

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Ingeniería en Sistemas Computacionales

**DESARROLLO DE UNA APLICACIÓN MÓVIL QUE OPTIMICE EL PROCESO DE
COMPRA DE BOLETOS DE BUSES INTERPROVINCIALES DE LA
COOPERATIVA DE TRANSPORTE “FLOTA IMBABURA” EN LA PROVINCIA DE
IMBABURA**

Trabajo de grado previo a la obtención del título de Ingeniero en Sistemas
Computacionales

Autor:

RUALES ROSERO LUIS CARLOS.

Director:

MSc. CARANQUI SÁNCHEZ VICTOR MANUEL.

Ibarra – 2019



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	0401517487
APELLIDOS Y NOMBRES:	RUALES ROSERO LUIS CARLOS
DIRECCIÓN:	AV. ATAHUALPA Y HERNÁN GONZÁLES DE SAA
EMAIL:	lcrualesr@utn.edu.ec, lucasruales@gmail.com
TELÉFONO FIJO:	-----
TELÉFONO MÓVIL:	0986863854

DATOS DE LA OBRA	
TÍTULO:	0401517487
AUTOR (ES):	RUALES ROSERO LUIS CARLOS
FECHA: DD/MM/AAAA	24/04/2019
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
ASESOR /DIRECTOR:	MSc. CARANQUI SÁNCHEZ VICTOR MANUEL

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 24 días del mes de abril de 2019

EL AUTOR:

.....
RUALES ROSERO LUIS CARLOS



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DIRECTOR DE TESIS

Por medio del presente yo MSc. Caranqui Sánchez Víctor Manuel, certifico que el trabajo de grado “DESARROLLO DE UNA APLICACIÓN MÓVIL QUE OPTIMICE EL PROCESO DE COMPRA DE BOLETOS DE BUSES INTERPROVINCIALES DE LA COOPERATIVA DE TRANSPORTE ‘FLOTA IMBABURA’ EN LA PROVINCIA DE IMBABURA”, ha sido desarrollado en su totalidad por la Sr. Luis Carlos Ruales Rosero, portador de cédula de identidad Nro. 040151748-7.

Es todo cuanto puedo certificar en honor a la verdad.

MSc. Víctor Caranqui.
DIRECTOR



Abril 2019

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

ACTA DE ENTREGA RECEPCIÓN

TEMA: Venta de boletos de buses interprovinciales desde una aplicación móvil.

En las instalaciones de la empresa BUSHIDO SOFTWARE STUDIOS CIA. LTDA., de la ciudad de Ibarra, el día 23 de abril de 2019, la Carrera de Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte, por medio del trabajo de titulación “DESARROLLO DE UNA APLICACIÓN MÓVIL QUE OPTIMICE EL PROCESO DE COMPRA DE BOLETOS DE BUSES INTERPROVINCIALES DE LA COOPERATIVA DE TRANSPORTE ‘FLOTA IMBABURA’ EN LA PROVINCIA DE IMBABURA”, hace la entrega de la aplicación móvil “Boleto Express”, al Dr. Paul Boada, Gerente General de la empresa BUSHIDO SOFTWARE STUDIOS CIA. LTDA.

El desarrollo de la aplicación móvil lo realizó el Sr. Luis Carlos Ruales Rosero con C.I 0401517487 bajo la tutoría del MSc. Víctor Caranqui, conforme a los requerimientos planteados por el Dr. Paúl Boada.

A continuación, se detalla los productos entregados:

- Aplicación móvil de venta de boletos (Código Fuente)
 - Visualización y selección de ruta y fecha de viaje.
 - Selección de horario de la ruta de viaje.
 - Selección de asientos disponibles.
 - Visualización de detalle de compra del boleto
 - Envío de correo electrónico de verificación de compra
- Pruebas contra especificaciones ESRE
- Manuales de Usuario
- Manuales Técnicos
- Capacitación al personal encargado

Atentamente, ENTREGA CONFORME

Sr. Luis Carlos Ruales
TESISTA
UNIVERSIDAD TÉCNICA DEL NORTE

Ing. Pedro Granda
Coord. Carrera Ingeniería en
Sistemas Computacionales

MSc. Víctor Caranqui
DOCENTE TUTOR
UNIVERSIDAD TÉCNICA DEL NORTE

Dr. Paul Boada
GERENTE GENERAL BUSHIDO
SOFTWARE STUDIOS

Dedicatoria

Dedico mi trabajo de grado, la carrera de ingeniería y mi vida entera a una persona que merece más mérito que yo en éste arduo camino, una persona a quién debo mi razón de ser.

¡Dedicado para usted Mami!

Agradecimientos

Quiero expresar mi gratitud a Dios, quien con su bendición llena siempre mi vida y la de toda mi familia.

Mi profundo agradecimiento a mis abuelitos, Carlitos y Rosita; quienes son pilar importante en esta meta alcanzada y en mi vida entera. A mi hermano Jhon, mi padre Oscar, mis tíos(as) y primos(as); por el apoyo que de una u otra manera contribuyeron a que pudiera alcanzar esta meta.

Al amor de mi vida la Lic. Elitha Chacón, por el apoyo y la preocupación constante de que todo lo que me propongo, salga bien; y, más aún, por su amor sincero e incondicional.

Un agradecimiento a mis grandes amigos y equipo de trabajo en Bushido Software Studios; al Dr. Paul Boada, Kevin Rodríguez, Richard Tarupí y Jorge Cuasés, quienes fueron parte fundamental en este trabajo de grado.

Finalmente quiero expresar mi gratitud al MSc. Víctor Caranqui, principal colaborador durante todo este proceso, quien con su dirección, conocimiento, enseñanza y colaboración permitió el desarrollo de este trabajo.

¡GRACIAS TOTALES!

Tabla de Contenidos

INTRODUCCIÓN	1
TEMA	1
PROBLEMA	1
ANTECEDENTES	1
SITUACIÓN ACTUAL	1
PROSPECTIVA	2
PLANTEAMIENTO DEL PROBLEMA	3
OBJETIVOS	5
OBJETIVO GENERAL	5
OBJETIVOS ESPECÍFICOS	5
ALCANCE	5
JUSTIFICACIÓN	6
CONTEXTO	7
CAPÍTULO 1	9
Marco teórico	9
1.1. Introducción a las aplicaciones móviles	9
1.1.1. ¿Qué es una aplicación móvil?	9
1.1.2. Tipos de aplicaciones móviles	10
1.2. Herramientas de desarrollo para aplicaciones móviles	11
1.2.1. Lenguajes de Programación	12
1.2.2. Plataformas de desarrollo para aplicaciones móviles	12
1.3. Sistema operativo Android	13
1.3.1. Características principales de Android	13
1.3.2. Arquitectura del sistema operativo Android	14
1.3.3. Kernel Linux	15
1.3.4. Runtime de Android	15
1.3.5. Bibliotecas o librerías	15
1.3.6. Framework de la aplicación	15
1.3.7. Aplicaciones	16
1.4. Optimización de procesos	16
1.4.1. ¿Qué es un proceso?	16
1.4.2. Optimización de procesos	17
1.5. Servidor de base de datos	19

1.5.1. Servidores de base de datos.....	19
1.5.2. Sistemas gestores de base de datos.....	20
1.6. Metodologías de trabajo.....	23
1.6.1. Marco de trabajo.....	23
1.6.2. Metodologías de desarrollo de software.....	23
1.6.3. SCRUM como marco ágil de trabajo.....	24
CAPÍTULO 2.....	28
Desarrollo del ciclo de vida de la aplicación.....	28
2.1. Toma de requerimientos.....	28
2.2. Planificación.....	28
2.2.1. Definición de roles y tareas.....	28
2.2.2. Definición de equipo de trabajo.....	29
2.2.3. Descripción de historias de usuario.....	30
2.3. Desarrollo.....	33
2.3.1. Arquitectura del sistema.....	33
2.3.1. Desarrollo de las historias de usuario.....	34
2.4. Pruebas.....	43
2.4.1. Pruebas contra Especificación (ESRE).....	43
CAPÍTULO 3.....	45
Análisis e interpretación de resultados.....	45
3.1. Hipótesis del trabajo.....	45
3.2. Obtención de datos.....	45
3.3. Manejo de datos.....	46
3.3.1. Frecuencia de viaje.....	46
3.3.2. Compra de boleto en condiciones normales.....	47
3.2.3. Compra de boleto en feriados.....	47
3.2.4. Acogida de la aplicación.....	48
CONCLUSIONES Y RECOMENDACIONES.....	50
CONCLUSIONES.....	50
RECOMENDACIONES.....	50
REFERENCIAS.....	51
ANEXOS.....	54

Índice de Figuras

Fig. 1 - Proceso de compra en línea a través de una pasarela de pago. (Fuente: (Urbano, 2016)).....	2
Fig. 2 - Diagrama de Ishikawa. (Fuente: Propia).....	4
Fig. 3 - Diagrama de la arquitectura del sistema (Fuente: Propia).....	6
Fig. 4 - Tienda de aplicaciones según Sistema Operativo (Fuente: Propia)	9
Fig. 5 - Representación de las aplicaciones móviles. (Fuente: Carballar, 2012).....	11
Fig. 6 - Sistema operativo Android en automóviles (Fuente: (LaPrensa, 2015)).....	13
Fig. 7 - Arquitectura Android (Fuente: (Tatés Lenin, 2018)).....	14
Fig. 8 - Actividades de un proceso (Fuente: Propia).....	17
Fig. 9 - Diagrama de interconexión de servidores de base de datos (Fuente: (Stein Steve, 2017)	19
Fig. 10 - Proceso del marco de trabajo ágil SCRUM (Fuente: (Pacheco, 2017a)	25
Fig. 11 - Arquitectura de la aplicación móvil (Fuente: Propia).....	34
Fig. 12 – Gráfico del modelamiento de la base de datos en Power Designer (Fuente: Propia).....	34
Fig. 13 - Estructura de clases del Servicio Web (Fuente: Propia).....	35
Fig. 14 - Consola de administración de la base de datos en Amazon Web Service (Fuente: Propia).....	35
Fig. 15 – Consola de administración del servicio web en Heroku (Fuente: Propia)	36
Fig. 16 - Pantalla de rutas. (Fuente: Propia).....	36
Fig. 17 - Pantalla de rutas con alerta de datos incompletos. (Fuente: Propia).....	37
Fig. 18 – Asistente de compra de boletos-Sección de rutas. (Fuente: Propia)	38
Fig. 19 - Asistente de compra de boletos-Sección de horarios. (Fuente: Propia)	39
Fig. 20 - Asistente de compra de boletos-Sección de asientos con alerta. (Fuente: Propia).....	40
Fig. 21 - Asistente de compra de boletos-Sección de asientos. (Fuente: Propia).....	40
Fig. 22 - Asistente de compra de boletos-Sección de asientos. (Fuente: Propia).....	41
Fig. 23 - Asistente de compra de boletos-Sección de asientos. (Fuente: Propia).....	42
Fig. 24 - Asistente de compra de boletos-Sección de asientos. (Fuente: Propia).....	43
Fig. 25 – Representación gráfica de frecuencias de viaje de la muestra (Fuente: Propia)	46
Fig. 26 - Representación gráfica de tiempos estimados de compra de boletos en condiciones normales (Fuente: propia)	47
Fig. 27 - Representación gráfica de tiempos estimados de compra de boletos en feriados (Fuente: propia).....	48

Fig. 28 - Representación gráfica de aceptación y uso de la aplicación (Fuente: propia)	49
Fig. 29 - Entrega de código fuente, aplicación móvil y firma de ACTA ENTREGA RECEPCIÓN	54

Índice de Cuadros

Tabla 1 - Comparativa de proyectos relacionados. (Fuente: Propia).....	8
Tabla 2 - Definición de roles y tareas. (Fuente: Propia).....	29
Tabla 3 - Definición de equipo de trabajo. (Fuente: Propia).....	29
Tabla 4 - Historia de usuario 1 (Fuente: Propia).....	30
Tabla 5 - Historia de usuario 2 (Fuente: Propia).....	30
Tabla 6 - Historia de usuario 3 (Fuente: Propia).....	31
Tabla 7 - Historia de usuario 4 (Fuente: Propia).....	31
Tabla 8 - Historia de usuario 5 (Fuente: Propia).....	32
Tabla 9 - Historia de usuario 6 (Fuente: Propia).....	32
Tabla 10 - Historia de usuario 7 (Fuente: Propia).....	33
Tabla 11 - Métricas y cumplimiento de las pruebas contra especificación ESRE (Fuente: propia)	44
Tabla 12 - Tabla de resultados de frecuencias de viaje (Fuente: Propia)	46
Tabla 13 - Tabla de resultados de tiempos aproximados de compra en condiciones normales (Fuente: Propia)	47
Tabla 14 - Tabla de resultados de tiempos aproximados de compra en feriados	48
Tabla 15 - Tabla de resultados de aceptación y uso de la aplicación (Fuente: propia)	48

Resumen

El presente proyecto muestra como optimizar el proceso de compra de boletos de viajes en buses interprovinciales a través de una aplicación móvil creada en Android Studio, lenguaje JAVA, aplicando la metodología de trabajo ágil SCRUM.

Dicha aplicación permite elegir la ruta de viaje, el horario en el cual se desee viajar y hasta 5 asientos disponibles por boleto, una vez finalizada la transacción, se envía un correo de verificación de compra al usuario previamente registrado.

El almacenamiento y lectura de la información relacionada a la aplicación se la realiza en una base de datos MySQL, subida a la plataforma Amazon Web Service, desde donde se gestiona todo tipo de transacción que la aplicación genere.

Se maneja un servicio web para la interacción transaccional de información entre la base de datos y la aplicación, dicho servicio está programado en el framework CodeIgniter de PHP y subido a la plataforma Heroku.

Finalmente, con los resultados, se comprueba la optimización del proceso de compra del boleto, en tiempo y comodidad, tanto para clientes como para vendedores.

Palabras claves:

Aplicación móvil, Optimización de Proceso de Compra, Servicios Web, Android Studio, SCRUM.

Abstract

This project shows how to optimize the process of buying travel tickets in interprovincial buses through a mobile application created in Android Studio, JAVA language, applying the agile SCRUM work methodology.

This application allows you to choose the travel route, the time in which you want to travel and up to 5 seats available per ticket, once the transaction is completed, a verification email is sent to the previously registered user.

The storage and reading of the information related to the application is done in a MySQL database, uploaded to the Amazon Web Service platform, from where all types of transactions generated by the application are managed.

A web service is managed for the transactional interaction of information between the database and the application, this service is programmed in the PHP CodeIgniter framework and uploaded to the Heroku platform.

Finally, with the results, the optimization of the ticket purchase process is verified, in time and comfort, for both clients and sellers.

Keywords:

Mobile application, Optimization of Purchase Process, Web Services, Android Studio, SCRUM

INTRODUCCIÓN

TEMA

Desarrollo de una aplicación móvil que optimice el proceso de compra de boletos de buses interprovinciales de la cooperativa de transporte “Flota Imbabura” en la provincia de Imbabura

PROBLEMA

ANTECEDENTES

Varias décadas atrás donde nuestro país y más específicamente la ciudad de Ibarra, no contaba con tecnología de automatización de procesos cotidianos, el manejo de cobro del pasaje de un viaje en bus interprovincial se lo realizaba al momento de subirse al mismo o en su defecto, el controlador o ayudante del chofer, cobraba asiento por asiento el pasaje de cada persona, dependiendo cual sea el destino del pasajero.

“La Cooperativa de Transporte Flota Imbabura, fue fundada el 20 de Septiembre de 1950, en la ciudad de Otavalo, provincia de Imbabura, con el firme propósito de brindar un mejor servicio a la colectividad imbabureña, carchense y del Ecuador en general.” (Flota Imbabura, s. f.)

Su actividad de viajes de turismo interprovincial inicia desde el año de 1962.

SITUACIÓN ACTUAL

En la actualidad se maneja un proceso de venta de boletos al público en general, a través de compras directas de boletos físicos en sus oficinas a nivel nacional, los cuales se integran al ERP¹, que tiene la cooperativa; dichos documentos impresos, sirven como evidencia de compra del boleto al momento de realizar el viaje.

Actualmente el ERP automatiza la asignación de asientos al momento que el personal encargado de la venta realiza la compra del boleto, éste mismo sistema, es el responsable de optimizar el orden de venta de los mismos y su posterior impresión en papel térmico, mediante una impresora POS².

¹ERP Enterprise Resource Planning (Sistema de planificación de recursos empresariales).

²POS Point of sale (Punto de venta).

La venta de dichos boletos también contempla los costos preferenciales para personas con discapacidad, tercera edad, niños menores de 2 años, etc.

La información del viaje, incluye datos como: hora de salida, ruta, ciudad destino, número de bus, etc.

PROSPECTIVA

Se ha agilizado el proceso de compra de boletos mediante el uso de TI³, a través de teléfonos inteligentes (Smartphone).

En 2016, la tenencia de los mismos creció 15,2 puntos del 2015 al 2016 al pasar del 37,7% al 52,9% de la población que tienen un celular activado. (INEC, 2016).

Dichos teléfonos permiten visualizar rutas, horarios, asientos y pagar con una tarjeta de débito crédito o prepago desde simple un botón de compra, gestionado por una pasarela de pago⁴, transacción efectuada entre el cliente (denominado Consumidor en la “Figura 1 - Proceso de compra online a través de una pasarela de pago.”) y el Comerciante.

A continuación, una representación del proceso de compra en línea mediante una pasarela de pago.



Fig. 1 - Proceso de compra en línea a través de una pasarela de pago. (Fuente: (Urbano, 2016))

³ TI Tecnologías de la información.

⁴ Una pasarela de pago es un intermediario de pago seguro mediante tarjetas de crédito, débito o prepago en una transacción en línea.

PLANTEAMIENTO DEL PROBLEMA

El mayor inconveniente al momento de comprar un boleto de viaje es realizar colas que en tiempo de feriados o simples fines de semana, se extienden por varios decámetros.

Si bien es cierto que los sistemas que se manejan actualmente ayudan a la organización de venta de boletos, no se ha pensado en una forma de optimizar el tiempo en que se venden los mismos desde una ventanilla, haciendo que se generen discusiones entre cliente - vendedor, debido al tiempo que tienen que esperar para comprar dicho boleto.

El tiempo de compra de boletos en feriados, varía mucho entre ciudades grandes como Quito, Guayaquil y Cuenca en comparación a otras no tan grandes como Tulcán o Ibarra. Normalmente de 10 a 30 minutos es el tiempo de espera, pero en feriados incrementa a más de una hora.

El aumento de la demanda del servicio en tiempo de feriados, hacen que el proceso de la oferta se ralentice a tal punto de afectar el proceso normal de compra. Una de estas anomalías es la alteración del precio del boleto, que llega hasta un 50% adicional del valor normal por concepto de envío de unidades (carros) extra, cosa que altera aún más a los usuarios, dejando una mala experiencia.

En la "Fig. 2 – Diagrama de Ishikawa", se pueden apreciar las principales problemáticas que tienen las empresas de transporte interprovincial, debido a que carecen de un proceso de compra ágil.

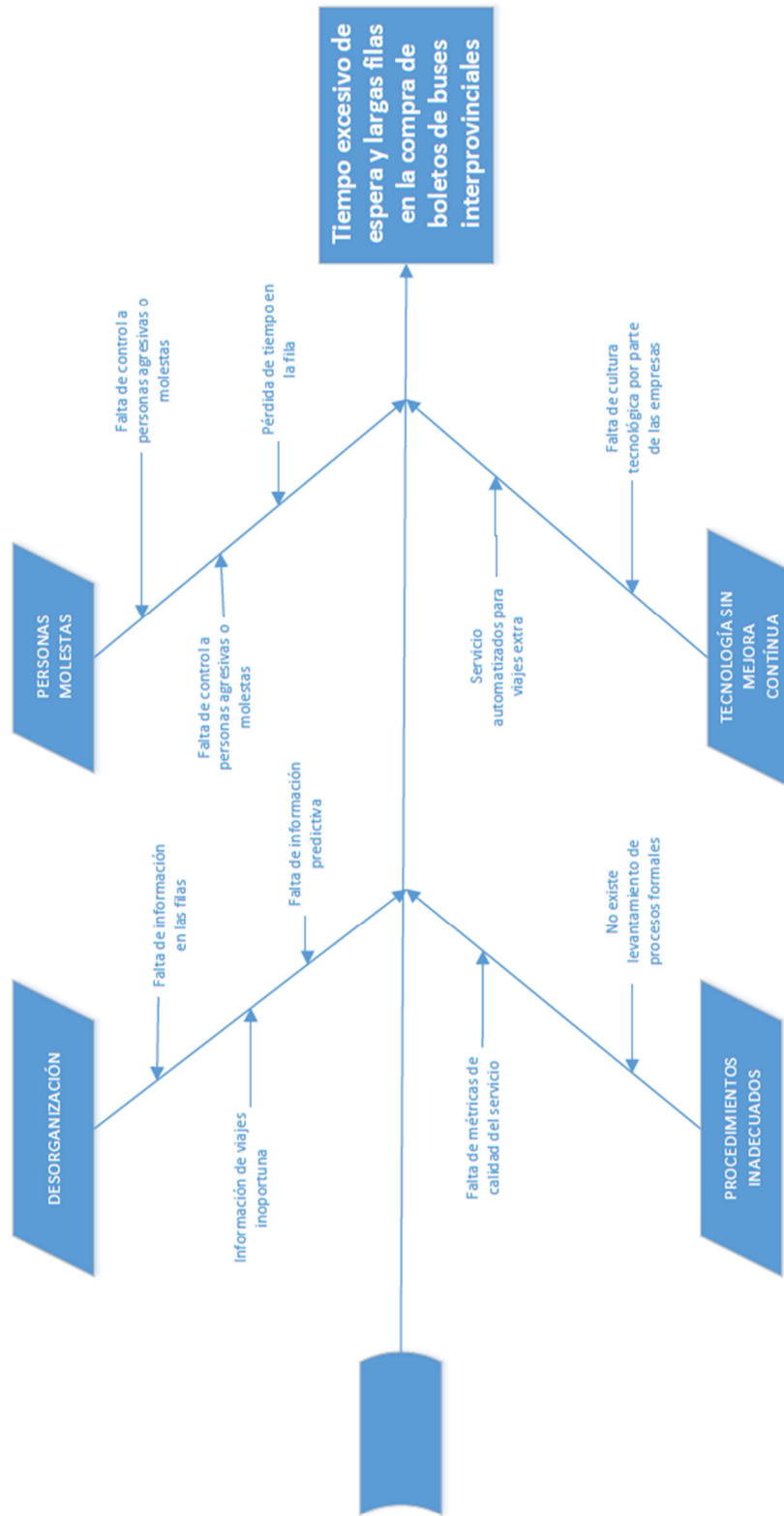


Fig. 2 - Diagrama de Ishikawa. (Fuente: Propia).

OBJETIVOS

OBJETIVO GENERAL

Desarrollar una aplicación móvil que optimice el proceso de compra de boletos de buses interprovinciales de la cooperativa de transporte “Flota Imbabura” en la provincia de Imbabura.

OBJETIVOS ESPECÍFICOS

- Recopilar información del proceso de ventas mediante automatización con software.
- Aplicar la metodología SCRUM como marco de trabajo ágil en el desarrollo de la aplicación móvil.
- Implementar el ciclo de vida de la aplicación.
- Validar los resultados del proyecto.

ALCANCE

El ciclo de vida del proyecto se compone de las fases que caracterizan a la metodología a utilizarse y son las siguientes:

- Toma de requerimientos
- Planificación
- Desarrollo
- Pruebas

Dichas fases del proyecto, se realizarán únicamente para Smartphone con sistema operativo Android.

De todos los diferentes Smartphone disponibles en el mercado actual, Android es el más dominante en toda América Latina. Según las investigaciones de StatsCounter⁵, los porcentajes de adopción de los Smartphone en Ecuador entre abril y junio de 2013 fueron los siguientes: Android (40%) – iOS (20%) – Nokia (14%) – Blackberry (11%). (Neomobile, 2014)

⁵ **StatsCounter** es una herramienta de análisis de tráfico web fundada en 1999 por la empresa con el mismo nombre.

La infraestructura a implementar se trata de un servicio web RESTful⁶ en el framework CodeIgniter de PHP, mismo que se encuentra subido a un servidor de aplicaciones en Heroku⁷ y a la vez, está conectado a una base de datos MySQL en un RDS⁸ de Amazon Web Service⁹, el cuál sirve de almacén de datos de rutas, boletos, horarios, etc., información a ser consumida por la aplicación.

En la “Fig. 3 – Diagrama de la arquitectura del sistema”, se pueden apreciar los componentes antes mencionados.

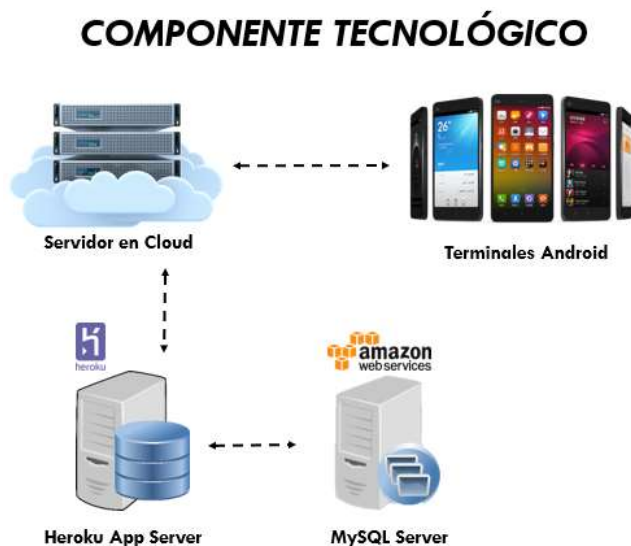


Fig. 3 - Diagrama de la arquitectura del sistema (Fuente: Propia)

JUSTIFICACIÓN

El proyecto tiene una justificación tecnológica. Los beneficios del presente trabajo son los siguientes:

- Medio ambiente: Al reducir la impresión de papel y por ende la contaminación que ésta genera en calles, alcantarillas, bosques, etc.

La ciencia y la tecnología pueden servir para ayudar a la conservación del medio ambiente. Algunos ejemplos son el reciclaje de determinados materiales como el vidrio, el papel, etc., evitando así la sobreexplotación de algunas materias primas (madera, etc.). (Sol Vásquez, 2010)

⁶ **REST** (Representational State Transfer) es una arquitectura que se ejecuta sobre HTTP. RESTful hace referencia a un servicio web que implementa la arquitectura REST.

⁷ **Heroku** es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación.

⁸ **RDS** Relational Database Service (Servicio de base de datos relacionales).

⁹ Amazon Web Services es una colección de servicios que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon.com.

- Crecimiento económico: Al introducir en el mercado una forma de cobro del servicio de ventas electrónicas y disminuir el gasto de tintas de impresoras POS en cooperativas de transporte.

La utilización de Sistemas de Venta sobre software (SVS), también conlleva ventajas financieras, como por ejemplo la realización de estadísticas de compra, venta, etc., facilitando considerablemente todas aquellas cuestiones contables y financieras que se necesiten para la gestión comercial del establecimiento. (Migues Pérez, 2011).

- Optimización del tiempo personal: Al entregar al público en general, una herramienta de compra segura y rápida de un boleto de viaje, desde su Smartphone, sin tener que esperar largas filas ni tener que desplazarse a los diferentes terminales terrestres.

En cuanto a las aportaciones más importantes del proyecto a desarrollarse, se puede destacar la secuela de la validación de resultados que dejará la realización del proyecto, dichos resultados de tiempo del proceso de compra de boletos, pueden servir como punto de referencia a futuras investigaciones o proyectos colaterales a desarrollarse.

El proyecto tiene características que ayudan al progreso de la agenda de los Objetivos de Desarrollo Sostenible¹⁰ en los puntos 12 y 9 que respectivamente mencionan “El consumo y la producción sostenible consisten en fomentar el uso eficiente de los recursos y la energía, la construcción de infraestructuras que no dañen el medio ambiente” y “Los servicios de transporte eficientes generan empleo y riqueza, e impulsan el desarrollo económico. Se calcula que, en 2015, los efectos económicos mundiales directos e indirectos del transporte, fueron de 2,7 billones de dólares.”(edX, 2018)

CONTEXTO

Se hace mención del hallazgo de una aplicación móvil que realiza funciones similares al proyecto a desarrollarse, pero en menor escala. La misma se encuentra en la PlayStore que ofrece Google en sus terminales Android y por sus características fue desarrollada por una persona vinculada a la Universidad Politécnica Salesiana del Ecuador. Cabe destacar que dicha aplicación móvil, no cuenta con la emisión de comprobantes de venta de boletos que luego puedan ser receptados por el controlador de turno del viaje, únicamente muestra rutas de algunas empresas de transporte, pero únicamente ofrece el servicio de venta de boletos para una sola cooperativa y no tiene validez real.

En cuanto al repositorio de tesis de la Universidad Técnica del Norte del Ecuador (UTN), se pudo encontrar dos temas afines al presente proyecto. El primero se titula “SISTEMA

¹⁰ Los ODS (Objetivos de desarrollo sostenible), son un llamado universal a la adopción de medidas para poner fin a la pobreza, proteger el planeta y garantizar que todas las personas gocen de paz y prosperidad.

WEB TRANSUR CON NODE.JS PARA LA GESTIÓN DE TRANSPORTE DE LA COOPERATIVA DE TRANSPORTE DE PASAJEROS INTERCANTONAL URCUQUÍ” (Cangás, 2015), y el segundo “IMPLEMENTACIÓN DE UNA APLICACIÓN PARA LA OBTENCIÓN DE SERVICIO SEGURO DE TAXIS MEDIANTE LA GEOLOCALIZACIÓN” (Cevallos, 2015).

Se realizó un análisis de las características de cada proyecto y en la “Tabla 1 – Comparativa de proyectos relacionados”, se puede visualizar las principales diferencias que tendrá el proyecto actual con todos los antes mencionados.

	App Flota	App Taxis	App Urcuqui	App UPS
Comprobante de venta de boleto	NO	NO	NO	NO
Aplicación Nativa	SI	SI	SI	SI
Servicio de Geolocalización	NO	SI	NO	NO
Gestión de la información	SI	NO	SI	SI

Tabla 1 - Comparativa de proyectos relacionados. (Fuente: Propia).

CAPÍTULO 1

Marco teórico

1.1. Introducción a las aplicaciones móviles

1.1.1. ¿Qué es una aplicación móvil?

Una aplicación nace de la necesidad de solucionar problemas en actividades periódicas como también de concebir una manera más fácil de realizar ciertas acciones que pueden automatizarse gracias a la tecnología móvil.

También llamada app, es un programa informático que funciona en teléfonos inteligentes, tabletas y otros dispositivos móviles, sirve para gestionar un sin número de actividades cotidianas que van desde trabajo, educación, entretenimiento, comunicación, geolocalización, etc., éstas pueden ser de pago o gratuitas y brindan al usuario agilidad y comodidad en su diario vivir.

Las aplicaciones se ejecutan en un sistema operativo móvil como Android¹¹, iOS¹², Windows Phone¹³, etc., y se encuentran disponibles en plataformas de distribución propia de cada sistema operativo. A continuación, se muestra en la “Fig.4 – Tienda de aplicaciones según sistema operativo” cada una de ellas.



Fig. 4 - Tienda de aplicaciones según Sistema Operativo (Fuente: Propia)

¹¹ Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles

¹² iOS es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone

¹³ Windows Phone es un sistema operativo desarrollado por Microsoft y funciona para dispositivos móviles de la misma empresa.

1.1.2. Tipos de aplicaciones móviles

Las aplicaciones se clasifican tomando en cuenta varios puntos de vista, entre ellos:

a) Por sus efectos psicosociales y psicopedagógicos:

- Aplicaciones de capacitación: aquellas que ofrecen material de aprendizaje de un tema de interés y que tienen como objetivo la enseñanza y ayuda académica de las personas interesadas en dicho tema. Por ejemplo, Duolingo es una app de aprendizaje de varios idiomas.

- Aplicaciones de dependencia: aquellas que nos ayudan a estar pendientes de nuestra salud y prevenir enfermedades con hábitos de vida saludables (cuidado de la alimentación, seguimiento de ejercicio, etc.). Entre las más populares se encuentran Samsung Health o Google Fit

b) Por los contenidos que ofrecen al usuario:

- De entretenimiento: donde se encuentran principalmente los juegos.

- De relación social: Orientadas a la comunicación interpersonal en grupos sociales de cualquier lugar del mundo.

- De producción o utilitarias: suministran material de resolución de tareas específicas que requieren inmediatez para solucionar conflictos, en especial en el sector empresarial y comercial.

- Educativas o informativas: son aquellas intermediarias de la información y el conocimiento entre el usuario y una plataforma de información, donde se prioriza el acceso a los contenidos y herramientas de búsqueda a través de una interfaz de navegación amigable al usuario.

- Creativas: ofrecen herramientas que incentivan la creatividad literaria, musical, fotográfica o video-gráfica.

- Publicitarias: con fines comerciales la gran mayoría son de distribución gratuita.

c) Por las condiciones de distribución: Se clasifican en gratuitas, de pago y freemium¹⁴, las cuales permiten su descarga gratuita para un uso limitado y básico, dando a futuro, el acceso a funcionalidades más avanzadas previo pago.

d) Por categorías de edad de los usuarios del contenido: El App Store impone una clasificación del contenido por tramos de edades de "4+, 9+, 12+ y 17+", que restringe la descarga de dicha aplicación.

¹⁴ Freemium es un modelo de negocio que funciona ofreciendo servicios básicos gratuitos, mientras se cobra dinero por otros servicios más avanzados o especiales.

e) Por el tipo de infraestructura: su desarrollo está diferenciado por el sistema operativo en el que funcionan y son las siguientes.

- Genéricas: prácticamente todo el diseño y programación de lenguaje es compatible con la mayoría de los dispositivos, pero son propensas a limitar sus funciones.
- Híbridas: determinados componentes de la programación son comunes para todos los Smartphone y otro porcentaje es específico, dependiendo del sistema operativo.
- Nativas: su programación en su totalidad es específica para cada sistema operativo y su gama de funcionalidades es bastante alta.



Fig. 5 - Representación de las aplicaciones móviles. (Fuente: Carballar, 2012)

1.2. Herramientas de desarrollo para aplicaciones móviles

Existen varias herramientas y tecnologías de desarrollo con las que se puede construir aplicaciones móviles capaces de cubrir los requerimientos del programador en cuanto a diseño y funcionalidad.

Con la existencia de varios sistemas operativos móviles, la construcción de aplicaciones se vuelve complicada para los desarrolladores y es por ese motivo que se han diseñado herramientas multiplataforma, que ayuden en el proceso de desarrollo de aplicaciones para diferentes sistemas. Las tendencias de las mismas están entre Android, iOS, y Windows Phone ya que son las más utilizados en la actualidad.

1.2.1. Lenguajes de Programación

En el desarrollo de aplicaciones y sistemas informáticos en general, el lenguaje de programación es determinante. Android trabaja con JAVA y C, mientras que iOS se compila en Objective-C y Swift, otro ejemplo es Windows Phone que usa el lenguaje Visual Basic.net y C#, en fin, existen varias tecnologías que funcionan de manera diferente, sin embargo existe la posibilidad de desarrollar aplicaciones que funcionen de la misma manera en diferentes sistemas operativos, esto gracias a ciertos framework que dan dichas características a las aplicaciones desarrolladas mediante tecnologías como HTML, CSS y JavaScript.

1.2.2. Plataformas de desarrollo para aplicaciones móviles

Una de las principales ventajas de la construcción de aplicaciones es que existe un sin número de opciones para realizar dicha actividad, algunas para desarrollo nativo, otras para crear apps híbridas, etc. A continuación, se enuncian algunas plataformas de desarrollo de aplicaciones y sus características.

- **Xcode.** Es una plataforma de desarrollo de aplicaciones móviles nativas para iOS, su proveedor es Apple y es considerada la más poderosa, utiliza un lenguaje Objctive C y Swift, posee una colección de compiladores y es capaz de compilar código C, C++, Objective-C++, Objective-C, Apple Script y Java ya que posee varios modelos de programación.
- **Xamarin Studio.** Es una plataforma para desarrollo de aplicaciones móviles multiplataforma, su principal característica es la capacidad que tiene para que el lenguaje de programación C# pueda ser traducido para ejecutarse en iOS, Android y Windows Phone, sin la necesidad de crear una app para cada sistema operativo ya que esta herramienta unifica las diferencias entre los mismos. (Setfree, s. f.)
- **B4A.** Existen herramientas que brindan eficiencia al momento de construir apps móviles y este es el caso de B4A, que significa “Basic for Android” y es un IDE¹⁵ que usa el lenguaje BASIC¹⁶, el cuál es fácil de aprender, además brinda la posibilidad de acceder a librerías propias de Android Studio, pudiendo desarrollar grandes cosas sin dejar de lado las funcionalidades necesarias. (Lasso, 2017)
- **Android Studio.** Es una plataforma de desarrollo de aplicaciones móviles nativas para Android, su proveedor es Google y su distribución es gratuita. Entre sus ventajas está la facilidad de integrar librerías compiladas desde internet o bien agregadas al

¹⁵ **IDE** Integrated Development Environment (Entorno de desarrollo integrado)

¹⁶ **BASIC** Beginners All Purpose Symbolic Instruction Code (Es una familia de lenguajes de programación de alto nivel)

proyecto, es por este motivo que en el actual trabajo de grado se utilizará dicha herramienta.

1.3. Sistema operativo Android

En la localidad de Palo Alto California - EEUU, en el año 2003, Andy Rubin, Rich Miner, Chris White y Nick Sears, buscaron desarrollar un sistema operativo para teléfonos móviles, y es así que crean la empresa Android Inc., que dos años más tarde fue adquirida por Google, donde conjuntamente con un equipo de trabajo, se creó una plataforma para dispositivos móviles, con el fin de proveer un sistema más fácil para los usuarios y a la vez actualizable. (Venturi, 2017)

Android, es una plataforma de código abierto para dispositivos móviles, está basada en Linux y fue desarrollada por Open Handset Alliance¹⁷, actualmente es el sistema operativo móvil con mayor presencia alrededor del mundo.

Al afirmar que es el sistema operativo más utilizado, no lo es únicamente en teléfonos inteligentes, sino también en relojes, televisores y cierto tipo de automóviles, ya que tiene la capacidad de integrar varios dispositivos, permitiendo que sea un sistema operativo funcional e interactivo. A continuación se muestra una representación de la presencia de Android en autos.



Fig. 6 - Sistema operativo Android en automóviles (Fuente: (LaPrensa, 2015))

1.3.1. Características principales de Android

El sistema operativo Android tiene las siguientes características:

¹⁷ **Open Handset Alliance** es una alianza comercial de 84 compañías que se dedica a desarrollar estándares abiertos para dispositivos móviles.

- ❖ Es una plataforma de código abierto.
- ❖ El núcleo está basado en el Kernel de Linux.
- ❖ Es adaptable a muchas pantallas y resoluciones.
- ❖ Navegador web (basado en WebKit¹⁸) incluido.
- ❖ Utiliza SQLite como almacén de datos.
- ❖ Soporte para Java y otros formatos multimedia.
- ❖ Soporte de HTML, HTML5, Adobe Flash Player.
- ❖ Herramientas para depuración de memoria y análisis de rendimiento de software.
- ❖ Tienda de aplicaciones gratuitas y pagadas donde pueden ser descargadas e instaladas desde Google Play.
- ❖ Multitarea real de aplicaciones. (Venturi, 2017)

1.3.2. Arquitectura del sistema operativo Android

La Arquitectura de Android se compone de cuatro capas estructurales que forman cinco secciones, entre las características más importantes se pueden mencionar que Android posee el Kernel de Linux y las capas están basadas en software libre, a continuación, una representación de la misma.

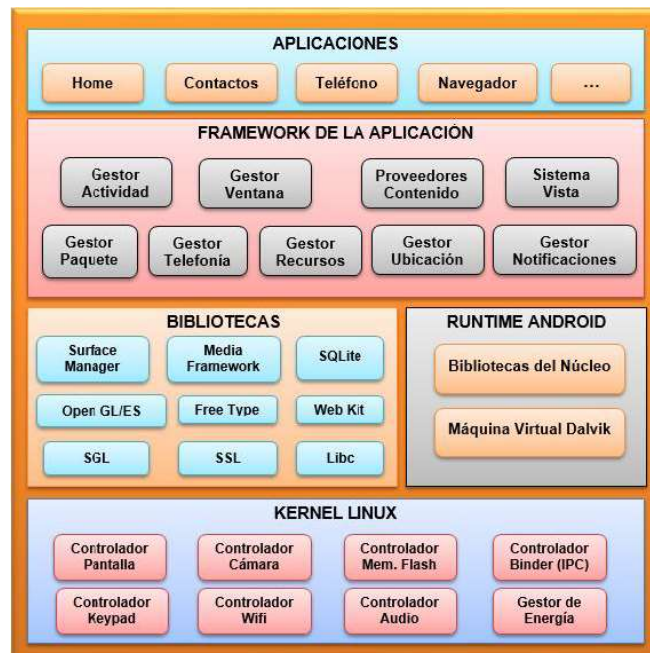


Fig. 7 - Arquitectura Android (Fuente: (Tatés Lenin, 2018))

¹⁸ **WebKit** es una plataforma para aplicaciones que funciona como base para el navegador web Safari, Opera, etc., desarrollado por Apple.

1.3.3. Kernel Linux

Una de las capas de mayor importancia en la arquitectura de Android es el Kernel de Linux, llamado también núcleo de Linux, que es el motor del sistema operativo. La versión de Linux 2.6, se encarga de proporcionar servicios y controles de seguridad, protocolos, multiprocesos y el manejo de memoria, entendiéndose que la capa posee los controladores de dispositivos de bajo nivel para todos los componentes del hardware de un dispositivo Android.

1.3.4. Runtime de Android

Es el entorno de ejecución de Android el cual está compuesto por las bibliotecas o librerías del núcleo que permiten a los desarrolladores generar aplicaciones Android utilizando Java y además se incluye la máquina virtual Dalvik, con la que cada aplicación se puede ejecutar en un proceso independiente, con una instancia de la máquina virtual Dalvik. (D'Andrea Edgar, 2015)

1.3.5. Bibliotecas o librerías

Esta capa contiene a las librerías principales de Android, éstas están escritas en lenguaje C/C++, y la mayor parte utilizan código abierto, su función principal es proporcionar las funcionalidades más importantes y representativas a los componentes, las librerías junto con el núcleo de Linux conforman la parte principal del sistema operativo Android.

1.3.6. Framework de la aplicación

Esta capa fue creada principalmente para simplificar la reutilización de componentes, es decir las aplicaciones pueden mostrar sus funcionalidades y otras pueden hacer uso de estas, este acceso a las funcionalidades de otras aplicaciones se maneja a través de varias restricciones de seguridad.

- **Gestor de Actividad.** Se encarga del manejo y gestión del ciclo de vida de las aplicaciones.
- **Gestor de Ventana.** Maneja las ventanas de las aplicaciones, utilizando la librería Surface Manager.
- **Gestor de Teléfono.** Contiene todas las API vinculadas a las funcionalidades propias del teléfono como las llamadas y mensajes.
- **Proveedores y Contenido.** Su función es permitir a cualquier aplicación Android el acceso a datos de otras aplicaciones como los contactos, agendas, mensajes.

- **Sistema de Vista.** Es el encargado de proveer de los elementos necesarios para construir la parte visual de los componentes o las interfaces de usuario.
- **Gestor de Ubicación.** Su función es posibilitar a las aplicaciones a obtener información de posición y localización.
- **Gestor de Notificaciones.** Permite que las aplicaciones muestren alertas personalizadas en la barra de estado.
- **Gestor de Recursos.** Encargado de proporcionar acceso que no contienen código. (Tomás J, 2016)

1.3.7. Aplicaciones

Esta capa se conforma de un grupo de aplicaciones que se ejecutan en la máquina virtual ya mencionada (Dalvik), con el objetivo de resguardar la seguridad en el sistema operativo Android, es decir, la capa contiene a las aplicaciones que vienen por defecto instaladas en el sistema operativo, de igual manera también se encuentran las aplicaciones que el usuario instale, ya sean descargadas o desarrolladas por otras empresas o también desarrolladas por el propio usuario.

1.4. Optimización de procesos.

El éxito de finalizar una actividad logrando un determinado propósito, se centra en realizar de manera eficiente y eficaz el proceso que conlleva hacer dicha actividad. Muchas veces para lograr un objetivo se realizan procedimientos no definidos ni estandarizados, lo que da resultados diferentes, a veces a favor y otras en contra del resultado que esperábamos.

1.4.1. ¿Qué es un proceso?

Está definido como la sucesión de actos o acciones realizados con cierto orden, que se dirigen a un punto o finalidad, así como también al conjunto de fenómenos activos y organizados en el tiempo. En el campo empresarial y económico, el proceso es una secuencia de actividades humanas, que transforman un conjunto específico de insumos en uno de rendimientos y resultados. Se pueden encontrar al proceso productivo, en donde el resultado es un producto o servicio; y al proceso de negocio, en éste se llevan y concluyen tareas de manera lógica como transferir mercancías, efectuar negociaciones, etc. (Concepto definicion.de, 2014)

La mayor parte de acciones que realiza una persona todos los días es un proceso, desde que se levanta hasta que su día termina; y muchas de las cosas que realiza diariamente, conllevan un determinado orden.

Un proceso recibe entradas (pueden ser el resultado de otro proceso), éstas pueden ser componentes, materia prima, instrucciones, información, criterios, etc., luego realiza actividades de valor agregado en esta entrada y finalmente crea una salida.

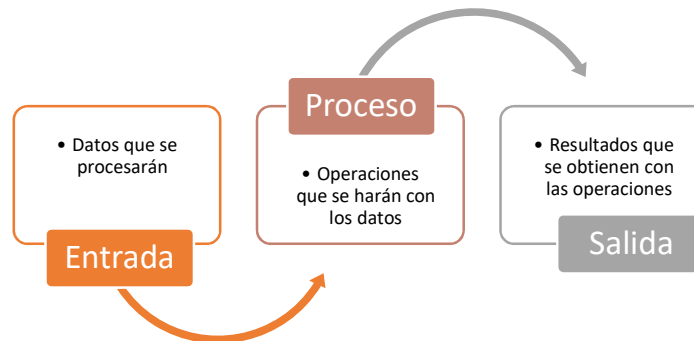


Fig. 8 - Actividades de un proceso (Fuente: Propia)

1.4.2. Optimización de procesos

Cuando se habla de optimizar procesos, se entiende que se estudiará dicho proceso de cierta actividad y a continuación se presentan alternativas de mejora tanto en tiempo, calidad y/o precio (hablando tanto de productos como servicios). Una vez se escoge la mejor alternativa de mejora, se aplica en el proceso actual, cambiando o adecuando las actividades que lo conforman, de tal manera que las salidas o resultados del mismo, reflejen superación o mejora, a los resultados del anterior proceso.

El ejemplo más práctico de optimización de procesos es cuando se establecen ciertas normas y parámetros que permitan dar como resultado un producto o servicio de calidad, en este caso, la aplicación móvil se ha desarrollado de tal manera que permita al usuario-pasajero, comprar su boleto de viaje desde su Smartphone sin necesidad de desplazarse a una oficina de la cooperativa OPTIMIZANDO el tiempo de compra del mismo, como también denotando CALIDAD DE SERVICIO AL CLIENTE brindándole la posibilidad de realizar la compra desde donde quiera que se encuentre, siempre y cuando tenga una conexión a Internet.

El propósito de la optimización de procesos es reducir o eliminar la pérdida de tiempo y recursos, gastos innecesarios, obstáculos y errores, a continuación, se muestra el paso a paso de cómo llegar a ese objetivo.

- **IDENTIFICAR.** En primer lugar, se debe buscar el motivo del problema por el cual nos centramos en cambiar la forma de cómo hacer las cosas. Para esto se deben realizar preguntas con el fin de determinar lo antes mencionado, las cuales se citan a continuación:

- a) ¿Cuál es el objetivo final de este proceso? ¿Cuál debe ser el resultado?
- b) ¿Dónde comienza el proceso y dónde termina?
- c) ¿Qué actividades son parte del proceso y lo llevan para adelante?
- d) ¿Qué departamentos y funcionarios están involucrados?
- e) ¿Qué información viaja entre los pasos?

- **REPENSAR.** Este es el momento de mapear el proceso, con la preocupación de cómo los pasos se realizan, cómo fluye el proceso, como parte de la optimización de procesos. Hay que preguntarse a sí mismo y al equipo lo siguiente:

- ¿Hay una mejor manera de llevar a cabo este proceso?
- ¿Cómo se conduce exactamente este proceso?
- ¿Cuánto papel (por ejemplo), se utiliza en este proceso?
- ¿Cuánto tiempo demora para que el proceso sea realizado por completo?
- ¿Cuánto tiempo se pierde en la reanudación y corrección de errores?
- ¿Dónde se paraliza el proceso?

- **IMPLEMENTAR.** Después de conocer el proceso en detalle e identificar las posibilidades de cambio y la necesidad de mejoras, es el momento de poner en práctica el proceso de una manera nueva. Esta es una parte delicada de la optimización de procesos. *Es crucial tanto para los objetivos del proceso como para la optimización de los mismos, que todos adopten el nuevo proceso desde el principio y apliquen todos los cambios que muestran.* Así se pueden comprobar los resultados, obtener información y ver si las mejoras fueron positivas o no. Puede suceder que el proceso no termine como estaba previsto, que el equipo no se acostumbre, que la aplicación no se haya hecho correctamente. En tales casos, es necesario iniciar el proceso de nuevo.

- **AUTOMATIZAR.** Insistir en los mismos errores y esperar resultados diferentes, es la receta del fracaso. Repetir y aumentar las prácticas exitosas, es la clave para prosperar. Hay que automatizar los procesos que han sido probados y aprobados, hay que distribuirlos por la empresa y ver los resultados en la reducción de gastos, la prevención de errores, la disminución de desperdicios y una mayor productividad. (Pacheco, 2017b).

1.5. Servidor de base de datos

Actualmente existen variedad de sistemas informáticos que han sido implantados para automatizar procesos industriales, económicos, administrativos, etc., y todos, o al menos en su gran mayoría, manejan un almacén de datos, el cual provee a la aplicación, de información que puede ser administrada desde el mismo sistema informático, como también conectarse mediante ciertas tecnologías con otros sistemas.

Éste almacén de información se lo ha denominado “Base de Datos (por sus siglas DB- DataBase), y para su correcto funcionamiento y gestión, necesita ser instalado en una máquina servidor, que bien puede estar en internet (también denominada NUBE), o puede administrarse localmente desde un PC, en donde se resalta que, si el sistema dependiente de dicha base de datos se encuentra en la misma PC, no es necesario tener una conexión a internet, caso contrario, si el sistema que depende de dicha base de datos se encuentra en otro dispositivo, es de vital importancia que el servidor cuente con una conexión activa y permanente a la red.

1.5.1. Servidores de base de datos.

Los servidores de bases de datos surgen en la década del 80 con motivo de la necesidad de las empresas de manejar grandes y complejos volúmenes de datos, al tiempo que requieren compartir la información con un conjunto de clientes (que pueden ser tanto aplicaciones como usuarios) de una manera segura y debe proporcionar servicios de forma global y, en la medida de lo posible, independientemente de la plataforma. (EcuRed, 2018b)

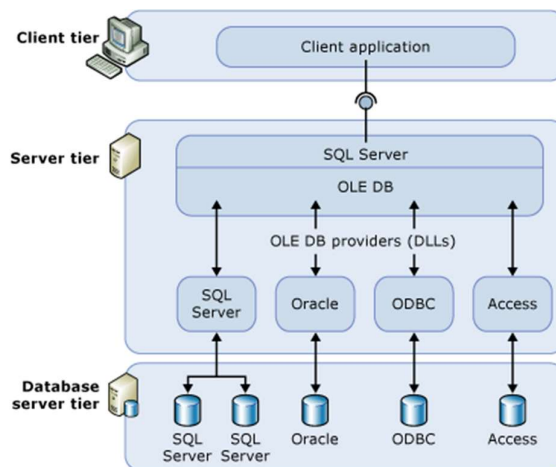


Fig. 9 - Diagrama de interconexión de servidores de base de datos (Fuente: (Stein Steve, 2017)

1.5.2. Sistemas gestores de base de datos.

Un sistema gestor de bases de datos (SGBD, a partir de ahora) ofrece soluciones de forma fiable, rentable y de alto rendimiento. Éstos son capaces de almacenar contenidos multimedia, objetos, documentos complejos, etc. Para que esto sea posible, se han creado herramientas de administración completas (que simplifiquen la tarea de la configuración, seguridad, creación y gestión de bases de datos al tiempo que proporcionan mecanismos de integración con otros sistemas y políticas de copias de seguridad) y herramientas que permitan su programación (tanto a nivel de diseño como a nivel de reglas y procedimientos que encapsulen la arquitectura de la base de datos, de tal manera que, a través de conectores a datos, las aplicaciones sólo tengan que pedir la información que necesitan sin preocuparse de cómo se encuentra almacenada). (Menéndez Rafael, 2000)

Existe variedad de SGBD, cada uno con funcionalidades que se acoplan a cierto tipo de aplicaciones y servicios. En forma general sus características son las siguientes:

- Permite una vista muy centralizada y clara de los datos para que sean accedidos de la mejor manera posible.
- Se encargan de gestionar adecuadamente los datos, evitando a los usuarios o programas que les requieren, tener que entender dónde se encuentran físicamente los datos.
- Estos sistemas disponen de un lenguaje de programación SQL¹⁹ para poder proteger y acceder a los datos.
- La necesidad de requerir de un lenguaje para su acceso y su autonomía como sistema, proporcionan integridad y seguridad a los datos.
- Suelen disponer de un sistema de bloqueo para el acceso simultáneo, lo que le da un plus de seguridad a la integridad de los datos.
- Estos sistemas de base de datos disponen de API's²⁰ muy visuales e intuitivas para poder gestionar los datos.
- Un correcto SGBD proporcionará economías de escala en el procesamiento de grandes cantidades de datos ya que está hecho para ese tipo de operaciones.
- Los SGBD proporcionan un nivel de abstracción entre la estructura lógica de la base de datos y el esquema físico que describe el contenido físico usado por la base de datos.

¹⁹ **SQL** (Structured Query Language), en español Lenguaje de consulta estructurada.

²⁰ **API** (Application Programming Interface) en español Interfaz de programación de aplicaciones.

- El programa de gestión de almacenamiento y su gestión de datos (servidor) es totalmente independiente del programa con el cuál se realizan las consultas (cliente).
- Los SGBD realizan eficientes almacenamientos de los datos, pero estos se hacen de forma oculta para el usuario y nada tiene que ver con lo que finalmente se le presenta.
- Son capaces de gestionar distintos tipos de bases de datos, por ejemplo: bases de datos relacionales (suelen ser las estándar) y bases de datos orientadas a objetos.
- Multiplicidad de acceso a los datos. (Kyocera, 2017)

En cuanto a tipos de SGBD, se listará a los más relevantes y con más afluencia en el área de software:

- **SQL Server.** Es un sistema gestor de base de datos relacionales creado por Microsoft. Es un sistema cliente/servidor que funciona principalmente como una extensión nata de Windows. Entre otras características proporciona integridad de datos, optimización de consultas, control de concurrencia, respaldos y recuperación.
Es prácticamente fácil de gestionar la gran mayoría de tareas de sistema y administración de bases de datos a través de un entorno gráfico. Utiliza varios servicios de Windows para ofrecer gran variedad de funcionalidades, como también ampliar la base de datos, envío-recepción de mensajes y gestión de seguridad de la conexión. Además, proporciona funciones de almacenamiento de datos que sólo estaban disponibles en Oracle y otros sistemas gestores de bases de datos más caros.
- **MySQL.** MySQL es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes.
Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un

gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras. (EcuRed, 2018c)

- **PostgreSQL.** Es un SGBD Relacional Orientado a Objetos, desarrollado en la Universidad de California, en el Departamento de Ciencias de la Computación de Berkeley. Es un software de gestión de bases de datos de código abierto, ofrece un MVCC²¹ que permite trabajar con extensos volúmenes de datos y soporta gran parte de la sintaxis SQL.

Ofrece funcionalidades como claves primarias, identificadores entrecomillados, conversión de tipos y entrada de enteros binarios y hexadecimales.

El código fuente está disponible para cualquier persona sin costo alguno y para 34 plataformas con la última versión estable.

Posee también una integridad referencial e interfaces nativas para lenguajes como ODBC, JDBC, C, C++, PHP, PERL, TCL, ECPG; PYTHON y RUBY. Funciona en todos los sistemas operativos Linux, UNIX, Mac OS X y Windows. Dado a la liberación de la licencia, PostgreSQL se puede utilizar, modificar y distribuir de forma gratuita para cualquier fin, sea comercial, privado o académico.

- **ORACLE Database.** Es un SGBD Relacional Orientado a Objetos, desarrollado por Oracle Corporation. Ocupó gran parte del mercado de servidores empresariales hasta que eventualmente apareció su competencia como es el sistema Microsoft SQL Server y otros SGBD con licencia libre como PostgreSQL o MySQL.

Las últimas versiones de Oracle han sido certificadas para trabajar bajo GNU/Linux.

- **Access.** Es un SGBD incluido en el paquete ofimático de Microsoft Office, se trata de un gestor de datos que se basa en los conceptos de bases de datos relacionales y puede gestionarse a través de consultas e informes. Como es

²¹ **MVCC** Control de concurrencia multi-versión (MVCC por sus siglas en inglés)

de esperar, está programado para recopilar datos de otras utilidades de Office como Excel, SharePoint, etc.

1.6. Metodologías de trabajo.

Actualmente las metodologías de desarrollo de proyectos, se han convertido en el eje principal del ciclo de vida de los mismos, haciendo que se utilicen cada vez más y con mayor frecuencia. Existen varios marcos de trabajo que se adaptan a los diferentes proyectos que un ente empresarial, educativo, comercial, etc., desarrolla y de éste depende el éxito de la correcta conclusión o finalización del mismo.

1.6.1. Marco de trabajo

Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como modelo a seguir, para buscar y resolver nuevos problemas con similares características.

En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. (Wikipedia, 2018)

1.6.2. Metodologías de desarrollo de software

Concluir con éxito un sistema informático eficiente, que cumpla con los requerimientos planteados, es una tarea ardua y sobre todo difícil de cumplir. Las metodologías para el desarrollo de software, exigen un proceso disciplinado con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas.

- **Las metodologías ágiles.** Se caracterizan por hacer énfasis en la comunicación cara a cara, es decir, se basan en una fuerte y constante interacción, donde clientes y desarrolladores trabajan constantemente juntos, estableciéndose así una estrecha comunicación. Estas metodologías están

orientadas al resultado del producto y no a la documentación; exige que el proceso sea adaptable, permitiendo realizar cambios de último momento. Se puede hacer mención dentro de las metodologías ágiles a: XP²², Scrum y Crystal Methodologies. (EcuRed, 2018)

- **Las metodologías robustas o tradicionales.** Se caracterizan por seguir una planificación robusta y estricta. Ponen énfasis en llevar una documentación exhaustiva de todo el proceso de desarrollo, como también cumplir con un plan de proyecto, definido en la fase inicial del mismo. Entre las metodologías robustas se encuentran: MSF²³, MÉTRICA 3 y RUP²⁴.

Cabe mencionar que no son Metodologías de Desarrollo de Software la "Programación estructurada" o la "Programación Orientada a Objetos", éstas se caracterizan por ser paradigmas o modelos de programación.

Los términos "Ciclo de vida en espiral", "Incremental", en "Cascada", con "prototipo", etc., indican esquemas generales de organización en las tareas del ciclo de vida, unas con respecto a otras y con respecto a otros aspectos como el tiempo, los requisitos o el riesgo. Actualmente se denominan "PATRONES" del ciclo de vida del software, aunque antaño fueron denominados simplemente distintos "Ciclos de vida". Indican ideas estructurales sencillas en el proceso de desarrollo, y no la manera en la que debe realizarse cada tarea del ciclo para un proyecto concreto, así pues, NO SON METODOLOGÍAS. El lenguaje UML²⁵ es un gran logro de la ingeniería. Aún con sus carencias, es algo muy importante: un lenguaje común para que todos los profesionales del desarrollo de sistemas-de software o no, expresen sus ideas, pero UML no le indica a nadie la manera de realizar las tareas en un proyecto concreto: tan solo es una herramienta para expresar ideas, así pues, NO ES UNA METODOLOGÍA. Sin embargo, algunas metodologías de las que hemos comentado, como RUP o METRICA hacen referencia a UML como lenguaje de modelado para expresar ideas. (EcuRed, 2018a)

1.6.3. SCRUM como marco ágil de trabajo.

Scrum se ha convertido en el marco ágil de trabajo más usado en el desarrollo de software en el mundo. Uno de los beneficios de Scrum es la entrega de incrementos de producto en periodos de tiempo cortos. El trabajo se basa en la colaboración y trabajo en equipo. Algunos aspectos como el descubrimiento, el empirismo y la inspección continua son claves para lograr un proceso de mejora continua y de innovación en el

²² **XP** Extreme Programming en español Programación Extrema.

²³ **MSF** Microsoft Solution Framework.

²⁴ **RUP** Rational Unified Process en español Proceso Unificado Racional.

²⁵ **UML** Unified Modeling Language en español Lenguaje Unificado de Modelado

producto a construir. Scrum está basado en un proceso de control empírico y sirve para construir productos complejos de forma sostenida. (Francia Joel, 2017)

Es un proceso en donde se aplican regularmente un conjunto de buenas prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Dichas prácticas se apoyan entre sí y su objetivo principal es fomentar la creación de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Proceso de la metodología:

Aplicando el marco de trabajo ágil antes mencionado, el proyecto se ejecuta en ciclos cortos con tiempos fijos que por lo general son iteraciones de 2 semanas, aunque puede variar de 3 a 4 semanas máximo, lo cual representa el tiempo de desarrollo del producto real. Cada iteración debe dar un resultado completo, un avance del producto final que tenga disponibilidad inmediata al cliente y que pueda ser incluido en producción.

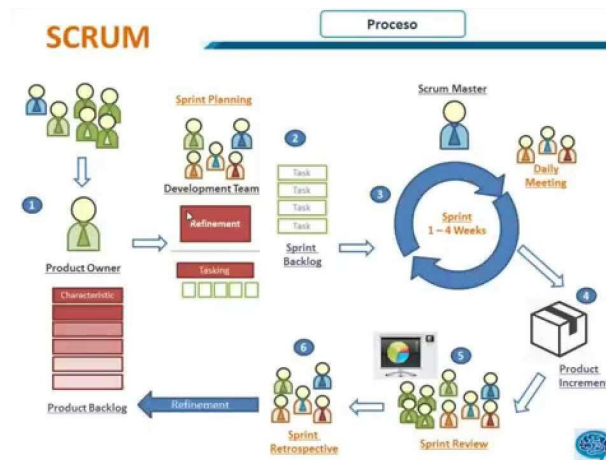


Fig. 10 - Proceso del marco de trabajo ágil SCRUM (Fuente: (Pacheco, 2017a)

El proceso inicia con una toma de objetivos/requisitos priorizada del producto, que se refleja como un plan de proyecto. En esta lista el cliente (llamado Product Owner) da la prioridad a los objetivos, equilibrando el valor que le aportan a su negocio respecto al coste de dichos objetivos, quedando repartidos en iteraciones y entregas.

Las actividades a llevarse a cabo en Scrum, con iteraciones de 2 semanas son las siguientes:

➤ **Planificación de la iteración**

Como primer paso de la iteración se ejecuta la reunión de planificación de la misma, la cual consta de dos partes:

- (1) **Selección de requisitos (2 horas)**. El Product Owner comunica al equipo la lista de requisitos priorizada del proyecto. El equipo de trabajo pregunta al Product Owner las posibles dudas que surjan y toman los requisitos de primer orden para completar la iteración, de tal forma que puedan ser entregados si el cliente lo solicita.
- (2) **Planificación de la iteración (2 horas)**. El equipo documenta la lista de tareas de la iteración a desarrollar, las cuales fueron dadas por el Product Owner. La estimación de esfuerzo de cada tarea se realiza en conjunto y los miembros del equipo seleccionan sus tareas, su organización de trabajo incluso puede ser en grupos de dos o más personas, con el objetivo de compartir conocimiento y así formar un equipo más resiliente como también resolver objetivos especialmente complejos.

➤ **Ejecución de la iteración**

Todos los días el equipo asiste a una reunión informativa de socialización y avance de las tareas del proyecto que dura aproximadamente 15 minutos, normalmente en una pizarra o proyector. El equipo se informa sobre el trabajo que los demás están realizando, donde se puede dialogar y debatir sobre dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo, etc., para hacer las conciliaciones necesarias que permitan cumplir con la perspectiva de los objetivos a mostrar al final de la iteración.

En esta reunión los miembros del equipo responden a tres preguntas:

- (1) ¿Qué tareas ha desarrollado desde la última reunión de sincronización para contribuir al equipo en el avance de su objetivo?
- (2) ¿Qué tareas realizará a partir de este momento para ayudar al equipo a cumplir la iteración?
- (3) ¿Cuáles son los impedimentos que reprimen la conclusión del objetivo?

Durante la iteración el líder del proyecto (llamado Scrum Master) es responsable de que el equipo pueda mantenerse enfocado en cumplir sus objetivos, como también cumplir con lo siguiente:

- (4) Reprimir los inconvenientes que el equipo pueda tener y no pueda resolver por sí mismo.
- (5) Evitar que el equipo tenga interrupciones externas que afecten al objetivo de la iteración o la productividad del mismo.

En el proceso de desarrollo de la iteración, el Product Owner junto con el equipo de trabajo, filtran la lista de requisitos con el fin de prepararlos para las siguientes iteraciones y, de ser necesario, cambian o planifican de nuevo los objetivos del proyecto, lo cual toma un 10% a 15% del tiempo de la iteración, con el objetivo de maximizar la utilidad del producto final y el retorno de inversión del cliente.

➤ **Inspección y adaptación**

El día de conclusión de la iteración se realiza una reunión de revisión de la iteración y se compone de dos partes:

- (1) **Revisión.** Dura aproximadamente hora y media, el equipo presenta al cliente una lista de las tareas completadas en la iteración, como también un producto listo para ser integrado en producción. En función de los resultados presentados y los cambios que se hayan hecho en el contexto del proyecto, el cliente realiza las integraciones o adaptaciones que sean necesarias.
- (2) **Retrospectiva.** Su duración es de aproximadamente hora y media, el equipo realiza un análisis de cómo ha sido forma de trabajar y los problemas que surgieron y a futuro pueden impedirles progresar en sus tareas, mejorando de manera continua su productividad. El líder de proyecto se encargará de suprimir o al menos reducir al máximo los obstáculos presentados que se encuentren más allá del accionar del equipo.

CAPÍTULO 2

Desarrollo del ciclo de vida de la aplicación

2.1. Toma de requerimientos.

En esta sección se especifican las tareas que el usuario desea automatizar y de qué forma quiere hacerlo a través de sus reglas de negocio, para lo cual se presenta a continuación la tabla que las especifica.

PRODUCT BACKLOG			
Prioridad	Historia de usuario	Puntos de historia	Tiempo estimado (h)
1	Selección de ruta de viaje	2	16
2	Selección de horarios de salida	1	8
3	Selección de asientos	3	24
4	Ventana de detalles de viaje	2	12

2.2. Planificación.

En el desarrollo de sistemas informáticos se cumplen varias etapas durante el ciclo de vida del software y la metodología de desarrollo es parte fundamental de éste. Se ha resuelto realizar el presente proyecto bajo las normativas establecidas por la metodología SCRUM, este modelo se compone de fases y una de ellas es la planificación, la cual precede a la creación del documento "Product Backlog", que recoge una serie de parámetros y requerimientos para el sistema. Seguidamente se definen roles y tareas para el desarrollo del proyecto y se establecen los integrantes del equipo de trabajo con las funciones a realizar.

2.2.1. Definición de roles y tareas.

ROL	DESCRIPCIÓN	RESPONSABILIDAD
Programador	Encargado de programar las funcionalidades de ventas de boletos necesarias en la aplicación.	Definir la estimación de tiempo para cada tarea planteada. Realizar el diseño de historias de usuario del sistema. Acudir periódicamente a reuniones sobre avances y funcionamiento del sistema.

		Informar y capacitar al usuario sobre el funcionamiento del sistema.
Usuario	Encargado de informar los procesos que ha de manejar la aplicación de Venta de boletos.	Facilitar los requerimientos principales del sistema. Solicitar una revisión periódica de los avances del sistema. Verificar el cumplimiento de los requisitos del sistema.
Tutor y Opositores	Encargados de brindar asesoría durante el proceso de desarrollo del sistema.	Realizar revisiones periódicas con el fin de verificar que todos los procesos se cumplan. Servir de testers para el aseguramiento de la calidad.

Tabla 2 - Definición de roles y tareas. (Fuente: Propia)

2.2.2. Definición de equipo de trabajo.

Es de vital importancia el socializar y documentar las funciones principales de las personas que conforman el equipo de trabajo por lo que se los detalla a continuación:

NOMBRE	DESCRIPCIÓN	RESPONSABILIDAD
Sr. Luis Carlos Ruales	Se encarga de desarrollar y entregar la aplicación al usuario final.	Programador y Tesista
Abg. Paul Boada.	Encargado de informar los procesos que ha de manejar la aplicación de Venta de boletos.	Facilitar los requerimientos principales del sistema. Solicitar una revisión periódica de los avances. Verificar el cumplimiento de los requisitos.
Ing. Víctor Caranqui. Msc.	Encargados de brindar asesoría durante el proceso de desarrollo del sistema.	Realizar revisiones periódicas con el fin de verificar que todos los procesos se cumplan.
Ing. Cosme Ortega. Msc.		Realizar las pruebas al sistema mediante un checklist.
Ing. Diego Trejo. Msc.		

Tabla 3 - Definición de equipo de trabajo. (Fuente: Propia)

2.2.3. Descripción de historias de usuario.

Una de las características importantes del marco de trabajo ágil Scrum son las Historias de Usuario, que se utilizan para definir y especificar los requisitos que debe tener el sistema, ya sean funcionales o no funcionales, en si describen de forma general las funciones que el cliente necesita en el software.

2.2.3.1. Historia de Usuario 1: Creación de base de datos bdboletos.

HU1 – CREACIÓN DE BASE DE DATOS “BDBOLETOS”	
Número: 1	Usuario: Abg. Paul Boada
Nombre de historia: Creación de base de datos bdboletos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Estimación (horas): 24	Iteraciones asignadas: 1
Programador responsable: Sr. Luis Carlos Ruales	
Descripción: Se debe diseñar la base de datos que almacenará y gestionará la información que genere la aplicación mediante un servicio web REST	
Observaciones: Deben implementarse los respectivos disparadores y procedimientos almacenados que aseguren la integridad de la información almacenada.	
Fecha:	
Firma:	

Tabla 4 - Historia de usuario 1 (Fuente: Propia)

2.2.3.2. Historia de Usuario 2: Desarrollo del servicio web.

HU2 – DESARROLLO DEL SERVICIO WEB	
Número: 2	Usuario: Abg. Paul Boada
Nombre de historia: Desarrollo del servicio web	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Estimación (horas): 24	Iteraciones asignadas: 1
Programador responsable: Sr. Luis Carlos Ruales	
Descripción: Se debe desarrollar un servicio web REST que pueda ser consumido por la aplicación móvil y así realizar las respectivas transacciones de lectura/escritura con la base de datos.	
Observaciones: Debe asignarse una ruta diferente por tabla en la programación del servicio web.	
Fecha:	
Firma:	

Tabla 5 - Historia de usuario 2 (Fuente: Propia)

2.2.3.3. Historia de Usuario 3: Subida de servicios a la nube.

HU3 – SUBIDA DE SERVICIOS A LA NUBE	
Número: 3	Usuario: Abg. Paul Boada
Nombre de historia: Subida de servicios a la nube	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Estimación (horas): 8	Iteraciones asignadas: 1
Programador responsable: Sr. Luis Carlos Ruales	
Descripción: El servicio web y la base de datos deben estar alojados en un servidor confiable en la nube para su posterior utilización.	
Observaciones: De preferencia subir el proyecto a GitHub y luego vincularlo a la plataforma Heroku por su característica de deploy automático cuando se realizan cambios. Por otra parte, la base de datos puede estar donde más convenga.	
Fecha:	
Firma:	

Tabla 6 - Historia de usuario 3 (Fuente: Propia)

2.2.3.4. Historia de Usuario 4: Selección de ruta de viaje.

HU4 - SELECCIÓN DE RUTA DE VIAJE	
Número: 4	Usuario: Abg. Paul Boada
Nombre de historia: Selección de ruta de viaje	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Estimación (horas): 16	Iteraciones asignadas: 1
Programador responsable: Sr. Luis Carlos Ruales	
Descripción: Se debe visualizar las rutas de viajes disponibles dando la posibilidad al usuario de elegir una ciudad de origen, ciudad destino, fecha de viaje y mostrar el costo de la ruta.	
Observaciones: Diseño de la ventana de ruta, donde se mostrará un campo de selección de ciudad de origen, posteriormente se cargarán las ciudades destino que tiene dicha ciudad origen, luego un selector de fecha y finalmente el costo que tiene la ruta elegida.	
Fecha:	
Firma:	

Tabla 7 - Historia de usuario 4 (Fuente: Propia)

2.2.3.5. Historia de Usuario 5: Selección de horarios de salida.

HU5 – SELECCIÓN DE HORARIOS DE SALIDA	
Número: 5	Usuario: Abg. Paul Boada
Nombre de historia: Selección de horarios de salida	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Estimación (horas): 8	Iteraciones asignadas: 1
Programador responsable: Sr. Luis Carlos Ruales	
Descripción: Creación de la ventana que mostrará los horarios de la ruta y fecha elegida anteriormente.	
Observaciones: Los horarios deben estar vinculados únicamente con la ruta elegida.	
Fecha:	
Firma:	

Tabla 8 - Historia de usuario 5 (Fuente: Propia)

2.2.3.6. Historia de Usuario 6: Selección de asientos.

HU6 – SELECCIÓN DE ASIENTOS	
Número: 6	Usuario: Abg. Paul Boada
Nombre de historia: Selección de asientos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Estimación (horas): 24	Iteraciones asignadas: 1
Programador responsable: Sr. Luis Carlos Ruales	
Descripción: Creación de la ventana que mostrará los asientos disponibles de forma gráfica, en el horario de viaje y ruta elegidos anteriormente.	
Observaciones: Solo se puede seleccionar un máximo de 5 asientos por boleto y pasado 5 minutos a partir de dicha selección, el(los) asiento(s) se deseleccionarán automáticamente.	
Fecha:	
Firma:	

Tabla 9 - Historia de usuario 6 (Fuente: Propia)

2.2.3.7. Historia de Usuario 7: Ventana de detalles de viaje.

HU7 – VENTANA DE DETALLES DE VIAJE	
Número: 7	Usuario: Abg. Paul Boada
Nombre de historia: Ventana de detalles de viaje	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Estimación (horas): 12	Iteraciones asignadas: 1
Programador responsable: Sr. Luis Carlos Ruales	
<p>Descripción: Creación de la ventana de detalles de viaje, que mostrará los datos del usuario, como son: nombres, apellidos, cédula, correo electrónico y desglosado el costo total del boleto, el cual contiene los valores a pagar de:</p> <ul style="list-style-type: none"> ➤ Precio del pasaje ➤ Comisión electrónica ➤ Comisión de tarjeta de débito/crédito <p>También tendrá un apartado para ingresar el número de cédula de la(s) persona(s) que viajan.</p>	
Observaciones: Dependiendo del número de asientos seleccionados, se creará el mismo número de campos para ingresar las cédulas de las personas que viajarán.	
Fecha:	
Firma:	

Tabla 10 - Historia de usuario 7 (Fuente: Propia)

2.3. Desarrollo.

2.3.1. Arquitectura del sistema.

El desarrollo de la aplicación móvil está realizado en lenguaje Java, utilizando librerías de envío de correo electrónico, manejadores de JSON y su respectivo consumo desde un servicio web (Desde ahora WS). La información que obtiene la aplicación (rutas, horarios y asientos disponibles) se obtiene del consumo de un WS RESTful desarrollado en el framework CodeIgniter de PHP, mismo que se encuentra subido a un servidor de aplicaciones en Heroku y a la vez, está conectado a una base de datos MySQL en un RDS de Amazon Web Service, el cuál sirve de almacén de datos de rutas, boletos, horarios, etc., información a ser consumida por la aplicación.



Fig. 11 - Arquitectura de la aplicación móvil (Fuente: Propia)

2.3.1. Desarrollo de las historias de usuario.

Para el desarrollo de las historias de usuario se detalla el proceso de construcción de cada una de ellas, haciendo énfasis en los requisitos detallados en la fase de planificación.

2.3.1.1. Historia de Usuario 1: Creación de base de datos bdboletos.

La estructura de la base de datos fue diseñada en el software de modelamiento Power Designer y luego exportado a lenguaje MySQL.

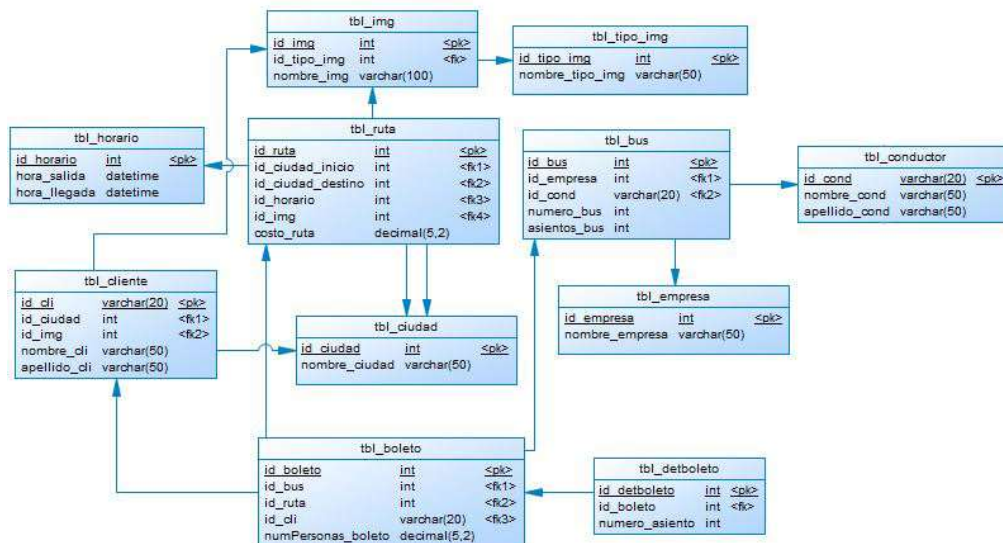


Fig. 12 – Gráfico del modelamiento de la base de datos en Power Designer (Fuente: Propia)

2.3.1.2. Historia de Usuario 2: Desarrollo del servicio web.

El WS tiene una arquitectura MVC, propia del framework del cual se lo desarrollo. Tiene diferentes rutas, cada una haciendo énfasis en los 4 pilares de información que maneja la aplicación como son boletos, ciudades, rutas y usuarios; cada uno con su respectiva función CRUD²⁶ y enlazados a la base de datos creada previamente.

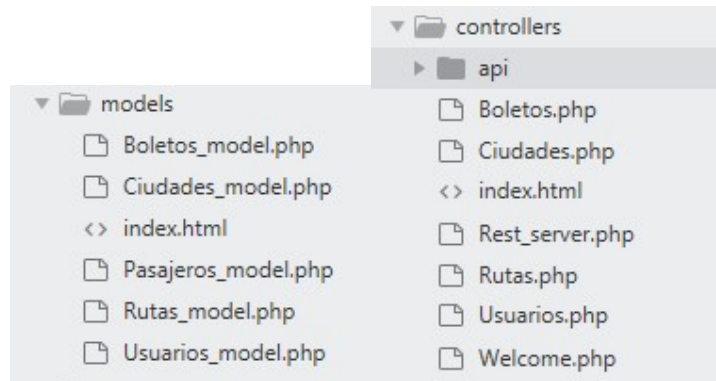


Fig. 13 - Estructura de clases del Servicio Web (Fuente: Propia)

2.3.1.3. Historia de Usuario 3: Subida de servicios a la nube.

Para el funcionamiento global de la aplicación se subió tanto la base de datos como el servicio web a plataformas gratuitas en la nube.

Base de datos:

Su estructura y contenido fue creada en el servicio RDS de Amazon Web Service, en donde se configuró las respectivas credenciales de acceso, reglas de seguridad y posteriormente su publicación en internet para usuarios autorizados.

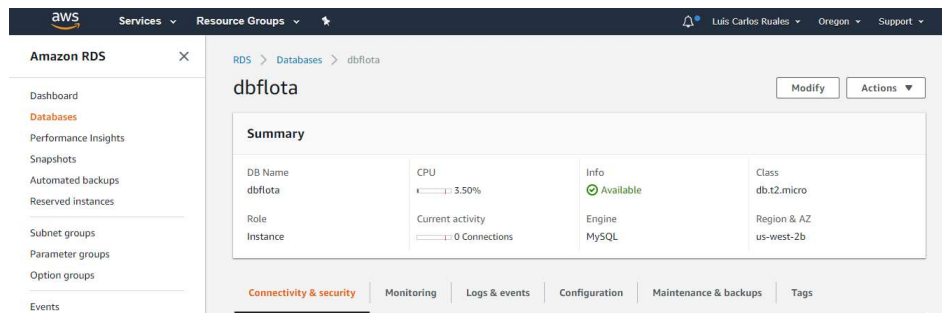


Fig. 14 - Consola de administración de la base de datos en Amazon Web Service (Fuente: Propia)

²⁶ **CRUD** Siglas de Create, Read, Update y Delete, en español Crear, Leer, Actualizar y Borrar

Servicio Web:

Para el deployment del servicio web, primero se subió el proyecto a un repositorio en GitHub, luego se vinculó dicho repositorio en un servicio basado en plataforma (Paas), desde donde se ejecuta y administra el mismo para su respectivo consumo desde cualquier parte del mundo.

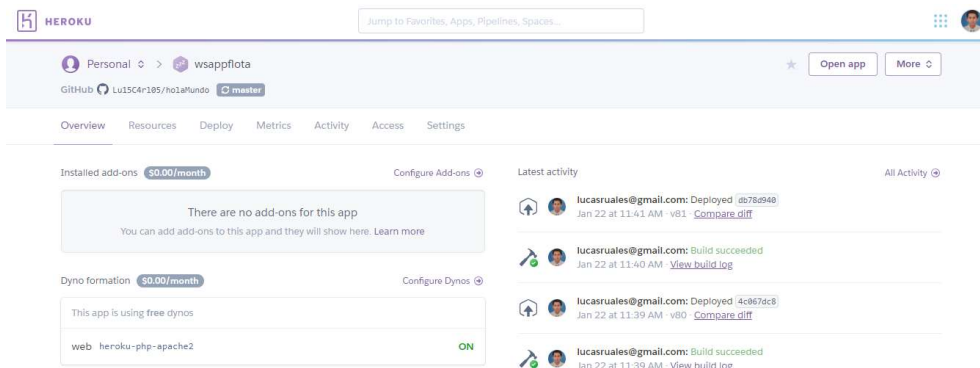


Fig. 15 – Consola de administración del servicio web en Heroku (Fuente: Propia)

2.3.1.4. Historia de Usuario 4: Selección de ruta de viaje.

El proceso de compra inicia con tres parámetros seleccionables que son ciudad origen, ciudad destino y fecha de viaje, después de haber seleccionado la ruta acorde al viaje, se mostrará el precio.

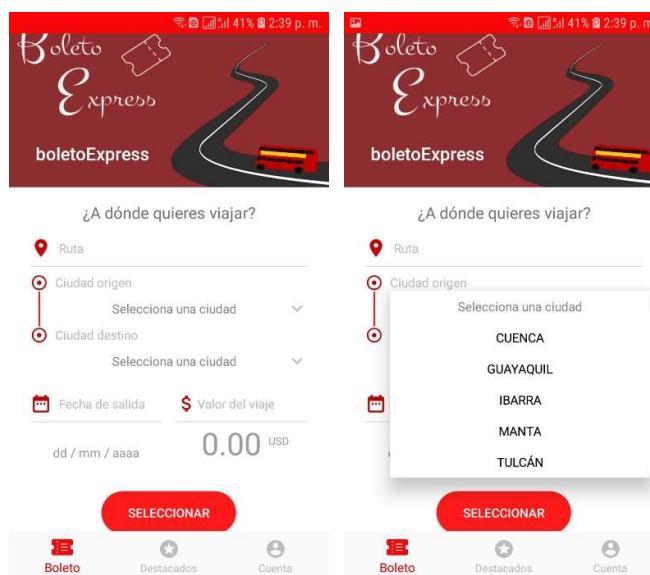


Fig. 16 - Pantalla de rutas. (Fuente: Propia)

Selección de rutas y fecha de viaje incompletos:

- ❖ Descripción
 - El usuario debe elegir la ruta en la que desea viajar mediante la ciudad de origen, destino y fecha.
- ❖ Entrada
 - El usuario no ingresa uno o más campos requeridos de la ruta.
- ❖ Resultado
 - Se muestra una alerta con el enunciado “**Asegúrese de seleccionar todas las opciones**”.



The screenshot shows a mobile application interface for 'boletoExpress'. At the top, there is a red banner with a white bell icon and the text 'AVISO' and 'Asegúrese de seleccionar todas las opciones'. Below this, the app's logo 'boletoExpress' is visible next to a graphic of a bus on a road. The main content area is titled '¿A dónde quieres viajar?' and contains a form with the following fields: 'Ruta' (with a location pin icon), 'Ciudad origen' (with a dropdown menu showing 'Selecciona una ciudad'), 'Ciudad destino' (with a dropdown menu showing 'Selecciona una ciudad'), 'Fecha de salida' (with a calendar icon and a date format 'dd / mm / aaaa'), and 'Valor del viaje' (with a dollar sign icon and a value of '0.00 USD'). A red button labeled 'SELECCIONAR' is positioned at the bottom of the form.

Fig. 17 - Pantalla de rutas con alerta de datos incompletos. (Fuente: Propia)

Selección de rutas y fecha de viaje completos:

- ❖ Descripción
 - El usuario debe elegir la ruta en la que desea viajar mediante la ciudad de origen, destino y fecha.
- ❖ Entrada
 - El usuario ingresa todos los campos requeridos de la ruta.
- ❖ Resultado
 - Se muestra la ventana de horarios de la ruta, en la fecha seleccionada.

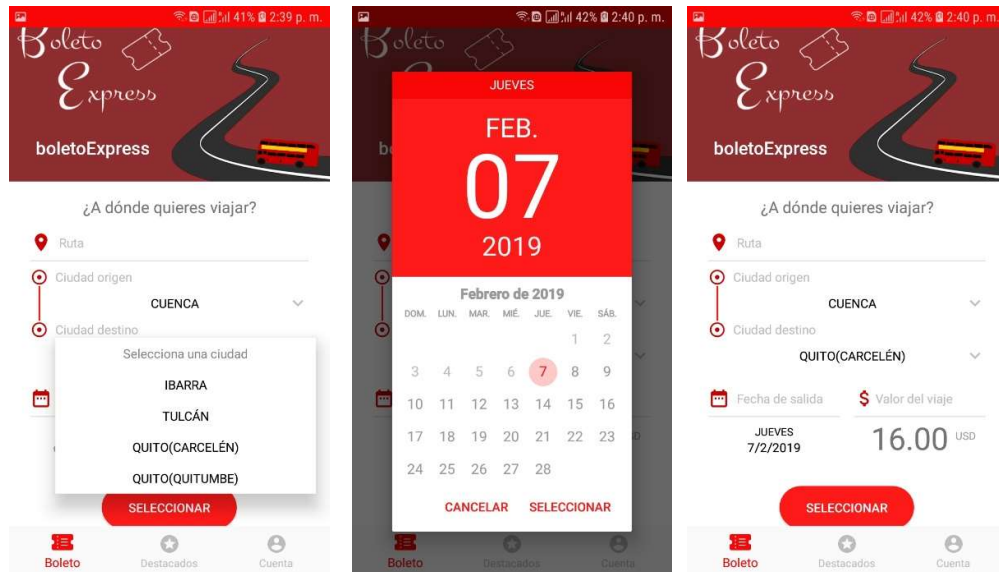


Fig. 18 – Asistente de compra de boletos-Sección de rutas. (Fuente: Propia)

2.3.1.5. Historia de Usuario 5: Selección de horarios de salida.

Una vez elegida la ruta se despliega una lista con los horarios de viaje disponibles para esa ruta.

Selección de horarios de la ruta especificada:

- ❖ Descripción
 - El usuario tiene a su disposición la información de horarios de la ruta y fecha seleccionados.
- ❖ Entrada
 - El usuario selecciona el horario en el que desea viajar.
- ❖ Resultado
 - Se muestra la ventana de horarios de la ruta, en la fecha seleccionada.

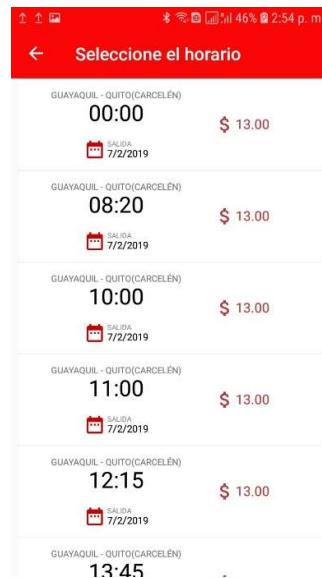


Fig. 19 - Asistente de compra de boletos- Sección de horarios. (Fuente: Propia)

2.3.1.6. Historia de Usuario 6: Selección de asientos.

Una vez elegido la hora, la aplicación muestra una ventana de selección de asientos.

No se selecciona ningún asiento:

- ❖ Descripción
 - El usuario debe elegir el asiento disponible de la ruta, fecha y hora elegida de la que desea viajar.
- ❖ Entrada
 - El usuario no selecciona ningún asiento.
- ❖ Resultado
 - Se muestra una alerta con el enunciado **“Asegúrese de seleccionar por lo menos un asiento”**.

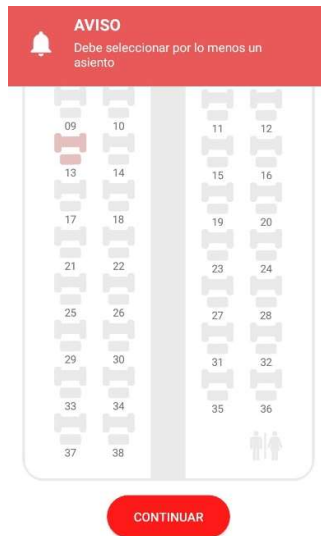


Fig. 20 - Asistente de compra de boletos-Sección de asientos con alerta. (Fuente: Propia)

Selección de uno a 5 asientos:

- ❖ Descripción
 - El usuario debe elegir el asiento disponible de la ruta, fecha y hora elegida de la que desea viajar.
- ❖ Entrada
 - El usuario selecciona de uno a cinco asientos.
- ❖ Resultado
 - Se muestra la ventana de detalle del viaje.

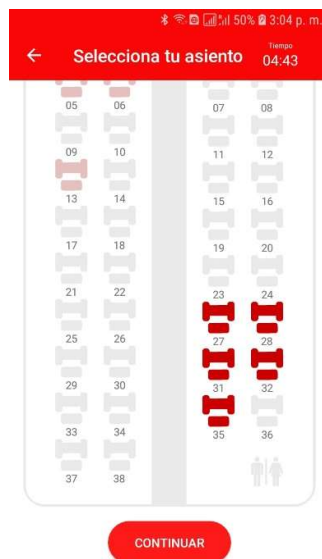


Fig. 21 - Asistente de compra de boletos-Sección de asientos. (Fuente: Propia)

Selección de más de 5 asientos:

- ❖ Descripción
 - El usuario debe elegir el asiento disponible de la ruta, fecha y hora elegida de la que desea viajar.
- ❖ Entrada
 - El usuario selecciona más de cinco asientos.
- ❖ Resultado
 - Se muestra una alerta con el enunciado **“No es posible seleccionar más de 5 asientos”**.

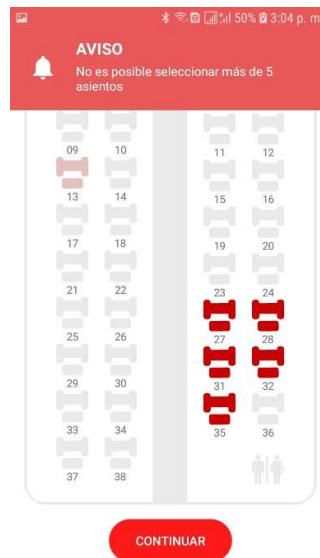


Fig. 22 - Asistente de compra de boletos-Sección de asientos. (Fuente: Propia)

2.3.1.6. Historia de Usuario 7: Ventana de detalles de viaje.

Después de haber elegido el o los asiento(s) de viaje, se mostrará la ventana de detalles de viaje, donde hay que ingresar el número de cédula de la(s) persona(s) que viajan.

Confirmación del detalle con cédula(s) en blanco:

- ❖ Descripción
 - El usuario debe verificar la información de viaje y seguidamente agregar el mismo de número de cédulas como asientos haya escogido para proceder con el pago.
- ❖ Entrada

- El usuario no ingresó la(s) cédula(s) de la(s) persona(s) que va(n) a viajar o no son válidas o están repetidas.
- ❖ Resultado
 - Se muestra una alerta de la ventana de detalle del viaje con el enunciado **“Una o más identificaciones de los pasajeros son inválidas o están repetidas, por favor verifíquelas”**.

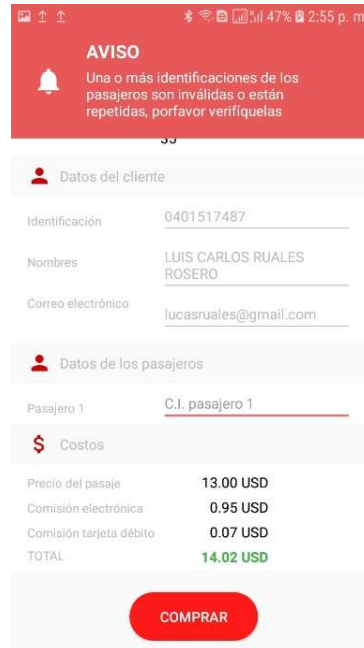


Fig. 23 - Asistente de compra de boletos- Sección de asientos. (Fuente: Propia)

Confirmación del detalle con campos de cédulas completas:

- ❖ Descripción
 - El usuario debe verificar la información de viaje y seguidamente agregar el mismo de número de cédulas como asientos haya escogido para proceder con el pago.
- ❖ Entrada
 - El usuario ingresó correctamente la(s) cédula(s) de la(s) persona(s) que va(n) a viajar.
- ❖ Resultado
 - Se envía un correo al usuario adjuntando su boleto y detalles de viaje.

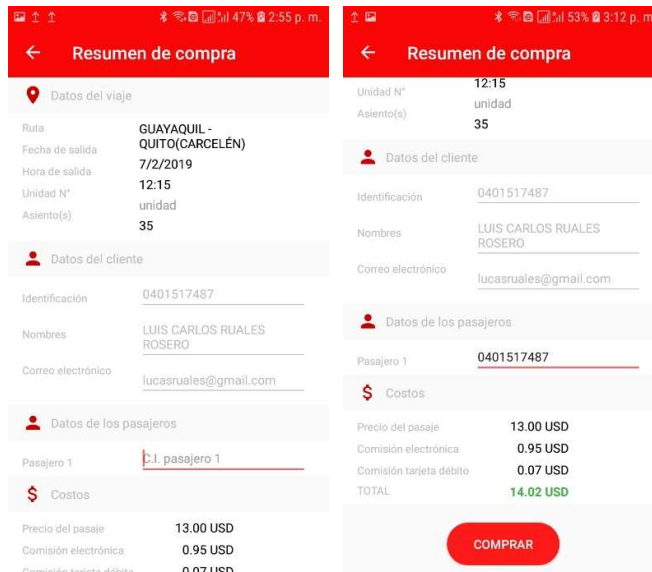


Fig. 24 - Asistente de compra de boletos-Sección de asientos. (Fuente: Propia)

2.4. Pruebas.

2.4.1. Pruebas contra Especificación (ESRE)

Las pruebas de especificaciones de requerimientos se realizan a partir de los requisitos, ya que éstos son importantes para verificar lo que en principio fue solicitado para funcionar en la aplicación móvil especificada en el capítulo anterior.

A continuación, se detalla un checklist del cumplimiento de las funcionalidades especificadas en los requerimientos, los resultados de dicho checklist fueron evaluadas por tres docentes, el tutor del trabajo de grado y los dos opositores.

MÉTRICAS	SI CUMPLE	OBSERVACIÓN
1. ¿Existe una ventana de rutas donde a partir de la ciudad de origen, se muestre el destino?	✓	Ninguna
2. ¿En la selección de fecha se puede elegir únicamente el lapso de un mes desde la fecha actual?	✓	Ninguna
3. ¿Existe una ventana de horarios de salida, vinculados a la ruta seleccionada?	✓	Ninguna
4. ¿Existe una ventana de selección de asientos?	✓	Ninguna
5. ¿Existe un temporizador de 5 minutos a partir de la selección de cualquier asiento?	✓	Ninguna

6. ¿La aplicación controla que, al terminar los 5 minutos a partir de la selección del asiento, este regrese al inicio del proceso (selección de ruta)?	✓	Ninguna
7. ¿La selección de asientos se limita a no más de 5 puestos por compra?	✓	Ninguna
8. ¿Existe una ventana que muestre los detalles (ruta, horario, asientos y costo detallado) del viaje seleccionado?	✓	Ninguna
9. ¿La ventana de detalles presenta un apartado donde se solicite las cédulas del mismo número de personas que van a viajar?	✓	Ninguna
10. Si la compra fue exitosa. ¿Existe el envío de un correo electrónico adjuntando el respectivo boleto?	✓	Ninguna

Tabla 11 - Métricas y cumplimiento de las pruebas contra especificación ESRE (Fuente: propia)

Los resultados de las pruebas realizadas reflejan una funcionalidad completa de los requerimientos dados para el desarrollo del proyecto, entendiéndose que se puede seguir con el proceso de pruebas funcionales, no funcionales, etc., lo cual no se efectuará en el presente proyecto.

CAPÍTULO 3

Análisis e interpretación de resultados

En este capítulo se realizaron procesos de orden, clasificación y presentación de resultados sobre la investigación en indicadores gráficos, elaborados y sistematizados en base a métodos estadísticos, para así hacerlos más comprensibles.

3.1. Hipótesis del trabajo

Al agilizar el tiempo del proceso de compra, la aplicación tendrá acogida inmediata por parte de la población que utiliza el servicio de transporte interprovincial, convirtiéndose en un instrumento de desarrollo social, económico y tecnológico dentro y fuera de la provincia.

3.2. Obtención de datos

La obtención de datos se realizó mediante una encuesta a 306 personas, dicho número representa el tamaño de la muestra con un nivel de confianza del 95% y un margen de error de 5% en una población de 1.500 personas, que es el número promedio mensual, de aquellos que compran un boleto por internet.

Para dicho cálculo se utilizó la fórmula que determina el tamaño de muestra de una población finita:

$$\text{Tamaño de muestra} = \frac{N * z^2 * p(1 - p)}{e^2 * (N - 1) + z^2 * p(1 - p)}$$

Donde:

N	Tamaño de la población
e	Margen de error (Porcentaje expresado en decimales)
z	Puntuación z
p	Probabilidad de que ocurra el evento estudiado (éxito) entre 0 y 1, en este caso 0.5

La puntuación z es la cantidad de desviaciones estándar que una proporción determinada se aleja de la media. Para encontrar la puntuación z adecuada, se consulta en la tabla a continuación:

Nivel de confianza deseado	Puntuación z
95%	1.96
99%	2.58

3.3. Manejo de datos

Siguiendo el proceso de interpretación de los resultados, se realiza un análisis descriptivo de la información obtenida.

3.3.1. Frecuencia de viaje

En una población de compradores de boletos de buses interprovinciales en Ecuador a través de internet, se obtuvieron las siguientes frecuencias de viaje con 306 personas:

Frecuencia	n_i	f_i
SIEMPRE	113	0.37
LA MAYORÍA DE VECES	61	0.2
POCAS VECES	127	0.41
NUNCA	5	0.02
TOTAL	306	1

Tabla 12 - Tabla de resultados de frecuencias de viaje (Fuente: Propia)

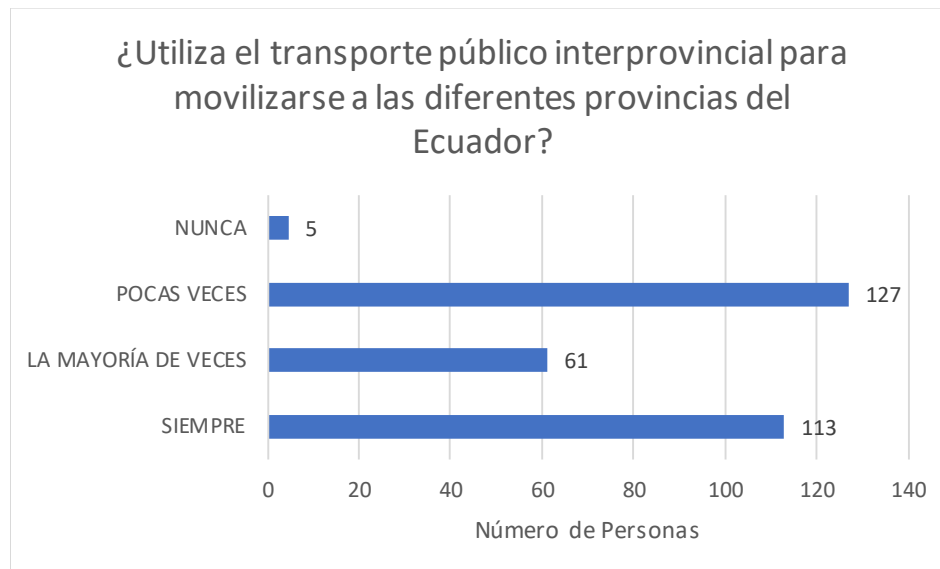


Fig. 25 – Representación gráfica de frecuencias de viaje de la muestra (Fuente: Propia)

En este punto se reducen 5 datos del tamaño de la muestra, manteniendo la información de las personas que de una u otra manera si se movilizan en transporte público interprovincial, que son 301.

3.3.2. Compra de boleto en condiciones normales

Filtrando al **72.09%** (217 personas) de las 301 que siempre compran un boleto en condiciones normales, se obtuvo la siguiente aproximación de tiempos:

Tiempo aproximado	n_i	f_i
MENOS DE 5 MINUTOS	38	0.18
ENTRE 5 Y 20 MINUTOS	142	0.65
ENTRE 20 Y 60 MINUTOS	28	0.13
MÁS DE UNA HORA	9	0.04
TOTAL	217	1

Tabla 13 - Tabla de resultados de tiempos aproximados de compra en condiciones normales (Fuente: Propia)

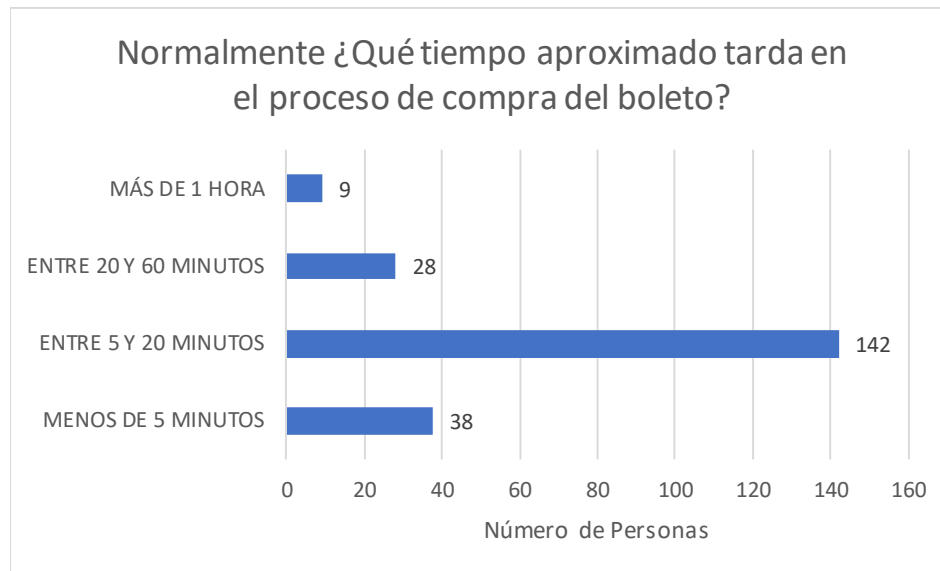


Fig. 26 - Representación gráfica de tiempos estimados de compra de boletos en condiciones normales (Fuente: propia)

3.2.3. Compra de boleto en feriados

Filtrando al **83.06%** (250 personas) de las 301, que siempre compran un boleto en condiciones normales, se obtuvo la siguiente aproximación de tiempos:

Tiempo aproximado	n_i	f_i
MENOS DE 5 MINUTOS	9	0.04
ENTRE 5 Y 20 MINUTOS	42	0.17

ENTRE 20 Y 60 MINUTOS	99	0.39
MÁS DE UNA HORA	100	0.4
TOTAL	250	1

Tabla 14 - Tabla de resultados de tiempos aproximados de compra en feriados

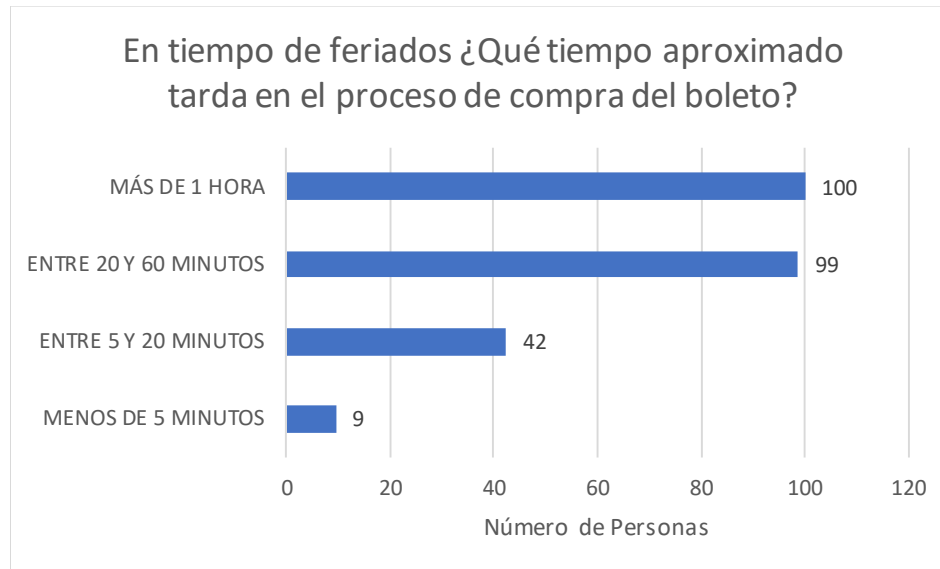


Fig. 27 - Representación gráfica de tiempos estimados de compra de boletos en feriados (Fuente: propia)

3.2.4. Acogida de la aplicación

Tomando una población de compradores de boletos de buses interprovinciales en Ecuador a través de internet, se determinó que el nivel de aceptación de la aplicación es el siguiente:

Aceptación	n_i	f_i
SÍ LA USARÍA	259	0.86
NO LA USARÍA	4	0.01
TAL VEZ LA USARÍA	38	0.13
TOTAL	301	1

Tabla 15 - Tabla de resultados de aceptación y uso de la aplicación (Fuente: propia)

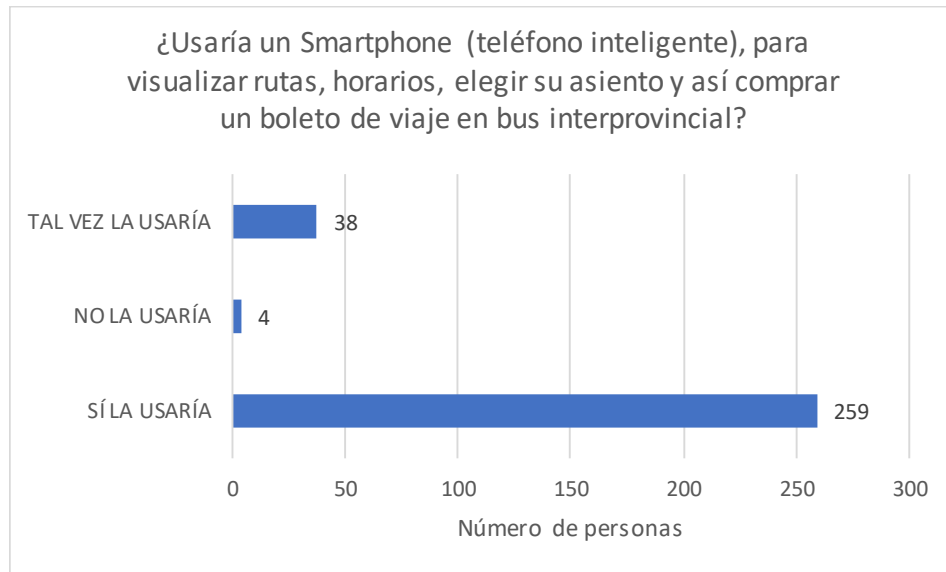


Fig. 28 - Representación gráfica de aceptación y uso de la aplicación (Fuente: propia)

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- ❖ El uso de la plataforma Android Studio en el desarrollo de aplicaciones móviles Android ha sido una herramienta muy útil y de gran aprendizaje, ya que contiene una serie de recursos y tecnologías que permiten a los usuarios explorar nuevos conocimientos.
- ❖ La metodología de desarrollo SCRUM, ha facilitado la elaboración del presente proyecto, gracias a las fases en las cuales está dividida, permitiendo cumplir con los tiempos establecidos.
- ❖ La optimización del proceso de compra de boletos, en cuanto a tiempo, mejoró al menos 5 veces más a diferencia de cuando se compra un boleto de forma tradicional.
- ❖ Al comprar un boleto desde la aplicación, la calidad de vida del usuario mejora al tener más tiempo libre, selección de asientos a su gusto y en ciertos casos ahorrar dinero.

RECOMENDACIONES

- ❖ Hay que desarrollar más aplicaciones móviles en el área de transporte, estando seguros que el uso de la tecnología avanzará y más servicios podrán ofertarse, dando así la posibilidad de crear un negocio rentable.
- ❖ Al momento de subir una base de datos o un servicio web a la nube, cerciorarse de implementar la seguridad de acceso y administración correspondientes, para evitar daños de personas malintencionadas.
- ❖ Se recomienda aplicar la metodología SCRUM asignando roles de test a los docentes tutores y/u opositores, con el fin de ver situaciones que el programador no tomaría en cuenta normalmente.

REFERENCIAS

Carballar, C. (2012, julio 12). ¿Qué son las aplicaciones móviles y para qué me sirven? Recuperado 13 de noviembre de 2018, de <http://blog.avansys.com.mx/2012/07/que-son-las-aplicaciones-moviles-y-para.html>

Conceptodefinicion.de. (2014, diciembre 17). Proceso. Recuperado 28 de noviembre de 2018, de <https://conceptodefinicion.de/proceso/>

D'Andrea Edgar. (2015). Arquitectura de Android | Edgar D'Andrea (.com). Recuperado 29 de noviembre de 2018, de <http://www.edgardandrea.com/arquitectura-de-android/>

EcuRed. (2018a). Metodologías de desarrollo de Software. Recuperado 29 de noviembre de 2018, de https://www.ecured.cu/Metodologias_de_desarrollo_de_Software

EcuRed. (2018b). Servidor Bases de Datos. Recuperado 29 de noviembre de 2018, de https://www.ecured.cu/Servidor_Bases_de_Datos

EcuRed. (2018c). Sistema Gestor de Base de Datos. Recuperado 29 de noviembre de 2018, de https://www.ecured.cu/Sistema_Gestor_de_Base_de_Datos

edX. (2018). ODS en la Agenda 2030 de las Naciones Unidas: Retos de los Objetivos de Desarrollo Sostenible. Recuperado de <https://www.edx.org/es/course/ods-en-la-agenda-2030-de-las-naciones-unidas-retos-de-los-objetivos-de-desarrollo-sostenible>

Flota Imbabura. (s. f.). Cooperativa de transportes - Flota Imbabura. Recuperado 6 de noviembre de 2018, de <http://flota-imabura.com/>

Francia Joel. (2017, agosto 17). Scrum no es una metodología es un Framework. Recuperado 3 de diciembre de 2018, de <https://www.scrum.org/resources/blog/scrum-no-es-una-metodologia-es-un-framework>

INEC. (2016). Tecnologías de la Información y Comunicaciones (TIC'S) 2016. Recuperado de http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2016/170125.Presentacion_Tics_2016.pdf

Kyocera. (2017, abril 7). Características avanzadas de un SGBD. Recuperado 29 de noviembre de 2018, de <https://smarterworkspaces.kyocera.es/blog/caracteristicas-avanzadas-sgbd/>

LaPrensa. (2015, mayo 26). El sistema Android de Google llega a los autos. Recuperado 28 de noviembre de 2018, de <https://www.laprensa.hn/economia/843748-410/el-sistema-android-de-google-llega-a-los-autos>

Lasso, I. (2017, abril 27). 7 plataformas diferentes para desarrollar Android Apps. Recuperado 27 de noviembre de 2018, de <https://tekzup.com/7-plataformas-diferentes-desarrollar-android-apps/>

Menéndez Rafael, A. B. (2000, enero 9). Servidores de bases de datos. Recuperado 29 de noviembre de 2018, de <https://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/sgbd.html>

Migues Pérez, M. (2011). *Técnicas de Venta*. (Ideaspropias Editorial, Ed.).

Neomobile. (2014, septiembre 9). Ecuador: Tendencias digitales y móviles. Recuperado 6 de noviembre de 2018, de <http://www.neomobile-blog.com/es/tendencias-digitales-movil-ecuador/>

Pacheco. (2017a, abril 9). Ingeniería de Software Procesos Ágiles - SCRUM. Recuperado 3 de diciembre de 2018, de <https://slideplayer.es/slide/3057200/>

Pacheco, J. (2017b, septiembre 22). ¿Qué es la optimización de procesos? Recuperado 28 de noviembre de 2018, de <https://www.heflo.com/es/blog/automatizacion-procesos/que-es-optimizacion-procesos/>

Setfree, L. (s. f.). Conociendo Xamarin, la herramienta para desarrolladores multiplataforma. Recuperado 27 de noviembre de 2018, de <https://www.vix.com/es/btg/tech/13263/conociendo-xamarin-la-herramienta-para-desarrolladores-multiplataforma>

Sol Vásquez. (2010). Tecnología y Medio Ambiente. Recuperado de <https://solvasquez.wordpress.com/2010/01/14/tecnologia-y-medio-ambiente/>

Stein Steve. (2017, marzo 13). Servidores vinculados (motor de base de datos). Recuperado 29 de noviembre de 2018, de <https://docs.microsoft.com/es-es/sql/relational-databases/linked-servers/linked-servers-database-engine>

Tatés Lenin. (2018). *IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL ANDROID PARA EL SEGUIMIENTO DE ASISTENCIA DE LOS ESTUDIANTES DE LA CISIC QUE REALIZAN VINCULACIÓN CON LA COLECTIVIDAD, UTILIZANDO LA PLATAFORMA ANDROID STUDIO*. Universidad Técnica del Norte, Ibarra.

Tomás J. (2016). *El Gran Libro de Android*. España: Marcombo.

Urbano, S. (2016, julio 31). Qué es y como funciona la pasarela de pago en ecommerce. Recuperado 6 de noviembre de 2018, de <https://www.actualidadecommerce.com/que-es-y-como-funciona-la-pasarela-de-pago-en-ecommerce/>

Venturi, B., Berteza, Borello, Castillo. (2017, septiembre 27). Android OS Documentation. Recuperado de <https://media.readthedocs.org/pdf/androidos/latest/androidos.pdf>

Wikipedia. (2018). Framework. En *Wikipedia, la enciclopedia libre*. Recuperado de <https://es.wikipedia.org/w/index.php?title=Framework&oldid=111644661>

ANEXOS



Fig. 29 - Entrega de código fuente, aplicación móvil y firma de ACTA ENTREGA RECEPCIÓN