

Inspección Digital de Equipos de Recaudo en Paradas y Terminales de la Metrovía

Gabriel Sosa Alvarado
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
papeles_2000@hotmail.com

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 extra 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.

Palabras Claves: *Android, Android Studio, SQLite, inspección, recaudo, móvil, Metrovía, MVRT, validador.*

Abstract

Metrovía Foundation, as a regulator entity for the Metrovía Transportation System, performs inspections to check the status of the control equipment at bus stops and terminals that make up the backbone of the system. These inspections are carried out by trained personnel, however, a problem arises on using procedures without any technological intervention; recording the findings on paper sheets, pre-printed forms. This, besides generating avoidable expenditure of paper, it can introduce human error in the process.

This creates an extra burden because an additional step is required, since someone must type the information from the paper, for being saved on digital information deposits in headquarters to be processed and analyzed.

Because of this, it was necessary to make a change of procedures, incorporating a technological element in the process: an application for mobile devices, the staff of inspectors will use to record the findings from the inspection.

This way it is improved the times that the information takes to be available from these inspections, errors margin is reduced, and a considerable saving of paper was achieved.

Keywords: *Android, Android Studio, SQLite, inspection, raised money, mobile, Metrovía, MVRT, validator.*

1. Introducción

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], supervisa la operación de cada una de las empresas que conforman el sistema, por ejemplo mediante inspecciones en sitio a equipos del recaudo tales como máquinas de venta y recarga de tarjetas, validadores, entre otros; pero 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.

Este artículo explica paso a paso como se fue implementando esta solución y el impacto que generó en los procedimientos involucrados.

2. Generalidades del Sistema Metrovía

La Metrovía es un conjunto de empresas privadas, entre estas, el Integrador Tecnológico que controla el recaudo del sistema, 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, mediante inspecciones en sitio a los equipos de recaudo, instalados en las paradas y terminales, verificando mediante muestreos su perfecto funcionamiento.

2.1. 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: Máquina Automática de Venta y Recarga de Tarjetas. El usuario por sí mismo adquiere una nueva tarjeta o hace su recarga, sin intervención de operador humano.
- Validador: Lector de tarjetas sin contacto, si encuentra saldo válido suficiente envía la señal eléctrica para desbloquear un torniquete.
- Taquilla.- para los usuarios que aun no poseen tarjeta, un operador humano le cobra su pasaje y libera el torniquete, permitiendo su ingreso al interior de la parada. Entre estos equipos consta la PC, mobiliario de la taquilla, etc. Su estado también juega un rol importante para la experiencia del usuario

2.2. 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, 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.

PROBLEMAS PREVIOS	CORRECCIONES PROYECTADAS
Hallazgos se anotan a mano en hoja de papel: - Correcciones son tachones - Las hojas se deterioran / traspapelan - Gasto de papel y tinta - Posible error humano	Datos guardados directamente en formato digital: - Correcciones son inmediatas - Los registros quedan asentados definitivamente - Información sólo reside en memoria del dispositivo - Se disminuye esta posibilidad
No hay constancia de las horas en las que se desarrolló la inspección	El aplicativo registraría las fechas y horas más importantes
Alguien debe luego digitar la información resultante: - Trabajo adicional - Se retrasa el reporte / análisis de la información - Posible error humano en la digitación	La información migra a los sistemas de la empresa sin trabajo adicional: - Ya no será necesaria la digitación - La información estará disponible en menor tiempo - Se elimina esta posibilidad

Figura 1. Problemas observados

3. Planificación de la solución

El objetivo del proyecto era 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.

Ciertas consideraciones particulares propias de la naturaleza de las actividades de los inspectores y de la empresa, condicionaron el diseño final:

- Debido a que las inspecciones son en sitio, era evidente que la solución debía ser móvil.
- Requería una aplicación móvil que permita al inspector registrar el estado de los equipos, luego pasar esa información a un depósito central, quedando a disposición de la Fundación para su procesamiento en los sistemas que ya existían para ese fin.
- El dispositivo móvil debía contar con una pantalla de tamaño medio, para que sea visible y manejable 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 y normados para toma de información.

4. Herramientas utilizadas

4.1. Android

Android es un sistema operativo sobre un kernel de Linux, especialmente diseñado para dispositivos móviles, y ampliamente aceptado por desarrolladores alrededor del mundo.

Las aplicaciones de usuario corren 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 pone 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]. Android 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.

4.2. 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 en java [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. Posee un buen debug, acceso rápido al emulador, ágil curva de aprendizaje para desarrolladores orientados a objetos y muy estable.

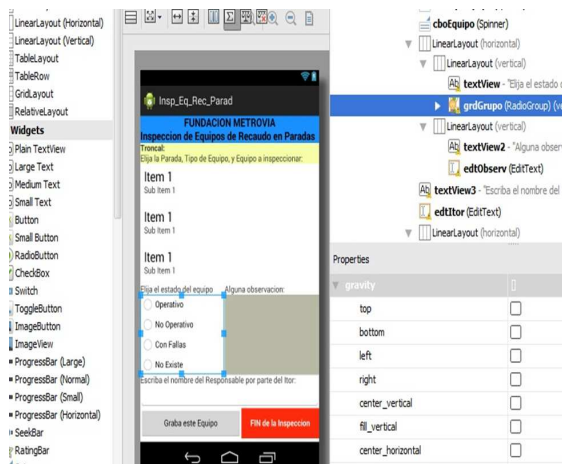


Figura 2. IDE Android Studio

4.3. SQLite

Un gestor de Base de Datos, muy ligero, todas las tablas, índices, relaciones, etc. se concentran en un solo archivo .DB. El soporte para SQLite viene embebido en cada dispositivo Android [4], por lo que las aplicaciones no necesitan conectarse con un proceso motor independiente, solo se enlazan con el gestor usando una simple librería.

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

5. Desarrollo de la solución

5.1. Entidades y clases

- Custodio de datos: Una clase heredada de SQLiteOpenHelper que maneja los niveles más bajos de abstracción, se encarga de las consultas y actualizaciones a las tablas. Usa un objeto de la clase SQLiteDatabase (db) que representa la base de datos, creada con un getWritableDatabase, mediante llamadas a sus métodos insert, execSQL, update, rawQuery, etc. Luego de usar esta clase db, la debe cerrar explícitamente con el método close. De esta manera proporciona toda la información que necesitan las consultas de las demás entidades, y hace todas las modificaciones a las tablas.

- Parada: Representa una parada de la Metrovía (código y nombre), el inspector debe seleccionarla de un listado.

```
/**
 * 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 3. Código constructor de la clase Parada

- Tipo de Equipos: Un objeto que representa una de las 3 clasificaciones de equipos para que el inspector pueda elegir.

- Equipo: Representa a un equipo (código y descripción).

- Encabezado de Inspección: Engloba todos los datos únicos de la Inspección (Id, nombre representativo, troncal, fecha programada, id del móvil, fechas-horas de inicio, fin y última modificación, estado actual de la inspección).

- Detalles de Inspección: Los datos de cada equipo inspeccionado (códigos de parada, del tipo de equipo, del equipo actual), estado del equipo (Operativo, No Operativo, con Fallas, o No Existe), observación.

5.2. Interfaz visual

Dentro de la IDE de Android Studio, se define la interfaz (ver la figura 2), probando los cambios con el emulador; se agregan uno a uno los controles al `activity_main.xml`: 3 Spinner (despliegan la parada, tipo de equipos, equipos), 4 RadioButton para cada posible estado del equipo (Operativo, No Operativo, con Fallas, No Existe), 2 EditText (uno para observación, y otro para un responsable de la empresa auditada), 2 botones (uno para Guardar cambios al equipo actual inspeccionado, 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).

```
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 4. Mapeo xml-a-objetos

Se observa el 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. Estos objetos ya pueden responder a eventos ejecutando funciones declaradas en la clase `MainActivity` de la aplicación.

5.3. Otras funciones importantes

- **INICIA_INSPECCION:** Declarada dentro del código de la clase de Encabezado de la Inspección. Se utiliza cuando se está entrando por primera vez al aplicativo, le pide al Manejador de tablas que guarde fecha-hora actuales como fecha de inicio de la inspección, y que guarde el estado de la inspección como ACT, que indica que la inspección esta activa.

- **GRABA_FILA:** Tanto en la clase de Encabezado como en la de Detalle de la inspección, le indica al Manejador de tablas que guarde los datos actuales de la inspección. En la tabla de Encabezado, se guarda el Responsable por parte de la empresa auditada, y la fecha-hora de cada modificación. En la tabla de Detalles, se guarda el Estado del Equipo y Observación.

- **FIN_INSPECCION:** En la clase del Encabezado, guarda Fecha-Hora de fin de la inspección y pone el estado como FIN, que indica que la inspección ha finalizado.

5.4. Interacción con programas externos

Otro sistema 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 5 se observa parte del código que verifica si el dispositivo está conectado.

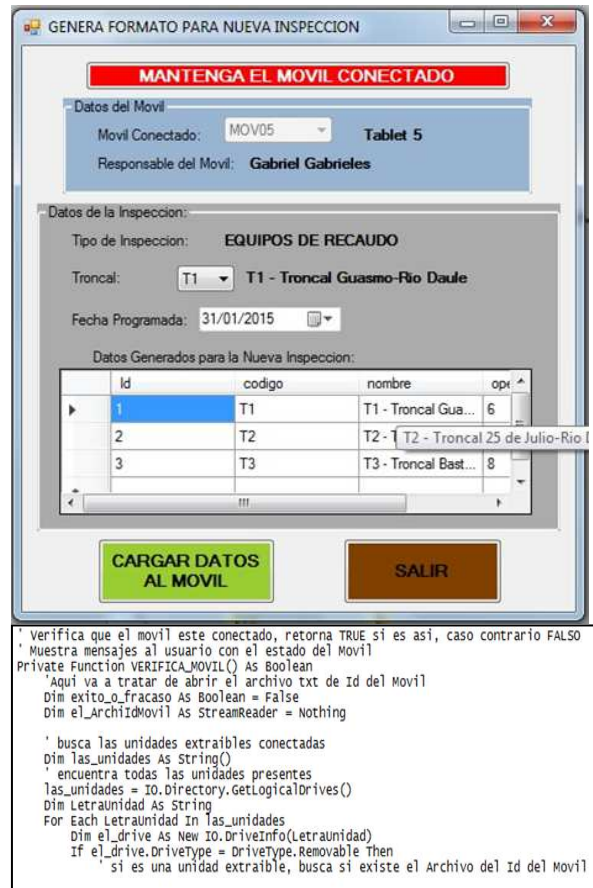


Figura 5. Interacción con sistema externo

6. Resultados: Una inspección más confiable

Mediante la aplicación de la solución detallada, se alcanzaron mejoras sustanciales en los procedimientos y en los resultados de las inspecciones.

- **Menor error humano:** Habiendo facilitado la tarea del inspector en sitio, y evitando la necesidad adicional de que alguien digite la información obtenida, se minimiza el riesgo de introducirse errores humanos al proceso. Aunque este riesgo no desaparece nunca, es inherente a todo proceso en el que intervenga el ser humano directa o indirectamente.
- **Ahorro en papel:** Con toda la importancia que tiene en la actualidad el tema ambiental, esto resulta muy gratificante.
- **Mejora de tiempos:** Resultado inmediato, reducción de tiempos de 1 semana a 1 día

7. Conclusiones

- Se mejoró un procedimiento con la aplicación de un elemento tecnológico simple bien utilizado.
- Inspección con resultados más confiables y realizada en menor tiempo redundó en un mejor servicio entregado al usuario final.
- El tiempo dedicado al análisis del problema y a la planificación de la solución rindieron sus frutos en un proyecto robusto y bien implementado.
- El procedimiento de la inspección aunque muy conocido y arraigado en el personal, era mejorable, y se lo consiguió con herramientas simples.
- Ventajas obtenidas de haber desarrollado la aplicación con herramientas bien seleccionadas como: Android Studio y SQLite.
- El aprovechamiento de un buen lenguaje de OOP, resultó en un código más fácil de entender, ordenado, y una curva de entrenamiento más ágil.

8. Recomendaciones

- 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; sin embargo, la experiencia de los usuarios se puede mejorar, generalmente con la aplicación de soluciones muy simples en todos los procedimientos.

- 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.

9. Agradecimientos

Mis más sinceros agradecimientos a todos mis amigos de la Fundación Metrovía por su apoyo y las facilidades para la realización de este proyecto; me brindaron el ánimo, las fuerzas y la confianza necesarias para la culminación.

10. Referencias

- [1] Metrovía, ¿Qué es la Metrovía?, <http://www.Metrovía-gye.com.ec/Metrovía>
- [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>
- [3] Developer.android, Android Studio Overview, <http://developer.android.com/tools/studio/index.html>
- [4] Vogella, Android SQLite database and content provider - Tutorial, <http://www.vogella.com/tutorials/AndroidSQLite/article.html>