

**DESARROLLO DE UN APLICATIVO MÓVIL EN ANDROID PARA LA
CREACIÓN E INTERCAMBIO DE TARJETAS PERSONALES Y
EMPRESARIALES DEPOSITADAS EN LA WEB**

**ANDRÉS GIRALDO CARVAJAL
CARLOS ALFREDO LONDOÑO SÁNCHEZ**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN
PEREIRA
2015**

**DESARROLLO DE UN APLICATIVO MÓVIL EN ANDROID PARA LA
CREACIÓN E INTERCAMBIO DE TARJETAS PERSONALES Y
EMPRESARIALES DEPOSITADAS EN LA WEB**

**ANDRES GIRALDO CARVAJAL
CARLOS ALFREDO LONDOÑO SÁNCHEZ**

**Proyecto de grado para optar por el título de Ingeniero de Sistemas y
Computación**

**Asesora
M. Sc. Paula Andrea Villa Sánchez**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN
PEREIRA
2015**

Nota de aceptación:

Firma del jurado

Pereira, 25 de Mayo de 2015

DEDICATORIA

A nuestras familias...

AGRADECIMIENTOS

A nuestras familias y amigos, que nos dieron su apoyo y confianza en todo momento. Gracias de corazón por estar ahí durante la construcción de este proyecto y por creer en que si lo lograríamos.

A nuestros profesores y a nuestra asesora, quienes nos acompañaron durante toda nuestra carrera, fomentando nuestras bases profesionales y permitiendo que este proyecto fuera realizable.

A la Universidad Tecnológica de Pereira, por ser el espacio apropiado para madurar profesionalmente en nuestra carrera y por hacernos entender la importancia de estudiar en una universidad de tan alta calidad educativa.

DERECHOS DE AUTOR

Teniendo en cuenta lo estipulado en la sección 5.5.2 (Entidades de Educación) del Manual de Derecho de Autor de la Dirección Nacional del Derecho de Autor del Ministerio del Interior y de Justicia de Colombia [1], específicamente cuando declara que:

“Para el caso de los trabajos de grado, que se pueden considerar obras literarias o artísticas (tesis de grado, monografías, programas de computador, documentos que recogen una investigación, etc) se debe establecer quién es el titular de los derechos de autor.

La persona que realizó la obra literaria o artística en que consiste el trabajo de grado, le imprime su inteligencia, ingenio y talento creativo, y es su expresión la que aparece plasmada en la obra protegible. Por ello, si el autor es un estudiante, en cabeza de él se reconocerán los derechos patrimoniales y morales.”

Se establece que los derechos de autor sobre el presente proyecto de grado, pertenecen a: Andrés Giraldo Carvajal con Cédula de Ciudadanía N° 1088296665 y Carlos Alfredo Londoño Sánchez con Cédula de Ciudadanía N° 1088297623, integrantes del proyecto.

En consecuencia, estos son, a la luz de la legislación vigente en materia de derecho de autor, los titulares de todas las prerrogativas y facultades que la misma conceda.

TABLA DE CONTENIDO

LISTA DE TABLAS	10
LISTA DE FIGURAS	11
LISTA DE ANEXOS	13
RESUMEN.....	14
1. INTRODUCCIÓN	15
2. PLANTEAMIENTO DEL PROBLEMA	16
3. JUSTIFICACIÓN.....	17
4. OBJETIVOS.....	18
4.1. OBJETIVO GENERAL.....	18
4.2. OBJETIVOS ESPECÍFICOS.....	18
5. MARCO DE REFERENCIA.....	19
5.1. MARCO DE ANTECEDENTES.....	19
5.1.1. CamCardLite – Bussiness Card.....	19
5.1.2. Bussiness Card Book	20
5.2. MARCO TEÓRICO	22
6. DISEÑO METODOLÓGICO	28
6.1. TIPO DE INVESTIGACIÓN	28
6.2. POBLACIÓN.....	28
6.3. MUESTRA.....	28
6.4. HIPÓTESIS	28
6.5. VARIABLES	29
7. ANÁLISIS.....	30
7.1. LEVANTAMIENTO DE REQUERIMIENTOS	30
7.1.1. Visión general del producto	30
7.1.2. HISTORIAS DE USUARIO.....	32
7.1.2.1.1. Requerimientos funcionales	32
7.1.2.2. Requerimientos no funcionales	45
7.2. DIAGRAMA DE CASOS DE USO	48
7.2.1. Especificación de casos de uso	49
8. DISEÑO Y ARQUITECTURA DE SOFTWARE.....	66

8.1.	DIAGRAMA DE CLASES.....	66
8.2.	DIAGRAMA DE COMPONENTES.....	70
8.3.	DIAGRAMA DE ACTIVIDADES.....	72
8.4.	DIAGRAMA DE DESPLIEGUE.....	79
8.5.	DIAGRAMA DE BASE DE DATOS.....	80
9.	PRUEBAS DE SOFTWARE.....	84
9.1.	PLAN DE PRUEBAS.....	84
9.1.1.	Introducción.....	84
9.1.2.	Alcance.....	84
9.1.3.	Requerimientos de las pruebas.....	85
9.1.4.	Estrategia de prueba.....	85
9.1.5.	Tipos de pruebas y técnicas.....	86
9.1.5.1.	Pruebas de integridad de la base de datos y de los datos.....	86
9.1.5.2.	Pruebas de funcionalidad.....	87
9.1.5.3.	Pruebas de interfaz de usuario.....	88
9.2.	MODELO DE PRUEBAS.....	89
10.	DESARROLLO DEL APLICATIVO.....	91
10.1.	LÍMITES Y PROYECCIONES RESPECTO AL DESARROLLO.....	94
10.2.	ESTRUCTURA DE CODIFICACIÓN.....	95
10.2.1.	Android.....	95
10.2.2.	Api WEB.....	96
10.3.	ESPECIFICACION DE ACTIVIDADES, CLASES Y FUNCIONALIDADES DEL APLICATIVO ANDROID.....	97
10.4.	ESPECIFICACIÓN DE CLASES Y ADMINISTRACIÓN DE LA BASE DE DATOS EN LA API WEB.....	105
10.5.	IMÁGENES DE LA APLICACIÓN EN FUNCIONAMIENTO.....	111
11.	PRUEBAS DE VALIDACION DE HIPOTESIS.....	113
11.1.	MEDICIONES.....	113
12.	RESULTADOS.....	120
12.1.	RESULTADOS - ACTUALIZACIÓN DE TARJETAS.....	121
12.2.	RESULTADOS - CREACIÓN DE TARJETAS.....	122
12.3.	RESULTADOS – BÚSQUEDA DE TARJETAS.....	125

12.4.	RESULTADOS – INTERCAMBIO DE TARJETAS.....	128
12.5.	VALIDACIÓN DE LA HIPÓTESIS	131
13.	PLATAFORMA WEB CARDMAILBOX (APORTE ADICIONAL).....	132
13.1.	MARCO TEÓRICO	132
13.2.	ANÁLISIS.....	133
13.2.1.	Requerimientos del sistema.....	133
13.2.2.	Diagrama y especificaciones de casos de uso.....	134
13.3.	DISEÑO Y ARQUITECTURA	135
13.3.1.	Diagrama de clases	135
13.3.2.	Diagrama de despliegue.....	138
13.3.3.	Diagrama de actividades.....	139
13.4.	DESARROLLO PLATAFORMA WEB	140
13.4.1.	Estructura de codificación	140
13.4.2.	Definición de clases y funcionalidades	142
13.5.	IMÁGENES DE LA PLATAFORMA WEB EN FUNCIONAMIENTO	147
14.	CONCLUSIONES	150
15.	MEJORAS FUTURAS.....	152
16.	BIBLIOGRAFÍA.....	153

LISTA DE TABLAS

Tabla 1: Comparación – Aplicaciones relacionadas con el proyecto	19
Tabla 2: Sistemas operativos móviles y su uso	22
Tabla 3: Actores - Casos de Uso	49
Tabla 4: Especificación de caso de uso - Visualizar entorno inicial.....	50
Tabla 5: Especificación de caso de uso - Registrarse	51
Tabla 6: Especificación de caso de uso - Aceptar términos y condiciones de uso.....	52
Tabla 7: Especificación de caso de uso - Verificar datos ingresados	53
Tabla 8: Especificación de caso de uso - Iniciar sesión	54
Tabla 9: Especificación de caso de uso - Cerrar sesión	55
Tabla 10: Especificación de caso de uso - Recuperación de contraseña	56
Tabla 11: Especificación de caso de uso - Crear Tarjeta	57
Tabla 12: Especificación de caso de uso - Modificar Tarjeta	58
Tabla 13: Especificación de caso de uso - Visualizar Tarjetas Creadas	59
Tabla 14: Especificación de caso de uso - Modificar información personal	60
Tabla 15: Especificación de caso de uso - Visualizar Tarjetas Agregadas a la cuenta.....	61
Tabla 16: Especificación de caso de uso - Ordenar Tarjetas	62
Tabla 17: Especificación de caso de uso - Buscar Tarjetas en el repositorio global	63
Tabla 18: Especificación de caso de uso - Agregar tarjeta a cuenta de usuario	64
Tabla 19: Especificación de caso de uso - Buscar tarjetas en la cuenta de usuario.....	65
Tabla 20: Especificaciones - Diagrama de Clases	69
Tabla 21: Revisión histórica - Plan de Pruebas	84
Tabla 22: Pruebas de integridad	86
Tabla 23: Pruebas de interfaz de usuario	88
Tabla 24: Modelo de Pruebas	90
Tabla 25: Funcionalidades de los Layouts de la Aplicación	104
Tabla 26: Valores y funcionalidades de la variable "tag" - Api WEB.....	107
Tabla 27: Métodos, módulo de funciones - Api WEB	110
Tabla 28: Resultados - Encuesta, Pregunta 1	120
Tabla 29: Resultados - Encuestas a litografías	122
Tabla 30: Resultados - Creación de tarjetas desde la aplicación	123
Tabla 31: Resultados - Búsqueda de tarjetas en formato físico	125
Tabla 32: Resultados - Causa de no encontrar tarjeta.....	125
Tabla 33: Resultados - Búsqueda de tarjetas en la aplicación móvil	126
Tabla 34: Resultados - Costo de distribución de tarjetas	129
Tabla 35: Resultados - Costo de distribución por tarjeta.....	129
Tabla 36: Diagrama de Clases - Plataforma WEB, Main.....	136
Tabla 37: Diagrama de Clases - Plataforma WEB, UserLogin.....	137

LISTA DE FIGURAS

Ilustración 1: CamCardLite	20
Ilustración 2: Bussiness Card Book 1.....	20
Ilustración 3: Bussiness Card Book 2 Ilustración 4: Bussiness Card Book 3	21
Ilustración 5: Modelo 4+1 Vistas.....	25
Ilustración 6: Historia de Usuario – Registro de Usuarios	32
Ilustración 7: Historia de Usuario - Términos y Condiciones de Uso	33
Ilustración 8: Historia de Usuario - Recuperación de Contraseña	34
Ilustración 9: Historia de Usuario - Menú de Opciones Principal	35
Ilustración 10: Historia de Usuario - Modificación de información personal del usuario	36
Ilustración 11: Historia de Usuario - Plataforma para la creación de tarjetas 1.....	38
Ilustración 12: Historia de Usuario - Plataforma para la creación de tarjetas 2.....	38
Ilustración 13: Historia de Usuario - Almacenamiento de tarjetas creadas	39
Ilustración 14: Historia de Usuario - Modificación de tarjetas creadas.....	40
Ilustración 15: Historia de Usuario - Búsqueda de tarjetas dentro del repositorio global	41
Ilustración 16: Historia de Usuario - Agregar tarjetas del repositorio a la cuenta de usuario	42
Ilustración 17: Historia de Usuario - Visualización de tarjetas almacenadas por el usuario	43
Ilustración 18: Historia de Usuario - Búsqueda de tarjetas agregadas en la cuenta de usuario	44
Ilustración 19: Historia de Usuario - Pantalla Inicial	45
Ilustración 20: Historia de Usuario - Login.....	46
Ilustración 21: Diagrama de Casos de Uso.....	48
Ilustración 22: Diagrama de Clases (Parte 1)	66
Ilustración 23: Diagrama de Clases (Parte 2)	67
Ilustración 24: Diagrama de Componentes	70
Ilustración 25: Diagrama de Actividades - Autenticación	72
Ilustración 26: Diagrama de Actividades – Registrarse.....	72
Ilustración 27: Diagrama de Actividades - Recuperar contraseña.....	73
Ilustración 28: Diagrama de Actividades - Visualizar entorno inicial.....	73
Ilustración 29: Diagrama de Actividades - Editar tarjeta	74
Ilustración 30: Diagrama de Actividades - Modificar datos.....	75
Ilustración 31: Diagrama de Actividades – Crear tarjeta	76
Ilustración 32: Diagrama de Actividades - Visualizar tarjetas agregadas a la cuenta.....	77
Ilustración 33: Diagrama de Actividades - Buscar tarjetas en cuenta de usuario	77
Ilustración 34: Diagrama de Actividades - Buscar tarjetas en repositorio global.....	78
Ilustración 35: Diagrama de Despliegue	79
Ilustración 36: Diagrama de Base de datos.....	80
Ilustración 37: Interfaz phpMyAdmin	86
Ilustración 38: Pruebas de funcionalidad.....	87
Ilustración 39: Módulos clave de la aplicación	88
Ilustración 40: Bloques de desarrollo de la aplicación.....	91

Ilustración 41: Módulos - Api WEB	93
Ilustración 42: Módulos - Desarrollo Java, XML.....	95
Ilustración 43: Módulos - Api WEB	96
Ilustración 44: Paquetes - Desarrollo Java	97
Ilustración 45: activities, Desarrollo Java.....	98
Ilustración 46: listviews, Desarrollo Java	98
Ilustración 47: fragments, Desarrollo Java.....	99
Ilustración 48: navdrawer, Desarrollo Java.....	99
Ilustración 49: Colorpicker, Paquete.....	99
Ilustración 50: Colorpicker, Vista Previa	99
Ilustración 51: JSON Library, Desarrollo Java.....	100
Ilustración 52: Layouts - Desarrollo XML	101
Ilustración 53: Pantalla de Aplicación 1	111
Ilustración 54: Pantalla de Aplicación 2	111
Ilustración 55: Pantalla de Aplicación 3	111
Ilustración 56: Pantalla de Aplicación 4	111
Ilustración 57: Pantalla de Aplicación 5	112
Ilustración 58: Pantalla de Aplicación 6	112
Ilustración 59: Pantalla de Aplicación 7	112
Ilustración 60: Pantalla de Aplicación 8	112
Ilustración 61: Diagrama circular - Índice de éxito al momento de actualizar una tarjeta creada.	121
Ilustración 62: Diagrama de barras - Tiempo de creación de tarjetas.....	124
Ilustración 63: Diagrama de barras - Costo por tarjeta creada.....	124
Ilustración 64: Diagrama de barras - Comparación, tiempo de búsqueda de tarjetas.....	127
Ilustración 65: Diagrama de barras - Tiempo destinado al intercambio de tarjetas	130
Ilustración 66: Diagrama de barras - Costo de distribución por tarjeta	130
Ilustración 67: Diagrama de Clases - Plataforma WEB, Parte 1	135
Ilustración 68: Diagrama de Clases - Plataforma WEB, Parte 2	136
Ilustración 69: Diagrama de Despliegue - Plataforma WEB.....	138
Ilustración 70: Estructura de codificación - Plataforma WEB	140
Ilustración 71: MVC en la plataforma WEB.....	141
Ilustración 72: Modelos MVC - Especificación de funcionalidades	143
Ilustración 73: Controladores MVC - Especificación de funcionalidades.....	146
Ilustración 74: Pantalla, Plataforma WEB #1	147
Ilustración 75: Pantalla, Plataforma WEB #2	147
Ilustración 76: Pantalla, Plataforma WEB #3	148
Ilustración 77: Pantalla, Plataforma WEB #4	148
Ilustración 78: Pantalla, Plataforma WEB #5	149
Ilustración 79: Pantalla, Plataforma WEB #6	149

LISTA DE ANEXOS

ANEXO A: Lista detallada de respuestas – Encuesta de Usabilidad.....	146
ANEXO B: Resumen de respuestas – Encuesta de Usabilidad.....	146

RESUMEN

El presente proyecto nace con la idea de brindar una solución eficaz, haciendo uso de herramientas tecnológicas, a un problema de tipo social y ambiental como lo es la administración ineficiente y el complejo proceso de actualización de tarjetas de presentación personal y empresarial.

El propósito de este proyecto es desarrollar un aplicativo móvil que permita crear y compartir tarjetas de presentación personal y empresarial depositadas en la web con el fin de mejorar la manera en cómo las personas, actualizan y administran eficientemente las mismas. La aplicación desarrollada le permitirá además al usuario disminuir el tiempo de búsqueda de una tarjeta en específico y el tiempo que toma este en hacerlas llegar a otros usuarios.

Se realizó un esfuerzo adicional que no estaba especificado en los requerimientos iniciales del proyecto para que el aplicativo pudiese también trabajar dentro de un entorno web, permitiendo así que un usuario pudiera disponer de todas las funcionalidades que tiene el aplicativo móvil desde internet.

Por otro lado, se adicionaron algunas mejoras respecto al requerimiento que especifica la creación de tarjetas, permitiendo al usuario además de visualizar la tarjeta creada con los campos y las imágenes que selecciona previamente, seleccionar para cada uno de estos elementos en la tarjeta un tamaño de letra, un color, un tipo de letra y una posición dentro de la misma.

Con todo esto el usuario podrá hacer uso de una aplicación muy completa en la que de forma práctica, le es posible crear, almacenar y compartir tarjetas de presentación personal y empresarial a muy bajo costo.

1. INTRODUCCIÓN

La empresa AVERY [1] expone: “En el ambiente actual de trabajo, las cosas cambian rápidamente. Nuevos empleados llegan y los actuales son transferidos o promovidos. Los asuntos y contactos de trabajo van cambiando. Nuevas promociones, eventos y citas se programan. ¿Alguna vez has pensado en que tus tarjetas se fueran actualizando con estos cambios?”. Con esta pregunta, la empresa resalta la necesidad de que las tarjetas de presentación cuenten con una información actualizada y susceptible a cambios futuros.

Si bien el costo de fabricación de tarjetas personales y empresariales ha bajado durante el transcurso de los últimos años [2], todavía se debe asumir un costo variable de diseño y fabricación, que en ocasiones, puede ser alto dependiendo del espacio de impresión en la tarjeta, del tipo de material y la técnica utilizada para la impresión de las mismas [3]. Así también, el tiempo de fabricación puede variar dependiendo de los aspectos mencionados anteriormente. Como se muestra en [3], el tiempo de fabricación de las tarjetas y la llegada de éstas al cliente, toma un total de 3 días, sin contar el tiempo de diseño que debe asumir algunas veces el mismo comprador.

Por otro lado, el celular o móvil ha revolucionado el mundo de forma sustancial desde su primera aparición en el mercado y aún hoy lo sigue haciendo ayudando en la tarea de conectar a miles de millones de personas en todo el mundo y convirtiéndose en un elemento indispensable de la sociedad actual. Con aproximadamente 7 mil millones de líneas de teléfonos celulares en todo el mundo, un número que igualaría la cifra de seres humanos vivos según un reporte de BBC Mundo [4], los dispositivos móviles son también una oportunidad enorme para que una solución de cualquier tipo adaptada a este entorno pueda replicarse y beneficiar a tantos.

El presente proyecto busca aprovechar el grado de expansión y aceptación de los dispositivos móviles, especialmente los Smartphones¹ y la facilidad que se tiene hoy en día de acceder económicamente a estos dispositivos para dar solución a las dificultades expuestas anteriormente al momento de crear y compartir tarjetas de presentación. Esto, desarrollando un aplicativo móvil que mejore la eficiencia en la administración de las tarjetas y la manera en cómo se actualiza la información contenida en las mismas.

Como un elemento adicional dentro del proyecto y con miras a llegar a más usuarios beneficiarios, la aplicación fue desarrollada también en un entorno web, permitiendo al usuario hacer uso de todas las funcionalidades que tiene la aplicación móvil.

¹ Smartphone o teléfono inteligente en español, es un dispositivo móvil con una mayor capacidad de almacenar datos y realizar actividades y con una mayor conectividad que un teléfono móvil convencional.

2. PLANTEAMIENTO DEL PROBLEMA

En la actualidad se presenta un intercambio constante de tarjetas personales y empresariales donde se refleja la necesidad de transmitir una información determinada a una población en concreto, tal como lo da a entender el Web Master de la empresa Globbereds [5] cuando enuncia que “las tarjetas personales, junto a las de carácter profesional, puede que sea una de las formas más comunes de relación social en nuestros días”. Este proceso presenta algunos obstáculos que generan un impacto negativo sobre las personas implicadas en el proceso:

- La imposibilidad de actualizar tarjetas personales y empresariales
- El consumo innecesario de tiempo para encontrar una tarjeta en específico.
- Los sistemas de impresión más tradicionales, mediante planchas, como pueden ser los sistemas de impresión offset, ofrecen unas calidades muy altas pero a precios prohibitivos [5]
- El esfuerzo significativo para hacer llegar estas tarjetas a las demás personas
- El impacto ambiental negativo que se presenta por el excesivo uso de papel en la fabricación de estas tarjetas y el desperdicio que se genera por desecharlas cuando no se requiere su uso.

Cada una de estas variables obstaculiza en mayor o menor medida el proceso de creación, intercambio, búsqueda y actualización de tarjetas de presentación personal o empresarial. La pregunta es ¿Cómo hacer frente a esas variables para facilitar cada uno de los anteriores procesos? o dicho de otra manera ¿Cómo es posible actualizar y administrar eficientemente tarjetas de presentación?

3. JUSTIFICACIÓN

La problemática que se menciona anteriormente no se percibe de forma clara en la comunidad, pero en realidad puede afectar considerablemente nuestro ambiente social, teniendo en cuenta lo expuesto por Renata Roa [6], consultora en Imagen Pública y Comunicación Facial, al decir que “la tarjeta de presentación es el pedazo de papel más importante para el profesionalista y emprendedor. Éste se convierte en la primera impresión que la otra persona tendrá de ti; es como la extensión de tu persona y personalidad, es el constante recordatorio de quién eres”.

Si este problema no es resuelto, las personas interesadas en ofrecer sus productos o servicios a través de una tarjeta de presentación tendrán que continuar acarreado con costos y tiempos de diseño, fabricación y distribución a veces demasiado altos, además de la imposibilidad de actualizar las tarjetas al momento de presentarse algún cambio en la información que contienen.

Por otra parte es más engorroso para una persona administrar sus tarjetas de presentación de la manera convencional, esto debido a lo siguiente:

- La búsqueda de una tarjeta en específico puede volverse complicada cuando la persona tiene que buscar dentro de un cúmulo más grande de estas, esto sin pensar en que no todas las personas tienen en un mismo lugar, ordenadas todas sus tarjetas y sin considerar el hecho de que pudo haber perdido o extraviado la tarjeta que busca.
- Puede pasar fácilmente que la información que se encuentra especificada en una tarjeta haya cambiado, lo que dificulta significativamente que la persona pueda acceder al servicio o producto que requiere.
- Si la persona cuenta con un número significativo de tarjetas de presentación, lo más probable es que no tenga cómo o no desee llevar consigo por razones de comodidad todas sus tarjetas, es decir, no tiene siempre a la mano la información de contacto de cualquiera de sus tarjetas si llegase a necesitarla.

La aplicación desarrollada en el presente proyecto hace frente a las dificultades que se mencionan anteriormente facilitando así la vida de las personas que hacen uso de esta. Una de las mayores motivaciones para la ejecución del proyecto, es sin duda tener la oportunidad de, por medio de ayudas tecnológicas, solventar un problema de tipo social y ambiental como el que aquí se aborda.

4. OBJETIVOS

4.1. OBJETIVO GENERAL

Desarrollar un aplicativo móvil que permita reducir costos de fabricación y distribución, optimizar la administración, y además, disminuir el tiempo de búsqueda de tarjetas de presentación.

4.2. OBJETIVOS ESPECÍFICOS

- Definir los requerimientos del sistema para conocer las funcionalidades que van a ser incluidas en el aplicativo a través de historias de usuario.
- Elaborar el diseño y la arquitectura del aplicativo. Garantizar atributos de calidad referentes a correctitud, confiabilidad, escalabilidad y usabilidad.
- Desarrollar el aplicativo móvil cumpliendo con los requerimientos, diseño y arquitectura establecidos al inicio del proyecto.
- Realizar las pruebas correspondientes para verificar los resultados entregados por el software.

5. MARCO DE REFERENCIA

5.1. MARCO DE ANTECEDENTES

A continuación se describen en un cuadro comparativo algunas aplicaciones que tienen relación con el presente proyecto, identificando en cada una de ellas las características más sobresalientes:

CARACTERÍSTICAS	CamCardLite [6]	Business Card Book [7]
Creación y diseño de tarjetas	x	✓
Sincronización de dispositivos	✓	x
Escaneo de tarjetas	✓	x
Compartir a través de redes sociales	x	✓
Organización de tarjetas	✓	✓
Búsqueda de tarjetas locales	✓	✓
Filtros de búsqueda	x	x
Actualización de los datos	x	x
Capacidad de encontrar tarjetas creadas por otras personas	x	x

Tabla 1: Comparación – Aplicaciones relacionadas con el proyecto

5.1.1. CamCardLite – Bussiness Card

Tal como se referencia en la página web de la Play Store [7], CamCardLite es un lector profesional de tarjetas comerciales. Puede capturar imágenes de la tarjeta comercial con la cámara del teléfono, reconocer el contenido de la imagen de tarjeta, y el resultado reconocido esta automáticamente organizado como un contacto y se guarda en el libro de dirección de teléfono y titular de la tarjeta.

La aplicación administra y organiza tarjetas comerciales escaneadas, y además sincroniza las tarjetas comerciales con otros dispositivos:



Ilustración 1: CamCardLite

5.1.2. Business Card Book

Como se muestra en la página web de la Play Store [8], Business Card Book es una innovadora aplicación móvil para crear, almacenar y compartir tarjetas de visita virtuales. Es también un moderno centro de comunicación, reuniendo toda la información de contactos y redes sociales en un solo lugar.

La aplicación permite compartir las tarjetas diseñadas a través de redes sociales:

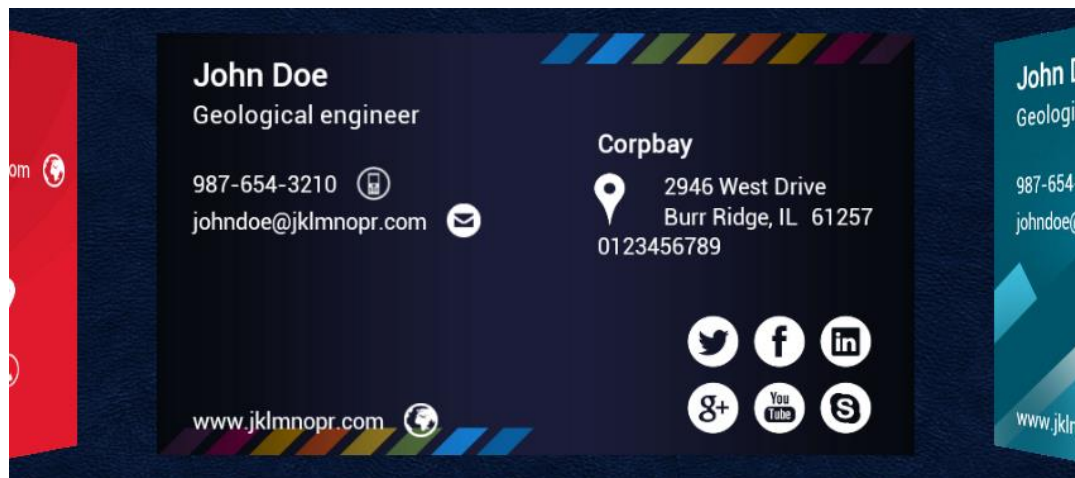


Ilustración 2: Business Card Book 1

La aplicación posee un editor dedicado a la creación de tarjetas basado en plantillas predefinidas por el programa:

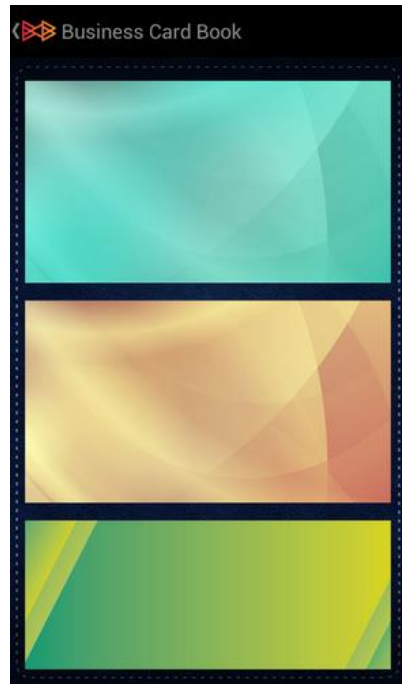


Ilustración 3: Bussiness Card Book 2



Ilustración 4: Bussiness Card Book 3

Las aplicaciones presentadas anteriormente abordan de cierta manera la problemática expuesta en el presente proyecto, cada una de ellas otorga un conjunto de funcionalidades que facilita a las personas la administración de sus tarjetas de presentación, sin embargo no brindan solución a problemas tan esenciales como la actualización de la información contenida en la tarjeta y la búsqueda de tarjetas creadas por otros usuarios para luego ser almacenadas a su cuenta personal.

5.2. MARCO TEÓRICO

Tarjetas personales y empresariales

Tal como lo expresa Marissa R. [9], las tarjetas de presentación son una representación de los datos más importantes de una empresa, de una persona o negocio. En esta misma se pretende lograr llamar la atención de la persona a la que va dirigida. Es una forma en la que las personas pueden conservar con facilidad información de contacto si la tarjeta tiene la información necesaria de una manera breve y con colores que llamen la atención. Los elementos básicos para una tarjeta personal o empresarial pueden ser:

- Nombre de la persona, de la empresa u organización a la que pertenece.
- Dirección de la oficina en que se labora.
- Número de Teléfono de oficina y celular.
- Dirección de correo electrónico.
- Página Web de la empresa y/o persona.
- Título o cargo de la Persona y logotipo.
- Imágenes y elementos decorativos para poder llamar la atención de manera visual.
- Slogan, si es que se cuenta con alguno, puede ser incluido en el mismo logotipo.

Smartphone

Como lo menciona Arletys González Madera [10], un teléfono inteligente es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades semejantes a una mini computadora y conectividad que un teléfono móvil convencional. El término «inteligente» hace referencia a la capacidad de usarse como un ordenador de bolsillo, llegando incluso a remplazar a un ordenador personal en algunos casos.

Según datos suministrados por Gartner Inc. En su sitio web [11], los sistemas operativos móviles más utilizados por teléfonos inteligentes son:

SISTEMA OPERATIVO	USO (%)
Android (Google)	81.9
iOS (Apple)	12.1
Windows Phone (Microsoft)	3.6
RIM (BlackBerry)	1.8
Symbian (Nokia)	0.2

Tabla 2: Sistemas operativos móviles y su uso

Android

Android es un sistema operativo basado en Linux, diseñado principalmente para móviles con pantalla táctil como teléfonos inteligentes o tabletas inicialmente desarrollados por Android Inc., que Google respaldó económicamente y más tarde compró en 2005.

Por otro lado, de acuerdo a lo referenciado en la página principal de Android [11], este sistema operativo es una pila de software de código abierto creado para una amplia gama de dispositivos con diferentes características en su forma. Los propósitos principales de Android son la creación de una plataforma de software abierto disponible para los fabricantes de equipos originales y desarrolladores para hacer que sus ideas innovadoras se conviertan en realidad y para introducir un producto de éxito en el mundo real.

Mik Mann expresa en su libro “Introducción a Android” [12], que la estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2.8 millones de líneas de lenguaje C, 2.1 millones de líneas de Java y 1.75 millones de líneas de C++.

Las características más relevantes del sistema operativo Android son:

- Adaptación al diseño de dispositivo
- Almacenamiento
- Mensajería
- Navegador web, soporte de Java
- Soporte multimedia y streaming
- Entorno de desarrollo
- Google Play
- Multi-táctil
- Bluetooth
- Video llamada y características basadas en voz
- Multitarea
- Tethering

Desarrollo de una aplicación para Android

Jesús Tomás Gironés, Docente de la Universidad Politécnica de Valencia, España, explica en su libro “El gran libro de Android” [13], que el desarrollo de una aplicación móvil, específicamente en sistemas operativos Android, requiere una serie de pasos importantes con el fin de construir de manera correcta el aplicativo. A continuación se listan algunos de los pasos más importantes que se deben desarrollar:

- La visión general y el entorno de desarrollo
- El diseño de la interfaz de usuario: Vistas y Layouts
- Las actividades e intenciones
- Gráficos
- Entradas: teclado, pantalla táctil y sensores
- Multimedia y Ciclo de Vida
- Seguridad y posicionamiento
- Servicios, notificaciones y receptores de anuncios
- Almacenamiento de datos
- Internet: sockets, HTTP y servicios web

Administración eficiente de tarjetas personales y empresariales

Para este proyecto la administración eficiente, se refiere a la disminución de tiempos y costos de fabricación, distribución y búsqueda de tarjetas de presentación.

UML

En una definición simple, el Lenguaje de Modelado Unificado (UML, del inglés Unified Modeling Language) es una familia de notaciones gráficas que ayudan a describir sistemas de software. UML [14] es la especificación más usada por OMG² (Object Management Group) y es la forma en que se estructura una aplicación, su comportamiento y su arquitectura así como los procesos de negocio y la estructura de los datos.

En el estándar UML 2.0 [15] se habla de un grupo de diagramas que son los más comunes y habituales y se realiza una clasificación de éstos dividiéndolos en Estructurales y de Comportamiento.

² OMG (Object Management Group) es un consorcio conformado para establecer estándares que soporten la interoperabilidad de sistemas orientados a objetos.

Los diagramas estructurales se usan para definir la arquitectura estática. Comprende construcciones tales como clases, objetos y componentes, y las relaciones entre éstos elementos. En este tipo de diagramas se encuentran: Diagrama de Clases, Diagrama de Componentes, Diagrama de Estructura Compuesta, Diagrama de Paquetes, Diagrama de Despliegue y Diagrama de Objetos.

Los diagramas de comportamiento se usan para representar la parte dinámica. Comprende construcciones de comportamiento tales como actividades, estados, líneas de tiempo y los mensajes que circulan entre diferentes objetos. Estos diagramas se utilizan para representar las interacciones entre los diversos elementos del modelo y los estados instantáneos a lo largo de un período de tiempo. En este tipo de diagramas se encuentran: Diagrama de Actividad, Diagrama de Casos de Uso, Diagrama de Estados, Diagrama de Secuencia, Diagrama de Comunicación, Diagrama de Interacción global y Diagrama de Tiempos.

Modelo 4 + 1 Vistas

Este modelo define 4 vistas principales y una vista “+1” que tiene la función de relacionar las otras 4, tal como se muestra en la siguiente figura:

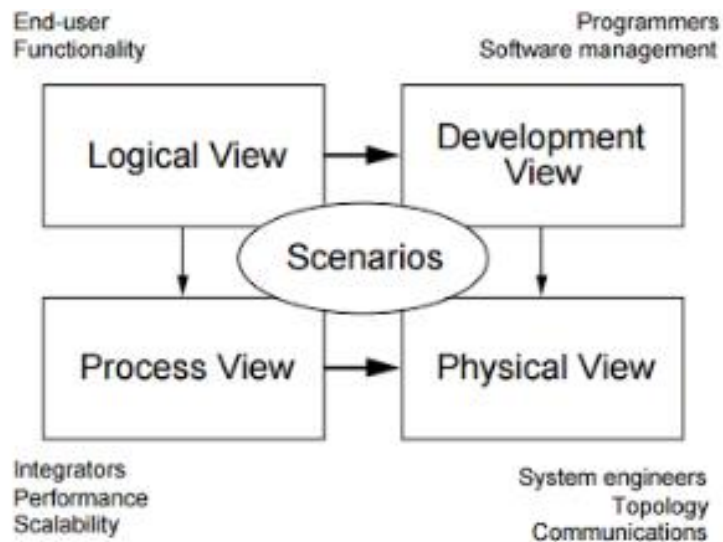


Ilustración 5: Modelo 4+1 Vistas

El modelo 4+1 vistas diseñado por Philippe Kruchten³ es un paradigma ampliamente aceptado a la hora de establecer las vistas necesarias para describir una arquitectura de software [16]. El objetivo de este modelo es mostrar, en cada una de las vistas, una perspectiva o visión de un conjunto de elementos del proyecto y sus relaciones.

En la vista lógica se tienen en cuenta los requerimientos funcionales que el sistema proporcionará a los usuarios, es decir lo que el sistema debe hacer y los servicios que ofrece. Para completar la documentación de esta vista se incluirá en este proyecto un diagrama de clases.

En la vista de despliegue o vista de desarrollo se muestra principalmente cómo está dividido el sistema de software en componentes y se muestran las dependencias e interacciones de estos componentes. Para completar la documentación de esta vista se incluirá en este proyecto un diagrama de componentes.

En la vista de procesos se muestran los procesos que se llevan a cabo en el sistema y la forma en que se comunican entre sí. Especifica las líneas de mando que ejecutan cada operación en cada una de las clases señaladas en la vista lógica. Para completar la documentación de esta vista se incluirá en este proyecto un diagrama de actividad.

En la vista física se representan los componentes físicos que conforman el sistema (incluyendo servicios) y las conexiones existentes entre ellos. Los elementos definidos en la vista lógica se mapean a componentes de software o de hardware. Para completar la documentación de esta vista se incluirá en este proyecto un diagrama de despliegue.

En la vista +1 o vista de escenarios se representan los casos de uso, que tiene como función unir las 4 anteriores vistas y relacionarlas, permitiendo una trazabilidad de componentes, clases, equipo, etc. Para completar la documentación de esta vista se incluirá en este proyecto un diagrama de casos de uso.

PHP: [17] (Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies.

³ Philippe Kruchten es un Ingeniero de Sistemas canadiense y profesor de la Universidad British Columbia conocido por ser el desarrollador del modelo 4+1 vistas y director responsable de RUP (Rational Unified Process).

Java: [18] Es un lenguaje de programación de alto nivel que se caracteriza por ser:

- Simple
- Orientado a objetos
- Distribuido
- Multiprocesos
- Dinámico
- Arquitectura neutral
- Portable
- Alto rendimiento
- Robusto
- Seguro

XML: [19] (eXtensible Markup Language). XML fue diseñado para describir datos. XML es una herramienta independiente de software y de hardware para llevar información.

JSON: [20] Es un formato de intercambio de datos ligero. Es fácil de leer y escribir para los seres humanos. Es fácil para las máquinas a la hora de analizar y generar. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza las convenciones que son familiares para los programadores de la familia de lenguajes C, incluyendo C, C ++, C #, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen de JSON un lenguaje ideal para el intercambio de datos.

Servidor: [21] Es la maquina o computador donde se almacena su página web. Toda la información publicada en cada sitio web se almacena en un espacio destinado para este fin. De lo contrario no habría forma de divulgar el contenido.

6. DISEÑO METODOLÓGICO

6.1. TIPO DE INVESTIGACIÓN

El tipo de investigación del presente proyecto es cuantitativa puesto que plantea una hipótesis que será verificada mediante pruebas, además se recolectará información y se medirán variables haciendo uso de instrumentos bien estructurados.

6.2. POBLACIÓN

La población que se tendrá en cuenta para el presente proyecto son todas aquellas personas que disponen de un dispositivo móvil con sistema operativo Android.

6.3. MUESTRA

Para el cálculo del tamaño de la muestra del presente proyecto se tendrá en cuenta la fórmula descrita en el libro “Ratios Financieros y Matemáticas de la Mercadotecnia” [22], que corresponde al cálculo de una muestra en una población infinita.

$$n = \frac{Z^2 * p * q}{E^2}$$

- $Z\alpha$ (coeficiente de confianza, para una seguridad del 90%)= 1.65
- p (proporción esperada, un 5% para maximizar el tamaño muestral) = 0.05
- $q = (1 - p) = 1 - 0.05 = 0.95$
- E (precisión, se usará un 5% de precisión) = 0.05

Hacemos el cálculo respectivo:

$$n = \frac{(1.65)^2(0.05)(0.95)}{(0.05)^2} = \frac{0.1293}{0.0025} = 51$$

Significa que para la población y las variables que se tuvieron en cuenta en el análisis, se tendría que estudiar a 51 personas.

6.4. HIPÓTESIS

El proyecto que se va a desarrollar permitirá actualizar y administrar eficientemente tarjetas personales y empresariales.

6.5. VARIABLES

Para el presente proyecto se tendrá en cuenta las siguientes variables de estudio:

Variables de tiempo:

- TC (Tiempo de creación): Es el tiempo que se tarda un usuario interesado en crear una tarjeta, desde el momento en que decide hacerla (o mandarla a hacer) hasta el momento en que está lista para ser distribuida.
- TB (Tiempo de búsqueda): Es el tiempo que se tarda un usuario interesado en encontrar una tarjeta en específico desde el momento en que decide buscarla hasta el momento en encontrarla.
- TD (Tiempo de distribución): Es el tiempo que se tarda un usuario interesado en compartir una tarjeta, desde el momento en que decide compartirla y hasta el momento en que llega a mano de otro usuario.

Variables de costos:

- CF (Costo de fabricación): Es el costo de fabricar una tarjeta de presentación.
- CD (Costo de distribución): Es el costo de distribuir una tarjeta de presentación

7. ANÁLISIS

7.1. LEVANTAMIENTO DE REQUERIMIENTOS

De acuerdo con Ralph Young [23], un requerimiento es un atributo necesario para el sistema a desarrollar, en el que se puede describir una funcionalidad o característica que tenga valor para los stakeholders (usuarios interesados) dentro del mismo. Para el presente proyecto se identificaron las siguientes necesidades o requisitos clasificados como requerimientos funcionales y no funcionales.

7.1.1. Visión general del producto

Requerimientos funcionales

Este tipo de requerimientos especifican funcionalidades o servicios que la aplicación debe proporcionar. Para el presente proyecto se definieron los siguientes requisitos funcionales:

- Que la aplicación permita a cada usuario crear una cuenta para administrar su propia información (Obligatorio crear cuenta).
- Que la aplicación permita aceptar “Términos de uso de la aplicación” al momento de crear una cuenta.
- Que la aplicación tenga la capacidad de recuperar la contraseña cuando sea olvidada.
- Que la aplicación permita al usuario modificar sus datos personales asociados a su cuenta (correo, contraseña).
- Que la aplicación posea un menú de opciones para elegir entre crear, buscar o administrar tarjetas agregadas del repositorio global.
- Que la aplicación permita crear tarjetas personales y empresariales a través de plantillas o imágenes prediseñadas subidas por el usuario.
- Que la aplicación permita almacenar las tarjetas creadas por el usuario.
- Que la aplicación permita al usuario modificar sus propias tarjetas (No tiene permisos para modificar tarjetas creadas por otros usuarios)
- Que la aplicación permita buscar tarjetas creadas por otros usuarios dentro del repositorio global, utilizando filtros de búsqueda.
- Que la aplicación permita agregar a mi cuenta tarjetas encontradas en el repositorio global.
- Que la aplicación permita listar las tarjetas agregadas por el usuario a su cuenta.
- Que la aplicación permita realizar una búsqueda local de tarjetas agregadas en la cuenta del usuario.
- Que la aplicación permita cerrar mi sesión de usuario
- Que la aplicación posea una pantalla principal para elegir entre “Iniciar Sesión” o “Registrarse”.

Requerimientos no funcionales

Un requerimiento no funcional especifica restricciones sobre los servicios y funcionalidades ofrecidos por el sistema. Para el presente proyecto se definieron los siguientes requisitos no funcionales:

- Que la aplicación cuente con un sistema de autenticación para el ingreso a las funcionalidades del sistema
- Que la aplicación sea compatible con versiones de Android superiores a 4.0 Android Ice Cream Sandwich.
- La aplicación debe correr con requisitos mínimos de hardware 512 RAM, Procesador 1Ghz de un núcleo y pantalla de mínimo 4.0”.
- La información almacenada por los usuarios debe estar hospedada en un servidor con un mínimo de 3 GB de capacidad.
- La base de datos debe correr sobre el motor de base de datos MySQL.
- El servidor debe soportar la conexión simultánea mínima de 3000 usuarios.

7.1.2. HISTORIAS DE USUARIO

7.1.2.1.1. Requerimientos funcionales

Número: 1

Nombre: Registro de usuarios

Tipo: Funcional

Descripción: Como cliente requiero que la aplicación permita a los usuarios crear una cuenta para administrar su información.

- La aplicación debe poseer una opción llamada “Registrarme” en la pantalla principal, que esté ubicado de tal manera que sea visible y de fácil acceso.
- En la opción registrarme deben estar los siguientes campos de entrada:
 - Nombre completo
 - Correo
 - Confirmación de correo
 - Contraseña
 - Confirmación de contraseña
- Además, se debe ingresar los siguientes campos para una posible recuperación de contraseña: Pregunta secreta, Respuesta secreta
- La aplicación debe poseer un botón para proseguir con el registro y realizar una comprobación de los datos ingresados. Si todo es correcto, se debe registrar el usuario en la base de datos.

REGISTRO

Nombre Completo

Correo

Confirmación de Correo

Contraseña

Confirmación de Contraseña

Pregunta para recuperación de contraseña

Pregunta secreta

Respuesta

Aceptar Términos de Uso

REGISTRARME

Ilustración 6: Historia de Usuario – Registro de Usuarios

Número: 2

Nombre: Términos y condiciones de uso.

Tipo: Funcional

Descripción: Como cliente me gustaría que la aplicación cuente con unos términos y condiciones de uso legibles y claramente especificados; además que me permita poder aceptarlos o rechazarlos de acuerdo a mi voluntad a la hora de crear una cuenta.

Observaciones:

- Las condiciones y términos de uso de la aplicación pueden ser modificadas sin previo aviso y por ende deben ser revisadas eventualmente.
- Las condiciones y términos de la aplicación deben incluir entre sus incisos el aseguramiento y la protección de los datos personales del cliente.
- Las condiciones y términos de la aplicación deben desplegarse en orden de importancia de tal manera que en la parte superior se incluyan aquellos con mayor relevancia.
- Las condiciones y términos de la aplicación deben categorizarse y organizarse garantizando una mejor lectura de los mismos.
- Las condiciones y términos de la aplicación deben mantenerse actualizados en la medida que cambien las políticas o procedimientos.

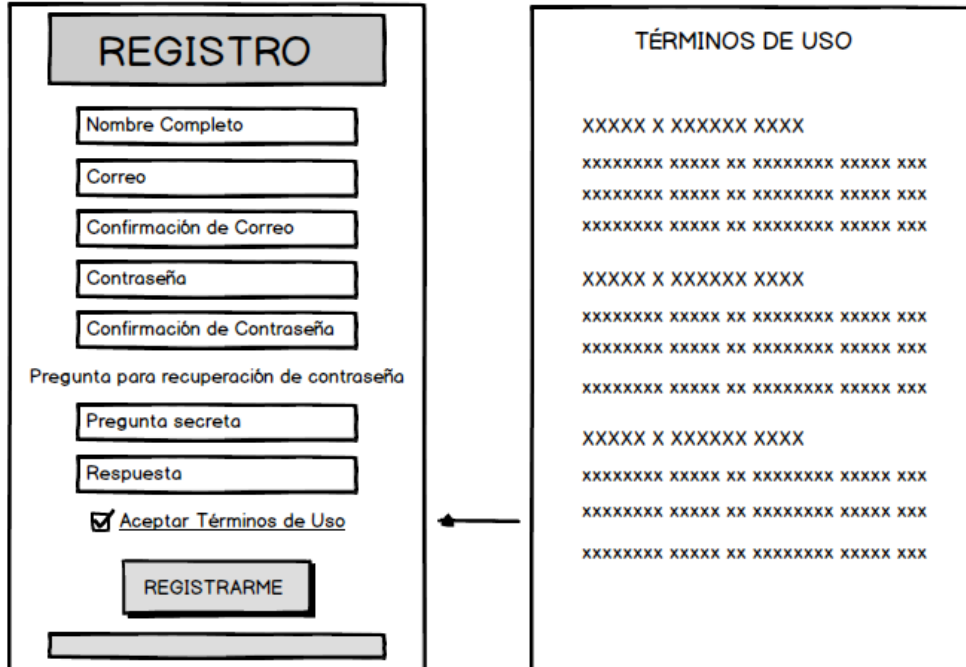


Ilustración 7: Historia de Usuario - Términos y Condiciones de Uso

Número: 3

Nombre: Recuperación de contraseña

Tipo: Funcional

Descripción: Como cliente me gustaría que la aplicación tuviese manera de recuperar mi contraseña cuando la haya olvidado.

- La recuperación de contraseña debe hacerse de manera fácil, rápida y segura.
- La recuperación de contraseña debe situarse en la misma pantalla del login de acceso.
- Para la recuperación de contraseña, se solicitará al usuario que ingrese el correo electrónico y de clic en recuperar contraseña. La aplicación le preguntará por la respuesta a la pregunta secreta ingresada en el registro y de ser correcta la información, el sistema le pide ingresar una nueva contraseña y confirmarla.

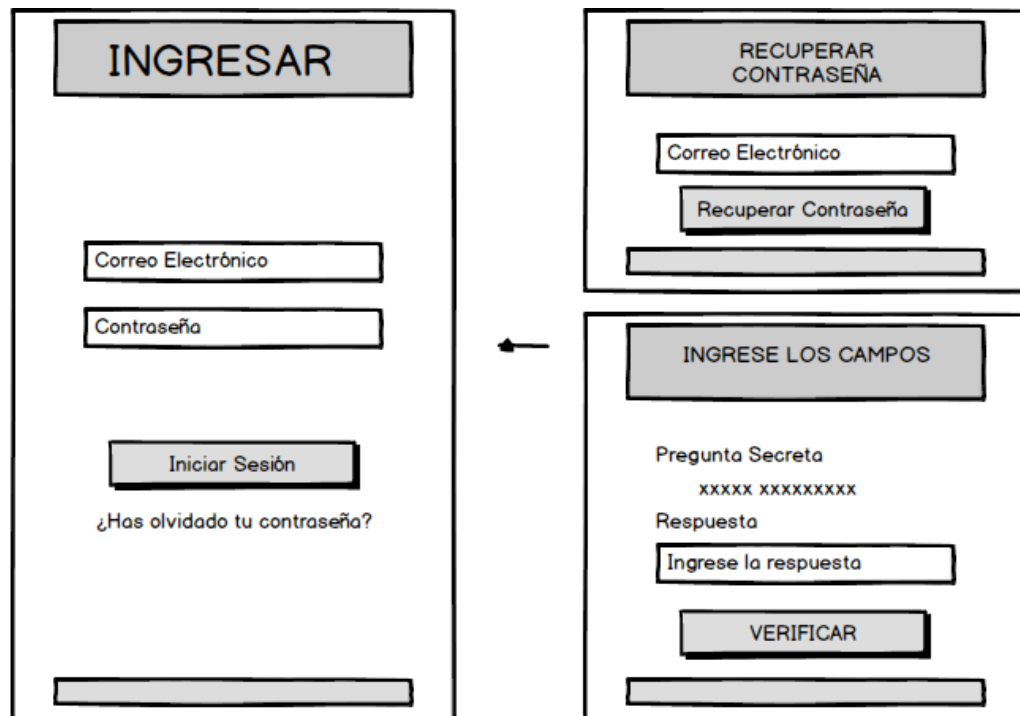


Ilustración 8: Historia de Usuario - Recuperación de Contraseña

Número: 4

Nombre: Menú de opciones principal

Tipo: Funcional

Descripción: Como cliente me gustaría que la aplicación me permitiera elegir entre las funcionalidades de la aplicación desde un menú principal de opciones centralizado una vez me encuentre en mi cuenta de usuario.

- Este menú de opciones es el principal, por lo cual puede accederse desde cualquier pantalla de la aplicación a través de una barra deslizable desde el borde izquierdo.
- El menú deberá presentarse con un estilo adecuado en forma de lista con la posibilidad de acceso con un solo toque.
- Las opciones que deben aparecer en el menú son:
 - Mis Tarjetas
 - Creación de tarjetas
 - Tarjetas Creadas
 - Buscar Tarjetas
 - Modificar Datos Personales
 - Salir

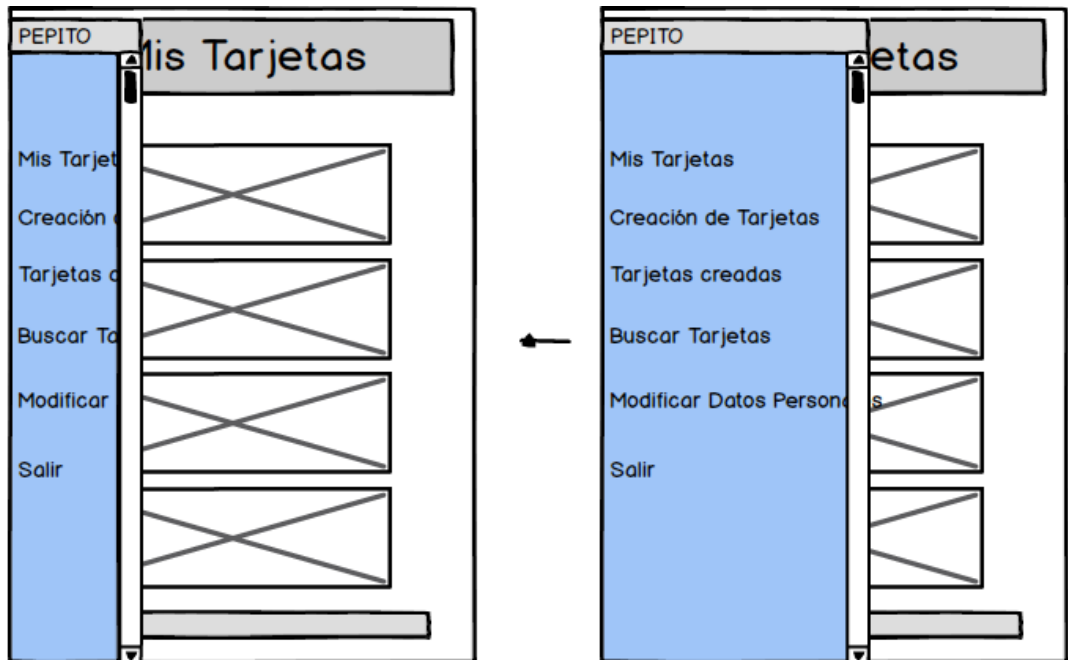


Ilustración 9: Historia de Usuario - Menú de Opciones Principal

Número: 5

Nombre: Modificación de información personal del usuario

Tipo: Funcional

Descripción: Como cliente requiero que la aplicación permita a los usuarios modificar sus datos personales.

- Esta opción deberá estar ubicada en el menú principal, después de haber pasado por el proceso de login correctamente.
- Esta opción permitirá al usuario modificar:
 - Nombre de usuario
 - Contraseña

Observaciones:

- En el momento que se hagan estas modificaciones, el sistema debe dar aviso al usuario de que los cambios fueron realizados exitosamente y llevarlo a la pantalla inicial para que pueda iniciar sesión con su nueva información.

El diagrama muestra una interfaz de usuario con un título 'Modificar Mis Datos' en un recuadro gris. Debajo del título, hay dos opciones seleccionadas con casillas de verificación: 'Cambiar Nombre' y 'Cambiar Contraseña'. Cada opción tiene un campo de entrada de texto correspondiente. Para 'Cambiar Nombre', el campo está etiquetado 'Nuevo Nombre'. Para 'Cambiar Contraseña', hay tres campos: 'Contraseña Anterior', 'Contraseña Nueva' y 'Confirmar Contraseña Nueva'. En la parte inferior de la interfaz, hay un botón rectangular con el texto 'ACTUALIZAR DATOS' y un campo de entrada de texto adicional.

Ilustración 10: Historia de Usuario - Modificación de información personal del usuario

Número: 6

Nombre: Plataforma para la creación de tarjetas

Tipo: Funcional

Descripción: Como cliente requiero que la aplicación posea una plataforma dedicada a la creación de tarjetas de presentación.

- Deben existir campos de entrada para la información de la tarjeta que el usuario quiere crear. Estos campos no son obligatorios y el usuario debe tener la opción de elegir cuáles campos desea utilizar. Los campos de entrada son:
 - Logo (Imagen subida por el usuario)
 - Nombre
 - Empresa
 - Cargo
 - Página Web
 - Dirección de contacto
 - Teléfono(s) de contacto
 - Dirección de correo
 - Descripción
- Debe existir un campo en el que se pueda elegir si la tarjeta es pública o no al momento de crearse.
- La plataforma debe permitir al usuario elegir el fondo de la tarjeta haciendo uso de alguna de las siguientes opciones:
 - Plantillas prediseñadas almacenadas en la aplicación.
 - Imagen subida por el usuario.
- Cuando la tarjeta esté creada, el usuario podrá decidir si guardarla y regresar a cambiar la selección de algún campo, sólo guardarla o crearla de una vez, con opciones ubicadas de forma visible y de fácil acceso.
- Si la tarjeta creada es guardada, se podrá visualizar desde la segunda opción “Tarjetas creadas”
- En la opción “Tarjetas creadas” se listan las tarjetas creadas por el usuario y cada una tendrá la opción de ser publicada o de permanecer de manera privada.

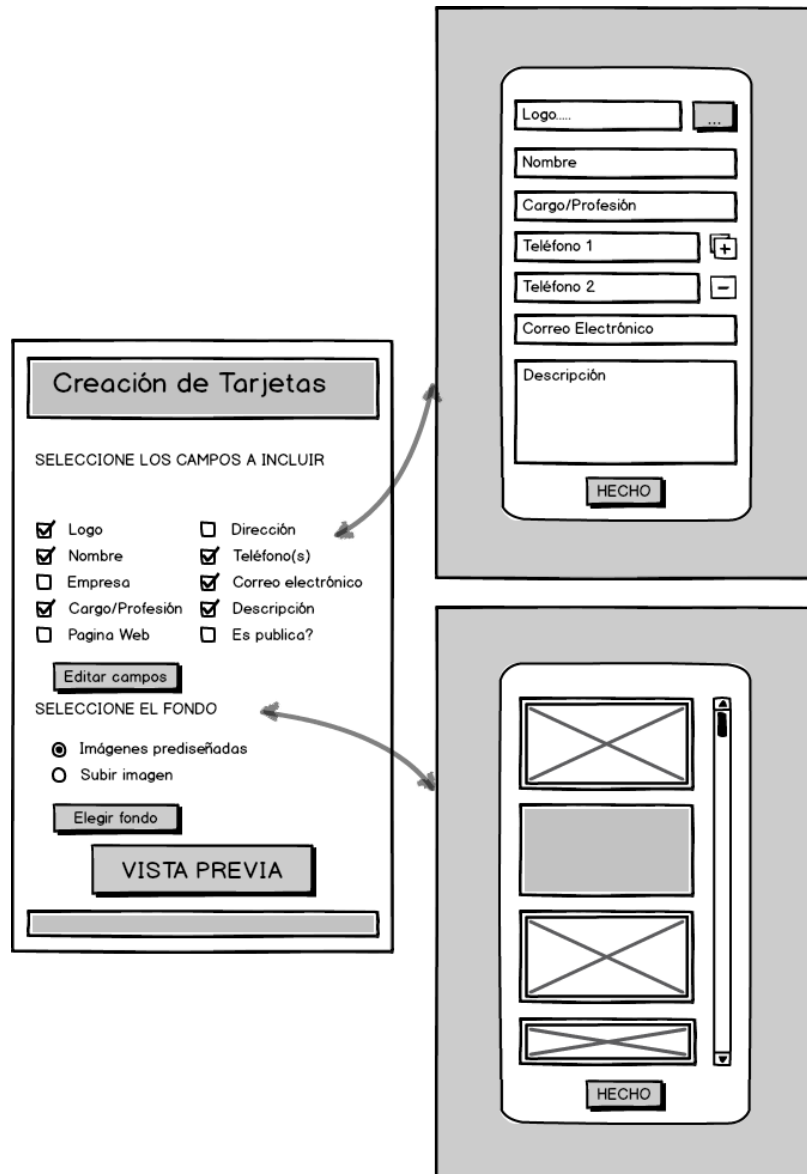


Ilustración 11: Historia de Usuario - Plataforma para la creación de tarjetas 1



Ilustración 12: Historia de Usuario - Plataforma para la creación de tarjetas 2

Número: 7

Nombre: Almacenamiento de tarjetas creadas

Tipo: Funcional

Descripción: Como cliente me gustaría que las tarjetas que el usuario cree, puedan ser almacenadas en su cuenta de usuario.

- Cada tarjeta tendrá las siguientes opciones:
 - Modificar
 - Eliminar

Observaciones:

- Es obligatorio que las tarjetas que el usuario cree, estén en un lugar separado de las tarjetas que ha tomado de otros usuarios. Estas tarjetas creadas estarán en la lista mostrada en la opción “Tarjetas Creadas”.
- Las tarjetas se visualizarán en forma de acordeón mostrando detalles de las tarjetas
- Cuando el usuario de clic en una tarjeta en específico debe visualizarse la pantalla completa de la misma

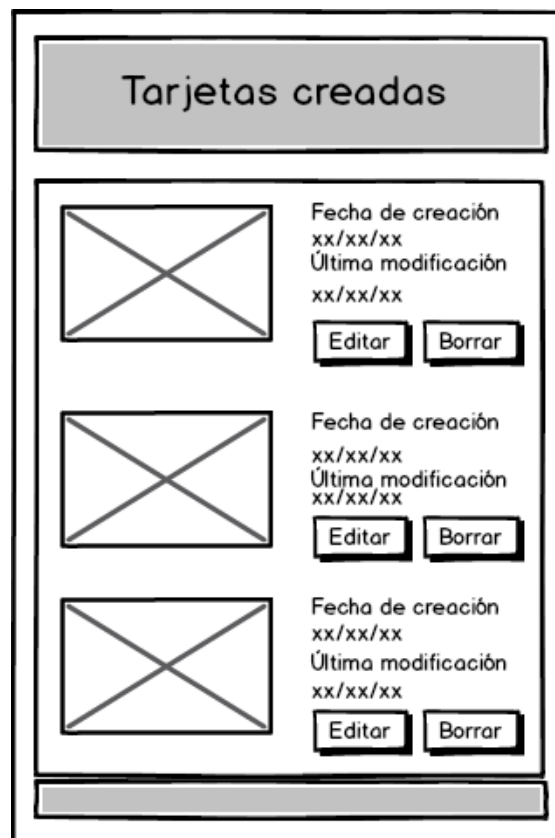


Ilustración 13: Historia de Usuario - Almacenamiento de tarjetas creadas

Número: 8

Nombre: Modificación de tarjetas creadas

Tipo: Funcional

Descripción: Como cliente requiero que la aplicación permita a los usuarios modificar tarjetas creadas con anterioridad.

- Al momento de presionar el botón “Modificar” en la opción de “Tarjetas Creadas”, automáticamente se cargarán los datos y configuración previamente guardados de la tarjeta a modificar.
- El usuario podrá modificar las tarjetas que se encuentran en la opción “Tarjetas Creadas”. Cada una de estas tarjetas debe tener un botón que permita acceder a la plataforma de edición de tarjetas.
- El usuario sólo podrá modificar las tarjetas que hayan sido creadas en su cuenta.
- El usuario podrá modificar:
 - Qué campos de usuario va a tener la tarjeta
 - El contenido de los campos de la tarjeta, incluyendo el logo
 - El fondo de la tarjeta

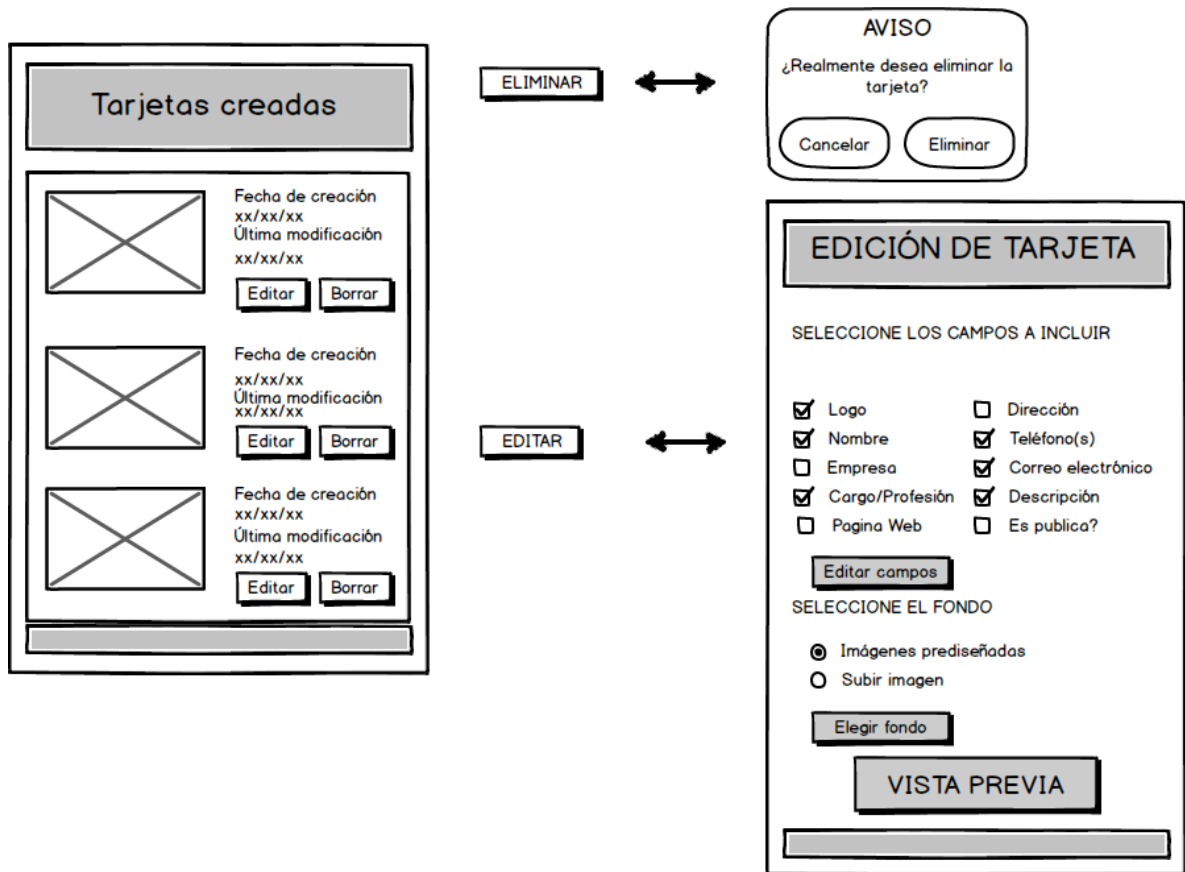


Ilustración 14: Historia de Usuario - Modificación de tarjetas creadas

Número: 9

Nombre: Búsqueda de tarjetas dentro del repositorio global.

Tipo: Funcional

Descripción: Como cliente me gustaría que la aplicación permita buscar tarjetas de cualquier persona, usando filtros de búsqueda.

- La opción de búsqueda debe estar en la pantalla principal después de que el usuario realice el login.
- Debe existir un campo de entrada de texto para que el usuario escriba lo que desea buscar.
- Deben existir filtros que faciliten la búsqueda de tarjetas. Estos filtros se harán a través de nombre, compañía, servicio, producto.
- Debe existir un botón que permita limpiar los resultados obtenidos y regrese a la pantalla de búsqueda nuevamente.

Observaciones:

- Se debe garantizar un tiempo de búsqueda máximo de 5 segundos.
- Cuando el usuario realice una búsqueda, lo hará sobre el total de las tarjetas almacenadas por los demás usuarios en la base de datos.

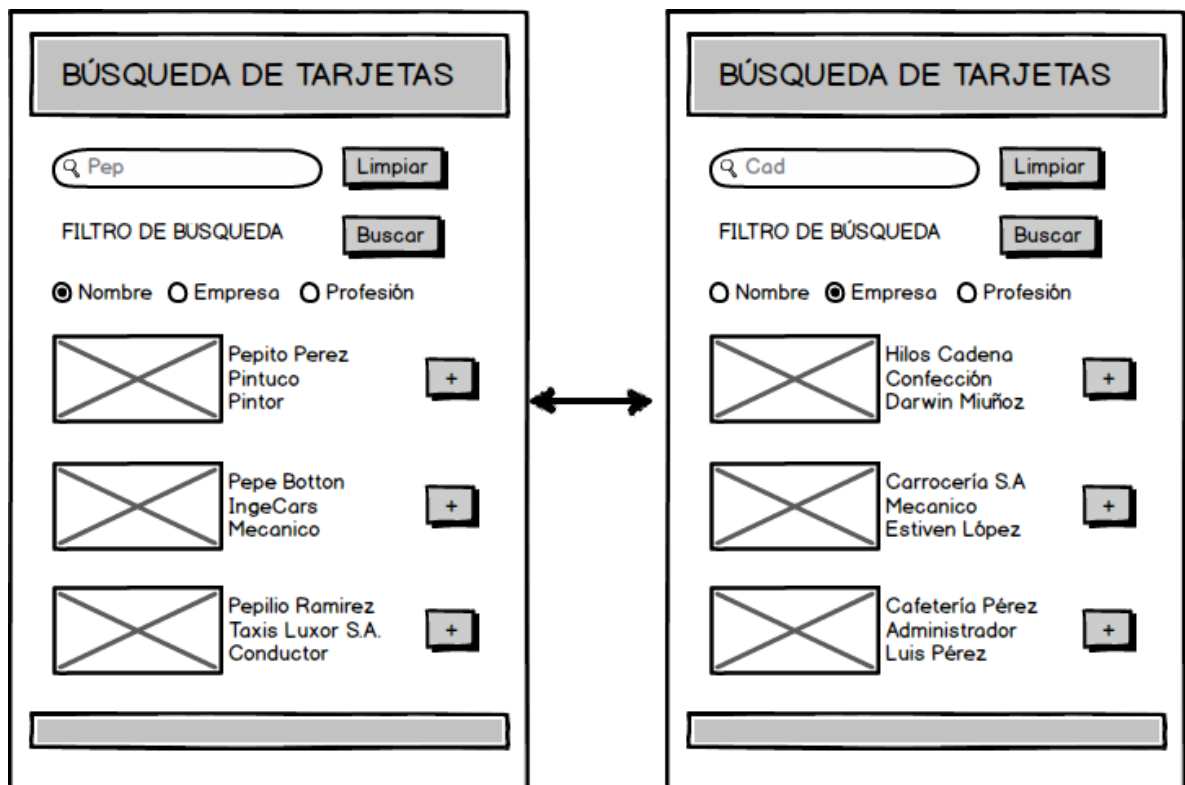


Ilustración 15: Historia de Usuario - Búsqueda de tarjetas dentro del repositorio global

Número: 10

Nombre: Agregar tarjetas del repositorio global a la cuenta de usuario

Tipo: Funcional

Descripción: Como cliente requiero que la aplicación permita al usuario almacenar en su cuenta las tarjetas que agregue del repositorio global.

- Después de realizar la búsqueda de una tarjeta en el repositorio global, cada resultado obtenido podrá visualizarse en pantalla completa después de hacer un toque sobre ella.
- Se podrá agregar la tarjeta a la cuenta del usuario a través del botón “Agregar a mi cuenta”.

Observaciones:

- Las tarjetas agregadas a la cuenta de usuario se podrán visualizar en la opción “Mis Tarjetas” en el menú principal.

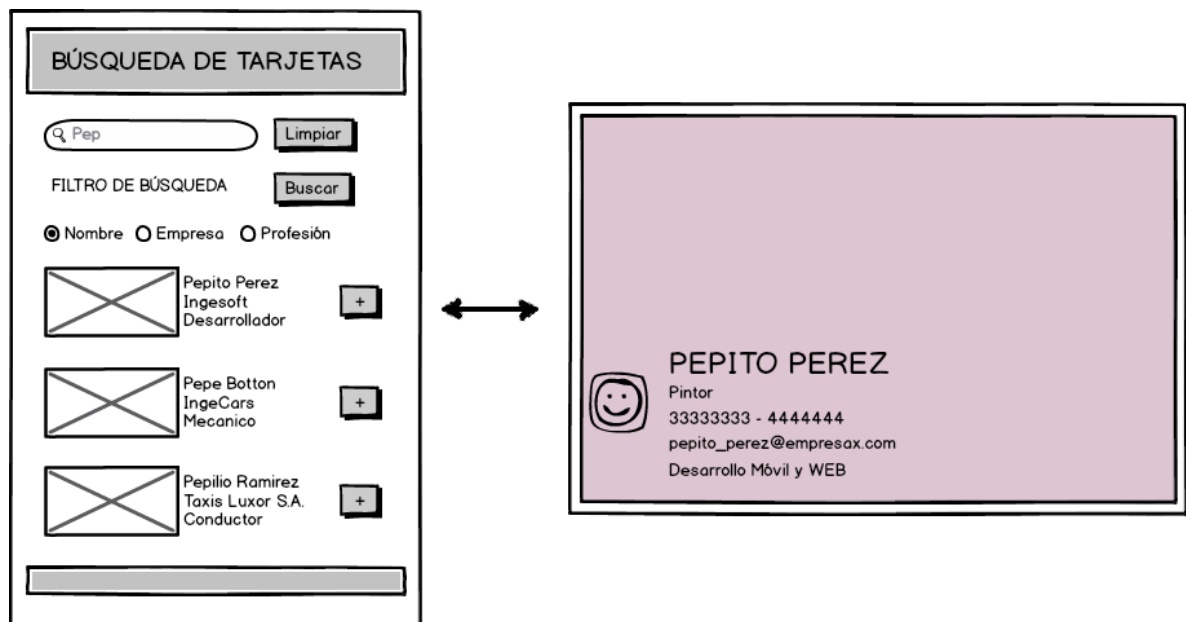


Ilustración 16: Historia de Usuario - Agregar tarjetas del repositorio a la cuenta de usuario

Número: 11

Nombre: Visualización de tarjetas almacenadas por el usuario

Tipo: Funcional

Descripción: Como cliente me gustaría que la aplicación le permitiera al usuario enlistar de manera organizada las tarjetas que ha agregado a su cuenta de usuario desde el repositorio global.

- Debe existir una opción que permita elegir el modo de ordenamiento de las tarjetas. (Fecha, alfabético o frecuencia de uso).
- Las tarjetas se deben mostrar en miniatura en forma de acordeón.
- Cada tarjeta de la lista debe tener accesos directos para llamada, SMS o correo electrónico.
- Cuando se dé un toque sobre un elemento de la lista, se debe mostrar la tarjeta en pantalla completa.

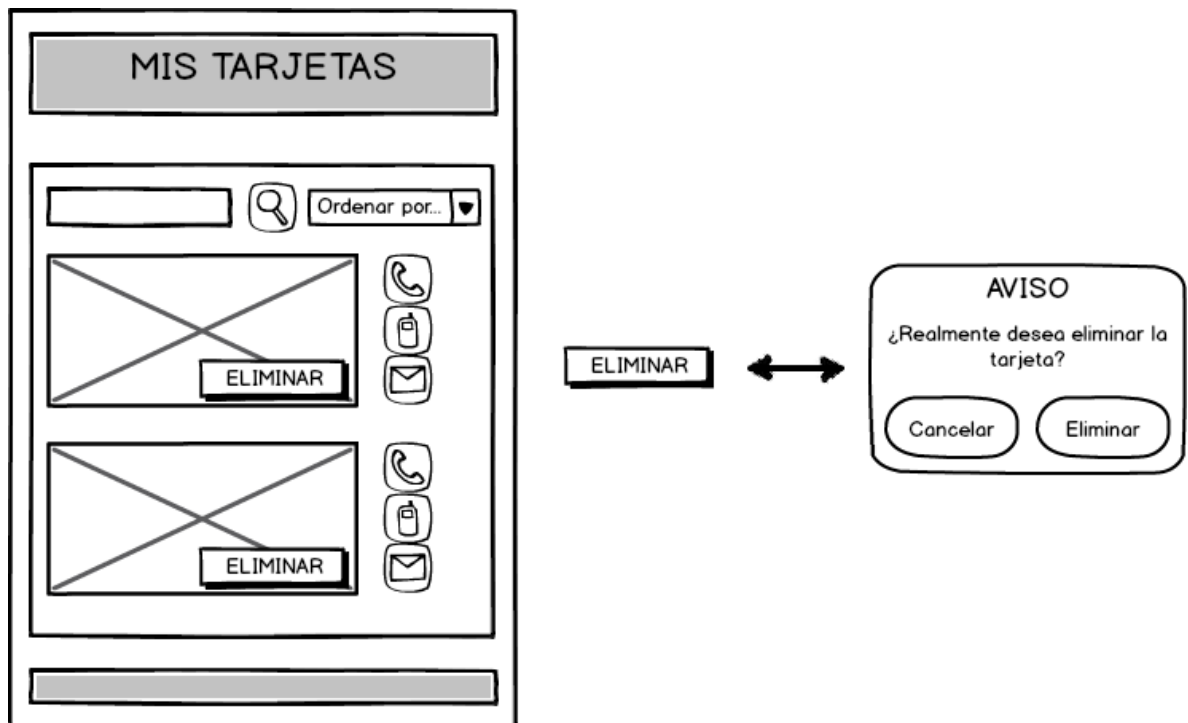


Ilustración 17: Historia de Usuario - Visualización de tarjetas almacenadas por el usuario

Número: 12

Nombre: Búsqueda de tarjetas agregadas en la cuenta de usuario

Tipo: Funcional

Descripción: Como cliente me gustaría que la aplicación permitiera realizar una búsqueda de las tarjetas que se han agregado a la cuenta de usuario.

- Debe existir un campo de entrada de texto para que el usuario escriba lo que desea buscar.
- Deben existir filtros que faciliten la búsqueda de tarjetas. Estos filtros se harán a través de nombre, compañía, servicio, producto, etc.
- Debe existir un botón que permita limpiar los resultados obtenidos y regrese a la pantalla de búsqueda nuevamente.

Observaciones:

- Se debe garantizar un tiempo de búsqueda máximo de 5 segundos.
- Cuando el usuario realice una búsqueda, lo hará sobre el total de las tarjetas almacenadas en su cuenta de usuario.

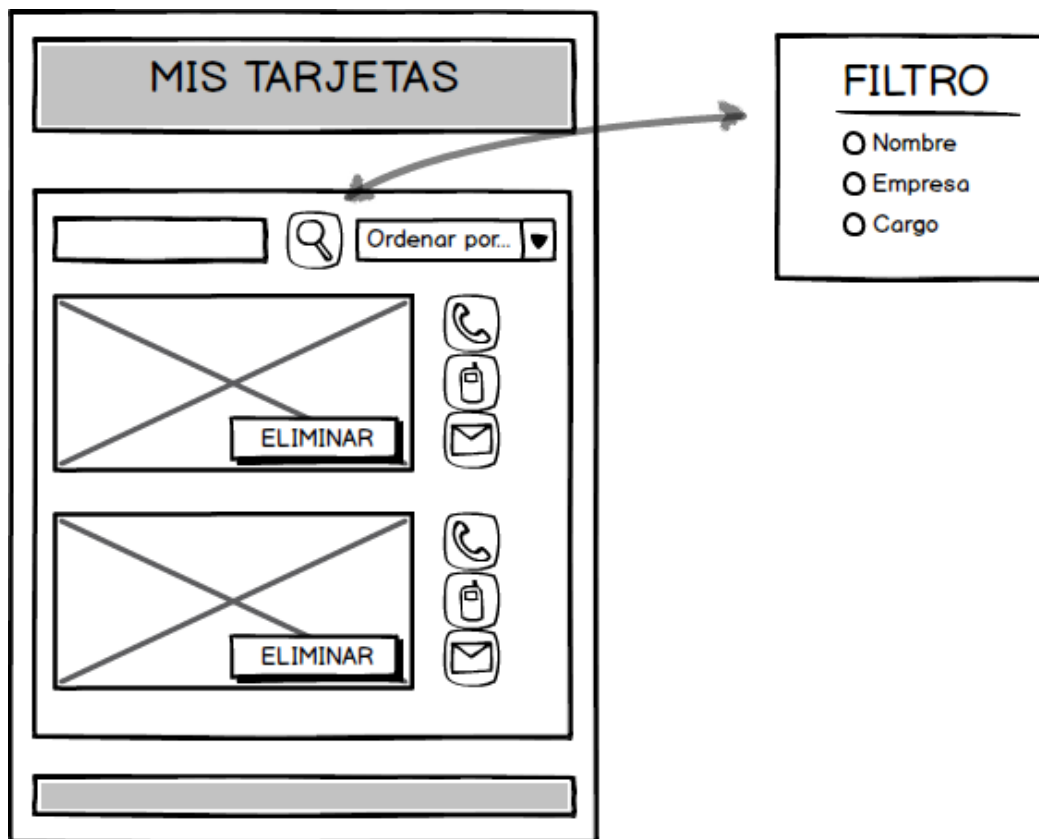


Ilustración 18: Historia de Usuario - Búsqueda de tarjetas agregadas en la cuenta de usuario

7.1.2.2. Requerimientos no funcionales

Número: 13

Nombre: Pantalla Inicial

Tipo: No Funcional

Descripción: Como usuario requiero que la aplicación posea una pantalla inicial al momento de abrir la aplicación.

- Debe poseer botones para inicio de sesión y registro.
- Debe poseer un botón para información sobre la aplicación y soporte.

Observaciones:

- La pantalla inicial debe poseer una interfaz amigable y sencilla

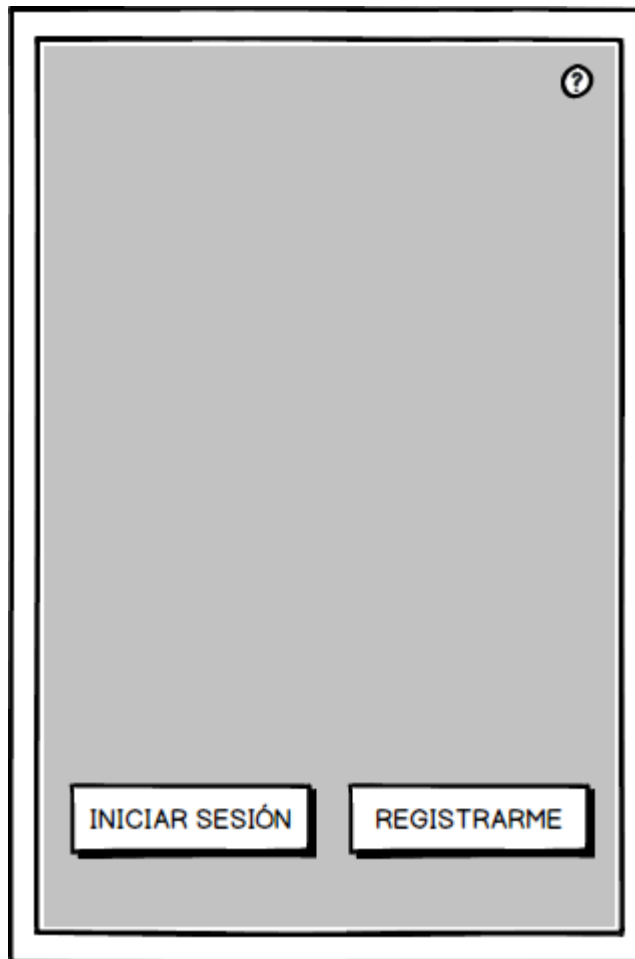


Ilustración 19: Historia de Usuario - Pantalla Inicial

Número: 14

Nombre: Login

Tipo: No Funcional

Descripción: Como cliente requiero que la aplicación permita a los usuarios ingresar a su cuenta por medio de un login.

- La pantalla inicial de la aplicación debe contener el sistema de login.
- En la sección de login, deben existir los siguientes campos de entrada:
 - E-mail
 - Contraseña
- Debe existir un botón que diga “Ingresar” para proseguir con el login. Si los datos ingresados por el usuario son correctos, se deberá cargar su cuenta con sus datos respectivos.

El diagrama muestra una interfaz de usuario para el inicio de sesión. En la parte superior, hay un botón rectangular con el texto "INGRESAR". Debajo de este botón, hay dos campos de entrada de texto: el primero está etiquetado como "Correo Electrónico" y el segundo como "Contraseña". Debajo de estos campos, hay un botón rectangular con el texto "Iniciar Sesión". Inmediatamente debajo del botón "Iniciar Sesión", se encuentra el texto "¿Has olvidado tu contraseña?". En la parte inferior del formulario, hay un campo de entrada de texto adicional que está actualmente vacío.

Ilustración 20: Historia de Usuario - Login

Número: 15

Nombre: Compatibilidad, versión de Android

Tipo: No Funcional

Descripción: Como cliente requiero que la aplicación pueda funcionar sobre sistemas operativos Android con versiones superiores a 4.0 Ice Cream Sandwich.

Observaciones: Los usuarios con sistemas operativos Android de versión más baja, no pueden tener permitida la instalación del aplicativo.

Número: 16

Nombre: Requisitos mínimos de hardware

Tipo: No Funcional

Descripción: Como cliente requiero que la aplicación pueda funcionar sobre los siguientes requisitos mínimos: 512 RAM, procesador 1Ghz de un núcleo y una pantalla de mínimo 4.0”.

Número: 17

Nombre: Hospedaje en servidor web

Tipo: No Funcional

Descripción: Como cliente requiero que la información de los usuarios que utilicen la aplicación, estén alojados en un servidor web.

Observaciones: Tanto la base de datos, como archivos multimedia de la aplicación, deben estar alojados en el servidor web.

Número: 18

Nombre: Base de datos MySQL

Tipo: No Funcional

Descripción: Como cliente requiero que la aplicación utilice una base de datos relacional MySQL.

Observaciones: La base de datos debe funcionar en el servidor con el fin de acceder a los datos desde cualquier dispositivo con solo tener acceso a internet.

Número: 19

Nombre: Soporte de 3000 usuarios

Tipo: No Funcional

Descripción: Como cliente requiero que el servidor tenga un soporte mínimo de 3000 usuarios en conexión simultánea.

7.2. DIAGRAMA DE CASOS DE USO

Para obtener una mejor conceptualización del funcionamiento de la aplicación, se construye un Diagrama de Casos de Uso con una descripción de procesos incluyendo funcionalidades específicas, actores del sistema y sus relaciones.

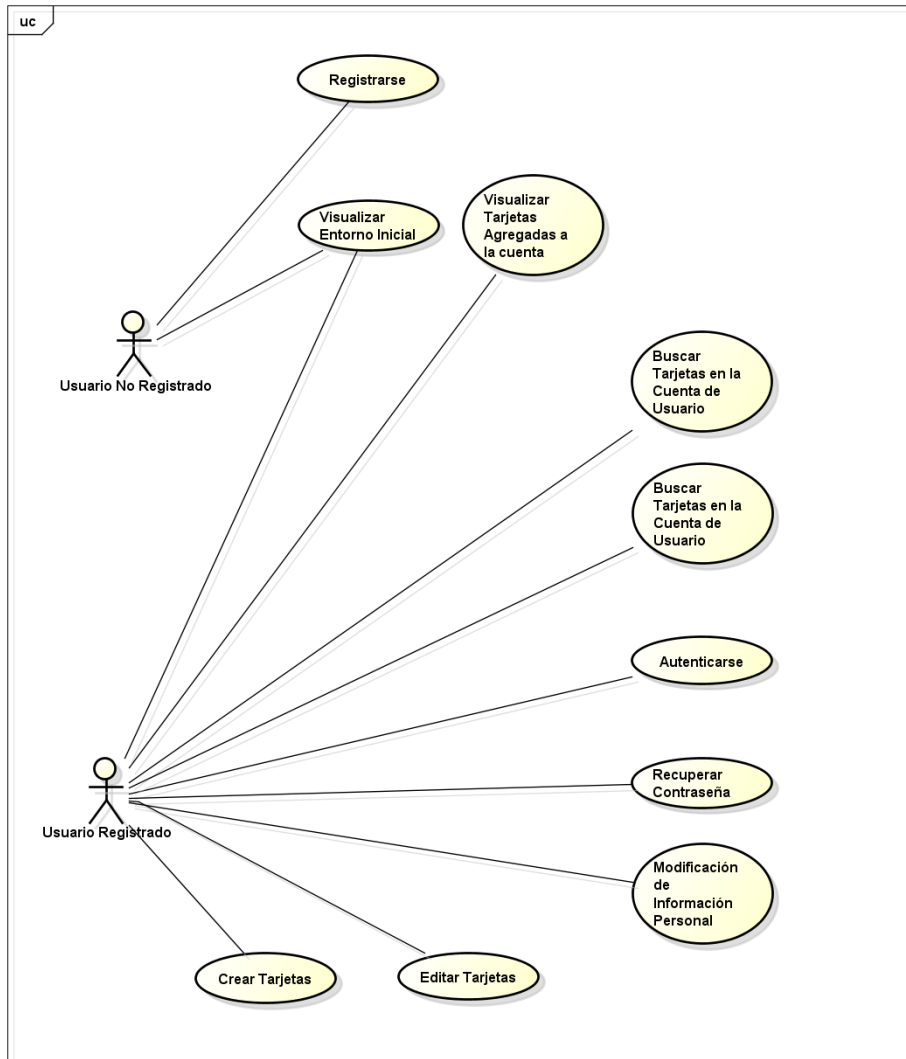


Ilustración 21: Diagrama de Casos de Uso

7.2.1. Especificación de casos de uso

ACTORES

A continuación se listan los actores que interactúan con la aplicación:

N°1	Nombre: Usuario no registrado
	Descripción: Representa a un usuario que no se ha identificado frente al sistema. Este usuario solo puede visualizar el entorno inicial de la aplicación en donde puede realizar un registro para obtener una cuenta de usuario.
N°2	Nombre: Usuario registrado
	Descripción: Representa a un usuario que está registrado en el sistema y posee una cuenta de usuario. Este usuario puede visualizar y tener acceso a todas las funcionalidades de la aplicación, además de la posibilidad de darse de baja para ser de nuevo un usuario no registrado.

Tabla 3: Actores - Casos de Uso

CASOS DE USO

Los actores mencionados anteriormente se relacionan con el sistema a través de unos casos de uso. A continuación se describen estos casos:

1	Nombre	Visualizar entorno inicial
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	Permite al usuario no registrado y al usuario registrado visualizar la pantalla inicial de la aplicación.
	Actores:	Usuario no registrado/Usuario registrado.
	Precondiciones:	Ingresar al sistema sin la dependencia de un registro.
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario ingresa a la aplicación 2. El sistema muestra la pantalla inicial 3. El usuario visualiza las opciones de inicio de sesión, registro y recuperación de contraseña.
	Flujo alternativo:	<ol style="list-style-type: none"> 2. A. El sistema muestra la pantalla inicial sin cargar datos previos de otro usuario. 2. B. El sistema muestra la pantalla inicial cargando los datos del usuario que haya usado la aplicación y no ha cerrado sesión.
Poscondiciones:	El sistema restringe la entrada a las funcionalidades de la aplicación al usuario no registrado.	

Tabla 4: Especificación de caso de uso - Visualizar entorno inicial

2	Nombre	Registrarse
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	Permite al usuario que no está registrado crear una cuenta de usuario para acceder a todas las funcionalidades de la aplicación.
	Actores:	Usuario no registrado
	Precondiciones:	El usuario que procede a registrarse no puede estar registrado con anterioridad
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de registro 2. Procede a llenar campos obligatorios para el registro 3. El sistema verifica los datos ingresados 4. El usuario acepta términos y condiciones de uso 5. Se completa el registro
	Flujo alternativo:	<ol style="list-style-type: none"> 3. A. El sistema comprueba los datos ingresados; si los datos son incorrectos se avisa al actor de ello permitiendo que pueda corregirlos. 4. B. Si el usuario no acepta los términos y condiciones de uso, el registro se considera nulo.
Poscondiciones:	El usuario no registrado ahora será un usuario registrado con su propia cuenta de usuario y con la posibilidad de acceder a las funcionalidades completas de la aplicación.	

Tabla 5: Especificación de caso de uso - Registrarse

3	Nombre	Aceptar términos y condiciones de uso
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	Durante el proceso de registro, el usuario no registrado debe aceptar términos y condiciones de uso para el manejo de la información personal del usuario. El usuario podrá leer toda la información referente al uso de sus datos personales y precondiciones para el uso de la aplicación.
	Actores:	Usuario no registrado
	Precondiciones:	El usuario debe estar en medio del proceso de registro para llegar al apartado de términos y condiciones de uso
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario no registrado realiza los pasos iniciales de registro en la aplicación. 2. Después de validar correctamente los datos ingresados, el usuario puede proceder a leer y aceptar los términos y condiciones de uso de la aplicación. 3. El usuario podrá continuar con el proceso de registro
	Flujo alternativo:	1. A. El usuario registrado no acepta los términos y condiciones de uso. La aplicación hace una última validación al usuario, y procede a anular el registro.
Poscondiciones:	Cuando un usuario acepta términos y condiciones de uso, se crea su cuenta y permite a la aplicación administrar su información personal.	

Tabla 6: Especificación de caso de uso - Aceptar términos y condiciones de uso

4	Nombre	Verificar datos ingresados
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	<p>Este caso de uso se encarga de verificar:</p> <ul style="list-style-type: none"> • Los datos ingresados por el usuario no registrado al momento de realizar el registro. • Los datos ingresados por el usuario registrado al momento de iniciar sesión.
	Actores:	Usuario no registrado/Usuario registrado.
	Precondiciones:	El proceso de verificación de datos se realiza cada vez que el usuario llena campos de texto para el proceso de registro e inicio de sesión
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario procede a llenar campos de registro o de inicio de sesión. 2. La aplicación se encarga de verificar si los datos ingresados son correctos. 3. La aplicación continúa con su proceso.
	Flujo alternativo:	2. A. Si los datos ingresados no cumplen los requisitos necesarios, la aplicación da aviso de los errores al usuario y permite corregir el texto ingresado en los campos de texto.
Poscondiciones:	Cuando la verificación culmina, la aplicación continúa con los procesos correspondientes.	

Tabla 7: Especificación de caso de uso - Verificar datos ingresados

5	Nombre	Iniciar sesión
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	El usuario registrado puede acceder a las funcionalidades de la aplicación a través de la plataforma de inicio de sesión. Para ello, debe ingresar sus datos en los campos correspondientes.
	Actores:	Usuario registrado
	Precondiciones:	Solo un usuario registrado puede iniciar sesión. El usuario no registrado que intente iniciar sesión, no pasará el proceso de verificación de datos.
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario registrado se dirige a la pantalla de inicio de sesión 2. El usuario ingresa los datos correspondientes (email y contraseña) 3. El sistema verifica la correctitud de los datos ingresados 4. El usuario ingresa a su cuenta con todas las funcionalidades de la aplicación
	Flujo alternativo:	3. A. Si los datos ingresados por el usuario son erróneos, el sistema da aviso, y permite corregirlos para realizar un nuevo intento de inicio de sesión.
Poscondiciones:	Después de que un usuario registrado inicia sesión correctamente, tiene acceso a todas las funcionalidades de la aplicación.	

Tabla 8: Especificación de caso de uso - Iniciar sesión

6	Nombre	Cerrar sesión
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	Después de que un usuario registrado inicia sesión correctamente, tiene la posibilidad de cerrar sesión con el fin de limitar el acceso a terceros y proteger sus datos personales.
	Actores:	Usuario registrado
	Precondiciones:	El usuario registrado debe haber iniciado sesión previamente.
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario inicia sesión correctamente 2. El usuario selecciona la opción de “Cerrar Sesión” 3. El sistema pide confirmación por parte del usuario para cerrar sesión. 4. El sistema procede a borrar los datos del usuario almacenados localmente.
	Flujo alternativo:	3. A. Si el usuario cancela el cierre de sesión, la aplicación continúa funcionando normalmente con la sesión abierta.
Poscondiciones:	El usuario debe proporcionar nuevamente sus datos si quiere iniciar sesión.	

Tabla 9: Especificación de caso de uso - Cerrar sesión

7	Nombre	Recuperar contraseña
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	El usuario tiene la posibilidad de recuperar su contraseña en caso de que esta sea olvidada.
	Actores:	Usuario registrado
	Precondiciones:	El usuario registrado debe haber establecido una pregunta y respuesta de seguridad al momento de registrarse en el sistema.
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla principal de la aplicación. 2. El usuario selecciona la opción “Recuperar Contraseña” 3. El usuario ingresa su correo electrónico para que el sistema verifique la existencia del usuario 4. El sistema le muestra al usuario la pregunta secreta. 5. El usuario procede a escribir la respuesta secreta. 6. El sistema verifica la validez de los datos. 7. El usuario escribe una nueva contraseña
	Flujo alternativo:	<p>3. A. El correo ingresado por el usuario no está registrado en la Base de Datos. El sistema procede a mostrar un mensaje de error y permite al usuario ingresar nuevamente su correo electrónico.</p> <p>6. A. El sistema detecta una respuesta secreta errónea. El sistema procede a mostrar un mensaje de error y permite al usuario intentar 2 veces más ingresar la respuesta secreta. Si el usuario sobrepasa el límite de intentos, el sistema no le permitirá recuperar la contraseña de dicho correo por las siguientes 2 horas.</p>
	Poscondiciones:	El usuario podrá iniciar sesión de forma normal con su nueva contraseña

Tabla 10: Especificación de caso de uso - Recuperación de contraseña

8	Nombre	Crear Tarjeta
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	El usuario registrado tiene un editor completo para crear tarjetas de presentación.
	Actores:	Usuario registrado
	Precondiciones:	El usuario debe haber iniciado sesión correctamente
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario ingresa a la sección de “Crear Tarjeta” 2. El usuario ingresa todos los datos requeridos por el sistema para la creación de la tarjeta. 3. El sistema valida los datos ingresados. 4. La tarjeta se crea y se registra en el sistema.
	Flujo alternativo:	3. A. Si el sistema detecta inconsistencias en los datos ingresados por el usuario, aparecerá un mensaje con el error y se le permitirá corregirlo.
Poscondiciones:	Después de crear la tarjeta satisfactoriamente, el usuario podrá modificarla en cualquier momento.	

Tabla 11: Especificación de caso de uso - Crear Tarjeta

9	Nombre	Modificar Tarjeta
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	El usuario tiene la opción de editar tarjetas creadas con anterioridad, con la posibilidad de hacer o no públicos los cambios.
	Actores:	Usuario registrado
	Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión correctamente. • La tarjeta a modificar, debe haber sido creada con anterioridad
	Flujo normal:	<ul style="list-style-type: none"> • El usuario se dirige a la sección de “Tarjetas Creadas” • Selecciona la tarjeta que quiere editar • Edita los campos que necesite de la tarjeta • El sistema verifica la validez de los nuevos datos • Los datos de la tarjeta se modifican en la Base de Datos.
	Flujo alternativo:	4. A. Si los nuevos datos ingresados no son válidos, el sistema muestra un mensaje de error y le permite al usuario modificarlos.
	Poscondiciones:	Los cambios realizados se ven reflejados en la Base de Datos lo que permite que todos los usuarios detecten los cambios de la tarjeta en tiempo real.

Tabla 12: Especificación de caso de uso - Modificar Tarjeta

10	Nombre	Visualizar Tarjetas Creadas
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	El usuario tiene la posibilidad de visualizar las tarjetas que ha creado previamente en su cuenta
	Actores:	Usuario registrado
	Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión correctamente. • La tarjeta a visualizar, debe haber sido creada con anterioridad.
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario se dirige a la sección de “Tarjetas Creadas” 2. El usuario puede visualizar la vista previa en miniatura de todas las tarjetas que ha creado.
	Flujo alternativo:	2. A. Si el usuario da clic en una de las tarjetas la podrá visualizar en tamaño completo.
Poscondiciones:	Una vez visualizadas las tarjetas, el usuario podrá hacer uso de las opciones de “Editar” y “Borrar” que se encuentran en la misma sección.	

Tabla 13: Especificación de caso de uso - Visualizar Tarjetas Creadas

11	Nombre	Modificar información personal
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	El usuario registrado tiene la posibilidad de modificar su información personal tal como su nombre y contraseña.
	Actores:	Usuario registrado
	Precondiciones:	El usuario debe haber iniciado sesión correctamente.
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario se dirige a la sección de “Modificar Mis Datos” 2. Selecciona el tipo de dato que desea modificar (Nombre, Contraseña) 3. Introduce los nuevos valores de los campos seleccionados 4. El usuario da clic en el botón “Actualizar Datos”
	Flujo alternativo:	<ol style="list-style-type: none"> 3. A. Si el sistema detecta inconsistencias en los datos ingresados por el usuario, aparecerá un mensaje con el error y se le permitirá corregirlo.
Poscondiciones:	El sistema modifica los datos, muestra un mensaje indicando que los cambios fueron realizados con éxito y dirige al usuario a la pantalla inicial de la aplicación.	

Tabla 14: Especificación de caso de uso - Modificar información personal

12	Nombre	Visualizar Tarjetas Agregadas a la cuenta
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	El usuario registrado tiene la posibilidad de visualizar las tarjetas que ha agregado previamente a su cuenta
	Actores:	Usuario registrado
	Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión correctamente • El usuario debe haber agregado la tarjeta a visualizar
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario se dirige a la sección “Mis tarjetas” 2. El usuario puede visualizar la vista previa en miniatura de todas las tarjetas que ha creado.
	Flujo alternativo:	<ol style="list-style-type: none"> 2. A. El usuario puede hacer uso de un filtro de búsqueda por Nombre, Empresa o Cargo para cargar una o varias tarjetas específicas. 2. B. Si el usuario da clic en la miniatura se visualiza la pantalla completa de la tarjeta.
Poscondiciones:	El usuario puede eliminar o hacer uso de las opciones llamar, enviar SMS o enviar un correo electrónico para cada una de las tarjetas de esta sección.	

Tabla 15: Especificación de caso de uso - Visualizar Tarjetas Agregadas a la cuenta

13	Nombre	Ordenar Tarjetas
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	El usuario podrá ordenar las tarjetas que ha agregado a su cuenta
	Actores:	Usuario registrado
	Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión correctamente • Haber agregado tarjetas desde el repositorio global a su cuenta
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario se dirige a la sección “Mis tarjetas” 2. Desde allí selecciona el tipo de ordenamiento que desee aplicar a la lista de tarjetas
Flujo alternativo:	1. A. El usuario utiliza el ordenamiento que el sistema trae por defecto para la visualización de las tarjetas que ha agregado desde el repositorio global	
Poscondiciones:	El usuario podrá visualizar las tarjetas de forma organizada de acuerdo al criterio de ordenamiento seleccionado	

Tabla 16: Especificación de caso de uso - Ordenar Tarjetas

14	Nombre	Buscar Tarjetas en el repositorio global
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	El usuario podrá buscar tarjetas de otros usuarios dentro de un repositorio global
	Actores:	Usuario registrado
	Precondiciones:	El usuario debe haber iniciado sesión correctamente
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario se dirige a la sección “Buscar Tarjetas” 2. Selecciona el filtro más apropiado para realizar la búsqueda 3. Ingresa el texto que deberá coincidir con el de la tarjeta que busca 4. El usuario da clic en el botón Buscar
	Flujo alternativo:	3. A. El usuario puede limpiar el texto que ha escrito haciendo uso del botón limpiar (x)
Poscondiciones:	El usuario puede agregar una o varias de las tarjetas que se enlistan como coincidencias de la búsqueda.	

Tabla 17: Especificación de caso de uso - Buscar Tarjetas en el repositorio global

15	Nombre	Agregar tarjeta a cuenta de usuario
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	El usuario podrá agregar tarjetas desde el repositorio global a su cuenta
	Actores:	Usuario registrado
	Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión correctamente • El usuario debe haber encontrado una tarjeta que coincida con sus parámetros de búsqueda
	Flujo normal:	<ol style="list-style-type: none"> 1. El usuario se dirige a la sección “Buscar Tarjetas” 2. Selecciona el filtro más apropiado para realizar la búsqueda 3. Ingresa el texto que deberá coincidir con el de la tarjeta que busca 4. Una vez encuentre la tarjeta que coincida con su búsqueda le da clic en el botón adicionar (+)
	Flujo alternativo:	El usuario decide no adicionar ninguna tarjetas de las encontradas
Poscondiciones:	El usuario puede visualizar las tarjetas que ha adicionado desde el repositorio global desde la sección “Mis Tarjetas”	

Tabla 18: Especificación de caso de uso - Agregar tarjeta a cuenta de usuario

16	Nombre	Buscar tarjetas en la cuenta de usuario
	Autor	Carlos Alfredo Londoño S. – Andrés Giraldo Carvajal
	Fecha	29/09/2014
	Descripción:	El usuario podrá buscar dentro de su propia cuenta, tarjetas que haya agregado previamente desde el repositorio global
	Actores:	Usuario registrado
	Precondiciones:	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión correctamente • Debe haber agregado tarjetas previamente desde el repositorio global
	Flujo normal:	<ul style="list-style-type: none"> • El usuario se dirige a la sección “Mis Tarjetas” • Selecciona el filtro más apropiado para realizar la búsqueda • Ingresa el texto que deberá coincidir con el de la tarjeta que busca • El usuario da clic en el botón de búsqueda
	Flujo alternativo:	3. A. El usuario puede limpiar el texto que ha escrito haciendo uso del botón “limpiar”
Poscondiciones:	El usuario puede visualizar sólo las tarjetas que coincidan con sus parámetros de búsqueda.	

Tabla 19: Especificación de caso de uso - Buscar tarjetas en la cuenta de usuario

8. DISEÑO Y ARQUITECTURA DE SOFTWARE

El presente capítulo tiene como objetivo exponer el diseño y la arquitectura de software sobre el que está soportado el aplicativo a desarrollar.

8.1. DIAGRAMA DE CLASES

A continuación se exponen las clases que componen el presente proyecto. Se especifican atributos, funciones y relaciones existentes entre ellas.

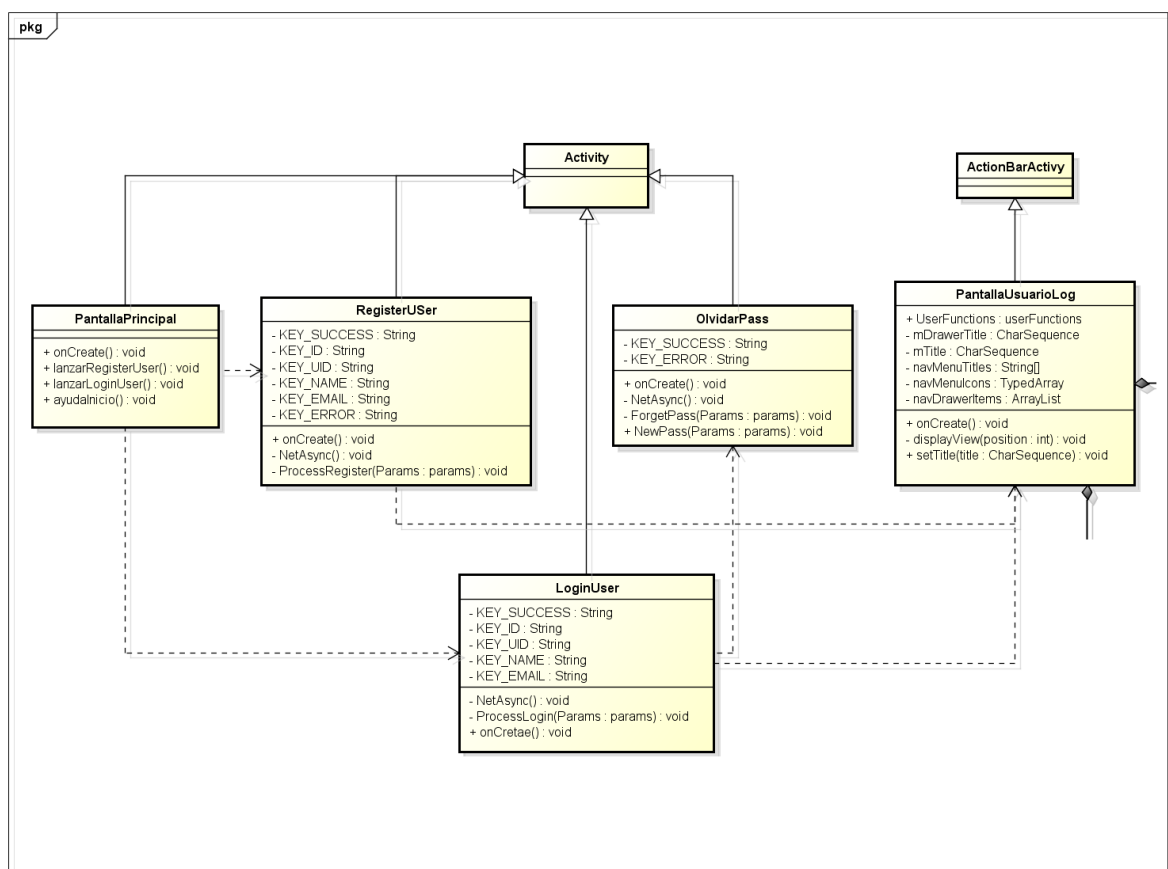


Ilustración 22: Diagrama de Clases (Parte 1)

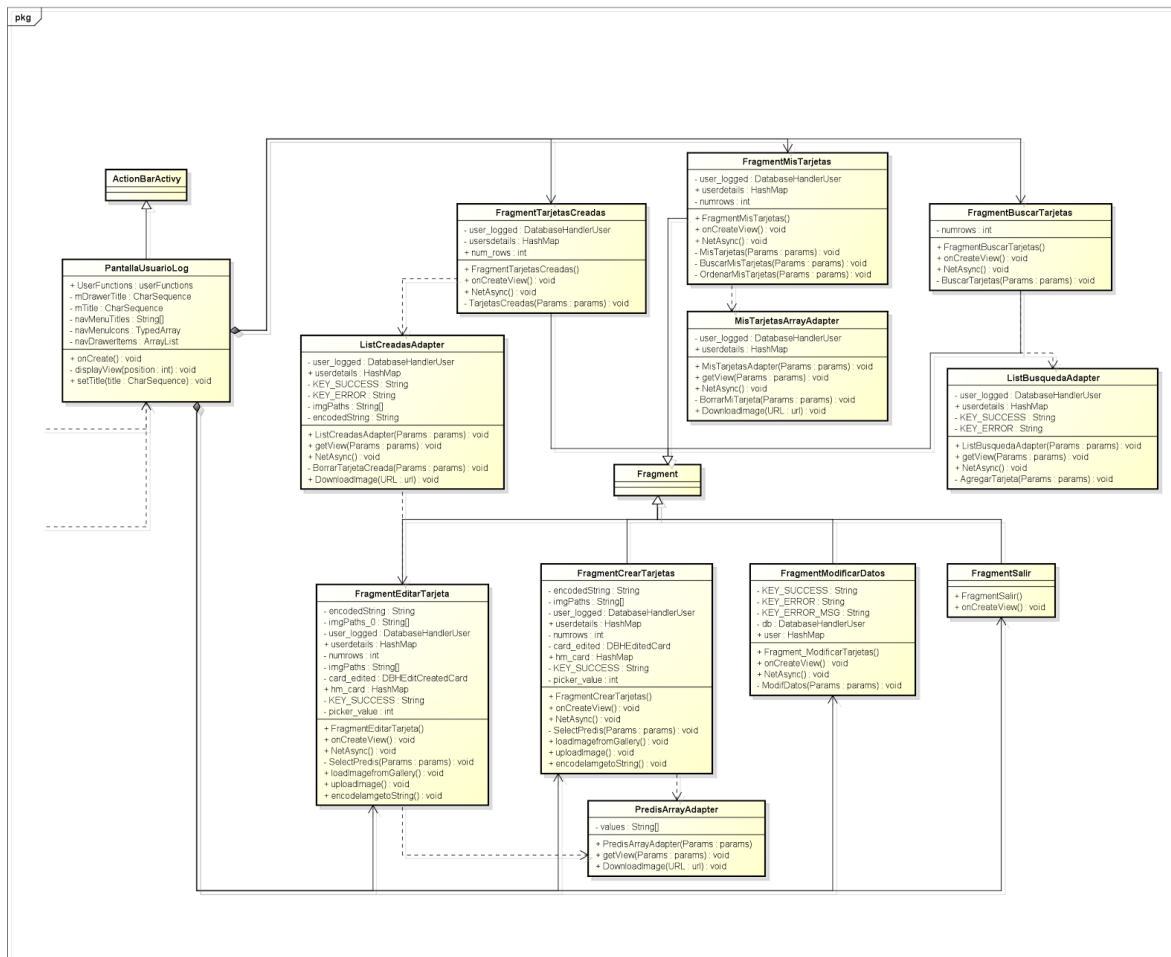


Ilustración 23: Diagrama de Clases (Parte 2)

Especificaciones – Diagrama de Clases:

Nro.	Nombre de la clase	Descripción
1	Activity	Proporciona una pantalla con la que los usuarios pueden interactuar con el fin de hacer algo. Una aplicación desarrollada para Android por lo general se compone de múltiples actividades que están débilmente unidas entre sí [24].
2	LoginUser	Es una clase hija de Activity que permite al usuario realizar el proceso de login. Esta funciona como un hilo independiente al igual que la mayoría de procesos en la aplicación. Desde esta actividad el usuario puede iniciar la funcionalidad de recuperar contraseña.
3	OlvidarPass	Es la encargada de recuperar la contraseña de un usuario que la haya olvidado.
4	PantallaPrincipal	Es la primera actividad que el usuario encuentra y le permite seleccionar entre realizar un registro o un login cuando el usuario ya se ha registrado.
5	RegisterUser	Es la clase encarga de proveer las funcionalidades para el proceso de registro de usuarios.
6	ActionBarActivity	Tiene como fin ubicar al usuario dentro de la aplicación y proveer modos de navegación haciendo uso de una barra superior con información de la aplicación.
7	PantallaUsuarioLog	Es una clase que hereda de ActionBarActivity y entrega al usuario, una vez este se haya logueado, la capacidad de desplazarse entre las distintas secciones de su cuenta.
8	Fragment	Representa un comportamiento o una porción de interfaz de usuario en una actividad [25]. Representa cada una de las secciones a las que un usuario puede acceder dentro de su cuenta.
9	FragmentCrearTarjetas	Es la que permite al usuario crear sus propias tarjetas. Esta contiene funciones entre otras cosas, para comunicarse con el servidor, administrar las rutas de las imágenes que el usuario seleccione desde su equipo y almacenar información en la Base de Datos local del móvil.
10	FragmentEditarTarjetas	Ofrece unas funciones muy similares a las del FragmentCrearTarjetas ayudando al usuario en

		la tarea de editar cualquiera de las tarjetas que haya creado previamente. Esta clase hace uso de una base de datos local independiente para no interferir con la información que se tiene almacenada de la tarjeta que está siendo creada en cierto momento.
11	PredisArrayAdapter	Es la clase encargada de proveer un ListView estructurado con las imágenes prediseñadas que FragmentCrearTarjetas y FragmentEditarTarjetas necesitan.
12	FragmentMisTarjetas	Contiene las funciones que permiten a un usuario interactuar con las tarjetas que ha agregado de otros usuarios, es decir que permite, con ayuda de la clase MisTarjetasArrayAdapter, visualizar las tarjetas de forma ordenada de acuerdo a un filtro dentro de un ListView, buscar dentro de su repositorio de tarjetas agregadas y acceder a las funcionalidades de llamar, enviar SMS o un correo electrónico al contacto de la tarjeta.
13	FragmentTarjetasCreadas	Junto con ListCreadasAdapter permiten al usuario interactuar con las tarjetas que ha creado previamente, presentándolas en una lista bien estructurada y proporcionando al usuario funciones para ver la tarjeta creada en pantalla completa, editarla o borrarla.
14	FragmentBuscarTarjetas	Provee una sección dentro de la aplicación para que el usuario realice una búsqueda global de tarjetas creadas por otros usuarios de acuerdo a sus preferencias y agregue a su repositorio la tarjeta que desee. Esta clase se apoya en ListBusquedaAdapter para listar de manera estructurada todas las tarjetas que cumplen con el filtro seleccionado por el usuario al momento de realizar cualquier búsqueda.
15	FragmentModificarDatos	Es la clase que brinda las funciones para modificar la información personal relacionada con la cuenta del usuario tales como su nombre y contraseña.
16	FragmentSalir	Contiene las funciones que permiten al usuario salir de su cuenta de usuario de manera segura.

Tabla 20: Especificaciones - Diagrama de Clases

8.2. DIAGRAMA DE COMPONENTES

A continuación se muestran los componentes principales del sistema y las dependencias existentes entre los mismos:

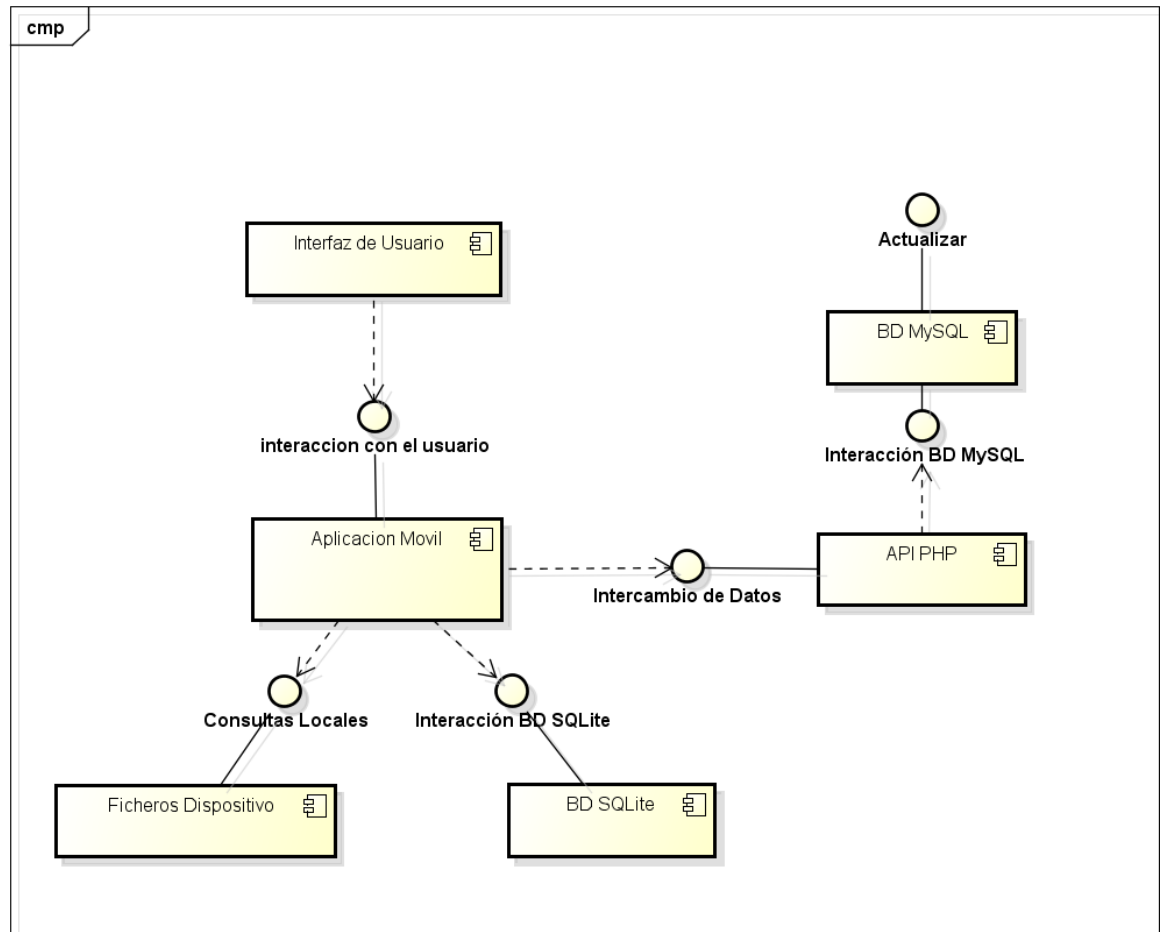


Ilustración 24: Diagrama de Componentes

La Aplicación Móvil es el componente principal del sistema, contiene todos los elementos necesarios para que un usuario pueda hacer uso de la aplicación en su dispositivo móvil.

La interfaz de usuario permite de forma amigable la interacción entre el usuario y la aplicación móvil.

El aplicativo móvil hace uso de una base de datos local SQLite para almacenar y consultar información que necesite durante los procesos que se lleven a cabo durante la ejecución de la aplicación. En procesos como la selección del logo y del

fondo la aplicación realiza consultas locales a los ficheros que se encuentran en el dispositivo.

El componente llamado API PHP alojado en el servidor, permite a la aplicación Android interactuar con la Base de Datos MySQL. Su trabajo consiste en recibir una solicitud del cliente en métodos GET/POST, interactuar con la base de datos mediante inserción o búsqueda de datos y, finalmente, dar la respuesta al cliente (Aplicación Móvil) en formato JSON.

8.3. DIAGRAMA DE ACTIVIDADES

A continuación se describe una serie de actividades que pueden ser realizadas por un usuario, así como las distintas rutas que puede tomar a medida que se mueve por las diferentes opciones de la aplicación.

I. Autenticación

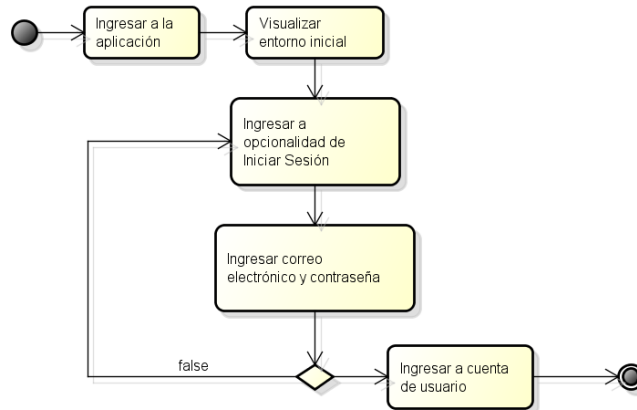


Ilustración 25: Diagrama de Actividades - Autenticación

El usuario requiere pasar por un proceso de autenticación para hacer uso de las funcionalidades de la aplicación.

II. Registrarse

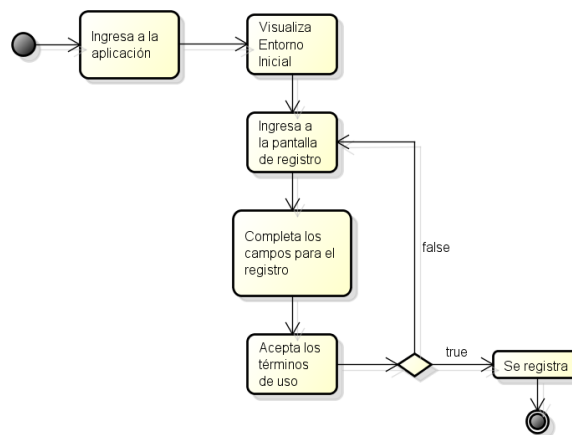


Ilustración 26: Diagrama de Actividades – Registrarse

Para registrarse, un usuario necesita ingresar a la aplicación y seleccionar la opción de Registrarse, completar los campos que son requeridos y si acepta los términos de uso queda finalmente registrado.

III. Recuperar contraseña

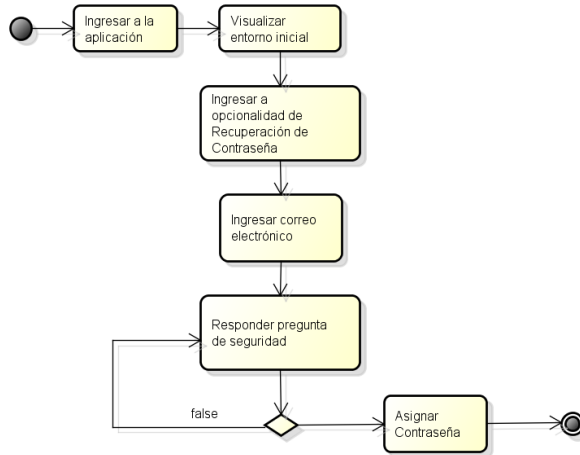


Ilustración 27: Diagrama de Actividades - Recuperar contraseña

El usuario puede recuperar su contraseña en caso de olvido. Para hacerlo, deben ingresar a la opción de recuperación de contraseña y seguir los pasos indicados.

IV. Visualizar entorno inicial

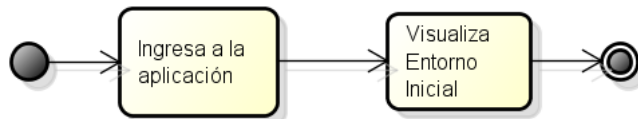


Ilustración 28: Diagrama de Actividades - Visualizar entorno inicial

Para visualizar el entorno inicial de la aplicación el usuario sólo necesita ingresar a la aplicación.

V. Editar tarjeta

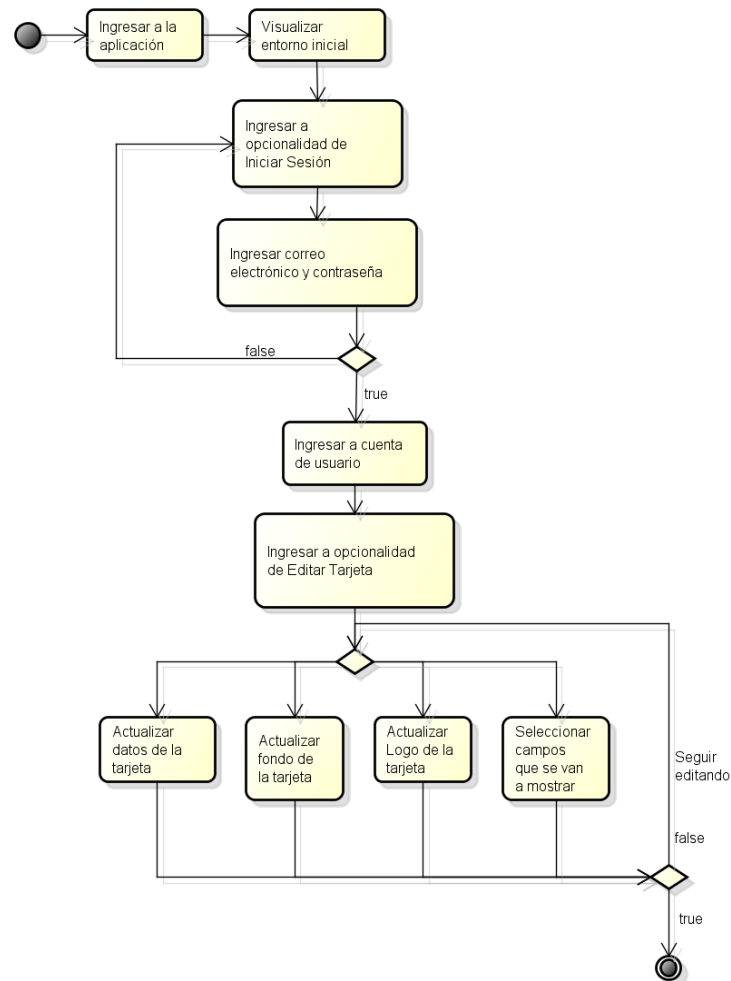


Ilustración 29: Diagrama de Actividades - Editar tarjeta

El usuario puede modificar una tarjeta creada. Para hacerlo, debe iniciar sesión y dirigirse a la opcionalidad de Tarjetas creadas, para después, decidir qué campos y qué elementos se van a actualizar en la tarjeta.

VI. Modificar datos

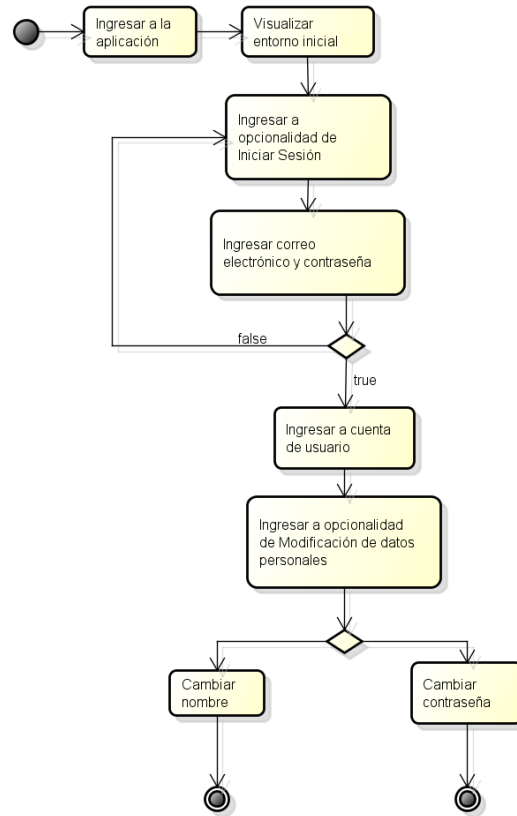


Ilustración 30: Diagrama de Actividades - Modificar datos

El usuario tiene la posibilidad de modificar sus datos personales. Para hacerlo, debe iniciar sesión y dirigirse a la opcionalidad de Modificar Mis Datos, para después, decidir si cambiar su nombre, su contraseña o ambos.

VII. Crear tarjeta

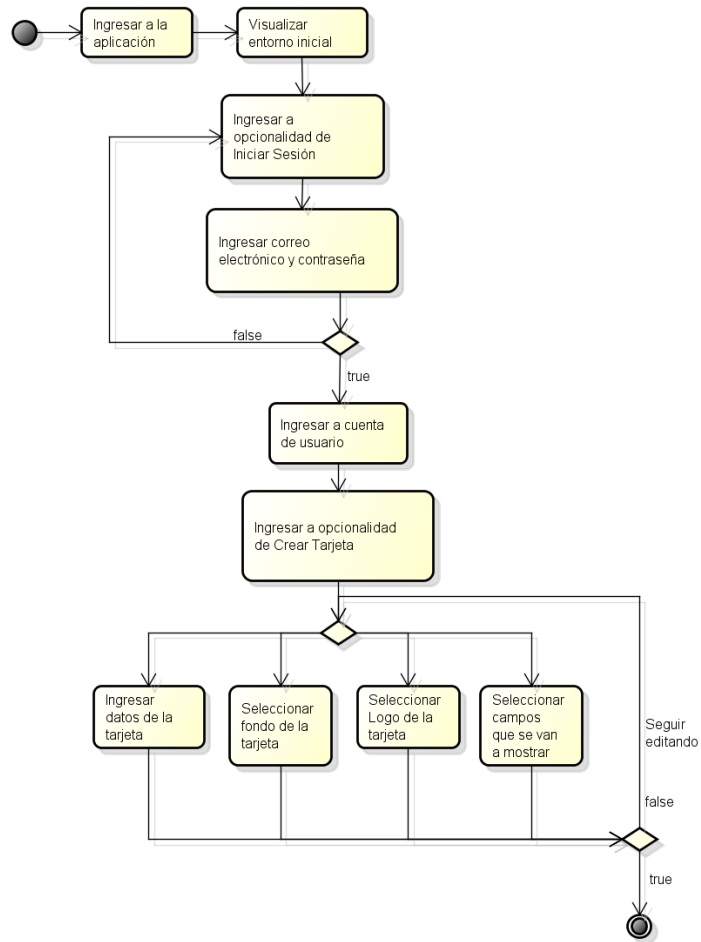


Ilustración 31: Diagrama de Actividades – Crear tarjeta

El usuario tiene la posibilidad de crear una tarjeta. Para hacerlo, debe iniciar sesión y dirigirse a la opción de Crear Tarjetas, para después, decidir qué campos y qué elementos se va a incluir en la tarjeta.

VIII. Visualizar tarjetas agregadas a la cuenta

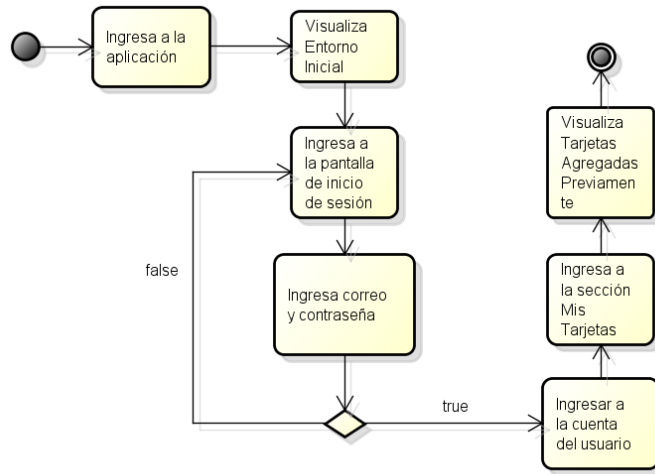


Ilustración 32: Diagrama de Actividades - Visualizar tarjetas agregadas a la cuenta

Para que un usuario pueda visualizar las tarjetas agregadas a su cuenta debe, después de iniciar sesión, seleccionar la opción “Mis Tarjetas” del menú de navegación. Con esto la aplicación le permitirá visualizar las tarjetas que ha agregado desde el repositorio global.

IX. Buscar tarjetas en cuenta de usuario

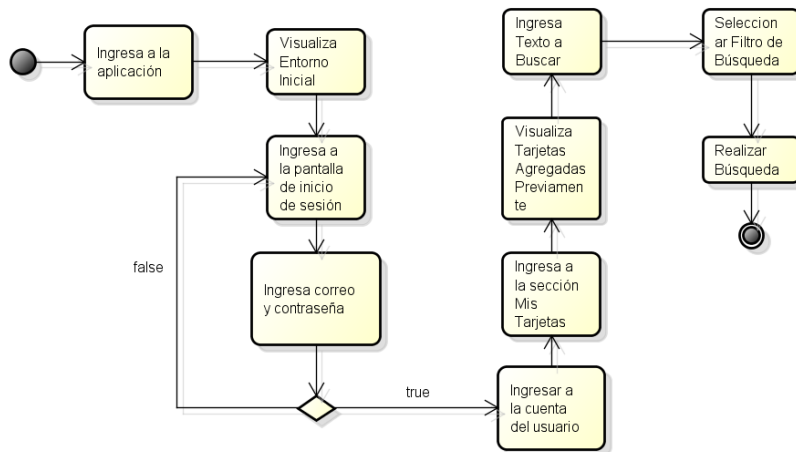


Ilustración 33: Diagrama de Actividades - Buscar tarjetas en cuenta de usuario

El usuario puede buscar alguna tarjeta en específico que ha agregado a su cuenta, dirigiéndose a la sección Mis Tarjetas y desde allí puede hacer uso del campo y filtro de búsqueda para encontrar la tarjeta que corresponde con los parámetros ingresados.

X. Buscar tarjetas en repositorio global

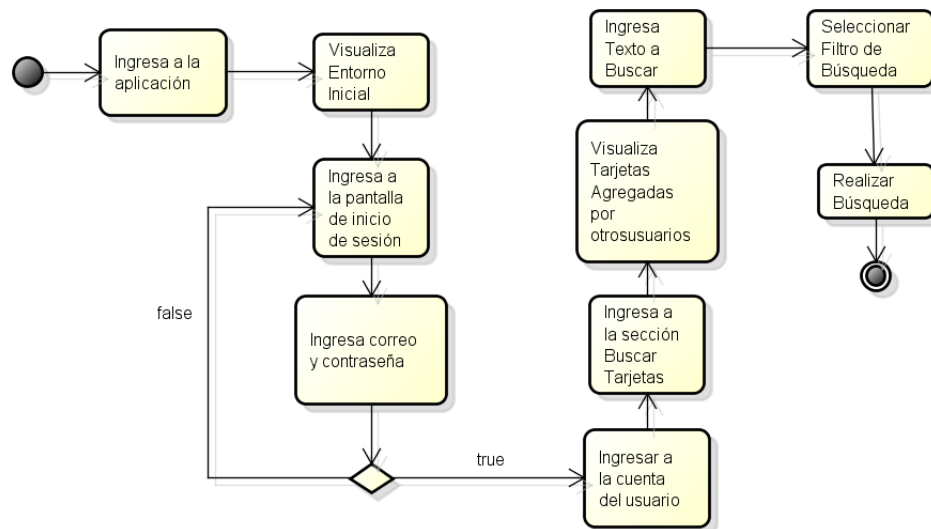


Ilustración 34: Diagrama de Actividades - Buscar tarjetas en repositorio global

Un usuario puede buscar tarjetas que otras personas han creado, ingresando a su cuenta de usuario y dirigiéndose a la sección “Buscar Tarjetas” desde aquí puede hacer uso del campo y filtro de búsqueda para encontrar cualquier tarjeta.

8.4. DIAGRAMA DE DESPLIEGUE

A continuación se muestra el diagrama de despliegue de la aplicación

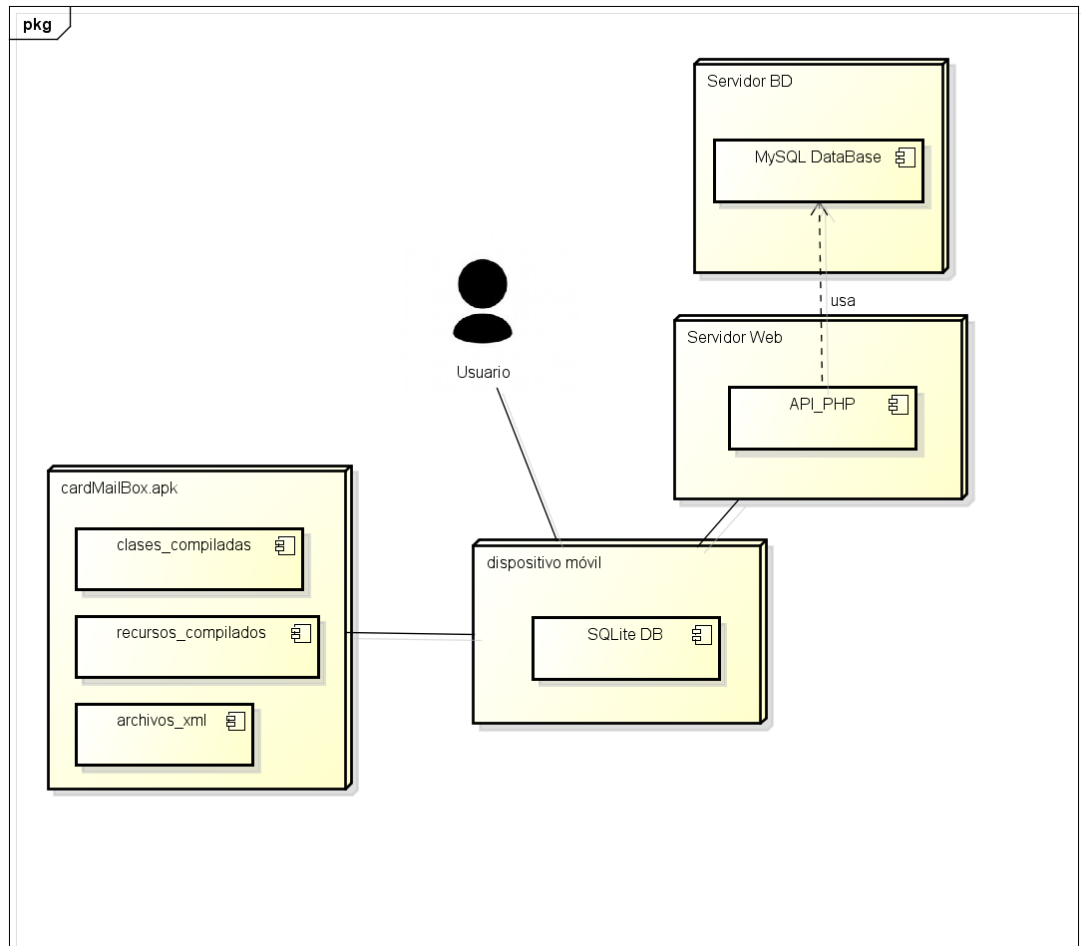


Ilustración 35: Diagrama de Despliegue

En este diagrama se puede visualizar los diferentes nodos que hacen parte de la aplicación.

El nodo dispositivo móvil, el cual contiene la base de datos local SQLite, comparte información con la API PHP que se encuentra dentro del servidor y esta a su vez interactúa con la BD MySQL ya sea para realizar consultas o ingresar nueva información proveniente del aplicativo móvil.

El nodo CardMailBox.apk contiene un conjunto de componentes que permiten a la aplicación ejecutarse en el dispositivo móvil, el cual a su vez se comunica finalmente con el usuario.

8.5. DIAGRAMA DE BASE DE DATOS

A continuación se muestra el diagrama de base de datos MySQL de la aplicación con sus tablas y sus respectivas relaciones:

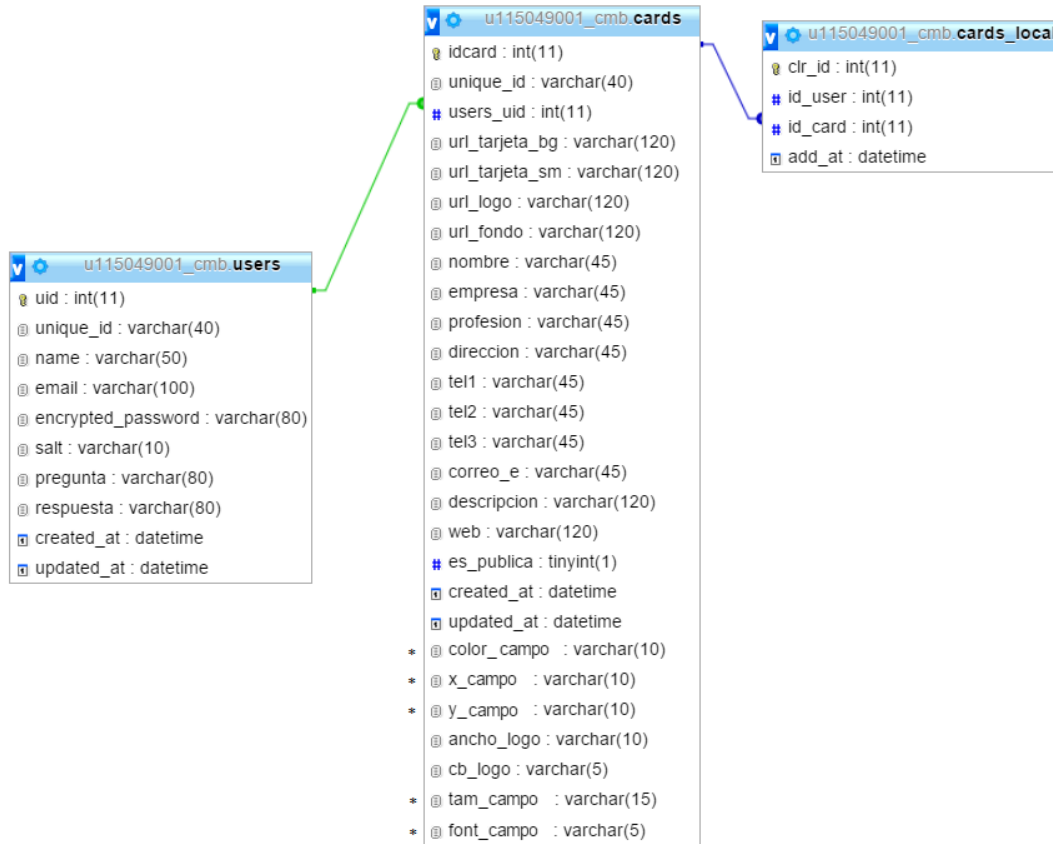


Ilustración 36: Diagrama de Base de datos

Para el correcto funcionamiento de la aplicación sólo se necesitan tres tablas:

Tabla “users”: para almacenar la información de los usuarios que se registran en la aplicación. De ellos es importante guardar su nombre que se mostrará una vez el usuario ingrese a su cuenta, su email con el que realizará el proceso de login, una contraseña encriptada y una salt (sal) usadas para garantizar la seguridad de la aplicación en cuanto al manejo de contraseñas (sha1+salt), una pregunta y una respuesta secreta usadas para recuperación de contraseña y finalmente unos campos que contienen información de la fecha en que fue creada y actualizada la cuenta del usuario. Además se guarda un uid que lleva la secuencia autoincremental de cada usuario que se registra y un unique_id que define de forma única cada uno de estos usuarios.

Tabla “cards”: para guardar información de todas las tarjetas que son creadas por los usuarios, sean públicas o no. Cuando un usuario cualquiera realiza una búsqueda de tarjetas para agregar a su cuenta, lo que hace la aplicación es dirigirse a esta tabla y buscar dentro de ella, tarjetas que correspondan a los parámetros de búsqueda que definió el usuario y que se encuentren públicas en ese momento para luego ser enlistadas en la aplicación. Los campos que se encuentran en esta tabla corresponden a lo siguiente:

- **idcard:** Lleva la secuencia incremental de cada tarjeta que ha sido creada
- **unique_id:** Es un identificador único de cada tarjeta, a diferencia de idcard no es autoincremental sino que se calcula a partir de valores random permitiendo que sea totalmente único en el tiempo de vida de la aplicación.
- **users_uid:** Es el identificador del usuario que creó la tarjeta.
- **url_tarjeta_bg:** Es la dirección web dentro del servidor donde se encuentra almacenada la tarjeta en tamaño completo (big).
- **url_tarjeta_sm:** Es la dirección web dentro del servidor donde se encuentra almacenada la tarjeta en tamaño de vista previa (small).
- **url_logo:** Es la dirección web dentro del servidor donde se encuentra almacenado el logo de una tarjeta en particular
- **url_fondo:** Es la dirección web dentro del servidor donde se encuentra almacenado el fondo de una tarjeta en particular.
- **nombre:** Campo que puede o no contener una tarjeta en particular, si el campo no es seleccionado al momento de crear la tarjeta, tendrá valor vacío en esta tabla.
- **empresa:** Campo que puede o no contener una tarjeta en particular, si el campo no es seleccionado al momento de crear la tarjeta, tendrá valor vacío en esta tabla.
- **profesion:** Campo que puede o no contener una tarjeta en particular, si el campo no es seleccionado al momento de crear la tarjeta, tendrá valor vacío en esta tabla.
- **direccion:** Campo que puede o no contener una tarjeta en particular, si el campo no es seleccionado al momento de crear la tarjeta, tendrá valor vacío en esta tabla.
- **tel1:** Campo que puede o no contener una tarjeta en particular, si el campo no es seleccionado al momento de crear la tarjeta, tendrá valor vacío en esta tabla.
- **tel2:** Campo que puede o no contener una tarjeta en particular, si el campo no es seleccionado al momento de crear la tarjeta, tendrá valor vacío en esta tabla.
- **tel3:** Campo que puede o no contener una tarjeta en particular, si el campo no es seleccionado al momento de crear la tarjeta, tendrá valor vacío en esta tabla.

- **correo_e:** Campo que puede o no contener una tarjeta en particular, si el campo no es seleccionado al momento de crear la tarjeta, tendrá valor vacío en esta tabla.
- **descripcion:** Campo que puede o no contener una tarjeta en particular, si el campo no es seleccionado al momento de crear la tarjeta, tendrá valor vacío en esta tabla.
- **web:** Campo que puede o no contener una tarjeta en particular, si el campo no es seleccionado al momento de crear la tarjeta, tendrá valor vacío en esta tabla.
- **es_publica:** Guarda un valor de 0 o 1 dependiendo si el usuario al crear la tarjeta especificó si es pública (1) o no (0)
- **created_at:** Especifica la fecha y hora en que se creó la tarjeta
- **updated_at:** Especifica la fecha y hora de la última modificación de la tarjeta
- **color_campo:** Se marca con un * ya que en la base de datos real de la aplicación no se llama color_campo sino que se tiene una columna por cada valor al que pueda cambiársele el color, es decir, en la base de datos real no se tiene un sólo campo llamado color_campo sino ocho: color_nombre, color_empresa, color_profesion, color_direccion, color_tel, color_correo, color_descripcion, color_web. Lo anterior solamente se hizo para facilitar la lectura del diagrama de base de datos, así mismo pasa con los otros atributos de la tabla que contienen la palabra “campo”.
- **x_campo, y_campo:** Estos dos campos hace referencia a las coordenadas x, y respectivamente que se almacenan de la posición de cada uno de los campos que puede contener la tarjeta. Es decir que en total, se tiene en la base de datos 16 campos que almacenan estas coordenadas.
- **ancho_logo:** Como el usuario puede aumentar o reducir el tamaño del logo en la tarjeta, este campo guarda una proporción de ese tamaño que indica qué tan grande quedó el logo dentro de la tarjeta.
- **cb_logo:** Almacena un 1 o un 0 dependiendo si el usuario decidió o no agregar un logo a la tarjeta.
- **tam_campo:** Se reemplaza en la BD real por ocho campos que almacenan el tamaño de la letra que se dejó en cada uno de los campos de la tarjeta al momento de crearla
- **font_campo:** Se reemplaza en la BD real por ocho campos que guardan la fuente de texto usada para cada uno de los campos de la tarjeta al momento de ser creada.

NOTA: Cada una de las url que se explicaron previamente contienen información del identificador único del usuario y del identificador único de la tarjeta a la que se haga alusión. Esto con el fin de que todas las imágenes relacionadas, se guarden de forma ordenada dentro del servidor.

La última tabla, llamada `cards_local` guarda información de las tarjetas que cada usuario agrega a su propia cuenta, por ende los campos que necesita son los siguientes:

- **clr_id:** Es un valor entero autoincremental que identifica cada relación usuario-tarjeta
- **id_user:** Es el identificador del usuario que tiene tarjetas almacenadas en su cuenta
- **id_card:** Es el identificador de la tarjeta del usuario que creó la tarjeta
- **add_at:** Guarda la fecha y hora en que el usuario agregó la tarjeta a su cuenta

9. PRUEBAS DE SOFTWARE

9.1. PLAN DE PRUEBAS

El plan de pruebas se basa en la norma “IEEE SDT 829”⁴ destinado a las pruebas de software, a continuación se definen los puntos que intervienen en el desarrollo del plan de pruebas del presente proyecto.

REVISIÓN HISTÓRICA

Fecha	Versión	Descripción	Autor
17/06/2015	1.0	Versión Inicial	Andrés Giraldo Carvajal
18/06/2015	2.0	Versión Final	Carlos A. Londoño Sánchez

Tabla 21: Revisión histórica - Plan de Pruebas

9.1.1. Introducción

Al momento de elaborar un proyecto de software, es de vital importancia construir un documento que muestre cada uno de los pasos que se tendrán en cuenta para el desarrollo de las pruebas del software. Allí se detallan los módulos de la aplicación, interfaces, características mínimas de software y hardware, el funcionamiento de la aplicación y un plan de contingencia para tratamiento de errores.

9.1.2. Alcance

Para un correcto funcionamiento del aplicativo móvil se debe probar y garantizar que los siguientes módulos clave de la aplicación trabajen adecuadamente:

- Módulo Registro: Permite a una persona convertirse en usuario de la aplicación, creando una cuenta con sus datos personales.
- Módulo Login: Permite el logueo de un usuario en la aplicación, ingresando su correo y contraseña.
- Módulo Creación de Tarjetas: La aplicación debe permitir a un usuario, crear tarjetas de presentación, ofreciendo una gama de posibilidades para personalizarla de acuerdo a sus gustos o necesidades.
- Módulo Actualizar Tarjetas: Permite editar cualquier información o elemento de una tarjeta en específico que el mismo usuario haya creado y que estos cambios se evidencien una vez finalice el proceso de edición.
- Módulo Buscar/Intercambiar Tarjetas: A través de este módulo el usuario puede compartir o publicar sus tarjetas de presentación y a la vez buscar de

⁴ IEEE SDT 829 es un estándar IEEE para la documentación de las pruebas de software

acuerdo a sus necesidades una tarjeta que corresponda con sus criterios de búsqueda.

9.1.3. Requerimientos de las pruebas

A continuación, se identifica los elementos (casos de uso, requisitos funcionales y no funcionales) que son objetivos de las pruebas, es decir, los elementos que vamos a probar.

Pruebas de integridad de la base de datos y de los datos

- Verificar el acceso a la Base de Datos
- Verificar la recuperación correcta de las modificaciones realizadas en la Base de Datos

Pruebas de funcionalidad

- Verificar el Caso de Uso Registrarse
- Verificar el Caso de Uso Autenticarse
- Verificar el Caso de Uso Crear Tarjetas
- Verificar el Caso de Uso Editar Tarjetas
- Verificar el Caso de Uso Buscar Tarjetas en el Repositorio Global

Pruebas de interfaz de usuario

- Verificar la facilidad de navegación a través de los diferentes casos de uso y las pantallas que presenta la aplicación.

9.1.4. Estrategia de prueba

A continuación se presenta la estrategia para probar el sistema y la manera en cómo se llevaran a cabo las pruebas.

9.1.5. Tipos de pruebas y técnicas

9.1.5.1. Pruebas de integridad de la base de datos y de los datos

Objetivos de la prueba	Comprobar que los métodos de acceso a la base de datos funcionan debidamente.
Técnicas	Invocar los métodos de acceso a la base de datos con datos válidos e inválidos.
Criterios de finalización	Cada uno de los métodos de acceso trabajan correctamente y no se presentan errores en los datos.
Consideraciones	El uso de un DBMS al momento de realizar la prueba permitirá recuperar o actualizar los datos en la BD de forma directa.

Tabla 22: Pruebas de integridad

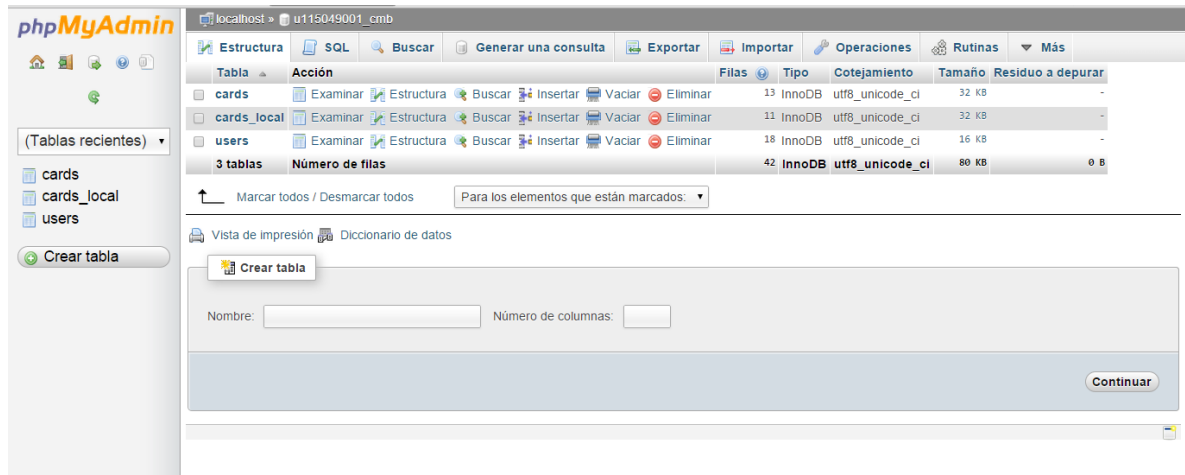


Ilustración 37: Interfaz phpMyAdmin

9.1.5.2. Pruebas de funcionalidad

Esta prueba tiene como objetivo verificar la aceptación, procesamiento y recuperación de datos y la adecuada implementación de los mismos. Se basa en verificar la aplicación interaccionando con las interfaces de usuario y analizando los resultados.

Objetivos de la prueba	Verificar que la navegación a través de los diferentes casos de uso en la aplicación se haga de forma correcta, la entrada de los datos, el procesamiento y recuperación de estos.
Técnicas	Ejecutar cada caso de uso y su respectivo flujo con datos válidos e inválidos para verificar: <ul style="list-style-type: none">• Que cuando se usan datos correctos, se obtienen los resultados esperados.• Que cuando se usan datos incorrectos se obtienen los mensajes de advertencia adecuados.
Criterios de finalización	Se ejecutaron todas las pruebas planificadas.
Consideraciones	Ninguna

Ilustración 38: Pruebas de funcionalidad

Se realizó pruebas a los módulos clave de la aplicación, los cuales se encuentran accesibles fácilmente en la aplicación:

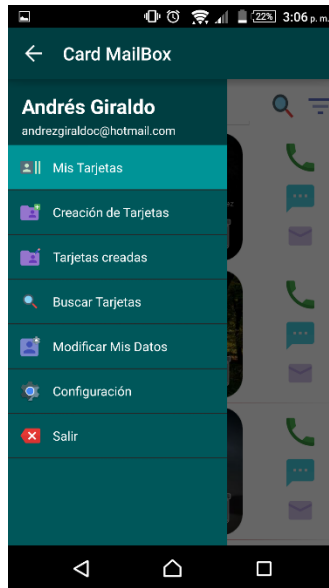


Ilustración 39: Módulos clave de la aplicación

9.1.5.3. Pruebas de interfaz de usuario

Esta prueba verifica la interacción del usuario con la aplicación.

Objetivos de la prueba	Verificar que la interfaz de usuario de la aplicación permita al usuario acceder y navegar a través de toda la funcionalidad de la aplicación.
Técnicas	Crear pruebas para cada módulo con el objetivo de verificar la correcta navegación y su estado
Criterios de finalización	Cada módulo se ha verificado con éxito
Consideraciones	Ninguna

Tabla 23: Pruebas de interfaz de usuario

9.2. MODELO DE PRUEBAS

CASO DE USO: Registrarse		
CASO DE PRUEBA: Registrar un usuario nuevo		
Entrada	Resultados	Condiciones
El usuario procede a llenar los campos obligatorios para el registro.	El sistema registra adecuadamente los datos en la Base de Datos, incluyendo un correo y una contraseña.	El usuario que procede a registrarse no puede estar registrado con anterioridad.

CASO DE USO: Autenticarse		
CASO DE PRUEBA: Realizar Login		
Entrada	Resultados	Condiciones
El usuario procede a ingresar sus datos en los campos correspondientes.	El sistema permite autenticar los usuarios de manera adecuada, con los respectivos mensajes de error cuando el usuario no procede de manera correcta.	Solo un usuario registrado puede iniciar sesión. El usuario no registrado que intente iniciar sesión, no pasará el proceso de verificación de datos.

CASO DE USO: Crear Tarjetas		
CASO DE PRUEBA: Crear una tarjeta		
Entrada	Resultados	Condiciones
El usuario ingresa todos los datos requeridos por el sistema para la creación de la tarjeta.	El sistema valida correctamente los datos ingresados y permite crear tarjetas adecuadamente	El usuario debe haber iniciado sesión correctamente

CASO DE USO: Editar Tarjetas		
CASO DE PRUEBA: Actualizar información de una tarjeta		
Entrada	Resultados	Condiciones
El usuario edita los campos que requiera actualizar de la tarjeta seleccionada	El usuario permite al usuario editar tarjetas creadas con anterioridad de forma adecuada, con la posibilidad de hacer o no públicos los cambios.	<ul style="list-style-type: none"> • El usuario debe haber iniciado sesión correctamente. • La tarjeta a modificar, debe haber sido creada con anterioridad

CASO DE USO: Buscar Tarjetas en el Repositorio Global		
CASO DE PRUEBA: Buscar una tarjeta específica de otro usuario		
Entrada	Resultados	Condiciones
El usuario ingresa el texto que deberá coincidir con el de la tarjeta que busca y selecciona un filtro de búsqueda	El sistema muestra correctamente todas las coincidencias relacionadas con el patrón de búsqueda definido por el usuario, además, el usuario puede agregar una o varias de esas tarjetas que se enlistan.	El usuario debe haber iniciado sesión correctamente

Tabla 24: Modelo de Pruebas

10. DESARROLLO DEL APLICATIVO

El desarrollo del aplicativo se hizo a partir de una estructura capaz de modularizar los procesos principales que posee. Se dividieron tareas y procesos, de tal manera que la aplicación nativa de Android fuese totalmente independiente de las tareas y procesos que se comunican con el servidor y la base de datos.

A continuación se describe gráficamente los bloques principales de la aplicación:

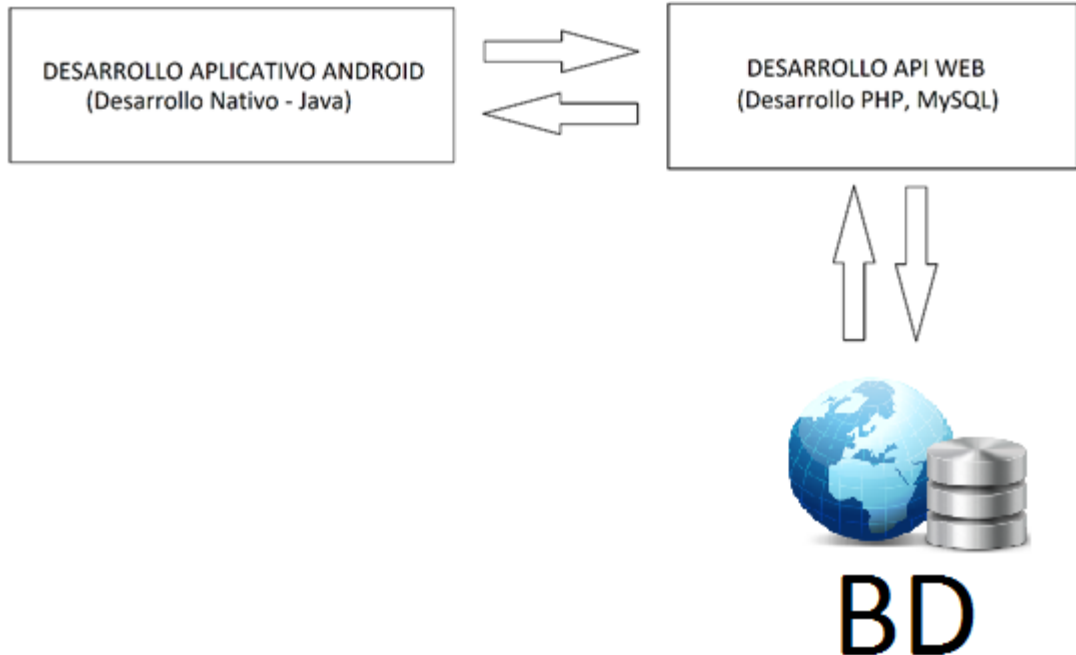


Ilustración 40: Bloques de desarrollo de la aplicación

Estos dos bloques conforman el desarrollo de toda la aplicación, y aunque cada bloque tiene tareas y procesos diferentes, entre ellas es obligatoria una comunicación correcta, estable y totalmente funcional.

Para lograr esto, se implementa un protocolo de comunicación a través de mensajes JSON (JSONObject y JSONArray), los cuales permiten el paso de mensajes a través de las dos plataformas de desarrollos.

Estos dos bloques de desarrollo trabajan conjuntamente con el fin de dar una correcta administración a la BD y al servidor.

Aplicativo Android

Este bloque contiene todo lo referente al desarrollo del aplicativo Android. Este desarrollo se hizo de forma nativa haciendo uso de:

- Lenguaje de programación Java.
- Lenguaje de marcas XML.
- Ambiente de desarrollo integrado (IDE), Android Studio.

Este bloque se encarga de agrupar el desarrollo de todas las funcionalidades que corren sobre el dispositivo. (Actividades, fragmentos, Layouts, vistas, etc). Cada una de estas funcionalidades trabaja directamente sobre el dispositivo y cumplen la tarea de mostrar al usuario todas las opciones de la aplicación, haciendo uso de menús, barras de navegación, vistas, bases de datos locales, etc.

El desarrollo sobre Android tiene unos pequeños módulos encargados de establecer la comunicación con el segundo bloque de desarrollo. Estos módulos usan protocolos HTTP e hilos con el fin de poder establecer la comunicación sin ser bloqueante y evitando el uso excesivo de recursos.

Api WEB

Este bloque de desarrollo se encarga de todas las tareas y procesos referentes a la comunicación directa con el servidor y con la base de datos. Cuando el aplicativo android requiere comunicarse con el servidor, esta Api web se encarga de funcionar como intermediario. Cada consulta y cada orden pasa por un “filtro” en la Api web, la cual determina como comunicarse y que comunicar hacia el servidor.

El desarrollo de este bloque se logró haciendo uso de:

- Lenguaje de programación PHP
- Desarrollo sobre editor de texto Sublime
- Sistema Gestor de Base de Datos Relacional MySQL

Esta Api WEB, a su vez también se divide en tres módulos:

- **INDEX:** encargado de la comunicación directa con la aplicación Android a través de mensajes JSON haciendo uso de métodos GET/POST
- **FUNCIONES:** encargado de las funciones que se comunican con la base de datos y el servidor.
- **CONFIGURACIÓN:** encargado de las configuraciones de conexión hacia la base de datos.

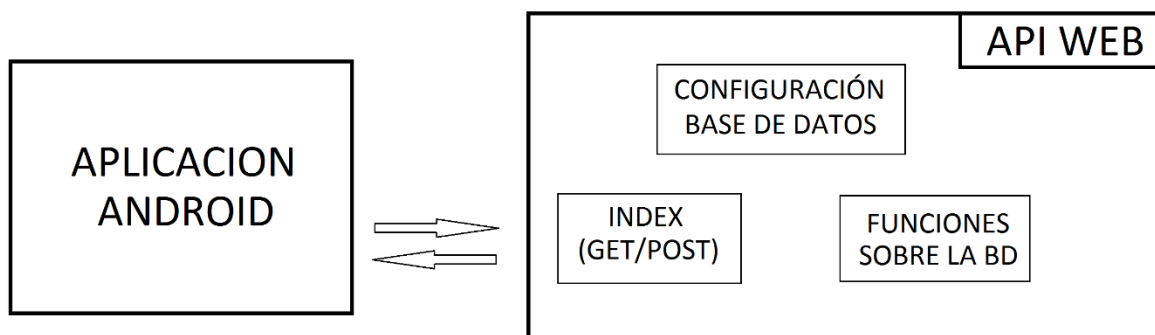


Ilustración 41: Módulos - Api WEB

10.1. LÍMITES Y PROYECCIONES RESPECTO AL DESARROLLO

Los límites respecto al desarrollo del presente proyecto se establecen a continuación:

- La aplicación a desarrollar sólo se ejecutará correctamente en Smartphones que cuenten con Sistema Operativo Android v 4.0 Ice Cream.
- La aplicación se desarrollará haciendo uso de una única gama de colores y por el momento el usuario no podrá cambiar estilos o colores.
- La aplicación a desarrollar se presentará en idioma español, el usuario por el momento no podrá seleccionar un idioma diferente a este.
- La aplicación que se desarrollará en el presente proyecto se ejecuta de forma correcta en dispositivos con tamaño de 4 pulgadas en adelante.

Las proyecciones respecto al desarrollo del presente proyecto se establecen a continuación:

- Implementación de la aplicación sobre Sistemas Operativos iOS y Windows Phone haciendo uso de un desarrollo nativo.
- Mejoramiento de la presentación gráfica de la aplicación haciendo uso de:
 - Más gama de colores
 - Iconos más amigables
 - Material Design
 - Animaciones
 - ListView con más interactividad
- Incluir otros idiomas en la aplicación, de manera que el usuario pueda seleccionar el más adecuado de acuerdo al lenguaje con el que se sienta más cómodo, además esto extiende las fronteras geográficas de la aplicación.
- Incluir dispositivos con un tamaño inferior a 4 pulgadas y evaluar la posibilidad de migrar la aplicación a otro tipo de dispositivos como SmartWatch.

10.2. ESTRUCTURA DE CODIFICACIÓN

10.2.1. Android

El desarrollo de la aplicación se hizo de forma nativa haciendo uso de programación orientada a objetos con lenguaje de programación Java y un lenguaje de marcas XML. Para una estructura de desarrollo organizada, se dividió el desarrollo en 3 módulos principales:

- **Java:** Contiene todas las actividades, fragmentos y clases del aplicativo. En resumen, contiene todo el desarrollo hecho sobre lenguaje Java.
- **Res:** Contiene todos los recursos multimedia de la aplicación, y además la especificación de todas los Layouts y vistas desarrollados en XML.
- **AndroidManifest:** Se encarga de administrar todos los permisos que posee la aplicación al momento de correr sobre un dispositivo.

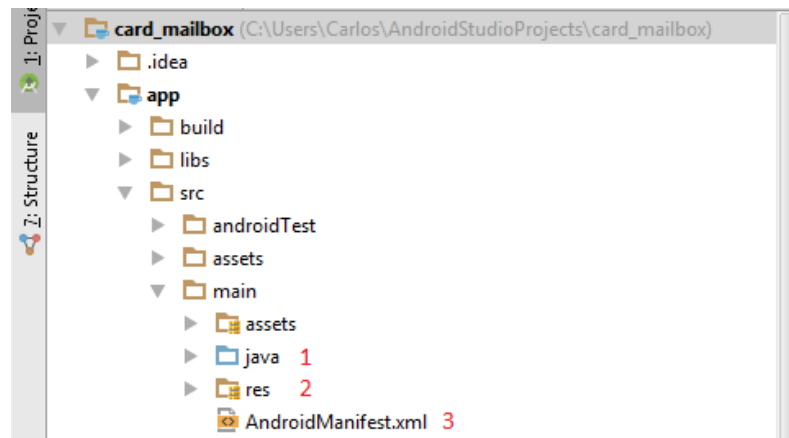


Ilustración 42: Módulos - Desarrollo Java, XML

10.2.2. Api WEB

El desarrollo de la Api WEB se hizo con lenguaje de programación PHP. Para una estructura de desarrollo organizada, se dividió el desarrollo en 3 módulos principales, los cuales están especificados al inicio del capítulo:

- Archivos de configuración
- Funciones
- Index (GET/POST)

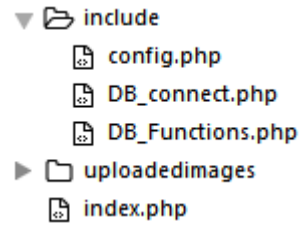


Ilustración 43: Módulos - Api WEB

10.3. ESPECIFICACION DE ACTIVIDADES, CLASES Y FUNCIONALIDADES DEL APLICATIVO ANDROID

Módulo de desarrollo Java:

El primer módulo Java está distribuido en paquetes o “Packages” con el fin de organizar tareas en diferentes lugares y evitar confusión de archivos. Cada uno de estos paquetes tiene una finalidad distinta:

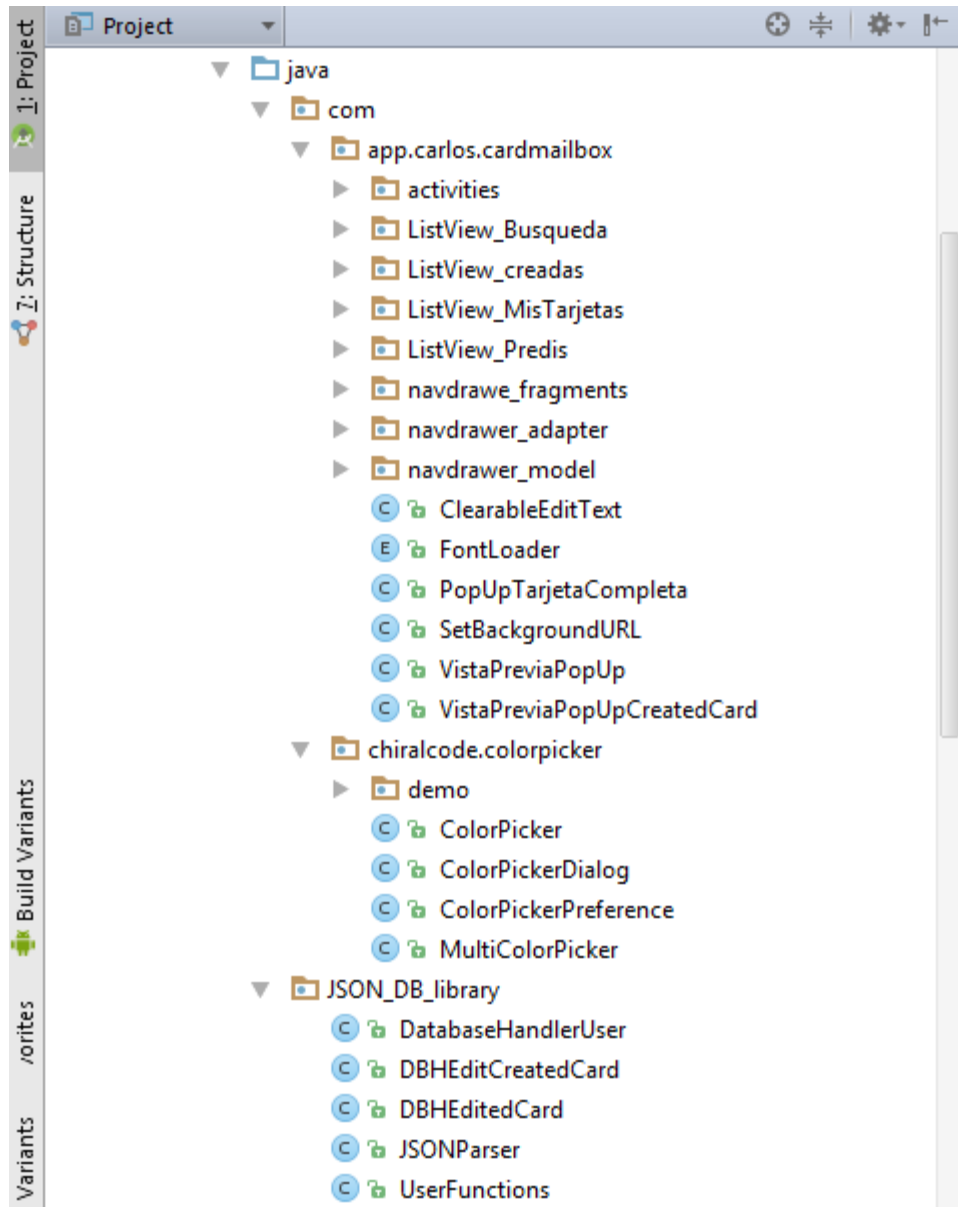


Ilustración 44: Paquetes - Desarrollo Java

- **Activities:** Conjunto de clases que conforman la base de la aplicación móvil. Estas actividades se encargan de la interacción directa con el usuario, y se caracterizan por ser las pantallas con las que el usuario interactúa mientras hace uso de la aplicación. Cada una de las clases que la conforma, se especifica en el diagrama de clases del Capítulo “Diseño y Arquitectura de Software”.

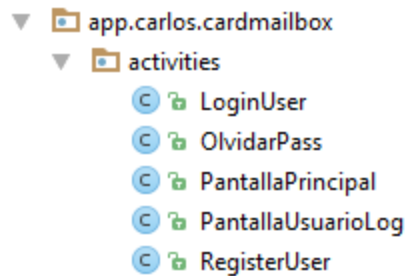


Ilustración 45: activities, Desarrollo Java

- **ListView_Busqueda, ListView_creadas, ListView_MisTarjetas, ListView_Predis:** Conjunto de clases que se encarga de enlistar un grupo de vistas determinadas, permitiéndole al usuario desplazarse por cada uno de los elementos y en algunos casos, interactuar con la aplicación a través de botones, campos de texto, etc.

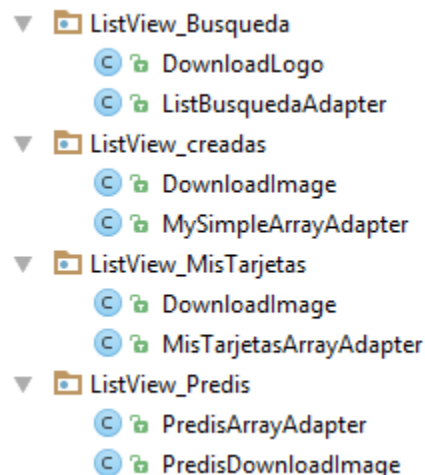


Ilustración 46: listviews, Desarrollo Java

Cada ListView requiere una clase Adapter que permita especificar qué vistas contiene cada elemento de la lista que se está mostrando. Cada lista está mostrando imágenes ubicadas en el servidor, y por ende, requiere de una clase que se encargue de manera asíncrona (Hilo), de descargar la imagen y mostrarla en el elemento de la lista.

- **Navdrawe_fragments:** Conjunto de clases encargadas de la administración de los fragmentos de la aplicación. Cada fragmento, es un espacio independiente en pantalla, pero que internamente, trabajan sobre la misma actividad. Cada uno de estos fragmentos, conforman las opciones principales a las que el usuario tiene acceso cuando esta logueado. Las especificaciones de estas clases, están en el diagrama de clases del capítulo “Diseño y Arquitectura de Software”.

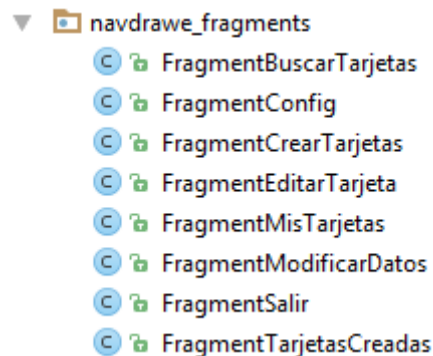


Ilustración 47: fragments, Desarrollo Java

- **Navdrawer_adapter, Nav_drawer_model:** Conjunto de clases encargado de la creación del Navigation Drawer de la aplicación. Permite visualizar las opciones de la aplicación a través de un menú lateral corredizo de izquierda a derecha. Cada una de las opciones del Navigation Drawer, instancia un fragmento según la opción escogida por el usuario.

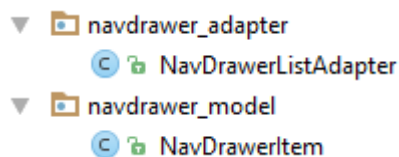


Ilustración 48: navdrawer, Desarrollo Java

- **Colorpicker:** Conjunto de clases encargadas de mostrar un selector de color interactivo en el que el usuario puede escoger un color al momento de editar campos de texto de la tarjeta.

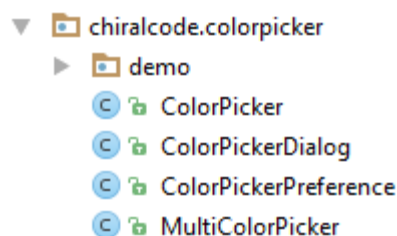


Ilustración 49: Colorpicker, Paquete



Ilustración 50: Colorpicker, Vista Previa

- **JSON_DB_Library:** Conjunto de clases que se encarga de administrar el intercambio de mensajes con la Api WEB a través de mensajes JSONObject y JSONArray. Para lograr este intercambio de mensajes, se hace uso de protocolos de comunicación HTTP. La clase “UserFunctions”, se encarga de definir las funcionalidades que requieren comunicarse con la Api WEB, por ejemplo: login, registro, crear tarjeta, añadir tarjeta, etc. La clase “JSONParser” se encarga de establecer la comunicación con la Api WEB y realizar la función establecida en “UserFunctions”.

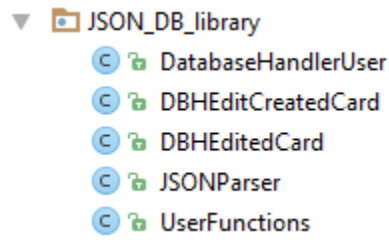


Ilustración 51: JSON Library, Desarrollo Java

Las 3 clases restantes son las que se encargan de crear y administrar las bases de datos locales del dispositivo:

- **DatabaseHandlerUser:** Base de datos local encargada de guardar los datos del usuario logueado con el fin de no tener que ingresar credenciales cada vez que quiera hacer uso de la aplicación.
- **DBHEditCreatedCard:** Base de datos local encargada de guardar los datos de una tarjeta ya creada y que se encuentre en modificación. Esto, con el fin de que el usuario pueda pausar la edición y continuarla en cualquier momento sin perder los cambios realizados sobre la tarjeta.
- **DBHEditedCard:** Base de datos local encarga de guardar los datos de una tarjeta que esté en medio de creación. Esto, con el fin de que el usuario no pierda la información ingresada durante la creación de la tarjeta si decide pausar o apagar el dispositivo.

Módulo de desarrollo XML

El desarrollo Java se complementa con todos los archivos XML que se encargan de definir la estructura gráfica de la aplicación. Cada uno de estos archivos, define vistas y componentes que conforman las pantallas de la aplicación. A continuación se listan los Layouts (Archivos XML) usados en la aplicación:

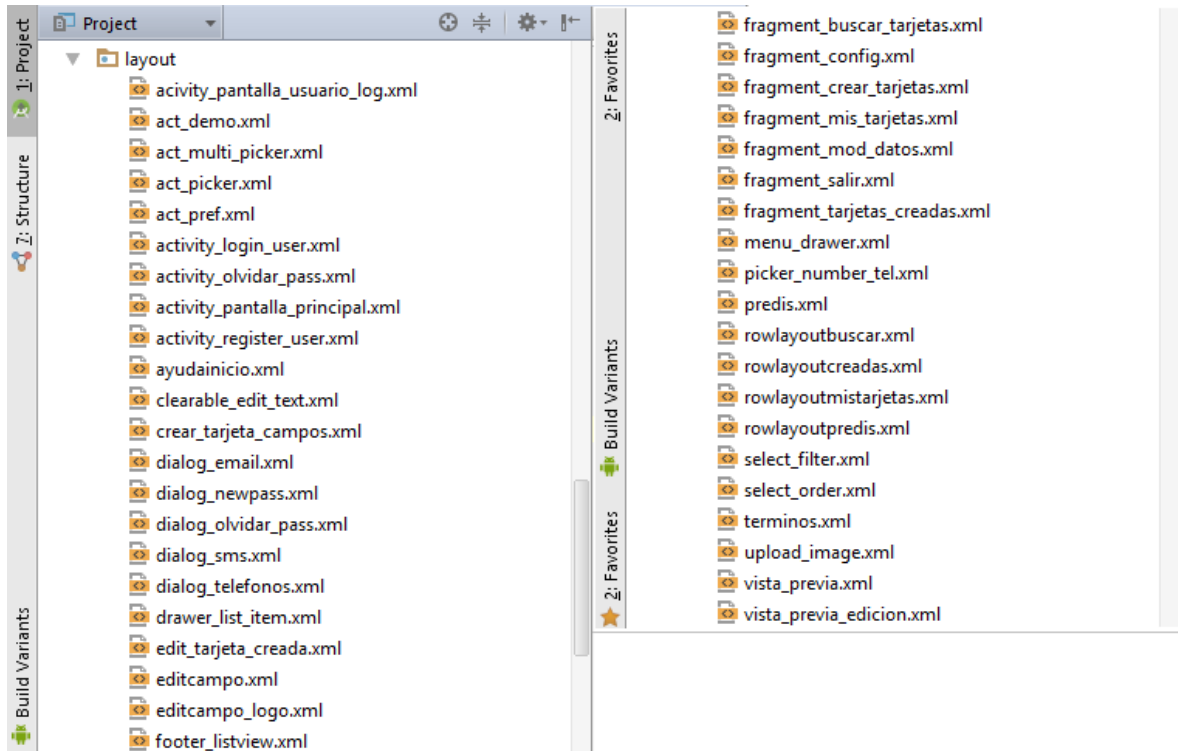


Ilustración 52: Layouts - Desarrollo XML

Cada uno de estos Layouts son instanciados desde las clases definidas anteriormente. Cuando una clase requiere mostrar datos al usuario, o requiere interactuar con el usuario, es necesario que se instancie un Layout donde se muestren, a través de vistas, respuestas por parte de la aplicación.

Si una clase en específico requiere que el usuario ingrese un dato o una orden, también se hace necesario instanciar un Layout en el que se definan botones, campos de texto, etc. Al momento de instanciar el Layout, la clase tiene dominio de todas las vistas incluidas en él y a su vez, la capacidad de obtener la información o la orden dada por el usuario.

A continuación se especifican todos los archivos XML de la aplicación y sus funcionalidades:

Nro.	Archivo XML	Función
1	activity_pantalla_usuario_log	Estructura gráfica de la actividad PantallaUsuarioLog. Este Layout es instanciado desde la clase.
2	activity_login_user	Estructura gráfica de la actividad LoginUser. Este Layout es instanciado desde la clase.
3	activity_olvidar_pass	Estructura gráfica de la actividad OlvidarPass. Este Layout es instanciado desde la clase.
4	activity_pantalla_principal	Estructura gráfica de la actividad PantallaPrincipal. Este Layout es instanciado desde la clase.
5	activity_register_user	Estructura gráfica de la actividad RegisterUser. Este Layout es instanciado desde la clase.
6	ayudainicio	PopUp con detalles de ayuda para el usuario. Se activa con el botón de ayuda que está en la Pantalla Principal de la aplicación
7	clearable_edit_text	Estructura gráfica para un campo de texto con la opción de borrar su contenido a través del botón "X".
8	crear_tarjeta_campos	PopUp con campos de texto y selección de Logo para la tarjeta que está en creación o edición.
9	dialog_newpass	PopUp para la asignación de una nueva contraseña en la funcionalidad de recuperación de contraseña.
10	dialog_olvidar_pass	Pantalla para ingresar y visualizar la pregunta y respuesta secreta al momento de recuperar contraseña.
11	dialog_email	PopUp encargado de listar los correos de tarjetas agregadas a Mis Tarjetas
12	dialog_sms	PopUp encargado de listar los teléfonos de tarjetas agregadas a Mis Tarjetas y permitir el envío de SMS.
13	dialog_telefonos	PopUp encargado de listar los teléfonos de tarjetas agregadas a Mis Tarjetas y permitir realizar llamadas.

14	drawer_list_item	Estructura gráfica encargada de contener los diferentes fragmentos de la aplicación.
15	edit_tarjeta_creada	Pantalla encargada de mostrar las opciones de edición de tarjetas ya creadas.
16	editcampo	PopUp que se visualiza al momento de tocar un campo de la tarjeta en medio de edición.
17	editcampo_logo	PopUp que se visualiza al momento de tocar el logo de la tarjeta en medio de edición.
18	footer_listview	Encabezado del Navigation Drawer que se encarga de mostrar el nombre y el correo de la persona logueada
19	fragment_buscar_tarjetas	Estructura gráfica de la Clase FragmentBuscarTarjetas. Este Layout es instanciado desde la clase.
20	fragment_config	Estructura gráfica de la Clase FragmentConfig. Este Layout es instanciado desde la clase.
21	fragment_crear_tarjetas	Estructura gráfica de la Clase FragmentCrearTarjetas. Este Layout es instanciado desde la clase.
22	fragment_mis_tarjetas	Estructura gráfica de la Clase FragmentMisTarjetas. Este Layout es instanciado desde la clase.
23	fragment_mod_datos	Estructura gráfica de la Clase FragmentModificarDatos. Este Layout es instanciado desde la clase.
24	fragment_salir	Estructura gráfica de la Clase FragmentSalir. Este Layout es instanciado desde la clase.
25	fragment_tarjetas_creadas	Estructura gráfica de la Clase FragmentBuscarTarjetas. Este Layout es instanciado desde la clase.
26	menu_drawer	Encargado de mostrar el menú lateral del Navigation Drawer. Contiene un ListView con las opciones de la aplicación.
27	picker_number_tel	PopUp que le permite al usuario seleccionar el número de teléfonos en la tarjeta a crear o editar.

28	rowlayoutbuscar	Especifica el contenido de cada elemento de la lista ListView_Busqueda
29	rowlayoutcreadas	Especifica el contenido de cada elemento de la lista ListView_creadas
30	rowlayoutmistarjetas	Especifica el contenido de cada elemento de la lista ListView_MisTarjetas
31	rowlayoutpredis	Especifica el contenido de cada elemento de la lista ListView_Predis
32	select_filter	PopUp que le permite al usuario elegir un filtro de búsqueda en Mis Tarjetas y en Buscar Tarjetas.
33	select_order	PopUp que le permite al usuario elegir un método de ordenamiento en Mis Tarjetas y Buscar Tarjetas.
34	terminos	PopUp encargado de mostrarle al usuario los términos y condiciones de uso.
35	upload_image	PopUp que se muestra al momento de seleccionar Subir Imagen en la creación y edición de tarjetas.
36	vista_previa	PopUp Full Screen encargado de mostrar la tarjeta seleccionada en pantalla completa.
37	vista_previa_edicion	PopUp Full Screen que contiene todas las opciones y funcionalidades para la creación y edición de tarjetas.

Tabla 25: Funcionalidades de los Layouts de la Aplicación

10.4. ESPECIFICACIÓN DE CLASES Y ADMINISTRACIÓN DE LA BASE DE DATOS EN LA API WEB

Módulo de configuración:

Para lograr la conexión entre la Api WEB y la Base de Datos, fue necesario crear un módulo que definiera los parámetros de conexión. Este módulo está compuesto por dos archivos:

- config.php
- connect.php

El archivo config.php, define los parámetros de uso para la base de datos:

- Nombre del servidor de la base de datos
- Usuario
- Contraseña
- Nombre de la base de datos.

El archivo connect.php, establece los detalles de conexión a la base de datos:

- Apertura de la conexión con los parámetros establecidos en config.php
- Establece el tipo de comandos a utilizar entre el código para las consultas a la base de datos, a través de `mysql_connect()`
- Selecciona la base de datos para ser usada en las funciones de la Api WEB.
- Establece la funcionalidad para cerrar la conexión a la base de datos.

Módulo index:

Como se ha explicado anteriormente, este módulo se encarga de la comunicación entre la aplicación Android y la Api WEB. Para lograr esto, se hace uso de una variable llamada "tag", la cual le indica al módulo qué tipo de función está requiriendo la aplicación Android. La variable "tag" es una cadena de texto, la cual es leída como un mensaje JSON proveniente de la aplicación Android, y después de ser filtrada, se procede a llamar al método específico del módulo de funciones. Esta variable puede contener los siguientes valores dependiendo de la función a implementar.

Nro.	"tag"	Funcionalidad	Retorna
1	"login"	Llamar a la función encargada del ingreso de usuario en el módulo de funciones.	Los datos del usuario que acaba de ingresar y una respuesta de éxito o error.
2	"register"	Llamar a la función encargada del registro de usuario en el módulo de funciones.	Los datos del usuario y una respuesta de éxito o error.
3	"modifdatos"	Llamar a la función encargada de la actualización del nombre del usuario en el módulo de funciones.	Una respuesta de éxito o error.
4	"modifpass"	Llamar a la función encargada de la actualización de la contraseña del usuario en el módulo de funciones.	Una respuesta de éxito o error.
5	"modifpass2"	Llamar a la función encargada de la actualización de la contraseña del usuario en caso de recuperación de contraseña, en el módulo de funciones.	Una respuesta de éxito o error.
6	"create_card"	Llamar a la función encargada de la creación de tarjetas en el módulo de funciones.	Id único de tarjeta y una respuesta de éxito o error.
7	"update_card"	Llamar a la función encargada de la actualización de tarjetas en el módulo de funciones.	Id único de tarjeta y una respuesta de éxito o error.
8	"getcards"	Llamar a la función encargada de retornar todas las tarjetas a nombre de un id de usuario determinado, en el módulo de funciones.	Todas las tarjetas pertenecientes al id de usuario recibido.
9	"searchcards"	Llamar a la función encargada de la búsqueda de tarjetas a través de una	Las tarjetas que coinciden con la palabra, el filtro y que sean públicas.

		palabra y un filtro, en el módulo de funciones.	
10	"addcard"	Llamar a la función encargada de la adición de tarjetas en el módulo de funciones.	Una respuesta de éxito o error.
11	"mycards"	Llamar a la función encargada de retornar las tarjetas agregadas en el módulo de funciones.	Todas las tarjetas agregadas por el usuario.
12	"forgetpass"	Llamar a la función encargada de la recuperación de contraseña en el módulo de funciones.	Una respuesta secreta para ser validada en la aplicación Android.
13	"borrarmitarjeta"	Llamar a la función encargada de la eliminación de tarjetas de Mis Tarjetas en el módulo de funciones.	Una respuesta de éxito o error.
14	"searchmycard"	Llamar a la función encargada de la búsqueda de tarjetas a través de una palabra y un filtro, en el módulo de funciones.	Las tarjetas que coinciden con la palabra, el filtro y que sean públicas.
15	"ordermycard"	Llamar a la función encargada de organizar las tarjetas de Mis Tarjetas en el módulo de funciones.	Las tarjetas de Mis Tarjetas con el filtro de ordenamiento.
16	"selectpredis"	Llamar a la función encargada de seleccionar la imagen prediseñada para el fondo de la tarjeta en el módulo de funciones.	Una respuesta de éxito o error.
17	"selectpredisedited"	Llamar a la función encargada de seleccionar la imagen prediseñada para el fondo de la tarjeta en el módulo de funciones.	Una respuesta de éxito o error.
18	"borrartarjetacreada"	Llamar a la función encargada de la eliminación de tarjetas de Tarjetas Creadas en el módulo de funciones.	Una respuesta de éxito o error.

Tabla 26: Valores y funcionalidades de la variable "tag" - Api WEB

Cuando la Api WEB recibe la orden a través del mensaje JSON, ésta, inmediatamente se comunica con el módulo de funciones para implementar la orden específica. Si el módulo de funciones retorna alguna respuesta, el módulo Index envía la respuesta a la aplicación Android a través de mensajes JSONObject o JSONArray.

Módulo de funciones:

Este módulo se pone en funcionamiento en el momento en que la orden de la aplicación Android ya ha sido filtrada por el módulo Index. El módulo de funciones contiene todos los métodos necesarios para realizar consultas y actualizaciones a la base de datos, es decir, que cada “tag” del módulo Index, llama a un método específico del módulo de funciones.

A continuación se listan las funciones de este módulo:

Nro.	Nombre del método	Funcionalidad
1	storeUser	Se encarga de insertar a la base de datos los datos del usuario que se está registrando, en la tabla “users”. También genera un identificador único con el cual se nombra su carpeta de archivos en el servidor.
2	forgetpass	Por medio del correo del usuario, se encarga de consultar a la base de datos, y retornar la pregunta y la respuesta secreta del usuario.
3	storeCard	Se encarga de insertar a la base de datos la información completa de la tarjeta que se está creando, en la tabla “cards”. También genera un identificador único, con el cual se nombra una carpeta que guarda en el servidor los archivos multimedia de la tarjeta.
4	updateCard	Se encarga de actualizar en la base de datos, la información de la tarjeta que se está editando, en la tabla “cards”. Se encarga de reemplazar los archivos multimedia de la tarjeta en el servidor.
5	addCard	Se encarga de insertar a la base de datos en la tabla “cards_local”, el identificador de la tarjeta que se quiere agregar y el

		identificador del usuario que la está agregando.
6	selectpredis	Se encarga de hacer una copia de la imagen prediseñada seleccionada, en la carpeta del usuario que está creando la tarjeta.
7	selectpredisedited	Se encarga de hacer una copia de la imagen prediseñada seleccionada, en la carpeta del usuario y en la carpeta de la tarjeta que se está editando.
8	getCardsByUserId	Se encarga de consultar a la base de datos, en la tabla "cards" y retornar las tarjetas que coincidan con el identificador del usuario.
9	myCards	Se encarga de consultar a la base de datos, en la tabla "cards_local" y retornar las tarjetas que coincidan con el identificador del usuario.
10	searchCardsByWord	Se encarga de consultar a la base de datos, en la tabla "cards", y retornar todas las tarjetas que coincidan con la palabra y el filtro de búsqueda.
11	deleteMyCard	Se encarga de borrar de la base de datos de la tabla "cards_local", la tarjeta con el identificador de usuario y el identificador de la tarjeta.
12	deleteCreatedCard	Se encarga de borrar de la base de datos de la tabla "cards", la tarjeta con el identificador de usuario y el identificador de la tarjeta.
13	searchMyCard	Se encarga de consultar a la base de datos, en la tabla "cards", y retornar todas las tarjetas que coincidan con la palabra y el filtro de búsqueda.
14	orderMyCard	Se encarga de consultar en la base de datos, en la tabla "cards_local", y retornar las tarjetas con el identificador de usuario, con un filtro de ordenamiento.
15	getUserByEmailAndPassword	Se encarga de consultar en la base de datos, en la tabla "users", y verificar que el email y contraseña ingresados por el usuario son válidos al momento de ingresar a la aplicación.

16	isCardExisted	Se encarga de consultar en la base de datos, en la tabla "cards", y retornar el número de filas que coinciden con el identificador de la tarjeta. Si la respuesta es mayor que cero, retorna true, sino retorna false.
17	modifname	Se encarga de actualizar en la base de datos, en la tabla "users", el nombre del usuario que está modificando sus datos.
18	modifpass	Se encarga de actualizar en la base de datos, en la tabla "users", la contraseña del usuario que está modificando sus datos.
19	isUserExisted	Se encarga de consultar en la base de datos, en la tabla "users", y retornar el número de filas que coinciden con el identificador del usuario. Si la respuesta es mayor que cero, retorna true, sino retorna false.
20	hashSSHA	Se encarga de cifrar una contraseña, haciendo uso del algoritmo sha1 y una sal. Esta función, retorna dos valores que conforman la contraseña.
21	checkhashSSHA	Se encarga de verificar si una contraseña es equivalente a las dos porciones de contraseña que están cifradas. Retorna true si las contraseñas coinciden, y de lo contrario, retorna false.
22	makeStringUTF8	Se encarga de convertir caracteres en codificación ISO-8859-1 a UTF-8, para el manejo de caracteres especiales del lenguaje español.

Tabla 27: Métodos, módulo de funciones - Api WEB

10.5. IMÁGENES DE LA APLICACIÓN EN FUNCIONAMIENTO

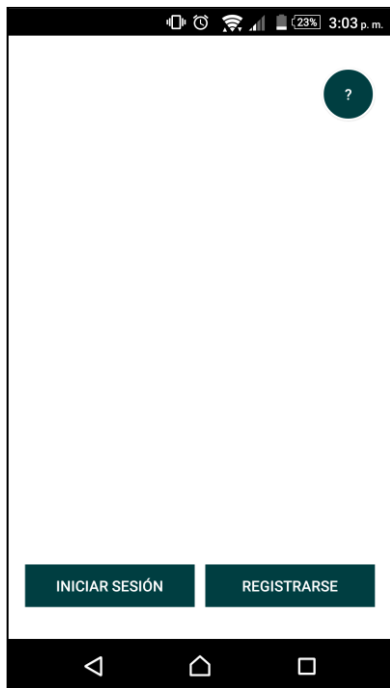


Ilustración 53: Pantalla de Aplicación 1

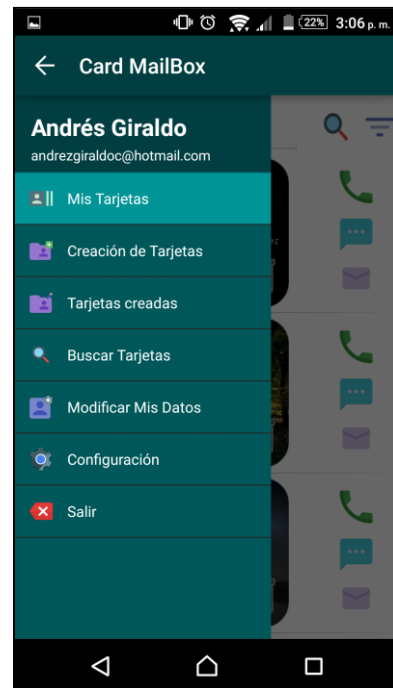


Ilustración 54: Pantalla de Aplicación 2

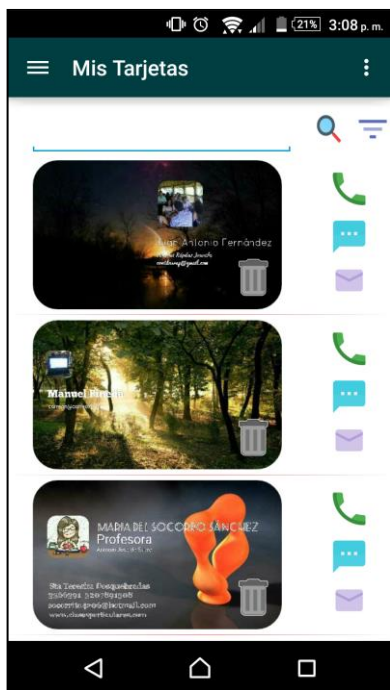


Ilustración 55: Pantalla de Aplicación 3



Ilustración 56: Pantalla de Aplicación 4



Ilustración 57: Pantalla de Aplicación 5



Ilustración 58: Pantalla de Aplicación 6

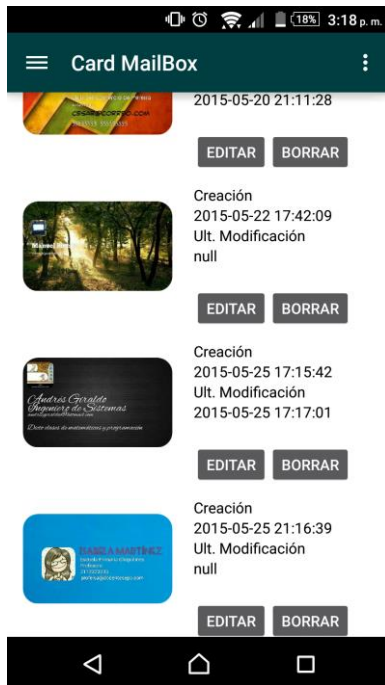


Ilustración 59: Pantalla de Aplicación 7

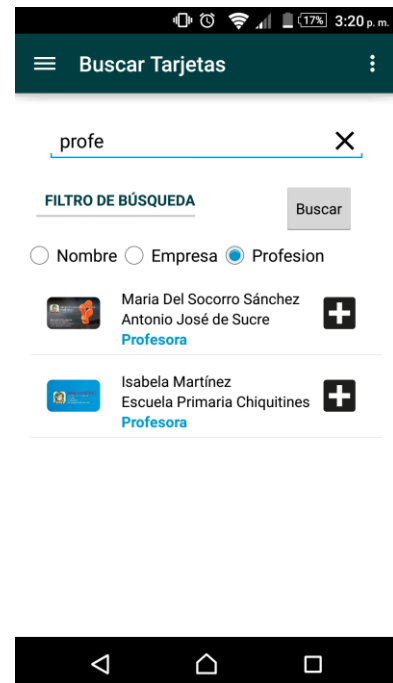


Ilustración 60: Pantalla de Aplicación 8

11. PRUEBAS DE VALIDACION DE HIPOTESIS

Tal y como se planteó en el diseño metodológico del presente proyecto, la hipótesis a validar es la siguiente:

“El proyecto que se va a desarrollar permitirá actualizar y administrar eficientemente tarjetas personales y empresariales”.

Para determinar la veracidad o falsedad de esta hipótesis deberán cumplirse entonces las siguientes condiciones:

- I. Que la aplicación permita al usuario actualizar la información contenida en las tarjetas de presentación que haya creado.
- II. Que la aplicación permita al usuario administrar eficientemente sus tarjetas de presentación.

11.1. MEDICIONES

Para la medición de las variables relacionadas con la validación de la hipótesis, se realizaron encuestas destinadas a conocer el estado actual de estas variables y cómo pueden verse afectadas luego, con el uso de la aplicación. Por una parte se realizó una encuesta basada en una prueba de usabilidad⁵ dirigida a un total de 51 personas que tuviesen un Smartphone con sistema operativo Android, este número de personas corresponde a la muestra calculada previamente. Por otro lado, para completar las mediciones, se encuestaron 7 litografías que prestan el servicio de fabricación de tarjetas de presentación.

A continuación, se muestra el formato de la prueba de usabilidad y de la encuesta practicada a los usuarios:

⁵ Una prueba de usabilidad se refiere a evaluar un producto mediante pruebas con los usuarios mismos, entregando información de cómo estos perciben y usan ese producto.

INSTRUCCIONES - PRUEBA DE USABILIDAD: Card MailBox

Una vez instalada la aplicación en su dispositivo móvil, proceda con los siguientes pasos:

- I.** Ejecute la aplicación
- II.** De clic en la opción “Registrarse” y complete el proceso de registro. Diríjase a la encuesta de usuario y conteste la pregunta número 2.
- III.** Si se ha registrado correctamente, la aplicación lo conducirá a su cuenta de usuario, vaya ahora al menú desplegable que aparece a la izquierda de la pantalla y seleccione la opción “Creación de Tarjetas”.
- IV.** Cuando esté en esta sección, asegúrese tener a la mano un reloj o alguna forma de medir tiempo (incluso puede hacer uso de la hora del celular) esto con el fin de cronometrar cuánto tiempo demora usted en crear una tarjeta.
- V.** Inicie el conteo de tiempo y proceda a crear una tarjeta a su entero gusto haciendo uso de las opciones que la aplicación le ofrece. Para llevar a cabo esta tarea, seleccione los campos a ser incluidos en la tarjeta (mínimo 3 para esta prueba) y déjela pública a otros usuarios. Usted sabrá si creó correctamente la tarjeta porque la aplicación le muestra un mensaje diciendo “Tarjeta creada correctamente”.
- VI.** Una vez aparezca el mensaje de confirmación de que se creó correctamente la tarjeta, detenga el conteo y diríjase a la encuesta de usuario para contestar las preguntas 3 y 4.
- VII.** Seleccione ahora en el menú de opciones la opción llamada “Tarjetas creadas”, allí se visualizan las tarjetas creadas por usted y podrá encontrar entonces la tarjeta que fabricó hace unos segundos, por favor ubíquela y dé clic sobre el botón “Editar” de esa tarjeta.
- VIII.** Modifique cualquiera de los datos ingresados previamente o agregue o elimine algún nuevo campo y guarde estos cambios procurando que aparezca el siguiente mensaje en la aplicación “Tarjeta actualizada correctamente”. Después vaya a la encuesta y responda la pregunta 5.
- IX.** Diríjase al menú de opciones y seleccione la opción “Buscar Tarjetas” desde allí podrá encontrar tarjetas de presentación de otros usuarios. Haciendo uso del campo de texto y los filtros de búsqueda, va a encontrar y agregar a su cuenta las tarjetas que corresponden a las siguientes descripciones:

- El nombre de la empresa es telecom
 - Una psicóloga
 - Un ingeniero de sistemas llamado Andrés Giraldo
- X.** Vaya a la opción “Mis Tarjetas” y compruebe que estas tarjetas recién agregadas, se encuentren allí en esta sección. Luego diríjase a la encuesta y por favor dé respuesta a la pregunta 6.
- XI.** Ahora, realice el siguiente ejercicio: Comience a cronometrar el tiempo y encuentre, el nombre de la persona o empresa que aparece en la última tarjeta de presentación física que usted haya recibido, una vez lo haya encontrado detenga el tiempo y vaya a la encuesta para contestar las preguntas de la 7 a la 9.
- XII.** Vaya a la opción “Mis Tarjetas” del menú de la aplicación y realice el ejercicio anterior, es decir, encuentre el nombre de la persona o empresa que aparece en la última tarjeta que agregó. No olvide contabilizar el tiempo que demora haciendo esta tarea y luego diríjase a la encuesta para contestar las preguntas de la 10 a la 12.
- XIII.** Realice el siguiente ejercicio: Si usted no tiene una tarjeta de presentación propia, suponga que tiene una consigo y trate de calcular el tiempo que demoraría en buscarla dentro de su cartera o bolsillo y pasarla a un compañero que se encuentra en frente suyo, luego vaya a la encuesta y responda la pregunta número 13.
- XIV.** Por último, para medir el tiempo que tomaría compartir su propia tarjeta de presentación a un compañero a través de la aplicación CardMailBox, teniendo en cuenta que él necesitaría registrarse y buscar su tarjeta, haga lo siguiente: Inicie el conteo de tiempo, registre un nuevo usuario (invente sus datos), ingrese a la cuenta de usuario, busque la tarjeta que usted creó e hizo pública previamente en el punto 5 y agréguela a la cuenta, luego detenga el tiempo y responda la pregunta número 14 de la encuesta.
- XV.** Finalmente vaya a la opción “Salir” del menú de opciones y diríjase a la encuesta para contestar las últimas preguntas del cuestionario. Muchas gracias por su participación.

ENCUESTA PRUEBA DE USABILIDAD: CardMailBox

A continuación se presenta una serie de preguntas destinadas a evaluar ciertos aspectos relacionados con el uso del aplicativo CardMailBox. Muchas gracias por el tiempo que ha dedicado en realizar completamente la encuesta.

*Obligatorio

Edad: _____

Sexo: M ___ F ___

1. ¿Guarda de forma ordenada las tarjetas de presentación que ha recibido de otras personas? *
 - No tengo un lugar específico dónde guardarlas
 - Sé dónde encontrarlas pero no las tengo de forma ordenada
 - Las tengo ordenadas y bien estructuradas de manera que es muy fácil encontrar cualquiera de ellas
-
2. ¿Cómo fue para usted realizar el proceso de registro en la aplicación? *
 - Muy Difícil
 - Difícil
 - Más o Menos
 - Fácil
 - Muy Fácil
-
3. ¿Cómo fue para usted realizar el proceso de creación de tarjetas haciendo uso de la aplicación?*
 - Muy Difícil
 - Difícil
 - Más o Menos
 - Fácil
 - Muy Fácil
-
4. ¿Cuánto tiempo tardó en crear la tarjeta en CardMailBox? Por favor, no olvide colocar la unidad de tiempo (horas, minutos, segundos, días, etc)
-

5. ¿Le fue posible actualizar la información de la tarjeta creada previamente en la aplicación? *

SI ___ NO ___

6. ¿Cómo fue para usted realizar el proceso de búsqueda de tarjetas de otros usuarios usando CardMailBox? *

- Muy Difícil
- Difícil
- Más o Menos
- Fácil
- Muy Fácil

7. ¿Le fue posible encontrar el nombre después de la búsqueda hecha entre sus tarjetas físicas? *

SI ___ NO ___

8. Si respondió afirmativamente la pregunta 7, continúe con esta pregunta. ¿Cuánto tiempo demoró realizando la búsqueda entre sus tarjetas en formato físico? Por favor, no olvide colocar la unidad de tiempo (horas, minutos, segundos, días, etc)

9. Si respondió negativamente la pregunta 7, continúe con esta pregunta. ¿Por qué no le fue posible encontrar el nombre buscado dentro de sus tarjetas físicas?

- No recuerdo cuál fue la última tarjeta de presentación que recibí
- Recuerdo cuál fue pero no la llevo conmigo en este momento
- Recuerdo cuál fue pero definitivamente no la pude encontrar
- Otra razón

10. ¿Le fue posible encontrar el nombre en la búsqueda hecha esta vez con la aplicación? *

SI ___ NO ___

11. Si respondió afirmativamente la pregunta 10, continúe con esta pregunta. ¿Cuánto tiempo demoró realizando la búsqueda dentro de la aplicación? Por favor, no olvide colocar la unidad de tiempo (horas, minutos, segundos, días, etc)

12. Si respondió negativamente la pregunta 10, continúe con esta pregunta. ¿Por qué no le fue posible encontrar el nombre buscado dentro de la aplicación?

13. ¿Cuánto tiempo se demoró haciendo el ejercicio de compartir una tarjeta en formato física? Por favor, no olvide colocar la unidad de tiempo (horas, minutos, segundos, días, etc)

14. ¿Cuánto tiempo se demoró haciendo el ejercicio de compartir una tarjeta a través de la aplicación? Por favor, no olvide colocar la unidad de tiempo (horas, minutos, segundos, días, etc)

15. ¿Alguna vez ha pagado para la distribución de sus tarjetas personales? *

SI ___ NO ___

16. Si respondió afirmativamente la pregunta 15, ¿Cuántas tarjetas entregó para que fueran repartidas?

17. Si respondió afirmativamente la pregunta 15, ¿Cuánto dinero pagó para que fueran repartidas ese número de tarjetas? *

18. ¿Seguiría usando CardMailBox para administrar y compartir sus tarjetas de presentación?

- Definitivamente no
- No
- Aún no me convengo
- Si

O Por supuesto que sí
Para las litografías se realizó la siguiente encuesta:

ENCUESTA A LITOGRAFÍAS

1. ¿Nombre de la empresa?
2. ¿Tienen servicio de impresión de tarjetas de presentación?
3. ¿Cuánto es el mínimo de tarjetas que imprimen?
4. ¿Cuánto cuesta imprimir ese mínimo de tarjetas?
5. ¿Cuánto tiempo se demora la empresa en entregar esa cantidad de tarjetas?

Muchas gracias por la información suministrada

12.RESULTADOS

La prueba de usabilidad fue realizada a una muestra de 51 personas entre las cuales se encontraban hombres y mujeres entre los 18 y 36 años de edad, quienes ya contaban con el conocimiento para hacer uso de dispositivos móviles. Para la realización de la prueba se le envió a cada participante el .apk de la aplicación (archivo para la instalación del software), las instrucciones de la prueba y la encuesta, la cual proporcionó los siguientes resultados:

La primera pregunta que aparece en la encuesta es la siguiente ¿Guarda de forma ordenada las tarjetas de presentación que ha recibido de otras personas? y los resultados de la misma se muestran a continuación:

Opción a elegir	Número de usuarios que seleccionaron esta opción
No tengo un lugar específico dónde guardarlas	16
Sé dónde encontrarlas pero no las tengo de forma ordenada	32
Las tengo ordenadas y bien estructuradas de manera que es muy fácil encontrar cualquiera de ellas	3

Tabla 28: Resultados - Encuesta, Pregunta 1

Estos primeros resultados muestran la necesidad que existe entre los usuarios encuestados de encontrar un mecanismo que facilite el orden y la administración de sus tarjetas, puesto que sólo 3 de las 51 personas en cuestión cumplen con ese objetivo.

12.1. RESULTADOS - ACTUALIZACIÓN DE TARJETAS

De los 51 usuarios que participaron de la prueba, a 49 de ellos les fue posible actualizar sin ningún tipo de inconvenientes, la información contenida en la tarjeta que habían creado previamente.

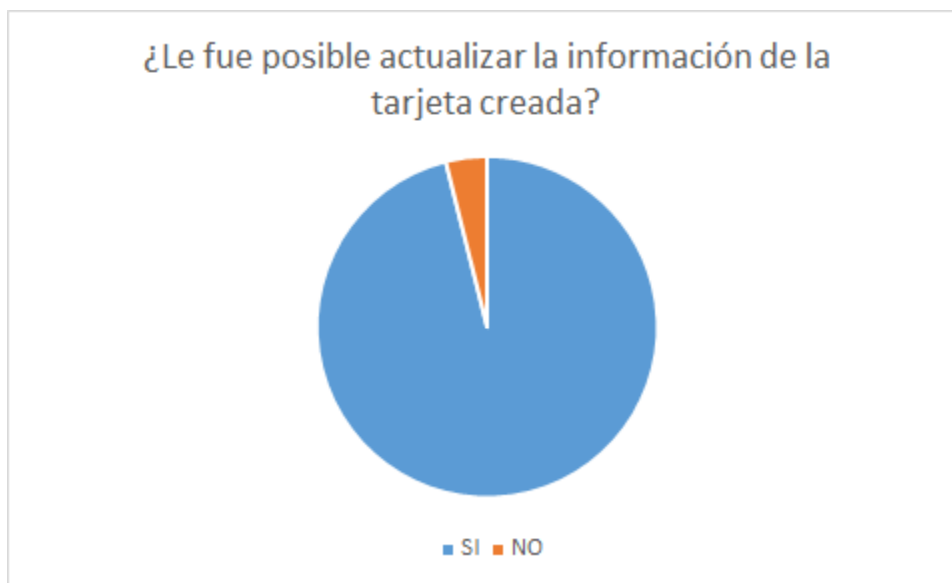


Ilustración 61: Diagrama circular - Índice de éxito al momento de actualizar una tarjeta creada

12.2. RESULTADOS - CREACIÓN DE TARJETAS

Para la creación de las tarjetas de presentación se realizaron dos tipos de encuestas, una aplicada a las litografías y otra a los usuarios finales.

Creación de tarjetas de la forma convencional

Los resultados de la encuesta que se aplicó a las litografías fueron los siguientes:

Empresa	¿Cuánto es el mínimo de tarjetas que imprimen?	¿Cuánto vale imprimirlas? Diseño Sencillo	¿Cuánto tiempo demora en entregarse?	Costo x unidad (costo en pesos colombianos)
Multigráficas SAS	100	45000	3 días hábiles	450
Tecno impresos	500	40000	4 días hábiles	80
ABC Litografía	1000	60000	3 días hábiles	60
Compugráficas	1000	60000	3 días hábiles	60
Tecni Arte Digital	1000	45000	3 días hábiles	45
Formas Gráficas	1000	50000	4 días hábiles	50
Litografía Elite Publicidad	1000	35000	3 días hábiles	35

Tabla 29: Resultados - Encuestas a litografías

En la mayoría de las litografías encuestadas el mínimo de tarjetas que se imprimen son 1000 tarjetas lo que significa que si un usuario necesita por ejemplo 20 tarjetas igual necesitará imprimir todo este monto y deberá asumir el costo de este. Ahora, haciendo el cálculo del costo por unidad con los anteriores datos (el costo del mínimo de tarjetas dividido por la cantidad mínima de tarjetas), en promedio al usuario le cuesta 111 pesos imprimir una sola tarjeta.

En cuanto al tiempo que demora la fabricación de las tarjetas, se tendrá en cuenta que la litografía tiene unos tiempos mínimos establecidos para la entrega de estas. Los tiempos mínimos se muestran en la tabla de resultados y el valor medio son 3 días aproximadamente, también es el valor correspondiente a la moda, por ende se tomará en cuenta este valor para la comparación con el tiempo de fabricación de tarjetas a través de la aplicación móvil.

Creación de tarjetas usando CardMailBox

Por otro lado, los resultados que arrojó la encuesta realizada a los usuarios con respecto al tema de la fabricación de tarjetas se describen a continuación.

En un inicio se preguntó a los usuarios cómo fue para ellos el proceso de creación de tarjetas haciendo uso de la aplicación, estas fueron las respuestas:

Opción a elegir	Número de usuarios que seleccionaron esta opción
Muy Difícil	0
Difícil	0
Más o Menos	2
Fácil	13
Muy Fácil	36

Tabla 30: Resultados - Creación de tarjetas desde la aplicación

Ahora bien, ante la pregunta ¿Cuánto tiempo tardó en crear la tarjeta en CardMailBox? las respuestas de los 51 usuarios mostraron que en promedio se tardan 541 segundos (9 minutos aproximadamente) en crear su tarjeta de presentación haciendo uso de la aplicación.

Por otra parte la aplicación es totalmente gratuita, por tanto los usuarios no tienen que pagar nada para crear sus tarjetas.

Comparación de los resultados obtenidos

A continuación se hace una comparación respecto a la fabricación de tarjetas de presentación desde el punto de vista de los dos escenarios, el de crear tarjetas de presentación con los servicios que presta una litografía y hacerlo usando el aplicativo móvil:

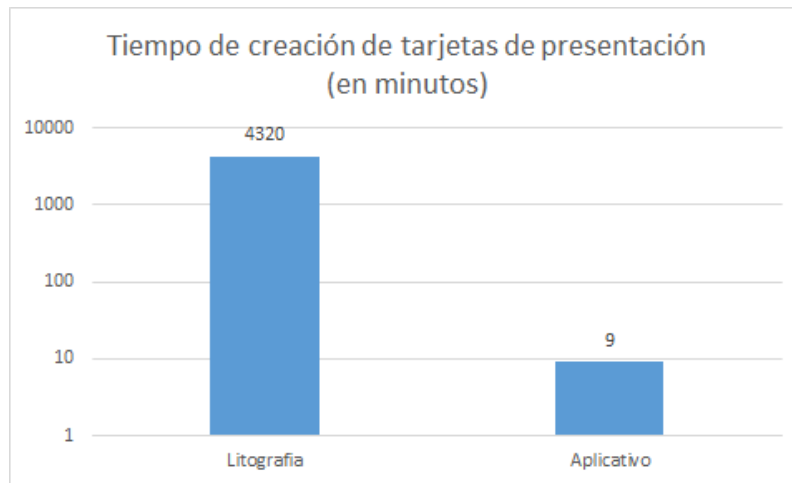


Ilustración 62: Diagrama de barras - Tiempo de creación de tarjetas

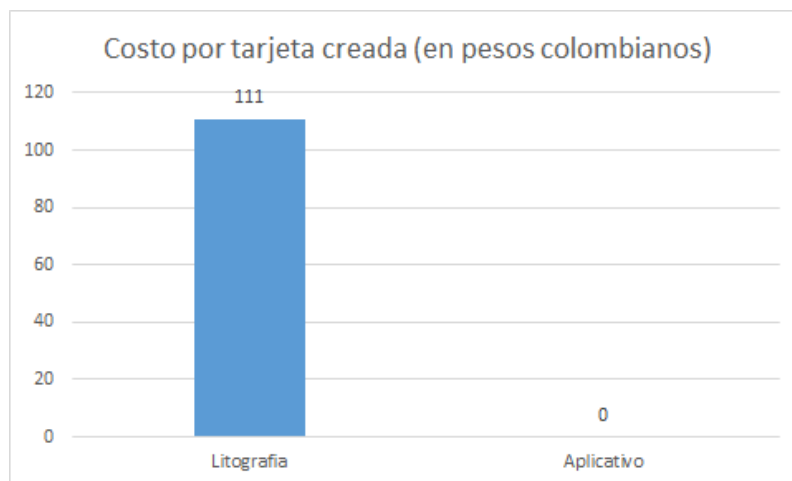


Ilustración 63: Diagrama de barras - Costo por tarjeta creada

12.3. RESULTADOS – BÚSQUEDA DE TARJETAS

Para la búsqueda de tarjetas de otros usuarios, se realizaron dos pruebas con el usuario encuestado, la primera consistía en preguntarle al usuario por el nombre de la persona o empresa que aparece en la última tarjeta de presentación física que él haya recibido, para esto, el usuario debía buscar dentro de sus tarjetas de presentación que tuviera guardadas y encontrar la tarjeta en cuestión para finalmente visualizar la información que se le solicita. La segunda prueba consistió en preguntarle al usuario por el nombre de la persona o empresa que aparece en la última tarjeta que agregó en su sección “Mis Tarjetas” de la aplicación, esto con el fin de comparar el tiempo que el usuario dedicó a realizar cada ejercicio y determinar cuál de las dos formas es más eficiente a la hora de buscar tarjetas de presentación personal.

Búsqueda de tarjetas en formato físico

Para evaluar la efectividad y eficiencia de la primera prueba llevada a cabo por el usuario, se le realizaron las siguientes preguntas relacionadas:

“¿Le fue posible encontrar el nombre después de la búsqueda hecha entre sus tarjetas físicas?”. Estos fueron los resultados:

Opción a elegir	Número de usuarios que seleccionaron esta opción
SI	20
NO	31

Tabla 31: Resultados - Búsqueda de tarjetas en formato físico

A los usuarios que respondieron negativamente la anterior pregunta, es decir aquellos que no pudieron completar con éxito la búsqueda, se les indagó luego por la causa relacionada con su respuesta, y esto fue lo que respondieron:

Opción a elegir	Número de usuarios que seleccionaron esta opción
No recuerdo cuál fue la última tarjeta de presentación que recibí	16
Recuerdo cuál fue pero no la llevo conmigo en este momento	7
Recuerdo cuál fue pero definitivamente no la pude encontrar	8
Otra razón	0

Tabla 32: Resultados - Causa de no encontrar tarjeta

Lo anterior permite ver tres dificultades evidentes al momento de buscar tarjetas de presentación de la manera corriente:

- Una persona puede olvidar fácilmente qué tarjetas ha recibido y los servicios o productos ofrecidos en cada una de ellas.
- Es muy probable que una persona no lleve siempre consigo todas sus tarjetas de presentación, esto por cuestiones de comodidad por ejemplo.
- Tal y como lo vimos en la encuesta, no es muy frecuente encontrar una persona que tenga de manera ordenada y bien estructurada sus tarjetas de presentación de manera que se le facilite la búsqueda de alguna de ellas.

Ahora, para los usuarios que pudieron culminar con éxito la prueba sobre la búsqueda del nombre dentro de sus tarjetas en formato físico, se les preguntó sobre el tiempo que dedicaron en realizar esta búsqueda y este fue el resultado:

En promedio, los 20 usuarios que pudieron culminar satisfactoriamente la búsqueda tardaron 99.75 segundos en llevarla a cabo.

Búsqueda de tarjetas a través de la aplicación móvil

Con respecto a la búsqueda de tarjetas a través de la aplicación móvil, se les preguntó a los usuarios si les fue posible encontrar el nombre en la búsqueda hecha con la aplicación y esta fue su respuesta:

Opción a elegir	Número de usuarios que seleccionaron esta opción
SI	50
NO	1

Tabla 33: Resultados - Búsqueda de tarjetas en la aplicación móvil

La única persona que expresó no haber encontrado el nombre que buscaba en la aplicación, argumentó que la tarjeta no apareció dentro de las opciones que entregó la aplicación, esto sin duda es un caso atípico puesto que los otros 50 usuarios pudieron hallar la información que buscaban, el error tal vez se deba a una inadecuada búsqueda por parte del usuario por qué no, a una posible falla en el sistema.

Por otro lado, a aquellas personas que respondieron SI a la anterior pregunta, se les indagó por el tiempo que se demoraron en efectuar tal tarea dentro de la aplicación y el resultado fue que las 50 personas se tomaron en promedio 10.76 segundos para efectuar la búsqueda.

Comparación de los resultados obtenidos

A continuación se hace una comparación respecto a la búsqueda de tarjetas de presentación desde el punto de vista de los dos escenarios, el de buscar tarjetas de presentación en formato físico y hacerlo usando el aplicativo móvil:

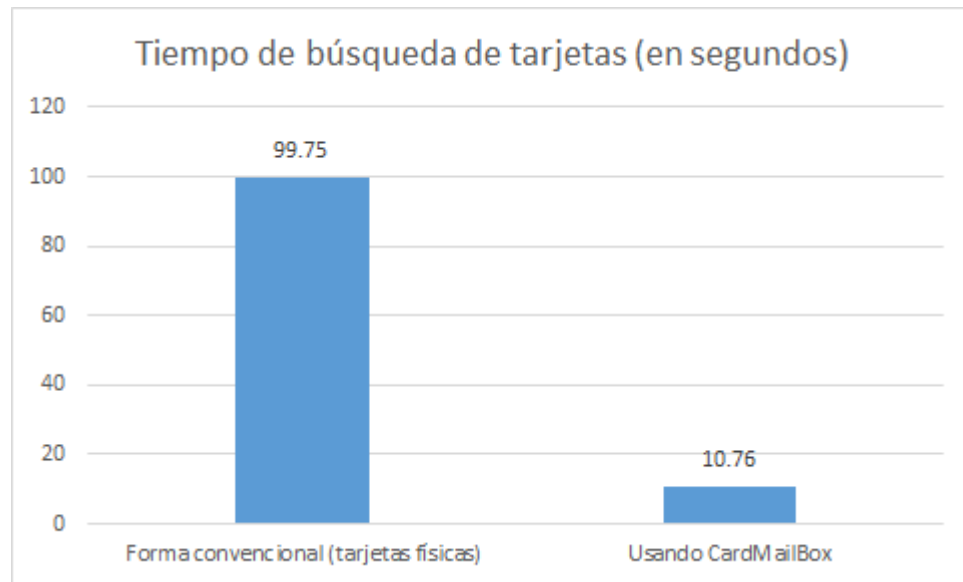


Ilustración 64: Diagrama de barras - Comparación, tiempo de búsqueda de tarjetas

12.4. RESULTADOS – INTERCAMBIO DE TARJETAS

Lo primero a tener en cuenta en este punto antes de mostrar cómo se obtuvieron los resultados, es que la forma en cómo se realiza el intercambio de tarjetas de presentación por medio de la aplicación cambia un poco el paradigma respecto a cómo se hace comúnmente este procedimiento cuando se tienen tarjetas en formato físico, pues en este último un usuario cuenta con un número determinado de tarjetas idénticas las cuales son compartidas a cierto número de personas objetivo hasta que estas se agoten, el usuario que la recibe puede o no necesitar esa tarjeta y él es quien decide qué hacer con ella si guardarla o tirarla. La aplicación en cambio permite a una persona publicar su tarjeta dentro de la aplicación de manera que cualquier otra persona que necesite de un servicio, producto o contacto en específico, buscará la tarjeta que mejor aplique y la almacenará en su cuenta de usuario.

Compartir una tarjeta de presentación de la forma en como se ha hecho comúnmente, implica de alguna manera un rol activo por parte de la persona que la está compartiendo, es por esto que se decidió basar la prueba en el hecho de que un usuario puede estar interesado no en publicar su tarjeta y esperar un resultado sino en compartirla a alguien en específico su tarjeta de presentación, si tiene su tarjeta de presentación dentro de la aplicación y no cuenta con una tarjeta de presentación en formato físico, necesita de alguna manera que la persona a quien entrega la tarjeta tenga una cuenta de usuario en la aplicación y busque su tarjeta para guardarla.

La prueba consiste entonces de dos partes, en la primera parte el usuario encuestado simula que busca dentro de su bolsillo o billetera una tarjeta de presentación física para luego entregarla a otra persona que tiene enfrente, el usuario debe medir el tiempo que se toma haciendo esto. En la segunda parte, el usuario hace la simulación de que desea compartir a alguien su tarjeta de presentación que se encuentra pública en la aplicación CardMailBox entonces motiva a esa otra persona a registrarse en la aplicación y buscar su tarjeta para que esta luego pueda ser guardada en la cuenta del nuevo usuario.

Intercambio de tarjetas en formato físico

Para evaluar la eficiencia de esta parte de la prueba, se le realizaron las siguientes preguntas al usuario:

“¿Cuánto tiempo se demoró haciendo el ejercicio de compartir una tarjeta en formato físico?”

De acuerdo con los resultados, las 51 personas encuestadas tardaron en promedio 7 segundos en realizar esta prueba.

Por otra parte se desea conocer si la aplicación mejora el tema de los costos de distribución de tarjetas, teniendo en cuenta que hay personas o empresas que pagan también por distribuirlas. Para evaluar este aspecto se planteó a los encuestados el siguiente interrogante:

“¿Alguna vez ha pagado para la distribución de sus tarjetas personales?”

Los resultados de la encuesta señalan lo siguiente:

Opción a elegir	Número de usuarios que seleccionaron esta opción
SI	4
NO	47

Tabla 34: Resultados - Costo de distribución de tarjetas

4 personas de las 51 encuestadas pagaron en algún momento para que se repartieran sus tarjetas ¿Cuánto dinero pagaron estas personas y cuántas tarjetas entregó para que fueran repartidas? Aquí están las respuestas:

Número de tarjetas a repartir	Cuánto pagó (pesos colombianos)	Costo de distribución por tarjeta (pesos colombianos)
500	50.000	100
500	30.000	60
100	15.000	150
250	40.000	160

Tabla 35: Resultados - Costo de distribución por tarjeta

Con los datos obtenidos de una pequeña parte de la muestra, se puede decir que para estos usuarios el costo medio de distribución por cada tarjeta es de 117.5 pesos.

Intercambio de tarjetas por medio de la aplicación

Para medir la eficiencia en el tiempo de distribución mediante el uso de la aplicación, se tuvo en cuenta las respuestas de los usuarios encuestados respecto a la pregunta ¿Cuánto tiempo se demoró haciendo el ejercicio de compartir una tarjeta a través de la aplicación? Los resultados se muestran a continuación:

Los usuarios encuestados se demoran en promedio 91.86 segundos compartiendo una tarjeta de presentación a un usuario específico por medio de la aplicación.

Por otra parte se recuerda que al ser la aplicación completamente gratuita, no cuenta con ningún tipo de costo de distribución de tarjetas.

Comparación de los resultados obtenidos

A continuación se hace una comparación de los resultados obtenidos respecto a la actividad de compartir tarjetas de presentación a una persona específica de la manera convencional haciendo uso de tarjetas en formato físico y haciendo uso por otro lado de la aplicación CardMailBox:

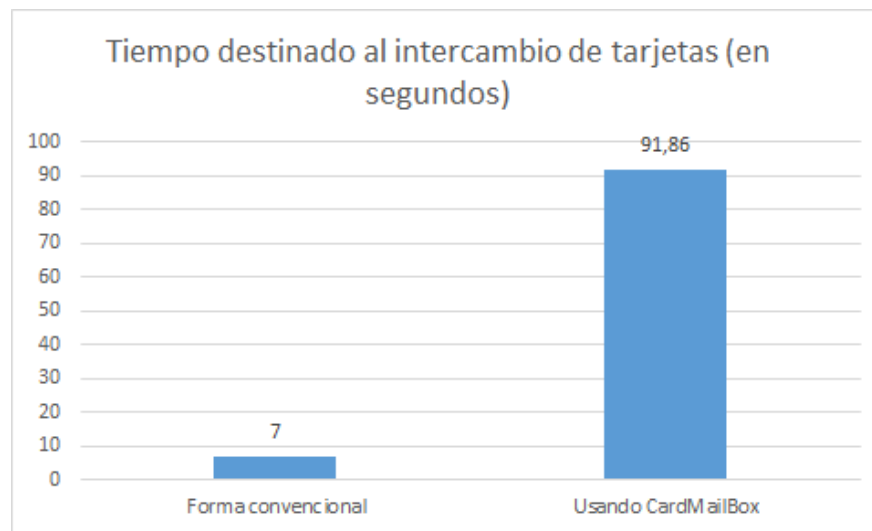


Ilustración 65: Diagrama de barras - Tiempo destinado al intercambio de tarjetas

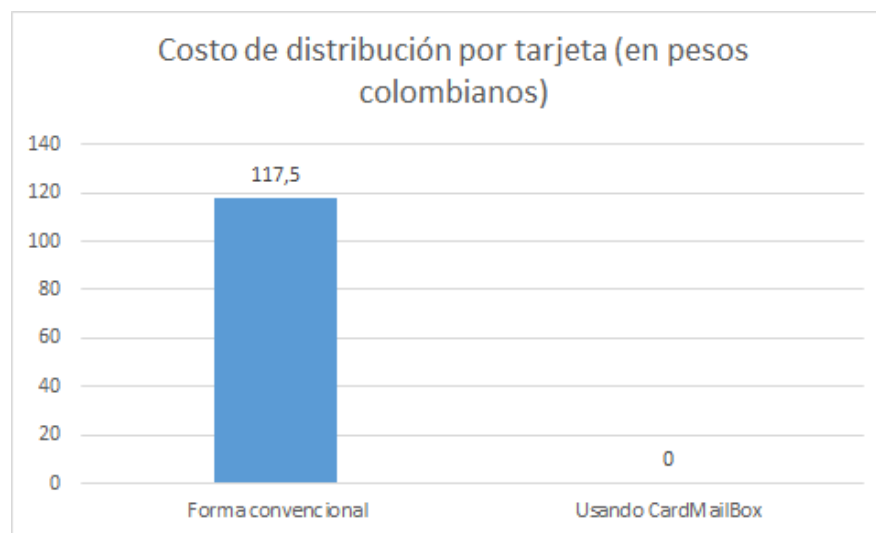


Ilustración 66: Diagrama de barras - Costo de distribución por tarjeta

12.5. VALIDACIÓN DE LA HIPÓTESIS

Para la validación de la hipótesis del presente proyecto, se comprobó por una parte que la aplicación permite la actualización de la información contenida en una tarjeta creada previamente por un usuario por lo menos en un 96% de las veces, por otro lado, en cuanto a administración eficiente de tarjetas de presentación se refiere, se pudo validar lo siguiente respecto a la administración convencional de las mismas:

- Que la aplicación es más eficiente en relación al tiempo de creación de tarjetas de presentación, logrando disminuir los 3 días que lleva comúnmente realizar este proceso a aproximadamente 9 minutos si se hace uso del aplicativo móvil.
- Que la aplicación es más eficiente en cuanto a costos de fabricación de tarjetas ayudando a disminuir los costos por tarjeta creada de 111 pesos colombianos a 0 pesos debido a que es totalmente gratuita.
- Que la aplicación es más eficiente en relación al tiempo dedicado a la búsqueda de tarjetas de presentación, disminuyendo el tiempo que comúnmente toma realizar esta actividad en aproximadamente 10 veces, pasando de unos 99.75 segundos a 10.76 segundos.
- Que la aplicación es más eficiente en cuanto a costos de distribución de tarjetas, logrando llevar los 117 pesos colombianos por tarjeta aproximadamente que le cuesta a un usuario mandar a distribuir sus tarjetas de presentación, a un valor de 0 pesos dado el carácter gratuito de la aplicación.
- Que la aplicación no resulta ser más eficiente en relación con el tiempo que conlleva compartir una tarjeta de presentación, si se evalúa de la misma forma en cómo este proceso es llevado a cabo con las tarjetas de presentación en formato físico. Este procedimiento tomaría unos 7 segundos si se realiza de la manera convencional mientras que con la aplicación el tiempo se eleva a unos 92 segundos aproximadamente.

No sobra decir que la manera en cómo se comparten (publican) eficientemente tarjetas de presentación en Card MailBox tiene otro ángulo respecto a la manera en cómo se comparten convencionalmente tarjetas de presentación.

Finalmente, la encuesta realizada a la muestra seleccionada revela una amplia aceptación hacia la aplicación ya que el 100% de los usuarios señalan que seguirían usando Card MailBox para administrar y compartir sus tarjetas de presentación.

13. PLATAFORMA WEB CARDMAILBOX (APORTE ADICIONAL)

Para mejorar la experiencia del usuario al usar la aplicación, se decidió anexar un objetivo extra a los establecidos desde la iniciación del proyecto. La aplicación móvil desarrollada para sistemas operativos Android, cumple con las metas establecidas en los objetivos del proyecto, pero puede mejorarse abriéndose paso en otras plataformas de desarrollo.

Se decidió crear un módulo extra que permita ser compatible con prácticamente cualquier dispositivo que tenga conexión a Internet, y que le permita al usuario disponer de todas las funcionalidades que el dispositivo móvil le brinda. Esta plataforma web, además de cumplir estos requerimientos, le permite al usuario trabajar sin ningún tipo de restricción sobre diferentes plataformas sin presentar inconsistencias de información o errores de escritura en la base de datos de la aplicación.

Este capítulo extra, encierra todas las especificaciones, requerimientos, diseño, arquitectura y desarrollo de la plataforma WEB. En resumen, este capítulo muestra como la aplicación móvil, también puede ser usada a través de un simple navegador de internet, o de un dispositivo móvil con sistema operativo diferente a Android, sin olvidar, que cada una de las funcionalidades definidas en Android, está incluida en esta plataforma.

13.1. MARCO TEÓRICO

MVC (Modelo, Vista, Controlador): [26] Es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos. Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores.

CodeIgniter: [27] Es un Framework (Herramienta) para el Desarrollo de Aplicaciones para la creación de webs usando PHP. Su meta es permitir desarrollar proyectos mucho más rápido que si se hiciera escribiendo el código desde cero, proporcionando una gran variedad de librerías para las tareas más corrientes, así como una interfaz simple y una estructura lógica para acceder a estas librerías.

Bootstrap: [28] Hace que el desarrollo front-end web sea más rápido y más fácil. Está hecho para la gente de todos los niveles, los dispositivos de todas las formas, y los proyectos de todos los tamaños. Bootstrap escala de manera fácil y eficiente sitios web y aplicaciones con una sola base de código, desde teléfonos hasta tabletas de escritorios.

13.2. ANÁLISIS

13.2.1. Requerimientos del sistema

Los requerimientos que se especifican a continuación, son similares al del aplicativo móvil, ya que las funcionalidades de la plataforma WEB son iguales.

Requerimientos funcionales:

- Que la plataforma WEB posea una pantalla principal para elegir entre “Iniciar Sesión” o “Registrarse”.
- Que la plataforma WEB permita a cada usuario crear una cuenta para administrar su propia información (Obligatorio crear cuenta).
- Que la plataforma WEB permita aceptar “Términos de uso de la aplicación” al momento de crear una cuenta.
- Que la plataforma WEB permita a un usuario loguearse con un correo y una contraseña.
- Que la plataforma WEB tenga la capacidad de recuperar la contraseña cuando sea olvidada.
- Que la plataforma WEB permita al usuario modificar sus datos personales asociados a su cuenta (correo, contraseña).
- Que la plataforma WEB posea un menú de opciones para elegir entre crear, buscar o administrar tarjetas agregadas del repositorio global.
- Que la plataforma WEB permita crear tarjetas personales y empresariales a través de plantillas o imágenes prediseñadas subidas por el usuario.
- Que la plataforma WEB permita almacenar las tarjetas creadas por el usuario.
- Que la plataforma WEB permita al usuario modificar sus propias tarjetas (No tiene permisos para modificar tarjetas creadas por otros usuarios).
- Que la plataforma WEB permita buscar tarjetas creadas por otros usuarios dentro del repositorio global, utilizando filtros de búsqueda.
- Que la plataforma WEB permita agregar a mi cuenta tarjetas encontradas en el repositorio global.

- Que la plataforma WEB permita listar las tarjetas agregadas por el usuario a su cuenta.
- Que la plataforma WEB permita realizar una búsqueda local de tarjetas agregadas en la cuenta del usuario.
- Que la plataforma WEB permita cerrar mi sesión de usuario

Requerimientos no funcionales:

- Que la plataforma WEB sea compatible con los siguientes Navegadores de Internet:
 - Google Chrome (Versión 35 en adelante)
 - Firefox (Versión 15 en adelante)
 - Safari (5.1 en adelante)
- La plataforma WEB debe correr con estos requisitos mínimos de hardware: 512 RAM, procesador uninúcleo 1Ghz y pantalla de más de 4.0”.
- La información almacenada por los usuarios debe estar hospedada en un servidor con un mínimo de 3 GB de capacidad.
- La base de datos debe correr sobre el motor de base de datos MySQL.
- El servidor debe soportar la conexión simultánea mínima de 3000 usuarios.

13.2.2. Diagrama y especificaciones de casos de uso

El diagrama de casos de uso, muestra principalmente, los usuarios que se relacionan con el sistema y las funcionalidades con las que el usuario interactúa. En este caso, la plataforma WEB posee exactamente los mismos usuarios del sistema y exactamente las mismas funcionalidades que la aplicación móvil.

Por ese motivo, el diagrama de casos de uso y la especificación de casos de uso, son exactamente iguales a las definidas en el Capítulo 8.

13.3. DISEÑO Y ARQUITECTURA

13.3.1. Diagrama de clases

A continuación se especifica todas las clases de la plataforma WEB. A su vez, se especifican también los atributos, funcionalidades y relaciones de estas clases.

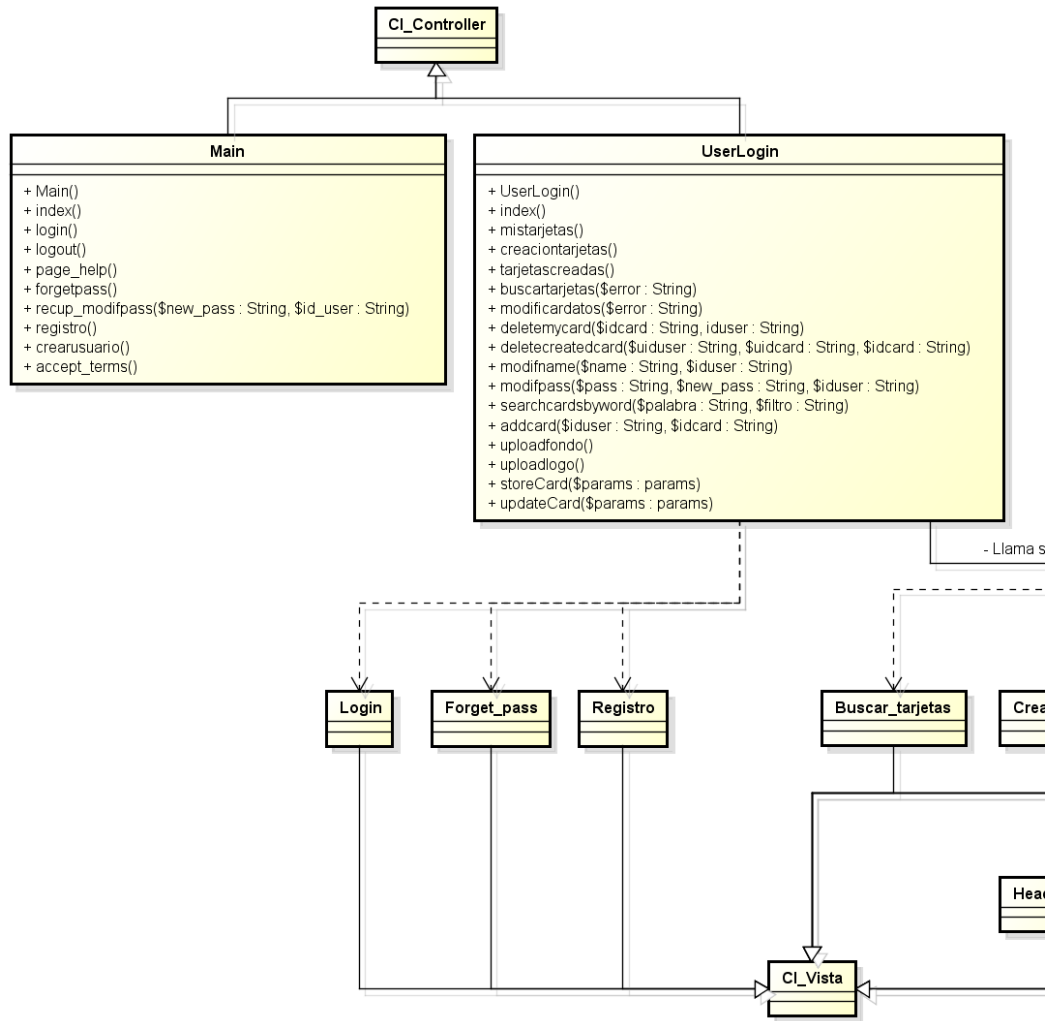


Ilustración 67: Diagrama de Clases - Plataforma WEB, Parte 1

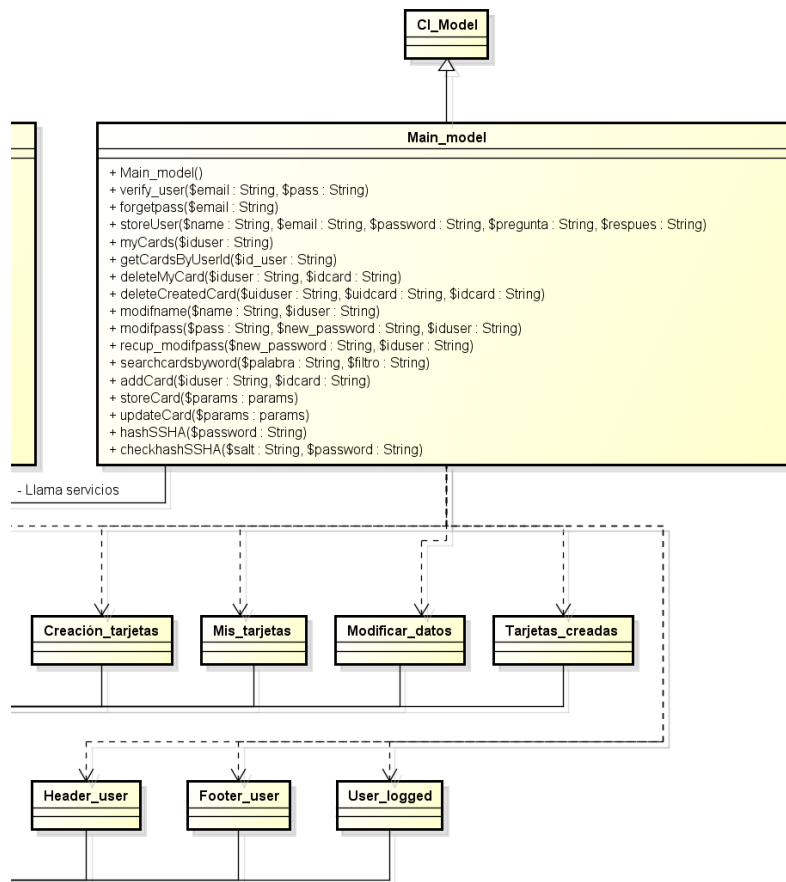


Ilustración 68: Diagrama de Clases - Plataforma WEB, Parte 2

Main: Hereda de una superclase llamada CI_Controller. Esta clase se encarga de la comunicación entre vistas y modelo cuando se hace uso de las funcionalidades:

Método Controlador	Vista que se instancia	Re direccionamiento a Controlador
Login	login	
Logout		main
page_help	forget_pass	
forgetpass	forget_pass	
recup_modifpass	forget_pass	
registro	registro	
crearusuario		UserLogin
accept_terms		

Tabla 36: Diagrama de Clases - Plataforma WEB, Main

UserLogin: Hereda de una superclase llamada CI_Controller. Esta clase se encarga de establecer comunicación entre vistas y modelo cuando se hace uso de las funcionalidades:

Método Controlador	Vista que se instancia	Re direccionamiento a Controlador
mistarjetas	mis_tarjetas	
creaciontarjetas	creacion_tarjetas	
tarjetascreadas	tarjetas_creadas	
buscartarjetas	buscar_tarjetas	
modificardatos	modificar_datos	
deleteMyCard		UserLogin/mistarjetas
deleteCreatedCard		UserLogin/mistarjetas
modifname		UserLogin/modificardatos
modifpass		UserLogin/modificardatos
searchCardsByWord	buscar_tarjetas	
addcard		UserLogin/buscartarjetas
upload_fondo	creacion_tarjetas	
upload_logo	creacion_tarjetas	
storeCard		UserLogin/storeCard
updateCard		UserLogin/updateCard

Tabla 37: Diagrama de Clases - Plataforma WEB, UserLogin

Main_model: Esta clase hereda de CI_Model. Tiene como función establecer la comunicación directa con la base de datos a través de consultas SQL. Cada vez que el usuario ejecuta una orden desde una vista y pasa por el controlador, el modelo es el responsable de que la orden se ejecute sobre la base de datos. Se compone de todas las funcionalidades de los controladores Main y UserLogin.

13.3.2. Diagrama de despliegue

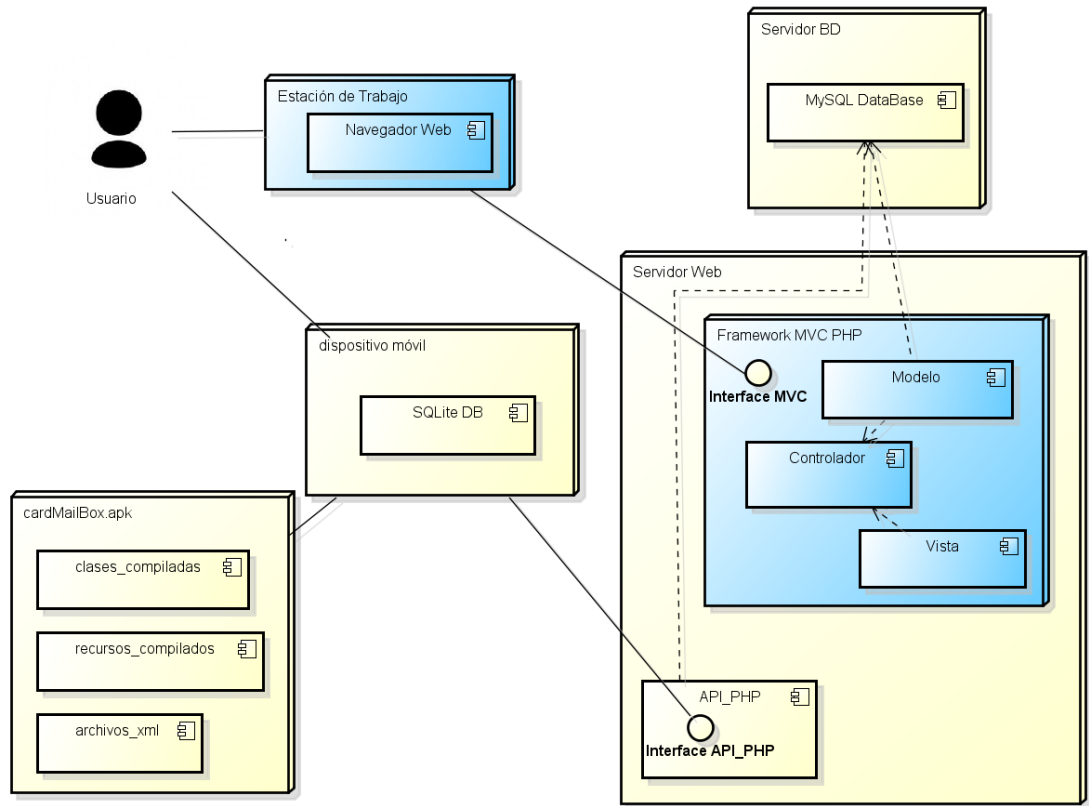


Ilustración 69: Diagrama de Despliegue - Plataforma WEB

Por medio de este diagrama es posible visualizar los diferentes nodos que hacen parte de la aplicación. Los nodos que sobresalen de color azul en el diagrama, componen lo que sería la plataforma web de la aplicación.

Lo primero que se puede observar en el diagrama, es que un usuario puede acceder a la aplicación a través de su dispositivo móvil o por medio de un navegador web.

Para que un usuario pueda hacer uso de las diferentes funcionalidades que tiene la aplicación dentro de un sitio web, es necesario crear primero toda una plataforma que permita al usuario entre otras cosas moverse de forma amigable por las diferentes opciones con las que cuenta la aplicación e interactuar de manera transparente con la Base de Datos donde será almacenada toda su información y la de sus tarjetas. Para la creación de esta plataforma es recomendable hacer uso de un Framework que sirva de apoyo y facilite todas las actividades relacionadas con el proceso de desarrollo Web, tal como se hizo en el presente proyecto.

El nodo llamado Framework MVC PHP utiliza un patrón de arquitectura de software que separa los datos y la lógica del negocio de la interfaz del usuario, tiene tres componentes que son el modelo, la vista y el controlador.

El modelo se encarga de gestionar el acceso a la información con la que el sistema opera, se puede observar que es quien permite en últimas la interacción con la Base de Datos.

El controlador responde a eventos (generalmente del usuario) y realiza peticiones al 'modelo' cuando se hace alguna solicitud sobre la información.

La vista presenta la información y la lógica del negocio en un formato adecuado para interactuar con el usuario.

Un último dato importante del diagrama con respecto a su versión web, es que el usuario accede a la misma información suya y de sus tarjetas. Esto se debe a que la plataforma web se comunica con la misma Base de datos a la que se comunica un usuario cuando accede desde su dispositivo móvil.

13.3.3. Diagrama de actividades

La plataforma web que se desarrolla como objetivo extra, posee un conjunto de actividades que constituyen la forma en que el usuario interactúa con la plataforma.

Estas actividades son exactamente las mismas construidas en el Diagrama de Actividades en el capítulo 8 del proyecto, ya que la plataforma WEB fue construida con el fin de mostrar al usuario las mismas funciones y los mismos mecanismos de interacción, que posee el aplicativo móvil.

Cada una de las actividades y cada una de las rutas del diagrama, poseen las mismas características que las especificadas en el capítulo 8.

13.4. DESARROLLO PLATAFORMA WEB

Para el desarrollo de la plataforma WEB, se hace uso de un marco de trabajo (framework) conocido como Modelo-Vista-Controlador (MVC). Este framework posee diferentes características que permiten una organización adecuada en el desarrollo de páginas web.

La plataforma WEB, al igual que la aplicación móvil, requiere una conexión a la base de datos e interactuar con el usuario, y ya que la Api WEB de la aplicación móvil está desarrollada sobre lenguaje PHP al igual que la configuración de conexión, se decide hacer uso de un framework MVC en PHP.

13.4.1. Estructura de codificación

El desarrollo de la plataforma WEB está hecho sobre CodeIgniter. Este framework MVC define toda la plataforma en tres módulos principales:

- **Modelo (Model):** Su funcionamiento es similar al del Módulo de funciones de la Api WEB de la aplicación móvil. Se encarga de comunicarse directamente a la base de datos a través de consultas SQL.
- **Controlador (Controller):** Su funcionamiento es similar al del Módulo Index de la Api WEB de la aplicación móvil. Se encarga de recibir las órdenes del usuario y procesarlas con el fin de que posteriormente sea llamado un método del Modelo y realice la comunicación a la base de datos.
- **Vista (View):** Se encarga de administrar toda la estructura visual de la plataforma WEB. Permite la interacción entre el usuario y la plataforma a través de lenguajes como HTML, CSS y JavaScript.

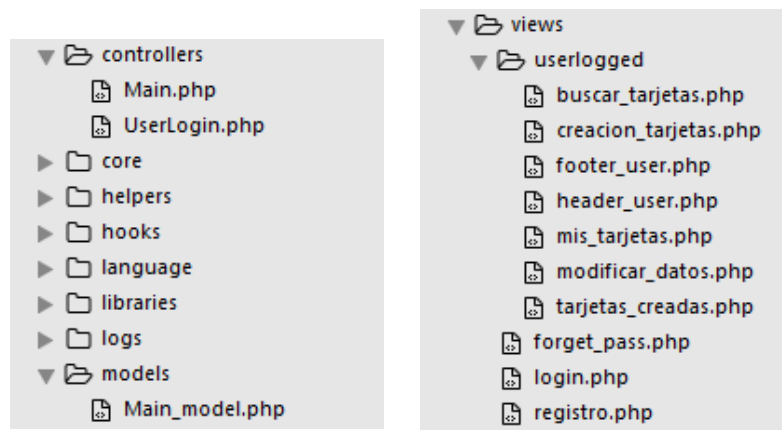


Ilustración 70: Estructura de codificación - Plataforma WEB

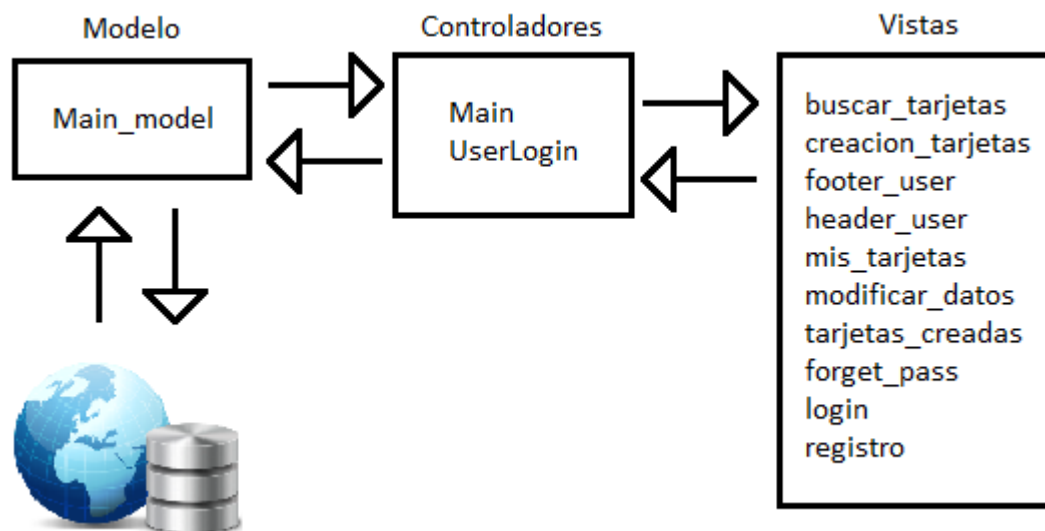


Ilustración 71: MVC en la plataforma WEB

13.4.2. Definición de clases y funcionalidades

Modelos

Nro.	Nombre	Funciones	Especificación
1	Main_model	construct	Constructor de la clase Main_model
		verify_user	Función encargada de verificar si los datos ingresados por el usuario al momento de ingresar a la plataforma WEB, son correctos.
		forgetpass	Por medio del correo del usuario, se encarga de consultar a la base de datos, y retornar la pregunta y la respuesta secreta del usuario.
		storeUser	Se encarga de insertar a la base de datos los datos del usuario que se está registrando, en la tabla "users". También genera un identificador único con el cual se nombra su carpeta de archivos en el servidor.
		myCards	Se encarga de consultar a la base de datos en la tabla "cards_local", y retornar las tarjetas que coincidan con el identificador del usuario.
		getCardsByUserId	Se encarga de consultar a la base de datos, en la tabla "cards" y retornar las tarjetas que coincidan con el identificador del usuario.
		deleteMyCard	Se encarga de borrar de la base de datos de la tabla "cards_local", la tarjeta con el identificador de usuario y el identificador de la tarjeta.
		deleteCreatedCard	Se encarga de borrar de la base de datos de la tabla "cards", la tarjeta con el identificador de usuario y el identificador de la tarjeta.
		modifname	Se encarga de actualizar en la base de datos, en la tabla "users", el nombre del usuario que está modificando sus datos.
		modifpass	Se encarga de actualizar en la base de datos, en la tabla "users", la contraseña del usuario que está modificando sus datos.

		recup_modifpass	Se encarga de actualizar en la base de datos, en la tabla "users", la contraseña del usuario que está recuperando su contraseña.
		searchCardsByWord	Se encarga de consultar a la base de datos, en la tabla "cards", y retornar todas las tarjetas que coincidan con la palabra y el filtro de búsqueda.
		addCard	Se encarga de insertar a la base de datos en la tabla "cards_local", el identificador de la tarjeta y el identificador del usuario.
		hashSSHA	Se encarga de cifrar una contraseña, haciendo uso del algoritmo sha1 y una sal. Esta función, retorna dos valores que conforman la contraseña.
		checkhashSSHA	Se encarga de verificar si una contraseña es equivalente a las dos porciones de contraseña que están cifradas.

Ilustración 72: Modelos MVC - Especificación de funcionalidades

Controladores

Nro.	Nombre	Funciones	Especificación
1	Main	construct	Constructor de la clase Main
		index	Función que corre por defecto cuando se instancia la clase Main sin ningún parámetro. Si el usuario no está logueado, index re direcciona al usuario hacia la página de login.
		login	Función encargada de validar los datos ingresados por el usuario en el formulario de login, a través de una consulta al modelo Main_model. Si los datos son correctos, re direcciona hacia la vista user_logged.
		logout	Destruye la sesión creada por el usuario y re direcciona hacia el controlador Main.

		page_help	Se encarga de re direccionar hacia la vista forget_pass.
		forgetpass	Se encarga de validar los datos ingresados por el usuario en el formulario de recuperación de contraseñas, a través de una consulta al modelo Main_model. Si el correo ingresado existe, re direcciona hacia la vista forget_pass con los datos de pregunta y respuesta secreta para la posterior recuperación de contraseña
		recup_modifpass	Se encarga de asignar una nueva contraseña a la cuenta del usuario después de haber pasado por el debido procedimiento de recuperación de contraseña.
		registro	Se encarga de cargar la vista registro.
		crearusuario	Se encarga de validar los datos ingresados por el usuario en el formulario de registro. Si los datos son correctos, se realiza el registro en la base de datos por medio del modelo Main_model y se re direcciona al controlador UserLogin.
		accept_terms	Se encarga de validar si el usuario acepta o no acepta los términos y condiciones de uso.
2	UserLogin	construct	Constructor de la clase Main
		index	Función que corre por defecto cuando se instancia la clase Main sin ningún parámetro. Si el usuario no está logueado, index re direcciona a la vista login.
		mistarjetas	Se encarga de llamar a la funcionalidad del modelo Main_model, encargada de listar las tarjetas agregadas por el usuario. Después de realizar la consulta, se re direcciona a la vista mis_tarjetas
		creaciontarjetas	Se encarga de cargar la vista que con todas las opciones para la creación de una tarjeta.

		tarjetascreadas	Se encarga de llamar a la funcionalidad del modelo Main_model, encargada de listar las tarjetas creadas por el usuario. Después de realizar la consulta, se re direcciona a la vista tarjetas_creadas.
		buscartarjetas	Se encarga de cargar la vista buscar_tarjetas, y si se presenta el caso, también de mostrar errores encontrados durante la búsqueda.
		modificardatos	Se encarga de cargar la vista modificar_datos, y si se presenta el caso, también de mostrar errores encontrados durante la modificación de datos.
		deleteMyCard	Se encarga de llamar a la funcionalidad del modelo Main_model encargado de eliminar una tarjeta agregada por el usuario. Después de realizar la eliminación, se re direcciona al controlador UserLogin/mistarjetas
		deleteCreatedCard	Se encarga de llamar a la funcionalidad del modelo Main_model encargado de eliminar una tarjeta creada por el usuario. Después de realizar la eliminación, se re direcciona al controlador UserLogin/mistarjetas
		modifname	Se encarga de llamar a la funcionalidad del modelo Main_model encargado de modificar el nombre del usuario. Después de realizar la modificación, se re direcciona al controlador UserLogin/modificardatos y como parámetro, se envía un número indicando si la modificación fue exitosa o fallida.
		modifpass	Se encarga de llamar a la funcionalidad del modelo Main_model encargado de modificar la contraseña de la cuenta de

			<p>usuario. Después de realizar la modificación, se re direcciona al controlador</p> <p>UserLogin/modificardatos y como parámetro, se envía un número indicando si la modificación fue exitosa o fallida.</p>
		searchCardsByWord	<p>Se encarga de llamar a la funcionalidad del modelo Main_model encargado buscar tarjetas por medio de una palabra y un filtro. Después de realizar la búsqueda, se re direcciona a la vista buscar_tarjetas junto con los datos de las tarjetas encontradas.</p>
		addCard	<p>Se encarga de llamar a la funcionalidad del modelo Main_model encargado agregar una tarjeta a la cuenta de usuario. Después de añadir la tarjeta, se re direcciona a la vista buscar_tarjetas y como parámetro, se envía un número indicando si la modificación fue exitosa o fallida.</p>
		uploadfondo	<p>Se encarga de la subida del fondo escogido por el usuario para la tarjeta que esté en creación o edición.</p>
		uploadlogo	<p>Se encarga de la subida del logo escogido por el usuario para la tarjeta que esté en creación o edición.</p>
		storeCard	<p>Se encarga de la creación de la tarjeta a través de la funcionalidad del modelo Main_model. Tiene como parámetros todos los campos que contiene una tarjeta.</p>
		updateCard	<p>Se encarga de la actualización de la tarjeta a través de la funcionalidad del modelo Main_model. Tiene como parámetros todos los campos que contiene una tarjeta.</p>

Ilustración 73: Controladores MVC - Especificación de funcionalidades

13.5. IMÁGENES DE LA PLATAFORMA WEB EN FUNCIONAMIENTO



Ilustración 74: Pantalla, Plataforma WEB #1

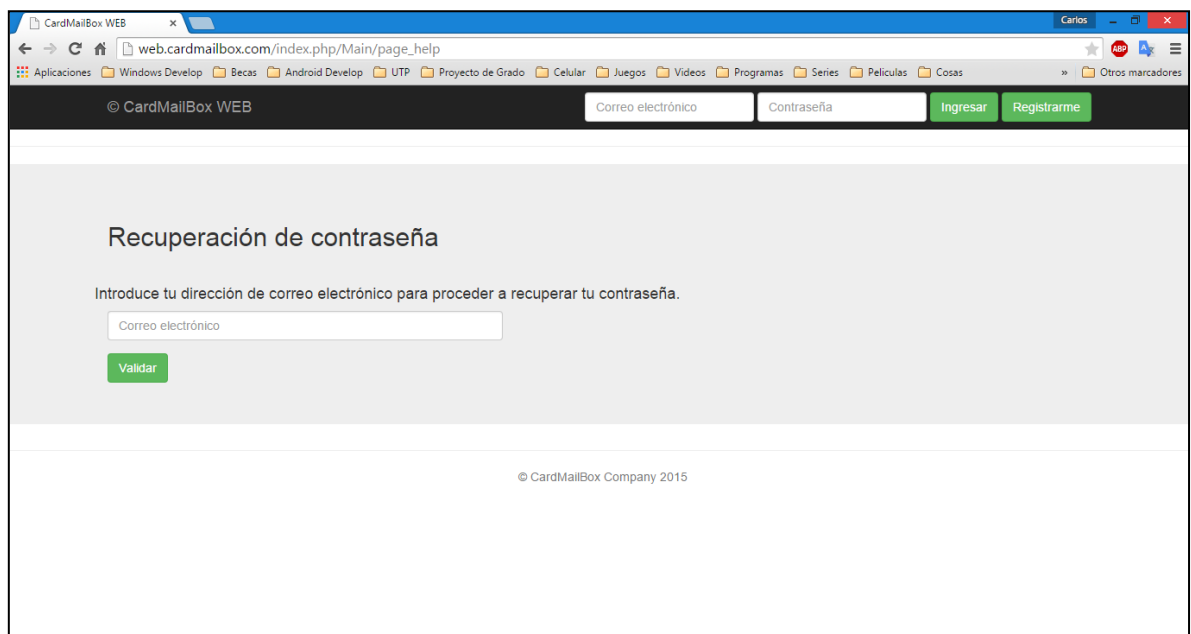


Ilustración 75: Pantalla, Plataforma WEB #2

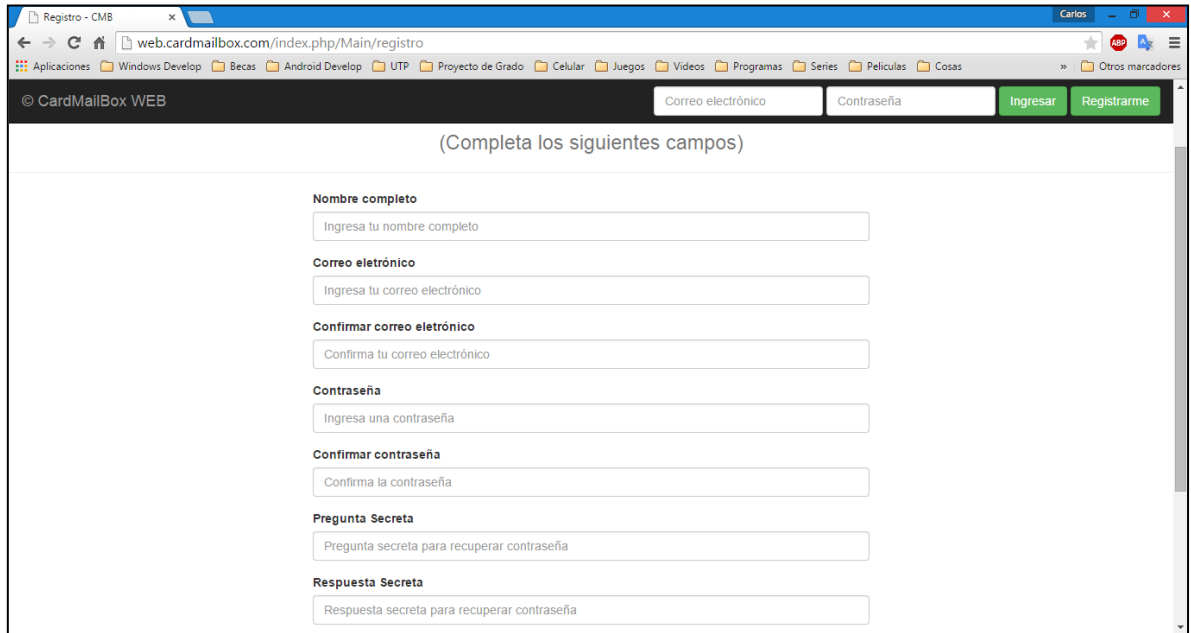


Ilustración 76: Pantalla, Plataforma WEB #3

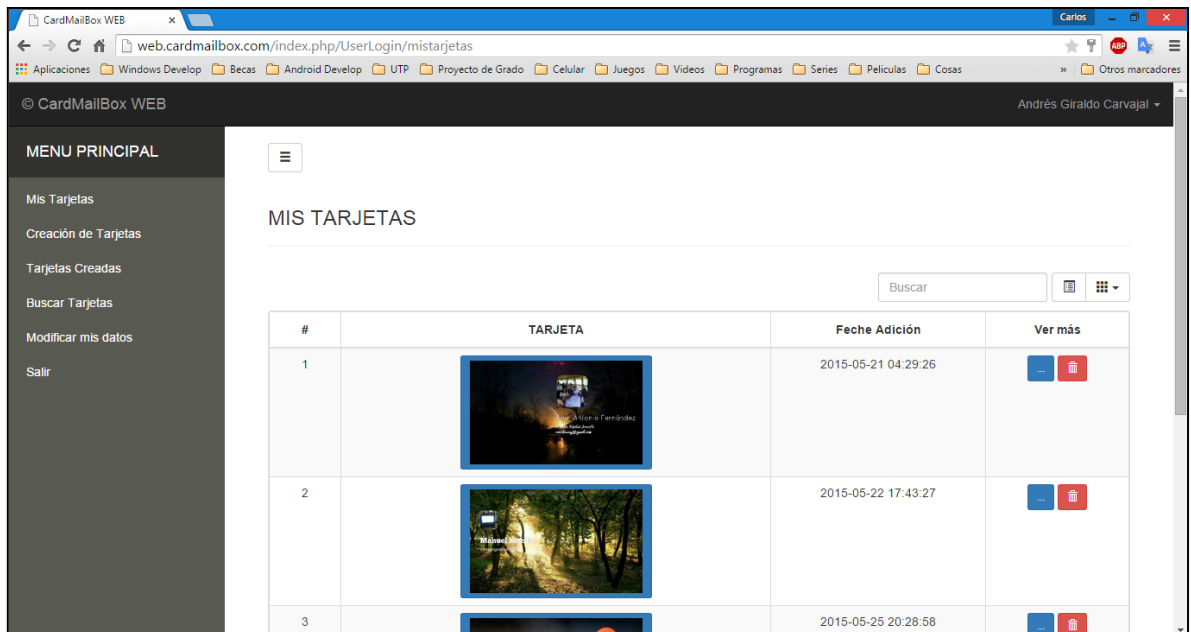


Ilustración 77: Pantalla, Plataforma WEB #4



Ilustración 78: Pantalla, Plataforma WEB #5

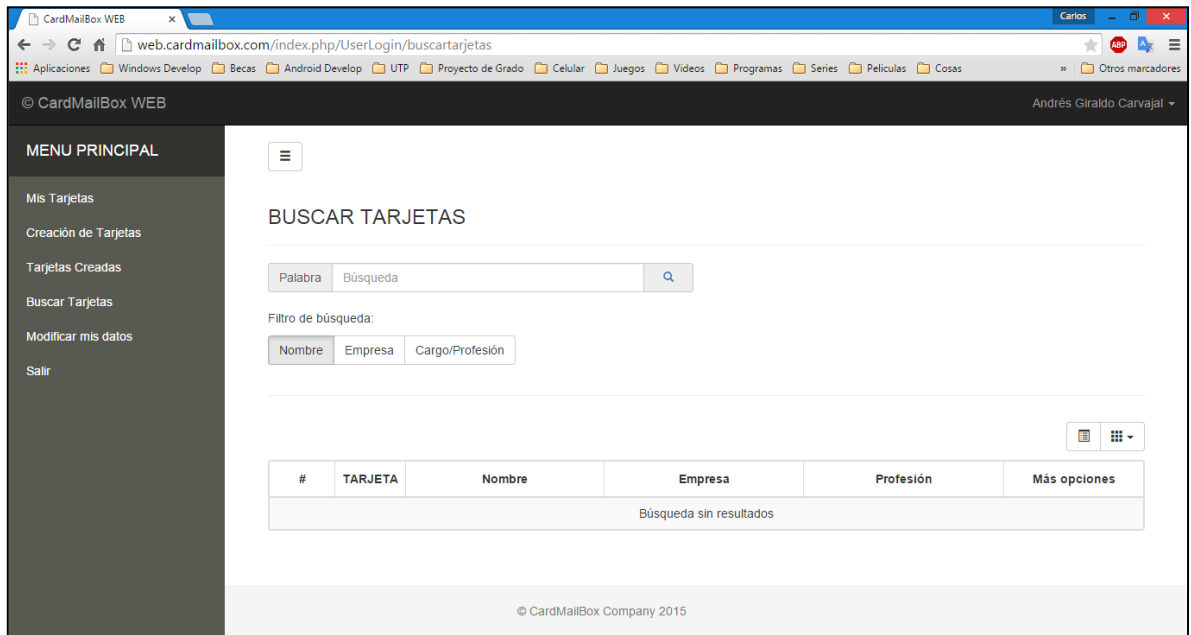


Ilustración 79: Pantalla, Plataforma WEB #6

14. CONCLUSIONES

Durante la realización de este documento, y durante el desarrollo de cada módulo de la aplicación móvil y la plataforma WEB, se establecieron las siguientes conclusiones:

- Es de vital importancia realizar un análisis exhaustivo sobre el desarrollo que se piense implementar. La lista de requerimientos y la definición de funcionalidades del sistema permite tener una idea no del todo precisa, pero que puede encaminar el proyecto hacia la dirección correcta.
- Es de vital importancia realizar un diseño y una arquitectura de software antes de comenzar con el desarrollo del aplicativo. Cada diagrama realizado en esa etapa de diseño permite estructurar, prever y facilitar el desarrollo y codificación, y sin duda, el tiempo de desarrollo del aplicativo se reduce considerablemente.
- Modularizar el desarrollo de un aplicativo permite llevar un orden y una estructura que evite confusiones y errores. Además, permite hacer un seguimiento más sencillo cuando se tienen problemas poco visibles o indetectables a primera vista.
- El control de versiones permite llevar un historial completo del desarrollo del aplicativo, lo que permite regresar a etapas anteriores del proyecto y evitar quedar estancado en errores poco perceptibles a la vista. Además de esto, el control de versiones facilita llevar el registro de fechas y modificaciones del desarrollo.
- Es de vital importancia desarrollar bajo un marco de trabajo (framework), y en el caso de este proyecto, sobre un framework MVC. El solo hecho de modularizar el desarrollo, permite la división de tareas, procesos y responsabilidades, lo cual se ve reflejado en la reducción de tiempo y facilidad de desarrollo.
- Desarrollar para el sistema operativo Android, es ventajoso, ya que es el sistema operativo móvil más usado del mundo y está en un crecimiento constante.
- El desarrollo nativo sobre Android permite un manejo más libre del dispositivo y un uso más eficaz sobre el hardware del dispositivo. Este desarrollo nativo permite hacer uso de hilos y tareas asíncronas, que permiten más fluidez de la aplicación y más compatibilidad sobre dispositivos que no sean de alta gama.

- El uso de Hilos y tareas asíncronas refleja instantáneamente sobre la aplicación, una mayor fluidez y un uso menor de recurso. Estos hilos a pesar de ser engorrosos al momento de implementar, enumeran una gran lista de ventajas, que al final del desarrollo, dan al usuario esa sensación de que el dispositivo corre de manera apropiada al usar el aplicativo.
- A pesar de que existen muchos lenguajes para desarrollar sobre un framework MVC, Php tiene la ventaja de poseer más compatibilidad en servidores web, por lo que facilita el desarrollo y pruebas.
- El IDE Android Studio es una herramienta muy completa que permite, además de desarrollar aplicativos para Android, trabajar de manera cómoda con la ventaja de tener un compilador muy bien estructurado, con un muy buen seguimiento de errores.
- Las pruebas de usabilidad permiten definir la aceptabilidad por parte de los usuarios, y acompañados de una encuesta, permiten evaluar el nivel de comprensión y satisfacción.
- Aprender a trabajar en equipo y aprender a usar la división de trabajo y responsabilidades, es vital al momento de desarrollar un proyecto. Sin esto, sencillamente el tiempo de desarrollo se vería alargado y el número de tareas desordenaría el desarrollo del proyecto.

15. MEJORAS FUTURAS

La aplicación desarrollada en este proyecto cumple con los objetivos establecidos al inicio del proyecto, pero es importante tener claro que el proyecto puede tener muchas más funcionalidades, más soporte, mejor estructura gráfica, etc.

A continuación se listan las mejoras futuras para la aplicación:

- Mejoramiento de la presentación gráfica de la aplicación haciendo uso de:
 - Más gama de colores
 - Iconos más amigables
 - Material Design
 - Animaciones
 - ListView con más interactividad
- Implementación de la aplicación sobre Sistemas Operativos iOS y Windows Phone haciendo uso de un desarrollo nativo.
- Implementación de tutoriales animados para los usuarios que utilicen por primera vez la aplicación.

16. BIBLIOGRAFÍA

[1] Alfredo Vega Jaramillo, Manual de Derecho de Autor, Dirección Nacional de Derecho de Autor, Ministerio del Interior y de Justicia [En línea] <[http://www.derechodeautor.gov.co/documents/10181/331998/Cartilla+derecho+derecho+de+autor+\(Alfredo+Vega\).pdf/e99b0ea4-5c06-4529-ae7a-152616083d40](http://www.derechodeautor.gov.co/documents/10181/331998/Cartilla+derecho+derecho+de+autor+(Alfredo+Vega).pdf/e99b0ea4-5c06-4529-ae7a-152616083d40)>

[1] AVERY, La Tarjeta de Presentación que es tan Versátil como tu Negocio. [En línea] <http://www.avery.com/avery/es_ar/Proyectos-e-Ideas/Articulos/La-Tarjeta-de-Presentacion-que-es-tan-Versatil-como-tu-Negocio.htm>

[2] Johnson, Bobbie, BBC, Las tarjetas de presentación agonizan en un mundo digital. [En línea] <<http://www.conexionbrando.com/1354028>>

[3] TARJETAS DE PRESENTACIÓN, Bogotá, Colombia. [En línea]. <<http://www.tarjetaspresentacion.com.co/index.html>>

[4] Equipo de Redacción, BBC Mundo. [En línea] <http://www.bbc.co.uk/mundo/noticias/2015/03/150302_economia_tarifas_celulares_lf> [citado el 3 de Marzo de 201506]

[5] GLOBEREDS – Web Master [En línea] <<http://www.otrascosas.com/bodas/categoria.asp?idcat=144>> [citado el 5 de Abril de 2006]

[6] Gutiérrez, Tatiana. Tarjeta de presentación, el brazo derecho de tu branding personal. [En línea] <<http://www.altonivel.com.mx/34387-branding-personal-posicionate-con-tus-tarjetas-de-presentacion.html>> [citado el 4 de Marzo de 2013]

[7] Google Play. CamCard Lite - Business Card R. [En línea] <<https://play.google.com/store/apps/details?id=com.intsig.BCRLite>> [Citado el 18 de mayo de 2014]

[8] Google Play. Business Card Book. [En línea] <<https://play.google.com/store/apps/details?id=mobi.businesscardbook>> [Citado el 28 de Marzo de 2014]

[9] R. Marissa. Tarjetas de presentación. [En línea] <<http://www.gropeimprenta.com/noticias/24-tarjetas-de-presentacion/32-tarjetas-de-presentacion.html>> [Citado el 1 de Marzo de 2012]

[10] González Madera, Arletys. La revolución de los teléfonos “inteligentes”: un medio de comunicación que se transformó hasta convertirse también en una forma de entretenimiento. [En línea] <<http://cybermambi.wordpress.com/2014/04/08/la->

revolucion-de-los-telefonos-inteligentes-un-medio-de-comunicacion-que-se-transformo-hasta-convertirse-tambien-en-una-forma-de-entretenimiento/> [Citado el 8 de Abril de 2014]

[11] Android Developing. The Android Source Code. [En línea] <<http://source.android.com/source/index.html>>

[12] Mann, Mik. INTRODUCCIÓN A ANDROID – Página 2 y Página 5

[13] TOMÁS, Jesús. Índice. El gran libro de Android. Ciudad: México. Editor: Alfaomega Grupo Editor, 2011.

[14] OMG (Object Management Group), "Official UML Specification," 2015. [Online]. Available: <http://www.omg.org/spec/UML/>.

[15] M. Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., 2004.

[16] P. (Rational S. Kruchten, "The 4+1 View Model of Architecture," Software, IEEE, no. November, 1995.

[17] The PHP Group - "Manual de PHP", Conceptos básicos 2015 <<http://php.net/manual/es/intro-what-is.php>>

[18] Oracle - "The Java Tutorials" , About the Java Technology 2015 <<http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>>

[19] W3Schools Home - "XML Tutorial" 2015 <<http://www.w3schools.com/xml/> >

[20] JSON.ORG - "Introducing JSON" 2015 <<http://json.org/>>

[21] Empresamia - "¿Qué es un servidor web?" 2013 <<http://empresamia.com/crear-empresa/crear/item/644-que-es-un-servidor-web>>

[22] Diplomado EASN. ACHING G. César. Ratios Financieros y Matemáticas de la Mercadotecnia. SERIE MYPES, Prociencia y Cultura S.A. Edición 2006.

[23] Young, Ralph. "The Requirements Engineering Handbook". Editorial Artech House. 2004.

[24] Android, API Guides, AppComponents <<http://developer.android.com/guide/components/activities.html>>

[25] API Guides, Android, AppComponents
<<http://developer.android.com/guide/components/fragments.html>>

[26] Álvarez, Miguel Angel “Qué es MVC” Desarrolloweb.com - Enero 2014
<<http://www.desarrolloweb.com/articulos/que-es-mvc.html>>

[27] CodeIgniter “Guía del usuario” 2012
<http://escodeigniter.com/guia_usuario/>

[28] Bootstrap “Introduction”
<<http://getbootstrap.com/>>

ANEXOS

Anexo A, se encuentra adjunto como archivo con el nombre “Resultado – Encuesta Usabilidad.xlsx”

Anexo B, se encuentra adjunto como archivo con el nombre “Encuesta CardMailBox – Formularios de Google.pdf”