



FACULTAD 2

Aplicación Móvil para GRHS (GDROID)

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores:

Luis Ángel Roche Broche
Yoán Jorge Nazábal Gómez

Tutor:

Ing. Yenlys Guerra Dávila

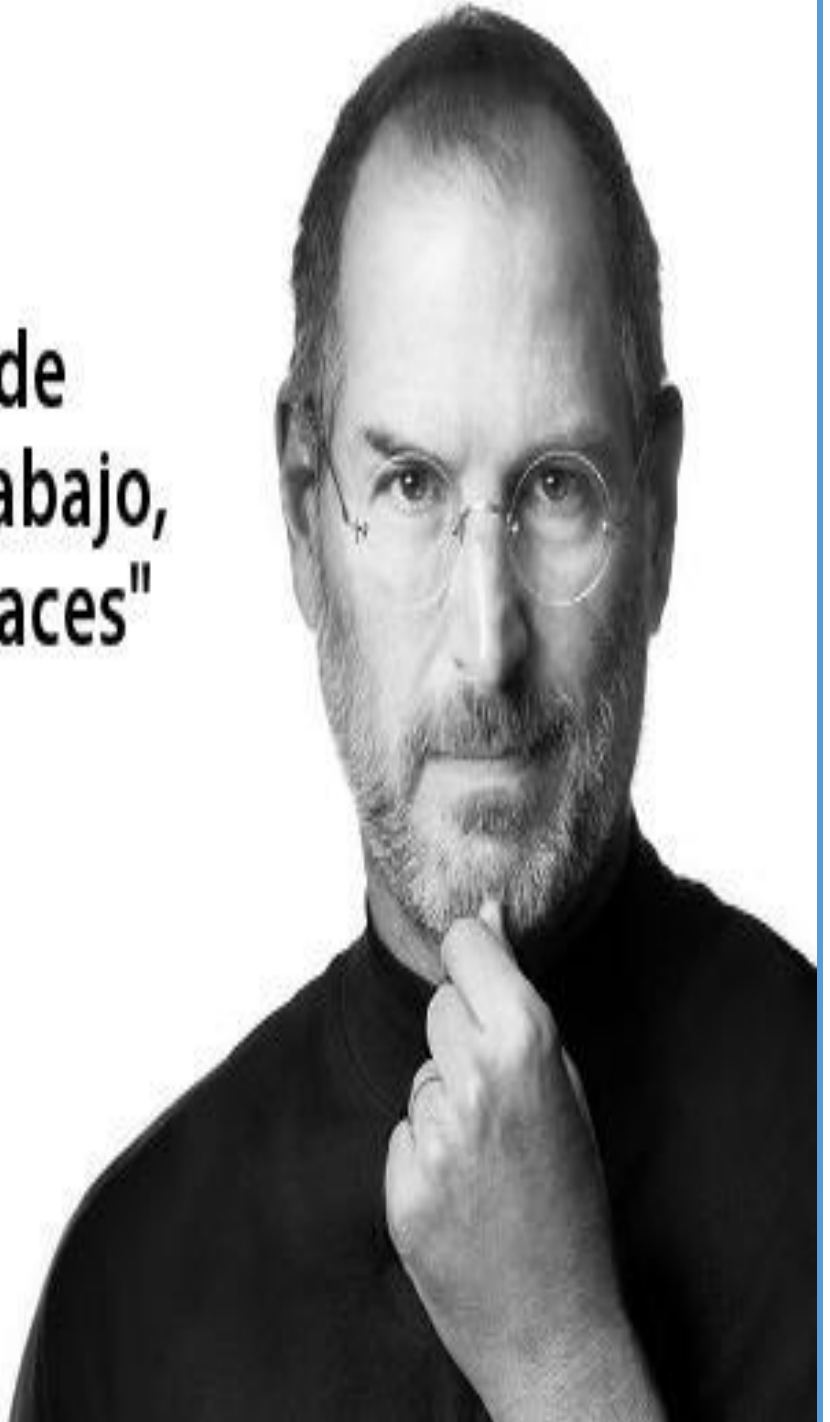
La Habana, junio, 2017.

“Año 59 de la Revolución”

FRASE

**"La única forma de
hacer un gran trabajo,
es amar lo que haces"**

—Steve Jobs



DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro de Telemática (TLM) de la Facultad 2 y a la Universidad de las Ciencias Informáticas (UCI) a hacer el uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Luis Ángel Roche Broche
(Autor)

Yoán Jorge Nazábal Gómez
(Autor)

Ing. Yenlys Guerra Dávila
(Tutor)

AGRADECIMIENTO

Yoán:

- *A mis padres Ignacio e Iraida, mi abuela Oraida por todo su esfuerzo, dedicación y apoyo a lo largo de mi vida.*
- *A mis abuelos América, Eligio y Domingo por ayudarme en su vida y desde el cielo a ser la persona que soy.*
- *A mi hermanito David por ser lo más importante en mi vida.*
- *A mi novia Ana Laura por su amor y paciencia, por estar a mi lado en todo momento.*
- *A mi suegra Odalys por ser para mí como una madre, por todos sus consejos y su confianza.*
- *A mi amigo Luis Ángel por realizar esta tarea juntos.*
- *A mi hermana Yashila y a Marisela por guiarme por el camino correcto.*

Luis Ángel:

- *A mis padres Ángel y Eneida y a mi abuelo Ramón por ser las 3 personas más importantes en mi vida, aunque solo una pueda estar presente en este momento, va dedicado en especial a ellos por serlo todo para mí.*
- *A mis primas Raiza y Roxana, a mi abuela Angelita por el apoyo que me han dado durante mi vida.*
- *A mi novia Xiomara por su apoyo y comprensión en toda esta trayectoria.*
- *A mis suegros Ledys y Pepe, a Osorio por acogerme como parte de ellos y ayudarme a pasar por los momentos más difíciles.*

Ambos:

- *A nuestros familiares y amigos, por la comprensión, afecto y apoyo incondicional que nos han brindado a lo largo de este difícil trayecto que es la formación académica y profesional.*

GDROID

- *A nuestra tutora Yenlys por acogernos, aconsejarnos y conducirnos por el camino correcto durante este trabajo.*
- *A nuestros amigos Juan Miguel, Yosiel y Julien por su amistad y compañía en todo este tiempo. Por compartir grandes momentos juntos.*
- *A nuestros compañeros de aula y de residencia.*
- *A todas aquellas personas que dentro y fuera de esta institución han sido un gran apoyo para nosotros. Gracias.*
- *A todos los que de una forma u otra han hecho posible este logro.*

DEDICATORIA

Yoán:

Le dedico esta tesis a mis padres, mis hermanos, mi novia, mis amigos y a todas aquellas personas importantes en mi vida, sin ellos no hubiese sido posible alcanzar mis objetivos.

Luis Ángel:

Dedico esta tesis a mis padres, mi abuelo, mi novia, mis suegros por haberme ayudado y apoyado a cumplir mis metas y propósitos en la vida.

GDROID

RESUMEN

La aparición de dispositivos móviles ha posibilitado el acceso a internet, desde cualquier lugar, de un mayor número de personas. Por ello numerosas empresas se han integrado al desarrollo de aplicaciones móviles con el objetivo de aumentar su alcance en el mercado.

El presente trabajo tiene como objetivo general, la implementación de nuevas funcionalidades para la aplicación móvil del Sistema Gestor de Recursos de Hardware y Software, desarrollado en el centro de Telemática de la Facultad 2 de la Universidad de Ciencias Informáticas, que permita el escaneo del inventario de la ficha técnica de un ordenador y mostrar la información que contiene dicho ordenador, y la representación de la información estadística mediante el uso de gráficos de barras. Para el desarrollo del presente trabajo se utiliza como metodología de desarrollo de software XP, como tecnologías del lado servidor, el lenguaje de programación Python en conjunto con el framework Django y el IDE de desarrollo PyCharm, además, como tecnologías del lado cliente se utiliza el lenguaje de programación Open JDK en conjunto con el framework de desarrollo Android API, el IDE de desarrollo Android Studio, Android SDK Tools, y para la compatibilidad con librerías escritas en C o C++, como es el caso de tess-two, Android NDK.

Palabras Claves: Escaneo, Tratamiento, Ficha técnica, Representación gráfica.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción.....	5
1.2 Estado del arte	5
1.2.1 Aplicación Gadmin Android para la administración de sistema Gestor de Hardware y Software:.....	5
1.2.2 Sistemas similares	8
1.2.3 Sistemas con escaneado y tratamiento de imágenes.....	8
1.2.4 Tratamiento de Datos.....	9
1.2.5 Análisis del estudio realizado	11
1.3 Tratamiento de información estadística.	12
1.4 Escaneo y tratamiento de imágenes.....	13
1.4.1 Técnica de Reconocimiento Óptico de Caracteres (OCR).....	13
1.4.2 Librerías utilizadas	14
1.5 Metodología de desarrollo	15
1.5.1 Metodología a utilizar	16
1.6 Tecnologías y herramientas.....	16
1.6.1 Open JDK 8	16
1.6.2 Android Studio 2.2.3.....	17
1.6.3 Android SDK Tools 23.0.5.....	17
1.6.4 Android API 21	17
1.6.5 Android NDK 14	18
1.7.6 Python 2.7.....	18
1.7.7 Django 1.4.....	18

GDROID

1.7.8 PyCharm 16.3.3	19
1.7 Conclusiones del capítulo	19
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN, EXPLORACIÓN Y PLANIFICACIÓN DEL SISTEMA GDROID.	20
2.1 Introducción	20
2.2 Modelo de propuesta de solución.	20
2.2.1 Escanear imagen	21
2.2.2 Mostrar datos estadísticos mediante el uso de gráficas	21
2.3 Características no funcionales del sistema.	21
2.3.1 Interfaz de usuario.	21
2.3.2 Consistencia de los datos.	21
2.3.3 Usabilidad.	22
2.3.4 Software.	22
2.3.5 Hardware.	22
2.3.6 Seguridad de la información.	22
2.4 Fase de exploración.	22
2.4.1 Historias de usuarios.	22
2.5 Fase de planificación.	26
2.5.1 Estimación del esfuerzo por HU.	27
2.5.2 Plan de Iteraciones.	27
2.5.3 Plan de duración de Iteraciones.	28
2.5.4 Plan de entrega.	28
2.6 Conclusiones del capítulo.	28
CAPÍTULO 3. DISEÑO, IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA.	29
3.1 Introducción	29
3.2 Diseño.	29
3.2.1 Arquitectura de software.	29

GDROID

3.2.2 Arquitectura Cliente/Servidor:.....	29
3.2.3 Modelo-Vista-Presentación (MVP)	30
3.2.4 Patrón de diseño.	31
3.2.5 Patrón Maestro/Esclavo.	33
3.2.6 Tarjetas CRC (Clases-Responsabilidad-Colaborador).	33
3.2.7 Tareas de Ingeniería.	34
3.3 Pruebas.	40
3.3.1 Pruebas Unitarias:.....	40
3.3.2 Pruebas de Aceptación:	42
3.4 Conclusiones del capítulo.	45
CONCLUSIONES GENERALES.....	46
RECOMENDACIONES	47
REFERENCIA BIBLIOGRÁFICA.....	48
BIBLIOGRAFÍA.....	51
ANEXOS.....	57
Anexos I: Muestra de Tarjetas CRC.	57
Anexos II: Pruebas unitarias por el método de camino básico.....	58

ÍNDICE DE FIGURAS

Ilustración 1: Arquitectura Cliente/Servidor en Gadmin Android	7
Ilustración 2: Datos de la consola móvil NetSupport DNA.	8
Ilustración 3: Ejemplo de gráfico Radar Polar en 3D Gráficos.	10
Ilustración 4: Forma de representación de los datos en SJOGraph.....	11
Ilustración 5: Propuesta de solución.....	20
Ilustración 6: Arquitectura Cliente/Servidor desglosada en sus 2 niveles.	30
Ilustración 7 Diseño de MVP	30
Ilustración 8: Resumen de Pruebas de Camino Básico.....	42
Ilustración 9 Resumen de pruebas de aceptación.....	43

ÍNDICE DE TABLAS

Tabla 1 HU-1 Tratamiento de imagen.	24
Tabla 2: HU-2 Mostrar resultados de la sincronización con Gserver.	25
Tabla 3: HU-3 Visualización de la información estadística.	26
Tabla 4: Estimación de esfuerzo por Historia de Usuario.	27
Tabla 5 Plan de duración de las iteraciones.....	28
Tabla 6: Plan de entrega de las iteraciones.	28
Tabla 7: Tarjeta CRC para la clase ActionStatisticFragmen.....	33
Tabla 8: Tareas de Ingeniería.....	34
Tabla 9: TI-1 Captura de la Imagen.....	35
Tabla 10: TI-2 Transformación de imagen a texto digital.....	36
Tabla 11: TI-3 Organización de los caracteres.....	36
Tabla 12: TI-4 Transformar caracteres en MB.....	36
Tabla 13: TI-5 Recibir el MB de la imagen.....	36
Tabla 14: TI-6 Buscar MB en la base de datos.....	37
Tabla 15: TI-7 Obtener datos del ordenador.....	37
Tabla 16: TI-8 Mostrar información al usuario.....	38
Tabla 17: TI-9 Obtener datos estadísticos de la base de datos remota de Gserver.....	38
Tabla 18: TI-10 Mostrar datos estadísticos de los agentes.....	38
Tabla 19: TI-11 Mostrar datos estadísticos de los inventarios.....	39
Tabla 20: TI-12 Mostrar datos estadísticos de las incidencias.....	39
Tabla 21: TI-13 Mostrar datos estadísticos de las acciones.....	40
Tabla 22: PU CB Método AddValuesToBARENTRY.....	41
Tabla 23: PA-1 Tratamiento de imagen.....	44
Tabla 24: PA-2 Mostrar resultados.....	44
Tabla 25: PA-3 Visualización de la información estadística.....	45
Tabla 26 TCRC ScanActivity.....	57
Tabla 27:TCRC ScanResult.....	57
Tabla 28: TCRC SyncAdapter.....	57
Tabla 29: TCRC ListAPITask.....	57
Tabla 30: TCRC InventoryStatisticFragment.....	58
Tabla 31: TCRC IncidencesStatisticFragment.....	58
Tabla 32: TCRC AgentStatisticFragment.....	58

GDROID

Tabla 33: TCRC DataBaseContractProvide	58
Tabla 34: PU CB Método GetPath	59
Tabla 35: PU CB Método onClick.....	59
Tabla 36: PU CB Método StartOCR	60
Tabla 37: PU CB Método StartCamaraActivity	61
Tabla 38: PU CB Método getText.....	61
Tabla 39: PU CB Método onActivityResult	62
Tabla 40: PU CB Método onActivityResult.....	62
Tabla 41: PU CB Método ParseMotherboardStats	63

INTRODUCCIÓN

Las Tecnologías de la Informática y las Comunicaciones (TIC) han tenido un profundo impacto en todos los ámbitos de la sociedad, permitiendo el desarrollo de diversas áreas del conocimiento tales como: Ingeniería, Educación, Salud, entre otras. Entre las tecnologías más utilizadas actualmente se encuentra la tecnología móvil, la cual permite la comunicación constante desde casi cualquier lugar a través de la voz digitalizada e Internet. Es por esto que la movilidad ha cobrado cada vez más importancia en múltiples tareas dentro del entorno empresarial, convirtiéndose en un apoyo para lograr una mayor productividad y optimizar los procesos dentro de una organización (1).

La información es un recurso vital para toda organización, y el buen manejo de esta puede significar la diferencia entre el éxito o el fracaso(2), por lo que es necesario contar con sistemas de información que permitan analizar los datos obtenidos para mostrar, resumir, organizar y visualizar los mismos, mediante la representación gráfica, ya que permite legibilidad y claridad a la hora de mostrar la información. Con un sistema de información se puede tener el control sobre el inventario con el que se cuenta y así saber en todo momento que es lo que se tiene y que es lo que se necesita (2).

En Cuba han sido varias las instituciones que han incluido la tecnología móvil como un apoyo para la mejora del rendimiento de sus procesos, conscientes del beneficio del uso de la tecnología y los dispositivos móviles, así como de las aplicaciones móviles “a la medida” (desarrolladas para necesidades particulares de productividad). La Universidad de Ciencias Informáticas (UCI) institución en la que se desarrolla aplicaciones para dispositivos móviles cuenta con la Aplicación Android para la administración del sistema Gestor de Recursos de Hardware y Software, que es un complemento para el sistema Gestor de Recursos de Hardware y Software (GRHS) producido en el Departamento de Aplicaciones del Centro de Telemática de la Facultad 2.

GRHS es una solución informática que permite realizar el inventario de hardware y software de las computadoras donde se encuentre instalado, así como la detección de incidencias que se puedan producir en las mismas, alertando a los interesados y generando estadísticas. GRHS está constituido por cuatro subsistemas: colector de inventarios, **gclient**; el centralizador de inventarios, **gserver**; la consola de administración, **gadmin**; y la de actualización, **gupdate**.

La aplicación móvil para GRHS Gadmin Android permite la consulta de la información relacionada con los inventarios de los recursos de hardware y software, así como el monitoreo de estadísticas

GDROID

y la gestión de incidencias de GRHS desde cualquier punto geográfico dentro del alcance de la red inalámbrica dentro de la entidad.

A pesar de que actualmente la aplicación desarrollada satisface las necesidades funcionales, no existe un mecanismo efectivo que proporcione, a los asesores de tecnología de cada área encargado de controlar los recursos de un local, comprobar que los datos de la ficha técnica, que se adquiere del chasis de cada ordenador con los siguientes datos: Local, Puesto, Inventario (MB), Serie, Motherboard (Tarjeta madre), Microprocesador, Memoria RAM, Disco Duro, Unidades Ópticas, sean correctos. Esta situación puede conllevar a una falta de organización y control de estos medios, facilitando una vía para la pérdida y/o cambio de los mismos, generando afectaciones económicas a la UCI. Además, la información estadística representada en dicha aplicación se muestra a través de listas, lo que puede conducir a la falta de legibilidad de la información.

Atendiendo a lo anteriormente descrito se plantea como **problema a resolver**:

La aplicación móvil Gadmin Android, presenta limitaciones en cuanto a la visualización de la información estadística y la muestra de datos contenidos en la ficha técnica de un ordenador, captada por una imagen de un dispositivo móvil, imposibilitando la comprobación y representación de los datos mostrados a los directivos y responsables de la seguridad informática.

Objeto de estudio:

El proceso de visualizar información estadística y mostrar los datos contenidos en la ficha técnica de un ordenador captada por una imagen de un dispositivo móvil con sistema operativo Android.

Objetivo general:

Desarrollar a la aplicación móvil Gadmin Android funcionalidades que resuelva las limitaciones en el proceso de visualizar información estadística y mostrar los datos contenidos en la ficha técnica de un ordenador captada por una imagen de un dispositivo móvil, que permita la comprobación y representación de los datos mostrados a los directivos y responsables de la seguridad informática.

Campo de acción:

GDROID

Herramientas para los procesos de visualizar la información estadística y mostrar los datos contenidos en la ficha técnica de un ordenador captada por una imagen para dispositivos móviles con sistema operativo Android.

Tareas de investigación:

- Estudio de la Aplicación móvil Gadmin Android para el desarrollo de la Aplicación GDroid.
- Estudio de herramientas gestoras de inventarios de hardware y software que presenten una Aplicación sobre el sistema operativo Android, con el objetivo de establecer similitudes con GDroid.
- Análisis de sistemas con escaneado y tratamiento de imágenes, con el objetivo de tener un mayor dominio acerca del problema a resolver.
- Análisis de las herramientas que permitan la representación de la información estadística en forma de gráficos, con el objetivo de conocer las tendencias de las herramientas informáticas para graficar.
- Análisis de la metodología de desarrollo de software para estructurar, planificar y controlar el proyecto correctamente.
- Selección de las herramientas de trabajo a utilizar para la implementación de la Aplicación.
- Análisis de los tipos de prueba propuestos por la metodología para verificar la correcta implementación de la aplicación.

Métodos teóricos:

Analítico-Sintético: utilizado en el estudio de las fuentes bibliográficas existentes referente al proceso de visualización de información estadística en Android para inventarios de recursos de hardware y software, identificando los elementos más importantes y necesarios para dar solución al problema planteado.

Histórico-Lógico: permitió el estudio del comportamiento y evolución de los diferentes sistemas de inventarios de recursos que cuentan con una Aplicación Android, tanto a nivel nacional, como internacional.

Comparativo: permitió establecer, mediante la comparación, las analogías y las diferencias entre los distintos sistemas de inventarios de recursos que cuentan con una Aplicación Android.

GDROID

Métodos empíricos:

Medición: utilizado para medir la Aplicación Android mediante las pruebas unitarias y de aceptación garantizando el correcto funcionamiento del mismo.

Estructura del documento

El presente trabajo de diploma está estructurado en 3 capítulos cómo se describe a continuación:

Capítulo 1. Fundamentación teórica: Tiene como objetivo analizar los diferentes aspectos teóricos relacionados con visualización de información estadística en Android para inventarios de recursos de hardware y software. Además, se analizan y adoptan los principales conceptos que facilitan el estudio y comprensión de la temática. Se definen las tecnologías y herramientas, que permitan la correcta elaboración de la actual investigación.

Capítulo 2. Propuesta de Solución, Exploración, Planificación del Sistema. Tiene como objetivo el diseño de la propuesta de solución, así como obtener los principales artefactos relacionados con la metodología de software utilizada.

Capítulo 3. Diseño, Implementación y pruebas del sistema: Se exponen los principales aspectos relacionados con la implementación de la solución informática. Además, se presenta los diseños de las pruebas de aceptación a utilizar en la validación del sistema y se analizan los resultados de las pruebas unitarias que permiten evaluar la calidad de la propuesta de solución.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

El objetivo fundamental del presente capítulo es abordar los aspectos más relevantes que conforman el fundamento teórico de la investigación. Se describen los conceptos principales. Se realiza un estudio de los principales sistemas de gestión de recursos de hardware y software existentes, basándose principalmente en su integración con la tecnología móvil y las ventajas que esta brinda. Se realiza además una valoración de las principales tecnologías seleccionadas para el desarrollo del sistema, así como las herramientas y la metodología a utilizar.

1.2 Estado del arte

En este epígrafe se presentan algunas de las soluciones que existen en el mundo que poseen relación con la presente investigación.

1.2.1 Aplicación Gadmin Android para la administración de sistema Gestor de Hardware y Software:

La aplicación Gadmin Android es la versión anterior de GDroid (Aplicación en desarrollo).

La aplicación Gadmin Android del sistema de GRHS permite la consulta de la información relacionada con los inventarios de los recursos de hardware y software, así como el monitoreo de estadísticas y la gestión de incidencias, desde cualquier punto geográfico dentro del alcance de la red inalámbrica de la entidad. Además, permite el control del estado de los recursos de hardware y software, así como del cumplimiento de las políticas de seguridad informática establecidas. Gadmin Android es capaz de mostrar de manera detallada los datos de cada uno de los agentes conectados a la red, que posean una instancia de gclient instalada y que el usuario esté autorizado a consultar; incluyendo su ubicación, dirección IP, estado del recurso (activo o inactivo) así como el inventario de sus componentes de hardware y software. Los datos obtenidos del servidor pueden consultarse aun cuando no haya conexión a la red de la institución, mediante un sistema de sincronización y una base de datos local en el dispositivo. La aplicación permite la configuración de parámetros de la cuenta tales como la dirección donde se encuentra el servidor de GRHS y el dominio de autenticación al que pertenece el usuario(3).

1.2.1.1 Gadmin Android presenta las siguientes funcionalidades

1.2.1.1.1 Sincronización de datos

Gadmin Android tiene implementado un mecanismo de sincronización, que se utiliza para las operaciones de transferencia de datos entre Gadmin Android y Gserver, por medio de una estructura centrada en un Sync Adapter (Componente que encapsula el código para las tareas

GDROID

de transferencia de datos entre el dispositivo y el servidor) y que permite realizar acciones en períodos de tiempo sin inicio ni fin determinado, manteniendo la integridad de la información(3).

1.2.1.1.2 Autenticación

Android incluye un componente llamado Authenticator, para manejar las cuentas asociadas cada aplicación, el cual es vital para el esquema de sincronización. La autenticación de Gadmin Android se utiliza para gestionar la creación de la cuenta local asociada a la aplicación y el manejo seguro de las credenciales de autenticación de un usuario del sistema GRHS. Estas credenciales se utilizaron para la obtención del token (objeto que contiene las credenciales de seguridad para una sesión de autenticación e identifica al usuario, los grupos a los que pertenece y sus privilegios) de seguridad necesario para la realización de solicitudes al servidor(3).

1.2.1.1.3 Consulta y manipulación de datos

Gadmin Android realiza la consulta y manipulación de datos mediante un componente llamado Content Provider, que gestiona el acceso a un repositorio central de datos y ofrece una interfaz estándar y consistente para los mismos. Este componente maneja la comunicación entre procesos y el acceso seguro a los datos, protegiendo así los datos del acceso no autorizado por parte de aplicaciones externas(3).

1.2.1.1.4 Gestión de almacenamiento de datos en el dispositivo

Gadmin Android presenta un mecanismo que permite controlar cuánta información puede descargar y mantener almacenada. Ante cada solicitud de sincronización realizada por el usuario, el sistema primero verifica un parámetro en las preferencias de la aplicación. Dicho parámetro corresponde al porcentaje de espacio libre mínimo que debe existir en el dispositivo para autorizar una descarga de datos del servidor. Si el porcentaje de espacio libre, en el momento de la solicitud, es menor que este parámetro, la aplicación abortará el proceso de sincronización y mostrará al usuario un mensaje, en caso contrario se realizará la sincronización de los datos satisfactoriamente(3).

1.2.1.2 Arquitectura Cliente/Servidor

La aplicación Gadmin Android ocupa el nivel Cliente, ya que es el demandante de información del servidor de GRHS y es el software que se ejecuta del lado del usuario. El modelo de solución plantea un escenario en el que múltiples instancias de Gadmin Android solicitan información a Gserver, aplicación principal del nivel Servidor, y este se encarga de decidir qué información proveer según los privilegios del usuario que hace la solicitud(3).

GDROID

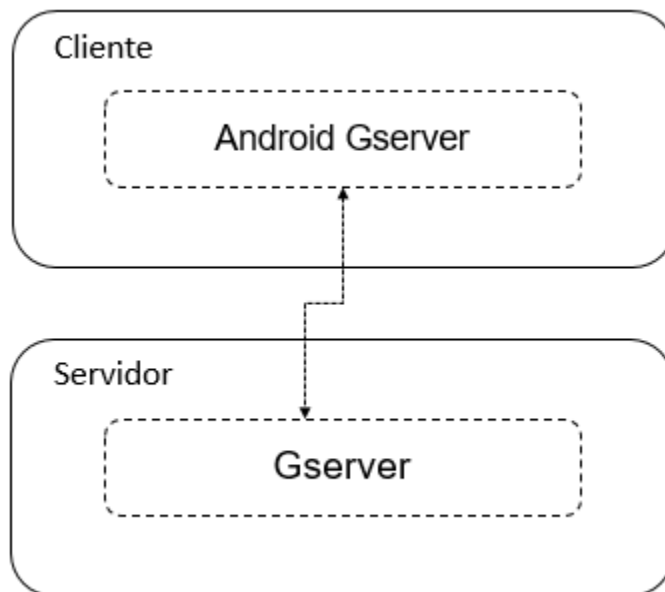


Ilustración 1: Arquitectura Cliente/Servidor en Gadmin Android

1.2.1.3 Seguridad de la información:

La conexión entre Gadmin Android y los servicios del API REST de GRHS debe realizarse mediante el protocolo HTTPS, en pos de proteger las credenciales de autenticación que se incluyen en la cabecera HTTP de cada solicitud (3).

Gadmin Android y Gserver se comunican de forma segura por medio del protocolo HTTPS garantizando la confidencialidad, integridad y disponibilidad de la información.

- **Confidencialidad:** La aplicación móvil Gadmin Android asegura que el acceso a la información está adecuadamente autorizado. Solo tienen acceso los usuarios autorizados.
- **Integridad:** La aplicación móvil Gadmin Android salvaguarda la precisión y completitud de la información y sus métodos de proceso.
- **Disponibilidad:** La aplicación móvil Gadmin Android asegura que los usuarios autorizados pueden acceder a la información cuando la necesitan.

1.2.1.4 Base de datos local de Gadmin Android:

La aplicación Gadmin Android utiliza el motor de gestión SQLite para el almacenamiento de la base de datos local. Se utiliza SQLite debido a que se enlaza con Gadmin Android pasando a ser parte integral del mismo, y su conjunto de base de datos se guarda como un solo fichero. En este motor se guardarán tantos datos como espacio libre tenga el dispositivo, ejecutando en cada sincronización la funcionalidad (Gestión de almacenamiento de datos en el dispositivo).

GDROID

1.2.2 Sistemas similares

1.2.2.1 Consola Móvil NetSupport DNA:

La aplicación móvil NetSupport DNA permite a un técnico, cuando está delante de su ordenador, buscar y ver un inventario detallado de hardware y software de cualquier PC de la red de la empresa. Dicha aplicación incluye un escáner de código QR para ayudar a identificar de forma instantánea cualquier ordenador, bien desde un código QR mostrado en la pantalla por DNA, o de una etiqueta fijada al dispositivo. La aplicación muestra un historial de todos los cambios efectuados al hardware, así como todas las instalaciones o desinstalaciones de software, además de las vistas de inventario y de historial. Además, indica las nuevas alertas que han surgido en la red(4).

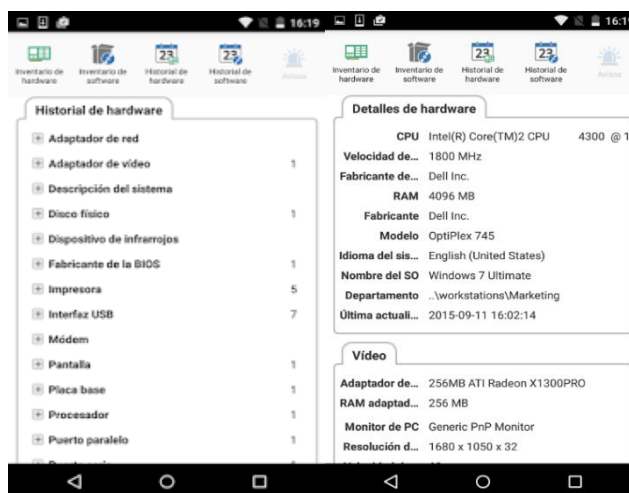


Ilustración 2: Datos de la consola móvil NetSupport DNA.

1.2.3 Sistemas con escaneado y tratamiento de imágenes

1.2.3.1 CamCard bussines:

Es un lector de tarjetas de visita profesional, digitaliza las tarjetas de visita en papel, directamente con la cámara del teléfono móvil o la tableta. Dicha aplicación realiza un escaneado del papel y presenta además una función de reconocimiento óptico de caracteres (OCR), lo que permite que la información obtenida pueda ser editada.

Para capturar la imagen se realiza una foto de la tarjeta de visitas, el software se encarga de buscar los bordes encuadrándolos correctamente y corrigiendo la perspectiva. Una vez introducidos los datos el usuario puede exportar o actualizar los datos de la agenda de contactos del móvil(5).

GDROID

1.2.3.2 Text Fairy:

Es una aplicación sobre el sistema operativo Android que permite la conversión de documentos de formato duro en digital, haciendo uso de la *tecnología de reconocimiento óptico de caracteres* realiza la conversión de una imagen captada por la cámara integrada al dispositivo en formato digital. También es posible realizar esta conversión a imágenes ya almacenadas en el dispositivo. En ambos casos, las fotografías deben estar bien iluminadas y enfocadas. Permite que se pueda encuadrar el texto que devuelve la imagen para corregir la perspectiva y se puede especificar en qué idioma está escrito(6).

La información digitalizada permite la corrección de errores y la edición del texto, así como copiarlo para ser pegado en otro documento.

1.2.4 Tratamiento de Datos

1.2.4.1 3D Gráficos:

Se utiliza para crear gráficos de aspecto en 3D (tres dimensiones). Dicha aplicación visualiza tipos de gráficos de gama en 3D y 2D (dos dimensiones). Establece colores, para las fotos y logos para una mayor comprensión de los mismos. Los gráficos realizados se guardan como imagen en 3D. La aplicación permite crear gráficos porcentuales, entre los que se encuentran gráficos pirámide, cono, anillo, entre otros; y gráficos combinados, entre los que se encuentran gráficos de línea, área, cilindro, escalera, entre otros. 3D Gráficos, como representa los gráficos en 3D, permite que los gráficos se puedan acercar, alejar, y moverse hacia la dirección deseada. Para su utilización se requiere una versión de Android 2.2 o superior(7).

GDROID

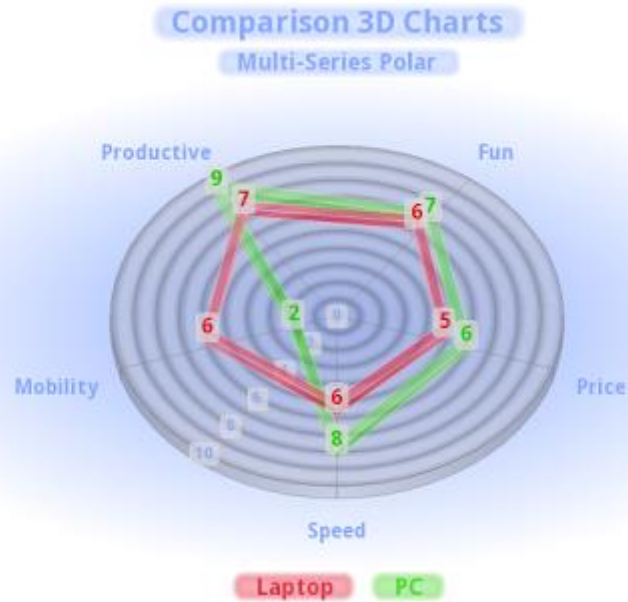


Ilustración 3: Ejemplo de gráfico Radar Polar en 3D Gráficos.

1.2.4.2 SJOGraph:

SJOGraph es una aplicación que permite crear gráficos de líneas, de barras y de pastel. Es de fácil comprensión para cualquier tipo de usuario, permite editar los datos, las leyendas, etiquetas y colores. SJOGraph permite, además, la impresión de datos y gráficos y contiene importación de datos externos, en hojas de cálculos. Para su utilización se requiere una versión de Android 2.2 o superior. La aplicación presenta un correcto mostrado de datos, que puede servir de ejemplo para el desarrollo de la aplicación GDroid(8).

GDROID



Ilustración 4: Forma de representación de los datos en SJOGraph.

1.2.5 Análisis del estudio realizado

Luego del análisis efectuado sobre la aplicación móvil Gadmin Android, se pudo concluir que el mismo no presenta funcionalidades que permitan realizar la captura y tratamiento de imágenes, las cuales son necesarias para obtener la información contenida en las fichas técnicas de los medios y poder comprobar si dicha información es correcta. Además, no cuenta con representación gráfica para mostrar la información estadística, sino que lo hace a través de listas.

En el caso de la consola móvil NetSupport DNA sirve como ejemplo ya que cuenta con todos los requisitos necesarios para dar solución al problema, sin embargo, es un software privativo que solo se puede utilizar en lugares donde esté instalado el servidor NetSupport DNA, ya que es un apoyo para este sistema.

De los sistemas de escaneo y tratamiento de imágenes se pudo obtener información de técnicas y métodos de trabajo que permiten hacer uso de tecnologías como OCR, utilizada para la captura y conversión de imágenes en datos específicamente en aplicaciones Android. Las aplicaciones estudiadas en la representación gráfica, permitieron conocer métodos y técnicas para lograr la representación de la información estadística mediante el uso de gráficas en dispositivos móviles, sobre el sistema operativo Android.

Luego del análisis de las aplicaciones anteriormente vistas, no existe una aplicación móvil capaz de juntar las funcionalidades de control de inventarios, representación gráfica de la visualización de estadísticas y escaneo y tratamiento de imágenes. Por lo que se puede concluir, que es

GDROID

necesario realizar una actualización a la aplicación móvil Gadmin Android, que contenga las funcionalidades de escaneo y tratamiento de imágenes y representación gráfica de las estadísticas.

1.3 Tratamiento de información estadística.

En la actualidad es fundamental el control y análisis de datos estadísticos, que origina con un estudio de los datos disponibles, donde se utilizan técnicas estadísticas para analizar la información o para explicar las condiciones que establecen la ocurrencia de algún fenómeno. Una de las principales formas de comprender el material disponible es la utilización de representaciones gráficas las cuales, capta la atención del cliente, presenta la información de forma sencilla, clara y precisa, facilita la comparación de datos y destaca las tendencias y las diferencias, no induce al error, analiza el conjunto de datos que representa para obtener conocimiento acerca de su estructura(9).

Para la representación gráfica de los datos estadísticos se utilizan los siguientes gráficos:

Gráfico de barras

Un gráfico de barras es una representación gráfica en un eje cartesiano de las frecuencias de una variable cualitativa o discreta. En uno de los ejes se posicionan las distintas categorías o modalidades de la variable cualitativa o discreta y en el otro el valor o frecuencia de cada categoría en una determinada escala. Los gráficos de barras se suelen usar para: magnitudes de varias categorías, ver la evolución en el tiempo en una magnitud concreta(10).

Gráfico de sectores o de pastel

Un gráfico de sectores es una representación circular de las frecuencias relativas de una variable cualitativa o discreta que permite, de una manera sencilla y rápida, su comparación. El círculo representa la totalidad que se quiere observar y cada porción, llamadas sectores, representan la proporción de cada categoría de la variable respecto al total. Suele expresarse en porcentajes(10).

Gráfico de líneas

Un gráfico de líneas es una representación gráfica en un eje cartesiano, de la relación que existe entre dos variables, reflejando con claridad los cambios producidos. En cada eje se representa cada una de las variables que se quiere observar. Se suelen utilizar para representar tendencias temporales, en el eje de las ordenadas se representa el tiempo, y en el eje de las abscisas se representa el desarrollo de una variable en el tiempo(10).

Gráfico de pareto

GDROID

Un gráfico de Pareto es un tipo de gráfico de barras vertical ordenado por frecuencias de forma descendente que identifica y da orden de prioridad a los datos. En el eje horizontal se representan las categorías de la variable que se quiere estudiar, en el eje vertical derecho se muestra la escala de porcentajes y en el eje vertical izquierdo la escala de frecuencias. Las barras muestran las frecuencias de las categorías de la variable y la línea representa el porcentaje acumulado de dichas frecuencias respecto al total(10).

Gráfico de dispersión

Este gráfico se utiliza para representar el grado de correlación entre dos variables. Muestra si el incremento o la disminución de los valores de una de las variables (denominada variable independiente, que se suele representar en el eje de horizontal) altera de alguna manera los valores de la otra (denominada variable dependiente, que se representa generalmente en el eje vertical)(10).

Luego de ver las facilidades que brinda cada gráfico y para que se utiliza, se decide utilizar los gráficos de barra, debido a que son utilizados para mostrar la frecuencia absoluta de cada variable, para la representación gráfica de los datos estadísticos contenidos en GRHS.

Para el desarrollo de la aplicación móvil GDroid se utiliza la librería MPAndroidchart.

MPAndroidchart:

Es una biblioteca de vista gráfica, soportando gráficos de barra de línea, radar, burbujas y velas, así como escalado, arrastrar y animaciones para el sistema operativo Android. Se ejecuta en API nivel 8 y niveles superiores(11).

1.4 Escaneo y tratamiento de imágenes

1.4.1 Técnica de Reconocimiento Óptico de Caracteres (OCR).

Reconocimiento Óptico de Caracteres, es una tecnología que permite convertir diferentes tipos de documentos tales como: documentos en papel escaneado, PDF archivos o imágenes captadas por una cámara digital en datos con opción de búsqueda y funcionalidad de editar(12). La tecnología OCR engloba un conjunto de técnicas basadas en estadísticas, en formas de caracteres, transformadas, que, complementándose entre sí, se emplean para distinguir de forma automática entre los diferentes caracteres alfanuméricos existentes. Se utiliza para identificar automáticamente a partir de una imagen, símbolo o caracteres, que pertenecen a un determinado alfabeto, un conjunto de datos comprensibles para un ordenador o dispositivo móvil. OCR para su correcto funcionamiento presenta cuatro etapas.

GDROID

Adecuación de la imagen (Preproceso): En esta fase el objetivo perseguido es eliminar la imagen de cualquier tipo de ruido o imperfección que no pertenezca al carácter, así como normalizar el tamaño de la imagen(13).

Selección de zona de interés (Segmentación): La fragmentación o segmentación es la operación que permite la descomposición de un texto en diferentes unidades lógicas. En esta fase el objetivo es obtener una imagen normalizada en la que se encuentre información susceptible(13).

Representación digital de la imagen (Extracción de características): En esta fase el objetivo perseguido es extraer las características de la imagen para su representación digital(13).

Distinción del carácter contenido en la imagen (Reconocimiento): En esta fase el objetivo principal es determinar los caracteres correspondiente de la imagen por medios de la minería de datos(13).

Hoy en día es habitual que se genere gran cantidad de contenido en multitud de soportes distintos. Entre esta multitud de tipos de soportes es habitual crear contenido en soporte manuscrito u tipográfico, por lo tanto, el reconocimiento óptico de caracteres puede ofrecer muchas ventajas competitivas. A continuación, se muestran algunas de ellas:

- En la digitalización de documentos supone un ahorro importante de tiempo, ya que la diferencia de tiempo entre entrar un texto de forma manual a reconocerlo de forma automática es muy grande.
- El reconocimiento de texto OCR transforma a los archivos escaneados en algo más que archivos de imagen. Aparte de la imagen escaneada, los archivos contendrán una parte de texto que se pueden usar para, la clasificación y ordenación de un archivo fotográfico o de imagen y facilitar la búsqueda de información en un archivo fotográfico o de imagen. De esta forma se podrá ofrecer un mayor servicio a terceros.
- El hecho de automatizar la transformación de imagen a texto implica un ahorro de recursos humanos, un incremento de la productividad y además puede ayudar a mejorar la calidad de algunos servicios que se ofrecen.
- El reconocimiento OCR y la digitalización de contenido suponen un ahorro importante de almacenamiento. Sin duda alguna, almacenar los documentos en formato digital supone un ahorro de espacio y de coste de almacenamiento muy importante.

1.4.2 Librerías utilizadas

Existen varias librerías para el desarrollo de la técnica de reconocimiento óptico de caracteres. Para el desarrollo de la aplicación GDroid se utilizarán las librerías Tess-two y Leptonica en la

GDROID

clase ScanActivity, la cual implementa la funcionalidad de escanear el inventario (MB) de una ficha técnica.

Tess-two:

El módulo tess-two contiene herramientas para compilar las bibliotecas Tesseract y Leptonica para su uso en la plataforma Android. Proporciona una API de Java para acceder a las API Tesseract y Leptonica compiladas de forma nativa. Contiene el código de procesamiento de imágenes adicional. Incluye funciones nativas para la detección de texto, detección de desenfoque, detección de flujo óptico y umbricación(14).

Leptonica:

Leptonica es una biblioteca de código abierto que contiene un software para el procesamiento de imágenes y aplicaciones de análisis de imágenes(15).

1.5 Metodología de desarrollo

En la actualidad existen numerosas propuestas de metodologías que inciden de distintas formas en el proceso de desarrollo del software. Estas metodologías de desarrollo son marcos de trabajo que constituyen un conjunto de procedimientos y técnicas usadas para estructurar, planificar y controlar este proceso de desarrollo de software generando la documentación necesaria para el mismo. Estos marcos de trabajo indican qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener, detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla. Dentro de estas metodologías existen dos grandes grupos, las metodologías tradicionales o pesadas y las ágiles. Las metodologías tradicionales se centran fundamentalmente en el control y definición de los procesos, tareas y herramientas a utilizar. Requieren de una extensa documentación ya que pretenden prever todo de antemano y pueden ser muy efectivas para proyectos de gran tamaño. Proponen un gran cúmulo de documentación de acuerdo al tamaño del proyecto y como consecuencia del equipo de desarrollo, es recomendable utilizarla en el desarrollo de proyectos medianos y grandes, por lo que no se consideró su utilización en el desarrollo del presente trabajo. Por otro lado, están las metodologías ágiles que se encargan de valorar al individuo y las iteraciones del equipo más que a las herramientas o los procesos utilizados. Al utilizar este tipo de metodología se hace mucho más significativo crear un producto de software que funcione, en lugar de escribir mucha documentación. Asimismo, se considera más importante la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan(16).

GDROID

1.5.1 Metodología a utilizar

Como guía para lograr el proceso de desarrollo de la propuesta de solución se decide emplear la metodología Programación Extrema (*XP, Xtreme Programming*) la cual se distingue, fundamentalmente, por su pronta, concreta y continua retroalimentación en ciclos cortos de tiempo, así como por su enfoque de planificación incremental, el cual permite definir un plan general que se espera evolucione durante la vida del proyecto(17).

La metodología XP posee también facilidad para flexibilizar la implementación de funcionalidades, respondiendo a los cambios que el negocio necesita. Se encuentra bien documentada con innumerables recursos en línea disponibles, comunidades libres y grupos de noticias, encontrándose una gran cantidad de proyectos desarrollados con esta metodología. Además, manifiesta la dependencia de un proceso de diseño evolutivo que dura todo el tiempo en que se desarrolla el sistema y su estrecha y cercana colaboración de los programadores con habilidades ordinarias(18).

A continuación, se muestran las razones de la selección:

- El proyecto es pequeño y todo el trabajo es llevado a cabo por una pareja de programadores.
- La generación de artefactos y roles en exceso no es necesaria ya que el proyecto es pequeño y está diseñado para ser realizado en el menor tiempo posible.
- Los requisitos suelen ser cambiados con frecuencia en la medida en que avanza el desarrollo del proyecto, así el cliente puede ir añadiendo HU, dividir las para agilizar el trabajo o eliminarlas simplemente. Esta metodología permite al equipo de trabajo modificar todos los planes conforme a lo anterior.
- Tanto cliente como desarrolladores forman parte del equipo de trabajo de forma tal que se logra una integración cliente-equipo de desarrollo, permitiendo la retroalimentación, corrección de errores y finalmente la realización de un producto capaz de satisfacer todas las necesidades del mismo.

1.6 Tecnologías y herramientas

1.6.1 Open JDK 8

Un Java Development Kit (JDK) es un entorno de desarrollo de programas para escribir aplicaciones Java. Se compone de un entorno de tiempo de ejecución que se encuentra en la parte superior de la capa del sistema operativo, así como las herramientas y la programación que se necesitan para compilar, depurar y ejecutar aplicaciones escritas en el lenguaje Java(19).

GDROID

Java es la plataforma de desarrollo utilizada en este proyecto, ya que las aplicaciones Android están escritas en este lenguaje.

1.6.2 Android Studio 2.2.3

Android Studio es el IDE (*Entorno de desarrollo integrado*) oficial para el desarrollo de aplicaciones Android. Entre sus capacidades, Android Studio ofrece:

- Un sistema de compilación flexible basado en Gradle (es una *herramienta de automatización en la construcción de código*).
- Plantillas de código para asistir en la construcción de características comunes de aplicaciones.
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android.
- Herramientas Lint (*herramienta de escaneo de código para identificar y corregir problemas de calidad estructural del mismo*) para capturar desempeño, usabilidad, compatibilidad entre versiones y otros problemas(20).

1.6.3 Android SDK Tools 23.0.5

El SDK (*Paquete de desarrollo de software*) de Android provee las bibliotecas API y las herramientas de desarrollo necesarias para compilar, probar y depurar aplicaciones Android(21). Se utiliza esta versión para la compatibilidad con la aplicación realizada anteriormente.

1.6.4 Android API 21

El nivel de API (*Interfaz de programación de aplicaciones*) es un valor entero que identifica de manera única la revisión de la Framework (*estructura conceptual y tecnológica de soporte, definido normalmente con artefactos o módulos concretos de software, que sirve de base para la organización y desarrollo de aplicaciones.*) API que ofrece una versión de la plataforma de Android. La misma proporciona que las aplicaciones se puedan usar para interactuar con el sistema de Android subyacente. En cada versión sucesiva de la plataforma Android se incluyen cambios y actualizaciones de la API. Estos cambios están diseñados de manera que cada nueva API sigue siendo compatible con las versiones anteriores(22).

La Framework API consiste en lo siguiente:

- Un conjunto básico de paquetes y clases.
- Un conjunto de atributos y elementos XML (eXtensible Markup Language, lenguaje de marcado desarrollado por el World Wide Web Consortium) para declarar un archivo de manifiesto (Brinda información que el sistema debe tener para ejecutar la aplicación).

GDROID

- Un conjunto de atributos y elementos XML para declarar recursos y acceder a estos.
- Un conjunto de intents (objeto de acción que puedes usar para solicitar una acción de otro componente de la aplicación).
- Un conjunto de permisos que las aplicaciones pueden solicitar, al igual que cumplimientos de permisos incluidos en el sistema.

La aplicación Android propuesta en el presente trabajo se implementará utilizando el framework API 21. Se utiliza esta versión para la compatibilidad con la aplicación realizada anteriormente.

1.6.5 Android NDK 14

Android NDK (Equipo de desarrollo nativo) es un conjunto de herramientas que permiten incorporar código nativo, C o C++, a tus aplicaciones para Android. Lo que facilita el uso de las bibliotecas de códigos escritas en lenguajes nativos. La capacidad de usar código nativo en aplicaciones para Android puede ser particularmente útil para los programadores que deseen realizar una o más de las siguientes acciones(23):

- Conectar sus aplicaciones entre plataformas.
- Reutilizar bibliotecas existentes o proporcionar sus propias bibliotecas para reutilizarlas.
- Aumentar el rendimiento en ciertos casos, en especial los de mayor exigencia computacional, como el de los juegos.

Para el desarrollo del sistema se utiliza el NDK 14 para poder utilizar la librería de reconocimiento de caracteres tess-two, que está escrita en lenguaje C++.

1.7.6 Python 2.7

Python es un lenguaje de programación interpretado que incluye programación orientada a objetos y ofrece una manera sencilla de desarrollar programas con componentes reutilizables. Elimina preocupaciones referentes a detalles de bajo nivel, como manejar la memoria empleada por el programa, permitiendo a los desarrolladores centrarse más en el código que en la sintaxis. La versión 2.7 de Python es utilizada en el desarrollo del sistema GRHS y se usará en la implementación de las modificaciones necesarias en el código fuente de Gserver(24).

1.7.7 Django 1.4

Django es un framework de aplicación de alto nivel, que fomenta el desarrollo rápido y el diseño limpio y pragmático. Se encarga de gran parte del desarrollo web, para que el desarrollador

GDROID

pueda centrarse en la escritura de su aplicación. Es gratuito y de código abierto, con una amplia comunidad de desarrollo y documentación detallada(25).

1.7.8 PyCharm 16.3.3

Es un IDE multiplataforma utilizado para desarrollar en el lenguaje de programación Python. Proporciona análisis de código, depuración gráfica, integración con VCS / DVCS y soporte para el desarrollo web con Django, entre otras bondades. PyCharm es desarrollado por la empresa JetBrains y debido a la naturaleza de sus licencias tiene dos versiones, la Community que es gratuita y orientada a la educación y al desarrollo puro en Python y la Professional, que incluye más características como el soporte a desarrollo web(26).

1.7 Conclusiones del capítulo

En el presente capítulo fueron estudiadas las herramientas informáticas que permiten el escaneo y tratamiento de imágenes, así como la muestra de información utilizando gráficas en dispositivos móviles sobre el sistema operativo Android lo que permitió la selección de herramientas adecuadas a utilizar para el desarrollo de las funcionalidades de GDroid. Fueron abordados los principales conceptos relacionados con la investigación para un mayor entendimiento de dicha investigación. Se expusieron las características por lo que fue escogida la metodología XP como guía del proyecto y las herramientas seleccionadas para el desarrollo del sistema.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN, EXPLORACIÓN Y PLANIFICACIÓN DEL SISTEMA GDROID.

2.1 Introducción

El objetivo fundamental del presente capítulo es el diseño de la propuesta de solución. En el mismo se realiza un análisis de las características del sistema a desarrollar, así como los resultados obtenidos en las fases de Exploración, Planificación y Diseño correspondientes a la metodología de desarrollo seleccionada.

2.2 Modelo de propuesta de solución.

La solución propuesta está elaborada para las entidades que hagan uso del sistema de GRHS, sobre dispositivos móviles que operen con el Sistema Operativo Android. Persigue como objetivo, las operaciones escaneo y la muestra de los datos alojados en la imagen de una ficha técnica y facilitar la información estadística, desde cualquier punto geográfico dentro del alcance de la red inalámbrica de la entidad. La solución debe permitir el control del estado de los recursos de hardware y software, así como del cumplimiento de las políticas de seguridad informática establecidas.



Ilustración 5: Propuesta de solución

Las solicitudes de información de la aplicación móvil GDroid al servidor será realizada por medio del protocolo de conexión segura HTTPS.

GDROID

Los datos obtenidos desde el servidor podrán ser mostrados aun cuando no exista conexión debido a funciones de sincronización implementadas, lo que permite la disponibilidad de los datos en todo momento.

2.2.1 Escanear imagen

GDroid será capaz de escanear el inventario (MB) de la ficha técnica de un ordenador, para recibir y mostrar información detallada de los detalles técnicos que contiene el ordenador en la base de datos remota de Gserver. Para ello GDroid permitirá que el usuario tome una foto del MB de una ficha técnica, comprueba que la foto se haya tomado adecuadamente, GDroid le da la opción al usuario de seleccionar el fragmento de imagen que desea escanear para evitar ruido innecesario a la hora de realizar el reconocimiento de datos. Por medio de las librerías Tess-two y Leptonica, se realiza el reconocimiento de datos y muestra un posible resultado. El cliente podrá comprobar si el MB que se le muestra es igual al escaneado, de no ser igual, se le da la opción de rectificarlo manualmente. Una vez el cliente comprueba que ambos MB son iguales, presiona el botón sincronizar, GDroid realiza una sincronización con Gserver y le muestra al usuario los datos que presenta del ordenador al que pertenece el MB escaneado.

2.2.2 Mostrar datos estadísticos mediante el uso de gráficas

GDroid mostrará la información estadística de los recursos almacenados en Gserver mediante el uso de gráficos de barras. Para ello se sincronizará con Gserver y obtendrá la información estadística. El usuario podrá seleccionar uno de las cuatro categorías existentes, Agentes, Acciones, Inventarios e Incidencias y mostrará mediante el uso de la librería MPAndroidchart los datos estadísticos mediante gráficos de barras. En caso de no tener red, se mostrará la información estadística recogida de la última sincronización con Gserver.

2.3 Características no funcionales del sistema.

Se muestran las principales características no funcionales del sistema con las que debe cumplir el producto para su correcto funcionamiento.

2.3.1 Interfaz de usuario.

La interfaz deberá ser entendible para el usuario, dirigida a los usuarios que se relacionen con el sistema GRHS.

2.3.2 Consistencia de los datos.

Los datos mostrados en GDroid mantendrán la consistencia con los datos almacenados en la base de datos de GRHS, esto quiere decir que, al escanear el MB de una ficha técnica, la aplicación verificará si el mismo ya existe en la base de datos local y lo actualizará si ha sufrido

GDROID

algún cambio. Esto evitará la existencia de datos duplicados y la realización de inserciones innecesarias.

2.3.3 Usabilidad.

Debe ser de fácil manejo y comprensión para el usuario que interactúe con el sistema GRHS.

2.3.4 Software.

El dispositivo móvil debe contar con sistema operativo Android, con una versión igual o superior a 4.0 y 256 Mb (megabyte) de RAM (memoria de acceso aleatorio) o más.

2.3.5 Hardware.

El dispositivo móvil debe contar con una cámara, y permitir la conexión por Wi-Fi, además, debe contar con un espacio interno de 42 Mb para el almacenamiento de la aplicación y un espacio mínimo de 227 kB (kilobyte), para realizar al menos una sincronización con el servidor, para el almacenamiento de la base datos local.

2.3.6 Seguridad de la información.

La comunicación entre GDroid y Gserver debe realizarse por medio del protocolo de conexión segura HTTPS. GDroid y Gserver por medio de HTTPS se comunican de forma segura garantizando la confidencialidad, integridad y disponibilidad de la información.

- Confidencialidad: La aplicación móvil GDroid asegura que el acceso a la información está adecuadamente autorizado. Solo tienen acceso los usuarios autorizados.
- Integridad: La aplicación móvil GDroid salvaguarda la precisión y completitud de la información y sus métodos de proceso.
- Disponibilidad: La aplicación móvil GDroid asegura que los usuarios autorizados pueden acceder a la información cuando la necesitan.

2.4 Fase de exploración.

Se definen las historias de usuario y se estiman los tiempos de desarrollos en base a esta información, los cuales podrían cambiar cuando se analicen con más detalles en cada iteración.

2.4.1 Historias de usuarios.

Las historias de usuarios (HU) son escritas por el cliente en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. Deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Las HU pueden ser programadas entre 1 y 3 semanas, las HU que sus

GDROID

estimaciones superan las 3 semanas deben ser divididas en 2 historias de usuarios, así como las que no alcanza una semana se debe combinar con otra(27).

Las HU se representan de la siguiente forma:

Número: Número de la HU.

Nombre de la HU: Para identificar la HU.

Iteración asignada: Número de la iteración asignada.

Prioridad en negocio:

- Alta: Constituyen funcionalidades principales del sistema, o forman parte esencial de la arquitectura del mismo.
- Media: Son funcionalidades importantes y de gran valor para el usuario pero que no impiden poner el proyecto en marcha si no se tienen.
- Baja: Son funcionalidades que sería deseable tener y podrían incluirse en caso de que hubiese recursos para ello.

Riesgo en desarrollo.

- Alto: En caso de tener algún error de implementación, pueden afectar la disponibilidad del sistema.
- Medio: En caso de presentar errores retrasan la entrega de la versión.
- Bajo: En caso de presentar errores, estos pueden ser tratados con facilidad y no afectan el desarrollo del proyecto.

Puntos estimados: Tiempo estimado para el desarrollo de la HU.

Descripción: Breve descripción de la HU.

Observaciones: Condiciones a tener en cuenta para desarrollar correctamente la HU.

Prototipo de Interfaz: Prototipo de interfaz.

Para el desarrollo de la solución propuesta se identificaron las siguientes HU detalladas a continuación:

HU- 1: Tratamiento de imagen.

HU- 2: Mostrar resultados de la sincronización con Gserver.

GDROID

HU- 3: Visualización de la información estadística.

Quedaron definidas como se muestran a continuación las historias de usuario que se realizaron para el desarrollo del trabajo.

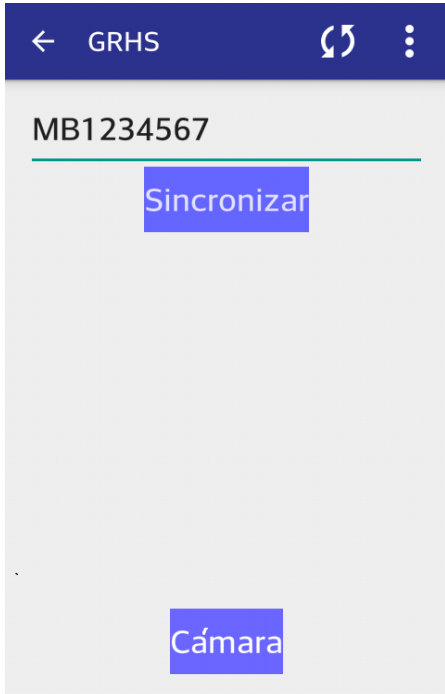
HISTORIA DE USUARIO	
Número: 1	Nombre: Tratamiento de imagen.
Usuario: Administrador.	
Prioridad del Negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador: Yoán Jorge Nazábal Gómez. Luis Ángel Roche Broche	
Descripción: El usuario coloca la cámara del dispositivo móvil sobre la ficha técnica para que este identifique la imagen que será procesada, en caso de que la foto no sea tomada correctamente se debe notificar para que la imagen sea tomada nuevamente. Se convierte en datos la imagen tomada que contiene el MB de la ficha técnica.	
Observaciones: <ul style="list-style-type: none">• Es necesario que la imagen se encuentre legible y que contenga todos los caracteres.	
Prototipo de Interfaz: 	

Tabla 1 HU-1 Tratamiento de imagen.

HISTORIA DE USUARIO

GDROID


Número: 2	Nombre: Mostrar resultados de la sincronización con Gserver.
Usuario: Administrador.	
Prioridad del Negocio: Media	Riesgo de desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 2
Programador: Yoán Jorge Nazábal Gómez. Luis Ángel Roche Broche	
Descripción: Se muestran al usuario los datos (Local, Puesto, Inventario, Serie, Motherboard (Tarjeta madre), Microprocesador, Memoria RAM, Disco Duro y Unidades Ópticas), contenidos en la base de datos remotas de Gserver, del ordenador que su MB coincida con el MB de la imagen tomada. De no existir conexión entre GDroid y Gserver, se muestra una notificación con el texto: "No existe conexión".	
Observaciones: <ul style="list-style-type: none">• Es necesario que haya conexión WIFI, entre GDroid y Gserver.• Es necesario que el MB obtenido de la imagen coincida con el MB de un ordenador contenido en la base de datos remota de Gserver.	
Prototipo de Interfaz:	
	

Tabla 2: HU-2 Mostrar resultados de la sincronización con Gserver.

HISTORIA DE USUARIO

GDROID

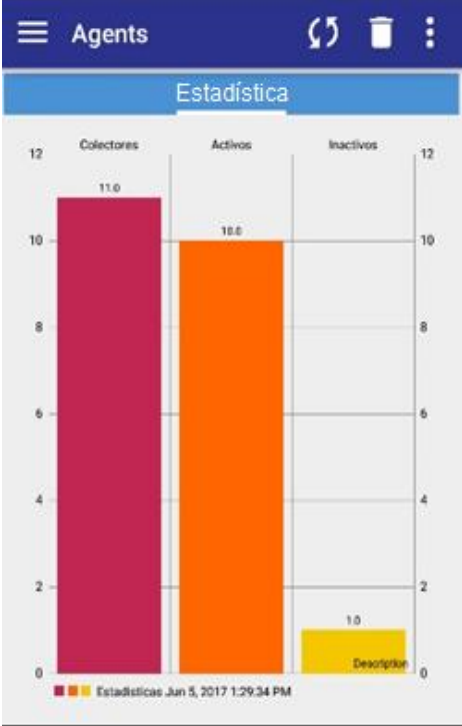
Número: 3	Nombre: Visualización de la información estadística.
Usuario: Administrador.	
Prioridad del Negocio: Media	Riesgo de desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 3
Programador: Yoán Jorge Nazábal Gómez. Luis Ángel Roche Broche	
Descripción: Se realiza una visualización de la información estadística, contenida en la base datos remota de Gserver, por medio de gráficas de barras.	
Observaciones: <ul style="list-style-type: none">• Es necesario que exista conexión WI-FI entre GDroid y Gserver.	
Prototipo de Interfaz:	
	

Tabla 3: HU-3 Visualización de la información estadística.

2.5 Fase de planificación.

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase

GDROID

de planificación toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología(16).

2.5.1 Estimación del esfuerzo por HU.

Para el desarrollo del sistema se realizó una estimación del esfuerzo para cada una de las HU identificadas, utilizando como medida el punto. Un punto equivale a una semana ideal de programación.

Historias de Usuario	Puntos Estimados
Tratamiento de imagen.	3
Mostrar resultados de la sincronización con Gserver.	3
Visualización de información estadística.	3

Tabla 4: Estimación de esfuerzo por Historia de Usuario.

2.5.2 Plan de Iteraciones.

Las HU seleccionada para cada entrega son desarrolladas y probadas en un ciclo de iteración de acuerdo al orden preestablecido. En base a lo antes mencionado, quedan conformadas las siguientes iteraciones.

Iteración 1.

Esta iteración tiene como objetivo la implementación de la historia de usuario 1, la cual es de prioridad alta y es esencial para el desarrollo del sistema. Al finalizar la misma se contará con una primera versión de prueba, la cual será mostrada al cliente.

Iteración 2.

Esta iteración tiene como objetivo la implementación de la historia de usuario 2, la cual es de prioridad media y es importante para el desarrollo de sistema. Al finalizar la iteración se contará con una versión funcional del sistema, donde se podrán probar las funcionalidades relacionadas con el tratamiento de datos.

Iteración 3.

Esta iteración tiene como objetivo la implementación de la historia de usuario 3, la cual es de prioridad media y es importante para el desarrollo del sistema. Al finalizar la iteración se contará con una primera versión del producto final, adicionando a las funcionalidades anteriores la representación gráfica de datos estadísticos. Como resultado de esta el sistema estará listo para su despliegue.

GDROID

2.5.3 Plan de duración de Iteraciones.

El plan de duración de iteraciones muestra el tiempo que demora cada iteración.

Iteración	Historia de Usuario a Implementar	Duración de la Iteración
Iteración 1	Tratamiento de imagen.	3 semanas
Iteración 2	Mostrar resultados de la sincronización con Gserver.	3 semanas
Iteración 3	Visualización de información estadística.	3 semanas

Tabla 5 Plan de duración de las iteraciones

2.5.4 Plan de entrega.

El plan de entrega consiste en establecer de manera conjunta entre el equipo de desarrollo y el cliente, la duración y la fecha de entrega de cada iteración para lograr el producto final. El proyecto empezará a desarrollarse el 06/03/2017

Iteración	Fecha de entrega
Iteración 1	24/03/2017
Iteración 2	14/04/2017
Iteración 3	05/05/2017

Tabla 6: Plan de entrega de las iteraciones.

2.6 Conclusiones del capítulo.

En el presente capítulo se analizó de forma detallada el funcionamiento del sistema propuesto. Con la culminación de este capítulo fue posible especificar las fases de exploración y planificación que propone la metodología XP, luego de definidas las HU, la construcción del plan de iteraciones, permitió conocer cuáles de ellas implementar en cada iteración y el orden de prioridad de cada una, además la confección del plan de entregas permitió conocer la fecha estimada en que se entregará una primera versión del producto al cliente.

CAPÍTULO 3. DISEÑO, IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA.

3.1 Introducción.

En el presente capítulo se describe el proceso de diseño de GDroid y los patrones que se utilizarán. Se expondrán las tarjetas CRC y las tareas de ingeniería como artefactos de la metodología XP. Se realizarán pruebas para evaluar la calidad de los resultados obtenidos, con el fin de obtener un producto satisfactorio.

3.2 Diseño.

A continuación, se describe la arquitectura de software, los patrones de diseños aplicados a la propuesta de solución y la descripción de las principales clases.

3.2.1 Arquitectura de software.

La Arquitectura de Software se refiere a las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos(28). Para el desarrollo de la presente aplicación móvil se utiliza la arquitectura de software Cliente/Servidor.

3.2.2 Arquitectura Cliente/Servidor:

Es un modelo para el desarrollo de sistemas distribuidos. Los clientes o programas que representan entidades que necesitan servicios y los servidores o programas que proporcionan servicios son objetos separados desde un punto de vista lógico y que se comunican a través de una red de comunicaciones para realizar una o varias tareas de forma conjunta(29). El proceso llamado Servidor se realiza la mayoría del trabajo mientras que, el proceso llamado Cliente se ocupa con de la interacción con el usuario. La forma más simple del modelo Cliente/Servidor consiste en una aplicación servidor a la que se accede directamente por varios clientes, a la cual se refiere como un estilo arquitectónico de 2 niveles(29). Para el desarrollo de la presente aplicación móvil se utilizó la arquitectura Cliente/Servidor porque GRHS cuenta con un nivel servidor llamado Gserver y la aplicación móvil GDroid ocuparía el nivel cliente.

GDROID

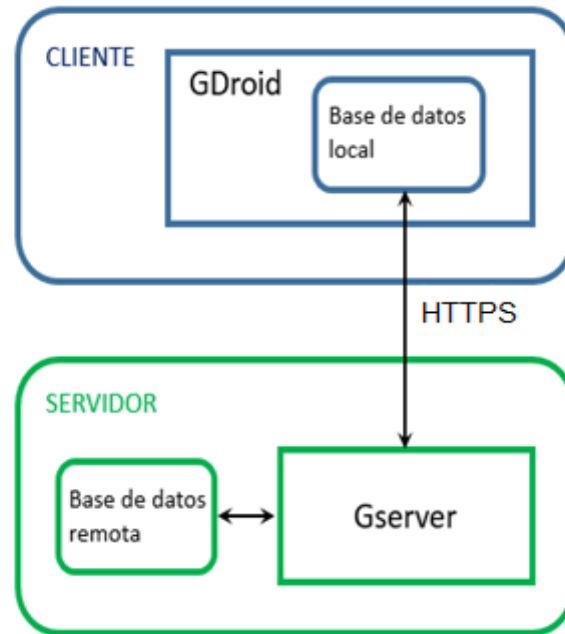


Ilustración 6: Arquitectura Cliente/Servidor desglosada en sus 2 niveles.

La aplicación móvil GDroid ocupa el nivel Cliente, ya que es la encargada de interactuar directamente con el usuario. El modelo solución plantea que múltiples instancias de GDroid solicitan información a Gserver, aplicación principal del nivel Servidor.

3.2.3 Modelo-Vista-Presentación (MVP)

Modelo-Vista-Presentación (MVP) es un patrón de arquitectura derivado del Modelo-Vista-Controlador (MVC). Permite separar los datos internos del *modelo* de una vista pasiva y enlazarlos median la *presentación* que maneja toda la lógica de la aplicación(30).

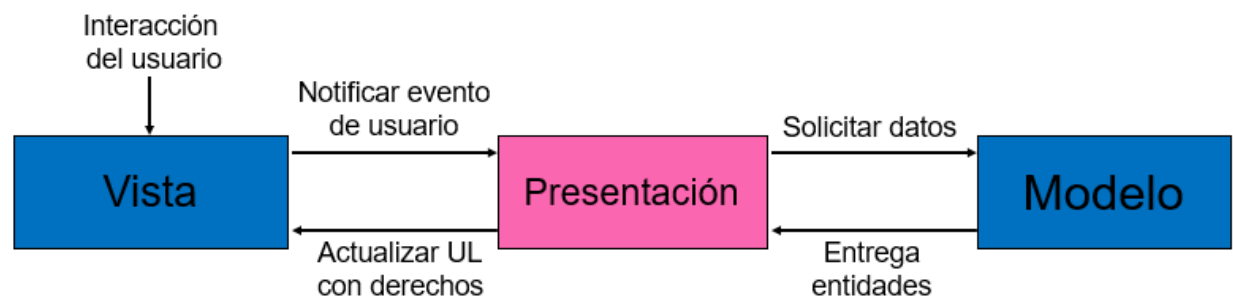


Ilustración 7 Diseño de MVP

MVP posee tres capas.

GDROID

- **Modelo:** Almacena y proporciona los datos internos.
- **Vista:** Maneja la visualización de los datos del *modelo*. Propaga las acciones del usuario a la *presentación*.
- **Presentación:** Enlaza las capas de *modelo* y *vista*. Sincroniza los datos mostrados en la *vista* con los almacenados en el *modelo* y actúa ante los eventos de usuario propagados por la *vista*.

La aplicación GDroid para su desarrollo utiliza el patrón arquitectónico MVP. Para dar solución a la funcionalidad "Escanear imagen" GDroid presenta la siguiente distribución de clases. Las clases `Data_Detail.xml` y `Scan_Activity.xml` pertenecen a la capa *vista*, la clase `DtaBaseContenProvider` pertenece a la capa *modelo* y las clases `ScanActivity` y `ScanResult` pertenecen a la capa *presentación*.

Para dar solución a la funcionalidad "Mostrar datos estadísticos mediante el uso de gráficas" GDroid presenta la siguiente distribución de clases. La clase `Activity_Graph_Fragment.xml` pertenece a la capa *vista*, la clase `DtaBaseContenProvider` pertenece a la capa *modelo* y las clases `IncidencesStatisticFragment`, `InventoryStatisticFragment`, `AgentStatisticFragment` y `ActionStatisticFragmen` pertenecen a la capa *presentación*.

3.2.4 Patrón de diseño.

Los patrones de diseño son soluciones para problemas típicos y recurrentes que se pueden encontrar a la hora de desarrollar una aplicación. Por tanto, un patrón de diseño es una solución demostrada, eficiente y reutilizable en diferentes contextos y escenarios en desarrollo de software(31).

Patrones GRASP

Los patrones GRASP (General Responsibility Assignment Software Patterns) describen los principios fundamentales del diseño de objeto y la asignación de responsabilidades, expresados como patrones(31). Es un sistema orientado a objetos, se compone de objetos que envíen mensajes a otros objetos para que lleven a cabo las operaciones requeridas.

Experto

El patrón Experto establece el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto, o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. El comportamiento se distribuye entre las clases que cuentan con la información requerida(31).

GDROID

Las clases ScanActivity y ScanResult contienen la información para cumplir con sus responsabilidades.

Creador

Es el encargado de guiar la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento(31). Las clases que tienen la responsabilidad de crear objetos contienen toda la información necesaria para crear los mismos.

Las clases ScanActivity, ScanResult, ListAPITask, DataBaseContractProvide crean objetos para obtener información necesaria para cumplir sus responsabilidades.

Alta Cohesión

Es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase, caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Esto presenta como problema que son difíciles de comprender, de reutilizar, conservar y las afectan constantemente los cambios(31). Se aplica en la mayoría de las clases del diseño, ya que en cada una solo se implementan las funcionalidades que le corresponden.

Las clases DataBaseContractProvide, SyncAdapter, ScanActivity, ScanResult, ListAPITask, ActionStatisticFragmen, InventoryStatisticFragment, IncidencesStatisticFragment, AgentStatisticFragment, permiten incrementar la claridad y facilitan la comprensión del diseño.

Bajo Acoplamiento

Este patrón estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecienten la oportunidad de una mayor productividad. Los cambios de las clases afines ocasionan cambios locales. Con el uso de este patrón los componentes no se afectan por cambios de otros componentes, son fáciles de entender por separado y fáciles de reutilizar(31). Se aplica en la mayoría de las clases del diseño, ya que cada clase se comunica con un número pequeño de clases.

La clase DataBaseContractProvide no afecta los cambios en otras clases y se reutilizan en varias funcionalidades.

GDROID

3.2.5 Patrón Maestro/Esclavo.

El patrón de diseño Maestro/Esclavo es otra arquitectura usada cuando se tienen dos o más procesos que necesitan ejecutarse simultánea y continuamente, pero a diferentes velocidades. Si estos procesos corren en un único bucle, pueden suceder problemas de temporización severos. Estos problemas de temporización ocurren cuando una parte del bucle tarda más en ejecutarse de lo esperado(32).

Para el desarrollo del presente trabajo se utilizan 2 hilos de ejecución. Uno de estos hilos es de la interfaz visual y el otro es el de sincronización con los datos, que se encarga de la lógica de los datos del servidor. El hilo maestro es el de la interfaz visual, ya que es el encargado de controlar el hilo de sincronización. Este proceso se utiliza en las clases ScanActivity, ActionStatisticFragmen, InventoryStatisticFragment, IncidencesStatisticFragment, AgentStatisticFragment. La comunicación de los hilos se realiza por medio de intents (es un objeto de mensajería que se utiliza para solicitar acciones entre componentes de una aplicación Android).

3.2.6 Tarjetas CRC (Clases-Responsabilidad-Colaborador).

Para el diseño de las aplicaciones, la metodología XP usa técnicas como las tarjetas CRC. Esta técnica se usa para guiar el sistema a través de análisis guiados por la responsabilidad donde las clases se examinan, se filtran y se refinan en base a sus responsabilidades con respecto al sistema. El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda, y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente(33).

A continuación, se muestra la tarjeta CRC perteneciente a la clase ActionStatisticFragmen, las restantes se encuentran en el [Anexo I](#).

Tarjeta CRC	
Clase: ActionStatisticFragmen	
Responsabilidades	Colaboradores
Genera una gráfica de barra para las Acciones, a partir de los datos obtenidos de la base de datos local.	<ul style="list-style-type: none">• DataBaseContractProvide• XYPlot

Tabla 7: Tarjeta CRC para la clase ActionStatisticFragmen

GDROID

3.2.7 Tareas de Ingeniería.

Las tareas de ingeniería son escritas por el equipo de desarrollo a partir de las historias de usuarios. Cada una de estas tareas describe a las historias de usuarios de una forma más detallada para su implementación, estimando un tiempo más aproximado a la realidad para el desarrollo de cada una de ellas(27).

Se muestra una tabla con las tareas de ingeniería definidas por cada historia de usuario.

Iteración:	Historia de Usuario:	Tarea de Ingeniería:
Iteración I	HU-1: Tratamiento de imagen.	TI-1: Captura de la imagen.
		TI-2: Transformación de imagen a texto digital.
		TI-3: Organización de los caracteres.
		TI-4: Transformar caracteres en MB.
Iteración II	HU-2: Mostrar resultados de sincronización con Gserver.	TI-5: Recibir el MB de la imagen.
		TI-6: Buscar MB en la base de datos.
		TI-7: Obtener datos del ordenador.
		TI-8: Mostrar datos al usuario.
Iteración III	HU-3: Visualización de la información estadística.	TI-9: Obtener datos estadísticos de la base de datos remota de Gserver.
		TI-10: Mostrar datos estadísticos de los agentes.
		TI-11: Mostrar datos estadísticos de los inventarios.
		TI-12: Mostrar datos estadísticos de las incidencias.
		TI-13: Mostrar datos estadísticos de las acciones.

Tabla 8: Tareas de Ingeniería

GDROID

Las tareas de la ingeniería serán representadas mediante tablas divididas por las siguientes secciones:

- **Número:** Número de la tarea.
- **Historia de Usuario:** Historia de usuario a la que pertenece la tarea.
- **Nombre de la Tarea:** Nombre que identifica la tarea.
- **Tipo de Tarea:** Las tareas pueden ser de tipo, Desarrollo, Corrección, Mejora.
- **Puntos Estimados:** Tiempo estimado para el desarrollo, cada punto equivale a una semana de desarrollo.
- **Fecha de Inicio:** Fecha en que comienza el desarrollo de la tarea.
- **Fecha de Fin:** Fecha en que termina el desarrollo de la tarea.
- **Programador Responsable:** Nombre del programador.
- **Descripción:** Breve descripción de la tarea.

Se muestran todas las tareas de ingeniería que se realizaron para el desarrollo del trabajo.

TAREA DE INGENIERÍA	
Número: 1	Historia de usuario: Tratamiento de imagen.
Nombre de la tarea: Captura de la imagen.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 06/03/2017	Fecha fin: 06/03/2017
Programador responsable: Yoán J. Nazábal Gómez Luis Ángel Roche Broche.	
Descripción: Permitirá digitalizar la imagen del MB de la ficha técnica. Se debe mostrar un botón que permita activar la cámara, para obtener la imagen. Se tomará una foto bien legible del MB de la ficha técnica de la máquina, abarcando todos los caracteres contenidos. Permitirá al usuario la opción de manipular la imagen realizada.	

Tabla 9: TI-1 Captura de la Imagen

TAREA DE INGENIERÍA	
Número: 2	Historia de usuario: Tratamiento de imagen.
Nombre de la tarea: Transformación de imagen a texto digital.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.8
Fecha de inicio: 07/03/2017	Fecha fin: 10/03/2017
Programador responsable: Yoán J. Nazábal Gómez Luis Ángel Roche Broche.	
Descripción:	

GDROID

Permitirá procesar los caracteres presentes en la imagen del MB de la ficha técnica, para que luego sean ordenados.

Tabla 10: TI-2 Transformación de imagen a texto digital.

TAREA DE INGENIERÍA	
Número: 3	Historia de usuario: Tratamiento de imagen.
Nombre de la tarea: Organización de los caracteres.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 13/03/2017	Fecha fin: 17/03/2017
Programador responsable: Yoán J. Nazábal Gómez Luis Ángel Roche Broche.	
Descripción: Permitirá ordenar los caracteres obtenidos de la imagen procesada anteriormente y eliminará la imagen del MB de la ficha técnica contenida en la base de datos local de GDroid.	

Tabla 11: TI-3 Organización de los caracteres

TAREA DE INGENIERÍA	
Número: 4	Historia de usuario: Tratamiento de imagen.
Nombre de la tarea: Transformar caracteres en MB.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 20/03/2017	Fecha fin: 24/03/2017
Programador responsable: Yoán J. Nazábal Gómez Luis Ángel Roche Broche.	
Descripción: Permitirá transformar los caracteres en un MB digital, para que sea comparado posteriormente.	

Tabla 12: TI-4 Transformar caracteres en MB

TAREA DE INGENIERÍA	
Número: 5	Historia de usuario: Mostrar resultados de la sincronización con Gserver.
Nombre de la tarea: Recibir el MB de la imagen.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 27/03/2017	Fecha fin: 28/03/2017
Programador responsable: Yoán J. Nazábal Gómez Luis Ángel Roche Broche.	
Descripción: Permitirá recibir el MB digitalizado de la imagen realizada.	

Tabla 13: TI-5 Recibir el MB de la imagen

TAREA DE INGENIERÍA

GDROID

Número: 6	Historia de usuario: Mostrar resultados de la sincronización con Gserver.
Nombre de la tarea: Buscar MB en la base de datos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 29/03/2017	Fecha fin: 04/04/2017
Programador responsable: Yoán J. Nazábal Gómez Luis Ángel Roche Broche.	
Descripción: Permitirá, primero buscar en la base de datos local de GDroid el ordenador cuyo MB coincida con el MB digital obtenido de la imagen. De no encontrarlo, permitirá buscar en la base de datos remota de Gserver el ordenador cuyo MB coincida con el MB digital obtenido de la imagen. Deberá existir conexión entre Gserver y GDroid por medio del protocolo de conexión segura HTTPS. En caso de no existir conexión se mostrará una notificación con el texto "No hay conexión". En caso de que no se encuentre ningún ordenador se mostrará una notificación con el texto "No hay coincidencias".	

Tabla 14: TI-6 Buscar MB en la base de datos

TAREA DE INGENIERÍA	
Número: 7	Historia de usuario: Mostrar resultados de la sincronización con Gserver.
Nombre de la tarea: Obtener datos del ordenador.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 05/04/2017	Fecha fin: 07/04/2017
Programador responsable: Yoán J. Nazábal Gómez Luis Ángel Roche Broche.	
Descripción: Permitirá obtener los datos (Local, Puesto, Inventario, Serie, Motherboard (Tarjeta madre), Microprocesador, Memoria RAM, Disco Duro y Unidades Ópticas) del ordenador, contenido en la base de datos remota de Gserver, cuyo MB coincide con el MB obtenido de la imagen. Deberá existir conexión entre Gserver y GDroid por medio del protocolo de conexión segura HTTPS.	

Tabla 15: TI-7 Obtener datos del ordenador

TAREA DE INGENIERÍA	
Número: 8	Historia de usuario: Mostrar resultados de la sincronización con Gserver.
Nombre de la tarea: Mostrar información al usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 10/04/2017	Fecha fin: 14/04/2017

GDROID

Programador responsable: Yoán J. Nazábal Gómez

Luis Ángel Roche Broche.

Descripción:

Permitirá mostrar al usuario los datos (Local, Puesto, Inventario, Serie, Motherboard (Tarjeta madre), Microprocesador, Memoria RAM, Disco Duro y Unidades Ópticas) del ordenador, encontrado. Deberá existir conexión entre Gserver y GDroid por medio del protocolo de conexión segura HTTPS. En caso de no existir conexión se mostrará una notificación con el texto "No hay conexión". En caso de que no se encuentre ningún ordenador se mostrará una notificación con el texto "No hay coincidencias".

Tabla 16: TI-8 Mostrar información al usuario

TAREA DE INGENIERÍA	
Número: 9	Historia de usuario: Visualización de la información estadística.
Nombre de la tarea: Obtener datos estadísticos de la base de datos remota de Gserver.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 17/04/2017	Fecha fin: 19/04/2017
Programador responsable: Yoán J. Nazábal Gómez Luis Ángel Roche Broche.	
Descripción: Se debe mostrar la información estadística en forma de gráfica de barras. Permitirá recibir los datos estadísticos contenidos en la base de datos remota de Gserver.	

Tabla 17: TI-9 Obtener datos estadísticos de la base de datos remota de Gserver

TAREA DE INGENIERÍA	
Número: 10	Historia de usuario: Visualización de la información estadística.
Nombre de la tarea: Mostrar datos estadísticos de los agentes.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 20/04/2017	Fecha fin: 24/04/2017
Programador responsable: Yoán J. Nazábal Gómez Luis Ángel Roche Broche.	
Descripción: Se debe mostrar la información estadística de los agentes en forma de gráficos de barras. Permitirá mostrar mediante un gráfico de barras, los agentes que se encuentran en la base de datos remota de Gserver.	

Tabla 18: TI-10 Mostrar datos estadísticos de los agentes

TAREA DE INGENIERÍA

GDROID

Número: 11	Historia de usuario: Visualización de la información estadística.
Nombre de la tarea: Mostrar datos estadísticos de los inventarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 25/04/2017	Fecha fin: 27/04/2017
Programador responsable: Yoán J. Nazábal Gómez Luis Ángel Roche Broche.	
Descripción: Se debe mostrar la información estadística de los inventarios en forma de gráfico barras. Permitirá mostrar mediante un gráfico de barras, los inventarios que se encuentran en la base de datos remota de Gserver.	

Tabla 19: TI-11 Mostrar datos estadísticos de los inventarios

TAREA DE INGENIERÍA	
Número: 12	Historia de usuario: Visualización de la información estadística.
Nombre de la tarea: Mostrar datos estadísticos de las incidencias.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 28/04/2017	Fecha fin: 02/05/2017
Programador responsable: Yoán J. Nazábal Gómez Luis Ángel Roche Broche.	
Descripción: Se debe mostrar la información estadística de las incidencias en forma de gráfico de barras. Permitirá mostrar mediante un gráfico de barras, las incidencias que se encuentran en la base de datos remota de Gserver.	

Tabla 20: TI-12 Mostrar datos estadísticos de las incidencias

TAREA DE INGENIERÍA	
Número: 13	Historia de usuario: Visualización de la información estadística.
Nombre de la tarea: Mostrar datos estadísticos de las acciones.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 03/05/2017	Fecha fin: 05/05/2017
Programador responsable: Yoán J. Nazábal Gómez Luis Ángel Roche Broche.	
Descripción:	

GDROID

Se debe mostrar la información estadística de las acciones en forma de gráfico de barras. Permitirá mostrar mediante un gráfico de barras, las acciones que se encuentran en la base de datos remota de Gserver.

Tabla 21: TI-13 Mostrar datos estadísticos de las acciones

3.3 Pruebas.

Las pruebas son una de las prácticas fundamentales en las cuales se basa XP. Esta actividad se realiza en forma continua a lo largo del proyecto. Existen dos tipos de pruebas, las unitarias y las de aceptación.

Las pruebas unitarias son definidas por los programadores antes de comenzar a escribir códigos. Éstas deben contemplar cada módulo del sistema que pueda generar fallas. Para poder integrar el código realizado al ya existente, el mismo debe aprobar satisfactoriamente todos los casos de pruebas definidos.

El cliente define las pruebas de aceptación para cada historia de usuario a principio de cada iteración. Las pruebas de aceptación se utilizan para validar que cada requerimiento implementado funciona como se había especificado(34).

3.3.1 Pruebas Unitarias:

Una prueba unitaria es la verificación de un módulo (unidad de código) determinado dentro de un sistema. Las pruebas unitarias aseguran que un determinado módulo cumpla con un comportamiento esperado en forma aislada antes de ser integrado al sistema. En XP los programadores deben escribir las pruebas unitarias para cada módulo antes de escribir el código(34).

Luego de escribir el código, los programadores ejecutan las pruebas, las cuales deben resultar 100% efectivas para que el código pueda integrarse al sistema. En caso contrario hay que solucionar los errores y ejecutar nuevamente los casos de prueba hasta lograr que ninguno de ellos(35).

El proceso de pruebas del código de GDroid se realizó mediante el uso de las pruebas de caja blanca "Camino básico".

La prueba de la caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de prueba.

Las pruebas de caja blanca intentan garantizar que:

- Se ejecutan al menos una vez todos los caminos independientes de cada módulo.

GDROID

- Se utilizan las decisiones en su parte verdadera y en su parte falsa.
- Se ejecuten todos los bucles en sus límites.
- Se utilizan todas las estructuras de datos internas.

Prueba del camino básico: El método del camino básico (propuesto por McCabe) permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez.

A continuación, se detallan los resultados obtenidos de las pruebas de camino básico. Las pruebas restantes se encuentran en el Anexo II.

Prueba de Caja Blanca: Camino Básico	
Probado por Luis Ángel Roche Broche	
<pre>public void AddValuesToBARENTRY(ArrayList<StatsListItem> values){ if(values.size()==0) { 1 BARENTRY.add(new BarEntry(1, 0)); 2 BARENTRY.add(new BarEntry(1, 1)); BARENTRY.add(new BarEntry(1, 2)); }else { for (int i = 2; i < 5; i++) { 3 if (!values.get(i).percent.equals("")) {4 String val = values.get(i).percent; 5 BARENTRY.add(new BarEntry(Float.parseFloat(val), i - 2)); } else { BARENTRY.add(new BarEntry(Float.parseFloat(values.get(i).description), i - 2));6 } 8 }</pre>	<pre>graph TD 1((1)) --> 2((2)) 1((1)) --> 3((3)) 2((2)) --> 5((5)) 2((2)) --> 8((8)) 3((3)) --> 4((4)) 3((3)) --> 6((6)) 4((4)) --> 5((5)) 4((4)) --> 6((6)) 5((5)) --> 8((8)) 6((6)) --> 8((8))</pre>
Complejidad Ciclomática: $V(G) = (\text{Cantidad de aristas} - \text{Cantidad de nodos}) + 2 = (9 - 8) + 2 = 3$	Caminos Básicos: (1-2-8) (1-3-4-5-8) (1-2-4-6-8)

Tabla 22: PU CB Método AddValuesToBARENTRY

A continuación, se detallan los resultados obtenidos de las pruebas de camino básico durante las 3 iteraciones realizadas. Se seleccionaron 9 fragmentos de código, 4 de la 1ra iteración, 2 de la 2da iteración y 3 de la 3ra iteración, detectándose 0 errores, por lo que se le da fin al proceso de pruebas del código.

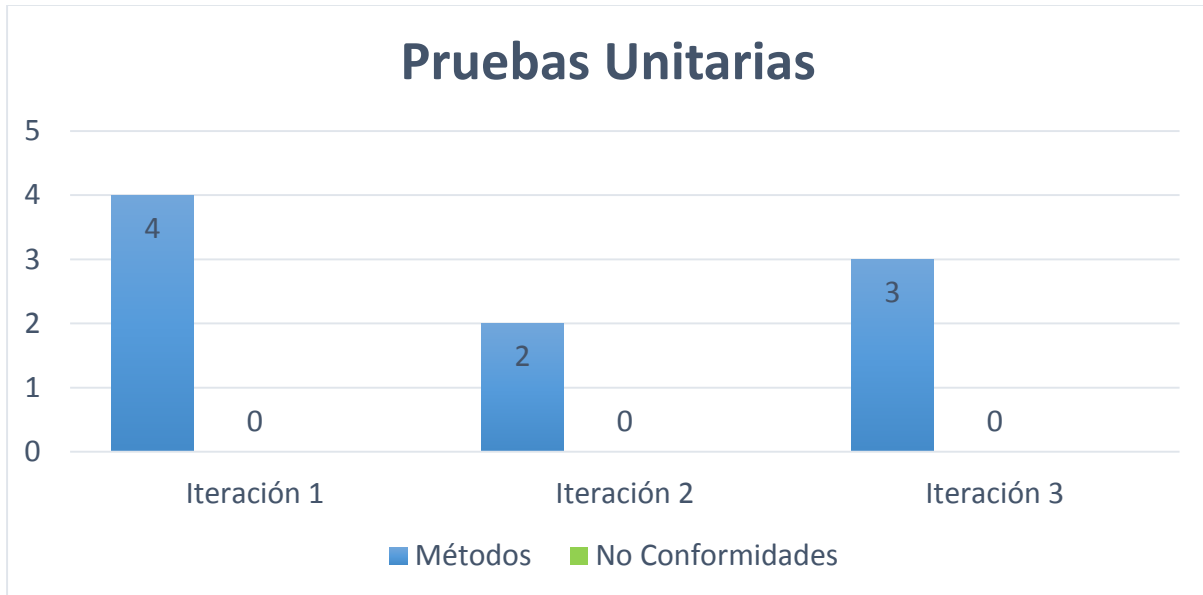


Ilustración 8: Resumen de Pruebas de Camino Básico

3.3.2 Pruebas de Aceptación:

Las pruebas de aceptación son pruebas de caja negra definidas por el cliente para cada historia de usuario, y tienen como objetivo asegurar que las funcionalidades del sistema cumplen con lo que se espera de ellas. En efecto, las pruebas de aceptación corresponden a una especie de documento de requerimientos en XP, ya que marcan el camino a seguir en cada iteración, indicándole al equipo de desarrollo hacia donde tiene que ir y en qué puntos o funcionalidades debe poner el mayor esfuerzo y atención. Al finalizar una iteración de desarrollo, una HU es aceptada por el cliente si la prueba de aceptación correspondiente a la misma es satisfactoria, si una prueba de aceptación falla, se realiza otra iteración de pruebas donde, tanto la HU como la prueba de aceptación correspondientes pueden sufrir cambios(36).

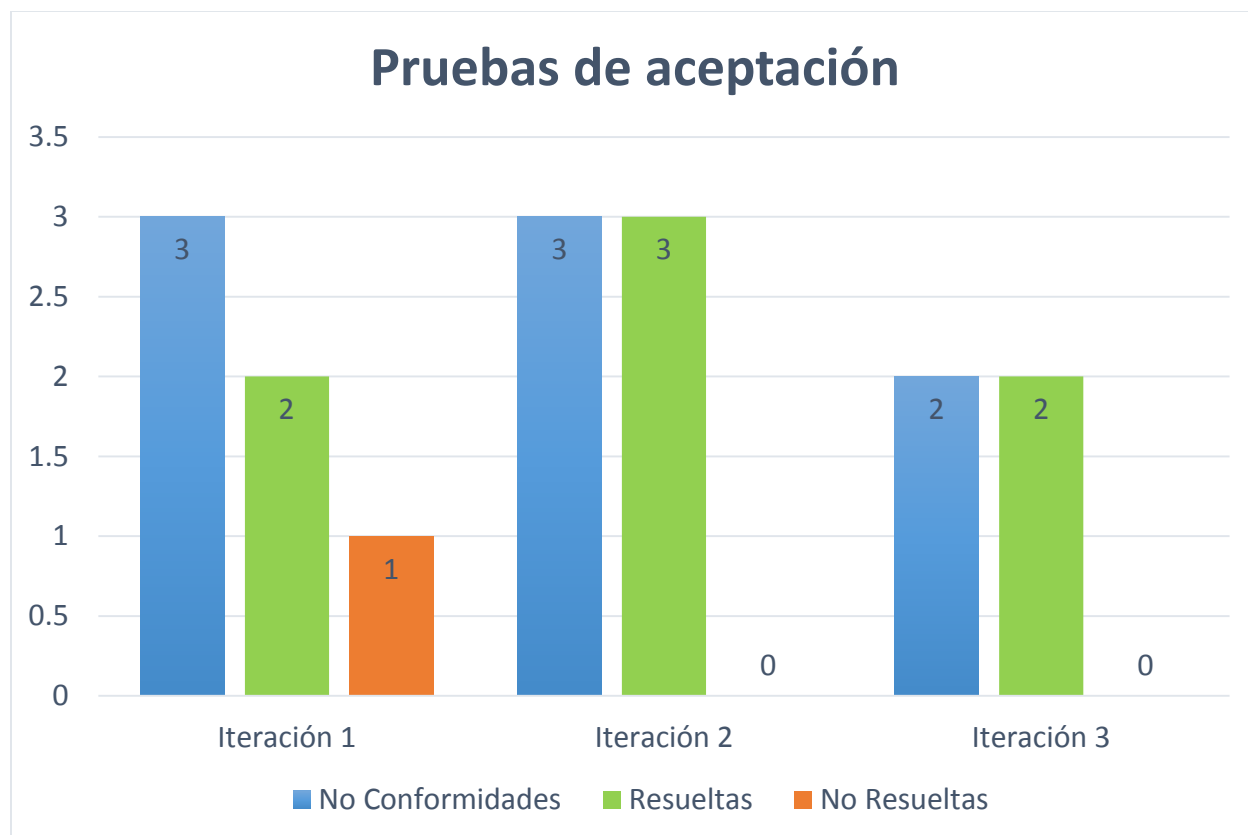


Ilustración 9 Resumen de pruebas de aceptación

Al realizar las pruebas de aceptación, se detectaron 3 no conformidades en la 1ra iteración, de las cuales 2 fueron resueltas en la misma 1ra iteración. En la 2da iteración se encontraron 2 no conformidades de la iteración y 1 proveniente de la 1ra iteración, las cuales fueron resueltas al finalizar la iteración. En la 3ra Iteración se encontraron 2 no conformidades las cuales fueron resueltas de forma satisfactoria. Al finalizar la 3ra iteración no se encontraron no conformidades, por lo que se dio fin al proceso de pruebas.

La siguiente tabla muestra los casos de prueba de aceptación realizada a las HU.

Caso de pruebas de aceptación	
Nombre:	PA-1 Tratamiento de imagen.
Historia de Usuario:	Tratamiento de imagen.
Descripción:	Se prueba que se realice de forma correcta la captura de la imagen del MB de la ficha técnica, el tratamiento y la transformación de los caracteres obtenidos de la imagen del MB de la ficha técnica.
Condiciones de ejecución:	<ul style="list-style-type: none"> La foto debe contener todos los caracteres, y ser legible La ficha técnica debe encontrarse en buen estado.

GDROID

Pasos de ejecución:	<ul style="list-style-type: none"> • Ejecutar la aplicación • Navegar hacia la pestaña "Escáner " • Tomar la imagen de la foto.
Resultado esperado – Datos Válidos:	Se realiza la foto satisfactoriamente, y se transforma la imagen en un número de serie digital.
Resultado esperado – Datos Inválidos:	Muestra una notificación de error con el texto "Error obteniendo la imagen".
Evaluación de prueba:	Satisfactoria.

Tabla 23: PA-1 Tratamiento de imagen.

Caso de pruebas de aceptación	
Nombre:	PA-2 Mostrar resultados
Historia de Usuario:	Mostrar resultados de la sincronización con Gserver
Descripción:	Se prueba que se realice de forma correcta el mostrado de los datos del ordenador cuyo número de serie coincide con el número de serie obtenido de la imagen.
Condiciones de ejecución:	<ul style="list-style-type: none"> • Existe una conexión activa a la red Wi-Fi • El servidor debe estar ejecutándose.
Pasos de ejecución:	<ul style="list-style-type: none"> • Se presiona la pestaña "Sincronización"
Resultado esperado – Datos Válidos:	Se muestran satisfactoriamente los datos del ordenador cuyo MB coincide con el MB obtenido de la imagen.
Resultado esperado – Datos Inválidos:	Se muestra una notificación de error con el texto " Conéctese a internet".
Evaluación de prueba:	Satisfactoria

Tabla 24: PA-2 Mostrar resultados

Caso de pruebas de aceptación	
Nombre:	PA-3 Visualización de la información estadística
Historia de Usuario:	Visualización de la información estadística
Descripción:	Se prueba que se muestre correctamente en forma de gráficas los datos estadísticos.
Condiciones de ejecución:	<ul style="list-style-type: none"> • Existe una conexión activa a la red Wi-Fi • El servidor debe estar ejecutándose.
Pasos de ejecución:	<ul style="list-style-type: none"> • Ejecutar la aplicación

GDROID

	<ul style="list-style-type: none">• Navegar hacia la primera vista de cualquiera de las 4 categorías principales (Agentes, Inventarios, Incidencias, Acciones)
Resultado esperado – Datos Válidos:	Se muestra satisfactoriamente la información estadística de cualquiera de las categorías seleccionadas por medio de gráficos de barra.
Resultado esperado – Datos Inválidos:	Se muestra una notificación de error con el texto “ Conéctese a una red”.
Evaluación de prueba:	Satisfactoria

Tabla 25: PA-3 Visualización de la información estadística

3.4 Conclusiones del capítulo.

En el presente capítulo se expuso el proceso de implementación de la solución propuesta, planteada por la metodología XP, el cual dio lugar a un producto que cumple con las pautas definidas en la propuesta de solución descrita en el Capítulo 2. Se crearon las tarjetas CRC y las tareas de ingeniería que dieron solución a las HU, logrando así una mayor organización y rapidez en el desarrollo del sistema. Se realizaron las pruebas unitarias y de aceptación al sistema, las cuales sirvieron como elemento clave para validar que la solución cumple con las expectativas del cliente.

CONCLUSIONES GENERALES

El desarrollo del presente trabajo de diploma permitió que se llevaran a cabo todas las tareas de investigación, a fin de dar cumplimiento al objetivo general, lo que permitió arribar a las siguientes conclusiones.

- La elaboración del marco teórico y el estudio del estado del arte realizado, acerca de: herramientas de gestión de recursos de hardware y software con plataforma Android, herramientas con escaneo y tratamiento de imágenes, y herramientas de representación gráfica, permitió al equipo de desarrollo validar la premisa inicial de la presente investigación.
- Las etapas de planificación, diseño, implementación y pruebas, definidas por la metodología de desarrollo de software XP, facilitaron el cumplimiento del plan de entrega, permitiendo el desarrollo de la aplicación por iteraciones y la integración de un producto final funcional que cumple con las expectativas deseadas.
- El desarrollo de un módulo de la aplicación Android para la administración del sistema GRHS permitió que dicha aplicación contara, además, con funcionalidades que permitan realizar la captura y tratamiento de imágenes y mostrar la información estadística en forma de gráficas.

Luego de lo anteriormente planteado se puede afirmar que la presente investigación alcanzó el objetivo propuesto.

GDROID

RECOMENDACIONES

Teniendo en cuenta lo anteriormente planteado se recomienda.

- Realizar un lector de código QR en la aplicación móvil del Sistema de Hardware y Software GDroid.

REFERENCIA BIBLIOGRÁFICA

1. SG Buzz. [online]. [Accessed 4 February 2017]. Available from: <https://sg.com.mx/revista/42/tecnologias-moviles-la-empresa>.
2. DANIEL COHEN KAREN and ENRIQUE ASÍN LARES. *Tecnologías de información en los negocios*. Monterrey. McGrawHill, 2009. [no date].
3. NESTOR CASTRO CASTILLO. *Aplicación Android para la administración del sistema Gestor de Recursos de Hardware y Software*. June 2016.
4. Soluciones de gestión de activos de TI para empresas e instituciones educativas. [online]. February 2017. [Accessed 6 June 2017]. Available from: <http://www.netsupportdna.com/ES/index.asp>
5. CamCard. [online]. 2015. [Accessed 6 June 2017]. Available from: <https://b.camcard.com/The Professional Business Card Reader>
6. Text Fairy (OCR Text Scanner) - Aplicaciones de Android en Google Play. [online]. February 2017. [Accessed 6 June 2017]. Available from: <https://play.google.com/store/apps/details?id=com.renard.ocr&hl=es>
7. ADVANCE-MEDIA. *3D Gráficos* [online]. Advance-Media, 2017. [Accessed 6 June 2017]. Available from: https://play.google.com/store/apps/details?id=air.A3DChartsMobile&hl=es_419
8. SjOGraph graphs & plotting - Android Apps on Google Play. [online]. February 2017. [Accessed 6 June 2017]. Available from: <https://play.google.com/store/apps/details?id=com.profitsearch.sjograph&hl=en>
9. Representación gráfica de datos estadísticos - hiru.eus. [online]. 2013. [Accessed 6 June 2017]. Available from: <http://www.hiru.eus/matematicas/representacion-grafica-de-datos-estadisticos>
10. *PowerPoint Presentation - pasos_tipos_graficos.pdf* [online]. [Accessed 6 June 2017]. Available from: http://www.ine.es/explica/docs/pasos_tipos_graficos.pdf
11. JAHODA, Philipp. *MPAndroidChart*: [online]. Java. 2017. [Accessed 7 June 2017]. Available from: <https://github.com/PhilJay/MPAndroidChart>
12. ¿Qué es Reconocimiento óptico de caracteres (OCR)? [online]. 2017. [Accessed 6 June 2017]. Available from: <https://www.abbyy.com/es-la/finereader/what-is-ocr/>
13. CARLOS JAVIER SANCHEZ, Victor Sandoní Consuegra. *Reconocimiento Óptico de Caracteres(OCR)*. Universidad Carlos III, 2008.
14. THEIS, Robert. *tess-two: Fork of Tesseract Tools for Android* [online]. C. 2017. [Accessed 7 June 2017]. Available from: <https://github.com/rmtheis/tess-two>
15. BLOOMBERG, Dan. *Leptonica* [online]. C. 2017. [Accessed 7 June 2017]. Available from: <https://github.com/DanBloomberg/leptonica>

GDROID

16. INGENIERIADESFTWARE. XP - Extreme Programing Ingenieria de Software. *ingenieriadesoftware* [online]. [Accessed 6 June 2017]. Available from: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html XP, Programacion Extrema
17. ADDISON WESLEY, Kent Beck. *Extremem Programing Explained*. Embrace Change, 1999.
18. PATRICIO LETELIER, M^a CARMEN PENADÉS. *Metodologias ágiles para el desarrollo de software: eXtreme Programing*. Universidad Politecnica de Valencia, 2006.
19. ORACLE CORPORATION. JDK 7. [online]. March 2017. [Accessed 7 June 2017]. Available from: <http://openjdk.java.net/projects/jdk7/>
20. SL, Media Ingea. Android Studio. [online]. March 2017. [Accessed 7 June 2017]. Available from: <https://android-studio.uptodown.com/windows>
21. SDK. Android SDK | Android Developers. [online]. October 2014. [Accessed 7 June 2017]. Available from: <https://stuff.mit.edu/afs/sipb/project/android/docs/sdk/index.html#>
22. USES-SDK. <uses-sdk> | Android Developers. [online]. February 2017. [Accessed 7 June 2017]. Available from: <https://developer.android.com/guide/topics/manifest/uses-sdk-element.html?hl=es>
23. NDK. Android NDK | Android Developers. [online]. June 2016. [Accessed 7 June 2017]. Available from: <https://developer.android.com/ndk/index.html?hl=es>
24. PYTHON. Welcome to Python.org. *Python.org* [online]. March 2017. [Accessed 7 June 2017]. Available from: <https://www.python.org/> The official home of the Python Programming Language
25. DJANGO. The Web framework for perfectionists with deadlines | Django. [online]. March 2017. [Accessed 7 June 2017]. Available from: <https://www.djangoproject.com/>
26. JETBRAINS. PyCharm. *JetBrains* [online]. March 2017. [Accessed 7 June 2017]. Available from: <https://www.jetbrains.com/pycharm/>
27. BECK and KENT. *Extreme Programming Explained: Embrace Change*. New York : Addison-Wesley, 2000. ISBN Longman Inc.
28. DAVID GARLAN and MARY SHAW. *An introduction to Software Architecture*. 2011.
29. RODRIGO PASZNIUK. Arquitectura Cliente-Servidor – Programación. [online]. July 2013. [Accessed 7 June 2017]. Available from: <https://www.programacion.com.py/varios/arquitectura-cliente-servidor>
30. DAVID MIGUEL LOZANO. Técnicas y herramientas — documentación de GoBees - 1.0. [online]. May 2016. [Accessed 6 June 2017]. Available from: http://go-bees.readthedocs.io/es/develop/documentacion/4_TecnicasHerramientas.html
31. CRAIG LARMAN. *UML y Patrones*. Prentice Hall, 2013.

GDROID

32. NATIONAL INSTRUMENT. Patrones de diseño de aplicaciones: Maestro/Esclavo - National Instruments. *National Instrument* [online]. March 2017. [Accessed 7 June 2017]. Available from: <http://www.ni.com/white-paper/3022/es/>
33. BECK and KENT. *Manifesto for Agile Software Development*. 2001.
34. LETELIER PH.D. *Métodologías Ágiles para el Desarrollo de Software: eXtreme Programming (XP)*. Técnica Administrativa. Buenos Aires, 2006.
35. ANDRES SOMMARIBA. Pruebas Unitarias, Parte 1: Introducción y utilización de objetos simulados (Mock). [online]. February 2014. [Accessed 7 June 2017]. Available from: <http://www.microgestion.com.ar/index.php/mg-developers/articulos/74-unit-test-part1-mock>
36. JOSÉ JOSKOWICZ. *Reglas y Prácticas en eXtreme Programming*. Universidad de Vigo, [no date].

BIBLIOGRAFÍA

1. USES-SDK. <uses-sdk> | Android Developers. [online]. February 2017. [Accessed 7 June 2017]. Available from: <https://developer.android.com/guide/topics/manifest/uses-sdk-element.html?hl=es>
2. ADVANCE-MEDIA. *3D Gráficos* [online]. Advance-Media, 2017. [Accessed 6 June 2017]. Available from: https://play.google.com/store/apps/details?id=air.A3DChartsMobile&hl=es_419
3. NDK. Android NDK | Android Developers. [online]. June 2016. [Accessed 7 June 2017]. Available from: <https://developer.android.com/ndk/index.html?hl=es>
4. SDK. Android SDK | Android Developers. [online]. October 2014. [Accessed 7 June 2017]. Available from: <https://stuff.mit.edu/afs/sipb/project/android/docs/sdk/index.html#>
5. SL, Media Ingea. Android Studio. [online]. March 2017. [Accessed 7 June 2017]. Available from: <https://android-studio.uptodown.com/windows>
6. DAVID GARLAN and MARY SHAW. *An introduction to Software Architecture*. 2011.
7. NESTOR CASTRO CASTILLO. *Aplicación Android para la administración del sistema Gestor de Recursos de Hardware y Software*. June 2016.
8. RODRIGO PASZNIUK. *Arquitectura Cliente-Servidor – Programación*. [online]. July 2013. [Accessed 7 June 2017]. Available from: <https://www.programacion.com.py/varios/arquitectura-cliente-servidor>
9. *Arquitectura de Software* | Facultad de Ingeniería. [online]. June 2015. [Accessed 7 June 2017]. Available from: <https://www.fing.edu.uy/cpap/cursos/arquitectura-de-software>
10. CamCard. [online]. 2015. [Accessed 6 June 2017]. Available from: <https://b.camcard.com/The Professional Business Card Reader>
11. *Conoce Android Studio* | Android Studio. [online]. [Accessed 7 June 2017]. Available from: <https://developer.android.com/studio/intro/index.html?hl=es-419>

GDROID

12. BECK and KENT. *Extreme Programming Explained: Embrace Change*. New York : Addison-Wesley, 2000. ISBN Longman Inc.
13. GOCR. [online]. March 2013. [Accessed 7 June 2017]. Available from: <http://jocr.sourceforge.net/index.html>
14. Google Goggles - Aplicaciones de Android en Google Play. [online]. March 2017. [Accessed 6 June 2017]. Available from: https://play.google.com/store/apps/details?id=com.google.android.apps.unveil&hl=es_419.
15. JULIO ANTONIO HERNANDEZ. GRHS: Gestor de Recursos de Hardware y Software (PDF Download Available). [online]. abril 2014. [Accessed 7 June 2017]. Available from: https://www.researchgate.net/publication/261511764_GRHS_Gestor_de_Recursos_de_Hardwre_y_Software
16. ORACLE CORPORATION. JDK 7. [online]. March 2017. [Accessed 7 June 2017]. Available from: <http://openjdk.java.net/projects/jdk7/>
17. BLOOMBERG, Dan. *Leptonica* [online]. C. 2017. [Accessed 7 June 2017]. Available from: <https://github.com/DanBloomberg/leptonica>
18. BECK and KENT. *Manifesto for Agile Software Development*. 2001.
19. *Memoria Final.pdf* [online]. [Accessed 7 June 2017]. Available from: <http://eprints.ucm.es/31485/1/Memoria%20Final.pdf>
20. *Metodología Ágil Programación Extrema XP - 62161.pdf* [online]. [Accessed 7 June 2017]. Available from: <http://repositorio.unan.edu.ni/1365/1/62161.pdf>
21. PATRICIO LETELIER, M^a CARMEN PENADÉS. *Metodologías ágiles para el desarrollo de software: eXtreme Programming*. Universidad Politecnica de Valencia, 2006.
22. LETELIER PH.D. *Métodologías Ágiles para el Desarrollo de Software: eXtreme Programming (XP)*. Técnica Administrativa. Buenos Aires, 2006.
23. SAMPIER, Roberto Hernandez. *metodologias de la investigacion 1*. La Habana, 2008.
24. Métodos Científicos de Investigación - EcuRed. [online]. June 2016.

GDROID

[Accessed 7 June 2017]. Available from:

https://www.ecured.cu/M%C3%A9todos_Cient%C3%ADficos_de_Investigaci%C3%B3n

25. MÉTODOS EMPÍRICOS. *Scribd* [online]. February 2017. [Accessed 7 June 2017]. Available from: <https://es.scribd.com/doc/21229743/METODOS-EMPIRICOS>
26. Modelo Vista Presentador (MVP) en Android. *Develapps* [online]. February 2016. [Accessed 5 June 2017]. Available from: <http://www.develapps.com/es/noticias/modelo-vista-presentador-mvp-en-android>
27. JAHODA, Philipp. *MPAndroidChart*: [online]. Java. 2017. [Accessed 7 June 2017]. Available from: <https://github.com/PhilJay/MPAndroidChart>
28. MVP como patrón de arquitectura Android – KATADE. [online]. 5 February 2016. [Accessed 5 June 2017]. Available from: <http://katade.com/2016/02/05/mvp-patron-arquitectura-android/>
29. Ocrad - GNU Project - Free Software Foundation (FSF). [online]. March 2017. [Accessed 7 June 2017]. Available from: https://www.gnu.org/software/ocrad/ocrad_es.html
30. MARIA FERNANDA FRIDSON TALLEDO. *OCR conversionales*. 2013.
31. NATIONAL INSTRUMENT. Patrones de diseño de aplicaciones: Maestro/Esclavo - National Instruments. *National Instrument* [online]. March 2017. [Accessed 7 June 2017]. Available from: <http://www.ni.com/white-paper/3022/es/>
32. *PowerPoint Presentation - pasos_tipos_graficos.pdf* [online]. [Accessed 6 June 2017]. Available from: http://www.ine.es/explica/docs/pasos_tipos_graficos.pdf
33. Presentación de datos estadísticos - Monografias.com. [online]. diciembre 2012. [Accessed 7 June 2017]. Available from: <http://www.monografias.com/trabajos81/presentacion-datos-estadisticos/presentacion-datos-estadisticos.shtml>
34. ANDRES SOMMARIBA. Pruebas Unitarias, Parte 1: Introducción y utilización de objetos simulados (Mock). [online]. February 2014. [Accessed 7 June 2017]. Available from: <http://www.microgestion.com.ar/index.php/mg-developers/articulos/74-unit-test-part1-mock>
35. JETBRAINS. PyCharm. *JetBrains* [online]. March 2017. [Accessed 7 June 2017].

GDROID

Available from: <https://www.jetbrains.com/pycharm/>

36. ¿Qué es Reconocimiento óptico de caracteres (OCR)? [online]. 2017.
[Accessed 6 June 2017]. Available from: <https://www.abbyy.com/es-la/finereader/what-is-ocr/>
37. CARLOS JAVIER SANCHEZ, Victor Sandoní Consuegra. *Reconocimiento Óptico de Caracteres(OCR)*. Universidad Carlos III, 2008.
38. *Reconocimiento Óptico de Caracteres(OCR)*.
39. JOSÉ JOSKOWICZ. *Reglas y Prácticas en eXtreme Programming*. Universidad de Vigo, [no date].
40. Representación gráfica de datos estadísticos - hiru.eus. [online]. 2013.
[Accessed 6 June 2017]. Available from: <http://www.hiru.eus/matematicas/representacion-grafica-de-datos-estadisticos>
41. Requisitos vs Casos de uso vs Historias de Usuario - Angel Lozano. [online]. February 2016. [Accessed 7 June 2017]. Available from: <http://www.angelozano.com/requisitos-del-sistema-vs-casos-uso-vs-historias-usuario/>
42. SDK Tools Release Notes | Android Studio. [online]. October 2014.
[Accessed 7 June 2017]. Available from: <https://developer.android.com/studio/releases/sdk-tools.html>
43. SG Buzz. [online]. [Accessed 4 February 2017]. Available from:
<https://sg.com.mx/revista/42/tecnologias-moviles-la-empresa>.
44. SjOGraph graphs & plotting - Android Apps on Google Play. [online]. February 2017.
[Accessed 6 June 2017]. Available from:
<https://play.google.com/store/apps/details?id=com.profitsearch.sjograph&hl=en>
45. Soluciones de gestión de activos de TI para empresas e instituciones educativas. [online]. February 2017. [Accessed 6 June 2017]. Available from:
<http://www.netsupportdna.com/ES/index.asp>
46. DAVID MIGUEL LOZANO. Técnicas y herramientas — documentación de GoBees - 1.0. [online]. May 2016. [Accessed 6 June 2017]. Available from: [54](http://go-</div><div data-bbox=)

GDROID

bees.readthedocs.io/es/develop/documentacion/4_TecnicasHerramientas.html

47. DANIEL COHEN KAREN and ENRIQUE ASÍN LARES. *Tecnologías de información en los negocios*. Monterrey. McGrawHill, 2009. [no date].
48. tesseract-ocr. *GitHub* [online]. March 2017. [Accessed 7 June 2017]. Available from: <https://github.com/tesseract-ocr>
49. THEIS, Robert. *tess-two: Fork of Tesseract Tools for Android* [online]. C. 2017. [Accessed 7 June 2017]. Available from: <https://github.com/rmtheis/tess-two>
50. Text Fairy (OCR Text Scanner) - Aplicaciones de Android en Google Play. [online]. February 2017. [Accessed 6 June 2017]. Available from: <https://play.google.com/store/apps/details?id=com.renard.ocr&hl=es>
51. *TFMMIJGarciaRodriguezRUO.pdf* [online]. [Accessed 7 June 2017]. Available from: <http://digibuo.uniovi.es/dspace/bitstream/10651/32457/6/TFMMIJGarciaRodriguezRUO.pdf>
52. DJANGO. The Web framework for perfectionists with deadlines | Django. [online]. March 2017. [Accessed 7 June 2017]. Available from: <https://www.djangoproject.com/>
53. TRABAJO_INVESTIGACIÓN 2016 XP- final.doc -. [online]. [Accessed 7 June 2017]. Available from: <https://www.coursehero.com/file/18398478/TRABAJO-INVESTIGACI%c3%93N-2016-XP-finaldoc/>
54. CRAIG LARMAN. *UML y Patrones*. Prentice Hall, 2013.
55. *WEB-Cliente-Servidor.ppt - web-cliente-servidor* [online]. [Accessed 7 June 2017]. Available from: <http://ocw.pucv.cl/cursos-1/arquitectura-de-sistemas-de-software/materiales-de-clases/web-cliente-servidor>
56. PYTHON. Welcome to Python.org. *Python.org* [online]. March 2017. [Accessed 7 June 2017]. Available from: <https://www.python.org/>The official home of the Python Programming Language
57. INGENIERIADESFTWARE. XP - Extreme Programing Ingenieria de Software. *ingenieriadesoftware* [online]. [Accessed 6 June 2017]. Available from: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.htmlXP,Programacion

GDROID

Extrema.

58. XP - Extreme Programming Ingenieria de Software. [online]. [Accessed 7 June 2017]. Available from: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programming.htmlXP,Programacion Extrema.

ANEXOS

Anexos I: Muestra de Tarjetas CRC.

Tarjeta CRC	
Clase: ScanActivity	
Responsabilidades	Colaboradores
<ul style="list-style-type: none">• Ejecuta la cámara.• Toma la foto.• Realiza el procesamiento de la foto para eliminar ruido.• Obtiene los datos del tratamiento de la imagen y los envía al servidor.	<ul style="list-style-type: none">• SyncAdapter

Tabla 26 TCRC ScanActivity

Tarjeta CRC	
Clase: ScanResult	
Responsabilidades	Colaboradores
<ul style="list-style-type: none">• Muestra los datos obtenidos del servidor, relacionados con la ficha técnica del ordenador.	<ul style="list-style-type: none">• SyncAdapter

Tabla 27:TCRC ScanResult

Tarjeta CRC	
Clase: SyncAdapter	
Responsabilidades	Colaboradores
<ul style="list-style-type: none">• Procesa los datos obtenidos del servidor, relacionados con la ficha técnica del ordenador.	<ul style="list-style-type: none">• ScanActivity• ScanResult• ListAPITask

Tabla 28: TCRC SyncAdapter

Tarjeta CRC	
Clase: ListAPITask	
Responsabilidades	Colaboradores
<ul style="list-style-type: none">• Se comunica con la API que se encuentra en el servidor.• Envía y recibe datos mediante en protocolo de conexión segura HTTPS	<ul style="list-style-type: none">• SyncAdapter

Tabla 29: TCRC ListAPITask

Tarjeta CRC	
-------------	--

GDROID

Clase: InventoryStatisticFragment	
Responsabilidades	Colaboradores
Genera una gráfica de barra para los Inventarios, a partir de los datos obtenidos de la base de datos local.	<ul style="list-style-type: none">• DataBaseContractProvide• XYPlot

Tabla 30: TCRC InventoryStatisticFragment

Tarjeta CRC	
Clase: IncidencesStatisticFragment	
Responsabilidades	Colaboradores
Genera una gráfica de barra para las Incidencias, a partir de los datos obtenidos de la base de datos local.	<ul style="list-style-type: none">• DataBaseContractProvide• XYPlot

Tabla 31: TCRC IncidencesStatisticFragment

Tarjeta CRC	
Clase: AgentStatisticFragment	
Responsabilidades	Colaboradores
Genera una gráfica de barra para los Agentes, a partir de los datos obtenidos de la base de datos local.	<ul style="list-style-type: none">• DataBaseContractProvide• XYPlot

Tabla 32: TCRC AgentStatisticFragment

Tarjeta CRC	
Clase: DataBaseContractProvide	
Responsabilidades	Colaboradores
Define el esquema de creación de las bases de datos local mediante clases estadísticas y constantes. Contiene las definiciones de Uri de contenido para el manejo de los datos para un ContentProvided.	<ul style="list-style-type: none">• ActionStatisticFragmen• InventoryStatisticFragment• AgentStatisticFragment• IncidencesStatisticFragment

Tabla 33: TCRC DataBaseContractProvide

Anexos II: Pruebas unitarias por el método de camino básico.

Iteración 1: Pruebas a la clase ScanActivity.

Prueba de Camino Básico: GetPath
Probado por Luis Ángel Roche Broche

GDROID

<pre>private void getPath() { if(state.equals(Environment.MEDIA_MOUNTED)) { 1 DATA_PATH = Environment.getExternalStorageDirectory().toString()+"/Tess"; 2 } else { 3 DATA_PATH = Environment.DIRECTORY_PICTURES.toString()+"/Tess"; 4 } }</pre>	<pre> graph TD 1((1)) --> 2((2)) 1((1)) --> 3((3)) 2((2)) --> 5((5)) 3((3)) --> 4((4)) 4((4)) --> 5((5)) </pre>
<p>Complejidad Ciclomática:</p> <p>$V(G) = (\text{Cantidad de aristas} - \text{Cantidad de nodos}) + 2 = (5 - 5) + 2 = 2$</p>	<p>Caminos Básicos:</p> <ol style="list-style-type: none"> 1. (1-2-5) 2. (1-3-4-5)

Tabla 34: PU CB Método GetPath

Prueba de Camino Básico: onClick	
Probado por Luis Ángel Roche Broche	
<pre>public void onClick(View v) { if (!editText.getText().toString().equals("")) { 1 if (NetUtils.hasNetworkConnection(getApplicationContext())) { 2 sync(); 3 } else { 4 Toast.makeText(getApplicationContext(), R.string.connect_to_internet, Toast.LENGTH_SHORT).show(); 5 } } else { 6 Toast.makeText(ScanActivity.this, "The Inventory is Empty", Toast.LENGTH_SHORT).show(); 7 } }</pre>	<pre> graph TD 1((1)) --> 2((2)) 1((1)) --> 6((6)) 2((2)) --> 3((3)) 2((2)) --> 4((4)) 3((3)) --> 8((8)) 4((4)) --> 5((5)) 5((5)) --> 8((8)) 6((6)) --> 7((7)) 7((7)) --> 8((8)) </pre>
<p>Complejidad Ciclomática:</p> <p>$V(G) = (\text{Cantidad de aristas} - \text{Cantidad de nodos}) + 2 = (9 - 8) + 2 = 3$</p>	<p>Caminos Básicos:</p> <ol style="list-style-type: none"> 1. (1-2-3-8) 2. (1-2-4-5-8) 3. (1-2-6-7-8)

Tabla 35: PU CB Método onClick

Prueba de Camino Básico: StartOCR
Probado por Luis Ángel Roche Broche

GDROID

<pre>private void StartOCR(Uri imageUri) { 1 try { BitmapFactory.Options option= new BitmapFactory.Options(); 2 option.inSampleSize = 7; Bitmap bitmap = BitmapFactory.decodeFile(imageUri.getPath(),option); String result = this.getText(bitmap); editText.setText(result); File file = new File(imageUri.getPath()); boolean a = file.delete(); if(a){ 3 Toast.makeText(getApplicationContext(), "", Toast.LENGTH_SHORT).show(); 4 } } catch (Exception e){ 5 Log.e(TAG,e.getMessage()); } } 6</pre>	
<p>Complejidad Ciclomática:</p> <p>$V(G) = (\text{Cantidad de aristas} - \text{Cantidad de nodos}) + 2 = (7-6) + 2 = 3$</p>	<p>Caminos Básicos:</p> <p>(1-2-6) (1-2-4-6) (1-5-6)</p>

Tabla 36: PU CB Método StartOCR

Prueba de Camino Básico: StartCamaraActivity	
Probado por Luis Ángel Roche Broche	
<pre>public void StartCamaraActivity() { 1 try { 2 String imagePath = DATA_PATH+"/image"; File dir = new File(imagePath); if (!dir.exists()){ 3 dir.mkdir(); 4 } Date date=new Date(); 5 String d= ""; Calendar c=Calendar.getInstance(); d=c.get(Calendar.YEAR)-1900+" "+date.getMonth()+" "+date.getDay()+" "+date.getHour(); String imageFilePath = imagePath+"/"+d+".jpg"; outputFileDir = Uri.fromFile(new File(imageFilePath)); final Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE); intent.putExtra(MediaStore.EXTRA_OUTPUT, outputFileDir); intent.putExtra("crop", "true"); intent.putExtra("outputX", 200); intent.putExtra("outputY", 200); intent.putExtra("aspectX", 1); intent.putExtra("aspectY", 1); intent.putExtra("scale", true); if (intent.resolveActivity(getPackageManager())!= null){ 6 startActivityForResult(intent, 100); 7 } } catch (Exception e){ 8 Log.e(TAG, e.getMessage()); } } 9</pre>	
<p>Complejidad Ciclomática:</p> <p>$V(G) = (\text{Cantidad de aristas} - \text{Cantidad de nodos}) + 2 = (12-9) + 2 = 5$</p>	<p>Caminos Básicos:</p> <p>(1-2-3-4-5-6-7-9) (1-2-3-5-6-7-9)</p>

GDROID

	(1-2-3-4-5-6-9)
	(1-2-3-5-6-9)
	(1-8-9)

Tabla 37: PU CB Método StartCamaraActivity

Iteración 2: Pruebas a la clase ScanResult.

Prueba de Camino Básico: getText	
Probado por Luis Ángel Roche Broche	
<pre>private String getText(Bitmap bitmap) { 1 try { tessBaseAPI = new TessBaseAPI(); 2 } catch (Exception e) { 3 Log.e(TAG, e.getMessage()); } tessBaseAPI.init(DATA_PATH, "eng"); 4 tessBaseAPI.setImage(bitmap); String rest = "No result"; try { rest = tessBaseAPI.getUTF8Text(); 5 } catch (Exception e) { 6 Log.e(TAG, e.getMessage()); } tessBaseAPI.end(); 7 return rest; 8 }</pre>	<pre> graph TD 1((1)) --> 2((2)) 1 --> 3((3)) 2 --> 4((4)) 3 --> 4 4 --> 5((5)) 4 --> 6((6)) 5 --> 7((7)) 6 --> 7 7 --> 8((8)) </pre>
<p>Complejidad Ciclomática:</p> <p>$V(G) = (\text{Cantidad de aristas} - \text{Cantidad de nodos}) + 2 = (10 - 8) + 2 = 4$</p>	<p>Caminos Básicos:</p> <ol style="list-style-type: none"> 1. (1-2-4-5-7-8) 2. (1-3-4-5-7-8) 3. (1-2-4-6-7-8) 4. (1-3-4-6-7-8)

Tabla 38: PU CB Método getText

Prueba de Camino Básico: onActivityResult
Probado por Luis Ángel Roche Broche

GDROID

<pre> protected void onActivityResult(int requestCode, int resultCode, Intent data) { 1 if (requestCode == 100 && resultCode == Activity.RESULT_OK) { 2 PrepareTestData(); StartOCR(outputFileDir); 3 } else { 4 Toast.makeText(getApplicationContext(), 5 } 6 </pre>	
<p>Complejidad Ciclomática: $V(G) = (\text{Cantidad de aristas} - \text{Cantidad de nodos}) + 2 = (6 - 6) + 2 = 2$</p>	<p>Caminos Básicos:</p> <ol style="list-style-type: none"> 1. (1-2-3-6) 2. (1-4-5-6)

Tabla 39: PU CB Método onActivityResult

Iteración 3: Pruebas a la clase StatisticFragment.

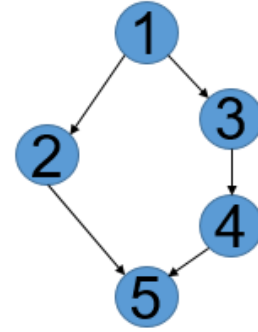
Prueba de Camino Básico: onActivityResult	
Probado por Luis Ángel Roche Broche	
<pre> public void onActivityResult(@Nullable Bundle savedInstanceState) { super.onActivityResult(savedInstanceState); 1 if (NetUtils.hasNetworkConnection(getActivity().getApplicationContext())) { 2 sync(); 3 } else { Cursor cursor = QueryUtils.getStats(getActivity().getApplicationContext()); 4 if (cursor.moveToFirst()) { 5 String date = cursor.getString(cursor.getColumnIndex(DataBaseContract .StatisticsColumns.STATISTICS_DATE)); String data = cursor.getString(cursor.getColumnIndex(DataBaseContract .StatisticsColumns.STATISTICS_DATA)); 6 parseStatisticsData(date, data); inicializarGrafo(values); } cursor.close(); 7 } } </pre>	
<p>Complejidad Ciclomática: $V(G) = (\text{Cantidad de aristas} - \text{Cantidad de nodos}) + 2 = (9 - 7) + 2 = 3$</p>	<p>Caminos Básicos:</p> <ul style="list-style-type: none"> (1-2-3-5-6-7) (1-2-3-5-7) (1-2-4-5-6-7) (1-2-4-5-7)

Tabla 40: PU CB Método onActivityResult

Prueba de Camino Básico: ParseMotherboardStats
Probado por Luis Ángel Roche Broche

GDROID

```
private void parseMotherboardStats(String data) {  
    Log.i(TAG, "Parsing motherboard statistics data..");1  
    try {  
        JSONObject motherboard = new JSONObject(data).getJSONObject("hardware")2  
            .getJSONObject("motherboard");  
        values.add(1, new StatsListItem("Registered assets",  
            motherboard.getString("count"), ""));  
        values.add(2, new StatsListItem("Types",  
            motherboard.getString("product"), ""));  
        values.add(3, new StatsListItem("Manufacturers",  
            motherboard.getString("manufacturer"), ""));  
    } catch (JSONException e) {3  
        e.printStackTrace();4  
    }5  
}
```



Complejidad Ciclomática:

$V(G) = (\text{Cantidad de aristas} - \text{Cantidad de nodos}) + 2 = (5 - 5) + 2 = 2$

Caminos Básicos:

1. (1-2-5)
2. (1-3-4-5)

Tabla 41: PU CB Método ParseMotherboardStats