



Facultad de Ingenierías y Ciencias Agropecuarias

Implementación de un prototipo de sistema de pago usando la tecnología NFC
para el Transporte publico

Trabajo de Titulación presentado en conformidad a los requisitos
establecidos para optar por el título de Ingeniero Electrónico y Redes de la
Información

Profesor Guía:

Ing. Eduardo Mauricio Campaña Ortega

Autor:

Carlos Ernesto Torres Vinueza

Año

2016

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos para un adecuado desarrollo del tema escogido, y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

Eduardo Mauricio Campaña Ortega
Ingeniero de Sistemas
C.I.:1708856701

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”.

Carlos Ernesto Torres Vinueza

C.I.: 1003826581

AGRADECIMIENTOS

A mis hermanos Jorge, María José y Francisco quienes juntos me han dado inspiración, fuerza y sobre todo valor para continuar en los momentos más difíciles.

Y a mí tutor el Ing. Mauricio Campaña quien me ha guiado con mucho empeño durante todo este proceso y ha sido un gran compañero en este paso tan importante de mi formación académica.

DEDICATORIA

A mis padres Jorge y Angélica, por ser mis cómplices y amigos en todo momento, por nunca dejar de confiar en mí y en todas mis metas propuestas.

RESUMEN

La tecnología NFC o Comunicación de Campo Cercano por sus siglas en inglés hoy en día representa el estándar RFid para dispositivos celulares, el cual en conjunto con tecnologías ya integradas se han creado un sin número de funciones y aplicaciones que obligan a los fabricantes a incluir este chip a más modelos cada día. Servicios como los de Apple Pay y Google Wallet no solo ofrecen un método innovador para la vinculación de tarjetas de créditos a los teléfonos celulares, sino que también entregan mucha más confianza a los procesos que un teléfono celular puede ofrecer en conjunto con la tecnología NFC.

En base a esto, en este proyecto se propone el diseño de un sistema que reemplace el actual método de pago para el transporte público, entendiendo como transporte público a las líneas de transporte Trolebús, Metrobús, Ecovía y líneas particulares de buses excluyendo al servicio de taxis debido a que las tarifas correspondientes a este servicio no son fijas, puesto que dependen de la distancia recorrida requerida por los usuarios.

El sistema a diseñar estará implementado bajo el sistema operativo Android y funcionara bajo los requerimientos que el sistema actual para el pago en el transporte público funciona actualmente.

Los usuarios podrán hacer uso del servicio a través de sus teléfonos celulares con chip NFC y podrán realizar consultas a sus datos mediante un aplicativo web basado en ASP.NET.

En este trabajo se detalla el proceso de diseño e implementación del sistema, utilizando un teléfono móvil como punto de partida para el desarrollo de un prototipo para cumplir con los objetivos propuestos.

ABSTRACT

The NFC technology or Near Field Communication for its acronym today represents the RFid standard for mobile devices, which in conjunction with already integrated technologies have been created a number of functions and applications which require manufacturers to include this chip in more models every day. Services like Apple Pay and Google Wallet not only offer an innovative method for bonding credit cards to cell phones, but it also delivered much more confidence to the processes that a cell phone can offer in conjunction with NFC technology.

On this basis, this project intends to design a system that will replace the current method of payment for public transportation, understanding public transport as transport lines like Trolebús, Metrobús, Ecovía, and private bus lines excluding taxi service since fees for this service are not static, since they depend on the distance required by the users.

The system design will be implemented under the operating system Android and will operate under the requirements that the current system for the payment in transportation service currently operates.

Users can make use of the service through their mobile phones with NFC chip and can make an enquiry to their data via a web application based on ASP.NET.

This work details the process of design and implementation of the system, using a mobile phone as a starting point for the development of a prototype to meet the proposed objectives.

ÍNDICE

1. INTRODUCCIÓN.....	1
1.1. Título del proyecto	2
1.2. Formulación del problema	2
1.3. Objetivo general	5
1.4. Objetivos específicos.....	5
2. MARCO TEÓRICO	6
2.1. El Transporte Urbano en la ciudad de Quito.....	6
2.1.1. Metrobús Q.....	6
2.1.2. Funcionamiento del Servicio	10
2.2. La Tecnología NFC	12
2.2.1. Funcionamiento.....	12
2.2.2. Principios de Comunicación	13
2.2.3. Modos de Funcionamiento	14
2.2.4. Estándares Relacionados.....	15
2.2.5. Integración de NFC en terminales Móviles.....	16
2.2.6. Aplicaciones y su Impacto en la tecnología Móvil	17
2.2.7. Ventajas frente a códigos QR.....	18
2.3. Herramientas de Desarrollo.....	19
2.3.1. Visual Studio	19
2.3.2. PhpMyAdmin	20
2.3.3. Android Studio.....	20
2.3.4. Funcionalidad NFC en SDK's de Android.....	21
2.3.5. Características de los mensajes NFC - Clases	23
2.3.6. Excepciones	27
2.4. Aplicaciones Web.....	27
2.5. Capas de una Aplicación Web	28

2.5.1.	Capa de Presentación	28
2.5.2.	Capa de Negocio.....	29
2.5.3.	Capa de Datos	29
2.6.	Servicios Web	30
2.6.1.	Estándares Relacionados.....	30
2.6.2.	Ventajas y Desventajas del uso de Servicios Web.....	31
2.6.3.	Plataformas	32
2.6.4.	Cloud Computing o Computación en la Nube	32
2.7.	Metodología	34
2.7.1.	Fases de la Metodología OOHDM	34
2.7.2.	Especificación de Requerimientos IEEE 830	35
3.	DISEÑO	36
3.1.	Análisis de Requerimientos	36
3.1.1.	Requerimientos	36
3.1.2.	Identificación de Actores y Tareas.....	41
3.1.3.	Especificaciones de Escenarios	43
3.1.4.	Especificaciones de casos de uso por Actores	44
3.2.	Metodología de Diseño de Hipermedia Orientada a Objetos (OOHDM)	53
3.2.1.	Diagramas de Secuencia	53
3.2.2.	Diagramas de Clases	56
3.2.3.	Diseño de Navegación	57
3.2.4.	Diseño de Interfaces.....	59
3.2.5.	Arquitectura del Sistema	67
3.3.	Consideraciones de implementación en la Fase 2.....	69
3.3.1.	Servidores de Aplicaciones	71
3.3.2.	Servidores de Base de Datos.....	72
4.	DESARROLLO Y PRUEBAS	73
4.1.	Desarrollo de la Base de Datos en MySQL	73

4.1.1.	Propósito de la Base de Datos	73
4.1.2.	Definición de Tablas Necesarias (Diseño Conceptual).....	73
4.1.3.	Definición de Campos Necesarios (Diseño Lógico)	74
4.1.4.	Definición de Relaciones (Diseño Físico)	79
4.2.	Desarrollo del Aplicativo Web en ASP.net.....	82
4.2.1.	Programación por Páginas	82
4.2.2.	Prueba de Desempeño (WEB)	85
4.3.	Desarrollo del Aplicativo en Android.....	86
4.3.1.	Interfaz Grafica	86
4.3.2.	Programación de Mensajes NDEF	88
4.3.3.	Intents para uso de la Comunicación P2P.....	88
4.3.4.	Envío de parámetros a través de JSON	89
4.3.5.	Configuración de Servicios Web en PHP	89
4.3.6.	Pruebas de Rendimiento (GameBench Ver. 3.2.2)	90
5.	CONCLUSIONES Y RECOMENDACIONES.....	93
5.1	Conclusiones	93
5.2	Recomendaciones	95
	REFERENCIAS.....	96

ÍNDICE DE TABLAS

Tabla 1. Porcentaje de versiones de distribución de SO Android Marzo 2016.	22
Tabla 2. Requerimiento Funcional RFQ-001: Registro de Usuarios.....	37
Tabla 3. Requerimiento Funcional RFQ-002: Clasificación de Usuarios.....	37
Tabla 4. Requerimiento Funcional RFQ-003: Registro de Uso	38
Tabla 5. Requerimiento Funcional RFQ-004: Registro de Datos	38
Tabla 6. Requerimiento Funcional RFQ-005: Facturación	39
Tabla 7. Requerimiento Funcional RFQ-006: Análisis de Datos	39
Tabla 8. Registrar (GUI Android).....	44
Tabla 9. Iniciar Sesión (GUI Android).....	45
Tabla 10. Generar Ticket (GUI Android).....	45
Tabla 11. Consultar (GUI Web).....	46
Tabla 12. Registrar (GUI WEB).....	47
Tabla 13. Iniciar Sesión (Android Prototipo).....	47
Tabla 14. Modo Lectura (GUI Android Prototipo).....	48
Tabla 15. Registrar (Prototipo).....	49
Tabla 16. Leer.....	50
Tabla 17. Conceder Acceso.....	50
Tabla 18. Comparación de aspectos técnicos.....	92

ÍNDICE DE FIGURAS

Figura 1. Tabla estadística del crecimiento de la posesión de celulares en el hogar.....	3
Figura 2. Tabla estadística de la posesión de teléfonos inteligentes en el Ecuador	4
Figura 3. Corredores del Metrobús-Q y sus estaciones de transferencia.....	7
Figura 4. Vehículo articulado usado en el servicio de Trolebús.	7
Figura 5. Estación Simón Bolívar del sistema Ecovía.	9
Figura 6. Diagrama de flujo del funcionamiento del servicio de transporte.	11
Figura 7. Funcionamiento de etiqueta NFC.....	12
Figura 8. Estándares relacionados a la tecnología NFC	16
Figura 9. Grafico comparativo de la Tabla 1.....	23
Figura 10. Actor Usuario y su relación con el actor GUI.....	41
Figura 11. Caso de uso del actor Usuario.	44
Figura 12. Caso de uso del actor Conductor.....	46
Figura 13. Casos de uso del GUI para Usuarios	48
Figura 14. Caso de uso del GUI para Conductores.....	49
Figura 15. Diagrama de Secuencia Registro (flujo normal).....	53
Figura 16. Diagrama de Secuencia Registro (flujo alternativo 1).	53
Figura 17. Diagrama de Secuencia Registro (flujo alternativo 2).	54
Figura 18. Diagrama de Secuencia Ingreso.	55
Figura 19. Diagrama de Secuencia Solicitar Ticket.....	55
Figura 20. Diagrama de Secuencia Consultar datos.	56
Figura 21. Diagrama de Clases del aplicativo Android (Prototipo).	56
Figura 22. Diagrama de Clases del aplicativo Android.....	57
Figura 23. Diagrama de Navegación Aplicativo Android	57
Figura 24. Diagrama de Navegación Aplicativo Android (Prototipo).....	58
Figura 25. Diagrama de Navegación Aplicativo We	58
Figura 26. Interfaz Android 1.1. Página de Inicio.....	59
Figura 27. Interfaz Android 1.2. Página de Inicio (Sesión Activa).....	60
Figura 28. Interfaz Android 1.3. Página Solicitar Acceso	61
Figura 29. Interfaz Android 1.4. Página Iniciar Sesión	62

Figura 30. Interfaz Android 1.5. Página Registro de Usuario	63
Figura 31. Interfaz Android 2.1. Página Principal	64
Figura 32. Interfaz Android 2.2. Página Principal (Sesión Activa)	65
Figura 33. Interfaz Web 3.1. Página Principal	66
Figura 34. Arquitectura del Sistema	67
Figura 35. Diagrama de flujo del sistema propuesto.	68
Figura 36. Diagrama de interacción entre app Android y Servidor de Base de Datos.	69
Figura 37. Placa base Raspberry Pi 3 + Modulo NFC.....	70
Figura 38. Tipos de Controles mecánicos de acceso.	71
Figura 39. Relaciones de la Tabla “user”.	79
Figura 40. Relaciones de la Tabla “ticket”.	79
Figura 41. Relaciones de la Tabla “bus”.....	80
Figura 42. Esquema final de la base de datos “Transporte Q”.	81
Figura 43. Página Default.aspx.	82
Figura 44. Página Login.aspx.....	82
Figura 45. Página Registro.aspx.	83
Figura 46. Página Consultas.aspx.....	84
Figura 47. Consola de Prueba de Google Chrome	85
Figura 48. Capturas de Pantalla Aplicativo Android para el Conductor.....	87
Figura 49. Comparación física de dispositivos usados en la prueba.	91
Figura 50. Comparación del uso de CPU.....	92
Figura 51. Comparación del uso de GPU.....	92

1. INTRODUCCIÓN

Hoy en día el servicio de transporte público no satisface la demanda básica del usuario y del lado del proveedor se registran numerosos incidentes, muchos de estos problemas no estimulan el uso de este servicio y esto se ve reflejado en el incremento del parque automotor.

El sistema de pago del servicio ha sido el mismo desde su creación y se puede considerar como la raíz de problemas como la inseguridad que sufre un chofer al recaudar el efectivo en la unidad como caja chica, en el cual se necesita una persona que esté a cargo del cobro en los diferentes tipos de transporte público (Trolebús, Metrobús Q, Ecovía y líneas particulares de buses) ya sea como cobrador en casetas ubicadas en cada parada de los articulados o como controlador de cada unidad en las líneas de buses.

Es por ello que nace la idea de desarrollar un prototipo que reemplace el actual método de pago, agregando seguridad al servicio al independizar el dinero en toda transacción.

Al cambiar el método de pago se aprovechan las propiedades de registro del software a implementar, en el cual se guardan datos del usuario, el conductor, la unidad de transporte, la hora de abordaje y la ruta establecida. Estos datos del lado del usuario, ayudarían a registrar su gasto destinado a este servicio. Del lado del proveedor, ayudaría a registrar los ingresos de una manera exacta, también, se tendría un control sobre las personas que ingresan a la unidad y en caso de existir un accidente o asalto se tendría los datos de todas las personas que están en el vehículo a determinada hora, mejorando así la seguridad. La base de datos beneficiaría también a la reorganización de las rutas establecidas y la cantidad de unidades dispuestas para cada ruta, utilizando datos estadísticos referentes a la demanda por sectores y en función de la hora en el día.

El prototipo a desarrollar será emulado en un teléfono celular ya que en este se disponen de todos los componentes necesarios para el funcionamiento del sistema y es una alternativa económica antes de la segunda fase que supondría la implementación del proyecto. Por esta razón se desarrollara 2

aplicativos basados en el sistema operativo Android, en los cuales uno servirá para el registro del conductor y la unidad de transporte y el segundo permitirá a los usuarios iniciar sesión, registrarse y finalmente pagar el servicio utilizando la sincronización que la tecnología NFC permite.

Finalmente, se realiza un análisis de resultados con la información obtenida de los resultados de las pruebas de conexión y funcionamiento. El prototipo estará conectado a la red y registrara en todo momento los datos de las interacciones con los usuarios. Dicho celular que conforma el prototipo dará acceso después de confirmar el registro valido de los usuarios, de esta forma no es imprescindible que el usuario tenga acceso a la red en el momento en que solicite el acceso si es que previamente se registró en su terminal móvil.

El usuario tendrá acceso al servicio mediante una aplicación móvil que lo podrá descargar con facilidad y únicamente requiera el registro de sus datos tomando en cuenta su cédula de identidad como número de usuario.

Desde dicho aplicativo el usuario podrá hacer las consultas relacionadas a su consumo. Para las funciones de pago deberá acceder por la herramienta web con su cuenta y definir las opciones de pago.

1.1. Título del proyecto

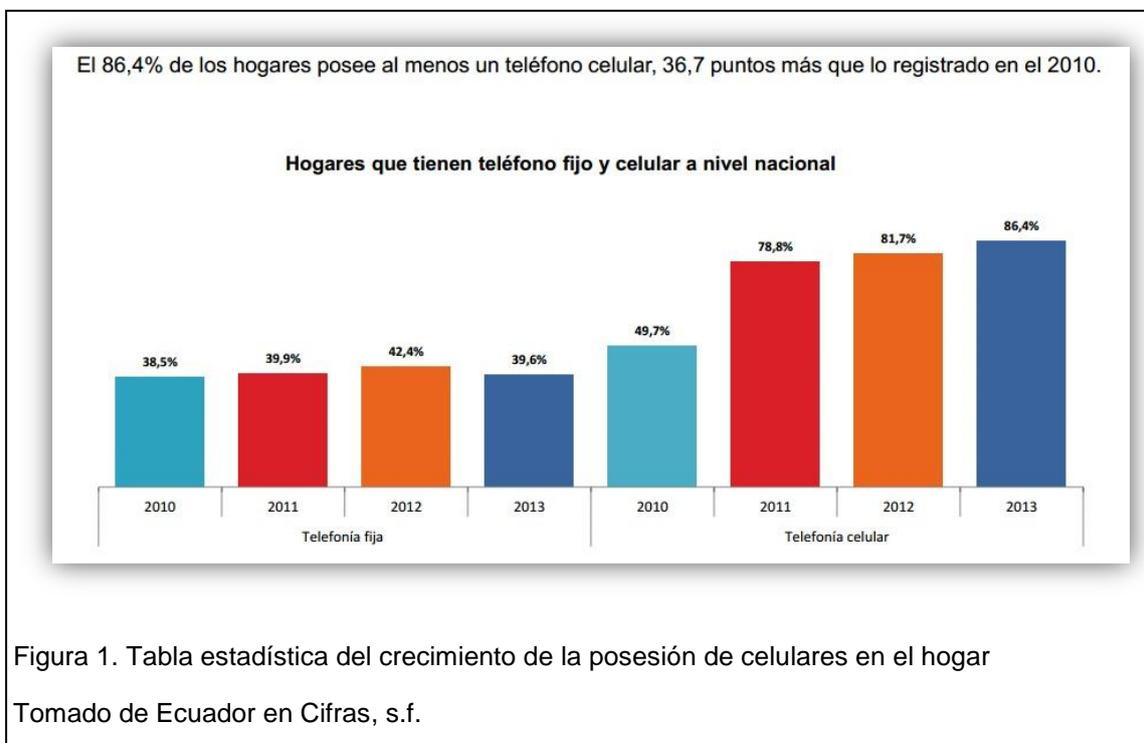
“Implementación de un prototipo de sistema de pago usando la tecnología NFC para el transporte público”

1.2. Formulación del problema

El Identificar la posibilidad de implementar varias tecnologías en el proceso normal de cobro del transporte público de la ciudad de Quito. Hoy en día se tiene a disposición una red de comunicación distribuida en toda la ciudad por la que se puede mantener una comunicación permanente en cualquier ruta establecida por las líneas de transporte público.

Además, actualmente se manejan terminales celulares capaces de registrar las transacciones que utilizamos a diario para usar este servicio. Se toma en consideración también el auge de la tecnología NFC en los dispositivos móviles, los cuales ya no requieren ser de gama alta para utilizar dicha comunicación.

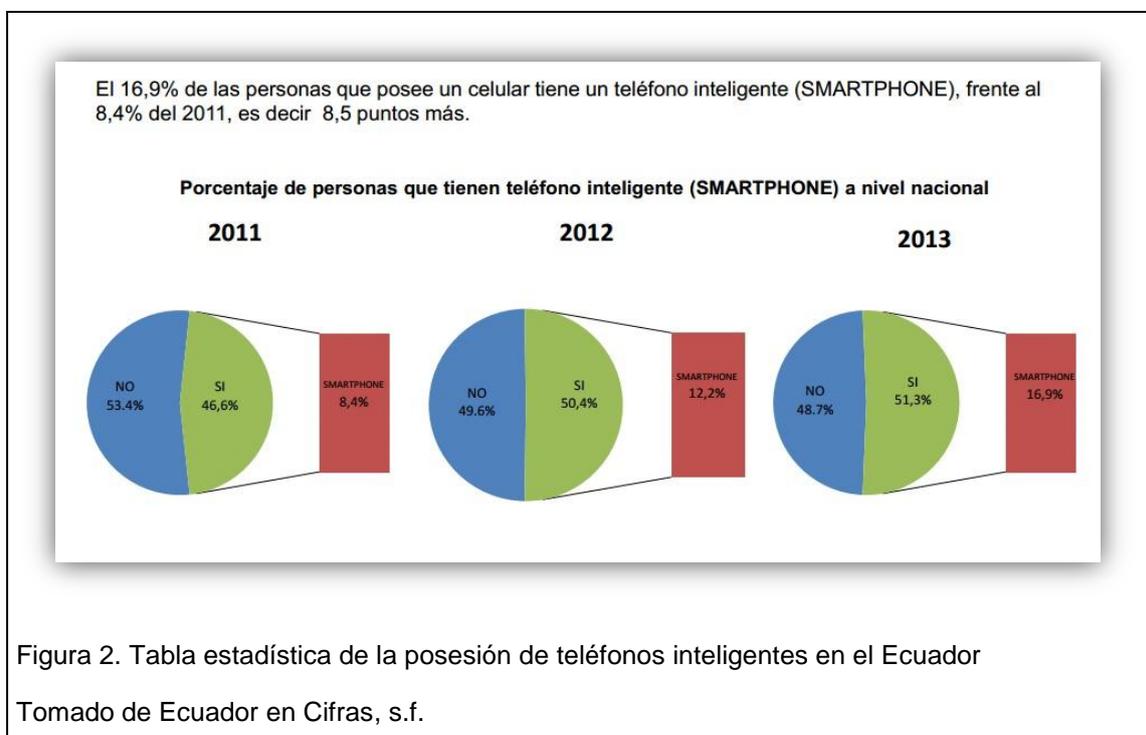
Bajo esta premisa el crecimiento de la posesión de teléfonos celulares en Ecuador también se ve reflejado. Como vemos en la figura 1, en 2013 se registra un porcentaje de 86.4% de posesión de teléfonos celulares en los hogares ecuatorianos.



De este porcentaje es importante recalcar que no todos los dispositivos son teléfonos inteligentes o tienen la capacidad de ejecutar aplicaciones complejas, por lo que supondría un limitante para aquellas personas que no poseen un teléfono inteligente. No obstante el crecimiento de modelos de teléfonos celulares inteligentes va en aumento debido a que las compañías fabricantes están desarrollando dispositivos más accesibles económicamente y que ofrecen capacidades similares a los modelos Premium.

Fabricantes como Apple y Samsung han lanzado versiones económicas de sus dispositivos Premium a un bajo costo, sacrificando materiales en su estructura, reduciendo las capacidades de procesamiento, almacenamiento, duración de batería, etc. Permitiendo que más personas puedan acceder a los teléfonos inteligentes y que hasta 2015 se registre que el acceso a internet se lo realiza en un 85% desde teléfonos celulares.

Esto de igual manera se ve reflejado en las estadísticas en países como Ecuador como podemos ver en la figura 2 que hasta el 2013 el 16.9% de teléfonos celulares en Ecuador, son Smartphones a comparación de 2011 en el que la cifra es de 8.4%.



Debido a esto, es necesario la creación de aplicaciones y servicios que satisfagan a este nuevo nicho de mercado, por lo que se ha optado por el desarrollo de un servicio que una a todas estas tecnologías para tener un mejor asistencia, optar por simplificar el proceso de cobro y sobre todo brindar seguridad tanto al conductor como a los usuarios de las unidades.

Es por esto que destinar dinero al pago de un tercero actor en el sistema se hace innecesario, debido a que se posee los elementos para desplegar un servicio orientado al consumidor, en donde se aprovechan las tecnologías ya

implementadas en el entorno. Dicho servicio no requiere ser afiliado a la seguridad social, tampoco se deben realizar los pagos de horas extras y suplementarias, pagos de décimo tercer sueldo, pagos de décimo cuarto sueldo, pagos del fondo de reserva, vacaciones anuales, pagos de la jubilación patronal, pago por licencia de paternidad ni maternidad ni tampoco el pago de utilidades. Valores adicionales al pago mensual de la remuneración normal que se recibe por el trabajo realizado de una persona.

1.3. Objetivo general

Implementar un sistema de pago aprovechando la tecnología NFC del móvil en los diferentes servicios de transporte.

1.4. Objetivos específicos

- Diseñar un sistema que reemplace el actual método de pago en el transporte público.
- Implementar un prototipo que funcione a través de la tecnología NFC con el fin de comunicar datos desde un teléfono móvil.
- Desarrollar una aplicación móvil que genere los datos necesarios para realizar las interacciones con el prototipo y por medio de un servicio web los almacene en una base de datos.
- Desarrollar una aplicación que despliegue la información de la base de datos y haga las funciones de servidor del sistema a implementar.

2. MARCO TEÓRICO

2.1. El Transporte Urbano en la ciudad de Quito

El transporte urbano es un servicio enfocado al transporte colectivo de pasajeros. A diferencia del transporte privado los usuarios de este servicio deben adaptarse a los horarios y rutas establecidas por cada operador. La demanda del servicio está dada por los pasajeros y la oferta está dada por los vehículos y se mantienen mediante el cobro directo a los usuarios.

El servicio varía en cada región en donde funciona debido a las diferencias geográficas, viales y sociales, por lo que cada servicio modifica sus rutas en tiempo y longitud de recorrido para satisfacer de mejor manera la demanda diaria.

2.1.1. Metrobús Q

En la ciudad de Quito se conoce como Metrobús Q al sistema de corredores exclusivos de transporte público, compuesto por buses articulados BRT y autobuses normales que alimentan a los corredores y amplían la cobertura a los sectores donde los articulados no llegan.

El Metrobús Q está conformado por cinco líneas de Corredores, siendo los primeros el corredor Trolebús y Corredor Ecovía, que fueron posteriormente extendidas con los corredores Central Norte, Sur Oriental y Corredor Sur Occidental. Al momento se estudia la posibilidad de construir un sexto corredor denominado Nor Oriental, además de la planificación de varios corredores hacia las principales parroquias rurales del distrito y hacia el Aeropuerto Internacional Mariscal Sucre.

A continuación se observa en la figura 3, la disposición de las diferentes líneas de corredores y estaciones principales que el sistema Metrobús Q actualmente conforma.



2.1.1.1. Trolebús

La primera etapa del Trolebús en la ciudad de Quito se inauguró con 14 unidades el 17 de diciembre de 1995 en donde se registró el traslado de un promedio de 50.000 pasajeros en el tramo comprendido entre la Estación Sur El Recreo y la calle Esmeraldas, más las líneas alimentadoras del Sur.

Este proyecto nació de la necesidad apremiante de mejorar la movilidad en la ciudad, disminuir la contaminación y ofrecer un servicio de transporte enfocado en sus usuarios. Proyecto que desde sus inicios en el año 1990 en conjunto con el Plan Maestro de Transporte de la ciudad busco solucionar uno de los problemas más relevantes en el ámbito del transporte en el distrito.



Figura 4. Vehículo articulado usado en el servicio de Trolebús.

Tomado de Trolebus, s.f.

Hoy a 20 años de su implementación conforma una parte fundamental del transporte urbano de la ciudad con 3 etapas integradas con éxito en donde se incluyeron vías desde El Recreo hasta La Colon sumando un total de 32 unidades el 18 de agosto del año 2000; año en el que se inauguró la tercera etapa, desde la Estación Sur El Recreo hasta la Estación Norte de La Y, con 113 unidades de doble vagón como se observa en la figura 4, teniendo un transporte promedio de 120.000 pasajeros.

Este sistema integrado es una red de líneas de transporte urbano que se enlazan entre sí mediante estaciones de transferencia y paradas de integración. Esto permite a los usuarios diseñar rutas que los trasladen a cualquier punto de la ciudad pagando un único pasaje, los 365 días del año, las 24 horas.

2.1.1.2. Ecovía

Planificado durante la alcaldía del economista Roque Sevilla, el corredor Ecovía fue inaugurado en el año 2001, a inicios de la administración de Paco Moncayo.

Ecovía fue el primer corredor del Sistema Metrobús Q en ocupar unidades de puerta izquierda y motor a diésel, característicos en la ciudad por su color rojo como se ve en la figura 5 junto a una de sus paradas. Convirtiéndose así en el segundo sistema planificado de transporte masivo de la urbe.

Mediante sus carriles exclusivos, Ecovía recorre parte del flanco nor-oriental de la capital a través de las avenidas 6 de Diciembre, Gran Colombia y Pichincha; conectándola desde el centro-norte, en la estación Rio Coca, hasta el centro histórico, en la estación multimodal Playón de la Marín. Al igual que el servicio de corredor Trolebús Q, este se extiende a los sectores periféricos mediante un sistema de buses urbanos o alimentadores.



Figura 5. Estación Simón Bolívar del sistema Ecovía.

Tomado de Trolebus, s.f.

El sistema Ecovía cuenta con 20 paradas a lo largo de su recorrido; una de ellas es de carácter integrador, es decir que sirve de conexión entre la Ecovía y los autobuses alimentadores que se dirigen a los barrios periféricos; 2 multimodales y finalmente 27 paradas regulares.

2.1.1.3. Buses Urbanos

Las rutas urbanas en Quito no poseen una frecuencia definida de recorrido. Existen numerosas cooperativas que la conforman y esto hace que la calidad del servicio varíe de empresa a empresa, que en conjunto “alimentan” los 5 corredores o sistemas integrados que manejan carriles exclusivos de superficie y paradas determinadas por estructuras uniformes.

A diferencia de los sistemas articulados, el servicio de buses urbanos no depende de paradas establecidas para la recolección de pasajeros, siendo este un factor que busca ser regulado por las entidades de control de tráfico y vialidad.

En Quito existen 134 rutas operadas por aproximadamente 2600 buses, identificadas por la pintura azul en las carrocerías que recorren el área de la

urbe alimentando los corredores principales y finalmente conformando la SITM-Q o Sistema Integrado de Transporte Metropolitano en conjunto con las compañías privadas de taxi y el sistema de bicicleta publica denominado BiciQuito.

2.1.2. Funcionamiento del Servicio

Pese a que el servicio de transporte urbano es un macro sistema integrado y que es manejado por el municipio en conjunto con varias compañías privadas estableciendo más de 200 rutas en toda la ciudad, el funcionamiento puede ser considerado constante en cada uno de ellos.

Se maneja una tarifa regular de \$0.25 y \$0.12 para estudiantes, personas de la tercera edad y personas con capacidades especiales.

El pago se lo realiza al ingreso de las estaciones (en el caso de los sistemas de buses articulados) o dentro de las unidades en los buses normales. Para ello existe la necesidad de un controlador o cobrador, que recolecte el dinero de los usuarios independizando al conductor en el proceso.

En el sistema se identifican a 3 actores: Los usuarios, el conductor y el controlador/cobrador. En algunos casos en el servicio de líneas particulares de buses no se cuenta con la presencia del controlador, por lo que el conductor hace las veces de cobrador interrumpiendo su función principal.

En el siguiente diagrama de flujo se representa el proceso por el que un usuario hace uso del servicio de transporte. Se simplifica en el mismo diagrama los casos en el que se ingresa directamente al medio de transporte o cuando se ingresa inicialmente a una estación.

El diagrama presentado servirá también para la recolección de requerimientos finales en el desarrollo del software del sistema. Al ser un servicio en el que se cumple un proceso fijo la obtención de requerimientos se simplifica y no depende de la recolección de información de los usuarios finales para alcanzar los objetivos del servicio.

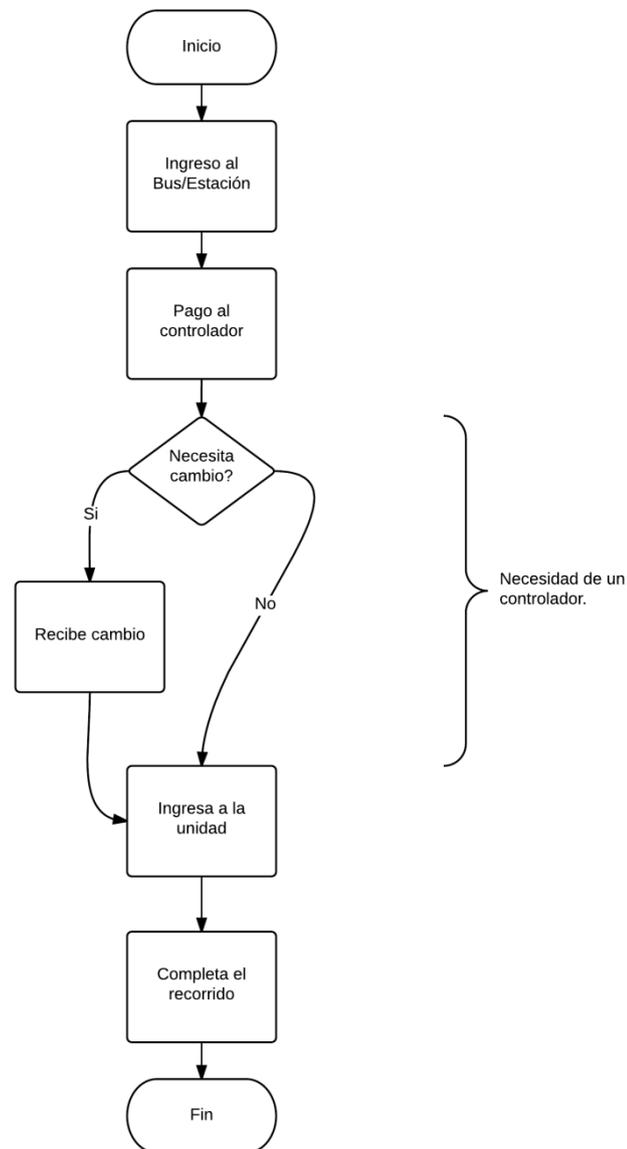


Figura 6. Diagrama de flujo del funcionamiento del servicio de transporte.

Como se puede observar en la figura 6, en el sistema actual de cobro se hace evidente la necesidad del controlador, en el caso de los sistemas articulados este actor representa a la persona que entrega el cambio en monedas con la denominación correcta para ingresar, pero su función en el sistema es similar.

El sistema propuesto pretende obviar esta necesidad y re direccionar el costo que representa este actor, en mejoras al servicio.

2.2. La Tecnología NFC

La tecnología NFC por sus siglas que hacen referencia al término Near Field Communication o “Campo de comunicación cercano” fue desarrollado a partir de la tecnología RFID en el año 2002 como un intento por conseguir compatibilidad entre las tecnologías Mifare y FeliCa de Philips y Sony respectivamente, pero no es hasta 2004 que se logra el primer intento exitoso de estandarizar protocolos y formatos para lograr una mayor promoción de esta tecnología en el NFC fórum, fundado en 2004 por Nokia, Phillips y Sony inicialmente.

2.2.1. Funcionamiento

NFC se comunica mediante inducción en un campo magnético, en donde 2 antenas de espiral son colocadas dentro de sus respectivos campos cercanos (Figura 7). Trabaja en la banda de los 13,56 MHz por lo que no se considera necesario que requiera una licencia para su uso.



Su velocidad de transferencia depende del entorno en el que se trabaje o en el acuerdo de ambos dispositivos en la transferencia y varía entre 106, 212, 424 o 848 Kbits/s, siendo esta velocidad configurada en cualquier instante de la comunicación.

Debido a su limitada cobertura, se considera una gran ventaja al resultar idóneo para atender servicios que impliquen una necesaria privacidad. Además, al tener ambos dispositivos tan cerca, se evitan errores en la comunicación y se asegura una mayor eficacia en la transmisión de datos.

2.2.2. Principios de Comunicación

A diferencia de otros dispositivos como Bluetooth y wifi, en NFC se necesita una aproximación lo suficiente cerca para que sus campos magnéticos entren en contacto y es ahí cuando se produce el acoplamiento magnético.

Un dispositivo NFC puede comunicarse con cualquier tarjeta inteligente, lector u otro dispositivo NFC. Dependiendo de la función que realice: enviar o recibir datos, el dispositivo NFC toma una de las siguientes funciones:

- **Iniciador (initiator):** Como su nombre lo indica es quien inicia y controla el intercambio de información (el equivalente al lector en los sistemas RFID).
- **Objetivo (target):** Es el dispositivo que responde a los requerimientos del iniciador.

Cualquier dispositivo con NFC (excepto una etiqueta NFC) puede operar de las dos formas: como iniciador o como objetivo.

La comunicación NFC consta de cinco fases las cuales siempre están presentes en el establecimiento de esta. Estas etapas son:

- **Descubrimiento:** Primera fase en la que ambos dispositivos se rastrean y posteriormente se reconocen.
- **Autenticación:** Se verifica si ambos dispositivos están autorizados o si deben establecer algún tipo de cifrado para la comunicación.
- **Negociación:** Se definen parámetros como: velocidad de transmisión, identificación del dispositivo, tipo de aplicación, tamaño y la acción a ser solicitada.
- **Transferencia:** Se procede a realizar el intercambio de datos.

- Confirmación: El dispositivo receptor confirma el establecimiento de la comunicación y la transferencia de datos.

Se debe aclarar que la Tecnología NFC no está destinada a la transferencia masiva de datos, pero es utilizada para reducir el establecimiento de la comunicación de otras tecnologías como Bluetooth o Wi-Fi, reduciendo considerablemente el tiempo comparadas a si se utilizaran esas tecnologías por si solas para efectuar el enlace.

2.2.3. Modos de Funcionamiento

Debido a su estándar NFCIP-1 todos los dispositivos deben soportar 2 tipos de funcionamiento:

- Activo (Solo dispositivos): ambos dispositivos generan su propio campo electromagnético, que utilizaran para transmitir sus datos.
- Pasivo (Dispositivos y etiquetas NFC): solo un dispositivo genera el campo electromagnético y el otro se aprovecha de la modulación de la carga para poder transferir los datos. El iniciador de la comunicación es el encargado de generar el campo electromagnético.

Una vez definido el tipo de funcionamiento, los dispositivos pueden optar por tres modos de funcionamiento distintas:

- Modo lector/grabador con capacidad de lectura y escritura de etiquetas. En esta configuración el dispositivo NFC es capaz de leer etiquetas compatibles con el estándar ISO-14443. Además, es posible la transferencia de una pequeña cantidad de información a la etiqueta. Esta información puede ser un texto, una página web o un número de identificación. Cada etiqueta puede ser re escrita cuantas veces se desee si es que no se asegura la información guardada, es decir que si se configura, la escritura puede ser permanente.

- Modo “Peer to Peer” usada para el intercambio de datos o el establecimiento de comunicaciones entre dispositivos NFC. Cuando la cantidad de datos intercambiada es pequeña se usa el mismo protocolo NFC. Para la transmisión de mayores cantidades de datos, NFC se usa para establecer parámetros de una conexión inalámbrica más avanzada.
- Modo emulación de tarjeta inteligente. En este modo el dispositivo se comporta como una etiqueta NFC, reemplazando una tarjeta inteligente usada ante un lector externo. En esta configuración se utilizan las características de seguridad avanzada, permitiendo la posibilidad de reemplazar una tarjeta de crédito o de débito, realizando una transacción mucho más rápida y sencilla que cualquier medio tradicional de pago.

2.2.4. Estándares Relacionados

Al derivarse de la tecnología RFID, NFC es una extensión de la norma ISO/IEC 14443. Estándar que define el uso de tarjetas electrónicas de identificación en especial las tarjetas inteligentes.

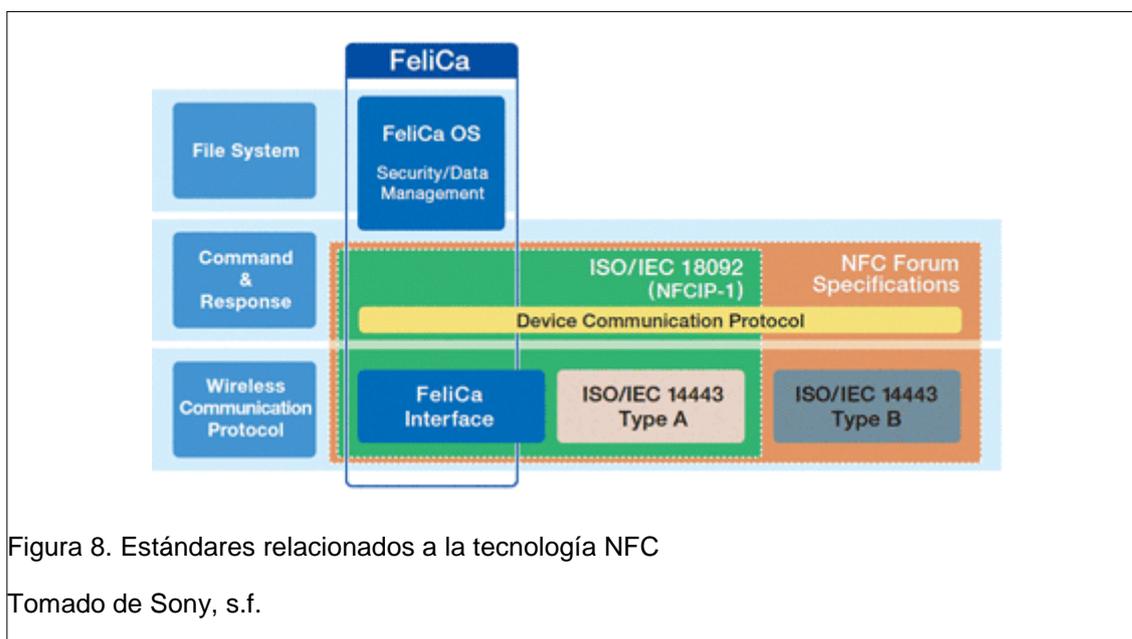
Dicho estándar especifica en 4 partes las características físicas, poder y señal de radiofrecuencia, inicialización/anticolisión y protocolo de transmisión. Mostrándonos las capacidades de esta tecnología con relación a otras soluciones inalámbricas.

De igual forma, otras organizaciones como la ASTM International, DASH7 Alliance y EPC Global han publicado sus propios estándares asociados a la tecnología RFID.

Además de la norma ISO/IEC 14443, NFC opera en la frecuencia no licenciada de ISM o Industrial, scientific and medical radio bands (Bandas de radio industriales, científicas y médicas), especificadas en la norma ISO/IEC 1800-3, norma que estandariza internacionalmente los parámetros asociadas a las interfaces de comunicación para RFID pasivas.

Similar a esta encontramos la ISO/IEC 7816 que se enfoca a las tarjetas electrónicas de identificación con contactos y en contra parte a FeliCa, que es una abstracción a Felicity Card en el que se describe una tarjeta de identificación sin contactos.

FeliCa es un sistema de tarjetas inteligentes de la compañía Sony es usada principalmente en tarjetas de dinero electrónico, usadas comúnmente en Hong Kong, Singapur, Japón y Estados Unidos.



2.2.5. Integración de NFC en terminales Móviles

La simplicidad que mostraba el funcionamiento de las tarjetas y etiquetas RFID en aplicaciones cotidianas impulsaron la integración a los móviles por parte de varias empresas fabricantes de estos dispositivos, lo que implicó un choque de incompatibilidad entre fabricantes y dispositivos debido a que cada fabricante imponía un sistema que pese a funcionar igual, utilizaba parámetros propios para la comunicación.

Es en el año 2004 cuando las compañías Nokia, Philips y Sony se asocian y forjan el termino NFC (Near Fiel Communication) como un intento por popularizar un solo estándar entre todos los sistemas creados hasta esa fecha.

El Nokia 6131 es el primer Smartphone con NFC, mientras que el Nexus S fue el primero con Android como sistema operativo. Actualmente, cada vez más fabricantes equipan a sus teléfonos móviles con emisores/receptores activos, teniendo esta función más como una parte primordial del móvil antes que una característica adicional.

Desde aplicaciones de identificación para conexión con otros sistemas de transferencia de archivos hasta aplicaciones de pago, la tecnología NFC se ha integrado completamente a los Smartphones de gama alta, media y baja, estando presente en aproximadamente 200 modelos de celulares.

2.2.6. Aplicaciones y su Impacto en la tecnología Móvil

Pese a la gran utilidad que muchas aplicaciones móviles tienen hoy en día, sufren de muchas limitaciones si los comparamos con aplicaciones complejas en las máquinas de escritorio y que no se comparan aun con aplicaciones de servidores en donde se manejan una gran cantidad de datos en comparación. Pero es desde 2007 con la aparición del iPhone que los desarrolladores tuvieron una nueva perspectiva del uso de un teléfono celular y sus aplicaciones en la vida cotidiana. Y es que sacando provecho de las características que ofrecen los terminales móviles, ahora las aplicaciones telefónicas no compiten con los software de escritorio, si no que se complementan, brindando funcionalidad de autenticación a los ya complejos sistemas que se manejan a diario.

Dichas características agregan movilidad, seguridad y dinamismos a tareas como las de por ejemplo, autenticarse a una página web utilizando un pin de seguridad enviado directamente al correo electrónico, subir una imagen a un portal web desde el lugar donde nos encontremos y muchos ejemplos más. Funciones que hacen uso de GPS, conectividad a la red, cámaras fotográficas, sensores de movimiento, giroscopios, etc. Y que con la inclusión de los chips NFC agregan un nuevo panorama de desarrollo.

El poder comunicar información sencilla inalámbricamente y ligada a la proximidad del dispositivo ha convertido a la tecnología NFC en el elemento base de muchos sistemas de identificación. Su función lo vincula a aplicaciones de publicidad, obtención de datos de presentación, pago con dinero electrónico y pago con tarjetas de crédito.

Actualmente el servicio Apple Pay de Apple y Google Wallet para Android son ejemplos de la funcionalidad que se obtiene al utilizar la tecnología NFC para identificación del usuario a sus tarjetas de crédito. Funciones que dan mucha más confianza al usuario en el manejo de su dinero que gracias a nuevos sistemas operativos ahora más seguros que en versiones pasadas se puede optar al manejo libre de esta información.

En terminales de gama alta, también se puede encontrar lectores de huella digital y funciones de reconocimiento fácil, elementos que agregan seguridad a la información y que permiten a los desarrolladores atreverse a brindar mayor libertad en el manejo de la información a la que los teléfonos celulares pueden acceder. Esta libertad representaría el poder controlar la seguridad de acceso a edificios, vehículos o inclusive las puertas de entrada a los hogares. Aplicaciones que van de la mano con el término “internet de las cosas”. Término que nos indica el flujo de información que rodea a la tecnología y que se complementa con aplicaciones simples y de gran utilidad.

2.2.7. Ventajas frente a códigos QR

Ciertos aplicativos móviles en los que se requiere la identificación de un dispositivo o la lectura de información pequeña se ha manejado a través de la implementación de códigos QR, los cuales por medio de la cámara del dispositivo móvil pueden ser leídos e interpretados para el despliegue y funcionamiento del aplicativo.

Pero representa también desventajas de funcionamiento como los listados a continuación:

- Requieren una luz adecuada de ambiente para la lectura de los mismos.
- No posee seguridad.
- Manejan múltiples estándares.

- Son códigos fijos, que limitan las interacciones en el sistema.
- Su funcionamiento no es soportado nativamente en dispositivos celulares.

2.3. Herramientas de Desarrollo

2.3.1. Visual Studio

Visual Studio es un entorno de desarrollo integrado en donde se albergan diferentes tipos de lenguaje de programación y entornos de desarrollo web. Maneja lenguajes de programación como C++, C#, Visual Basic, Java, Php, entre otros. Es una herramienta desarrollada por Microsoft y trabaja con el sistema operativo Windows.

Se usara el entorno de desarrollo web ASP.Net para crear el aplicativo web en donde se podrán ver visualizados los datos guardados en la base de datos.

ASP.Net permite la programación bajo cualquier lenguaje que este soportado por el .Net Framework. Gracias a esto se ha optado por el lenguaje de programación C#.

La versión de Visual Studio que se usara en este proyecto será Visual Studio 2013 Update 4 el cual es registrado como la última versión más estable del IDE. Existiendo también Visual Studio 2015 la cual es la última versión liberada el 12 de noviembre del 2014.

Actualmente Visual Studio representa uno de los entornos de desarrollo integrado más completo y enfocado en herramientas visuales, lo cual es una excelente ayuda al momento de programar aplicativos web e interfaces amigables para los usuarios.

Al tener cualquier entorno que soporte las plataformas .NET se pueden programar aplicaciones que se comuniquen entre Pc's, páginas web, aplicaciones móviles, entre otras.

2.3.2. PhpMyAdmin

PhpMyAdmin es una herramienta escrita bajo el lenguaje PHP que administra bases de datos en MySQL. Su interfaz permite la creación, modificación y monitoreo de bases de datos a través de páginas web. Es capaz de ejecutar sentencias en sql y su interfaz amigable está disponible en 72 idiomas, posicionándolo como una de las herramientas predilectas para el uso en equipos con servicios web.

Creada por Tobias Ratschiller en el año 1998, se convirtió rápidamente en una de las herramientas PHP más populares. Con contribuciones de Linux, su comunidad rápidamente creció con usuarios y administradores.

En este proyecto será utilizado para generar la base de datos basada en MySQL y se la usara para monitorear las interacciones entre aplicativos web y móviles.

2.3.3. Android Studio

Android Studio al igual que Visual Studio es un entorno de desarrollo integrado (IDE) pero exclusivo para la plataforma Android, sistema operativo creado por Google en el año 2008.

Anunciado el 16 de mayo del 2013, reemplazó a Eclipse como IDE oficial de desarrollo de aplicativos Android. Pese a tener similitudes a Eclipse, debido a que su programación es basada en Java, Android Studio ofrece renderización en tiempo real de interfaces, consola de desarrollador con estadísticas de uso, refactorización específica de Android, plantillas para el desarrollo rápido y soporte para la implementación de aplicativos en diferentes versiones del sistema operativo Android.

Esta herramienta será usada para la elaboración del aplicativo móvil el cual interactuara e identificara al usuario mediante una interfaz que usara el chip NFC del dispositivo para transmitir los datos necesarios para el registro del servicio a implementar.

2.3.4. Funcionalidad NFC en SDK's de Android

La versión mínima de SDK que una aplicación Android puede soportar funciones NFC es la API de nivel 9, que soporta envío a etiquetas mediante el método "ACTION_TAG_DISCOVERED", y únicamente provee acceso a mensajes NDEF mediante el extra "EXTRA_NDEF_MESSAGES". Ninguna otra propiedad u operación es accesible en este nivel.

A partir del API de nivel 10 se agregó el soporte para lectura y escritura, así como el empuje de los mensajes NDEF en primer plano.

Desde el API de nivel 14 se provee una forma más sencilla de empujar los mensajes NDEF hacia otros dispositivos con Android Beam y métodos convenientes para la creación de archivos NDEF.

En el archivo AndroidManifest.xml se puede limitar el SDK mínimo y máximo con el que el aplicativo desarrollado se puede ejecutar. Limitando así en base a las propiedades de cada versión de SDK según las funciones que se ejecutan en un aplicativo.

```
<uses-sdk android:minSdkVersion="14"/>
```

Para acceder a los elementos de hardware de NFC y manejar los Intents de NFC se deben declarar estos ítems en el archivo "AndroidManifest.xml" del aplicativo:

```
<uses-permission android:name="android.permission.NFC" />
```

En el caso de distribución del aplicativo, si se desea que la aplicación esté disponible únicamente para dispositivos que tienen el hardware de NFC se debe definir el "uses_feature" de la siguiente manera:

```
<uses-feature android:name="android.hardware.nfc" android:required="true" />
```

En este último caso se debe definir este parámetro si el aplicativo realiza procesos cruciales con la funcionalidad NFC, de no ser este el caso se debe omitir.

Al limitar el nivel de API en las aplicaciones de Android se corre el riesgo de limitar la cuota de mercado al que la aplicación puede llegar, por lo que es importante manejar la limitación del aplicativo únicamente si es crucialmente necesario.

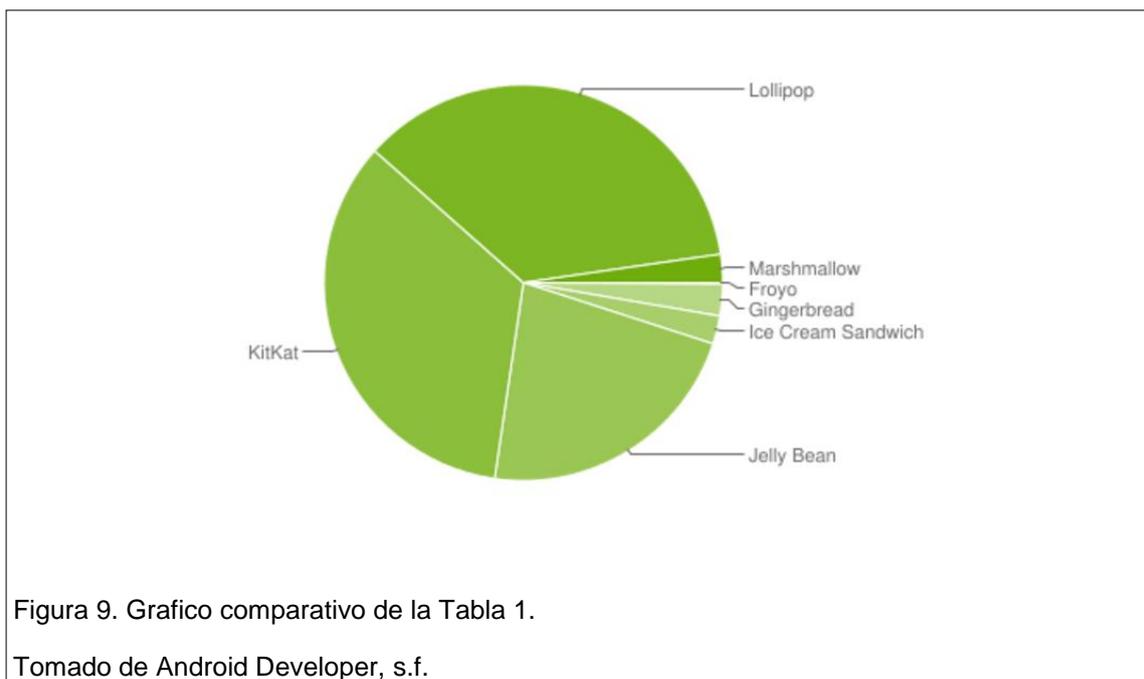
Como se observa en la tabla 1, el aplicativo a desarrollar no se verá limitado en cuota de mercado al usar un API mínimo de nivel 14 debido a que se estaría apuntando a un 97% de todos los dispositivos Android actualmente.

Tabla 1. Porcentaje de versiones de distribución de SO Android Marzo 2016

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.3%
4.1.x	Jelly Bean	16	8.1%
4.2.x		17	11.0%
4.3		18	3.2%
4.4	KitKat	19	34.3%
5.0	Lollipop	21	16.9%
5.1		22	19.2%
6.0	Marshmallow	23	2.3%

Data collected during a 7-day period ending on March 7, 2016.

Tomado de Android Developer, s.f.



Como podemos observar el grafico de la figura 9, el porcentaje que representa las versiones Froyo y Gingerbread del sistema operativo Android que no serían compatibles con el sistema propuesto, es inferior al 5% por lo que no representa una limitante.

2.3.5. Características de los mensajes NFC - Clases

NDEF (NFC Data Exchange Format)

El estándar internacional ISO/IEC 18092, Near Field Communication – Interfaces y Protocolos (NFCIP-1), define una interfaz y protocolo simple para conexiones inalámbricas de 2 dispositivos cercanos operando en una frecuencia de 13.56 MHz.

El NFC Data Exchange Format (NDEF) o NFC Formato de Intercambio de Datos, define el formato del intercambio de los mensajes encapsulados, sean estos dispositivos con chips NFC o etiquetas NFC. La clase definida tiene el nombre de “NdefMessage”.

NDEF es un formato binario de datos ligeros que puede ser usado para el encapsulamiento de uno o más aplicaciones para la definición del tipo y tamaño del mensaje, además de un identificador opcional.

Estas especificaciones definen únicamente la estructura del formato para el intercambio de datos entre servicios o aplicaciones. Esto no incluye la lectura o escritura de ningún tipo de los datos en detalle. Tampoco se especifica el intercambio de datos entre los 2 dispositivos, únicamente se motiva la revisión del protocolo NFCIP-1. Un ejemplo claro de esto es cuando dos dispositivos con NFC están próximos uno del otro y un mensaje NDEF es intercambiado mediante cualquier protocolo establecido, de igual forma si un mensaje es leído de una etiqueta NFC, en ambos casos independientemente del modelo del dispositivo o la etiqueta, el mensaje NDEF es procesado.

Debido al gran número de formatos de encapsulamiento, protocolos de grabado y protocolos de multiplexado conviene más hablar sobre los objetivos de diseño:

- Encapsulamiento arbitrario de documentos y entidades, incluyendo datos encriptados, documentos XML, fragmentos XML, datos de imágenes, etc.
- Encapsulamiento de documentos y entidades inicialmente de tamaños desconocidos. Esta capacidad puede ser usada para encapsular dinámicamente contenido generado en entidades muy grandes o series de pedazos de archivos.
- Agregación de múltiples documentos que son asociados lógicamente en un solo mensaje.
- Encapsulación compacta de pequeñas cargas útiles, sin la necesidad de introducir complejidad a los analizadores.

Para lograr la eficiencia y simplicidad, los mecanismos provistos en estas especificaciones han sido deliberadamente limitados a servir estos propósitos. NDEF no ha sido diseñado como una descripción general de un mensaje o formato de documento como lo es en cambio MIME o XML por ejemplo. Pero las aplicaciones NFC pueden aprovecharse de esto y tomar ventaja de dichos formatos encapsulándolos en mensajes NDEF.

NFC RTD (NFC Record Type Definition)

La clase definida en NFC es conocida como “NdefRecord”. Especifica el formato y las reglas para construir estándares de tipos de grabado usados por aplicaciones NFC que son basadas en los formatos de datos NDEF. Las especificaciones RTD proveen una vía eficiente para definir formatos de grabado para nuevas aplicaciones y da a los usuarios oportunidades de crear sus propias aplicaciones basadas en especificaciones NFC.

- Text RTD

Provee de una manera eficiente un método para albergar cadenas de texto en múltiples lenguajes usando mecanismos RTD y con formato NDEF. Un ejemplo de esto son las especificaciones usadas en los Smart Poster RTD.

- URI RTD

Provee de una manera eficiente un método para albergar recursos uniformes de identificadores o Uniform Resource Identifiers (URI), al usar mecanismos RTD con formatos NDEF.

- Smart Poster RTD

Define el cómo poner URL, SMS o números de teléfonos en una etiqueta NFC o como transportarlos entre dispositivos. Un Smart Poster usa los URI RTD y los Text RTD como bloques de construcción.

Signature RTD

Especifica el formato usado cuando se inscriben uno o varios registros NDEF. Define los requerimientos y firmas opcionales de los campos RTD, también provee una lista de algoritmos y tipos de certificados que pueden ser usados para crear firmas.

No define un sistema de certificación o define un nuevo algoritmo para el uso de firmas “RTD (Signatures RTD)”.

Política de Certificación RTD

Define los requerimientos del proceso y operación que el NFC espera adherir de una autoridad de certificación o Certificate Authorities (CAs), mientras se emiten y manejan certificados para crear firmas para los mensajes NDEF.

NfcAdapter

Incluido desde el API de nivel 9, es una clase que ayuda a recuperar el adaptador por defecto en un dispositivo Android con el método “getDefaultAdapter(Context)”. Este objeto posee diversas clases anidadas, así como constantes y métodos públicos para identificar las capacidades del dispositivo referente al chip NFC que posee integrado en la tarjeta. También, ayuda a identificar las propiedades por defecto de los dispositivos que se encuentran en el rango de funcionamiento, con el fin de interactuar en los procesos cuando 2 dispositivos están próximos el uno del otro.

NfcEvent

Incluido desde el API de nivel 14, es una clase que envuelve información asociada a cualquier evento NFC. Un objeto NfcEvent es comúnmente incluido en las llamadas desde NfcAdapter, verifica la documentación de cada llamada y revisa que ambiente debería ser establecido.

Este objeto es usado en vez de parametrizar la llamada de vuelta, porque permite agregar nuevos ambientes sin romper la compatibilidad de los API.

NfcManager

Incluido desde el API de nivel 10, es una clase que mediante la ayuda de algunos métodos es usado para obtener una instancia del NfcAdapter.

Tag

También incluido desde el API de nivel 10, es un objeto que representa de manera inmutable al estado de una etiqueta NFC desde el momento en que es descubierto. Puede ser usado para manejar clases de “Tagtechnology” para ejecutar operaciones avanzadas.

Un objeto Tag es creado cada vez que una etiqueta es descubierta (se aproxima al rango de funcionamiento NFC), incluso cuando es la misma etiqueta física. Si una etiqueta es retirada del rango y luego colocada nuevamente, entonces el último objeto Tag creado exitosamente es usado.

2.3.6. Excepciones

Excepciones de Formato – FormatExceptions

Clase publica derivada del extends Exception. Utiliza los constructores públicos “FormatException()”, “FormatException(String message)” y “FormatException(String message, Throwable e)”. Todos incluidos desde el API de nivel 9 a excepción del tercero que fue incluido en el API de nivel 16. Es usado al igual que las excepciones capturadas en un bloque Try, pero definidas para específicamente para los mensajes NDEF.

Excepciones de Perdida de Etiqueta - TagLostException

Se deriva del extends “java.io.IOException” que se deriva a su vez del extends Exception. Utiliza los constructores públicos “TagLosException()” y “TagLostException(String message)”, ambos incluidos en el API de nivel 10. Es usado para capturar las excepciones en un bloque Try cuando una etiqueta es retirada o el dispositivo no logra identificarlo a tiempo.

2.4. Aplicaciones Web

Una aplicación web son todas las herramientas a las que un usuario puede acceder a través de un navegador, ya sea conectado a internet o a una intranet.

La popularidad de las aplicaciones web radica en su capacidad de ser ejecutada desde cualquier navegador de internet, independizando la instalación y actualizaciones locales en los equipos de usuarios potenciales y la versión de sistema operativo desde donde son ejecutadas. Entre los tipos de aplicaciones

más populares se encuentran los web mails, páginas wiki, tiendas en línea y videojuegos entre otras.

Las aplicaciones web utilizan una serie de páginas con un formato estándar como el HTML o XHTML y son soportados generalmente por cualquier navegador de internet. En el caso de ejecutar funciones especiales se utilizan “plugins” que funcionan como lenguajes interpretados del lado del usuario, un ejemplo de ello es el JavaScript, Flash, entre otros.

Su presentación es interpretada por el navegador de internet y se muestra al usuario de una manera dinámica, con la capacidad de interactuar con la información mostrada ya sea en forma de juegos, formularios o gestores de base de datos.

Su uso representa ventajas como la de optimización del tiempo al realizar tareas simples, no tienen problemas de compatibilidad, ahorran espacio en las máquinas de los usuarios, las actualizaciones se manejan del lado del servidor, consumen bajos recursos, su disponibilidad es alta y los navegadores cada vez ofrecen mayor tipo de funciones.

Como desventajas se deben considerar que tienen limitadas funciones en comparación con aplicaciones de escritorio y su disponibilidad depende del proveedor de conexión a internet.

2.5. Capas de una Aplicación Web

2.5.1. Capa de Presentación

La capa de presentación es todo lo que ve el usuario, toda la información que es transmitida a él y los datos con los que interactúa. También conocida como la interfaz gráfica, en las aplicaciones web es el código interpretado por el navegador de internet. Aquí se descifra el código en forma de páginas web y es donde se encuentra presente al usuario. Se captura la información provista o modificada por el usuario y se filtra previamente para evitar errores.

Su principal característica es la de ser entendible y fácil de usar, debe ser intuitiva y cada parte de ella debe tener un propósito para el usuario.

En aplicaciones web, la capa de presentación podría estar conformada por un sin número de dispositivos que acceden a la capa de negocio y ejecutan el sistema. Estos dispositivos pueden ser de cualquier tipo (Pc, laptops, tablets, teléfonos celulares, etc.) y su única característica en común es la de poseer acceso al servidor de la capa de negocio. Esta capacidad está delimitada por el proveedor de internet o el enlace y por la versión de navegador web que posee cada dispositivo.

Los ejemplos más comunes de aplicativos web que tienen este amplio margen de disponibilidad para dispositivos son las redes sociales. En ellas se entregan los mismos servicios independientemente del tipo de dispositivo desde el cual se quiere acceder.

2.5.2. Capa de Negocio

En esta capa residen las aplicaciones y la lógica del negocio, puede estar representada por un servidor en donde se encuentren una o varias aplicaciones funcionando en conjunto.

Es en este nivel donde se definen las reglas que deben cumplirse en el sistema, determinando valores de entrada, salida, formatos y funciones. Dependiendo de la complejidad de la programación, el aplicativo podrá estar ubicado en un solo computador o en varios, a esta estructura se la determina por el término "nivel".

2.5.3. Capa de Datos

En la capa de datos se representa enteramente a la base de datos y los gestores de la misma. Se procesan solicitudes de lectura y escritura desde la capa de negocio, de igual forma dependiendo de la complejidad del sistema la capa de datos podrá ser conformada por uno o varios servidores.

En los tipos de aplicaciones cliente/servidor es muy común encontrar que en la capa de presentación se encuentra el mayor número de equipos, debido al tipo de conexión que realiza. Por lo que la capa de negocio y de datos puede residir en un mismo equipo/servidor, y dependiendo de la complejidad del sistema este puede extenderse a más equipos como en el caso de servidores de respaldo situados geográficamente separados de los originales.

2.6. Servicios Web

Los servicios Web son aplicaciones que se comunican a través del protocolo HTTP definido en el World Wide Web Consortium (W3C). Y se encargan de proveer un nexo entre aplicaciones y dispositivos que funcionen a través de un Framework establecido.

Gracias a los servicios web, aplicativos simples pueden ofrecer servicios de valor añadido, interactuando con sistemas similares o de una lógica de negocio diferente pero que trabajen con los mismos tipos de datos.

2.6.1. Estándares Relacionados

En un proceso de ejecución de un servicio Web intervienen una serie de tecnologías que hacen posible la circulación de la información. A continuación se lista una serie de estándares empleados en los servicios Web:

- SOAP (Simple Object Access Protocol). – Se encarga de establecer el protocolo del intercambio de información.
- XML (Extensible Markup Language).- Formato estándar para los datos que se vayan a intercambiar.
- Web Services Protocol Stack.- También conocido como pila de protocolos para servicios Web. Es como se lo denomina al conjunto de servicios y protocolos en los servicios Web.
- HTTP, FTP y SMTP.- Son protocolos de envío normal de archivos y también son usados en estos servicios.

- WSDL (Web Service Description Language). - Es el lenguaje usado por la interfaz. Está basada en XML y describe los requisitos funcionales para establecer una comunicación con los servicios Web.
- UDDI (Universal Description, Discovery and Integration). - Es un protocolo que publica la información de los servicios Web. Ayuda a identificar qué servicios están disponibles.
- WS-Security (Web Service Security). - Es el protocolo de seguridad adoptado por la Organización para el Avance de Estándares de Información Estructurada o por sus siglas en inglés OASIS.
- REST (Representation State Transfer).- Realiza operaciones entre las aplicaciones del servicio web y el cliente a través del protocolo HTTP.

2.6.2. Ventajas y Desventajas del uso de Servicios Web

Entre las ventajas del uso de los servicios Web tenemos el aporte de la interoperabilidad entre aplicaciones. La comunicación entre sistemas independientemente de donde estén instalados puede transformar aplicaciones simples en sistemas muy útiles.

Gracias a los servicios Web, cualquier aplicación puede interactuar con otra sin importar su ubicación geográfica. También su uso no depende de una sola compañía ni el tipo de software integrado.

En cambio, del lado de las desventajas tenemos que su funcionamiento con HTTP podría representar un salto a las seguridades basadas en firewall. Pese a que la utilización de HTTP sobre TCP es uno de sus principales atractivos, no es el único protocolo sobre el que se pueden utilizar.

Su rendimiento es bajo debido a su adopción del formato basado en texto y tampoco se tiene especificado un modelo en la eficacia del procesamiento.

2.6.3. Plataformas

Entre los diferentes servidores de aplicaciones para servicios Web tenemos:

- IBM Lotus Domino
- Axis
- JBoss
- Oracle Fusion Middleware
- WebLogic
- Websphere
- JAX-WS
- ColdFusion
- Java Web Service Development Pack o JWSDP
- Microsoft .NET
- Novell extend
- Zope
- VERASTREAM

2.6.4. Cloud Computing o Computación en la Nube

La computación en la nube o servicios en la nube permite la oferta de servicios de computación a través del internet. Permite el almacenamiento permanente de información en servidores y es enviado temporalmente al cliente, según sus requerimientos.

Estos servicios proveen al usuario de seguridad en la información, mientras se opta por una optimización de recursos en los proyectos al no tener que invertir en la compra de servidores, componentes de red, hardware, mantenimiento, etc.

Es definido por las siguientes características:

- Reducción de costos globales en los proyectos.
- Seguridad en el almacenamiento y gestión de la información.
- No requiere mantenimiento.
- Consume de manera eficiente la energía, al ser accedidos únicamente cuando lo dispone el cliente.

Cloud IaaS: Infrastructure as a Service

IaaS o Infraestructura como Servicio es un modelo de cómputo a través de la nube, el cual se encarga de entregar recursos de cómputo en un ambiente virtualizado por medio de conexión a la red o enlaces directos al servicio. En el caso de IaaS el servicio entrega hardware virtualizado, almacenamiento básico, servidores, enrutadores o en otras palabras infraestructura de cómputo.

Cloud PaaS: Platform as a Service

PaaS o Plataforma como Servicio, se encarga del encapsulamiento de componentes de la lógica de un sistema. En este caso, el servicio se encargaría de contener módulos, APIs y complementos de un sistema configurado y listo para integrarse a un entorno de desarrollo.

Este servicio ofrece soporte a todas las fases del ciclo de desarrollo y pruebas de software, o puede estar dirigida a un área en particular.

Cloud SaaS: Software as a Service

El Software como Servicio se caracteriza por entregar una aplicación completa ofrecida como servicio. Es una sola instancia del software que es ejecutada a nivel de proveedor y es usada para varias instancias de clientes.

Generalmente es accedido a través de navegadores de internet y es el servicio que mejor se adapta a las necesidades de las aplicaciones web, permitiendo a los usuarios configuraciones básicas de los datos pero no del aplicativo.

Su principal ventaja es la de eliminar cualquier tipo de instalación del lado del usuario, simplificando su acceso y mejorando completamente la disponibilidad del aplicativo.

2.7. Metodología

Se hace uso de UML para el modelamiento y documentación de la construcción del sistema propuesto. Esta metodología será útil para la construcción de diagramas de secuencia y diagramas de clase que se complementara con la metodología OOHDM.

En el desarrollo del software se ha definido el uso de la metodología de Diseño de Hipermedia Orientado a Objetos u OOHDM por sus siglas en ingles. Esta consideración está basada en el enfoque de esta metodología a la programación en torno a la interfaz del software, característica que beneficia a la programación de aplicativos móviles, los cuales requieren una mayor atención a la interacción de la interfaz con el usuario.

Esta metodología permitirá la estructuración del sistema propuesto basado en el patrón de arquitectura de software Modelo Vista Controlador (MVC). El cual ayudara a definir las capas del sistema a implementar.

2.7.1. Fases de la Metodología OOHDM

Para la aplicación de esta metodología se consideran las siguientes fases:

- Análisis de Requerimientos.
- Diseño Conceptual.
- Diseño de Navegación.
- Diseño de Interfaces.
- Implementación.

2.7.2. Especificación de Requerimientos según el estándar IEEE 830

La recopilación de requisitos será obtenido en base al funcionamiento actual del sistema de transporte público en la ciudad de Quito (Figura 6.) y serán organizados en el presente documento según el estándar IEEE 830, el cual define requisitos para la obtención de dichos requerimientos.

3. DISEÑO

3.1. Análisis de Requerimientos

A continuación se analizan las funciones del sistema a implementar propuesto en el presente trabajo. Se detalla el comportamiento requerido, y son definidos por el comportamiento del sistema actual del transporte público en la ciudad de Quito.

Dichos comportamientos serán identificados con el fin de ser innovados en el tiempo de ejecución, recursos a utilizarse y actores involucrados. Además, se agregan funciones proporcionadas por las herramientas a utilizarse.

Como introducción al análisis de requerimientos debemos tomar en cuenta que el proyecto es basado en un sistema existente, por lo que se toma como referencia las entradas y salidas del sistema actual.

En dicho sistema y como se puede observar en la Figura 5, se identifican la participación de 3 actores: El Usuario, el Controlador y el Conductor. En el sistema propuesto en este trabajo, el actor Controlador es reemplazado por el prototipo y el aplicativo en Android, por lo que se identifican los actores: GUI Android y Prototipo.

Adicional, se identifica un cuarto actor denominado GUI Web, en el que recaen las funciones de despliegue de la información de la base de datos.

3.1.1. Requerimientos

Requerimientos Funcionales

Se especifican las capacidades del sistema para satisfacer las necesidades del negocio. Se determina como punto de partida el tipo de negocio actual definido por el servicio de transporte público en la ciudad de Quito.

Las funciones aquí definidas serán analizadas en la definición de los roles y tareas para clasificarlos desde el diseño en los diferentes softwares que conforman el sistema.

Para que el sistema propuesto pueda identificar a los usuarios, este debe tener la capacidad de registrarlos desde el aplicativo móvil (Tabla 2.), este requerimiento tiene una importancia alta debido a que esta propiedad del sistema es el que permitirá a los usuarios registrarse y posteriormente identificarse. El sistema debe tener la capacidad de diferenciar a los usuarios, dependiendo si estos son adultos, tercera edad, menores de edad o poseen una discapacidad. Esta clasificación es necesaria debido a que en el sistema actual del servicio de transporte, el valor de pasaje es reducido en un 50% considerando esta clasificación (Tabla 3.).

Tabla 2. Requerimiento Funcional RFQ-001: Registro de Usuarios

RFQ-001	Registro de Usuarios
DESCRIPCIÓN	El aplicativo Android permitirá a los pasajeros registrarse al servicio usando su documento de identificación (Cédula de Identidad o Pasaporte).
OBJETIVO	Objetivo Especifico #3
IMPORTANCIA	ALTA
ESTADO	DISPONIBLE
ESTABILIDAD	ALTA
COMENTARIOS	El registro de usuarios será posible de realizarlo desde el aplicativo web.

La estabilidad de este requerimiento es medio, debido a que el servicio podría proponer cambios tanto en la clasificación de los usuarios ya sea agregando o quitando grupos, o el valor del descuento al que pueden acceder.

Tabla 3. Requerimiento Funcional RFQ-002: Clasificación de Usuarios

RFQ-002	Clasificación de Usuarios
DESCRIPCIÓN	El sistema registrara a los usuarios clasificándolos en los grupos: Adultos, Tercera Edad, Menor de Edad y Con Discapacidad.
OBJETIVO	Objetivo Especifico #1
IMPORTANCIA	ALTA
ESTADO	DISPONIBLE
ESTABILIDAD	MEDIA
COMENTARIOS	

El sistema propuesto debe tener la capacidad de registrar a los usuarios que hacen uso de él. En la Tabla 4 y Tabla 5 se registran los requerimientos relacionados a esta propiedad del funcionamiento del servicio. Dichos datos almacenados es la información que conforma el valor agregado al servicio actual.

Hoy en día, el funcionamiento del servicio de transporte no ofrece identificación de los usuarios ni del chofer en cada unidad de transporte, por lo que en situaciones de accidentes, estos no pueden ser responsables de los hechos sucedidos. También, cada usuario que ingresa es registrado, por lo que representaría una ventaja para evitar los asaltos dentro de las unidades.

Tabla 4. Requerimiento Funcional RFQ-003: Registro de Uso

RFQ-003	Registro de Uso
DESCRIPCIÓN	El sistema registrara la cantidad de veces en que el usuario utilizo los medios de transporte público ligados al sistema.
OBJETIVO	Objetivo Especifico #1
IMPORTANCIA	ALTA
ESTADO	DISPONIBLE
ESTABILIDAD	ALTA
COMENTARIOS	Se registrara la información en una tabla "Ticket".

Tabla 5. Requerimiento Funcional RFQ-004: Registro de Datos

RFQ-004	Registro de Datos
DESCRIPCIÓN	El sistema registrara los datos del usuario, la unidad de transporte, el conductor, la ruta elegida y la hora en que se realizó el acceso.
OBJETIVO	Objetivo Especifico #4
IMPORTANCIA	ALTA
ESTADO	DISPONIBLE
ESTABILIDAD	ALTA
COMENTARIOS	

Con relación a la posibilidad de visualizar dicha información en el sistema, este debe ser capaz de desplegar la información amigablemente y dispuesta para la facturación del servicio, esto con el sentido de vincular los gastos realizados por el usuario y poder disponer de una factura del servicio. Esto más adelante podría ser considerado para la deducción de impuestos con el motivo de promocionar el servicio inicialmente. Estos requerimientos serán registrados en las Tablas 6 y Tabla 7 tendrán una importancia Alta debido a que de esto depende el valor agregado del servicio, además de una estabilidad Media, debido a que este es sometido a las decisiones de funcionamiento que las rige el municipio y son aceptadas por las empresas proveedores del servicio de transporte.

Tabla 6. Requerimiento Funcional RFQ-005: Facturación

RFQ-005	Facturación
DESCRIPCIÓN	El sistema emitirá una factura a final de mes similar a la proporcionada por las entidades gubernamentales que proveen los servicios básicos.
OBJETIVO	Objetivo Especifico #4
IMPORTANCIA	ALTA
ESTADO	DISPONIBLE
ESTABILIDAD	MEDIA
COMENTARIOS	Los datos de la factura se podrán visualizar a través del aplicativo Web.

Tabla 7. Requerimiento Funcional RFQ-006: Análisis de Datos

RFQ-006	Análisis de Datos
DESCRIPCIÓN	El usuario será capaz de acceder a los datos generados por el sistema a través de un aplicativo web.
OBJETIVO	Objetivo Especifico #4
IMPORTANCIA	ALTA
ESTADO	DISPONIBLE
ESTABILIDAD	MEDIA
COMENTARIOS	El usuario podrá acceder a los datos desde el aplicativo web por medio de su usuario y contraseña.

Requerimientos no Funcionales

Las definiciones aquí planteadas servirán como ambiente para el diseño de las funciones del negocio.

Rendimiento

- El acceso a la red de los dispositivos que se conecten por medio de la interfaz web o el aplicativo móvil, tendrá que ser mayor o igual a 200Kb/s.
- El acceso a la base de datos tendrá que ser requerido únicamente cuando se realicen actualizaciones o consultas, el resto del tiempo deberá permanecerá desconectado.

Disponibilidad

- La disponibilidad del aplicativo estará vinculada con la disponibilidad del servidor de la base de datos.
- El sistema estará disponible en los horarios de atención del actual sistema de transporte público.

Accesibilidad

- El usuario deberá habilitar las funciones NFC en su celular.
- Para realizar el registro por medio del aplicativo Android el teléfono celular deberá estar conectado a internet.

Operacionales

- Las interfaces contarán con obligatoriedad de campos.
- Las interfaces restringirán el ingreso de tipos inválidos en los diferentes campos.
- El sistema verificará usuarios y contraseñas para el inicio de sesión.

Hardware

- El usuario podrá ejecutar el aplicativo en su celular si cuenta con un sistema operativo Android y su versión mínima funcional será la versión 4.0 Android Ice Cream Sandwich.
- El usuario deberá poseer un dispositivo celular con un chip NFC integrado, compatible con los estándares establecidos en el NFC Forum.
- Para acceder al aplicativo web se necesita tener instalado cualquier navegador de internet (Google Chrome, Firefox, Internet Explorer).

3.1.2. Identificación de Actores y Tareas

Actores:

- Usuario

Se identifica al usuario como el actor que solicita el servicio de transporte. Este actor tiene acceso al aplicativo Android y a la base de datos únicamente para visualizar los datos referentes a su cuenta.

- Conductor

El Conductor es el actor que se encarga de proveer el servicio de transporte al usuario final y en el sistema propuesto también se identificará con sus credenciales.

- GUI

La GUI es el actor que despliega la información recopilada por el sistema hacia el usuario y al conductor mediante una interfaz amigable. Esta se divide en la GUI para el sistema Android para Usuarios y Conductores y GUI Web como se muestra en la Figura 10.

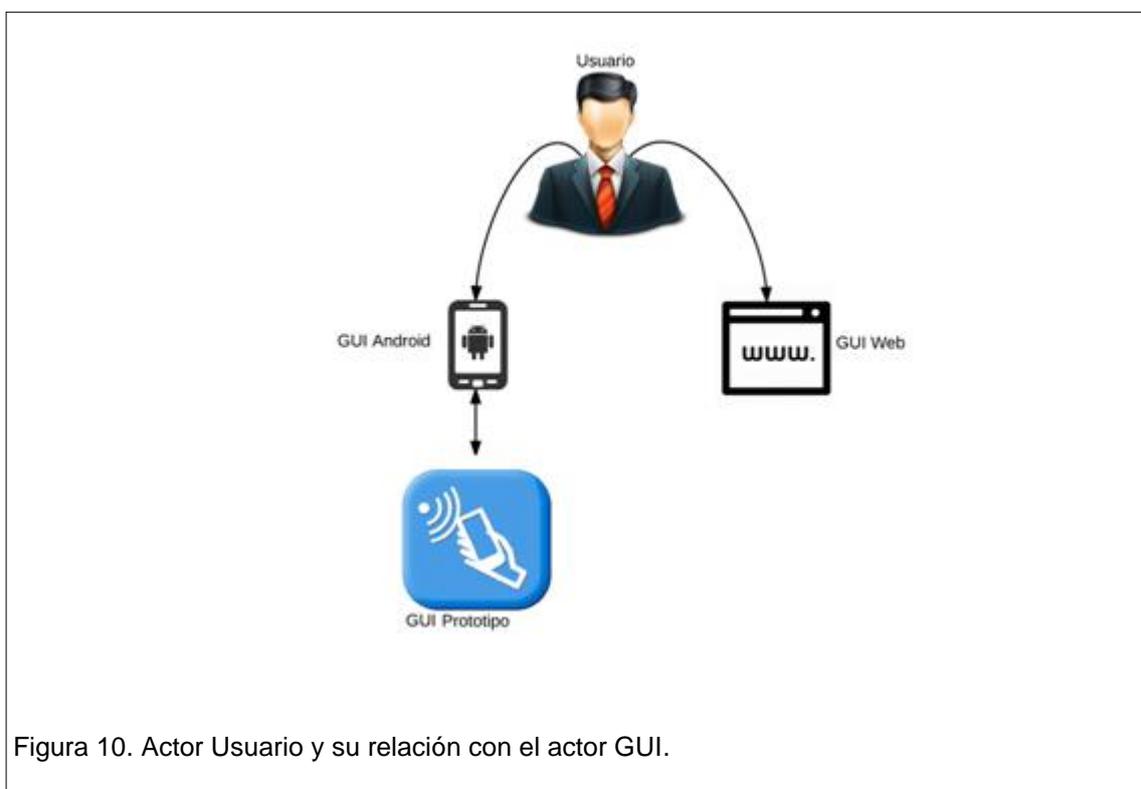


Figura 10. Actor Usuario y su relación con el actor GUI.

Tareas:

Usuario

- Registrarse con sus datos personales solicitados en la aplicación.
- Identificarse para iniciar una sesión en el aplicativo con sus datos.
- Seleccionar la opción “Generar Ticket” en la interfaz de Android.
- Acercar su teléfono móvil al prototipo para iniciar la secuencia de identificación y registrar el acceso. La distancia entre el teléfono celular del usuario y el prototipo deberá ser inferior a los 3cm.
- Ingresar al transporte público.

Conductor

- Registrarse con sus datos personales solicitados en la aplicación Prototipo.
- Activar la función “Modo lectura” en el aplicativo para que este pueda empezar a registrar el ingreso de los usuarios.
- Trasladar a los usuarios en las unidades de transporte, manteniendo las rutas establecidas.

GUI

- Conexión a la base de datos.
- Registro de usuario.
- Mostrar mensajes informativos al usuario referente al registro de datos.
- Mostrar mensajes informativos al usuario referente al inicio de sesión.
- Desplegar la opción de generar la solicitud de acceso al transporte público.
- Registrar ingreso de usuarios con la función “Modo Lectura”.
- Desplegar la información de la factura generada por el sistema, en base a los datos de determinado usuario, unidad de transporte y conductor del mismo.

3.1.3. Especificaciones de Escenarios

Rol Usuario

- Opción Registrar: El usuario ingresa en el aplicativo Android los datos CI o Pasaporte, el grupo al que pertenece (Adulto, Tercera Edad, Menor de Edad o Con Discapacidad), su nombre completo, su número de teléfono, su dirección y una contraseña para la cuenta que se le asignara.
- Opción Ingresar: El usuario ingresara su id, que será su Cédula de Identidad o su Pasaporte y su clave para iniciar sesión en el aplicativo.
- Opción Solicitud de Acceso: El usuario seleccionara la opción en el aplicativo “Solicitar Ticket” y acercara su dispositivo móvil al prototipo para completar la identificación.
- Opción Consulta: El usuario inicia sesión en el aplicativo web y realiza la consulta del estado de su cuenta.

Rol Conductor

- Opción Registrar (Conductor): El GUI Web registrara los datos del conductor.
- Opción Iniciar Sesión (Conductor): Cuando el Conductor ingrese su ID y su contraseña, se realizara una verificación de sus datos para verificar que tiene una cuenta creada.
- Opción “Modo Lectura”: Una vez registrado el conductor, el GUI prototipo podrá entrar en estado de lectura, esperando la detección de cualquier mensaje NDEF.

Rol GUI

- Opción Registrar: Una vez que el usuario ingresa los datos en la interfaz gráfica del aplicativo Android, este registra y crea una cuenta nueva en la base de datos.
- Opción Iniciar Sesión (Usuario): Cuando el usuario ingrese su ID y su contraseña, se realizara una verificación de sus datos para verificar que tiene una cuenta creada.
- Opción Iniciar Sesión (Conductor): Cuando el usuario ingrese su ID y su contraseña, se realizara una verificación de sus datos para verificar que tiene una cuenta creada.
- Opción Ticket: El usuario solicita el acceso al medio de transporte una vez que inicio sesión en la GUI Android.
- Opción Consulta: El usuario despliega la información sobre su cuenta y los datos recopilados.

3.1.4. Especificaciones de casos de uso por Actores

Actor: Usuario

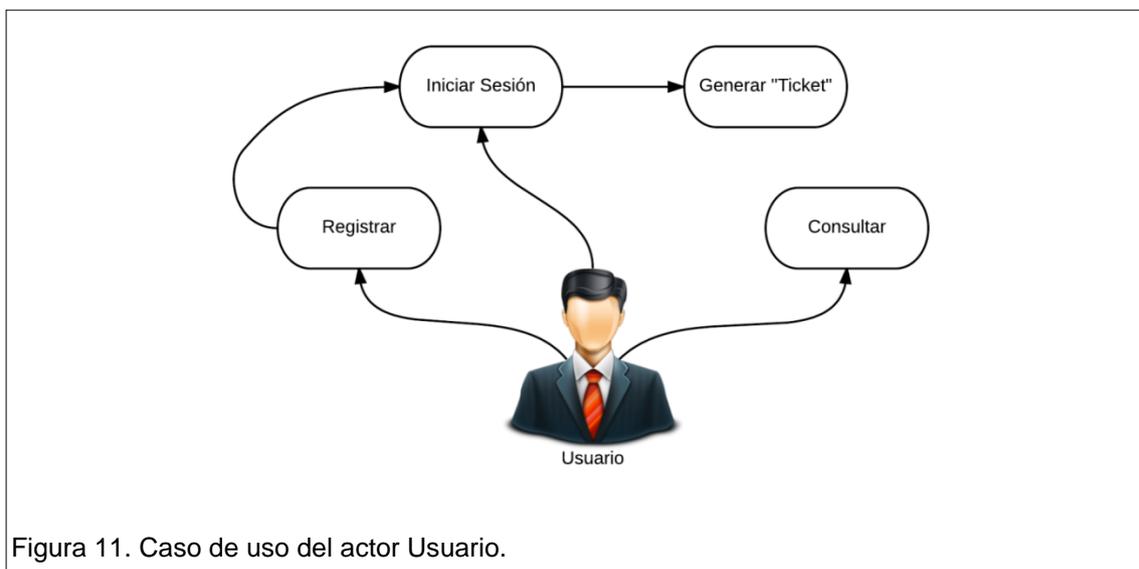


Figura 11. Caso de uso del actor Usuario.

La figura 11 muestra la relación del actor usuario con las funciones propuestas por el sistema. En base a este esquema se identificara las especificaciones de casos de uso, por actores.

Tabla 8. Registrar (GUI Android).

Nombre	Registrar(GUI Android)
Autor	Carlos Torres Vinueza
Fecha	13/09/2015
Descripción	El Usuario crea su cuenta en el sistema.
Actores	<ul style="list-style-type: none"> • Usuario • GUI Android
Precondiciones	<ul style="list-style-type: none"> • Descargar el aplicativo en el celular. • Seleccionar la opción "Registrar". • Ingreso de los siguientes datos en la interfaz: ID, tipo de ID, grupo, Nombre, Teléfono, Dirección, Contraseña.
Flujo normal	<ol style="list-style-type: none"> 1. Selección de la función "Registrar". 2. Ingreso de los datos solicitados. 3. Guardar información.
Flujo Alternativo	<ol style="list-style-type: none"> 1.1. Selección del botón Home o retroceso. 2.1. Ingreso parcial de los datos solicitados. 2.2. Ingreso incorrecto de los datos solicitados. 2.3. Ingreso de datos ya incluidos en la base.
Post condiciones	<ul style="list-style-type: none"> • Ingreso de los datos en la base. • Mensaje de confirmación.

El usuario tiene la capacidad de registrarse en el sistema desde el aplicativo web y el aplicativo móvil, por lo que no está ligado a ningún proceso externo para crear un usuario en el sistema.

Tabla 9. Iniciar Sesión (GUI Android).

Nombre	Iniciar Sesión(GUI Android)
Autor	Carlos Torres Vinueza
Fecha	13/09/2015
Descripción	El Usuario inicia sesión con su cuenta en el sistema.
Actores	<ul style="list-style-type: none"> • Usuario • GUI Android
Precondiciones	<ul style="list-style-type: none"> • Seleccionar la opción "Iniciar Sesión". • Ingreso de los siguientes datos en la interfaz: ID, Contraseña.
Flujo normal	<ol style="list-style-type: none"> 1. Selección de la función "Iniciar Sesión". 2. Ingreso de los datos solicitados. 3. Selección del botón "Ingresar". 4. Confirmación de inicio de sesión completo.
Flujo Alternativo	<ol style="list-style-type: none"> 1.2. Selección del botón Home o retroceso. 2.1. Ingreso parcial de los datos solicitados. 2.2. Ingreso incorrecto de los datos solicitados. 2.3. Ingreso de datos no registrados en la base.
Post condiciones	<ul style="list-style-type: none"> • Sesión activa en la interfaz Android. • Mensaje de confirmación.

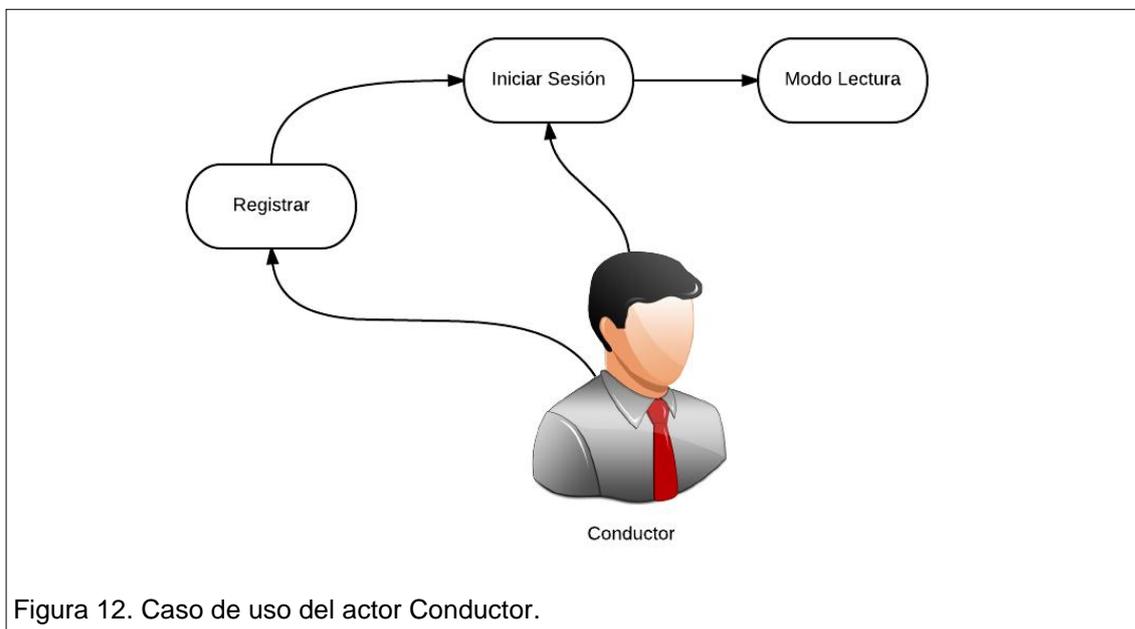
Tabla 10. Generar Ticket (GUI Android).

Nombre	Generar Ticket(GUI Android)
Autor	Carlos Torres Vinueza
Fecha	13/09/2015
Descripción	El Usuario selecciona la opción para generar un ticket de acceso a la unidad de transporte.
Actores	<ul style="list-style-type: none"> • Usuario • GUI Android
Precondiciones	<ul style="list-style-type: none"> • Iniciar sesión en el aplicativo. • Seleccionar la opción "Generar Ticket".
Flujo normal	<ol style="list-style-type: none"> 1. Selección de la función "Generar Ticket". 2. Selección del botón "Generar".
Flujo Alternativo	<ol style="list-style-type: none"> 1.1. Selección del botón Home o retroceso. 2.1. Ninguna selección.
Post condiciones	<ul style="list-style-type: none"> • Ticket virtual mostrado en pantalla.

Tabla 11. Consultar (GUI Web).

Nombre	Consultar(GUI Web)
Autor	Carlos Torres Vinueza
Fecha	15/09/2015
Descripción	El Usuario consulta los datos referentes a su cuenta en la interfaz Web.
Actores	<ul style="list-style-type: none"> • Usuario • GUI Web
Precondiciones	<ul style="list-style-type: none"> • Iniciar sesión en la Interfaz Web. • Seleccionar la opción "Consultar".
Flujo normal	<ol style="list-style-type: none"> 1. Selección de la función "Consultar". 2. Despliegue de la información solicitada.
Flujo Alternativo	<ol style="list-style-type: none"> 1.1. Sesión no iniciada. 2.1. Ningún dato disponible.
Post condiciones	<ul style="list-style-type: none"> • Información desplegada en pantalla.

Actor: Conductor



A diferencia del usuario, el conductor únicamente puede registrarse desde el aplicativo web y su interacción con el aplicativo móvil prototipo únicamente es el de inicio de sesión y activación del modo lectura, el cual una vez activado registra a cualquier usuario que acerque su dispositivo celular.

Tabla 12. Registrar (GUI WEB).

Nombre	Registrar(GUI WEB)
Autor	Carlos Torres Vinueza
Fecha	14/09/2015
Descripción	La interfaz crea una cuenta con los datos del conductor.
Actores	<ul style="list-style-type: none"> • Conductor • GUI
Precondiciones	<ul style="list-style-type: none"> • Seleccionar la opción "Registro". • Ingreso de los siguientes datos en la interfaz: ID, tipo de ID, grupo, Nombre, Teléfono, Dirección, Contraseña.
Flujo normal	<ol style="list-style-type: none"> 1. Selección de la función "Registrar". 2. Ingreso de los datos solicitados por parte del usuario. 3. Guardar información en la base de datos.
Flujo Alternativo	<ol style="list-style-type: none"> 1.1. Selección del botón Home o retroceso. 2.1. Ingreso parcial de los datos solicitados. 2.2. Ingreso incorrecto de los datos solicitados. 2.3. Ingreso de datos ya incluidos en la base.
Post condiciones	<ul style="list-style-type: none"> • Ingreso de los datos en la base. • Mensaje de confirmación.

Tabla 13. Iniciar Sesión (Android Prototipo).

Nombre	Iniciar Sesión(GUI Android Prototipo)
Autor	Carlos Torres Vinueza
Fecha	14/09/2015
Descripción	La interfaz inicia sesión con los datos del conductor.
Actores	<ul style="list-style-type: none"> • Conductor • GUI
Precondiciones	<ul style="list-style-type: none"> • Seleccionar la opción "Iniciar Sesión". • Ingreso de los siguientes datos en la interfaz: ID, Contraseña.
Flujo normal	<ol style="list-style-type: none"> 1. Selección de la función "Iniciar Sesión". 2. Ingreso de los datos solicitados por parte del usuario. 3. Guardar información en la base de datos.
Flujo Alternativo	<ol style="list-style-type: none"> 1.1. Selección del botón Home o retroceso. 2.1. Ingreso parcial de los datos solicitados. 2.2. Ingreso incorrecto de los datos solicitados. 2.3. Ingreso de datos ya incluidos en la base.
Post condiciones	<ul style="list-style-type: none"> • Sesión activa en la interfaz Android. • Mensaje de confirmación.

Tabla 14. Modo Lectura (GUI Android Prototipo).

Nombre	Modo Lectura (GUI Android Prototipo)
Autor	Carlos Torres Vinueza
Fecha	14/09/2015
Descripción	El prototipo entra en modo de lectura, esperando que cualquier usuario acerque su teléfono celular solicitando acceso.
Actores	<ul style="list-style-type: none"> • Conductor • GUI
Precondiciones	<ul style="list-style-type: none"> • Aplicación en el prototipo ejecutada. • Identificación de los datos del conductor en el aplicativo del prototipo.
Flujo normal	<ol style="list-style-type: none"> 1. Selección de la función "Leer". 2. El prototipo entra en modo "Leer".
Flujo Alternativo	1.1. Selección del botón Home o retroceso.
Post condiciones	<ul style="list-style-type: none"> • Pantalla de modo Leer activada. • Disponible para leer mensaje NDEF.

Actor: GUI

Aplicativo Android para Usuarios

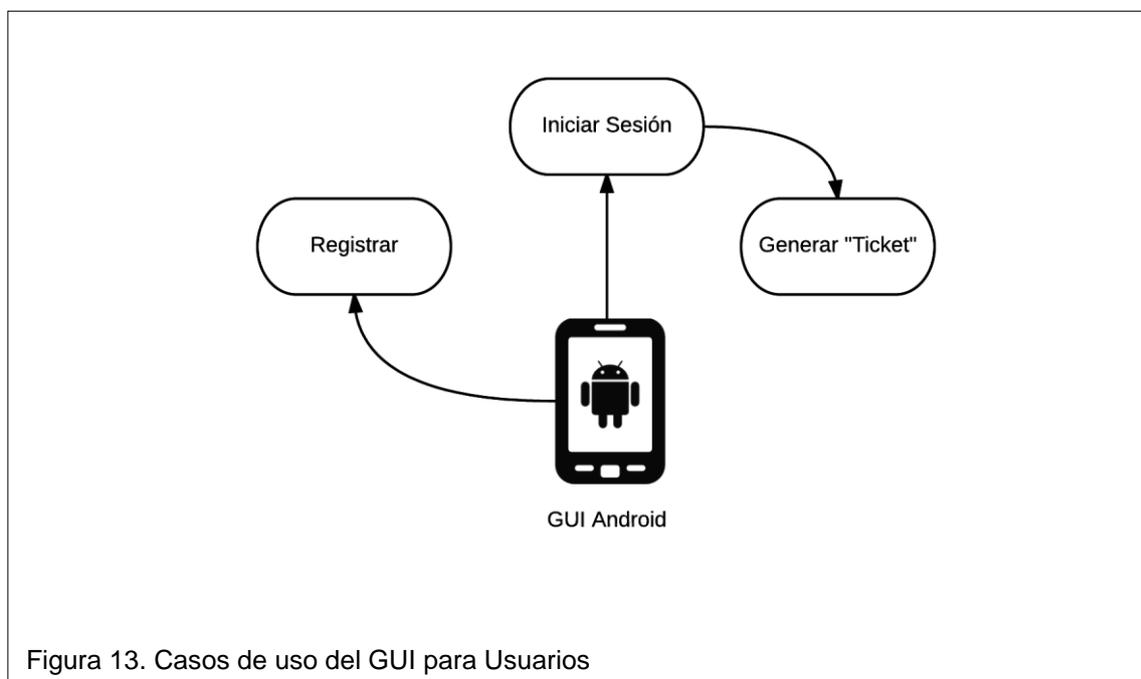
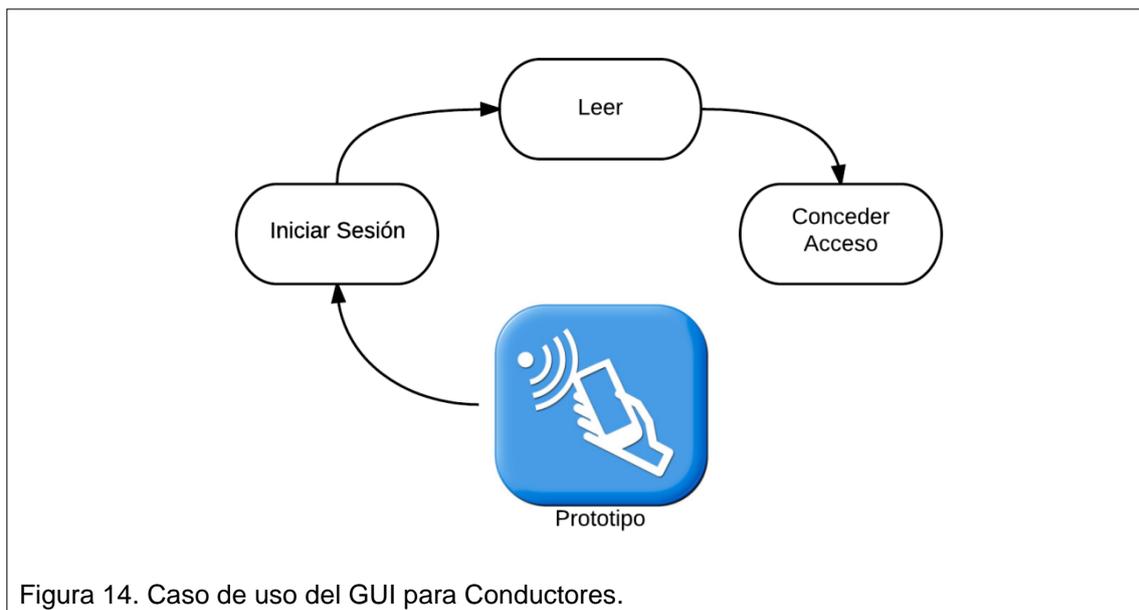


Figura 13. Casos de uso del GUI para Usuarios

Aplicativo Android para conductores (Prototipo)



La GUI para usuarios y conductores comparten similitudes para el inicio de sesión (Figura 13 y 14). Tanto usuarios como conductores deben iniciar sesión en el sistema. El usuario inicia la opción generar ticket, el cual empieza a transmitir la información por NFC y el conductor activa la opción “Modo Leer” el cual recepta dicha información y verifica el acceso del usuario.

Tabla 15. Registrar (Prototipo).

Nombre	Registrar (Prototipo)
Autor	Carlos Torres Vinueza
Fecha	15/09/2015
Descripción	El prototipo registra y vincula las funciones a los datos del conductor.
Actores	<ul style="list-style-type: none"> • Prototipo
Precondiciones	<ul style="list-style-type: none"> • Aplicación en el prototipo ejecutada.
Flujo normal	<ol style="list-style-type: none"> 1. Selección de la función “Registrar”. 2. Ingreso de los datos del conductor. 3. Habilidad de la función Leer.
Flujo Alternativo	<ol style="list-style-type: none"> 1.1. Selección del botón Home o retroceso.
Post condiciones	<ul style="list-style-type: none"> • Vinculación de los datos del conductor al prototipo. • Mensaje de confirmación. • Opción Leer habilitada.

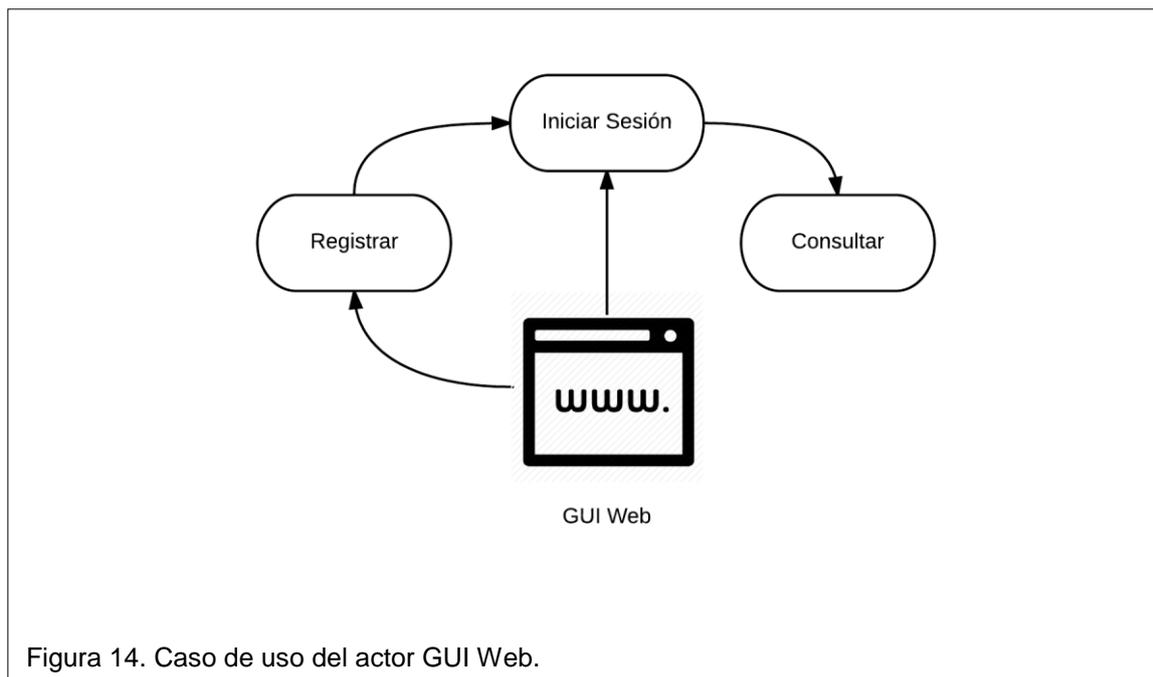
Tabla 16. Leer.

Nombre	Leer
Autor	Carlos Torres Vinueza
Fecha	15/09/2015
Descripción	El prototipo entra en modo de lectura, esperando que cualquier usuario acerque su teléfono celular solicitando acceso.
Actores	<ul style="list-style-type: none"> • Prototipo
Precondiciones	<ul style="list-style-type: none"> • Aplicación en el prototipo ejecutada. • Identificación de los datos del conductor en el aplicativo del prototipo.
Flujo normal	<ol style="list-style-type: none"> 3. Selección de la función "Leer". 4. El prototipo entra en modo "Leer".
Flujo Alternativo	<ol style="list-style-type: none"> 1.2. Selección del botón Home o retroceso.
Post condiciones	<ul style="list-style-type: none"> • Pantalla de modo Leer activada. • Disponible para leer mensaje NDEF.

Tabla 17. Conceder Acceso.

Nombre	Conceder Acceso
Autor	Carlos Torres Vinueza
Fecha	15/09/2015
Descripción	Mientras el prototipo se encuentra en modo "Leer", se aproxima un dispositivo celular con la opción Generar Ticket y es leído.
Actores	<ul style="list-style-type: none"> • Prototipo • GUI Android
Precondiciones	<ul style="list-style-type: none"> • Aplicación en el prototipo ejecutada. • Identificación de los datos del conductor en el aplicativo del prototipo. • Modo Leer activado.
Flujo normal	<ol style="list-style-type: none"> 1. El prototipo se encuentra en estado Leer. 2. Un dispositivo se acerca al prototipo y le comunica un mensaje NDEF. 3. El prototipo verifica los datos leídos. 4. Copia los datos del mensaje NDEF y los une a los datos del conductor y lo guarda en la base.
Flujo Alternativo	<ol style="list-style-type: none"> 1.1. Selección del botón Home o retroceso. 4.1. Datos leídos no válidos.
Post condiciones	<ul style="list-style-type: none"> • Lectura de mensaje NDEF. • Mensaje de confirmación.

Aplicativo Web



Usuarios y conductores pueden registrarse a través de esta herramienta. Pero para usuarios la página trabaja para desplegar los datos de sus interacciones en el sistema.

Tabla 17. Registrar (GUI Web).

Nombre	Registrar(GUI Web)
Autor	Carlos Torres Vinueza
Fecha	16/09/2015
Descripción	La interfaz crea una cuenta con los datos del usuario.
Actores	<ul style="list-style-type: none"> • GUI Web • Usuario
Precondiciones	<ul style="list-style-type: none"> • Seleccionar la opción "Registro". • Ingreso de los siguientes datos en la interfaz: ID, tipo de ID, grupo, Nombre, Teléfono, Dirección, Contraseña.
Flujo normal	<ol style="list-style-type: none"> 1. Selección de la función "Registrar". 2. Ingreso de los datos del usuario. 3. Guardar información en la base de datos.
Flujo Alternativo	<ol style="list-style-type: none"> 1.1. Selección del botón regresar. 2.1. Ingreso parcial de los datos solicitados. 2.2. Ingreso incorrecto de los datos solicitados. 2.3. Ingreso de datos ya incluidos en la base.
Post condiciones	<ul style="list-style-type: none"> • Ingreso de los datos en la base. • Mensaje de confirmación.

Tabla 18. Iniciar Sesión (GUI Web).

Nombre	Iniciar Sesión (GUI Web)
Autor	Carlos Torres Vinueza
Fecha	16/09/2015
Descripción	La interfaz inicia sesión con los datos del usuario.
Actores	<ul style="list-style-type: none"> • GUI Web • Usuario
Precondiciones	<ul style="list-style-type: none"> • Seleccionar la opción "Iniciar Sesión". • Ingreso de los siguientes datos en la interfaz: ID, Contraseña.
Flujo normal	<ol style="list-style-type: none"> 1. Selección de la función "Iniciar Sesión". 2. Ingreso de los datos solicitados por parte del usuario. 3. Guardar información en la base de datos.
Flujo Alternativo	<ol style="list-style-type: none"> 1.1. Selección del botón regresar. 2.1. Ingreso parcial de los datos solicitados. 2.2. Ingreso incorrecto de los datos solicitados. 2.3. Ingreso de datos ya incluidos en la base.
Post condiciones	<ul style="list-style-type: none"> • Sesión activa en la interfaz Web. • Mensaje de confirmación.

Tabla 19. Consultar (GUI Web).

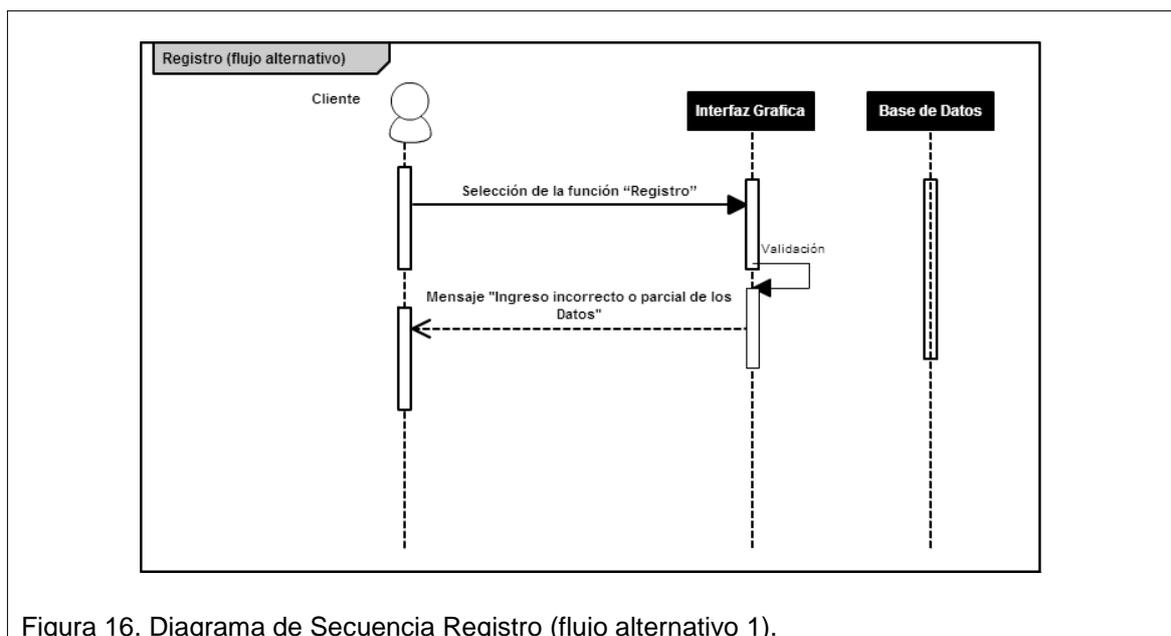
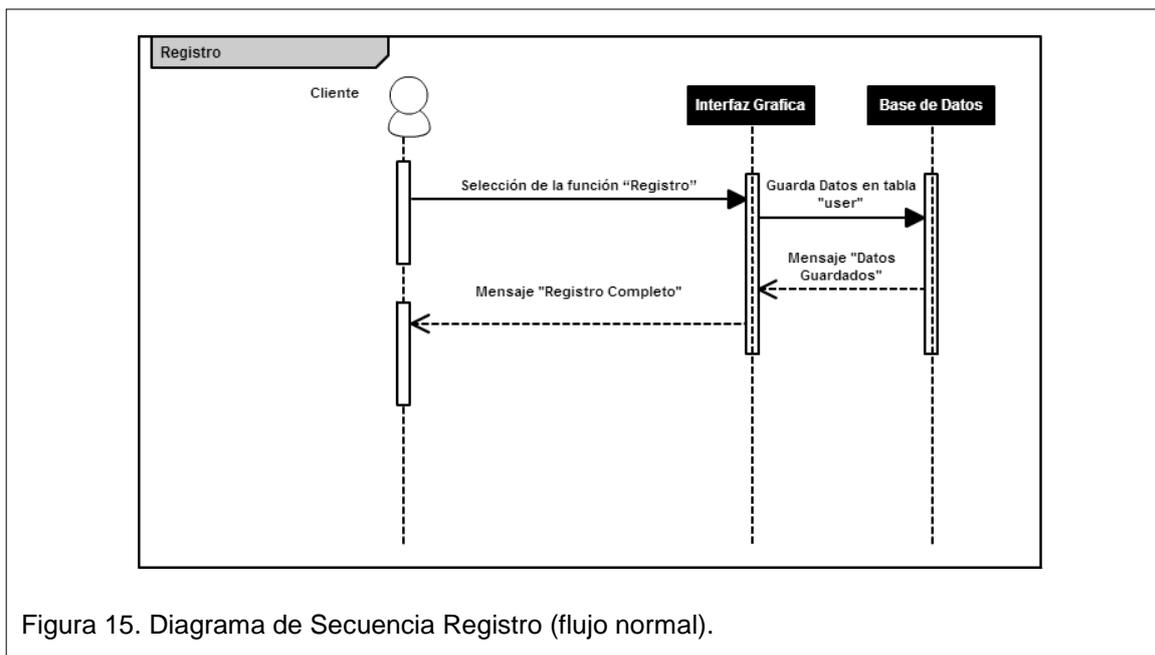
Nombre	Consultar (GUI Web)
Autor	Carlos Torres Vinueza
Fecha	16/09/2015
Descripción	El Usuario consulta los datos guardados en la base referente a su cuenta.
Actores	<ul style="list-style-type: none"> • GUI Web • Usuario
Precondiciones	<ul style="list-style-type: none"> • Aplicación en el prototipo ejecutada. • Identificación de los datos del conductor en el aplicativo del prototipo.
Flujo normal	<ol style="list-style-type: none"> 1. Inicia Sesión. 2. Selección de la función "Consultar". 3. Se despliega toda la información referente a la cuenta del usuario.
Flujo Alternativo	<ol style="list-style-type: none"> 1.1. Selección del botón regresar.
Post condiciones	<ul style="list-style-type: none"> • Información de la cuenta desplegada en pantalla. • Disponible los botones regresar e imprimir.

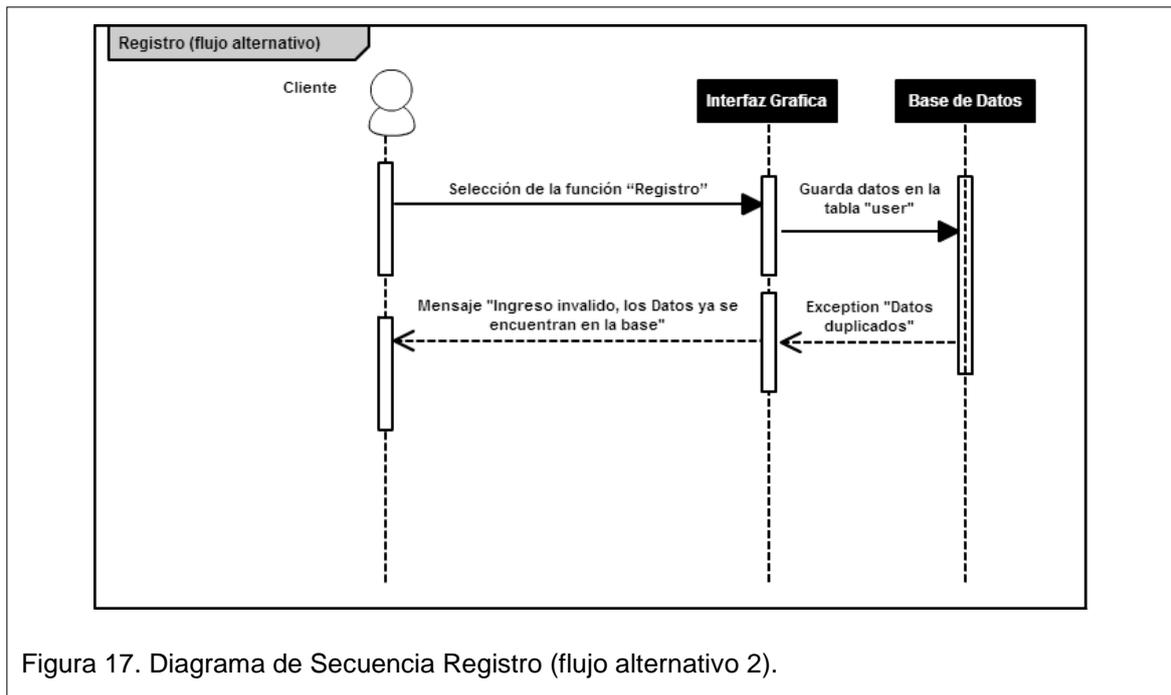
3.2. Metodología de Diseño de Hipermedia Orientada a Objetos (OOHDM)

3.2.1. Diagramas de Secuencia

Se presentan los flujos o secuencias previamente analizados en los casos de uso. A continuación se observaran los diagramas de secuencia de los casos de uso por el actor Usuario, en este caso representado como el Cliente.

Registro:





El flujo de la operación Registro en la figura 15, 16 y 17 verifica que no existan usuarios con el mismo ID o user name, para evitar inconsistencias en la base de datos. También se encarga de filtrar el ingreso de datos, validando campos y valores de igual forma para evitar errores al momento de guardar o desplegar la información de la base de datos.

Ingreso:

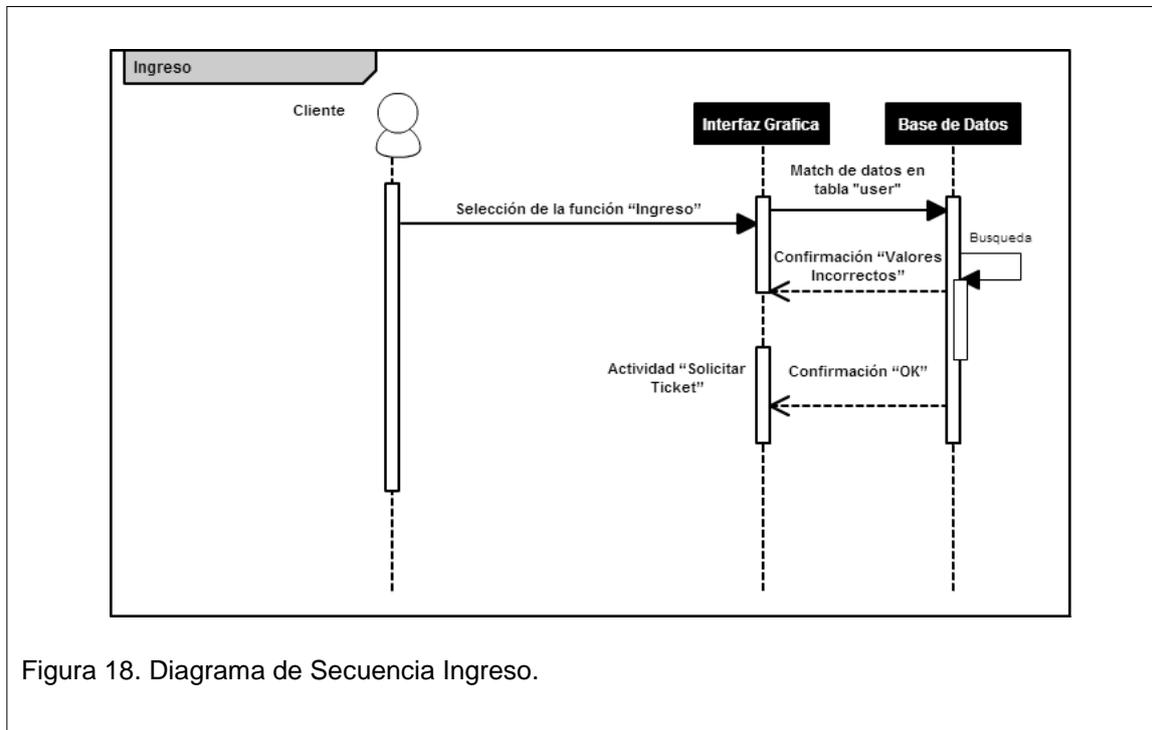


Figura 18. Diagrama de Secuencia Ingreso.

Solicitar Ticket:

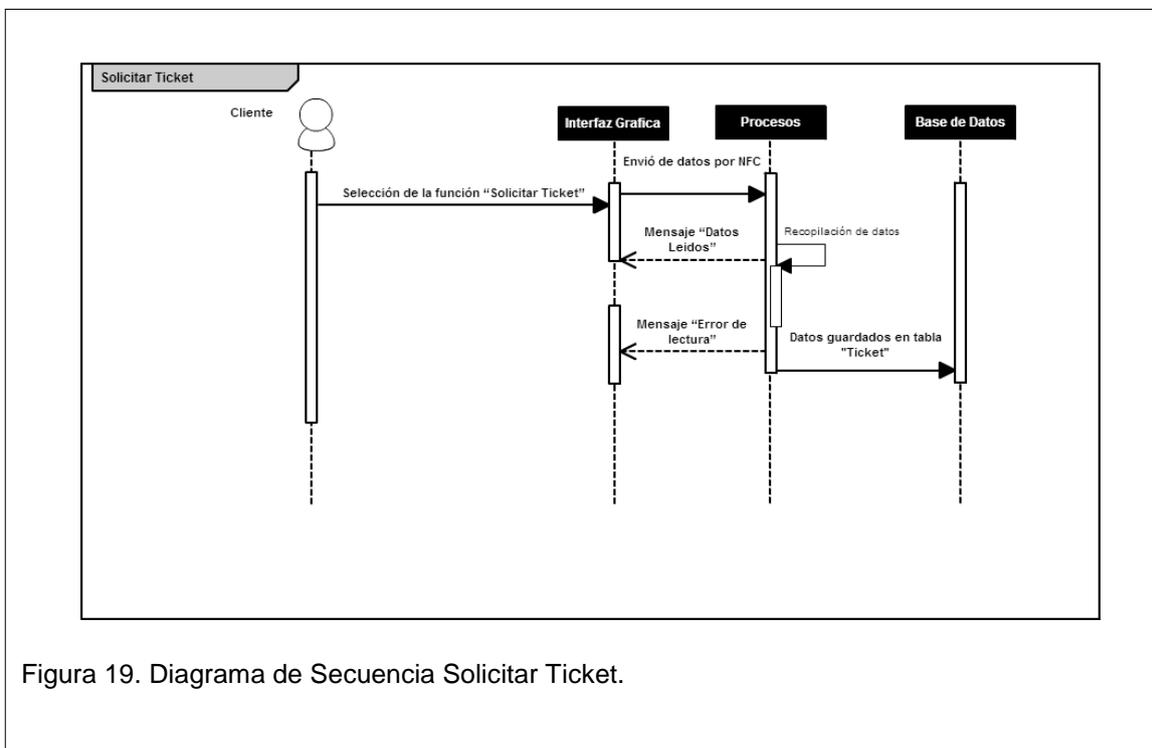


Figura 19. Diagrama de Secuencia Solicitar Ticket.

Consultar Datos:

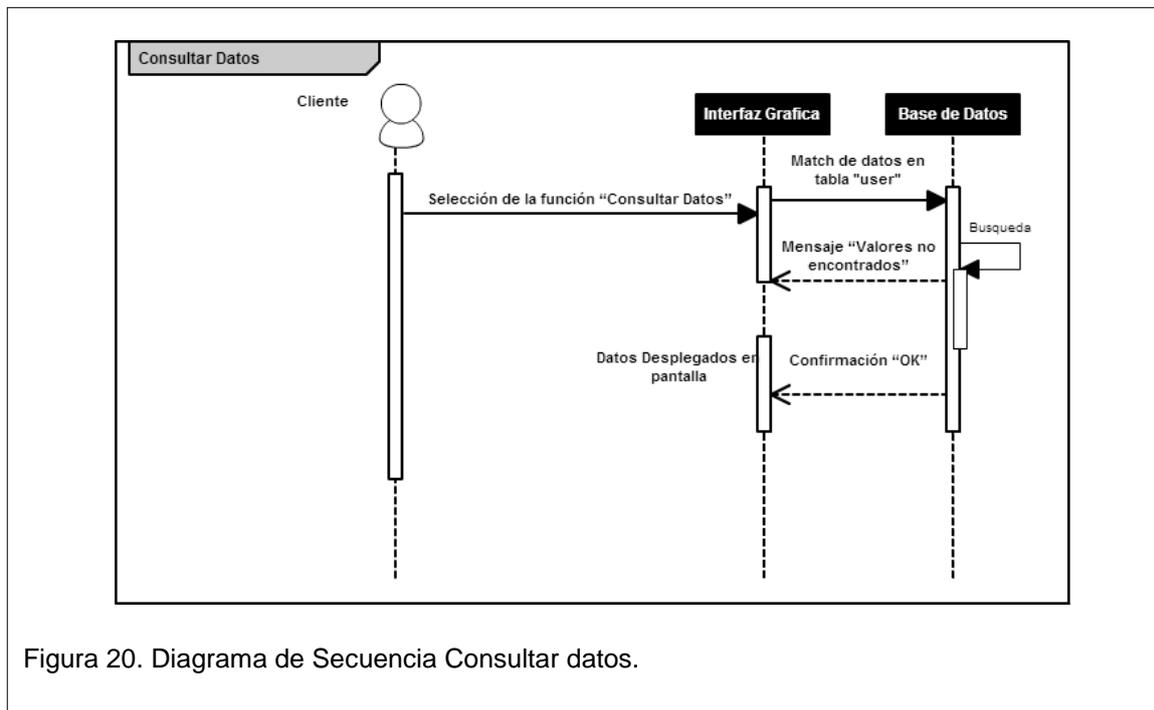


Figura 20. Diagrama de Secuencia Consultar datos.

3.2.2. Diagramas de Clases

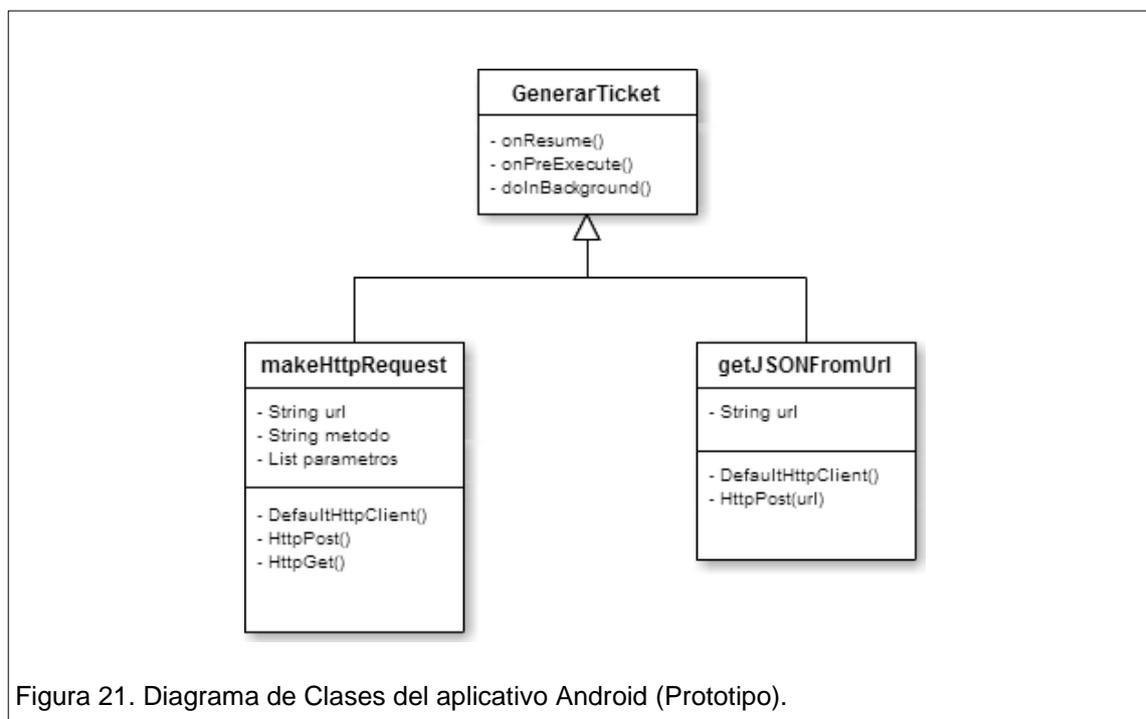
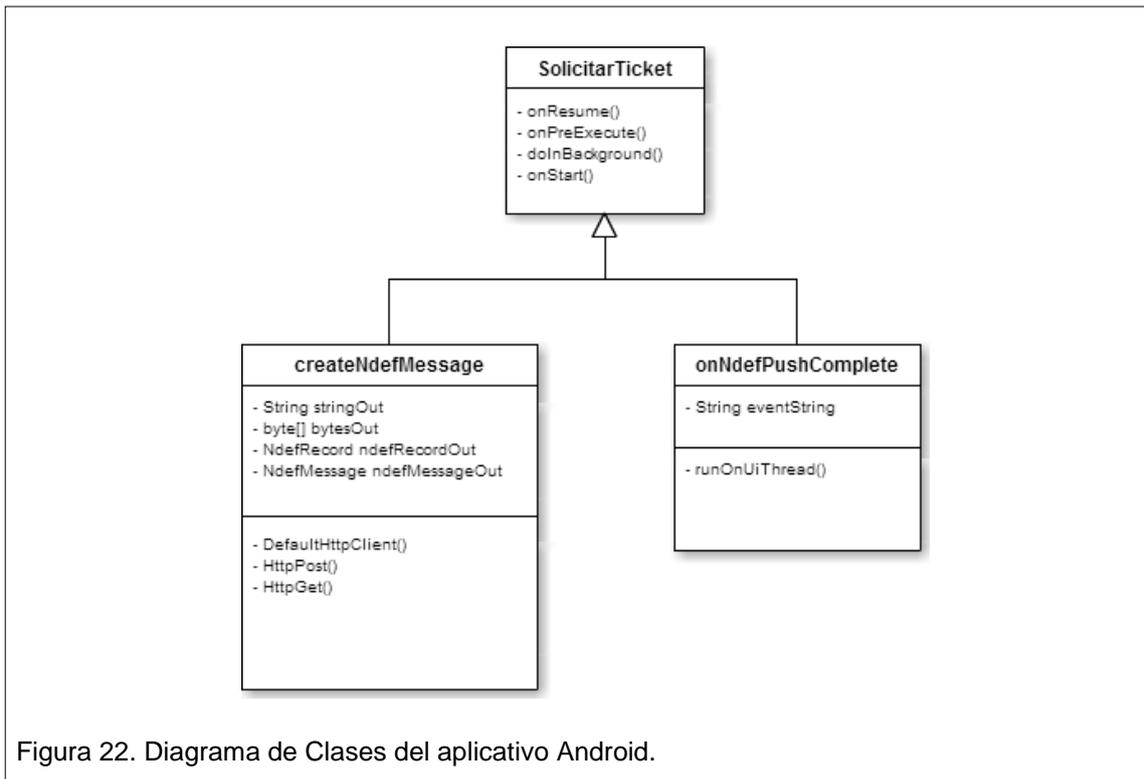


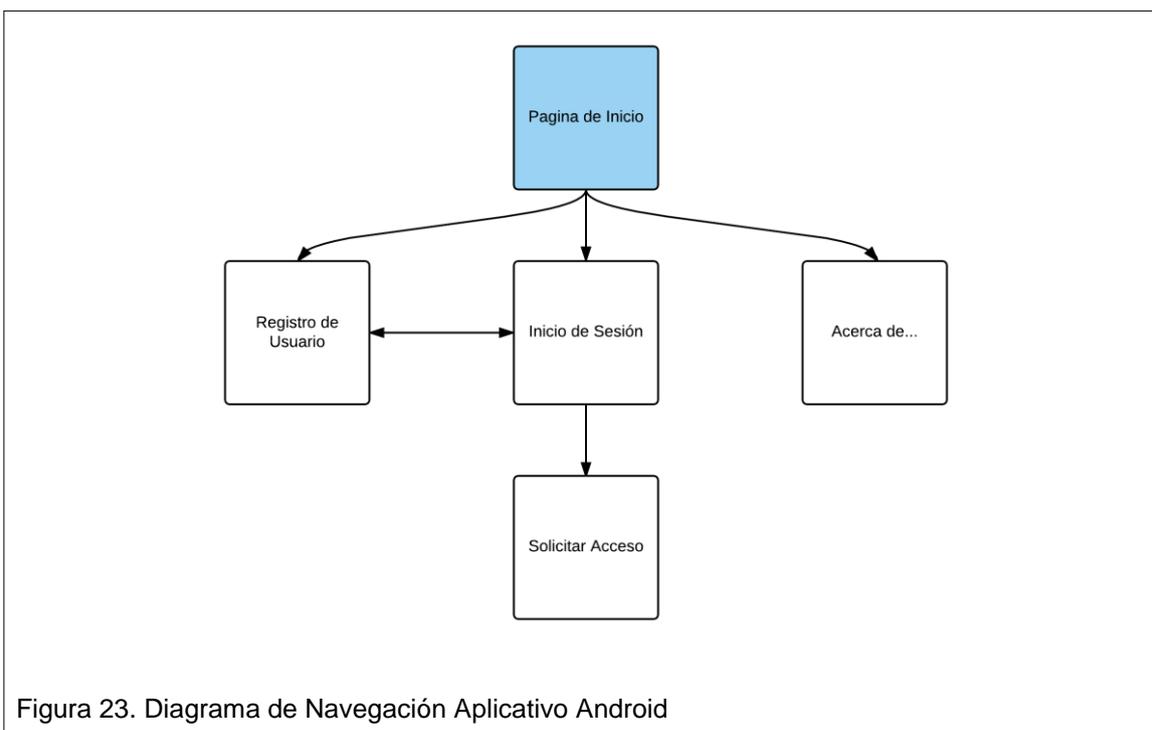
Figura 21. Diagrama de Clases del aplicativo Android (Prototipo).



El diagrama de clases para el aplicativo Android brinda una idea de las variables de entrada y salida, en los métodos programados resultando útil en el momento de implementar cambios o mejoras en el aplicativo.

3.2.3. Diseño de Navegación

GUI Android:



GUI Prototipo:

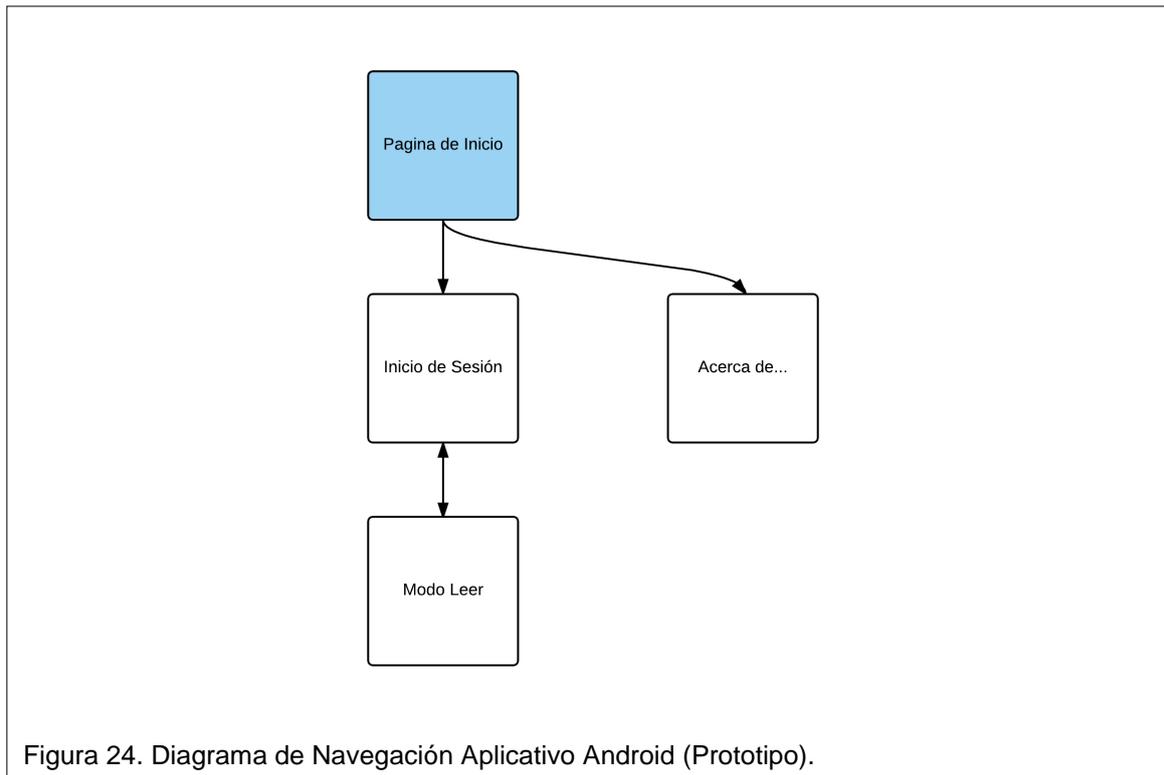


Figura 24. Diagrama de Navegación Aplicativo Android (Prototipo).

GUI Web:

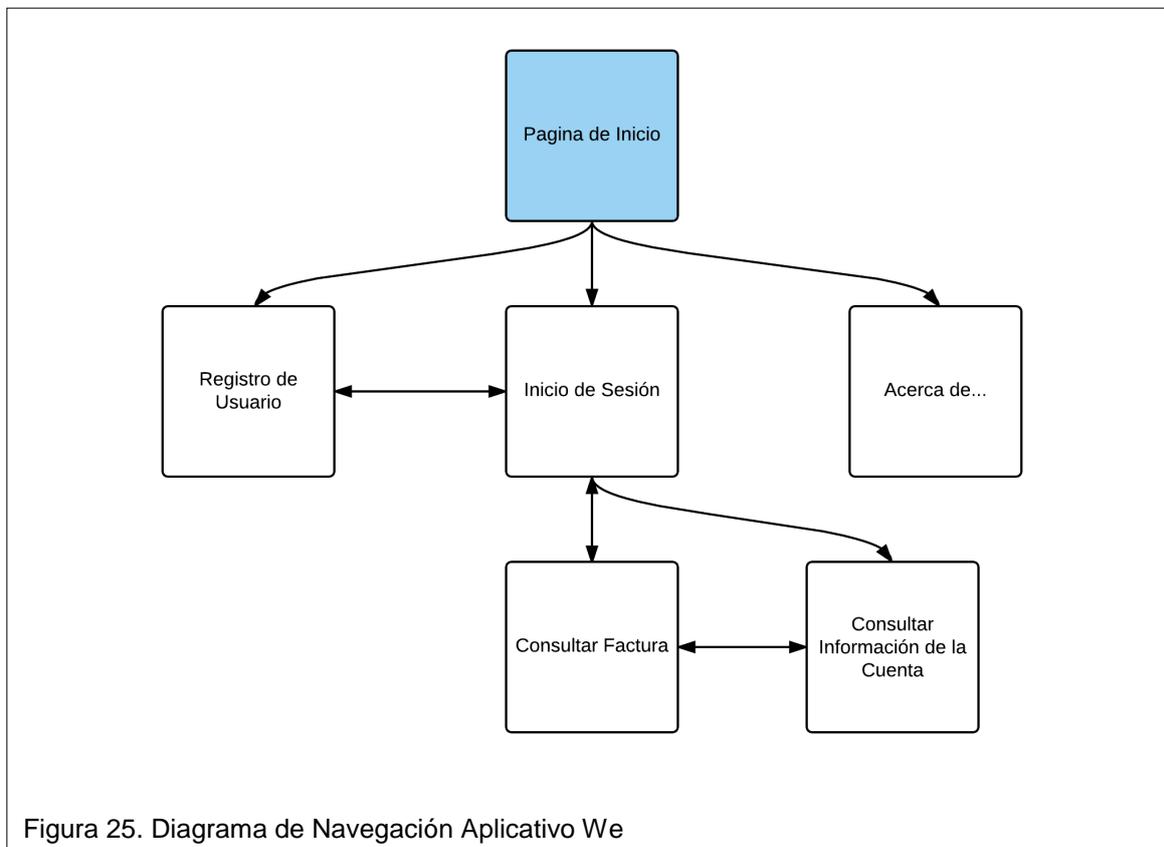
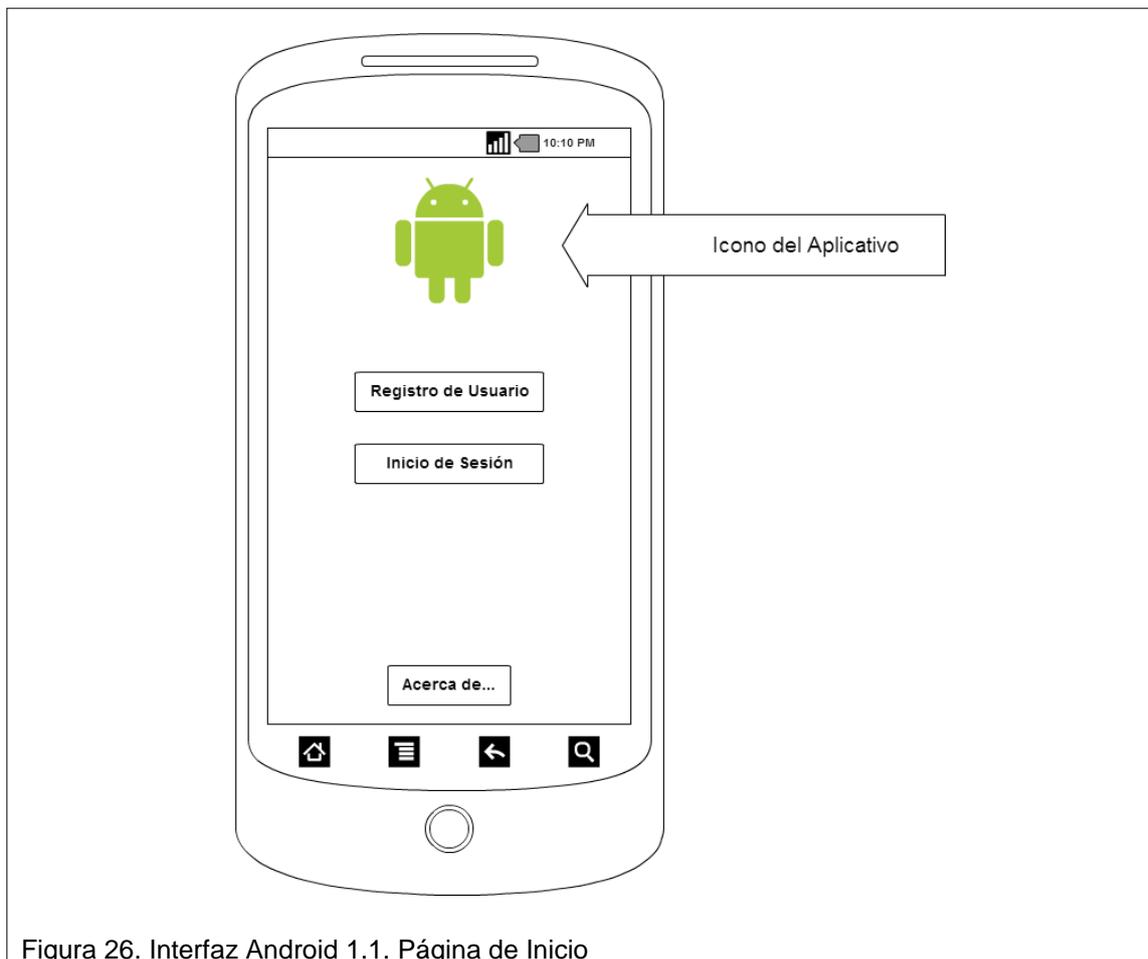


Figura 25. Diagrama de Navegación Aplicativo We

3.2.4. Diseño de Interfaces

GUI Android:

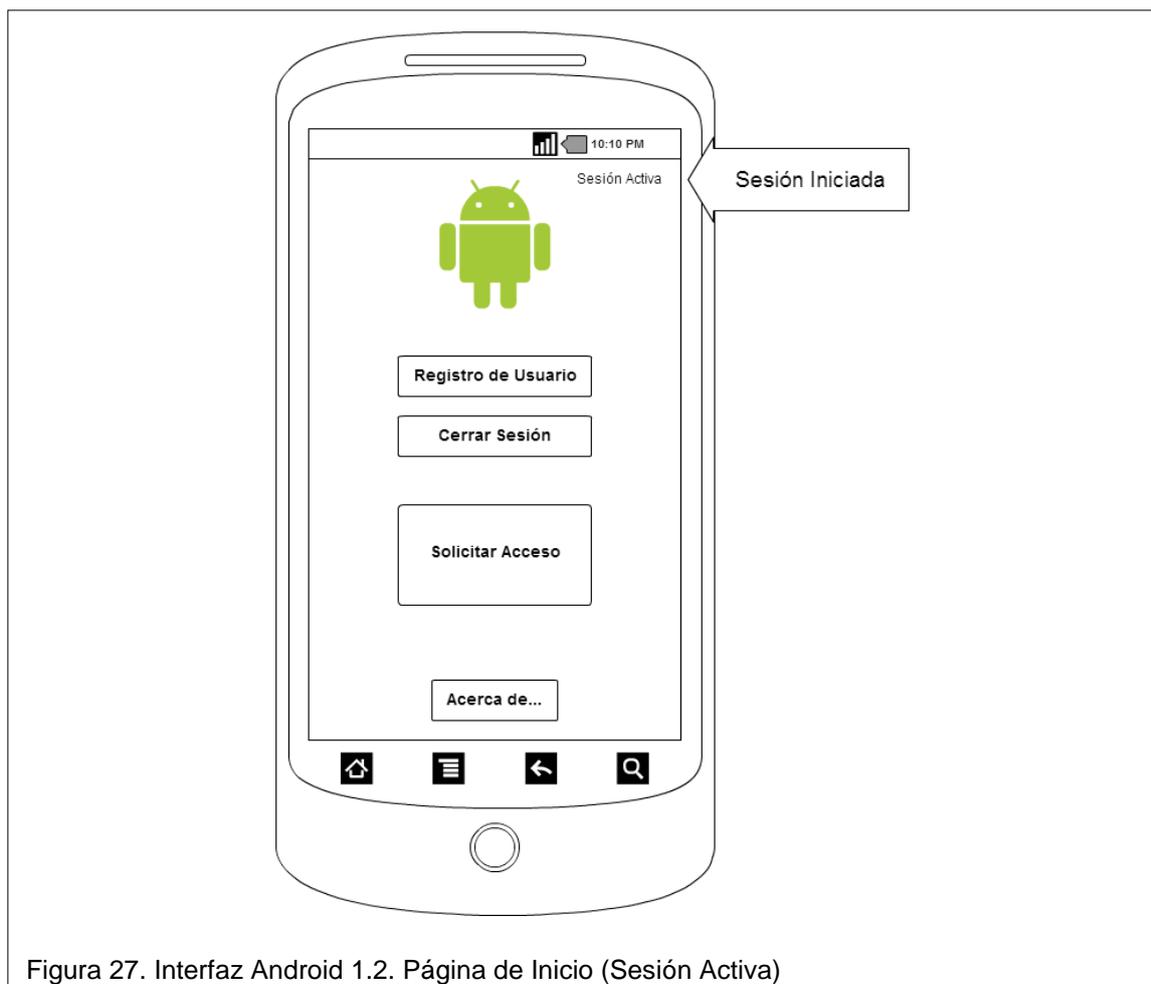


La pantalla inicial de la página de inicio mostrara los siguientes elementos:

- Registro de Usuario(Botón)
- Inicio de Sesión(Botón)
- Acerca de...(Botón)

Una vez iniciada la sesión, el botón “Solicitar Acceso” será visible, y el botón “Iniciar Sesión” desaparecerá.

El botón “Registro de Usuario” será visible aunque se inicie sesión en el aplicativo.



La pantalla inicial con una sesión activa, tendrá activado los siguientes elementos:

- Registro de Usuario(Botón)
- Cerrar Sesión(Botón)
- Solicitar Acceso(Botón)
- Acerca de...(Botón)
- Sesión Activa(Etiqueta)

Adicional existirá una etiqueta informativa con la frase “Sesión Activa”.

El botón “Acerca de...” nos mostrara la información del desarrollador.

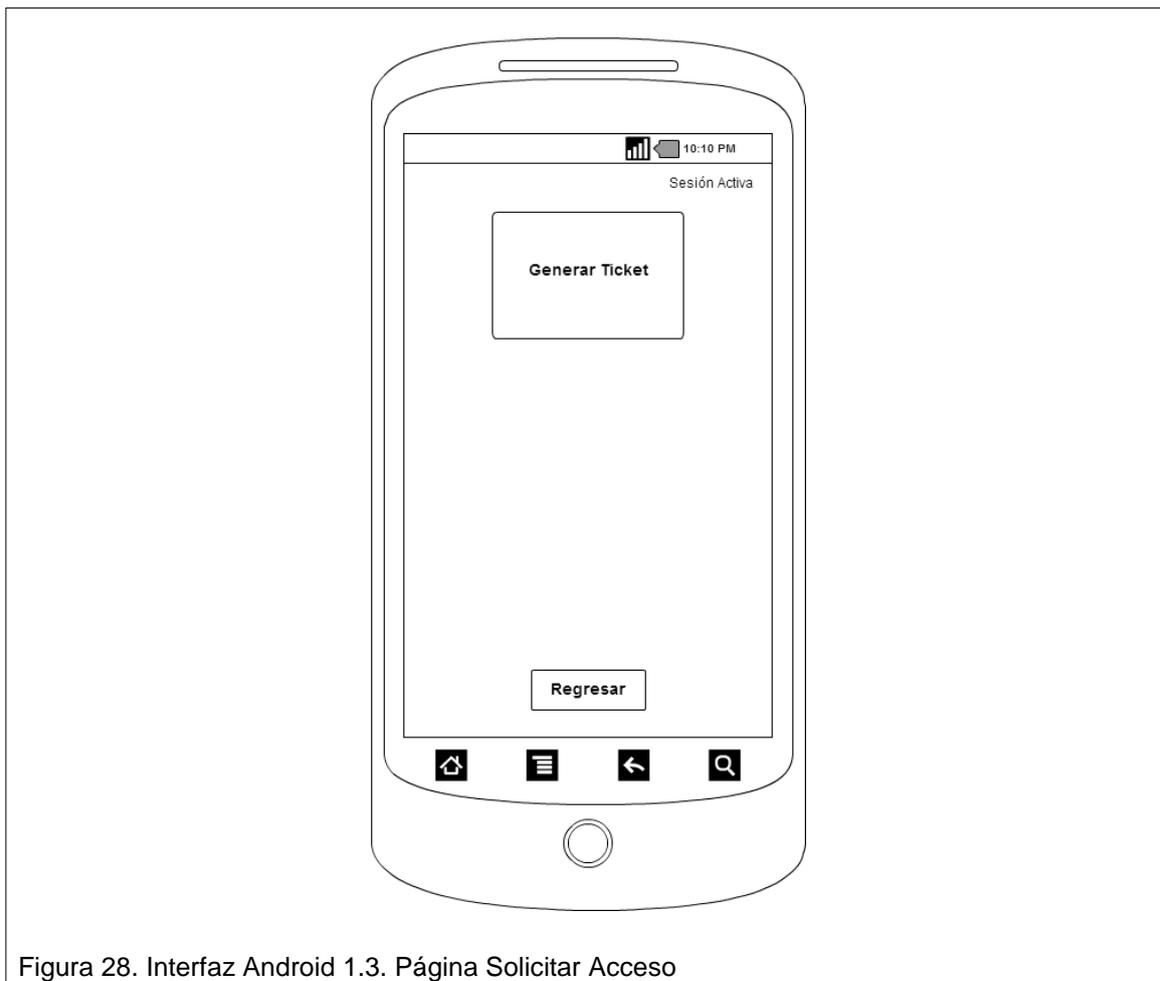


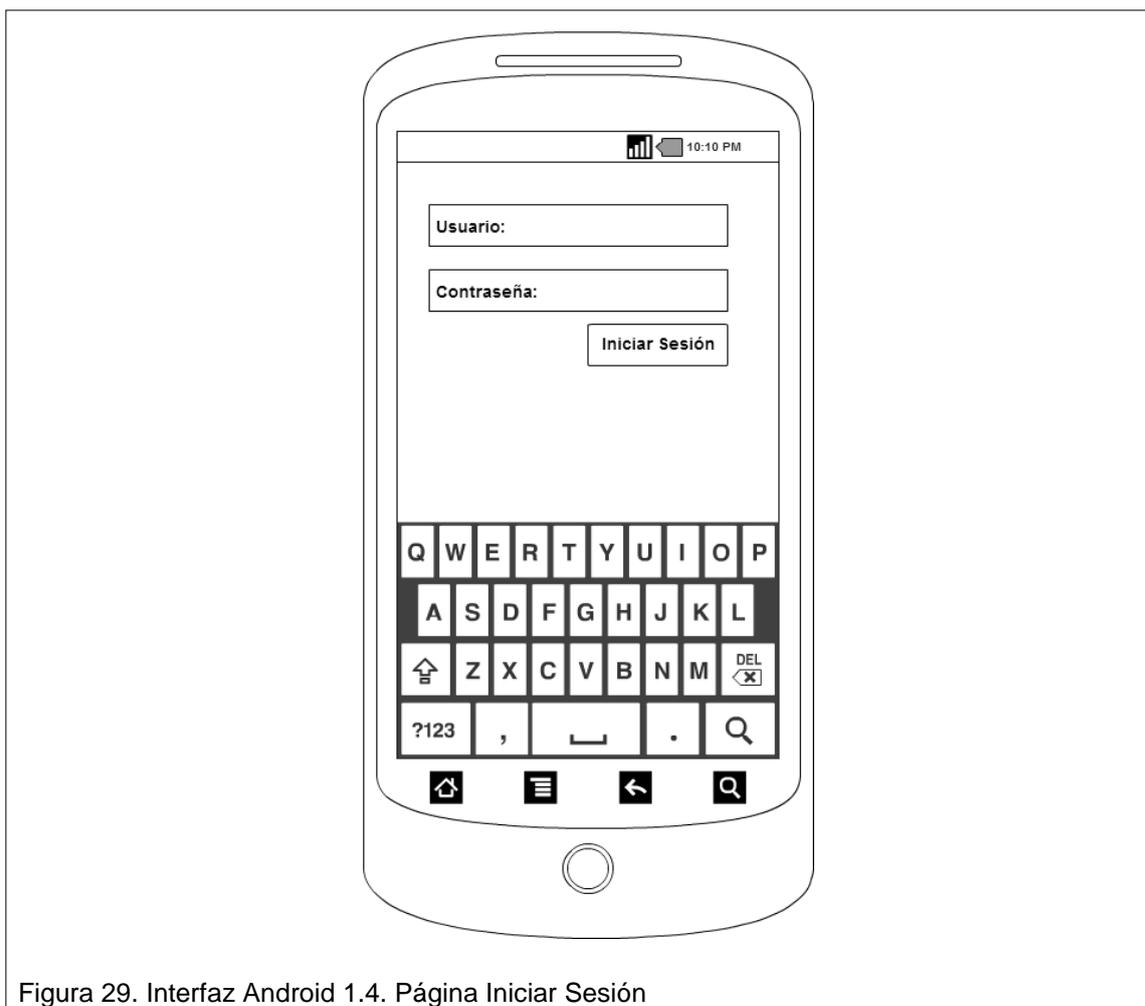
Figura 28. Interfaz Android 1.3. Página Solicitar Acceso

En esta página se muestran los siguientes elementos:

- Generar Ticket(Botón)
- Regresar(Botón)
- Sesión Activa(Etiqueta)

El botón “Generar Ticket” mostrara en pantalla un mensaje el cual indique al usuario que ya puede acercar su dispositivo al prototipo para realizar la identificación.

El botón “Regresar” podrá ser usado para volver a la pantalla de inicio.



En esta pantalla se muestran los siguientes elementos:

- Usuario(Campo de Texto)
- Contraseña(Campo de Texto)
- Iniciar Sesión(Botón)

Los elementos “Usuario” y “Contraseña” permitirán ingresar al usuario sus datos para la verificación e inicio de sesión.



Figura 30. Interfaz Android 1.5. Página Registro de Usuario

En esta pantalla se muestran los siguientes elementos:

- Usuario(Campo de Texto)
- Tipo(Botones Radiales)
 - Cédula
 - Pasaporte
- Grupo(Botones Radiales)
 - Adulto
 - Tercera Edad
 - Menor de Edad
 - Con Discapacidad
- Nombre(Campo de Texto)
- Teléfono(Campo de Texto)
- Dirección(Campo de Texto)

- Contraseña (Campo de Texto)
- Registro(Botón)

GUI Prototipo:



Figura 31. Interfaz Android 2.1. Página Principal

En esta pantalla se muestran los siguientes elementos:

- Iniciar Sesión(Botón)
- Acerca de...(Botón)

En esta pantalla el conductor podrá seleccionar la opción de iniciar sesión, en donde además de sus datos tendrá que ingresar la información referente a la unidad de transporte.



Figura 32. Interfaz Android 2.2. Página Principal (Sesión Activa)

En esta pantalla se muestran los siguientes elementos:

- Cerrar Sesión(Botón)
- Modo Lectura(Botón)
- Acerca de...(Botón)
- Sesión Activa(Etiqueta)

En esta pantalla el conductor podrá seleccionar la opción de modo lectura. Esta opción es la que permite la lectura de cualquier dispositivo que ingrese al rango de funcionamiento del NFC.

GUI Web:

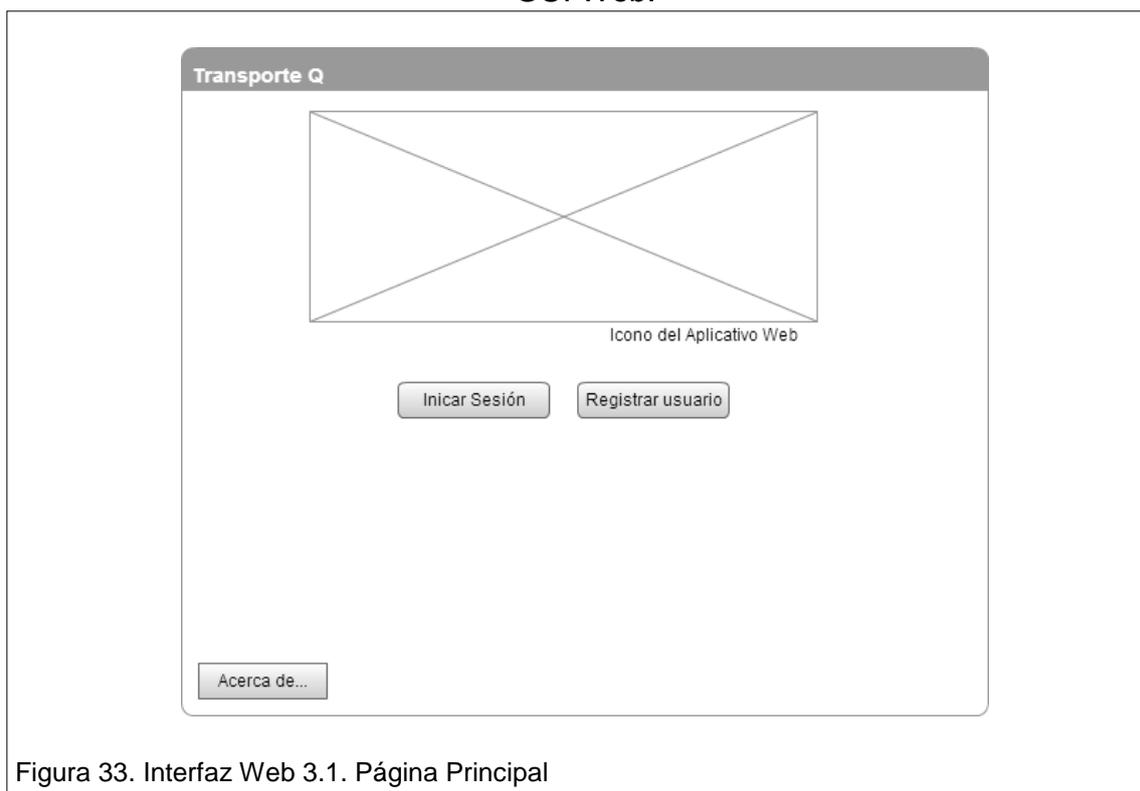


Figura 33. Interfaz Web 3.1. Página Principal

En esta pantalla se muestran los siguientes elementos:

- Iniciar Sesión(Botón)
- Registrar Usuario(Botón)
- Acerca de...(Botón)

3.2.5. Arquitectura del Sistema

El sistema está conformado por 3 elementos, Servidor (aplicativo Web y Base de Datos), Aplicación en Android para el usuario y la aplicación en Android para el conductor.

Entre la aplicación del usuario y el conductor, el método de conexión para la transferencia de datos es NFC. La comunicación se establece a una distancia máxima de 3 cm y los datos son verificados vía WAN al servidor de base de datos. En este sistema, el servidor de aplicación Web y de base de datos es simulado en una laptop (Figura 34.), pero para la implementación del proyecto en una segunda fase se debe considerar tenerlos en 2 servidores separados y considerar servidores de contingencia y desarrollo.

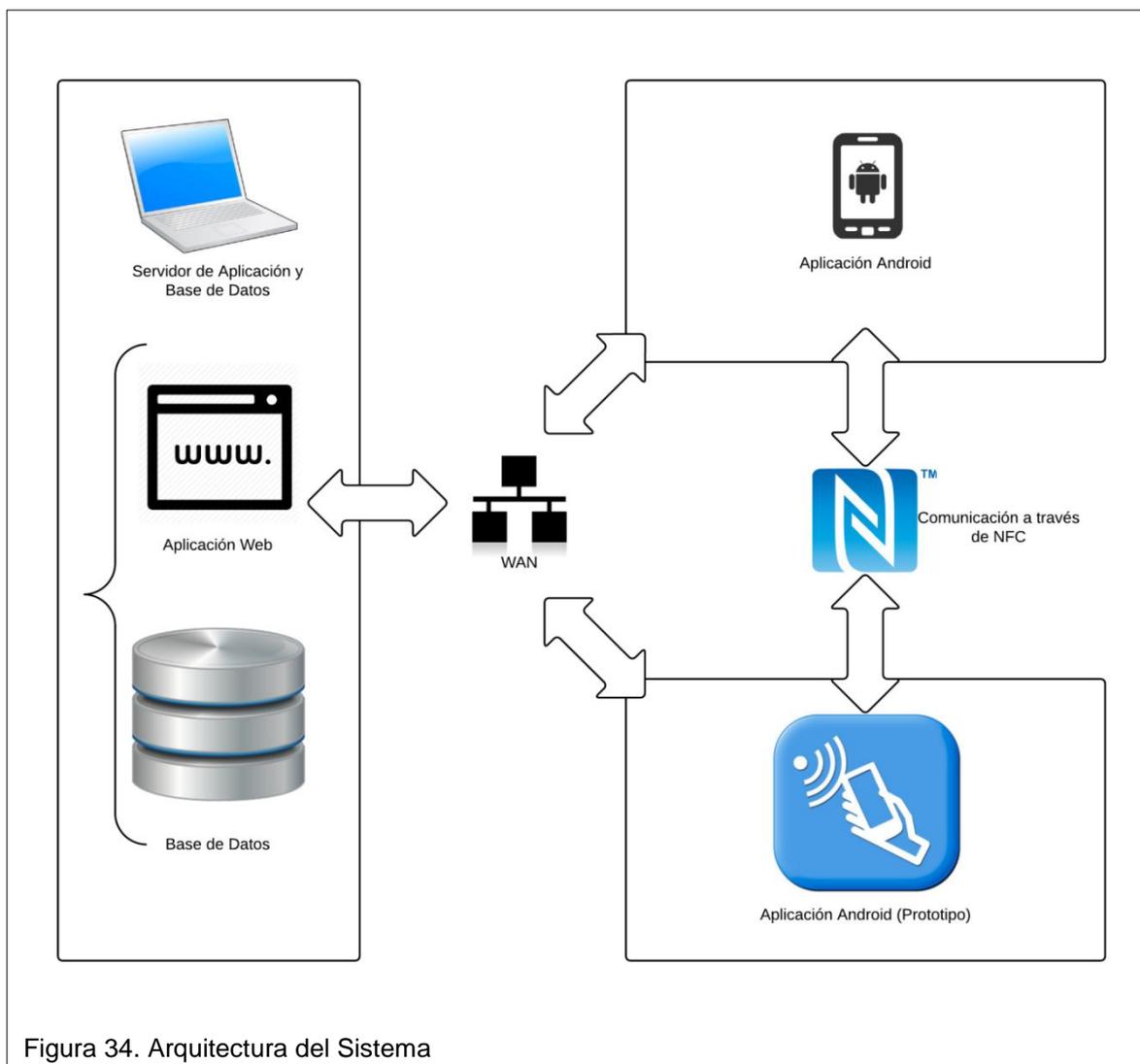


Figura 34. Arquitectura del Sistema

- DELETE

Una vez que el servicio Web obtiene la información del usuario y es aplicado cualquiera de las funciones listadas se analiza por medio de JSON, el cual es un formato de texto ligero ideal para el intercambio de datos.

Se maneja acceso a la base de datos e interacción de datos, de esta forma es como los usuarios pueden crear una cuenta en el sistema, ingresar al sistema y realizar el pago en las unidades de transporte.

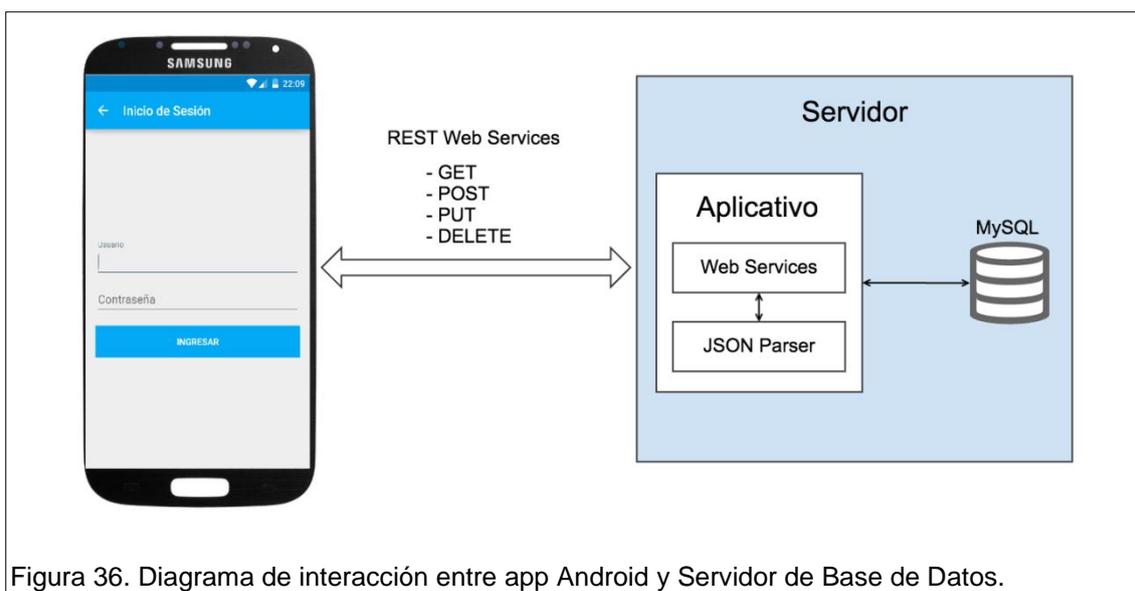


Figura 36. Diagrama de interacción entre app Android y Servidor de Base de Datos.

De esta forma la información sale del servidor en forma de objetos JSON, los cuales son leídos e interpretados en objetos Java que pueden ser manipulados por el adaptador de Android y colocarlos finalmente en variables dentro del aplicativo.

Esta información puede ser desplegada en variables, campos de texto o ListViews dependiendo de las necesidades y utilidad de la información. Para el aplicativo que emula al prototipo el funcionamiento es exactamente igual.

3.3. Consideraciones de implementación en la Fase 2

Es importante tener en cuenta ciertas consideraciones en la implementación real de este proyecto, debido a que en este trabajo el prototipo es emulado en su totalidad a través de un dispositivo móvil, puesto que los elementos a

implementar en el prototipo incluían el poder soportar el sistema operativo Android, debido a que de esta forma se aprovechan los protocolos de seguridad de la tecnología NFC.

De igual forma el servidor de aplicaciones y base de datos fue simulada a través de un computador, ya que la capacidad del servidor requerido puede ser física o virtual para la fase inicial de pruebas.

El sistema puede seguir funcionando aún, si el usuario o el conductor no tienen conexión a la red, ya que la información es almacenada localmente hasta que se restablezca la conexión a la red.

Prototipo

En la fase 2 o fase de implementación del proyecto el prototipo tendrá que estar conformado por una mini placa base con sistema operativo Android y un módulo NFC para poder identificar a los usuarios como vemos en la figura 36:

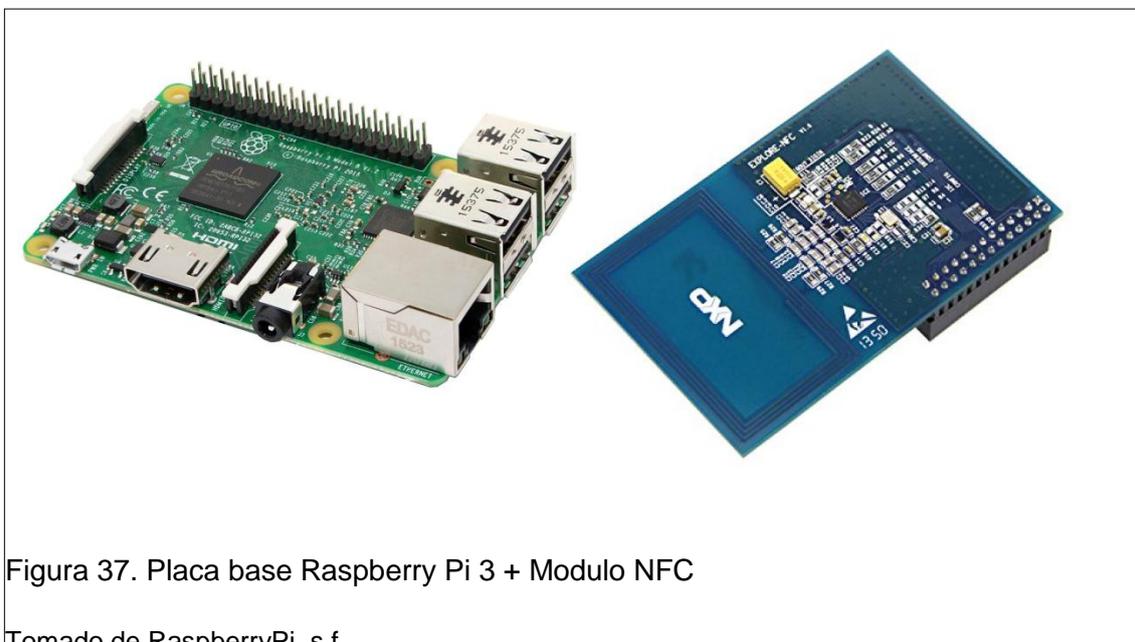


Figura 37. Placa base Raspberry Pi 3 + Modulo NFC

Tomado de RaspberryPi, s.f.

Actualmente Raspberry Pi soporta el sistema operativo Android mediante versiones modificadas para optimizar la capacidad del dispositivo con los elementos gráficos del SO.

La ventaja de utilizar Raspberry Pi, es la facilidad de conexión a módulos de relé el cual puede ser aprovechado para la conexión de los tornos de acceso en las unidades de transporte o en las estaciones de los diferentes sistemas de transporte.

En el mercado existen 3 tipos de tornos de acceso o accesos mecánicos, en los que se consideran los lugares a los cuales se requieren ser instalados, es decir que su forma y función dependen únicamente del lugar en donde serán instalados. Para este proyecto pueden estar dispuestos a la entrada de cada estación de espera o dentro de cada unidad y su forma está dispuesta como se muestra en la figura 37:



Los controles de acceso mecánicos de cercamiento y cuerpo entero son óptimos para el funcionamiento en estaciones y casetas de espera de las líneas de buses, mientras que los controles de acceso de medio cuerpo son mejores para la instalación dentro de cada unidad debido a su menor tamaño.

3.3.1. Servidores de Aplicaciones

Para la implementación del proyecto se requiere un servidor dedicado para aplicaciones exclusivamente, en donde se albergara el aplicativo Web y herramientas de análisis de Data.

Es importante recalcar que la capacidad de dichos servidores depende únicamente de la cantidad de consultas realizadas a la aplicación y la optimización del sistema operativo, debido a que mientras aumentan la cantidad de usuarios conectados al aplicativo Web, no importa cuanta memoria se agregue pueden surgir problemas relacionados al servicio Apache.

Con estas consideraciones se pueden calcular aproximadamente 4000 usuarios máximo por servidor web configurados con 4 cores Xeon y 4 Gb RAM. El tener varios servidores Web permite balancear la carga y repartirla entre todos los servidores del clúster, además de que se obtiene la posibilidad de agregar o quitar servidores dependiendo de la carga que el aplicativo web tenga.

También se debe considerar un esquema similar con ambiente de contingencia, en donde se disponga dentro de lo posible la misma capacidad que el ambiente de producción soporte.

3.3.2. Servidores de Base de Datos

Para el servidor MySQL se puede considerar una configuración con capacidad mayor, es decir 8 cores con 8 Gb de RAM cada uno, para obtener un margen de carga estimada cercano al 100%.

Esta configuración está pensada para un sistema desarrollado íntegramente en PHP y MySQL, y el número de servidores dependerá de la cantidad de usuarios estimado y la idea de mantener la base de datos separada es de tener el contenido estático del sistema totalmente independiente.

4. DESARROLLO Y PRUEBAS

4.1. Desarrollo de la Base de Datos en MySQL

4.1.1. Propósito de la Base de Datos

El propósito del desarrollo de una base de datos en este aplicativo radica en la necesidad de vincular los datos del proveedor del servicio con los datos del usuario. El ideal de vincular todas las tecnologías mencionadas tiene como fin proporcionar información sobre las interacciones que se realizan diariamente a los actores involucrados.

Además, se definen los siguientes beneficios:

- La recopilación inteligente de estos datos simplificará el funcionamiento del servicio, entregando información simple pero cernida a los requerimientos propuestos.
- Al interactuar con sistemas aislados, existe la necesidad de mantener un formato en el tipo de datos que se almacenan.
- Se requiere el almacenamiento único de los datos, para entregar consistencia en la información. Esto no hace referencia al respaldo que pueden existir de la base.
- La base de datos proporciona privacidad a la información, agregando seguridad al sistema.

4.1.2. Definición de Tablas Necesarias (Diseño Conceptual)

Tablas Principales:

- User
- Bus
- Ticket

La tabla “user” es la tabla del Usuario. Recopilará la información pertinente a los datos de cada usuario del sistema. Esta tabla es considerada principal debido a que en ella se guardarán los datos más relevantes para el usuario.

La tabla “bus” será la tabla encargada de guardar la información relacionada a la unidad de transporte. Es considerada principal ya que estará presente en la generación de los tickets virtuales que permitirán el acceso a los usuarios.

Como se mencionó anteriormente, la información del usuario (tabla “user”) y la información del medio de transporte (tabla “bus”) iría vinculados en un ticket virtual, por lo que se necesitan almacenar en una tabla diferente (tabla “ticket”).

Tablas Secundarias:

- Tipo
- Grupo
- Cuenta
- Conductor
- Bustipo
- Cooperativa
- Ruta

Las tablas “tipo” y “grupo” hacen referencia al tipo de documento de identificación (Cédula, Pasaporte) y al grupo al que el usuario pertenece (Adulto, Tercera Edad, Menor de Edad, Con Discapacidad) respectivamente, mientras que “bustipo”, “cooperativa” y “ruta” almacenaran información relativa al medio de transporte y los datos producto de sus funciones en el servicio.

La tabla “cuenta” guardara información detallada del estado de la cuenta del usuario con el objetivo de separar la relación de un usuario con los detalles comunes de su cuenta, así el detalle de la activación de una cuenta podrá ser identificada como “activa” o “deshabilitada”, sin eliminar ni alterar la información del usuario.

Finalmente la tabla “conductor” guardara la información y detalles del conductor del medio de transporte, similar a la tabla “user” pero únicamente mostrando campos útiles para el sistema y el usuario.

4.1.3. Definición de Campos Necesarios (Diseño Lógico)

Aquí se definen las columnas necesarias para el funcionamiento de los procesos propuestos.

Tabla “user”

En esta tabla se considerará la información necesaria para cumplir con los procesos de Ingreso y Registro, por lo que serán necesarias columnas para el identificador de usuario o “username”(clave primaria de la tabla), “cedula de identidad”, “grupo” haciendo referencia a la capacidad del sistema de diferenciar a los usuarios en 4 tipos (Adulto, Tercera Edad, Menor de Edad, Con Discapacidad), “tipo” refiriéndose al tipo de identificación en la cedula de identidad (Cédula, Pasaporte), “nombre”, “teléfono”, “dirección” y finalmente “password” el cual será almacenado con encriptación MD5, proceso encargado por el gestor de la base de datos (phpMyAdmin).

De esta manera se define para la tabla “user” las siguientes columnas:

- user_id (clave primaria)
- cedulaidentidad
- grupo_id (clave foránea)
- tipo_id (clave foránea)
- nombre
- teléfono
- cuenta_id (clave foránea)
- dirección
- password

Tabla “bus”

En la tabla “bus” se definirán la información necesaria para identificar el número de unidad de transporte único, la cooperativa a la que pertenece y el tipo (Trolebús, Ecovía y Bus urbano).

Se define para la tabla “bus” las siguientes columnas:

- bus_id (clave primaria)
- bustipo_id (clave foránea)
- coop_id (clave foránea)

Tabla “ticket”

La tabla “ticket” almacenará la información del usuario, conductor, unidad de transporte y la relacionara con un campo que almacene la fecha y hora, así se lograra reunir la información necesaria para la creación de un ticket virtual.

Se define para la tabla “ticket” las siguientes columnas:

- ticket_id (clave primaria)
- fecha
- user_id (clave foránea)
- bus_id (clave foránea)
- conductor_id (clave foránea)

Tabla “tipo”

Aquí se definen los tipos de identificación que se asignarán en la tabla “user”, estos pueden ser Cédula o Pasaporte. Esta tabla nos ayudara a mantener un orden si se requiere aumentar o detallar el tipo de documentación de identificación ingresado.

Se define para la tabla “tipo” las siguientes columnas:

- tipo_id (clave primaria)
- descripción

Tabla “grupo”

Al igual que en la tabla “tipo”, en la tabla “grupo” se define y se detalla la información relacionada al grupo al que pertenece el usuario. Siendo estos valores Adulto, Tercera Edad, Menor de Edad o Con Discapacidad.

Se define para la tabla “grupo” las siguientes columnas:

- grupo_id (clave primaria)
- descripción

Tabla “cuenta”

En la tabla “cuenta” se lleva el registro del saldo del usuario y la información si esta cuenta esta activa o deshabilitada.

Se define para la tabla “cuenta” las siguientes columnas:

- cuenta_id (clave primaria)
- estado
- saldo

Tabla “conductor”

Esta tabla contendrá únicamente información vinculada al conductor de la unidad de transporte. Esta información está separada de la tabla “bus” debido a que un conductor puede manejar varias unidades de transporte.

Se define para la tabla “conductor” las siguientes columnas:

- conductor_id (clave primaria)
- nombre
- dirección
- teléfono
- cedulaidentidadc

Tabla “bustipo”

En la tabla “bustipo” se detalla cada tipo de buses que conforman el sistema Metrobús Q, siendo estos valores Trolebús, Ecovía y Buses Urbanos.

Se define para la tabla “bustipo” las siguientes columnas:

- bustipo_id (clave primaria)
- descripcionbus

Tabla “cooperativa”

En la tabla “cooperativa” de igual forma se detallan los nombres de las compañías de transporte asignándoles un número de identificación. Esta tabla permitirá la inclusión de futuras compañías sin comprometer los datos de las demás tablas.

Se define para la tabla “cooperativa” las siguientes columnas:

- coop_id (clave primaria)
- descripcioncoop

Tabla “ruta”

En esta tabla se asigna un número de identificación a las diferentes rutas del sistema Metrobús Q. Esta tabla permitirá la inclusión de rutas nuevas en caso de expansión del servicio.

Se define para la tabla “ruta” las siguientes columnas:

- ruta_id (clave primaria)
- descripción

4.1.4. Definición de Relaciones (Diseño Físico)

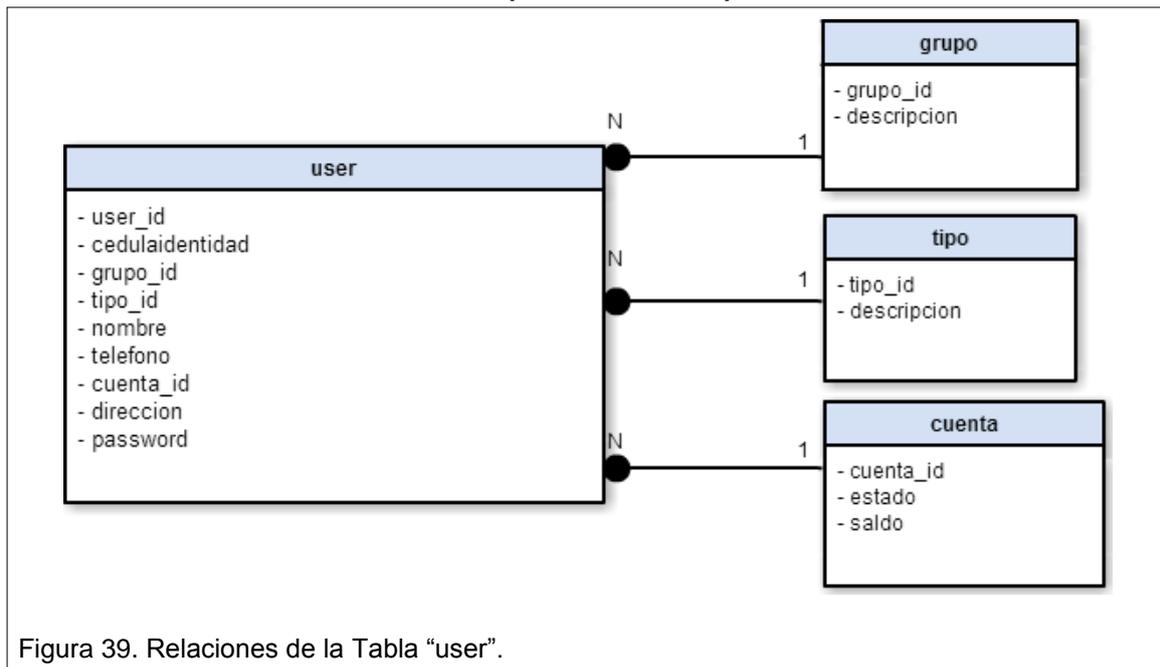


Figura 39. Relaciones de la Tabla "user".

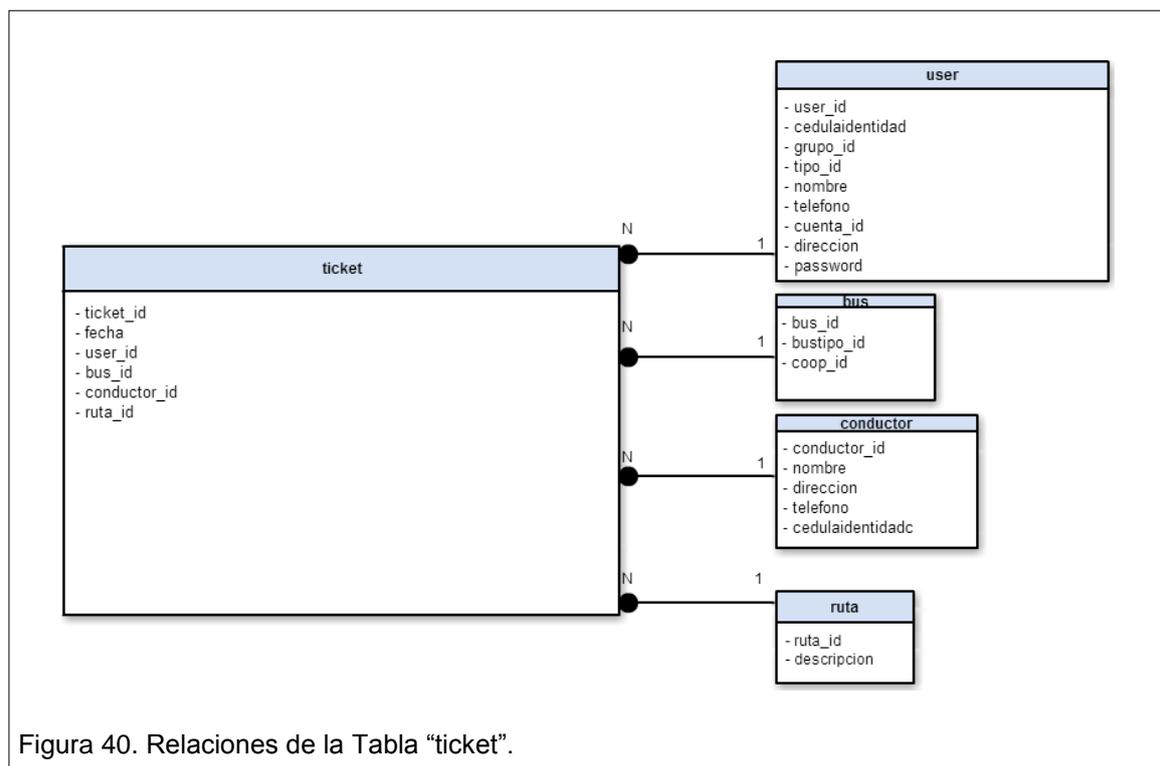


Figura 40. Relaciones de la Tabla "ticket".

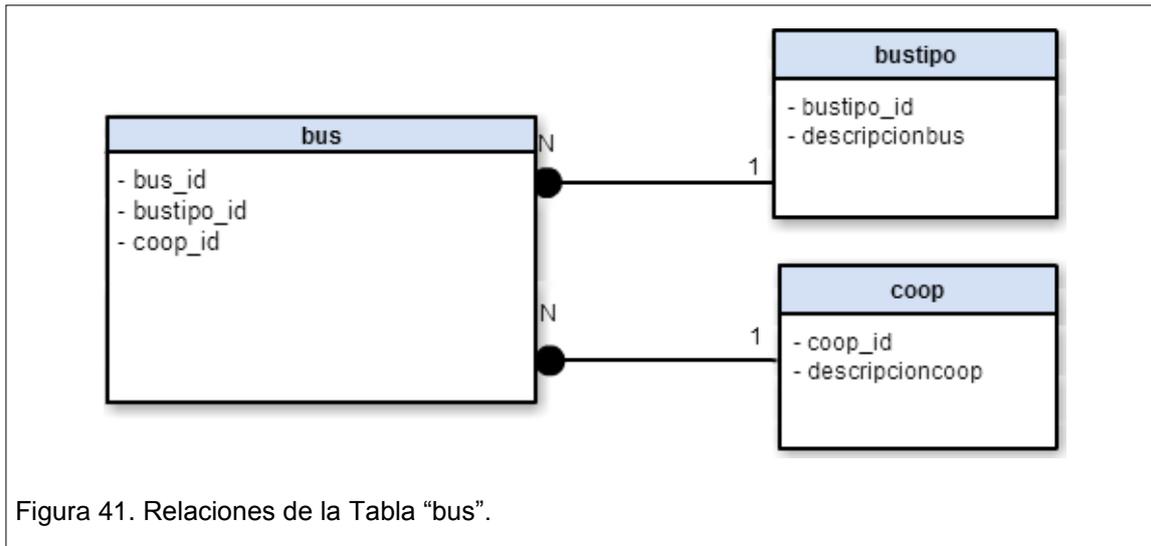


Figura 41. Relaciones de la Tabla "bus".

La relación entre tablas es de vital importancia, debido a que el sistema se regirá al comportamiento de los datos dentro de la base de datos. Es en este proceso de diseño en donde la lógica de negocio es definida (Figura 42).

4.1.5. Implementación – Esquema Final

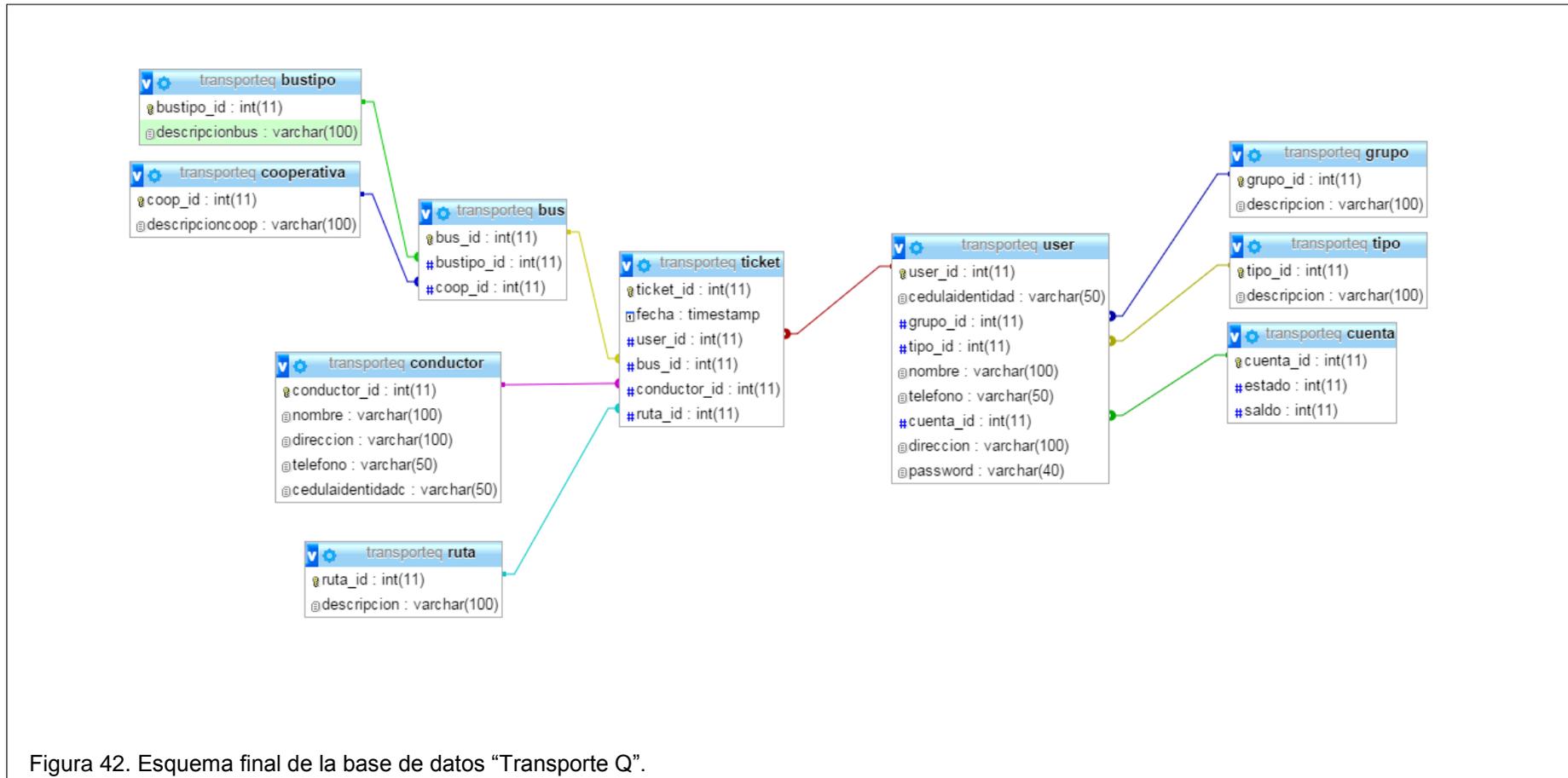


Figura 42. Esquema final de la base de datos “Transporte Q”.

4.2. Desarrollo del Aplicativo Web en ASP.net

4.2.1. Programación por Páginas

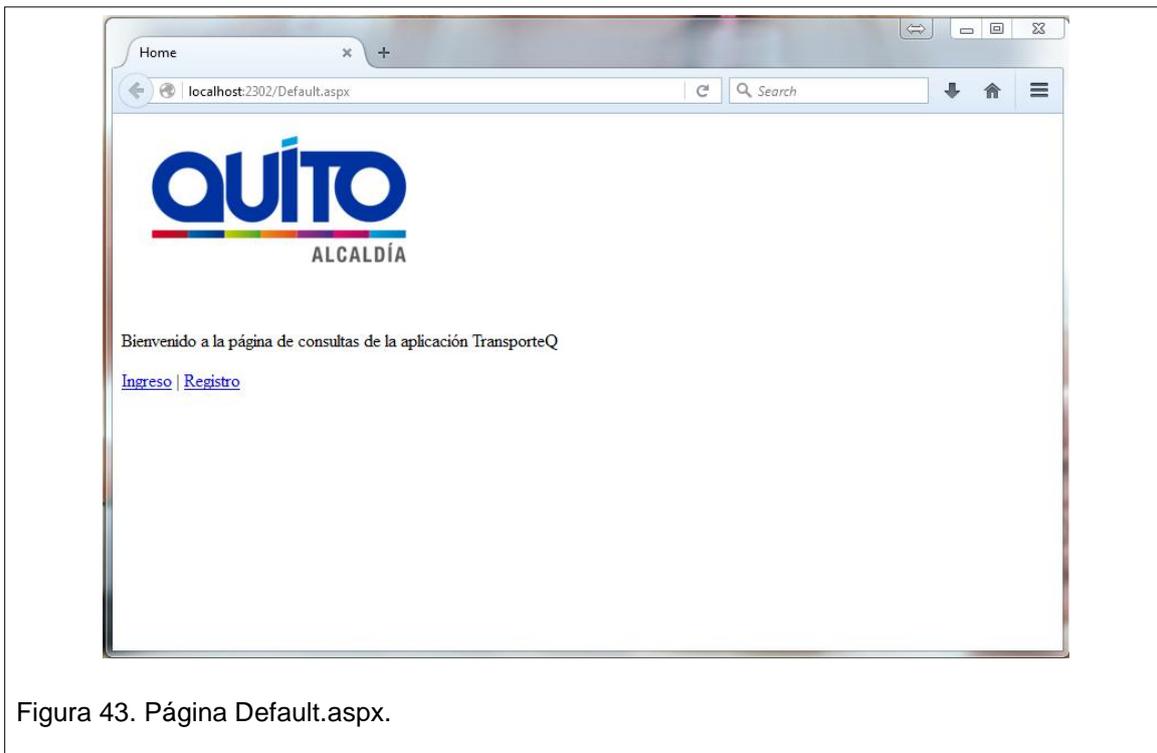


Figura 43. Página Default.aspx.

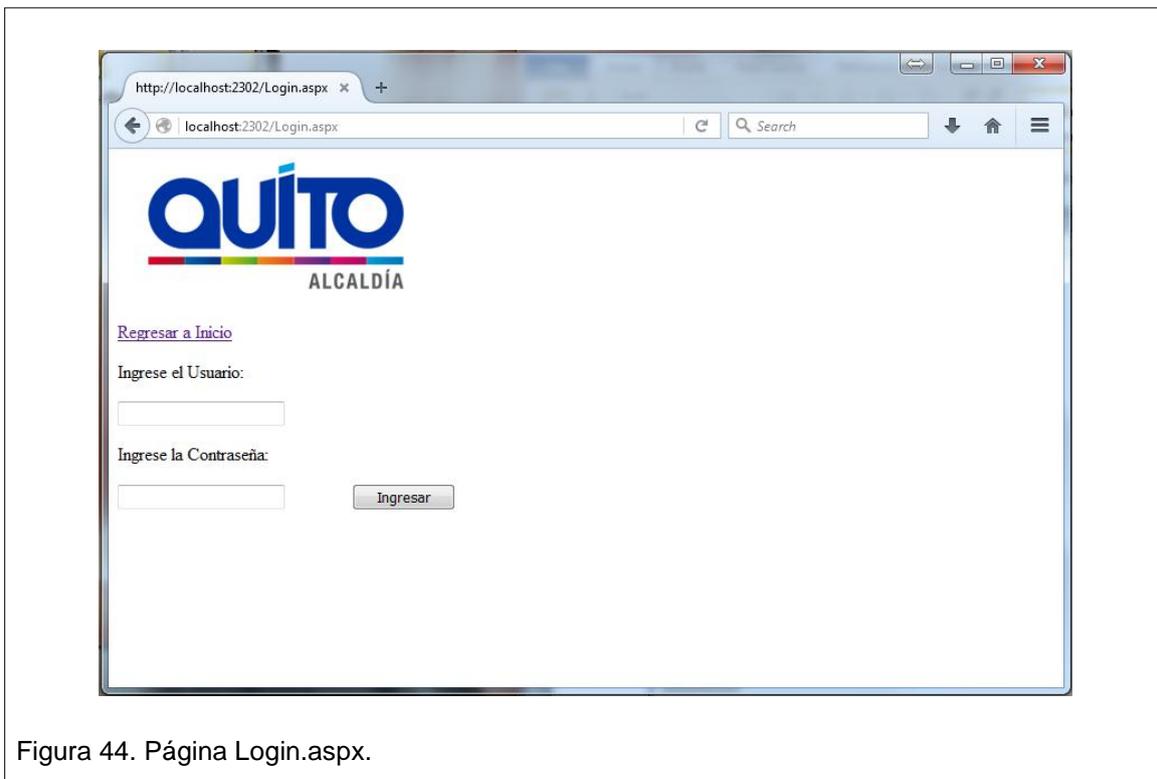


Figura 44. Página Login.aspx.

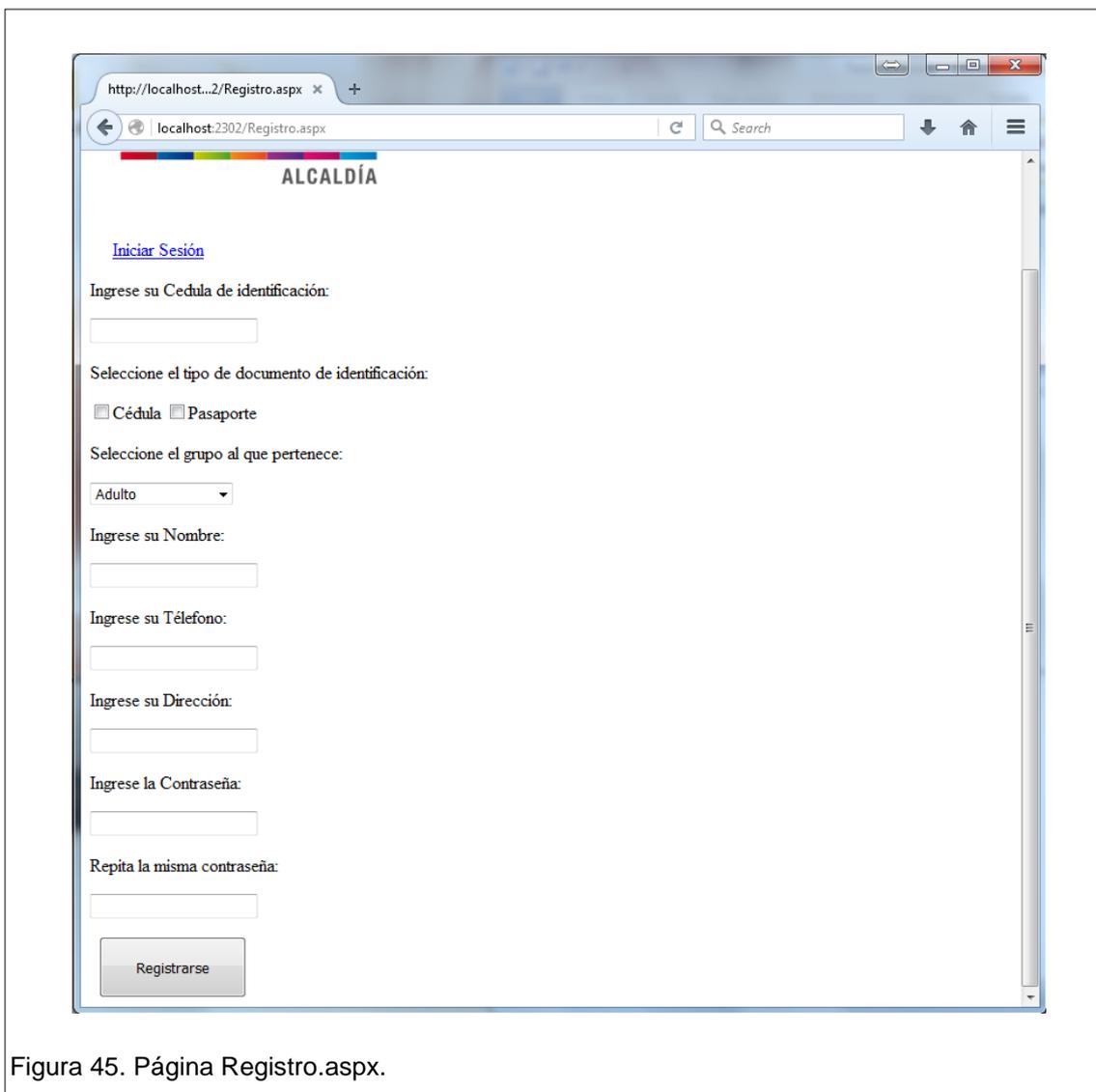


Figura 45. Página Registro.aspx.

Las páginas mostradas en la Figura 38 y 39 acceden directamente a la base de datos por medio de una cadena de conexión en donde se especifica el nombre de la base y los datos del usuario con los permisos necesarios para realizar los procesos de cambios en la información de la tabla.

```
String MyConnectionString =  
"Server=127.0.0.1;Database=NAME;Uid=USER;Pwd=PASSWORD";  
MySqlConnection connection = new MySqlConnection(MyConnectionString);  
connection.Open();
```

La codificación usada para las contraseñas es MD5, la cual es utilizada en las solicitudes de registro de contraseña y comparación para el inicio de sesión. Al utilizar la función que proporciona MySQL, la solicitud de codificación se la realiza en las líneas de comando a la base de datos:

```
cmd = connection.CreateCommand();  
cmd.CommandText = "SELECT nombre FROM `user` WHERE user_id =@ID AND password =  
md5(@Password)";
```

Se realiza la validación de los datos ingresados tanto en la página de “Inicio de Sesión: como la de “Registro” para usuarios nuevos, limitando el envío de formularios con campos vacíos, datos duplicados en la base e ingreso de variables con diferente tipo de formato. Se consideran dos niveles de filtrado para validación de datos, el primero directamente en la interfaz vinculando los campos a condicionantes “if” sin enviar ninguna información. Y la segunda enviando información a la base de datos con el fin de comparar los datos ingresados.

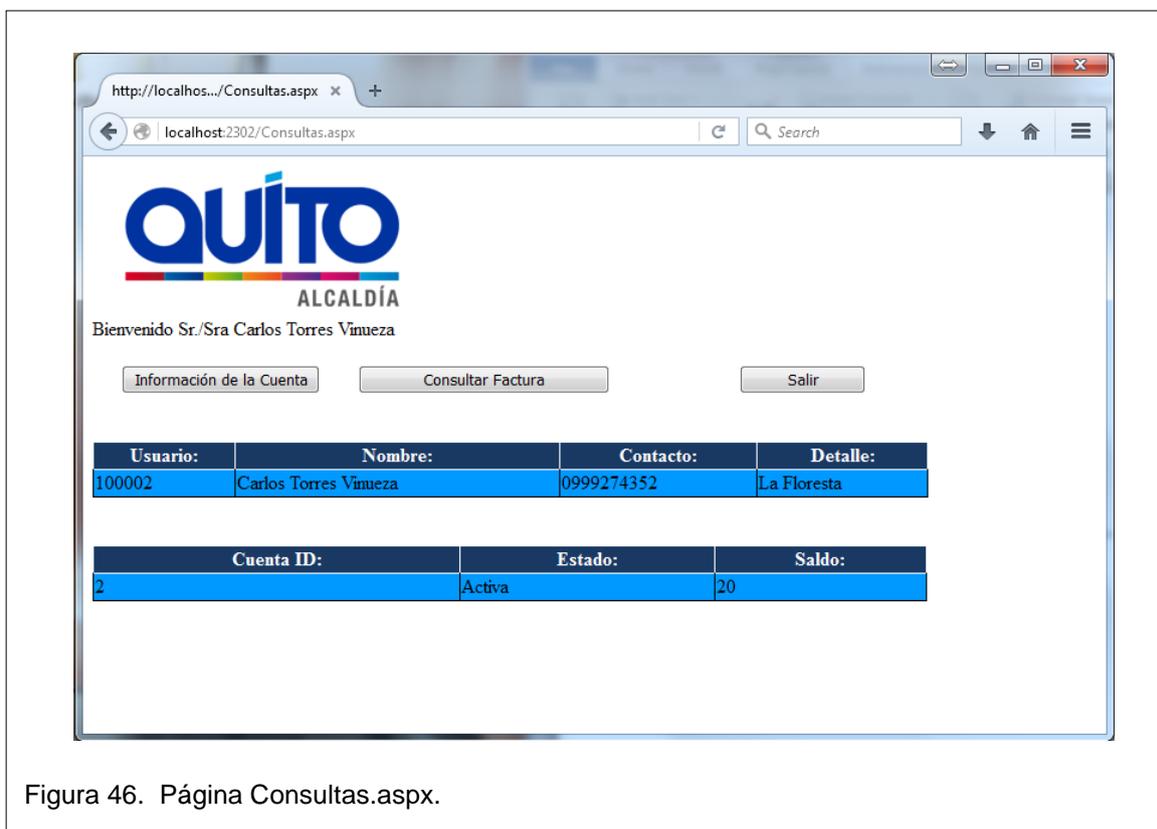


Figura 46. Página Consultas.aspx.

4.2.2. Prueba de Desempeño (WEB)

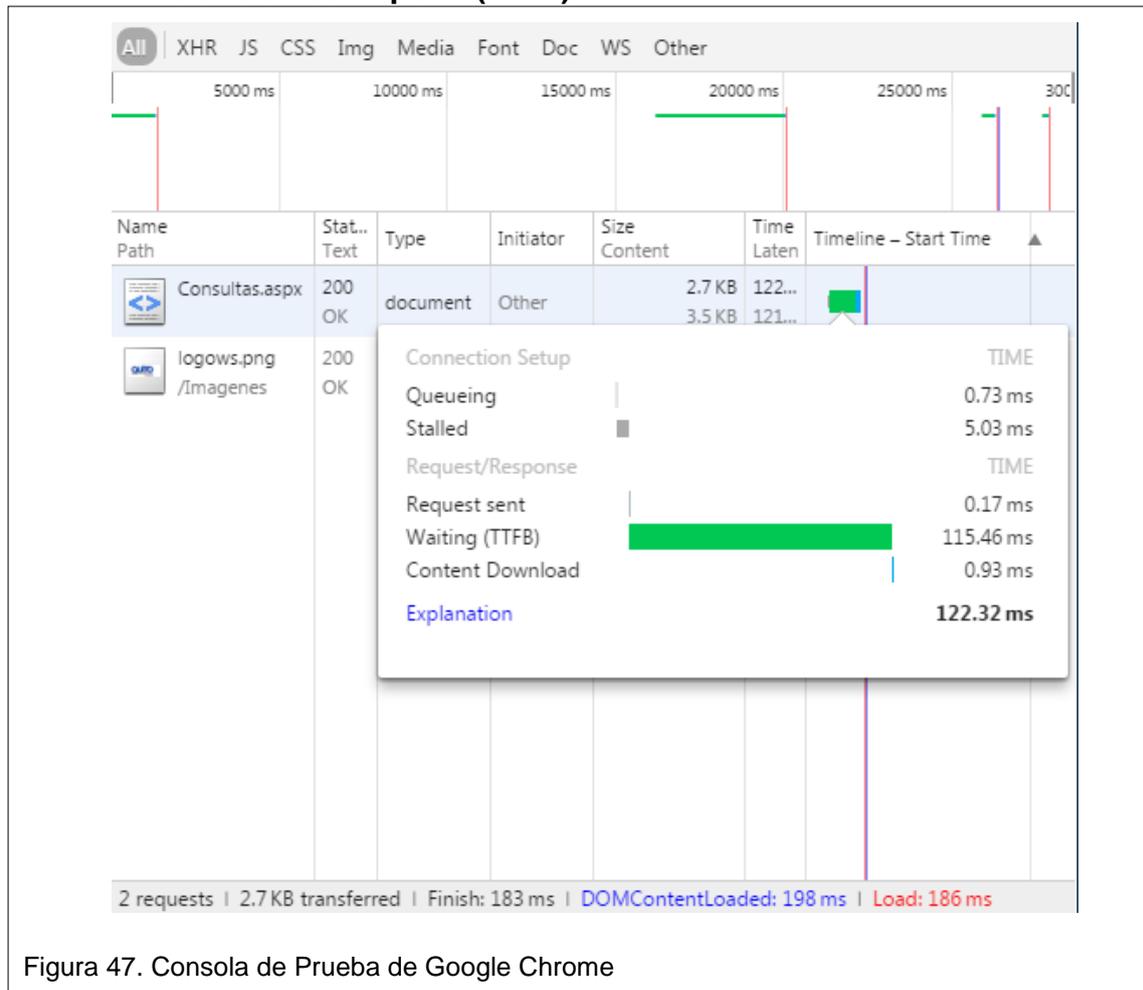


Figura 47. Consola de Prueba de Google Chrome

Los resultados obtenidos de la consola de pruebas del explorador Google Chrome nos muestra una respuesta de 122.32 mili segundos en la consulta realizada y 2.7 K de información transferidos.

- Página: Consultas.aspx
- Tiempo de respuesta total: 122.32 mili segundos
- Tamaño de paquetes descargados: 2.7 KB

4.3. Desarrollo del Aplicativo en Android

4.3.1. Interfaz Grafica

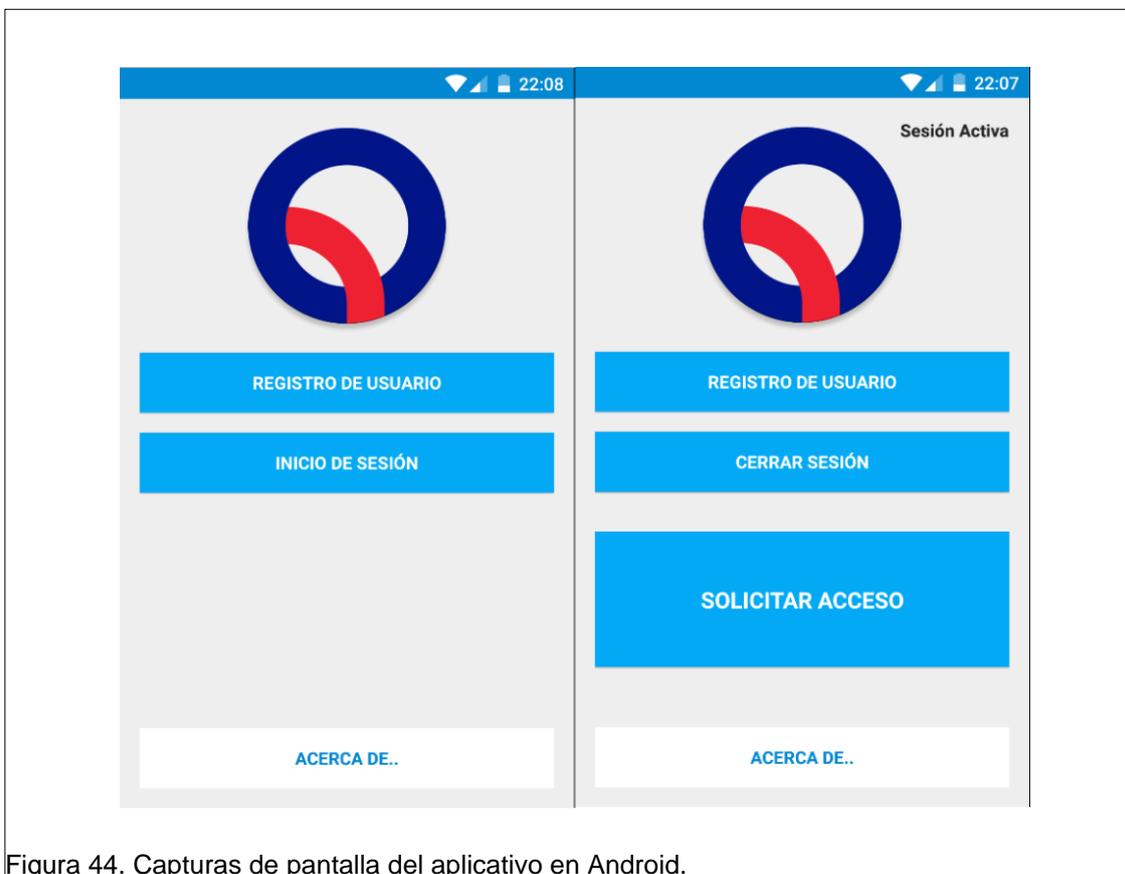


Figura 44. Capturas de pantalla del aplicativo en Android.

En la figura 44, podemos observar la pantalla inicial del aplicativo Android para usuarios y el cambio cuando se inicia sesión. Una vez iniciada la sesión el botón “Solicitar Acceso” se activa y tiene la capacidad de generar el ticket virtual.

Layouts:

- Activity_main.xml
- Activity_ingreso.xml
- Activity_registrar.xml
- Activity_solicitar_ticket.xml

En Android, la conexión a la base de datos se la realizara mediante web services. Del lado del dispositivo se utilizará JSON, el cual es un formato de intercambio de datos de lectura simple para los dispositivos vinculados.

Del lado del servidor, se utilizara PHP para realizar el acceso a la base de datos y el manejo de los mensajes a JSON.

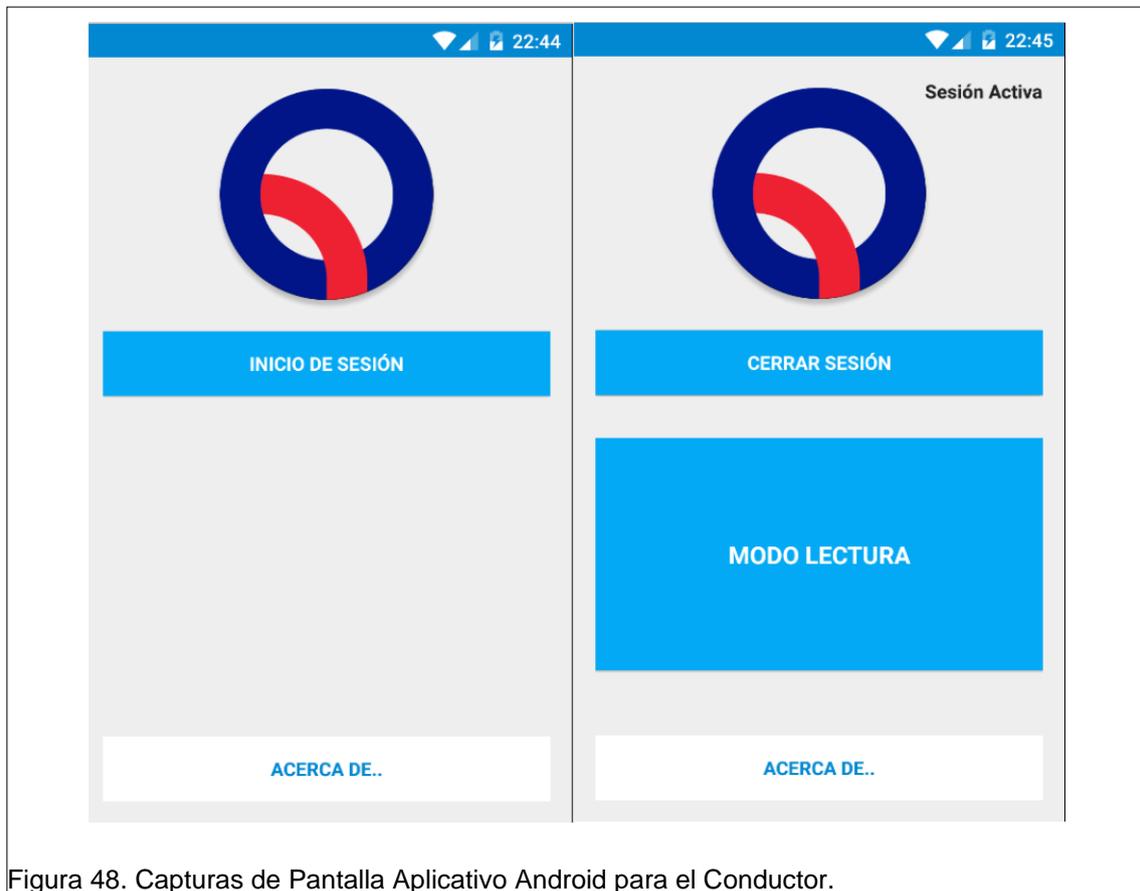


Figura 48. Capturas de Pantalla Aplicativo Android para el Conductor.

Como podemos ver en la Figura 48, el aplicativo que emula al prototipo únicamente permite al conductor iniciar sesión y activar el modo lectura.

De igual forma, la interfaz nos comunicara por medio de un layer en la parte superior derecha si es que en el dispositivo se encuentra una sesión activa.

4.3.2. Programación de Mensajes NDEF

Para realizar la comunicación de datos a través de una conexión NFC se debe estructurar los mensajes en formato NDEF. Se establece dicho mensaje en un NdefRecord que representa las reglas para el envío del mensaje.

```
String stringOut = Usuario.getText().toString();
byte[] bytesOut = stringOut.getBytes();

NdefRecord ndefRecordOut = new NdefRecord(
    NdefRecord.TNF_MIME_MEDIA,
    "text/plain".getBytes(),
    new byte[] {},
    bytesOut);

NdefMessage ndefMessageOut = new NdefMessage(ndefRecordOut);
return ndefMessageOut;
```

Para que la actividad pueda identificar un nuevo mensaje NDEF se debe configurar el archivo AndroidManifest.xml con los siguientes parámetros en la actividad:

```
<activity android:name=".SolicitarTicket">
    <intent-filter>
        <action android:name="android.nfc.action.NDEF_DISCOVERED" />
        <data android:mimeType="text/plain" />
    </intent-filter>
</activity>
```

4.3.3. Intents para uso de la Comunicación P2P

Se configuran los siguientes Intents en el método onCreate de la actividad "SolicitarTicket" en orden de definir el funcionamiento de la comunicación P2P entre ambos dispositivos:

- mNfcPendingIntent
- nDefDetected
- NfcAdapter.ACTION_NDEF_DISCOVERED

El siguiente paso es habilitar el envío de mensajes NDEF en el método onResume de la actividad.

```
private void enableNdefExchangeMode() {
    mNfcAdapter.enableForegroundNdefPush(NfcAddFriendsActivity.this,
        NfcUtils.createNdefMessage(mUserId));
    mNfcAdapter.enableForegroundDispatch(this, mNfcPendingIntent,
        mNdefExchangeFilters, null);
}
```

4.3.4. Envío de parámetros a través de JSON

Una vez recuperado los datos que se requieren enviar al servicio web para que sea almacenado en la base de datos, se realiza la creación de una lista de parámetros que será enviado en un objeto del tipo JSONObject.

```
List<NameValuePair> params = new ArrayList<NameValuePair>();
params.add(new BasicNameValuePair("username", username));
params.add(new BasicNameValuePair("password", password));

JSONObject json = jsonParser.makeHttpRequest(
    LOGIN_URL, "POST", params);
```

4.3.5. Configuración de Servicios Web en PHP

Para poder configurar servicios web inicialmente se debe instalar un servidor web en la máquina que se utilizará para emular el servidor de la base de datos. Para ello se consideró XAMMP, el cual es una distribución de Apache y que ofrece el módulo de Apache y MySQL en la misma herramienta.

Una vez levantado el servidor web, se configuran los servicios web en un bloc de notas. Dichos archivos se guardaran con la extensión “.php” y estarán ubicados en la ruta:

```
C:\xampp\htdocs\webservice
```

Se define el primer archivo el responsable de realizar la conexión a la base de datos. Este archivo tendrá el nombre de “config.inc.php”.

Se definen parámetros de conexión a la base de datos:

```
$host = "localhost";

$dbname = "transporteq";
```

```
$username = "admin";
```

```
$password = "PASSWORD";
```

Al tener almacenados los archivos en el mismo equipo que contiene la base de datos, se hace referencia al host como localhost en reemplazo de la ip de acceso a la base.

A continuación se procede con la conexión:

```
$db = new PDO("mysql:host={$host};dbname={$dbname};charset=utf8",  
$username, $password);
```

```
session_start();
```

Este archivo configurara el acceso a la base de datos de todos los servicios web que se ejecutaran en el equipo.

- Config.inc.php
- Login.php
- Register.php
- Ticket.php

4.3.6. Pruebas de Rendimiento (GameBench Ver. 3.2.2)

Se toman en consideración las acciones capturadas en la transferencia de datos a través de NFC de los aplicativos en Android (App Usuario y App Prototipo). Se utilizara el aplicativo GameBench para realizar la captura de la información de uso de la CPU y GPU de ambos dispositivos mientras corren los aplicativos.

Se descarga el aplicativo en ambos dispositivos desde el siguiente vínculo:

<https://play.google.com/store/apps/details?id=com.gamebench.metricscollector>

Una vez instalado el aplicativo GameBench se procede a activar las funciones de monitoreo, para lo cual se necesita descargar el servicio de GameBench en el computador desde esta ruta:

<https://www.gamebench.net/gamebench-setup>

Se conectan los dispositivos al computador y se ejecuta el servicio. Una vez mostrado el mensaje de confirmación se puede empezar a realizar las pruebas de rendimiento.

Los dispositivos móviles que serán puestos a prueba tienen chips NFC y corren bajo el sistema operativo Android. Ambos dispositivos fueron probados para cumplir con los requerimientos mínimos necesarios para ejecutar las funciones programadas en este proyecto.



Figura 49. Comparación física de dispositivos usados en la prueba.

Tomado de PhoneArena, s.f.

Tabla 18. Comparación de aspectos técnicos.

	Sony Xperia Z3	Samsung Galaxy S III
System Chip	Qualcomm Snapdragon 801 8974-AC	Exynos 4 Quad 4412
CPU	Quad-core, 2500 MHz, Krait 400	Quad-core, 1400 MHz, ARM Cortex-A9
GPU	Adreno 330	ARM Mali-400/MP4

Prueba de Uso de CPU

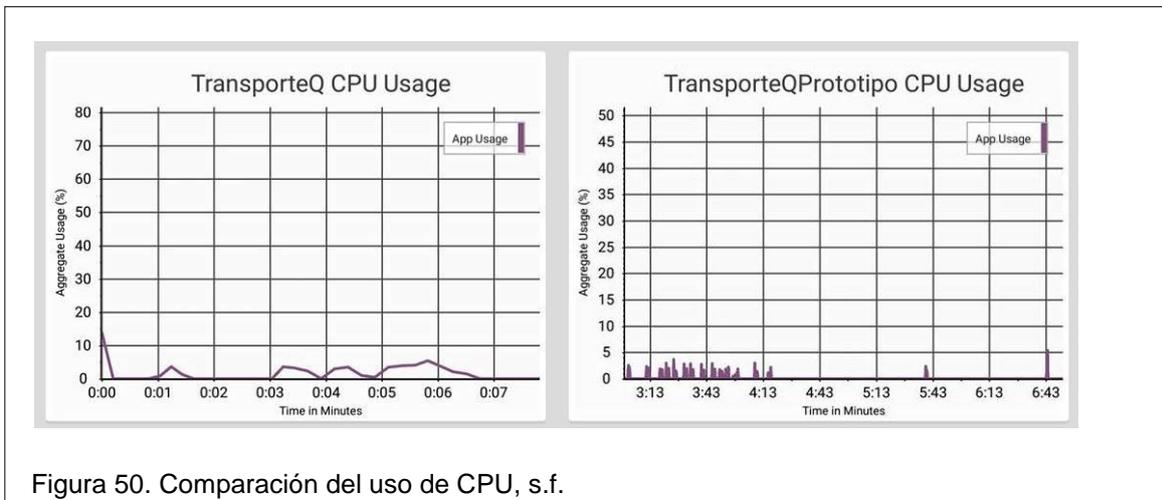


Figura 50. Comparación del uso de CPU, s.f.

A la izquierda consumo de la CPU mientras se ejecuta el aplicativo “TransporteQ”. Aplicativo diseñado para el Usuario final. A la derecha el consumo realizado por el aplicativo “TransporteQPrototipo”. Aplicativo diseñado para funcionar en el prototipo.

Prueba de Uso de GPU

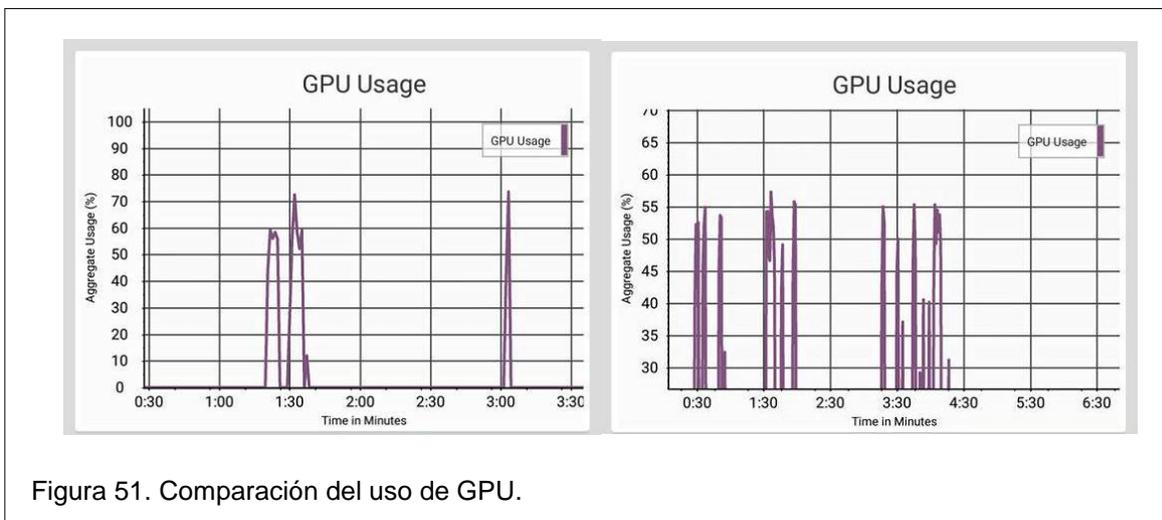


Figura 51. Comparación del uso de GPU.

5. CONCLUSIONES Y RECOMENDACIONES.

5.1 Conclusiones

Es imprescindible el análisis de los requerimientos como etapa inicial de un proyecto, y más aún cuando vincula la integración de varios aplicativos funcionando con la misma base de datos. En la etapa inicial es donde se deben registrar cada una de las necesidades del usuario y traducirlas en variable de entrada y salida, ya que una vez definido correctamente cada uno de estos valores, la creación de la base de datos será el inicio del desarrollo e implementación de un proyecto. Esto ayuda a la estabilidad del software creado, evitando cambios inoportunos que retrasen el cronograma propuesto.

El uso de la metodología OODHM o Metodología de Diseño de Hipermedia Orientada a Objetos para el desarrollo del sistema propuesto en conjunto con el patrón de diseño MVC o Modelo Vista Controlador brindaron una estructura sencilla para el diseño e implementación de los aplicativos desarrollados.

El uso de la tecnología NFC es una excelente herramienta para integrar en sistemas que requieren la certificación del usuario. Su funcionamiento ligado a la cercanía de los dispositivos (menor a 20cm) obliga la presencia física del dispositivo, lo que representa interacción virtual mediante acciones físicas. Al trabajar en un rango tan limitado en vez de restarle propiedades a su funcionamiento, lo enfoca en el ámbito de las tecnologías de seguridad.

Gracias al surgimiento del NFC Forum, cada día más dispositivos integran esta tecnología nacida a partir del Rfid. Debido a que la tecnología Rfid presentaba tantas aplicaciones de uso cotidiano, las empresas productoras de teléfonos celulares empezaron a crear adaptaciones de esta tecnología en cada uno de sus dispositivos. Pero gracias al NFC Forum, la tecnología Rfid tiene una nueva dirección y se perfila a ser un elemento indispensable en los teléfonos celulares.

En nuestro entorno se pueden identificar varias tecnologías que tienen un funcionamiento constante. Los celulares cada vez presentan mejores servicios y están a nuestra disposición. Además, funcionan a través de una red de telefonía que ahora soporta altas velocidades y se encuentra ligada a la densidad poblacional de un sector geográfico, lo cual permite el desarrollo de sistemas que se adapten a las necesidades de los usuarios y tengan una comunicación constante con ellos.

La herramienta de desarrollo Android Studio ofrece el monitoreo en tiempo real de la ejecución de las aplicaciones que se crean en ella. Es posible el estudio de los recursos de red, memoria, procesamiento y almacenamiento que se ocupa y se pueden crear registros con el objetivo de mejorar las aplicaciones o identificar posibles desperdicios de recursos. Estas propiedades no se limitan al monitoreo en la ejecución de emuladores, sino que también es utilizable en el proceso de depuración a través de USB, en donde la prueba se realiza directamente en un dispositivo móvil.

En el desarrollo de este proyecto la depuración mediante USB resulta muy ventajosa, debido a que al trabajar con un chip de proximidad como lo es NFC, es imprescindible realizar pruebas constantes de funcionamiento. Existen emuladores que funcionan en conjunto con el emulador de dispositivos de Android Studio pero no ofrecen resultados fidedignos e inclusive se pueden omitir errores debido al modelo o marca de los chips NFC con los que se va a trabajar.

Una aplicación Web es una herramienta muy útil para realizar el despliegue de información. Se adapta con facilidad a la base de datos y su interfaz puede ser programada para que sea amable con los usuarios. El registro de datos desde una página web puede resultar más cómoda para un usuario que al realizarlo mediante una interfaz en un dispositivo celular, debido a las prestaciones que ofrece un monitor frente a la pantalla o display de un teléfono celular.

5.2 Recomendaciones

Se recomienda la contratación de un Hosting tanto para la aplicación Web como para la base de datos, ya que de esta forma se brinda una mejor disponibilidad del aplicativo, funcionando en todo momento sin depender del equipo o servidor donde trabaja. Esta medida no solo mejora la disponibilidad del aplicativo, sino que también lo protege de ataques y lo hace accesible a los diferentes dispositivos que pueden conectarse a la red de internet.

Se recomienda el uso de Android Studio en el desarrollo inicial de aplicaciones debido al alto número de documentación que existe sobre el manejo de APIs y clases. En la página Android Development se puede encontrar la documentación necesaria para el desarrollo en esta herramienta, además desde otras fuentes existen tutoriales e información para la implementación de aplicativos. Pese a que la mayoría de la información aún se encuentra en inglés es muy fácil de entender gracias al Android Studio.

Es recomendable en la programación de aplicativos para Android, especificar desde el inicio la versión de API a la cual será dirigida. Esto debido a que dependiendo de los recursos de software y hardware que requiera el aplicativo, se limitara el acceso a dichos dispositivos que no cuenten con los requerimientos necesarios. Ya sea uso de internet, acceso a la cámara, mapas, o como en el caso de este proyecto al chip NFC del dispositivo.

REFERENCIAS

- Berzal F. Cortijo F. Cubero J. (2005). *Desarrollo Profesional de Aplicaciones Web con ASP.NET*. Recuperado el 13 de abril de 2015 de <http://elvex.ugr.es/decsai/csharp/aspnet.html>
- Campa A. (2011). *Desarrollo de una Aplicación de Pago a Través de la Tecnología NFC*. Madrid, España: Universidad Carlos III de Madrid.
- Chavarría D. (2011). *Tecnología de Comunicación de Campo Cercano (NFC) y sus Aplicaciones*. San José, Costa Rica: Universidad de Costa Rica.
- Ecuador en Cifras. (s.f.). *Resultados Principales Tecnologías de la Información y Comunicación (TIC's) 2013*. Recuperado el 20 de enero de 2015 de <http://www.ecuadorencifras.gob.ec/tecnologias-de-la-informacion-y-comunicacion-tic/>
- Flores S. (2015). *Curso: PHP con MySQL*. Recuperado el 14 de enero de 2016 de <http://www.miholamundo.com/2015/12/php-con-mysql.html>
- Google Inc. (2016). *Beaming NDEF Messages to Other Devices*. Recuperado el 13 de enero de 2016 de <http://developer.android.com/intl/es/guide/topics/connectivity/nfc/nfc.html#p2p>
- Google. (2016). *NFC Basics*. Recuperado el 11 de enero de 2016 de <http://developer.android.com/intl/es/guide/topics/connectivity/nfc/nfc.html>
- Hébuterne S. Pérochon S. (2014). *Android: Guía de Desarrollo de Aplicaciones para Smartphones y Tabletas*. (2ª. Ed.). Barcelona, España: Ediciones ENI.
- IEEE Org. (2008). *Especificaciones de Requisitos según el estándar de IEEE 830 - 1998*. IEEE.
- Igoe T. Coleman D. Jepson B. (2014). *Beginning NFC. Near Field Communication with Arduino, Android & Phonegap*. (1ra. ed.). Estados Unidos: O'Really Media Inc.
- Junta de Andalucía. (2013). *Atributos de los Requisitos*. Recuperado el 12 de octubre de 2015 de <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/409>
- Méndez, M. (2011). *Evaluación de la Factibilidad de Implementar la Tecnología NFC en el Ecuador*. Cuenca, Ecuador: Universidad Tecnológica Israel.

- NFC FORUM. (2016). *NFC Forum Technical Specifications*. Recuperado el 18 de agosto de 2015 de <http://nfc-forum.org/our-work/specifications-and-application-documents/specifications/nfc-forum-technical-specifications/>
- NFC Forum. (2016). *NFC Product Showcase*. Recuperado el 18 de agosto de 2015 de <http://nfc-forum.org/nfc-product-showcase/>
- Phone Arena. (2016). *Comparison: Sony Xperia Z3 vs Samsung Galaxy S III*. Recuperado el 20 de enero de 2016 de <http://www.phonearena.com/phones/compare/Sony-Xperia-Z3,Samsung-Galaxy-S-III/phones/8764,6330>
- Pressman R. (2010). *Software Engineering: A Practitioner's Approach*. (7a. ed.). New York. Estados Unidos: The McGraw-Hill Companies.
- Quito Adventure. Cando, E. (s.f.). *Sistemas Integrados de Transporte en Quito*. Recuperado el 21 de enero de 2015 de <http://www.quitoadventure.com/espanol/informacion-ecuador/transporte-ecuador/quito/quito-transporte-integrado.html>
- RapidNFC Ltd. (2016). *NFC Enabled Phones and Tablets*. Recuperado el 12 de octubre de 2015 de http://rapidnfc.com/nfc_enabled_phones