

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES

TEMA:

APLICACIÓN MÓVIL PARA APOYAR LA GESTIÓN DE LA
SEGURIDAD Y SALUD EN EMPRESAS CEMENTERAS USANDO LA
METODOLOGÍA ÁGIL MOBILE-D

AUTOR:

ALEXANDER OSWALDO ORBES ORBES

DIRECTOR:

ING. DIEGO JAVIER TREJO ESPAÑA

IBARRA , 2018



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO		
CÉDULA DE IDENTIDAD:	100359342-1	
APELLIDOS Y NOMBRES:	ORBES ORBES ALEXANDER OSWALDO	
DIRECCIÓN:	Colimbuela – Imantag – Cotacachi	
EMAIL:	alex_251093@hotmail.com	
TELÉFONO FIJO:	TELÉFONO	0989158376
	MÓVIL:	

DATOS DE LA OBRA	
TÍTULO:	APLICACIÓN MÓVIL PARA APOYAR LA GESTIÓN DE LA SEGURIDAD Y SALUD EN EMPRESAS CEMENTERAS USANDO LA METODOLOGÍA ÁGIL MOBILE-D

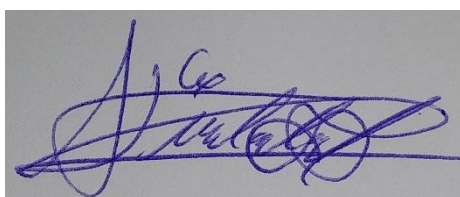
AUTOR (ES):	ORBES ORBES ALEXANDER OSWALDO	
FECHA:	01/08/2018	
SOLO PARA TRABAJOS DE GRADO		
PROGRAMA:	PREGRADO	POSGRADO
	<input checked="" type="checkbox"/>	<input type="checkbox"/>
TITULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES	
ASESOR/DIRECTOR:	MCs. DIEGO TREJO	

2. CONSTANCIAS

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

Ibarra, a los 01 días del mes de Agosto de 2018

EL AUTOR:



(Firma).....

Nombre: Alexander Oswaldo Orbes Orbes



UNIVERSIDAD TÉCNICA DEL NORTE

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, **ALEXANDER OSWALDO ORBES ORBES**, con cédula de identidad Nro. **100359342-1**, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor (es) de la obra o trabajo de grado denominado: **“APLICACIÓN MÓVIL PARA APOYAR LA GESTIÓN DE LA SEGURIDAD Y SALUD EN EMPRESAS CEMENTERAS USANDO LA METODOLOGÍA ÁGIL MOBILE-D”**, que ha sido desarrollado para optar por el título de: **INGENIERO EN SISTEMAS COMPUTACIONALES** en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Ibarra, a los 01 días del mes de Agosto de 2018

(Firma)

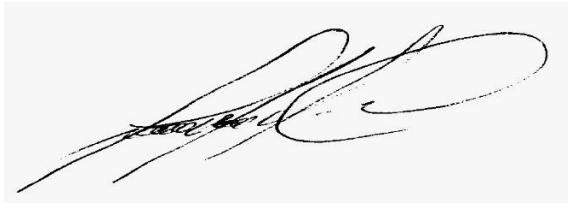
Nombre: Alexander Oswaldo Orbes Orbes

Cédula: 100359342-1

Ibarra, 01 de Agosto de 2018

CERTIFICACIÓN DIRECTOR

Certifico que la tesis realizada por el señor Alexander Oswaldo Orbes Orbes se ha trabajado en el desarrollo del trabajo de grado “APLICACIÓN MÓVIL PARA APOYAR LA GESTIÓN DE LA SEGURIDAD Y SALUD EN EMPRESAS CEMENTERAS USANDO LA METODOLOGÍA ÁGIL MOBILE-D”, previo a la obtención del título de ingeniero en sistemas computacionales, realizándola con interés profesional y responsabilidad, lo cual certifico en honor a la verdad.

A handwritten signature in black ink, appearing to read 'Diego Trejo', is centered on a light gray rectangular background.

Ing. Diego Trejo

Director de Tesis

Perugachi, 25 de junio de 2018

CERTIFICADO

La Empresa **UNACEM ECUADOR S.A.**, certifica que el **Sr ALEXANDER OSWALDO ORBES ORBES**, portador de la cédula de ciudadanía No. 1003593421, desarrolló de manera satisfactoria y según la planificación establecida, el proyecto académico de tesis titulado: "Aplicación Móvil para apoyar la gestión de la seguridad y salud en empresas cementeras usando la metodología Ágil Mobile-D".

El señor Marlon Cadena Subia, Jefe de Seguridad Industrial fue designado Tutor Interno para el mencionado proyecto académico conforme se había notificado en comunicación de fecha 5 de mayo de 2017, al estudiante Sr. Alexander Orbes.

Lo certifico,

 **UNACEM**
ECUADOR

RECURSOS HUMANOS

Geovany Estrella Silva
Gerente de Talento Humano
UNACEM Ecuador S.A.
geovanny.estrella@unacem.com.ec

DEDICATORIA

Dedico enteramente este trabajo a mis padres José Luis Orbes Chávez y María Tránsito Obres Bolaños, quienes me han apoyado incondicionalmente en todo momento de mi vida.

Alexander Orbes

AGRADECIMIENTO

Agradezco a todos mis profesores que me hayan ayudado de manera directa o indirecta mi formación a lo largo de mi vida universitaria. Y de manera especial a mi tutor Diego Trejo que me ayudó a culminar este presente trabajo. También a mis hermanos Diego y Cristian con quienes nos hemos ayudado en el transcurso de la vida personal y universitaria, por último, a mi amigo Carlos Proaño quien de igual manera me ha apoyado en la culminación de la tesis.

Alexander Orbes

ÍNDICE DE CONTENIDOS

BIBLIOTECA UNIVERSITARIA	i
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO.....	iii
CERTIFICACIÓN DIRECTOR	iv
DEDICATORIA.....	vi
AGRADECIMIENTO.....	vii
ÍNDICE DE CONTENIDOS.....	1
ÍNDICE DE FIGURAS	8
ÍNDICE DE TABLAS.....	11
RESUMEN.....	13
ABSTRACT	14
CAPITULO I.....	0
1 Introducción.....	0
1.1 Antecedentes	0
1.2 Situación actual.....	1
1.3 Prospectiva.....	2
1.4 Planteamiento del problema.....	2
1.5 Objetivos.....	2
1.5.1 Objetivo General	2
1.5.2 Objetivos específicos.....	3
1.6 Alcance	3
1.7 Justificación	5
CAPÍTULO II.....	6
2 Marco teórico.....	6

2.1	Industrias cementeras en el Ecuador.....	6
2.2	Unacem	6
2.3	Seguridad y Salud Industrial.....	8
2.3.1	Condiciones y actos inseguros	9
2.3.2	ATS (Análisis de trabajo seguro).....	10
2.3.3	Estándares de seguridad industrial	11
2.4	El mundo de las aplicaciones móviles	11
2.4.1	Android.....	12
2.4.2	IOS	13
2.4.3	¿Android o iOS?.....	14
2.4.4	Android SDK.....	15
2.4.5	Android Studio	15
2.4.6	Gradle	16
2.4.7	Sincronización de datos.....	17
2.4.8	Sync adapter	19
2.4.8.1	Token de autenticación.....	19
2.4.8.2	Content providers.....	19
2.4.9	Modelos de comunicación.....	20
2.4.9.1	Modelo de conexión por intranet.....	20
2.4.9.2	Modelo de conexión por internet.....	20
2.4.10	Persistencia en Android.....	20
2.4.11	ORM.....	23
2.4.12	GreenDAO	24
2.4.12.1	Estructura de GreenDao.....	25

2.4.12.2	El Core de GreenDao.....	26
2.4.13	Material Design	26
2.5	Patrones de Diseño y Arquitectónicos en Android	27
2.5.1	Patrones de diseño	27
2.5.1.1	Patrón de diseño facade	28
2.5.1.2	Patrón de diseño observer.....	28
2.5.2	Patrones arquitectónicos.....	28
2.5.2.1	MVC (Modelo Vista Controlador)	28
2.5.2.2	MVP (Modelo Vista Presentador)	29
2.6	Metodologías	29
2.6.1	Metodologías ágiles.....	30
2.6.2	Metodologías ágiles en el desarrollo de aplicaciones móviles.....	30
2.6.3	Justificación de la metodología	31
2.6.4	Mobile-D	33
2.6.4.1	Exploración.....	33
2.6.4.2	Inicialización.....	34
2.6.4.3	Producción	35
2.6.4.4	Estabilización.....	35
2.6.4.5	Pruebas del Sistema	36
CAPÍTULO III		38
3	Desarrollo de la solución	38
3.1	Fase de exploración	38
3.1.1	Establecimiento de partes interesadas	38

3.1.2	Especificación de requerimientos.....	39
3.1.2.1	Requerimientos iniciales.....	39
3.1.2.2	Diagramas de casos de uso	39
3.1.2.3	Caso de uso módulo Login	40
3.1.2.4	Caso de uso Home	41
3.1.2.5	Caso de uso módulo incidentes registro, consulta y sincronización.....	42
3.1.2.6	Caso de uso registro, consulta y sincronización de IVL.....	43
3.1.2.7	Caso de uso registro, consulta y sincronización de planes vivenciales.	45
3.1.2.8	Caso de uso de inspecciones de área, nueva acción, subir registros.....	46
3.1.2.9	Diagrama caso de usos inspecciones estándar, agregar nuevo.	47
3.1.3	Diagrama de Componentes.	49
3.1.4	Requerimientos Funcionales	49
3.1.5	Módulos de la Aplicación	50
3.1.6	Establecimiento del Proyecto	52
3.2	Inicialización del proyecto	53
3.2.1	Diseño de la base de datos.....	53
3.2.2	Planificación de fases	55
3.2.3	Lista de verificación del Plan de Proyecto	58
3.3	Fase de Producción	59
3.3.1	Configuración del proyecto.....	59
3.3.2	Historia de usuario Login.....	60
3.3.2.1	Diseño de interfaz de login.	61
3.3.3	Historia de usuario Home.....	61

3.3.3.1	Diseño interfaz home	63
3.3.4	Historia de usuario incidentes	64
3.3.4.1	Diseño interfaz incidentes.....	65
3.3.5	Historia de usuario IVL.....	66
TABLA 21. Historia de usuario IVL.....		66
Fuente: Propia.....		66
3.3.5.1	Diseño interface módulo IVL	67
3.3.6	Historia de usuario Inspecciones estándar	69
3.3.6.1	Diseño interfaz módulo de inspecciones de área	70
3.3.7	Historia de usuario plan vivencial.....	71
3.3.7.1	Diseño interfaz módulo plan vivencial.....	72
3.3.8	Historia de usuario inspecciones estándar.....	73
3.3.9	Diseño de interfaces de módulo de inspecciones estándar.....	74
3.4	Pruebas de aceptación	77
3.4.1	Prueba de aceptación login.....	77
3.4.2	Prueba de aceptación home	78
3.4.3	Prueba de aceptación módulo incidentes	78
3.4.4	Prueba de aceptación módulo IVL	79
3.4.5	Pruebas de aceptación módulo inspecciones de área	80
3.4.6	Prueba de aceptación módulo de plan vivencial	81
3.4.7	Prueba de aceptación módulo de inspecciones estándar	82
3.5	Fase de Estabilización.....	83
3.5.1	Historial de commits locales.....	84

3.5.2	Push repositorio remoto.	85
3.5.3	Repositorio remoto.....	86
3.6	Fase de correcciones y pruebas.....	86
3.6.1	Compilación de la aplicación	86
3.6.1.1	General APK firmado	86
3.6.2	Publicarlo aplicación en Play Store.....	87
3.6.3	Inicio de pruebas Alfa.	87
3.6.3.1	Crear versión alfa.....	88
3.6.3.2	Resultados prueba alfa.	89
3.6.4	Inicio de pruebas beta.....	90
3.6.4.1	Resultados de pruebas Beta	91
3.6.5	Pruebas con Firebase	92
3.6.5.1	Resultados de pruebas con Firebase	93
3.7	Lanzamiento a producción.....	94
3.8	Fin de pruebas y documentación.....	95
CAPITULO IV		96
4	Impactos, conclusiones y recomendaciones	96
4.1	Impacto operativo	97
4.1.1	Análisis.....	97
4.2	Impacto administrativo	98
4.2.1	Análisis.....	98
4.3	Impacto ecológico.....	99

4.3.1	Análisis.....	100
4.4	Conclusiones y recomendaciones	101
4.5	Conclusiones.....	101
4.6	Recomendaciones	102
5	Bibliografía.....	103
	ANEXOS.....	108

ÍNDICE DE FIGURAS

Fig. 1. Módulos para desarrollarse	5
Fig. 2. Los 10 elementos del sistema de gestión de salud y seguridad	9
Fig. 3. Actos y Condiciones Inseguras a abril del 2017	10
Fig. 4. Valores en números de los incidentes por Estándar a abril del 2017	11
Fig. 5. Capas de la Arquitectura de Android	13
Fig. 6. Arquitectura de iOS	14
Fig. 7. Interfaz y estructura de un proyecto Android	16
Fig. 8. Gradle	16
Fig. 9. Consumir Servicio Rest sin usar SynAdaper	18
Fig. 10. Consumir Servicios Rest con SyncAdapter	18
Fig. 11. Content Provider	19
Fig. 12. La arquitectura cliente/servidor con un RDBMS	22
Fig. 13. Arquitectura con SQLite	23
Fig. 14. Ejemplo de mapeo de una tabla a objetos en java	24
Fig. 15. Arquitectura de GreenDao	25
Fig. 16. Estructura de Proyecto de GreenDao	25
Fig. 17. Clases principales del Core de GreenDAO	26
Fig. 18. Patrón Modelo Vista y Controlador	29
Fig. 19. Patrón Modelo Vista Presentador	29
Fig. 20. Fases y Etapas de la Metodología Mobile-D	33
Fig. 21. Caso de uso login	40
Fig. 22. Caso de uso home	41
Fig. 23. Caso de uso registro, consulta y sincronización de incidentes	42
Fig. 24. Caso de uso registro, consulta y sincronización de IVL	43

Fig. 25. Caso de uso registro, consulta y sincronización de Planes Vivenciales	45
Fig. 26. Caso de uso inspecciones de área, agregar nuevo, subir registros.	46
Fig. 27. Caso de usos inspecciones estándar, agregar nuevo, subir registros.	47
Fig. 28. Diagrama de componentes	49
Fig. 29. Diseño de la base de datos	54
Fig. 30. Inicialización de Android Studio	59
Fig. 31. Inicialización de Jboss	60
Fig. 32. Diseño Login	61
Fig. 33. Diseño home sincronizar	63
Fig. 34. Diseño home sincronizado	63
Fig. 35. Menú home	63
Fig. 37. Diseño agregar nuevo incidente.	65
Fig. 38. Diseño agregar fotos incidentes	65
Fig. 39. Diseño reporte incidentes	65
Fig. 40. Nuevo plan vivencial	73
Fig. 41. Planes vivenciales registrados	73
Fig. 42. Menú inspecciones estándar.	74
Fig. 43. Tipos de inspecciones estándar.	75
Fig. 44. Nueva inspección estándar-datos obligatorios	75
Fig. 45. Nueva inspección estándar- datos específicos.	76
Fig. 46. Nueva inspección estándar-agregar fotos.	76
Fig. 47. Commit de módulo de inspecciones estándar.	84
Fig. 48. Push Proyecto al branch development	85
Fig. 49. Repositorio remoto-bitbucket	86
Fig. 50. Generar APK	86

Fig. 51. Generar aplicación firmada	87
Fig. 52. Seleccionar destino de la aplicación firmada	87
Fig. 53. Versiones de una aplicación.	88
Fig. 54. Creación de versión alfa	88
Fig. 55. Agregar usuarios a pruebas Alfa	89
Fig. 56. Panel de control de versiones beta.	91
Fig. 57. TestLab firebase	93
Fig. 58. Pruebas con errores Firebase	93
Fig. 59. Pruebas exitosas Firebase	94
Fig. 60. Lanzar aplicación a producción.	94
Fig. 61. Aplicación publicada en Play Store	95

ÍNDICE DE TABLAS

Tabla 1. Checklist Android vs iOS	14
Tabla 2. Ventajas y desventajas de metodologías ágiles	30
Tabla 3. Checklist metodologías.	31
TABLA 4. Descripción del caso de uso login	40
TABLA 5. Descripción del caso de uso Home	41
TABLA 6. Descripción del caso de uso módulo de incidentes	42
TABLA 7. Descripción del caso de uso módulo de IVL	44
TABLA 8. Descripción del caso de uso módulo de plan vivenciales	45
TABLA 9. Descripción de caso de uso de inspecciones de área	46
TABLA 10. Descripción de caso de uso inspecciones estándar.	48
TABLA 11. Requerimientos funcionales	49
TABLA 12. Descripción de módulos	50
TABLA 13. Tecnologías seleccionadas para la implementación del proyecto	52
TABLA 14. Planificación de iteraciones.	55
TABLA 15. Artefacto de la fase Exploración del Proyecto	58
TABLA 16. Historia de usuario iniciar sesión	60
TABLA 17. Historia de usuario home	61
Tabla 18. Diseño interfaz home	63
Tabla 19. Historia de usuario menú incidentes	64
TABLA 20. Diseño interfaz incidentes	65
TABLA 21. Historia de usuario IVL	66
TABLA 22. Diseño Interfaces módulo de inspecciones de área	67
TABLA 23. Historia de usuarios inspecciones estándar	69
TABLA 24. Diseño del módulo de inspecciones de área	70

TABLA 25. Diseño interfaz módulo plan vivencial	72
TABLA 26. Historia de usuario inspecciones estándar	73
TABLA 27. Diseño interfaz módulo inspecciones de área	74
TABLA 28. Prueba de aceptación del login	77
TABLA 29. Hoja de prueba de aceptación home	78
TABLA 30. Prueba de aceptación módulo incidentes	78
TABLA 31. Prueba de aceptación módulo IVL	79
TABLA 32. Prueba de aceptación módulo de inspecciones de área	80
TABLA 33. Prueba de aceptación módulo de plan vivencial	81
TABLA 34. Prueba de aceptación módulo inspecciones estándar	82
TABLA 35. Commits locales de la aplicación	84
TABLA 36. Resultado pruebas alfa	89
TABLA 37. Resultado pruebas beta	91
TABLA 38. Checklist finalización de desarrollo	95
Tabla 39. Impacto operativo	97
Tabla 40. Impacto administrativo.	98
Tabla 41. Impacto ecológico	99

RESUMEN

Este proyecto de tesis detalla el desarrollo de una aplicación Android enfocado a temas de seguridad y salud en empresas donde la seguridad industrial tiene gran prioridad. Por ello se desarrolla la solución enfocada en minimizar y mitigar los riesgos dentro del funcionamiento de una empresa.

El documento en el capítulo dos analiza la realidad de la seguridad industrial en empresas cementeras, continuando con la teoría necesaria para el desarrollo del proyecto, se analiza las diferentes opciones de desarrollo de aplicaciones móviles, la ingeniería de software en el desarrollo en aplicaciones móviles, haciendo énfasis en la metodología Mobile-D.

El capítulo tres detalla el desenlace del desarrollo de la aplicación móvil propuesta que va de la mano con la metodología seleccionada, donde se hace uso de diagramas UML, historias de usuario y otros artefactos necesarios para completar el proceso de desarrollo.

El capítulo cuatro analiza el impacto de la aplicación móvil haciendo uso de una metodología de análisis prospectivo y se finaliza con conclusiones y recomendaciones.

ABSTRACT

This project of thesis details the development of an Android application focus in security and health topics on factories where industrial security has greater priority.

For this reason, the solution is developed focused in minimize and mitigate risks inside factory operation.

In the document in chapter two analyze the industrial security context in cement companies, following with the necessary theory to develop this project. It's analyzed the different options to develop mobile applications, the software engineering in mobile application development, making focus in Mobile-D methodology.

The third chapter details the outcome of the development of the project proposed that goes in hand with the methodology selected, where is used UML diagrams, user histories and other necessary artifacts to complete the development process.

The four chapter analyzes impact of the mobile application using a prospective analysis methodology and finalize with conclusions and recommendations.

CAPITULO I

1 Introducción

1.1 Antecedentes

La Unión Andina de Cementos en el año 2104 adquirió el grupo francés Lafarge Cementos de Ecuador, a partir de esa fecha UNACEM ha tomado la posta y ha continuado con la operatividad de la empresa. (Cementos, 2018) Uno de los aspectos más importantes que ha caracterizado a esta compañía es la seguridad y salud de los empleados, es por eso por lo que las operaciones que se llevan a cabo dentro de la compañía son llevadas bajo estrictos estándares y normas de seguridad para evitar riesgos en la ejecución de las actividades, y para eso se dictan cursos de inducción de seguridad donde se da a conocer las normas que un empleado debe cumplir en su desenvolvimiento. La gestión de la seguridad y salud industrial se hace a través de las personas expertas, que realizan su seguimiento a través de una plataforma digital llamada “Vida” en donde se hace la gestión de la información que es generada por los usuarios. Esta plataforma ha tenido un gran impacto al permitir la gestión de la información, porque con ella se previene y evita accidentes laborales.

Actualmente está en auge las aplicaciones móviles en varias plataformas existentes en el mercado, donde la más común y utilizada es Android, en las aplicaciones móviles la característica principal es que no es necesaria una conexión a Internet para funcionar, esta característica permite el desarrollo de aplicaciones para diferentes ámbitos de la industria, entretenimiento, marketing o incluso la seguridad industrial. (Enriquez & Casas, 2014)

1.2 Situación actual

La gestión de la seguridad y salud se la hace a través de un Sistema de Gestión Integral “SGI¹” que habilita las políticas de operaciones de la empresa, además posee un sistema “Vida” el cual administra la información generada por el SGI y genera indicadores reactivos y proactivos para con esto prevenir y evitar accidentes laborales.

El ingreso de la información se la hace a través de la plataforma web Vida, en base a los reportes que los empleados hacen según las observaciones en inspecciones que realizaron en el campo industrial, se realizan las respectivas notificaciones a los gerentes encargados de cada área con la finalidad de que procesen esta información y tomen las decisiones pertinentes. La plataforma gestiona las inspecciones periódicas que realizan los competentes de cada estándar de seguridad que son aplicados a los empleados para verificar su cumplimiento y realizar futuras correcciones en caso de no estar cumpliéndolas.

Una de las debilidades de la plataforma digital de seguridad y salud Vida se debe a que al momento de ingresar la información ya que esta llega de manera desorganizada, ya que los usuarios lo ingresan luego de hacer las inspecciones y observaciones en el campo industrial, esto en muchas ocasiones lleva a que el empleado se olvide de los detalles de las observaciones. Además, en el campo industrial no existe redes inalámbricas para acceder a la plataforma web o también hay sitios en donde no hay cobertura de datos. Por lo cual hay la necesidad de que exista la posibilidad de registrar las observaciones al instante ocurrido mediante un dispositivo móvil el cual trabajará de manera local y posteriormente sincronizar con el servidor (Joyanes Aguilar, 2012).

¹ SGI. Sistema de Gestión Integral

1.3 Prospectiva

Con la implementación de la aplicación móvil permitirá que las personas que trabajan en el campo industrial puedan aportar con sus ideas y observaciones para el mejoramiento de la seguridad y salud del personal. De esta manera se puede prevenir y evitar accidentes (CAVERO DORIA, 2017). Existen organismos externos que auditan el ambiente de trabajo en la planta industrial, a lo cual se prevé tener las mejores calificaciones para dejar a la empresa con una buena imagen en la gestión de la Seguridad y Salud.

La información que se toma en el campo tendrá un formato y será detallada al momento que se hace una inspección y posteriormente se va a sincronizar con el servidor, de esta manera se evita que haya distorsión de información cuando finalice una inspección cualquiera para el cumplimiento de los verificadores.

La aplicación estará diseñada para funcionar sin conexión activa a internet (Luján Castillo, 2015) esto se conoce como el lado fuerte de las aplicaciones móviles, hay lugares en donde los datos celulares o WiFi no están disponibles hecho que no será un impedimento para que un usuario registre un incidente o desviación. La aplicación estará desarrollada de tal forma que se podrán integrar los módulos que sean necesarios para la gestión de la seguridad y salud en la empresa.

1.4 Planteamiento del problema

¿Por qué la información de riesgos industriales no es precisa?

1.5 Objetivos

1.5.1 Objetivo General

Implementar una Aplicación Móvil para apoyar la gestión de la Seguridad y Salud en empresas cementeras usando la metodología Ágil Mobile-D.

1.5.2 Objetivos específicos

- Construir la aplicación con enfoque a escenarios offline² para garantizar el ingreso de datos en cualquier campo industrial.
- Implementar una sincronización bidireccional entre la aplicación móvil y el servidor de Aplicaciones.
- Garantizar la integridad y la seguridad de la información con la implementación de la aplicación móvil.
- Analizar la metodología Ágil Mobile-D³ para el desarrollo de aplicaciones móviles.

1.6 Alcance

Con la implementación de esta aplicación móvil se va a gestionar de manera clara los datos generados con respecto a la seguridad industrial, los empleados pueden registrar sus observaciones y reportarlas al instante, sin la necesidad de que haya una conexión activa de internet, se tomarán datos como estadísticas, observaciones fotos, documentos y además gestionar tal información en la misma aplicación móvil como actualizar y eliminar un registro (Ribas Lequerica, 2016).

La aplicación móvil se va a desarrollar de manera que exista una sincronización bidireccional (Gironés, 2016), esto permitirá que se carguen los datos del Servidor dentro de la aplicación como son el usuario, los departamentos, las áreas, los parámetros de seguridad que están almacenados en el servidor externo; y de la misma manera se va a subir la información al servidor cuando el usuario tenga una conexión activa a Internet.

² Offline. Fuera de línea.

³ Mobile-D. Metodología de desarrollo ágil para aplicaciones móviles.

La aplicación móvil tendrá varios módulos para la seguridad y salud, dos de ellos son:

- Registro de Incidentes/Desviaciones

Mediante este módulo se registrará las desviaciones que son condiciones que se consideran un peligro para la seguridad del empleado, así como también los incidentes que son condiciones en el que el riesgo ya está presente.

- Inspecciones de Área

Este módulo registrará las inspecciones que se realizan en determinada fecha por un grupo de personas calificadas, con la aplicación se tomarán las evidencias pertinentes, detallarlas en el mismo campo y posteriormente sincronizar con el servidor para su gestión.

- Inspecciones Estándar

En este módulo se registran las inspecciones que se realizan cotidianamente para calificar el cumplimiento de los estándares de la empresa por parte de los empleados, existen varias categorías de inspecciones estándar.

- Interacciones visibles de liderazgo

Se trata de acompañamientos que un empleado hace a otros para verificar el cumplimiento de sus objetivos de seguridad.

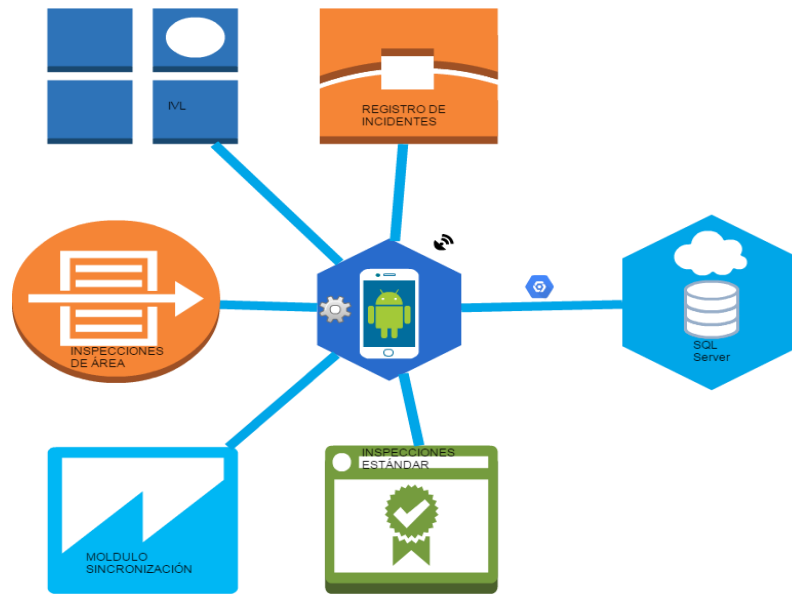


Fig. 1. Módulos para desarrollarse
Fuente: Propia

1.7 Justificación

El Desarrollo de la aplicación móvil permitirá mitigar los incidentes que pongan en riesgo la salud o la vida del personal gracias a la participación proactiva de los usuarios reportando las desviaciones que pueden observar durante el desempeño de sus operaciones dentro del campo industrial, además promoverá una cultura responsable hacia la seguridad y salud de los empleados mejorando sustancialmente la calidad del ambiente de trabajo (Razynskas Sosa, 2015).

La seguridad y salud es el tema principal en una empresa de industrial de alto riesgo donde se debe mantener por mínimos los indicadores de riesgos, haciendo uso de los dispositivos móviles que los empleados poseen, se les dará la facilidad de aportar con sus ideas de mejora a la seguridad industrial de la empresa. Además de aportar positivamente en la gestión de la información para permitir mitigar futuros daños ambientales, biológicos o humanos (González Maestre, 2015).

CAPÍTULO II

2 Marco teórico

2.1 Industrias cementeras en el Ecuador

En el Ecuador, el cemento se constituye como la principal materia prima en el sector de la construcción, una actividad que en los últimos años se ha caracterizado por su crecimiento constante, aún en situaciones económicas adversas; como crisis económicas y financieras. (Hernández-Gaytán, Santos-Burgoa, Becker-Meyer, Macías-Carrillo, & López-Cervantes, 2000)

Las principales compañías cementeras del Ecuador son: UNACEM, HOLCIM siendo estas multinacionales y UCEM que surge de la fusión de dos grandes cementeras: Cemento Chimborazo S.A e Industrias Guapán S.A en los que el 95,72% del capital pertenece al Estado y el 4,28% corresponde a inversionistas privados; por tal motivo se la considera como empresa pública (Izquierdo, A.N.E., 2015).

2.2 Unacem

La Unión Andina de Cementos de Ecuador (UNACEM) es una compañía cementera radicada en la provincia de Imbabura y cantón Otavalo, esta ha venido transformándose durante los años y a través de las diferentes empresas que han estado a cargo de la producción y comercialización del cemento.

El inicio de esta empresa data en el año 1974 como Cemento Selva Alegre S.A. con participación de los sectores público y privado, para la fabricación de cemento, contribuyendo desde su inicio al desarrollo industrial del Ecuador y a la generación de fuentes de trabajo directas e indirectas en la zona norte del país. se produce el primer saco de Cemento Selvalegre, producto que, por cumplir y superar siempre las normas de calidad, fue prontamente acogido por el sector de la construcción y por la población en

general. Desde el año 1994, dentro del programa de modernización del país, la empresa pasa a ser administrada por el sector privado y sigue su camino hacia la consolidación como una de las empresas industriales más eficientes del Ecuador. En diciembre de 2004, la exitosa trayectoria de Cementos Selva Alegre S.A. se fortalece al ser adquirida por el Grupo Lafarge, que le otorga una nueva visión de excelencia y la integra al grupo de materiales de construcción más grande del mundo (Del & General).

UNACEM Ecuador se conformó en noviembre de 2014, producto de la adquisición de Lafarge Cementos S.A. por UNACEM, empresa con más de 60 años de experiencia y líder en el sector cementero peruano. A través de sus subsidiarias, ha diversificado sus negocios para abastecer de manera integral a los sectores de construcción y energía. Hoy, su estrategia de crecimiento lo ha llevado a tener presencia en cinco países: Perú, Ecuador, Chile, Colombia y Estados Unidos (“<http://www.unacem.com.ec/>,” n.d.).

En el año 2015, UNACEM Ecuador recibe la máxima certificación como Empresa Ecoeficiente de parte del Ministerio del Ambiente, gracias a sus proyectos de aprovechamiento de residuos para la producción de clínker⁴ y optimización del consumo de energía. La empresa se convierte en la primera cementera a nivel nacional en recibir la Licencia para el Coprocesamiento de Desechos Peligrosos (Unacem Historia).

UNACEM Ecuador tiene como misión ser la empresa preferida en la industria de la construcción y para lograrlo se apalancan de valores tales como la excelencia, ética, compromiso y por último y muy importante la seguridad y salud. Para este último punto se tiene implementado un Sistema de Gestión Integrado con el que controlan 3 aspectos muy fundamentales que son la calidad de los productos, el aspecto ambiental y la seguridad y salud en el trabajo. Se tiene implementado el ISO 9001, “Sistema de Gestión de la

⁴ Clínker. Resulta de la calcinación en horno de calizas arcillosas preparadas artificialmente a base de otras materias cementosas.

calidad”, por cumplir con los más altos estándares de calidad en producción y comercialización de cemento.

En el Ecuador, a través de su producto Selvalegre, UNACEM Ecuador ha contribuido al desarrollo de la infraestructura del país, durante más de tres décadas. UNACEM Ecuador cuenta con oficinas en Quito y una planta de producción de cemento ubicada en Otavalo. Con el aumento de la capacidad instalada a un total de 1.6 millones de toneladas métricas al año, ampliamos nuestra disponibilidad y cobertura para servir mejor a nuestros clientes en todo el país (Reporte de sostenibilidad 2016, 2016).

2.3 Seguridad y Salud Industrial

Los hechos que ponen en riesgo la vida o la salud del hombre han existido desde siempre. En consecuencia, también desde siempre, el hombre ha tenido la necesidad de protegerse. Pero cuando estos hechos o condiciones de riesgo se circunscriben al trabajo, históricamente, el tema de la producción ha recibido mayor importancia que el de la seguridad, ya que es sólo recientemente que el hombre, como persona natural y como persona jurídica, ha tomado conciencia de la importancia que reviste la salud ocupacional y la seguridad en el trabajo (Lizando & Gallegos, 2012).

En UNACEM se ha implementado un sistema de seguridad de gestión de la seguridad y este ha sido verificado por el Seguro General de Riesgos del Trabajo del Instituto Ecuatoriano de Seguridad Social del Ecuador “IESS”, en conformidad a lo dispuesto por el Sistema Nacional de Gestión de Prevención de Riesgos Laborales. Esto implica que UNACEM Ecuador identifica, evalúa y controla los riesgos asociados a sus actividades y que su gestión integra con eficiencia la mejora continua de la salud y la seguridad en sus instalaciones (“<http://www.unacem.com.ec/>,” n.d.).



Fig. 2. Los 10 elementos del sistema de gestión de salud y seguridad
Fuente: (“Reporte de sostenibilidad 2016,” 2016)

La eliminación de los accidentes dentro de la industria es de vital interés por parte del público, unos accidentes laborales producen pérdidas económico – sociales, la productividad se reduce o se frena y se reduce el aumento del nivel de vida. Como en toda industria se cuenta con procesos, maquinaria y tecnología y estos al interrelacionarse con los trabajadores, las oraciones se vuelven riesgosas donde el individuo puede contraer enfermedades, lesiones en incluso la pérdida de la vida. (“Istas Salud Laboral,” n.d.)

2.3.1 Condiciones y actos inseguros

En la UNACEM se maneja un concepto de seguridad industrial donde el empleado tiene la prioridad sobre las operaciones. Para reducir mitigar los peligros en el campo industrial la empresa tiene una cultura de reportar cada incidente que se observe o de la que un individuo fue parte, este incidente puede ser por un acto inseguro que es por parte

de malas prácticas del empleado o de una condición insegura que tiene que ver con el ambiente de trabajo que no es apto para las labores diarias.

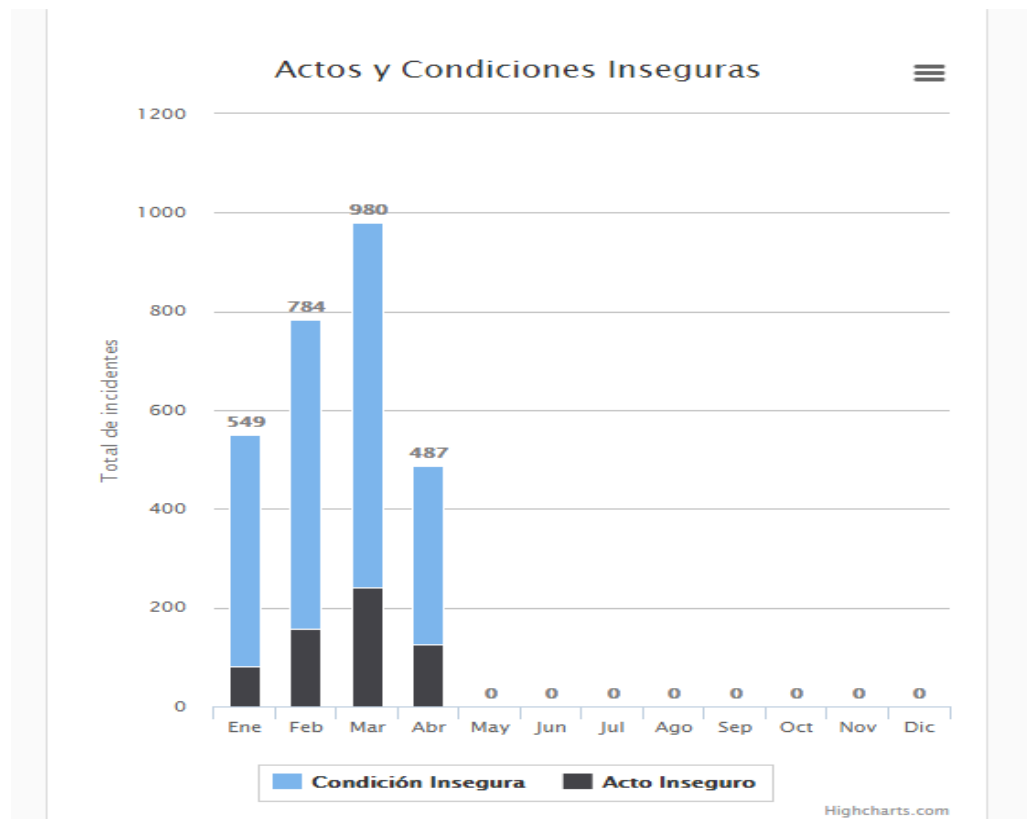


Fig. 3. Actos y Condiciones Inseguras a abril del 2017
Fuente. Plataforma Vida de UNACEM

2.3.2 ATS (Análisis de trabajo seguro)

El departamento de seguridad industrial del UNACEM ha implementado un mecanismo para disminuir los riesgos al ejercer las labores en el campo industrial, el Análisis de Trabajo Seguro donde se detalla los pasos a seguir para realizar determinada actividad, este es analizado por los competentes en el área para aprobarla o rechazarla si se considera que los pasos son riesgos para el trabajador. La persona que apruebe el análisis de trabajo seguro es responsable de la ejecución de la actividad, responsabilizándose si ocurre accidentes.

2.3.3 Estándares de seguridad industrial

Se estandariza los peligros existentes dentro del campo industrial, y se analiza cuál de ellos tiene mayor peligro para tomar las medidas correspondientes al trabajar con tal estándar.

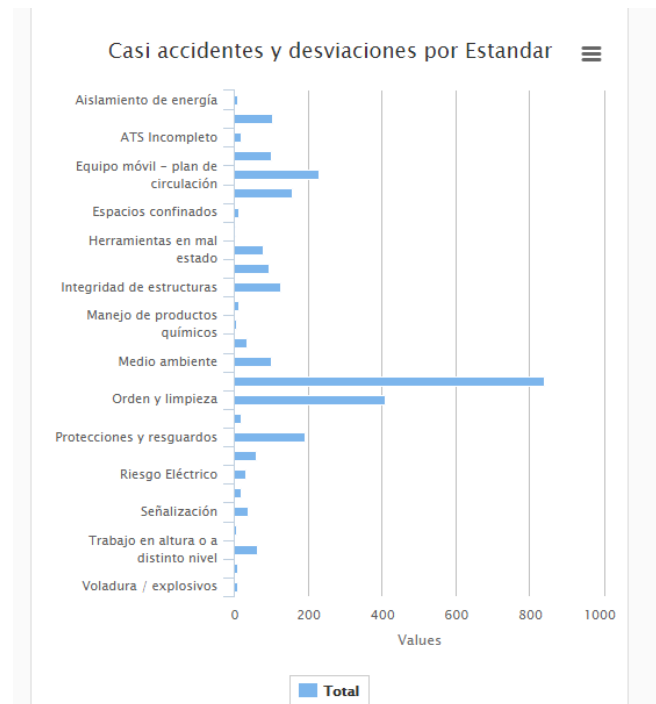


Fig. 4. Valores en números de los incidentes por Estándar a abril del 2017
Fuente: Plataforma Vida Unacem

2.4 El mundo de las aplicaciones móviles

El masivo desarrollo de las aplicaciones móviles comenzó a finales del siglo XX, en donde comúnmente se desarrollaban juegos, calculadoras, calendarios y editores de tonos para las distintas plataformas que existían en aquel entonces. El desarrollo de estas aplicaciones salía de lo convencional que era desarrollar sistemas para computadoras, pero con la demanda de los equipos móviles inteligentes las aplicaciones móviles seguían ganando terreno. (Clarke, 2012).

Los sistemas operativos móviles que operaban en ese entonces eran Symbian que abarcaba la mayoría del mercado, seguido de Android, IOS y RIM, Windows Phone entre

otras. De los cuales actualmente solo han logrado mantenerse en el mercado son Android e IOS, hoy en día, lideran el mercado. (David Stoffel, n.d.).

2.4.1 **Android.**

Android es un sistema operativo que inicialmente fue ideado para dispositivos móviles y está basado en una versión modificada de Linux⁵, nació como un startup que posteriormente fue adquirida por Google y que a partir de ese momento éste lo distribuyó bajo la Licencia de Apache, en otras palabras, es Open Source⁶. Android está diseñado para ejecutarse en un sin número de terminales, los fabricantes vieron opción como una ventaja y remplazaron sus sistemas operativos por Android, para de una, manera ganar terreno contra los iPhone que han tenido mucho éxito con sus productos (Lee, 2012).

Ver figura 5.

⁵ Linux. Sistema operativo de software libre.

⁶ Open Source. Licencia de Código abierto.

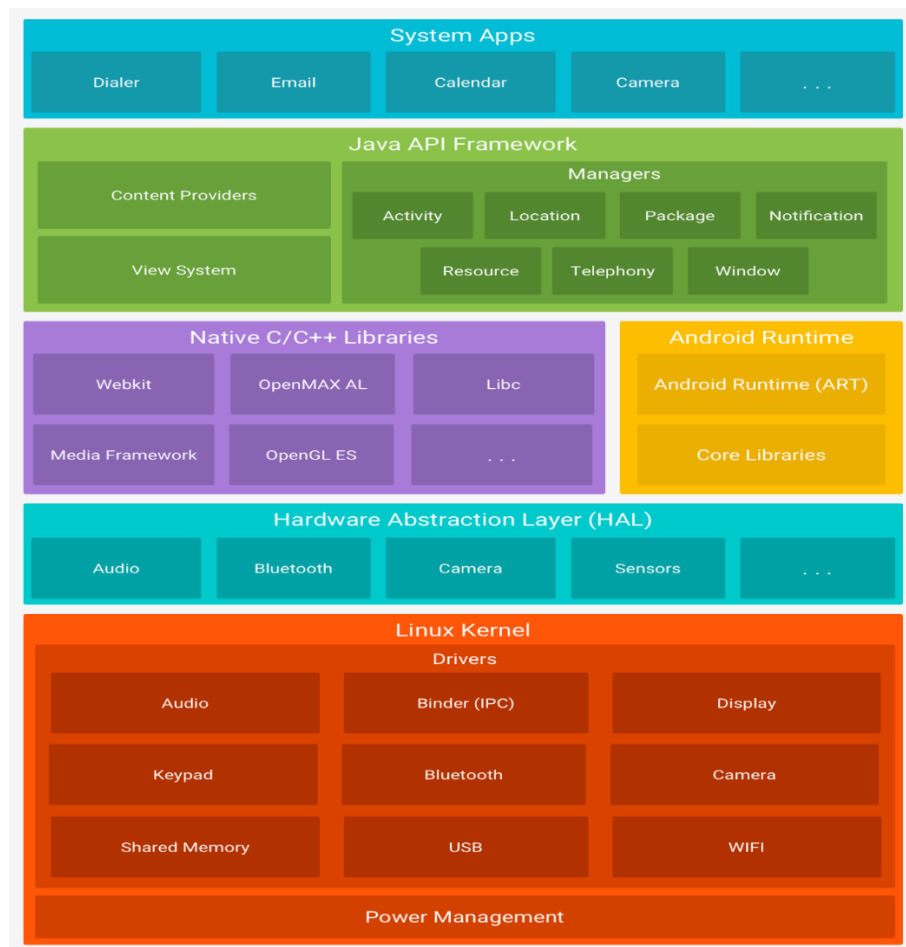


Fig. 5. Capas de la Arquitectura de Android
Fuente: (Báez, Borrego, Cordero, & Cruz, 2012)

2.4.2 IOS

IOS es un sistema operativo para dispositivos móviles de la multinacional Appel, el cual, mostró al mundo en el 2007 cuando Steve Jobs lanzó el iPhone. El lanzamiento del iPhone marca un importante acontecimiento en el mercado de los smartphones ya que vino con innovaciones bastantes atractivas como la pantalla táctil. (“A Brief History of iOS,” n.d.)

Su arquitectura se basa en el sistema operativo de Mac simplificado y optimizado para el iPhone y se compone de cuatro capas.

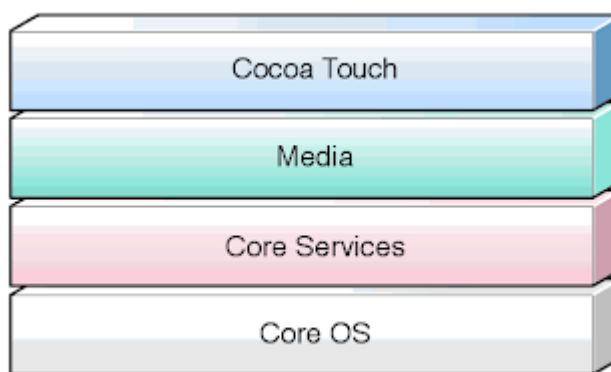


Fig. 6. Arquitectura de iOS
Fuente: <http://bit.ly/2Nljuar>

2.4.3 ¿Android o iOS?

Decidirse por uno de estos sistemas operativos depende de los gustos del usuario, cada plataforma ofrece una particularidad muy importante, que a continuación se describe ventajas y desventajas de cada uno.

Tabla 1. Checklist Android vs iOS

Ítem	Android	iOS
1. Código Abierto.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2. Aplicaciones gratuitas.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3. Aplicaciones de paga.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4. Personalización del sistema operativo.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5. Se ejecuta en diversas marcas de dispositivos.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6. Seguridad	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7. Accesible a todos los usuarios.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8. Compartir archivos con dispositivos diferentes	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Fuente: Propia

Se puede concluir que ninguno es mejor que otro, Android o iOS, puesto que tienen sus puntos fuertes y sus puntos débiles, el usuario final es quien decide con qué sistema operativo se siente más cómodo. (Aguilar, 2017).

2.4.4 Android SDK

Es el elemento más importante de software de Android puesto que esta contiene librerías, depurador, un emulador y documentación del código. El SDK se ejecuta juntamente con el JDK del CPU que debe estar instalado con anterioridad para su funcionamiento.

2.4.5 Android Studio

Android Studio es el IDE oficial para el desarrollo de aplicaciones nativas Android, se basa en IntelliJ IDEA, que es otro, IDE de Java. Ofrece diversas funcionalidades como:

- Integración de Gradle para la administración de librerías y compilación de la aplicación.
- Emulador integrado, el cual permite desplegar la aplicación e incluye una ejecución instantánea el cual evita que se compile nuevamente el APK.
- Entorno en el cual se hace el desarrollo de todo tipo de aplicaciones para dispositivos Android (Smartphone, Tablet, SmartWatch)
- Posibilidad de Añadir NDK, con el que se puede añadir código en lenguajes nativos como C o C++ (Conoce Android Studio, s.f.).
- Soporte de Kotlin.

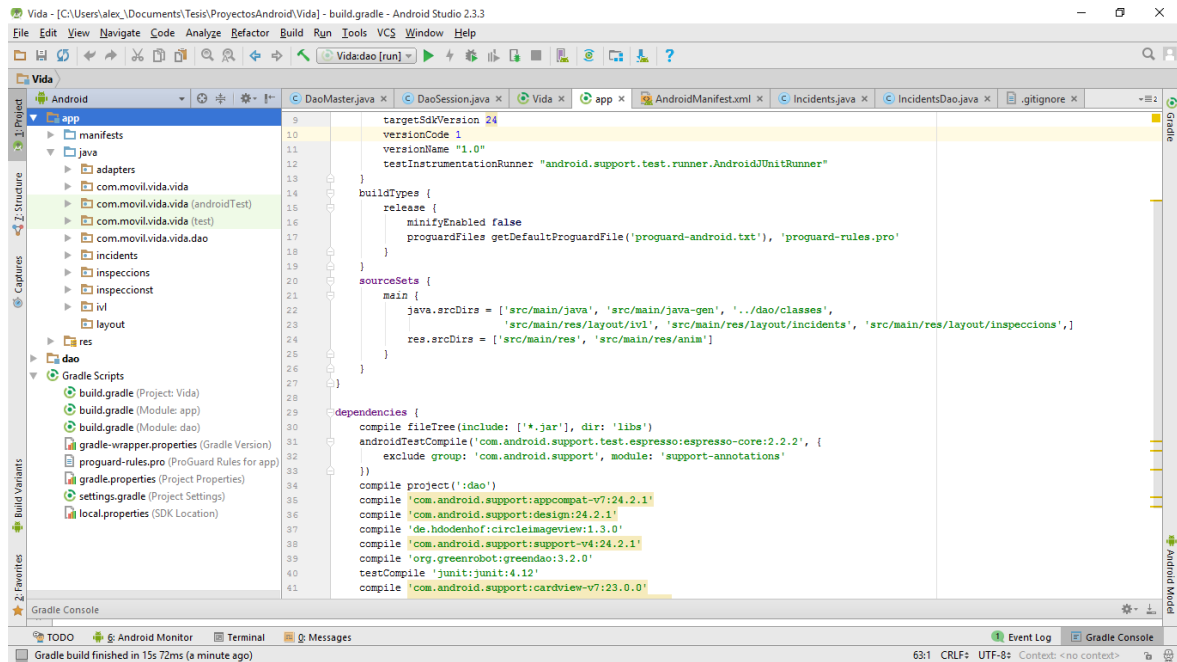


Fig. 7. Interfaz y estructura de un proyecto Android
Fuente: Propia

2.4.6 Gradle

Gradle es una herramienta que se integra al IDE de Android Studio que ayuda en la compilación del código, está basado en JMV (Java Virtual Machine) por lo que es posible crear scripts personalizados para que sea ejecutados junto con el proyecto de la Aplicación. Este plugin facilita la actualización y exportación del proyecto en un solo paquete (Android Application Package) (Android Studio Faqs, 2016).

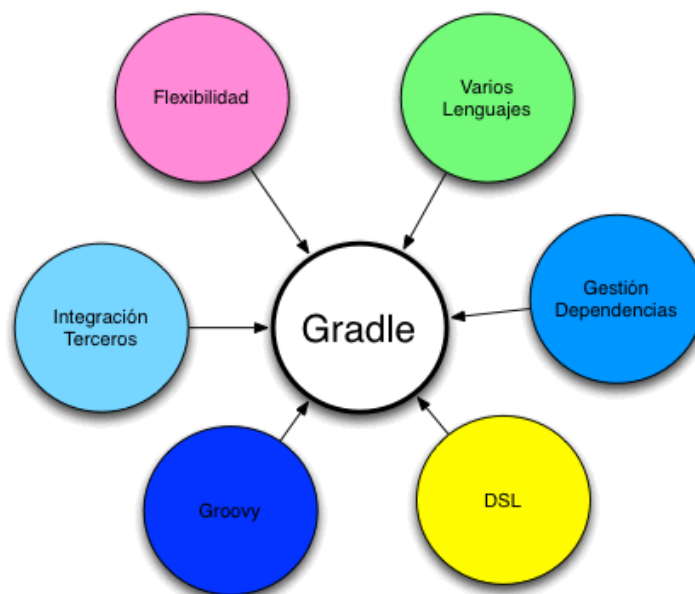


Fig. 8. Gradle
Fuente: (Álvarez Caules, 2016)

2.4.7 Sincronización de datos

La sincronización de datos es una técnica por la cual se extrae datos de un host o a su vez se envía datos hacia él, sirve para replicar datos, hacer copias de seguridad o corregir errores (Dar & Iqra, 2016). La sincronización de datos aumenta la usabilidad de la aplicación móvil, tal como preservar los datos del usuario en servidores remotos los cuales pueden ser recuperados en caso de perder el dispositivo.

- Sincronización Local

Esta sincronización es la que se hace el dispositivo descargando el contenido del servidor remoto y se lo almacena en el dispositivo. Esta técnica permite trabajar con la aplicación sin conexión a una red.

- Sincronización Remota

Esta sincronización es la que se hace con el servidor remoto, la aplicación envía datos y el servidor los recibe y los une con los datos almacenados. Hay herramientas que permiten mantener sincronizados estos datos ya que muchas veces hay problemas de información duplicada.

La sincronización con los dispositivos móviles puede llegar a ser tediosa si no se lo implementa con una arquitectura adecuada y para la práctica existen varios métodos de sincronización. Lo más habitual es a través de web services tipo REST.

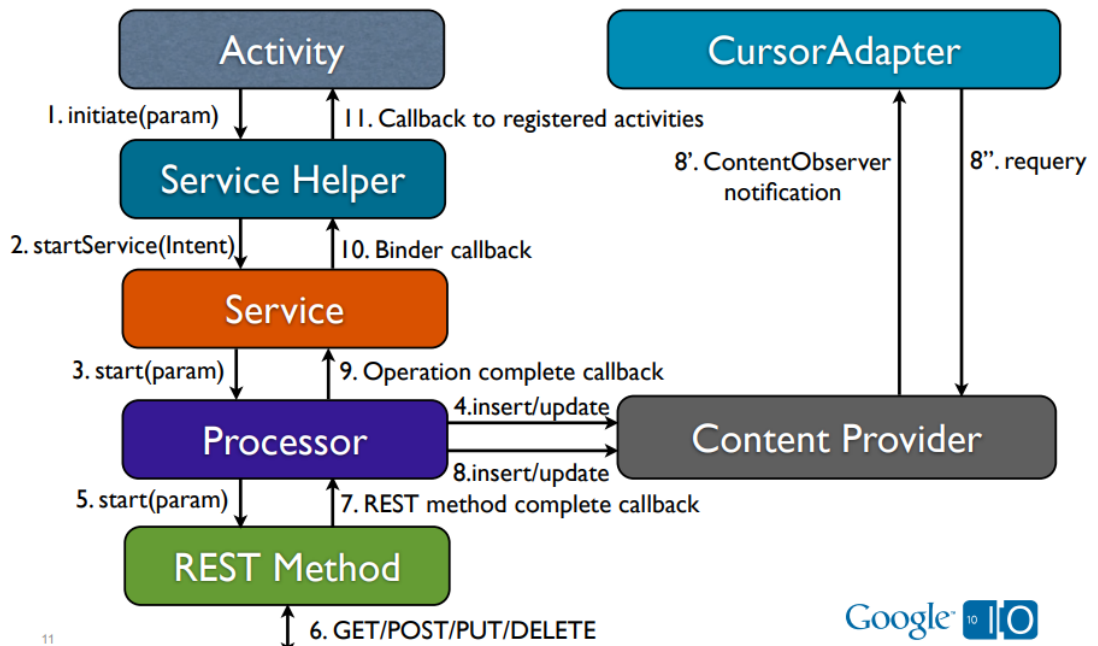


Fig. 9. Consumir Servicio Rest sin usar SynAdaper
Fuente: (Dobjanschi, 2010)

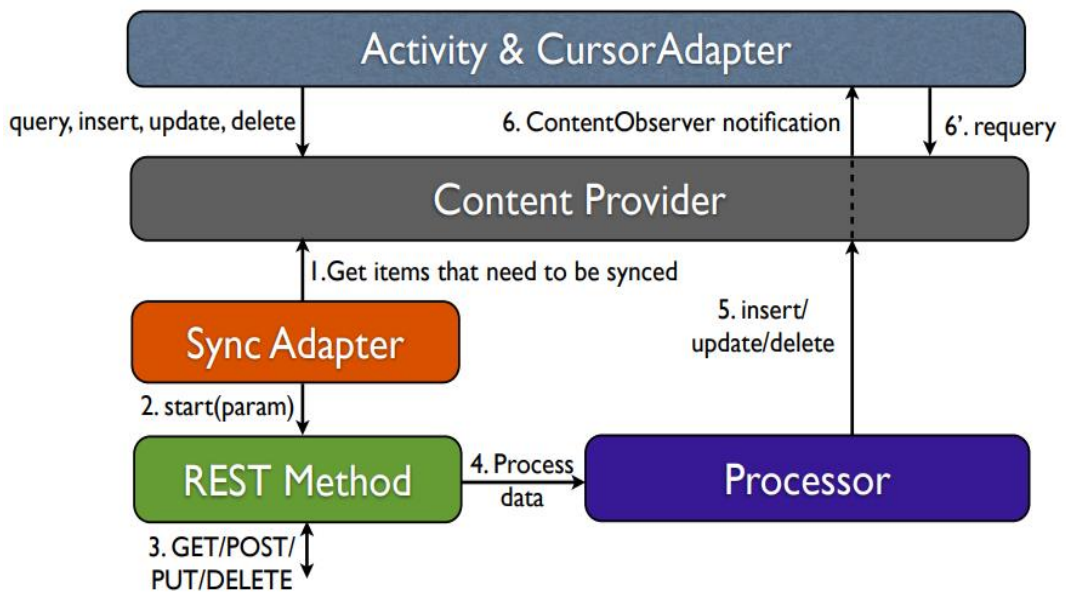


Fig. 10. Consumir Servicios Rest con SyncAdapter
Fuente: (Dobjanschi, 2010)

Este método de sincronización usa un Sync Adapter el cual detecta los cambios en el modelo de datos y hace sus respectivas llamadas a los métodos Rest para realizar la sincronización. De esta manera se evita el excesivo uso de ancho de banda o datos de los dispositivos móviles enviando y descargando toda la información.

2.4.8 Sync adapter

Sync Adapter es un componente de android que permite la sincronización local y remota entre el dispositivo móvil y un servidor remoto. Este componente se ejecutará según la configuración del usuario y necesita de otros componentes para el envío de datos (Developer Android, 2016).

2.4.8.1 Token de autenticación

Para el uso de sync adapter es necesario crear un token que permita la autenticación entre el dispositivo y el servidor remoto, esto crea una cuenta el dispositivo y es administrado por las cuentas de Google. Esto permite a los usuarios registrarse una sola vez y las posteriores ingresar sin la petición de Login tal como lo hacen las aplicaciones de Facebook, Whatsapp, Instagram, etcétera.

2.4.8.2 Content providers

Un Content Provider es una clase que permite que otras aplicaciones o la misma aplicación acceda de manera segura a los datos almacenados en la base de datos embebida SQLite. Cuando una aplicación tiene como objetivo compartir sus datos esa debe incluir un Content Provider para que el acceso a ellos sea ordenado.

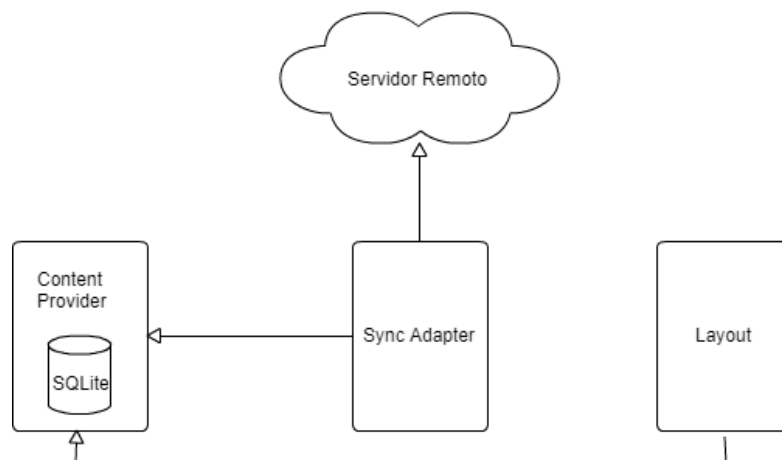


Fig. 11. Content Provider
Fuente: Propia

2.4.9 Modelos de comunicación

Los modelos de conexión en modo remoto permiten la integración de las configuraciones hardware y software, para la obtención de varias vías de acceso a la información (Comparación de sistemas de comunicación remota para transferir información de un registrador de fallas, 2008). En el desarrollo de aplicaciones móviles se presenta varios tipos de modelos de comunicación.

2.4.9.1 Modelo de conexión por intranet

Este modelo de comunicación se caracteriza por el aislamiento de conexiones exteriores a una red que está registrada para un dominio en particular. Es habitual que exista este tipo de comunicación en organizaciones y empresas donde los datos que se manejan en ellos son de carácter privado o en algunos casos sensible hacia el público.

2.4.9.2 Modelo de conexión por internet

En este modelo el acceso exterior es autenticado ya que el nivel de confianza es baja en comparación con el anterior modelo. Los servidores se configuran de manera que no se permita ataques exteriores en contra de la organización ya que al estar expuesta la amenaza de robo o adulteración de la información aumenta.

2.4.10 Persistencia en Android

Existen muchas maneras de guardar información en el SDK de Android tales como:

- Shared Preferences

“La clase SharedPreferences provee de un framework que permite almacenar y recuperar la información. Los datos son almacenados en pares nombre-valor, este último puede ser cualquier tipo de datos primitivo: boolean, floats, int, longs o String. Este tipo de almacenamiento suele utilizarse para el almacenamiento de preferencias de usuario.” (Narváez & David, 2012).

- Archivos internos

Es una manera de almacenar la información en la memoria interna del dispositivo en objetos Files al cual solo se podrá acceder mediante la aplicación quien creó esos archivos, una vez se desinstale la aplicación toda esa información se borra, de esa forma se asegura que solo la aplicación acceda a tal información (Developer Android, 2016).

- Archivos Externos

Para guardar los datos se necesita de medios externos como dispositivos Micro SD o USB, con este tipo de almacenamiento es posible compartir la información generada por las aplicaciones a otros dispositivos (Developer Android, 2016).

- SQLite

SQL lite es un paquete que provee de un sistema de almacenamiento relacional de datos en el cual se puede guardar grandes datos en tablas. Lite no significa que su poder de almacenamiento sea bajo, al contrario, llega a ser poderoso cuando se lo configura de la manera adecuada. Las características que lo definen son:

- a. No necesita un servidor por que las aplicaciones acceden a ella directamente sin la necesidad de tener un servicio a parte en el sistema operativo.
- b. No necesita de una configuración previa, es tan sencillo como hacer una instancia del objeto para empezar a usarlo.

- c. No requiere de administración ya que toda la información se la guarda en un solo archivo
- d. Una sola librería contiene todo el sistema de base de datos y está integrada en el dispositivo.
- e. SQL lite es completamente transaccional de esta manera asegurando la atomicidad, consistencia, aislamiento y durabilidad. Esto permite el acceso seguro desde varias aplicaciones.
- f. Es muy confiable y además soporta varias características del lenguaje de consultas SQL.

En resumen, SQL lite provee de una base de datos relacional flexible y funcional que consumen recursos mínimos y no representa molestias a los desarrolladores (Kreibich, 2010).

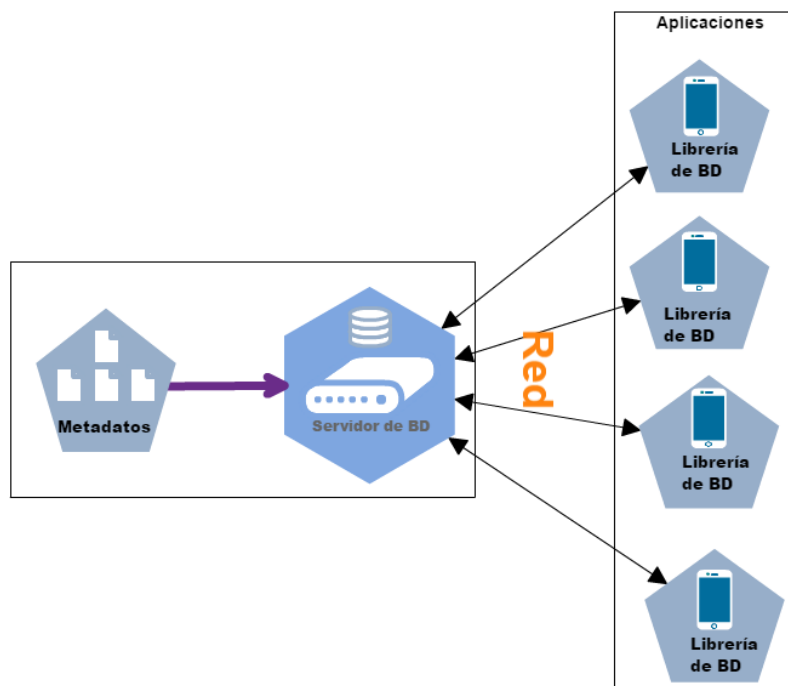


Fig. 12. La arquitectura cliente/servidor con un RDBMS
Fuente: (Kreibich, 2010)

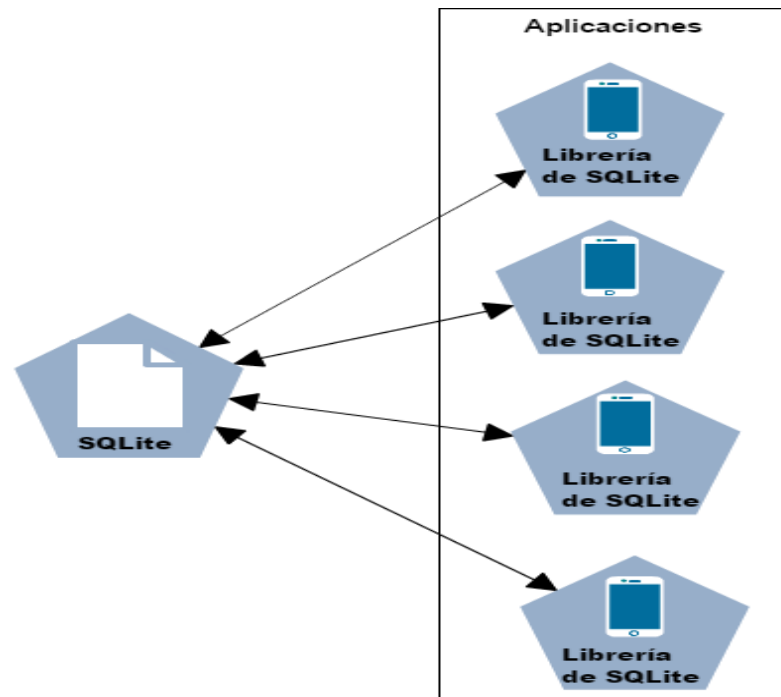


Fig. 13. Arquitectura con SQLite
Fuente: (Kreibich, 2010)

- Almacenamiento en la nube

Muchas aplicaciones que se usa diariamente adoptan este tipo de almacenamiento como es el caso de Facebook o WhatsApp en donde la información del usuario viaja a través del internet para almacenarse en proveedores de almacenamiento externo o como se lo conoce Cloud Storage. Esta técnica permite al usuario la disponibilidad de su información en cualquier momento mientras haya conexión hacia el sitio de almacenamiento con tan solo realizar una sincronización.

2.4.11 ORM

ORM (Object Relational Mapping) es una técnica para transformar datos entre sistemas incompatibles en lenguajes de programación orientado a objetos. Este es una capa entre la base de datos y la aplicación y se encarga de mapear objetos entre la aplicación y las propiedades como las columnas, claves primarias, claves foráneas, atributos de la base de datos, de esa manera se evita el mapeo manual de las consultas en la Aplicación (Juneau, 2013).

Un ORM reduce el tiempo y trabajo requerido para construir sentencias SQL de bajo nivel, un ORM provee de independencia entre el esquema de base datos que estará definida en la aplicación y el proveedor de base de datos por el hecho de que se puede definir el modelo lógico en la capa de persistencia de la aplicación. Existen varios frameworks que utilizan JPA o Hibernate para Java EE, y en dispositivos móviles ORMLite, Sugar ORM, GreenDAO, etcétera.

```
1 package com.capsule.safedrive.model.base;
2
3 import java.util.Set;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 @Entity
21 @Table(name="user")
22 public class User {
23
24     @Id
25     @GeneratedValue(strategy=GenerationType.IDENTITY)
26     private Long id;
27
28     @Column(name="login", length=12, nullable=false)
29     private String login;
30
31     @Column(name="name", length=30, nullable=false)
32     private String name;
33
34     @Column(name="password", length=100, nullable=false)
35     private String password;
36
37     @Column(name="email", length=80, nullable=false)
38     private String email;
39
40
41     @ManyToOne(fetch=FetchType.LAZY)
42     @JoinColumn(name="enterprise_id")
43     private Enterprise enterprise;
44
45     @ManyToMany(fetch=FetchType.EAGER)
46     @JoinTable(name = "user_role",
47         joinColumns = @JoinColumn(name = "user_id"),
48         inverseJoinColumns = @JoinColumn(name = "role_id"))
49 }
```

Fig. 14. Ejemplo de mapeo de una tabla a objetos en java
Fuente: Propia

2.4.12 GreenDAO

GreenDao es un API de persistencia para el desarrollo en Android con la base de datos SQLite que permite realizar operaciones DDL y DML mediante objetos con lo cual se evita la escritura de sentencias SQL y hace que el desarrollo de sea ágil y la experiencia con SQLite sea amigable. Este ORM además aumenta el performance de la aplicación por que permite que SQLite se ejecute en un hilo secundario y no interfiera con el hilo

principal que controla la aplicación, de esta manera el usuario no se verá afectado con las lecturas y escrituras hacia la base de datos mientras navega en la interface.

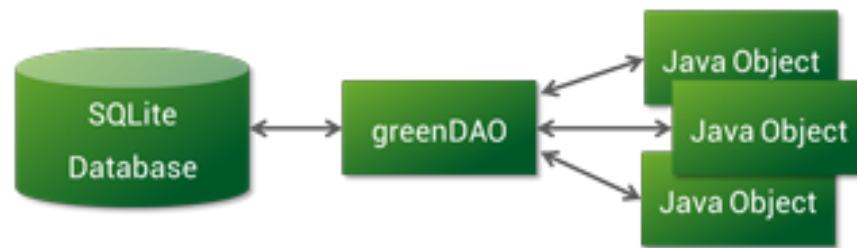


Fig. 15. Arquitectura de GreenDao
Fuente: (GreenRobot, 2016)

Características de GreenDAO

- **Fácil de Usar.** Su API cubre todas operaciones DML incluyendo las juntas, su rendimiento aumenta considerablemente.
- **Minimalista.** Su librería pesa menos que 100KB, acompañado de bajo consumo de memoria.
- **Open Source.** Es código libre, sin licenciamiento. Muchas de las aplicaciones en la Google Play integran este ORM (Juneau, 2013).

2.4.12.1 Estructura de GreenDao

La definición del esquema de la base de datos se lo hace en un proyecto Java (no Android) el cual al ser ejecutado genera las entidades que serán usadas en el proyecto Android.

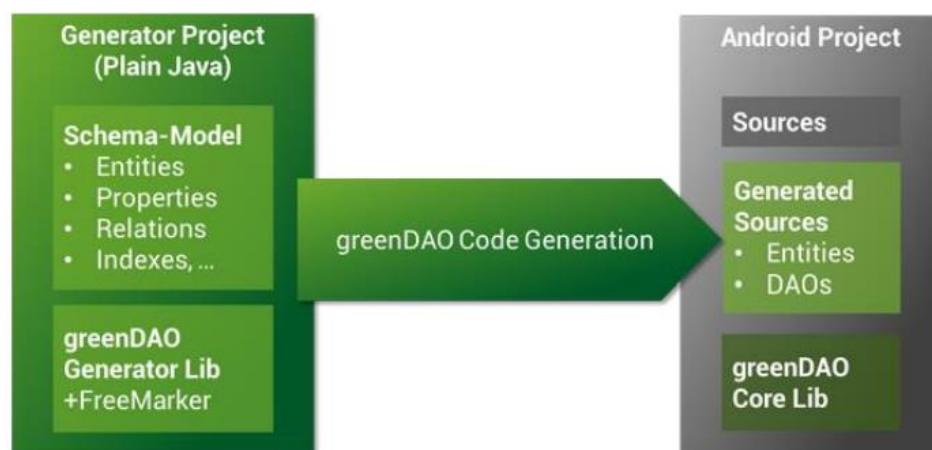


Fig. 16. Estructura de Proyecto de GreenDao
Fuente: (Juneau, 2013)

2.4.12.2 El Core de GreenDao

Está confirmada por 4 clases principales, las cuales son:

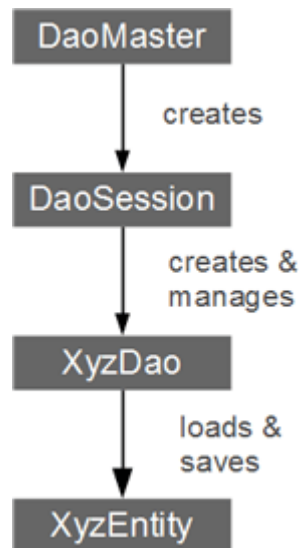


Fig. 17. Clases principales del Core de GreenDAO
Fuente: Propia

- **DaoMaster.** Esta clase hereda las propiedades de SQLiteOpenHelper el cual administra la creación y borrado de los esquemas en la base de datos SQLite.
- **DaoSession.** Se genera a partir de DaoMaster y tiene como finalidad administrar el acceso a los Data Access Object (DAO) para evitar sobrecargas en memoria caché. Además provee los query's genéricos como insert, update, read y delete.
- **DAOs.** Esta clase implementa la persistencia de consultas y entidades, a parte de los métodos genéricos posee métodos como findBy, count, sum, loadAll, etcétera.
- **Entity.** Son clases que representan una tabla y columnas de base de datos los cuales van a ser usado por los DAO para llevar a cabo la persistencia.

En el desarrollo se especificará los pasos para generar estas clases.

2.4.13 Material Design

Google en el año 2014 en una de sus conferencias anunció un nuevo patrón de diseño basado en los colores y formas de los materiales de la naturaleza de ahí su nombre “Material”. Desde tal fecha se ha buscado de que todas las aplicaciones adapten este diseño

con la ideología de que será intuitivo para el usuario por la combinación de colores y formas. También se busca homogeneizar el diseño de las aplicaciones para que su performance no se vea afectado en las diferentes versiones de Android y teléfonos existentes en el mercado.

2.5 Patrones de Diseño y Arquitectónicos en Android

Cuando los desarrolladores construyen aplicaciones con un buen desempeño y que sea mantenible, la solución es implementar patrones de diseño y arquitectónicos en la construcción de la aplicación.

2.5.1 Patrones de diseño

Los patrones de diseño ofrecen elegantes soluciones a problemas cotidianos, estos manejan la creación de objetos, composición de objetos en largas estructuras, y el control de flujo entre objetos (Gamma, Helm, Johnson, & Vlissides, 1996).

El nombre que se le da a un patrón de diseño debe describir el diseño que se le da al problema esto ayuda a la comunicación efectiva entre los desarrolladores. A los patrones de diseño se los categorizan en tres categorías.

- **Patrones de Diseño Creacionales**

Describe la creación de objetos

- **Patrones de diseño Estructurales**

Describe la simplificación de la relación objetos/entidad

- **Patrones de Diseño de Comportamiento**

Describe como asignar responsabilidades entre objetos

2.5.1.1 **Patrón de diseño facade**

Es un patrón de diseño estructural, y este engloba sistemas complejos en una sola interface lo cual simplifica el uso por parte del cliente.

2.5.1.2 **Patrón de diseño observer**

Este patrón de diseño es usado por los desarrolladores de interfaces en el front-end, se define en observadores los cuales esperan a leer los cambios en el proveedor de los servicios.

2.5.2 **Patrones arquitectónicos**

En el campo de la ingeniería de software los patrones arquitectónicos proporcionan una solución frente a los problemas presentados en las arquitecturas de software.

Android no define ningún modelo arquitectónico, el desarrollador es quien decide que patrón adaptar al desarrollo de la aplicación.

2.5.2.1 **MVC (Modelo Vista Controlador)**

El Modelo MVC es un patrón de diseño que permite implementar sistemas basados en interfaces, dando como resultado un producto de software más robusto y con un ciclo de vida adecuado, facilitando la reutilización de código y diferenciando claramente la vista o código “html”, del modelo “capa de manejo de datos” y del controlador “lógica del negocio”.

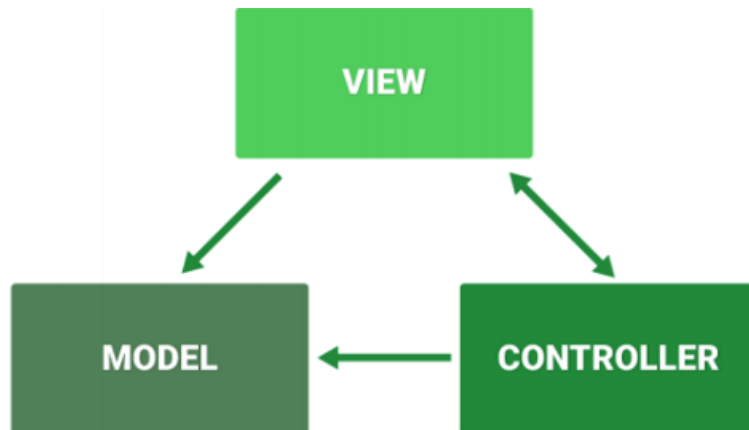


Fig. 18. Patrón Modelo Vista y Controlador
Fuente: (Bui Duy, 2017)

2.5.2.2 MVP (Modelo Vista Presentador)

El modelo MVP puede ser considerado una mejora del MVC, donde la vista no tiene comunicación con el modelo y donde no existe ningún tipo de código de procesamiento, el presentador es independiente y es el encargado de dotarle de inteligencia a la vista y el modelo es la capa de datos que toma el presentador y que luego se envía a la vista para que sea mostrada en la interfaz de usuario (Imaginanet, 2012).

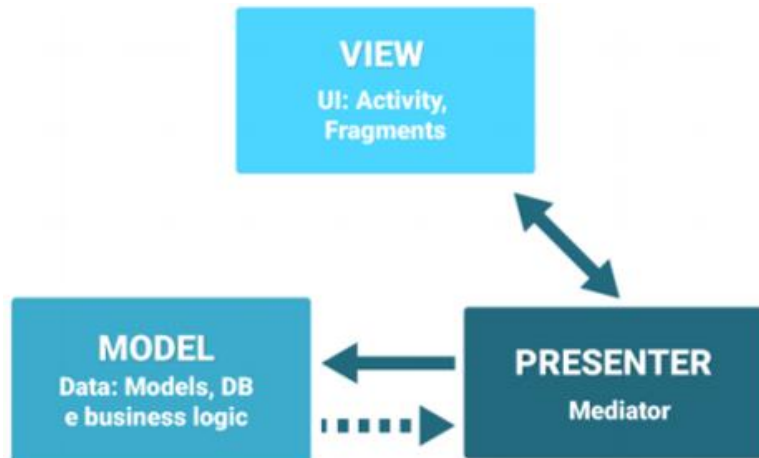


Fig. 19. Patrón Modelo Vista Presentador
Fuente: Propia

2.6 Metodologías

“Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de

sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo” (Amaya Balaguera, 2013).

2.6.1 Metodologías ágiles

Las metodologías ágiles surgen como alternativa a las metodologías tradicionales y se caracterizan por un desarrollo iterativo para adaptarse a los cambios en los requerimientos. Se adaptan fácilmente a proyectos donde el tiempo es corto y el cliente solicita verificar avances periódicamente. En cada una de las iteraciones se procura obtener un producto entregable, con una corrección o un reléase mejorado. Las metodologías exponentes en esta categoría son SCRUM y Extreme Programming(XP).

2.6.2 Metodologías ágiles en el desarrollo de aplicaciones móviles.

El desarrollo de aplicaciones móviles se caracteriza por tener cortos periodos para la entrega de los productos, por lo que se debe tener un contacto directo con el cliente para cumplir con el alcance de la aplicación. Existe varias metodologías que se usa en desarrollo móvil como SCRUM, pero hay una que surgió en el tiempo de auge de las aplicaciones móviles, la cual es llamada Mobile-D.

Tabla 2.Ventajas y desventajas de metodologías ágiles

Metodología	Ventajas	Desventajas
SCRUM	<ul style="list-style-type: none"> -Entregables en cada sprint. -Corrección de errores a tiempo -Buena organización de roles en el equipo. 	<ul style="list-style-type: none"> -No es aplicable en proyectos de gran escala. -El equipo debe poseer alta cualificación para cumplir los plazos.

Rational Unified Process (RUP)	-Metodología para grandes proyectos con una buena organización de responsabilidades.	-Muy pesado en proyectos pequeños.
Extreme Programming(XP)	-Desarrollo orientado a corrección de errores. -Constante retroalimentación.	-Los inconvenientes llegan a ser costosos.
Mobile-D	-De fácil entendimiento. -Documentación liviana. -Comunicación continua con el cliente. -Detección de errores en fases iniciales.	-Los equipos no pueden ser de más de 10 personas.

Fuente: Propia

2.6.3 Justificación de la metodología

Tabla 3. Checklist metodologías.

Ítem	Scrum	XP	RUP	Mobile-D
1. Equipos conformados por menos de 10 personas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2. Documentación liviana.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3. Iteración en la misma etapa	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4. Especificaciones de requerimientos de desarrollo móvil.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5. Desarrollo orientado a pruebas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

6. Integración continua.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7. Alta disciplina en la aplicación de la metodología.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8. Programación en pareja	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
9. Comunicación continua con el cliente.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
10. Entregables en cada sprint(iteración)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11. Detección de errores en fases iniciales	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
12. Gran documentación.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13. Orientado a dispositivos móviles	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
14. Inicio de desarrollo en menos de una semana	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
15. Se puede agregar más funcionalidades en pleno desarrollo.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fuente: Propia

Para el desarrollo de la aplicación se eligió Mobile-D, ya que el alcance da a conocer que no es un desarrollo nuevo si no la ampliación de un software existente, una aplicación móvil. Además, los requerimientos están bien definidos, así como el alcance del proyecto.

2.6.4 Mobile-D

Esta metodología está diseñada para trabajar con no más de 10 personas, por el hecho de que el desarrollo de aplicaciones móviles se la hace en equipos pequeños y el tiempo es corto. **Consta de 5 fases: Exploración, Inicialización, Producción, Estabilización y Pruebas del Sistema (Agile, s.f.).**

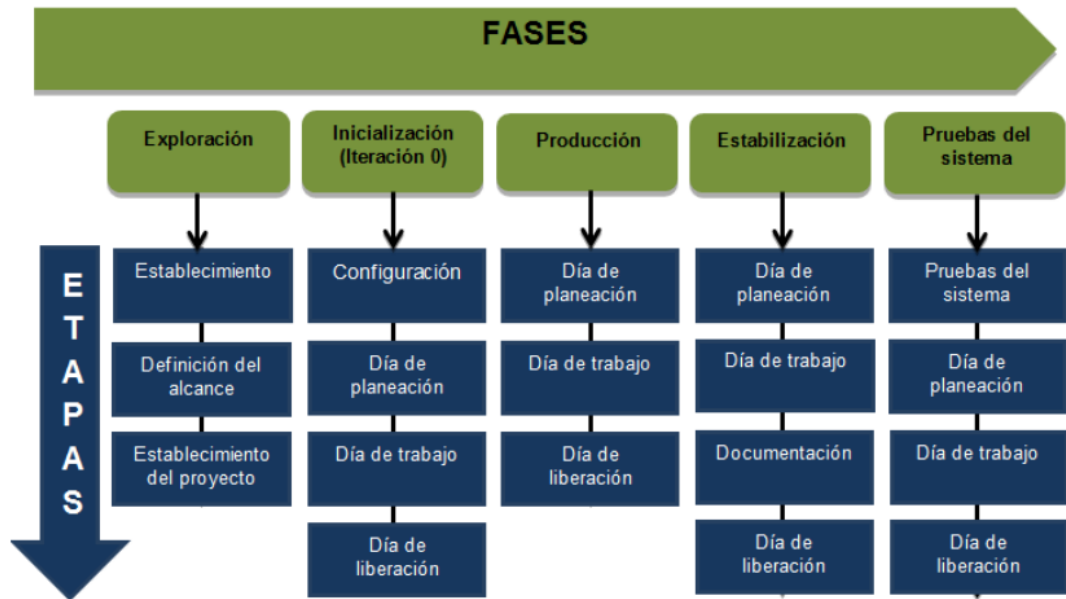


Fig. 20. Fases y Etapas de la Metodología Mobile-D
Fuente: (Escobar Bonilla & Campaña Guzmán, 2014)

2.6.4.1 Exploración

El propósito de esta fase es organizar una reunión entre los desarrolladores y el cliente, para definir el alcance. El equipo conformado discutirá sobre el escenario que van a cubrir, los problemas posibles al seleccionar una tecnología y expresar al cliente la solución que proponen. Aunque se puede obviar esta fase con 0 iteraciones no es aconsejable ya que es aquí donde se definen los roles de participación por parte de cada uno de los actores.

✓ Establecimiento

En esta etapa de la exploración se definen las personas o grupos que van a revisar continuamente el avance del proyecto con propósito de que al culminarlo se realice una implementación satisfactoria.

✓ **Definición del Alcance**

Fase en la cual se establecen objetivos a cumplir y además el tiempo que tomará para cumplirlos. Se realiza la primera colección de los requerimientos y una primera planificación.

✓ **Establecimiento del Proyecto**

Se establecen los escenarios para el equipo de desarrollo y las herramientas o la arquitectura que usarán para el desarrollo para evitar confusiones durante el proceso. En esta etapa se identifican las fallencias de los desarrolladores y se busca una solución o caso contrario buscar personas que se acoplen a los requerimientos.

2.6.4.2 **Inicialización**

El propósito de esta fase es asegurar el éxito de las siguientes fases de la metodología para ello se prepara y verifica puntos críticos que podrían ser un obstáculo para cumplir con todos los requerimientos del cliente.

✓ **Inicialización**

En esta etapa se hacen las configuraciones físicas y tecnológicas así mismo un ambiente donde se visualice su avance. Es aquí donde se establece la manera de comunicación con el cliente y se escuchan las sugerencias por su parte en caso de no estar de acuerdo con algún término.

✓ **Día de Planeación**

En esta etapa se verifica el buen entendimiento de las fases del proyecto por parte del equipo de desarrollo y preparar planes para solventar errores críticos.

✓ **Día de trabajo**

Se configuran todos los requerimientos técnicos para asegurarse de que todo esté listo para la implementación del desarrollo. También el propósito es iniciar con el desarrollo del código o solucionar algún problema relacionado para resolverlo antes de iniciar totalmente con el desarrollo. Si en este punto el cliente decide implementar nuevos requerimientos no debe ser de gran prioridad a menos que este cause el cambio de la arquitectura ya definida para el desarrollo.

✓ **Día del Release**

Se libera el primer reléase del aplicativo móvil el cual es entregable con las primeras funcionalidades que el cliente haya solicitado.

2.6.4.3 **Producción**

En esta fase se implementa las funcionalidades requeridas en el producto aplicando las iteraciones necesarias.

✓ **Día de planeación**

En esta etapa se hace un análisis sobre el proceso de desarrollo de software está acoplándose al equipo de trabajo, además se hace la priorización de los requerimientos y la calendarización de las siguientes iteraciones en el orden que se los defina. Se hace tareas de verificación de los requerimientos que el cliente solicitó y también se coordina la comunicación con el equipo de desarrollo.

✓ **Día de trabajo**

En esta etapa se libera un reléase completamente funcional sobre el requerimiento que se ha estado trabajando, y una de las tareas que se desencadena es las pruebas piloto del release, si se detecta alguna falla se hace cambios y se verifica que no represente daño a los demás componentes de la aplicación. El equipo de desarrollo estará en continuo aprendizaje en los métodos de resolución de problemas. Al final se va a evaluar honestamente el avance que se ha hecho para presentarlo al cliente.

✓ **Día del Release**

Si hay componentes que han estado desarrollándose en paralelo en esta etapa se los integra en un solo producto, hecho esto se hace un testing de la integración, lo siguiente es realizar pruebas de funcionalidad con todos los componentes integrados hasta que se alcance un nivel de confianza aceptable.

2.6.4.4 **Estabilización**

Las aplicaciones complejas pueden haber estado desarrollándose en partes en pequeños subsistemas, y el propósito de esta fase es unir todo en un producto único

✓ **Día de planeación**

En esta etapa se planifica el escenario correspondiente para las iteraciones tomando en cuenta que si los requerimientos del cliente han sido solventados. Una de las tareas es mejorar el proceso de desarrollo de software del equipo.

✓ **Día de trabajo**

El objetivo en esta etapa es obtener el release funcional, se inicia con la tarea de documentación para la tarea es necesario el testing intensivo y la retroalimentación de los desarrolladores.

✓ **Documentación**

En esta etapa se completa la documentación del sistema caso contrario sería un desastre al momento de comunicarse con el cliente.

✓ **Día del release**

La tarea de esta etapa es verificar si todo ha sido correctamente unido, las auditorías se hacen presentes.

2.6.4.5 **Pruebas del Sistema**

En esta etapa se hará pruebas y correcciones si se presenta errores en caso de que no se cumpla con los requerimientos del cliente.

✓ **Pruebas del Sistema**

Las pruebas en esta fase se retroalimentan de las pruebas que se hicieron en las anteriores fases y se corrobora si solucionaron los defectos encontrados.

✓ **Día de Planeación**

Se planifica los contenidos o las pruebas necesarias para culminar el desarrollo para la tarea se estable un porcentaje de aceptación en del testing.

✓ **Día de Trabajo**

En esta etapa se obtendrá el release final del sistema desarrollado, se documentará los últimos artefactos y se informará a al cliente esperando una retroalimentación.

✓ **Día del Release**

Se publica el release final luego de que haya superado las pruebas necesarias para corroborar que todo ha sido implementado correctamente.

CAPÍTULO III

3 Desarrollo de la solución

El presente capítulo explica la aplicación propuesta mediante el uso de la metodología Mobile-D, detallada en el capítulo dos.

3.1 Fase de exploración

La fase de exploración permite definir las partes interesadas, el alcance y las iteraciones necesarias para cumplir los requerimientos.

3.1.1 Establecimiento de partes interesadas

Define las partes involucradas en el proyecto.

Equipo cliente

- Cliente (Empresa): UNACEM.
- Representante de la Empresa: Marlon Cadena.
- Usuarios Finales: Trabajadores de la planta industrial de cemento y oficinas Quito.
- Participación del Cliente: Sponsor y Tester de la aplicación.

Equipo desarrollo

- Líder del Proyecto: Alexander Orbes.
- Tutor de proyecto: Ing. Diego Trejo.
- Arquitecto y Analista Sponsor: Christian Suárez.

3.1.2 Especificación de requerimientos

3.1.2.1 Requerimientos iniciales

Se estable los requerimientos iniciales que fueron recolectados mediante con la participación de los grupos de interés.

Los requerimientos manifestados son:

- Inicio sesión con usuarios registrados en la plataforma VIDA.
- Registro, reporte, edición, eliminación de incidentes, incluir imágenes.
- Registro, reporte, edición, eliminación de incidentes reporte, edición, eliminación de incidentes de IVL y acciones de IVL, incluir imágenes.
- Registro, reporte, edición, eliminación de incidentes de acciones de inspecciones de área, incluir imágenes.
- Registro reporte, edición, eliminación de incidentes de plan vivencial
- Registro, reporte, edición, eliminación de incidentes de inspecciones estándar, incluir imágenes.
- Información almacenada localmente.
- Subir información a la plataforma VIDA
- Descargar información de la plataforma VIDA.

3.1.2.2 Diagramas de casos de uso

Se definen los requerimientos funcionales, analizando la arquitectura y problemas técnicos. Son demostrables mediante los diagramas de casos de uso UML.

Con los diagramas de casos de uso se define la usabilidad de la aplicación y las acciones de ejecución de quien está a cargo de ejecutarla, además señala los procesos a ejecutarse antes del inicio de cada caso de uso.

3.1.2.3 Caso de uso módulo Login

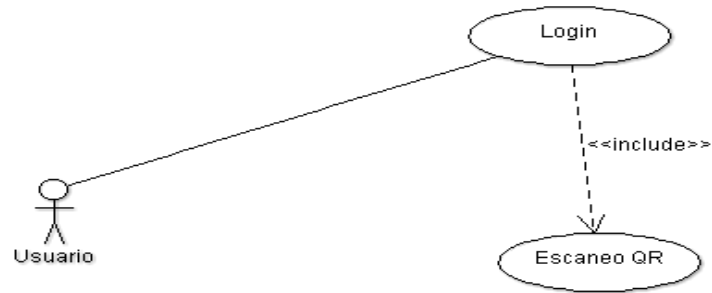


Fig. 21. Caso de uso login
Fuente: Propia.

La figura anterior muestra la tarea necesaria para la autenticación en la aplicación.

TABLA 4. Descripción del caso de uso login

CASO DE USO:	Login
Descripción	El usuario final es el que accede al aplicativo móvil, y para ello debe encontrarse registrado previamente en VIDA, y desde ahí escanea el código QR con la información de su usuario.
Actor	<ul style="list-style-type: none">• Usuario final
Condiciones previas	<ul style="list-style-type: none">• Encontrarse registrado con un rol de usuario no administrador en VIDA.
Flujo básico de eventos	<ul style="list-style-type: none">• Escanear código QR
Flujos alternativos	Configurar url/puerto del servidor
Escenario clave	Disponer de una cuenta de usuario en VIDA
Post-condiciones	Ninguno.

Fuente: Propia

3.1.2.4 Caso de uso Home

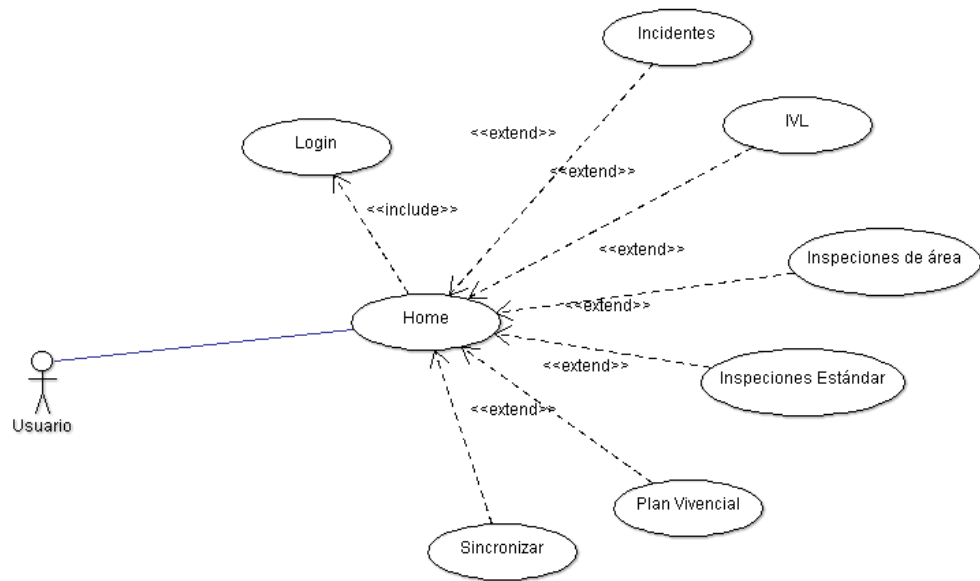


Fig. 22. Caso de uso home
Fuente: Propia

La figura anterior describe las funcionalidades a las que puede el usuario final en el home.

TABLA 5. Descripción del caso de uso Home

CASO DE USO:	Home
Descripción	El usuario final es el encargado de registrar los eventos en la aplicación móvil que posteriormente se sincronizará con el sistema central y para ello hace uso del home que le redirigirá a cada opción disponible en el aplicativo móvil.
Actor	<ul style="list-style-type: none">• Usuario final
Condiciones previas	<ul style="list-style-type: none">• Encontrarse registrado con un rol de usuario no administrador.• Encontrarse autenticado en la aplicación.
Flujo básico de eventos	<ul style="list-style-type: none">• Navegar hacia incidentes.• Navegar hacia IVL.

	<ul style="list-style-type: none"> • Navegar hacia inspecciones de área. • Navegar hacia plan vivencial • Navegar hacia inspecciones estándar • Sincronizar datos de la aplicación con el servidor y viceversa.
Flujos alternativos	No aplica.
Escenario clave	Disponer de una cuenta de usuario.
Post-condiciones	Ninguno.

Fuente: Propia.

3.1.2.5 Caso de uso módulo incidentes registro, consulta y sincronización.

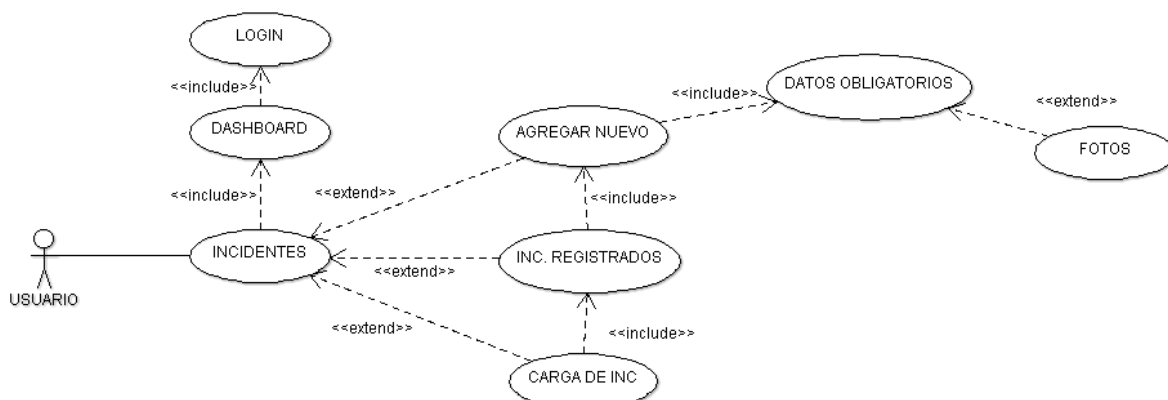


Fig. 23. Caso de uso registro, consulta y sincronización de incidentes

Fuente: Propia.

La figura anterior muestra el flujo de trabajo que el usuario final debe ejecutar para el registro, actualización y sincronización de incidentes.

TABLA 6. Descripción del caso de uso módulo de incidentes

CASO DE USO:	Registro, consulta y sincronización de incidentes
Descripción	El usuario final es el encargado de registrar incidentes mediante el aplicativo móvil ingresando toda la información necesaria para disponer de un registro óptimo y preciso.

Actor	<ul style="list-style-type: none"> • Usuario final
Condiciones previas	<ul style="list-style-type: none"> • Encontrarse registrado con un rol de usuario no administrador. • Encontrarse autenticado en la aplicación.
Flujo básico de eventos	<ul style="list-style-type: none"> • Navegar hacia el módulo de incidentes. • Navegar hacia registro de un nuevo incidente. • Navegar hacia el reporte de incidentes registrados. • Subir los incidentes registrados en la aplicación al servidor
Flujos alternativos	No aplica.
Escenario clave	Disponer de una cuenta de usuario.
Post-condiciones	Ninguno.

Fuente: Propia.

3.1.2.6 Caso de uso registro, consulta y sincronización de IVL

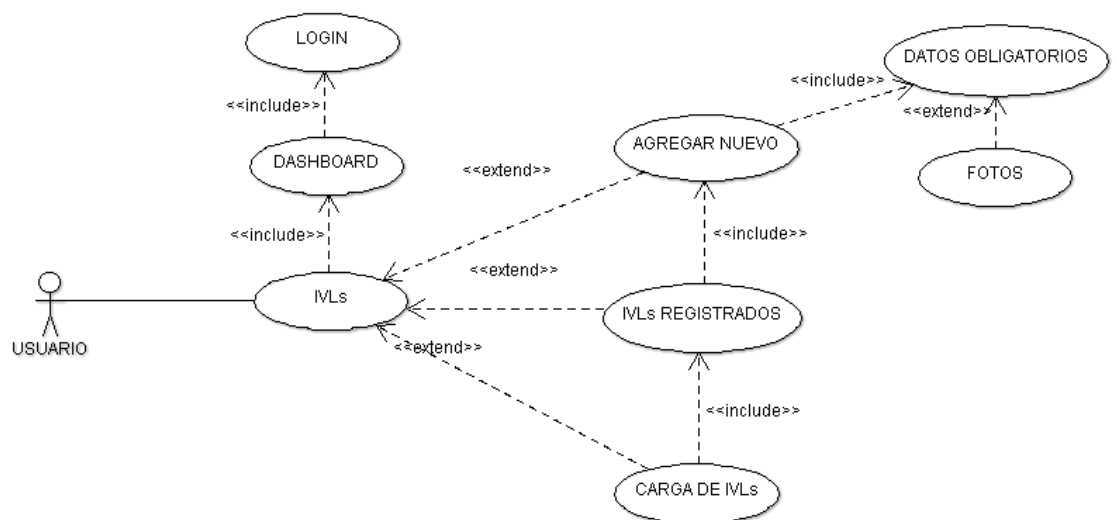


Fig. 24. Caso de uso registro, consulta y sincronización de IVL

Fuente: Propia.

La figura anterior describe el flujo de trabajo que el usuario final debe seguir para realizar el registro, consulta y sincronización de IVL.

TABLA 7. Descripción del caso de uso módulo de IVL

CASO DE USO:	Registro, consulta y sincronización de IVL
Descripción	El usuario final es el encargado de registrar IVL mediante el aplicativo móvil ingresando toda la información necesaria para disponer de un registro óptimo y preciso.
Actor	<ul style="list-style-type: none"> • Usuario final
Condiciones previas	<ul style="list-style-type: none"> • Encontrarse registrado con un rol de usuario no administrador. • Encontrarse autenticado en la aplicación.
Flujo básico de eventos	<ul style="list-style-type: none"> • Navegar hacia el módulo de IVL. • Navegar hacia registro de un nuevo IVL. • Navegar hacia el reporte de IVL registrados. • Subir de los IVL registrados en la aplicación al servidor
Flujos alternativos	No aplica.
Escenario clave	Disponer de una cuenta de usuario.
Post-condiciones	Ninguno.

Fuente: Propia.

3.1.2.7 Caso de uso registro, consulta y sincronización de planes vivenciales.

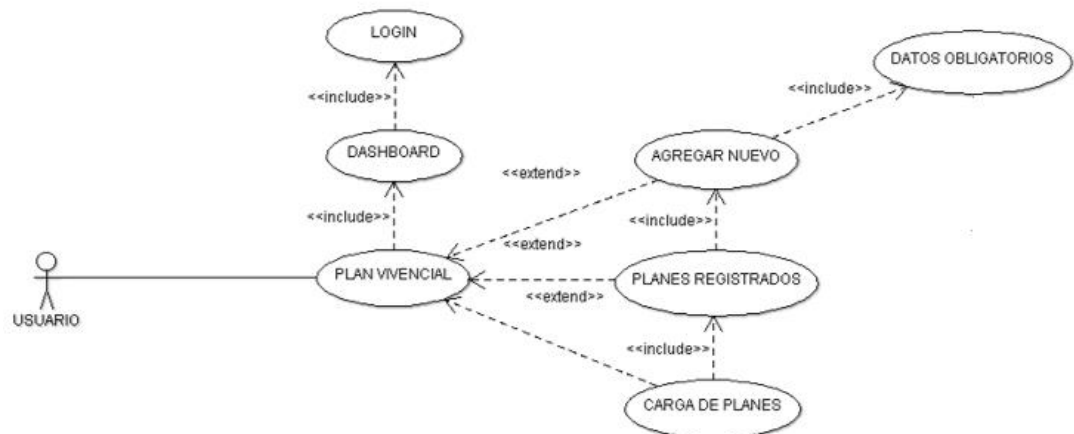


Fig. 25. Caso de uso registro, consulta y sincronización de Planes Vivenciales
Fuente: Propia.

La figura anterior permite mostrar el flujo de trabajo que el usuario final debe seguir para realizar el registro, consulta y sincronización de planes vivenciales.

TABLA 8. Descripción del caso de uso módulo de plan vivenciales

CASO DE USO:	Registro, consulta y sincronización de planes vivenciales
Descripción	El usuario final es el encargado de registrar planes vivenciales mediante el aplicativo móvil ingresando toda la información necesaria para disponer de un registro óptimo y preciso.
Actor	<ul style="list-style-type: none"> Usuario final
Condiciones previas	<ul style="list-style-type: none"> Encontrarse registrado con un rol de usuario no administrador. Encontrarse autenticado en la aplicación.
Flujo básico de eventos	<ul style="list-style-type: none"> Navegar hacia el módulo de Plan Vivencial Navegar hacia registro de un nuevo Plan Vivencial Navegar hacia el reporte de planes vivenciales registrados. Subir de los planes vivenciales registrados en la aplicación con al servidor.

Flujos alternativos	No aplica.
Escenario clave	Disponer de una cuenta de usuario.
Post-condiciones	Ninguno.

Fuente: Propia.

3.1.2.8 Caso de uso de inspecciones de área, nueva acción, subir registros.

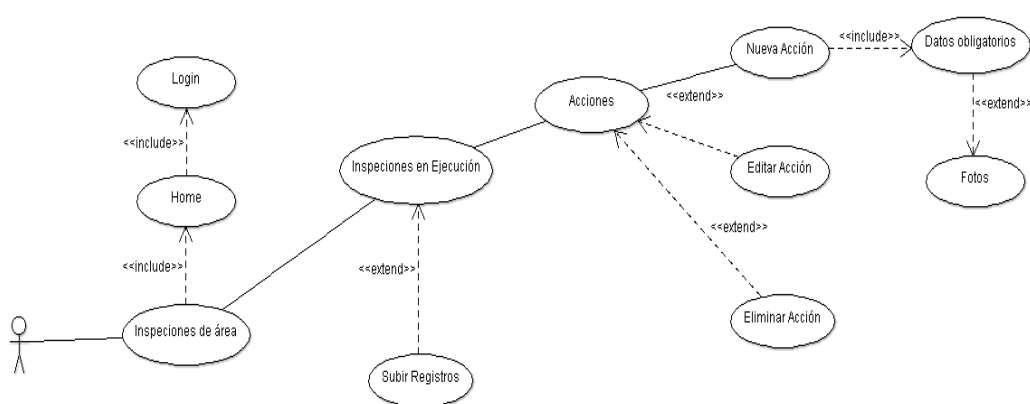


Fig. 26. Caso de uso inspecciones de área, agregar nuevo, subir registros.

Fuente: propia

La figura anterior muestra el flujo a seguir para agregar un registro inspecciones de área desde inspecciones de área existente.

TABLA 9. Descripción de caso de uso de inspecciones de área

CASO DE USO:	Registro, consulta y subida de acciones de inspecciones de área
Descripción	El usuario final registras acciones de inspecciones de área, editar eliminar los registros
Actor	<ul style="list-style-type: none"> Usuario final
Condiciones previas	<ul style="list-style-type: none"> Encontrarse registrado con un rol de usuario no administrador.

	<ul style="list-style-type: none"> • Encontrarse autenticado en la aplicación. • Descargar inspecciones de área en ejecución
Flujo básico de eventos	<ul style="list-style-type: none"> • Navegar hacia el módulo de Inspecciones de área • Navegar hacia nueva acción • Registrar nueva acción • Navegar hacia el reporte de acciones registradas • Navegar hacia inspecciones en ejecución, subir acciones al servidor.
Flujos alternativos	<ul style="list-style-type: none"> • Eliminar acciones. • Editar acciones.
Escenario clave	<ul style="list-style-type: none"> • Haber iniciado sesión en la aplicación • Descargar inspecciones de área en ejecución
Post-condiciones	Ninguno.

Fuente: Propia

3.1.2.9 Diagrama caso de usos inspecciones estándar, agregar nuevo.

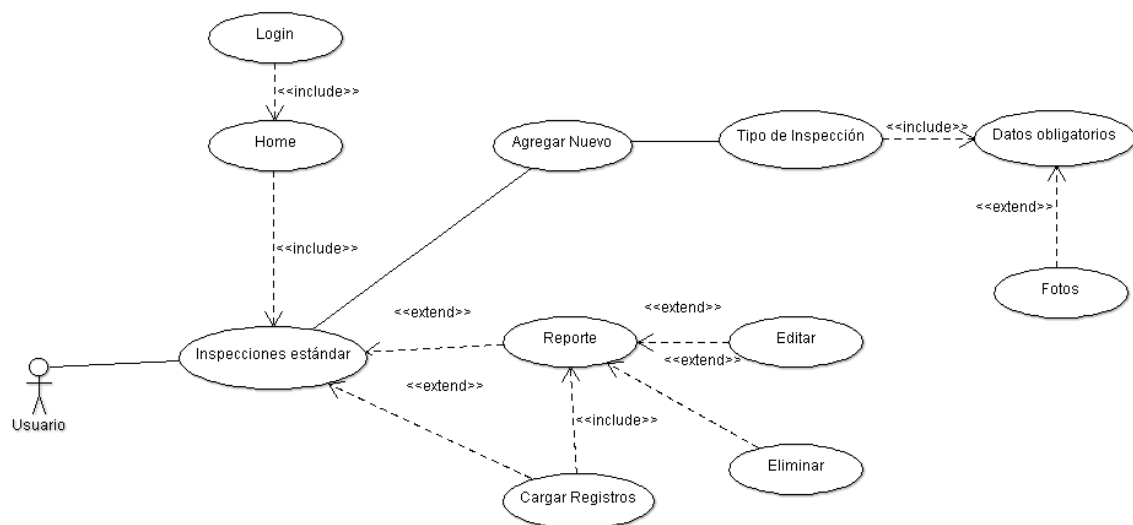


Fig. 27. Caso de usos inspecciones estándar, agregar nuevo, subir registros.

Fuente: Propia

La figura anterior muestra el flujo para agregar una nueva inspección estándar de cualquier tipo que haya escogido el usuario, Además los flujos alternativos para subir los registros al servidor, editar, eliminar registros.

TABLA 10. Descripción de caso de uso inspecciones estándar.

CASO DE USO:	Registro, consulta y subida de inspecciones estándar
Descripción	El usuario final registras acciones de inspecciones estándar, de cualquier tipo disponible, editar eliminar registros.
Actor	<ul style="list-style-type: none"> • Usuario final
Condiciones previas	<ul style="list-style-type: none"> • Encontrarse registrado con un rol de usuario no administrador. • Encontrarse autenticado en la aplicación. • Haber descargado los tipos de inspecciones estándar del servidor.
Flujo básico de eventos	<ul style="list-style-type: none"> • Navegar hacia el módulo de Inspecciones Estándar • Navegar hacia nuevo registro • Registrar nuevo • Navegar hacia el reporte de los registros
Flujos alternativos	<ul style="list-style-type: none"> • Eliminar acciones. • Editar acciones. • Subir inspecciones al servidor
Escenario clave	<ul style="list-style-type: none"> • Haber iniciado sesión en la aplicación • Descargar tipos de inspecciones estándar
Post-condiciones	Ninguno.

Fuente: Propia

3.1.3 Diagrama de Componentes.

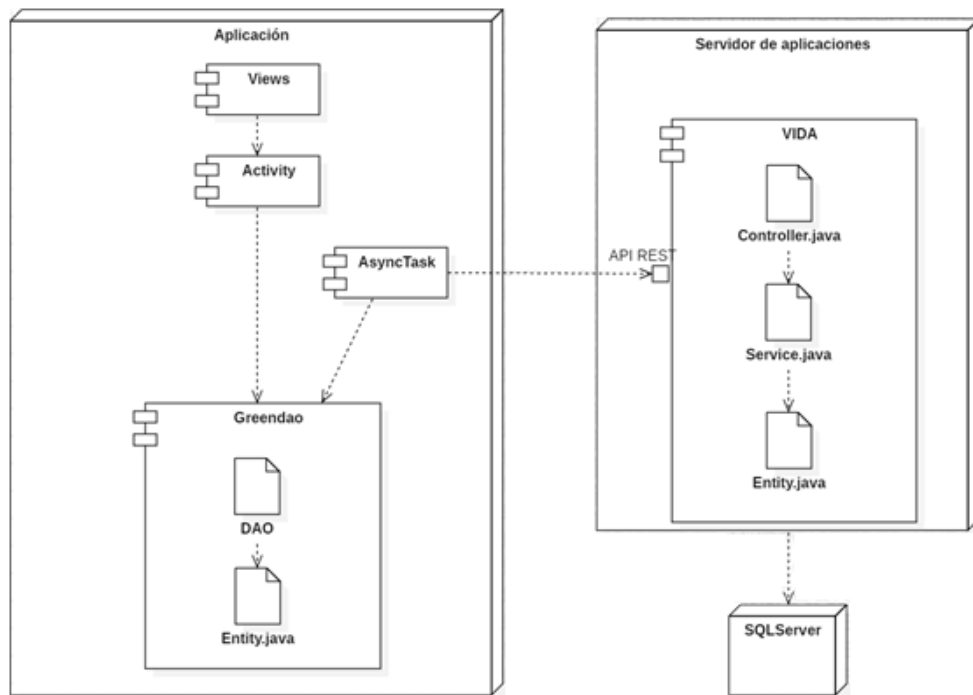


Fig. 28. Diagrama de componentes
Fuente: Propia

En el gráfico anterior se puede analizar la orquestación de los diferentes componentes que van a implementarse en la solución, la aplicación se va a comunicar con el servidor mediante un API Rest.

3.1.4 Requerimientos Funcionales

TABLA 11. Requerimientos funcionales

Código	Descripción	Categoría
RF001	La aplicación permitirá al usuario, registrar, consultar y actualizar los incidentes de campo.	Requerimiento
RF002	La aplicación permitirá al usuario, registrar, consultar y actualizar las interacciones visibles de liderazgo.	Requerimiento

RF003	La aplicación permitirá al usuario, registrar, consultar y actualizar información sobre las acciones de inspecciones de área	Requerimiento
RF004	La aplicación permitirá al usuario, registrar, consultar y actualizar información sobre los planes vivenciales.	Requerimiento
RF005	La aplicación permitirá el acceso mediante login a través de Código QR.	Mejora
RF006	La aplicación dispondrá de un home que permita al usuario acceder a las opciones de forma intuitiva.	Mejora
RF007	La aplicación permitirá al usuario recuperar sus datos guardados en la plataforma VIDA	Requerimiento

Fuente: Propia.

3.1.5 Módulos de la Aplicación

TABLA 12. Descripción de módulos

Módulo	Código	Proceso	Requerimientos
Módulo de Incidentes	M001	Registro general de incidentes en el campo detectado por parte de los usuarios, estos pueden ser por parte de personal planta y por personal de logística, por lo que se debe añadir categorías al incidente.	RF001

Módulo de ILV	M002	Registro de IVL y sus respectivas Acciones, Además se debe permitir al usuario recibir notificaciones de acciones ya cerradas	RF002
Módulo de Registro del Inspecciones de Área	M003	Se debe registrar las acciones de inspecciones que haga el grupo asignado, se debe agregar acciones solo a inspecciones en ejecución	RF003
Módulo de Inspecciones Estándar	M004	Se debe crear 13 tipo de inspecciones para el registro de las de las inspecciones con los diferentes estándares existentes	RF003
Módulo de planes vivenciales	M005	Se debe registra los acompañamientos a conductores mediante el plan vivencial, los cuales posteriormente se subirán a VIDA	RF004
Módulo de navegación	M006	Se debe navegar dentro de la aplicación mediante	RF006

		un Home que contenga los accesos directos a cada módulo.	
Módulo de Sesión	M007	Se debe iniciar sesión en la aplicación para disponer de acceso a los datos guardados por cada cuenta.	RF005, RF007

Fuente: Propia.\

3.1.6 Establecimiento del Proyecto

En esta etapa de la primera fase permite definir las tecnologías a utilizarse, lenguajes de programación, IDE de desarrollo, patrones de diseño, maquetación del sistema y equipos necesarios para una correcta implementación.

TABLA 13. Tecnologías seleccionadas para la implementación del proyecto

Tecnología de desarrollo	Android
Lenguaje de Programación	Java (JDK 8)
IDE de Desarrollo	Android Studio 3.1.3
Versión de Android	Desde 4.4 Kitkat o superior
Tecnología de Servidor	Jboss AS 7.1.1
Tecnología de la Plataforma Web	Spring MVC versión 3
Equipos	Laptop Dell Inspiron i7(desarrollo), Xiaomi Mi A1(pruebas)
Metodología de Desarrollo	Mobile-D
Bases de Datos	SQLite y SQLServer en Servidor
Librerías	GreenDao, ResTemplate, StepOneLibrary, BarCodeScannder.
Patrones arquitectónicos	MVC

Fuente: Propia.

3.2 Inicialización del proyecto

En esta fase se realiza el análisis de los aspectos críticos para desarrollar la aplicación, entrenamiento necesario del equipo, planificaciones, configuraciones del entorno.

3.2.1 Diseño de la base de datos

Se realiza el diseño entidad relación de la base de datos. Véase en la Fig. 28.

3.2.2 Planificación de fases

La planificación inicial se culmina con la planeación de las fases y las iteraciones que tendremos en el desarrollo.

TABLA 14. Planificación de iteraciones.

Fase	Iteración	Descripción
Exploración	Iteración 0	Requerimientos iniciales, casos de usos, establecimiento del proyecto.
Inicialización	Iteración 1	Planificación del proyecto, esquema de la base de datos
Producción	Iteración 2	Creación de los API Rest en Spring MVC, para descargar lista de usuarios, lista de departamentos, lista de áreas, inicio de sesión, registro de dispositivo móvil, etc.
	Iteración 3	Implementación del generador de código QR en la plataforma VIDA, control de usuarios
	Iteración 4	Implementación de login con escáner de código QR generado en VIDA. Registro de dispositivos.
	Iteración 5	Implementación de la capa de persistencia,

		creación de entidades con greendao.
	Iteración 6	Implementación del home, creación de menús de navegación drawer. Descarga de información del servidor.
	Iteración 7	Implementación del módulo incidentes, generación de historias de usuario, diseño de la interface, pruebas de aceptación.
	Iteración 8	Refactorización de módulo de incidentes, creación de dos tipos adiciones de incidentes
	Iteración 9	Implementación de componente agregar fotos, desde galería y desde cámara, será usado en todos los módulos donde se necesite imágenes
	Iteración 10	Implementación del módulo de IVL, acciones de IVL y diseño de interface, historias de usuario y pruebas de aceptación.
	Iteración 11	Implementación del módulo de

		inspecciones de área, acciones de inspecciones de área, diseño de interface, historias de usuario y pruebas de aceptación.
	Iteración 12	Implementación del módulo de plan vivencial, diseño interfaz, historias de usuario, pruebas de aceptación
	Iteración 13	Implementación del módulo de inspecciones estándar, creación dinámica de vistas para 13 tipos de inspecciones estándar, historias de usuario, diseño de interfaz, pruebas de aceptación,
	Iteración 14	Pruebas de aceptación general, depuración interfaz, debug, refactorización de funcionalidades, corrección de errores los API Rest,

Fuente: Propia.

3.2.3 Lista de verificación del Plan de Proyecto

TABLA 15. Artefacto de la fase Exploración del Proyecto

Fase de Exploración			
Requerimientos Iniciales	Si	No	N/A
1. Todos los requerimientos funcionales iniciales han sido incluidos en el plan de proyecto	✓		
2. Todos los requisitos no funcionales iniciales han sido incluidos en el plan del proyecto	✓		
Horario y Ritmo			
1. El horario general se ha incluido en el plan del			✓
2. El ritmo previsto (número y duración de cada fase y sus iteraciones) se han definido en el plan del proyecto			✓
Recursos	Si	No	N/A
1. Plan del proyecto se ha actualizado con los grupos de interés identificados y sus miembros (por ejemplo, grupo de clientes, manejo de grupo, grupo de arquitectura)	✓		
2. Plan del proyecto se ha actualizado la información sobre las herramientas de desarrollo de software seleccionado, terminales, etc.	✓		
3. Plan del proyecto se ha actualizado con los miembros del equipo proyecto identificado			✓
	Si	No	N/A
1. Capacitación necesita de equipo del proyecto se han incluido en el plan del proyecto	✓		
2. Programa de formación ha sido incluido en el plan del proyecto			✓
Documentación	Si	No	N/A
1. Los documentos que se producirán en el proyecto han sido incluidos en el plan del proyecto (por ejemplo, arquitectura y documentación de la interfaz de usuario)	✓		
2. La vida útil de cada documento se ha incluido en el plan del proyecto (por ejemplo, contenidos, actualización)	✓		
Aseguramiento de la calidad	Si	No	N/A
1. El aseguramiento de la calidad procedimientos se han definido en el plan del proyecto para cada producto de trabajo (documentación, código y producto)	✓		
Iteración 0			

Requisitos	Si	No	N/A
1. Se ha actualizado el plan de proyecto relativa a los requisitos de prueba seleccionados para la iteración 0	✓		
2. El plan del proyecto se ha actualizado con respecto a la realización de los requisitos de prueba seleccionados para la iteración 0	✓		
3. Definición de la línea arquitectura ha sido incluido en el plan del proyecto	✓		

Fuente: Propia.

3.3 Fase de Producción

En esta fase se desarrollarán 4 etapas donde se configurará el ambiente de desarrollo hasta lanzar un primer release de la aplicación.

3.3.1 Configuración del proyecto

Se configura y sincroniza las herramientas de desarrollo.

Instalación de Android Studio.



Fig. 30. Inicialización de Android Studio
Fuente: Propia.

Se configura el Backend⁷ en el servidor remoto para hacer uso de los servicios Rest y crear una comunicación segura entre las partes.

⁷ Backend: Servidor donde está desplegados los API Rest.

```

C:\WINDOWS\system32\cmd.exe
Calling "C:\Users\alex_\jboss-as-7.1.0.Final\bin\standalone.conf.bat"
-----
JBoss Bootstrap Environment
JBOSS_HOME: C:\Users\alex_\jboss-as-7.1.0.Final
JAVA: C:\Program Files\Java\jdk1.7.0_75\bin\java
JAVA_OPTS: -XX:TieredCompilation -Dprogram.name=standalone.bat -Xms64M -Xmx512M -XX:MaxPermSize=256M -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true -Djboss.modules.system.pkgs=org.jboss.byteman -Djboss.server.default.config=standalone.xml
-----
01:04:45,222 INFORMACIÓN [org.jboss.modules] JBoss Modules version 1.1.1.GA
01:04:47,113 INFO [org.jboss.msc] JBoss MSC version 1.0.2.GA
01:04:47,241 INFO [org.jboss.as] JBA5015899: JBoss AS 7.1.0.Final "Thunder" starting

```

Fig. 31. Inicialización de Jboss

Fuente: Propia.

3.3.2 Historia de usuario Login

Estas se realizan en base a la plantilla proporcionada por la metodología y los requerimientos funcionales.


TABLA 16. Historia de usuario iniciar sesión

TABLA 10. Historia de usuario iniciar sesión

ID	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
1	Nueva	Fácil	Fácil	10h	20h	Alta
	Arreglo	Moderada	Moderada			
	Mejora	Difícil	Difícil			
Descripción: Login						
Un usuario que posea una cuenta activa en VIDA puede:						
Éxito: Código QR Válido.						
1. El usuario puede iniciar sesión						
2. El usuario accede a home						
Falla: Código QR Invalido, permisos hardware						
1. El usuario visualiza mensaje de error						
2. Usuario no puede ingresar a Home						
3. Usuario procede a verificar dirección del servidor.						
4. El usuario no ha dado permisos a la cámara.						

Fuente: Propia

3.3.2.1 Diseño de interfaz de login.

 <p>Fig. 32. Diseño Login</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • La pantalla de bienvenida es un login donde se pide al usuario escanear un código QR • La opción de Escanear Código QR abre la cámara con el lector del código. • En caso de actualizar la dirección del servidor puede hacerlo en la opción Configuraciones
--	--

3.3.3 Historia de usuario Home

TABLA 17. Historia de usuario home

ID	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
	Nueva	Fácil	Fácil	10h	15h	Alta
	Arreglo	Moderada	Moderada			
	Mejora	Difícil	Difícil			
Descripción: Home						
Cuando el usuario inicia sesión exitosamente, él puede						
Éxito: Se muestra opción de descargar datos.						
1. El usuario puede iniciar descarga de datos del servidor						
2. El usuario visualiza información de usuario y correo en menú drawer						

3. El usuario puede cerrar sesión
4. El usuario puede navegar a los 5 módulos de la aplicación

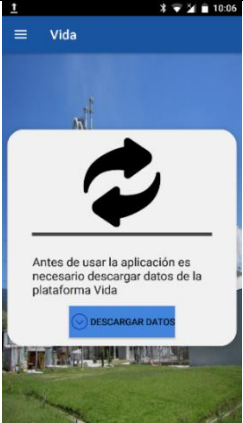

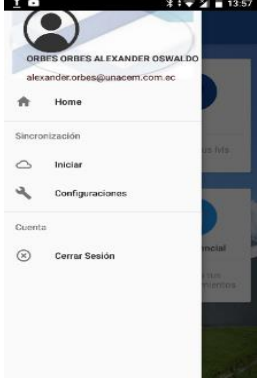
Falla: El usuario no puede visualizar Home

1. El usuario inicia descarga de datos
2. El usuario cierra sesión

-++Fuente: Propia

3.3.3.1 Diseño interfaz home

Tabla 18. Diseño interfaz home

 <p>Fig. 33. Diseño home sincronizar</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • El usuario al iniciar la sesión por primera vez se le solicita descargar los datos del servidor
 <p>Fig. 34. Diseño home sincronizado</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • Home presenta 5 módulos a los cuales el usuario puede acceder. • Incidentes • IVL • Inspecciones de área • Plan Vivencial • Inspecciones estándar
 <p>Fig. 35. Menú home</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • En el menú drawer el usuario puede iniciar la descarga de datos • El usuario puede configurar dirección del servidor. • El usuario puede cerrar sesión

Fuente: Propia

3.3.4 Historia de usuario incidentes

Tabla 19. Historia de usuario menú incidentes

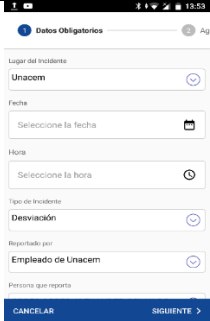
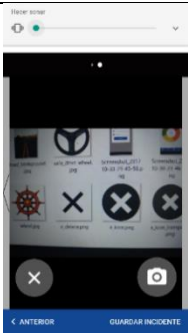

ID	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
3	Nueva	Fácil	Fácil	10h	30h	Alta
	Arreglo	Moderada	Moderada			
	Mejora	Difícil	Difícil			
Descripción: Módulo incidentes						
Cuando el usuario selecciona incidentes en home, él puede						
Éxito: Se visualiza opciones de incidentes						
1. El usuario puede agregar un nuevo incidente						
2. El usuario puede navegar a los registros.						
3. El usuario puede iniciar subida de registros a VIDA						
Falla: Mensaje de error						
1. El usuario no puede registrar incidentes						
2. El usuario no puede agregar imágenes.						
3. No se completó los campos obligatorios						

Fuente: Propia

3.3.4.1 Diseño interfaz incidentes

Ver el diseño en la tabla 20.

TABLA 20. Diseño interfaz incidentes

 <p>Fig. 36. Diseño agregar nuevo incidente.</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • El usuario inicia el registro de incidentes, todos los campos están validados. • El usuario puede cancelar el registro o continuar con el siguiente paso
 <p>Fig. 37. Diseño agregar fotos incidentes</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • El usuario puede agregar varias fotos al registro. • El usuario puede agregar fotos desde cámara o galería. El usuario puede eliminar imágenes agregadas
 <p>Fig. 38. Diseño reporte incidentes</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • El usuario puede visualizar los registros de incidentes. • Con la opción editar se puede modificar la información del registro. • Con la opción eliminar se borra el registro de la base de datos local.

Fuente: propia

3.3.5 Historia de usuario IVL

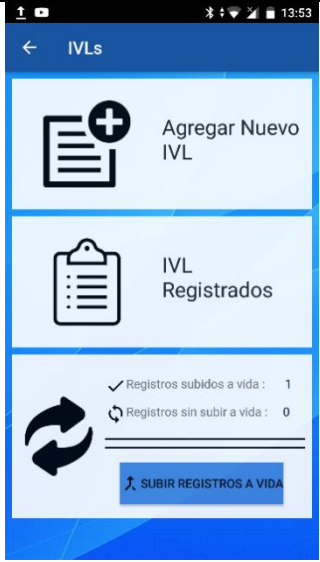
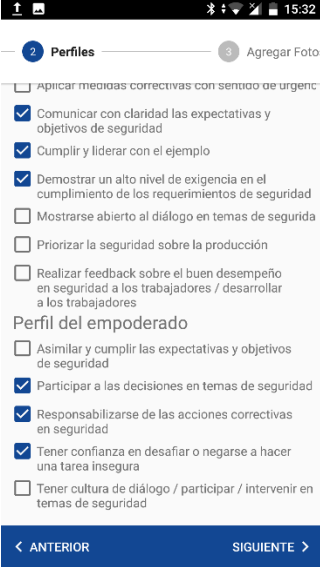
TABLA 21. Historia de usuario IVL

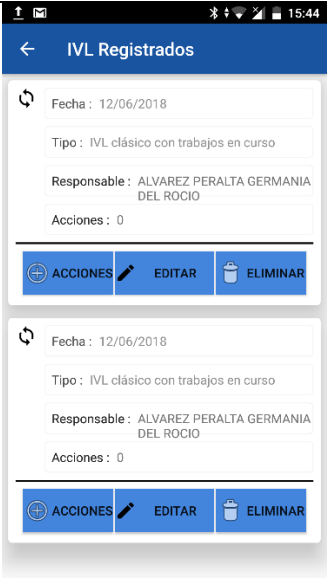
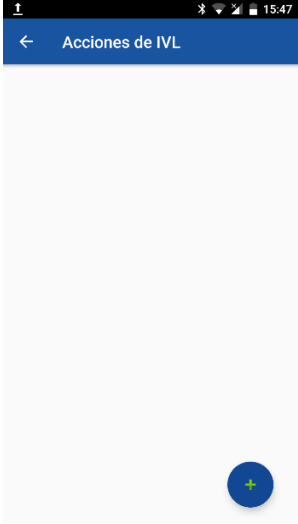
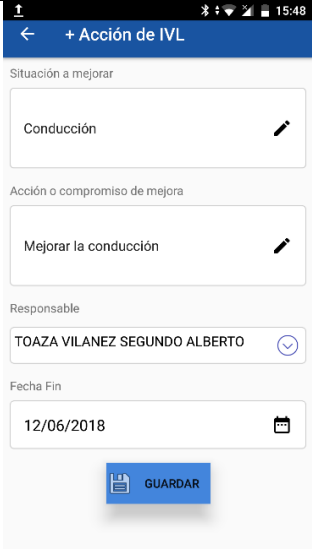
ID	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
	Nueva	Fácil	Fácil	10h	20h	Alta
	Arreglo	Moderada	Moderada			
	Mejora	Difícil	Difícil			
Descripción: Módulo IVL						
<p>Quando el usuario selecciona IVL en home, él puede</p> <p>Éxito: Se visualiza menú IVL</p> <ol style="list-style-type: none"> 4. El usuario puede agregar un nuevo IVL 5. El usuario puede navegar a los registros. 6. El usuario puede agregar acciones a IVL 7. El usuario puede iniciar subida de registros a VIDA <p>Falla: Mensaje de error</p> <ol style="list-style-type: none"> 4. El usuario no puede registrar IVL. 5. El usuario no puede agregar imágenes. 6. No se completó los campos obligatorios. 7. El usuario no puede agregar acciones a IVL. 						

Fuente: Propia

3.3.5.1 Diseño interface módulo IVL

TABLA 22. Diseño Interfaces módulo de inspecciones de área

 <p>Fig. 40</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • Se muestra el menú del IVL • Con la opción agregar nuevo el usuario inicia el registro del IVL • El usuario puede visualizar los registros de IVL • El usuario puede subir registros al servidor con la opción Subir Registros a Vida
	<ul style="list-style-type: none"> • En el segundo paso del wizard agregar IVL el usuario puede seleccionar los perfiles de IVL. • El usuario retrocede con el botón anterior • El usuario Avanzar con el botón siguiente. • El paso no es obligatorio

	<ul style="list-style-type: none"> • El usuario puede navegar en los registros de IVL • El usuario puede editar un registro • El usuario puede eliminar un registro • El usuario puede agregar acciones al IVL
	<ul style="list-style-type: none"> • Con acciones de IVL el usuario navega al listado de acciones de IVL • El usuario puede agregar uno nuevo con el botón flotante.
	<ul style="list-style-type: none"> • El usuario puede agregar acciones al IVL registrado. • Los campos están debidamente validados. • La misma pantalla se usa en la edición de acciones.

Fuente: Propia.

3.3.6 Historia de usuario Inspecciones estándar

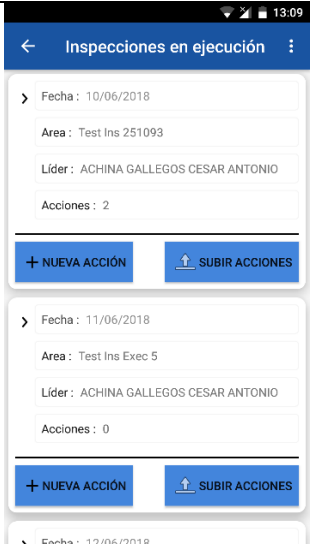
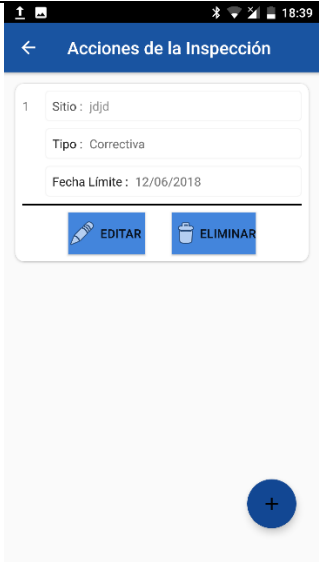
TABLA 23. Historia de usuarios inspecciones estándar

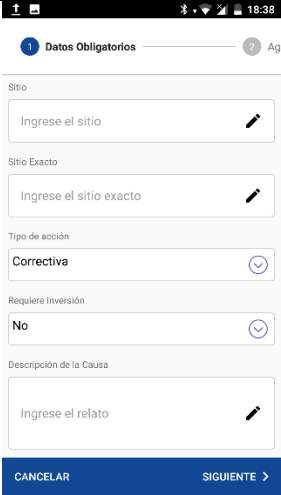
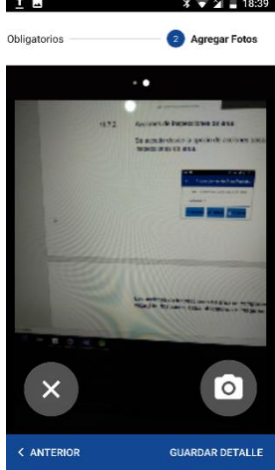
ID	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
5	Nueva	Fácil	Fácil	10h	20h	Alta
	Arreglo	Moderada	Moderada			
	Mejora	Difícil	Difícil			
Descripción: Módulo Inspecciones de área						
Cuando el usuario selecciona inspecciones de área en home, él puede						
Éxito: Se visualiza inspecciones en ejecución						
8. El usuario puede agregar descargar inspecciones en ejecución desde VIDA						
9. El usuario puede agregar acciones a inspecciones en ejecución						
10. El usuario puede iniciar subida de acciones a VIDA						
Falla: Mensaje de error						
8. El usuario no actualizar inspecciones en ejecución						
9. El usuario no puede agregar acciones						
10. El usuario no puede subir acciones a VIDA						

Fuente: Propia

3.3.6.1 Diseño interfaz módulo de inspecciones de área.

TABLA 24. Diseño del módulo de inspecciones de área

Lista de inspecciones en ejecución	
	<ul style="list-style-type: none"> • El usuario puede visualizar una lista de lista de inspecciones estándar. • El usuario puede actualizar la lista realizando un swipe hacia abajo. • El usuario puede agregar nuevas acciones. • El usuario puede subir acciones al servidor.
Lista de acciones	
	<ul style="list-style-type: none"> • El usuario puede visualizar la listado inspecciones en ejecución • El usuario puede editar acciones registradas • El usuario puede eliminar acciones registradas • El usuario con el botón flotante puede agregar nueva acción.
Agregar nueva acción	

	<ul style="list-style-type: none"> • El usuario ingresa los datos obligatorios en el primer paso del wizard. • Los campos obligatorios están debidamente validados. • Con la opción cancelar retorna a la lista de acciones. • Con la opción siguiente continua con el siguiente paso.
<p style="text-align: center;">Agregar fotos nueva acción.</p>	
	<ul style="list-style-type: none"> • El usuario puede agregar una o varias fotos desde galería o cámara. • El usuario puede eliminar imágenes agregadas. • Con la opción anterior el usuario regresa al paso anterior. • Con la opción siguiente el usuario guarda la acción.

Fuente: propia

3.3.7 Historia de usuario plan vivencial.

TABLA. Historia de usuario plan vivencial

ID	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
5	Nueva	Fácil	Fácil	10h	10h	Alta
	Arreglo	Moderada	Moderada			
	Mejora	Difícil	Difícil			

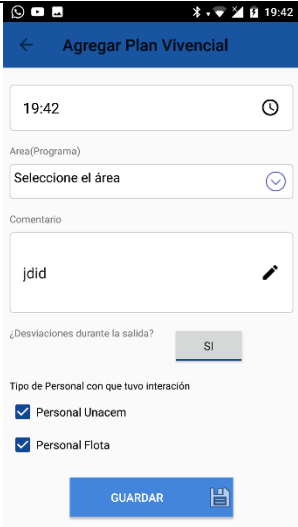
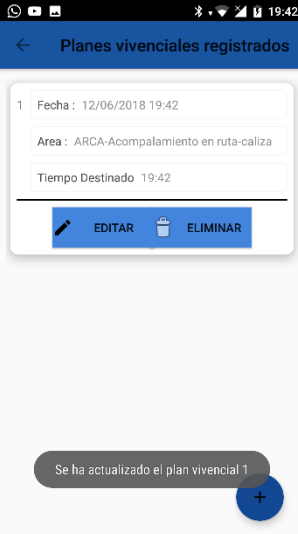
Descripción: Módulo de Plan Vivencial
<p>Cuando el usuario selecciona plan vivencial en home, él puede</p> <p>Éxito: Se visualiza menú de plan vivencial</p> <ol style="list-style-type: none"> 1. El usuario puede agregar nuevo plan vivencial 2. El usuario puede visualizar registros de planes vivenciales 3. El usuario puede iniciar subida de registros a VIDA <p>Falla: Mensaje de error</p> <ol style="list-style-type: none"> 1. El usuario no puede agregar nuevos registros 2. El usuario no puede subir acciones a VIDA

Fuente: Propia.

3.3.7.1 Diseño interfaz módulo plan vivencial.

TABLA 25. Diseño interfaz módulo plan vivencial

Menú plan vivencial	
	<ul style="list-style-type: none"> • El usuario puede agregar un nuevo registro con la primera opción. • El usuario puede visualizar registros con la segunda opción. • El usuario visualiza en la tercera opción registros sin subir a VIDA y registros subidos a vida y un botón para subir los registros.
Agregar un nuevo plan vivencial	

 <p>Fig. 39. Nuevo plan vivencial</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • El usuario puede agregar un nuevo registro y editarlo en la misma pantalla. • Los campos están debidamente validados
Listado de planes vivenciales	
 <p>Fig. 40. Planes vivenciales registrados</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • El usuario puede visualizar registros de planes vivenciales • El usuario puede editar registros • El usuario puede eliminar registros • El usuario puede agregar nuevo registro.

Fuente: Propia

3.3.8 Historia de usuario inspecciones estándar

TABLA 26. Historia de usuario inspecciones estándar

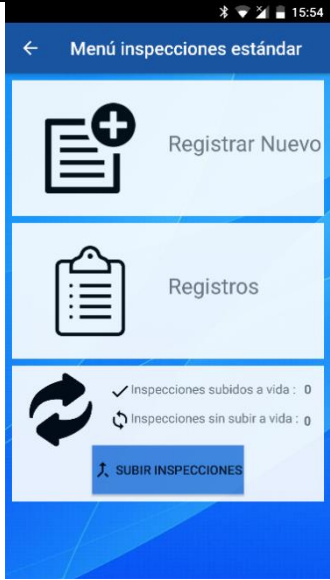
ID	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
6	Nueva	Fácil	Fácil	20h	30h	Alta
	Arreglo	Moderada	Moderada			


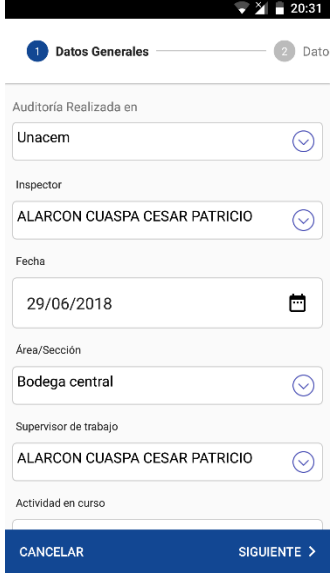
	Mejora	Difícil	Difícil			
Descripción: Módulo de inspecciones estándar						
<p>Cuando el usuario selecciona inspecciones estándar en home, él puede</p> <p>Éxito: Se visualiza menú de inspecciones estándar</p> <ol style="list-style-type: none"> 1. El usuario puede agregar nueva inspección estándar 2. El usuario puede visualizar registros de inspecciones estándar 3. El usuario puede iniciar subida de registros a VIDA <p>Falla: Mensaje de error</p> <ol style="list-style-type: none"> 1. El usuario no puede agregar nuevos registros 2. El usuario no puede subir acciones a VIDA. 3. El usuario no puede seleccionar tipos de inspecciones estándar. 						

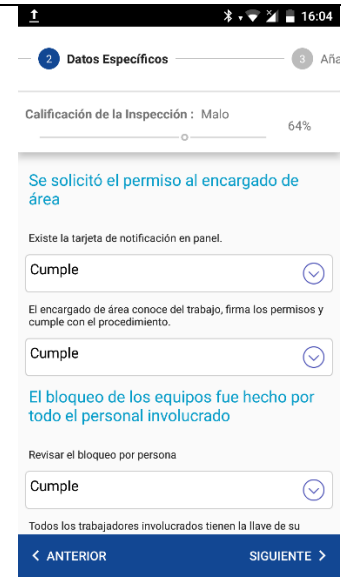
Fuente: Propia

3.3.9 Diseño de interfaces de módulo de inspecciones estándar.


TABLA 27. Diseño interfaz módulo inspecciones de área

Menú inspecciones estándar	
 <p>Fig. 41. Menú inspecciones estándar.</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • El usuario visualiza menú de inspecciones estándar. • El usuario con la primera opción puede navegar hacia la selección de tipos de inspección estándar. • El usuario con la segunda opción registros puede navegar a los registros de inspecciones estándar. • El usuario con la tercera opción puede iniciar subida de registros al servidor
Tipos de inspecciones estándar	

 <p>Fig. 42. Tipos de inspecciones estándar.</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> El usuario puede escoger trece tipos de inspecciones estándar y navegar hacia el wizard para registrar la inspección.
<p>Nueva Inspección estándar: Datos obligatorios</p>	
 <p>Fig. 43. Nueva inspección estándar-datos obligatorios</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> En el primer paso el usuario debe ingresar los datos generales de la inspección estándar. Los campos se cargan dinámicamente según el tipo de inspección estándar haya escogido el usuario. Con el botón siguiente el usuario avanza al siguiente paso.
<p>Nueva Inspección estándar: Datos específicos</p>	

 <p>Fig. 44. Nueva inspección estándar- datos específicos.</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • El usuario puede calificar la inspección estándar con los ítems cargados dinámicamente según el tipo de inspección estándar. • Los campos están debidamente validados • El usuario con la opción siguiente continúa con el último paso.
---	---

Nueva Inspección estándar: Agregar fotos

 <p>Fig. 45. Nueva inspección estándar-agregar fotos.</p> <p>Fuente: Propia</p>	<ul style="list-style-type: none"> • El usuario puede agregar fotos desde cámara o desde galería. • El usuario puede agregar cualquier cantidad de evidencia fotográfica. • El usuario completa el wizard con la opción guardar inspección.
---	--

Fuente: Propia

3.4 Pruebas de aceptación

Luego de culminar con la fase de desarrollo de los módulos se procede a realizar las respectivas pruebas de aceptación para verificar el cumplimiento de las respectivas historias de usuario.

3.4.1 Prueba de aceptación login

TABLA 28. Prueba de aceptación del login

Hoja de prueba de aceptación
TEST ID: 001
STORY: Login
Fecha de la Documentación: 10/05/2018
Fecha de la Ejecución: 10/05/2017
Superado: Sí
Descripción: <ol style="list-style-type: none">1. La pantalla debe mostrarse igual al diseño de la interfaz2. Se pide permisos de acceso a la cámara.3. Código QR incorrecto.4. Código QR correcto.5. Se abre el diálogo de configuración de ip/puerto
Resultado Esperado: <ol style="list-style-type: none">1. Inspección visual2. Se procede a escanear el código QR3. Se muestra error de escaneo.4. Se muestra un mensaje de inicio de sesión.5. Se muestra el dialogo con valores por default

Fuente: Propia.

3.4.2 Prueba de aceptación home

TABLA 29. Hoja de prueba de aceptación home

Hoja de prueba de aceptación
TEST ID: 002
STORY: Home
Fecha de la Documentación: 11/05/2018
Fecha de la Ejecución: 11/05/2017
Superado: Sí
Descripción: <ol style="list-style-type: none">1. La pantalla debe mostrarse igual al diseño de la interfaz2. Mensaje con opción de inicio de descarga de datos.3. Iniciar descarga de datos4. Navegar al menú lateral5. Cerrar sesión
Resultado Esperado: <ol style="list-style-type: none">1. Inspección visual2. Inspección visual3. Se inicia la descarga de datos indicando el porcentaje avanzado.4. Se muestra usuario y email del usuario.5. Se muestra mensaje de cierre de sesión y posterior cierre de sesión de la aplicación.

Fuente: Propia

3.4.3 Prueba de aceptación módulo incidentes

TABLA 30. Prueba de aceptación módulo incidentes

Hoja de prueba de aceptación
TEST ID: 003
STORY: Módulo incidentes
Fecha de la Documentación: 12/05/2018
Fecha de la Ejecución: 12/05/2017

Superado: Sí
Descripción: <ol style="list-style-type: none"> 1. La pantalla debe mostrarse igual al diseño de la interfaz. 2. Agregar nuevo incidente se debe mostrar tres opciones de incidente. 3. Se debe mostrar la cantidad de incidentes subidos y sin subir a VIDA. 4. Navegar al reporte de incidentes. 5. Agregar un incidente tipo planta mina. 6. Agregar un incidente tipo vehículo de logística planta/mina 7. Agregar un incidente tipo vehículo de logística carretera. 8. Eliminar un incidente en reporte. 9. Editar un incídete en reporte. 10. Agregar fotos en incidentes. 11. Subir incidentes a VIDA
Resultado Esperado: <ol style="list-style-type: none"> 1. Inspección visual 2. Inspección visual. Se muestra tres tipos de incidentes. 3. Inspección visual. Se muestra la cantidad de registros sincronizados. 4. Se muestra el listado de incidentes registrados. 5. Se abre un wizard con dos pasos: Datos obligatorios, Agregar fotos. 6. Se abre un wizard con tres pasos: Datos del vehículo, Datos del incidente, Fotos del incidente. 7. Se abre un wizard con cuatro pasos: Datos del vehículo, Datos del incidente, Infracciones, Fotos del incidente. 8. Se muestra mensaje de confirmación antes de eliminar. 9. Se abre un wizard de edición del incidente dependiendo del tipo de incidente a editar. 10. Se muestra un diálogo para agregar fotos desde galería o cámara. 11. Se muestra un diálogo de confirmación de subida de registros.

Fuente: Propia.

3.4.4 Prueba de aceptación módulo IVL

TABLA 31. Prueba de aceptación módulo IVL

Hoja de prueba de aceptación
TEST ID: 004
STORY: Módulo IVL
Fecha de la Documentación: 13/05/2018
Fecha de la Ejecución: 13/05/2017

Superado: Sí
Descripción: <ol style="list-style-type: none"> 1. La pantalla debe mostrarse igual al diseño de la interfaz. 2. Se debe mostrar la cantidad de IVL subidos y sin subir a VIDA. 3. Navegar al reporte de IVL. 4. Agregar nuevo IVL 5. Eliminar un IVL en reporte. 6. Editar un IVL en reporte. 7. Agregar fotos en IVL. 8. Subir IVL a VIDA 9. Agregar acciones a IVL en reporte de IVL 10. Agregar acciones en reporte de acciones 11. Guardar acción de IVL 12. Edita Acción de IVL
Resultado Esperado: <ol style="list-style-type: none"> 1. Inspección visual 2. Inspección visual. Se muestra la cantidad de registros sincronizados. 3. Se muestra el listado de IVL registrados. 4. Se abre un wizard con tres pasos: Perfiles, Agregar fotos. 5. Se muestra mensaje de confirmación antes de eliminar. 6. Se abre un wizard de edición del IVL con datos precargados. 7. Se muestra un diálogo para agregar fotos desde galería o cámara. 8. Se muestra un diálogo de confirmación de subida de registros. 9. Navega a reporte de acciones de IVL 10. Navega a formulario de nueva acción 11. Valida campos obligatorios, guarda la acción. 12. Navega a formulario de edición de acción con datos precargados.

Fuente: Propia

3.4.5 Pruebas de aceptación módulo inspecciones de área

TABLA 32. Prueba de aceptación módulo de inspecciones de área

Hoja de prueba de aceptación
TEST ID: 005
STORY: Módulo inspecciones de área
Fecha de la Documentación: 15/05/2018
Fecha de la Ejecución: 15/05/2017

Superado: Sí
Descripción: <ol style="list-style-type: none"> 1. El módulo debe mostrarse igual al diseño de la interfaz de la documentación. 2. Navegar a inspecciones de área. 3. Agregar acciones. 4. Subir acciones. 5. Agregar acciones en lista. 6. Agregar fotos. 7. Guardar acción. 8. Eliminar acción.
Resultado Esperado: <ol style="list-style-type: none"> 1. Inspección visual 2. Cuando es por primera vez debe aparecer la opción de actualizar con un swipe. 3. Agregar acciones navega a listado de acciones. 4. Se muestra diálogo de confirmación para iniciar subida de registros. 5. Se abre un wizard con dos pasos: Datos obligatorios, Agregar fotos. 6. Se abre diálogo de selección fotos desde cámara o galería 7. Muestra mensaje “Nueva acción registrada”. 8. Muestra diálogo de confirmación de eliminación de acción.

Fuente: Propia

3.4.6 Prueba de aceptación módulo de plan vivencial

TABLA 33. Prueba de aceptación módulo de plan vivencial

Hoja de prueba de aceptación
TEST ID: 006
STORY: Módulo de plan vivencial
Fecha de la Documentación: 16/05/2018
Fecha de la Ejecución: 16/05/2017
Superado: Sí
Descripción: <ol style="list-style-type: none"> 1. La pantalla debe mostrarse igual al diseño de la interfaz. 2. Se debe mostrar la cantidad de registros subidos y sin subir a VIDA. 3. Navegar al reporte de planes vivenciales.

<ol style="list-style-type: none"> 4. Agregar nuevo plan vivencial. 5. Guardar plan vivencial. 6. Eliminar un plan vivencial en reporte. 7. Editar un plan vivencial en reporte. 8. Subir planes vivenciales a VIDA
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. Inspección visual 2. Inspección visual. Se muestra la cantidad de registros sincronizados. 3. Se muestra el listado de planes vivenciales registrados. 4. Se abre el formulario de registro de plan vivencial. 5. Validad campos obligatorios, se muestra mensaje de confirmación. 6. Se muestra mensaje de confirmación antes de eliminar. 7. Se abre el formulario de edición con datos precargados 8. Se muestra un diálogo de confirmación de subida de registros.

Fuente: Propia.

3.4.7 Prueba de aceptación módulo de inspecciones estándar

TABLA 34. Prueba de aceptación módulo inspecciones estándar

Hoja de prueba de aceptación
TEST ID: 006
STORY: Módulo de inspecciones estándar.
Fecha de la Documentación: 17/05/2018
Fecha de la Ejecución: 17/05/2017
Superado: Sí
<p>Descripción:</p> <ol style="list-style-type: none"> 1. La pantalla debe mostrarse igual al diseño de la interfaz. 2. Se debe mostrar la cantidad de registros subidos y sin subir a VIDA. 3. Navegar al reporte de inspecciones estándar. 4. Agregar nueva inspección estándar. 5. Seleccionar tipo de inspección estándar 6. Navegar al primer paso de la inspección estándar 7. Navegar al segundo paso de la inspección estándar y calificar inspección. 8. Agregar fotos al registro de la inspección.

9. Eliminar inspección
10. Editar inspección.
11. Subir inspecciones a VIDA

Resultado Esperado:

1. Inspección visual
2. Inspección visual. Se muestra la cantidad de registros sincronizados.
3. Se muestra el listado de inspecciones estándar registrados.
4. Se la pantalla de selección de tipos de inspecciones estándar.
5. Se abre un wizard con tres pasos: Datos generales, Datos específicos, Agregar fotos.
6. Se posiciona en el paso Datos generales con los campos generados dinámicamente según el tipo de inspección seleccionado.
7. Se posiciona en el paso Datos específicos con los ítems generados dinámicamente según el tipo de inspección y al calificar el porcentaje sube o baja, solo se puede continuar si se califica todos los ítems.
8. Se muestra diálogo con opciones de selección, cámara y galería
9. Se muestra diálogo de confirmación
10. Se abre el wizard con los campos precargados según el tipo de inspección estándar.
11. Se muestra diálogo de confirmación.

Fuente: Propia

3.5 Fase de Estabilización.

En esta fase se hace el merge(unión) de proyectos en caso de que haya desarrollado módulos en diferentes ambientes.

En este caso no se ha desarrollado soluciones aparte ya que hay un solo encargado, aun así el código fue versionado usando la herramienta Git y se subió a un repositorio remoto si se decidiera incluir más desarrolladores.

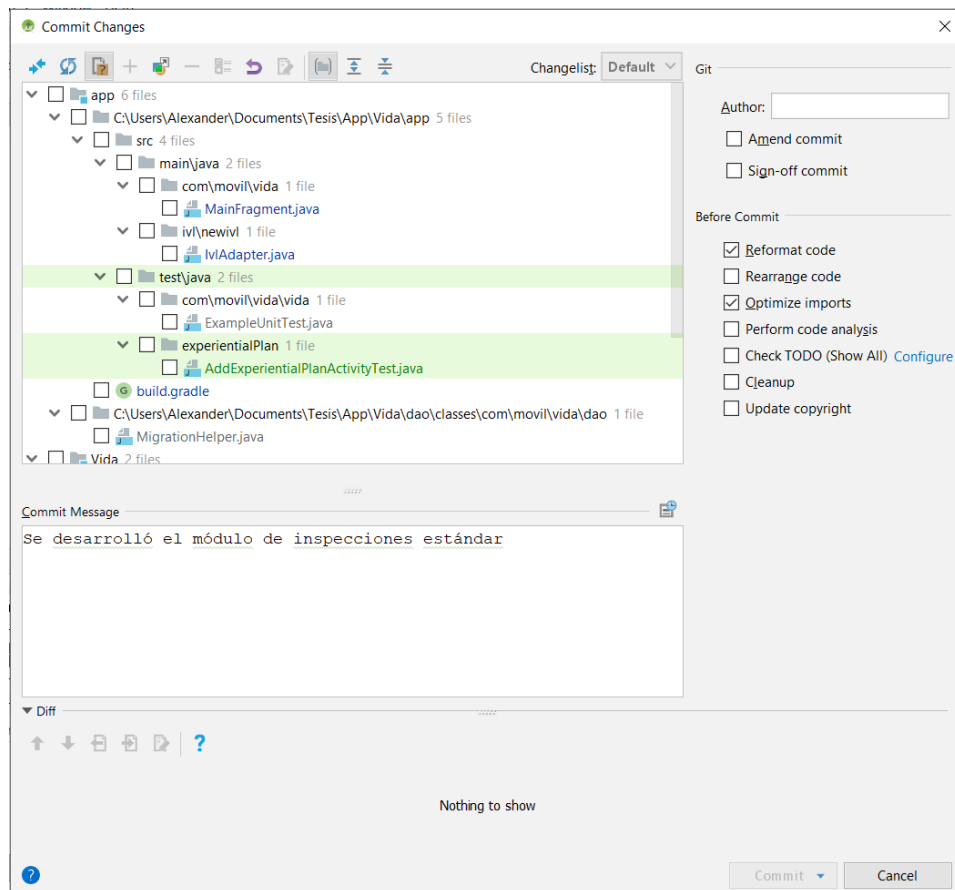


Fig. 46. Commit de módulo de inspecciones estándar.
Fuente: Propia

3.5.1 Historial de commits locales.

TABLA 35. Commits locales de la aplicación

Versión	Fecha	Autor	Mensaj e
ae3bf14	29/01/201	Alexande	Commit
	8	r	inicial
0a70ae1	05/02/201	Alexande	Home
	8	r	culminado
ec14691	12/02/201	Alexande	Módulo
	8	r	de incidentes culminada
9f7cd3a	29/02/201	Alexande	Módulo
	8	r	de inspecciones de IVL culminada

caa3296	8	25/04/201	r	Alexande	Módulo de inspecciones de área culminada
e63d2fe	8	25/05/201	r	Alexande	Módulo de inspecciones de plan vivencial culminada
7 b50c9d	8	25/06/201	r	Alexande	Módulo de inspecciones estándar culminada

Fuente: Propia

3.5.2 Push repositorio remoto.

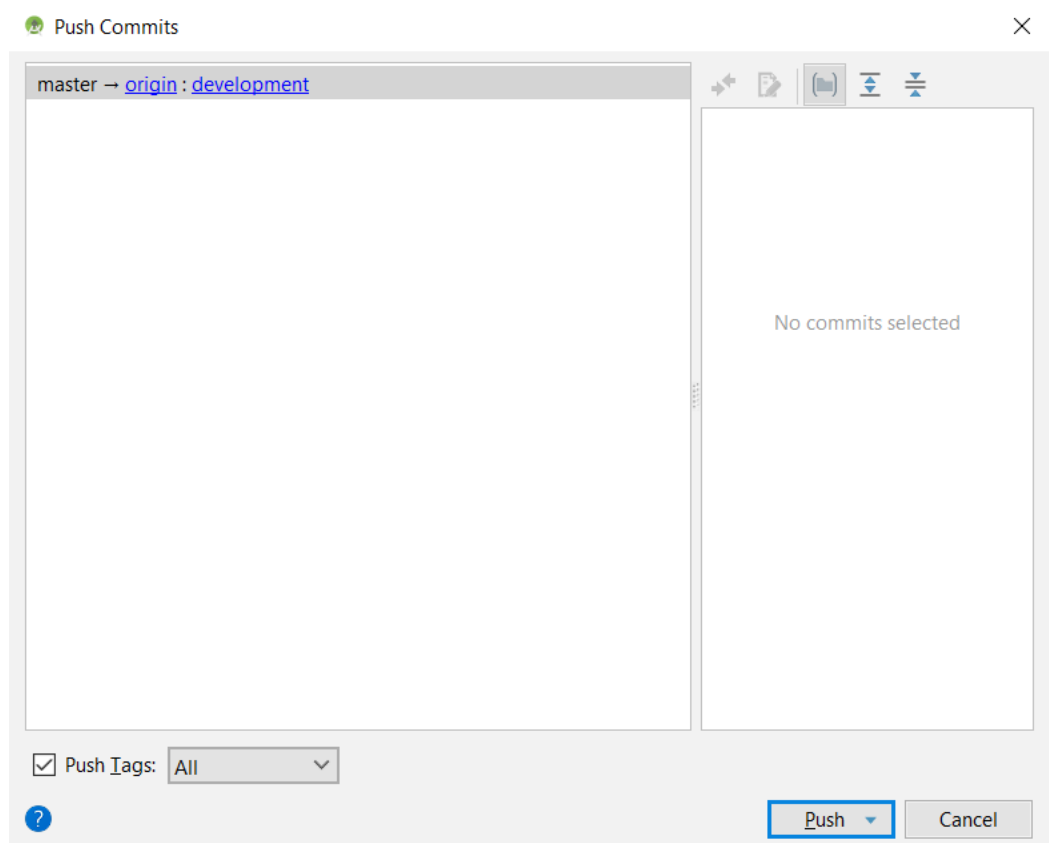


Fig. 47. Push Proyecto al branch development
Fuente: Propia

3.5.3 Repositorio remoto

Se hizo uso del servicio de bitbucket, que permite alojar los repositorios en privado que es una ventaja frente a su competencia GitHub.

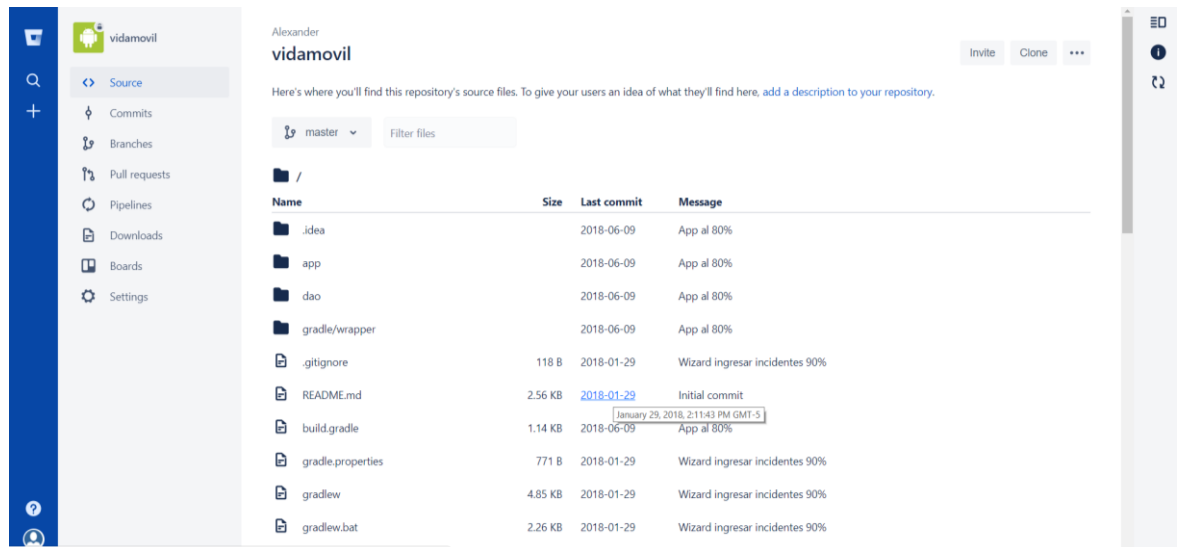


Fig. 48. Repositorio remoto-bitbucket
Fuente: Propia

3.6 Fase de correcciones y pruebas

En esta fase se va a subir la aplicación a la Play Store para iniciar con las pruebas alfa, beta y posterior puesta en producción. Se hará uso de herramientas de dispositivos emulados, físicos y herramientas como firebase para realizar las pruebas. En esta finaliza el proceso de desarrollo y la documentación.

3.6.1 Compilación de la aplicación

3.6.1.1 General APK firmado

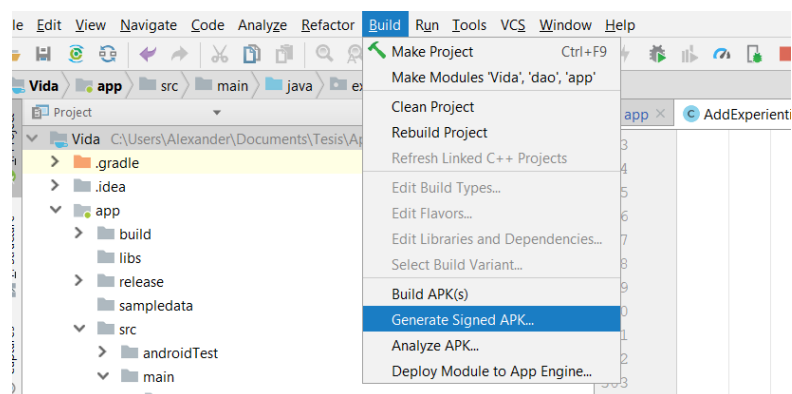


Fig. 49. Generar APK
Fuente: Propia

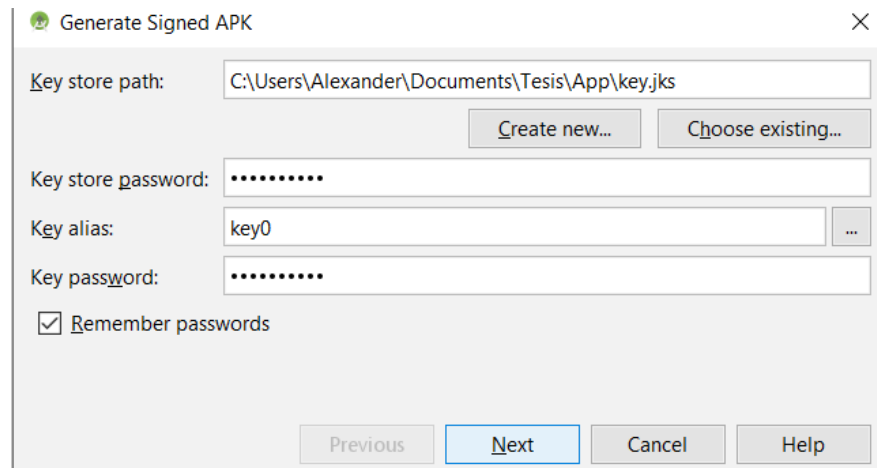


Fig. 50. Generar aplicación firmada
Fuente: Propia

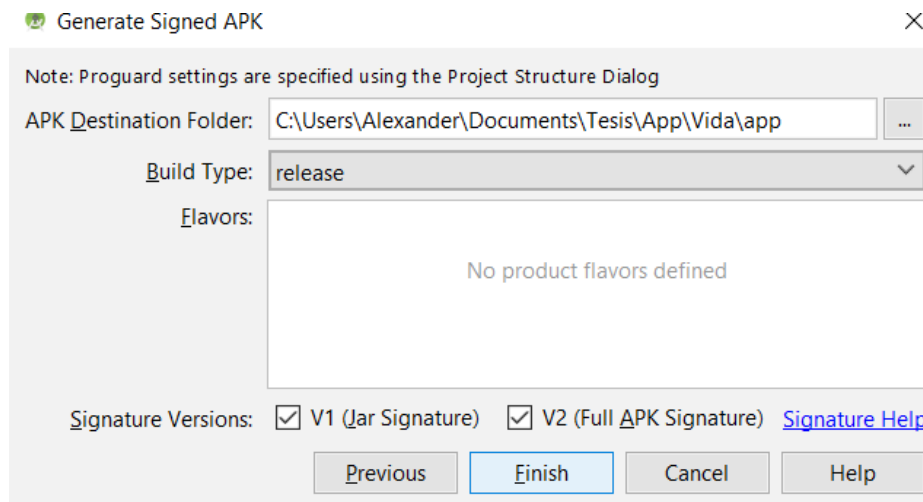


Fig. 51. Seleccionar destino de la aplicación firmada
Fuente: Propia

3.6.2 Publicarlo aplicación en Play Store

Antes de nada, se debe hacer un pago de 25 USD para poder usar la consola de aplicaciones de Play Store.

Este proceso puede encontrado en la página oficial de asistencia de Google:

<https://support.google.com/googleplay/android-developer/answer/113469?hl=es-419>

3.6.3 Inicio de pruebas Alfa.

Aunque se haya agregado una pista de prueba cerrada, en este caso se hará uso desde la pista de pruebas alfa.

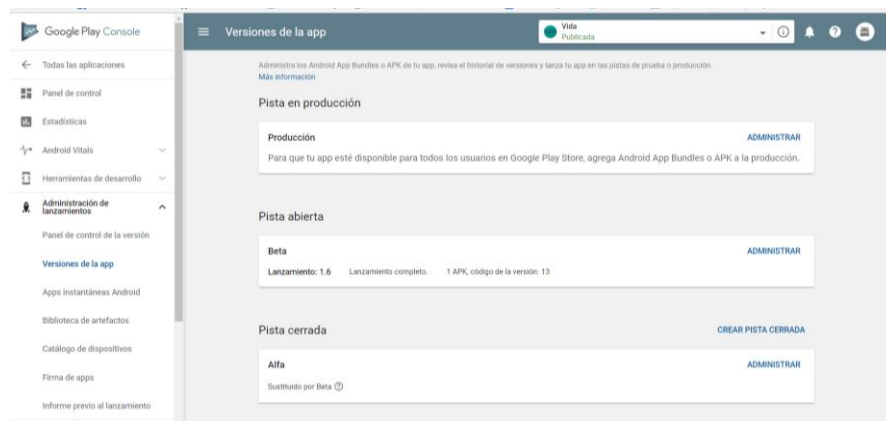


Fig. 52. Versiones de una aplicación.
Fuente: Propia

3.6.3.1 Crear versión alfa

Una aplicación en versión Alfa se lanza para probar el comportamiento de la aplicación en diferentes dispositivos y versiones de Android, las funcionalidades aún pueden no estar desarrolladas al 100%

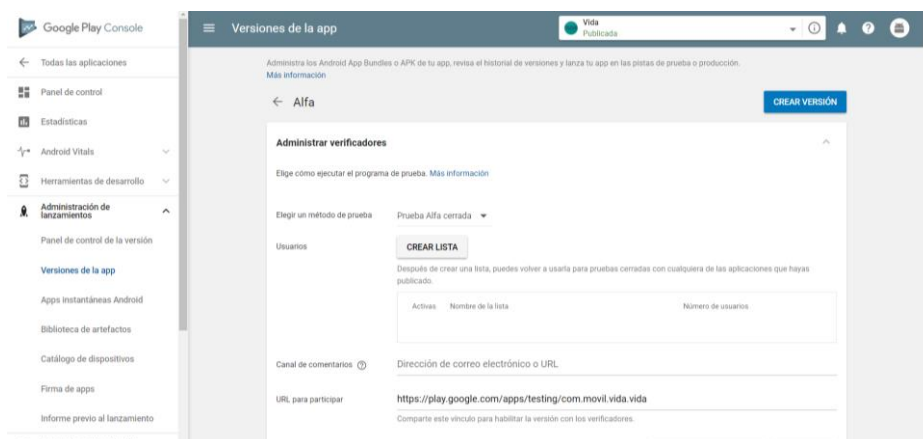


Fig. 53. Creación de versión alfa
Fuente: Propia

Dentro de las pruebas alfa, se crea los usuarios quienes hará los test respectivos en sus dispositivos móviles.

Crear lista de verificadores

Nombre de la lista Usuarios Test Alfa

Direcciones de correo electrónico de verificadores

Separa las direcciones de correo electrónico con cc

alex_251093@hotmail.com	X
marlon.cadena@unacem.com.ec	X
alex2510939394@gmail.com	X

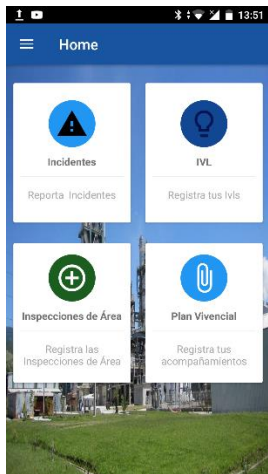
CARGAR UN NUEVO ARCHIVO CSV

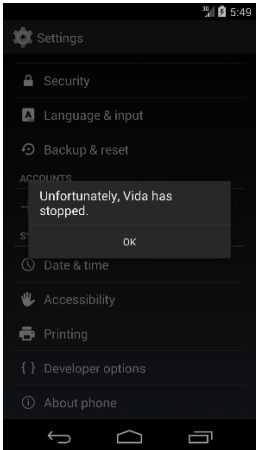

CANCELAR GUARDAR

Fig. 54. Agregar usuarios a pruebas Alfa
Fuente: Propia

3.6.3.2 Resultados prueba alfa.

TABLA 36. Resultado pruebas alfa

Dispositivo	Captura
<p>Xiaomi Mi A1</p> <p>Api 26</p> <p>Android 8.0.0 Oreo</p> <p>Novedades</p> <ul style="list-style-type: none"> Ninguna, todo funciona correctamente 	

<p>Pixel (Emulador)</p> <p>Api 19</p> <p>Android 4.4 Kitkat</p> <p>Novedades</p> <ul style="list-style-type: none"> • Se tuvo que remplazar las funciones donde haga uso de lambda ya que la versión de Android no soporta el JDK 8. • Se cambia la manera de obtener recurso de la aplicación. 	
<p>Dispositivo genérico</p> <p>Api 22</p> <p>Android 5.1</p> <p>Novedades</p> <ul style="list-style-type: none"> – Se tuvo que remplazar las funciones donde haga uso de lambda ya que la versión de Android no soporta el JDK 8. 	

Fuente: Propia

3.6.4 Inicio de pruebas beta

Luego de verificar el comportamiento de aplicaciones en la versión Alfa, se procede a lanzar la versión beta donde la aplicación ya tiene facilidades implementadas y están en espera de ser probadas por los Usuarios testers.

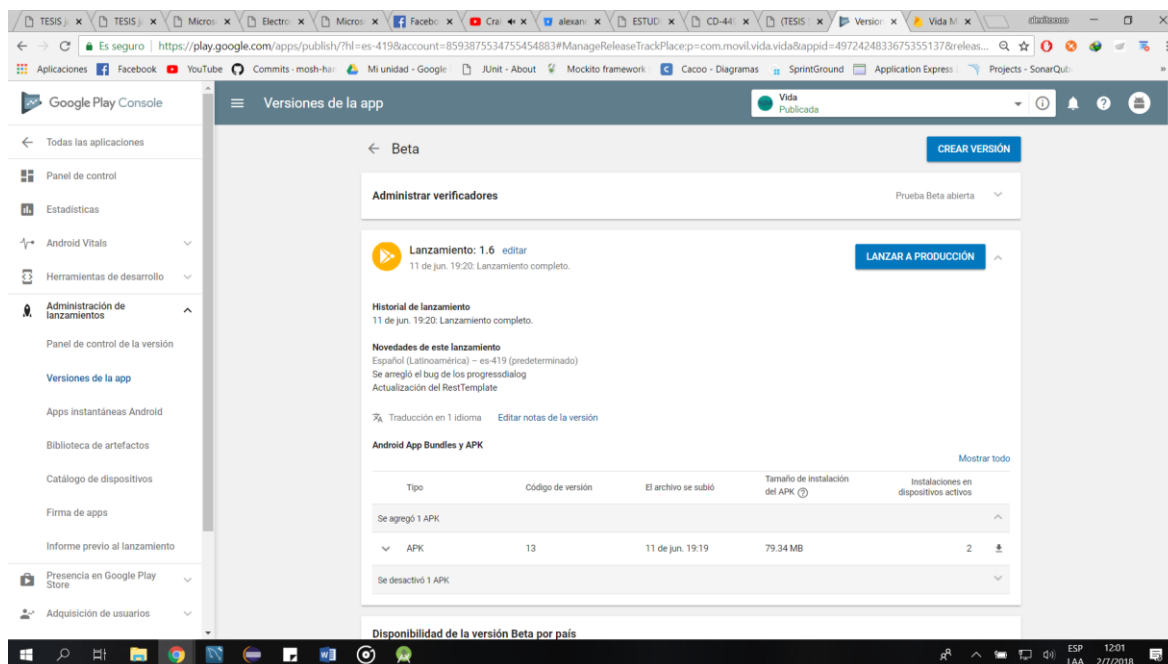

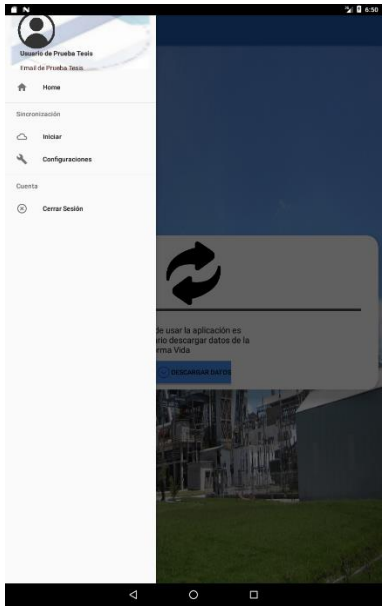


Fig. 55. Panel de control de versiones beta.
Fuente: Propia

3.6.4.1 Resultados de pruebas Beta

TABLA 37. Resultado pruebas beta

Dispositivo	Captura
<p>Xiaomi Mi A1</p> <p>Api 26</p> <p>Android 8.0.0 Oreo</p> <p>Novedades</p> <p>— Ninguna</p>	

<p>Dispositivo Genérico (Emulador)</p> <p>Api 22</p> <p>Android 5.1 Lollipop</p> <p>Novedades</p> <ul style="list-style-type: none"> – Se ha arreglado la compatibilidad con JDK 8 	
<p>Tablet (Emulador)</p> <p>Api 24</p> <p>Android 7.0 Nougat</p> <p>Novedades.</p> <ul style="list-style-type: none"> – Se tiene que crear vistas acordes a la dimensión de la pantalla del dispositivo. 	

Fuente: Propia

3.6.5 Pruebas con Firebase

Firebase es una plataforma de servicios para el desarrollo de aplicaciones web y móviles, este ofrece una herramienta para testear aplicaciones en dispositivos físicos y virtuales.

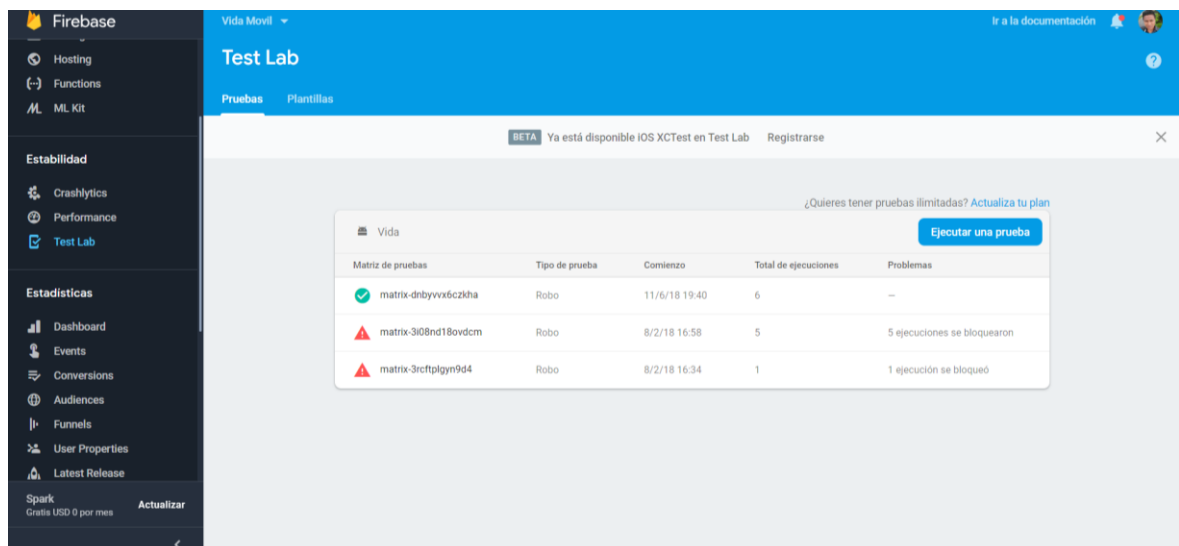


Fig. 56. TestLab firebase
Fuente: Propia

3.6.5.1 Resultados de pruebas con Firebase

Se realizó varias pruebas y correcciones hasta que se pueda concluir exitosamente las pruebas.

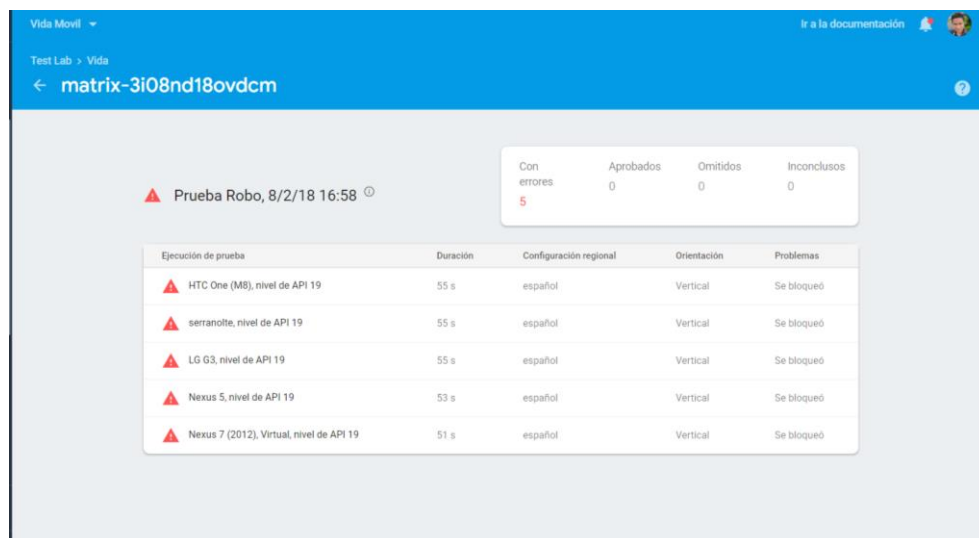


Fig. 57. Pruebas con errores Firebase
Fuente: Propia

Con errores		Aprobados	Omitidos	Inconclusos
0		6	0	0

Ver clústeres de capturas de pantalla →

Ejecución de prueba	Duración	Configuración regional	Orientación	Problemas
✓ Nexus 5X, Virtual, nivel de API 23	5 min 10 s	español (España)	Vertical	—
✓ Nexus 9, Virtual, nivel de API 21	5 min 17 s	español (España)	Vertical	—
✓ Nexus 10, Virtual, nivel de API 19	5 min 13 s	español (España)	Vertical	—
✓ Huawei Mate 9, nivel de API 24	5 min 7 s	español (España)	Vertical	—
✓ Nexus 7 (2012), Virtual, nivel de API 19	5 min 7 s	español (España)	Vertical	—
✓ Samsung Galaxy S9 (US), nivel de API 26	5 min 12 s	español (España)	Vertical	—

Fig. 58. Pruebas exitosas Firebase
Fuente: Propia

3.7 Lanzamiento a producción

Se lanza a producción la versión 1.6 de la aplicación

← Confirmar lanzamiento de versión en producción: 1.6

Preparar versión (✓) Revisión y lanzamiento (2)

Resumen de reseñas
Esta versión está lista para su lanzamiento.

APK de esta versión [Mostrar todo](#)

Tipo	Código de versión	El archivo se subió	Tamaño de instalación del APK	Instalaciones en dispositivos activos
Se agregó 1 APK				
APK	14	2 minutos atrás	79.89 MB	Sin datos

Novedades de este lanzamiento
Español (Latinoamérica) – es-419 (predeterminado)
Se completó el desarrollo general de aplicación 99%
Traducción en 1 idioma

ANTERIOR DESCARTAR **INICIAR LANZAMIENTO DE VERSIÓN EN PRODUCCIÓN**

Fig. 59. Lanzar aplicación a producción.
Fuente: propia

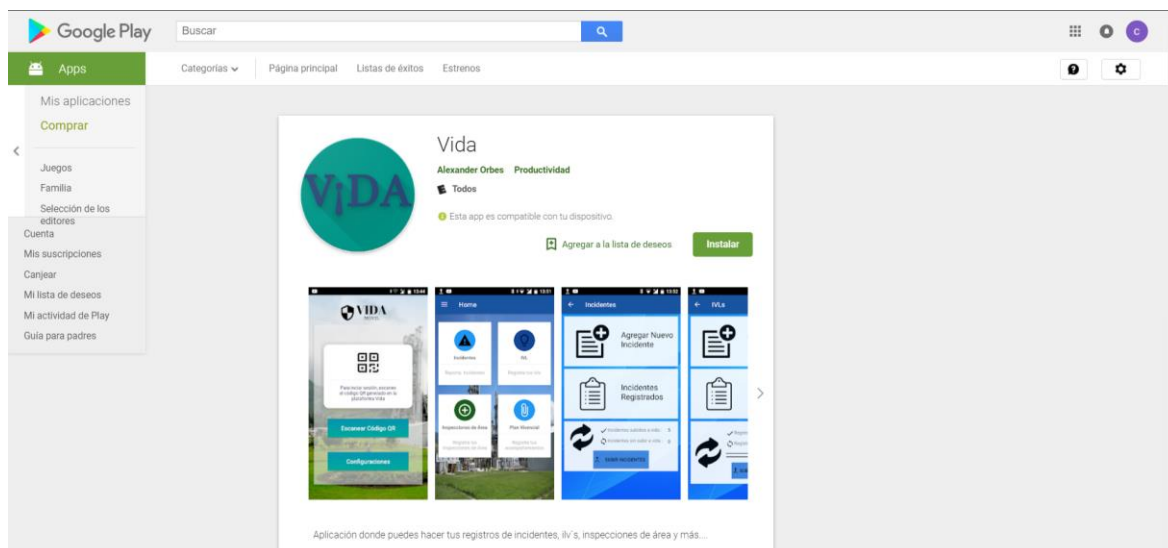


Fig. 60. Aplicación publicada en Play Store
Fuente: Propia

3.8 Fin de pruebas y documentación

TABLA 38. Checklist finalización de desarrollo

Release Final			
Fecha: 01/07/2018 Release:1.7			
Requerimientos	Si	No	N/A
1. Todas las StoryCads han sido verificadas	✓		
2. Todas las tareas han sido revisadas	✓		
3. Han sido resueltas los problemas antes del	✓		
4. En el testing surgió problemas (documentado)	✓		
5. Documentos al día	✓		
6. Se revisó las métricas, logs, duplicación de código o vulnerabilidades	✓		
Comentarios: Se completó las fases de desarrollo y pruebas y se culminó con el lanzamiento de la aplicación a producción.			

Fuente. Propia

CAPITULO IV

4 Impactos, conclusiones y recomendaciones

La implementación de un producto de software en las empresas provoca diversas reacciones en el ambiente laboral, productivo y administrativo y es necesario señalar las más relevantes.

Este análisis se lo va a hacer usando la primera metodología que nos propone el libro Proyectos, Tesis y Marco Lógico: Planes e informes de investigación del Autor Miguel Ángel Posso, en el cual expone dos tipos de análisis de impactos, prospectivos y retrospectivos.

El análisis retrospectivo se hace con datos resultantes del proyecto ejecutado en cambio el análisis prospectivo se lo realiza mediante ponderaciones de que es lo que se espera obtener implementando el proyecto.

Se hará un análisis de impacto prospectivo ya que el aplicativo tiene poco tiempo de implementación y que se espera luego de periodo de funcionamiento. Se usará los niveles de impacto indicados en la siguiente figura.

-3	Impacto alto negativo.
-2	Impacto medio negativo.
-1	Impacto bajo negativo.
0	No hay impacto.
1	Impacto bajo positivo.
2	Impacto medio positivo
3	Impacto alto positivo.

Fig. 62. Niveles de impacto.
Fuente: (Yépez, 2016)

4.1 Impacto operativo

Tabla 39. Impacto operativo

Niveles de impacto Indicador	-3	-2	-1	0	1	2	3
-Efectividad en prevención de accidentes.						✓	
-Participación del personal.							✓
-Portabilidad.							✓
-Usabilidad de la aplicación.						✓	
-Buenas prácticas e ideas de mejora						✓	
Total						6	6
$\sum = 12$ $\text{Nivel de Impacto Operativo} = \frac{\sum}{\text{Numero de indicadores}}$ $\text{NI} = \frac{12}{5} = 2.4$ <p>Nivel de impacto Operativo = Medio positivo.</p>							

Fuente: Propia

4.1.1 Análisis

- Al reportar incidentes los jefes de seguridad pueden tomar acciones preventivas de seguridad así evitando futuros conflictos en la operación diaria.
- Con una aplicación móvil en cada teléfono celular, el personal puede reportar sus inspecciones personales y también se incentiva a la participación continua mediante charlas.
- Las inspecciones de área que se hacen en las minas tienen una mejor precisión en la interpretación de datos, dada la situación geográfica del lugar no existe acceso a internet y, con la aplicación móvil se puede registrar datos precisos que luego serán sincronizados en la plataforma.

- En esta década el uso de teléfonos celulares y aplicaciones móviles hace que las personas desarrollen cierta habilidad al usar una nueva aplicación por lo que no hace falta leer un tutorial o un manual para poder empezar a usar la aplicación.
- La participación continua del personal hace que los empleados puedan exponer sus ideas de mejora a los jefes inmediatos. Además, que también se inculca ideas de mejora dentro de la operatividad de la empresa.

4.2 Impacto administrativo

Tabla 40. Impacto administrativo.

Niveles de impacto Indicador	-3	-2	-1	0	1	2	3
-Gestión integral de la seguridad						✓	
-Fortalecimiento de la cultura de seguridad						✓	
-Asignación de recursos SGI.						✓	
-Incentivación de liderazgo						✓	
-Resultados evidenciables							✓
Total						8	3
$\sum = 11$ $\text{Nivel de Impacto Operativo} = \frac{\sum}{\text{Numero de indicadores}}$ $NI = \frac{11}{5} = 2.2$ <p>Nivel de impacto Operativo = Medio positivo.</p>							

Fuente: Propia

4.2.1 Análisis

- Los gerentes de la empresa apoyan la gran labor que se hace en el sistema de gestión integral de la seguridad, puesto que la aplicación móvil ayuda a mantener y mejorar los estándares de calidad y seguridad que se han implementado
- La aplicación provee de un ambiente participativo en el personal porque gracias a él tienen el medio para fortalecer la cultura de seguridad ocupacional, refiriéndose a los reportes de incidentes.

- Los gerentes asignan recursos a los departamentos siempre y cuando hay una necesidad evidenciable, con la aplicación móvil la recolección de datos es un gran apoyo para la hora de reportar el desempeño del departamento de seguridad industrial y de la misma manera lograr mayor asignación de presupuesto para mejorar la gestión de la seguridad integral.
- Las personas que participan activamente pueden ser las que lideren equipos y logren la ejecución de una acción de mejora en la cultura de seguridad ocupacional dentro de la empresa.

4.3 Impacto ecológico

Tabla 41. Impacto ecológico

Indicador	Niveles de impacto						
	-3	-2	-1	0	1	2	3
-Gestión de la contaminación por polvo						✓	
-Reducción del uso de papel							✓
-Buenas prácticas en conducción						✓	
-Tratamiento de desechos						✓	
-Compromiso con el medio ambiente					✓		
Total					1	6	3
$\sum = 10$ <p>Nivel de Impacto Operativo = $\frac{\sum}{\text{Numero de indicadores}}$</p> <p>$NI = \frac{10}{5} = 2$</p> <p>Nivel de impacto Operativo = Medio positivo.</p>							

Fuente: Propia

4.3.1 Análisis

- UNACEM se cataloga como una empresa de alto riesgo para el ambiente, es por eso por lo que trabaja continuamente para realizar las labores con el menor impacto ambiental, de esta forma, la aplicación móvil permite la toma de datos puntuales en lugares donde los trabajos pueden llegar a desencadenar un perjuicio ambiental con es la polución por polvo.
- Se reduce el uso de papel en gran medida ya que se evita usar registros físicos en papel en campo.
- Los conductores de flotas hacen mejor uso del combustible y derivados, al aplicarse el plan vivencial el encargado inspecciona su uso efectivo.
- El tratamiento de residuos tiene un impacto negativo en comunidades aledañas las cuales se quejan continuamente de desechos en las vías públicas. Se propuso a tratar el asunto registrando evidencias y organizando reuniones con los dirigentes.
- La participación constante del personal apoyado con charlas y herramientas como la aplicación móvil hace que las personas se involucren en el cuidado de los espacios de trabajo y del medio ambiente.

4.4 Conclusiones y recomendaciones

4.5 Conclusiones

- Con la implementación del aplicativo móvil los trabajadores tienen una participación más activa y responsable con la seguridad y salud empresarial ya pueden reportar o comunicar incidentes, en el lugar del hecho, dando como resultado la disminución de incidentes graves que involucre la salud de los trabajadores.
- En el ámbito de desarrollo de aplicaciones, existen una gran variedad de herramientas y lenguajes de programación, por lo cual escoger una en la que se tenga experiencia siempre es la mejor opción para desarrollar software, actualmente Java representa una gran ventaja para los desarrolladores, puesto que ya muchos son afines de este lenguaje en su desarrollo del día a día.
- En el mercado hay una infinidad de dispositivos móviles funcionando con el sistema operativo de Android, por lo cual, las aplicaciones para estos deben cubrir la mayoría de sus versiones, pero se debe enfatizar en el desarrollo de aplicaciones para las versiones más reciente, ya que estas ofrecen un mejor rendimiento y seguridad para el usuario final, de esta manera también se promueve a que los usuarios tengan sus equipos actualizados.
- Antes de iniciar el desarrollo de aplicaciones para Android se debe analizar las diferentes herramientas de desarrollo sean estas híbridas, multiplataforma o nativas. Cada elección tiene sus ventajas y desventajas.
- El testeo de aplicaciones para Android se hace de mejor manera usando las herramientas de Play Store; lanzando versiones alfa o beta con grupos de usuarios definidos en dispositivos reales, de esta manera se puede detectar errores y corregirlos a tiempo.

4.6 Recomendaciones

- Se recomienda que los usuarios sean capacitados antes de comenzar con el uso de la aplicación móvil, con el fin de optimizar el tiempo de manipulación del aplicativo, así como también, para evitar registros incorrectos con la información.
- Es recomendable que los usuarios dispongan de un dispositivo móvil con el sistema operativo móvil Android compatible con la aplicación, ya que esto les permitirá tener un mayor desempeño y fluidez.
- Se sugiere que los dispositivos móviles dispongan de una conexión a datos móviles, para que, a través de estos, la información ingresada sea sincronizada en ese momento, o en el caso de encontrarse en una zona alejada en donde la cobertura tenga un difícil acceso, la información captada sea registrada en cuanto la cobertura sea estable, con el fin de los datos sean registrados lo más pronto posible.
- El uso de metodologías de desarrollo permite organizar el ambiente y equipos de desarrollo y durante el desarrollo de aplicaciones móviles la metodología Mobile-D ofrece los procesos más adecuados tanto para equipos de desarrollo de aplicaciones como para dispositivos móviles, optimizando el trabajo del desarrollador.
- Para el desarrollo de aplicaciones Android, en donde se necesite hacer uso de una base de datos local, es recomendable usar una API de persistencia, dado que el desarrollo se agiliza.
- Es recomendable desarrollar aplicaciones Android con herramientas nativas cuando el usuario final busca mayor experiencia de usuario y rendimiento.
- La integración de aplicaciones móviles y servidores remotos se debe usar tecnologías de comunicación como RestFull o las tecnologías más actuales, ya que el envío de información se lo hace de manera ágil usando el protocolo HTTP.

5 Bibliografía

“<http://www.unacem.com.ec/>,” n.d. (s.f.). <http://www.unacem.com.ec/>.

Agile. (s.f.). *Agile*. Obtenido de MOBILE-D: <http://agile.vtt.fi/mobiled.html>

Álvarez Caules, C. (2016). *Arquitectura Java*. Obtenido de ¿Qué es Gradle?:
<https://www.arquitecturajava.com/que-es-gradle/>

Amaya Balaguera, Y. D. (2013). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual. *Revista de Tecnología / Journal Technology*, 12(2), 111-124. Obtenido de http://m.uelbosque.edu.co/sites/default/files/publicaciones/revistas/revista_tecnologia/volumen12_numero2/12Articulo_Rev-Tec-Num-2.pdf

Android Studio Faqs. (2016). *Android Studio Faqs*. Obtenido de ¿Qué es gradle en Android Studio? [AndroidStudioFAQs](https://androidstudiofaqs.com/conceptos/que-es-gradle-en-android-studio) te lo cuenta:
<https://androidstudiofaqs.com/conceptos/que-es-gradle-en-android-studio>

Báez, M., Borrego, A., Cordero, J., & Cruz, L. (2012). *Introducción a android*. Obtenido de <http://ceavitoria.com.es/Sergio/Android/Apuntes/android.pdf>

Bui Duy, T. (2017). *Reactive Programming and Clean Architecture in Android Development*. Finlandia. Obtenido de <https://www.theseus.fi/bitstream/handle/10024/126982/Thesis2016-Sunshine.pdf?sequence=1>

C, T. D. C. D. T. C. (2015). *PRODUCTIVIDAD (MIPRO) PARA INVESTIGAR LA DINÁMICA DE LA INDUSTRIA CEMENTERA EN EL ECUADOR Y SU PROYECCIÓN A 20 AÑOS*.

CAVERO DORIA, M. F. (2017). *PROPUESTA DE MEJORA DE SEGURIDAD Y*. Obtenido de

http://repositorioacademico.upc.edu.pe/upc/bitstream/10757/621566/5/CAVERO_DM.pdf

Comparación de sistemas de comunicación remota para transferir información de un registrador de fallas. (2008).

Conoce Android Studio. (s.f.). *Android Studio*. Obtenido de Conoce Android Studio: <https://developer.android.com/studio/intro/index.html>

Dar, S. A., & Iqra, J. (2016). *Synchronization of Data Between SQLite (Local Database) and SQL Server (Remote Database)*, 7.

Del, C., & General, G. (s.f.). *Informe de Responsabilidad Corporativa Lafarge*.

Developer Android. (2016). *Developers*. Obtenido de Transferring Data Using Sync Adapters: <https://developer.android.com/training/sync-adapters/index.html>

Dobjanschi, V. (2010). *Developing Android REST client applications. Google I/O 2010*. Obtenido de <https://dl.google.com/googleio/2010/android-developing-RESTful-android-apps.pdf>

Escobar Bonilla, G. S., & Campaña Guzmán, A. I. (Abril de 2014). *Repositorio Dspace*. Obtenido de Diseño e implementación de una aplicación móvil que cumpla la función de Estación en Tierra para el monitoreo de UAV'S en el Centro de Investigación y Desarrollo de la Fuerza Aérea Ecuatoriana: <http://repositorio.espe.edu.ec/xmlui/handle/21000/8189>

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1996). *Design Patterns: Elements of Reusable Software. Addison-Wesley Professional Computing Series*. doi:<https://sophia.javeriana.edu.co/~cbustaca/docencia/DSBP-2016-03/recursos/Erich%20Gamma,%20Richard%20Helm,%20Ralph%20Johnson,%20John%20M.%20Vlissides->

- Design%20Patterns_%20Elements%20of%20Reusable%20Object-Oriented%20Software%20%20-Addison-Wesley%20Professi
- Gironés, J. T. (2016). *El gran libro de android 5ta edicion*. Madrid: Marcombo.
- González Maestre, D. (2015). *Ergonomía y psicosociología*.
- GreenRobot. (2016). *Green Robot*. Obtenido de greenDAO: Android ORM for your SQLite database: <http://greenrobot.org/greendao/>
- Imaginanet. (2012). *Imaginanet.com*. Obtenido de Patrón MVP: <https://www.imaginanet.com/blog/patron-mvp.html>
- Instituto Politécnico Nacional. (s.f.).
- Izquierdo, A.N.E. (2015). *Responsabilidad Social y Empresarial como estrategia de comunicación de Unión Cementera Nacional UCEM*.
- Joyanes Aguilar, L. (2012). *Fundamentos de programación : algoritmos y estructura de datos y objetos*. Madrid, España: McGraw-Hill.
- Juneau, J. (2013). *Object-Relational Mapping*. Obtenido de https://link.springer.com/chapter/10.1007/978-1-4302-4426-4_8
- Kreibich, J. A. (2010). *Using SQLite (First)*. O'Reilly.
- Lee, W. M. (2012). *Android 4 desarrollo de aplicaciones*.
- Lizando, W., & Gallegos, A. (2012). *OCUPACIONAL Y LA SEGURIDAD INDUSTRIAL HISTORICAL REVIEW ABOUT OCCUPATIONAL HEALTH*.
- Luján Castillo, J. D. (2015). *Android: Aprende desde cero a crear aplicaciones*.
- Narváez, M., & D. E. (2012). *Desarrollo de una aplicación para dispositivos móviles con sistema operativo android aplicando extreme programming*. Obtenido de <http://dspace.udla.edu.ec/handle/33000/2597>
- Razynskas Sosa, M. A. (2015). *Manejo de emergencias con materiales peligrosos*.
- Reporte de sostenibilidad 2016*. (2016).

Ribas Lequerica, J. (2016). *Desarrollo de aplicaciones para android*. Madrid: Anaya Multimedia.

Unacem Historia. (s.f.).

ANEXOS

ANEXOS

Anexo A. Manual de usuario (en CD)