de usuario..... | 44 |
| Figura C.10: <i>Screenshot</i> de la vista de visualización de saldo y movimientos realizados en el perfil de usuario | 44 |
| Figura C.11: <i>Screenshot</i> de la vista de visualización de categorías..... | 45 |
| Figura C.12: <i>Screenshot</i> de la vista de pasillo de compra de productos por categoría..... | 45 |
| Figura C.13: <i>Screenshot</i> de la vista de visualización de abonos realizados | 46 |
| Figura C.14: <i>Screenshot</i> de la vista del formulario de abono de saldo..... | 46 |

LISTA DE ABREVIATURAS

| | |
|------|---|
| API | <i>Application Programming Interface</i> |
| APK | <i>Android Application Package</i> |
| HTTP | <i>Hypertext Transfer Protocol</i> |
| IDE | <i>Integrated Development Environment</i> |
| JSON | <i>JavaScript Object Notation</i> |
| MVC | Modelo Vista Controlador |
| REST | <i>Representational State Transfer</i> |
| SDK | <i>Software Development Kit</i> |
| URI | <i>Uniform Resource Identifier</i> |
| XML | <i>eXtensible Markup Language</i> |

INTRODUCCIÓN

El presente documento contextualiza y describe el proceso de desarrollo de la aplicación móvil asociada al sistema web Club Mercado, así como el proceso de desarrollo de su API (*Application Programming Interface*) como proyecto de pasantías realizado por su autor.

Está organizado por capítulos de la siguiente manera: la Introducción al proyecto de pasantía. En el Capítulo 1 se describe el entorno laboral para familiarizar al lector con la estructura empresarial de IDBC Group. En el Capítulo 2 se describe el marco teórico necesario para la comprensión del problema y abordar su solución. En el Capítulo 3 se presentan y describen las herramientas utilizadas para el desarrollo del proyecto. En el Capítulo 4 se especifica la metodología utilizada (SCRUM). En el Capítulo 5 se detalla el proceso de desarrollo en cada una de las iteraciones realizadas y al mismo tiempo, se describen tanto las actividades como los artefactos utilizados, así como los problemas encontrados en cada una de ellas. Por último, se exponen las Conclusiones y recomendaciones.

- **Antecedentes**

Club Mercado es una aplicación web de compras colaborativas que permite a sus usuarios, organizados en grupos independientes, utilizar tecnología y procesos de logística para comprar directo a fabricantes, productores e importadores (de forma cómoda, práctica y segura) productos frescos y de buena calidad a bajo costo.

- **Justificación e importancia**

Actualmente, el sistema de Club Mercado sólo es accesible para sus usuarios a través de un explorador web, lo que resulta poco práctico para aquellos que desean realizar pedidos (o participar), abonar a su saldo o consultar y actualizar su información desde cualquier lugar, sin restricciones; por lo que atender dichos requerimientos resulta de gran importancia para el negocio.

- **Planteamiento del problema**

Se identificó la necesidad de desarrollar la versión móvil nativa para la plataforma Android del sistema web existente, con el fin de poder brindarle a los usuarios que utilizan dispositivos móviles de dicha plataforma, algunas de las principales funcionalidades de Club Mercado; tales como el registro de usuario, inicio de sesión, visualización del saldo y estado de cuenta del usuario, visualización de compras disponibles para participar, así como el historial de participaciones en dichas compras.

Para lograr esto, es necesario también el desarrollo de un API sencillo para el sistema web Club Mercado que permita la conexión con otras aplicaciones, incluyendo la versión móvil a desarrollar. Asimismo, se requiere la creación de una base de datos local perteneciente a la aplicación móvil que se sincronice con los datos correspondientes a cada funcionalidad a través del propio API, con miras a permitir ciertas acciones en ausencia de conexión a internet.

- **Objetivos**

A continuación, se exponen los objetivos generales y específicos que se busca alcanzar en este desarrollo con la finalidad de contextualizar al lector respecto al informe de este proyecto de pasantía.

- **Objetivo general**

El objetivo general de este proyecto es desarrollar una aplicación móvil para dispositivos de la plataforma Android que brinde una interfaz sencilla y que contenga las funcionalidades principales de Club Mercado. Adicionalmente, se busca la creación de un API que conecte con el sistema de Club Mercado y que permita la conexión con otras aplicaciones para solicitar o modificar contenido.

- **Objetivos específicos**

- Crear un API sencillo que permita la conexión con la plataforma web Club Mercado.
- Desarrollar un tour de inicio que muestre las principales funcionalidades disponibles en la aplicación móvil.
- Desarrollar un módulo para el acceso y autenticación de la aplicación que comprenda el inicio y cierre de sesión, la recuperación de contraseñas y la validación de los roles y privilegios de cada usuario.
- Desarrollar un módulo que permita la visualización y edición del perfil de usuario.
- Desarrollar un módulo que permita agregar productos a un carrito temporal de compra y cerrar la compra (también llamada participación) actual.
- Desarrollar un módulo que permita el abono al saldo de un usuario mediante la carga de los datos de una transferencia realizada.
- Desplegar la aplicación móvil en la plataforma Google Play

CAPÍTULO 1

ENTORNO EMPRESARIAL

En este capítulo se describe el entorno empresarial en el cual tuvo lugar el desarrollo del proyecto de pasantía. Se presentan los objetivos, misión y visión de IDBC Group.

1.1 IDBC Group

IDBC Group, es una empresa de desarrollo ágil de software con más de nueve años de experiencia, orientada al desarrollo de servicios y soluciones tecnológicas ágiles, eficientes y de alto impacto.

La empresa obtiene resultados increíbles, todos impulsados por su definición de equipos integrados de producto que siguen una visión compartida. Fomenta equipos de trabajo permanentes que se mantienen juntos y son los correctos para cada proyecto.

IDBC Group trabaja directamente con distintos segmentos cuyas necesidades, aunque parecidas, varían, por naturaleza, desde su raíz: particulares, corporativos y emprendedores (Inganzo, 2016).

1.1.1 Misión

“Crear impacto positivo, sostenible y social a través de la tecnología, con una actitud humilde y divertida” (Inganzo, 2016).

1.1.2 Visión

“Posicionarnos como una empresa de desarrollo tecnológico que abarque los diversos mercados existentes, logrando así diversidad de soluciones tecnológicas de excelencia y calidad en América Latina.” (Inganzo, 2016)

1.1.3 Estructura organizacional

En la Figura 1.1 se muestra la estructura organizacional de IDBC Group:

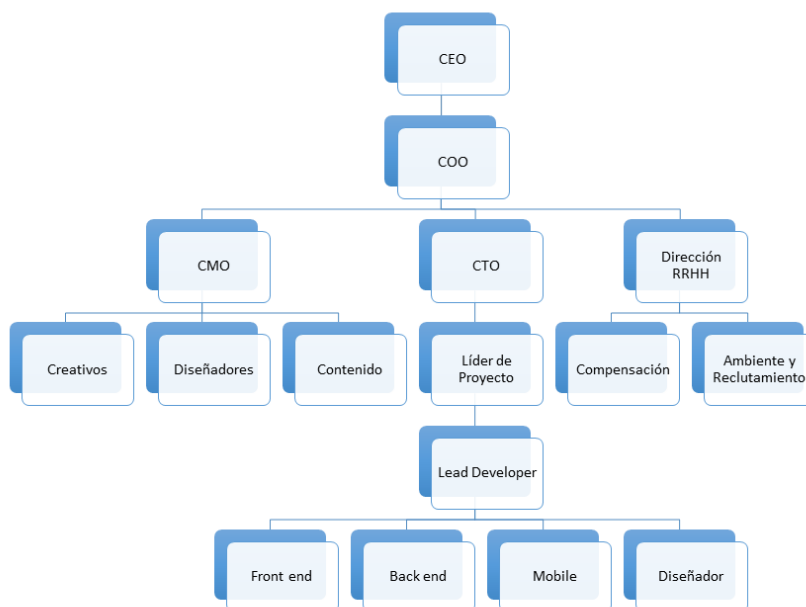


Figura 1.1: Estructura organizacional de IDBC Group

1.1.4 Cargo ocupado por el pasante

Durante la realización del proyecto de pasantía, el pasante ocupó el puesto de desarrollador de aplicaciones móviles, dentro del módulo *Mobile*, cuyo nivel en la jerarquía de la empresa se muestra en la Figura 1.1.

CAPÍTULO 2

MARCO TEÓRICO

En el presente capítulo es presentado y descrito el conjunto de conceptos, términos y pilares teóricos relevantes sobre el cuales se basó este proyecto.

2.1 API

La interfaz de programación de aplicaciones (*Application Programming Interface* o API, por su abreviatura en inglés) es un conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general evitándose así el trabajo de programar todo desde el principio (Merino, 2014).

2.2 Arquitectura Cliente – Servidor

La arquitectura cliente - servidor consiste básicamente, como su nombre indica, en un *cliente*, es decir: un programa ejecutable que participa activamente en el establecimiento de conexiones y peticiones de recursos a otro programa, el *servidor*, encargado de atender dichas solicitudes y proporcionar los recursos solicitados.

Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora, es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras, la interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes (EcuRed, Arquitectura Cliente-Servidor, (s.f))

2.3 Modelo Vista Controlador

Modelo-Vista-Controlador (MVC) es un patrón de arquitectura de software, descrito por primera vez en el año 1979, por Trygve Reenskaug y James O. Coplien (Trygve & Coplien, 2009). Este patrón propone la separación de los datos y la lógica del negocio de la interfaz de usuario y los módulos encargados de gestionar los eventos. Para ello, el patrón se basa en tres capas descritas de la siguiente forma:

- **Modelo:** gestiona la representación de los datos del sistema y las operaciones asociadas a ellos, tales como consultas o actualizaciones.
- **Vista:** encargada de presentar visualmente al usuario la información en un formato adecuado para interactuar con ella.
- **Controlador:** definido como la interfaz o intermediario entre la vista y el modelo. Su principal labor es la de responder a eventos e invocar peticiones al modelo cuando se hace alguna solicitud. Asimismo, interactúa con la vista asociada a él para modificar la presentación de un modelo.

La idea detrás de mantener estas tres capas por separado es la de aumentar la independencia de las mismas, disminuyendo así el acoplamiento del código.

El proyecto de pasantía es desarrollado basándose en este patrón, pues proporciona facilidades a la hora de reciclar y de mantener código.

2.4 *Representational State Transfer*

El estilo de arquitectura REST (*Representational State Transfer*) está basado en la transferencia de recursos desde un servidor a un cliente. Un servicio web RESTful es identificado por su URI (*Uniform Resource Identifier*) y se comunica haciendo uso del protocolo HTTP (*Hypertext Transfer Protocol*), respondiendo a los métodos *GET*, *POST*, *PUT* y *DELETE*, devolviendo una representación de los recursos.

Se decidió la utilización del estilo REST debido a que es más ligero y proporciona formas más simples y flexibles de hacer las cosas en comparación con otros estilos tales como SOAP (*Simple Object Access Protocol*). Asimismo, haciendo uso del formato JSON (*JavaScript Object Notation*, un formato ligero para el intercambio de datos), por ser de menor tamaño que el formato XML (*eXtensible Markup Language*, lenguaje de marcas desarrollado por la W3C

utilizado para almacenar datos de forma legible), lo cual es de gran importancia a la hora de realizar una aplicación que consiste principalmente en la transferencia de datos.

2.5 Estrategia de desarrollo de aplicaciones móviles en entornos nativos

Para el desarrollo de aplicaciones móviles existe un conjunto de estrategias, siendo la estrategia de entornos de desarrollo nativos una de ellas. Este tipo de desarrollo permite que la experiencia al usuario sea óptima, pues se trata de aplicaciones diseñadas e implementadas para su contexto de ejecución. De esta manera, a su vez, se aprovechan los SDKs (*Software Development Kit*) y lenguajes de programación específicos de la plataforma para desarrollar la aplicación, así como los *market places* o mercados de aplicaciones dependientes de la plataforma para su distribución.

Dado al gran número de usuarios Android en Venezuela, el proyecto de pasantía hizo uso de la estrategia de desarrollo de aplicaciones móviles en entornos nativos para el desarrollo de la aplicación Club Mercado.

CAPÍTULO 3

MARCO TECNOLÓGICO

A continuación, se presentan los aspectos tecnológicos relacionados con el desarrollo del proyecto de pasantía, se describen brevemente y se indica su uso como parte de la solución propuesta.

3.1 *Python*

Es un lenguaje de programación interpretado, multiplataforma, multiparadigma y de código abierto cuya filosofía enfatiza una sintaxis que favorezca un código legible, explícito y transparente. Fue creado por Guido van Rossum a finales de la década de los ochenta como sucesor del lenguaje de programación ABC (Python).

3.2 *Django*

Django es un *framework* web de código abierto para el lenguaje *Python* utilizado para crear todo tipo de aplicaciones web. Construido por desarrolladores con experiencia, *Django* se ocupa de muchas de las molestias que pueden surgir durante el desarrollo web con miras a que el desarrollador pueda concentrarse en escribir aplicaciones rápida y limpiamente con un diseño pragmático sin necesidad de reinventar la rueda. Se mantiene en constante actualización gracias a la ayuda de más de tres mil cien colaboradores (Django Software Foundation).

Django utiliza el patrón MVC (Modelo-Vista-Controlador), pero con una interpretación donde la “vista” describe el dato que es presentado al usuario. Esto es: no es necesariamente cómo se ve el dato, sino qué dato se muestra. La vista describe cuál dato ve, no cómo lo ve.

3.3 *Django REST Framework*

Django REST Framework es un poderoso y flexible conjunto de herramientas para construir *Application Programming Interfaces*, o APIs, web (*Django REST Framework*). A continuación, se presentan algunas de las razones por las cuales *Django REST Framework* resulta útil:

- Una API web navegable es una gran ventaja de usabilidad para los desarrolladores.
- Las políticas de autenticación incluyen paquetes para OAuth1 y OAuth2.
- Es altamente personalizable.
- Posee una documentación extensa y una gran comunidad de apoyo.
- Es usado por compañías internacionalmente reconocidas como *Mozilla*, *Red Hat*, *Heroku* y *Eventbrite*.

3.4 *SQLite*

SQLite es un sistema de gestión de bases de datos relacional contenida en una biblioteca escrita en C relativamente pequeña. *SQLite* es un proyecto de dominio público.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de *SQLite* no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca *SQLite* se enlaza con el programa pasando a ser parte integral del mismo (*SQLite*).

En el desarrollo de aplicaciones Android, *SQLite* funge como el manejador de base de datos local de la aplicación. En el contexto MVC, el código escrito en *SQLite* corresponde a la capa del modelo.

3.5 XML

XML (*eXtensible Markup Language*) es un meta-lenguaje que permite definir lenguajes de marcas desarrollado por el *World Wide Web Consortium* (W3C) utilizado para almacenar datos en forma legible (W3C).

El aspecto visual y la interfaz de usuario en Android es definido por el código XML; es decir, el mismo corresponde al nivel de la vista en el contexto MVC.

3.6 Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible (Java).

En el desarrollo de aplicaciones Android, el código escrito en Java es el que define al controlador de la aplicación en el contexto del patrón MVC.

3.7 *Android Software Development Kit*

Es un *kit* de desarrollo de *software* para plataforma Android de *Google* para el desarrollo de aplicaciones nativas. El SDK (*Software Development Kit*) ofrece las APIs y las herramientas de desarrollo necesarias para desarrollar, probar y depurar aplicaciones para Android (Android).

3.8 Herramientas para el control de versiones y planificación

Durante el desarrollo de este proyecto de pasantías, se utilizaron diversas herramientas para llevar una correcta planificación del mismo, así como el control de versiones de sus archivos fuente generados. Estas herramientas se describen a continuación.

3.8.1 Git

Sistema de control de versiones distribuido diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente (Git).

3.8.2 *GitHub*

Servicio web de alojamiento para proyectos de desarrollo de software que utilizan el sistema de control de versiones Git (Finley, 2012).

3.8.3 Taiga

Taiga es una plataforma de código abierto destinada al manejo de proyectos para desarrollos ágiles y gerentes de proyectos que quieren una herramienta eficiente que haga del trabajo algo agradable (Taiga).

CAPÍTULO 4

MARCO METODOLÓGICO

En este capítulo se describe la metodología de trabajo utilizada en el desarrollo de este proyecto de pasantía, adaptada dada la naturaleza individual del mismo.

La metodología que ha adoptado la empresa IDBC Group para la realización de sus proyectos de desarrollo de software es SCRUM y por esto, es también la metodología utilizada durante el desarrollo de la aplicación móvil y API (*Application Programming Interface*) de Club Mercado. Esta metodología tiene un proceso de desarrollo iterativo incremental basado en entregas parciales y regulares del producto final al cliente (Softeng), lo cual la hace flexible y de rápida adaptación ante cualquier cambio en cada iteración o *Sprint*. Dicha metodología está definida por los elementos descritos en las secciones siguientes de este capítulo.

4.1 Roles

Cada miembro de un equipo SCRUM tiene especificado uno de los siguientes roles dentro del mismo:

4.1.1 Dueño del Producto o *Product owner*

Es aquel miembro del equipo que administra y define los requisitos del proyecto de desarrollo de *software* así como sus objetivos, agregando y organizando estos requisitos de acuerdo a prioridades para “maximizar el valor del producto”. Asimismo, representa a todas las personas interesadas en los resultados del mismo.

Para este proyecto de pasantías, el Ing. Carlos Inguanzo asumió el papel de *Product Owner*.

4.1.2 Equipo o *Team*

Equipo de profesionales autoorganizado, multidisciplinario y con un sistema jerárquico horizontal que desarrollan el proyecto. Preferiblemente, el equipo debe estar compuesto con un número suficientemente pequeño de miembros como para mantener las características de “trabajo ágil”, pero lo suficientemente grande como para cumplir a tiempo todas las tareas.

Este proyecto de pasantía fue realizado de manera estrictamente individual, por lo que el pasante asumió el papel de *Team*.

4.1.3 Facilitador o *Scrum master*

Es la persona encargada de liderar al equipo en miras de que todos los procesos internos se lleven de la mejor manera, cumpliendo con las reglas de SCRUM, a lo largo de todo el desarrollo del proyecto. Sirve de mediador entre el equipo de desarrollo y el dueño del producto, facilitando las reuniones y eliminando los impedimentos que puedan presentarse durante el desarrollo.

Para este proyecto de pasantías, el Br. Roberto Romero asumió el papel de *Scrum Master*.

4.1.4 *Stakeholders*

Son aquellas personas para quienes el proyecto producirá el beneficio esperado que justifica su producción, pues son las interesadas en la realización del proyecto de desarrollo. Su participación se limita a las revisiones de cada *sprint*.

4.2 Eventos

Los eventos son todas aquellas reuniones planificadas para el seguimiento del proyecto de desarrollo y pueden ser:

4.2.1 *Sprint*

Es aquel período, de un tiempo previamente fijado y constante para todo el proyecto, durante el cual el equipo trabaja para convertir un subconjunto de requerimientos en una nueva versión del *software* totalmente operativo. Los *sprints* para este proyecto de pasantía tuvieron una duración aproximada de seis días en promedio, por lo que se realizaron dieciséis *sprints*, que se detallan en el Capítulo 5.

4.2.2 *Sprint Planning*

El *Sprint Planning* es una reunión que se realiza antes del inicio de cada *Sprint*, donde el equipo de desarrollo determina la carga de trabajo que se compromete a completar en ese *sprint*, realizando la planificación del mismo. Al ser este proyecto de pasantía de carácter individual, esta reunión no se realizó. Sin embargo, sí se realizó la planificación al inicio de cada *sprint* para seleccionar aquellos elementos del *Product Backlog* que serían desarrollados en dicho *sprint*.

4.2.3 *Daily Scrum*

Reunión diaria, de máximo quince minutos, en la que el equipo informa sobre el estado del proyecto. Cada miembro responde a las siguientes tres preguntas:

- ¿Qué hiciste ayer?
- ¿Qué harás hoy?
- ¿Has tenido algún impedimento para alcanzar tu objetivo?

Debido a que la pasantía se realizó de forma individual, esta reunión no se llevó a cabo.

4.2.4 *Sprint Review*

Reunión que debe realizarse al final de cada *sprint* en la que el equipo de desarrollo presenta el trabajo completado durante el mismo a los interesados.

A los efectos de este proyecto de pasantía, esta reunión se realizó de manera informal con el Ing. Carlos Inguanzo, CEO de IDBC Group.

4.2.5 *Sprint Retrospective*

Después de cada *sprint* se lleva a cabo una retrospectiva del mismo, en la cual todos los miembros del equipo dan su opinión acerca del *sprint* recién superado en miras de mejorar continuamente el proceso de desarrollo. Como el proyecto de pasantía se realizó individualmente, esta reunión no se llevó a cabo.

4.3 Artefactos

La metodología *Scrum* hace uso de una serie de documentos que permiten su correcto funcionamiento y la comunicación del equipo completo. Entre ellos tenemos:

4.3.1 *Product Backlog*

Es un documento de alto nivel para todo el proyecto que consiste en una pila dinámica de requisitos denominados historias, descritos en un lenguaje no técnico y priorizados por valor de negocio. Decimos que es dinámica, pues los requisitos y prioridades se revisan y ajustan durante el curso del proyecto. Aquí, el *Product Owner* lista las características, funcionalidades, mejoras y correcciones del producto.

4.3.2 *Sprint Backlog*

Es un documento detallado y administrado por el equipo de desarrollo donde se describen, con una lista dinámica, todas las tareas a realizar para llevar a cabo las historias de un *sprint*.

CAPÍTULO 5

DESARROLLO DE LA APLICACIÓN

En este capítulo se describe el proceso de desarrollo del proyecto de pasantía. Realizado bajo las directrices de la metodología SCRUM y a lo largo de dieciséis *sprints*, comprendiendo las fases: especificación y análisis de requerimientos, diseño e implementación. A continuación, se describen las actividades realizadas en cada fase, las dificultades encontradas, artefactos generados y las soluciones tomadas a lo largo del desarrollo de cada *sprint*.

5.1 Primer *Sprint*

5.1.1 Objetivos

- Familiarizarse con el proyecto y con las herramientas a utilizar.

5.1.2 Actividades

5.1.2.1 Familiarización con los lenguajes de programación y herramientas a utilizar

Para cada uno de los aspectos involucrados en el desarrollo del proyecto, se procedió a realizar un estudio evaluando cuál sería el alcance del mismo.

Para lograr un buen manejo de las herramientas en un tiempo aceptable, se determinó como tarea realizar los tutoriales de *Django*, *Django REST Framework* y *Android Studio*.

5.1.2.2 Asistencia a charla sobre la metodología y procesos de IDBC Group

Como se especificó en el capítulo anterior, la empresa IDBC Group utiliza la metodología SCRUM en sus procesos de desarrollo. Para el estudio y comprensión de la misma, se dictó un charla introductoria en donde se dieron a conocer y explicaron detalladamente todos los elementos que la componen, especificando a su vez los roles de cada uno de los involucrados.

5.2 Segundo Sprint

5.2.1 Objetivos

- Familiarizarse con las funcionalidades de compra, notificaciones, control de acceso y manejo de saldo en Club Mercado.

5.2.2 Actividades

5.2.2.1 Familiarización con el modelo de negocio de Club Mercado

Club Mercado busca innovar y redefinir la manera en la que se hacen compras colaborativas de productos alimenticios a través de internet y es por esto, que su modelo de negocio puede resultar de difícil comprensión a primera instancia. En miras de aclararlo, se procedió a realizar una charla introductoria en donde se especificaron detalladamente las interacciones que tienen los usuarios entre sí para poder organizarse, realizar sus compras (también conocido como «participar») y recibirlas satisfactoriamente, así como la interacción con los proveedores, el manejo del saldo y el manejo de compras (o «participaciones»).

5.2.2.2 Estudio de la plataforma web desarrollada para Club Mercado

Durante esta actividad se estudió el servicio web de Club Mercado, así como el modelo de la base de datos del sistema, como actividad fundamental previa al desarrollo del API (*Application Programming Interface*) y de la aplicación móvil.

5.3 Tercer Sprint

5.3.1 Objetivos

- Diseñar un API para el sistema web Club Mercado e implementar sus funcionalidades básicas.

5.3.2 Actividades

5.3.2.1 Familiarización con la creación de APIs en *Python* (comparación entre *API Frameworks*)

En la ejecución de esta actividad se investigaron y compararon las funcionalidades ofrecidas por dos de los *API Frameworks* más usados para trabajar en conjunto con *Django*: *Tastypie* y *Django REST Framework*, eligiendo este último dada su extensa documentación.

5.3.2.2 Diseño de los servicios web a ser un API para el sistema web Club Mercado

Durante esta actividad, se procedió a diseñar un API para el sistema web Club Mercado que contemplara todas aquellas funcionalidades y modelos a tomar en cuenta por la aplicación móvil para su correcto funcionamiento. Se determinaron qué modelos de la base de datos serían representados, así como la selección de aquellos atributos considerados necesarios y suficientes para esto. Asimismo, se definieron las funcionalidades que el API debía satisfacer para cumplir con los requerimientos de este proyecto de pasantía.

5.3.2.3 Creación del API para comunicación cliente-servidor de Club Mercado

Haciendo uso de la herramienta *Django REST Framework*, se implementaron las clases apropiadas para la representación de los modelos de la base de datos a formato JSON (*JavaScript Object Notation*), llamadas *serializers*, de aquellos modelos cuyos datos son de acceso público y de sólo lectura, así como las rutas de acceso a dichos datos a través de la web. Finalmente, se implementaron en el API las funcionalidades del manejo de sesión mediante *tokens*.

5.4 Cuarto Sprint

5.4.1 Objetivos

- Implementar las funcionalidades avanzadas en el API del sistema web Club Mercado.

5.4.2 Actividades

5.4.2.1 Implementación del módulo de permisos sobre el API

Durante esta actividad se procedió a definir el módulo de permisos, los cuales determinan quiénes pueden, a través del API, realizar ciertas acciones tales como crear, actualizar o listar instancias de la base de datos. Esto fue posible gracias al módulo de permisos *de Django REST Framework* que contempla, entre otros, permisos a nivel de objetos y a nivel de usuarios.

5.4.2.2 Implementación de las funcionalidades para la muestra, creación y actualización de los datos correspondientes a los modelos de interés de la base de datos a través del API

Una vez definidos los permisos, se implementaron todos los métodos necesarios para la muestra (a través del método HTTP GET¹), creación y actualización (usualmente a través del método HTTP POST²) de instancias en la base de datos del sistema web Club Mercado correspondientes a los servicios web a ser consumidos, a saber: los servicios correspondientes al manejo de compras, usuarios y saldos.

5.5 Quinto *Sprint*

5.5.1 Objetivos

- Desarrollar un tour de inicio que muestre las principales funcionalidades disponibles en la aplicación móvil.

5.5.2 Actividades

5.5.2.1 Análisis, diseño e implementación de la interfaz el tour de inicio que desemboque en el registro o inicio de sesión de usuarios

¹ El método de petición GET está diseñado para obtener información de un servidor web.

² El método de petición POST está diseñado para hacer peticiones a un servidor que acepte el tipo de datos adjunto en el cuerpo del mensaje, siendo enviado para su almacenamiento.

A partir del diseño gráfico creado previamente para la interfaz del tour de inicio, se procedió a su implementación. Es la primera vista que se presenta al usuario al abrir la aplicación sólo si no ha iniciado sesión. Esta interfaz muestra información a través de cuatro pantallas o *tabs* acerca de cómo funcionan los procesos de compra en Club Mercado en miras de familiarizarlo con éste y, desde ella, se puede acceder, mediante dos botones, a la interfaz del módulo de registro de usuario o a la interfaz de módulo de inicio de sesión y restablecimiento de contraseñas.

Como artefactos generados para este *sprint*, se agregaron capturas de pantalla o *screenshots* disponibles en las Figuras C.1, C.2, C.3 y C.4 del Apéndice C y el código fuente generado.

5.6 Sexto Sprint

5.6.1 Objetivos

- Desarrollar un módulo para el acceso y autenticación del usuario.

5.6.2 Actividades

5.6.2.1 Análisis, diseño e implementación de la interfaz de inicio y cierre de sesión

Se implementó la interfaz de inicio de sesión a partir de diseños previamente establecidos y utilizando la librería *HttpClient* de Apache en el dispositivo Android.

En esta vista, se muestra un formulario con los dos campos correspondientes al correo electrónico y a la contraseña, así como un botón para realizar la acción de inicio de sesión. Dicho inicio de sesión se realiza mediante una petición HTTP POST al servicio web de Club Mercado, donde se autentica al usuario y se retorna una respuesta en formato JSON con sus credenciales (o *token*) las cuales se almacenan en las preferencias compartidas de la aplicación, que son accedidas cada vez que se necesite hacer una petición al servidor donde se precise la autenticación del usuario. Al ser una autenticación basada en *tokens*, no es necesario el manejo de *cookies* por parte del servidor.

El cierre de sesión, en cambio, puede realizarse desde cualquier vista en la aplicación. Al realizar dicha acción, se eliminan los datos almacenados tanto en las preferencias compartidas

como en la base de datos local de la aplicación, redireccionando finalmente al usuario a la vista de inicio de sesión.

Como artefactos generados para este *sprint*, se agregó una captura de pantalla o *screenshot* disponible en la Figura C.5 del Apéndice C y el código fuente generado.

5.6.2.2 Análisis, diseño e implementación de la interfaz de recuperación de contraseñas

Al igual que para la actividad anterior, se implementó la interfaz de inicio de sesión utilizando la librería *HttpClient* de Apache en el dispositivo Android y la librería *Django-rest-auth* en el servidor. En esta interfaz se muestra un campo correspondiente al correo electrónico y un botón para realizar la acción de restablecimiento de contraseñas, el cual envía una petición HTTP POST al servicio web de Club Mercado, donde se envía un mensaje al correo electrónico especificado que contiene los pasos para restablecer la contraseña exitosamente.

Como artefactos generados para este *sprint*, se agregó una captura de pantalla o *screenshot* disponible en la Figura C.6 del Apéndice C y el código fuente generado.

5.7 Séptimo Sprint

5.7.1 Objetivos

- Desarrollar un módulo para el registro de usuario.
- Desarrollar la base de datos local y el motor de sincronización de datos para los modelos afectados.

5.7.2 Actividades

5.7.2.1 Análisis, diseño e implementación de una base de datos local y motor de sincronización de datos para los modelos correspondientes al módulo de manejo de usuarios

Debido a que el tamaño de los datos a ser almacenados por la aplicación móvil es relativamente pequeño, se optó por implementar una base de datos relacional mediante la librería *SQLite*, la cual comprende un ligero motor de base de datos de código abierto que permite el almacenamiento persistente de los datos de forma sencilla.

Para ello, se optó por replicar, a través de sus clases correspondientes, las tablas ya existentes en el servidor, a saber: *User*, *UserAttributes* y *Address*, pues la información de usuario es suministrada por el API de dicha forma.

5.7.2.2 Análisis, diseño e implementación de la interfaz de registro de usuario

La interfaz de registro de usuario se implementó a través de la librería *HttpClient*. Dicha interfaz es comprendida por ocho campos que corresponden al nombre, apellido, cédula de identidad, número celular, correo electrónico, contraseña, confirmación de contraseña y dirección del usuario, así como tres listas desplegables correspondientes al estado, ciudad y municipio a la cual pertenece la dirección especificada y un botón que realiza la validación de los datos proporcionados y el registro del usuario.

Para cargar la información de las listas desplegables, se hace una petición HTTP GET al API de Club Mercado, que retorna los estados, ciudades y municipios disponibles en la plataforma en formato JSON. Finalmente, para realizar el registro de usuario, se realizan, en orden, tres peticiones HTTP POST con los datos correspondientes a las tablas *User*, *Address* y *UserAttributes* respectivamente.

Como artefactos generados para este *sprint*, se agregó una captura de pantalla o *screenshot* disponible en la Figura C.7 del Apéndice C y el código fuente generado.

5.8 Octavo Sprint

5.8.1 Objetivos

- Desarrollar la primera fase de la interfaz de un perfil de usuario que comprenda la visualización o edición de los datos personales, compras realizadas y estado de cuenta del usuario.

5.8.2 Actividades

5.8.2.1 Análisis y diseño de la interfaz de un perfil de usuario que comprenda la visualización o edición de los datos personales, compras realizadas y estado de cuenta del usuario

Durante esta actividad se procedió a analizar la mejor manera de implementar una interfaz que integrara, en un sólo módulo, las tres funcionalidades básicas requeridas que comprende el perfil de usuario en la aplicación móvil Club Mercado. Se determinó la implementación de tres pestañas (o *tabs*) correspondientes a cada una de las tres funcionalidades: la pestaña de "Datos", la pestaña de "Colecciones" y la pestaña de "Saldo". El perfil de usuario es accedido mediante la opción "Perfil" desde el menú de la aplicación.

5.8.2.2 Implementación de la interfaz de visualización y modificación de datos personales en el perfil de usuario

La implementación de esta interfaz viene dada con la obtención de los datos del usuario (desde la base de datos local) para ser mostrados en la interfaz gráfica. Una vez obtenidos y mostrados, los datos pueden ser modificados por el usuario. Dichas modificaciones pueden ser canceladas o salvadas, siendo entonces, en este último caso, enviada una petición HTTP POST con los datos actualizados en formato JSON al API web de Club Mercado y actualizándose la base de datos local de ser exitosa dicha petición.

Para visualizar o modificar los datos, se debe seleccionar la opción "Perfil" desde el menú de la aplicación. Asimismo, la edición de los campos que pueden ser modificados es activada mediante el botón "Modificar información".

Como artefactos generados para este *sprint*, se agregó una captura de pantalla o *screenshot* disponible en la Figura C.8 del Apéndice C y el código fuente generado.

5.9 Noveno *Sprint*

5.9.1 Objetivos

- Desarrollar la segunda fase de la interfaz de un perfil de usuario que comprenda la visualización o edición de los datos personales del usuario, compras realizadas y estado de cuenta del usuario.

5.9.2 Actividades

5.9.2.1 Diseño e implementación de la interfaz de visualización de las estadísticas de compra realizadas en el perfil de usuario

La visualización de las estadísticas de compra realizadas por el usuario se implementó con la generación de una lista de las colecciones³ en las que ha participado el usuario junto con la cantidad neta de dinero gastado en ellas, así como su porcentaje respecto a compras realizadas a otras colecciones. La información de esta lista es obtenida a través de una petición HTTP GET al API web de Club Mercado mediante el controlador *UserCollectionsAdapter.java* cada vez que el usuario accede a su perfil.

Para visualizar las estadísticas concernientes a las compras realizadas por el usuario, se debe acceder al perfil (como se especificó en la sección anterior) y deslizar la pantalla una vez hacia la derecha.

Como parte de los artefactos del caso de uso "visualizar estadísticas de compra", se tiene la captura de pantalla de la Figura C.9 del Apéndice C y el código fuente generado.

³ Se entiende por "colección" en Club Mercado al conjunto de productos ofrecidos por un proveedor en específico, los cuales generalmente comparten características en común.

5.10 Décimo Sprint

5.10.1 Objetivos

- Desarrollar la tercera fase de un perfil de usuario que comprenda la visualización o edición de los datos personales del usuario, compras realizadas y estado de cuenta del usuario.

5.10.2 Actividades

5.10.2.1 Diseño e implementación de la interfaz de visualización de saldo y movimientos realizados en el perfil de usuario

La implementación de esta interfaz viene dada por la muestra del saldo actual del usuario, obtenido a través de la base de datos local mediante una consulta a la tabla *UserAttributes*; así como la de una lista generada con todos los movimientos realizados sobre el mismo (donde cada fila contiene información acerca del tipo de movimiento realizado, la cantidad de dinero involucrada en él y la fecha en la que se realizó), obtenidos mediante una petición HTTP GET al API web de Club mercado a través del controlador *UserBalanceMovementsAdapter.java* cada vez que el usuario accede a su perfil. Además, se implementó un botón que lleva a la funcionalidad de abonar, la cual es descrita en secciones posteriores de este capítulo.

Para visualizar los movimientos de saldo del usuario, se debe acceder al perfil (como se especificó en la sección anterior) y deslizar la pantalla dos veces hacia la derecha.

Como parte de los artefactos del caso de uso "visualizar movimientos sobre el saldo", se tiene la captura de pantalla de la Figura C.10 del Apéndice C y el código fuente generado.

5.11 Décimo primer Sprint

5.11.1 Objetivos

- Implementar la primera fase del módulo de compras.

5.11.2 Actividades

5.11.2.1 Análisis, diseño e implementación de la interfaz para la visualización de categorías que desemboque en el pasillo de sus productos asociados

Esta interfaz se implementó mediante una lista, cargada dinámicamente a partir de una petición HTTP GET al API web de ClubMercado gracias al controlador *ActiveCollectionsAdapter.java*, con los datos más relevantes de las categorías de productos, tales como sus nombres, descripciones, fechas de inicio y cierre, así como sus fotos y eslóganes. Adicionalmente, atado a cada colección o categoría, se implementó un botón que lleva al listado de los productos asociados a las mismas, funcionalidad descrita en el siguiente punto. En esta actividad se implementó también la ejecución de todos los algoritmos para la actualización de los datos encontrados en la base de datos local, sincronizándolos con aquellos encontrados en el servidor; esto, a través de peticiones HTTP GET a los servicios ofrecidos por el API correspondientes. Por último, en esta interfaz se implementó el menú de la aplicación mediante las librerías y módulos ofrecidos por el entorno Android para tal fin.

La visualización de categorías o "colecciones" funge como la actividad principal de la aplicación móvil Club Mercado. Dicha interfaz es accedida cada vez que se ejecuta la aplicación si el usuario ya ha iniciado sesión, o desde el formulario de inicio de sesión si los datos proporcionados en él son válidos.

Como parte de los artefactos del caso de uso "visualizar categorías", se tiene la captura de pantalla de la Figura C.11 del Apéndice C y el código fuente generado.

5.11.2.2 Análisis, diseño e implementación de la interfaz para la visualización de productos dentro de categorías

Para esta interfaz resultó necesaria la implementación de una lista, cargada dinámicamente, con la información de los productos pertenecientes a dicha colección activa junto a sus detalles: nombre, foto, cantidad y unidad de venta y precio. La carga de los datos en esta lista se logra a través de una petición HTTP GET al API web de Club Mercado, donde se hace

uso del *id* de la participación, el cual es pasado como argumento al *Activity* mediante una instancia de la clase *Bundle* proporcionada por la librería Android.

La interfaz para la visualización de productos dentro de categorías o colecciones activas es accedida mediante el botón "Participar", el cual se encuentra anidado en cada colección de la lista de colecciones activas previamente descrita.

5.12 Décimo segundo Sprint

5.12.1 Objetivos

- Implementar la segunda fase del módulo de compras

5.12.2 Actividades

5.12.2.1 Implementación de la funcionalidad de visualizar y agregar al carrito de compras

Con la lista de productos cargada se procedió a añadir, para cada uno de estos productos, un campo numérico que puede ser editado por el usuario de la aplicación, correspondiente a la cantidad de dicho producto que el usuario desea integrar a su compra.

Al modificar este campo se actualiza, con el identificador del producto añadido y su respectiva cantidad, la instancia de la clase *Cart* (definida dentro del alcance de la clase controlador asociada a la interfaz) activa en la ejecución de la actividad. Asimismo, se actualiza el monto total estimado del carrito de compra correspondiente a la participación (o compra) que se muestra al usuario. Para lograr esto, se hizo uso de la clase *TextWatcher* provista por Android.

Durante el transcurso de esta actividad, se vio la necesidad de replantear y diseñar de nuevo el concepto de carrito de compra definido previamente por cuestiones relacionadas a la practicidad, pues los mismos no podían ser implementados según el modelo de la base de datos del servidor dado a su costo de procesamiento. Además, un cambio en el modelo de negocios solicitado por el *Product Owner* del proyecto, hizo del carrito de compras una funcionalidad meramente informativa en contraste con el paso previo a la realización de la compra que estaba pensado ser.

5.13 Décimo tercer *Sprint*

5.13.1 Objetivos

- Implementar la tercera fase del módulo de compras

5.13.2 Actividades

5.13.2.1 Implementación de la funcionalidad de búsqueda de productos

Es habitual que una sola lista pueda llegar a tener hasta cien productos y, es por esto, que resulta práctico el tener una funcionalidad que permita la búsqueda de los productos deseados. Para ello, se integró en la interfaz del módulo de compras un campo de texto que, habilitado después de presionar un botón, permite al usuario escribir una cadena de caracteres que éste considere lo ayudará en su búsqueda mientras que, a medida que lo hace, se va filtrando la lista de productos mostrando sólo aquellos cuyos nombres la contengan.

Esta funcionalidad fue implementada mediante la utilización de la clase *TextWatcher* provista por Android sobre el campo de texto y realizando una búsqueda lineal sobre los campos correspondientes al nombre de los productos, ocultando aquellas filas que no cumplieran con la condición de búsqueda.

5.13.2.2 Implementación del cierre de compra

El cierre de compra o "participación" viene dado una vez que el usuario está de satisfecho con los productos que ha agregado al carrito de compras y con el precio total del mismo. Para realizarlo, el mismo debe presionar el botón "Participar", que envía el resumen del carrito de compras generado por la aplicación al servidor para que éste lo procese, generando a su vez su historial de participación correspondiente, entre otras cosas.

Para lograr esto, se hizo uso de la librería *HttpClient* para enviar una petición HTTP POST al API web de Club Mercado, la cual contiene, en formato JSON, los datos correspondiente al carrito de compras (es decir, los identificadores de los productos así como sus

cantidades requeridas), el identificador del usuario que hace la compra y el identificador de la colección en la cual se está participando.

Como parte de los artefactos del caso de uso "visualizar categorías", se tiene la captura de pantalla de la Figura C.12 del Apéndice C y el código fuente generado.

5.14 Décimo cuarto *Sprint*

5.14.1 Objetivos

- Implementar el módulo de abonos al saldo

5.14.2 Actividades

5.14.2.1 Análisis y diseño de la interfaz para el abono de saldo y visualización de abonos realizados

Se requería que esta interfaz implementara dos funcionalidades: la carga de los datos de una transferencia realizada por el usuario con anterioridad mediante un formulario y el listado de todos los abonos de saldo hechos por el mismo. Por ello, se determinó nuevamente el uso de pestañas (o *tabs*), una para cada funcionalidad.

La interfaz para el abono de saldo puede ser accedida mediante la opción "Abonar" del menú principal de la aplicación o desde el perfil de usuario, en la sección correspondiente al saldo, mediante el botón "Abonar".

5.14.2.2 Implementación de la interfaz de visualización de abonos realizados

En esta pestaña se genera, dinámicamente, una lista con todos los abonos (transferencias bancarias) realizadas por el usuario, así como su información más relevante: fecha de la realización, banco emisor, banco receptor, nombre y cédula de identidad del depositante, el número de transacción bancaria y el monto abonado.

La información de esta lista es obtenida a través de una petición HTTP GET al API web de Club Mercado mediante el controlador *PaymentHistoryAdapter.java* cada vez que el usuario accede a la interfaz.

Para visualizar los abonos de saldo realizados por el usuario, se debe acceder a la interfaz de abonar (como se especificó en la sección anterior) y deslizar la pantalla una vez hacia la derecha.

Como parte de los artefactos del caso de uso "visualizar abonos de saldo", se tiene la captura de pantalla de la Figura C.13 del Apéndice C y el código fuente generado.

5.15 Décimo quinto *Sprint*

5.15.1 Objetivos

- Implementar la segunda fase del módulo de abonos al saldo

5.15.2 Actividades

5.15.2.1 Implementación de la interfaz del formulario de abono de saldo

Se requería que esta interfaz implementara un formulario conformado por cinco campos de texto correspondientes al nombre, correo electrónico y cédula del depositante, monto y número de transacción de la transferencia; dos menús desplegables correspondientes al banco receptor y emisor; un campo de inserción de fechas correspondiente a la fecha en que se realizó la transferencia y un botón para realizar la acción de abonar. Para llevar a cabo dicha implementación, se hizo uso de las clases provistas por el entorno Android para la obtención de datos del usuario, tales como las clases *EditText*, *Spinner* y *Datepicker*. Una vez obtenidos los datos requeridos, los mismos son enviados en formato JSON (luego de ser verificados) en una petición HTTP POST al API Web de Club Mercado, mostrando un mensaje de éxito de ser exitosa dicha petición, esto último gracias a la clase *Toast* de Android.

Para realizar la carga de datos de una transferencia bancaria realizada, se debe acceder a la interfaz del módulo de abonos, tal y como se especificó en secciones anteriores.

Como parte de los artefactos del caso de uso "visualizar abonos de saldo", se tiene la captura de pantalla de la Figura C.14 del Apéndice C y el código fuente generado

5.16 Décimo sexto *Sprint*

5.16.1 Objetivos

- Desplegar la aplicación móvil en la plataforma Google Play.

5.16.2 Actividades

5.16.2.1 Realización del despliegue de la aplicación móvil en la plataforma *Play Store* de *Google*

La realización del despliegue de la aplicación en la plataforma *Play Store* de *Google* se logró mediante lanzamientos progresivos bajo las pruebas *Alpha* y *Beta* provistas por la plataforma.

En primer lugar, se subió en su versión *Alpha* para que la misma fuese probada por un número controlado y limitado de usuarios en miras a corregir los errores que pudiera presentar.

Una vez que la aplicación fue considerada (tanto por el desarrollador como por el *Product Owner* del proyecto) lista para probarse en versión *Beta*, se procedió a hacer su lanzamiento progresivo en dicha versión.

Finalmente, como el sistema web de Club Mercado aún se encuentra en desarrollo, la aplicación queda pendiente por ser desplegada y publicada completamente en producción una vez que éste se encuentre también en producción.

5.17 Dificultades generales encontradas

Las principales dificultades encontradas a lo largo del desarrollo del proyecto de pasantía fueron:

- Falta de personal familiarizado con la plataforma Android en la empresa: a medida que el desarrollo del proyecto avanzaba y se requerían nuevas funcionalidades, el mismo se veía afectado, pues no se contaba con alguien familiarizado con la plataforma Android en la empresa que ayudara a esclarecer las dudas de implementación que pudieran surgir, retrasando, a menudo, dicho desarrollo.
- Cambio constante en los requerimientos del sistema web: al ser aún un proyecto de desarrollo bajo la metodología SCRUM, el sistema web de Club Mercado se veía afectado constantemente por cambios en sus requerimientos, afectando también, directa o indirectamente, el desarrollo de su aplicación móvil, pues los mismos implicaban, a menudo, cambios (a veces drásticos) en el API Web y por consiguiente, en la aplicación móvil.

CONCLUSIONES Y RECOMENDACIONES

El desarrollo de este proyecto de pasantías finalizó exitosamente y en el tiempo estipulado, derivando en la primera versión de la aplicación móvil Club Mercado para dispositivos de la plataforma Android, la cual cumple con todos los requerimientos propuestos al principio del mismo. Para ello, se desarrollaron e implementaron tanto el API (*Application Programming Interface*) web de Club Mercado como los módulos de autorización, autenticación, usuarios, compras y abono de saldo de la aplicación móvil; permitiendo a los usuarios con dispositivos de la plataforma Android acceder a las funcionalidades básicas e indispensables del sistema web.

El proyecto fue realizado bajo la metodología SCRUM, la cual permitió que su desarrollo fuese planificado, eficiente y organizado, llevando a cabo los cambios necesarios para adaptarse a los nuevos requerimientos que constantemente fueron surgiendo a lo largo del mismo. Como se explicó en el Capítulo 4, SCRUM es una metodología que trabaja mejor con equipos de desarrollo conformados por varias personas; pero, al ser individual la realización del proyecto de pasantía, se optó por adaptar dicha metodología a esta condición. El proyecto estuvo comprendido por dieciséis *sprints* en donde se fueron tomando y desarrollando las historias de usuarios encontradas en el *sprint backlog*. Por su parte, el *product backlog* se creó y gestionó con la ayuda de la herramienta Taiga, la cual permitió también el desarrollo organizado de la aplicación.

La aplicación móvil para Android se realizó bajo el patrón de arquitectura MVC de forma nativa. Durante la implementación del proyecto de pasantía, se utilizaron herramientas como el Android SDK y la librería *Django REST Framework*, las cuales facilitaron enormemente su desarrollo.

Como recomendaciones tras la finalización de este proyecto de pasantía, se tiene, en primer lugar, realizar las pruebas automatizadas para cada uno de sus componentes, a manera de poder contar con un producto más robusto y con el menor número de errores posible. Finalmente,

la segunda recomendación va orientada hacia la mantenibilidad de la aplicación, dada la relevancia que tienen las diferentes versiones del sistema operativo Android y cómo estas versiones afectan directamente el rendimiento de la aplicación, resulta imperante mantenerse al día con estos avances.

REFERENCIAS BIBLIOGRÁFICAS

- Android. (s.f.). *Android*. Recuperado el 16 de Noviembre de 2016, de Android: <https://developer.android.com/studio/index.html>
- Django REST Framework. (s.f.). *Django REST Framework*. Recuperado el 16 de Noviembre de 2016, de Django REST Framework: <http://www.django-rest-framework.org/>
- Django Software Foundation. (s.f.). *Django Project*. Recuperado el 19 de Febrero de 2017, de Django Project: <https://www.djangoproject.com/start/overview/>
- EcuRed. (s.f.). *EcuRed*. Recuperado el 19 de Febrero de 2017, de EcuRed: <https://www.ecured.cu/Cliente-Servidor>
- Finley, K. (14 de Julio de 2012). *TechCrunch*. Recuperado el 16 de Noviembre de 2016, de TechCrunch: <http://techcrunch.com/2012/07/14/what-exactly-is-github-anyway>
- Git. (s.f.). *Git SCM*. Recuperado el 19 de Febrero de 2017, de Git SCM: <https://git-scm.com/>
- Inguanzo, C. (15 de Julio de 2016). Declaración de Principios, Reglas y Valores de IDBC Group. Chacao, Miranda, Venezuela.
- Inguanzo, C. (2016, Julio 15). Declaración de Principios, Reglas y Valores de IDBC Group. Chacao, Miranda, Venezuela.
- Java. (s.f.). *Java*. Recuperado el 19 de Febrero de 2017, de Java: https://www.java.com/en/download/faq/whatis_java.xml
- Merino, M. (12 de Julio de 2014). *Tic Beat*. Recuperado el 19 de Febrero de 2017, de Tic Beat: <http://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/>
- Python. (s.f.). *Python*. Recuperado el 19 de Febrero de 2017, de Python: <https://www.python.org/doc/essays/blurb/>
- Softeng. (s.f.). *Softeng*. Recuperado el 17 de Noviembre de 2016, de Softeng: <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>
- SQLite. (s.f.). *SQLite*. Recuperado el 19 de Febrero de 2017, de SQLite: <https://www.sqlite.org/about.html>
- Taiga. (s.f.). *Taiga*. Recuperado el 16 de Noviembre de 2016, de Taiga: <https://taiga.io/>

Trygve, R., & Coplien, J. O. (20 de Marzo de 2009). *Artima Developer*. Recuperado el 19 de Febrero de 2017, de Artima Developer: http://www.artima.com/articles/dci_vision.html

W3C. (s.f.). W3C. Recuperado el 19 de Febrero de 2017, de W3c: https://www.w3schools.com/xml/xml_what_is.asp

APÉNDICE A. Diagrama de Casos de Uso

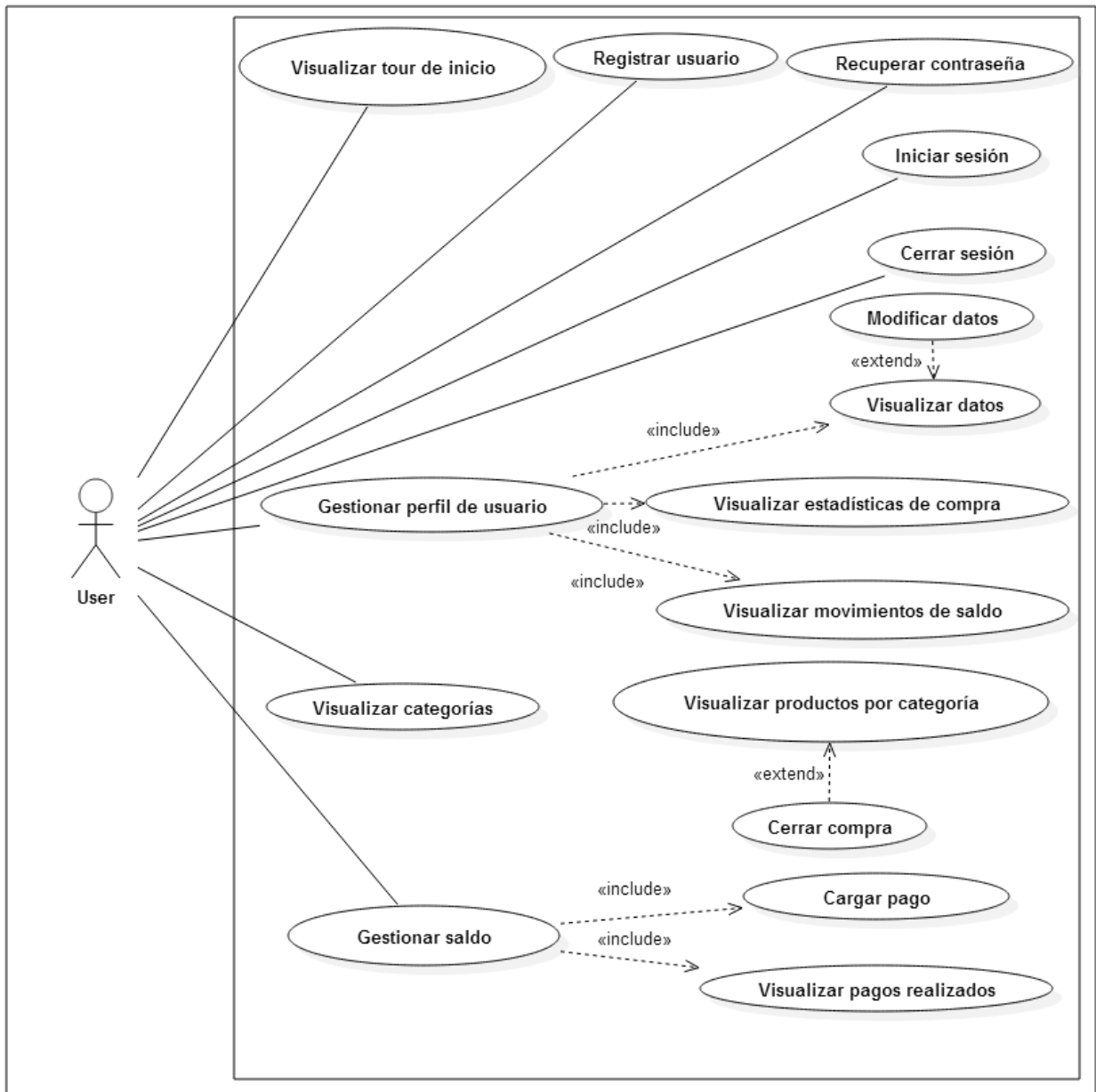


Figura A.1: Diagrama de Casos de Usos de la aplicación móvil Club Mercado para Android

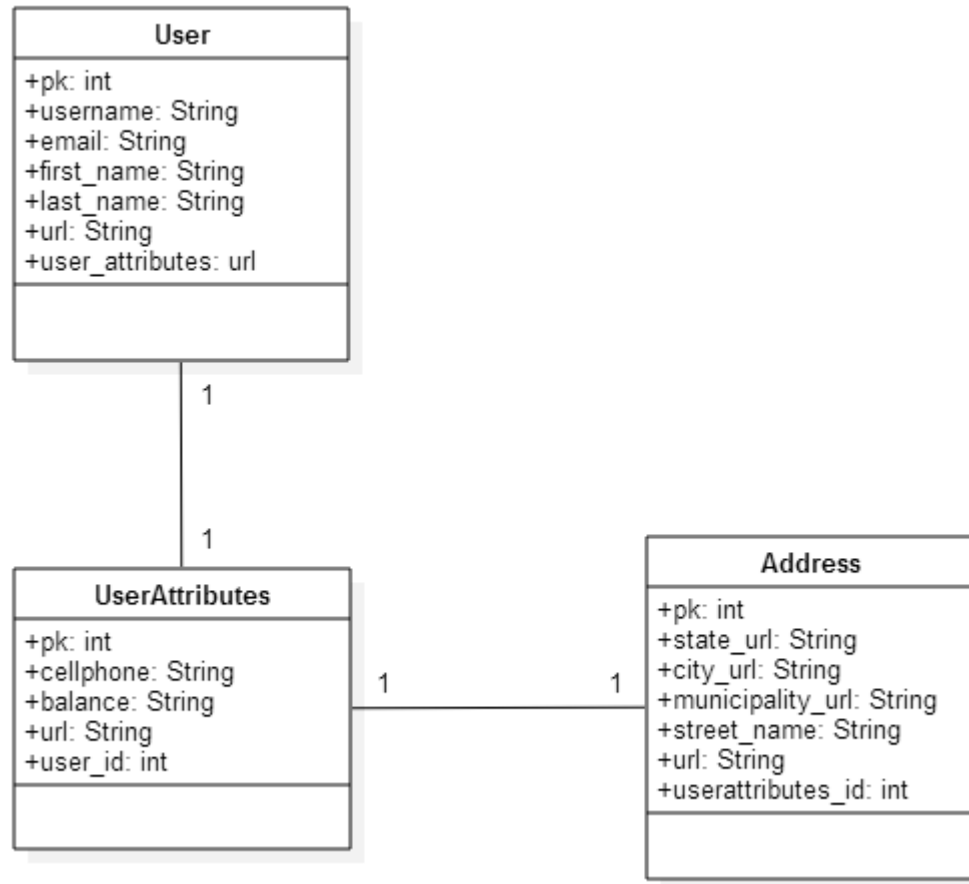
APÉNDICE B. Diagrama de Clases

Figura B.1: Diagrama de Clases de la aplicación móvil Club Mercado para Android

APÉNDICE C. *Screenshots* del Sistema Contenido

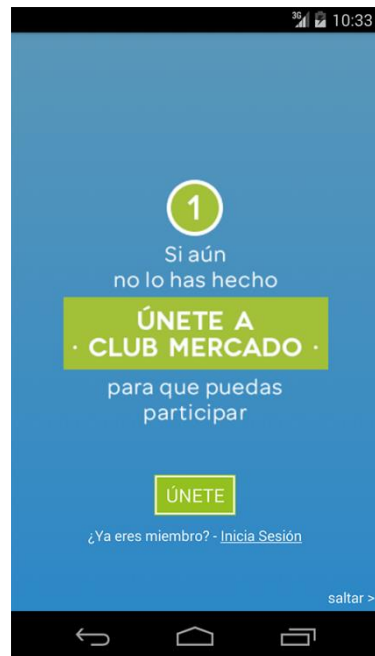


Figura C.1: *Screenshot* de la vista de tour de inicio (1)



Figura C.2: *Screenshot* de la vista de tour de inicio (2)



Figura C.3: *Screenshot* de la vista de tour de inicio (3)



Figura C.4: *Screenshot* de la vista de tour de inicio (4)

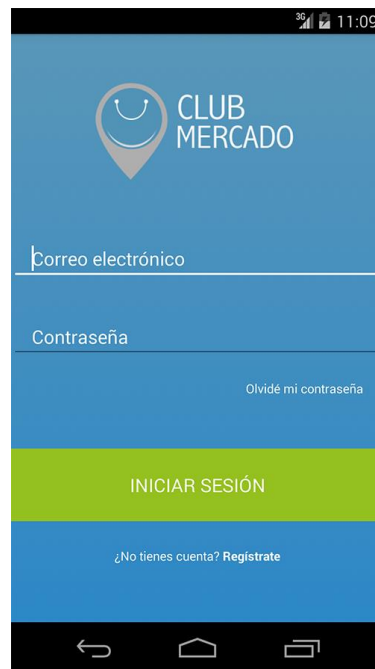
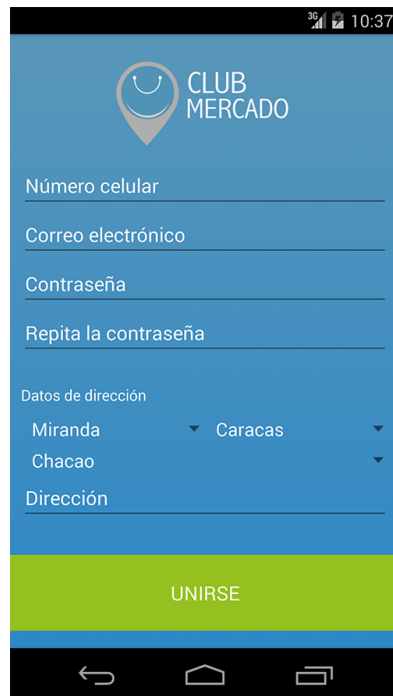


Figura C.5: *Screenshot* de la vista de inicio de sesión

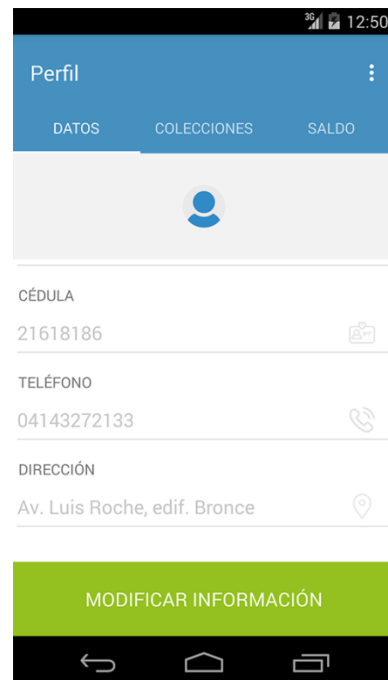


Figura C.6: *Screenshot* de la vista de reinicio de contraseña



The screenshot shows the user registration screen of the CLUB MERCADO app. At the top, there is a logo with a smiley face inside a location pin and the text "CLUB MERCADO". Below the logo, there are four input fields: "Número celular", "Correo electrónico", "Contraseña", and "Repita la contraseña". Under these fields, there is a section titled "Datos de dirección" with two dropdown menus: "Miranda" and "Caracas", and a "Chacao" dropdown menu. Below these, there is a "Dirección" input field. At the bottom of the form is a green button labeled "UNIRSE". The status bar at the top shows the time as 10:37.

Figura C.7: *Screenshot* de la vista de registro de usuario



The screenshot shows the user profile management screen of the CLUB MERCADO app. At the top, there is a header "Perfil" with a menu icon. Below the header, there are three tabs: "DATOS", "COLECCIONES", and "SALDO". The "DATOS" tab is selected. Below the tabs, there is a profile picture placeholder. Underneath, there are three rows of personal data: "CÉDULA" with the value "21618186", "TELÉFONO" with the value "04143272133", and "DIRECCIÓN" with the value "Av. Luis Roche, edif. Bronce". Each row has an edit icon. At the bottom of the form is a green button labeled "MODIFICAR INFORMACIÓN". The status bar at the top shows the time as 12:50.

Figura C.8: *Screenshot* de la vista de gestión de datos personales en el perfil de usuario

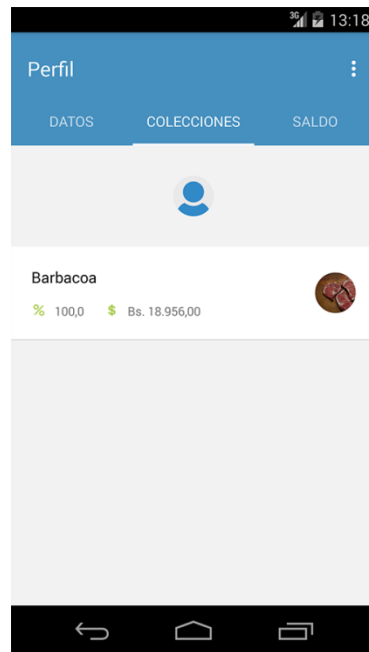


Figura C.9: *Screenshot* de la vista de visualización de estadísticas de compra en el perfil de usuario



Figura C.10: *Screenshot* de la vista de visualización de saldo y movimientos realizados en el perfil de usuario

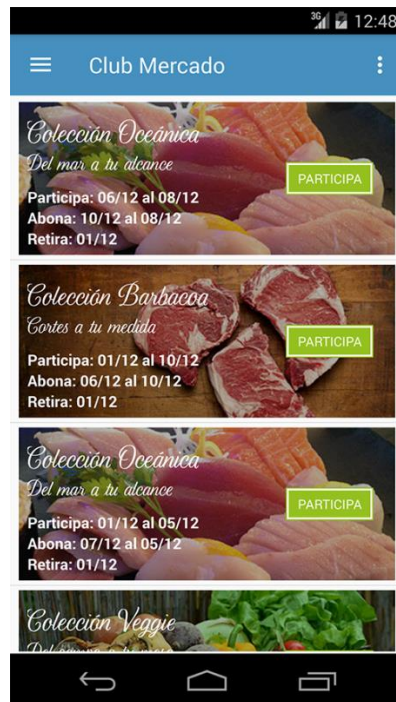


Figura C.11: *Screenshot* de la vista de visualización de categorías

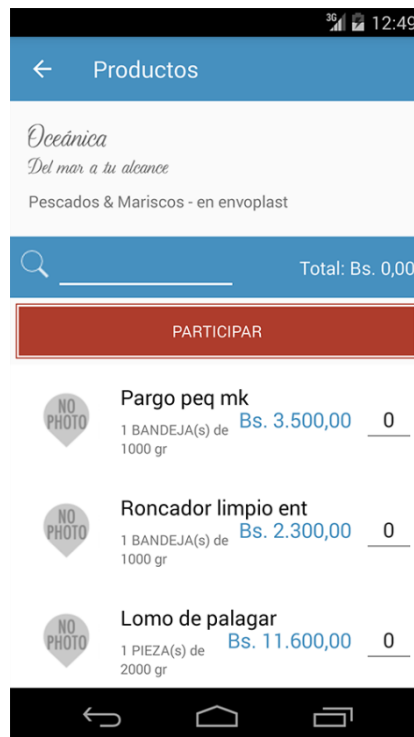


Figura C.12: *Screenshot* de la vista de pasillo de compra de productos por categoría

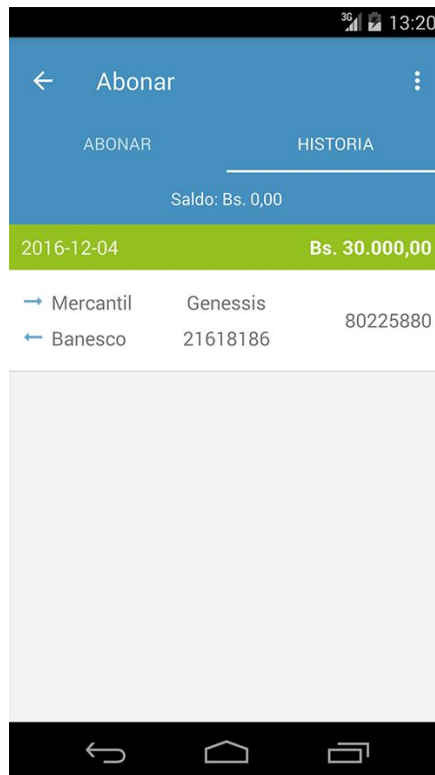


Figura C.13: *Screenshot* de la vista de visualización de abonos realizados

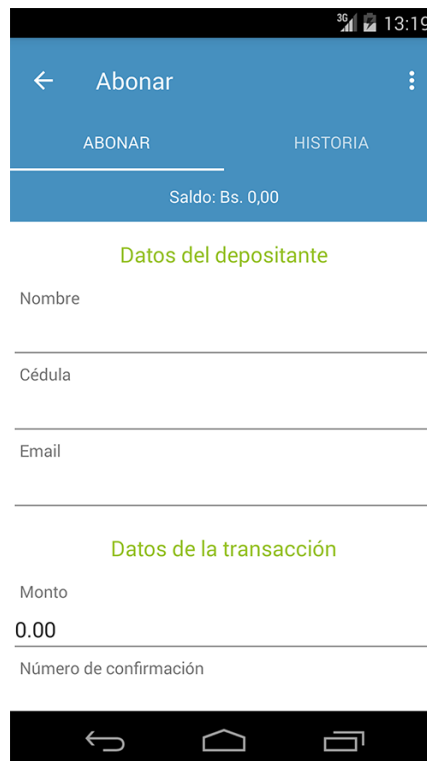


Figura C.14: *Screenshot* de la vista del formulario de abono de saldo