

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

**“INSPECCION DIGITAL DE EQUIPOS DE RECAUDO EN
PARADAS Y TERMINALES DE LA METROVÍA”**

EXÁMEN DE GRADO (COMPLEXIVO)

Previa a la obtención del grado de:

**INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
ESPECIALIZACIÓN COMPUTACION**

GABRIEL ERNESTO SOSA ALVARADO

GUAYAQUIL – ECUADOR

AÑO: 2015

AGRADECIMIENTO

Mis más sinceros agradecimientos a todos mis amigos de la Fundación Metrovía que siempre me apoyaron, recibí todas las facilidades para la realización de este proyecto; me brindaron el ánimo, las fuerzas y la confianza para culminar por fin mis estudios.

Eternamente agradecido a la ESPOL, mi universidad, por haberme proporcionado el entorno preciso para adquirir todos los conocimientos y destrezas requeridas en mi carrera.

DEDICATORIA

El presente proyecto está dedicado a mis padres, quienes gracias a Dios estuvieron a mi lado durante todo el tiempo de mis estudios, brindándome su apoyo en todos los sentidos, y han esperado tanto tiempo para ver los frutos de nuestro esfuerzo conjunto: nunca es demasiado tarde.

Asimismo lo dedico a mi familia, mis hijas y mi esposa, que en la actualidad han sufrido igual que yo estos últimos sacrificios.

Especial mención a mis amigos en la Fundación Metrovía, seguimos adelante.

TRIBUNAL DE SUSTENTACIÓN

M.Sc. Cruz Falcones M.

PROFESOR DELEGADO

POR LA FIEC

M.Sc. Marisol Villacrés F.

PROFESOR DELEGADO

POR LA FIEC

DECLARACIÓN EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestas en este Informe me corresponde exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL).

.....
Gabriel Sosa Alvarado

RESUMEN

La Fundación Metrovía, como ente regulador del Sistema de Transporte Metrovía, efectúa inspecciones para revisión del estado de los equipos de control del recaudo en las paradas y terminales que componen las troncales del Sistema.

Estas inspecciones las lleva a cabo personal capacitado, sin embargo, el problema surge al utilizar procedimientos sin ninguna intervención tecnológica; registrando los hallazgos en hojas de papel, en formatos pre-impresos. Esto, además de generar un gasto de papel evitable, puede introducir errores humanos en el procedimiento.

Se genera una carga adicional al requerirse luego otro paso adicional, ya que alguien debe digitar la información que descansa en el papel a fin de introducirla en los depósitos de información digital de la empresa para poder ser procesada y analizada.

Por estos motivos, se hizo necesaria una modificación de los procedimientos, incorporando un elemento tecnológico en el proceso: una aplicación para dispositivos móviles, que el personal de inspectores utilizará para registrar los hallazgos resultantes de la inspección.

De esta manera se mejoraron mucho los tiempos requeridos para tener a disposición la información de estas inspecciones, se redujo el margen de errores, y se logró un considerable ahorro de papel.

ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA.....	III
TRIBUNAL DE SUSTENTACIÓN.....	IV
DECLARACIÓN EXPRESA	V
RESUMEN	VI
ÍNDICE GENERAL.....	VIII
INTRODUCCIÓN	X
CAPÍTULO 1	1
1. MEJORANDO UNA INSPECCIÓN.....	1
1.1 Generalidades del Sistema Metrovía.	1
1.2 Equipos de Recaudo.....	2
1.3 Importancia de una Inspección confiable.	3
1.4 Planificación de la solución.....	4
1.5 Consideraciones particulares.	4

1.6 Herramientas utilizadas.....	5
1.6.1 Android.....	5
1.6.2 Android Studio.....	7
1.6.3 SQLite.....	7
1.7 Desarrollo de la solución.....	8
1.7.1 Entidades y Clases.....	8
1.7.2 Interfaz Visual.....	12
1.7.3 Otras Funciones importantes.....	14
1.7.4 Interacción con otros programas externos.....	14
CAPÍTULO 2.....	16
2. UNA INSPECCIÓN MÁS CONFIABLE.....	16
2.1 Menor error humano.....	16
2.2 Ahorro en papel.....	17
2.3 Mejora de tiempos.....	17
CONCLUSIONES Y RECOMENDACIONES.....	18
BIBLIOGRAFÍA.....	20
ANEXOS.....	21

INTRODUCCIÓN

La Metrovía en Guayaquil es un sistema de transporte masivo urbano con ámbito y alcances muy importantes, que brinda servicio a miles de usuarios al día, cubriendo casi toda la ciudad con sus carriles troncales y sus rutas alimentadoras.

La Fundación Metrovía, cumpliendo su objetivo de administrar y regular en forma coordinada el Sistema Integrado de Transporte Urbano Masivo de Guayaquil - "Sistema METROVÍA" [1], debe supervisar la operación de cada una de las empresas que conforman el sistema, entre ellas al Integrador Tecnológico (encargado del cobro y manejo del dinero recaudado por concepto de pasajes). Esto se realiza mediante inspecciones en sitio a equipos del recaudo tales como máquinas de venta y recarga de tarjetas, validadores, entre otros. Por diversos motivos se concluyó que estas inspecciones aplicaban procedimientos que podían ser mejorados.

Al aplicar una mejora tecnológica a sus procedimientos, las inspecciones generaban una información más confiable, y en menor tiempo, logrando una mejor satisfacción de los usuarios del Sistema Metrovía.

En el Capítulo 1 se expondrá más detenidamente la importancia de mejorar estas inspecciones, así como las repercusiones de una inspección no confiable. Además, todas las consideraciones especiales que forman parte de los análisis y las

tecnologías evaluadas y aplicadas a la solución de este caso, mientras que en el Capítulo 2 se detallarán los resultados obtenidos de la aplicación de esta solución.

El presente informe explica paso a paso como se fue implementando esta solución y el impacto que generó, para de esta manera, dejar demostrado un caso de la vida real en el cual se aplicaron conocimientos que recibí durante mis años de estudio en la ESPOL.

CAPÍTULO 1

1. MEJORANDO UNA INSPECCIÓN

En este capítulo se detalla paso a paso cómo se implementó una solución tecnológica, ante las oportunidades detectadas para mejorar las inspecciones a los equipos de recaudo del Sistema Metrovía.

Se ha dividido este capítulo en varios temas de especial importancia que surgieron y tuvieron que ser abordados durante la realización del proyecto.

1.1 Generalidades del Sistema Metrovía.

La Metrovía es un conjunto de varias empresas privadas, entre estas, el Integrador Tecnológico (ITOR), el cual se encarga de controlar el recaudo del sistema, de las actividades que se relacionan con el medio de pago del pasaje desde que el usuario llega a una parada o bus del sistema para ser

transportado a su destino, del cobro automatizado en máquinas o el cobro directo por parte de un conductor a bordo de un bus alimentador, y de la movilización y cuadro de los dineros y su final distribución a las empresas que perciben estos ingresos.

A pesar de las virtudes del Sistema, que ha cambiado para bien la transportación en la ciudad, no podemos desconocer las necesidades y oportunidades de mejora del sistema, que se hacen evidentes en la experiencia diaria de los pasajeros.

En este contexto se propone la aplicación de nuevas tecnologías, pues lo que hoy parece óptimo, probablemente no lo será en el futuro; un equilibrio sustentable entre todos los factores que intervienen en un negocio privado como lo es la Metrovía, debería reflejarse en usuarios satisfechos.

El sistema está regido por la Fundación Municipal Metrovía, que supervisa el cumplimiento de las obligaciones de cada empresa, por ejemplo el control al ITOR (referente a este informe) mediante inspecciones en sitio a sus equipos de recaudo, instalados en las paradas y terminales, verificando mediante muestreos su perfecto funcionamiento.

1.2 Equipos de Recaudo.

Son equipos electrónicos instalados en la entrada de cada una de las paradas y terminales, aseguran el cobro del pasaje correspondiente a cada usuario utilizando los medios de pago establecidos.

- **MVRT.-** Maquina Automática de Venta y Recarga de Tarjetas. Permite que el mismo usuario ingrese su dinero y haga la recarga de pasajes en su propia tarjeta, o que adquiera una nueva tarjeta, sin intervención de un operador humano.
- **Validador de tarjetas.-** El usuario debe acercarle su tarjeta, la lee sin contacto, y determina si contiene saldos suficientes para enviar la orden a un torniquete de permitir o negar el ingreso al usuario.
- **Taquilla.-** Aún existen usuarios que no poseen su tarjeta, por lo tanto se mantiene operando ciertas taquillas de cobro con un operador humano, en paradas y terminales de mayor afluencia de pasajeros. Su estado también juega un rol importante para la experiencia del usuario.

1.3 Importancia de una Inspección confiable.

Las inspecciones a estos equipos se efectuaban sin ninguna ventaja tecnológica, se registraban las fallas detectadas en hojas de papel, las cuales posteriormente eran digitadas por otra persona en la oficina central; estos procedimientos comprometían la integridad de la información al introducir riesgo de error humano en el proceso, reduciendo la confiabilidad de la inspección.

Sabiendo que el buen estado de los equipos representa un mejor servicio al pasajero, minimiza las posibilidades de fraude, garantiza el respeto a la tarifa de cada usuario, agiliza su ingreso, entre otras muchas ventajas; se hace evidente la importancia de que la inspección sea lo más confiable y segura. Adicionalmente, el mal estado de los equipos acarrea sanciones a la empresa

responsable, por lo que una inspección no confiable tendría importantes repercusiones.

1.4 Planificación de la solución.

El objetivo del proyecto fue mejorar tecnológicamente la inspección de equipos de recaudo del Sistema Metrovía, para lograr mejores rendimientos en cuanto a tiempos totales de toma de información, y eliminar las causas que atenten contra la pureza de la misma.

Debido a que las inspecciones son en sitio, era evidente que la solución debía ser móvil, desarrollando una aplicación móvil que permita al inspector registrar el estado de los equipos de recaudo; luego esa información debía pasar a un depósito central, quedando a disposición de los sistemas computacionales de la Fundación para su procesamiento en los sistemas que ya existían para ese fin.

1.5 Consideraciones particulares.

Consideraciones adicionales propias de la naturaleza de las actividades de los inspectores y de la empresa, condicionaron el diseño final:

- El dispositivo móvil debía contar con una pantalla de tamaño medio, ni muy pequeña que no sea visible, ni muy grande que no permita ser manejado cómodamente en ambientes externos.
- El dispositivo debía ser resistente, para uso al aire libre y su movilización a bordo de los buses.

- El traspaso de la información sería en entorno desconectado, quedando almacenada en el dispositivo hasta ser descargada directamente a la central, debido a que por políticas de la empresa no se proporcionaba planes de datos. La información no viajaría en línea.
- Se debían respetar formatos establecidos para toma de información.

1.6 Herramientas utilizadas.

Considerando los puntos anteriormente descritos, se escogió desarrollar una aplicación para tablets con la plataforma Android, por la amplia disponibilidad de equipos móviles que operan con este sistema operativo, lo extendido de las herramientas de desarrollo, y los bajos costos de estos dispositivos: tablets marca AOC con pantallas táctiles de 7 pulgadas.

1.6.1 Android

Es un sistema operativo sobre un kernel de Linux, especialmente diseñado para dispositivos móviles, y ampliamente aceptado por desarrolladores alrededor del mundo. Existe variedad de dispositivos que utilizan Android.

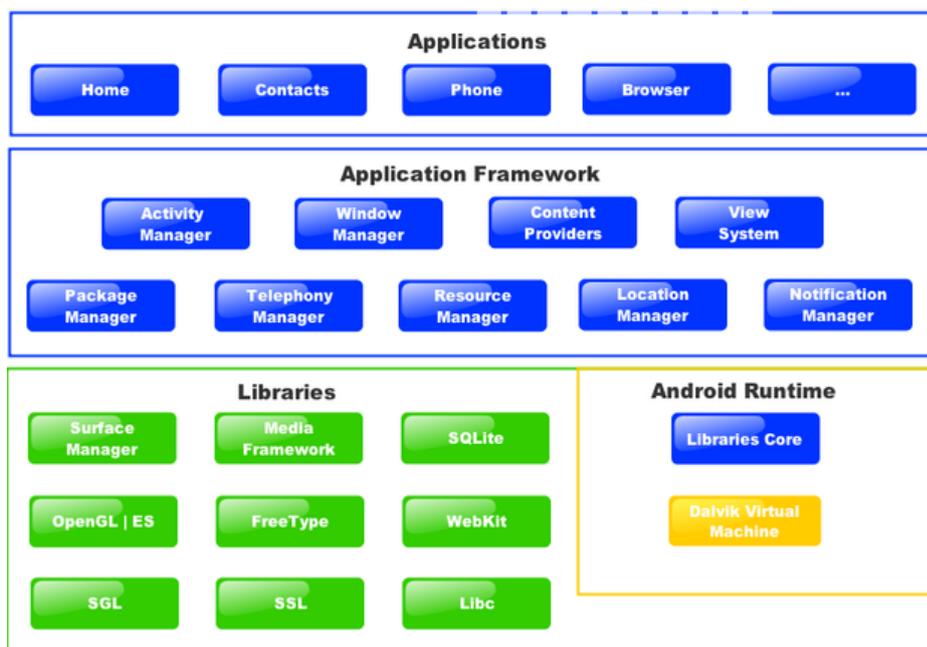


Figura 1.1. Arquitectura Android

En la figura 1.1 se observa un diagrama de la arquitectura de Android, se muestran las capas diferenciadas: las aplicaciones de usuario corriendo sobre un framework Java que soporta Programación Orientada a Objetos, y soporte nativo para base de datos SQLite. Se aprovecharon estas características en este proyecto. Android brinda muchas facilidades para los desarrolladores, poniendo al alcance del código los recursos hardware del dispositivo móvil, el programador recibe acceso completo a todas las APIs del framework que utilizan las aplicaciones base del sistema operativo [2]. Heredó las fortalezas en seguridad, manejo de memoria, procesos, y demás que caracterizan a Linux.

Las aplicaciones se generan como un solo archivo comprimido .APK, facilitando su transporte e instalación con cualquier explorador de archivos.

Android garantiza su vigencia con el desarrollo de nuevas versiones, librerías y cada vez más nuevas aplicaciones.

1.6.2 Android Studio

Todas las virtudes de Java, serían desaprovechadas sin un buen entorno de desarrollo, por eso se eligió Android Studio, que es el IDE (por Integrated Development Environment) oficial para el desarrollo de aplicaciones Android [\[3\]](#), por su similitud con otros entornos de amplia aceptación, autocompletado, ayudas, consejos de corrección y optimización del código fuente; aprovecha toda la pantalla del computador para mostrar las secciones más necesarias durante el desarrollo. Todo lo necesario está justo a la mano.

Posee un buen debug, acceso rápido al emulador, rápido aprendizaje y mucha familiaridad para los desarrolladores orientados a objetos (en java) y muy estable.

1.6.3 SQLite

Es un gestor de base de datos, concentra todo en un solo archivo .db muy pequeño, como datos, tablas, índices, relaciones.

El soporte para SQLite viene embebido en cada dispositivo Android [\[4\]](#), por eso las aplicaciones no necesitan conectarse con un proceso motor independiente, solo se enlazan con él mediante un objeto que gestionará la BD y el resto serán llamadas a funciones.

Una opción liviana y segura para las necesidades de este proyecto.

Se utilizó SQLiteManager como visor de tablas para verificar cada tanto la calidad de la información que se estaba guardando. Es un programa muy ligero, amigable y definitivamente una buena elección.

1.7 Desarrollo de la solución.

Aprovechando las ventajas de la Programación Orientada a Objetos (OOP) que brinda el lenguaje java, las herramientas seleccionadas se utilizaron en conjunto para desarrollar en una PC de la empresa la aplicación para Android.

En los Anexos [1](#) y [2](#) se incluyen el cronograma de etapas y cuadro de presupuestos referenciales del proyecto.

1.7.1 Entidades y Clases

Para el desarrollo de la aplicación, primero se reconocieron todas las entidades que participan, definiendo sus características principales y sus interrelaciones (ver Anexo [3](#)), para luego codificarlas como clases en java, utilizando conceptos básicos de la OOP como abstracción, herencia y polimorfismo.

- **Parada.**- Es una entidad que simboliza una parada del sistema Metrovía (se inspecciona parada por parada en una troncal); el aplicativo debe mostrar un listado de las paradas para que el inspector elija la que está inspeccionando. En la figura 1.2 se observa la declaración de sus atributos (código y nombre), y el método constructor que a través de otra entidad (custodio o manejador de datos, detallado posteriormente) consulta las tablas para obtener el listado de paradas.

```
/**
 * La clase que maneja un objeto abstracto que representa los datos de las
 * Paradas
 */
public class Clase_Parada {

    // los items seleccionados
    public String codigo = null;
    public String nombre = null;

    // creando el Constructor
    public Clase_Parada(Manejador_BD ManejaBD) {
        // al crear, deja apuntado al primer elemento de la lista
        ConsultaReg_Paradas_TODO(ManejaBD,true);
    }
}
```

Figura 1.2. Código Constructor de la Clase Parada

Otra consulta adicional encuentra el código de la parada a partir de su nombre.

El manejador de datos es quien busca en las tablas la información requerida en todas las funciones de consulta utilizadas por estas clases de más alto nivel.

- **Tipo de Equipos.**- Los equipos están agrupados en 3 clasificaciones: Validadores, MVRTs, y Equipos de Taquillas; el inspector al momento de identificar un equipo primero debe categorizarlo por su tipo.

Usa 2 funciones de consulta similares a la de la clase Parada.

- **Equipo.**- Es una entidad que representa a un equipo de recaudo.

Está definido por su tipo, código y descripción.

Usa más funciones de consulta: búsqueda del código, de la descripción, por tipo de equipo, y una consulta general de todos los equipos.

- **Encabezado de Inspección.**- Una entidad que engloba todos los datos únicos para la inspección, representa la inspección en sí.

Definida por su Id única, un nombre representativo, la troncal inspeccionada, fecha programada para la revisión, id del móvil utilizado, 3 fechas y horas: de inicio, de última modificación, y de fin de la inspección. El nombre de una persona responsable por parte de la empresa auditada (opcional), y estado actual de la inspección (nueva inspección, inspección activa, o inspección ya finalizada).

Posee múltiples funciones de edición de registros, cada vez que el programa necesita, por ejemplo, actualizar fecha y hora de modificación, o cambiar el estado de la inspección; además funciones de consulta, por ejemplo por estado, y 3 funciones especiales INICIA_INSPECCION, GRABA_FILA y FIN_INSPECCION detalladas posteriormente.

- **Detalle de Inspección.**- Contiene los datos de cada equipo inspeccionado (definido por el código de la parada, del tipo de

equipo, y del equipo actual), además del estado del equipo, y una observación opcional.

Otras funciones como crear un registro, cambiar el estado de un equipo (Operativo, No operativo, con Fallas, o no Existe); otras muchas funciones de consulta, y una función especial GRABA_FILA, detallada posteriormente.

- **Custodio de los datos.**- O manejador de la Base de Datos, alguien que tenga que lidiar con niveles más bajos de abstracción, que garantice que la información que usan las 5 entidades anteriores siempre esté guardada, la busque y la tenga disponible para cuando sea solicitada.

Se usó una clase que extiende y hereda de la clase SQLiteOpenHelper e interactúa directamente con abstracciones inferiores para leer/escribir de las tablas de la base de datos.

Usa funciones como Inserta_Reg, Edita_Reg, y Consulta_Reg, que insertan/editan/consultan un registro en determinada tabla de la base de datos.

```
// Inserta un nuevo registro en una Tabla
public int Inserta_Reg(String nombre_tabla, ContentValues valores_reg) {
    // obtengo un objeto db
    SQLiteDatabase db = this.getWritableDatabase();

    // Inserta el registro en la db
    long num_reg = db.insert(nombre_tabla, null, valores_reg);
    db.close();
    return (int) num_reg;
}
```

Figura 1.3. Código Función Inserta_Reg

El manejador de datos utiliza una instancia de una clase SQLiteDatabase (objeto db en la figura 1.3), es un objeto

abstracto que representa a la Base de Datos, mediante la cual podrá utilizar registros de las tablas.

Debido al cuidado especial que se debe tener de los recursos limitados con los que cuenta un dispositivo móvil, se debe cerrar explícitamente el objeto de la base de datos (`db.close()`) una vez que ya se logró la tarea con la tabla.

1.7.2 Interfaz Visual

El entorno de Android Studio es muy estable al momento de armar la interfaz visual y facilita el uso del emulador con el AVD (Android Virtual Device) creado previamente, para ir probando los cambios.

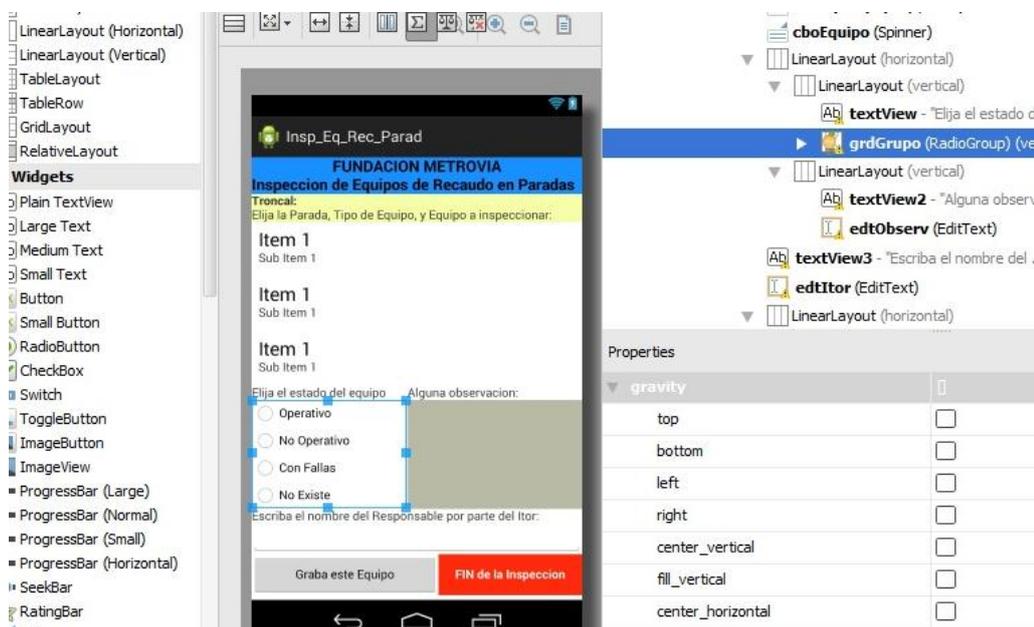


Figura 1.4. Edición de la Interfaz Visual con Android Studio

Uno a uno se fue agregando al lienzo (al `activity_main.xml`) cada control que el inspector va a necesitar para plasmar la información en pantalla, tal como se muestra en la figura 1.4: 3 de tipo `Spinner` que despliegan las listas de datos que el inspector debe elegir (paradas, tipos de equipos, y listado de equipos), un grupo de 4 `RadioButton` que muestran las opciones posibles del estado del equipo (Operativo, No operativo, con Fallas, No Existe), 2 de tipo `EditText` (uno para alguna observación y otro para digitar el nombre de la persona responsable de la empresa auditada), muchos `TextView` para mostrar títulos, y 2 botones (uno para Grabar los cambios a un equipo, y otro botón para Finalizar la inspección).

Cada control se enlaza a un objeto mediante la función `findViewById` (recibe de parámetro una `Tag` del `activity_main.xml`, y lo enlaza con un objeto en memoria). En la figura 1.5 se observa este mapeo de controles a objetos para el combo de selección del equipo a inspeccionar, y para el grupo de controles `radioButton` para marcar el estado del equipo:

```
cboEquipo = (Spinner) findViewById(R.id.cboEquipo);
RadGrupo_Estado = (RadioGroup) findViewById(R.id.grdGrupo);
rad_O = (RadioButton) findViewById(R.id.rado);
rad_X = (RadioButton) findViewById(R.id.radX);
rad_F = (RadioButton) findViewById(R.id.radF);
rad_N = (RadioButton) findViewById(R.id.radN);
```

Figura 1.5. Función `findViewById`

Estos objetos responderán a eventos ejecutando funciones declaradas en la clase `MainActivity` de la aplicación.

1.7.3 Otras Funciones importantes

Se utilizaron tres funciones importantes, con objetivos específicos:

- La función INICIA_INSPECCION, declarada dentro del código de la clase del Encabezado de la Inspección. Utilizada cuando se ingresa por primera vez al aplicativo, le pide al Manejador de tablas que guarde fecha y hora actuales como inicio de la inspección, y que guarde el estado de la inspección como ACT (la inspección está activa).
- GRABA_FILA, en las clases de Encabezado y de Detalle de la inspección, le indica al Manejador de tablas que guarde los datos actuales de la inspección.
- FIN_INSPECCION, contraria a la de INICIA_INSPECCION, esta función se utiliza cuando se ha dado click al botón de Finalizar la inspección; le pide al Manejador de tablas que guarde fecha y hora actuales como fin de la inspección, y que guarde el estado de la inspección como FIN; si el usuario intenta ingresar al aplicativo, esta inspección ya no estará disponible.

1.7.4 Interacción con otros programas externos

El ámbito de este aplicativo abarca el trabajo dentro del entorno de Android: por las condicionantes particulares que se expresaron anteriormente, la aplicación no tiene interacción con otros aplicativos, pues el dispositivo trabaja desconectado de la red. Sin embargo, otro

sistema de la Fundación Metrovía (escrito en Visual Basic .NET) que requiere la información de la inspección, la intercambia con el aplicativo Android a través de un puerto USB de la PC: en un módulo se recibe el archivo .db que la aplicación Android ha guardado en SQLite, y en otro módulo el sistema deja preparados los archivos de la inspección, los cuales se copian por USB al dispositivo Android. En la figura 1.6 se observa parte del código que verifica si el dispositivo está conectado:

	Id	código	nombre	ope
▶	1	T1	T1 - Troncal Gua...	6
	2	T2	T2 - T2 - Troncal 25 de Julio-Rio I	
	3	T3	T3 - Troncal Bast...	8

```

' verifica que el movil este conectado, retorna TRUE si es asi, caso contrario FALSO
' Muestra mensajes al usuario con el estado del Movil
Private Funcion VERIFICA_MOVIL() As Boolean
'Aqui va a tratar de abrir el archivo txt de Id del Movil
Dim exito_o_fracaso As Boolean = False
Dim el_ArchIdMovil As StreamReader = Nothing

' busca las unidades extraibles conectadas
Dim las_unidades As string()
' encuentra todas las unidades presentes
las_unidades = IO.Directory.GetLogicalDrives()
Dim Letraunidad As String
For Each LetraUnidad In las_unidades
    Dim el_drive As New IO.DriveInfo(LetraUnidad)
    If el_drive.DriveType = DriveType.Removable Then
        ' si es una unidad extraible, busca si existe el Archivo del Id del Movil
    
```

Figura 1.6. Otro sistema que consume la información levantada

CAPÍTULO 2

2. UNA INSPECCIÓN MÁS CONFIABLE

Mediante la aplicación de la solución detallada en el capítulo anterior, se lograron mejoras sustanciales en los procedimientos y en los resultados de las inspecciones.

2.1 Menor error humano

Antes de la aplicación de la mejora, se corría el riesgo de introducir errores humanos: durante la toma de información, ya que el inspector utilizaba una hoja de papel para asentar sus hallazgos. Si cometía un error se formaba un tachón, ilegible para quien manipule la hoja. Otro momento en el que se introducía un posible error humano era cuando otra persona tenía que digitar la información desde la hoja de papel para ingresarla al sistema de la Fundación.

Luego de haber entrado en uso la aplicación, el riesgo de error humano se reduce a mínimos niveles (no es cero mientras siga interviniendo el ser humano en el proceso), el inspector en sitio puede corregir en la Tablet cualquier error con solo un click; al viajar la información en un archivo no necesita ser digitada.

2.2 Ahorro en papel

Anteriormente a la implantación de la solución desarrollada, cada inspector utilizaba 6 hojas de papel, entre los 4 inspectores, se utilizaban 24 hojas de papel por cada inspección para las 3 troncales. Todas estas hojas se ahorrarán con la solución implementada.

2.3 Mejora de tiempos

Todo el proceso, desde la impresión del formato en papel hasta el fin de la digitación de la información recabada, tomaba hasta 1 semana en el peor de los casos. Luego de aplicada la solución, se tendrá la información lista a más tardar a primera hora del día siguiente, y solo si el inspector no logra retornar a oficina el mismo día de la inspección.

Todas estas mejoras redundan en inspecciones más confiables, y un mejor servicio a los usuarios del Sistema Metrovía.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. La aplicación de un elemento tecnológico bien utilizado en la inspección mejoró los procedimientos.
2. La inspección dió resultados más confiables y se realizó en menor tiempo, logrando un mejor control de cumplimientos y por tanto un mejor servicio entregado al usuario final.
3. El tiempo dedicado al análisis del problema y a la planificación de la solución rindieron sus frutos en un proyecto robusto y bien implementado.
4. El procedimiento de la inspección aunque muy conocido y arraigado en el personal, era mejorable, y se lo consiguió con herramientas simples.
5. Son innegables las ventajas obtenidas de haber desarrollado la aplicación con herramientas bien seleccionadas como: Android Studio y SQLite.

6. El aprovechamiento de un buen lenguaje de programación orientado a objetos, y un buen entorno de desarrollo, resultó en un código más fácil de entender, ordenado, y una ágil curva de entrenamientos ágil.

Recomendaciones

1. Está en plena vigencia el tema del servicio en el transporte público, en el caso analizado de la Metrovía, sus usuarios tienen pleno derecho de expresar libremente sus inconformidades en el servicio; la experiencia de los usuarios se puede mejorar, generalmente con la aplicación de soluciones muy simples en todos los procedimientos.
2. Actualmente está en franco crecimiento el número de usuarios de smartphones, la mayoría con Android como sistema operativo. Esto debe motivarnos a desarrollar aplicativos para estos móviles, y empezar a pensar más fuera del entorno de la oficina y mover nuestro bagaje de códigos un poco hacia el mundo móvil.

BIBLIOGRAFÍA

[1] Metrovía, ¿Qué es la Metrovía?,
<http://www.Metrovía-gye.com.ec/Metrovía> ,
Fecha de consulta febrero 2015

[2] Android SO, Qué es Android? Historia y características del sistema operativo,
<http://www.android-so.com/que-es-android-historia-y-caracteristicas-del-sistema-operativo> ,
Fecha de consulta febrero 2015

[3] Developer.android, Android Studio Overview,
<http://developer.android.com/tools/studio/index.html> ,
Fecha de consulta febrero 2015

[4] Vogella, Android SQLite database and content provider - Tutorial,
<http://www.vogella.com/tutorials/AndroidSQLite/article.html> ,
Fecha de consulta febrero 2015

ANEXOS

Anexo 1: Cronograma de fases y presupuestos referenciales del Proyecto

CRONOGRAMA DE FASES DEL PROYECTO	
ETAPA	SEMANAS
Análisis del problema y requerimientos	semana 1
Diseño: entidades, relaciones, datos, selección de herramientas	semanas 2 y 3
Programación	semanas 4 al 10
Implementación	semana 11
Pruebas y retroalimentación	semanas 12 y 13
Ajustes y correcciones	semanas 14 al 16
TOTAL	16 semanas

Anexo 2: Cronograma de fases y presupuestos referenciales del Proyecto

PRESUPUESTOS REFERENCIALES DEL PROYECTO			
CANT.	ITEM	COSTOS	
		UNITARIO	TOTAL
4	Tablets AOC ViewPad 7"	\$ 190.00	\$ 760.00
4	Licencia uso Android	\$ 0.00	\$ 0.00
4	Licencia uso SQLite	\$ 0.00	\$ 0.00
1	Licencia uso Android Studio	\$ 0.00	\$ 0.00
1	Licencia uso SQLiteManager Demo	\$ 0.00	\$ 0.00
1	Mano de Obra Programador	\$ 0.00	\$ 0.00
	TOTAL		\$ 760.00

Anexo 3: Diagrama Entidad-Relación

