

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

Departamento de Computación

TRABAJO DE DIPLOMA

Título del trabajo: "Tablero de control de indicadores de desempeño para plataforma Android"

Autores del trabajo: Carlos Miguel García Rafael

Tutores del trabajo: Dra. Ana María García Pérez

Ing. Elieser Bello Ross

Lic. Osmani Rosado Falcón

Santa Clara, Junio 2018
Copyright©UCLV

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

“Nunca consideres el estudio como una obligación,
sino como una oportunidad para penetrar en el bello y maravilloso mundo del
saber.”

(Albert Einstein)

Dedicatoria

A todas las personas que me han apoyado en estos cinco años, en especial a mi familia, que sin su apoyo no hubiese llegado a obtener este resultado.

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

El presente trabajo de diploma se centra en el desarrollo de una aplicación denominada “Bimas Bion Android” para dispositivos móviles sobre plataforma Android, que posibilite el control y seguimiento de los indicadores de desempeño en las organizaciones. Esta aplicación obtiene la información necesaria desde un servicio brindado por Bimas, sistema de seguimiento a estrategias empresariales desarrollado por la empresa Desoft de Villa Clara.

El propósito de “Bimas Bion Android” es dotar a los directivos de esta valiosa información facilitando su movilidad. Para esto se abordan conceptos relacionados con el desarrollo de las aplicaciones móviles, la conexión mediante el protocolo de comunicación HTTP y bases de datos para dispositivos móviles, metodologías de desarrollo para aplicaciones móviles, así como las técnicas de visualización de indicadores de desempeño.

Abstract

The present diploma work focuses on the development of an application called "Bimas Bion Android" for mobile devices on Android platform, which enables the control and monitoring of performance indicators in organizations. This application obtains the necessary information from a service provided by Bimas, a tracking system for business strategies developed by the company Desoft of Villa Clara.

The purpose of "Bimas Bion Android" is to provide managers with this valuable information facilitating their mobility. This is addressed concepts related to the development of mobile applications, the connection through the HTTP communication protocol and databases for mobile devices, development methodologies for mobile applications, as well as the techniques of visualization of performance indicators.

Índice

Introducción.....	1
CAPÍTULO 1. CONSIDERACIONES TEÓRICAS SOBRE EL DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES Y SOBRE LA HERRAMIENTA BIMAS 4.0.	4
1.1 Metodologías y métodos para el desarrollo de aplicaciones para dispositivos móviles.	4
1.2 Indicadores de desempeño en las Organizaciones.....	17
1.3 Conclusiones del capítulo:	23
CAPÍTULO 2: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN.	24
2.1 Primera Fase: Análisis	24
2.1.1 Exploración	24
2.2 Segunda Fase: Diseño	26
2.2.1 INICIALIZACIÓN	26
2.3 Tercera Fase: Codificación	48
2.3.1 PRODUCCIÓN Y ESTABILIZACIÓN	48
Conclusiones del Capítulo:	52
CAPÍTULO 3. CUARTA FASE: PRUEBAS.	53
3.1 PRUEBAS DEL SISTEMA Y ARREGLOS.....	53
3.1.1 Pruebas de Interfaz de Usuario.....	53
3.1.2 Pruebas con Monkop	58
3.1.3 Manual de Usuario	59
Conclusiones del Capítulo:	71
Conclusiones.....	72
Recomendaciones.....	73
Referencias bibliográficas	74
Anexos	76

Tabla de Figuras

Figura 1: Ciclo de vida de Scrum para dispositivos móviles.....	9
Figura 2: Ciclo de vida de Mobile-D	10
Figura 3: Ejemplo de gráfico de barra	15
Figura 4: Las 4 perspectivas clásicas de un Cuadro de Mando Integral	20
Figura 5: Arquitectura del sistema.....	27
Figura 6: Prototipo de Pantalla: Pantalla Principal	32
Figura 7: Prototipo de Pantalla: Menú de Navegación Lateral	33
Figura 8: Prototipo de Pantalla: Pantalla Perspectiva	34
Figura 9: Prototipo de Pantalla: Pantalla Grupo	36
Figura 10: Prototipo de Pantalla: Pantalla Indicadores.....	37
Figura 11: Prototipo de Pantalla: Pantalla Indicador	38
Figura 12: Prototipo de Pantalla: Pantalla Seleccionar Perspectiva	39
Figura 13: Prototipo de Pantalla: Pantalla Seleccionar	40
Figura 14: Prototipo de Pantalla: Pantalla Gráfico Indicadores	42
Figura 15: Prototipo de Pantalla: Pantalla Opciones	43
Figura 16: Prototipo de Pantalla: Pantalla Acerca de	45
Figura 17: Prototipo de Pantalla: Pantalla Configuración	47

Introducción

En nuestro país se han implantado de forma exitosa varios sistemas de gestión contables. Los balances contables de cada mes permiten a los directivos analizar el cumplimiento o las desviaciones en los indicadores económicos planificados para la entidad. Sin embargo, hacer estos análisis solo de manera mensual resulta insuficiente. De esta forma se hace necesaria la existencia de herramientas que permitan dar un seguimiento sistemático al comportamiento no sólo de los indicadores económicos, sino de otros indicadores que caracterizan la marcha de la empresa, en función de los objetivos a lograr. Tales herramientas son conocidas como “cuadros de mando integrales” (CMI) o “balanced scorecard” (BSC por sus siglas en inglés). Pueden existir muchos otros informes que se reciben con periodicidad distinta y responden a actuaciones de cualquier área: informes de ventas, cierres de resultados etc. Se pretende presentar indicadores que puedan servir como señales de alerta, para lograr que las empresas cumplan las estrategias propuestas.

Los indicadores presentados en los CMI son reglas de cálculo que sirven para medir y valorar el cumplimiento de los objetivos estratégicos. Es muy importante su selección y definición ya que son los encargados de que la empresa avance en la dirección correcta.

En la Empresa Desoft de Villa Clara se desarrolló la herramienta Bimas la cual permite la planificación estratégica de una empresa y su seguimiento utilizando el concepto de Cuadro de Mando Integral (Perspectivas, Objetivos e Indicadores por objetivos). Su funcionamiento se basa en el uso de servicios de extracción, transformación y carga de datos desde fuentes y formatos diversos. Integra toda la información en un almacén de datos, para la posterior confección de indicadores, reportes y gráficos, para hacer uso controlado de dichos elementos de información.

Bimas está conformado por los siguientes sistemas:

1. BIMAS Server (v4.x): Aplicación web que permite la configuración de la estrategia (perspectivas, objetivos, relaciones entre objetivos e indicadores que se asocian a objetivos), distribución de reportes a destinatarios y brinda además variadas aplicaciones colaborativas como: calendarios corporativos e individuales, foros de discusión, encuestas, entre otros.
2. BIMAS Smifor (v2.x): Aplicación desktop que extrae los datos de los sistemas de gestión mediante SQL, y envía el resultado como tabla de datos hacia BIMAS Server, en un formato (DBF, XLS, CSV), por un canal (Email, FTP, HTTP, FS) y con una frecuencia determinada.
3. BIMAS Bion. Aplicación desktop sobre Windows para visualizar los indicadores seleccionados por el usuario con la frecuencia de actualización que el mismo defina. Muy útil a directivos, los cuales no tienen que conectarse a la página web de BIMAS a fin de localizar los

valores de aquellos indicadores que más les interesen, sino que les avisa sobre sus valores de manera paralela al resto de sus actividades sobre la PC.

En estos momentos no se cuenta con la posibilidad de dotar de movilidad la información sobre los indicadores del cuadro de mando a los directivos, ya que el componente BIMAS Bion se ejecuta solamente en PC con sistemas operativos Windows. Por esta razón los directivos tienen que trasladar la información hacia reuniones, o hacia otros directivos de nivel superior, por la vía de informes en formato PDF.

En este trabajo se define como **Problema de Investigación:**

Al no existir la posibilidad de dotar de movilidad la información actualizada sobre los indicadores de desempeño de una organización, los directivos pierden tiempo en la generación de los informes adecuados para mostrar en las reuniones, visitas u otras actividades de control planificadas.

Por lo que se propone como **Objetivo General:**

Desarrollar una herramienta sobre la plataforma Android para la visualización de los indicadores de desempeño implementados en el producto BIMAS Server, a fin de dotar a los directivos de esta valiosa información facilitando la movilidad de la misma.

Los objetivos específicos son los siguientes:

- 1- Caracterizar el marco teórico referencial de la investigación, en especial lo relativo a las metodologías y herramientas para el desarrollo de aplicaciones informáticas sobre el sistema operativo Android, así como las técnicas para la implementación y visualización de los indicadores de desempeño en las organizaciones.
- 2- Diseñar una aplicación para dispositivos móviles sobre la plataforma Android que permita visualizar los indicadores seleccionados por el usuario.
- 3- Implementar la conexión con el servicio API RESTful del producto Bimas Server y visualizar los datos sobre dispositivos móviles haciendo uso de gráficos.
- 4- Diseñar una Base de Datos que almacene los datos obtenidos del servicio y permita conservar los datos en las operaciones de sincronización con el servidor.

Preguntas de Investigación

- ¿Qué herramienta es la más adecuada para desarrollar una aplicación Android?
- ¿Qué lenguaje de base de datos y gestor es más apropiado para utilizar en una aplicación Android?

-¿Cuáles son las bibliotecas más usadas para implementar una conexión con una API RESTful y visualizar gráficos?

Justificación

Esta aplicación facilitará la presentación del nivel de cumplimiento de los indicadores de una empresa en las diferentes actividades que lo requieran, haciendo portable dicha información, lo cual proporcionará un mejor control de las empresas del país al basar la toma de decisiones en indicadores calculados a partir de los datos primarios registrados, en vez de estar basada en los informes que tradicionalmente se elaboran por los directivos.

Este trabajo consta de tres capítulos.

En el capítulo 1 se describen los principales conceptos, métodos, tecnologías y aplicaciones existentes en la actualidad, relacionados con la posible solución del problema de investigación planteado. Se llega a conclusiones sobre los métodos, tecnologías y software a desarrollar.

El capítulo 2 muestra la arquitectura de la solución y se documentan los principales artefactos de la metodología de desarrollo seleccionada. Haciendo uso para algunos de UML 2.0.

En el capítulo 3 se muestran las distintas pruebas aplicadas a la aplicación, así como las interfaces de usuario y una descripción de cómo se utiliza la aplicación desarrollada por los usuarios.

CAPÍTULO 1. CONSIDERACIONES TEÓRICAS SOBRE EL DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES Y SOBRE LA HERRAMIENTA BIMAS 4.0.

En este capítulo 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 al desarrollo de aplicaciones móviles. Se llega a conclusiones sobre los métodos y tecnologías a utilizar, los cuales permitirán una aplicación acorde al avance de los dispositivos móviles en cuanto a la compatibilidad con las distintas versiones del sistema operativo Android así como una interfaz de usuario acorde a lo sugerido por la empresa Google, la cual es la creadora de dicho sistema operativo. Se describen los principales aspectos relacionados con indicadores de desempeño en las Organizaciones y las técnicas para su representación de forma que se logre que los directivos entiendan correctamente la información mostrada.

1.1 Metodologías y métodos para el desarrollo de aplicaciones para dispositivos móviles.

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.(Lequerica, 2013)

Cuando el uso de dispositivos móviles se hizo algo común, también la programación de aplicaciones para estos se tomó más en serio, así diversas compañías crearon entornos de programación o complementos a Integrated Development Enviroment (IDE).

La lista siguiente muestra algunos de los IDEs y emuladores existentes:

- Pda Toolbox,
- Satellite Forms,
- Scotbuilder,
- Palm Os Developer Studio,

- HS Pascal,
- Mobile Visual Basic
- Embedded Visual Tools

Con el auge que han alcanzado las aplicaciones para dispositivos móviles y con los distintos sistemas operativos existentes para dichos dispositivos se han desarrollado plataformas de desarrollo que permiten la creación multiplataforma de aplicaciones con el mínimo de configuración para su implementación en ambientes específicos como son:

- Xamarin
- Appcelerator Titanium
- Phonegap

1.1.1 Tipos de aplicaciones móviles

Actualmente existen tres tipos de aplicaciones móviles en base a como son desarrolladas: (DeLl\ia et al., 2013)

- Apps nativas
- Web apps
- Apps híbridas

Aplicación Nativa: Son todas aquellas aplicaciones instaladas en el teléfono, diseñadas especialmente para el sistema operativo que porta el dispositivo.

Aplicación Web: Son aquellas web a las cuales se accede desde el navegador del dispositivo móvil, las cuales tienen que tener una versión pensada para utilizar en dispositivos pequeños brindando controles sencillos e intuitivos.

Aplicación Híbrida: Son aquellas aplicaciones que mezclan las características de las aplicaciones web y nativas. Pueden hacer uso de casi todas las aplicaciones del dispositivo.

1.1.2 Metodologías de desarrollo de software

“Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub - fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo.”(Avison, Cole and Fitzgerald, 2006)

Las metodologías tradicionales inician el desarrollo de un proyecto con un riguroso proceso de establecimiento de requerimientos, previo a etapas de análisis y diseño. En las metodologías tradicionales se concibe un solo proyecto, de grandes dimensiones y estructura definida; se sigue un proceso secuencial en una sola dirección y sin marcha atrás; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto. En la década de los noventa surgieron metodologías de desarrollo de software ligeras, más adelante nombradas como metodologías ágiles, que buscaban reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en los proyectos de desarrollo de software. Estas metodologías nacieron como reacción a las metodologías existentes con el propósito de disminuir la burocracia que implica la aplicación de las metodologías tradicionales en los proyectos de pequeña y mediana escala.(Cadavid, Martínez and Vélez, 2013)

Metodologías Ágiles:

(Poole, 2009) Define al desarrollo ágil como aquel que, en comparación con el desarrollo tradicional, provee beneficios de mayor flexibilidad, retorno de inversión más alto, realización más rápida del retorno de inversión, alta calidad y mayor visibilidad.

En febrero del 2001, tras una reunión celebrada en Utah, nace el término "ágil" aplicado al desarrollo de software. El objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y responder a los cambios que pueden surgir a lo largo del proyecto. Esto pretende ser una alternativa a los procesos de desarrollo tradicionales caracterizados por su total rigidez y muy dirigidos a la documentación que se genera tras cada una de las actividades desarrolladas. (Blanco *et al.*, 2009)

Tras la reunión se creó *The Agile Alliance* dedicada a promover el desarrollo ágil de software y ayudar a las empresas que lo adoptaran. El punto de partida fue el Manifiesto Ágil, documento que resume esa filosofía y que expone cuatro valores a tener en cuenta:

- El individuo y las interacciones del equipo de desarrollo están por encima del proceso y las herramientas. Construir un buen equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- Desarrollar software que funciona más que conseguir buena documentación. No producir documentos a menos que sean necesarios de una forma inmediata. Si el software no funciona, los documentos no valen de nada.
- La colaboración con el cliente es más importante que la negociación de contratos. Tiene que haber una interacción constante entre el cliente y el equipo de desarrollo.
- La respuesta ante el cambio es más importante que el seguimiento de un plan. La planificación no debe ser estricta sino flexible y abierta, la

habilidad de responder a los cambios que surjan determina el éxito o el fracaso del proyecto.

Algunas Metodologías Ágiles:

A continuación se describen algunas metodologías ágiles relacionadas al desarrollo de aplicaciones para dispositivos móviles.

Extreme Programing (XP):

Se centra en las mejores prácticas para el desarrollo de software. Consta de doce prácticas: el juego de planificación, pequeñas emisiones, la metáfora, el diseño sencillo, las pruebas, la refactorización, la programación en parejas, la propiedad colectiva, integración continua, semana 40-h, los clientes en el lugar, y los estándares de codificación. (Balaguera, 2013)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Las historias del usuario son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. (Canós and Letelier, 2012)

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración. (Canós and Letelier, 2012)

La metodología Extreme Programing (XP) posee las fases: planificación, diseño, desarrollo y pruebas.

Scrum:

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 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 el sprint siguiente.

La Figura 1 muestra el ciclo de vida de Scrum para dispositivos móviles destacándose la planificación de los 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. (Canós and Letelier, 2012)

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)

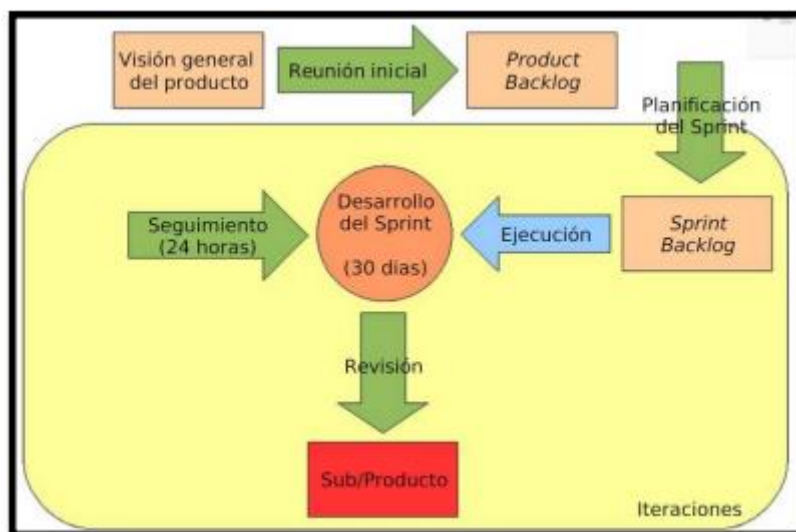


Figura 1: Ciclo de vida de Scrum para dispositivos móviles

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 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 2 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.

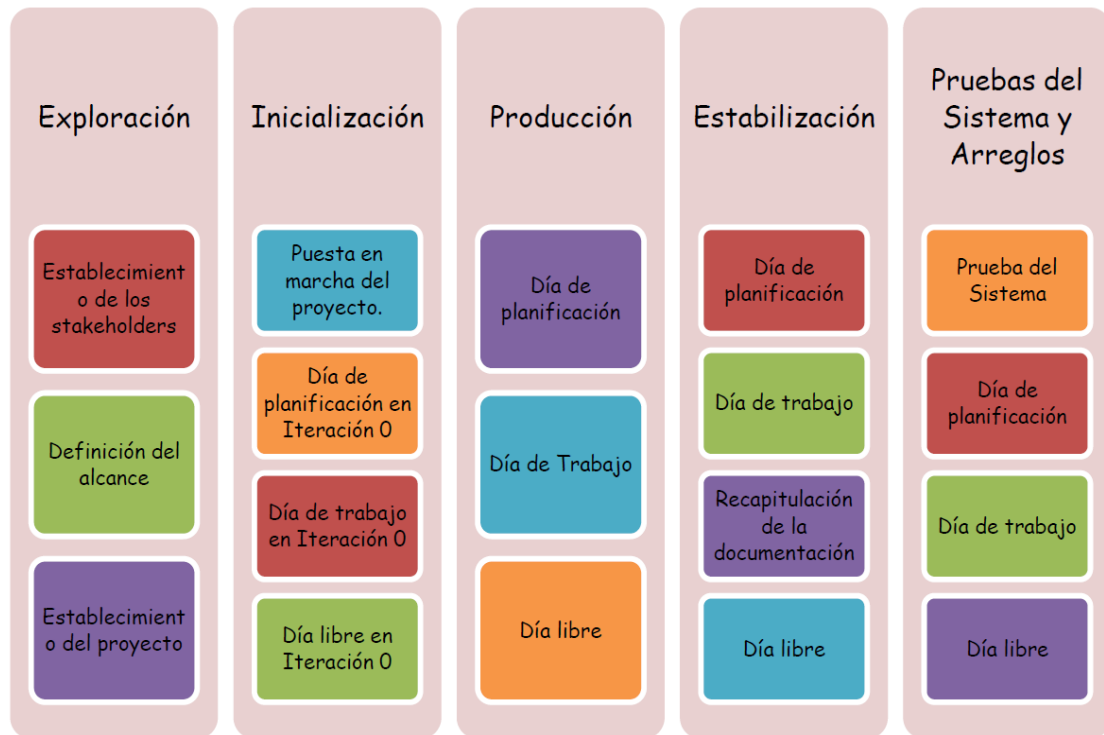


Figura 2: Ciclo de vida de Mobile-D

Fase de Exploración:

En la primera fase, Explorar, el equipo de desarrollo debe generar un plan y establecer las características del proyecto. Esto se realiza en tres etapas: establecimiento actores, definición del alcance y el establecimiento de proyectos. Las tareas 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 y 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 todos ellos sean corregidos con prontitud en el final de 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 la fase de producción se repite la programación de tres días (planificación, trabajo, liberación) se repite iterativamente hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de

requisitos y tareas a realizar. 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, se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funciona correctamente. Esta será la fase más importante en los proyecto multi-equipo con 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 Prueba del Sistema:

La última fase (prueba 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:

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.
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.

Desventajas:

1. No se adapta a equipos demasiado grandes o segmentados.

Tabla comparativa entre las tres metodologías mencionadas anteriormente.

	Metodología Extreme Programing	Metodología Scrum para móviles	Metodología Mobile D
Fases	Fases: planificación, diseño, desarrollo y pruebas.	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	Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo	El Product Owner maneja la comunicación con el cliente	La comunicación con el cliente es menor que en Scrum.
Programación	Programación en parejas, en jornadas largas revisión de código mutuo.	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 son las historias de usuario y tarjetas CRC.	Lo principal lo constituyen las historias de usuario.	Lo principal es el modelo del dominio, storycards y storyboards.
Pruebas	Se realizan unidades de pruebas, pruebas de aceptación y pruebas unitarias.	Se realizan pruebas unitarias de pruebas de integración, pruebas de aceptación.	Se realizan pruebas unitarias, pruebas de interfaz, pruebas de datos ingresados y pruebas funcionales.

Luego de analizar la tabla comparativa se determinó que las fases de las metodologías XP y Mobile-D se parecen a las fases de 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 por cada iteración. La documentación en la metodología XP y en Scrum tiene un nivel bajo en comparación con la metodología Mobile-D. Sin embargo, Mobile-D no exige una gran cantidad de documentación, solo lo necesario para guiarse en el desarrollo de la aplicación. (Chinchay Cuenca, 2016)

Después de comparadas las metodologías anteriores, teniendo en cuenta que el grupo de desarrollo está conformado por una persona y se dispone de poco 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.

1.1.3 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.(Hohensee, 2014)

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 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)

1.1.4 SQLite:

Las aplicaciones Android profesionales se diseñan cada vez más como los sistemas de escritorio, con bases de datos para gestionar el almacenamiento y recuperación de información, que ayudan a mejorar sus capacidades y prestaciones. (Vanegas, 2012)

En Android hay varias posibilidades para trabajar con base de datos. Podemos utilizar un sistema embebido como SQLite o conectar de diversas formas a una base de datos externa, como MySQL. Además, podemos implementar un ORM, que nos facilite la gestión mediante una base de datos virtual orientada a objetos, o un Content Provider, que nos permita compartir datos entre aplicaciones.

El gestor de base de datos SQLite provee memoria interna y base de datos relacionales, fácilmente persistente y de fácil acceso a sus datos de forma estructurada habilitada para aplicaciones Android (Cinar, 2015) . Es código abierto, estable y popular en muchos pequeños dispositivos. No necesita servidor ni proceso corriendo es básicamente una biblioteca y es un solo

archivo. El Framework de Android ofrece diversas formas para usarlo. (Gargenta and Nakamura, 2014)

1.1.5 Material Design

Es un Lenguaje de diseño desarrollado por Google que sintetiza los principios del buen diseño junto con la innovación y la tecnología. Empezó utilizándose en Android Lollipop 5.0. Está enfocado a mejorar la experiencia de usuario y tiene muy en cuenta los dispositivos móviles/táctiles. (Benbourahla, 2015)

Está basado en formas, iconografía, tipografía, espacio, color, imágenes-guía, tratamientos visuales y todo ello ordenado de forma consciente para crear una jerarquía, comunicar un significado y siempre desde un enfoque concreto. El diseño es simple pero deliberado: todo está intencionado. (*Material Design de Google _ GUÍA EN ESPAÑOL _ Agencia MDW*, no date)

Destaca por el uso de colores vibrantes sobre ambientes apagados y un guiño constante a la arquitectura.

1.1.6 WireframeSketcher

Es una herramienta basada en Eclipse y con todos los elementos nativos de Android, también tiene los de IOS y Windows Phone entre otros. Da la posibilidad de utilizar el paquete de iconos nativos para poner los elementos de menú, el ícono de la aplicación o cualquier otra imagen. Permite simular la navegación entre las pantallas, lo que ayuda a visualizar y revisar todos los enlaces entre ellas creando un Storyboard. Este tipo de fichero nos permite agregar pantallas de nuestro proyecto las cuales podemos luego exportar como fichero HTML con los enlaces entre ellas. Entre las principales funcionalidades de esta herramienta se encuentra la exportación de archivos lo cual permite interactuar con el prototipo y así ver cómo quedaría la aplicación. (Ko *et al.*, 2013)

La primera forma es exportar un Storyboard a HTML, teniendo así un subconjunto de pantallas navegables a través del navegador, algo que podemos entregar a un cliente para que valide que es así como se debe organizar la aplicación.

Otras formas más estáticas de exportar nuestro trabajo es usando ficheros PDF e imágenes en formato PNG.

Por lo anterior expuesto vemos que es una herramienta extensible y muy fiel al diseño final. Su uso es bastante sencillo y nos da la opción de exportar a HTML, algo ideal para enviar a clientes o compañeros.

1.1.7 MPAndroidChart

Es una biblioteca gratuita ideal para la creación de gráficas (ver ejemplo en la Figura 3) para dispositivos Android. Posee dos grandes grupos de tipos de gráficas: basadas en líneas y del tipo tarta. Brinda la posibilidad de modificar grandemente estos para que su resultado final sea esperado. Entre sus funcionalidades podemos destacar que permite gestos, se puede escalar y arrastrar fácilmente. También permite exportar los gráficos como imágenes. Su instalación es sumamente sencilla y se hace de forma muy similar a la de cualquier otro elemento que comúnmente manipulemos en nuestras aplicaciones Android. (Pablo, 2014)

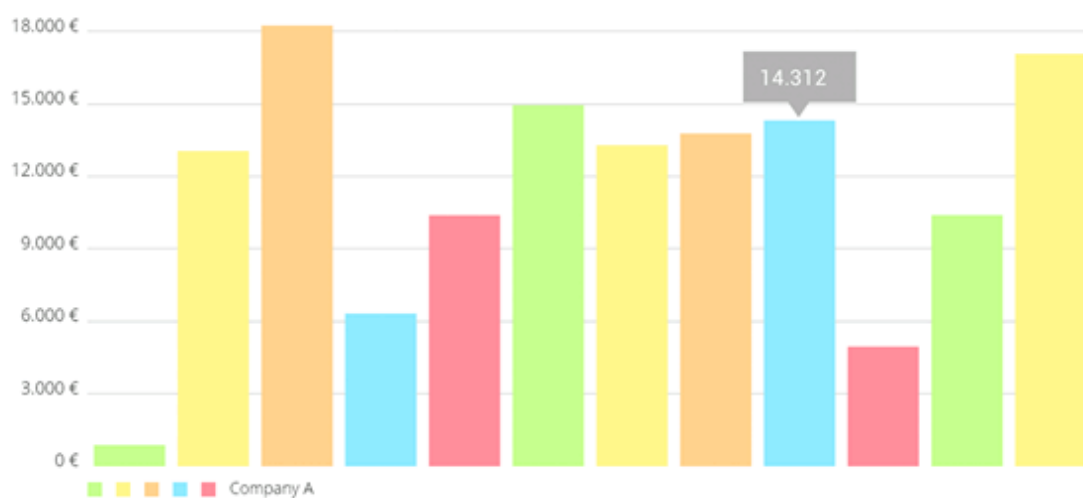


Figura 3: Ejemplo de gráfico de barra

1.1.8 Web Services

El término Web Services describe una forma estandarizada de integrar aplicaciones WEB mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de la Internet. XML es usado para describir los datos, SOAP se ocupa para la transferencia de los datos, WSDL se emplea para describir los servicios disponibles y UDDI se ocupa para conocer cuáles son los servicios disponibles. Uno de los usos principales es permitir la comunicación entre las empresas y sus clientes. Los Web Services permiten a las organizaciones intercambiar datos sin necesidad de conocer los detalles de sus respectivos Sistemas de Información.

A diferencia de los modelos Cliente/Servidor, tales como un servidor de páginas Web, los Web Services no proveen al usuario una interfaz gráfica (GUI). En vez de ello, los Web Services comparten la lógica del negocio, los datos y los procesos, por medio de una interfaz de programas a través de la red. Es decir conectan programas, por tanto son programas que no interactúan

directamente con los usuarios. Los desarrolladores pueden por consiguiente agregar a los Web Services la interfaz para usuarios, por ejemplo mediante una página Web o un programa ejecutable, tal de entregar a los usuarios una funcionalidad específica que provee un determinado Web Service.

Estos mensajes van de un sistema a otro, por lo general, vía HTTP. El sistema receptor interpreta el mensaje, hace lo que se supone que debe hacer, y devuelve una respuesta con la forma de otro mensaje SOAP.

Tipos de Web Service:

SOAP:

SOAP (Simple Object Access Protocol) es un protocolo que ha sido elaborado para dialogar con un web services. SOAP se ocupa de la transmisión de datos, la llamada de métodos, el paso de argumentos y el envío al cliente de los valores de retorno.

Implica el envío de un mensaje. Estos mensajes se componen de XML, que es un estándar abierto basado en texto, accesible por cualquier persona desde cualquier aplicación. Esto amplía el mundo de su aplicación para que cualquier persona pueda acceder a ella en su red.

La asociación HTTP/XML hace a los mensajes SOAP independientes de los sistemas operativos, los lenguajes y las restricciones de hardware. Un web services escrito con un lenguaje .NET puede, por ejemplo, ser explotado en una petición Java o Delphi, y viceversa. El web services y su aplicación cliente pueden estar albergados en sistemas operativos distintos. La comunicación a través de internet se consigue con la utilización de HTTP. Se utiliza el puerto 80, lo que hace que los contrafuegos (firewall) no supongan ningún obstáculo, a diferencia de otros protocolos binarios que necesitan la apertura en los contrafuegos de puertos suplementarios, lo cual supone un problema de seguridad. (Pintado *et al.*, 2015)

REST:

Define un set de principios arquitectónicos por los cuales se diseñan web services haciendo foco en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren por HTTP hacia los clientes escritos en diversos lenguajes. REST emergió en los últimos años como el modelo predominante para el diseño de servicios. De hecho, REST logró un impacto tan grande en la web que prácticamente logró desplazar a SOAP y las interfaces basadas en WSDL por tener un estilo bastante más simple de usar. (Seta, no date)

Una implementación concreta de un servicio web REST sigue cuatro principios de diseño fundamentales:

- Utiliza los métodos HTTP de manera explícita.
- No mantiene estado.

- Expone URLs con forma de directorios.
- Transfiere XML, JavaScript Object Notation (JSON), o ambos.

REST hace que los desarrolladores usen los métodos HTTP explícitamente de manera que resulte consistente con la definición del protocolo. Este principio de diseño básico establece una asociación uno-a-uno entre las operaciones de crear, leer, actualizar y borrar y los métodos HTTP. De acuerdo a esta asociación:

- Se usa POST para crear un recurso en el servidor.
- Se usa GET para obtener un recurso.
- Se usa PUT para cambiar el estado de un recurso o actualizarlo.
- Se usa DELETE para eliminar un recurso.

1.1.9 RESTful API:

Se conoce como API (del inglés: “Application Programming Interface”), o Interfaz de programación de Aplicaciones, al conjunto de rutinas, funciones y procedimientos (métodos) que permiten utilizar recursos de un software por otro, sirviendo como una capa de abstracción o intermediario. Podemos describir un API como método generalizado de consumir un servicio. Un API se crea con el objetivo de que cualquier software pueda “consumir” nuestros recursos, por lo general datos, sin importar en que lenguaje o plataforma en que sea creado. RESTful API es el sucesor de métodos anteriores como SOAP y WSDL cuya implementación y uso son un poco más complejos y requieren mayores recursos y especificaciones al ser usados. (Federico Guzman, 2016)

Un RESTful API es un servicio que funciona como un estándar para compartir información, en un sistema de doble vía: Consulta y Respuesta (Request -> Response). Por lo general y mejor práctica, el cuerpo (Body) de la respuesta de un API es una estructura en formato JSON. Aunque también puede ser una estructura XML o cualquier otra estructura de datos de intercambio, incluso una personalizada. Sin embargo, como el objetivo es permitir que cualquier cliente pueda consumir el servicio de un API, lo ideal es mantener una estructura estándar, por lo que JSON es la mejor opción. (Federico Guzman, 2016)

1.2 Indicadores de desempeño en las Organizaciones.

Una Organización es una entidad social orientada hacia objetivos específicos y estructurada de manera deliberada. La organización es una entidad social porque está constituida por personas, y se orienta hacia objetivos porque está diseñada para alcanzar resultados. Siendo los indicadores la forma de medirlos.

Un **indicador** es la expresión cuantitativa del comportamiento o del desempeño de toda una organización o una de sus partes, cuya magnitud al ser comparada

con algún nivel de referencia, puede señalar una desviación sobre la cual se tomarán acciones correctivas o preventivas según el caso.

El **valor del indicador** es el resultado de la medición del indicador y constituye un valor de comparación, referido a su meta asociada. Los indicadores son la expresión del cumplimiento de los objetivos.

Los indicadores son de suma importancia ya que (Armijo, 2009):

- Son herramientas de gestión que proveen un valor de referencia a partir del cual se puede establecer una comparación entre las metas planeadas y el desempeño logrado.
- Un indicador de desempeño es una herramienta que entrega información cuantitativa respecto del logro o resultado de los objetivos de la institución.
- Los indicadores de desempeño son medidas que describen cuán bien se están desarrollando los objetivos de una institución, a qué costo y con qué nivel de calidad.

Clasificación de los indicadores:

Según qué miden: (dimensiones del desempeño):

- Eficacia
- Calidad
- Eficiencia
- Economía

Indicador de eficacia:

- Mide el grado de cumplimiento de un objetivo
- No considera los recursos asignados para ello

Ejemplos:

- Porcentaje de becas entregadas en relación a las programadas
- Reducción de la tasa de analfabetismo

Indicador de eficiencia:

- Mide la relación entre los productos y servicios generados con respecto a los insumos o recursos utilizados.

Aproximaciones a este concepto:

- Costos de producción (costos medios, costos unitarios, gastos administrativos en relación al gasto total)
- Productividad media de los distintos factores de producción

Ejemplos:

- Costo promedio anual por alumno atendido
- Solicitudes tramitadas por funcionario

Indicador de calidad:

- Mide atributos, capacidades o características que deben tener los bienes o servicios para satisfacer adecuadamente los objetivos del programa.
- Calidad es una dimensión de la eficacia, pero por su importancia se la distingue con nombre propio

Ejemplos:

- Grado de satisfacción de los usuarios
- Tiempo de respuesta a requerimientos de los beneficiarios

Indicador de economía:

- Mide la capacidad del programa para generar o movilizar adecuadamente los recursos financieros.
- Economía es una dimensión de la eficiencia.

Ejemplos:

- Ingresos propios generados sobre Presupuesto Total
- Presupuesto ejercido respecto del presupuesto total

1.2.1 Cuadro de Mando Integral:

El término “Cuadro de Mando Integral” (CMI) surge en 1990 cuando el profesor Robert Kaplan de la Universidad de Harvard y David Norton (ver Figura 4), consultor empresarial, deciden hacer un estudio de investigación sobre un grupo de empresas, explorando nuevos métodos para medir la actividad y los resultados obtenidos conociendo que las medidas financieras de la actividad empresarial no eran adecuadas en el entorno competitivo. En años fue adoptado por diferentes empresas, obteniendo resultados muy positivos, por lo que éste se fue consolidando como herramienta clave para el despliegue de la estrategia empresarial. Ha tenido una influencia revolucionaria en la forma de controlar la empresa moderna. La razón no es otra que su estricto enfoque

hacia la maximización del beneficio empresarial haciendo el mejor uso posible de los recursos disponibles.(KAPLAN and NORTON, 2005)

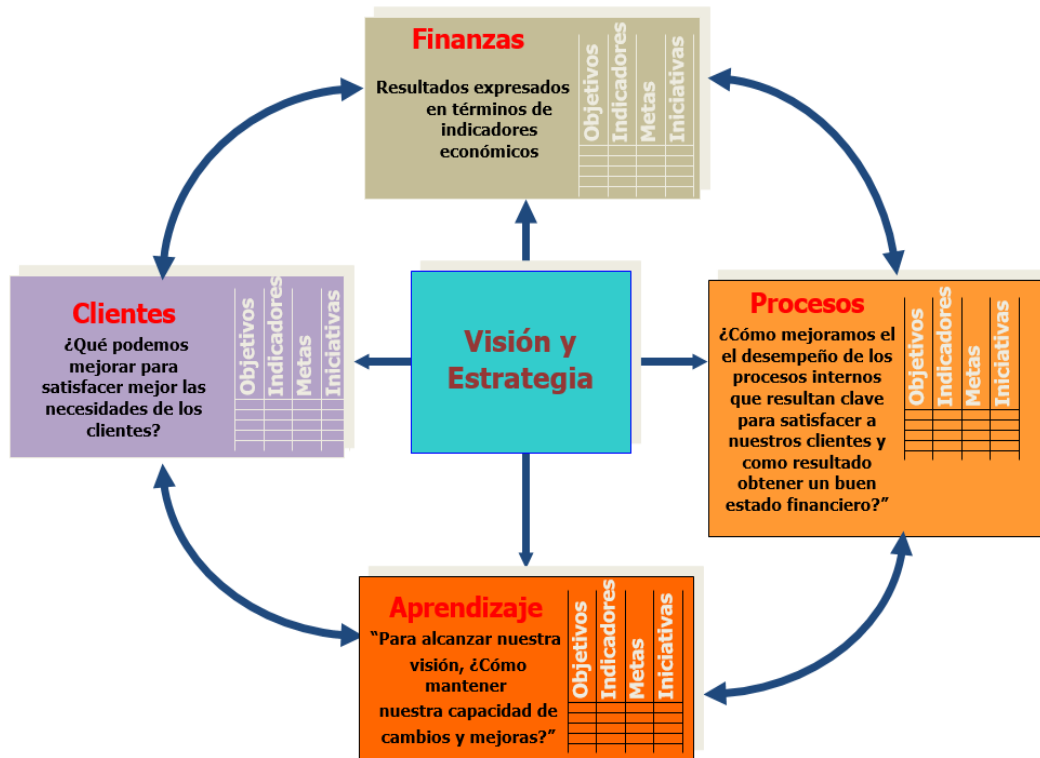


Figura 4: Las 4 perspectivas clásicas de un Cuadro de Mando Integral

El cuadro de mando es un componente más del "sistema de información" de la dirección. Pueden existir muchos otros informes que se reciben con periodicidad distinta y responden a actuaciones de cualquier área: informes de ventas, cierres de resultados etc. Lo que pretende es presentar indicadores que pueden servir como señales de alerta, para lograr que las empresas cumplan las estrategias propuestas.

Los indicadores presentados en los CMI son reglas de cálculo que sirven para medir y valorar el cumplimiento de los objetivos estratégicos. Es muy importante su selección y definición ya que son los encargados de que la empresa avance en la dirección correcta.

Existen dos tipos de indicadores:

Inductores: son los que miden las acciones que se realizan para conseguir el objetivo.

Resultados: son los que miden el grado de obtención de resultados.

El Cuadro de Mando Integral (CMI) es una herramienta útil para dirigir empresas de forma proactiva en el corto y en el largo plazo. Es una técnica de gestión y planificación. Su eficiencia radica en una buena comprensión de sus

fundamentos y una aplicación completa que incluya a toda la dirección. Se elabora un mapa estratégico donde se reflejan las estrategias y los objetivos a conseguir en cuatro áreas fundamentales: financiera, procesos internos, clientes e innovación y formación.(KAPLAN and NORTON, 2005)

Es considerado una herramienta la cual usa medidas en cuatro categorías: desempeño financiero, conocimiento del cliente, procesos internos de negocios y; aprendizaje y crecimiento, para alinear iniciativas individuales, organizacionales y departamentales e identifica procesos enteramente nuevos para cumplir con objetivos del cliente y accionista.

Está compuesto por los siguientes elementos:

- Objetivos estratégicos
- Perspectivas
- Indicadores
- Metas
- Mapas Estratégicos
- Proyectos Estratégicos

Entre las ventajas que la aplicación del CMI puede aportar a las organizaciones, destacan las siguientes:

- Clarificar y actualizar la estrategia de la corporación.
- Ayudar a comunicar la misión a toda la organización; es decir, ayuda a pasar de las grandes declaraciones de intenciones al trabajo diario de cualquiera de los colaboradores de una organización.
- Identificar y alinear las iniciativas estratégicas.
- Enlazar los objetivos estratégicos con las metas a largo plazo y los presupuestos anuales.
- Alinear las revisiones estratégicas y operacionales.
- Obtener retroalimentación para aprender acerca de la estrategia y mejorarla; es decir, contribuye a la revisión permanente de la estrategia.

1.2.2 BIMAS 4.0

En nuestro país está consolidado el uso de sistemas de gestión contable financiera, siendo el Balance contable un informe que refleja la situación económica y financiera de una empresa en cada mes del año.

En nuestro país se han implantado con éxito sistemas de gestión contables. Los balances contables de cada mes permiten a los directivos analizar el cumplimiento o las desviaciones en los indicadores económicos planificados para la entidad. Sin embargo, hacer estos análisis solo de manera mensual resulta insuficiente. Es necesario poder contar con herramientas que permitan dar seguimiento más sistemático al comportamiento no sólo de los indicadores económicos, sino de otros indicadores que caracterizan la marcha de la empresa, en función de los objetivos a lograr. Tales herramientas son

conocidas como “cuadros de mando integrales” (CMI) o “balanced scorecard” (BSC por sus siglas en inglés).

Bimas es un instrumento para la planificación estratégica que proporciona herramientas de seguimiento estratégico, basadas en la extracción de datos y su transferencia para garantizar la oportunidad en el acceso a la información. Entre sus funcionalidades se encuentra la definición de Perspectivas, Objetivos e indicadores para medir el desempeño, el cálculo automático de indicadores a partir de fórmulas que consultan fuentes de datos diversas y cálculo de reportes para sustituir los informes manuales.

Esta herramienta posee una arquitectura cliente-servidor y está dividida en tres componentes:

1. BIMAS Server (v4.x): Aplicación web que permite la configuración de la estrategia (perspectivas, objetivos, relaciones entre objetivos e indicadores que se asocian a objetivos), distribución de reportes a destinatarios, cuadro de mando y brinda además variadas aplicaciones colaborativas como: calendarios corporativos e individuales, foros de discusión, encuestas, entre otros.
2. BIMAS Smifor (v2.x): Aplicación desktop que extrae los datos de los sistemas de gestión mediante SQL, y envía el resultado como tabla de datos hacia BIMAS Server, en un formato (DBF, XLS, CSV), por un canal (Email, FTP, HTTP, FS) y con una frecuencia determinada.
3. BIMAS Bion. Aplicación desktop sobre Windows para visualizar indicadores seleccionados por el usuario con la frecuencia de actualización que el mismo defina. Muy útil a directivos, los cuales no tienen que conectarse a la página web de BIMAS a fin de localizar los valores de aquellos indicadores que más les interesen, sino que les avisa sobre sus valores de manera paralela al resto de sus actividades sobre la PC.

En estos momentos no se cuenta con la posibilidad de dotar de movilidad la información sobre los indicadores del cuadro de mando, a los directivos. En su lugar, estos tienen que trasladar la información hacia reuniones u otros directivos de nivel superior por la vía de informes en formato PDF, que ocasiona que los directivos pierdan tiempo en la generación de los informes adecuados para mostrar en las reuniones, visitas u otras actividades de control planificadas.

Por lo que se propone desarrollar una herramienta sobre la plataforma Android para la visualización de los indicadores de desempeño implementados en el producto BIMAS Server, a fin de dotar a los directivos de esta valiosa información facilitando la movilidad de la misma.

1.3 Conclusiones del capítulo:

En este capítulo se fundamentan las herramientas y la metodología utilizada para dar solución, al desarrollo de la aplicación para plataforma Android de Tablero de control de indicadores de desempeño. Para así mejorar el desempeño empresarial actual con la posibilidad de la movilidad.

Después de lo expuesto en este capítulo se concluye que:

- 1- La aplicación a desarrollar se clasifica como nativa, ya que estará diseñada específicamente para el sistema operativo Android y no podrá ser accedida desde el navegador.
- 2- Para el desarrollo de la aplicación se decidió optar por una metodología ágil ya que estas tienen un tiempo corto para la entrega del producto final y son las más usadas en el desarrollo de aplicaciones para dispositivos móviles.
- 3- Entre los Framework de desarrollo de aplicaciones móviles para plataforma Android es recomendable Android Studio, ya que es el IDE oficial para dicho sistema operativo y brinda muchas facilidades.
- 4- Para la visualización de los indicadores se escogió la biblioteca MPAndroidChart. Esta es una de las más utilizadas para la creación de los diferentes tipos de gráficos.
- 5- La aplicación a desarrollar para la plataforma Android debe permitir la sincronización con el servicio que brinda BIMAS guardando la información referente al CMI de las organizaciones en una base de datos local permitiendo así su control sin tener que mantener la conexión en todo momento.
- 6- Se requiere proveer al usuario de gráficos mediante los cuales se muestre la información perteneciente a los indicadores de desempeño, así como la posibilidad de compartir estos datos en un documento PDF mediante las distintas formas permitidas por los dispositivos móviles.

CAPÍTULO 2: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN.

En este capítulo se documentan los artefactos de la aplicación “Bimas Bion Android”, de acuerdo con las fases de la metodología ágil Mobile-D, metodología seleccionada para la concepción y desarrollo de la aplicación. Se describen los storyboard y storycard, los cuales representan la principal documentación de la metodología Mobile-D, estos muestran visualmente como se observara cada pantalla de la aplicación. Se confecciona el mapa de navegación, el cual permite conocer cómo acceder a cada pantalla. Se describen los métodos y clases de más interés de la implementación.

2.1 Primera Fase: Análisis

En esta fase se busca realizar el análisis de los requisitos funcionales mediante el estudio de la problemática.

2.1.1 Exploración

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

Establecimiento de los Grupos de Interés o Stakeholders:

- **Desarrollador:** Es la persona encargada del análisis, desarrollo y pruebas de la aplicación.
- **Desarrolladores de la herramienta BIMAS:** Personas que conocen el funcionamiento de BIMAS, así como del servicio a consumir por la aplicación móvil y los requerimientos funcionales de la misma. También definen junto al desarrollador la interfaz de usuario.
- **Directivos empresariales:** Todo personal de organizaciones donde se despliegue la herramienta BIMAS que posea autorización para acceder a la información de los indicadores de desempeño de la organización.

Requerimientos Iniciales:

Se pretende desarrollar una aplicación móvil para el sistema operativo Android que permita visualizar los indicadores de desempeño de cualquier empresa que despliegue la herramienta BIMAS.

Requerimientos Funcionales:

Tabla 2. Requerimientos funcionales

Código	Descripción
RF1	Listar organización, así como sus sub organizaciones.
RF2	Listar Perspectivas por organización.
RF3	Listar Grupos por organización.
RF4	Listar indicadores de desempeño por grupo.
RF5	Visualizar información de cada indicador.
RF6	Permitir configuración de conexión con el servidor de datos.
RF7	Autenticación para actualización de datos.
RF8	Actualización de la información.
RF9	Compartir información.
RF10	Visualizar gráficos de los indicadores.

Requerimientos no Funcionales

Tabla 3. Requerimientos no funcionales

Código	Descripción
RNF1	El sistema será desarrollado para la plataforma Android.
RNF2	El Sistema será desarrollado bajo la plataforma de programación Java.
RNF3	El sistema utilizará interfaz amigable y responsive (sensible).
RNF4	El sistema permitirá el acceso a los usuarios sin necesidad de autenticarse menos la primera vez para actualizar la base de datos.
RNF5	El sistema debe ser fácilmente instalable por los usuarios en equipos móviles con la versión del sistema operativo Android 4.0 o superiores

Definición del Alcance

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

Limitaciones

Las limitaciones de la aplicación son:

- La aplicación sólo puede ser ejecutada en dispositivos con el Sistema Operativo Android.
- Para la actualización de la información brindada por la aplicación es necesario estar conectado al servidor del sistema BIMAS Server.

Supuestos y dependencias

Los supuestos y dependencias de la aplicación son:

- La aplicación actualiza los datos de los indicadores de desempeño a través de un servicio API RESTful.
- Los usuarios manejan como idioma principal el español, siendo éste el lenguaje que se muestra en la interfaz de la aplicación.

2.2 Segunda Fase: Diseño

A través del diseño se desean obtener los diagramas que representen el funcionamiento de la aplicación, los procesos que va a ejecutar con el fin de cumplir los requerimientos definidos en la primera fase; además la descripción de cada uno de ellos y las configuraciones necesarias para desarrollar la aplicación móvil.

2.2.1 INICIALIZACIÓN

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

Configuración del Ambiente de Desarrollo:

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

Configuración para aplicaciones móviles Android:

- **Tipo de Proyecto:** Android Application Project.
- **Configuraciones:** Inclusión de la biblioteca mpandroidchart para la visualización de gráficos, de la biblioteca itextg para la creación del documento PDF a compartir y android-async-http-1.4.9 para la conexión con el servicio.

Diseño del Sistema:

El diseño general del sistema está compuesto por un servidor principal de aplicaciones web y uno de base de datos, el servicio RESTful API y el usuario móvil. (Ver Figura 5)

El servidor de base de datos, administrado bajo el gestor MongoDB, es quien proporciona la información acerca de los indicadores de desempeño al servidor principal y estos datos se consumen utilizando el API que proporciona BIMAS Server.

El API RESTful es el encargado de enviar la respuesta a la consulta hecha por la aplicación móvil.

La arquitectura de la aplicación móvil utiliza el estilo arquitectónico de llamada y retorno Modelo-Vista-Controlador (MVC) y está orientada a servicios.

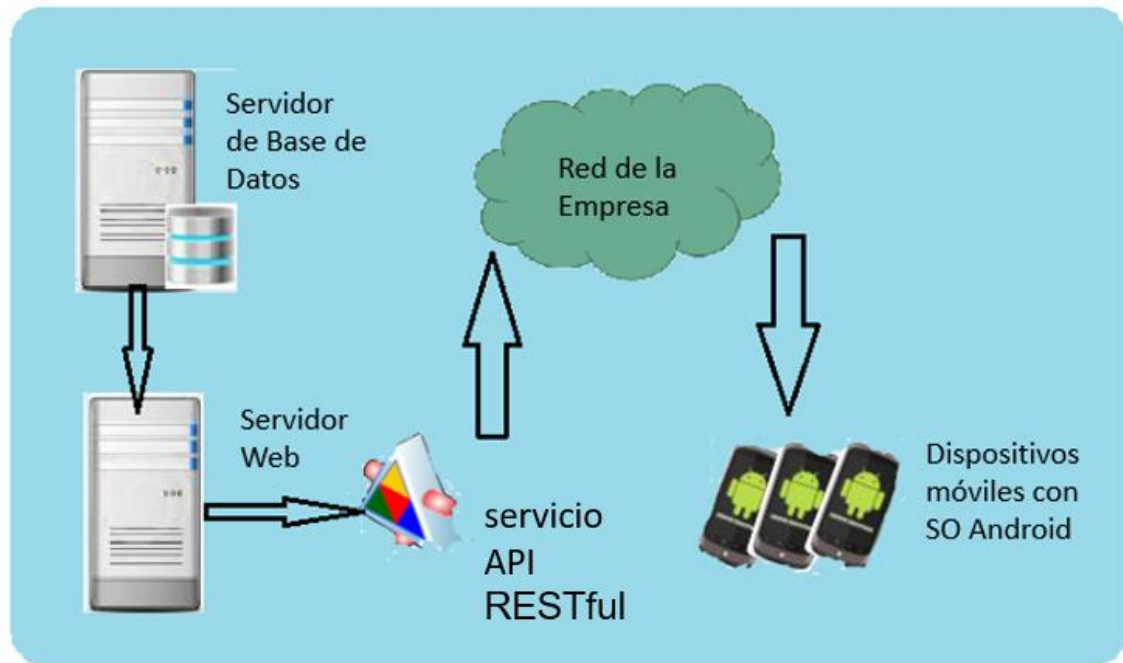


Figura 5: Arquitectura del sistema

Modelo del Dominio:

A continuación se detallan las clases fundamentales que componen el sistema “Bimas Bion Android”. Siendo la clase Indicador la principal ya que es el objeto principal de análisis de la aplicación, es la que posee todos los atributos a los que se le quiere dar seguimiento. Para ello se necesitan las demás clases, las cuales componen la estructura del **CMI**, como se puede observar en el diagrama 1.

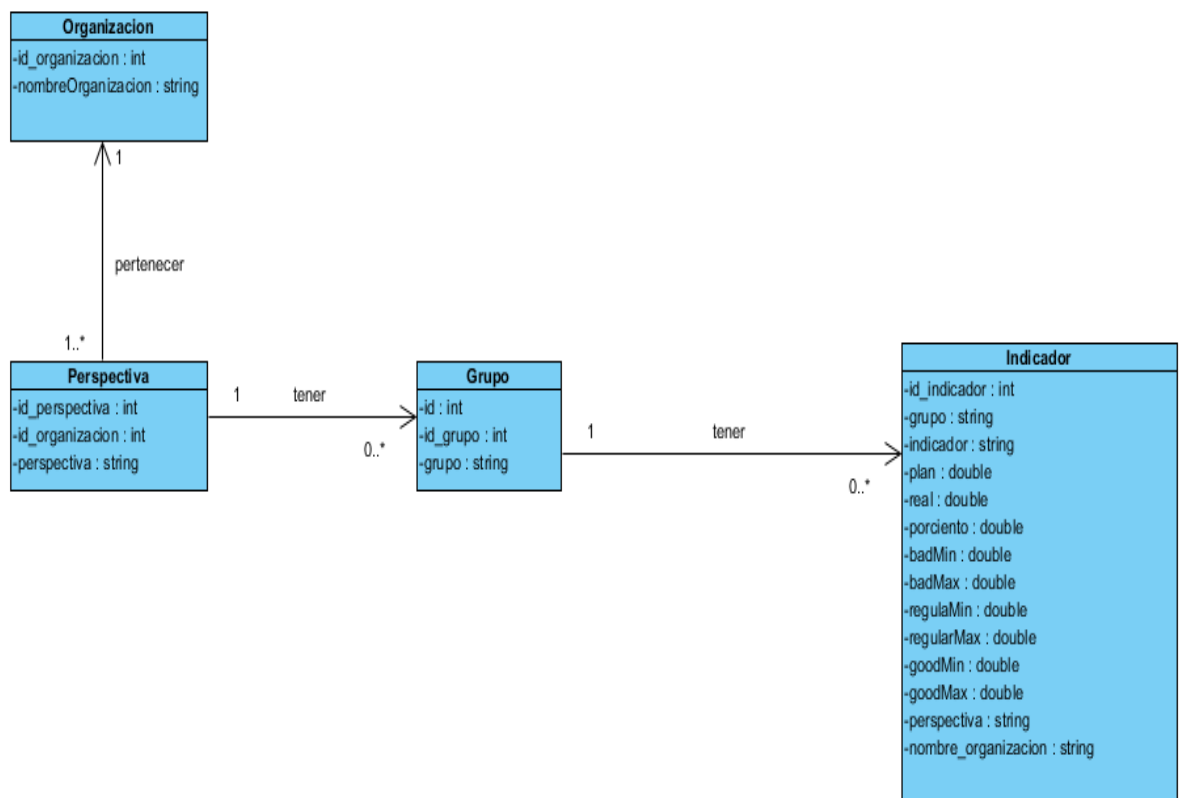


Diagrama 1: Modelo de dominio

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. A continuación, el esquema diseñado para la base de datos del sistema "Bimas Bion Android" el cual sirve como fuente de información para dicha aplicación. (Ver diagrama 2)

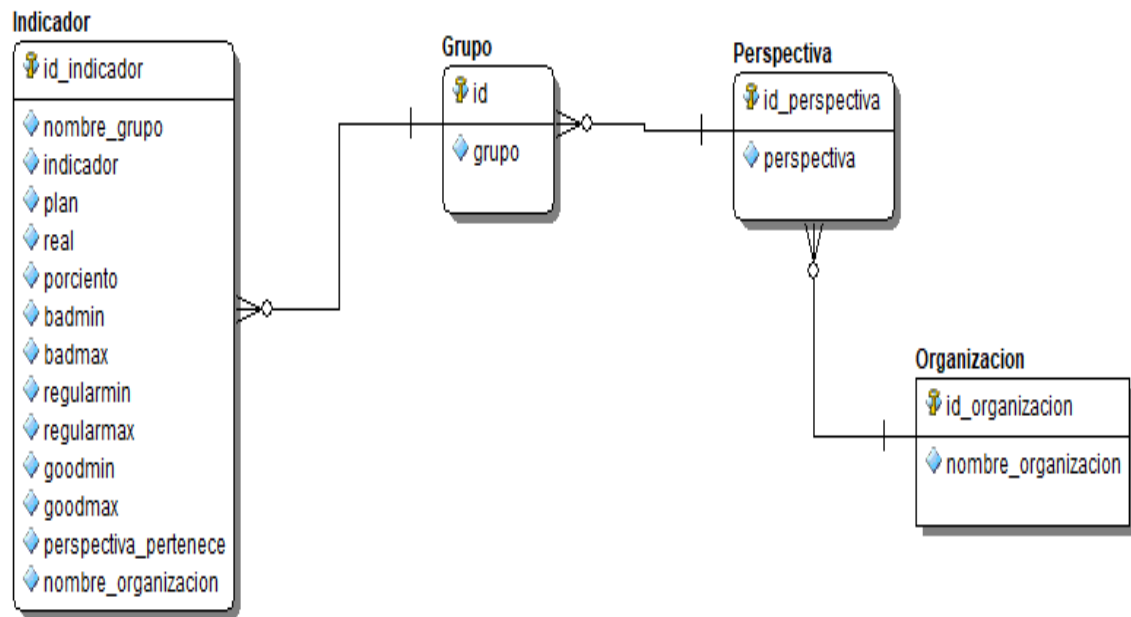


Diagrama 2: Modelo lógico

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. (Ver diagrama 3)

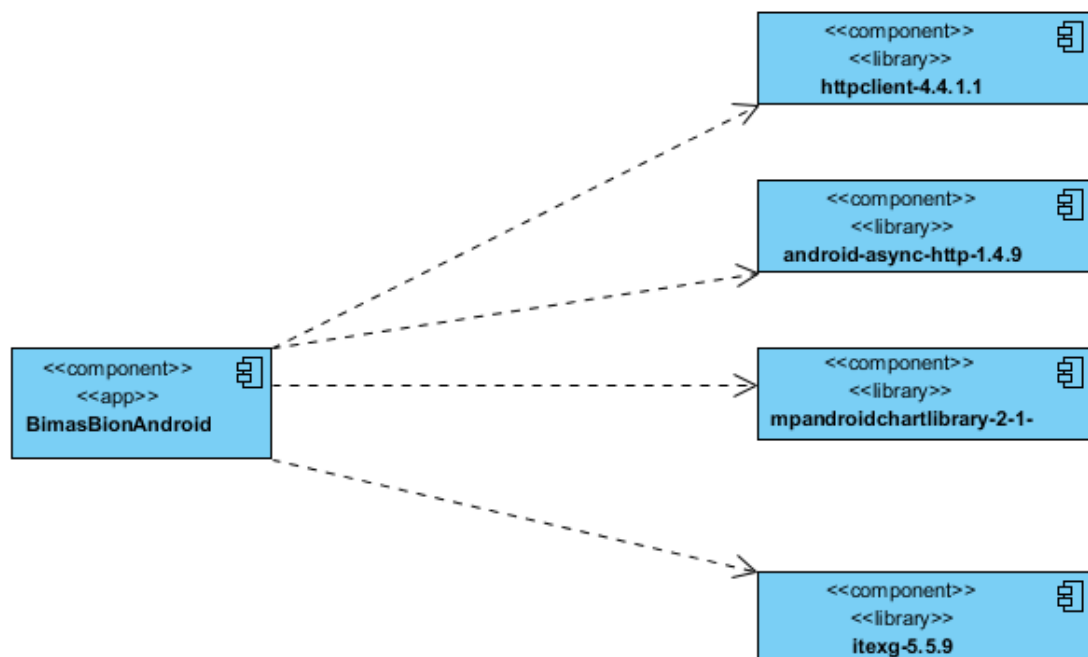


Diagrama 3. Diagrama de componentes

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. (Ver Diagrama 4)

Aunque no se visualiza en el diagrama 4, al instalar y ejecutar por primera vez la aplicación la primera pantalla será la guía rápida, con la cual se pretende dar al usuario una breve explicación de cómo utilizar las principales funcionalidades de la aplicación.

Posteriormente se puede visualizar el Splash, pantalla en blanco con el icono de la aplicación en el centro. Normalmente pasaría a visualizarse la pantalla principal la cual posee un listado con la organización y sub organizaciones, a las cuales tiene autorización para darle seguimiento el usuario autenticado, pero si es primera vez que se ejecuta la aplicación la base de datos estará vacía por lo cual se visualiza la pantalla configuración. Esta pantalla permite configurar los parámetros fundamentales para establecer la conexión con el API de Bimas Server como son: dirección y puerto.

La pantalla principal también cuenta con un menú lateral, el cual permite tres funcionalidades, las cuales son: actualizar, visualizar gráficos de indicadores y compartir.

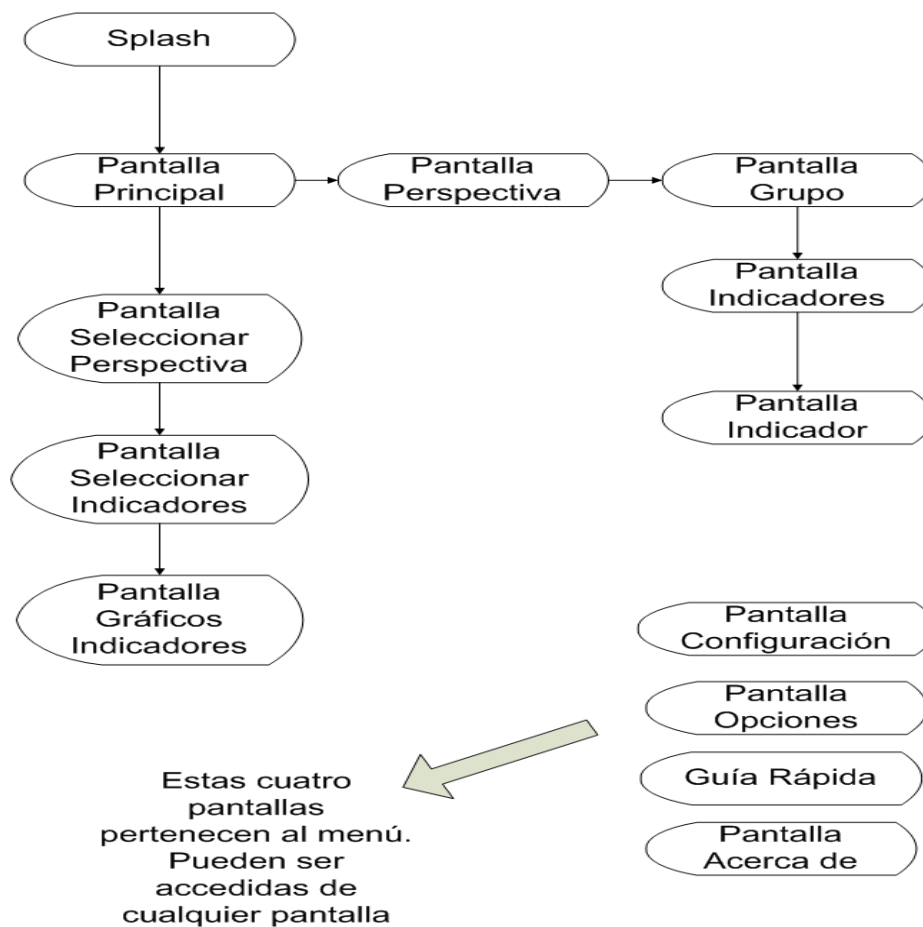


Diagrama 4. Storyboard de la Aplicación

- **Pantalla Principal**

La pantalla principal está formada por una barra de acción superior, un menú de navegación lateral y un listado de las organizaciones a las cuales se tiene acceso para su seguimiento. Además será precedida de un splash con el logo de la aplicación. Se observa la pantalla en las dos orientaciones posibles del móvil. (Ver Figura 6)

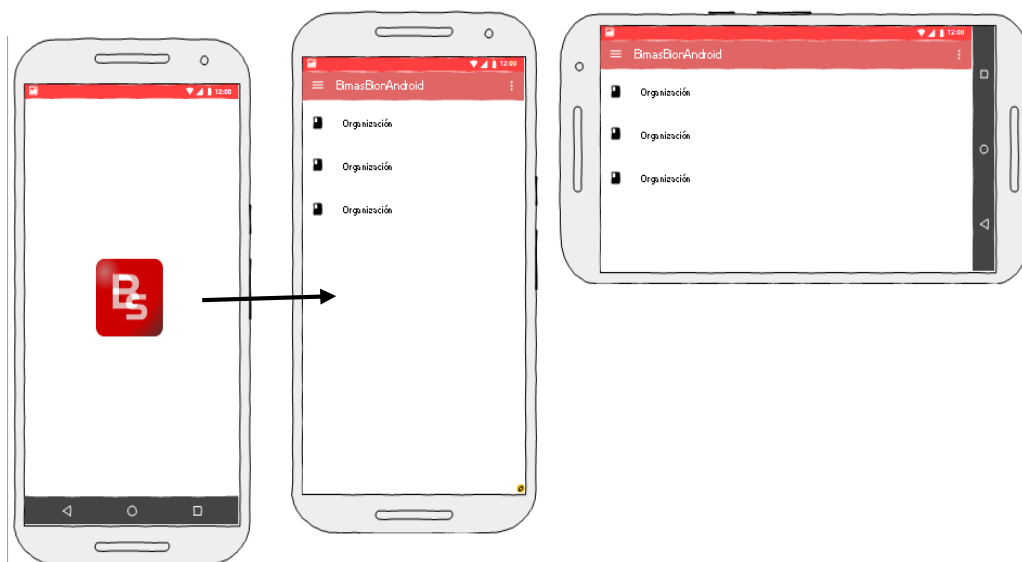


Figura 6: Prototipo de Pantalla: Pantalla Principal

En la tabla 4, se describe el StoryCard de prototipo de la pantalla principal. El storycard muestra una descripción del funcionamiento de la Pantalla Principal y de las fechas de su desarrollo.

Tabla 4:
STORYCARD DE LA PANTALLA PRINCIPAL

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
01	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Cuando el usuario abre la aplicación se visualiza el Splash, posteriormente se visualiza el listado de las organizaciones. Seleccionando cada una de ellas se pasa al listado de las perspectivas. Cuenta con un menú en el barra superior donde se puede acceder a la configuración, opciones, guía rápida y acerca de. También con un menú de navegación lateral.							
Fecha	Estado	Comentario					
10/01/18	Definido	Sin comentario					
15/01/18	Implementado	Sin comentario					
20/01/18	Hecho	Sin comentario					
10/02/18	Verificado	Sin comentario					

- **Menú de Navegación Lateral**

El menú de navegación lateral, muestra una lista con las opciones: actualizar, gráficos indicadores y compartir. Se observa la pantalla en las dos orientaciones posibles del móvil. (Ver Figura 7)

- Actualizar la base de datos, necesitándose estar conectado a la Wi-Fi, para consumir el servicio de la herramienta BIMAS Server.
- Visualizar los gráficos de barra de uno o dos indicadores de la misma organización y perspectiva que seleccionó el usuario. Donde se muestran los valores: plan y real.
- Compartir la información de todos los indicadores en un documento PDF dividido en tablas y por organización donde se muestran los planes, el valor real y el porcentaje.



Figura 7: Prototipo de Pantalla: Menú de Navegación Lateral

En la tabla 5, se describe el StoryCard del Prototipado de Pantalla: Menú de Navegación Lateral.

Tabla 5:
STORYCARD DEL MENÚ DE NAVEGACIÓN LATERAL

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
02	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
El usuario presiona el botón de navegación, y aparece un menú lateral con las opciones de actualizar, gráficos indicadores y compartir.							
Fecha	Estado	Comentario					
12/01/18	Definido	Sin comentario					
16/01/18	Implementado	Sin comentario					
21/01/18	Hecho	Sin comentario					
10/02/18	Verificado	Sin comentario					

- **Pantalla Perspectiva**

La pantalla perspectiva cuenta con una lista de todas las perspectivas existentes en la organización seleccionada en la pantalla principal. También posee el mismo menú en la barra superior que la pantalla principal. Se observa la pantalla en las dos orientaciones posibles del móvil. (Ver Figura 8)



Figura 8: Prototipo de Pantalla: Pantalla Perspectiva

En la tabla 6, se describe el StoryCard de prototipo de la pantalla perspectiva. El storycard muestra una descripción del funcionamiento de la Pantalla Perspectiva.

Tabla 6:
STORYCARD DE LA PANTALLA PERSPECTIVA

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
03	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Se visualiza el listado de perspectivas existentes en la organización seleccionada en la pantalla principal. Donde al presionar una se pasa a la pantalla de grupo. El título de la barra superior es el nombre de la organización.							
Fecha	Estado	Comentario					
16/01/18	Definido	Sin comentario					
18/01/18	Implementado	Sin comentario					
23/01/18	Hecho	Sin comentario					
10/02/18	Verificado	Sin comentario					

- **Pantalla Grupo**

La Pantalla grupo muestra un listado con los grupos existentes en la perspectiva seleccionada en la pantalla anterior. También cuenta con el menú de la barra superior. Se observa la pantalla en las dos orientaciones posibles del móvil. (Ver Figura 9)

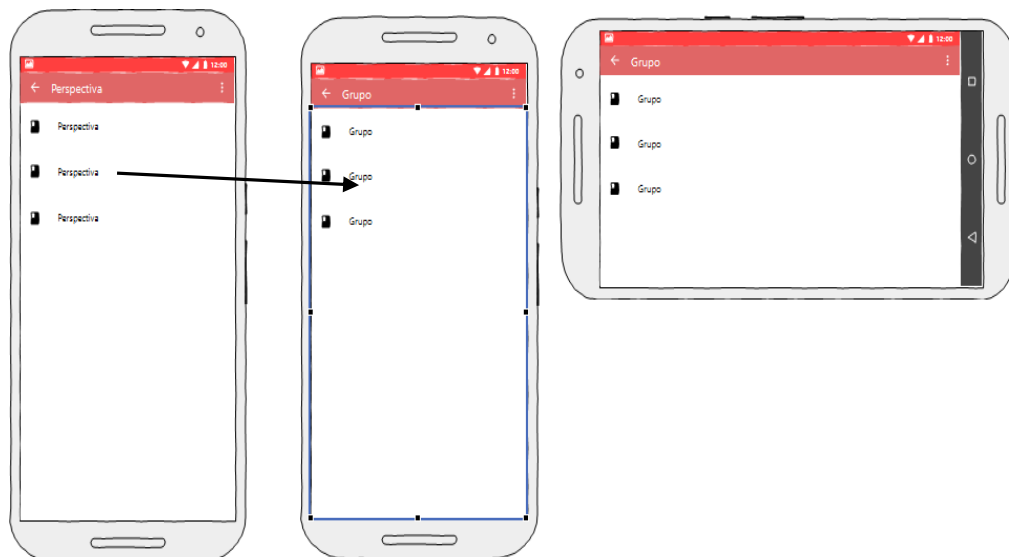


Figura 9: Prototipo de Pantalla: Pantalla Grupo

En la tabla 7, se describe el StoryCard de prototipo de la pantalla grupo. El storycard muestra una descripción del funcionamiento de la Pantalla Grupo.

Tabla 7:

STORYCARD DE LA PANTALLA GRUPO

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
04	Nuevo	Fácil	Fácil			Baja	
	Fijo	Moderado	Moderado			Media	
	Mejora	Duro	Duro			Alta	
Descripción							
Se visualiza el listado de grupos existentes en la perspectiva seleccionada en la pantalla anterior. Donde al presionar un grupo se pasa a la pantalla de indicadores. El título de la barra superior es el nombre de la perspectiva.							
Fecha		Estado		Comentario			
18/01/18		Definido		Sin comentario			
20/01/18		Implementado		Sin comentario			
23/01/18		Hecho		Sin comentario			
10/02/18		Verificado		Sin comentario			

- Pantalla Indicadores**

La pantalla indicadores muestra un listado con todos los indicadores pertenecientes al grupo seleccionado en la pantalla anterior, siendo este el título de la barra superior en la cual también tenemos el menú descrito

en las pantallas anteriores. Se observa la pantalla en las dos orientaciones posibles del móvil. (Ver Figura 10)

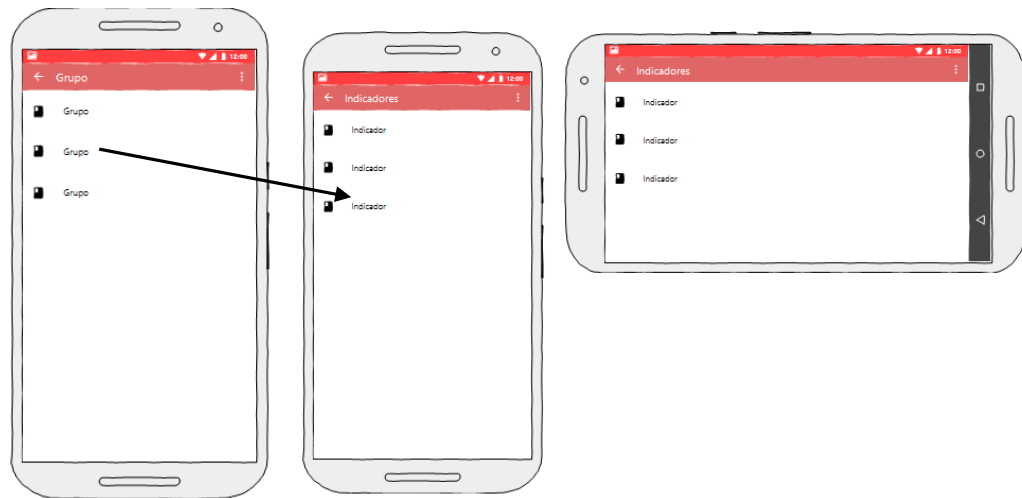


Figura 10: Prototipo de Pantalla: Pantalla Indicadores

En la tabla 8, se describe el StoryCard de prototipo de la pantalla indicadores. El storycard muestra una descripción del funcionamiento de la Pantalla Indicadores.

Tabla 8:

STORYCARD DE LA PANTALLA INDICADORES

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
05	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Se visualiza el listado de indicadores existentes en el grupo seleccionado en la pantalla anterior. Donde al presionar uno, se pasa a la pantalla del indicador, donde se muestra la información del mismo. El título de la barra superior es el nombre del grupo.							
Fecha	Estado	Comentario					
20/01/18	Definido	Sin comentario					
21/01/18	Implementado	Sin comentario					
23/01/18	Hecho	Sin comentario					
10/02/18	Verificado	Sin comentario					

- Pantalla Indicador**

La pantalla indicador nos muestra la información del indicador seleccionado en la pantalla indicadores. También cuenta con el menú de

la barra superior. Se observa la pantalla en las dos orientaciones posibles del móvil. (Ver Figura 11)

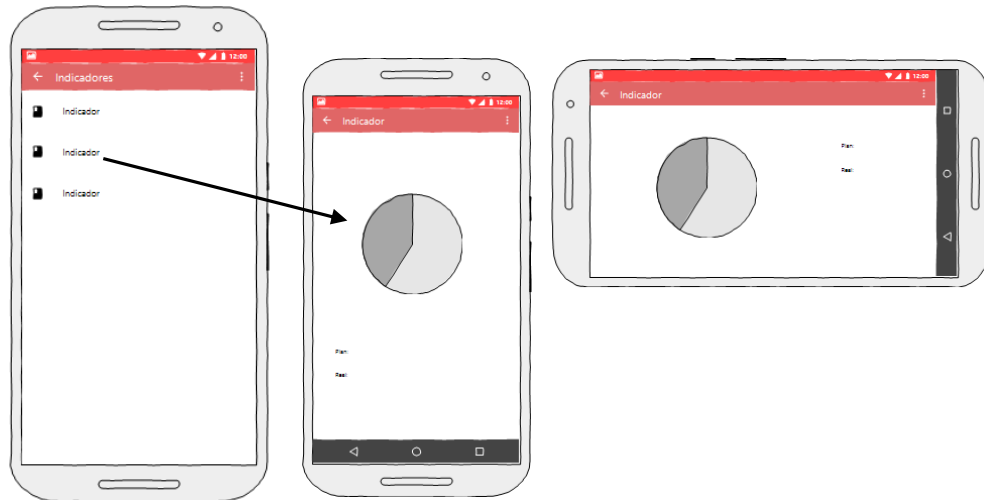


Figura 11: Prototipo de Pantalla: Pantalla Indicador

En la tabla 9, se describe el StoryCard de prototipo de la pantalla indicador. El storycard muestra una descripción del funcionamiento de la Pantalla indicador.

Tabla 9:
STORYCARD DE LA PANTALLA INDICADOR

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
06	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Se visualiza un gráfico donde se muestra el plan y el cumplimiento del indicador seleccionado, en el centro del gráfico está el porcentaje que representa lo alcanzado de lo propuesto. El título de la barra superior es el nombre del indicador.							
Fecha	Estado	Comentario					
21/01/18	Definido	Sin comentario					
22/01/18	Implementado	Sin comentario					
25/01/18	Hecho	Sin comentario					
12/02/18	Verificado	Sin comentario					

- **Pantalla Seleccionar Perspectiva**

Esta pantalla muestra una lista expandible donde se seleccionara una organización, luego una perspectiva de la cual se pasa a los indicadores de la misma en otra pantalla. También cuenta con el menú de la barra superior. Se observa la pantalla en las dos orientaciones posibles del móvil. (VerFigura 12)

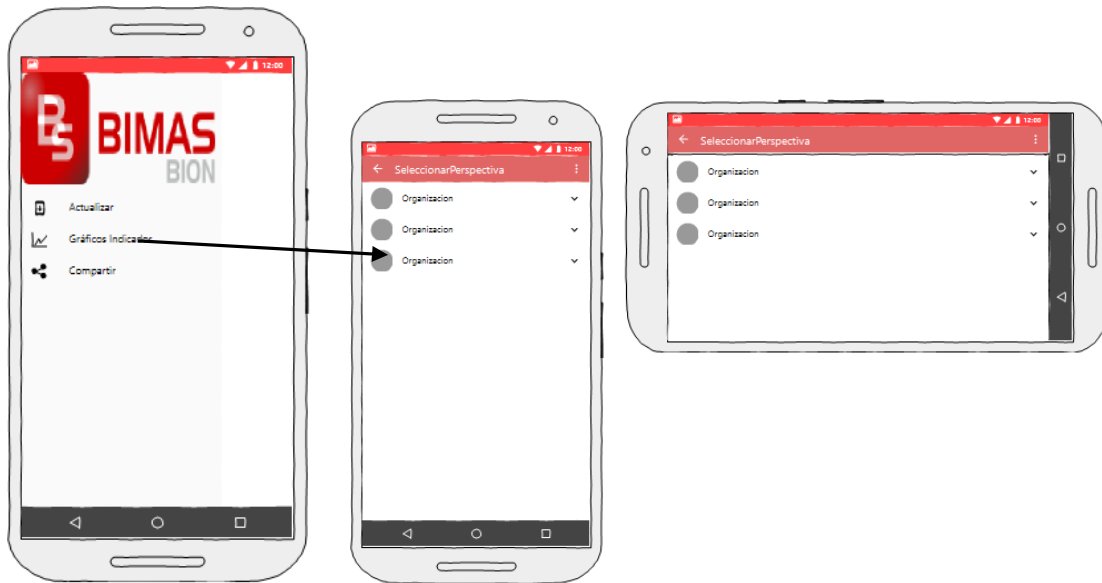


Figura 12: Prototipo de Pantalla: Pantalla Seleccionar Perspectiva

En la tabla 10, se describe el StoryCard de prototipo de la pantalla seleccionar perspectiva. El storycard muestra una descripción del funcionamiento de la Pantalla seleccionar perspectiva.

Tabla 10:
STORYCARD DE LA PANTALLA SELECCIONAR PERSPECTIVA

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
07	Nuevo	Fácil	Fácil			Baja	
	Fijo	Moderado	Moderado			Media	
	Mejora	Duro	Duro			Alta	
Descripción							
Se visualiza una lista desplegable que contiene todas las organizaciones. Al desplegar una se muestran sus perspectivas. Si selecciona una perspectiva se muestran sus indicadores en una nueva pantalla.							
Fecha	Estado	Comentario					
21/01/18	Definido	Sin comentario					
22/01/18	Implementado	Sin comentario					
25/01/18	Hecho	Sin comentario					
12/02/18	Verificado	Sin comentario					

- **Pantalla Seleccionar**

Esta pantalla tiene un listado con los indicadores de la perspectiva seleccionada en la pantalla anterior. También cuenta con el menú de la barra superior. Se observa la pantalla en las dos orientaciones posibles del móvil. (Ver Figura 13)

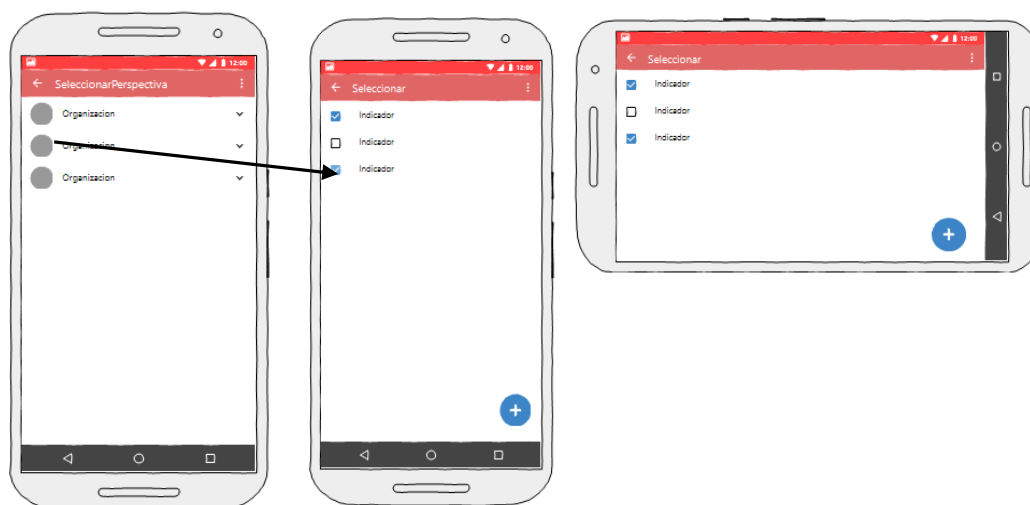


Figura 13: Prototipo de Pantalla: Pantalla Seleccionar

En la tabla 11, se describe el StoryCard de prototipo de la pantalla seleccionar. El storycard muestra una descripción del funcionamiento de la Pantalla seleccionar.

Tabla 11:
STORYCARD DE LA PANTALLA SELECCIONAR

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
08	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Se visualiza una lista que contiene indicadores y cada uno tiene un cuadro para señalarlo, al tener uno marcado aparece en la esquina derecha inferior un botón flotante y de ser dos se inhabilitan los demás cuadros. Al presionar el botón pasamos a otra pantalla con los gráficos de barra de dichos indicadores. Si queremos marcar otros indicadores primero es necesario desmarcar los seleccionados para que se habiliten los demás.							
Fecha	Estado	Comentario					
21/01/18	Definido	Sin comentario					
22/01/18	Implementado	Sin comentario					
25/01/18	Hecho	Sin comentario					
15/02/18	Verificado	Sin comentario					

- **Pantalla Gráfico Indicadores**

Esta pantalla muestra los gráficos de barra de los indicadores seleccionados en la pantalla anterior. También cuenta con el menú de la barra superior. Se observa la pantalla en las dos orientaciones posibles del móvil. (Ver Figura 14)



Figura 14: Prototipo de Pantalla: Pantalla Gráfico Indicadores

En la tabla 12, se describe el StoryCard de prototipo de la pantalla Gráfico Indicadores. El storycard muestra una descripción del funcionamiento de la Pantalla Gráfico Indicadores.

Tabla 12:

STORYCARD DE LA PANTALLA GRÁFICO INDICADORES

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
09	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Se visualizan los gráficos de barra de los indicadores seleccionados en la pantalla anterior. Estos gráficos muestran los valores del plan y de lo realmente alcanzado. Se ajusta cuando el móvil cambia de orientación.							
Fecha	Estado	Comentario					
25/01/18	Definido	Sin comentario					
28/01/18	Implementado	Sin comentario					
01/02/18	Hecho	Sin comentario					
16/02/18	Verificado	Sin comentario					

- **Pantalla Opciones**

Esta pantalla es de opciones de la aplicación. También cuenta con el menú de la barra superior. Se observa la pantalla en las dos orientaciones posibles del móvil. (Ver Figura 15)

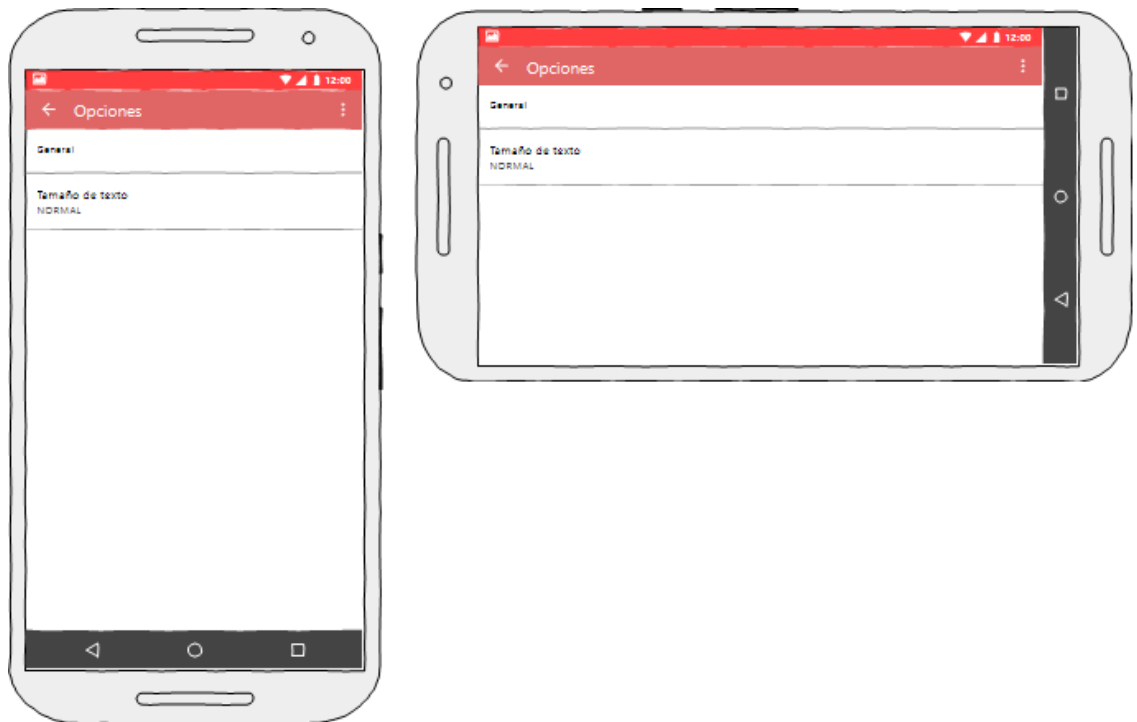


Figura 15: Prototipo de Pantalla: Pantalla Opciones

En la tabla 13, se describe el StoryCard de prototipo de la pantalla Opciones. El storycard muestra una descripción del funcionamiento de la Pantalla Opciones.

Tabla 13:
STORYCARD DE LA PANTALLA OPCIONES

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
10	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Permite al usuario cambiar el tamaño de letra de los textos. Cuando presionamos en la opción tamaño del texto nos aparece un diálogo con tres opciones posibles.							
Fecha	Estado	Comentario					
3/02/18	Definido	Sin comentario					
5/02/18	Implementado	Sin comentario					
6/02/18	Hecho	Sin comentario					
16/02/18	Verificado	Sin comentario					

- **Pantalla Acerca de**

Esta pantalla muestra información general acerca del producto. Se observa la pantalla en las dos orientaciones posibles del móvil. (Ver Figura 16)

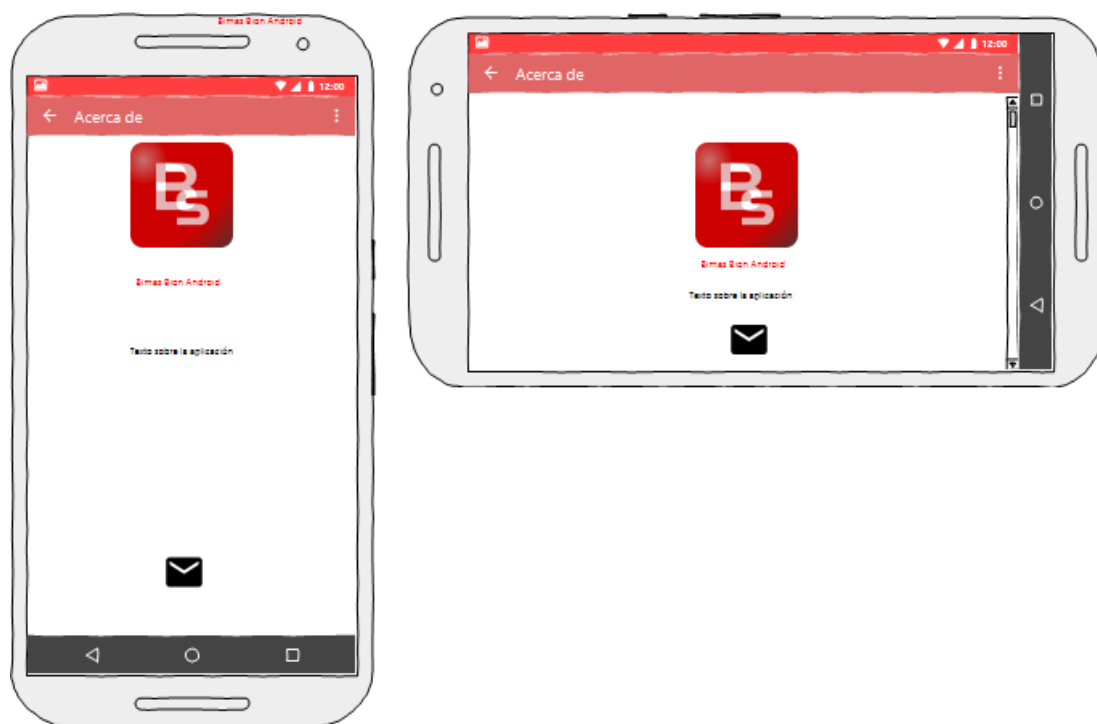


Figura 16: Prototipo de Pantalla: Pantalla Acerca de

En la tabla 14, se describe el StoryCard de prototipo de la pantalla Acerca de. El storycard muestra una descripción del funcionamiento de la Pantalla Acerca de.

Tabla 14:
STORYCARD DE LA PANTALLA ACERCA DE

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
11	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
Esta pantalla muestra información general de la aplicación, como la versión y los derechos reservados. También posee un icono de correo que al presionarlo permite abrir la aplicación determinada como favorita de correo en el teléfono, en el nuevo mensaje aparece en el destinatario la dirección electrónica de Desoft VC y un asunto predeterminado.							
Fecha	Estado	Comentario					
5/02/18	Definido	Sin comentario					
8/02/18	Implementado	Sin comentario					
10/02/18	Hecho	Sin comentario					
18/02/18	Verificado	Sin comentario					

- **Pantalla Configuración**

Esta pantalla es donde se introducen los datos de configuración para establecer la conexión con el servicio. Se observa la pantalla en las dos orientaciones posibles del móvil. (Ver Figura 17)



Figura 17: Prototipo de Pantalla: Pantalla Configuración

En la tabla 15, se describe el StoryCard de prototipo de la pantalla Configuración. El storycard muestra una descripción del funcionamiento de la Pantalla Configuración.

Tabla 15:
STORYCARD DE LA PANTALLA CONFIGURACIÓN

Número /id	Tipo	Dificultad		Esfuerzo		Prioridad	Nota
		Antes	Después	Estimado	Gastado		
12	Nuevo Fijo Mejora	Fácil Moderado Duro	Fácil Moderado Duro			Baja Media Alta	
Descripción							
En esta pantalla es donde se introducen los parámetros necesarios para establecer la conexión con el servicio de Bimas Server. Solo después de configurar la conexión es posible actualizar la base de datos de la aplicación							
Fecha	Estado	Comentario					
10/02/18	Definido	Sin comentario					
13/02/18	Implementado	Sin comentario					
16/02/18	Hecho	Sin comentario					
20/02/18	Verificado	Sin comentario					

2.3 Tercera Fase: Codificación

En esta fase se realiza el desarrollo de las funcionalidades planteadas en el diseño mediante un lenguaje de programación. Se utiliza java porque el Sistema Operativo Android esta basado en el mismo; además se plantean los estándares de codificación y la estructuración de la aplicación.

2.3.1 PRODUCCIÓN Y ESTABILIZACIÓN

En la producción se realiza la codificación de los procesos diseñados y la posterior integración entre cada uno de ellos para obtener el total funcionamiento de la aplicación.

Estándares de Codificación:

- 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 separar cada una de ellas.

Estructura de Directorios:

En el diagrama 5, se describe la estructura de la aplicación mostrando los directorios mas importantes.

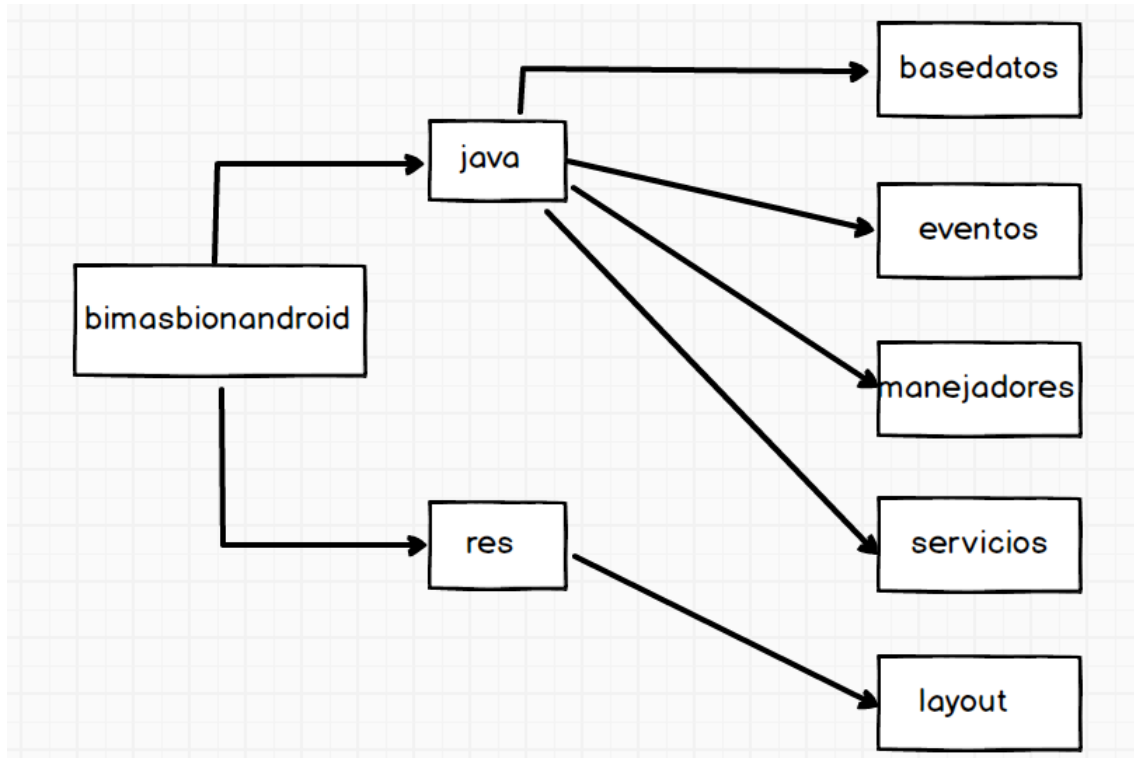


Diagrama 5. Estructura de directorios más significativos

El directorio eventos posee todos los eventos utilizados en la aplicación para controlar los procesos que se realizan dentro de la misma. Son de gran importancia para el directorio manejadores, el cual se encarga de desarrollar la lógica de la aplicación en segundo plano y mediante eventos se controla la terminación de métodos o invocación de objetos, siendo también donde se procesa la información obtenida del servicio de Bimas Server. El directorio servicios posee la clase que se encarga de brindar la posibilidad de salir de la aplicación mientras se actualiza la información de la base de datos.

Codificación:

La codificación se realizó basándose en cada uno de los storycards realizados en la fase anterior. A continuación se muestran las clases y métodos más importantes.

Pantalla Principal y Menú de Navegación Lateral:

Para la pantalla principal se realizó la construcción del Splash, se implementó la conexión al servicio, el almacenamiento de los indicadores en la base de datos interna y la funcionalidad de compartir.

- **Construcción del Splash**

Para la creación del Splash se creó una actividad llamada SplashActivity y se utilizaron varios métodos como los siguientes:

setRequestedOrientation(): Este método sirve para fijar una orientación al dispositivo, en este caso ActivityInfo.SCREEN_ORIENTATION_PORTRAIT se utiliza para el modo retrato; permite que el dispositivo no cambie de orientación mientras se carga el splash y así éste no se reinicie con algún movimiento del dispositivo. (Ver figura 1 de los anexos)

Thread: Se utilizó esta clase para que el splash se ejecutara durante dos segundos.

- **Obtención de los datos a través del servicio API RESTful**

Para esto es necesario desplegar el menú lateral de navegación y presionar la opción Actualizar Datos, al realizar esta acción se ejecuta el método `onNavigationItemSelectedListener()` (Ver figura 2 de los anexos). Aquí se verifica que estén guardados los parámetros de conexión con el servicio, de estarlo se le envía un mensaje con dicha información a la clase `BackgroundHandler`. (Ver figura 3 de los anexos) Esta es la encargada de establecer la conexión en un segundo plano y con la utilización de un servicio Android permite salir de la aplicación mientras se actualiza la información. (Ver figura 4 de los anexos)

En esta sección de código se envía un mensaje con la opción conectar y los datos necesarios para dicho proceso.

```
BackgroundHandler handler =  
BackgroundHandler.getInstance(getApplicationContext());  
Message message = Message.obtain(handler, BackgroundHandler.CONECTAR);  
message.getData().putStringArray("data", new String[]{host, puerto,  
user, pass});  
message.sendToTarget();
```

Aquí se hace la llamada al servicio Android encargado de almacenar en la base de datos la información obtenida del API RESTful.

```
Intent intent = new Intent(context(getApplicationContext()),  
ActualizarDatosService.class);  
Bundle bundle = new Bundle();  
bundle.putString("Json", object_json);
```

```
intent.putExtras(bundle);
intent.setAction("cu.desoftvc.bimas.bimasbionandroid.ACTUALIZAR_DATOS"
);
context.startService(intent);
```

- **Funcionalidad compartir**

Para poder compartir la información de la aplicación presionamos en la opción compartir del menú lateral de navegación. Esto hace una llamada a la clase BackgroundHandler la cual es la encargada de salvar los indicadores en un archivo PDF (Ver figura 5 de los anexos). Para luego compartirlo por todas las opciones posibles como adjunto de correo, entre otras.

En esta sección de código se observa cómo se comparte el documento PDF. Y se puede observar el uso de la biblioteca eventbus al utilizar la anotación Subscribe.

```
@Subscribe(threadMode = ThreadMode.MAIN)
public void salvadoPdf(VisualExportarPdfEvent event) {
    Toast.makeText(this, "Salvado Pdf", Toast.LENGTH_SHORT).show();
    Intent shareIntent = new Intent();
    Uri uri = Uri.fromFile(event.getFile());
    shareIntent.setAction(Intent.ACTION_SEND);
    shareIntent.putExtra(Intent.EXTRA_STREAM, uri);
    shareIntent.setType("application/pdf");
    startActivity(Intent.createChooser(shareIntent, "Compartir"));
}
```

Otras clases y bibliotecas utilizadas:

JsonManipulador:

Esta clase es llamada por el servicio Android y es la encargada de analizar y procesar el objeto JSON obtenido del servicio API RESTful. Realiza la lógica de inserción en la base de datos creando hilos en base a la cantidad de procesadores que posea el dispositivo móvil y repartiendo los datos a guardar de una manera equitativa.

```
handlerThreads = new
HandlerThread[Runtime.getRuntime().availableProcessors() - 1];
JsonHandler[] jsonHandlers = new JsonHandler[handlerThreads.length];
int cant = orgArray.length() / handlerThreads.length;
int mod = (orgArray.length() < handlerThreads.length)? 0 :
orgArray.length() % handlerThreads.length;
cant = Math.max(1, cant);
```

- android-async-http-1.4.9:

Biblioteca utilizada para establecer la conexión con el servicio de Bimas Server.

- itextg-5.5.9

Biblioteca de Google para la construcción de documentos, se utilizó en la funcionalidad de compartir, para el diseño del PDF con los indicadores organizados en tablas por cada una de las organizaciones.

- eventbus

Biblioteca que permite manejar eventos a nivel global dentro de la aplicación con gran facilidad. Se utiliza mediante la suscripción de los métodos que esperan eventos para su ejecución. Estos eventos pueden ser definidos por el desarrollador.

- MPAndroidChart

Esta Biblioteca es la que permite la visualización de los indicadores mediante gráficos.

Conclusiones del Capítulo:

En este capítulo se diseñaron los principales artefactos de la metodología Mobile-D. Estos permitieron:

1. Implementar cada uno de los storyboard confeccionados junto al usuario.
2. Desarrollar cada pantalla sin errores, ya que Mobile-D propone durante la fase de producción realizar varias pruebas durante cada iteración.
3. Hacer un correcto diseño de la base de datos así como del diagrama de clases del dominio lo que facilita y evita errores durante la fase de producción de Mobile-D.

CAPÍTULO 3. CUARTA FASE: PRUEBAS.

En este capítulo se describen los resultados de las distintas pruebas realizadas a la aplicación desarrollada, con el fin de comprobar si su funcionamiento es el esperado, tanto en varios dispositivos con diferentes versiones del Sistema Operativo Android así como en diferentes densidades y tamaños de pantalla comprobando si se afecta la visualización de los indicadores y los gráficos. También se documenta un pequeño manual de usuario que explica las principales funcionalidades de la aplicación.

3.1 PRUEBAS DEL SISTEMA Y ARREGLOS

Luego de haber terminado con la codificación se deben realizar las pruebas que comprueben que es una aplicación de calidad y que cumple con los requerimientos por los que fue creada.

Las pruebas que la metodología Mobile-D establece para una aplicación son la comprobación de requerimientos, los resultados en tiempos de respuestas y tiempos de accesos, además de una lista de dispositivos en dónde se ha comprobado el adecuado funcionamiento de la aplicación.

3.1.1 Pruebas de Interfaz de Usuario

Estas pruebas consisten en comprobar si la interfaz de la aplicación se ve correctamente en dispositivos de todas las resoluciones disponibles.

Para realizarlas se ha ejecutado la aplicación en diferentes dispositivos de diferentes tamaños y resoluciones y se ha comprobado que no había una sobrecarga de campos en la pantalla, ni campos que se desplacen por falta de espacio, etc.

Descripción de dispositivos móviles utilizados:

Modelo	Huawei Ascend Y530
Tipo de dispositivo	Teléfono
Versión de Android	4.3(Jelly Bean)
Resolución de pantalla	480x854
Pantalla en diagonal	4.5 pulgadas

Modelo	LO-T1073
Tipo de dispositivo	Tableta
Versión de Android	4.2.2(Jelly Bean)
Resolución de pantalla	800x480
Pantalla en diagonal	6.7 pulgadas

Pantalla Indicador

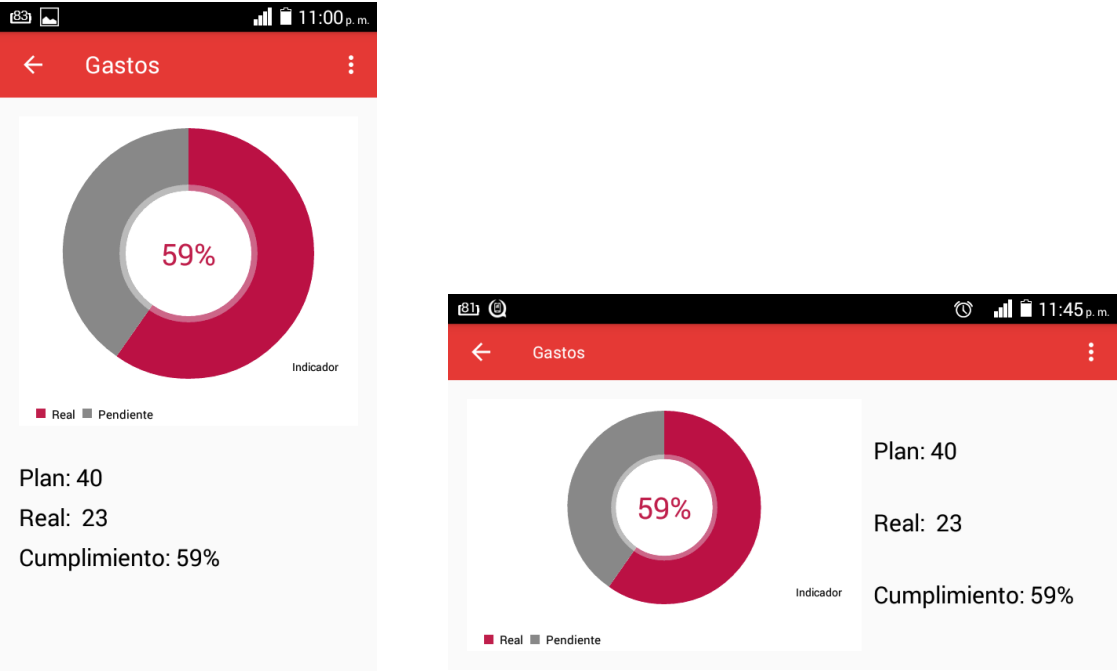


Figura 18. Pantalla Indicador sobre Teléfono Huawei

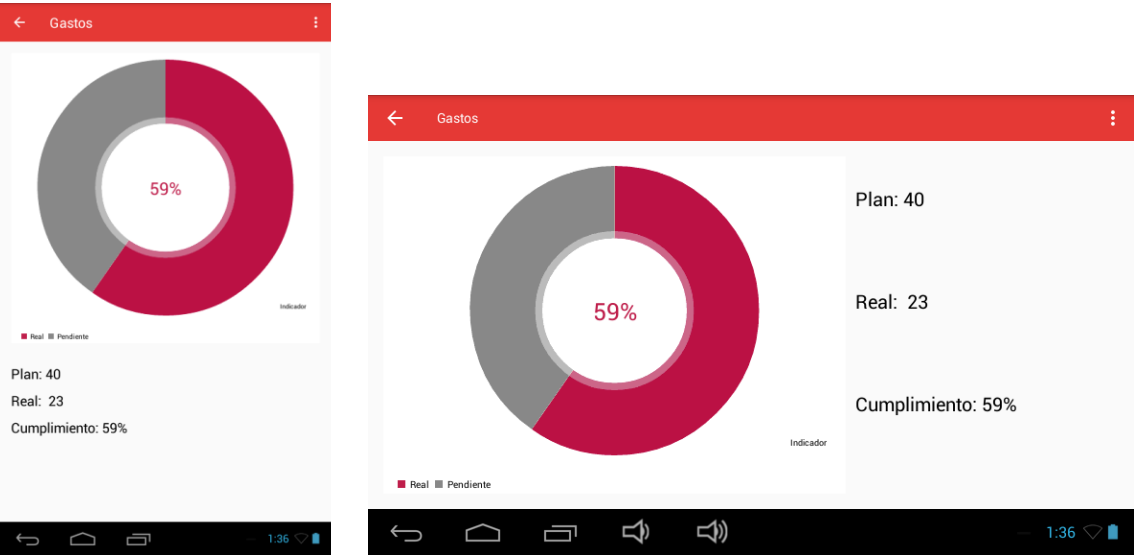


Figura 19. Pantalla Indicador sobre Tableta LO-T1073

Pantalla Gráficos Indicadores:



Figura 20. Pantalla Gráficos Indicadores sobre Teléfono Huawei



Figura 21. Pantalla Gráficos Indicadores sobre Teléfono Huawei

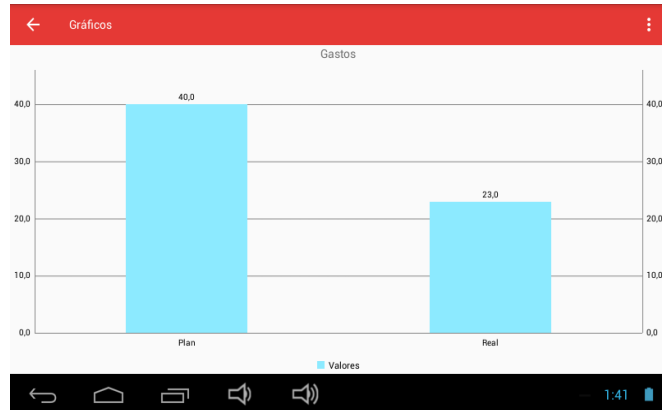
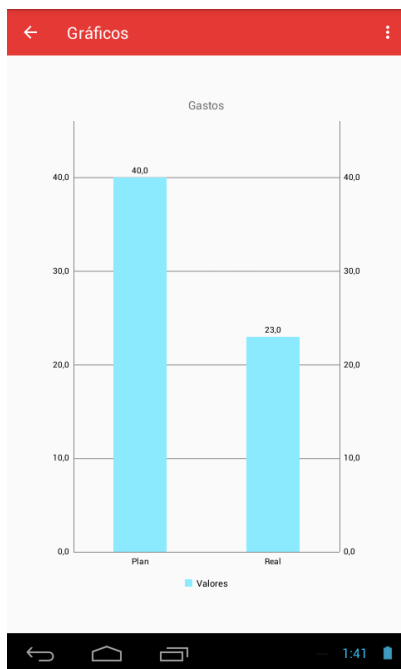


Figura 22. Pantalla Gráficos Indicadores sobre Tableta LO-T1073

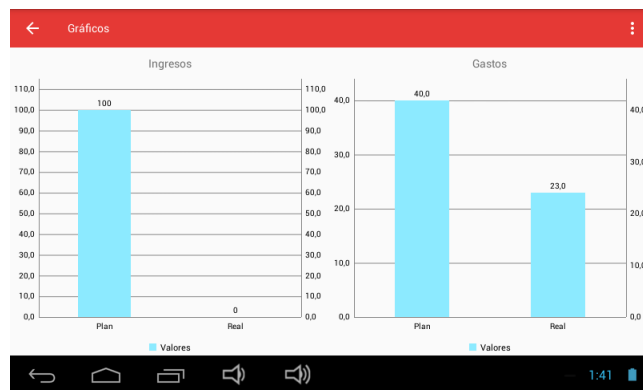
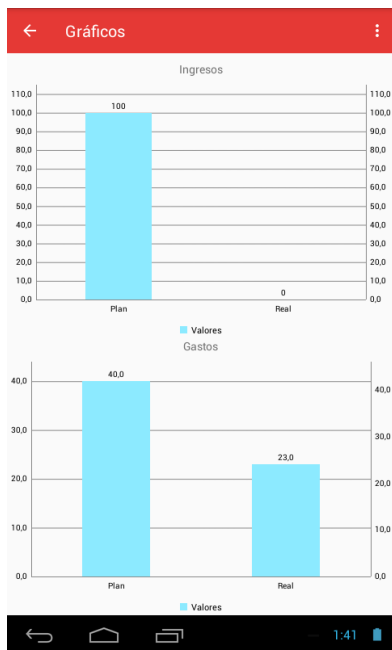


Figura 23. Pantalla Gráficos Indicadores sobre Tableta LO-T1073

Pantalla Configuración

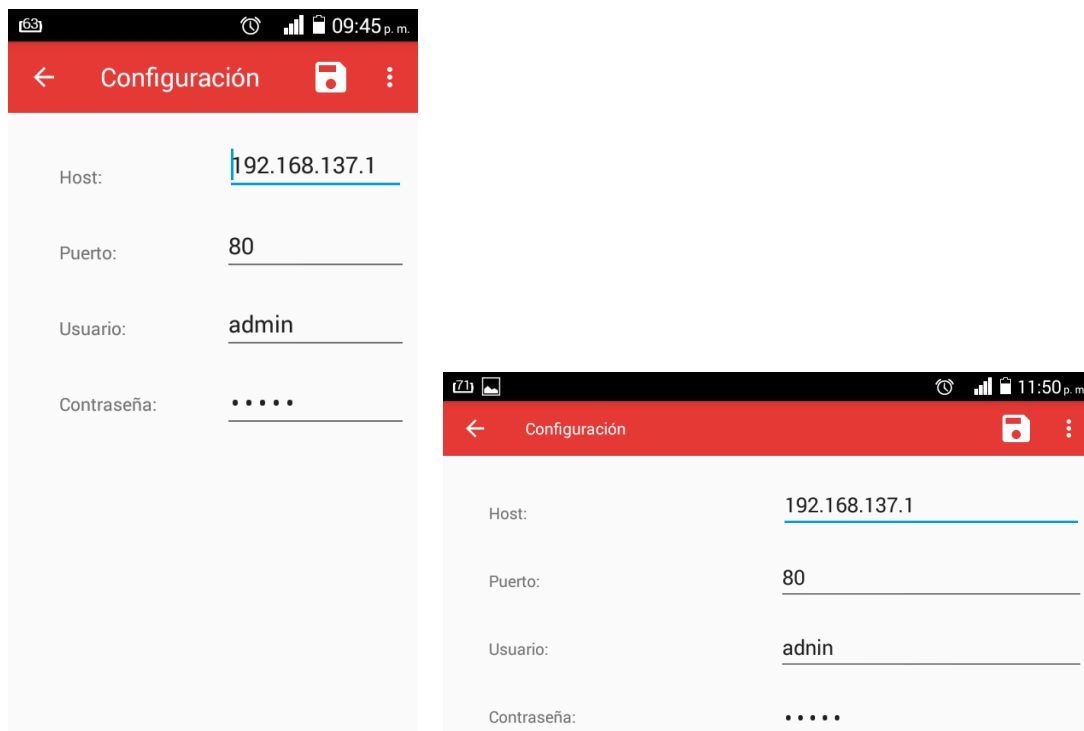


Figura 24. Pantalla Configuración sobre Teléfono Huawei

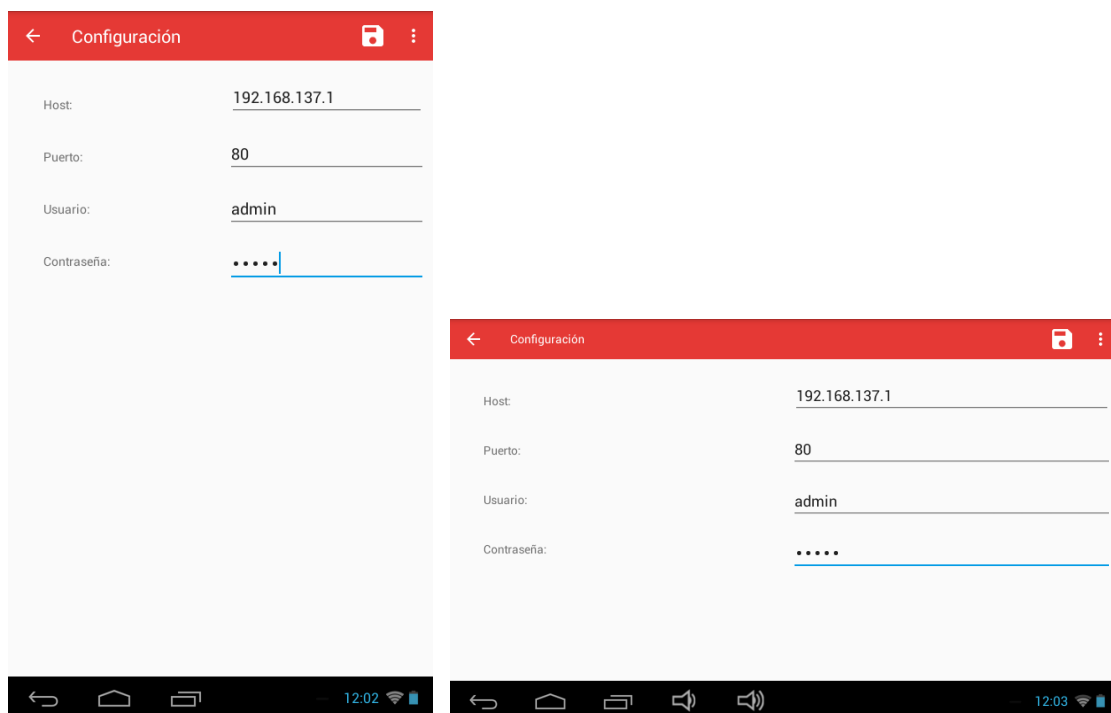


Figura 25. Pantalla Configuración sobre Tableta LO-T1073

3.1.2 Pruebas con Monkop

Es una herramienta online para hacer pruebas sobre una aplicación Android. Monkop hace un reporte pormenorizado de una aplicación con solo pasarle su apk. Ingresa datos aleatorios en todos los componentes que encuentre, selecciona algún elemento de una lista, hace taps sobre los botones que encuentra, etc. De esta manera va chequeando el comportamiento de la aplicación, y obteniendo estadísticas que luego muestra en un reporte.

Como resultado final muestra un reporte con información sobre el uso de la batería, sobre el rendimiento, bibliotecas utilizadas por la aplicación, bibliotecas deprecadas, actividades de la aplicación, controles en cada pantalla, servicios activos, uso de la CPU, errores y mucho más. Algo realmente útil para evitar hacer pruebas manuales con muchos datos y combinaciones de datos sobre cada pantalla de la aplicación. Pero también muestra información de utilidad para el desarrollador, como por ejemplo que bibliotecas está usando y que están siendo deprecadas, el uso de la CPU, etc.

Dispositivos	Sistema Operativo	Tamaño/ Densidad	Ejecución	Tiempo de Inicio
Google Nexus 5	5.1 (API:22)	normal/xxhdpi	10m04s.726	.581
Samsung Galaxy Note 3 (SM-N900V)	4.4 (API:19)	normal/xxhdpi	10m03s.806	.870
Google Nexus 10	5.1 (API:22)	xlarge/xhdpi	10m04s.126	.739
Moto G4 Plus(XT1644)	7.0 (API:24)	normal/xxhdpi	10m01s.423	.719

Tabla 4. Ejecución de la aplicación con Monkop

Dispositivos	Errores de respuesta	Errores de Bloqueo
Google Nexus 5	0	0
Samsung Galaxy Note 3 (SM-N900V)	0	0
Google Nexus 10	0	0
Moto G4 Plus(XT1644)	0	0

Tabla 5. Cantidad de errores de respuesta y de bloqueo

Resultados de las pruebas con Monkop

Durante las pruebas de Monkop la aplicación funcionó correctamente en los dispositivos seleccionados. Utilizando pocos recursos de los móviles sin la ocurrencia de errores de respuesta o de bloqueo. El tiempo de inicio de la aplicación fue rápido y no mostró errores en los elementos de las pantallas.

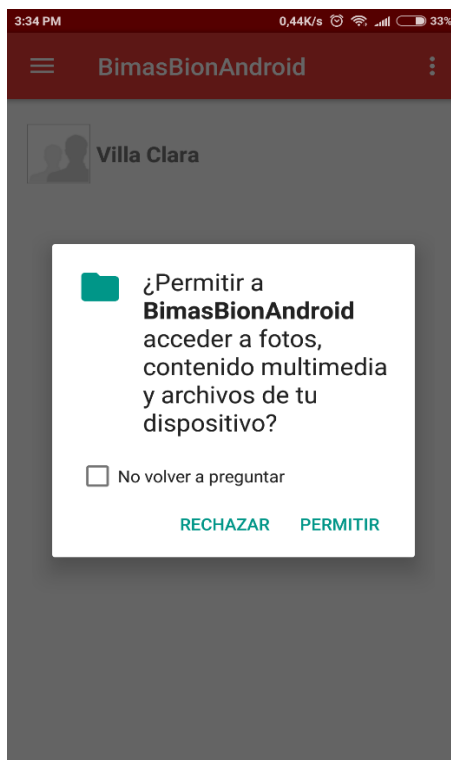
3.1.3 Manual de Usuario



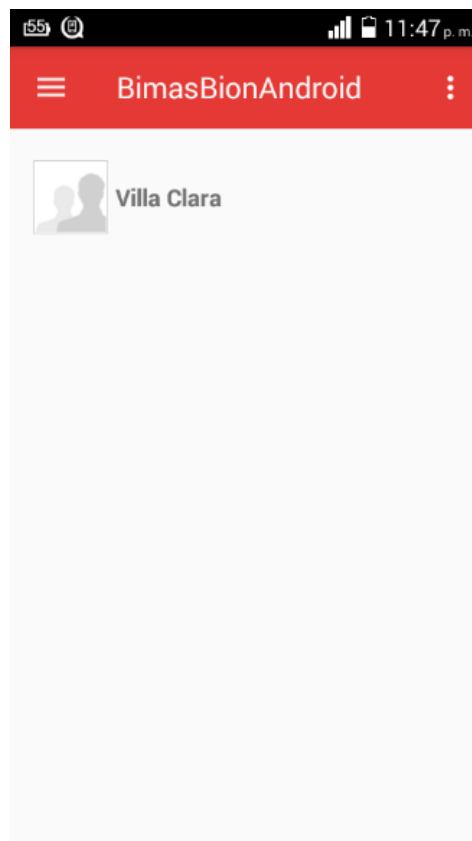
BimasBionAndroid

Requisitos del Sistema:

1. Soporte: La versión del sistema operativo Android debe ser 4.0.2 (Jelly Bean) o superior.
2. Seguridad: A partir de la versión 6.0 (Marshmallow) del sistema operativo Android el usuario debe otorgar manualmente los permisos de escritura y lectura de la memoria interna.



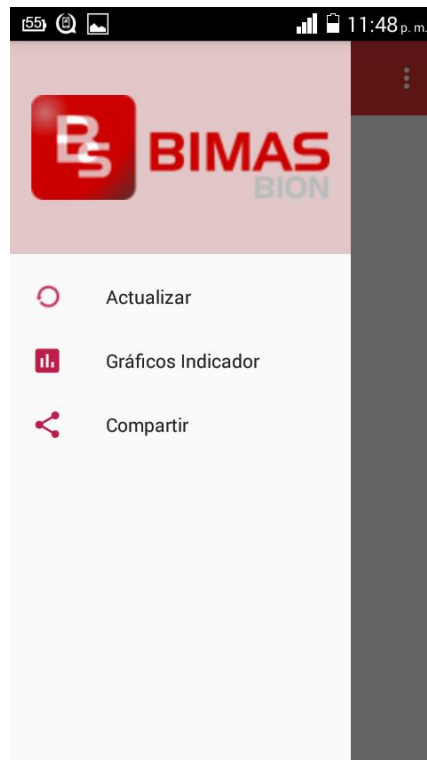
Interfaz Principal de la aplicación



Menú

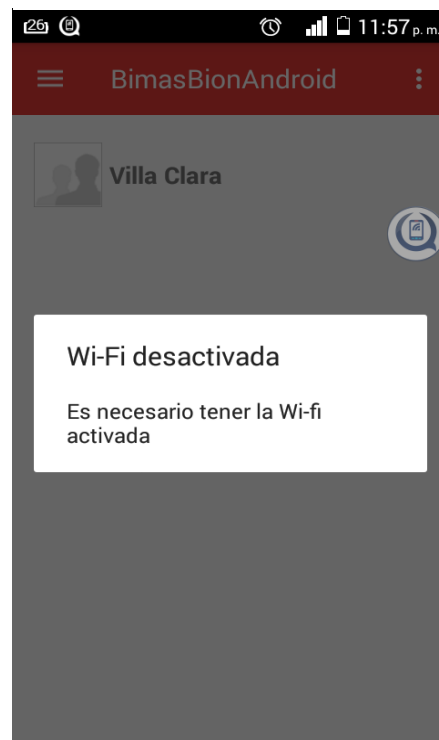
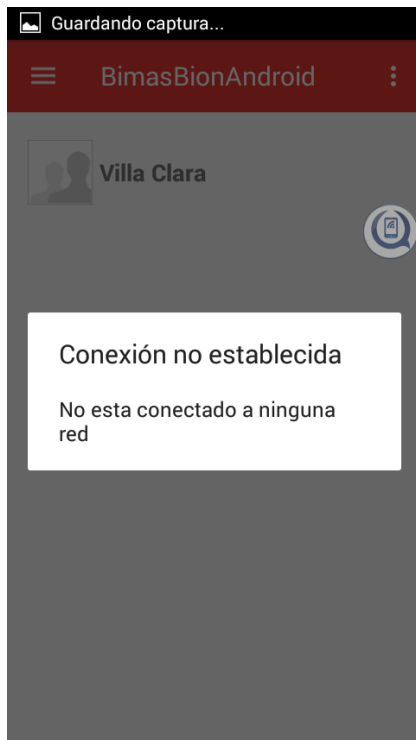


Presione el menú de navegación lateral o deslice el dedo hacia la derecha para que aparezca. El cual brinda tres funcionalidades.

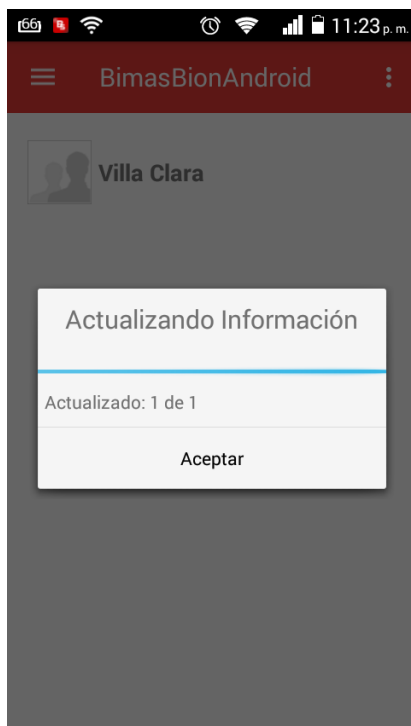


Si se presiona la opción Actualizar esta nos permite sincronizar con el servidor de Bimas Server y actualizar los datos pertenecientes a los indicadores definidos en la organización.

Es necesario para esta acción tener la Wi-Fi del dispositivo encendida y estar conectado a la red de la empresa sino se muestran los siguientes mensajes:

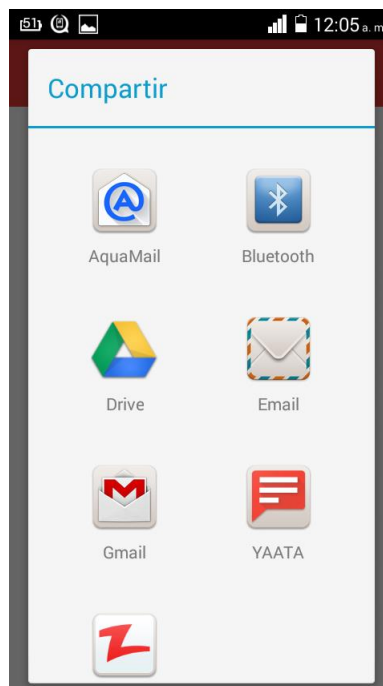


Durante el proceso de sincronización y actualización de la información se mostrará un mensaje con las acciones en curso así como una barra de progreso. La aplicación cuenta con una notificación para, si se quiere salir de ella mientras se actualiza, saber cuándo termina el proceso.

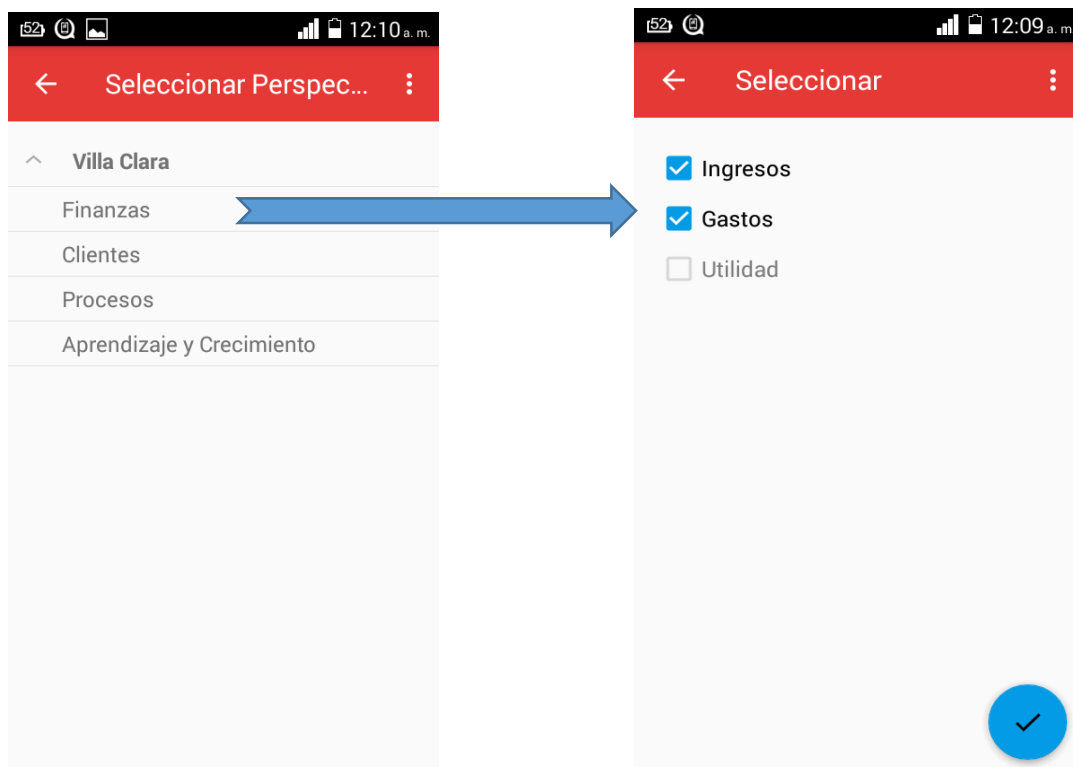


Notificación de la aplicación

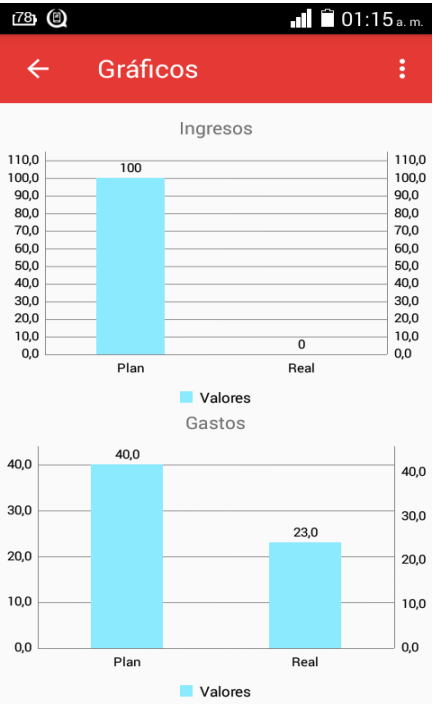
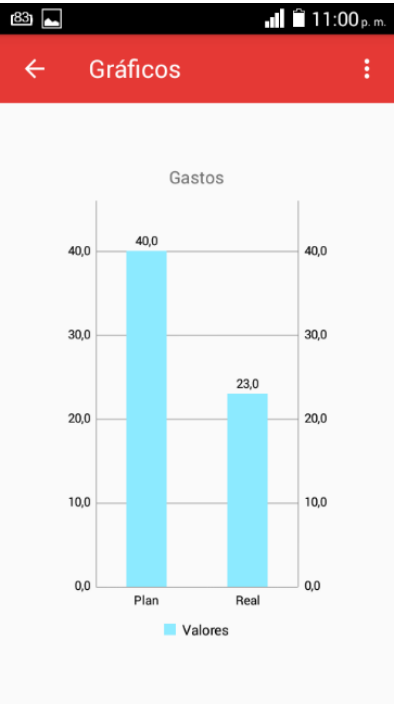
Otra opción del menú lateral es la de compartir, esta nos permite crear un documento de tipo PDF con la información almacenada de la empresa y posteriormente compartirlo mediante todas las formas permitidas por el dispositivo ejemplo Bluetooth, Correo , Zappy, etc.



La opción Gráficos nos permite escoger una perspectiva de la empresa.

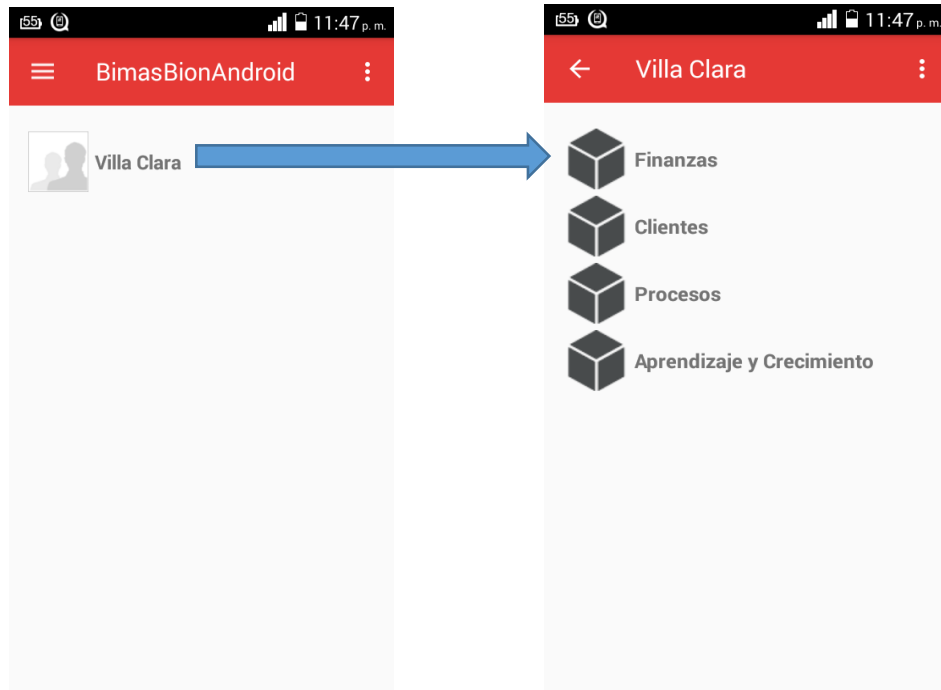


Después de seleccionar la perspectiva aparece un listado con los indicadores pertenecientes a la misma. Después de seleccionado uno o dos de los indicadores aparece en la parte inferior derecha de la pantalla un botón flotante para mostrar los valores: plan y real en un gráfico de barras.



Interfaz Perspectivas

En la pantalla principal se muestra el listado de sub organizaciones, si se selecciona una de ellas se mostrara el listado de las perspectivas de la misma.



Interfaz Grupos

Si se presiona una perspectiva de la interfaz anterior se muestra el listado de grupos pertenecientes a dicha selección.



Interfaz Indicadores

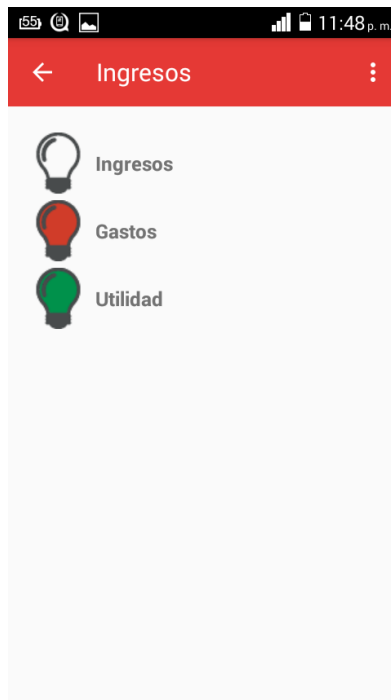
Si se presiona un grupo de la interfaz anterior se muestra el listado de indicadores pertenecientes al mismo. Los colores de los iconos indican la evaluación del indicador.

Blanco: sin definir

Rojo: Mal

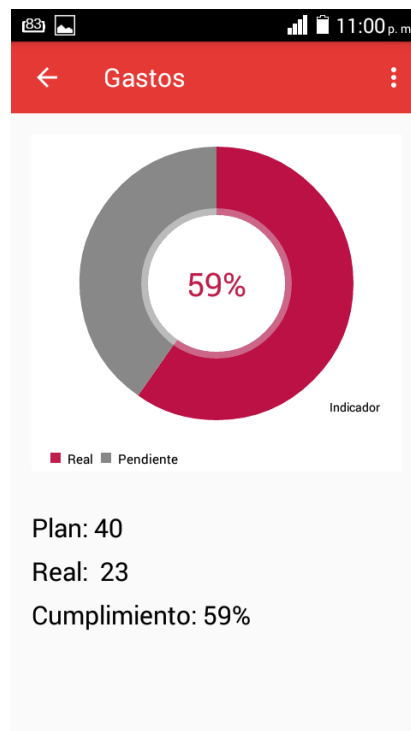
Amarillo: Regular

Verde: Bien



Interfaz Vista Indicador

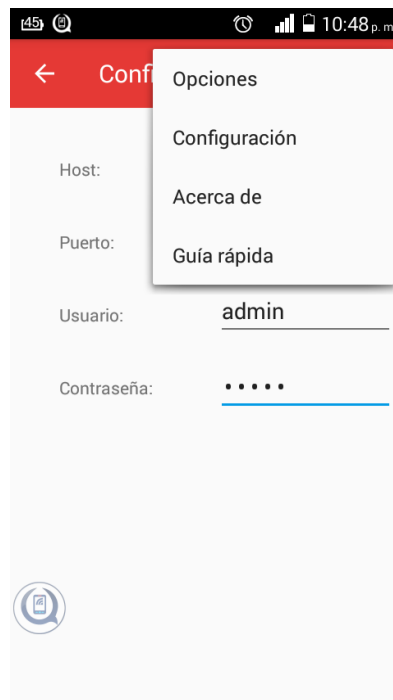
Si se selecciona un indicador de la interfaz anterior se muestran los valores: real, plan y cumplimiento de dicho indicador así como un gráfico de pastel que muestra el porcentaje de cumplimiento.



Otras Opciones

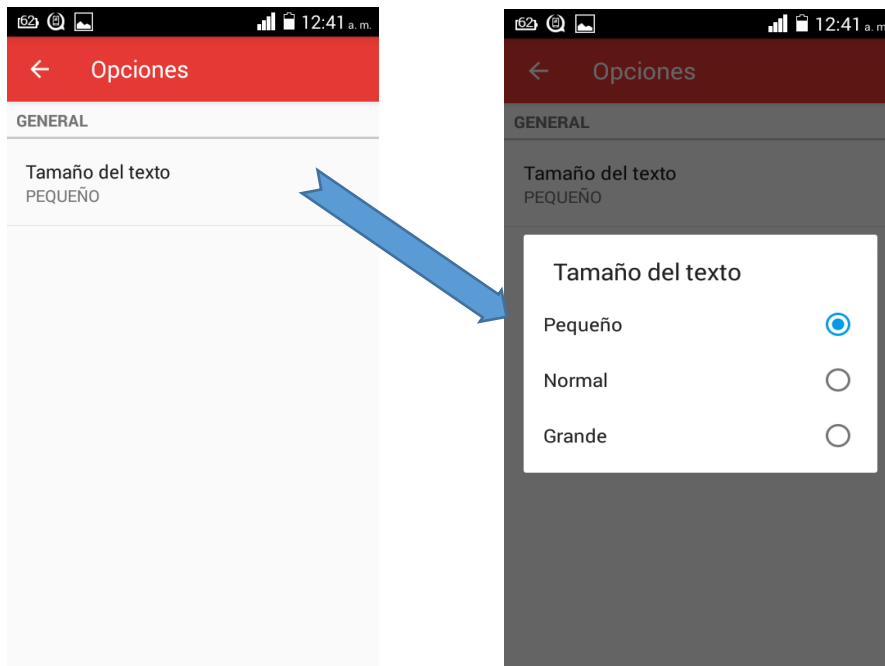


Al presionar este menú se puede acceder a las opciones de acerca de, guía rápida, opciones y configuración.



Interfaz Opciones

Si presionamos en opciones tenemos la posibilidad de cambiar el tamaño de fuente de algunos textos de la aplicación.



Interfaz Acerca de

En esta opción se muestra los derechos sobre la aplicación así como información de los desarrolladores.

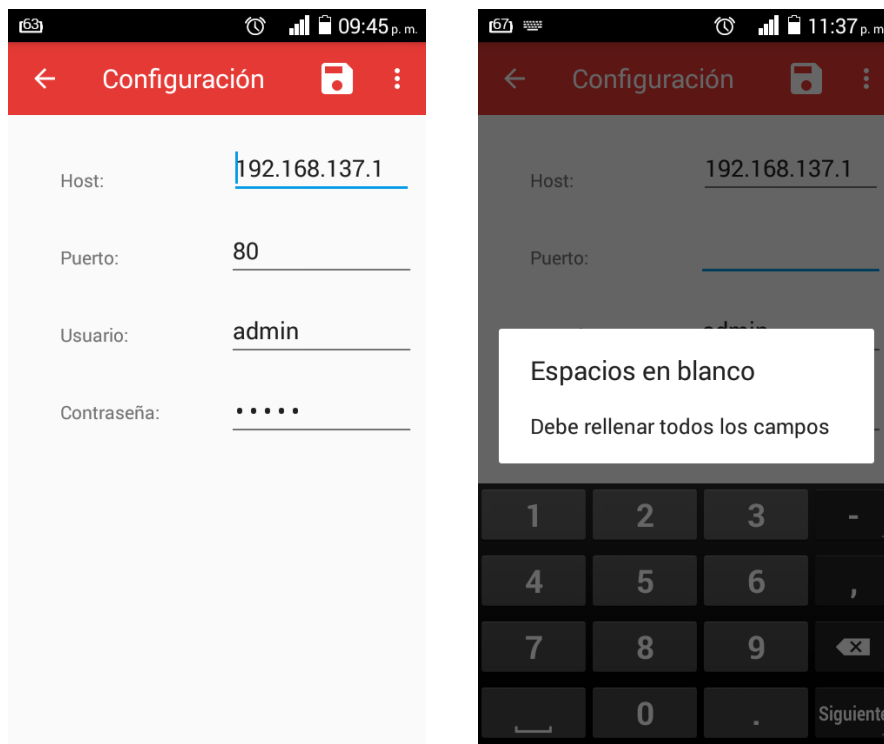


Presionar en el símbolo de correo permite seleccionar una aplicación de correo existente en el dispositivo poniéndose automáticamente un asunto predeterminado y la dirección de correo de la empresa Desoft Villa Clara.

Interfaz Configuración

En esta interfaz se configuran los parámetros de conexión con el servidor de Bimas Server así como el usuario y la contraseña. Cuando se abre por primera vez la aplicación después de la Guía Rápida y el Splash aparece esta pantalla y no la principal para configurar dichos parámetros.

Después de introducida la información necesaria se presiona el botón en la barra superior para ser guardada, si se deja algún campo en blanco aparecerá un mensaje de error.



Conclusiones del Capítulo:

En este capítulo se describen las pruebas a la aplicación obteniéndose resultados satisfactorios:

1. Las pruebas realizadas por la herramienta online Monkop sobre los dispositivos seleccionados, mostró que la aplicación utiliza pocos recursos físicos de los dispositivos móviles.
2. El tiempo de inicio es rápido y sin errores de ejecución o de bloqueo.
3. La interfaz es responsive ya que se adapta a distintas dimensiones de pantalla sin perder información ni sobrecargas de elementos.
4. Se confeccionó un pequeño manual de usuario para facilitar el uso de la aplicación.

Conclusiones

1. Para el desarrollo de la aplicación se decidió optar por una metodología ágil ya que estas tienen un tiempo corto para la entrega del producto final y son las más usadas en el desarrollo de aplicaciones para dispositivos móviles. Entre los Framework de desarrollo de aplicaciones móviles para la plataforma Android es recomendable Android Studio, ya que es el IDE oficial para dicho sistema operativo y brinda muchas facilidades. Para la visualización de los indicadores se escogió la biblioteca MPAndroidChart. Esta es una de las más utilizadas para creación de los diferentes tipos de gráficos.
2. La base de datos diseñada evita la necesidad de estar conectado al servicio de Bimas Server para visualizar los indicadores. Permitiendo el seguimiento en todo momento de los indicadores de desempeño.
3. Se realizaron pruebas a la aplicación desarrollada comprobando que la misma funciona tanto en dispositivos con diferentes versiones del Sistema Operativo Android así como en diferentes densidades y tamaños de pantalla.
4. Se confeccionó un manual de usuario que explica las principales funcionalidades de la aplicación.

Recomendaciones

1. Para próximas versiones se deben agregar otros tipos de gráficos para la visualización de los indicadores de desempeño.

Referencias bibliográficas

- Armijo, M. (2009) 'Lineamientos metodológicos para la construcción de indicadores de desempeño', *Santiago de Chile: ILPES/CEPAL/CAPRADE. Recuperado el*, 3.
- Avison, D., Cole, M. and Fitzgerald, G. (2006) 'Reflections on teaching information systems analysis and design: from then to now!', *Journal of Information Systems Education*, 17(3).
- Balaguera, Y. D. A. (2013) 'Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual', *Revista de Tecnología*. Universidad El Bosque, 12(2), pp. 111–123.
- Benbourahla, N. (2015) *Android 5: Principios del desarrollo de aplicaciones Java*. Ediciones Eni.
- Blanco, P. et al. (2009) 'Metodología de desarrollo ágil para sistemas móviles. Introducción al desarrollo con Android y el iPhone', *Dr. en Ing. Sist. Telemáticos*, pp. 1–30.
- Cadavid, A. N., Martínez, J. D. F. and Vélez, J. M. (2013) 'Revisión de metodologías ágiles para el desarrollo de software', *Prospectiva*. Universidad Autónoma del Caribe, 11(2), pp. 30–39.
- Canós, J. H. and Letelier, M. C. P. P. (2012) 'Metodologías ágiles en el desarrollo de software'.
- Chinchay Cuenca, M. J. (2016) *Desarrollo de una aplicación móvil android para la bûsqueda de plazas disponibles en un parqueadero*.
- Cinar, O. (2015) *Android quick APIs reference*. Apress.
- Del'ia, L. et al. (2013) 'Un análisis experimental de tipo de aplicaciones para dispositivos móviles', in *Congreso Argentino de Ciencias de la Computación (CACIC)*.
- Federico Guzman (2016) *RESTful API: ¿Qué es y para que sirve?*
- Gargenta, M. and Nakamura, M. (2014) *Learning Android: Develop Mobile Apps Using Java and Eclipse*. ' O'Reilly Media, Inc.'
- Hohensee, B. (2014) *Introducción A Android Studio. Incluye Proyectos Reales Y El Código Fuente*. Babelcube Inc.
- KAPLAN, R. and NORTON, D. (2005) 'Síntesis del Libro Cuadro de Mando Integral', *Universidad de Carabobo. Diplomado en Finanzas*. Año.
- Ko, Y. et al. (2013) 'Suggestions of Prototyping Tool for Service Design Co-Creation', URL: <http://design-cu.jp/iasdr2013/papers/1048-1b.pdf>. Accessed, 7.
- Lequerica, J. R. (2013) *Desarrollo de aplicaciones para Android*. Ediciones Anaya Multimedia.
- Material Design de Google _ GUÍA EN ESPAÑOL _ Agencia MDW (no date). Available at: <https://www.agenciamdw.es/material-design-google-guia-en-espanol/>.
- Pablo (2014) *Una biblioteca para mostrar gráficas (Android)*.

Pintado, R. *et al.* (2015) *Análisis, diseño y desarrollo de un prototipo de endpoint para mejorar la producción de web services*.

Poole, D. B. (2009) 'Systems and methods for determining the software components included in a view of a software development project at a particular time'. Google Patents.

Seta, L. De (no date) *Introducción a los servicios web RESTful*.

Vanegas, C. A. (2012) 'Desarrollo de aplicaciones sobre Android', *Revista v{\'i}nculos*, 9(2), pp. 129–145.

Anexos

```
public class SplashActivity extends AppCompatActivity {
    SharedPreferences m_sharedPreferences;
    private static final String PREFS = "prefs";
    private boolean init;
    private boolean forced;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        m_sharedPreferences = getSharedPreferences(PREFS, MODE_PRIVATE);
        setContentView(R.layout.activity_splash);

        if (isFirstTime()){
            startActivity(new Intent(SplashActivity.this, GuiaRapidaActivity.class).putExtra("primera", true));
        } else {

            final Thread timer = run() -> {
                try {
                    sleep(2000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                } finally {

                    Intent intent = new Intent(SplashActivity.this, OrganizacionActivity.class);
                    startActivity(intent);
                    overridePendingTransition(R.anim.move_left_in_activity,
                        R.anim.move_right_out_activity);

                    finish();
                }
            };
            timer.start();
        }
    }

    @Override
    protected void onPause() {
        super.onPause();
        finish();
    }

    //Evita salir dando atras, activar:
    @Override
    public void onBackPressed() { return; }

    private boolean isFirstTime(){
        return m_sharedPreferences.getBoolean("primera_vez_inicia", true);
    }
}
```

Figura 1: Construcción del Splash

```

@Override
public boolean onNavigationItemSelected(MenuItem item) {
    boolean pressed = false;
    switch (item.getItemId()) {
        case R.id.nav_actualizar:

            host = m_sharedPreferences.getString("host", "");
            puerto = m_sharedPreferences.getString("puerto", "");
            user = m_sharedPreferences.getString("user", "");
            pass = m_sharedPreferences.getString("pass", "");

            // System.out.println(host);

            if (host.equals("") || puerto.equals("") || user.equals("") || pass.equals("")) {
                AlertDialog alertDialog = new AlertDialog.Builder(this).
                    setTitle("Parámetros incompletos").
                    setMessage("Es necesario ir a configuración").create();
                alertDialog.show();
            } else {

                WifiManager connManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);

                if (connManager.getWifiState() == WIFI_STATE_DISABLED) {

                    AlertDialog alertDialog = new AlertDialog.Builder(this).
                        setTitle("Wi-Fi desactivada").
                        setMessage("Es necesario tener la Wi-fi activada").create();
                    alertDialog.show();

                } else {
                    if (connManager.getWifiState() == WIFI_STATE_ENABLED && connManager.getConnectionInfo().getNetworkId() == -1) {

                        AlertDialog alertDialog = new AlertDialog.Builder(this).
                            setTitle("Conexión no establecida").
                            setMessage("No esta conectado a ninguna red").create();
                        alertDialog.show();

                    } else {
                        System.out.println(pass);
                        DialogFragment dialogFragment = new ActualizandoDialog();
                        dialogFragment.setCancelable(false);
                        dialogFragment.show(getSupportFragmentManager(), "Actualizando Dialog");
                        BackgroundHandler handler = BackgroundHandler.getInstance(getApplicationContext());
                        Message message = Message.obtain(handler, BackgroundHandler.CONECTAR);
                        message.getData().putStringArray("data", new String[]{host, puerto, user, pass});
                        message.sendToTarget();
                    }
                }
            }
    }
}

```

Figura 2: Método onNavigationItemSelected ()

```

    case CONECTAR:
        mgr.notify(ID_NOTIF_JSON, buildActualizarDatos(System.currentTimeMillis(), true).build());
        String[] data = msg.getData().getStringArray("data");
        String host = data[0];
        String puerto = data[1];
        String user = data[2];
        String pass = data[3];
        final String auth = user + ":" + pass;
        String encoded = new String(Base64.encode(auth.getBytes(), Base64.NO_WRAP));

        client = new AsyncHttpClient();

        client.setConnectTimeout(3000); // default is 10 seconds, minimum is 1 second
        client.setResponseTimeout(4000); // as above
        //client.setTimeout(4000); // both connection and socket timeout
        client.setMaxRetriesAndTimeout(2, 1000); // times, delay

        client.addHeader("Authorization", "Basic " + encoded);
        // hasta aquí el logueo
        String service="";
        if (host.charAt(0)>'0' && host.charAt(0)<'9') {
            service = "http://" + host + ":" + puerto + INDICATOR_SERVICE;
        }else{
            service=host + ":" + puerto + INDICATOR_SERVICE;
        }

        requestHandle = client.get(service, asyncHttpResponseHandler);
        break;
    }
}

```

Figura 3: Conexión con la API RESTful de BIMAS Server

```

@Override
public void onSuccess(int i, cz.msebera.android.httpclient.Header[] headers, byte[] bytes) {
    // called when response HTTP status is "200 OK"
    String strJson = new String(bytes);
    JSONObject json = null;
    System.out.println(strJson);

    try {
        json = new JSONObject(strJson);
        String object_json = json.getString("objectjson");

        //llamada al servicio el cual es el encargado de actualizar los datos
        Intent intent = new Intent(context.getApplicationContext(), ActualizarDatosService.class);
        Bundle bundle = new Bundle();
        bundle.putString("Json", object_json);
        intent.putExtras(bundle);
        intent.setAction("cu.desoftvc.bimas.bimashionandroid.ACTUALIZAR_DATOS");
        context.startService(intent);

    } catch (JSONException e) {
        try {
            //verificando si usuario o contraseña incorrecta
            JSONObject error=new JSONObject(strJson);
            if (error.getString("exception").equals("Authenticated access required"))
                mensaje = USER_PASS_MAL;
            mgr.notify(ID_NOTIF_JSON, buildActualizarDatos(-1).setOngoing(false).build());
            EventBus.getDefault().postSticky(new ActualizarProgressEvent(preprocessDoneCant, preprocessDoneTotal, false, USER_PASS_MAL));
        } catch (JSONException e1) {
        }
    }
}

```

Figura 4: Llamada al servicio de Android con los datos obtenidos


```

private File exportarPdf(List<Organizacion> organizaciones) throws FileNotFoundException, DocumentException {
    File path = Environment.getExternalStorageDirectory();
    File f = new File(path, "Indicadores.pdf");
    FileOutputStream stream = new FileOutputStream(f);
    Document document = new Document();
    PdfWriter.getInstance(document, stream);
    document.open();
    for (int i = 0; i < organizaciones.size(); i++) {
        Organizacion temp = organizaciones.get(i);
        document.add(new Paragraph(temp.getNombreOrganizacion()));
        //creando tabla de indicadores en el pdf
        PdfPTable table = new PdfPTable(new float[]{5, 2, 2, 2});
        table.setSpacingBefore(40);
        table.setSpacingAfter(40);
        table.setWidthPercentage(100);
        table.setPaddingTop(Element.ALIGN_JUSTIFIED);
        table.addCell("Indicadores");
        table.addCell("Plan");
        table.addCell("Real");
        table.addCell("Porcentaje");
        IndicadorMapper indicadorMapper = new IndicadorMapper();
        List<Indicador> indicadores = indicadorMapper.listarIndicadores(bd, false, temp.getId_organizacion(), true);

        for (int j = 0; j < indicadores.size(); j++) {
            Indicador indicador = indicadores.get(j);
            table.addCell(indicador.getIndicador());
            table.addCell(((int) indicador.getPlan() + ""));
            table.addCell(((int) indicador.getReal() + ""));
            table.addCell(((int) indicador.getPorcentaje() + " %"));
        }
        document.add(table);
    }
    document.close();
    EventBus.getDefault().post(new VisualExportarPdfEvent(f));
    return f;
}

```

Figura 5: Método para exportar documento PDF