

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS Y COMPUTACIÓN



**Pontificia Universidad
Católica del Ecuador**

**DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS Y COMPUTACIÓN**

TEMA

**“DESARROLLO DE UN PROTOTIPO DE UNA APLICACIÓN, BASADA EN
ANDROID QUE PERMITA AUTENTICAR EL TAXI Y SU CONDUCTOR.”**

AUTOR

LEONARDO DAVID PUNGACHO SANCHEZ

QUITO, 2017

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

AGRADECIMIENTOS

A Dios y a la virgen dolorosa por acompañarme, guiarme e iluminarme en todo este camino.

A mis padres Jorge y Yolanda gracias por todo su apoyo, paciencia y amor, gracias a ustedes fue posible todo lo que he conseguido.

A mis abuelitos que no pude conocer pero que siempre están mandándonos sus bendiciones desde el cielo. A papito miguel que fue parte de mi infancia y que genero recuerdos bonitos en mi memoria. A mamita melita por sus sabios consejos, palabras, anécdotas, juegos y chistes te extraño mucha melita. A mis hermanos Mary y Eduardo por apoyarme siempre y tenerme paciencia. A mi tío y padrino más querido Carlos Miguel gracias por tus consejos y todo tu cariño.

A Cata que siempre me apoyo desde el primer momento que nos conocimos.

Y un agradecimiento en especial por ser mi completa inspiración para seguir adelante en busca de un futuro mejor Dome, Mauri, Sophie, Amelita y a toda mi familia los amo mucho.

DEDICATORIAS

Este logro académico está dedicado a la virgencita dolorosa y a Dios que siempre estuvieron presentes en todo mi camino recorrido hasta aquí y estarán en el camino por recorrer.

A mis padres y toda mi familia, que siempre son mi inspiración.

A la Pontificia Universidad Católica del Ecuador y a todos los ingenieros que formaron parte de mi aprendizaje.

A mi herencia latina, Héctor Lavoe, Willie Colon, Tito Puente, Ray Barreto y a un sinnúmero de personajes que hicieron grande a la Salsa. A mi maestro Kike Gaitán que me enseñó que el querer es poder y que la perseverancia hace la diferencia. A mi amigo Pato Alomia que compartimos grandes momentos, logros y que siempre me brindó su apoyo cuando lo necesite. Y a mi querida Salsa por todos los valores que me enseñó y a la cual extraño mucho.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Contenido

AGRADECIMIENTOS.....	2
DEDICATORIAS.....	2
INTRODUCCION	12
1. ANÁLISIS DE REQUERIMIENTOS PARA AUTENTIFICAR UN TAXI Y SU CHOFER	13
1.1 MARCO TEORICO.....	13
1.1.1 APLICACIÓN MOVIL.....	13
1.1.2 ANDROID	14
1.1.3 APLICACIONES DISTRIBUIDAS.....	14
1.1.4 SERVICIO REST	15
1.1.5 MOBILE VISION.....	16
1.1.6 API	16
1.1.7 REGISTRO MUNICIPAL DE TAXIS.....	16
1.2 ANÁLISIS DE PARÁMETROS PARA REGISTRAR UN TAXI EN EL MUNICIPIO DE QUITO.....	17
1.3 ANÁLISIS DE PARÁMETROS PARA REGISTRA UN CONDUCTOR DE TAXI.....	19
2. METODOLOGIA Y HERRAMIENTAS	21
2.1. Metodología de programación Extreme Programming (XP).....	21
2.1.1. Introducción	21
2.1.2. Características Fundamentales	23
2.1.3 Roles	25

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

2.1.4 Ciclo de Vida	26
2.1.5 Ventajas y Desventajas.....	35
2.2. Lenguaje de programación C#.....	36
2.2.1. Introducción.	36
2.2.2. Características fundamentales.....	37
2.2.3. Ventajas y desventajas.	38
2.3. Gestor de base de datos SQL server	39
2.3.1. Introducción.	39
2.3.2. Características fundamentales.....	40
2.3.3. Ventajas y desventajas.	40
2.4. Herramienta de desarrollo Visual Studio	41
2.4.1. Introducción.	41
2.4.2. Características fundamentales.....	42
2.4.3. Ventajas y desventajas.	43
2.5. Herramienta de desarrollo Android Studio.....	44
2.5.1. Introducción.	44
2.5.2. Características fundamentales.....	44
2.5.3. Ventajas y desventajas.	45
3. APLICACIÓN DE LA METODOLOGIA PROGRAMACION EXTREMA.....	46
3.1. Iteración 1	46
3.1.1 Planificación	46
a) Historia de usuario	46

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

b) Planificación de lanzamiento	49
c) Plan de iteraciones	49
3.1.2. Diseño.....	50
a) Diseño simple	50
b) Metáforas del sistema	53
c) Tarjetas C.R.C	54
d) Funcionalidades extra	55
e) Refactorización.....	55
3.1.3. Desarrollo	56
a) Integración continua	56
3.1.4. Pruebas.....	59
a) Pruebas de aceptación	59
b) Pruebas unitarias	59
3.2. Iteración 2	60
3.2.1. Planificación	60
a) Historia de usuario	60
b) Planificación de lanzamiento	60
c) Plan de iteraciones	61
3.2.2. Diseño.....	61
a) Diseño simple	61
b) Metáfora del sistema	63
c) Tarjetas C.R.C.	63

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

3.2.3. Desarrollo	65
a) Integración continua	65
3.2.4. Pruebas.....	68
a) Pruebas de aceptación	68
b) Pruebas unitarias.....	68
3.3. Iteración 3	71
3.3.1. Planificación	71
a) Historia de usuario	71
b) Planificación de lanzamiento	71
c) Plan de iteraciones	72
3.3.2. Diseño.....	72
a) Diseño simple	72
b) Metáfora del sistema	73
c) Tarjetas C.R.C.	74
3.3.3. Desarrollo	76
a) Integración continua	76
3.3.4. Pruebas.....	77
a) Pruebas de aceptación	77
b) Pruebas unitarias.....	78
3.4. Iteración 4	79
3.4.1. Planificación	79
a) Historia de usuario	79

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

b) Planificación de lanzamiento	79
c) Plan de iteraciones	80
3.4.2. Diseño.....	80
a) Diseño simple	80
b) Metáfora del sistema	81
c) Tarjetas C.R.C.	82
3.4.3. Desarrollo	85
a) Integración continua	85
3.4.4. Pruebas.....	85
a) Pruebas de aceptación	85
b) Pruebas unitarias.....	86
4. Conclusiones y Recomendaciones	87
4.1. Conclusiones.....	87
4.2. Recomendaciones	88
5. Bibliografía.....	89
Sitios Web.....	89
Videos.....	90
Libros.....	90
6. Anexos.....	91

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Contenido de Ilustraciones

<i>Ilustración 1. Comportamiento General del Sistema</i>	12
<i>Ilustración 2. Registro Municipal</i>	18
<i>Ilustración 3. Registro municipal en taxi</i>	19
<i>Ilustración 4. Ciclo de vida en v</i>	22
<i>Ilustración 5. Programación Extrema</i>	22
<i>Ilustración 6. Ciclo de vida XP</i>	27
<i>Ilustración 7. Ciclo de vida XP explicado</i>	27
<i>Ilustración 8. Pruebas de Aceptación</i>	34
<i>Ilustración 9. C#</i>	36
<i>Ilustración 10. Common Language Infrastructure (CLI)</i>	37
<i>Ilustración 11. Sql Server 2014</i>	39
<i>Ilustración 12. Visual Studio 2012</i>	41
<i>Ilustración 13. Modelo Vista Controlador</i>	42
<i>Ilustración 14. Android Studio 2.2</i>	44
<i>Ilustración 15. Historia de Usuario</i>	50
<i>Ilustración 16. Diseño Conceptual BD</i>	51
<i>Ilustración 17. Diseño Lógico BD</i>	51
<i>Ilustración 18. Diseño Físico BD</i>	52
<i>Ilustración 19. Metáfora del sistema BD</i>	53
<i>Ilustración 20. Metáfora del Sistema General BD</i>	53
<i>Ilustración 21. Creando una base</i>	56
<i>Ilustración 22. Corriendo el modelo físico</i>	57
<i>Ilustración 23. Verificando la BD</i>	57
<i>Ilustración 24. Tabla conductor</i>	58
<i>Ilustración 25. Tabla cooperativa</i>	58
<i>Ilustración 26. Tabla vehículo</i>	58

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

<i>Ilustración 27. Iteración 2.....</i>	<i>61</i>
<i>Ilustración 28. Arquitectura 3 capas.....</i>	<i>62</i>
<i>Ilustración 29. Modelo Vista Controlador.....</i>	<i>62</i>
<i>Ilustración 30. Metáfora del Sistema Servicio Rest</i>	<i>63</i>
<i>Ilustración 31. Conexión al modelo de datos.....</i>	<i>65</i>
<i>Ilustración 32. Modelo de datos como clases.....</i>	<i>66</i>
<i>Ilustración 33. Capa de datos</i>	<i>66</i>
<i>Ilustración 34. Capa lógica delo negocio</i>	<i>67</i>
<i>Ilustración 35. Servicio Rest.....</i>	<i>67</i>
<i>Ilustración 36. Pruebas unitarias</i>	<i>69</i>
<i>Ilustración 37. Iteración 3.....</i>	<i>72</i>
<i>Ilustración 38. Diseño interfaz Android</i>	<i>73</i>
<i>Ilustración 39. Metáfora del sistema Cliente Android</i>	<i>73</i>
<i>Ilustración 40. Prueba unitaria Android</i>	<i>78</i>
<i>Ilustración 41. Iteración 4.....</i>	<i>80</i>
<i>Ilustración 42. Diseño interfaz Android OCR.....</i>	<i>81</i>
<i>Ilustración 43. Metáfora del sistema Cliente Android OCR</i>	<i>82</i>

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Contenido de Tablas

<i>Tabla 1. Historia de usuario</i>	28
<i>Tabla 2. • Planificación de lanzamiento</i>	29
<i>Tabla 3. Plan de iteraciones</i>	29
<i>Tabla 4. Tarjeta CRC</i>	31
<i>Tabla 5. Funcionalidad extra</i>	31
<i>Tabla 6. Prueba de aceptación</i>	34
<i>Tabla 7. Historia de usuario 1</i>	46
<i>Tabla 8. Historia de usuario 2</i>	47
<i>Tabla 9. Historia de usuario 3</i>	47
<i>Tabla 10. Historia de usuario 4</i>	48
<i>Tabla 11. Historia de usuario 5</i>	48
<i>Tabla 12. Plan de lanzamiento Iteración 1</i>	49
<i>Tabla 13. Plan de iteraciones Iteración 1</i>	49
<i>Tabla 14. Tarjeta CRC1 Iteración 1</i>	54
<i>Tabla 15. Tarjeta CRC2 Iteración 1</i>	54
<i>Tabla 16. Tarjeta CRC3 Iteración 1</i>	55
<i>Tabla 17. Funcionalidad extra Iteración 1</i>	55
<i>Tabla 18. Prueba de aceptación Iteración 1</i>	59
<i>Tabla 19. Historia de usuario 3 Iteración 2</i>	60
<i>Tabla 20. Planificación de lanzamiento Iteración 2</i>	60
<i>Tabla 21. Plan de iteraciones Iteración 2</i>	61
<i>Tabla 22. Tarjeta CRC1 Iteración 2</i>	63
<i>Tabla 23. Tarjeta CRC2 Iteración 2</i>	64
<i>Tabla 24. Tarjeta CRC3 Iteración 2</i>	64
<i>Tabla 25. Funcionalidad extra Iteración 2</i>	65

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

<i>Tabla 26. Prueba de aceptación Iteración 2</i>	<i>68</i>
<i>Tabla 27. Prueba unitaria 1 Iteración 2</i>	<i>69</i>
<i>Tabla 28. Prueba unitaria 2 Iteración 2</i>	<i>70</i>
<i>Tabla 29. Historia de usuario 4 Iteración 3.....</i>	<i>71</i>
<i>Tabla 30. Planificación de lanzamiento Iteración 3.....</i>	<i>71</i>
<i>Tabla 31. Plan de iteraciones Iteración 3.....</i>	<i>72</i>
<i>Tabla 32. Tarjeta CRC1 Iteración 3</i>	<i>74</i>
<i>Tabla 33. Tarjeta CRC2 Iteración 3</i>	<i>74</i>
<i>Tabla 34. Tarjeta CRC3 Iteración 3</i>	<i>75</i>
<i>Tabla 35. Tarjeta CRC4 Iteración 3</i>	<i>75</i>
<i>Tabla 36. Funcionalidad extra Iteración 3</i>	<i>76</i>
<i>Tabla 37. Pruebas de aceptación Iteración 3.....</i>	<i>77</i>
<i>Tabla 38. Historia de Usuario Iteración 4</i>	<i>79</i>
<i>Tabla 39. Planificación de lanzamiento Iteración 4.....</i>	<i>79</i>
<i>Tabla 40. Plan de iteraciones Iteración 4.....</i>	<i>80</i>
<i>Tabla 41. Tarjeta CRC1 Iteración 4</i>	<i>82</i>
<i>Tabla 42. Tarjeta CRC2 Iteración 4</i>	<i>83</i>
<i>Tabla 43. Tarjeta CRC3 Iteración 4</i>	<i>84</i>
<i>Tabla 44. Funcionalidad extra Iteración 4</i>	<i>84</i>
<i>Tabla 45. Prueba de aceptación Iteración 4</i>	<i>86</i>
<i>Tabla 46. Prueba unitaria Iteración 4.....</i>	<i>86</i>

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

INTRODUCCION

Hoy en día, en nuestra sociedad, el método más común de movilización de la gente es mediante transporte público, ya sea en buses o por unidades de taxi, a cualquier hora del día o de la noche. El transporte público en especial las cooperativas de taxis de la ciudad de Quito están controladas bajo el registro municipal de la ciudad.

El presente trabajo, pretende brindar seguridad a los usuarios de los taxis para lo cual se desarrollará un prototipo que permita validar que el chofer del taxi es quien dice ser y que el taxi sea el que está registrado por las autoridades pertinentes.

Esta verificación se la realizará mediante el uso del código único de registro municipal del taxi o por la placa del taxi, incluyendo el reconocimiento de caracteres, mediante la detección del registro municipal o de la placa con la cámara de los dispositivos Android y como resultado devolverá la información importante relacionada al vehículo y al conductor.

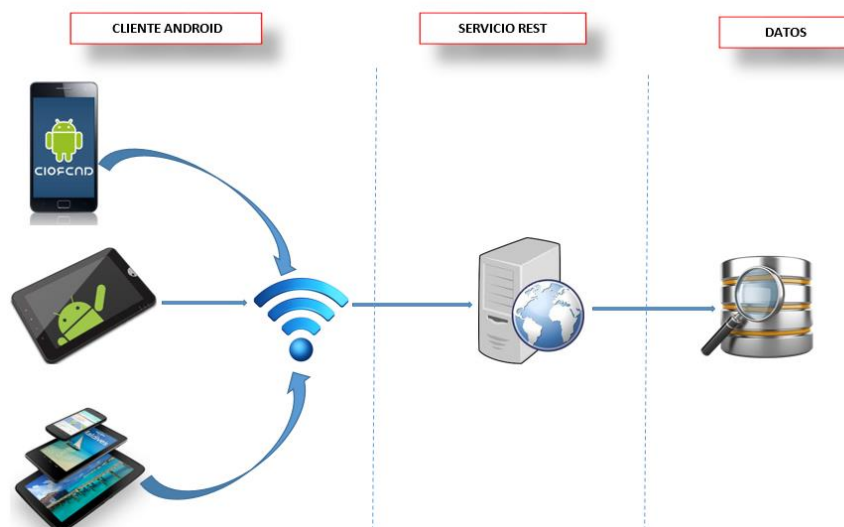


Ilustración 1. Comportamiento General del Sistema

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

1. ANÁLISIS DE REQUERIMIENTOS PARA AUTENTICAR UN TAXI Y SU CHOFER

1.1 MARCO TEORICO

1.1.1 APLICACIÓN MOVIL

En la actualidad la tecnología ha ido avanzando a pasos agigantados, facilitando la creación de un sinfín de herramientas que nos ayudan a mejorar nuestras vidas, es aquí donde entran las aplicaciones móviles o app, estas aplicaciones informáticas tienen el fin de realizar uno o varios tipos de tareas, estos pueden ser: informar, entretener, organizar, comunicar, etc. Las apps están diseñadas para poder ser ejecutadas en dispositivos de última generación, como los son: teléfonos inteligentes, tablets, entre otros dispositivos móviles.

En el mundo de las aplicaciones móviles existen una gran gama de opciones, limitándose a la capacidad del dispositivo móvil que la va a ejecutar, ya que puede depender del tamaño de la pantalla, la resolución de la pantalla, la capacidad de procesamiento, la capacidad de almacenamiento del dispositivo, entre otras, es así que existen aplicaciones que hacen uso de geolocalización, lectores de código QR, comandos de voz, etc.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

1.1.2 ANDROID

Es un sistema operativo basado en Linux, es software libre, multiplataforma y de código abierto. Usa dos tipos de máquinas virtuales: Dalvik o Android Runtime, que son variaciones de la JVM (Java Virtual Machine).

Android Inc. Fue fundada en 2003 y adquirida por Google en 2005. El 5 de noviembre de 2007 se anuncia la primera versión de Android: Android 1.0 Apple Pie. La versión actual es Android 5.0 Lollipop.

Es el sistema operativo para móviles más exitoso del mundo, superando a sus competidores IOS y Windows Phone.

1.1.3 APLICACIONES DISTRIBUIDAS

Hoy en día, debido a que la mayoría de personas se encuentran conectadas a internet, muchas veces desde más de un dispositivo, se ha hecho común el uso de aplicaciones distribuidas.

Una aplicación distribuida es aquella en la que sus componentes están localizados en una red de computadores. Estos componentes, que tienen un objetivo en común, se comunican entre sí, coordinando sus acciones mediante el envío de mensajes.

El modelo más utilizado dentro de las aplicaciones distribuidas es el de cliente/servidor. El equipo cliente hace pedidos a un servidor. El servidor por lo general es una máquina muy potente que recibe y gestiona los pedidos de los equipos cliente y les brinda servicios.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Una aplicación distribuida que sigue el modelo antes mencionado tiene los siguientes componentes:

- ✓ Lado servidor. Es software que se ejecuta en el servidor conectado a la red y responde a las peticiones de los clientes. Estas peticiones se manejan mediante puertos (un servidor web escucha en el puerto 80).
- ✓ Lado cliente. Es software que ejecuta el usuario final. Este hace peticiones al servidor a través de la red. Un ejemplo común es el navegador web.
- ✓ Protocolo de aplicación. Define el tipo de los mensajes que se manejan entre el cliente y el servidor. Por ejemplo, HTTP (Hypertext Transfer Protocol) es el protocolo de aplicación entre el navegador y el servidor web.

Algunos ejemplos de aplicaciones distribuidas son correo electrónico, streaming, navegación web, telefonía IP, entre otras.

1.1.4 SERVICIO REST

El servicio Rest (REpresentational State Transfer) es una arquitectura de servicios web distribuidos, diseñada para desarrollar servicios. Fue introducida en el año 2000, en la tesis doctoral de Roy Fielding, uno de los gestores del protocolo HTTP.

Existen ciertos requisitos para que una arquitectura de servicios distribuidos sea catalogada como REST: debe ser cliente/servidor, sin estado, debe poder apoyarse en la cache, sus servicios son uniformes y por último su arquitectura ha de ser en capas.

Algunas empresas que han utilizado REST para sus aplicaciones son: Amazon, eBay, Microsoft, Yahoo! Entre otras.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

1.1.5 MOBILE VISION

Mobile visión es un api de código abierto desarrollado por Google liberada en agosto del 2015, este nuevo servicio de google está dedicado al trabajo con reconocimiento de imágenes, que mediante el uso de detectores en videos y fotografías se puede localizar objetos y manipularlos. El api contiene funcionalidades para: detección de rostros, escaneo de código de barras y reconocimiento de texto.

1.1.6 API

Las API (Application Programming Interface) son código de programas que trabajan como interfaces, permitiendo y facilitando la interacción y la comunicación entre diversas aplicaciones, tienen el mismo objetivo que una interfaz de usuario, que es hacer amigable la comunicación entre humano y software. Las apis más usadas en el mundo de desarrollo de aplicaciones para dispositivos móviles son: Google Maps, Google play services, Google Apps Activity, entre otras.

1.1.7 REGISTRO MUNICIPAL DE TAXIS

En la ciudad de Quito, los taxis legales cuentan con un distintivo denominado Registro Municipal de Transporte Comercial, que consta de un número de cuatro dígitos y es otorgado por la AMT (Agencia Metropolitana de Tránsito). Este distintivo tiene la forma de un adhesivo verde adherido a las puertas, el parabrisas y a la ventana posterior del automotor.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

1.2 ANÁLISIS DE PARÁMETROS PARA REGISTRAR UN TAXI EN EL MUNICIPIO DE QUITO.

Considerando que en el artículo 264, numeral 6, de la Constitución de la República del Ecuador se considera como obligación exclusiva de los gobiernos municipales: *“Planificar, regular y controlar el tránsito y el transporte terrestre dentro de su territorio cantonal”* (Quito E. c., Ordenanza municipal No 0047, 2011, pág. 1).

Se considera también que en el numeral 2 del artículo 2 de la Ley Orgánica de Régimen para el Distrito Metropolitano de Quito, se determina que es obligación *“expedir las normas que sean necesarias para planificar, regular y coordinar todo lo relacionado con el transporte público y privado dentro de su jurisdicción”* (Quito E. c., Ordenanza municipal No 0047, 2011, pág. 1).

Según el artículo 18 numeral 2 de la Ordenanza Municipal, el Municipio de Quito es quien se encarga de calificar, aprobar y emitir el registro de Auto Taxi, ya que sin este registro ninguna operadora podrá ejercer el servicio de taxi, esta calificación de auto taxi solo será emitida al titular del vehículo que va a ser destinado como taxi, o a su vez al conductor/a que fuese contratado para conducir el mismo.

La ordenanza municipal del servicio de taxi en el distrito metropolitano de quito clasifica al servicio de taxis en dos tipos de clases: Convencional y Ejecutivo.

Convencional

Es el servicio público de transporte mediante el uso de taxi que se brinda al usuario en general en las zonas urbanas del distrito metropolitano de quito, este servicio puede ser tomado en la

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

vía pública de la ciudad, pero solo en las paradas establecidas para el uso de transporte público o en el centro de operaciones del taxi a ser usado.

Ejecutivo

Es el servicio público de transporte mediante el uso de taxi que se brinda al usuario en general en las zonas urbanas del distrito metropolitano de quito o a las afueras del mismo, este servicio debe ser solicitado únicamente mediante un centro de operaciones (llamada telefónica).

La Agencia Metropolitana de Transito es la encargada de administrar la regulación del transporte tipo comercial, entre otros, que mediante una revisión técnica del vehículo y previamente cumpliendo todos los requisitos necesarios establecidos en la ordenanza municipal, es la encargada de emitir el sello del Registro Municipal de Transporte Comercial, que para el caso de los taxis es de color verde y se debe encontrar pegado en la ventana posterior, las dos puertas delanteras y en el parabrisas del vehículo que va a funcionar como unidad de taxi.



Ilustración 2.Registro Municipal

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.



Ilustración 3. Registro municipal en taxi

El objetivo de los artículos de ley expuestos anteriormente es aclarar el panorama sobre el cual los taxis operan en la ciudad, para poder entender que autoridades rigen control sobre los mismos y poder obtener los requerimientos adecuados que garanticen el desarrollo del prototipo con total veracidad sobre los datos obtenidos del chofer y del vehículo.

1.3 ANÁLISIS DE PARÁMETROS PARA REGISTRA UN CONDUCTOR DE TAXI.

Otro factor importante para la construcción de este prototipo es quien va a operar el taxi, por eso vamos a analizar los requisitos fundamentales para poder encontrar y determinar los parámetros que nos van a ayudar en el desarrollo del prototipo.

La agencia nacional de tránsito del Ecuador es la encargada de gestionar todas las licencias de conducción, entre la cual se encuentra la licencia tipo c que es para taxis ejecutivos o convencionales, esta agencia establece una serie de requisitos para la obtención de la licencia tipo c entre los cuales vamos a destacar los más importantes que son:

- ✓ Ser mayor de edad.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

- ✓ Haber aprobado el primer año de bachillerato.
- ✓ Haber aprobado un curso de conductor profesional, dentro de una escuela de conducción de choferes profesionales legalmente autorizadas.
- ✓ Aprobar las evaluaciones que se deben rendir en la ANT.

Una vez cumplidos todos los requerimientos establecidos por la ANT el aspirante a obtenido la licencia profesional tipo c.

El paso siguiente a la obtención de la licencia profesional tipo c es cumplir con los requisitos impuestos por la ordenanza municipal, a continuación, se listará los requisitos más destacados para el fin de la construcción del prototipo.

- ✓ Presentar la licencia profesional tipo c.
- ✓ Contar con el Sistema Público Para Pago de Accidentes de Tránsito(SPPAT).
- ✓ La matrícula del vehículo actualizada al año correspondiente.
- ✓ Calificación de Auto Taxi.
- ✓ Permiso de Operación.

Una vez analizados los parámetros necesarios para registrar un vehículo como taxi y a una persona como conductor de este vehículo, llegamos a la conclusión que el registro municipal y la placa del vehículo, nos van a brindar la información necesaria de manera confiable ya que esta información está regulada por dos organismos de control que son el Municipio de Quito y la Agencia Nacional de Tránsito.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

2. METODOLOGIA Y HERRAMIENTAS

2.1. Metodología de programación Extreme Programming (XP)

2.1.1. Introducción

En el mundo del desarrollo de software ha pasado por varias etapas de mejora y de constante evolución. En sus inicios por los años 50 el desarrollo de software se basaba en inspiraciones personales aplicando “Programa y corrige” que principalmente su filosofía se basaba en ensayo y error. A mediados de los años 60 aparecen nuevas necesidades en el desarrollo de software, como son los sistemas multiusuario y la multiprogramación, sin embargo, la industria del desarrollo de software no pudo cubrir estas necesidades y se dio inicio a la gran crisis en la industria del software, debido al incumplimiento de contratos, a la pésima calidad del producto, al desperdicio de recursos, retraso en las entregas, etc. Como solución a esta crisis la OTAN (La Organización del Tratado del Atlántico Norte) crea el término de Ingeniería de Software con el objetivo de crear una formalidad en la construcción de software aplicando técnicas y metodologías ingenieriles, a partir de este momento la industria del software experimento una gran mejora, introduciendo los ciclos de vida de desarrollo de software, entre los más conocidos están el ciclo en cascada, en v, es espiral, etc.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Ciclo en V

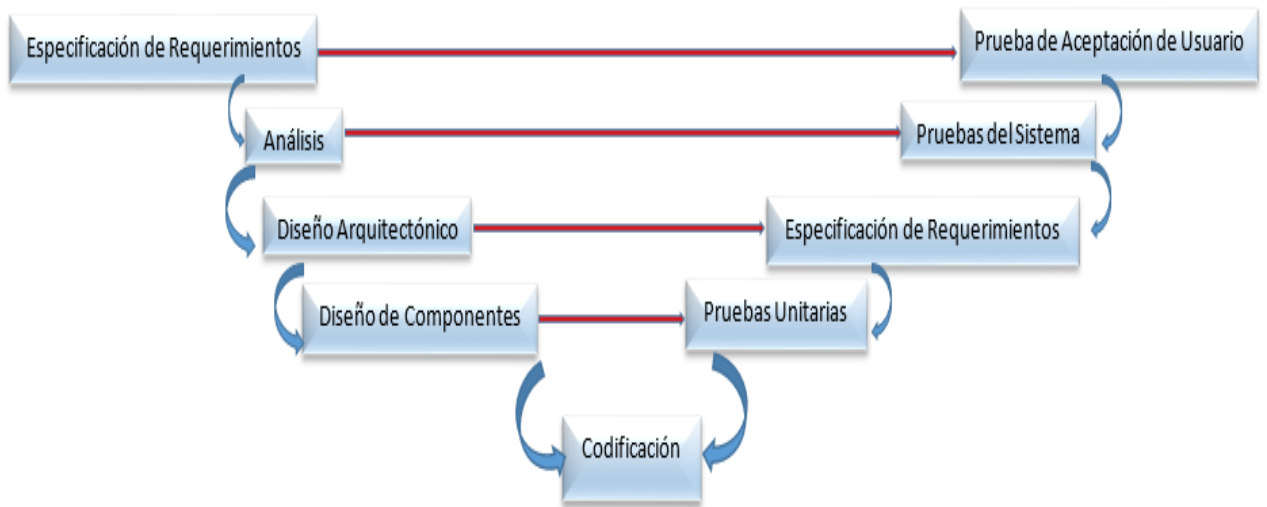


Ilustración 4. Ciclo de vida en v

Actualmente la tecnología ha ido creciendo a pasos agigantados, creando nuevos equipos con características cada vez más amplias, como lo son los dispositivos móviles, esto también ha creado una revolución en la industria del desarrollo de software ya que la creación de software para dispositivos móviles no se adapta a las técnicas y metodologías que anteriormente mencionamos debido a las características del entorno como lo son: un ambiente dinámico (según PlayStore de Android para finales del 2016 existen alrededor de 14000 diferentes modelos de dispositivos usando Android), la programación se pueda dar en parejas o individualmente, entre otras.



Ilustración 5. Programación Extrema

Extreme Programming es una metodología ágil de programación creada por Kent Beck en 1999. La principal característica y razón por la cual se escogió esta metodología para el desarrollo del prototipo es por su filosofía de poner énfasis en la **adaptabilidad al cambio** que,

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

en el manejo de la previsibilidad, esto quiere decir que esta metodología nos ayuda a enfrentar los cambios de requerimientos que se pueden dar sobre la marcha, debido a que estamos en un medio constantemente cambiante, como lo son los dispositivos móviles.

2.1.2. Características Fundamentales

La programación extrema considera esencial 5 valores para poder realizar un producto de calidad.

Simplicidad

Este valor apunta directamente a la ideología de adaptabilidad al cambio, se busca simplificar el diseño, con el objetivo de reducir la documentación para facilitar el mantenimiento o algún cambio sobre el mismo, para lograr la reducción de documentación se sugiere comentar el código, hacerlo más entendible, usar estándares de programación para el nombramiento de variables, métodos, clases, etc.

Comunicación

Este valor hace referencia a la auto-documentación del código, debe ser clara y precisa, se debe adquirir una técnica que me permita hacerme entender con el resto del equipo, o si la misma persona desea realizar algún cambio, debe entender el código que fue escrito hace algún tiempo atrás.

La comunicación también la encontramos al realizar pruebas unitarias de lo que vamos construyendo.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Retroalimentación

El objetivo principal al establecer este valor es crear ciclos cortos, ya que esto me permite controlar los resultados que me arroja cada ciclo y poder verificar si estos resultados se acercan a las necesidades del cliente y si no lo hacen, me permite enfocarme en las partes más importantes para poder lograrlo, sin desperdiciar una gran cantidad de recursos. En comparación a las viejas técnicas de desarrollo esto me permite optimizar el tiempo, ya que si se plantean ciclos de meses y existen algún cambio de requerimiento o existen algún mal entendido al inicio del ciclo, esto podría terminar en el desperdicio del ciclo y en los recursos del mismo.

Valentía

La valentía en programación extrema está directamente atada al dicho "No dejes para mañana lo que se puede realizar hoy", esto quiere decir que los programadores no deben estancarse en el diseño, caso contrario esto dificultara el resto de pasos como son la programación o realizar algún cambio en el sistema. La valentía también está relacionada hacia la perseverancia ya que los programadores se pueden encontrar con obstáculos que sean difíciles de resolver, pero la constante lucha y esfuerzo para salir adelante los llevara a lograr los objetivos establecidos.

Respeto

El respeto es la obligación moral que debe existir entre cada miembro del equipo, se debe mantener respeto en cada instante, por ejemplo, cuando se realiza algún cambio en el código, se debe respetar lo que ya antes funcionaba exitosamente y no dañarlo, no se debe menospreciar a ningún miembro del equipo, ya que esto fomenta el trabajo en equipo y ayuda a alcanzar el objetivo principal para el cual el equipo está trabajando que sería la creación de un producto de software con calidad.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

2.1.3 Roles

Desarrollador o Programador

Este rol dentro de la programación extrema es considerado como uno de los más esenciales, debido a que es el encargado de ejecutar la construcción del software. El programador debe tener habilidades de comunicación y de trabajo en equipo que le permitirán acoplarse a lo necesario dentro del proyecto.

Ciente

El cliente juega un rol importante dentro del proyecto, el cliente debería tener perfectamente claro que es lo que se va a programar a diferencia del programador que tiene claro el cómo. El cliente es el encargado de expresar todas sus necesidades a través de las historias de usuario con el objetivo de llegar a un acuerdo sobre lo que se va a construir.

Gerente

El gerente del proyecto dentro de la programación extrema es el encargado de organizar las reglas del proyecto, es decir va a calendarizar y planificar las reuniones, cuando empieza y cuando termina cada iteración del proyecto, es el encargado de familiarizar al equipo con el/los clientes y en algunos casos también debe actuar como mediador.

Entrenador

Como su nombre lo dice es la persona encargada de entrenar o guiar a cada miembro del equipo a lo largo del proyecto para que cada uno cumpla con los objetivos establecidos, si bien es cierto que cada persona es responsable de sus propios actos, este rol es de crucial importancia cuando un miembro nuevo se integra al equipo, ya que su obligación es ayudar a este nuevo miembro a adaptarse a las características del equipo lo más rápido posible.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Tracker o Seguidor

El tracker debe estar involucrado a lo largo de todo el proyecto, desde el inicio hasta la finalización del mismo, el tracker es el encargado de generar retroalimentaciones hacia el equipo en base a las estimaciones que se realizan al inicio del proyecto comparadas con los resultados reales obtenidos, el objetivo es mantener al equipo informado sobre cómo está llevando el proyecto.

Tester o Probador funcional

La principal funcionalidad del Tester es diseñar, implementar y ejecutar las pruebas de funcionalidad del proyecto, una vez realizadas las pruebas el tester está en la obligación de procesar estos resultados en un lenguaje lo más natural posible, es decir usando gráficos, estadísticas, etc. con el fin de que pueda ser entendido por todos y no solo por los encargados de la parte técnica del proyecto.

Consultor

El rol de consultor es un rol de apoyo hacia el equipo, el consultor debe ser una persona externa con conocimientos específicos, si en algún punto de ejecución del proyecto el equipo tiene algún problema, este podría solicitar ayuda al consultor que se supone tiene conocimientos específicos sobre el área en la que se está trabajando.

[2.1.4 Ciclo de Vida](#)

El ciclo de vida de la programación extrema está dividido en iteraciones que se pueden determinar según la necesidad de cada proyecto, cada iteración está definida por 4 fases que son:

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

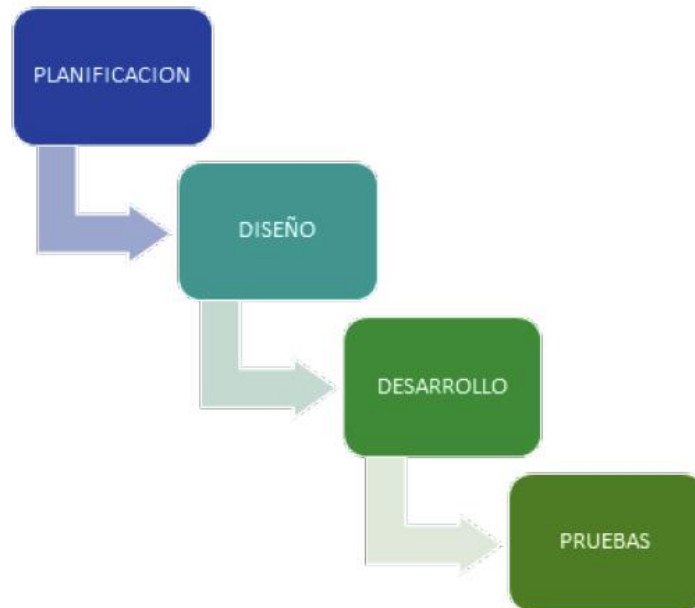


Ilustración 6. Ciclo de vida XP

Una vez que hemos establecido las 4 fases fundamentales por las cuales el proyecto atravesará, es importante entender de que está compuestas cada una. Es fundamental recordar que desde el primer instante hasta el último en el que se aplica esta metodología se debe trabajar siempre bajo los valores y las características mencionadas anteriormente.



Ilustración 7. Ciclo de vida XP explicado

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

1. Planificación

La fase más importante de toda la metodología es la planificación, ya que es aquí donde se va determinar qué es lo que se va a construir, es decir las necesidades del usuario. Esta fase está compuesta por:

- Historias de Usuario

Las historias de Usuario son realizadas por el cliente, deben ser realizadas en un lenguaje de alto nivel, es decir lo más natural posible sin usar términos técnicos, no va a contener creación o descripción de modelo de datos, algoritmos, clases, etc. deben ser claras, precisas, concisas, de una longitud de 3 o 4 líneas. Hay que tener en cuenta que la historia de usuario me puede servir para la etapa de pruebas, ya que debo verificar si al final se cumple con lo necesitado.

Ejemplo

Historia de Usuario	
Numero de Historia:	
Fecha:	
Tipo (nueva/mejorar/cambiar):	
Nombre de Historia:	
Usuario:	
Prioridad en Negocio (alta/media/baja):	
Riesgo (alto/medio/bajo):	
Iteración Asignada:	
Descripción:	

Cliente

Responsable

Tabla 1. Historia de usuario

- Planificación de lanzamiento

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Una vez que se obtienen todas las historias de usuario necesarias para el proyecto, se procede a crear un plan para ejecutarlas; “Release planning”. Este plan debe definir qué historias van a ser creadas en cada versión del proyecto, calendarizando las fechas tentativas para el lanzamiento de estas versiones.

Ejemplo

Release Planning						
Versión del sistema	Fecha publicación versión	Prioridad de historia	Numero de historia	Nombre de historia	Tiempo de implementación	# de personas desarrollo

Tabla 2. • Planificación de lanzamiento

- **Plan de iteraciones**

En el plan de iteraciones es donde se va a designar que es lo que se va a ejecutar en cada iteración, cada iteración durara aproximadamente 3 semanas.

Ejemplo

Plan de Iteraciones					
Numero de Iteración	Fecha inicio	Fecha fin	Numero de historia	Nombre de historia	Responsable

Tabla 3. Plan de iteraciones

2. Diseño

El diseño es el proceso previo que se debe realizar para poder plantearse la solución que se pretende dar a las historias de usuario adquiridas. En la programación extrema se diseñan aquellas historias que previamente fueron seleccionadas por el usuario para ser ejecutadas en esa iteración. Las principales características del diseño son:

- **Diseño simple**

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

El diseño al igual que los demás componentes de la programación extreme deben sujetarse a la filosofía y a los valores de la metodología, en este caso la simplicidad, ya que se considera que, si un diseño está bien hecho y es simple, se lo puede implementar en corto tiempo y siempre este diseño debe apuntar a la historia de usuario correspondiente.

- **Metáfora**

La metáfora es la representación de manera global de cómo el sistema se va a comportar, por esta característica la metáfora debe ser construida en lenguaje a alto nivel, es decir que sea entendible por la mayoría de personas involucradas en el proyecto, desde los técnicos hasta los usuarios finales de la misma. Las metáforas van a involucrar cada historia de usuario que exista y va a crear una descripción preferible de manera gráfica de cómo interactúan cada una de ellas, estandarizando terminologías, apoyándose con diccionario de datos o estándares de programación.

- **Tarjetas C.R.C**

Las tarjetas de clase, responsabilidad, colaboración se enfocan en lo que conocemos como programación orientada a objetos. Son tarjetas donde se van a asignar uno o dos responsables de las mismas para administrarlas. Para la elaboración de estas tarjetas pueden participar algunos miembros técnicos del equipo. Mediante una lluvia de ideas se determinan las clases que están involucradas, se debe analizar y clasificar estas clases obtenidas según la utilidad para el proyecto y descartando las menos utilizadas después de clasificar las clases, se las asignan a la tarjeta en orden de importancia.

Ejemplo

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Tarjeta CRC	
Nombre del Proyecto:	
Numero de Iteración:	
Fecha de Reunión:	
Nombre de Clase:	
Responsabilidades:	Colaboradores:

Tabla 4. Tarjeta CRC

- **Funcionalidades extras o soluciones de punta**

Esta parte del diseño está enfocada para cuando a lo largo del proyecto se presenten características difíciles o que representen algún problema para los equipos que trabajan en el proyecto, si esta situación aparece se puede asignar a dos técnicos que específicamente trabajen, investiguen y estudien el problema con el fin de encontrar una solución a este y que deje de ser una amenaza para el equipo.

Ejemplo

Funcionalidad Extra	
Nombre:	
Fecha inicio:	
Fecha fin:	
Integrantes:	
Descripción:	

Tabla 5. Funcionalidad extra

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

- **Refactorización**

Una vez que todos los pasos anteriores se realizaron la programación extreme recomienda volver a analizar y revisar todo lo hecho hasta el momento con el objetivo principal de mantener la filosofía de la metodología sobre el diseño que es la simplicidad, para que en la fase de desarrollo no existe mayor problema y puede ser ejecutada de una manera rápida. Si en el transcurso de la refactorización se encuentra alguna anomalía se debe corregirla o ponerla en términos simples y ágiles inmediatamente.

3. Desarrollo

La fase de desarrollo en la programación extrema como en las metodologías ágiles en general ha marcado una diferencia importante con las metodologías tradicionales de desarrollo de software ya que debido al corto tiempo asignado para cada iteración el desarrollo incluso podría ir ejecutándose al mismo tiempo con el diseño. A continuación, vamos a destacar las principales características de la fase de desarrollo.

- **Programación en parejas**

La programación extrema recomienda que los equipos de desarrollo deben ser establecidos en un máximo de 10 personas, agrupándolos por parejas. Estas dos personas van a trabajar en un mismo computador, una persona se va a encargar de escribir el código, por ejemplo, una función determinada de una clase, esta persona se debe esmerar por escribir este código con todos los estándares de programación que el proyecto demande y de la manera más eficiente posible. La segunda persona se va a encargar de analizar cómo se va a comportar esta función con los demás componentes, va a analizar si esta función es necesaria, si hace lo que se necesita, si la función puede ser optimizada o mejorada, etc. Las parejas de programación se pueden rotar en sus roles, así se cubrirá falencias posibles que puedan existir dentro de cada pareja.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

En este caso en específico para la construcción del proyecto, el trabajo de la programación en parejas se va a recargar en una sola persona, que debido a las características del proyecto no representara mayor inconveniente.

- **Integración continua**

Como la esencia del desarrollo en la programación extreme es la simplicidad y la programación en parejas, en algún punto esto puede llegar a convertirse en una amenaza para el equipo, debido a que cada pareja está concentrada en realizar su trabajo, causando una situación ajena entre el desenvolvimiento de las parejas colegas y por ende se crea la necesidad del versionamiento y la necesidad de evitar el trabajo ambiguo que se crea como resultado a la misma metodología.

Para combatir los posibles problemas que se pueden presentar en el desarrollo mencionados anteriormente, se debe utilizar la integración continua, de esta manera cada pareja está obligada a actualizar e integrar su trabajo con el sistema en general después de aplicar todas las pruebas necesarias que garanticen que el trabajo hecho ese día es el adecuado, todo esto debe pasar máximo al final de cada día, evitando que la integración continua se acumule.

- **Bucle de repetición (1. Planificación)**

Una de las principales características de las metodologías ágiles es la adaptabilidad al entorno y los cambios, si en este punto de ejecución del proyecto el cliente decide aumentar, cambiar o mejorar alguna historia de usuario el equipo debe analizar este impacto empezando desde el paso 1. Planificación, después de analizar la petición el equipo debe tomar las decisiones adecuadas sobre la misma.

4. Pruebas

La fase de pruebas es la garantía de que todo lo que el equipo ha venido construyendo es óptimo, funcional y puede ser utilizado en el proyecto. Las pruebas deben realizarse a la par de

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

cada nuevo producto creado para el proyecto en curso y no esperar hasta el último, con el objetivo de construir un programa robusto que pueda perdurar en el tiempo y que sea capaz de resistir y adaptarse a los cambios.

- **Pruebas de aceptación**

Las pruebas de aceptación son elaboradas en base a las historias del cliente, este tipo de pruebas son denominadas como caja negra, debido a que nos interesa el requerimiento del inicio y el resultado final, sin enfocarse en el procedimiento que puede existir en el medio. El objetivo principal de estas pruebas es garantizar que las funcionalidades del sistema que se está construyendo cumplan con los requerimientos de inicio (historias de usuario).



Ilustración 8. Pruebas de Aceptación

Ejemplo

Prueba de Aceptación	
Nombre:	
Código:	
Descripción:	
Condiciones de ejecución:	
Entradas:	
Pasos de ejecución:	
Resultado esperado:	
Evaluación de la prueba:	

Tabla 6. Prueba de aceptación

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

- **Pruebas unitarias**

Las pruebas unitarias consisten en dividir los elementos por módulos, en el caso de la programación orientada a objetos esto quiere decir en clases, se toma cada clase del proyecto y se analiza su comportamiento de manera individual y se verifica que funcione de la forma necesitada antes de integrarla al sistema en general.

Es importante que las pruebas sean determinadas antes de la construcción de cada clase o modulo, las pruebas unitarias son de vital importancia en los módulos que el equipo considere difícil o peligro para el proyecto.

2.1.5 Ventajas y Desventajas

Ventajas

- Optimización de recursos: al ejecutarse el proyecto mediante iteraciones de corto plazo, se puede realizar cambios en la marcha sin tener que esperar a que se acabe toda la iteración y en el caso de ser una iteración de meses, esta podría ser tiempo perdido.
- Mayor calidad: al realizar iteraciones más cortas, se puede verificar con el cliente los resultados de las mismas y confirmar o corregir para que estos resultados sean lo más cercanos al cliente.
- Mejor comunicación y organización: debido a una de las características intrínsecas de la programación extrema, el desarrollador debe mantener estándares de programación que servirán para facilitar el entendimiento, el mantenimiento y la comunicación entre los miembros del equipo.

Desventajas

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

- La principal desventaja que esta metodología presenta es que es aplicable en su mayoría para proyectos a corto plazo.
- Tampoco es aplicable para equipos grandes de desarrollo, máximo 10 personas, enfocada en la programación en parejas o individual.
- En algunas ocasiones puede ser difícil aplicar esta metodología ya que todas las personas integrantes del proyecto deben tener la mentalidad y la predisposición para ajustarse a los valores y principios de la programación extrema.

2.2. Lenguaje de programación C#



Ilustración 9. C#

2.2.1. Introducción.

C# es un lenguaje de programación, con un paradigma orientado a objetos, desarrollado, controlado y distribuido por Microsoft. C# está aprobado por la asociación europea de fabricantes de ordenadores (ECMA por sus siglas en inglés) y también está regido bajo la organización internacional de estándares ISO. Este lenguaje de programación fue creado principalmente por Adres Hejlsberg originario de Dinamarca en los años 2000.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

2.2.2. Características fundamentales.

- **Amigable para aprendices.**

Este lenguaje de programación es amigable debido a que no es un lenguaje completamente nuevo en el mercado, el mundo de construcción de software ya conoce a sus antecesores que son C, C++ y al gran parecido con java, esto representa confianza en los programadores a la hora de introducirse en este lenguaje.

- **Common Language Infrastructure (CLI)**

C# está construido bajo la infraestructura de lenguaje común, esto quiere decir que está sujeto a estándares, el CLI se encarga de administrar un entorno virtual para que las aplicaciones que están construidas con diferentes lenguajes a alto nivel puedan ser ejecutadas en plataformas de diferente tipo tanto a nivel de hardware como al nivel de software.

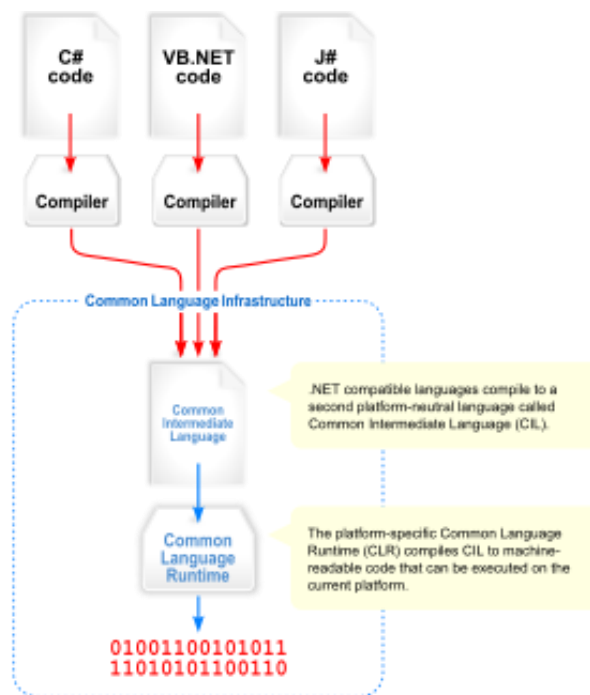


Ilustración 10. • Common Language Infrastructure (CLI)

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

- **Escalabilidad**

Al ser orientado a objetos, la escalabilidad es una propiedad inherente en este lenguaje, si se quiere potenciar un programa construido en C# solo basta con usar una carga adicional que puede ser el uso de alguna biblioteca (conjunto de clases), evitando hacer un cambio extremadamente extensivo en la aplicación construida.

- **Comunidad**

Cuando estamos por escoger un lenguaje de programación con el cual vamos a trabajar nuestro siguiente proyecto es de vital importancia tener en mente que en algún punto nos podemos estancar y vamos a necesitar la ayuda de alguien, es aquí donde entra en juego la comunidad de soporte que tiene dicho lenguaje, en el caso de C# cuenta con sitios oficiales donde millones de programadores comparten y resuelven dudas online que en algunas ocasiones serán de gran ayuda.

2.2.3. Ventajas y desventajas.

Ventajas

- **Reusabilidad:** si diseñamos y codificamos bien nuestras clases en c#, podemos volverlas a utilizarlas en varios proyectos o en varias partes del mismo programa, esto será de gran ayuda a la optimización de código.
- **Mantenimiento:** debido a que este tipo de lenguajes como C# o Java tienen un alto grado de abstracción con las clases y objetos, es muy fácil identificar los detalles más importantes para el proyecto.
- **Confianza:** al ser un lenguaje respaldado por una empresa de la importancia de Microsoft, brinda un respaldo y seguridad a los usuarios de este lenguaje.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Desventajas

- Una desventaja es que para ingresar en el paradigma de la programación orientada a objetos es importante cambiar toda la mentalidad de la programación tradicional e involucrarse por completo en el lenguaje y la filosofía del mismo.
- Para poder trabajar con un lenguaje de este tipo es necesario ser ordenado y metodológico, ya que debemos ser capaz primero de imaginarnos la situación para poder lograr una buena abstracción del problema para poder trabajar de manera adecuada con el lenguaje.

2.3. Gestor de base de datos SQL server



Ilustración 11. Sql Server 2014

2.3.1. Introducción.

Sql Server es un gestor de base de datos desarrollado por Microsoft, este gestor está basado en bases de datos de tipo relacional. Las principales funciones de SQL Server son crear tablas con relaciones definidas entre cada tabla y la manipulación de estos datos mediante la creación, edición, eliminación o la consulta de los mismos. SQL Server trabaja con un lenguaje de búsqueda estructurado (Transact SQL) que es un derivado del lenguaje SQL.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

2.3.2. Características fundamentales.

- **Transaccionalidad**

La Transaccionalidad es una de las características más importantes de SQL Server, ya que permite garantizar que una operación se va a realizar o no al 100%, es decir si estoy haciendo un retiro en la base de datos y en el medio de este proceso surge algún inconveniente, la operación se quedara tal y como estaba antes de empezarla, de esta manera se evita pérdidas o creación de información ambigua.

- **Procedimientos almacenados**

Un procedimiento almacenado es un conjunto de sentencias de tipo sql que están previamente compiladas con el objetivo de realizar algún proceso deseado, este procedimiento almacenado puede ser llamado para ser ejecutado desde varios programas.

- **Managment Studio**

Sql server cuenta con una interfaz gráfica que es muy amigable al usuario y permite realizar todas las funciones disponibles, en muchos de los casos facilitando la ejecución de dichas funciones que si deberían ser realizadas manualmente podrían emplear varias líneas de código de SQL.

2.3.3. Ventajas y desventajas.

Ventajas

- Seguridad: Sql Server es un gestor robusto que permite controlar y asignar roles para el acceso a la información, de esta manera se está protegiendo la información contenida en las bases de datos.
- Diversidad: otra ventaja es que estamos permitidos trabajar en diferentes ambientes, como lo es cliente servidor, de esta manera garantizamos que los datos del cliente

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

estarán almacenados en el modo servidor y los puntos de acceso al servidor trabajarán solamente como terminales de acceso.

Desventajas

- La principal desventaja de SQL server es que es una herramienta de pago, si bien tiene versiones gratuitas, si se quiere realizar un proyecto de gran alcance, tendremos que adquirir una de estas licencias que no son nada baratas.

2.4. Herramienta de desarrollo Visual Studio



Ilustración 12. Visual Studio 2012

2.4.1. Introducción.

Visual Studio es la interfaz gráfica de trabajo oficial de Microsoft (IDE), al principio estaba destinado para funcionar solamente en sistemas operativos de Windows, pero a finales del año 2016 Microsoft anuncio que se lanzaría una versión para IOS.

Microsoft lanzo la primera versión de visual estudio en 1997 de ahí en adelante Microsoft ha continuado dando soporte y creando nuevas versiones de Visual hasta la actualidad, debido al gran impacto que esta herramienta causo en el mercado de desarrollo de software, en especial de aplicativos web.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

2.4.2. Características fundamentales.

- Visual Studio en su versión 2012 soporta un entorno de trabajo o framework llamado **Asp.net** que permite crear a los desarrolladores aplicaciones de tipo web, basadas en servicios como SOAP, WCF o REST.
- **MVC** (modelo vista controlador) es un marco de referencia que contiene formularios web que ayudan a la creación de aplicaciones web dividiéndolas en 3 partes como su nombre lo indica, la parte de la lógica del manejo de datos, la interfaz que el usuario va a ver al terminar la aplicación y los controladores que están encargados de manejar todas las acciones ejecutadas por el usuario.

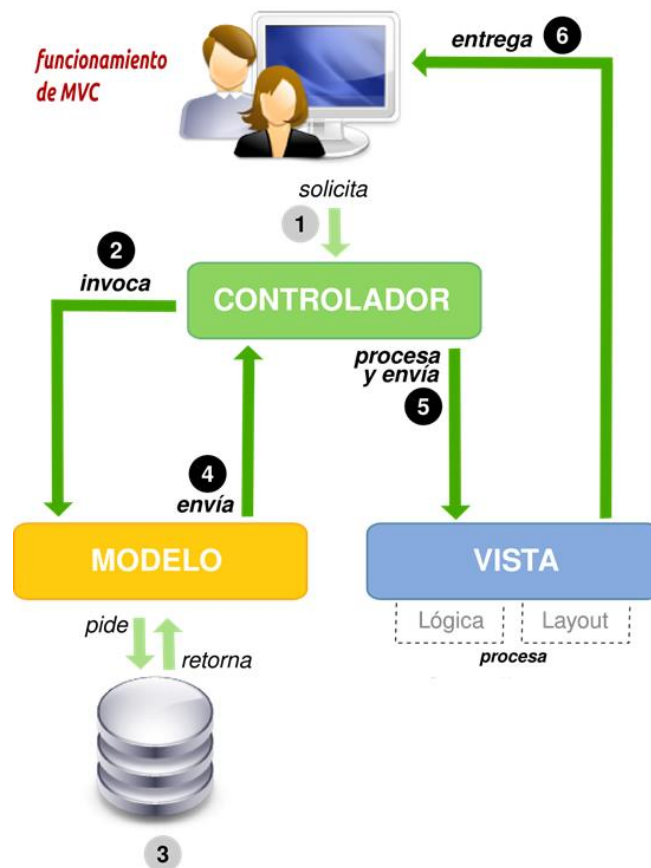


Ilustración 13. Modelo Vista Controlador

- **Herramienta Multiplataforma**, visual studio ha desarrollado marcos de referencia y de apoyo como ASP.NET, que permite estandarizar la creación de aplicaciones, al tener

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

todo estandarizado se facilita la comunicación entre diferentes plataformas como son dispositivos móviles, aplicaciones web, aplicaciones de escritorio etc.

2.4.3. Ventajas y desventajas.

Ventajas

- La principal ventaja que nos brinda este IDE es la facilidad para crear aplicaciones, ya que debido a su framework la mayoría de las conexiones siguen estándares que ya están automatizados en el IDE lo que permite desarrollar eficazmente nuestras aplicaciones.
- La parte del diseño de usuario y la implementación del mismo están muy bien conectados y se puede trabajar a la par, permitiendo crear aplicaciones más robustas y en menor tiempo.

Desventajas

- La principal desventaja que nos presenta esta herramienta es que es de tipo propietario, a pesar de que existen versiones liberadas, si se desea probar las características totales que esta herramienta posee se deberá adquirir una licencia.
- Hasta finales del año 2016 era imposible migrar el código a cualquier otra plataforma diferente de Windows, a partir del 2017 ya se lo puede hacer solamente hacia IOS.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

2.5. Herramienta de desarrollo Android Studio



Ilustración 14. Android Studio 2.2

2.5.1. Introducción.

Anteriormente los desarrolladores de móviles del sistema operativo Android trabajaban oficialmente con la herramienta Eclipse hasta que Android Studio fue lanzado oficialmente en su primera versión en el año 2014 por IntelliJ. La comercialización de Android studio está dada de manera libre, ya que rige bajo la licencia de apache2.0 permitiendo de esta manera llegar a millones de desarrolladores en todo el mundo. La última versión disponible a inicios del año 2017 es la 2.2.2.0.

2.5.2. Características fundamentales.

- Una característica que ayuda bastante a la hora del desarrollo es la consola de desarrollo, ya que esta nos permite acceder a estadísticas de uso, alerta de consumo de memoria, recomendaciones para optimización, guía de ejecución, etc.
- Android studio está construido en base al lenguaje de programación java, lo que significa que va a soportar todas las características de este lenguaje, como lo es el paradigma de programación orientado a objetos.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

- Está orientado a ser multiplataforma y puede ser ejecutado en varios ambientes y sistemas operativos.
- Cuenta con herramientas para controlar el versionamiento de las aplicaciones y facilita el uso de plantillas para estandarizar la creación de componentes (botones, campos de texto, etc.) en la parte de diseño.

2.5.3. Ventajas y desventajas.

Ventajas

- **Uso libre:** esto representa una gran ventaja, debido a que todo el mundo puede acceder al uso de esta potente herramienta sin tener que pagar ninguna licencia.
- **Comunidad:** existe una gran comunidad de apoyo atrás de esta herramienta, con millones de desarrolladores dispuestos a proponer problemas y soluciones sobre la herramienta mediante foros online, canales de YouTube, páginas web, etc.
- **Autodidacta:** si se está empezando por primera vez en el desarrollo de aplicaciones para Android, existen excelentes cursos online de prestigiosas instituciones como lo es el Tecnológico de monterrey que ofertan estos cursos en modalidad gratis o pagada.

Desventajas

- La principal desventaja que presenta esta herramienta es que se necesita una alta exigencia de recursos en hardware sobre las maquinas en las que se va a trabajar, partiendo mínimamente de: 8 gb RAM, procesador i7. En requerimientos menores, se puede trabajar, pero se va a notar problemas en la maquina como lo es la lentitud para ejecutar procesos.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

3. APLICACIÓN DE LA METODOLOGIA PROGRAMACION

EXTREMA

3.1. Iteración 1

3.1.1 Planificación

La fase de planificación va a servir de arranque para todo el proyecto y para cada iteración que se cree.

a) Historia de usuario

A continuación, vamos a registrar la primera historia de usuario, que es la necesidad del cliente y la cual va a servir de base para las demás historias de usuario y del proyecto en general.

Historia de Usuario	
Numero de Historia:	1
Fecha:	2015-08-10
Tipo (nueva/mejorar/cambiar):	nueva
Nombre de Historia:	Taxi seguro
Usuario:	Clientes de taxis
Prioridad en Negocio (alta/media/baja):	Alta
Riesgo (alto/medio/bajo):	Alto
Iteración Asignada:	1-4
Descripción: Una aplicación móvil que valide que el chofer de taxi es quien dice ser y que el taxi sea el que está registrado por las autoridades pertinentes.	

Cliente

Responsable

Tabla 7. Historia de usuario 1

A partir de la segunda historia de usuario hasta la quinta se entra en detalle sobre el contenido oficial que debe tener la aplicación.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Historia de Usuario	
Numero de Historia:	2
Fecha:	2015-08-10
Tipo (nueva/mejorar/cambiar):	nueva
Nombre de Historia:	Información oficial
Usuario:	Clientes de taxis
Prioridad en Negocio (alta/media/baja):	Media
Riesgo (alto/medio/bajo):	Alto
Iteración Asignada:	1
Descripción: Debe contener la información oficial del conductor, por ejemplo: Apellido, nombre y foto del chofer. Del vehículo: Marca, modelo, color, placa y registro municipal.	

Cliente

Responsable

Tabla 8. Historia de usuario 2

Historia de Usuario	
Numero de Historia:	3
Fecha:	2015-08-10
Tipo (nueva/mejorar/cambiar):	Nueva
Nombre de Historia:	Disponibilidad
Usuario:	Clientes de taxis
Prioridad en Negocio (alta/media/baja):	Alta
Riesgo (alto/medio/bajo):	Alto
Iteración Asignada:	2
Descripción: La información debe provenir de fuentes oficiales y debe estar actualizada al momento en que los usuarios accedan a la aplicación.	

Cliente

Responsable

Tabla 9. Historia de usuario 3

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Historia de Usuario	
Numero de Historia:	4
Fecha:	2015-08-10
Tipo (nueva/mejorar/cambiar):	nueva
Nombre de Historia:	Dispositivos Android
Usuario:	Clientes de taxis
Prioridad en Negocio (alta/media/baja):	Alta
Riesgo (alto/medio/bajo):	Alto
Iteración Asignada:	3
Descripción: La aplicación que va a ser usada por los clientes debe estar dirigida para dispositivos que usan Android.	

Cliente

Responsable

Tabla 10. Historia de usuario 4

Historia de Usuario	
Numero de Historia:	5
Fecha:	2015-08-10
Tipo (nueva/mejorar/cambiar):	nueva
Nombre de Historia:	Interacción con cámara
Usuario:	Clientes de taxis
Prioridad en Negocio (alta/media/baja):	Baja
Riesgo (alto/medio/bajo):	Alto
Iteración Asignada:	4
Descripción: La aplicación debe permitir al usuario capturar con la cámara del dispositivo la información para realizar la consulta.	

Cliente

Responsable

Tabla 11. Historia de usuario 5

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

b) Planificación de lanzamiento

Release Planning						
Versión del sistema	Fecha publicación versión	Prioridad de historia	Numero de historia	Nombre de historia	Tiempo de implementación	# de personas desarrollo
1.0	Febrero-2017	Alta	1	Taxi seguro	12 semanas	1
1.0	Febrero-2017	Media	2	Información oficial	3 semanas	1
1.0	Febrero-2017	Alta	3	Disponibilidad	3 semanas	1
1.0	Febrero-2017	Alta	4	Dispositivos Android	3 semanas	1
1.0	Febrero-2017	Baja	5	Interacción con cámara	3 semanas	1

Tabla 12. Plan de lanzamiento Iteración 1

c) Plan de iteraciones

Plan de Iteraciones					
Numero de Iteración	Fecha inicio	Fecha fin	Numero de historia	Nombre de historia	Responsable
1	31-10-2016	18-11-2016	2	Información oficial	David Pungacho
2	21-11-2016	9-12-2016	3	Disponibilidad	David Pungacho
3	12-12-2016	6-01-2017	4	Dispositivos Android	David Pungacho
4	9-01-2017	27-01-2017	5	Interacción con cámara	David Pungacho

Tabla 13. Plan de iteraciones Iteración 1

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

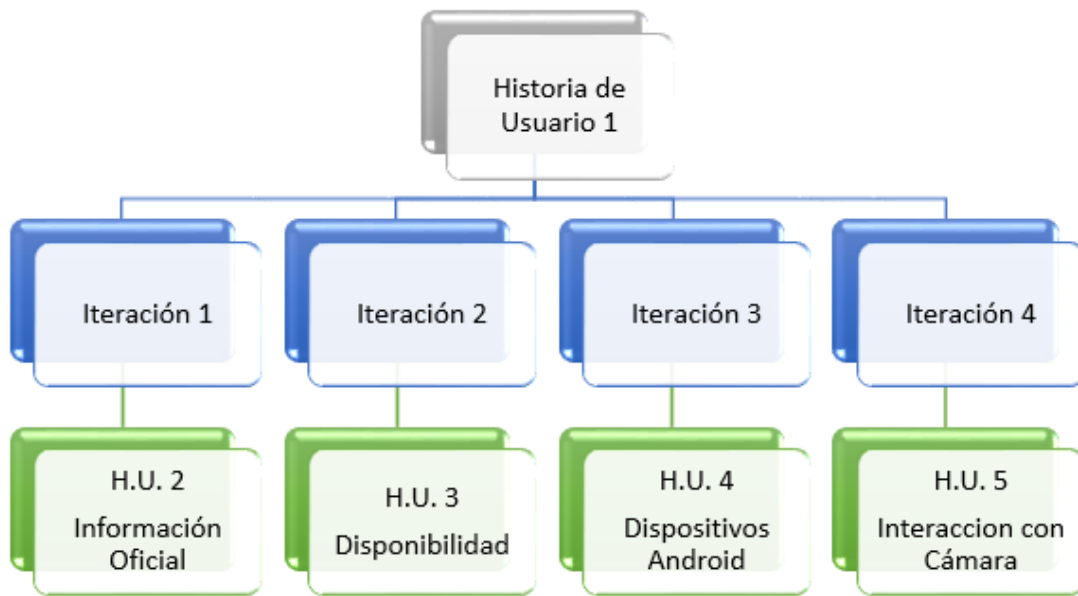


Ilustración 15. Historia de Usuario

3.1.2. Diseño

a) Diseño simple

Una vez que se estableció en la historia de usuario número 2 que información es la solicitada para mostrar en la aplicación, vamos a construir una base de datos relacional, tomando como referencia los campos encontrados en la agencia nacional de tránsito para los conductores y los campos encontrados en el registro municipal de taxis para los vehículos utilizando la herramienta power designer y SQL server. Para construir el modelo de datos vamos crear 3 modelos: conceptual, lógico y físico.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Diseño conceptual

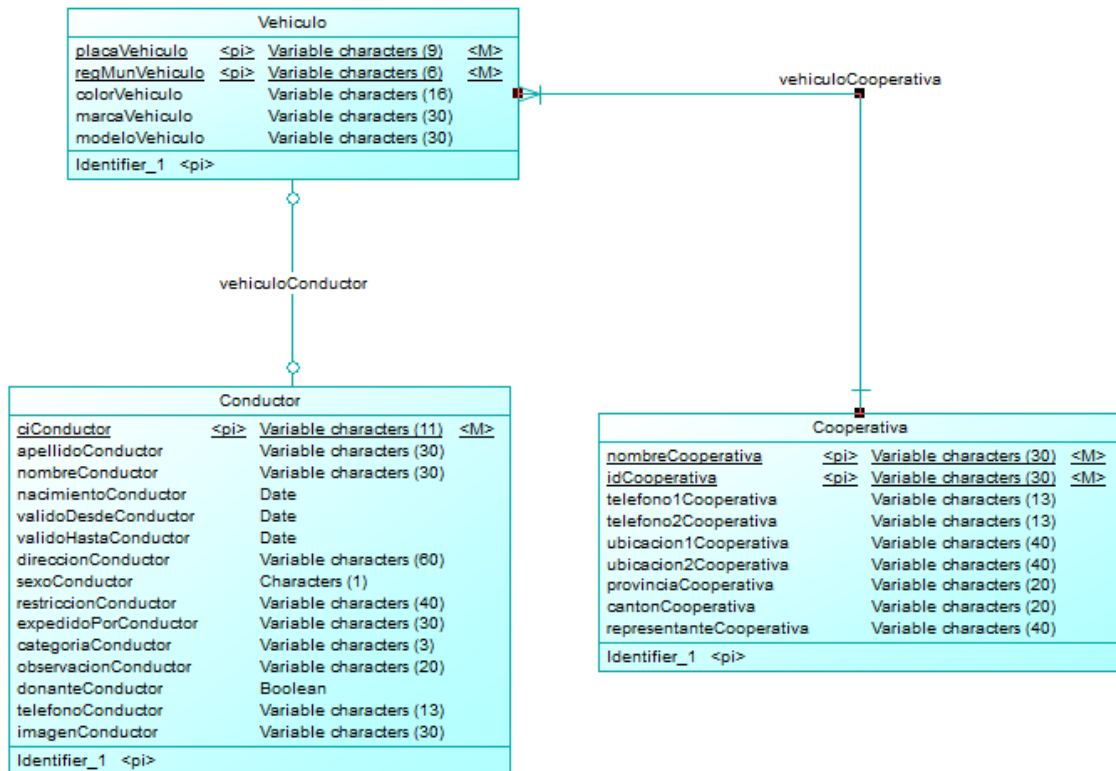


Ilustración 16. Diseño Conceptual BD

Diseño lógico

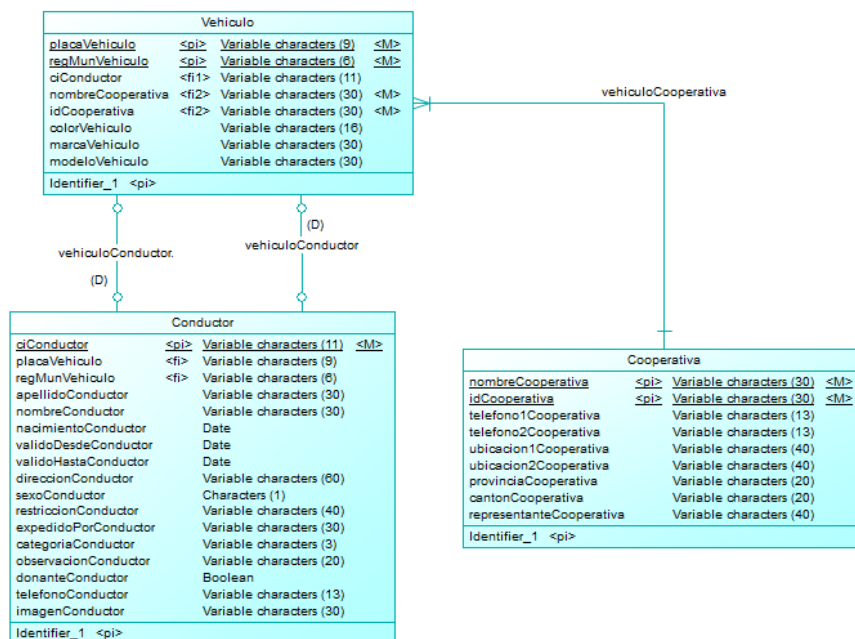


Ilustración 17. Diseño Lógico BD

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Diseño físico

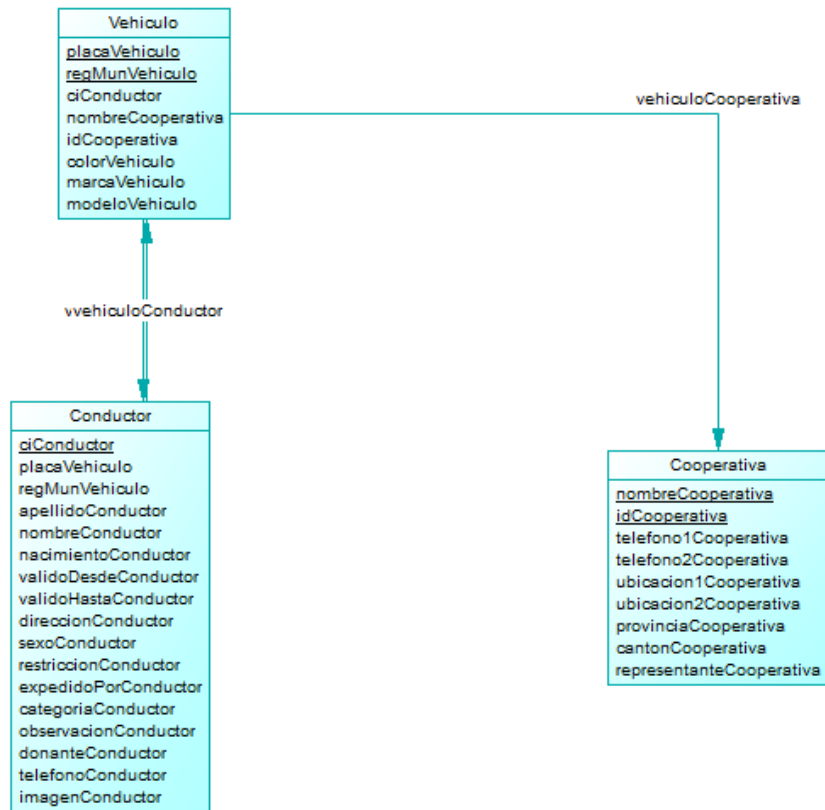


Ilustración 18. Diseño Físico BD

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

b) Metáforas del sistema

Comportamiento interno del modelo de datos

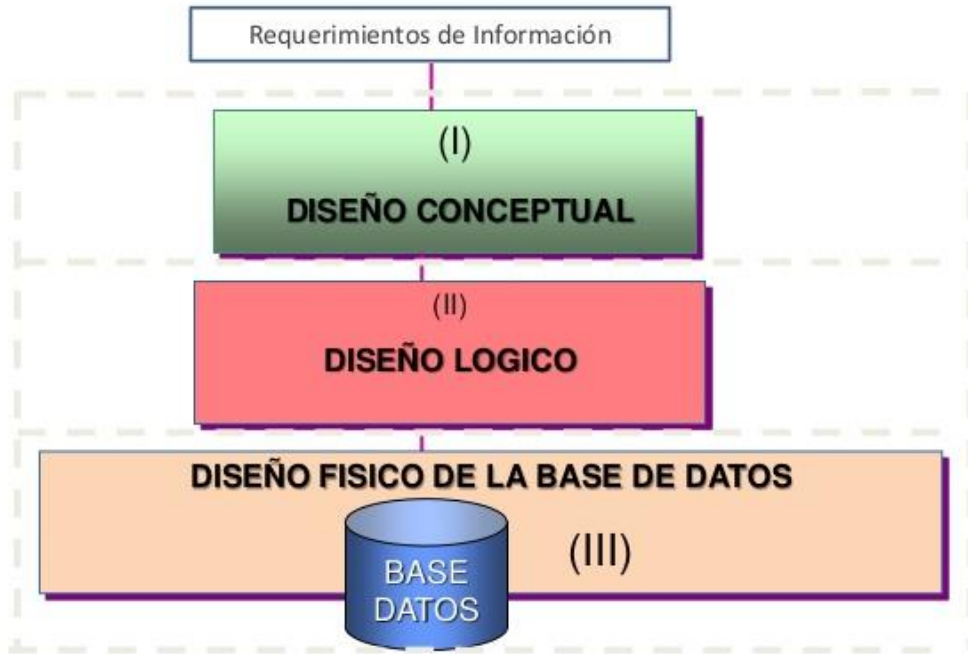


Ilustración 19. Metáfora del sistema BD

Comportamiento del modelo de datos dentro del sistema (parte resaltada).

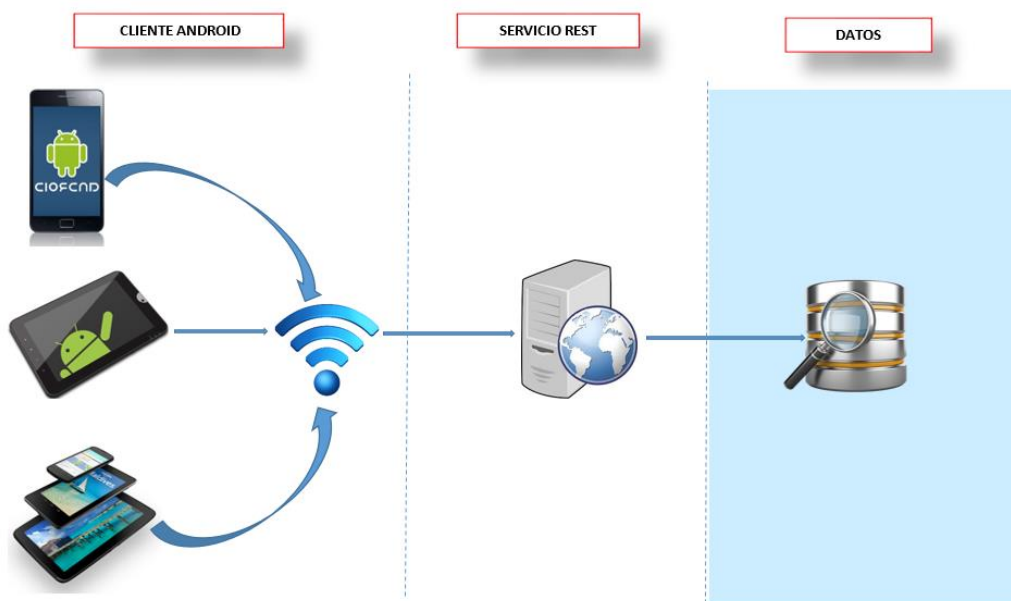


Ilustración 20. Metáfora del Sistema General BD

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

c) Tarjetas C.R.C

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	1
Fecha de Reunión:	04-11-2016
Nombre de Clase:	Vehículo
Responsabilidades: <ul style="list-style-type: none"> • Datos oficiales proporcionados por el municipio de quito para registrar taxis. • Solamente consulta de datos. • Relación con conductor y cooperativa. 	Colaboradores: <ul style="list-style-type: none"> • David Pungacho

Tabla 14. Tarjeta CRC1 Iteración 1

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	1
Fecha de Reunión:	04-11-2016
Nombre de Clase:	Conductor
Responsabilidades: <ul style="list-style-type: none"> • Datos oficiales proporcionados por la agencia nacional de tránsito para ser conductor profesional. • Solamente consulta de datos. • Relación con vehículo. 	Colaboradores: <ul style="list-style-type: none"> • David Pungacho

Tabla 15. Tarjeta CRC2 Iteración 1

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	1
Fecha de Reunión:	04-11-2016
Nombre de Clase:	Cooperativa
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> Datos oficiales proporcionados por el municipio de quito para cooperativas de taxis. Solamente consulta de datos. Relación con vehículo. 	<ul style="list-style-type: none"> David Pungacho

Tabla 16. Tarjeta CRC3 Iteración 1

d) Funcionalidades extra

Funcionalidad Extra	
Nombre:	Interacción con cámara
Fecha inicio:	1-11-2016
Fecha fin:	6-01-2017
Integrantes:	David Pungacho
Descripción:	
<ul style="list-style-type: none"> La funcionalidad extra es investigar cómo utilizar el recurso de la cámara de los dispositivos que usan Android para poder capturar una imagen, transformarla a texto y pasarla como parámetro para poder realizar la consulta en la aplicación. 	

Tabla 17. Funcionalidad extra Iteración 1

e) Refactorización

La refactorización fue realizada con éxito y se mejoraron las siguientes fases:

- d) Funcionalidades extra.
- b) Metáforas del sistema.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

3.1.3. Desarrollo

a) Integración continua

- Creamos una nueva base de datos en el gestor de SQL server

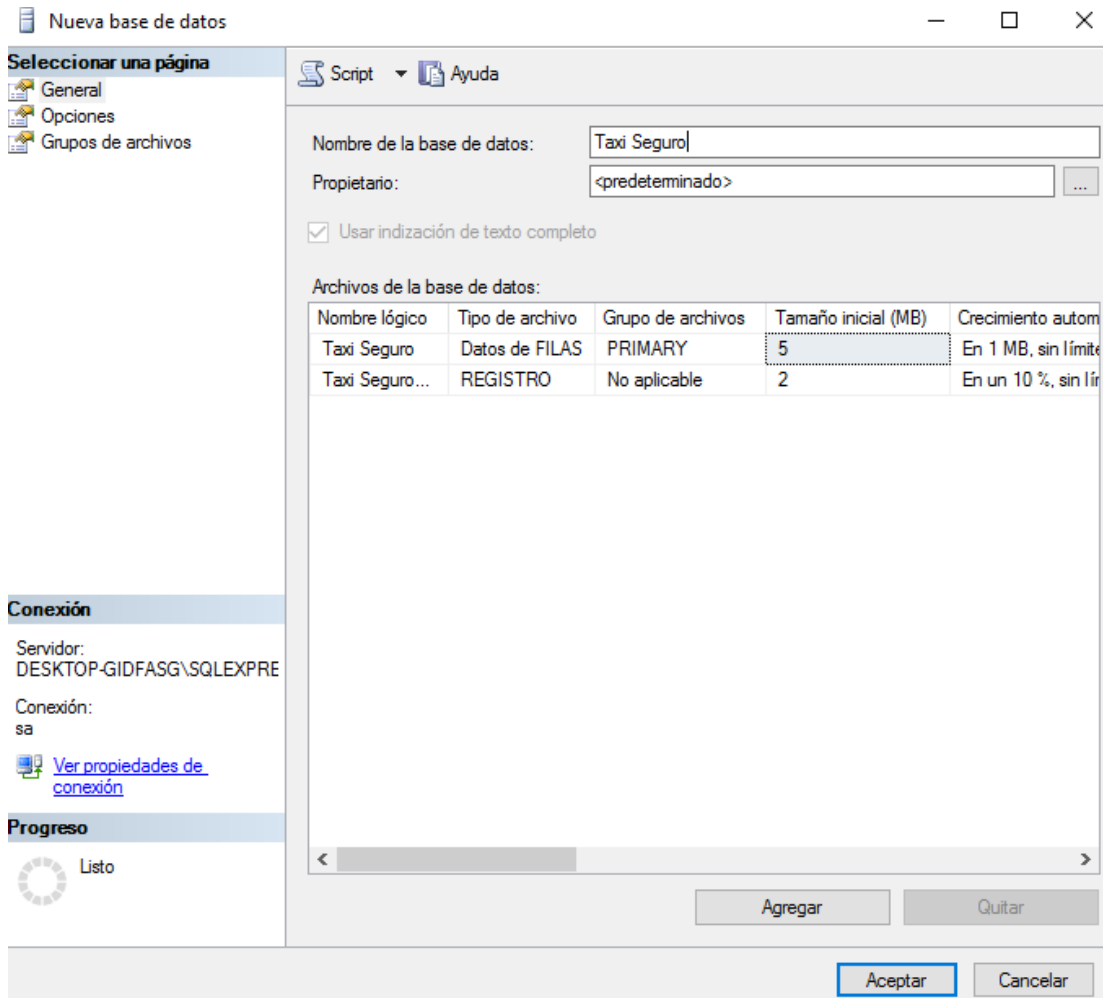


Ilustración 21. Creando una base

- Agregamos el script generado en base al modelo físico de la base

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

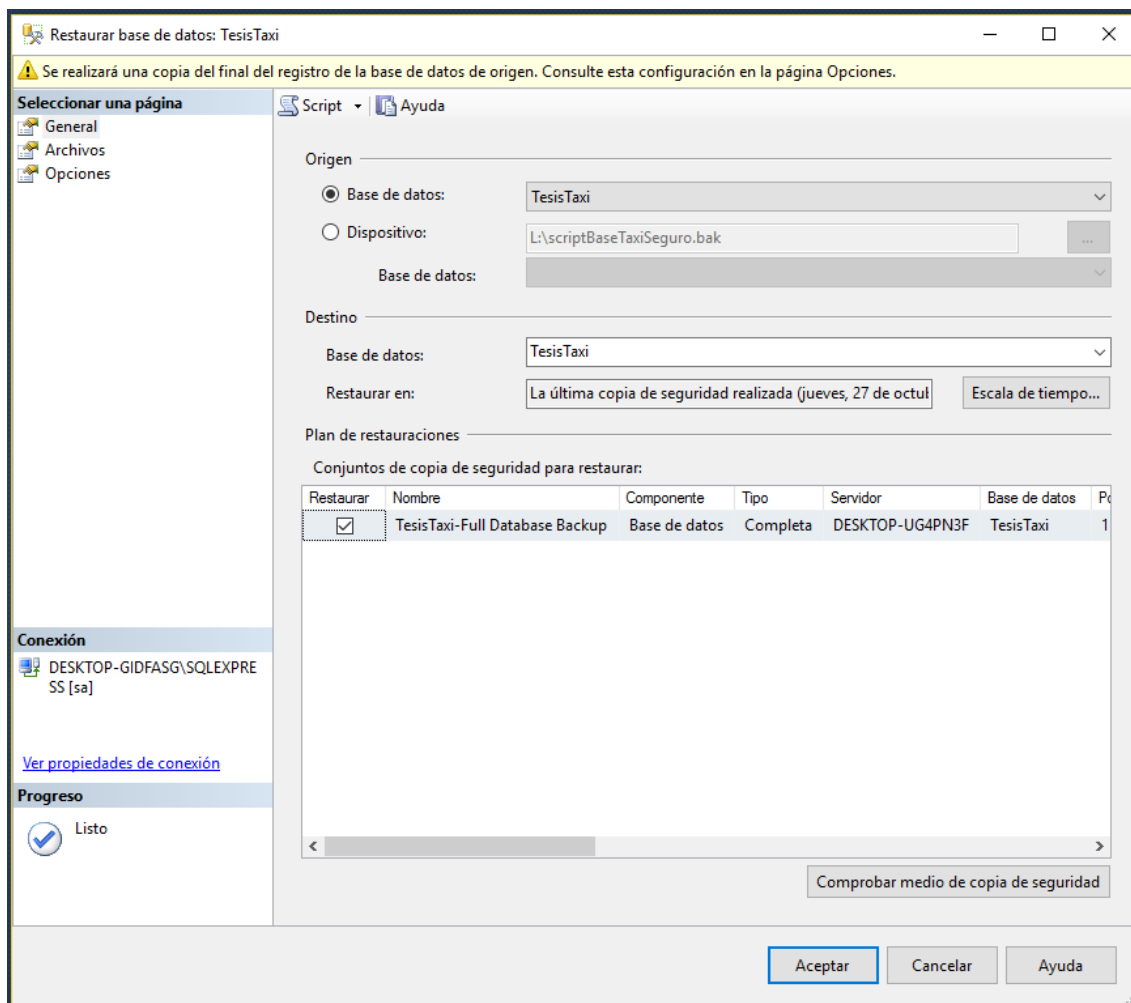


Ilustración 22. Corriendo el modelo físico

- En el explorador de objetos podemos observar que nuestra base esta creada

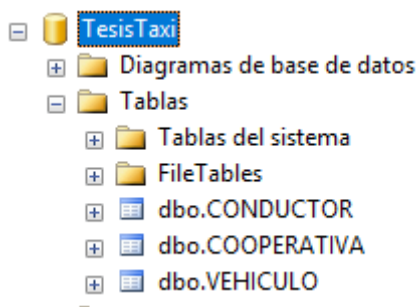


Ilustración 23. Verificando la BD

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

- Tabla conductor

CICONDUCTOR	PLACAVEHICULO	REGMUNVEHICULO	APELLIDOCONDUCTOR	NOMBRECONDUCTOR	NACIMIENTOCONDUCTOR	VALIDODESDECONDUCTOR	VALIDOHASTAONDUCTOR
1722875323	psc-4526	3254	Zapata	Andres	1990-02-03 00:00:00.000	2014-03-03 00:00:00.000	2017-06-06 00:00:00.000
1722875324	psc-4527	3255	Alarcon	Sandra	1990-02-03 00:00:00.000	2014-04-08 00:00:00.000	2018-05-09 00:00:00.000
1722875325	psc-4528	3256	Suarez	Carlos	1979-08-12 00:00:00.000	2015-08-07 00:00:00.000	2018-09-04 00:00:00.000
1722875326	psc-4529	3257	Villavicencio	Andrea	1984-07-12 00:00:00.000	2014-12-12 00:00:00.000	2019-04-06 00:00:00.000
1722875327	psc-4530	3258	Sanchez	Eduardo	1981-08-10 00:00:00.000	2014-08-06 00:00:00.000	2017-03-03 00:00:00.000

Ilustración 24. Tabla conductor

- Tabla cooperativa

NOMBRECOOPERATIVA	IDCOOPERATIVA	TELEFONO1COOPERATIVA	TELEFONO2COOPERATIVA	UBICACION1COOPERATIVA	UBICACION2COOPERATIVA	PROVINCIACOOPERATIVA	CANTONCOOPERATIVA
Aeropuerto	c1	022535353	022424242	tababela	bicentenario	Pichincha	Quito
Mariscal	c2	023478569	0999856478	amazonas	foch	pichincha	quito
Magdalena	c3	022123456	0987485998	napo	centro historico	Pichincha	Quito
23 de Octubre	c4	023458791	0984536971	brasil	prensa	Pichincha	Quito
El Comercio	c5	022124760	0985105904	condado	cotocollao	Pichincha	Quito

Ilustración 25. Tabla cooperativa

- Tabla vehículo

PLACAVEHICULO	REGMUNVEHICULO	CICONDUCTOR	NOMBRECOOPERATIVA	IDCOOPERATIVA	COLORVEHICULO	MARCAVEHICULO	MODELOVEHICULO
psc-4526	3254	1722875323	Aeropuerto	c1	amarillo/rojo	nissan	sentra
psc-4527	3255	1722875324	Mariscal	c2	amarillo/verde	chevrolet	aveo
psc-4528	3256	1722875325	Magdalena	c3	amarillo/negro	renault	family
psc-4529	3257	1722875326	23 de Octubre	c4	amarillo	chevrolet	corsa
psc-4530	3258	1722875327	El comercio	c5	amarillo/rojo	hyundai	elantra

Ilustración 26. Tabla vehículo

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

3.1.4. Pruebas

a) Pruebas de aceptación

Prueba de Aceptación	
Nombre:	Información oficial
Código:	1
Descripción:	Debe contener la información oficial del conductor, por ejemplo: Apellido, nombre y foto del chofer. Del vehículo: Marca, modelo, color, placa y registro municipal.
Condiciones de ejecución:	Debe existir información registrada en las 3 tablas
Entradas:	Registro municipal del vehículo Placa del vehículo
Pasos de ejecución:	<ol style="list-style-type: none"> 1. Obtener el registro municipal/placa del vehículo a consultar. 2. Identificamos la información que se desea mostrar. 3. Se construye el query para buscar la información en Sql server. 4. Verificar la información devuelta.
Resultado esperado:	Los datos que debe devolver la consulta son: <ul style="list-style-type: none"> • Apellido (conductor) • Nombre (conductor) • Código de la foto (conductor) • Marca • Modelo • Color • Placa • Registro municipal
Evaluación de la prueba:	La prueba realizada a la base de datos que se construyó concluyo con éxito al 100% ya que muestra la información solicitada por el cliente.

Tabla 18. Prueba de aceptación Iteración 1

b) Pruebas unitarias

Las pruebas unitarias en la primera iteración no tienen aplicación, debido a que son pruebas que se realizan sobre código escrito por ejemplo a cada clase construida o si de ser necesario a cada método construido.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

3.2. Iteración 2

3.2.1. Planificación

a) Historia de usuario

Historia de Usuario	
Numero de Historia:	3
Fecha:	2015-08-10
Tipo (nueva/mejorar/cambiar):	Nueva
Nombre de Historia:	Disponibilidad
Usuario:	Clientes de taxis
Prioridad en Negocio (alta/media/baja):	Alta
Riesgo (alto/medio/bajo):	Alto
Iteración Asignada:	2
Descripción: La información debe provenir de fuentes oficiales y debe estar actualizada al momento en que los usuarios accedan a la aplicación.	

Cliente

Responsable

Tabla 19. Historia de usuario 3 Iteración 2

b) Planificación de lanzamiento

Release Planning						
Versión del sistema	Fecha publicación versión	Prioridad de historia	Numero de historia	Nombre de historia	Tiempo de implementación	# de personas desarrollado
1.0	Febrero-2017	Alta	1	Taxi seguro	12 semanas	1
1.0	Febrero-2017	Media	2	Información oficial	3 semanas	1
1.0	Febrero-2017	Alta	3	Disponibilidad	3 semanas	1
1.0	Febrero-2017	Alta	4	Dispositivos Android	3 semanas	1
1.0	Febrero-2017	Baja	5	Interacción con cámara	3 semanas	1

Tabla 20. Planificación de lanzamiento Iteración 2

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

c) Plan de iteraciones

Plan de Iteraciones					
Numero de Iteración	Fecha inicio	Fecha fin	Numero de historia	Nombre de historia	Responsable
1	31-10-2016	18-11-2016	2	Información oficial	David Pungacho
2	21-11-2016	9-12-2016	3	Disponibilidad	David Pungacho
3	12-12-2016	6-01-2017	4	Dispositivos Android	David Pungacho
4	9-01-2017	27-01-2017	5	Interacción con cámara	David Pungacho

Tabla 21. Plan de iteraciones Iteración 2



Ilustración 27. Iteración 2

3.2.2. Diseño

a) Diseño simple

Basándonos en los requerimientos aprobados por el usuario, vamos a construir una aplicación en visual studio bajo la arquitectura de 3 capas que son: Cliente, lógica del negocio y los datos

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

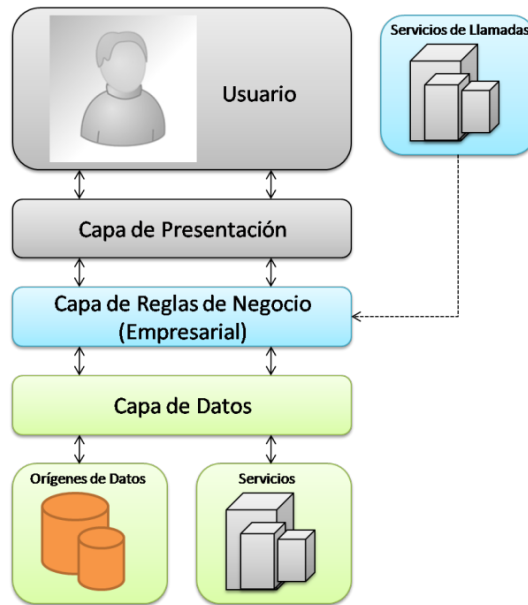
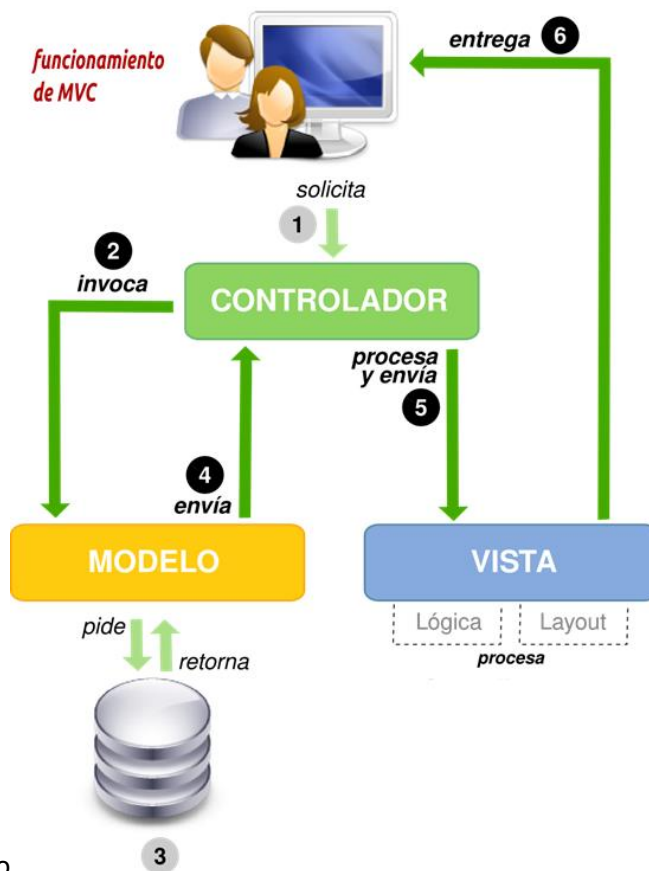


Ilustración 28. Arquitectura 3 capas

Para garantizar la accesibilidad a los datos desde un cliente externo vamos a construir un servicio de tipo Rest para ello utilizaremos la funcionalidad de Modelo Vista Controlador de



visual studio

Ilustración 29. Modelo Vista Controlador

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

b) Metáfora del sistema

La parte resaltada indica el comportamiento de la iteración actual con todo el sistema

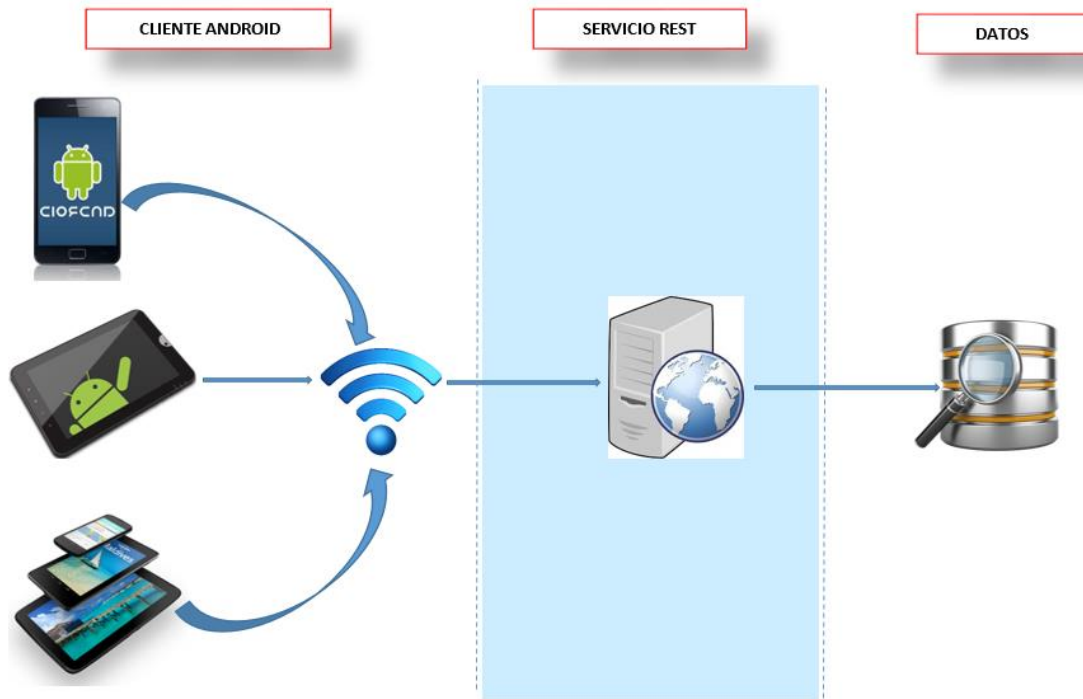


Ilustración 30. Metáfora del Sistema Servicio Rest

c) Tarjetas C.R.C.

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	2
Fecha de Reunión:	21-11-2016
Nombre de Clase:	DatosTaxi
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> La clase debe contener los métodos necesarios para obtener la información solicitada por el cliente desde la base de datos creada. Solamente consulta de datos. Relación con la capa de Acceso a datos. 	<ul style="list-style-type: none"> David Pungacho

Tabla 22. Tarjeta CRC1 Iteración 2

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	2
Fecha de Reunión:	21-11-2016
Nombre de Clase:	LogicaTaxi
Responsabilidades: <ul style="list-style-type: none"> • La clase debe contener los métodos necesarios para recoger la información de la capa de Datos. • Solamente consulta de datos. • Relación con la capa de Datos. 	Colaboradores: <ul style="list-style-type: none"> • David Pungacho

Tabla 23. Tarjeta CRC2 Iteración 2

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	2
Fecha de Reunión:	21-11-2016
Nombre de Clase:	Controlador
Responsabilidades: <ul style="list-style-type: none"> • La clase debe contener los métodos necesarios para recoger la información de la capa de Lógica. • La clase debe tener las características de servicio rest. • Solamente consulta de datos. • Relación con la capa de Datos. 	Colaboradores: <ul style="list-style-type: none"> • David Pungacho

Tabla 24. Tarjeta CRC3 Iteración 2

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

d) Funcionalidades extra

Funcionalidad Extra	
Nombre:	Interacción con cámara
Fecha inicio:	1-11-2016
Fecha fin:	6-01-2017
Integrantes:	David Pungacho
Descripción: <ul style="list-style-type: none">• La funcionalidad extra es investigar cómo utilizar el recurso de la cámara de los dispositivos que usan Android para poder capturar una imagen, transformarla a texto y pasarla como parámetro para poder realizar la consulta en la aplicación.	

Tabla 25. Funcionalidad extra Iteración 2

e) Refactorización

La refactorización fue realizada con éxito y se realizó una mejora en la fase de:

- a) Diseño simple

3.2.3. Desarrollo

a) Integración continua

- Creamos la conexión al modelo de datos creado en la iteración 1.

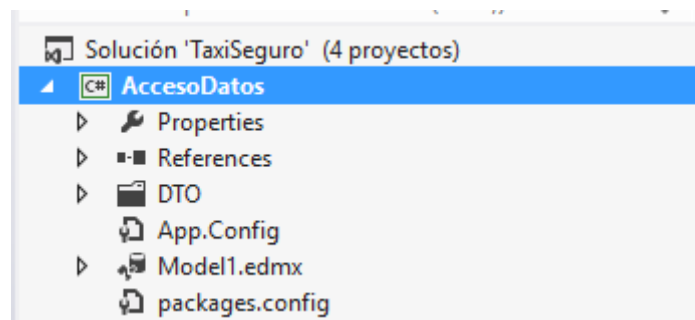


Ilustración 31. Conexión al modelo de datos

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

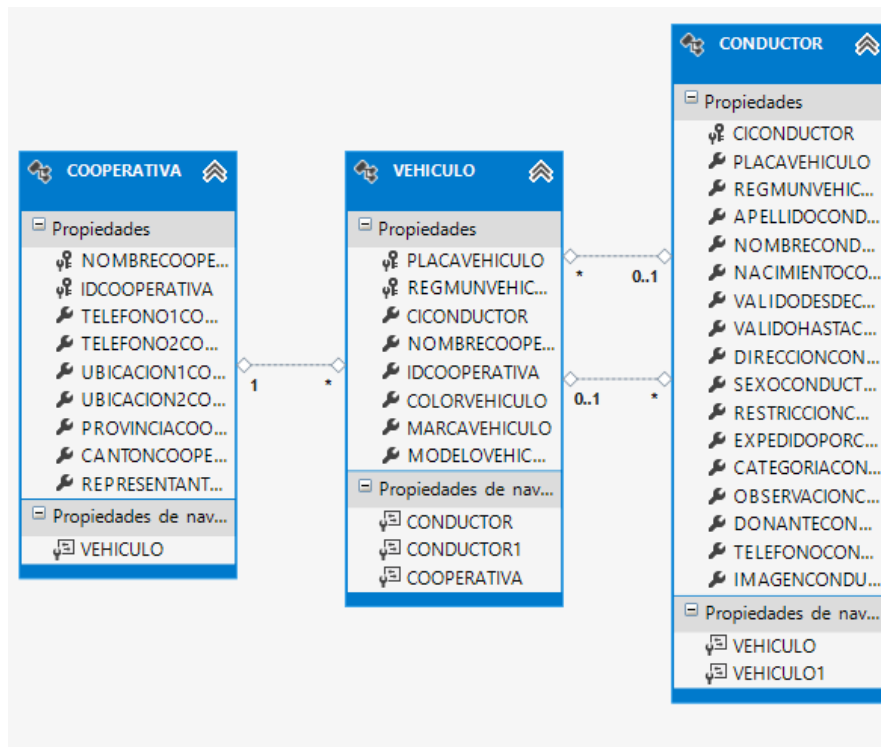


Ilustración 32. Modelo de datos como clases

- Creamos la capa de Datos que va a contener los métodos de consulta hacia la base de datos.

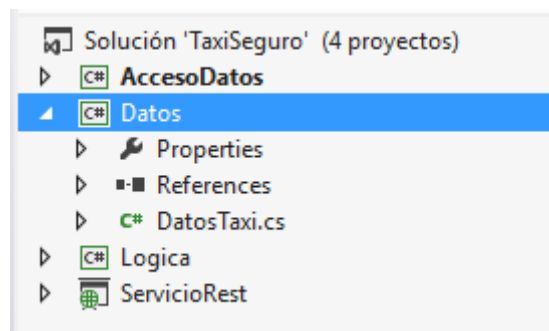


Ilustración 33. Capa de datos

- Creamos la capa de Lógica que va a contener los métodos necesarios para utilizar la información proveniente de la capa de Datos.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

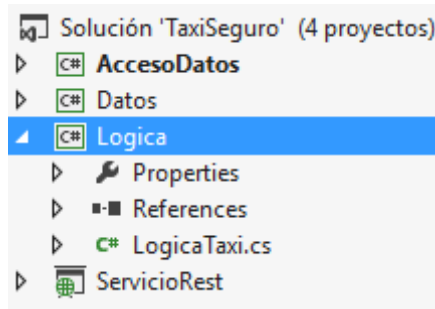


Ilustración 34. Capa lógica del negocio

- Finalmente creamos el servicio rest y el controlador.

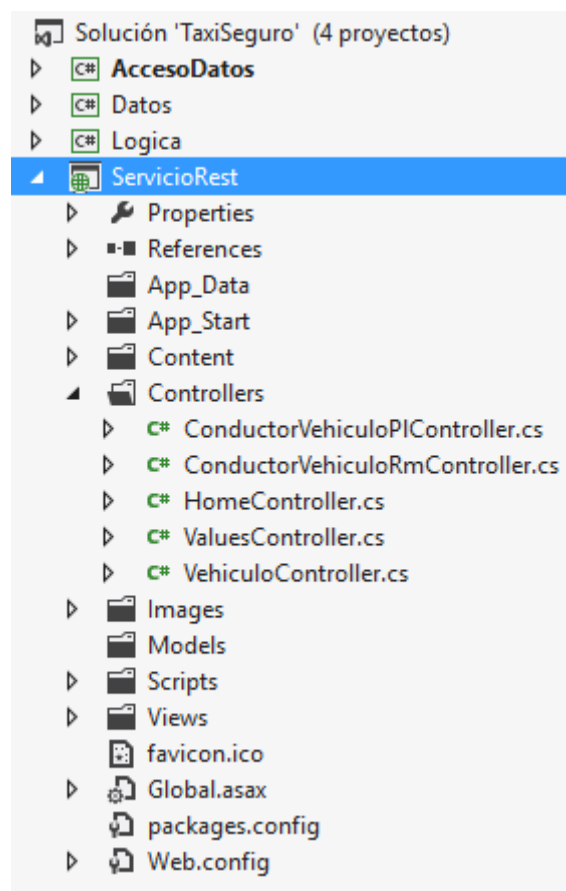


Ilustración 35. Servicio Rest

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

3.2.4. Pruebas

a) Pruebas de aceptación

Prueba de Aceptación	
Nombre:	Disponibilidad
Código:	2
Descripción: La información debe provenir de fuentes oficiales y debe estar actualizada al momento en que los usuarios accedan a la aplicación.	
Condiciones de ejecución:	<ul style="list-style-type: none"> • Debe existir información registrada en la base de datos. • Debe existir conexión a internet. • El servicio rest debe estar subido al Internet Information Service.
Entradas:	<ul style="list-style-type: none"> • Registro municipal del vehículo. • Placa del vehículo.
Pasos de ejecución:	<ol style="list-style-type: none"> 1. Obtener el registro municipal/placa del vehículo a consultar. 2. Abrir un navegador de preferencia Google Chrome. 3. Construir el link de consulta: http://localhost:8081/api/ConductorVehiculoPl/psc-4526 4. Verificar la información obtenida.
Resultado esperado:	Los datos que debe devolver la consulta son: <ul style="list-style-type: none"> • Apellido (conductor) • Nombre (conductor) • Código de la foto (conductor) • Marca • Modelo • Color • Placa • Registro municipal
Evaluación de la prueba:	La prueba realizada a la aplicación Rest que se construyó concluyó con éxito al 100% ya que muestra la información solicitada por el cliente y está disponible como servicio Rest.

Tabla 26. Prueba de aceptación Iteración 2

b) Pruebas unitarias

Para realizar las pruebas unitarias nos apoyamos en una funcionalidad de visual studio denominada Pruebas unitarias.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

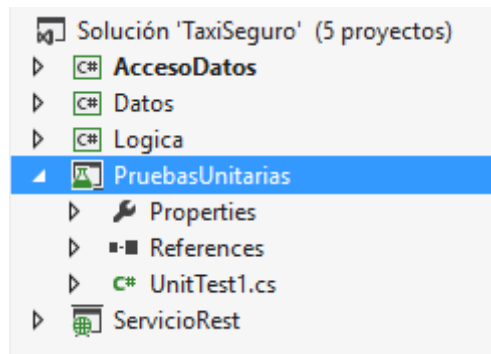


Ilustración 36. Pruebas unitarias

Con esta funcionalidad vamos a probar los métodos de consulta para el registro municipal y para la placa del vehículo.

Prueba Unitaria 1	
Numero de iteración:	2
Nombre de la prueba:	Query registro municipal
Descripción:	
<pre> public ConductorVehiculo seleccionarVehiculoPorRM(string rmVehiculo) { var consultaRes = from C in contexto.CONDUCTOR join V in contexto.VEHICULO on C.REGMUNVEHICULO equals V.REGMUNVEHICULO where C.REGMUNVEHICULO == rmVehiculo select new ConductorVehiculo { APELLIDOCONDUCTOR = C.APELLIDOCONDUCTOR, NOMBRECONDUCTOR = C.NOMBRECONDUCTOR, PLACAVEHICULO = C.PLACAVEHICULO, REGMUNVEHICULO = C.REGMUNVEHICULO, COLORVEHICULO = V.COLORVEHICULO, MARCAVEHICULO = V.MARCAVEHICULO, MODELOVEHICULO = V.MODELOVEHICULO, IMAGENCONDUCTOR = C.IMAGENCONDUCTOR }; ConductorVehiculo txResult = consultaRes.SingleOrDefault(); return txResult; } </pre>	
Resultado:	El método devuelve la información necesitada.

Tabla 27. Prueba unitaria 1 Iteración 2

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Prueba Unitaria 2	
Numero de iteración:	2
Nombre de la prueba:	Query placa del vehículo
Descripción:	
<pre> public ConductorVehiculo seleccionarVehiculoPorPL(string plVehiculo) { var consultaRes = from C in contexto.CONDUCTOR join V in contexto.VEHICULO on C.REGMUNVEHICULO equals V.REGMUNVEHICULO where C.PLACAVEHICULO == plVehiculo select new ConductorVehiculo { APELLIDOCONDUCTOR = C.APELLIDOCONDUCTOR, NOMBRECONDUCTOR = C.NOMBRECONDUCTOR, PLACAVEHICULO = C.PLACAVEHICULO, REGMUNVEHICULO = C.REGMUNVEHICULO, COLORVEHICULO = V.COLORVEHICULO, MARCAVEHICULO = V.MARCAVEHICULO, MODELOVEHICULO = V.MODELOVEHICULO, IMAGENCONDUCTOR = C.IMAGENCONDUCTOR }; ConductorVehiculo txResult = consultaRes.SingleOrDefault(); return txResult; } </pre>	
Resultado:	El método devuelve la información necesitada.

Tabla 28. Prueba unitaria 2 Iteración 2

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

3.3. Iteración 3

3.3.1. Planificación

a) Historia de usuario

Historia de Usuario	
Numero de Historia:	4
Fecha:	2015-08-10
Tipo (nueva/mejorar/cambiar):	nueva
Nombre de Historia:	Dispositivos Android
Usuario:	Clientes de taxis
Prioridad en Negocio (alta/media/baja):	Alta
Riesgo (alto/medio/bajo):	Alto
Iteración Asignada:	3
Descripción: La aplicación que va a ser usada por los clientes debe estar dirigida para dispositivos que usan Android.	

Cliente

Responsable

Tabla 29. Historia de usuario 4 Iteración 3

b) Planificación de lanzamiento

Release Planning						
Versión del sistema	Fecha publicación versión	Prioridad de historia	Numero de historia	Nombre de historia	Tiempo de implementación	# de personas desarrollo
1.0	Febrero-2017	Alta	1	Taxi seguro	12 semanas	1
1.0	Febrero-2017	Media	2	Información oficial	3 semanas	1
1.0	Febrero-2017	Alta	3	Disponibilidad	3 semanas	1
1.0	Febrero-2017	Alta	4	Dispositivos Android	3 semanas	1
1.0	Febrero-2017	Baja	5	Interacción con cámara	3 semanas	1

Tabla 30. Planificación de lanzamiento Iteración 3

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

c) Plan de iteraciones

Plan de Iteraciones					
Numero de Iteración	Fecha inicio	Fecha fin	Numero de historia	Nombre de historia	Responsable
1	31-10-2016	18-11-2016	2	Información oficial	David Pungacho
2	21-11-2016	9-12-2016	3	Disponibilidad	David Pungacho
3	12-12-2016	6-01-2017	4	Dispositivos Android	David Pungacho
4	9-01-2017	27-01-2017	5	Interacción con cámara	David Pungacho

Tabla 31. Plan de iteraciones Iteración 3

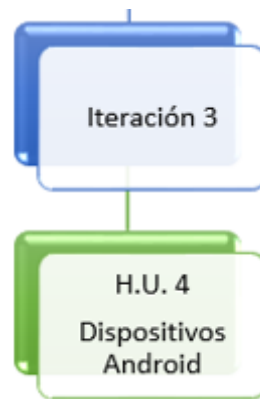


Ilustración 37. Iteración 3

3.3.2. Diseño

a) Diseño simple

La interfaz de usuario debe ser simple, amigable y fácil de acceder para el usuario, debido a que el objetivo principal es consultar la información del taxi lo más rápido posible. Para resolver esta necesidad se ha propuesto crear una sola pantalla donde se va a realizar la consulta y se va a mostrar la información.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

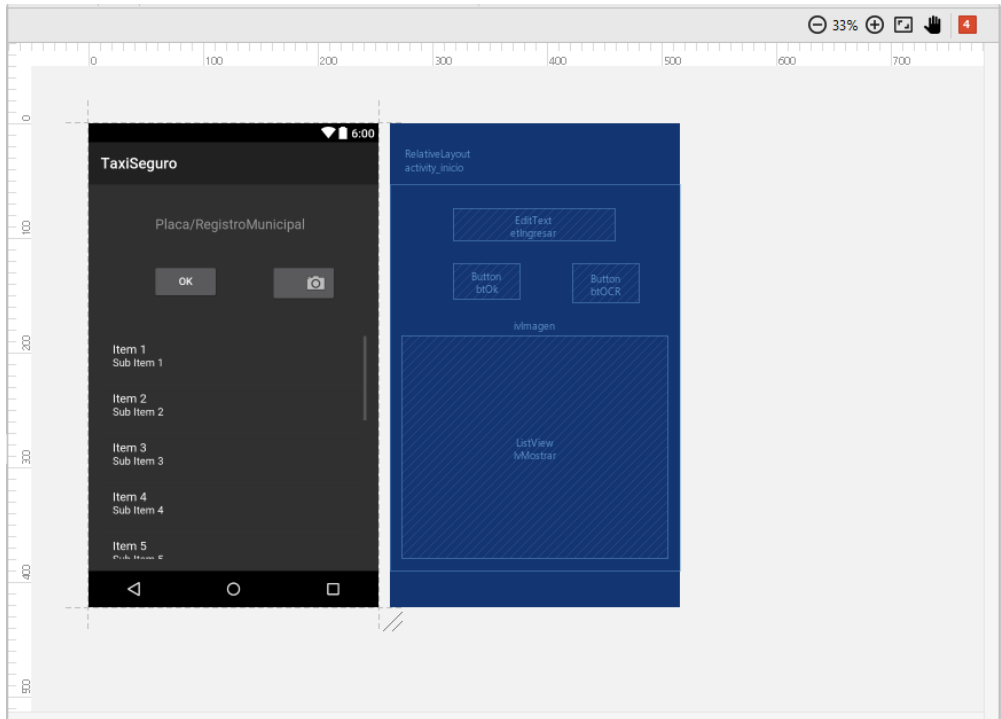


Ilustración 38. Diseño interfaz Android

b) Metáfora del sistema

Comportamiento del cliente Android dentro del sistema (parte resaltada).

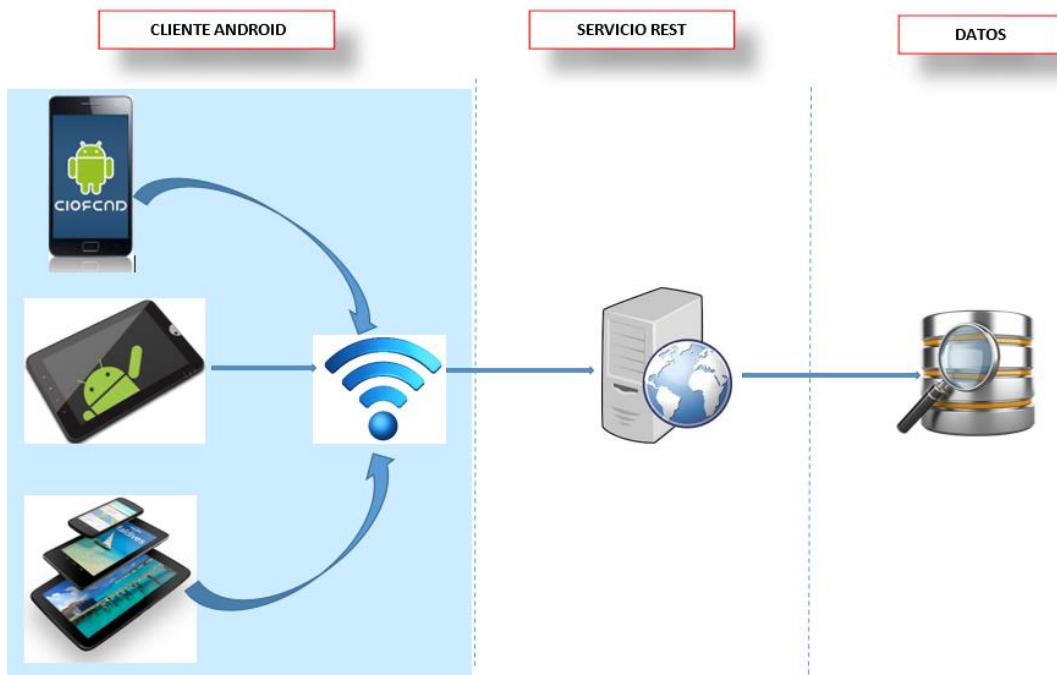


Ilustración 39. Metáfora del sistema Cliente Android

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

c) Tarjetas C.R.C.

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	3
Fecha de Reunión:	12-12-2016
Nombre de Clase:	Inicio
Responsabilidades: <ul style="list-style-type: none"> • La clase deberá controlar todos los componentes de la interfaz gráfica. • La clase deberá tener los "Action Listener" necesarios para que la aplicación realice lo solicitado. • La clase deberá validar los resultados obtenidos. • La clase debe de ser a prueba de usuarios. 	Colaboradores: <ul style="list-style-type: none"> • David Pungacho

Tabla 32. Tarjeta CRC1 Iteración 3

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	3
Fecha de Reunión:	12-12-2016
Nombre de Clase:	BuscarPorPlaca
Responsabilidades: <ul style="list-style-type: none"> • La clase deberá ser de tipo Asíncrona para que soporte conexiones http. • La clase deberá consumir el servicio Rest utilizando como parámetro la placa del vehículo. • La clase deberá deserializar el objeto xml obtenido y colocar la información de una manera visible para el usuario. • La clase deberá tomar el id de la foto del conductor y deberá construir un URL para poder traer una imagen de un servidor de imágenes dentro del IIS. 	Colaboradores: <ul style="list-style-type: none"> • David Pungacho

Tabla 33. Tarjeta CRC2 Iteración 3

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	3
Fecha de Reunión:	12-12-2016
Nombre de Clase:	BuscarPorRegistroMunicipal
Responsabilidades: <ul style="list-style-type: none"> • La clase deberá ser de tipo Asíncrona para que soporte conexiones http. • La clase deberá consumir el servicio Rest utilizando como parámetro el registro municipal. • La clase deberá deserializar el objeto xml obtenido y colocar la información de una manera visible para el usuario. • La clase deberá tomar el id de la foto del conductor y deberá construir un URL para poder traer una imagen de un servidor de imágenes dentro del IIS. 	Colaboradores: <ul style="list-style-type: none"> • David Pungacho

Tabla 34. Tarjeta CRC3 Iteración 3

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	3
Fecha de Reunión:	12-12-2016
Nombre de Clase:	CargarImagenes
Responsabilidades: <ul style="list-style-type: none"> • La clase deberá ser de tipo Asíncrona para que soporte conexiones http. • La clase deberá usar el URL creado por BuscarPorPlaca y BuscarPorRegistroMunicipal. • La clase deberá traer la imagen desde un servidor de imágenes en IIS y presentarla en la interfaz del usuario. 	Colaboradores: <ul style="list-style-type: none"> • David Pungacho

Tabla 35. Tarjeta CRC4 Iteración 3

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

d) Funcionalidades extra

Funcionalidad Extra	
Nombre:	Interacción con cámara
Fecha inicio:	1-11-2016
Fecha fin:	6-01-2017
Integrantes:	David Pungacho
Descripción: <ul style="list-style-type: none">• La funcionalidad extra es investigar cómo utilizar el recurso de la cámara de los dispositivos que usan Android para poder capturar una imagen, transformarla a texto y pasarla como parámetro para poder realizar la consulta en la aplicación.	

Tabla 36. Funcionalidad extra Iteración 3

e) Refactorización

La refactorización fue realizada con éxito y se mejoraron las siguientes fases:

- c) Tarjetas C.R.C

3.3.3. Desarrollo

a) Integración continua

Para el desarrollo de la interfaz de usuario con todos sus componentes nos basamos en el requisito fundamental del usuario que es la simplicidad de la misma. La pantalla principal es donde se ingresa el parámetro de búsqueda y también se presenta la información, esta pantalla cuenta con: un campo de texto para escribir, 2 botones (ok y cámara) y una lista de elementos donde se va a mostrar los datos obtenidos.

Todo el código fue desarrollado en base a las Tarjetas de Clase Responsabilidad Colaborador y esta adjuntado como anexo a este trabajo.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

3.3.4. Pruebas

a) Pruebas de aceptación

Prueba de Aceptación	
Nombre:	Dispositivos Android
Código:	3
Descripción: La aplicación que va a ser usada por los clientes debe estar dirigida para dispositivos que usan Android.	
Condiciones de ejecución:	<ul style="list-style-type: none"> • Debe existir información registrada en la base de datos. • Debe existir conexión a internet. • El servicio rest debe estar subido al Internet Information Service. • El acceso a IIS debe estar habilitado. • Para que la información se muestre el parámetro de búsqueda debe estar registrado en la base de datos.
Entradas:	<ul style="list-style-type: none"> • Registro municipal del vehículo. • Placa del vehículo.
Pasos de ejecución:	<ol style="list-style-type: none"> 1. Obtener el registro municipal/placa del vehículo a consultar. 2. Ingresar el registro municipal/placa del vehículo en el campo de texto de la primera pantalla 3. Visualizar la información.
Resultado esperado:	<p>Los datos que debe devolver la consulta son:</p> <ul style="list-style-type: none"> • Foto (conductor) • Apellido (conductor) • Nombre (conductor) • Marca • Modelo • Color • Placa • Registro municipal
Evaluación de la prueba:	La prueba realizada a la aplicación Android que se construyó concluyó con éxito al 100% ya que muestra la información solicitada por el cliente y está disponible para dispositivos Android.

Tabla 37. Pruebas de aceptación Iteración 3

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

b) Pruebas unitarias

Las pruebas unitarias se las realizaron apoyándose en la herramienta de Android studio que cuenta con una funcionalidad para poder probar los métodos de importancia, devolviendo un archivo html con la descripción de la prueba. Los métodos que se probaron fueron las dos consultas por placa de vehículo o por registro municipal, el resultado obtenido fue.

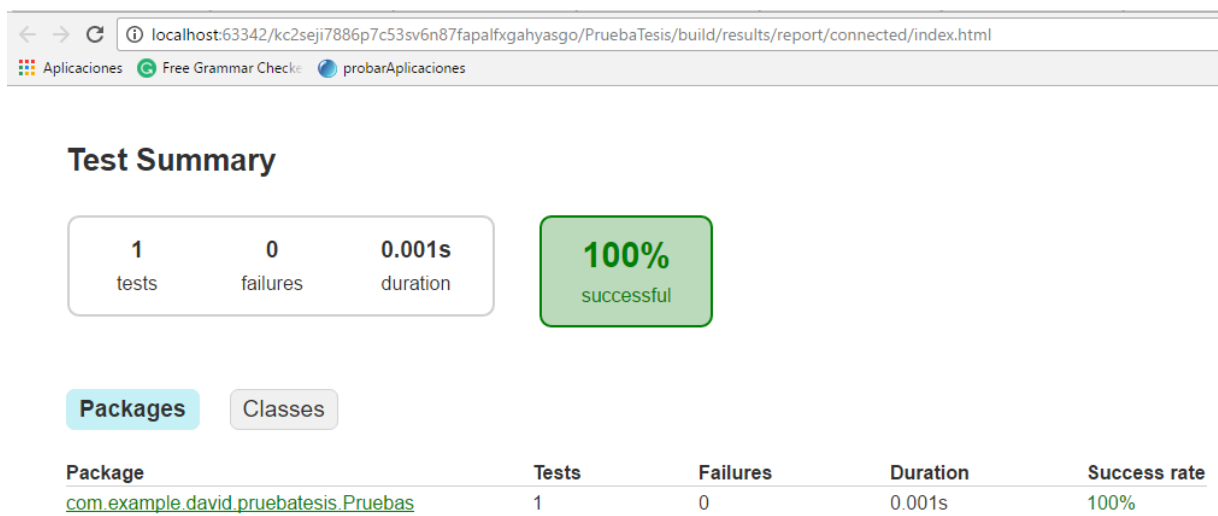


Ilustración 40. Prueba unitaria Android

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

3.4. Iteración 4

3.4.1. Planificación

a) Historia de usuario

Historia de Usuario	
Numero de Historia:	5
Fecha:	2015-08-10
Tipo (nueva/mejorar/cambiar):	nueva
Nombre de Historia:	Interacción con cámara
Usuario:	Clientes de taxis
Prioridad en Negocio (alta/media/baja):	Baja
Riesgo (alto/medio/bajo):	Alto
Iteración Asignada:	4
Descripción: La aplicación debe permitir al usuario capturar con la cámara del dispositivo la información para realizar la consulta.	

Cliente

Responsable

Tabla 38. Historia de Usuario Iteración 4

b) Planificación de lanzamiento

Release Planning						
Versión del sistema	Fecha publicación versión	Prioridad de historia	Numero de historia	Nombre de historia	Tiempo de implementación	# de personas desarrollo
1.0	Febrero-2017	Alta	1	Taxi seguro	12 semanas	1
1.0	Febrero-2017	Media	2	Información oficial	3 semanas	1
1.0	Febrero-2017	Alta	3	Disponibilidad	3 semanas	1
1.0	Febrero-2017	Alta	4	Dispositivos Android	3 semanas	1
1.0	Febrero-2017	Baja	5	Interacción con cámara	3 semanas	1

Tabla 39. Planificación de lanzamiento Iteración 4

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

c) Plan de iteraciones

Plan de Iteraciones					
Numero de Iteración	Fecha inicio	Fecha fin	Numero de historia	Nombre de historia	Responsable
1	31-10-2016	18-11-2016	2	Información oficial	David Pungacho
2	21-11-2016	9-12-2016	3	Disponibilidad	David Pungacho
3	12-12-2016	6-01-2017	4	Dispositivos Android	David Pungacho
4	9-01-2017	27-01-2017	5	Interacción con cámara	David Pungacho

Tabla 40. Plan de iteraciones Iteración 4

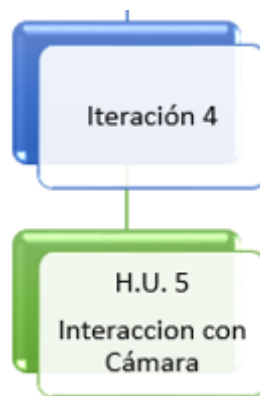


Ilustración 41. Iteración 4

3.4.2. Diseño

a) Diseño simple

La funcionalidad pico del proyecto es la interacción con la cámara de fotos, esto corresponde al reconocimiento de imágenes, para ello la segunda pantalla es la interfaz de la cámara de fotos que se va a abrir si el usuario da click al botón con el icono de cámara de fotos.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

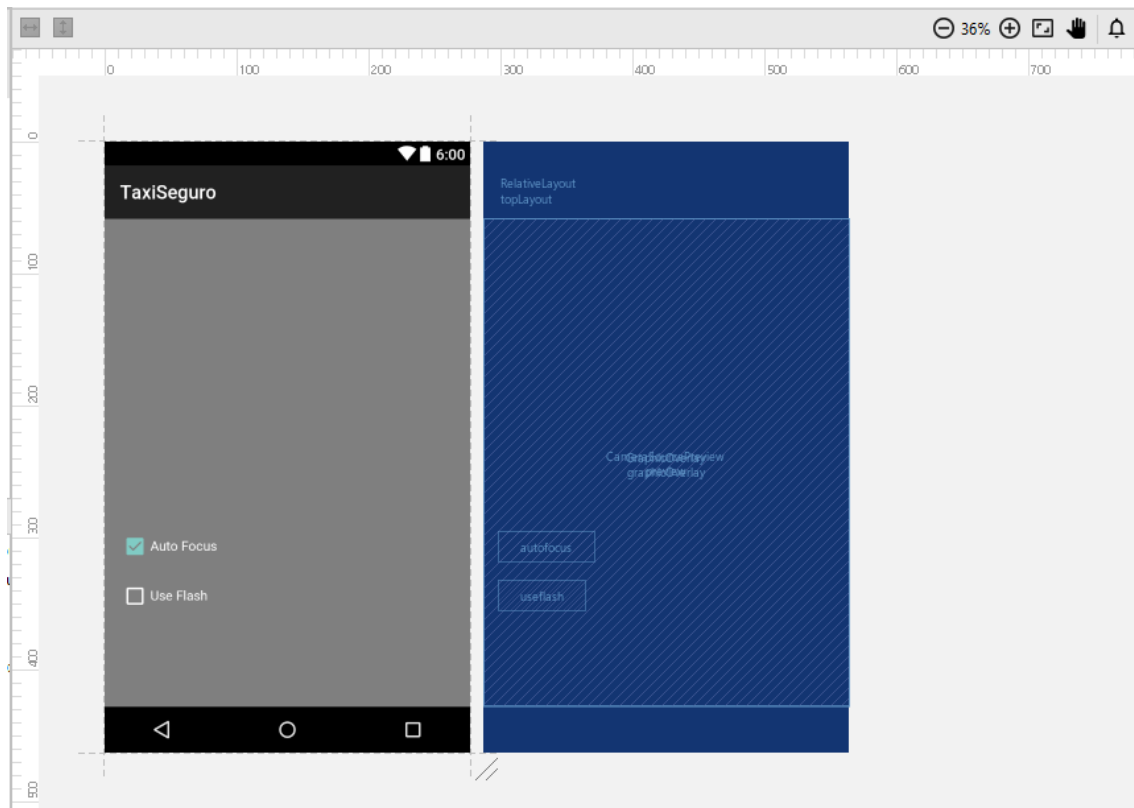


Ilustración 42. Diseño interfaz Android OCR

b) Metáfora del sistema

Comportamiento del cliente Android dentro del sistema (parte resaltada).

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

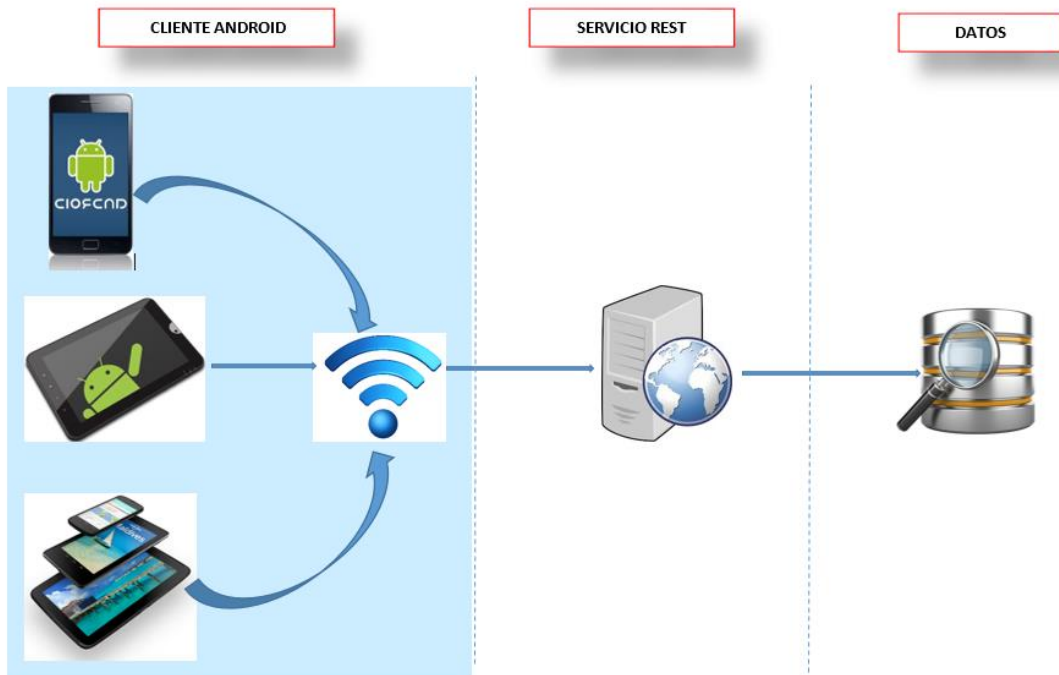


Ilustración 43. Metáfora del sistema Cliente Android OCR

c) Tarjetas C.R.C.

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	4
Fecha de Reunión:	09-01-2017
Nombre de Clase:	OcrCaptureActivity
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> La clase deberá tener la configuración necesaria para soportar el api de google OCR visión. La clase deberá ser de tipo AppCompatActivity para que pueda manejar la interfaz gráfica de la cámara. La clase deberá solicitar los permisos necesarios al usuario de la aplicación para poder ser utilizada. La clase deberá tener mensajes de alerta informando que está sucediendo en la aplicación, para que sea amigable al usuario. La clase deberá tener sobrescritos todos los métodos necesarios para su correcto funcionamiento. 	<ul style="list-style-type: none"> David Pungacho

Tabla 41. Tarjeta CRC1 Iteración 4

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	4
Fecha de Reunión:	09-01-2017
Nombre de Clase:	OcrDetectorProcessor
Responsabilidades: <ul style="list-style-type: none"> • La clase deberá tener la configuración necesaria para soportar el api de google OCR visión. • La clase deberá ser de tipo Detector.Processor para que pueda manejar las imágenes detectadas por la cámara. • La clase deberá solicitar los permisos necesarios al usuario de la aplicación para poder ser utilizada. • La clase deberá tener mensajes de alerta informando que está sucediendo en la aplicación, para que sea amigable al usuario. • La clase deberá tener sobrescritos todos los métodos necesarios para su correcto funcionamiento. 	Colaboradores: <ul style="list-style-type: none"> • David Pungacho

Tabla 42. Tarjeta CRC2 Iteración 4

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Tarjeta CRC	
Nombre del Proyecto:	Taxi Seguro
Numero de Iteración:	4
Fecha de Reunión:	09-01-2017
Nombre de Clase:	OcrGraphic
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> • La clase deberá tener la configuración necesaria para soportar el api de google OCR visión. • La clase deberá ser de tipo GraphicOverlay.Graphic para que pueda dibujar un rectángulo sobre la información obtenida por la cámara. • La clase deberá solicitar los permisos necesarios al usuario de la aplicación para poder ser utilizada. • La clase deberá tener mensajes de alerta informando que está sucediendo en la aplicación, para que sea amigable al usuario. • La clase deberá tener sobrescritos todos los métodos necesarios para su correcto funcionamiento. 	<ul style="list-style-type: none"> • David Pungacho

Tabla 43. Tarjeta CRC3 Iteración 4

d) Funcionalidades extra

Funcionalidad Extra	
Nombre:	Interacción con cámara
Fecha inicio:	1-11-2016
Fecha fin:	6-01-2017
Integrantes:	David Pungacho
Descripción:	<ul style="list-style-type: none"> • La funcionalidad extra es investigar cómo utilizar el recurso de la cámara de los dispositivos que usan Android para poder capturar una imagen, transformarla a texto y pasarla como parámetro para poder realizar la consulta en la aplicación.

Tabla 44. Funcionalidad extra Iteración 4

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

e) Refactorización

La refactorización fue realizada con éxito y se mejoraron las siguientes fases:

- a) Diseño simple

3.4.3. Desarrollo

a) Integración continua

Todo el código fue desarrollado en base a las Tarjetas de Clase Responsabilidad Colaborador y esta adjuntado como anexo a este trabajo.

3.4.4. Pruebas

a) Pruebas de aceptación

Prueba de Aceptación	
Nombre:	Interacción con cámara
Código:	4
Descripción: La aplicación debe permitir al usuario capturar con la cámara del dispositivo la información para realizar la consulta.	
Condiciones de ejecución:	<ul style="list-style-type: none">• Debe existir información registrada en la base de datos.• Debe existir conexión a internet.• El servicio rest debe estar subido al Internet Information Service.• El acceso a IIS debe estar habilitado.• Para que la información se muestre el parámetro de búsqueda debe estar registrado en la base de datos.• El área que va a ser escaneada debe estar iluminada.• La placa o registro municipal que va a ser leído por la cámara debe ser legible.
Entradas:	<ul style="list-style-type: none">• Registro municipal del vehículo.• Placa del vehículo.
Pasos de ejecución:	<ol style="list-style-type: none">1. Abrir la aplicación.2. Dar clic al botón con imagen de cámara.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

	<ol style="list-style-type: none"> 3. Enfocar con la cámara el objeto que se quiere leer. 4. Una vez seguro del objeto deseado, dar click sobre el rectángulo. 5. Si la información que se escribe en el cuadro de texto de la pantalla 1 es la deseada, aplastar ok, caso contrario paso 3. 6. Analizar la información.
Resultado esperado:	<p>Los datos que debe devolver la consulta son:</p> <ul style="list-style-type: none"> • Foto (conductor) • Apellido (conductor) • Nombre (conductor) • Marca • Modelo • Color • Placa • Registro municipal
Evaluación de la prueba:	<p>La prueba realizada con la interacción de la cámara concluyo con un factor de éxito del 80% debido a que el api liberado por google es una tecnología nueva(2015) y porque depende de factores externos como la legibilidad de las placas/registro municipal o la luminosidad del lugar.</p>

Tabla 45. Prueba de aceptación Iteración 4

b) Pruebas unitarias

Una vez que la funcionalidad extra o pico concluyo al 100% vamos a realizar las pruebas unitarias enfocadas a diferentes dispositivos que usen Android.

Marca	OCR
Samsung Galaxy S7 Edge	90%
Samsung Galaxy S6	90%
Samsung Galaxy S5	90%
Samsung Galaxy S4 mini	80%
Samsung Galaxy S3	80%
HTC one	90%

Tabla 46. Prueba unitaria Iteración 4

Es importante recalcar que se deben tomar en cuenta factores primordiales como lo son: la iluminación, la distancia, el hardware del dispositivo y que el api con el que está construido el reconocimiento de imágenes fue liberado a finales del año 2015 por google. Hay que tomar en

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

cuenta los factores externos, en algunos casos la poca legibilidad de las placas o registros municipales de los vehículos, debido a las malas condiciones de los mismos.

4. Conclusiones y Recomendaciones

4.1. Conclusiones

- Al final de todo el proceso se pudo desarrollar un prototipo de aplicación, basada en Android que consume un servicio Rest y permite consultar la información ya sea por el ingreso manual de texto o mediante la captura de imágenes con conversión a texto utilizando la cámara de fotos de los dispositivos.
- Para concluir el presente trabajo es importante mencionar que la mayoría del conocimiento adquirido en el transcurso de la carrera se puso en práctica en este proyecto, desde como estructurar una base de datos, como crear un servicio hasta la aplicación de estándares de programación.
- Para poder concluir el proyecto, también se puso en práctica la parte investigativa que la universidad a lo largo de la carrera nos enseñó, la investigación fue ocupada en su totalidad en como consumir el servicio rest, es decir, la parte de Android y el reconocimiento de caracteres.
- El poder de la tecnología ha crecido exponencialmente, con trabajos como este prototipo realizado, se pueden generar ideas que brinden un aporte tecnológico a la sociedad en el ámbito de la seguridad en el transporte público.
- Toda la formación brindada por la Universidad Católica lo largo de la carrera de Ingeniería en Sistemas y Computación, es una información amplia y general, ya que no se especializa en un solo ámbito, es decir la universidad desarrolla en nosotros varias habilidades, pasando por la programación, las bases de datos, la gerencia de proyectos, contabilidad, finanzas, procesos y calidad, entre otras. Esto es una ventaja

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

ya que brinda una visión sobre todos los campos de aplicación que tiene esta hermosa carrera, pudiendo especializarse sobre alguno de ellos en un estudio de cuarto nivel.

4.2. Recomendaciones

- En base a la experiencia obtenida en el desarrollo de este proyecto se puede recomendar a las autoridades pertinentes la estandarización de toda la información del vehículo utilizado como taxi y de su conductor con el fin de crear un código de escaneo QR que facilite la verificación de los datos.
- La fase de reconocimiento de caracteres está construida con un api de código abierto creado por Google liberada a finales del año 2015. Se recomienda investigar a futuro más sobre este api una vez que se liberen versiones más estables.
- El programa está construido para soportar una versión 2. Se recomienda que con apoyo de la universidad se realicen los trámites necesarios para obtener los permisos ante los organismos de control correspondientes para poder acceder a la información original y construir una versión dos de este proyecto.
- Para la parte del desarrollo en Android se recomienda aplicar en una versión 2 del proyecto la metodología de “Experiencia de Usuario” lo cual puede ayudar a que la aplicación tenga un mayor impacto en la parte comercial.
- Se recomienda la ampliación del campo de estudio dentro de la carrera de sistemas sobre el desarrollo de dispositivos móviles, ya que es un mercado relativamente joven, que crece muy rápidamente y que ha generado un gran impacto en la sociedad.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

5. Bibliografía

Sitios Web

Android. (s.f.). *Android Developer*. Obtenido de <https://developer.android.com/index.html>

Android Developer. (s.f.). Obtenido de <https://developer.android.com/index.html>

App Quality Alliance. (2014). Obtenido de <http://www.appqualityalliance.org/>

CCM. (Diciembre de 2016). Obtenido de <http://es.ccm.net/contents/223-ciclo-de-vida-del-software>

Ecuador, G. N. (s.f.). *Agencia Nacional de Transito*. Obtenido de <http://www.ant.gob.ec/>

Extreme Programming: A gentle introduction. (s.f.). Obtenido de <http://www.extremeprogramming.org/>

Google. (2015). *Mobile Vision*. Obtenido de <https://developers.google.com/vision/>

Ingenieria de Software. (s.f.). Obtenido de http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html

Microsoft. (29 de febrero de 2016). *Microsoft Doc*. Obtenido de <https://docs.microsoft.com/es-es/azure/app-service-web/web-sites-dotnet-rest-service-aspnet-api-sql-database>

Microsoft. (s.f.). *IIS*. Obtenido de <https://www.iis.net/>

Quito, M. d. (s.f.). *Municipio de Quito*. Obtenido de <http://www.quito.gob.ec/>

Rouse, M. (2006). *Search Software Quality*. Obtenido de <http://searchsoftwarequality.techtarget.com/definition/Extreme-Programming>

Seta, L. D. (05 de febrero de 2010). *Dos Ideas*. Obtenido de <http://www.dosideas.com/noticias/metodologias/822-una-introduccion-a-extreme-programming.html>

Torres, R. (5 de Enero de 2015). *Desarrolladores*. Obtenido de <http://desarrolladores.me/2015/01/c-creando-nuestro-primer-servicio-web-restful-api/>

Tutorials point. (s.f.). Obtenido de https://www.tutorialspoint.com/extreme_programming/extreme_programming_introduction.htm

¿Qué es Android? (s.f.). Obtenido de <http://www.xatakandroid.com/sistema-operativo/que-es-android>

Alerta en Linea. (Septiembre de 2011). Obtenido de <https://www.alertaenlinea.gov/articulos/s0018-aplicaciones-m%C3%B3viles-qu%C3%A9-son-y-c%C3%B3mo-funcionan>

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

Android 5.0, Lollipop. (s.f.). Obtenido de <https://www.android.com/versions/lollipop-5-0/>

App Design Book. (s.f.). Obtenido de <http://appdesignbook.com/es/contenidos/las-aplicaciones/>

Banda, J. (20 de Abril de 2015). *El Bit Pragmatico*. Obtenido de <http://www.dineroenimagen.com/blogs/el-bit-pragmatico/como-iniciar-en-el-desarrollo-de-aplicaciones-moviles/54254>

Características. (s.f.). Obtenido de <http://androidos.readthedocs.org/en/latest/data/caracteristicas/>

Entorno cliente/servidor. (s.f.). Obtenido de <http://es.ccm.net/contents/148-entorno-cliente-servidor>

Información general sobre las aplicaciones distribuidas. (s.f.). Obtenido de [https://msdn.microsoft.com/es-es/library/aa291332\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa291332(v=vs.71).aspx)

LoremSolutions. (23 de diciembre de 2012). *SlideShare*. Obtenido de es.slideshare.net/LoremSolutions/proceso-de-desarrollo-de-una-aplicacion-mvil

QODE. (s.f.). Obtenido de <http://qode.pro/blog/pasos-para-el-desarrollo-de-aplicaciones-moviles/>

Krug, S. (s.f.). *No me hagas pensar*. Pearson.

Videos

o, v. d. (29 de diciembre de 2013). *Youtube*. Obtenido de Consumir WebService NET con Android: <https://www.youtube.com/watch?v=evx1S-Ssqfg>

Avidosprogramer. (17 de enero de 2014). *Youtube*. Obtenido de Introducción a ASP.NET Web API 2-1 Como Crear un Servicio REST en Visual Studio 2013: https://www.youtube.com/watch?v=1tc_HUGL7iM

Libros

Sierra, J. C. (2012). *Visual C#: Interfaces graficas y aplicaciones para Internet con WPF, WCF y Silverlight*. Madrid: RA-MA.

Soriano, J. E. (2012). *El gran libro de programacion avanzada con Android*. Barcelona: Alfaomega.

Miguel, R. M. (2012). *Desarrollo de Aplicaciones para Android*. Madrid: RA-MA.

Lequerica, J. R. (2016). *Desarrollo de aplicaciones para Android*. Madrid: Anaya.

Arana, B. O. (s.f.). *Desarrollo de aplicaciones C# con Visual Studio.NET*. Madrid: Altaria.

Desarrollo de un prototipo de una aplicación, basada en Android que permita autenticar el taxi y su conductor.

6. Anexos

- Código del sistema (CD).
- Documento de pruebas generales en varios dispositivos (Excel).