



Departamento de Computación

TRABAJO DE DIPLOMA

“Aplicación móvil para el control de eventos, visitas y licencias para la Empresa Provincial de Acopio Villa Clara”

Autor: Victor Manuel Rodríguez Seijo

Tutor: Juan Carlos Ortega Camacho

Curso Académico: 2020-2021

Santa Clara
Copyright©UCLV

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

DIPLOMA THESIS

**"Aplicación móvil para el control, administración y trámite
de eventos, controles, visitas y licencias para la Empresa
Provincial de Acopio Villa Clara"**

Author: Victor Manuel Rodriguez

Tutor: Juan Carlos Ortega Camacho

Este documento es Propiedad Patrimonial de la Universidad Central “Marta Abreu” de Las Villas, y se encuentra depositado en los fondos de la Biblioteca Universitaria “Chiqui Gómez Lubian” subordinada a la Dirección de Información Científico Técnica de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual

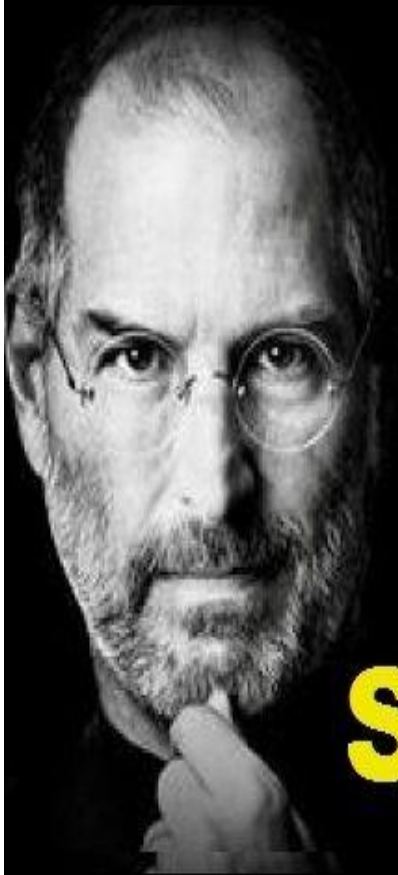


Para cualquier información contacte con:

Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las Villas. Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830
Teléfonos.: +53 01 42281503-1419

Pensamiento

"No dejes que el ruido
de las opiniones ajenas
apaguen tu voz interna."



Steve J  **bs**

Dedicatoria

A todas las personas que me han apoyado en estos años, en especial a mi familia, amistades y compañeros de aula.

Agradecimientos

A mi familia, amigos y compañeros de aula. A todos los que de una forma u otra han contribuido en la realización de este trabajo.

Resumen

En la actualidad, las aplicaciones móviles son utilizadas en numerosos campos como recursos prácticos, intuitivos, portables y de fácil manejo para facilitar tareas en el entorno laboral. La estrategia de usar estas herramientas es utilizada por todo tipo de empresas para realizar con eficacia toda tarea automatizable. El presente trabajo responde a una problemática real que enfrenta la Empresa Provincial de Acopio de Villa Clara, se presenta una aplicación móvil que da solución al proceso de automatizar el trámite, control y administración de los eventos que desarrolla dicha empresa con sus mercados, puntos de venta y demás anexos, que permite a cualquier empleado con un dispositivo móvil usarla para optimizar tiempo, trabajo y esfuerzo en sus tareas cotidianas. Para el desarrollo de la aplicación se utiliza la metodología ágil Mobile-D, lo que permite una reducción del tiempo empleado en este proceso.

Abstract

Currently, mobile applications are used in many fields as practical, intuitive, portable and user-friendly resources to facilitate tasks in the work environment. The strategy of using these tools is used by all types of companies to effectively operate any automatable task. This paper responds to a current problem that the Acopio Provincial Enterprise in Villa Clara faces. A mobile application is presented, which provides a solution to the process of automating the processing, control and administration of the events that the company develops with its markets, points of sale and other annexes and permits any worker with a mobile device can use it in order to optimize time, work and effort in their daily jobs. During application development agile methodology Mobile-D is used, its permits a significant time reduction employed in this process.

Contenidos

Agradecimientos.....	6
Resumen	7
Abstract	8
INTRODUCCIÓN	13
CAPÍTULO I. CONSIDERACIONES TEÓRICAS	16
1.1 Objeto de Estudio.....	16
1.2 Tecnologías Móviles en la Gestión Empresarial	17
1.3 Metodologías Ágiles.....	17
1.2.1 Scrum.....	17
1.2.2 Mobile-D.....	18
1.2.3 Comparativa de Metodologías	21
1.4 Android.....	22
1.3.1 iOS	22
1.3.2 ¿Por qué Android?	22
1.3.3 Ventajas de Android	23
1.3.4 Android Studio.....	24
1.5 Kotlin	25
1.5 SQLite.....	26
Conclusiones del Capítulo	27
CAPÍTULO II: ANÁLISIS Y DEFINICIÓN DE REQUISITOS	28
2.1 Mobile-D	28
2.2 Exploración	28
2.2.1 Establecimiento de los Grupos de Interés o Stakeholders	28
2.2.3 Definición del Alcance	29
2.2.4 Establecimiento del Proyecto	29
2.3 Definición de los Requisitos.....	29
2.3.1 Requisitos Iniciales.....	29
2.3.2 Requisitos Funcionales	29
2.2.3 Requisitos No Funcionales.....	30
2.2.4 Casos de uso del sistema.	30
Conclusiones del Capítulo	32
CAPÍTULO III. DISEÑO E IMPLEMENTACIÓN	33
3.1 Inicialización	33
3.1.1 Configuración del Ambiente de Desarrollo	33

3.1.2 Diagrama de Componentes	33
3.2 Diseño de la Aplicación.....	34
3.2.1 Modelo de Clase	34
3.2.2 Diseño Conceptual de la Base de Datos.....	35
3.2.3 Modelo Entidad-Relación	37
3.3 Descripción de la Interfaz de Usuario.....	37
3.3.1 StoryCard	38
3.3.2 Descripción de las Vistas.	39
3.4 Producción.....	50
3.4.1 Estándares de Codificación.....	51
3.4.2 Codificación.....	51
Conclusiones del Capítulo	54
CONCLUSIONES.....	55
RECOMENDACIONES	56
BIBLIOGRAFÍA	57
ANEXOS	58

Índice de figuras

Figura 1. Ciclo de desarrollo de Mobile-D	19
Figura 2. Porcentaje de usuarios según su tecnología móvil	23
Figura 3. Uso de Kotlin	26
Figura 4. Proceso de Mobile -D	28
Figura 5. Casos de Uso del Sistema	31
Figura 6. Diagrama de Componentes	34
Figura 7. Modelo de Clase	35
Figura 8. Modelo Entidad Relación	37
Figura 9. Mapa de Navegación	38
Figura 10. Pantalla Principal.....	40
Figura 11. Menú Plegable	41
Figura 12. Tipos de Eventos.....	42
Figura 13. Ajustes	43
Figura 14. Vista Diaria	44
Figura 15. Vista Semanal	45
Figura 16. Vista Anual	46
Figura 17. Vista Mes + Día	48
Figura 18. Vista Acerca de	49
Figura 19. Tipo de Mercado	50
Figura 20. Notificaciones	52
Figura 21. Función de Gráficos	53
Figura 22. Exportar pdf.....	53
Figura 23. Dependencias	54

Índice de tablas

Tabla 1. Comparación entre las metodologías	21
Tabla 2. Requisitos funcionales.....	30
Tabla 3. Requisitos no funcionales.....	30
Tabla 4. Descripción de los campos de la tabla de la base de datos	35
Tabla 5. Descripción de la entidad merchants.....	35
Tabla 6. Descripción de la entidad events.....	36
Tabla 7. Descripción de la entidad event tipe.....	36
Tabla 8. Descripción del storycard	38
Tabla 9. Vista Mercados.....	40
Tabla 10. Menú Lateral	41
Tabla 11. Tipos de Eventos.....	42
Tabla 12. Ajustes.....	43
Tabla 13. Vista Diaria	44
Tabla 14. Vista Semanal	45
Tabla 15. Vista Anual	47
Tabla 16. Vista Mes +Día	48
Tabla 17. Vista Acerca de	49
Tabla 18. Vista Estadísticas	50

INTRODUCCIÓN

A lo largo de los últimos años el avance de la tecnología para dispositivos móviles ha sido tal que un gran número de acciones han sido absorbidas completamente por el uso de teléfonos inteligentes, desde la comunicación instantánea entre dos personas, hasta la compra de algún producto por internet. A diario salen a la luz nuevas aplicaciones, cada vez más fáciles y cómodas, que hacen posible la realización de tareas cotidianas desde la palma de la mano.

Es indudable que las ventajas superan los inconvenientes, pues este tipo de tecnología ha ayudado a mejorar muchos aspectos de la vida cotidiana, tales como el acceso a la información, el almacenamiento de datos de forma segura o la compra de manera inmediata y sencilla desde cualquier lugar. (Mojica, 2018)

En el sector agrícola ya se han tenido en cuenta algunas variantes para lograr buenos y mejores resultados usando las tecnologías, ejemplo de esto es la aplicación móvil creada en la UCI para el control del *"trips"* del frijol, la cual permite escanear la hoja del cultivo, otro ejemplo es la aplicación SIMBA, se trata de una herramienta que ayuda a proyectar inversiones y trazar estrategias en cuanto a la forma de aplicación de la maquinaria a los cultivos en los campos. No solo se han desarrollado aplicaciones relacionadas con la forma de cultivar sino también relativas a la correcta y más accesible distribución de los productos, un ejemplo de esto es la muy usada Merca Zona.

En la Empresa Provincial de Acopio Villa Clara existe una limitada cantidad de programas informáticos que se dedican a realizar las tareas de organizar el trabajo y las prioridades de los departamentos, estas son de suma importancia para la optimización del tiempo y el correcto funcionamiento de la entidad, además ayudan a desarrollar una vida laboral más organizada, planificada y eficiente en cuanto al uso del tiempo, de este modo es posible concentrarse en las tareas más importantes. Una aplicación móvil sería muy útil para realizar los trámites y las consultas de los vencimientos de licencias de mercados de toda la provincia, así como el tratamiento que se les otorgará, además de permitir la planificación de las visitas y los controles que se deben realizar.

La empresa presenta dificultades para la tramitación de las tareas y las prioridades que se le deben asignar, uno de los principales problemas es la tramitación de las licencias de venta de cada una de las unidades con las que cuenta la empresa, estas licencias tienen una fecha de vencimiento y no existe una base de datos en la que se guarde esta información.

Otro problema es que cada unidad debe recibir visitas periódicas del departamento de calidad para la comprobación del funcionamiento de los equipos de medición y las condiciones del local, estas visitas se deben cumplir conforme lo señala el plan trimestral abarcando la totalidad de unidades con la que cuenta la provincia. Sería útil contar con una aplicación informática que permita realizar la planificación de estas visitas.

Para dar solución a esta **problemática** nos proponemos como **objetivo general** de este trabajo:

Desarrollar una aplicación móvil que permita gestionar los eventos de la empresa provincial de Acopio Villa Clara, para hacer más eficiente el proceso de planificación de las visitas a los mercados.

Para su cumplimiento se plantean como **objetivos específicos**:

1. Evaluar las metodologías y herramientas existentes para el desarrollo de aplicaciones informáticas sobre el sistema operativo Android.
2. Diseñar una aplicación utilizando la metodología de desarrollo seleccionada.
3. Implementar una aplicación que permita la gestión de los eventos de la empresa Provincial de Acopio Villa Clara.

El desarrollo de este producto **se justifica** plenamente puesto que la actividad que va a realizar la aplicación se hace actualmente en la institución de forma manual, poco ágil y poco óptima, siendo una tarea muy engorrosa para los miembros de la empresa.

Se considera como **hipótesis** que la implementación de una aplicación Android para la gestión de los eventos de la Empresa Provincial de Acopio permitirá realizar esta actividad de manera más ágil y factible por el personal que allí labora.

Una vez concluida esta aplicación traerá consigo un **aporte práctico** a la solución del problema planteado, ya que en la institución se podrá desarrollar una planificación más orientada, dándole tratamiento a las tareas que se enfrentan día a día, empleándose el tiempo de forma correcta, además se facilitará a los trabajadores crear hábitos informáticos para el manejo de este producto y otros similares.

Estructura del documento:

CAPÍTULO I. Consideraciones Teóricas

En este capítulo se lleva a cabo la modelación conceptual y caracterización de las tecnologías de programación a utilizar en el trabajo, y se realiza una descripción teórica de las herramientas que se utilizan.

CAPÍTULO II: Análisis y Definición de Requisitos.

En este capítulo se documentan los artefactos de la aplicación, de acuerdo con las fases de la metodología ágil usada, se incluyen los requisitos funcionales y no funcionales y el diagrama de casos de uso del sistema.

CAPÍTULO III. Diseño e Implementación.

En este capítulo se describen el modelo de entidad relación, el diseño de la base de datos, el diagrama de componentes y los storycards, que representan la principal documentación de la metodología usada. Se confecciona el mapa de navegación, que permite conocer cómo acceder a cada pantalla. Se describen los métodos y clases de más interés de la implementación.

CAPÍTULO I. CONSIDERACIONES TEÓRICAS

En este capítulo se define la misión y visión de la Empresa Provincial de Acopio Villa Clara, se describen los principales conceptos, métodos, tecnologías y aplicaciones existentes en la actualidad, relacionados con la posible solución del problema de investigación planteado, para esto se realiza un estudio detallado de las metodologías de desarrollo de software relacionadas a las aplicaciones móviles, además se analiza el porqué de la utilización de Android y se describen las principales características de las herramientas Kotlin y SQLite.

1.1 Objeto de Estudio

La Empresa Provincial de Acopio Villa Clara se subordina a la Unión Nacional de Acopio (UNA) perteneciente al Ministerio de la Agricultura, es una entidad netamente comercializadora, cuenta con un total de 14 Unidades Empresariales de Base (UEB) en la provincia que se subordinan a la Delegación del Ministerio de la Agricultura Provincial. **Su misión** es garantizar la logística al sector agropecuario en beneficio de un desarrollo próspero y sostenible y la seguridad alimentaria del país, con una óptima atención al cliente, mientras que **su visión** es ser un grupo empresarial de éxito que comercializa y circula insumos y equipamiento de forma mayorista y minorista y presta servicios técnicos y especializados, para esto cuenta con cuadros de alta cultura integral de dirección y colectivos laborales motivados y de alta profesionalidad.

La entidad tiene un amplio flujo de trabajo en todas sus áreas, con una gran red que se encarga de llevar a cabo las actividades comerciales, ejemplo de esto son:

- Ciclo de metrología de los instrumentos: es un proceso cíclico al que cada mercado debe someter sus instrumentos, mayormente de pesaje.
- Aseguramiento metrológico: es un evento que surge a partir de alguna inconformidad o de forma fortuita, ya sea por el control a los mercados o por la sospecha de alguna irregularidad.
- Licencias de higiene: son las licencias con la que cuenta cada mercado para la venta de los productos alimenticios.
- Eventos regulares: estos son para el control de los precios y el buen desempeño del personal.

Con las nuevas transformaciones que sufrió la empresa se dispuso de nuevos tipos de mercados, pues además de los locales tradicionales se utilizan agromóviles (camiones para la distribución de los alimentos) y biciagros (ciclos de distribución), de los que no se tiene una base de datos informatizada; esto complejiza el trabajo en cuanto al control de los mercados. Por otra parte, con las nuevas formas de contratación los mercados pueden estar activos o no para la empresa en un momento dado, sería de mucha ayuda tener un medio automatizado donde registrar este dato.

CAPÍTULO I. Consideraciones Teóricas

Gran parte de las actividades y procesos anteriores son realizados de forma manual, como resultado los trabajadores tienen que ejecutar cada día un conjunto de tareas engorrosas, por lo que se hace necesario la automatización de algunas de estas tareas, para mejorar y facilitar tanto la experiencia de trabajo de los empleados, como la calidad de los servicios de la empresa, fomentando y haciendo énfasis en el uso de las tecnologías para un mejor aprovechamiento.

1.2 Tecnologías Móviles en la Gestión Empresarial

Es indiscutible que actualmente, cualquier negocio va a ser mucho más eficiente en todos los aspectos, si cuenta con dispositivos tecnológicos que faciliten todo tipo de tareas; a partir de las cajas registradoras modernas y los ordenadores, la tecnología ha ido escalando hasta disponer hoy de teléfonos inteligentes que permiten interactuar a los empleados en tiempo real; lo que les permite por ejemplo, mostrar el catálogo de productos a los clientes o utilizar aplicaciones para hacer encuestas de satisfacción sencillísimas y muy rápidas.

La tecnología, y en especial, los dispositivos móviles, hacen que se pueda dar un plus de calidad y eficiencia a la empresa, usándolos en todos los niveles, logrando con ello el máximo de perfección, consiguiéndose así una mejor gestión de la empresa.

Los dispositivos móviles en la actualidad están presentes en todas las esferas de la vida, tanto en lo profesional como lo personal, y a pesar de que no hace mucho resultaba complejo construir aplicaciones para ellos, hoy en día es relativamente sencillo desarrollar programas, basta tomar algo de la abundante literatura existente, elegir un lenguaje de programación y descargar el software necesario que permita la instalación de un entorno de desarrollo apropiado para dicha actividad (García Rafael, 2018).

1.3 Metodologías Ágiles.

Aparecieron de forma experimental en el umbral de los años 90, pero fue en el año 2001 en el que apenas nació el término como tal debido a la aparición del manifiesto ágil, resultado de una reunión en la que intervinieron sus representantes y otras personas interesadas, para tratar temas relativos al desarrollo de software y conformar lo que luego se conocería como Alianza Ágil (Gómez, Ruiz, & López, 2016).

En los últimos años se han utilizado varios métodos ágiles por investigadores en diversos campos, a continuación nos referiremos a dos de los procesos ágiles más populares.

1.2.1 Scrum

Esta metodología centra la gestión de proyectos en situaciones en las que es difícil planificar el futuro, con mecanismos de control “proceso empírico”, donde los bucles de realimentación constituyen el elemento central. El software es

CAPÍTULO I. Consideraciones Teóricas

desarrollado por un equipo de auto-organización en incrementos (llamados “sprints”), empezando por la planificación y finalizando con un comentario. Las características que deben aplicarse en el sistema se registran en un backlog. Entonces, el dueño del producto decide qué elementos del backlog se deben desarrollar en un sprint.

Un Sprint corresponde a una ventana de tiempo donde se crea una versión utilizable del producto (incremento). Cada Sprint es considerado como un proyecto independiente. Su duración máxima es de un mes. Un Sprint se compone de los siguientes elementos: reunión de planeación del Sprint, Daily Scrum, trabajo de desarrollo, revisión del Sprint y retrospectiva del Sprint. Utiliza un enfoque incremental que tiene como fundamento la teoría de control empírico de procesos. Esta teoría se fundamenta en transparencia, inspección y adaptación; la transparencia, que garantiza la visibilidad en el proceso de las cosas que pueden afectar el resultado; la inspección, que ayuda a detectar variaciones indeseables en el proceso; y la adaptación, que realiza los ajustes pertinentes para minimizar el impacto de las mismas (Cadavid, Martínez, & Vélez, 2013).

Scrum define tres roles:

1. Scrum master, el dueño del producto y el equipo de desarrollo. El Scrum master tiene como función asegurar que el equipo está adoptando la metodología, sus prácticas, valores y normas; es el líder del equipo pero no gestiona el desarrollo.
2. El dueño del producto es una sola persona y representa a los interesados, es el responsable de maximizar el valor del producto y el trabajo del equipo de desarrollo; tiene entre sus funciones gestionar la lista ordenada de funcionalidades requeridas o Product Backlog.
3. El equipo de desarrollo, por su parte, tiene como responsabilidad convertir lo que el cliente quiere, el Product Backlog, en iteraciones funcionales del producto; el equipo de desarrollo no tiene jerarquías, todos sus miembros tienen el mismo nivel y cargo: desarrollador. El tamaño óptimo del equipo está entre tres y nueve personas.

Los miembros del equipo coordinan su trabajo en un diario de stand-up de la reunión. Un miembro del equipo, el “Scrum Master” (equivalente al gerente del proyecto), es el encargado de resolver los problemas que impiden que el equipo trabaje eficazmente, este equipo generalmente es de diez o menos componentes (Balaguera, 2013).

1.2.2 Mobile-D

Una metodología de desarrollo, especialmente diseñada para el desarrollo de aplicaciones móviles, recibe el nombre de Mobile-D y es propuesta por Pekka Abrahamsson y su equipo del VTT (Valtion Teknillinen Tutkimuskeskus, en inglés

CAPÍTULO I. Consideraciones Teóricas

Technical Research Centre of Finland) en Finlandia que lideran una corriente muy importante de desarrollo ágil muy centrada en las plataformas móviles. Las prácticas asociadas a Mobile-D incluyen desarrollo basado en pruebas, la programación en parejas, integración continua y refactorización, así como las tareas de mejora de procesos de software; según Abrahamsson Mobile-D debe ser utilizado por un equipo de no más de diez desarrolladores, trabajando en conjunto para suministrar un producto listo en un plazo máximo de diez semanas (Balaguera, 2013).

La aproximación de Mobile-D se ha apoyado en muchas otras soluciones bien conocidas y consolidadas: Extreme Programming (XP), Crystal Methodologies y Rational Unified Process (RUP). Los principios de programación extrema se han reutilizado en lo que se refiere a las prácticas de desarrollo, las metodologías Crystal proporcionaron un input muy valioso en términos de la escalabilidad de los métodos y el RUP es la base para el diseño completo del ciclo de vida.

En la figura 1 se muestran las cinco fases de Mobile-D: exploración, iniciación, producción, estabilización y prueba del sistema. Cada una de estas fases tiene un número de etapas, tareas y prácticas asociadas.



Figura 1. Ciclo de desarrollo de Mobile-D

Fase de Exploración:

En esta fase el equipo de desarrollo debe generar un plan y establecer las características del proyecto, esto se realiza en tres etapas: establecimiento de actores, definición del alcance y establecimiento de proyectos. Las tareas

CAPÍTULO I. Consideraciones Teóricas

asociadas a esta fase incluyen el establecimiento del cliente (los clientes que toman parte activa en el proceso de desarrollo), la planificación inicial del proyecto, los requisitos de recogida y el establecimiento de procesos.

Fase de Inicialización:

El propósito de esta fase es posibilitar el éxito de las siguientes fases del proyecto preparando y verificando todos los problemas críticos del desarrollo, de manera que sean corregidos con prontitud al finalizar la fase de aplicación de los requisitos, además se preparan todos los recursos físicos, tecnológicos y de comunicaciones para las actividades de producción.

Fase de Producción:

En esta fase se repite iterativamente la programación de tres días (planificación, trabajo, liberación) hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar, para esto se preparan las pruebas de la iteración de antemano. Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el último día se lleva a cabo la integración del sistema (en caso de que estuvieran trabajando varios equipos de forma independiente), seguida de las pruebas de aceptación.

Fase de Estabilización:

En la fase de estabilización es donde se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funcione correctamente. Esta será la fase más importante en los proyectos multi-equipo formados por diferentes subsistemas desarrollados por equipos distintos. En esta fase los desarrolladores realizarán tareas similares a las que debían desplegar en la fase de “producción”, aunque en este caso todo el esfuerzo se dirige a la integración del sistema. Adicionalmente se puede considerar en esta fase la producción de documentación.

Fase de Pruebas del Sistema:

La última fase (pruebas y reparación del sistema) tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos del cliente y se eliminan todos los defectos encontrados.

Ventajas de Mobile-D:

1. Los costos son bajos cuando se necesita realizar algún cambio en el proyecto.
2. La entrega de resultados es rápida.
3. Se enfoca a trabajar con grupos de trabajo pequeños.
4. Permite mejorar el producto mediante iteraciones pequeñas.
5. Se centra en la satisfacción del usuario final.
6. En la liberación del producto la densidad de defectos es baja.
7. La comunicación entre los miembros del o de un equipo es constante.

CAPÍTULO I. Consideraciones Teóricas

8. Mobile-D cuenta con certificación CMMI de nivel 2.
9. Se basa en el desarrollo basado en pruebas que es una de las mejores formas de asegurar la calidad.

1.2.3 Comparativa de Metodologías

Tabla 1. Comparación entre las metodologías

	Metodología Scrum para Móviles	Metodología Mobile-D
Fases	Se define por Sprints de dos semanas aproximadamente y de un producto entregable.	Fases: exploración, inicialización, producción, estabilización y prueba del sistema.
Comunicación con el cliente	El Product Owner maneja la comunicación con el cliente	La comunicación con el cliente es menor que en Scrum.
Programación	El tiempo de programación se determina de acuerdo a la puntuación dada a cada tarea.	Es la parte más importante, y que mayor tiempo posee junto con las pruebas.
Documentación	Lo principal lo constituyen las historias de usuario.	Lo principal es el modelo del dominio, storycards y storyboards.
Pruebas	Se realizan pruebas unitarias, pruebas de integración, pruebas de aceptación.	Se realizan pruebas unitarias, pruebas de interfaz, pruebas de datos ingresados y pruebas funcionales.

En la tabla 1 se puede observar que Mobile-D tiene fases más definidas: análisis, diseño, codificación y pruebas; en cambio Scrum se caracteriza por ser iterativo, se podría decir que realiza las fases de análisis, diseño, codificación y pruebas en cada iteración. La documentación en Scrum tiene un nivel bajo en comparación con la metodología Mobile-D, en cambio Mobile-D no exige una gran cantidad de documentación, solo lo necesario para guiarse en el desarrollo de la aplicación.

A partir del análisis de las metodologías anteriores, teniendo en cuenta que el grupo de desarrollo estuvo conformado por una persona y se disponía de poco

CAPÍTULO I. Consideraciones Teóricas

tiempo, se decidió elegir la metodología Mobile-D, ya que esta fue creada especialmente para el desarrollo de aplicaciones móviles y la entrega de resultados es rápida.

1.4 Android

Android es un sistema operativo basado en el kernel de Linux y diseñado principalmente para dispositivos móviles con pantalla táctil, tales como teléfonos inteligentes o tabletas. Fue inicialmente desarrollado por Android Inc, y respaldado económicamente por Google, que más tarde, en el año 2005 adquirió la empresa.

Uno de los aspectos fundamentales del sistema operativo de Android fue su orientación a la multiplataforma, algo realmente novedoso, pues anteriormente un sistema operativo se asociaba a un único dispositivo. Esta característica rápidamente hizo que Android alcanzara sus objetivos, convirtiéndose en el sistema operativo más utilizado (Gómez Matesanz, 2014).

1.3.1 iOS

iOS es el sistema operativo que se ejecuta en dispositivos iPad, iPhone y iPod touch. El sistema operativo gestiona el hardware del dispositivo y proporciona las tecnologías necesarias para implementar aplicaciones nativas. Este sistema operativo también incluye varias aplicaciones del sistema, como Phone, Mail y Safari, que proporcionan servicios estándar al usuario (Correa Ríos, 2014).

1.3.2 ¿Por qué Android?

Se utiliza Android para darle solución a la problemática, primeramente analicemos que la mayoría de los dispositivos con que cuentan los trabajadores de la empresa son de esa tecnología y al ser este sistema operativo para móviles resulta idóneo, en la actualidad prácticamente el 100% de la personas utilizan los móviles como un asistente cotidiano además no se pudo dejar de analizar las características de Android y sus muchas de las ventajas.

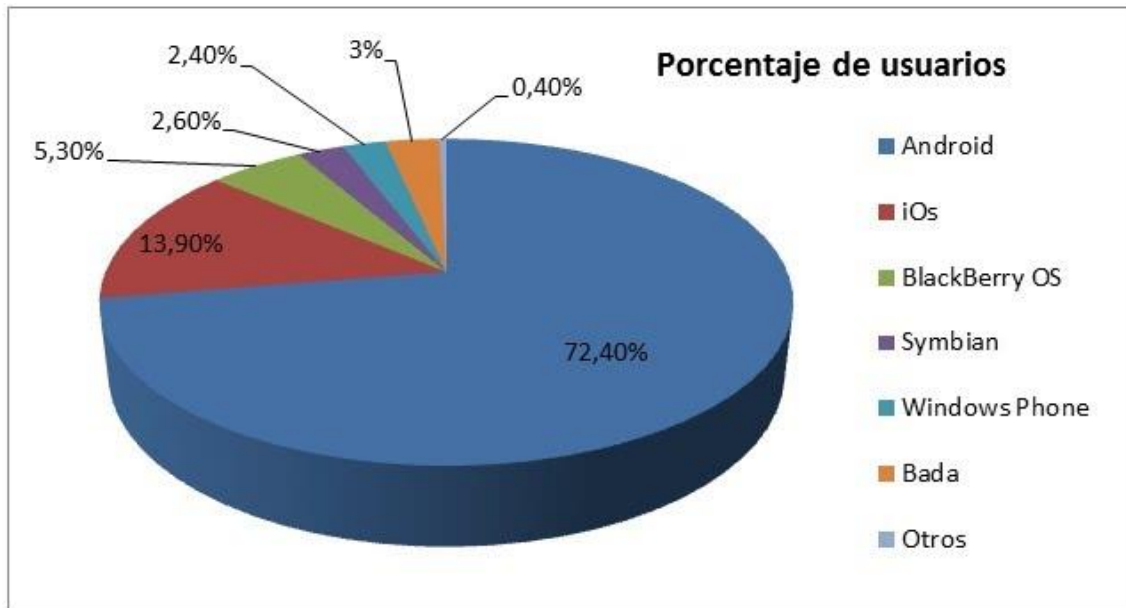


Figura 2. Porcentaje de usuarios según su tecnología móvil

Características de Android:

- Código abierto.
- Núcleo basado en el Kernel de Linux.
- Adaptable a muchas pantallas y resoluciones.
- Utiliza SQLite para el almacenamiento de datos.
- Ofrece diferentes formas de mensajería.
- Navegador web basado en WebKit incluido.
- Soporte de Java y muchos formatos multimedia.
- Soporte de HTML, HTML5, Adobe Flash Player, etc.
- Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software.
- Catálogo de aplicaciones gratuitas o pagas en el que pueden ser descargadas e instaladas (Google Play).
- Bluetooth.
- Google Talk desde su versión HoneyComb, para realizar videollamadas.
- Multitarea real de aplicaciones.

1.3.3 Ventajas de Android

1. Código abierto

La primera ventaja de Android, y la más notable, es que funciona con un código abierto, lo que se traduce en que cualquiera puede crear aplicaciones y contribuir a aumentar la oferta de esta en el universo Android, siendo la gran mayoría de ellas, gratuitas.

CAPÍTULO I. Consideraciones Teóricas

El hecho de que su código sea abierto, también posibilita una gran variedad de ventajas más, como por ejemplo el hecho de que los errores puedan ser revisados y reparados con mayor rapidez, y otras ventajas a continuación.

2. Mayor libertad

Otra característica positiva que define este sistema operativo, que se desprende de la primera, es la libertad total con la que trabajan los desarrolladores de Android para realizar todo tipo de aplicaciones sin límites y sin pedir ningún tipo de permiso.

Además de eso, también es destacable el hecho de que Android no depende de ningún fabricante u operadora para implementar las mejoras del sistema, lo que redundará en su nivel de libertad.

3. Diversidad y versatilidad

Por todo lo comentado anteriormente, Android es también el sistema operativo que cuenta con un mayor número de fabricantes y está presente en la mayoría de marcas comerciales, así como de usuarios, lo que lo convierte en el sistema más utilizado.

A ese hecho también añadir la versatilidad de dispositivos en los que se encuentra este sistema operativo además de teléfonos móviles, tabletas, relojes inteligentes, ordenadores, dispositivos GPS y toda clase de electrodomésticos de última generación que dispongan de la tecnología

1.3.4 Android Studio

Desde noviembre del 2007 fue liberado el SDK para desarrollar aplicaciones para el Sistema Operativo Android, Eclipse era uno de los pocos software utilizados para desarrollar aplicaciones nativas para Android. Pero Google decide apostar por crear su propio IDE en conjunto con IntelliJ IDEA (uno de los IDE para java de primer nivel), de esa fusión de conocimientos nace el nuevo y mejorado Android Studio.

Android Studio se perfila desde el año 2014 como el único e inmejorable IDE para desarrollar aplicaciones Android, no sólo el hecho de que este es multiplataforma sino que, no se debe instalar el SDK por separado tal como se hacía anteriormente con Eclipse, ya que este por ser promocionado por Google ya viene incorporado en sus diferentes instaladores, el único requisito que se solicita tener instalado es la versión más reciente del JDK (Java Developer Kit).

Entre las características positivas que posee están que utiliza Gradle lo que facilita en gran medida reusar código y recursos, facilita configurar, extender y personalizar el proceso, facilita la distribución del código y por tanto trabajar en equipos. Gestiona las dependencias de una forma cómoda y potente (está basado en Maven). Hace increíblemente fácil crear distintas versiones de la

CAPÍTULO I. Consideraciones Teóricas

aplicación, por ejemplo, para distintos dispositivos, o una versión de prueba que carga distintos recursos, apunta a webservices distintos. Siendo el IDE oficial para Android.(Cinar, 2015)

1.5 Kotlin

Kotlin es un lenguaje de programación pragmático pensado para funcionar con Máquina Virtual de Java (JVM) y Android. Además, puede ser compilado a código fuente de Javascript(Vogel, 2021).

Puede ser utilizado en cualquier tipo de desarrollo, basado en servidor, web de cliente o Android. Dispone de soporte para otras plataformas como sistemas integrados macOS e iOS. Actualmente, su uso está muy extendido en el desarrollo de aplicaciones móviles en Android, pero también se utiliza en data science, proyectos del lado servidor y del lado cliente con JavaScript JavaFX.

Se caracteriza por una perfecta combinación de características claramente orientadas a la funcionalidad durante la programación, centrándose en la seguridad, la claridad y la interoperabilidad. Los programas en Kotlin pueden utilizar los frameworks y librerías de Java existentes. Así mismo, su interoperabilidad no requiere capas de adaptación.

Fue bautizado en honor a la isla del mismo nombre, cercana a San Petersburgo, sede de la empresa JetBrains, cuyo equipo de programadores se encargó de su desarrollo.

Su sintaxis es igualmente intuitiva. Tiene un considerable parecido con Scala por ejemplo, los puntos y comas como cierre de frase son opcionales, y los saltos de línea suelen bastar para que el compilador entienda que la declaración ha terminado, no obstante, es más simple y permite una compilación más rápida. La adopción de Kotlin presenta un riesgo muy bajo gracias a su interoperabilidad con Java.

Puede ser puesto a prueba en tan solo una parte del código, sin necesidad de alterar la totalidad del proyecto. Se exporta una API de Java, cuya apariencia es idéntica al código de Java. Si, finalmente, se decide continuar utilizando Kotlin, se puede migrar el código existente o mantener el código Java legado, utilizando Kotlin para las nuevas clases y funcionalidades(Mesonero Rodríguez, 2021).

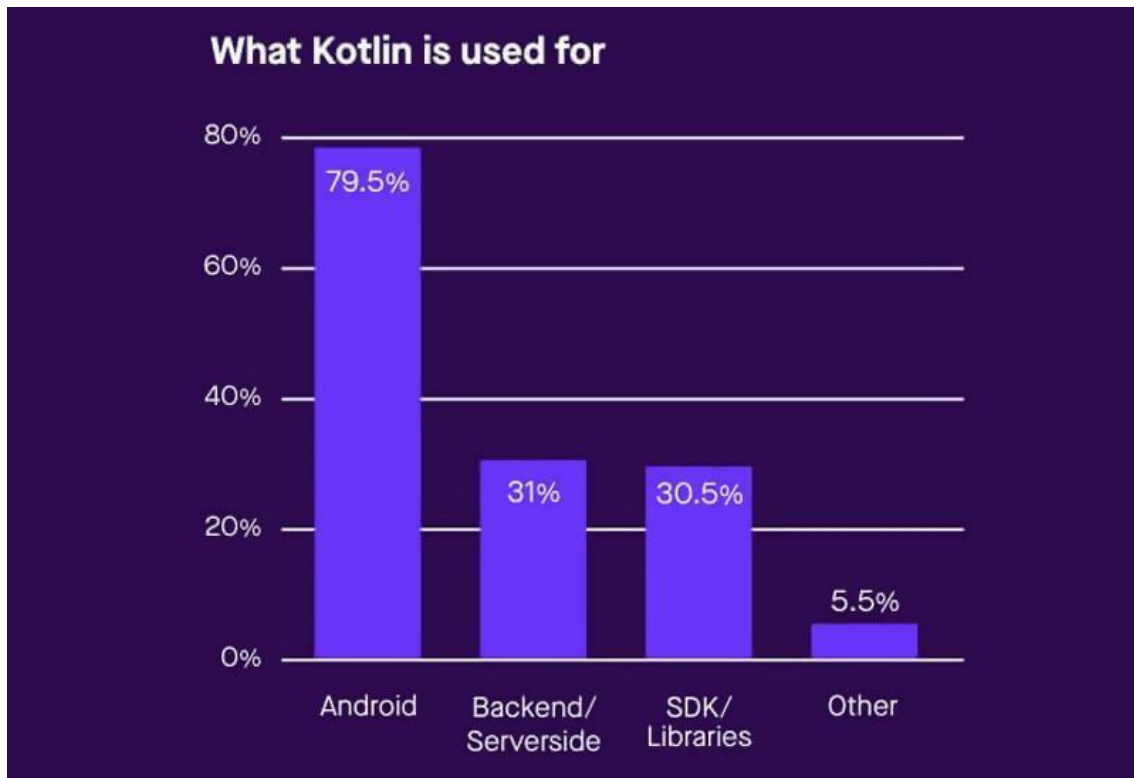


Figura 3. Uso de Kotlin

1.5 SQLite

SQLite es una herramienta de software libre, que permite almacenar información en dispositivos empujados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser una PDA o un teléfono celular. SQLite implementa el estándar SQL92 y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo. Esto permite que SQLite soporte desde las consultas más básicas hasta las más complejas del lenguaje SQL, y lo más importante es que se puede usar tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos, ya que existe compatibilidad al 100% entre las diversas plataformas disponibles, haciendo que la portabilidad entre dispositivos y plataformas sea transparente (Narváez Argeñal, Pérez Rodríguez, & Vallecillo Dávila, 2019).

SQLite apareció en mayo del año 2000 de la mano de su creador D. Richard Hip, quién ha liberado las diferentes versiones de SQLite en base a la licencia GPL (Licencia Publica General) por lo que su código es de dominio público y puede ser modificado por cualquier persona. Gracias a esto, SQLite ha sido mejorada a lo largo de 7 años por un gran número de colaboradores y también ha sido migrada a diversas plataformas. (González Márquez (2019))

Características principales de SQLite:

CAPÍTULO I. Consideraciones Teóricas

- La base de datos completa se encuentra en un solo archivo.
- Puede funcionar enteramente en memoria, lo que la hace muy rápida.
- Es totalmente autocontenida (sin dependencias externas).
- Cuenta con librerías de acceso para muchos lenguajes de programación.
- Soporta texto en formato UTF-8 y UTF-16, así como datos numéricos de 64 bits.
- Soporta funciones SQL definidas por el usuario (UDF).
- El código fuente es de dominio público y se encuentra muy bien documentado.

Conclusiones del Capítulo

Se describieron los principales conceptos, métodos, tecnologías y aplicaciones existentes en la actualidad, relacionados con la posible solución del problema de investigación planteado, además de lo anterior:

- Se realizó un estudio de las metodologías de desarrollo de software relacionadas con el desarrollo de aplicaciones móviles, escogiéndose la metodología Mobile-D.
- Se seleccionó Android para el desarrollo de la aplicación basado en sus múltiples ventajas y por ser el medio que la gran mayoría de los trabajadores de la empresa tienen a su disposición.
- Se determinaron las principales herramientas que se van a utilizar para el desarrollo de la aplicación: el lenguaje de programación Kotlin, SQLite para la implementación de la base de datos y el ambiente de desarrollo Android Studio.

CAPÍTULO II: ANÁLISIS Y DEFINICIÓN DE REQUISITOS

En este capítulo se documentan los artefactos de la aplicación, de acuerdo con las fases de la metodología ágil Mobile-D, metodología seleccionada para la concepción y desarrollo de la aplicación. Se describen los requisitos funcionales y no funcionales, además de los artefactos gráficos para el desarrollo de la aplicación.

2.1 Mobile-D

Esta metodología se utiliza especialmente en las pequeñas empresas debido a los tiempos cortos de desarrollo, lo que produce como resultado la minimización de los costos de producción y garantiza que sea asequible para pequeñas organizaciones que tienen poco personal y recursos (Muñoz Muñoz, 2020).

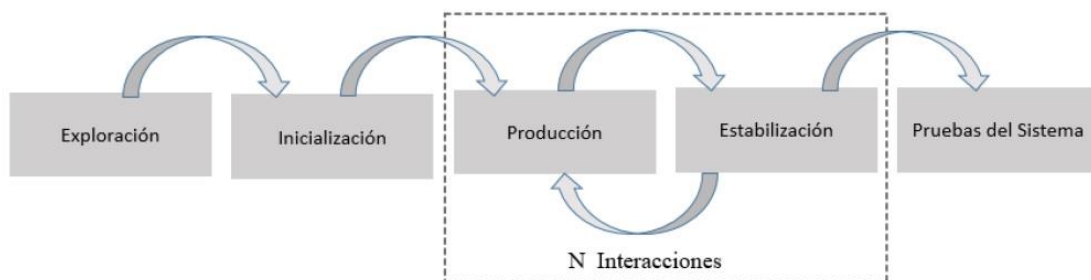


Figura 4. Proceso de Mobile -D

2.2 Exploración

En esta fase se definen los requisitos, el alcance, así como las bases fundamentales para el desarrollo de un producto consecuente con lo que se desea.

Los objetivos de esta fase son establecer los grupos de partes interesadas, que son uno de los principales indicadores de lo que se espera de la aplicación móvil, definir los objetivos de la aplicación móvil y elegir qué miembros van a participar en el desarrollo de la aplicación móvil.

2.2.1 Establecimiento de los Grupos de Interés o Stakeholders

Los grupos de interés involucrados en este proyecto son:

- **Desarrollador:** es la persona encargada del análisis, desarrollo y pruebas de la aplicación.
- **Empleados:** son los trabajadores de la empresa que van a hacer uso de la aplicación para recolectar información.
- **Directivos empresariales:** se refiere a todo personal de la empresa donde se utilice la aplicación “MercaEventos”, que posea autorización para acceder a la información recolectada.

CAPÍTULO II: Análisis y Definición de Requisitos

2.2.3 Definición del Alcance

El alcance del proyecto se determina a través de las limitaciones y de los supuestos y dependencias.

Las limitaciones son:

- La aplicación sólo puede ser ejecutada en dispositivos con el sistema operativo Android.
- Para el trabajo con la aplicación es necesario tener conocimiento de los atributos de los mercados, así como de los eventos a programar.

Los supuestos y dependencias son:

- La aplicación no es compatible para los dispositivos de versión Android antiguas, deben ser superior a la versión 5.0 de Android.
- Los usuarios manejan como idioma principal el español, siendo éste el lenguaje que se muestra en la interfaz de la aplicación.

2.2.4 Establecimiento del Proyecto

En esta etapa indicamos que tipo de herramientas se usaron para el desarrollo de la aplicación móvil.

Las herramientas utilizadas son:

- Android Studio: ambiente integrado para el desarrollo de aplicaciones Android
- Kotlin: lenguaje de programación
- SQLite: herramienta para la creación y manejo de la base de datos

2.3 Definición de los Requisitos

La especificación de los requisitos de un sistema es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describen todas las interacciones que tendrán los usuarios con la aplicación.

2.3.1 Requisitos Iniciales

Se pretende desarrollar una aplicación móvil para el sistema operativo Android que permita el correcto control y tramite de las tareas de los mercados de toda Villa Clara con la empresa provincial de Acopio de dicha provincia.

2.3.2 Requisitos Funcionales

Los requisitos funcionales son las capacidades o funciones que el sistema debe cumplir, pueden ser frases muy generales sobre lo que el sistema debería hacer.

Para lograr el cumplimiento de los objetivos presentados, las funcionalidades que el sistema debe cumplir son:

CAPÍTULO II: Análisis y Definición de Requisitos

Tabla 2. Requisitos funcionales

Código	Descripción
RF1	Crear mercados y sus eventos
RF2	Programar eventos
RF2	Listar mercados.
RF3	Listar eventos.
RF4	Permitir el trabajo con el calendario.
RF5	Visualizar información de los mercados y sus eventos asociados.
RF6	Permitir la edición de los datos de los mercados.
RF7	Permitir la edición de los datos de los eventos.
RF8	Actualización de la información.
RF9	Notificar eventos.
RF10	Crear y eliminar tipos de eventos.
RF11	Crear la valoración estadística

2.2.3 Requisitos No Funcionales

Los requisitos no funcionales son restricciones que afectan a los servicios funcionales del sistema, son propiedades o cualidades que el producto debe tener, representan las características que hacen al producto atractivo, usable, rápido y confiable. Estos requisitos pueden ser más críticos que los funcionales, pudiendo marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

A continuación, se dan a conocer los requisitos no funcionales del sistema:

Tabla 3. Requisitos no funcionales

Código	Descripción
RNF1	El programa será desarrollado para la plataforma Android.
RNF2	El programa será desarrollado bajo el lenguaje de programación Kotlin. .
RNF3	El programa utilizará interfaz amigable.
RNF4	El sistema permitirá el acceso a los usuarios sin necesidad de autenticación.
RNF5	El sistema debe ser fácilmente instalable por los usuarios en equipos móviles con la versión del sistema operativo Android 5.0 o superiores
RNF6	Mostrar Gráficos estadísticos comprensibles

2.2.4 Casos de uso del sistema.

CAPÍTULO II: Análisis y Definición de Requisitos

La metodología Mobile-D no hace uso del siguiente diagrama UML, pero se consideró que en el presente trabajo se incorpora el diagrama de casos de uso del sistema para mayor claridad y comprensión.

Un caso de uso describe una interacción entre el sistema y un agente externo, capta siempre una función visible para el usuario, logra un objetivo concreto y específico para el usuario y puede ser algo simple o algo complejo (Hernández, 2018).

Los casos de uso del sistema son:

CU 1: Gestionar mercados.

CU 2: Gestionar eventos.

CU 3: Gestionar tipos de evento.

CU 4: Consultar estadísticas.

En la figura 5 se pueden observar los casos de uso del sistema con sus atributos.

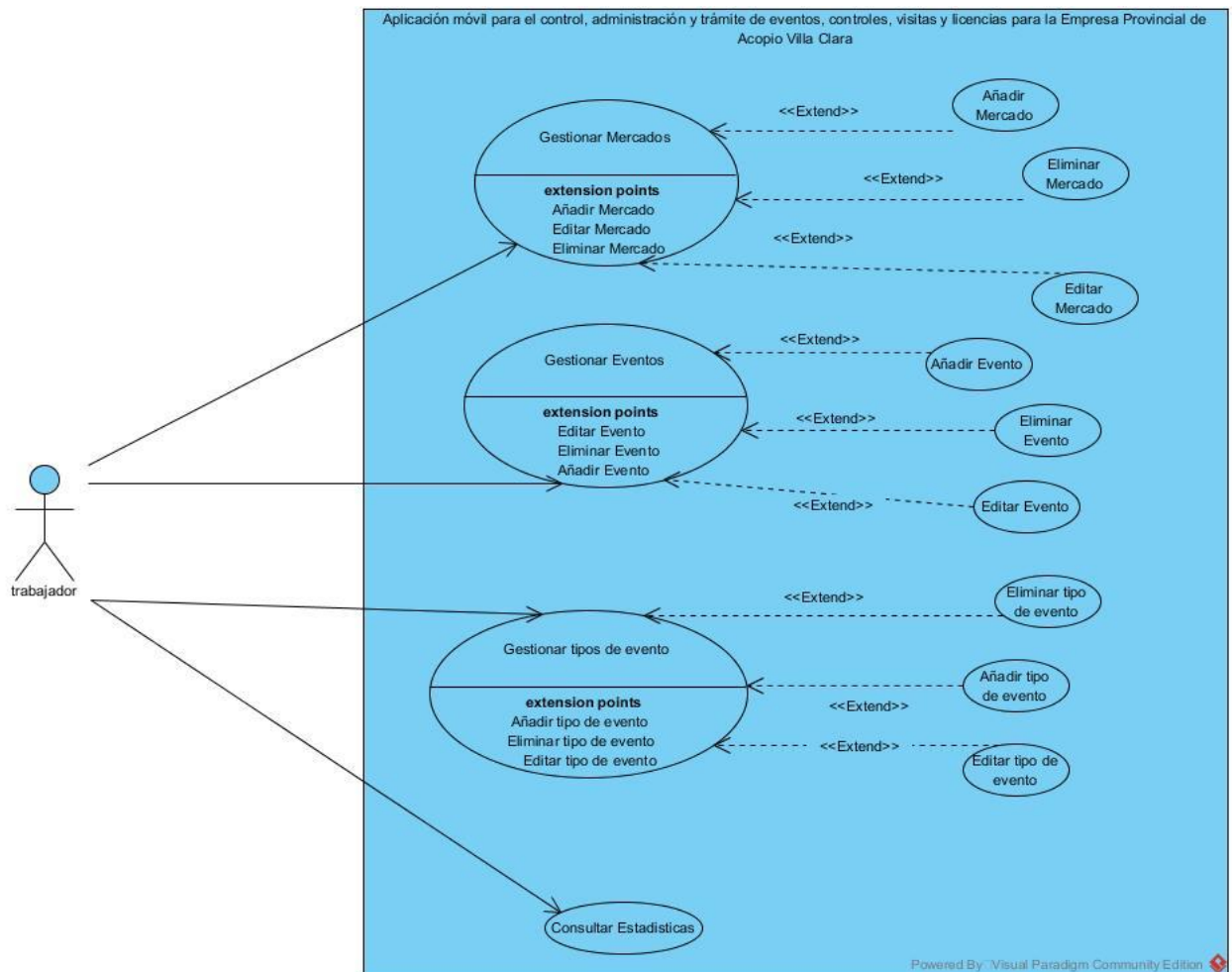


Figura 5. Casos de Uso del Sistema

A continuación se describen en más detalle los principales casos de uso:

CAPÍTULO II: Análisis y Definición de Requisitos

Gestionar mercados: este caso de uso permite al trabajador el tratamiento de los mercados, para tener una información actualizada de cada uno de los atributos que los definen.

Gestionar eventos: este caso de uso se encarga de los eventos asociados a los mercados, que los trabajadores deben gestionar de forma óptima.

Gestionar tipos de evento: cada evento tiene sus características y necesitan de una habilidad especial de cada trabajador, además cada tipo de evento tiene su prioridad en el ámbito interno del departamento.

Consultar estadísticas: permite consultar los gráficos que se generan como resultado del procesamiento estadístico de la información.

Conclusiones del Capítulo

Se documentaron los principales artefactos de la aplicación, acorde a las diferentes etapas de la primera fase o fase de exploración de la metodología Mobile-D, en este sentido se definieron los grupos de interés, los objetivos y el alcance de la aplicación y se describieron las herramientas que se usaron durante el desarrollo del proyecto

Se realizó una descripción completa del comportamiento del sistema, con este fin se definieron los requisitos funcionales y no funcionales y se establecieron los casos de uso del sistema.

CAPÍTULO III. DISEÑO E IMPLEMENTACIÓN

Se describen los métodos y clases de más interés de la implementación, además se describe el modelo de entidad relación y se hace el diseño de la base de datos. Se muestra el storyboard o mapa de navegación, que permite conocer cómo acceder a cada pantalla y los storycard, que representan la principal documentación de la metodología usada, estos muestran visualmente como se observará cada pantalla de la aplicación.

3.1 Inicialización

En esta fase se realizan las actividades relacionadas a la configuración del ambiente de desarrollo y al diseño de la aplicación.

3.1.1 Configuración del Ambiente de Desarrollo

En esta actividad se realizó la configuración de los elementos técnicos, en este caso se configuró el entorno de desarrollo para aplicaciones Android.

Configuración para desarrollo de aplicaciones móviles Android:

- Tipo de Proyecto: Android Application Project.
- Configuraciones: Inclusión de las bibliotecas Picasso para el trabajo con imágenes, *joda_time* para el trabajo con el calendario y *room* para el trabajo con la base de datos.

3.1.2 Diagrama de Componentes

El diagrama de componentes proporciona una visión física de la construcción del sistema de información muestra la organización de los componentes de software, sus interfaces y las dependencias entre ellos.

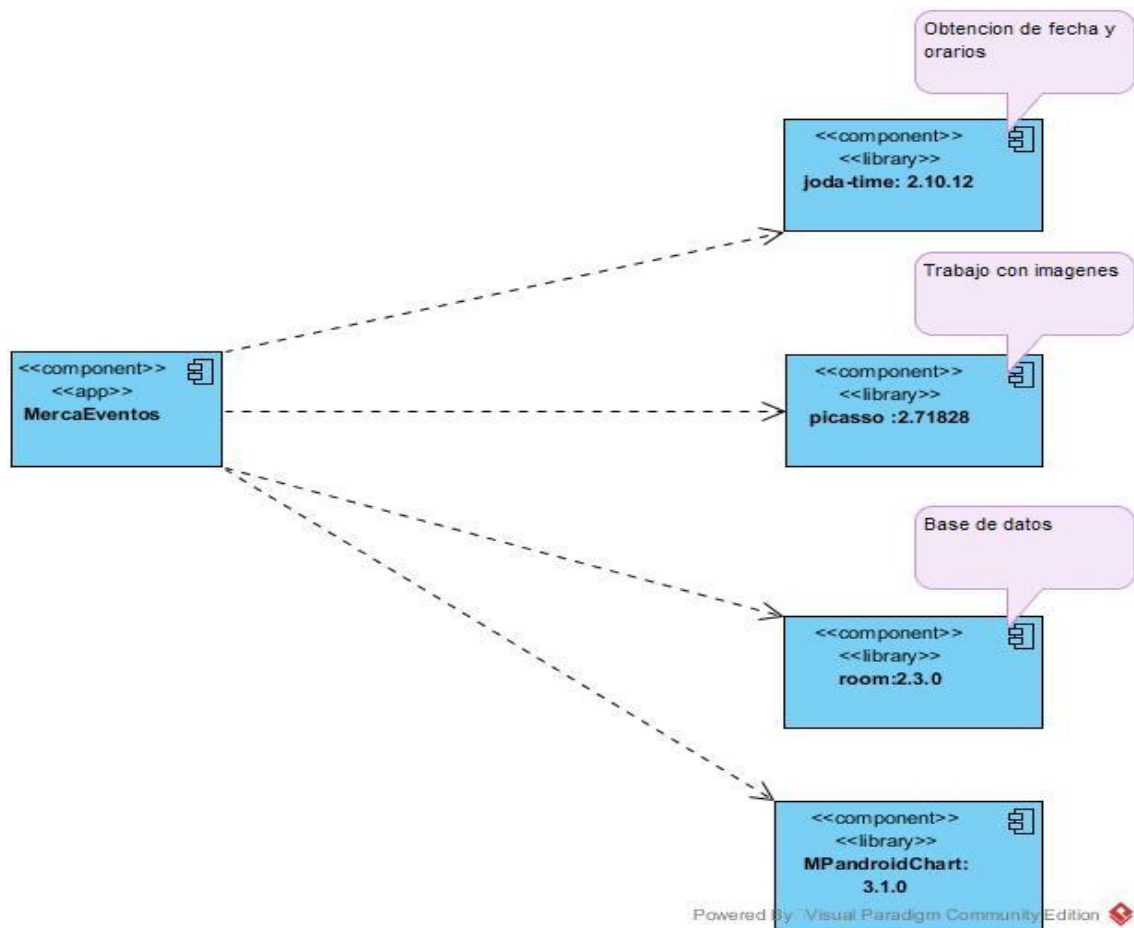


Figura 6. Diagrama de Componentes

3.2 Diseño de la Aplicación

A través del diseño se obtienen los diagramas que representan el funcionamiento de la aplicación y de los procesos que se van a ejecutar, con el fin de cumplir los requisitos definidos en la primera fase.

3.2.1 Modelo de Clase

A continuación se detallan las clases fundamentales que componen la aplicación "MercaEventos" siendo la clase *events* la más amplia ya que cuenta con la mayor cantidad de atributos, como se puede observar en la figura 7.

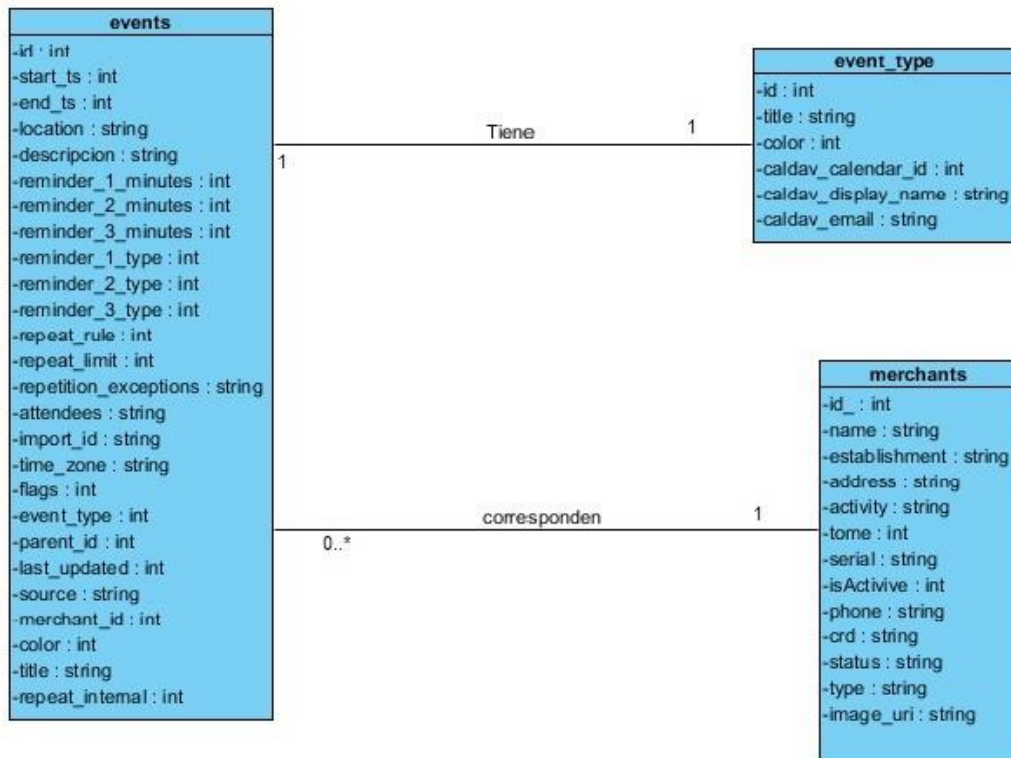


Figura 7. Modelo de Clase

3.2.2 Diseño Conceptual de la Base de Datos

En la elaboración del diagrama conceptual de la base de datos se identifican y analizan las entidades que comprenden el modelo de los datos que serán utilizados para el almacenamiento.

Tabla 4. Descripción de los campos de la tabla de la base de datos

Campos de las Tablas	
Entidad	Nombre de la Entidad
Atributo	Nombre del campo que define una característica de una entidad
Tipo	Define el tipo de dato del atributo
Longitud	Tamaño máximo de caracteres del atributo
Descripción	Detalla o describe al atributo

Diseño de entidad *merchants*:

La información del mercado será almacenada en el dispositivo móvil para que esté disponible.

En la tabla 5 se muestra el tipo y la descripción de los principales atributos de la entidad *merchants*.

Tabla 5. Descripción de la entidad *merchants*

CAPÍTULO III. Diseño e Implementación

Descripción de entidad <i>merchants</i>			
Atributo	Tipo	Longitud	Descripción
+id : integer(10)	integer	10	Id del Mercado
name	Varchar	100	Clave primaria que identifica al tipo de evento
Establishment	Varchar	100	Nombre que se le otorga al evento
address	Varchar	100	Dirección del Mercado
image_uri	Varchar	100	Foto del Mercado
activity	integer	100	Actividades que realiza
status	Varchar	10	Estado del Mercado

Diseño de entidad *events*:

La información de los eventos será almacenada en el dispositivo móvil para el que esté disponible.

En la tabla 6 se muestra el tipo y la descripción de los principales atributos de la entidad *events*.

Tabla 6. Descripción de la entidad events

Descripción de entidad <i>events</i>			
Atributo	TIPO	Longitud	Descripción
id	integer	10	Clave primaria que identifica al evento
title	Varchar	100	Nombre que se le otorga al evento
location	Varchar	100	Localización del mercado
description	Varchar	100	Descripción del evento
merchant_id	integer	10	Mercado Asociado al Evento
event_type	integer	10	Tipo de evento asociado
start_ts	integer	10	Comienzo del Evento
end_ts	integer	10	Fin del evento
repeat_interval	integer	10	Repetición del intervalo

Diseño de Entidad *events_type*:

La información de los tipos de evento será almacenada en el dispositivo móvil para el que esté disponible.

En la tabla 7 se muestra el tipo y la descripción de los principales atributos de la entidad *events_type*.

Tabla 7. Descripción de la entidad event tipe

Descripción de Entidad <i>event_type</i>			
Atributo	TIPO	Longitud	Descripción

id	integer	10	Clave primaria que identifica al tipo de evento
tittle	Varchar	100	Nombre que se le otorga al evento
color	integer	10	Color asociado al tipo de evento
caldav_calendar_id	integer	10	Atributo del calendario

3.2.3 Modelo Entidad-Relación

El modelo Entidad-Relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos, es una técnica de análisis para definir las entidades y las relaciones entre ellas.

En la figura 8 se muestra el esquema diseñado para la base de datos del sistema “MercaEventos” que sirve como fuente de información para esta aplicación.

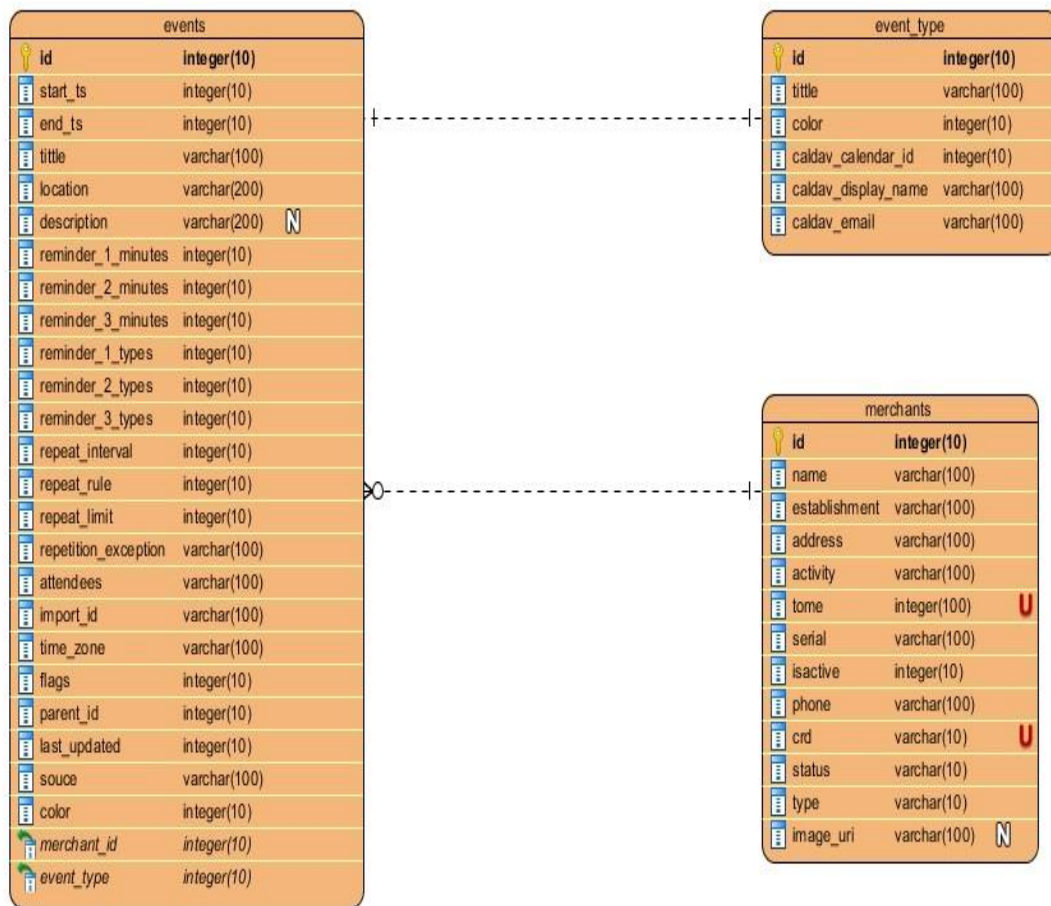


Figura 8. Modelo Entidad Relación

3.3 Descripción de la Interfaz de Usuario

A continuación se describe el storyboard o mapa de navegación dentro de la aplicación, es decir los vínculos entre las distintas pantallas que conforman la aplicación.

CAPÍTULO III. Diseño e Implementación

Al abrir la aplicación primeramente se visualizará la vista Mercados (cuando se utilice por primera vez estará vacía), allí se observará el listado de todos los mercados; además hay un botón de búsqueda y otro para añadir nuevos mercados a la base de datos.

La pantalla principal también cuenta con un menú lateral, que permite cuatro funcionalidades: Eventos, Tipo de eventos, Ajustes y Acerca de.

En Eventos se encuentra toda la información que el usuario programa para las tareas definidas en su interacción con los mercados.

En Tipo de Evento está el listado de los tipos de eventos iniciales, con su respectivo código de colores y la opción de edición.

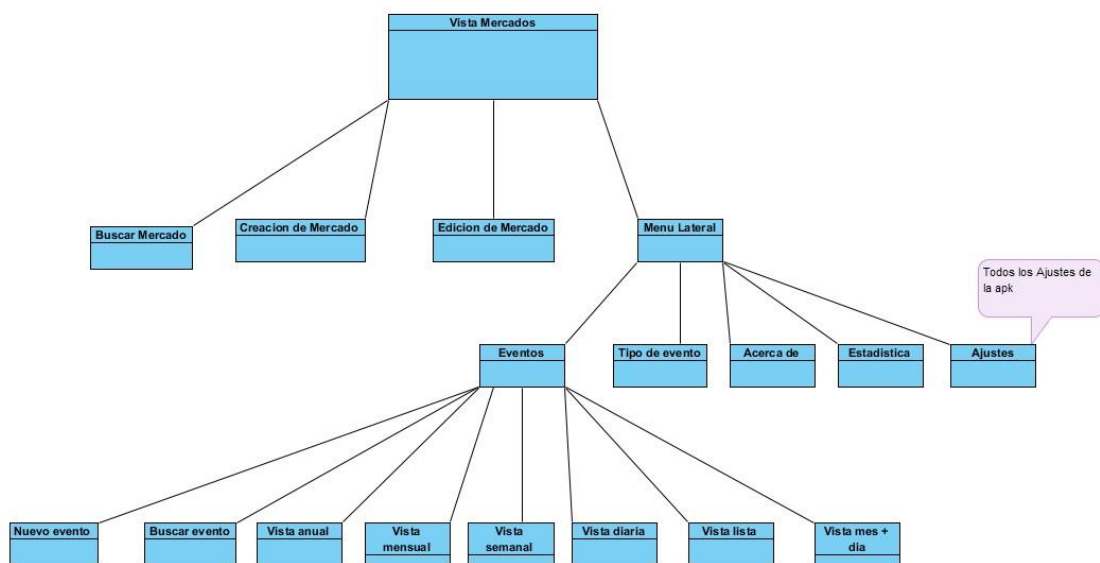


Figura 9. Mapa de Navegación

3.3.1 StoryCard

El propósito de una storycard es escribir cómo un proyecto devolverá valor al usuario. Entonces es el trabajo del equipo de desarrollo ocuparse de cómo desarrollar el código que satisfaga los requisitos de la historia del usuario. En el mejor de los casos, los desarrolladores colaboran estrechamente con los propietarios de negocios y las partes interesadas para aclarar los detalles a medida que se desarrolla el código.

Tabla 8. Descripción del storycard

Número	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
		Fácil	Fácil			Baja	
		Moderado	Moderado			Media	
		Duro	Duro			Alta	

CAPÍTULO III. Diseño e Implementación

Descripción		
Fecha	Estado	Comentario
	Definido	
	Implementado	
	Hecho	
	Verificado	

- En el campo ID se asigna un número identificador a la historia de usuario.
- En el campo TIPO se define qué tipo de actividad se va a describir en esta historia de usuario. Puede ser la implementación de un nuevo requisito, el arreglo o la mejora de un requisito ya implementado. Este valor varía dependiendo de la fase en la que se encuentre el proyecto. En la fase inicial el tipo siempre se define como Nuevo.
- En el campo DIFICULTAD se describe el nivel de dificultad que representará para el equipo de desarrollo el implementar ese requisito. Se tiene que ingresar un valor estimado antes y después de la implementación del requisito. Los valores referenciales recomendados son Fácil, Moderado, Difícil.
- En el campo ESFUERZO se detalla el número de horas que le tomará al equipo de desarrollo implementar este requisito. Al igual que con el campo anterior se debe ingresar un valor estimado antes de la implementación y el valor real que tomó la implementación.
- El campo PRIORIDAD debe incluir un valor que denote lo sustancial de este requisito para el proyecto. Este campo acepta valores referenciales sugeridos de Baja, Normal y Alta.
- El campo DESCRIPCIÓN lleva el nombre de la historia de usuario que se está cumpliendo y en el campo inferior se describe el desempeño que se va a llevar a cabo en esta historia de usuario, así como los posibles escenarios de éxito y falla que puedan generarse. Esta descripción, según se señala en los buenos procedimientos de las metodologías ágiles, debe ser lo más común posible, sin profundizar en demasiados detalles técnicos.

3.3.2 Descripción de las Vistas.

- **Pantalla Principal. Vista Mercados**

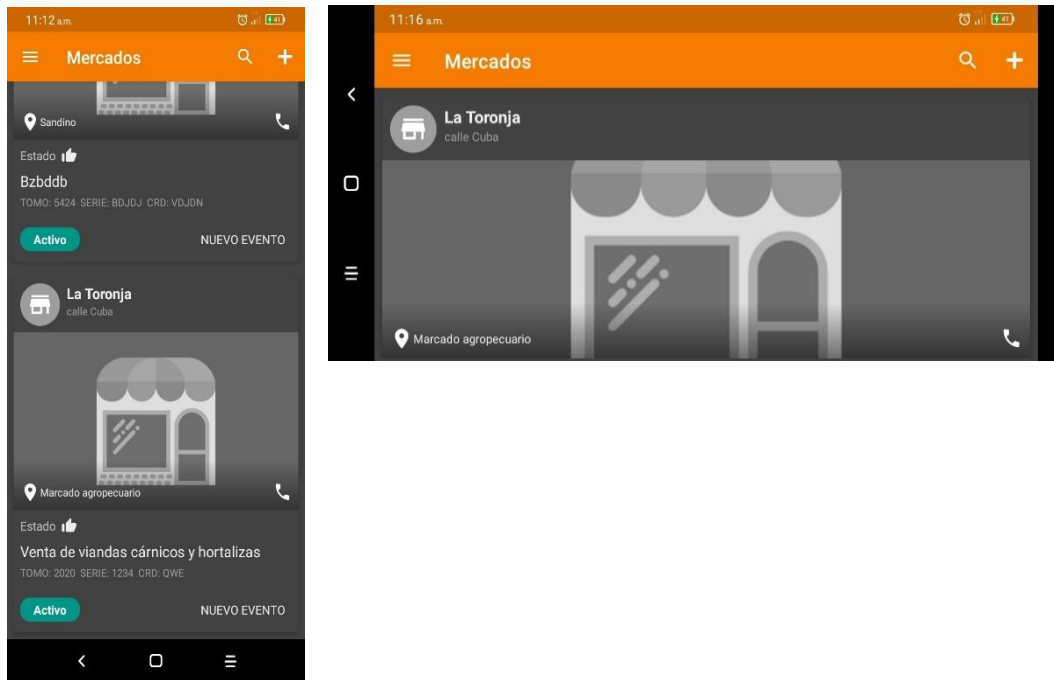


Figura 10. Pantalla Principal

La pantalla principal está formada por una barra de acción superior, un menú de navegación lateral y un listado de los mercados a los cuales se tiene acceso para su seguimiento.

El storycard descrito en la tabla 9 muestra una descripción del funcionamiento de la pantalla principal y las fechas de su desarrollo

Tabla 9. Vista Mercados

Número	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
01	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro	12	15	Baja Media Alta	
Descripción							
Cuando el usuario abre la aplicación se visualiza la pantalla de Mercados, la cual tiene una lista con todos los mercados añadidos, un botón de búsqueda y otro para agregar un nuevo mercado. Cuenta con un menú en la barra superior donde se puede acceder al menú lateral.							
Fecha	Estado	Comentario					
10/04/21	Definido	Sin comentario					
15/05/21	Implementado	Sin comentario					
20/07/21	Hecho	Sin comentario					
10/08/21	Verificado	Sin comentario					

- **Menú de Navegación Lateral**

CAPÍTULO III. Diseño e Implementación

El menú de navegación lateral muestra una lista con las opciones: Eventos, Tipo de eventos, Ajustes y Acerca de.

En la figura 11 se observa la pantalla en las dos orientaciones posibles del móvil:

- **Eventos:** se encuentra toda la información que el usuario programa para las tareas definidas en su interacción con los mercados.
- **Tipo de evento:** listado de los tipos de eventos iniciales, con su respectivo código de colores y la opción de su edición.
- **Ajustes:** se encuentran todos los ajustes de la aplicación.
- **Acerca de:** información de la aplicación.

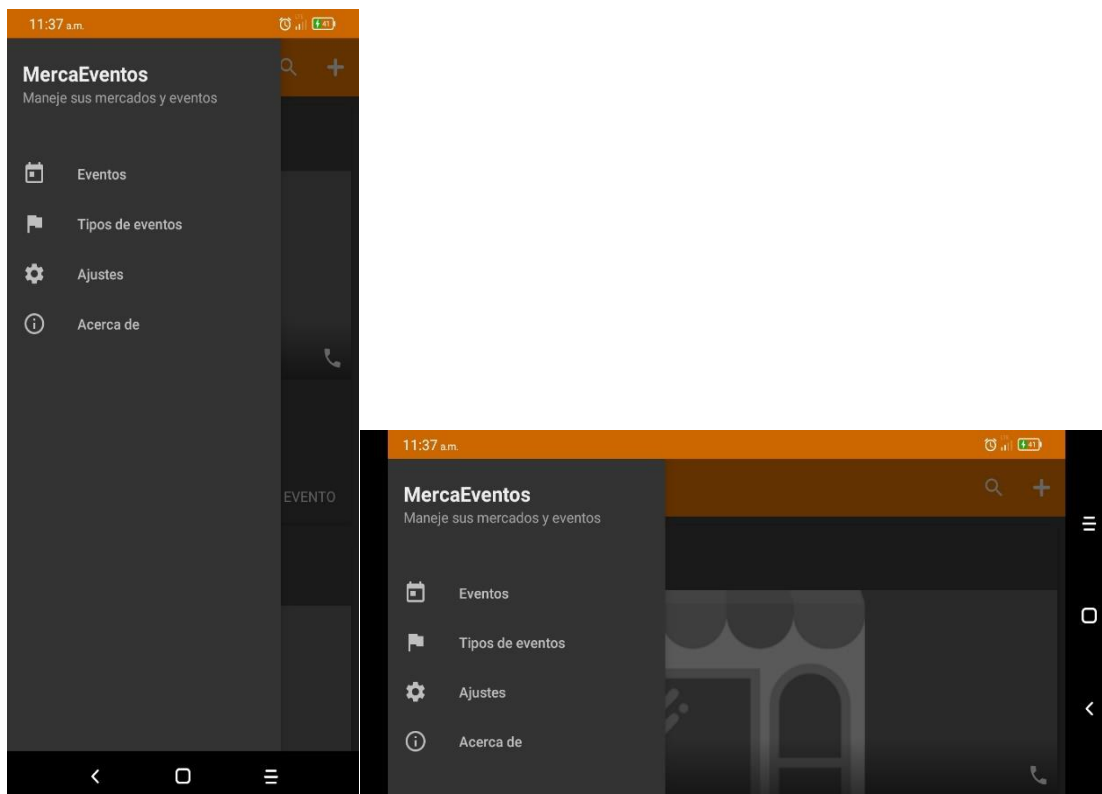


Figura 11. Menú Plegable

En la tabla 10, se describe la pantalla Menú de Navegación Lateral

Tabla 10. Menú Lateral

Número	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
02	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro	6	5	Baja Media Alta	
Descripción							
El usuario presiona el botón de navegación, y aparece un menú lateral con las opciones de Eventos, Tipo de eventos, estadística, Ajustes y Acerca de.							

Fecha	Estado	Comentario
12/04/21	Definido	Sin comentario
16/05/21	Implementado	Sin comentario
21/07/21	Hecho	Sin comentario
10/08/21	Verificado	Sin comentario

- **Vista Tipos de Eventos**

Se muestran en primer lugar los eventos predeterminados que trae la aplicación, que fueron los más prioritarios y necesarios durante la etapa investigativa. Cuenta con un botón para añadir nuevos tipos.

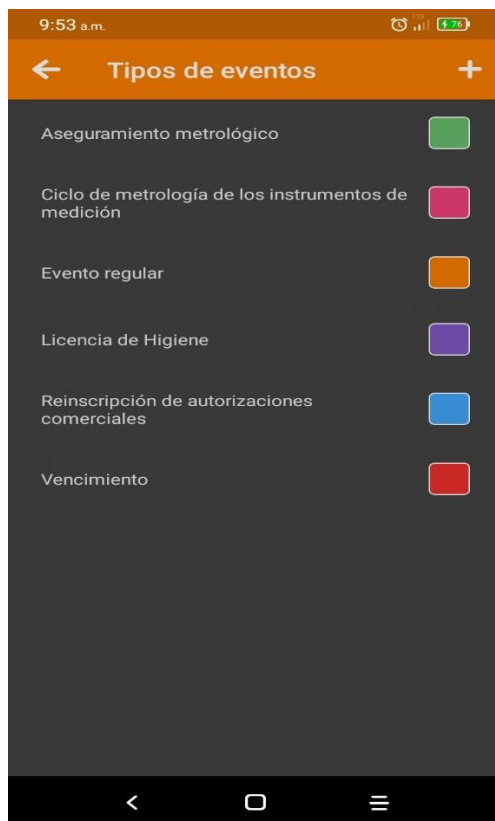


Figura 12. Tipos de Eventos

En la tabla 11, se describe la pantalla Tipos de Eventos. Se muestra una descripción del funcionamiento de la vista Tipo de Evento.

Tabla 11. Tipos de Eventos

Número	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
03	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro	10	12	Baja Media Alta	
Descripción							

CAPÍTULO III. Diseño e Implementación

Al el usuario seleccionar tipos de eventos, se mostrara un listado con los tipos de eventos existentes para asignar a los eventos, donde también podrá crear nuevos tipos y editar los existentes.

Fecha	Estado	Comentario
12/04/21	Definido	Sin comentario
17/05/21	Implementado	Sin comentario
21/07/21	Hecho	Sin comentario
10/08/21	Verificado	Sin comentario

- **Ajustes de la aplicación**

Esta es la vista de Ajustes donde el usuario puede modificar la amplia gama de opciones de la aplicación. Se permite la personalización de las notificaciones, del formato de la hora, de los colores, el tiempo de repetición, el tamaño de la fuente, etc.

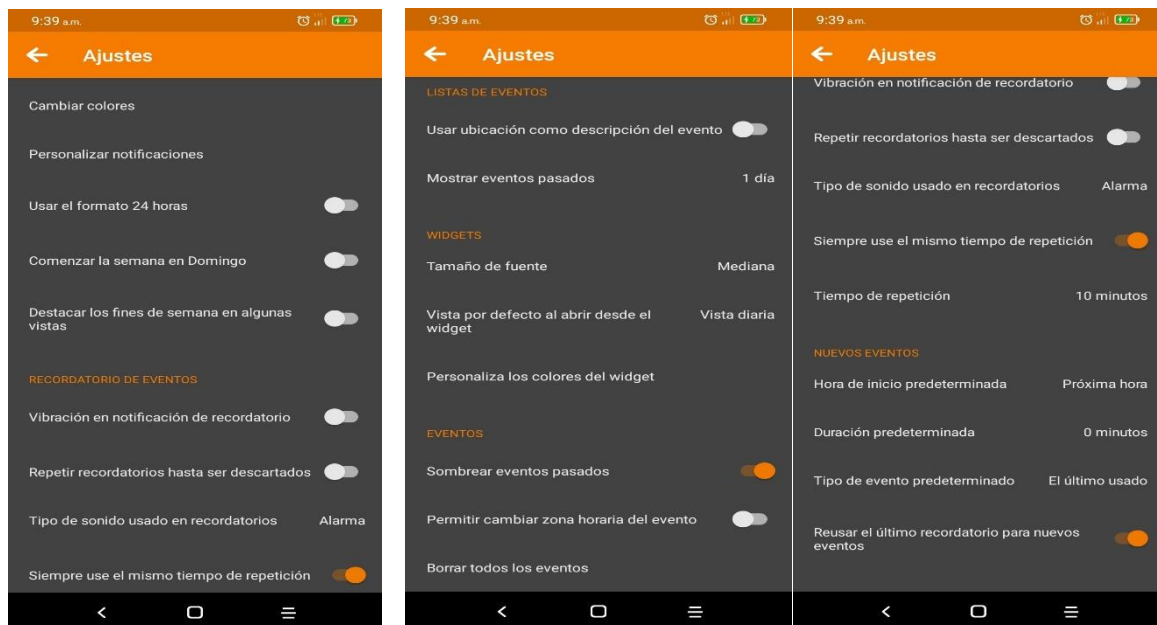


Figura 13. Ajustes

En la tabla 12 se describe la pantalla Ajustes. Se muestra una descripción del funcionamiento da la vista Ajustes.

Tabla 12. Ajustes

Número	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
04	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro	15	25	Baja Media Alta	
Descripción							

CAPÍTULO III. Diseño e Implementación

Ajustes donde el usuario puede modificar la amplia gama de opciones de la aplicación. Se permite la personalización de las notificaciones, el formato de hora, los colores, el tiempo de repetición, tamaño de la fuente, etc.

Fecha	Estado	Comentario
12/04/21	Definido	Sin comentario
18/05/21	Implementado	Sin comentario
21/07/21	Hecho	Sin comentario
10/08/21	Verificado	Sin comentario

- **Eventos – Vista Diaria**

En esta pantalla se visualizan los eventos de forma diaria, o sea los que están pendientes en el día que corre, permite además desplazarse por los demás días para verificar y planificar nuevos eventos.

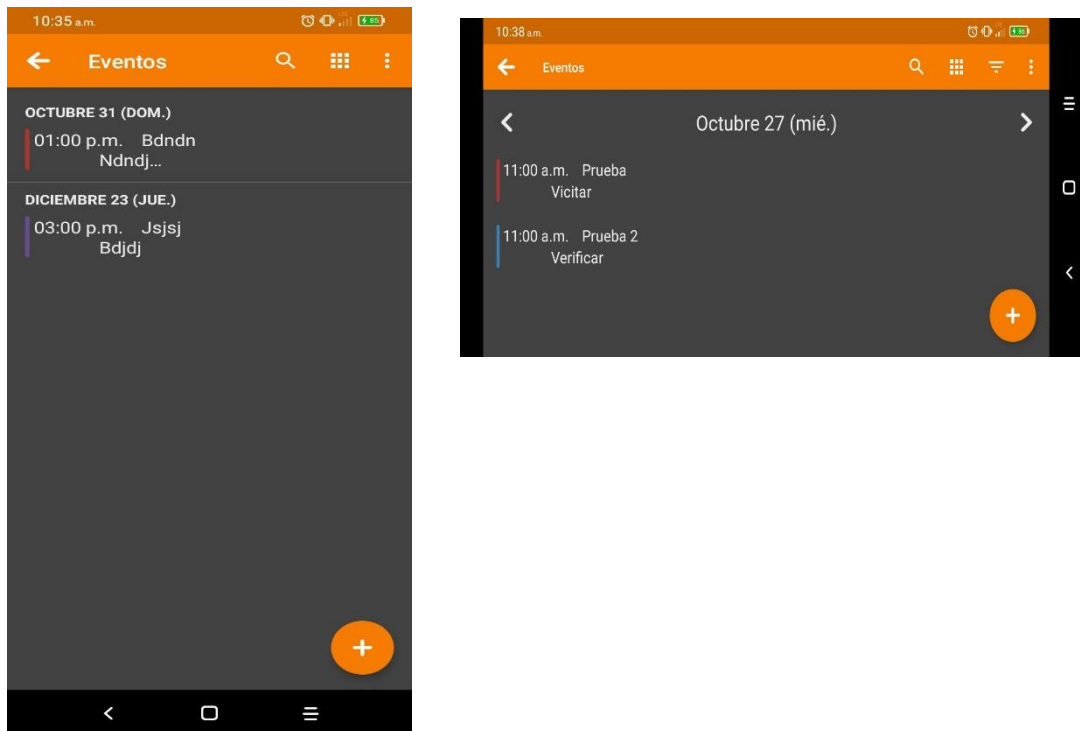


Figura 14. Vista Diaria

En la tabla 13 se describe la pantalla Vista Diaria, se muestra una descripción del funcionamiento de la Vista Diaria.

Tabla 13. Vista Diaria

Número	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
05	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro	10	8	Baja Media Alta	
Descripción							

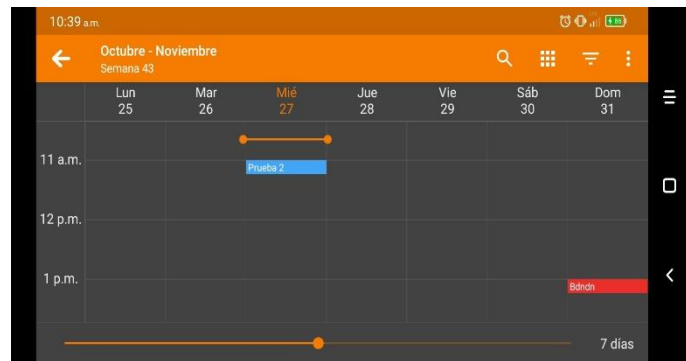
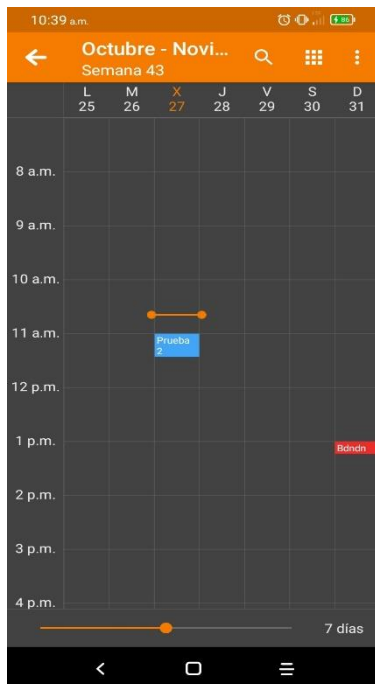
CAPÍTULO III. Diseño e Implementación

Se visualizan los eventos de forma diaria, los que están pendientes en el día que corre o desplazarse por los demás días para verificar y planificar nuevos eventos. Cuenta en la barra superior con la opción de buscar ,la opción de seleccionar vista , filtrar eventos , ir a la fecha y la opción de imprimir, estas tres últimas varían según la orientación del móvil

Fecha	Estado	Comentario
12/04/21	Definido	Sin comentario
15/05/21	Implementado	Sin comentario
21/07/21	Hecho	Sin comentario
11/08/21	Verificado	Sin comentario

• Eventos – Vista semanal

En esta pantalla se visualizan los eventos de forma semanal, los que están programados en la semana actual, permite desplazarse en un rango de 14 días.



En la tabla 14 se describe la pantalla: Vista Semanal, se muestra una descripción del funcionamiento de la Vista Semanal.

Tabla 14. Vista Semanal

Número	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
06	Nuevo	Fácil	Fácil	5	4	Baja	
	Fijo	Moderado	Moderado			Media	
	Mejora	Duro	Duro			Alta	

CAPÍTULO III. Diseño e Implementación

Descripción							
En esta pantalla se visualizan los eventos de forma semanal, los que están programados en la semana que corre o desplazarse en un rango de 14 días. Para agregar un nuevo evento solo es necesario pulsar el día que se quiera programar. Cuenta en la barra superior con la opción de buscar ,la opción de seleccionar vista , filtrar eventos , ir a la fecha y la opción de imprimir, estas tres últimas varían según la orientación del móvil							
Fecha	Estado	Comentario					
12/04/21	Definido	Sin comentario					
15/05/21	Implementado	Sin comentario					
21/07/21	Hecho	Sin comentario					
11/08/21	Verificado	Sin comentario					

• Eventos – Vista anual

En esta pantalla se visualizan los eventos de forma anual, los que están programados en el año.

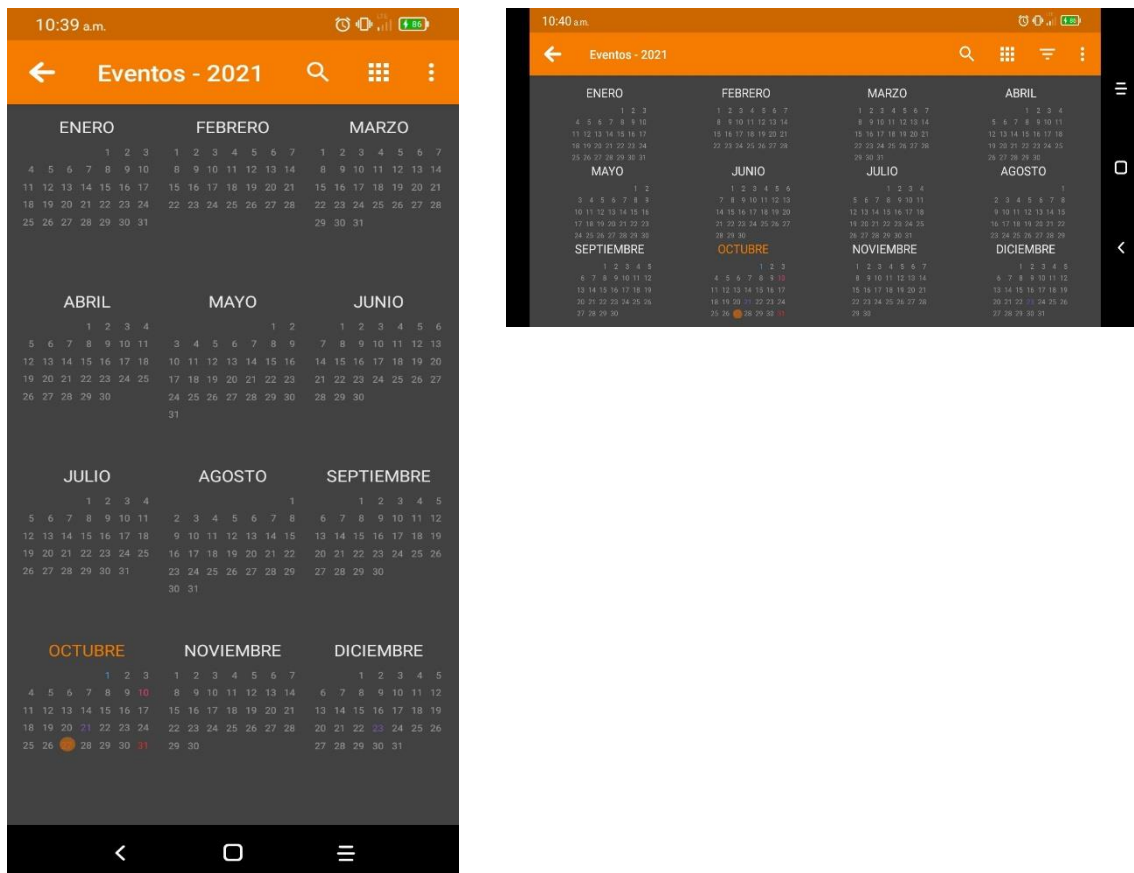


Figura 16. Vista Anual

CAPÍTULO III. Diseño e Implementación

En la tabla 15 se describe la pantalla Vista Anual, se muestra una descripción del funcionamiento de la Vista Anual.

Tabla 15. Vista Anual

Número	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
07	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro	5	3	Baja Media Alta	
Descripción							
En esta pantalla se visualizan los eventos de forma anual, los que están programados en el año. Para agregar un nuevo evento solo es necesario pulsar el día que se quiera programar. Cuenta en la barra superior con la opción de buscar ,la opción de seleccionar vista , filtrar eventos , ir a la fecha y la opción de imprimir, estas tres últimas varían según la orientación del móvil							
Fecha	Estado	Comentario					
12/04/21	Definido	Sin comentario					
15/05/21	Implementado	Sin comentario					
21/07/21	Hecho	Sin comentario					
11/08/21	Verificado	Sin comentario					

- **Eventos – Vista mes + día**

En esta pantalla se visualizan los eventos correspondientes a un mes, organizados por semanas que inician por lunes, se incluye un botón de crear evento.

CAPÍTULO III. Diseño e Implementación

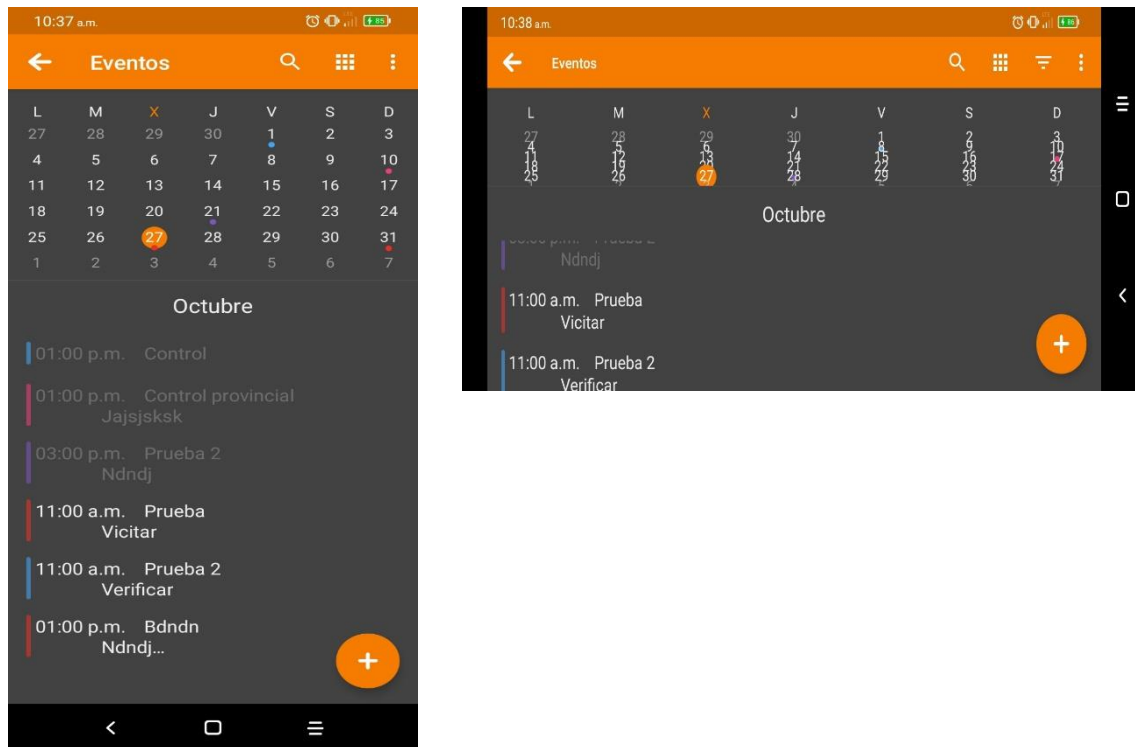


Figura 17. Vista Mes + Día

En la tabla 16 se describe la pantalla Vista Mes + Día, se muestra una descripción del funcionamiento de la Vista Mes + Día.

Tabla 16. Vista Mes +Día

Número	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
08	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro	6	8	Baja Media Alta	
Descripción							
Se visualizan los eventos correspondientes a un mes, organizados x semanas que inician por lunes y una lista de los eventos del mes además del botón de crear evento. Cuenta en la barra superior con la opción de buscar ,la opción de seleccionar vista , filtrar eventos, ir a la fecha y la opción de imprimir, estas tres últimas varían según la orientación del móvil							
Fecha	Estado	Comentario					
12/04/21	Definido	Sin comentario					
15/05/21	Implementado	Sin comentario					
21/07/21	Hecho	Sin comentario					
11/08/21	Verificado	Sin comentario					

- Acerca de

CAPÍTULO III. Diseño e Implementación

Esta pantalla muestra información general acerca del producto, se observa la pantalla en las dos orientaciones posibles del móvil.

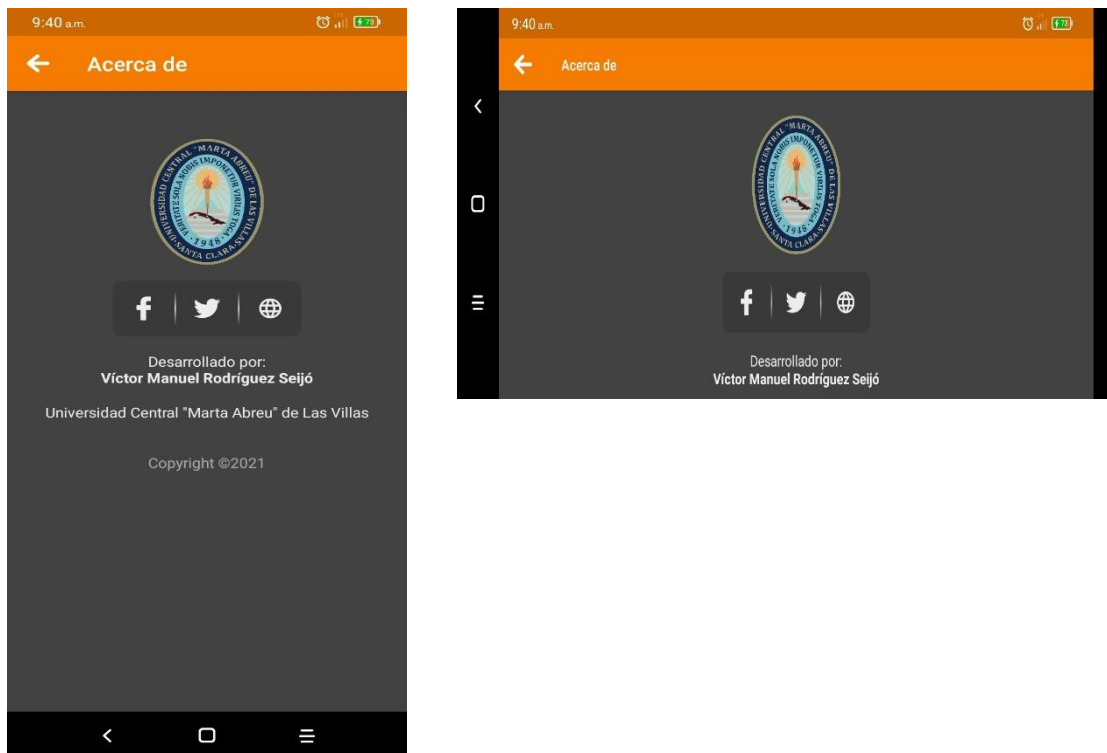


Figura 18. Vista Acerca de

En la tabla 17 se describe la pantalla Acerca de, se muestra una descripción del funcionamiento de la Vista Acerca de.

Tabla 17. Vista Acerca de

Número	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
09	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro	3	2	Baja Media Alta	
Descripción							
Esta pantalla muestra información general de la aplicación, como la versión y los derechos reservados. El logo de la uclv, y el nombre del desarrollador							
Fecha	Estado	Comentario					
14/04/21	Definido	Sin comentario					
8/05/21	Implementado	Sin comentario					
10/07/21	Hecho	Sin comentario					
18/08/21	Verificado	Sin comentario					

- **Estadística**

Crea una estadística de los datos que contiene la aplicación mostrándola en gráficos de pastel, de fácil comprensión para el usuario.

En la figura 19 se muestra el porcentaje de cada tipo de mercado.

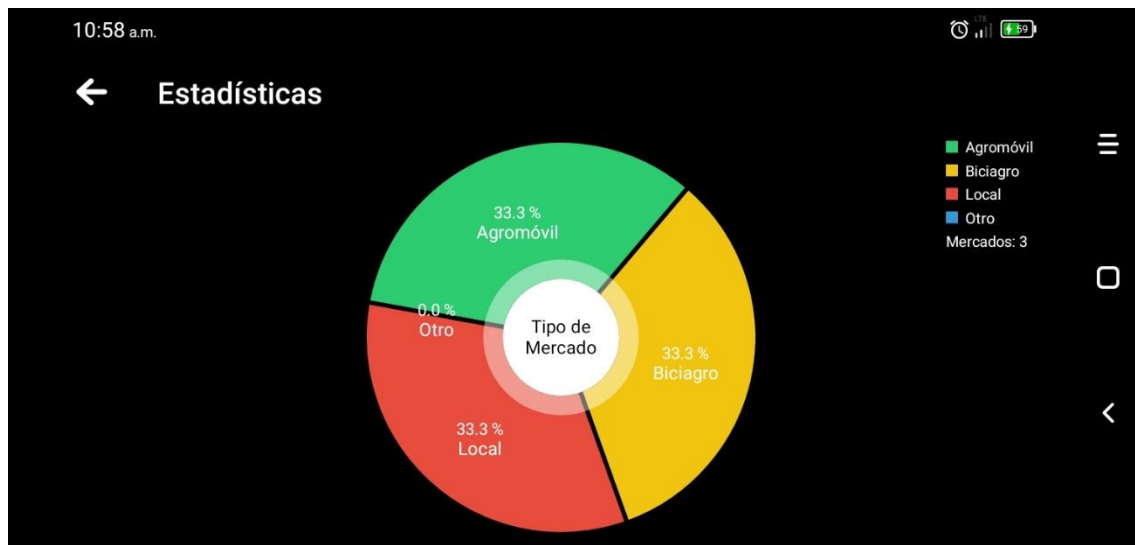


Figura 19. Tipo de Mercado

Tabla 18. Vista Estadísticas

Número	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
07	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro	5	3	Baja Media Alta	
Descripción							
En esta pantalla se visualiza el gráfico generado por los tipos de mercados con su porcentaje correspondiente							
Fecha	Estado	Comentario					
14/04/21	Definido	Sin comentario					
15/05/21	Implementado	Sin comentario					
21/07/21	Hecho	Sin comentario					
23/08/21	Verificado	Sin comentario					

3.4 Producción

Tanto en la fase de producción como en la de estabilización se trata de proceder a realizar la funcionalidad de la aplicación móvil junto con todos los requisitos recolectados en las fases anteriores, para asegurar que esta cumpla con la calidad deseada de una manera correcta.

La fase de producción incluye la implementación real, se divide en tres etapas: día de planificación, día de lanzamiento y días laborables y se itera consecutivamente entre ellas hasta obtener el resultado final.

Los días de planificación tienen como objetivo analizar, mejorar y priorizar los requisitos, planificar los contenidos de la iteración actual y preparar los casos de prueba de aceptación que se usará el día del lanzamiento.

CAPÍTULO III. Diseño e Implementación

En los días de lanzamiento se lanza una versión funcional para pruebas de aceptación del cliente, utilizando los casos de prueba desarrollados durante los días de planificación.

En los días laborables se implementan funcionalidades en el desarrollo guiado por pruebas de software (TDD).

Además del modelo informativo y de vista de usuario, durante esta fase el equipo de desarrollo utiliza el diagrama de casos de uso, el de componentes, el diagrama de clases de la actividad y el patrón MVC para guiar la implementación y alinear la comprensión de los miembros del equipo. (Alsabi & Dahanayake, 2016).

Una vez culminada esta fase se recolecta la siguiente documentación:

- Funcionalidades desarrolladas.
- Anotaciones del desarrollo.
- Esquemas de la interfaz de usuario de la aplicación.
- Storycards.
- Requisitos modificados. (Darwin, Duldulao, Pagtaconan, & Galang, 2015).

En esta fase además se plantean los estándares de codificación y la estructuración de la aplicación.

3.4.1 Estándares de Codificación

Los estándares de codificación definidos son:

- Paquetes: el nombre de los paquetes está escrito en minúscula.
- Clases: el nombre de las clases está escrito con la primera letra en mayúscula, si se compone de dos o más palabras la primera letra de cada una de ellas está en mayúscula.
- Métodos: los métodos llevan un nombre relacionado al proceso que ejecutan, están escritos con la primera letra en minúscula, si se compone de dos o más palabras la primera letra de ellos está en mayúscula, a excepción de la primera palabra.
- Layouts: los layouts tienen todo su nombre en minúsculas, si poseen más de dos palabras tendrán un guión bajo para separarlas.

3.4.2 Codificación

La codificación se realizó basándose en los storycards realizados en la fase anterior.

A continuación se muestran las clases y métodos más importantes.

- **SnoozeService**

Esta es la clase que se ocupa de que las notificaciones de la aplicación sean tratadas por el sistema, tramita con el sistema del teléfono informando o actualizando cada evento a tener en cuenta para notificar.

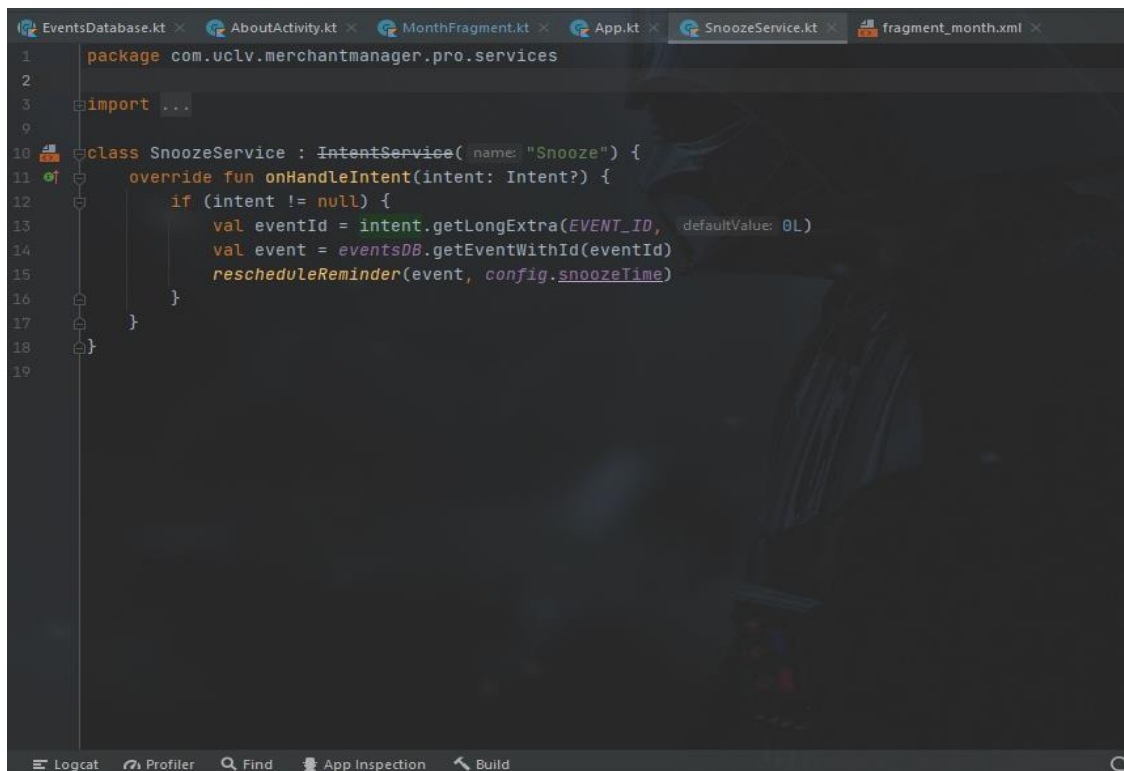


Figura 20. Notificaciones

- **MPAndroidChart**

Esta biblioteca es la que permite la visualización de los indicadores mediante gráficos, fue utilizada para la consulta estadística que se realiza a partir de los datos de mercados almacenados para su posterior análisis.

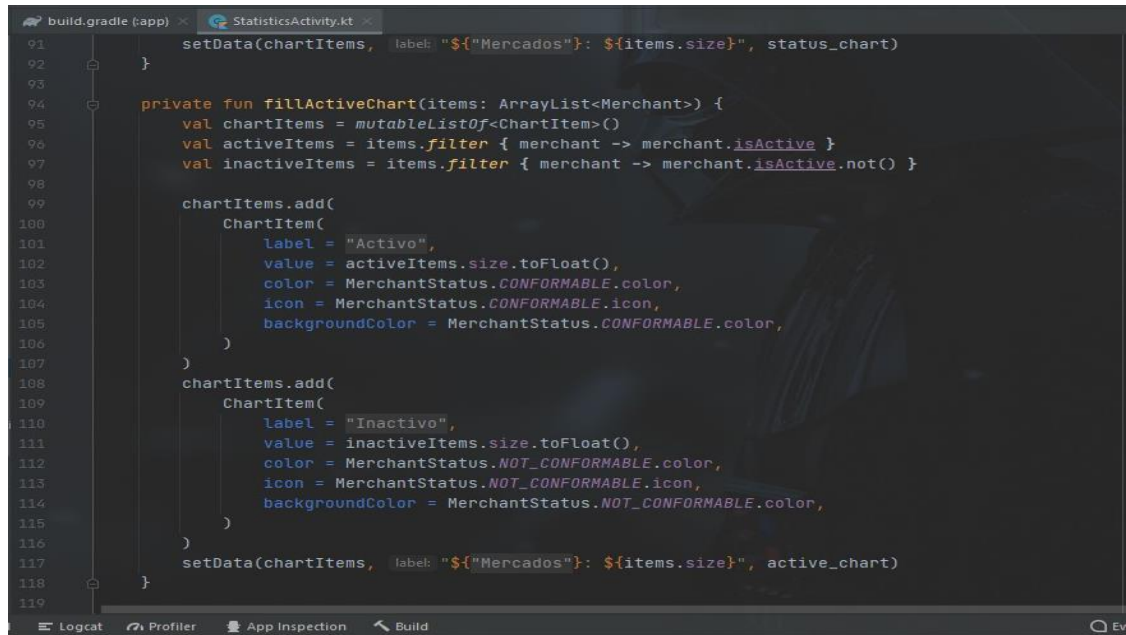


Figura 21. Función de Gráficos

- **Función compartir**

Para compartir los eventos creados, ya sea por días o meses, la aplicación utilizó la siguiente función para compartir la información en formato pdf.

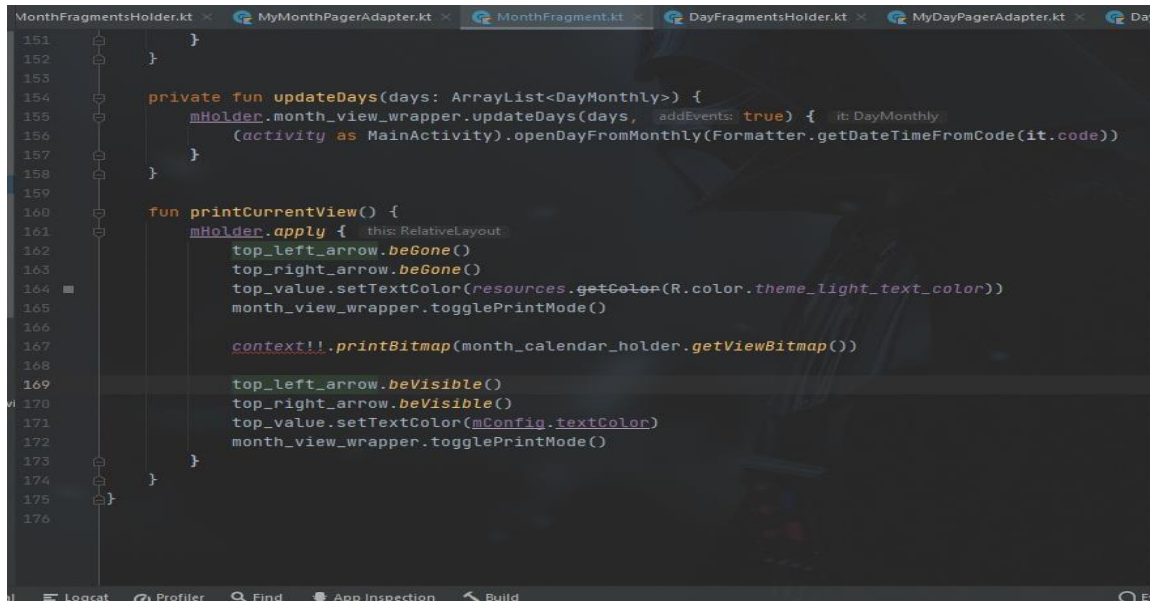


Figura 22. Exportar pdf

- **Dependencias de la aplicación**

```
dependencies {  
    implementation project(path: ':commons')  
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk8:$kotlin_version"  
    implementation 'androidx.core:core-ktx:1.6.0'  
    implementation 'androidx.appcompat:appcompat:1.3.1'  
    implementation 'joda-time:joda-time:2.10.12'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.1'  
    implementation "androidx.print:print:1.0.0"  
    implementation 'com.github.dhaval2404:imagepicker:2.1'  
    implementation 'com.squareup.picasso:picasso:2.71828'  
    implementation 'androidx.swiperefreshlayout:swiperefreshlayout:1.1.0'  
    implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0'  
  
    kapt 'androidx.room:room-compiler:2.3.0'  
    implementation 'androidx.room:room-runtime:2.3.0'  
    annotationProcessor 'androidx.room:room-compiler:2.3.0'  
}
```

Figura 23. Dependencias

Conclusiones del Capítulo

En este capítulo se diseñaron los principales artefactos de la metodología Mobile-D, con este objetivo:

- Se configuró el entorno de desarrollo y se elaboró el diagrama de componentes.
- Se implementaron las clases del sistema.
- Se realizó el diseño conceptual de la base de datos en base al modelo entidad relación.
- Se confeccionó el storyboard o mapa de navegación, que permite conocer cómo acceder a las diferentes pantallas.
- Se definieron todas las storycards que permiten mostrar visualmente como se observará cada pantalla de la aplicación.
- Se definieron los estándares de codificación.

CONCLUSIONES

- Se utilizó la metodología Mobile-D que permitió una reducción del tiempo durante la etapa de desarrollo.
- La documentación recolectada en las distintas fases de la metodología usada, permitió garantizar la calidad y usabilidad del producto.
- Se implementó una aplicación que permite la gestión de los eventos de la empresa Provincial de Acopio Villa Clara.
- Se concibió una interfaz intuitiva y sencilla para proporcionar una mejor interacción y fácil manejo de los usuarios con el sistema.

RECOMENDACIONES

Considerando la investigación realizada en el presente trabajo se recomienda:

- Mantener un adecuado control de la funcionabilidad de los mercados.
- Tener establecidos los tipos de eventos con los que trata cada trabajador.
- Formar en los trabajadores la necesidad del uso de las tecnologías para un mejor funcionamiento de los procesos.
- Continuar dándole soporte a la aplicación para la implementación de futuras versiones, con el objetivo de adaptarse al desarrollo de los dispositivos tecnológicos.
- Considerar la integración con otras aplicaciones con el interés de llevar todos los procesos comerciales que realiza la empresa hacia una plataforma común.

BIBLIOGRAFÍA

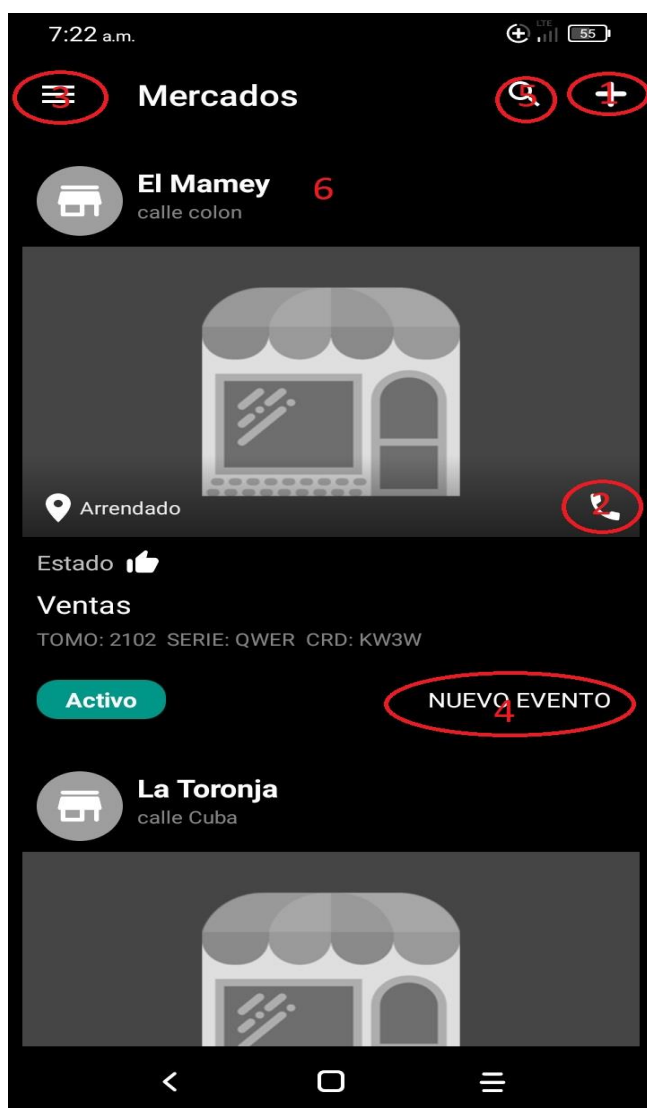
- Balaguera, Y. D. A. (2013). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual. *Revista de Tecnología*, 12(2), 111-123.
- Cadavid, A. N., Martínez, J. D. F., & Vélez, J. M. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva*, 11(2), 30-39.
- Correa Ríos, M. (2014). Composición del sistema operativo móvil iOS de Apple y el hardware y software que lo utilizan.
- García Rafael, C. M. (2018). *Tablero de control de indicadores de desempeño para plataforma Android*. Universidad Central "Marta Abreu" de Las Villas.
- Gómez Matesanz, A. (2014). *Aplicacion android para la empresa Travelling-Service*.
- Gómez, E., Ruiz, J., & López, F. (2016). El conflicto en el desarrollo ágil: una perspectiva desde el SCRUM. *Revista Gestion y Region*, 121-150.
- González Márquez, D. (2019). Creación de una práctica de bases de datos relacionales con SQLite.
- Hernández, A. d. I. C. (2018). *Sistema Web para el control de la fuerza de trabajo de la Empresa Tabaco Torcido Villa Clara*. Universidad Central "Marta Abreu" de Las Villas.
- Mesonero Rodríguez, D. (2021). *Avanza Mobility*.
- Muñoz Muñoz, C. A. (2020). *Aplicación de la metodología Mobile-D en el desarrollo de una app móvil para gestionar citas médicas del centro Jel Riobamba*. Riobamba: Universidad Nacional de Chimborazo.
- Narváez Argeñal, M. d. I. C., Pérez Rodríguez, Y. Y., & Vallecillo Dávila, L. Á. (2019).
- Desarrollo de un software para la gestión de préstamos (mayo-septiembre 2019).
- Vogel, S. M. (2021). *Aplicación móvil para la asistencia*.

ANEXOS

Manual de usuario

- **Vista mercado**

Cuando el usuario abre la aplicación se visualiza la pantalla de Mercados, la cual tiene una lista con todos los mercados añadidos, un botón de búsqueda y otro para agregar un nuevo mercado. Cuenta con un menú en la barra superior donde se puede acceder al menú lateral.



Vista donde se listan los mercados, cuenta con los siguientes componentes:

1. Crear nuevo mercados, opción para añadir un nuevo mercado
2. Llamar, opción para llamar o enviar mensaje
3. Menú lateral , despliega el menú lateral
4. Crea un evento vinculado con el mercado seleccionado

5. Busca , busca en la lista por cualquiera de los atributos del mercado
6. Mercado , entra al mercado para editar o consultar sus atributos

- **Añadir mercado**

Cuando el usuario necesita añadir un nuevo mercado o editar la información de uno ya existente tiene acceso a esta vista

7:23 a.m. 55%

Añadir mercado

1

Nombre 3

Establecimiento 4

Dirección 5

Actividades 6

Teléfono 7

CRD 8

Tomo 9

Serie 10

8:46 a.m. 55%

Añadir mercado

14

Local 11

Activo 12

Conforme 13

Nombre

Establecimiento

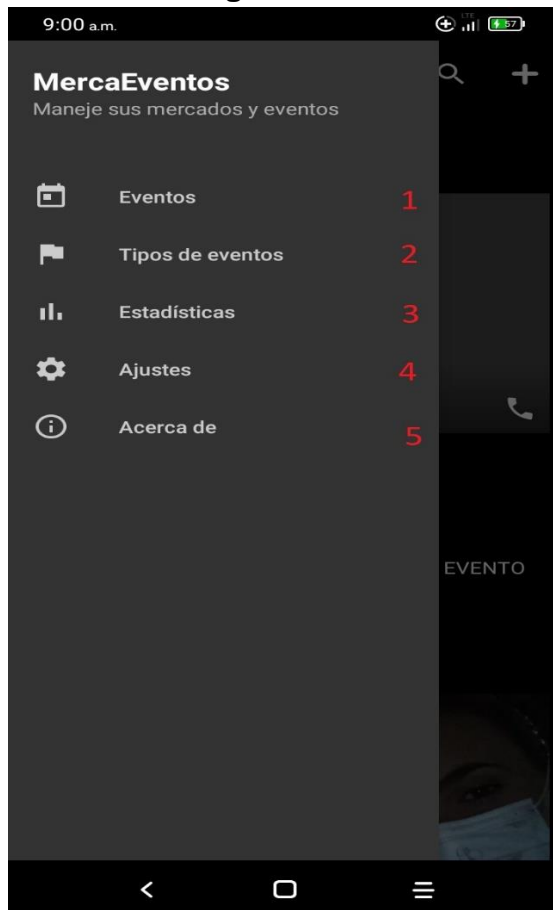
Dirección

Vista para añadir un nuevo mercado, cuenta con los siguientes componentes:

1. Confirmar
2. Atrás , sale de la vista y retrocede a la anterior
3. Se añade el nombre del mercado
4. Se añade el establecimiento del mercado
5. Se añade la dirección del mercado
6. Se añade las actividades del mercado
7. Se añade el teléfono del mercado
8. Se añade el CRD del mercado
9. Se añade el tomo del mercado

10. Se añade el serie del mercado
11. Tipo de mercado , Puede ser local, biciagro, agromóviles o añadir un nuevo tipo
12. Disponibilidad del mercado, puede estar activo o no
13. Criterio de conformidad del mercado , puede ser conforme o no conforme
14. Imagen del mercado añadir una foto o editarla

- **Menú de Navegación Lateral**



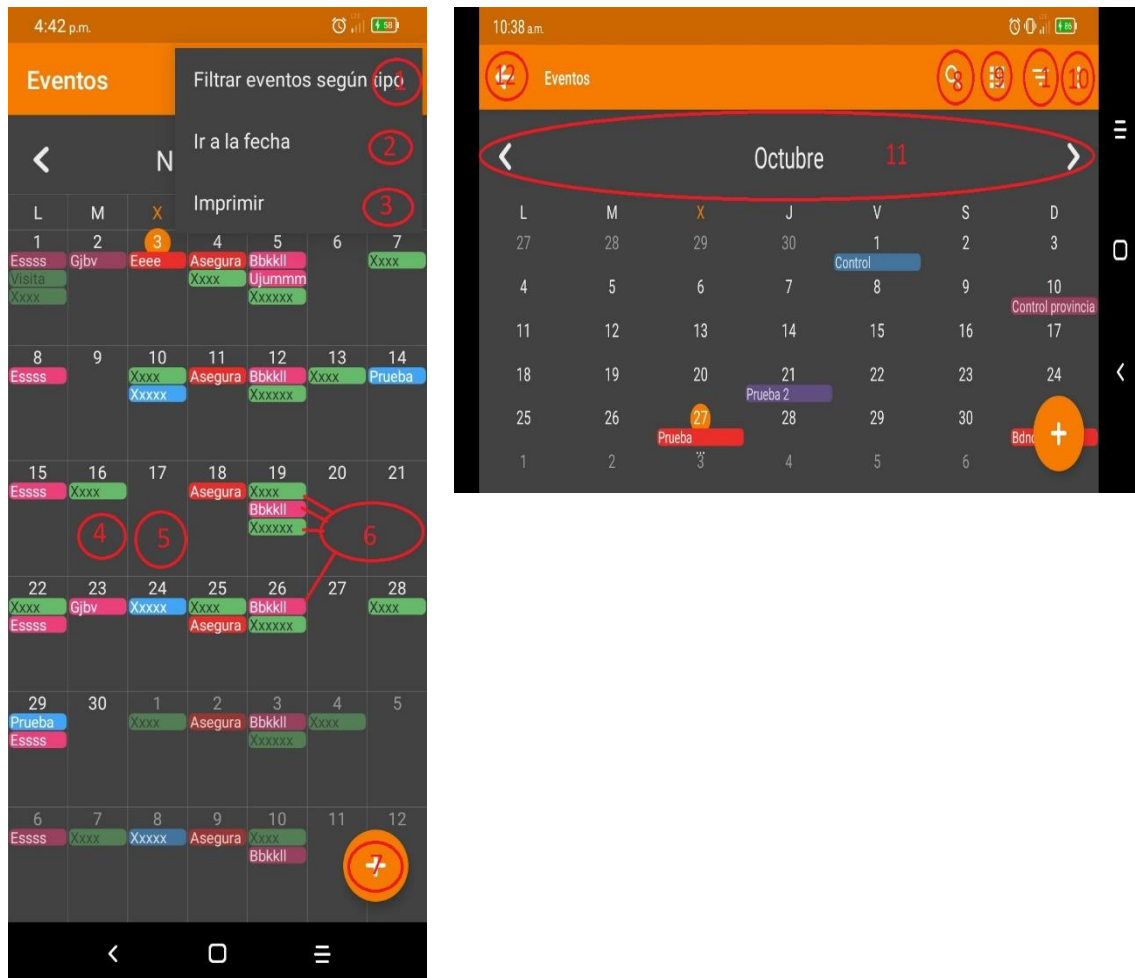
El usuario presiona el botón de navegación, y aparece un menú lateral con las opciones de Eventos, Tipo de eventos, estadística, Ajustes y Acerca de.

El menú de navegación lateral muestra una lista con las opciones: Eventos, Tipo de eventos, estadísticas, Ajustes y Acerca de.

1. Eventos se encuentra toda la información que el usuario programa para las tareas definidas en su interacción con los mercados.
2. Tipo de evento está el listado de los tipos de eventos iniciales, con su respectivo código de colores y la opción de su edición.
3. Estadísticas
4. Ajustes, se encuentran todos los ajustes de la aplicación.
5. Acerca de, información de la aplicación.

- **Vista eventos**

Se visualizan los eventos, los que están pendientes en el día que corre o desplazarse por los demás días para verificar y planificar nuevos eventos. Cuenta en la barra superior con la opción de buscar, la opción de seleccionar vista, filtrar eventos, ir a la fecha y la opción de imprimir.

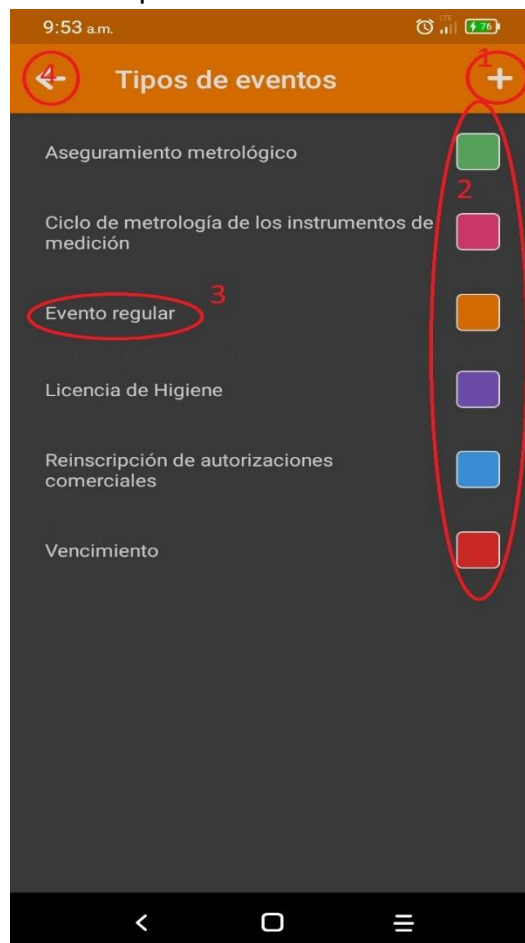


1. Filtrar eventos según tipo , permite la visualización o no de los eventos según el tipo que prefieras
2. Ir a la fecha , permite la búsqueda manual de determinado fecha con los eventos correspondientes a ella
3. Imprimir , imprime la información de la vista o permite la exportación de la misma para luego ser compartida
4. Día con un evento planificado
5. Día vacío
6. Eventos por código de color
7. Añadir nuevo evento
8. Buscar
9. Cambiar de vista

- 10. Más opciones
- 11. Desplazarse por las fechas
- 12. Atrás

- **Tipo de Evento**

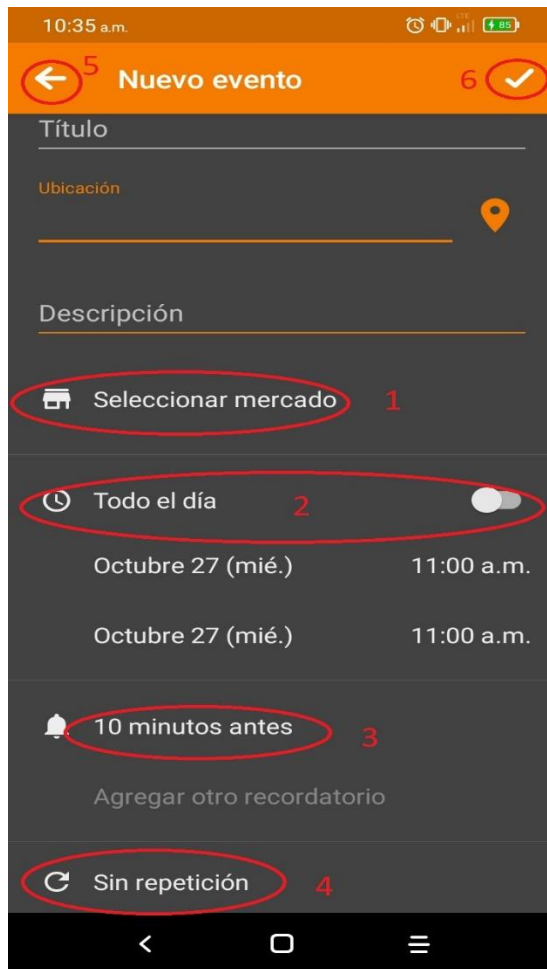
Los tipos de eventos se muestran los eventos primeramente que trae la aplicación predeterminada, los cuales mediante la parte investigativa fueron los más prioritarios y necesarios. Cuenta con un botón para añadir nuevos tipos.



- 1. Añadir tipo de evento , se crea con un nombre y un código de color
- 2. Código de color de los eventos creados
- 3. Nombre del evento creado
- 4. Atrás

- **Nuevo evento**

Vista para crear un nuevo evento

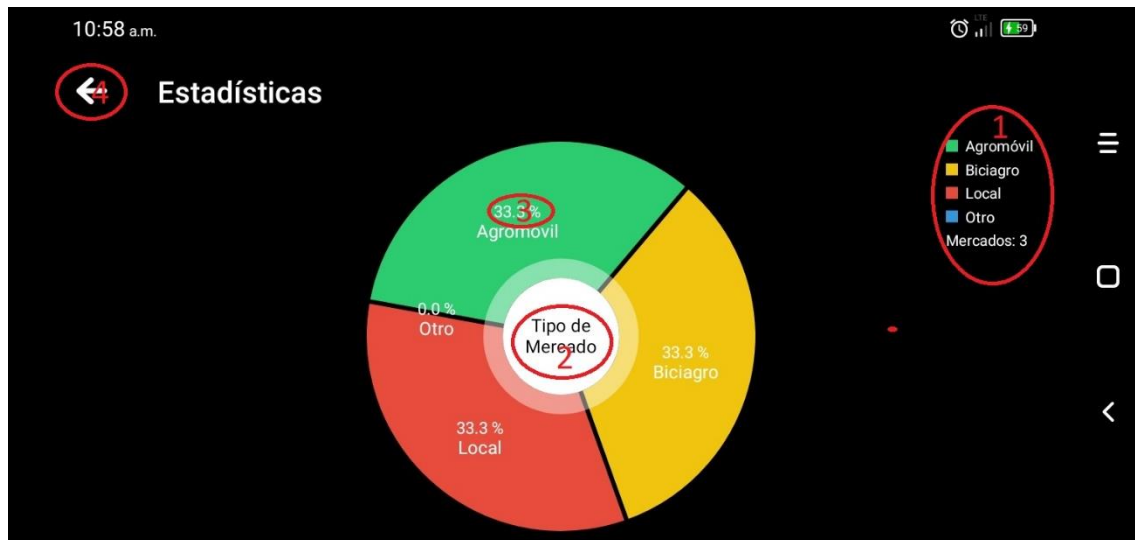


1. Seleccionar mercado , selecciona el mercado asociado con ese evento
2. Repetición de los notificaciones
3. Configuración del tiempo
4. Planificar este evento en fechas futuras si es un evento cíclico

- **Estadística**

Crea una estadística de los datos que contiene la aplicación mostrándola en gráficos de pastel, de fácil comprensión para el usuario.

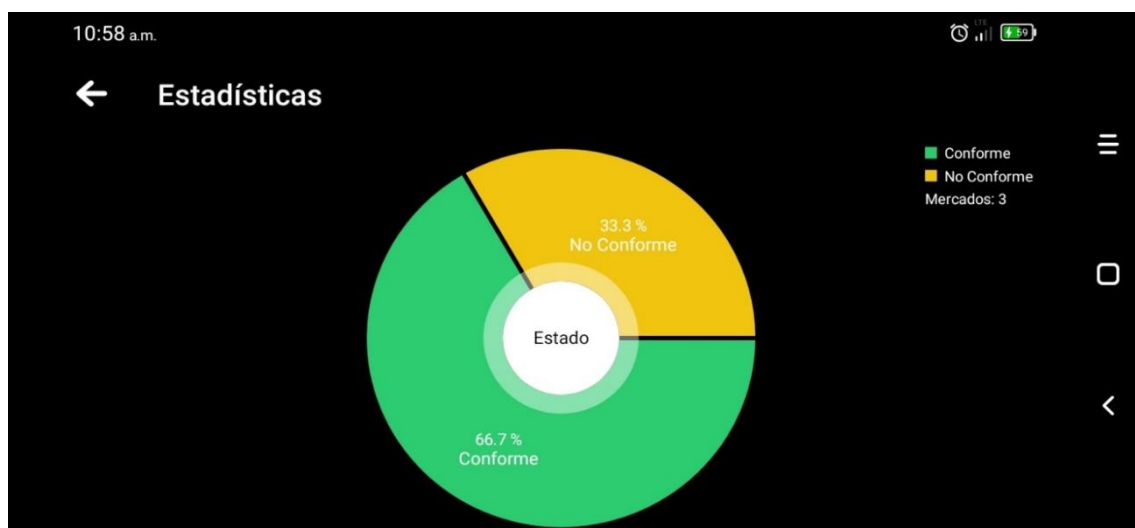
- Analiza el porcentaje de los mercados en cuanto a si se encuentran activos o inactivo generando un gráfico de pastel de útil y de fácil comprensión.



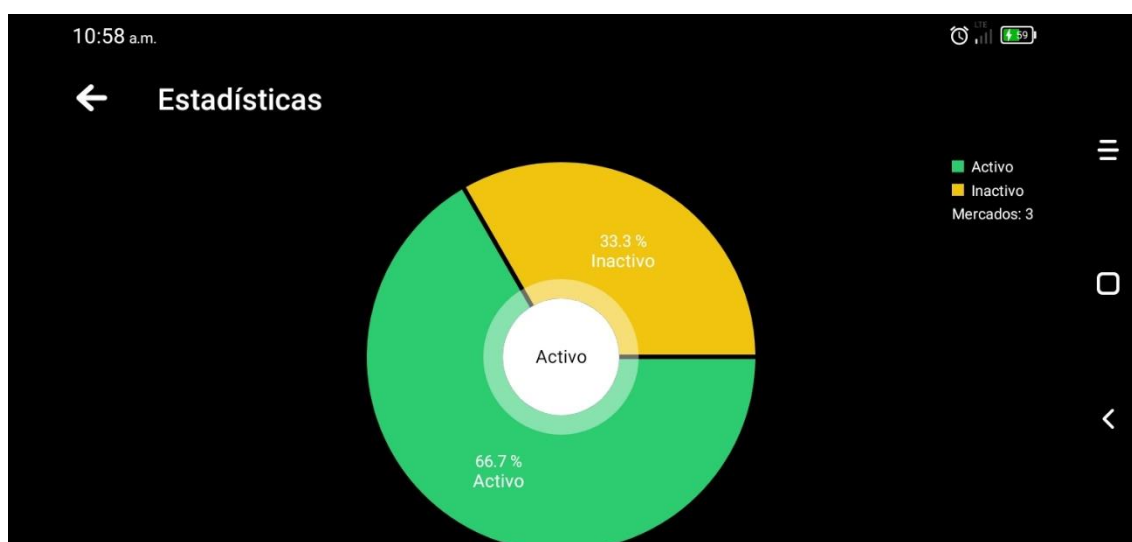
1. Leyenda del grafico
2. Nombre de la gráfica
3. Porcentaje q representa
4. Atrás

Otras vistas de la aplicación

- Mediante el parametro de conformidad de los mercados analiza el porcinto según el estado conforme y no conforme.



- Analiza el porcentaje de los mercados en cuanto a si se encuentran activos o inactivo generando un gráfico de pastel de útil y de fácil comprensión.



- Configuración de las notificaciones

10:37 a.m.

← Nuevo evento ✓

Descripción

☐ Sin recordatorio

☐ Al inicio

☐ 1 minuto antes

☐ 5 minutos antes

☐ 10 minutos antes

☐ 30 minutos antes

☒ 1 hora antes

☐ Personalizado

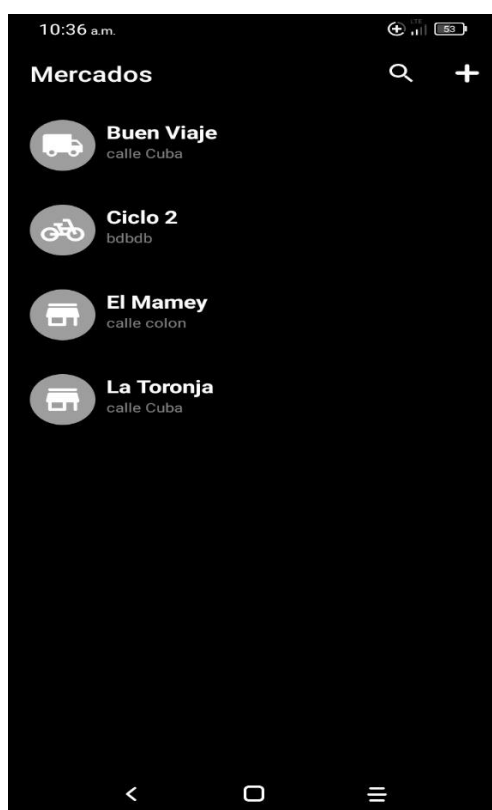
☒ Sin repetición

Ciclo de metrología de los instrumentos de medición

- Buscar por fecha



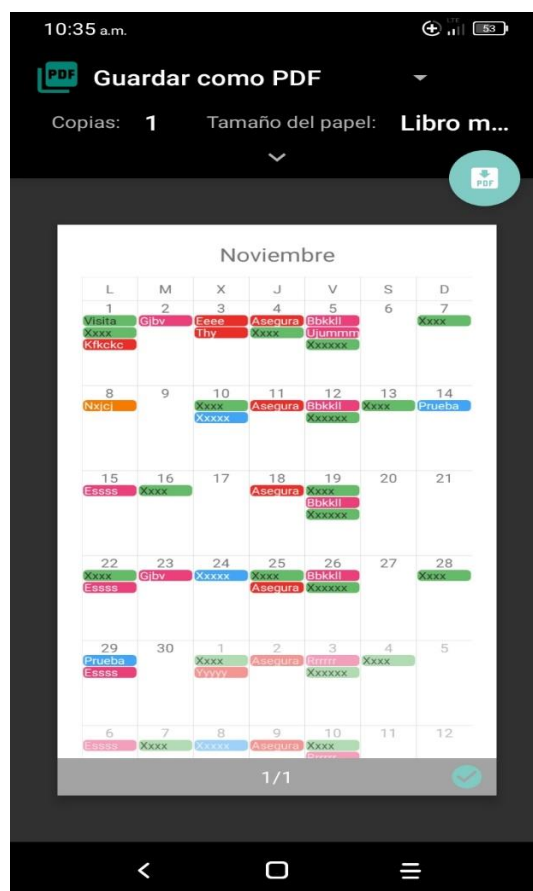
- **Seleccionar mercado**



- **Buscar de impresora**



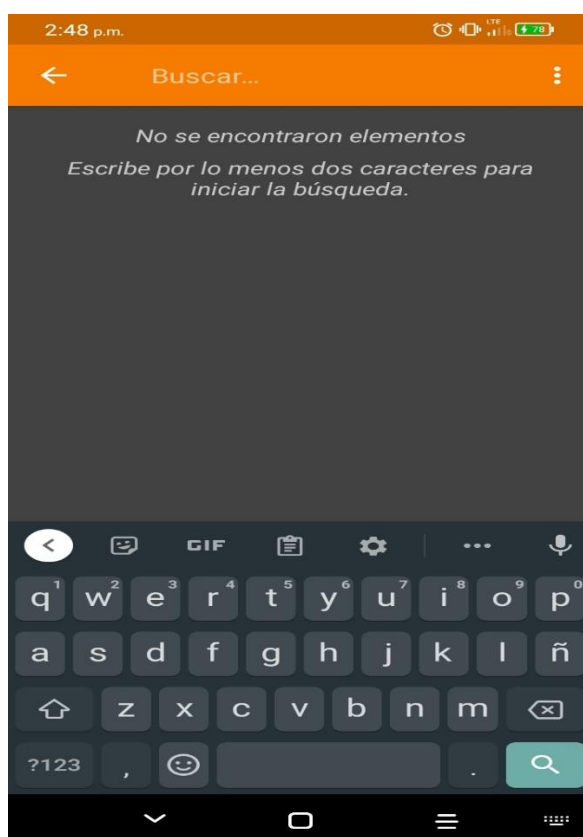
- Importar a pdf



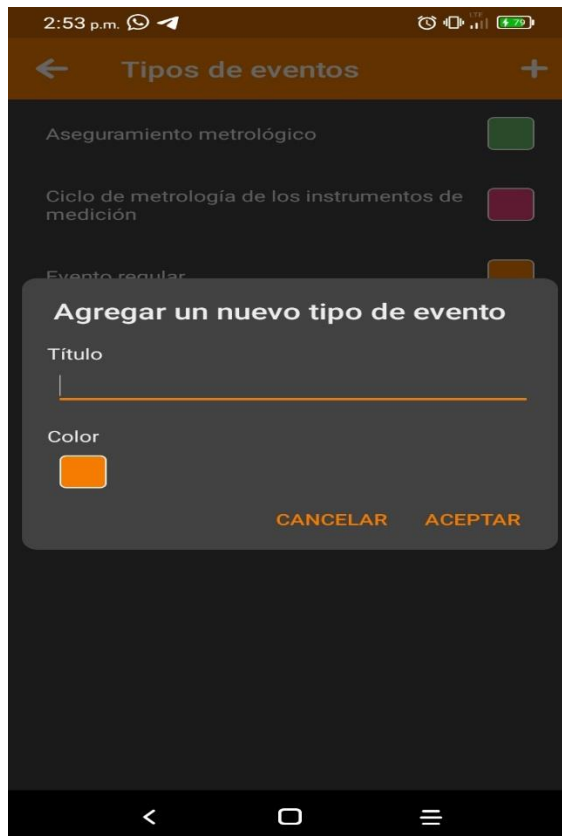
- Selección de tipo de mercado



- **Buscar**



- **Agregar evento**



- Paleta de Colores



