

Maintenance Management – Lift IL App Android

Gestión de mantenimiento - Lift IL App Android

Nombre del estudiante

Oscar Mauricio Pastás Bastidas

Master en Desarrollo de Aplicaciones para Dispositivos móviles
Android (Desarrollo web front-end/app) – Drupal (Desarrollo web backend)

Nombre Consultor/a

Eduard Martin Lineros

Nombre Profesor/a responsable de la asignatura

Carles Garrigues Olivella

Junio-2019

Créditos / Copyright



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

© **Oscar Pastás**

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Gestión de Mantenimiento – Lift</i>
Nombre del autor:	<i>Oscar Pastás Bastidas</i>
Nombre del consultor/a:	<i>Eduard Martin Lineros</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	06/2019
Titulación::	<i>Plan de estudios del estudiante</i>
Área del Trabajo Final:	<i>Android – Desarrollo Web</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Lift, Mantenimiento de equipos de transporte vertical, inspección, ascensor, elevador</i>
Resumen del Trabajo:	
<p>Lift Ingenieria S.A. es una empresa del sector de la construcción dedicada a la venta y mantenimiento de equipos de transporte vertical. Con presupuesto limitado para su base tecnológica, cuenta con un dominio (web) disponible y personal con dispositivos móviles.</p> <p>Mediante el estudio de la necesidad, viabilidad económica, entrevistas/encuestas se determinarán los requerimientos necesarios tanto en desarrollo tecnológico, como en la capacitación. Esto permitirá mejorar la gestión de mantenimiento aplicando la sistematización y análisis de los reportes (actualmente son escritos).</p> <p>Las herramientas, asignaturas y metodologías vistas a lo largo del Master como: diseño centrado en usuario, modelo vista controlador, diseño de aplicaciones móviles en Android, bases de datos, programación de servidores web, serán empleadas para dar una solución adecuada al requerimiento definido como objetivo principal.</p> <p>Luego del proceso de revisión, se procede a plantear el diseño de la app empleando Android Studio (java), como CMS se elige Drupal. Por otro lado, el manejo de datos se hace mediante los tipos de contenido definidos. El servidor web http es Apache y lenguaje de programación empleado es PHP v.7.</p> <p>Los resultados obtenidos son demostrados mediante un video de capacitación registrado en la entrega. El personal de Lift Ingeniería S.A., fue capacitado tanto en el uso de Drupal, como a la aplicación en Android. Los temas de capacitación fueron: Creación de máquinas, usuarios, reportes y envío.</p> <p>La arquitectura de solución propuesta permite conexión de múltiples plataformas y sesiones, siendo una alternativa adecuada para escalar el sistema, cumpliendo con los diferentes objetivos planteados.</p>	

Abstract:

Lift Ingenieria S.A. is a construction company dedicated to the sale and maintenance of vertical transport equipment. With a limited budget for its technological base, it has an available web host and personnel with knowledge in app for mobile devices.

Studying needs, economic viability, interviews/surveys the requirements will be determined in technological development and training. Systematization and report analysis are required since current reports are being written in papers.

Methodologies, tools and subjects taken in Master's program, such as: user centered design, MVC-Model-View-Controller, databases, web Server programming, will be used to provide an integral solution to the requirement defined as the main objective.

After the review stage, app design is proposed with Android Studio (Java), Drupal is chosen as CMS. On the other hand, data handling is done using content types (Drupal). Http web server is Apache and PHP is the programming language used.

The evidence of results are demonstrated in training video recorded in Lift Ingenieria S. A, where staff practiced in basic usage of Drupal CMS and Android app. Training topics were: machine records and creation, user creation, reports. The proposed architecture allows connection of multiple platforms and sessions, giving to the system scalability and fulfilling whole specific and main objectives.

Índice

1.	Introducción	1
1.1	Contexto y justificación del Trabajo	1
1.2	Objetivos del Trabajo.....	8
1.2.1	Objetivo General	8
1.2.2	Objetivos Específicos.....	8
1.3	Enfoque y método seguido.....	8
1.4	Planificación del Trabajo	11
1.5	Breve resumen de productos obtenidos	14
2.	Diseño y Análisis.....	15
2.1	Breves criterios revisados para el diseño centrado en usuario de la app	15
2.2	Necesidades detectadas para la app:	15
2.3.	Enfoque del producto y/o servicio	15
2.4.	Requisitos de usuario (Fichas realizadas a partir de la observación de usuarios en la empresa).....	16
2.5.	Puntos clave de la app (A partir de encuestas y entrevistas realizadas)	16
2.6.	Procedencia de público objetivo.....	16
2.7.	Frase descriptiva de la aplicación	17
2.8.	Contexto de uso	17
3.	Diseño de wireframes y esquema de navegación.....	18
3.1.	Wireframes y bocetos de app.....	18
3.2.	Esquema de navegación	24
3.3.	Criterios generales de diseño en wireframes y navegación	25
3.4.	Diseño de alta definición	25
3.5.	Test con usuarios – Usabilidad – Evaluación:	26
3.6.	ARBOL DE NAVEGACIÓN USUARIO TIPO ADMIN:	27
3.7.	ARBOL DE NAVEGACIÓN USUARIO TIPO NORMAL:.....	28
4.	Base de datos	29
4.1.	Casos de uso	29
4.2.	Actores y guiones.....	29
4.2.1.	Identificación de actores	29
4.2.2.	Guiones.....	30
4.3.	Modelo Entidad – Relación.....	30
4.4.	Modelo Relacional de base de datos.....	31
4.4.	Implementación de base de datos.....	31
5.	Arquitectura de sistema	33
6.	Valoración económica preliminar	34
7.	Implementación de base de datos y entorno de desarrollo en Drupal - servidor	35
7.1.	Implementación de base de datos en Dreamhost:	35
7.2.	Instalación de Drupal.....	36
7.3.	Herramienta JSON – API de Drupal:.....	36
7.4.	Creación de tablas y administración de contenido	37
7.4.1.	Tabla reporte técnico:	38
7.4.2.	Tabla de usuarios:	39
7.4.3	Tabla Máquina:	39

7.4.4. Tabla de sitios:.....	40
7.5. Contenido de prueba para la aplicación	41
7.6. Servicios REST y JSON API	42
7.7. Herramienta Postman:.....	43
8. Implementación de aplicación en Android Studio	45
7.1. Configuración de Android Studio:.....	45
7.2. Pantalla de ingreso.....	45
7.2. Pantalla de menú principal	47
7.3. Pantalla de recursos.....	47
7.4. Pantalla de Crear Reporte.....	48
7.5. Pantalla de emergencias	48
7.6. Pantalla de Historial.....	49
7.7. Icono de Ingreso.....	50
7.8. Uso de librería Retrofit.....	50
9. Conclusiones	51
10. Glosario	53
11. Bibliografía.....	54
12. Anexos.....	59
A. PERFILES DE USUARIO:	59
B. ENTREVISTA, ENCUESTA Y RESULTADOS	61
B. 1. FICHA ENCUESTA APLICADA Y RESULTADOS:	61
B. 2. FICHA ENTREVISTA APLICADA Y RESULTADOS:.....	65

Lista de figuras

Ilustración 1. Ejemplo de actividades de inspección. Fuente: Propia	1
Ilustración 2. Ejemplo de actividades de mantenimiento preventivo. Fuente: Propia	2
Ilustración 3. Ejemplo de actividades de mantenimiento correctivo. Fuente: Propia	2
Ilustración 4. Aplicación Lift Maintenance	4
Ilustración 5. Aplicación Happy Service	5
Ilustración 6. Aplicación App Sheet	5
Ilustración 7. Aplicación Easy Maintenance Reminder	6
Ilustración 8. Aplicación Entire Maintenance Solutions	6
Ilustración 9. Iteración en DCU. Fuente: Propia	9
Ilustración 10. Modelo Vista Controlador (MVC). Fuente: Propia	10
Ilustración 11. Esquema – metodología – Plan de trabajo. Fuente: Propia	10
Ilustración 12. Menú Principal – Usuario Administrador – Prototipo de alta definición. Fuente: Propia.....	26
Ilustración 13. Árbol de navegación – Usuario Administrador. Fuente: Propia.	27
Ilustración 14. Árbol de navegación – Usuario Normal. Fuente: Propia	28
Ilustración 15. Casos de uso. Fuente: Propia	29
Ilustración 16. Diagrama de entidad-relación. Fuente: Propia.....	30
Ilustración 17. Diagrama de base de datos relacional y convenciones. Fuente: Propia	31
Ilustración 18. Base de datos creada en Firebase. Fuente: Propia	31
Ilustración 19. Base de datos creada en dreamhost web. Fuente: Propia	32
Ilustración 20. Arquitectura de sistema. Fuente: Propia	33
Ilustración 21. Edición de usuario y configuración de acceso	35
Ilustración 22. Usuario y selección de herramientas, configuración de dominio.	36
Ilustración 23. Estructura definida a partir de tipos de contenido	37
Ilustración 24. Tablas creadas dentro del administrador de contenido.....	37
Ilustración 25. Entidades creadas en table. Consultadas a través de: Inicio – Administración – Informes.	38
Ilustración 26. Tabla de Reporte técnico – campos configurados	38
Ilustración 27. Ejemplo de creación de reporte técnico	39
Ilustración 28. Usuarios – Configuración y creación.....	39
Ilustración 29. Configuración de campos en tabla máquina	40
Ilustración 30. Ejemplo de creación de máquina	40
Ilustración 31. Creación y configuración de campos en la tabla Sitio.....	41
Ilustración 32. Ejemplo de sitio creado	41
Ilustración 33. Ejemplos de creación de reportes, sitios y máquinas	42
Ilustración 34. JSON API.....	42
Ilustración 35. Habilitación de servicios REST	43
Ilustración 36. Petición GET en Postman.....	43
Ilustración 37. Simulación de operaciones GET, POST, etc. en Postman	44

Ilustración 38. Configuración de Android Studio – simulación con Firebase Auth y RealtimeDatabase.....	45
Ilustración 39. Configuración de Firebase – Authentication con usuario opastas@uoc.edu	46
Ilustración 40. Pantalla de ingreso	46
Ilustración 41. Pantalla de Menú Principal	47
Ilustración 42. Pantalla de recursos	47
Ilustración 43. Pantalla de crear reporte.....	48
Ilustración 44. Pantalla de emergencias.....	49
Ilustración 45. Pantalla de historial.....	49
Ilustración 46. Perfil 1. Fuente: Propia	59
Ilustración 47. Perfil 2. Fuente: Propia	60
Ilustración 48. Perfil 3. Fuente: Propia	60

Lista de tablas

Tabla 1. Distribución de personal técnico - operativo – Lift IL	3
Tabla 2. Dispositivos móviles disponibles para personal técnico - operativo – Lift IL	3
Tabla 3. Resumen de ventajas y desventajas de otras apps de mantenimiento:	8
Tabla 4. Resumen de horas de Trabajo por etapa	12
Tabla 5. Lista de tareas/actividades del TFM	14
Tabla 6. Costos actuales de operación – Lift IL.....	34
Tabla 7. Costos de desarrollo de Lift IL App.....	34
Tabla 8. Tiempo de retorno de inversión aproximado	34

1. Introducción

1.1 Contexto y justificación del Trabajo

Existen diferentes apps de gestión de mantenimiento como recurso informático, pero no se encuentra una opción adecuada especialmente para la empresa Lift Ingeniería Ltda. (en adelante Lift IL), personalizada a sus necesidades.

La empresa Lift IL se dedica a la venta y mantenimiento de equipos de transporte vertical tales como: rampas eléctricas, elevadores/ascensores. Dentro de su labor de mantenimiento existen varias modalidades:

- “Inspección: Corresponde a una tarea o actividad de verificación de condiciones de operación del equipo empleando los sentidos y/o instrumentos de medida (observar tensión de alimentación en indicadores, escuchar ruidos, observar tensión mecánica de cable, verificar experiencia de usuario en la máquina, medir corriente y tensión en el motor). Las inspecciones son actividades programadas.

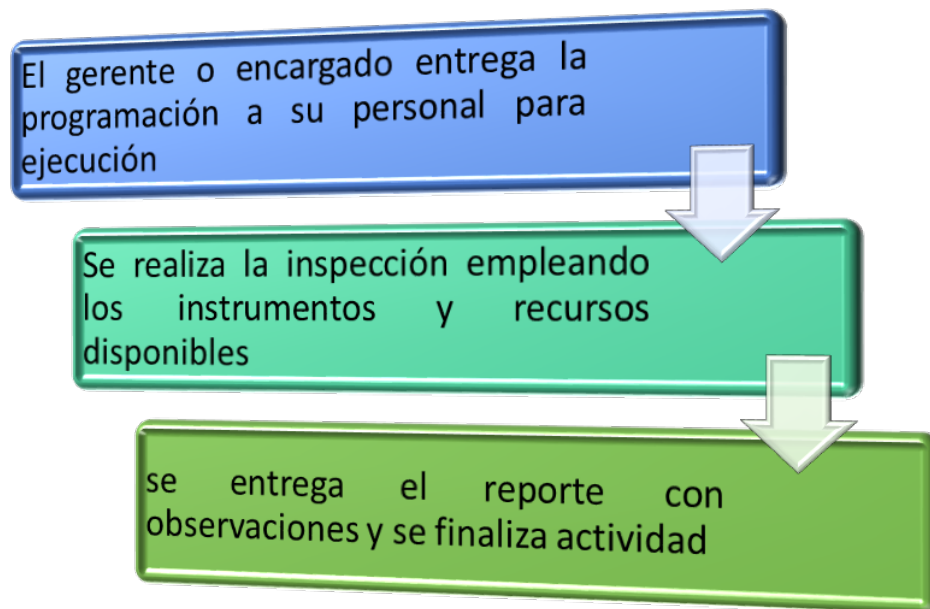


Ilustración 1. Ejemplo de actividades de inspección. Fuente: Propia

- Mantenimiento preventivo: A través de una serie de actividades programadas con personal técnico, herramientas y materiales se evita que el equipo pierda su función principal, evitando las consecuencias asociadas.”¹ Por ejemplo, la consecuencia de una parada no programada en un día de gran afluencia de público en un centro comercial sería catalogada como muy grave, por lo tanto es conveniente

¹ Extraído y traducido de: **Redding, L.; Roy R.** (2015), Through-life Engineering Services, ISBN: 978-3-319-12110-9, Springer.

programar una serie de paradas (de mantenimiento preventivo) para revisar el estado del equipo en días donde la consecuencia sea baja.

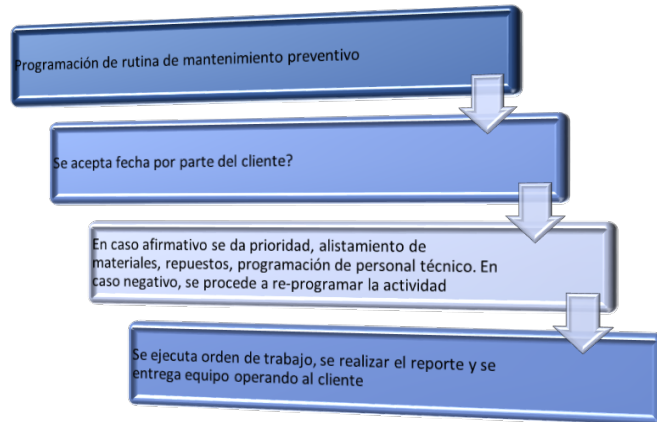


Ilustración 2. Ejemplo de actividades de mantenimiento preventivo. Fuente: Propia

- **Mantenimiento correctivo:** Se lleva a cabo, una vez que el equipo ha perdido su función. Si un ascensor tiene una falla, por pérdida de fluido eléctrico, o alguna de sus partes mecánicas o sistema eléctrico o electrónico sufre daño, el personal técnico debe corregir la falla después de recibir un llamado de emergencia.

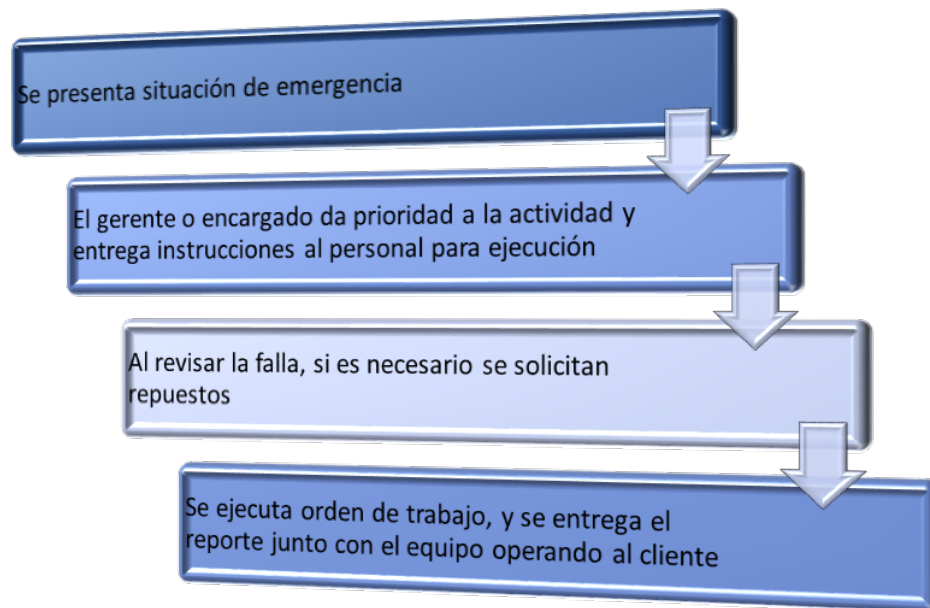


Ilustración 3. Ejemplo de actividades de mantenimiento correctivo. Fuente: Propia

Los requerimientos a cubrir en el área de mantenimiento que se identifican de acuerdo a una evaluación preliminar, son (obtenidos a partir de una entrevista al personal técnico y gerente operativo):

“Optimizar la gestión del mantenimiento de la empresa Lift IL, diseñando un sistema de planificación, recolección de datos de mantenimiento teniendo una

base de datos de los equipos elevadores activos instalados en una jerarquía definida. Se pretende mostrar datos estadísticos en diagramas de Pareto con ciertos indicadores definidos por el personal de mantenimiento“

La empresa se desempeña en el sector de construcción, realizando labores de mantenimiento y puesta en marcha de equipos de transporte vertical.

Los recursos humanos (personal técnico) y su estado (disponible, en inspección, en mantenimiento), serán modificados en una base de datos alojados en un servidor, y serán visualizados a través de la app.

El gerente de mantenimiento tendrá una aplicación instalada en su dispositivo para poder visualizar el estado de sus recursos humanos, la tarea finalizada y la orden de trabajo ejecutada con su respectivo reporte.

El sistema requiere presentar los técnicos disponibles para el trabajo, mostrando su estado actual, a través de una herramienta o aplicación. El sistema desarrollado permitirá enviar un reporte vía web, al finalizar el trabajo, que a su vez podrá ser enviado por el gerente al cliente, dejando constancia de la atención.

Actualmente dentro de la compañía, los técnicos y el gerente de mantenimiento tienen equipos móviles (sistema operativo Android y plan de datos móviles). La distribución de personal es la siguiente:

ZONA	NÚMERO DE TÉCNICOS/PERSONAL
NORTE	3
SUR	3
TOTAL RECURSO HUMANO DISPONIBLE	6

Tabla 1. Distribución de personal técnico - operativo – Lift IL

El personal técnico cuenta con los siguientes dispositivos móviles:

EQUIPOS DISPONIBLES PARA PRUEBAS	CANTIDAD
MOTO G4	1
SAMSUNG GALAXY A1	6
HUAWEI P10	1
TOTAL	8

Tabla 2. Dispositivos móviles disponibles para personal técnico - operativo – Lift IL

La constancia o certificación de los trabajos realizados actualmente se maneja a través de reportes escritos entregados al cliente al finalizar la labor/trabajo. Una vez finalizada la app se termine, se ahorrará papel, se mejorará la forma de utilizar los recursos disponibles (móviles y plan) y se permitirá llevar

estadísticas que fácilmente pueden ser revisadas por el gerente de mantenimiento.

El objetivo de Lift IL será mejorar sus indicadores de disponibilidad, y por supuesto conservar una buena relación disponibilidad/costos mínimos de mantenimiento. La aplicación busca mejorar el uso de los recursos actuales y en un futuro puede llevar indicadores que permitan realizar seguimiento estadístico a los equipos y mediante una investigación formal, un pronóstico o recomendación del tipo de mantenimiento requerido junto con su frecuencia.²

Lift IL presta servicios de mantenimiento a los siguientes equipos ubicados dentro de Bogotá D. C. – Colombia:

Existen otras aplicaciones en el mercado que serán analizadas con el fin de implementar una alternativa competitiva en Lift IL:

- Lift Maintenance³

Es una aplicación para supervisores, técnicos e ingenieros de servicio, que realizan inspección de diferentes partes y parámetros de elevadores. El personal puede enviar reportes y permitir revisar la localización del dispositivo.

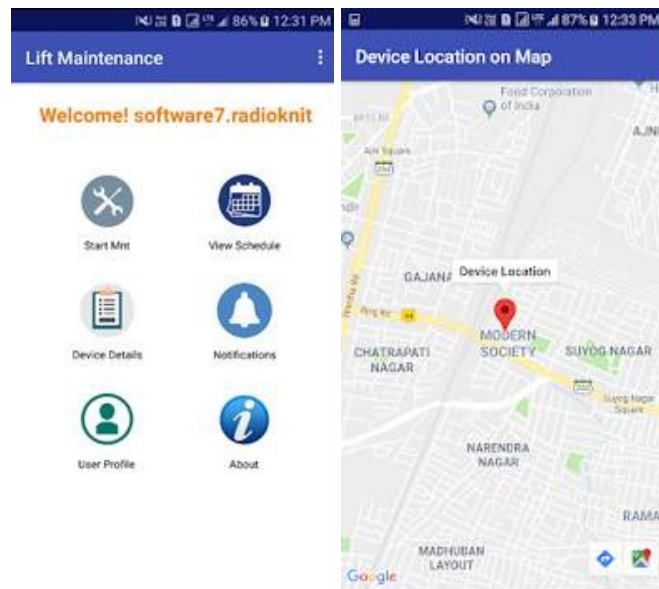


Ilustración 4. Aplicación Lift Maintenance

- Happy Service⁴

² Traducido y adaptado de: **DHILLON B. (2002)**, Engineering Maintenance – a modern approach (2da edición), ISBN: 978-1-4200-3184-3, CRC Press.

³ Ref.: Lift Maintenance (2019) [en línea]. [Fecha de consulta: 22 de febrero de 2019] <<https://play.google.com/store/apps/details?id=com.radioknit.liftmaintenance>>

⁴ Ref.: Happy Service (2019) [en línea]. [Fecha de consulta: 22 de febrero de 2019] <<https://play.google.com/store/apps/details?id=com.happyservice>>

Es una aplicación para supervisores, técnicos e ingenieros de servicio, que realizan inspección de diferentes partes y parámetros de elevadores. El personal puede enviar reportes y permitir revisar la localización del dispositivo.

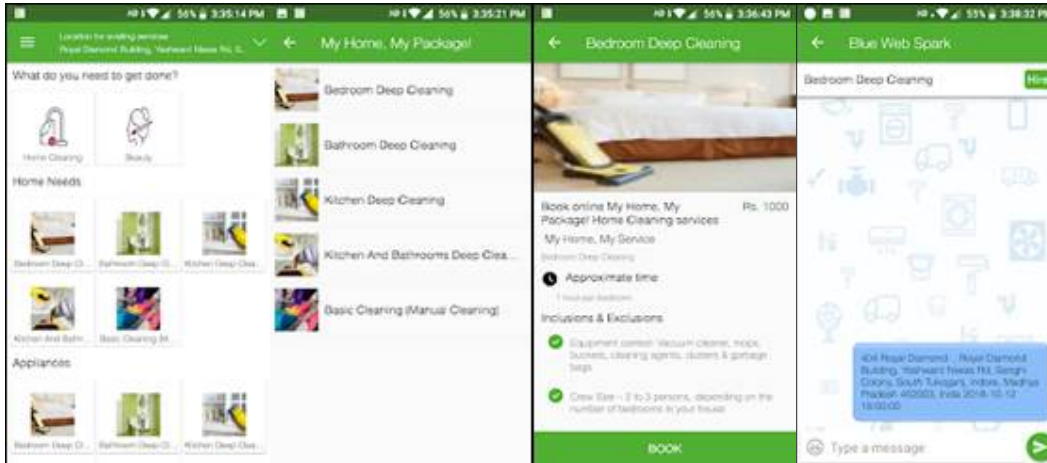


Ilustración 5. Aplicación Happy Service

- AppSheet⁵

Esta plataforma permite crear apps a partir de datos e integraciones con API, por ejemplo guardar datos en las hojas de cálculo de GoogleDrive, mediante una interfaz gráfica diseñada.

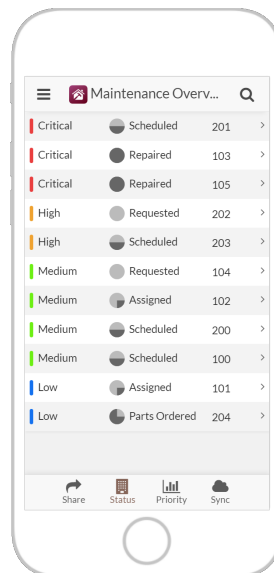


Ilustración 6. Aplicación App Sheet

- Easy Maintenance Reminder⁶

⁵ Ref.: AppSheet (2019) [en línea]. [Fecha de consulta: 22 de febrero de 2019] <<https://www.appsheet.com/>>

A través de esta app se pueden crear diferentes categorías de elementos y recordatorios de tareas (con cierta frecuencia, por ejemplo mantenimiento cada 6 meses del filtro de un vehículo, o cambio de aceite de un motor cada 3 meses).

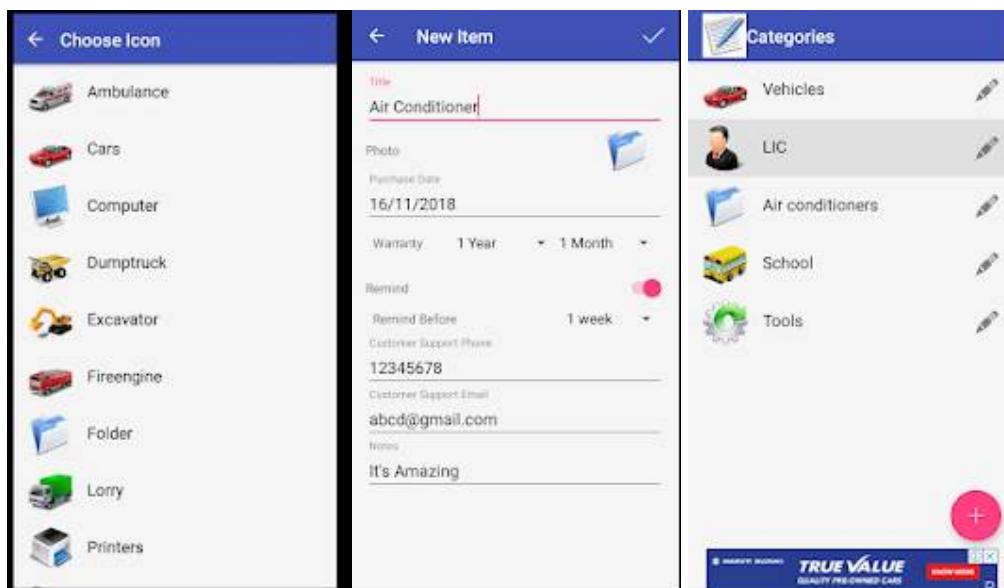


Ilustración 7. Aplicación Easy Maintenance Reminder

- EMS – Entire Maintenance Solutions⁷

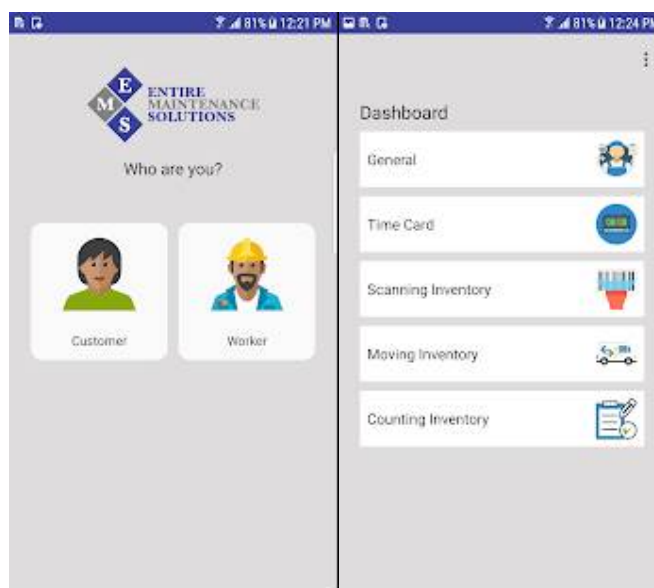


Ilustración 8. Aplicación Entire Maintenance Solutions

⁶ Ref.: Easy Maintenance Reminder (2019) [en línea]. [Fecha de consulta: 22 de febrero de 2019] < <https://play.google.com/store/apps/details?id=com.easy.maintenancereminder> >

⁷ Ref.: EMS Maintenance (2019) [en línea]. [Fecha de consulta: 22 de febrero de 2019] < <https://play.google.com/store/apps/details?id=com.maintenancesolution.ems> >

Resumen de ventajas y desventajas de otras apps de mantenimiento:

APLICACIÓN ANALIZADA	VENTAJAS	DESVENTAJAS
LIFT MAINTENANCE	<p>ES DE FÁCIL USO EL USUARIO DEBE REGISTRARSE PREVIO A SU USO SOLICITUD DE PERMISOS ESTABLECIDA VARIAS FUNCIONALIDADES PERMITE TENER ADMINISTRADOR Y USUARIOS QUE REPORTAN DENTRO DE LA APP</p>	<p>DISEÑO POCO AMIGABLE AL USUARIO A VECES FALLA EL RESPONSIVE LAYOUT EN ALGUNOS DISPOSITIVOS NO PERMITE PERSONALIZACIÓN PARA LA EMPRESA (DISEÑO DE LOGO, COLORES, ETC.) NO PERMITE COMPARTIR REPORTE EN PDF U EXPORTAR A OTROS FORMATOS</p>
HAPPY SERVICE	<p>PERMITE RESERVAR UN TÉCNICO PARA REALIZAR UNA TAREA ESPECIALIZADA EN VIVIENDA SOPORTE MULTI-DISPOSITIVO INTERFAZ GRÁFICA DE USUARIO ADECUADA PERMISOS ADECUADOS MANEJADOS DENTRO DE HAPPY SERVICE PERMITE ORGANIZAR SERVICIOS POR LISTADO Y PROYECTOS CHAT INTEGRADO EN LA APP (WE-CHAT)</p>	<p>DISEÑO NO PERMITE PERSONALIZACIÓN PARA UNA EMPRESA NO PERMITE COMPARTIR O EXPORTAR REPORTE EN PDF U OTROS FORMATOS DISEÑO NO PERMITE ASOCIAR RUTINAS DE MANTENIMIENTO A EQUIPOS - DE FORMA ADECUADA A LA NECESIDAD</p>
APP SHEET	<p>ES DE FÁCIL USO EL USUARIO DEBE REGISTRARSE PREVIO A SU USO SOLICITUD DE PERMISOS ESTABLECIDA VARIAS FUNCIONALIDADES DEBE CONSTRUIRSE LA APLICACIÓN PARA TENER TODAS LAS FUNCIONALIDADES REQUERIDAS, PERO NO PUEDE SER PERSONALIZADA, COMO SI LO PERMITE EL DESARROLLO MEDIANTE ANDROID STUDIO</p>	<p>PARA TENER TODAS LAS FUNCIONALIDADES Y UTILIZAR MAYOR CANTIDAD DE DATOS DEBE PAGARSE POR EL ACCESO. PARA AMPLIAR MAYOR CANTIDAD DE CARACTERÍSTICAS Y USUARIOS (> 10) SE REQUIERE DE PAGO. LA EXPERIENCIA DE USUARIO ES DE NIVEL MEDIO</p>
Easy Maintenance Reminder	<p>FÁCIL Y ÚTIL PARA ESTABLECER RECORDATORIOS DE EVENTOS. PERMITE PROGRAMACIÓN DE ACTIVIDADES Y ORGANIZAR POR CATEGORÍAS</p>	<p>NO TIENE FORMA DE CREAR ADMINISTRADOR Y PERSONAL TÉCNICO/USUARIOS DE APP PARA SUPERVISION DE LABORES</p>

EMS MAINTENANCE	FÁCIL USO, PERMITE CREAR NUEVOS EQUIPOS Y JERARQUIA, ADEMÁS DE LLEVAR INVENTARIO DE PARTES	NO TIENE FORMA DE CREAR ADMINISTRADOR Y PERSONAL TÉCNICO/USUARIOS DE APP PARA SUPERVISION DE LABORES, NO PERMITE CREAR RECORDATORIOS
------------------------	--	--

Tabla 3. Resumen de ventajas y desventajas de otras apps de mantenimiento:

1.2 Objetivos del Trabajo

1.2.1 Objetivo General

El objetivo del trabajo es optimizar la gestión del mantenimiento a través de la creación de una app, teniendo previamente la programación del recurso humano disponible.

1.2.2 Objetivos Específicos

- Crear los diferentes wireframes de la app.
- Realizar un diseño teniendo en cuenta “responsive layout” para diferentes versiones de Android OS.
- El diseño será centrado en usuario (DCU).
- Implementar los diferentes permisos requeridos: red, cámara, contactos, posición/ubicación/geolocalización, etc.
- Guardar el reporte de inspección o trabajo, con un campo de observaciones, fecha, firma, con opción de compartir para enviar el respectivo informe generado.
- Enviar reporte al cliente del informe una vez realizado
- La app debe permitir revisar la programación del personal técnico
- En caso de registrarse una llamada por falla de un equipo, el técnico codificará la falla una vez se realice la revisión con una de las opciones disponibles.
- Capacitar al usuario en el uso de la App, para asegurar el correcto registro de los datos.
- Escribir una documentación adecuada con el fin de continuar el desarrollo de la app en un futuro, y con fines de capacitación.

1.3 Enfoque y método seguido

El desarrollo de la app se tiene planeado sobre sistema Android, por ser en este caso un desarrollo de bajo costo, para una gran cantidad de dispositivos móviles existentes en el mercado, para lo cual se utilizará el IDE de Android Studio (lenguaje: Java)

Como parte adicional/opcional del Trabajo, para la simulación de los recursos disponibles, se creará un servidor web donde se alojará una base de datos.

Se revisarán las siguientes opciones de arquitectura para el manejo y visualización de datos:

Servidor – clientes: aplicación web para visualización de datos.

Servidor – clientes: aplicación nativa Android para visualización de datos.

Se incluirá la evaluación económica de costos operativos (actualmente realizando mediante llamadas y papel, la programación de actividades). El desarrollador tiene experiencia en el área de mantenimiento, siendo determinante en la metodología de trabajo.

“La metodología de trabajo tomará como modelo el desarrollo ágil basado en iteraciones, donde cada iteración, se realiza todas las fases de desarrollo. Cada ciclo tendrá su retroalimentación, por parte del tutor, y empresa Lift IL, analizando los requerimientos, implementando las funcionalidades y evaluando los resultados”⁸

“El diseño se centrará en usuario (DCU), siendo la iteración, aspecto clave del DCU, enriqueciendo cada fase con la respuesta del usuario para mejorar y adaptar los elementos necesarios. El proceso avanza de forma cíclica”⁹

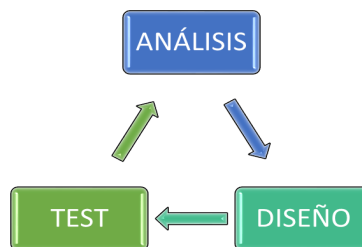


Ilustración 9. Iteración en DCU. Fuente: Propia

El modelo de programación a seguir tendrá en un mayor porcentaje el MVC, que propondrá separar la parte visual por un lado, del objeto/modelo y por otro del control lógico del programa. El objetivo de la visualización será presentar el modelo de forma adecuada, generando una gran experiencia de usabilidad. El controlador estará encargado de responder a los eventos de usuario, sirviendo de enlace entre el modelo y la vista.

⁸ Ref.: RAMIREZ, R.; BOLTÀ H. (2016), Métodos para el desarrollo de aplicaciones móviles PID_00246016, Universitat Oberta de Catalunya - UOC.

⁹ Ref.: FLAMARICH, Jordi, Conceptualización PID_00245395 (2018) [en línea]. [Fecha de consulta: 20 de febrero de 2019] <http://materials.cv.uoc.edu/daisy/Materials/PID_00245395/pdf/PID_00245395.pdf>



Ilustración 10. Modelo Vista Controlador (MVC). Fuente: Propia

Esquema de Metodología/Plan de trabajo:

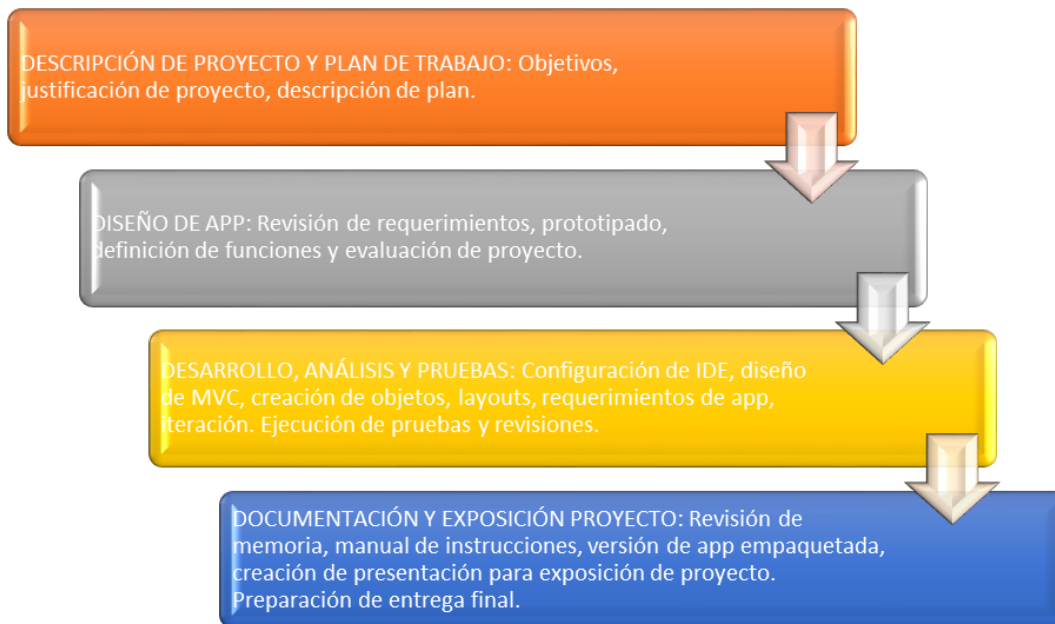
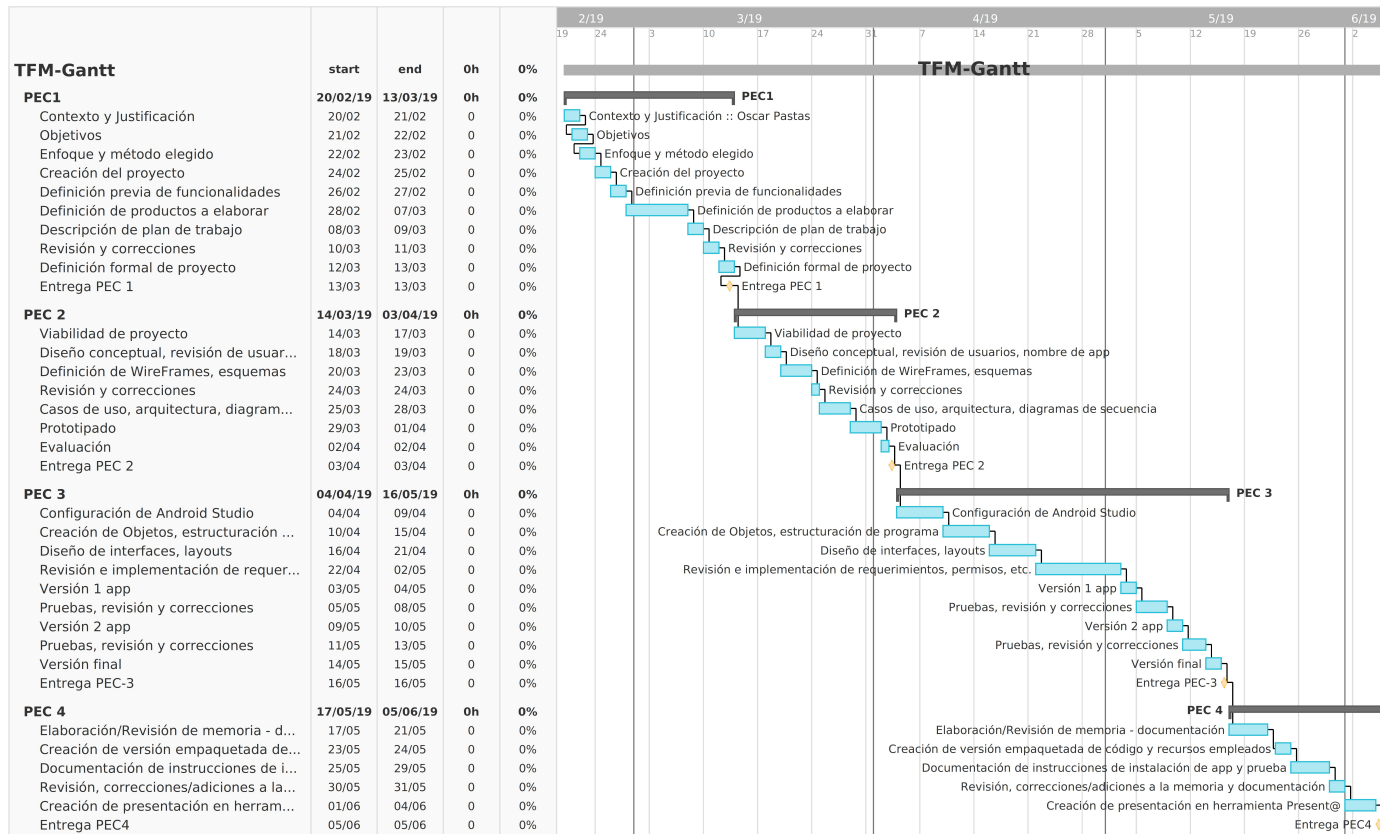


Ilustración 11. Esquema – metodología – Plan de trabajo. Fuente: Propia

1.4 Planificación del Trabajo

A lo largo del semestre, se presentan las diferentes actividades que permitirán alcanzar el objetivo del desarrollo de la aplicación descrita. El gráfico de Gantt permite planificar cada una de las tareas necesarias, marcando los hitos de cada una de las entregas del plan de trabajo.



Las tareas se han agrupado de acuerdo a las entregas a realizar:

- PEC1: Definición de proyecto
- PEC2: Diseño
- PEC3: Programación, análisis y pruebas
- PEC4: Ajuste de documentación y preparación de presentación de proyecto.

La dedicación diaria para las diferentes actividades será:

- Lunes a domingo: 4 horas semanales

Total de horas semanales: 27 horas

A continuación se describe el resumen de horas para realizar el proyecto:

	HORAS DE TRABAJO
PEC 1 - PLAN DE TRABAJO	92
PEC 2 - DISEÑO	80
PEC3 - IMPLEMENTACIÓN	160
PEC 4 - ENTREGA FINAL	76

Tabla 4. Resumen de horas de Trabajo por etapa

Lista de tareas/actividades que se llevarán a cabo durante el TFM:

HORAS DE TRABAJO	
PEC 1 - PLAN DE TRABAJO	92
<i>Contexto y justificación</i>	8
<i>Objetivos</i>	8
<i>Enfoque y método elegido</i>	8
<i>Creación del proyecto</i>	8
<i>Definición previa de funcionalidades</i>	8
<i>Definición de productos a elaborar</i>	28
<i>Descripción de plan de trabajo</i>	8
<i>Revisión y correcciones</i>	8
<i>Definición formal de proyecto</i>	8
Entrega PEC 1	
HORAS DE TRABAJO	
PEC 2 - DISEÑO	80
<i>Viabilidad de proyecto</i>	16
<i>Diseño conceptual, revisión de usuarios, nombre de app</i>	8
<i>Definición de WireFrames, esquemas</i>	16
<i>Revisión y correcciones</i>	4
<i>Casos de uso, arquitectura, diagramas de secuencia</i>	16
<i>Prototipado</i>	16
<i>Evaluación</i>	4
Entrega PEC 2	
HORAS DE TRABAJO	
PEC3 - IMPLEMENTACIÓN	160
<i>Configuración de Android</i>	24
<i>Creación de Objetos, estructuración de programa</i>	24
<i>Diseño de interfaces, layouts</i>	24
<i>Revisión e implementación de requerimientos, permisos, etc.</i>	44
Versión 1	8
<i>Pruebas, revisión y correcciones</i>	16
Versión 2	8

<i>Pruebas, revisión y correcciones</i>	12
<i>Versión final</i>	8
<i>Entrega PEC 3</i>	
	HORAS DE TRABAJO
PEC 4 - ENTREGA FINAL	76
<i>Elaboración/Revisión de memoria - documentación</i>	20
<i>Creación de versión empaquetada de código y recursos empleados</i>	12
<i>Documentación de instrucciones de instalación de app y prueba</i>	20
<i>Revisión, correcciones/adiciones</i>	8
<i>Creación de presentación en herramienta Present@</i>	16
<i>Entrega PEC4</i>	

Tabla 5. Lista de tareas/actividades del TFM

1.5 Breve resumen de productos obtenidos

Los productos a entregar una vez finalizado el TFM serán:

- APP EN ANDROID desarrollada completamente instalada en diferentes dispositivos y configuraciones, cumpliendo diseño “responsive layout”, donde se muestren las diferentes funcionalidades.
- Video con explicación del funcionamiento de la aplicación desarrollada
- Memoria y documentación del trabajo desarrollado
- Manual de usuario de app.

2. Diseño y Análisis

2.1 Breves criterios revisados para el diseño centrado en usuario de la app

- Experiencia propia del creador del proyecto en gestión de mantenimiento y requerimientos en equipos
- Utilidad para la empresa que va dirigida la aplicación
- Necesidades identificadas de acuerdo a entrevistas realizadas
- Opciones de accesibilidad, cumpliendo los requisitos de la empresa
- Contenidos deben ser fácilmente manejables por el usuario
- Debe existir documentación y capacitación adecuada en el manejo de la app
- La aplicación permitirá la creación de datos siendo almacenados en una base, útil para consulta de nuevos casos, adquirir estadísticas, y fuente de conocimiento para mejorar el área de mantenimiento en la prestación del servicio

2.2 Necesidades detectadas para la app:

De forma preliminar se detectan los siguientes requerimientos dentro de la app:

- Creación de reporte de actividades, de acuerdo a la necesidad: preventivo, correctivo, inspección.
- No hay registro actualmente de los reportes realizados por el personal técnico.
- Facilidad de manejo y usos, para una app que no está desarrollada a la medida
- Se realizarán encuestas y entrevistas para confirmar las necesidades detectadas de forma preliminar, permitiendo formular las funcionalidades de la aplicación adecuadamente.

2.3. Enfoque del producto y/o servicio

La aplicación cuenta con dos enfoques: en primer lugar, permite realizar consultas y observar el comportamiento del recurso humano del área de servicios y mantenimiento. En segundo permite realizar programación de labores, y verificar actividades, estableciendo una programación para cada uno de los equipos dentro de la app.

La aplicación podrá ser instalada en dispositivos Android / laptops (a futuro en iOS), compatible con la gran parte de dispositivos móviles actuales en el mercado.

2.4. Requisitos de usuario (Fichas realizadas a partir de la observación de usuarios en la empresa)¹⁰

Segmento No. 1: Personal que labora en el área de mantenimiento, segmento 25-60 años. Con ingresos medios, habilidades medias de uso de tecnología. Género: Hombre o mujer. Estudios: profesional técnico. Clase social: media/alta. Posibilidad de acceso a tecnología móvil o datos para instalación o ingreso a la aplicación porque cuenta con el equipo y plan por parte de la empresa.

Segmento No. 2: Personal administrativo en empresa de mantenimiento, segmento 20-70 años. Habilidades altas de uso de tecnología. Género: Hombre o mujer: Estudios: profesional/profesional técnicos: Clase social: media/alta. Posibilidad de acceso a tecnología móvil o datos para instalación o ingreso a la aplicación porque cuenta con el equipo y plan por parte de la empresa.

2.5. Puntos clave de la app (A partir de encuestas y entrevistas realizadas¹¹)

Como usuario administrador:

Puede programarse/crear actividad/reporte a cada persona /usuario normal registrado en una o varias actividades.

Envío de reportes pendientes a clientes mediante opción de revisar reporte (Los reportes han sido creados por el personal técnico).

Opción de recursos, para verificar lo necesario para una actividad programada. Revisión de emergencias por cada equipo y estadísticas sencillas para seguimiento.

Historial de las diferentes actividades de mantenimiento en equipos.

Revisión de emergencias teniendo en cuenta el listado de eventos consultados.

Como usuario normal:

Puede programarse a cada persona /usuario normal registrado en una o varias actividades.

Envío de reportes a usuario administrador mediante opción “Enviar reporte”.

La opción “Tareas” permite consultar el listado de tareas que tiene pendientes o ha llevado a cabo el usuario.

Historial de las diferentes actividades de mantenimiento en equipos.

2.6. Procedencia de público objetivo

Segmento No. 1, son personas que viven en la ciudad de Bogotá inicialmente. En nuestro caso Bogotá sería la ciudad elegida porque está la mayor concentración de personas de la empresa en Colombia y es sencillo realizar las pruebas necesarias con el personal. Personas en el segmento de 25-60 años. Con ingresos medios, habilidades básicas de uso de tecnología. Género:

¹⁰ Ver Anexo correspondiente

¹¹ Ver Anexo correspondiente.

Hombre o mujer. Estudios: estudiante, graduado universitario. Clase social: media, alta. (Ver Entrevistas y encuestas realizadas).¹²

Segmento No. 2, son personas que viven en la ciudad de Bogotá inicialmente. En nuestro caso Bogotá sería la ciudad elegida porque está la mayor concentración de personas de la empresa en Colombia y es sencillo realizar las pruebas necesarias con el personal. Personas en el segmento de 25-70 años. Con ingresos medios, habilidades avanzadas de uso de tecnología. Género: Hombre o mujer. Estudios: estudiante, graduado universitario. Clase social: media, alta. (Ver Entrevistas y encuestas realizadas).

2.7. Frase descriptiva de la aplicación

Como palabras o breve descripción de la aplicación se ha elegido: Programación, mantenimiento e historial en una sola aplicación.

En la pantalla de inicio cómo título se escribe: “[Nombre de] APP”

2.8. Contexto de uso

Está claramente definido, durante el día se realizará la programación de personal técnico, o de acuerdo a la falla que presente el equipo. Los usuarios ingresarán los datos, desde la oficina, edificio en el caso del administrador, mientras que el personal operativo lo realizará desde el sitio de inspección/mantenimiento. La mayor parte de los sitios permite acceso a internet.


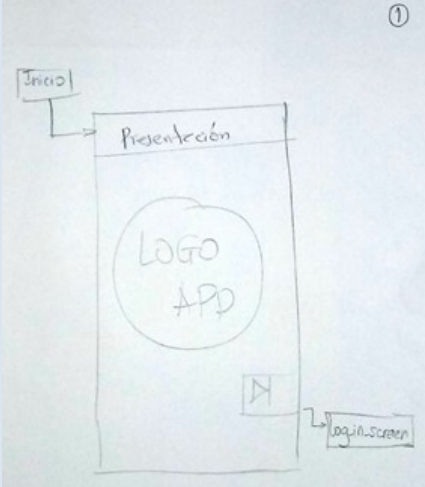

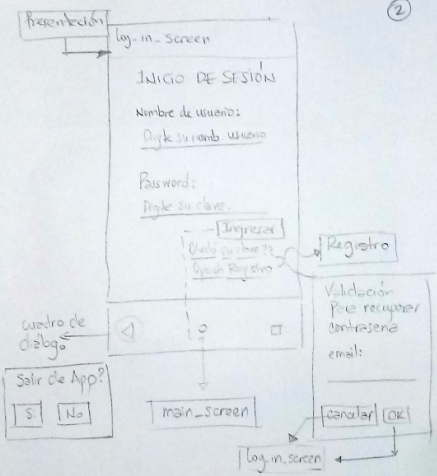
Como funcionalidad a futuro, en sitios donde no haya conexión de datos, se puede proceder a guardar los datos dentro de la app, para ser enviados una vez el móvil tenga conexión de red disponible para envío de los datos.

¹² Ver Anexo correspondiente

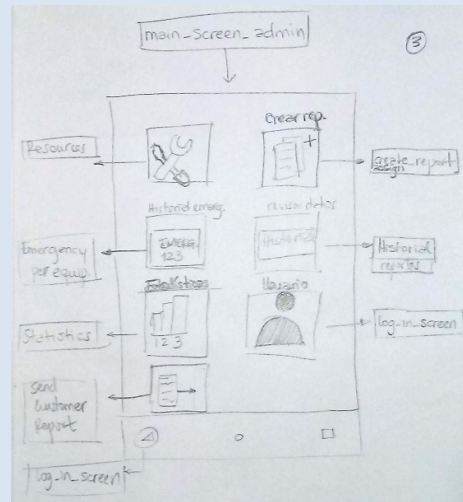
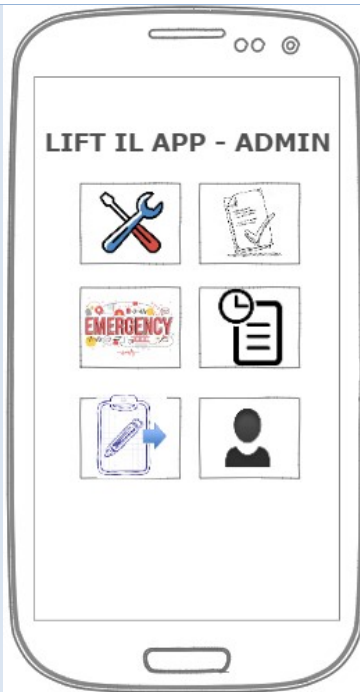
3. Diseño de wireframes y esquema de navegación

3.1. Wireframes y bocetos de app

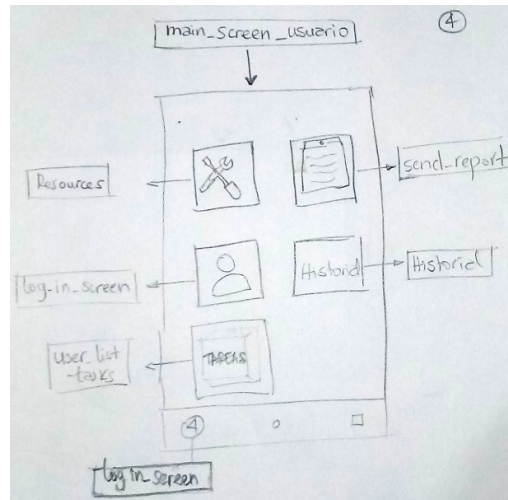
Se han realizado las diferentes pantallas/actividades en bocetos y en diseño definido empleando <https://ninjamock.com>.

NOMBRE DE PANTALLA	PROTOTIPO	BOSQUEJO BÁSICO
<p>HOME – Pantalla de presentación de la aplicación</p>		
<p>LOG-IN: Se verifica usuario y password registrados. Al realizar click en Aun no estás registrado?</p>		

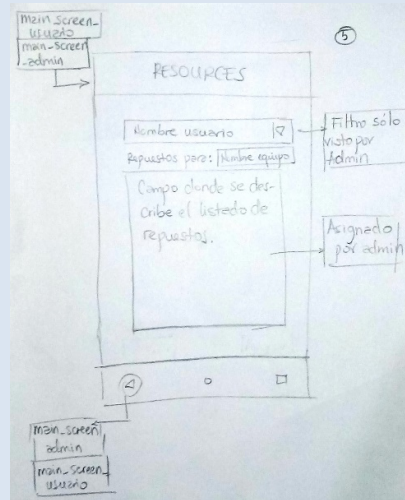
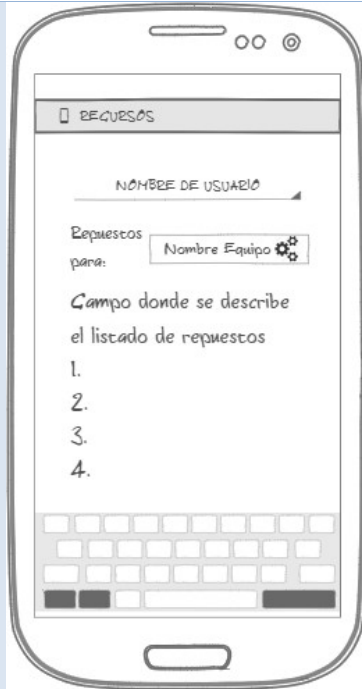
PANTALLA PRINCIPAL – ADMIN: Menú de opciones para usuario administrador



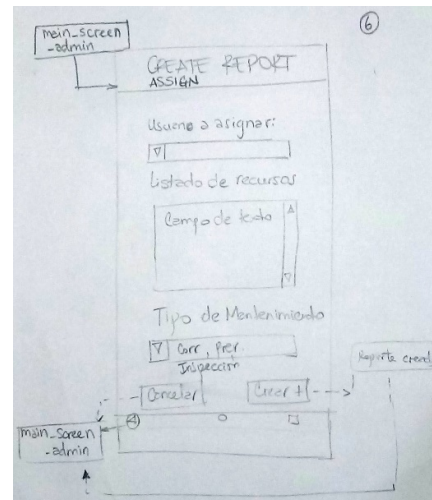
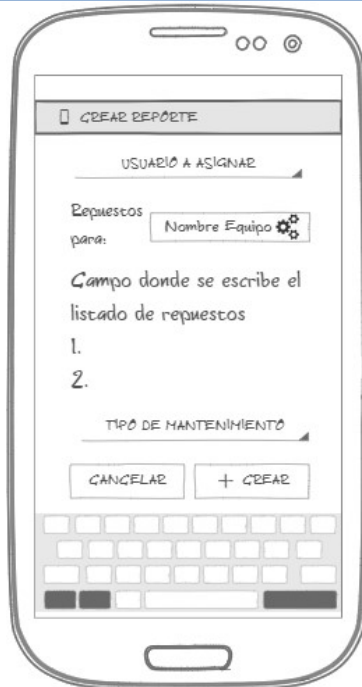
PANTALLA PRINCIPAL – USUARIO: Menú de opciones para usuario normal.



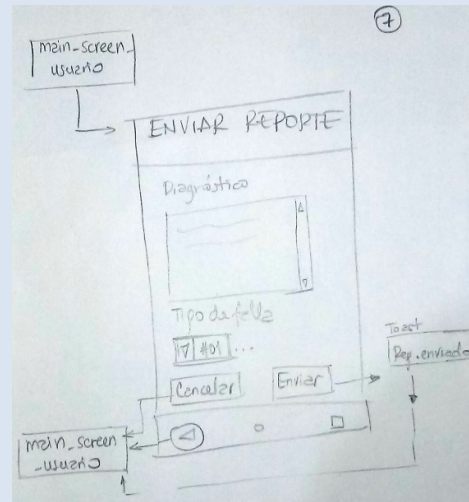
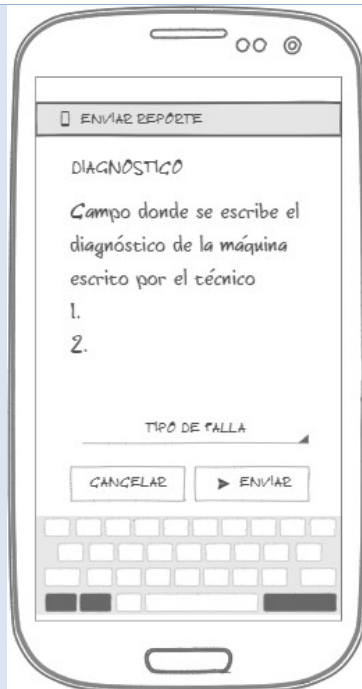
RESOURCES:
Pantalla que permite consultar los recursos por el equipo donde se ha programado algún mantenimiento asignado en fecha cercana.



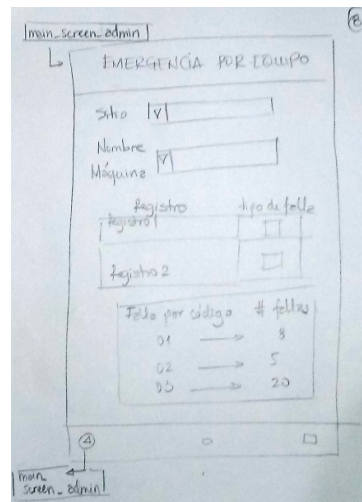
CREATE REPORT – ASSIGN: A partir de esta pantalla, el administrador asigna una tarea con los parámetros de listado de repuesto, el nombre del equipo y el tipo de mantenimiento necesario.



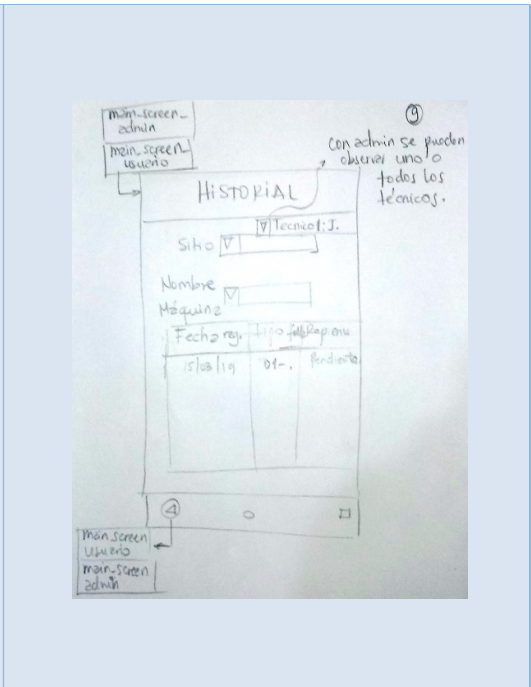
SEND – REPORT – USUARIO: A partir de esta opción, el usuario – técnico, envía el reporte de diagnóstico una vez finalizada la labor. Tiene la opción de adjuntar una fotografía.



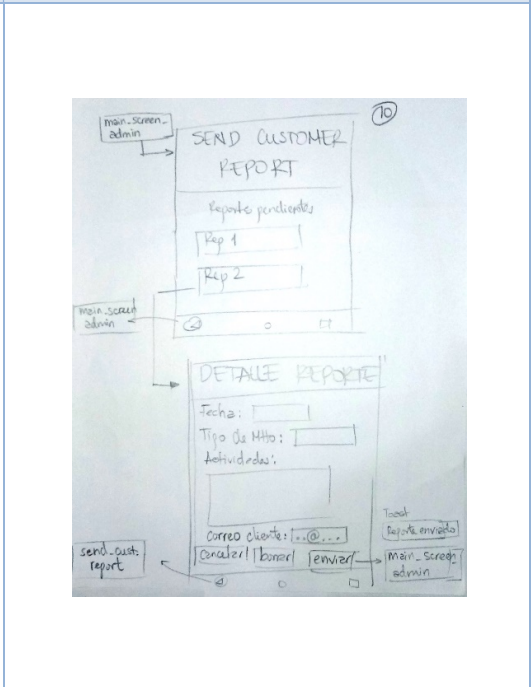
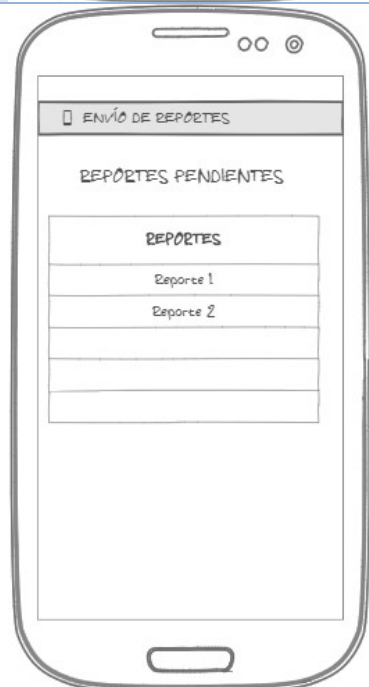
EMERGENCY PER EQUIPMENT: Pantalla que permite realizar una consulta de acuerdo al sitio, y nombre de máquina, de las labores llevadas a cabo y el tipo de falla.

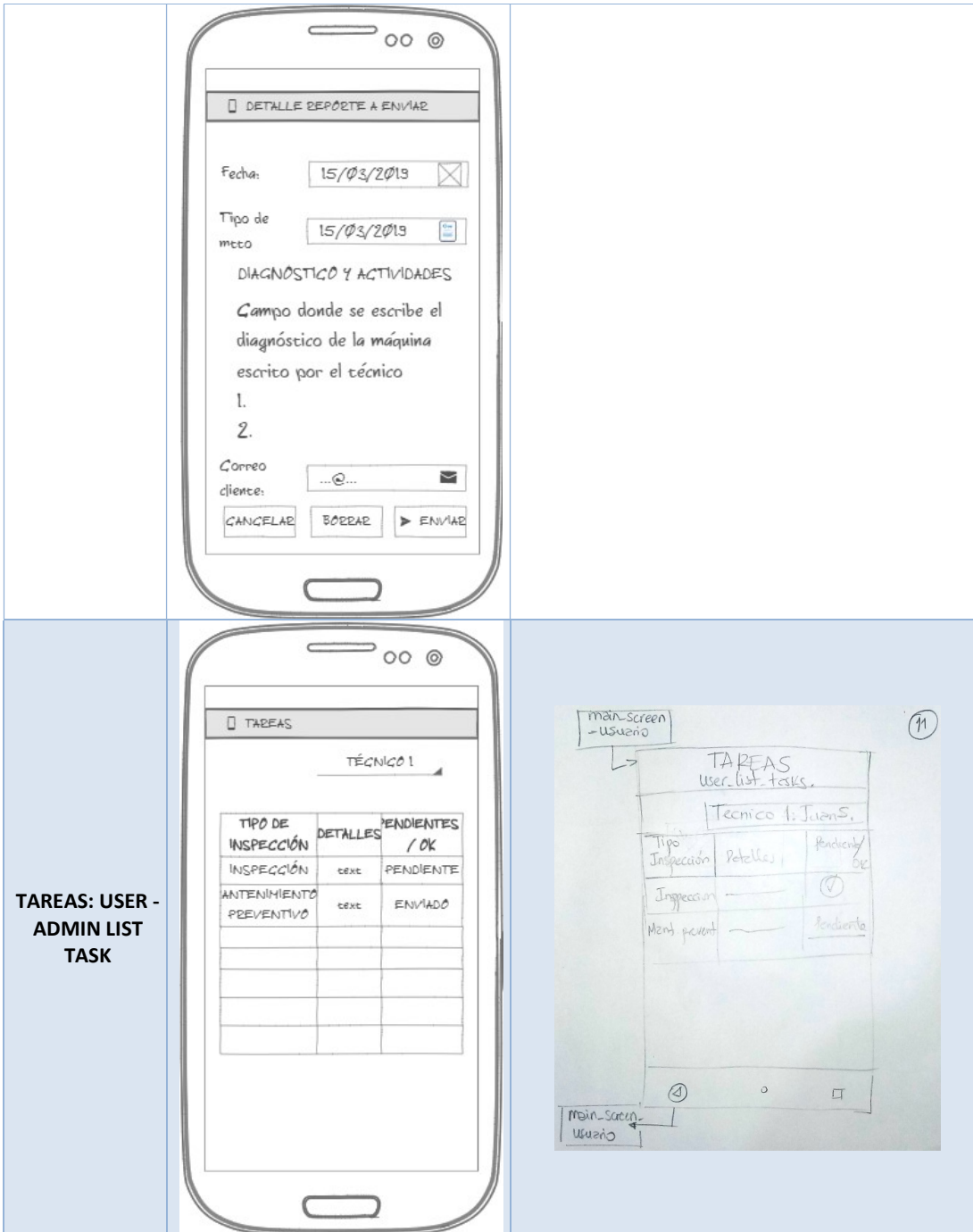


HISTORIAL:
Permite visualizar tanto al técnico(usuario normal), como al administrador



SEND CUSTOMER REPORT





TAREAS: USER - ADMIN LIST TASK

3.2. Esquema de navegación

a. Navegación para usuario tipo Administrador:

La navegación en la app se realiza mediante:

- Botones de navegación de sistema operativo: Back y home están habilitados dentro del prototipo para ir hacia atrás o a la pantalla de “Home” de la app. Al realizar doble click sobre el botón de back, la app va a la pantalla de inicio “Home”
- Paginación: Fácilmente el usuario puede saber dónde se encuentra tanto por el título, como por el contexto de la pantalla (Nombre de la pantalla)
- Botones: Los botones de cada pantalla son representativos e intuitivamente se puede conocer que función realizan.
- Menú Administrador: Existe una pantalla principal, que permite la navegación, con las siguientes opciones: Recursos, asignar actividad, emergencia por equipo, historial, editar usuario, enviar reportes pendientes a cliente.
- Elementos de control: Dentro de las pantallas que requieren elementos de control, por ejemplo botón de ingreso en el log-in, crear un reporte, registrarse dentro de la app, etc.

b. Navegación para Usuario normal – User:

- Las opciones descritas anteriormente aplican, solamente cambia el menú principal de acuerdo al tipo de usuario:
- Existen opciones diferentes habilitadas para el usuario normal:
- Existe una pantalla principal, que permite la navegación, con las siguientes opciones: Recursos, enviar reporte, historial, editar datos de usuario, listado de tareas llevadas a cabo/pendientes.

c. Mostrar información:

Para mostrar información relevante se usan cuadros de texto, labels sencillos y toast donde se reporta por ejemplo: Datos de registro de usuarios, datos de un reporte, tareas asignadas, al crear un reporte exitosamente se mostrará un toast confirmando la operación de forma correcta, etc.

Se emplean listados simples dentro de la app, para selección de alternativas en el registro, para país, tipo de tarjeta, etc.

Hay pantallas que ofrecen detalles específicos, por ejemplo en la pantalla de envío de reportes pendientes un mensaje se despliega para entregar mayor información de acuerdo al ítem seleccionado

d. Entrada de datos:

Para el ingreso de datos en la app, se emplea teclado normal y/o numérico según sea el caso.

- Campos de texto y formularios

Se presentan pantallas de registro para el perfil de usuario. Para el ingreso de datos, se emplea listados de selección, teclado, botones de control para confirmación de registro.

3.3. Criterios generales de diseño en wireframes y navegación

Como criterios de diseño generales se tuvieron en cuenta los siguientes aspectos:

- a. Se evita una pantalla para cada cosa, y se trata de reunir varios temas en una sola pantalla, resumida, clara y con la información relevante para el usuario.
- b. El árbol de navegación de la app se encuentra definido, por lo cual se ha establecido las diferentes relaciones entre pantallas.
- c. Cada pantalla tiene un uso intuitivo para cada botón y gráfico. Adicionalmente se indica cuáles son los items donde se puede realizar clicks (clickables), mediante un mensaje sencillo en cada elemento.
- d. Existen pantallas donde el ingreso de datos se realiza mediante controles, o en las fechas se emplea un calendario desplegable a fin de minimizar el esfuerzo para introducir datos.
- e. El diseño se realiza pensando en que los elementos tienen el tamaño apropiado para cualquier usuario (incluyendo dedos grandes).
- f. El contenido en primer plano se cumple en todas las pantallas, por esto las barras de búsqueda y navegación se han colocado en la parte inferior y superior, dejando el contenido relevante en el centro de la pantalla y en primer plano.

3.4. Diseño de alta definición

Luego de crear los bocetos y prototipo sencillo, se procedió a crear un diseño de alta definición utilizando el software Axure RP (licencia student) y Share, Team edition, versión v. 8.1.0.3382

Este prototipo fue publicado en la web de Axure Share, con la cuenta creada por opastas@uoc.edu:

Link del prototipo: <https://tbut08.axshare.com>

Contraseña: tfm1234

Usuario prueba: email: pec3@uoc.edu

Contraseña: 1234

Con este diseño definido los usuarios realizaron test de usabilidad que permitirán que la implementación sea óptima y eficiente.



**Ilustración 12. Menú Principal – Usuario Administrador – Prototipo de alta definición.
Fuente: Propia¹³**

3.5. Test con usuarios – Usabilidad – Evaluación:

Las pruebas con usuarios se llevaron a cabo una vez terminado y publicado de alta definición. En este caso, el observador y facilitador fue la misma persona (quien presenta la PEC, Oscar P.). Tres usuarios (Natalia P., Adriana Q., Fabiola B.) nos dieron sus comentarios (de acuerdo a los comentarios recibidos, se procedió a corregir las respectivas fallas observadas, y se mejoró la experiencia de usuario):

Por qué los botones de back no tienen funcionalidad dentro del prototipo?

R/ Se ha modificado el prototipo y los botones back y home tienen toda la funcionalidad estilo app. Incluso al dar doble click sobre el botón de back, la aplicación nos conduce a la pantalla home.

Se pueden crear máquinas dentro del diseño?

R/ Se tiene contemplado crear máquinas utilizando inicialmente la base de datos.

Donde se pueden consultar las diferentes actividades asignadas a usuarios?

R/ Dentro de la consulta de reportes, tanto como del historial, se podrá observar la programación de cada uno de los usuarios normales.

El sistema almacena un nuevo usuario creado?

R/ No se crearon variables de almacenamiento en un inicio, pero para mejorar la experiencia de usuario se crearon variables para este fin.

¹³ Para observar las pantallas creadas en el modelo de alta definición, utilizar el enlace descrito en la sección "Diseño de alta definición"

Que sucede si hay una notificación de Whatsapp u otro programa si estoy empleando la aplicación

R/ La app permite interactuar normalmente con las notificaciones entrantes. Se incluye simulación de una notificación en varias pantallas.

Es posible tener barras de navegación dentro de la aplicación para facilitar la creación de reportes?

R/ No se crearon barras de navegación dado que cada pantalla muestra lo requerido para el manejo de reportes. Se evaluará si una barra de navegación mejora la interfaz en la etapa de implementación.

3.6. ARBOL DE NAVEGACIÓN USUARIO TIPO ADMIN:

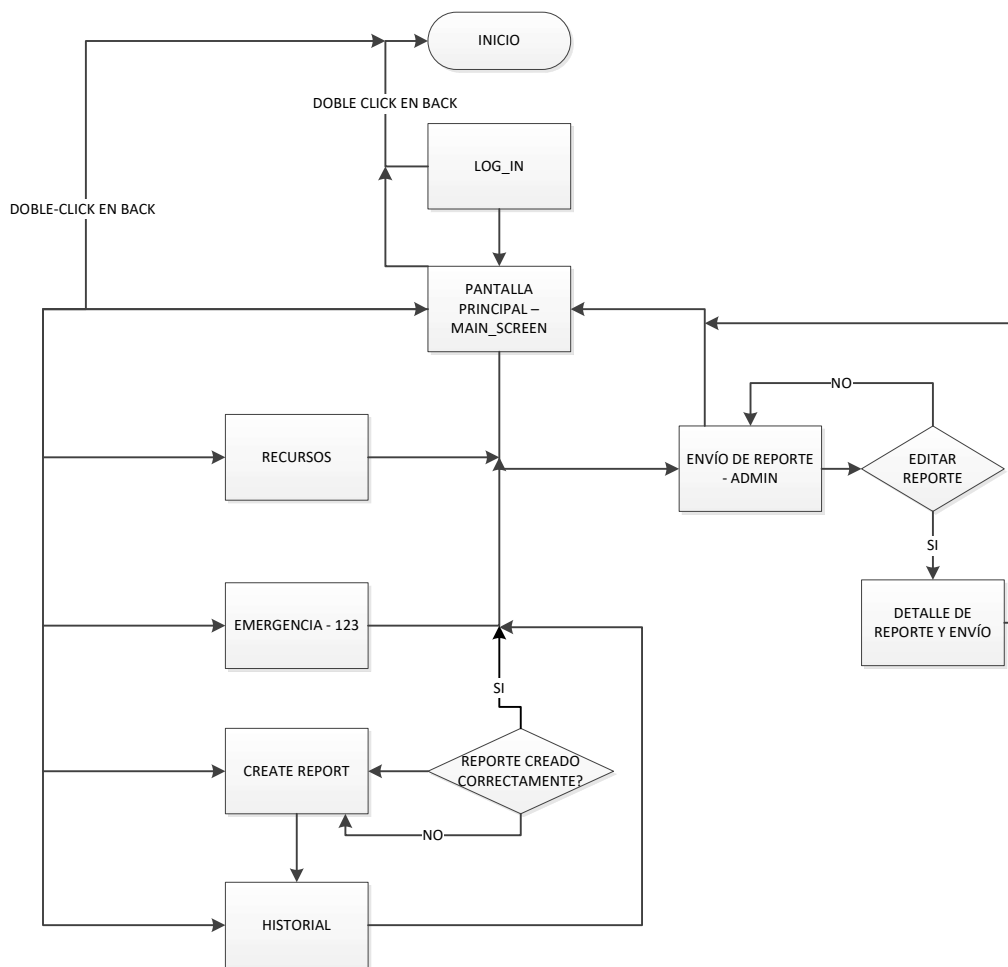


Ilustración 13. Árbol de navegación – Usuario Administrador. Fuente: Propia

3.7. ARBOL DE NAVEGACIÓN USUARIO TIPO NORMAL:

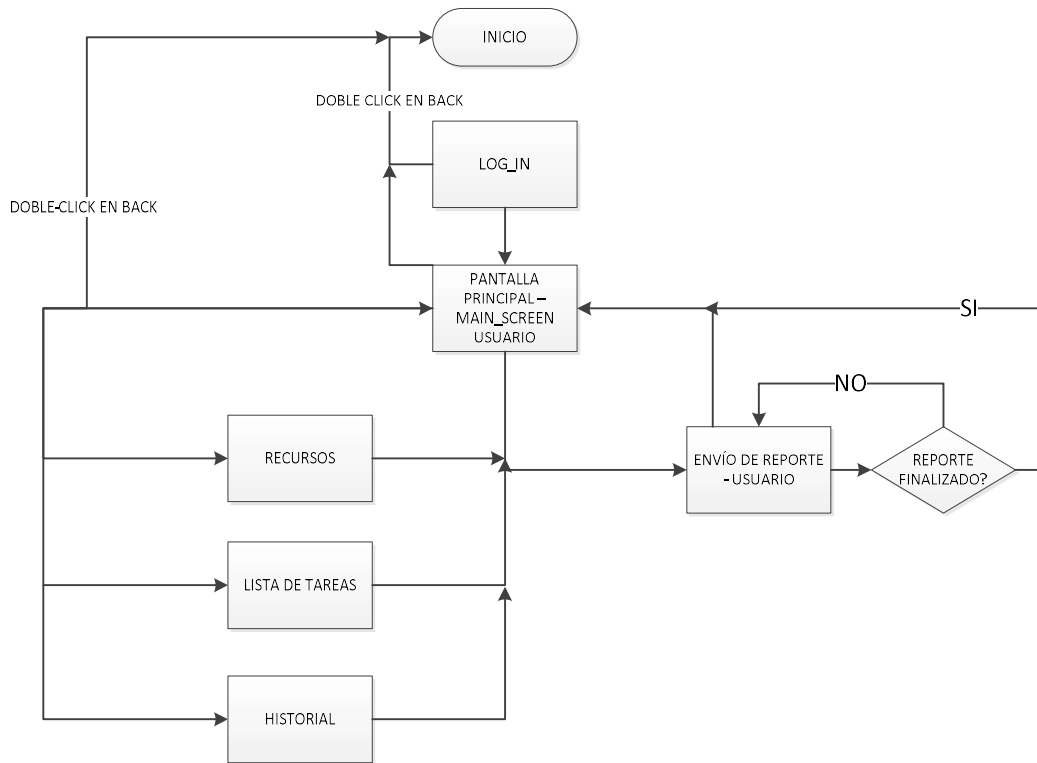


Ilustración 14. Árbol de navegación – Usuario Normal. Fuente: Propia

4. Base de datos

Aunque no es del alcance de este trabajo realizar el diseño de base de datos, se incluye porque es importante para las pruebas de la aplicación, y más adelante poder contar con persistencia de datos.

4.1. Casos de uso

Los casos de uso preliminares serán los siguientes:

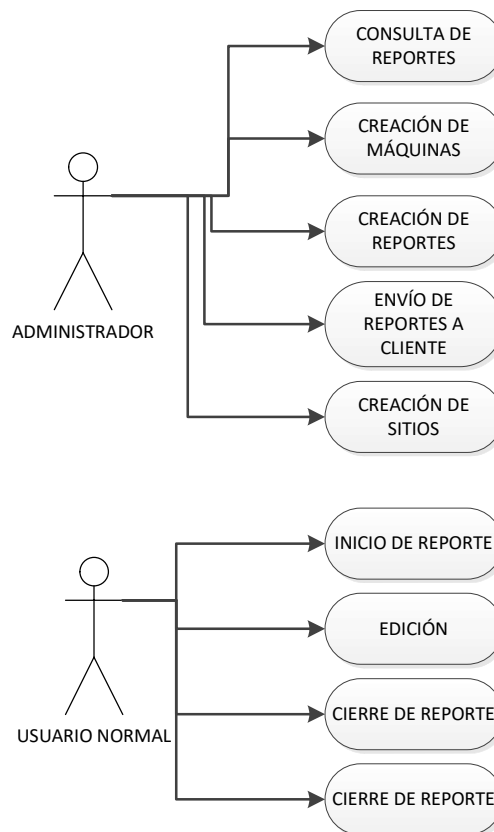


Ilustración 15. Casos de uso. Fuente: Propia

4.2. Actores y guiones

4.2.1. Identificación de actores

Los actores en este caso serán: usuario normal y usuario administrador.

El usuario administrador crea máquinas, equipos, realiza consultas y programa reportes/actividades. Adicionalmente edita los reportes para ser enviados a clientes.

El usuario normal, abre los reportes creados (inicio), consulta reportes pendientes, cierra reportes y envía a revisión.

4.2.2. Guiones

1. Un usuario administrador puede crear o consultar varias máquinas. Una máquina pueden ser creadas por un usuario administrador.
2. Un usuario administrador puede crear uno o varios sitios. Un sitio puede ser creado por un usuario administrador.
3. Un usuario administrador puede crear uno o varios reportes. Un reporte sólo puede ser creado por un usuario administrador.
4. Un reporte solamente puede tener una máquina asociada. Una máquina puede tener varios reportes asociados a ella.
5. Un sitio puede tener una o varias máquinas asociadas. Una máquina sólo puede estar asociada/dentro de un sitio.
6. Un usuario normal puede consultar o diagnosticar varias máquinas. Una máquina puede ser consultada o diagnosticada por varios usuarios normales.
7. Un usuario normal puede diligenciar uno o varios reportes. Un reporte sólo puede ser consultado o diligenciado por un usuario.
8. Un usuario normal puede consultar un sitio a la vez. Un sitio puede ser consultado por varios usuarios normales al mismo tiempo.

4.3. Modelo Entidad – Relación

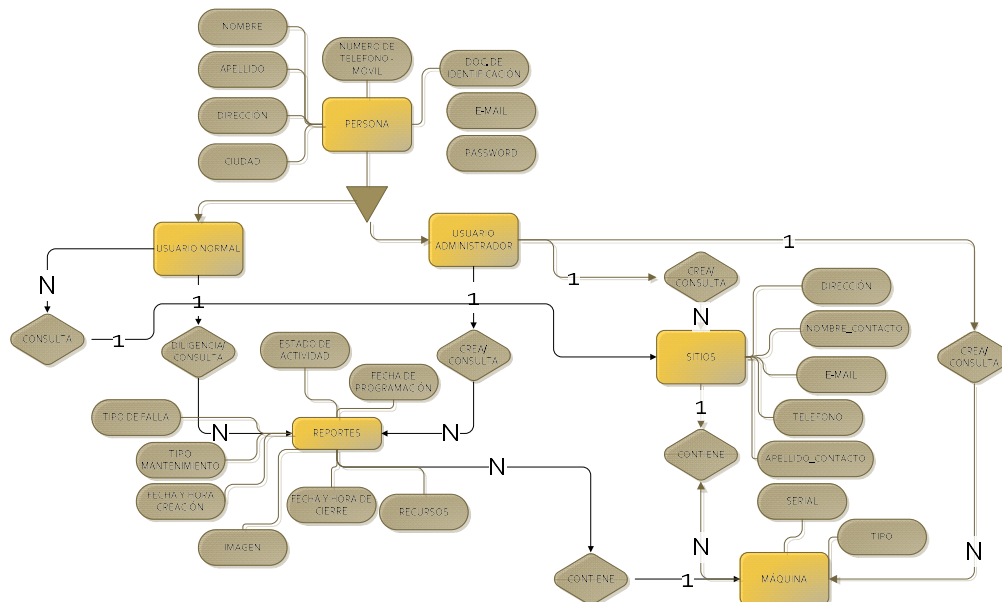


Ilustración 16. Diagrama de entidad-relación. Fuente: Propia

4.4. Modelo Relacional de base de datos

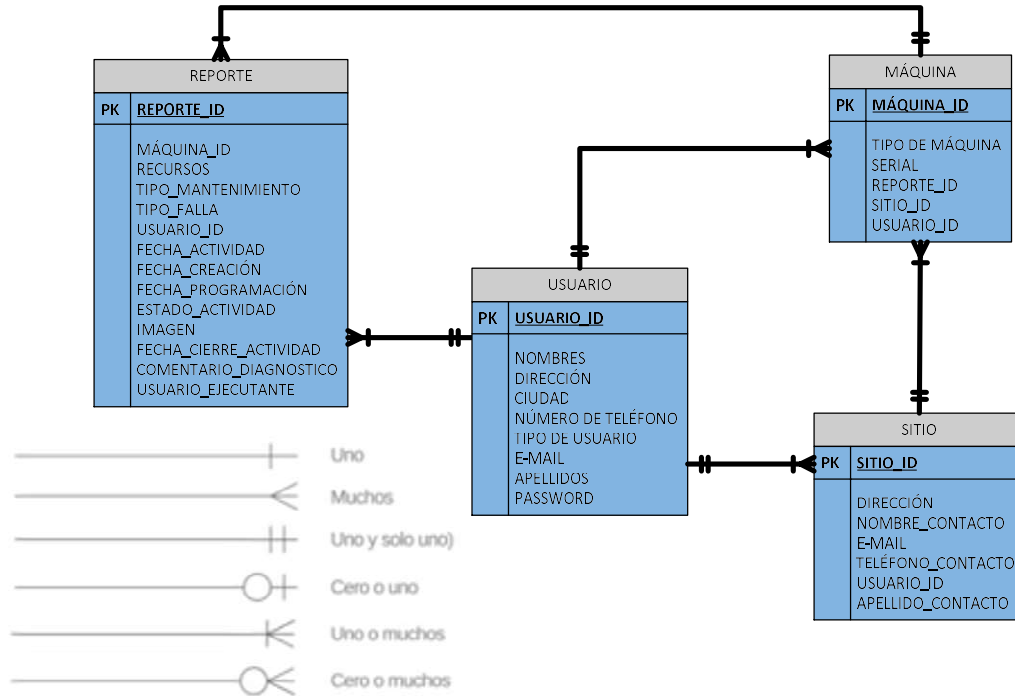


Ilustración 17. Diagrama de base de datos relacional y convenciones. Fuente: Propia

4.4. Implementación de base de datos

Para el desarrollo de base de datos, se emplearán una de las siguientes opciones:

- Base de datos en Firebase, conectada a aplicación en Android Studio:

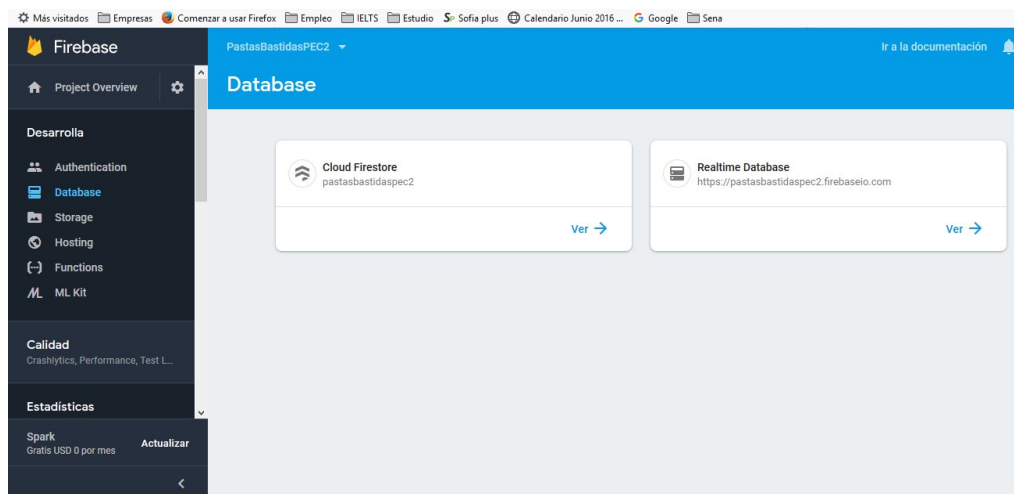


Ilustración 18. Base de datos creada en Firebase. Fuente: Propia

- Desarrollo de app con API de tipo REST en Android Studio conectado a base de datos con MySQL (Por la experiencia del autor del trabajo, se considera esta segunda opción como adecuada y viable para una expansión futura de la aplicación dada la gran cantidad de datos que puede manejar la base a largo plazo). Se tiene disponible para pruebas un servicio de web hosting:

Para la segunda opción se crea la base de datos con nombre mlvapp:

```
mysql -u mlvapp -p -h mysql.mlvtec.com mlvapp
```

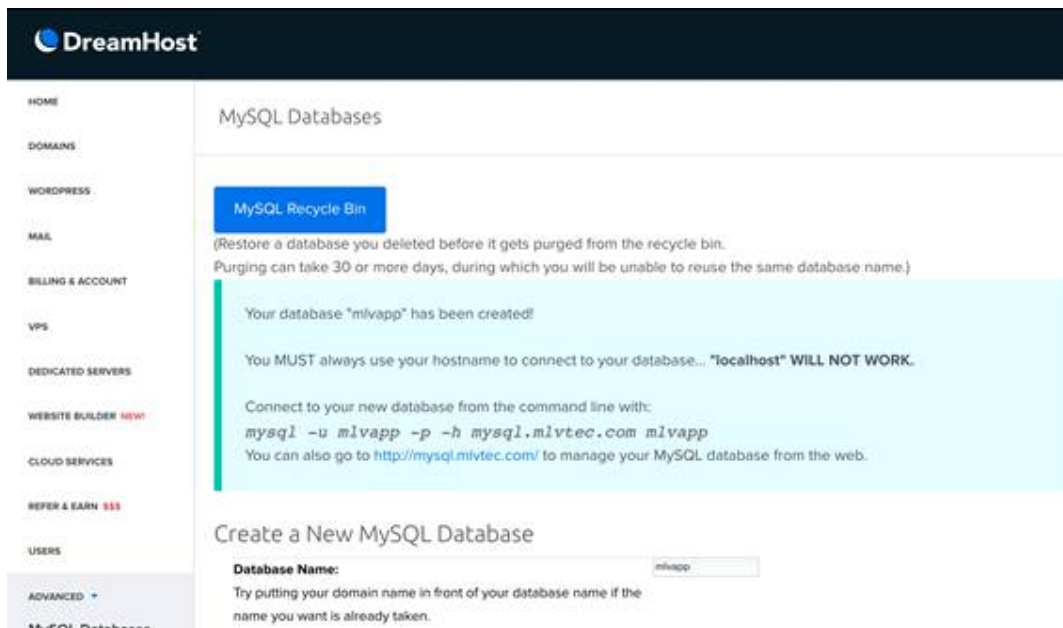


Ilustración 19. Base de datos creada en dreamhost web. Fuente: Propia

5. Arquitectura de sistema

El diseño propuesto en este caso es el modelo MVC, de forma general:

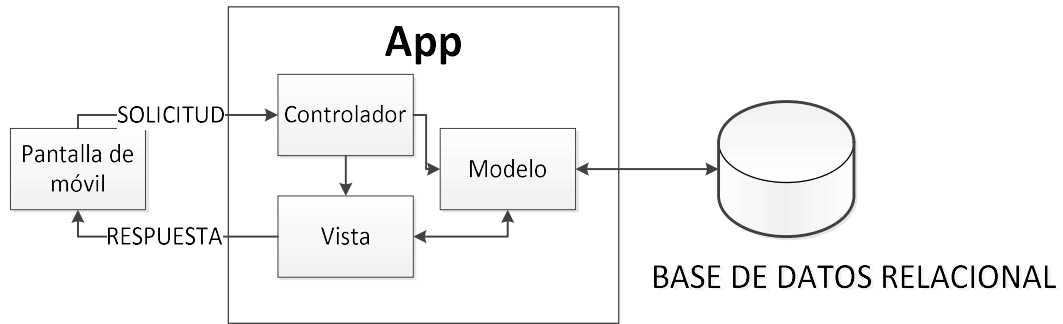


Ilustración 20. Arquitectura de sistema. Fuente: Propia

Dentro de la aplicación, también se tendrá en cuenta dicho modelo para crear las diferentes clases.

6. Valoración económica preliminar

A continuación se presentan los costos actuales asociados a operación, importantes para el propósito del TFM (Se asume como tasa de conversión 1 EUR (€) - COP (\$) = \$ 3.555,9):

TOTAL COSTO PLAN DATOS	\$ 1.200.000
TOTAL GASTOS PAPELERÍA MENSUAL	\$ 30.000
GASTOS MENSUALES OPERACIÓN COP (\$)	\$ 1.230.000
GASTOS MENSUALES OPERACIÓN EUR (\$)	345,9 €

Tabla 6. Costos actuales de operación – Lift IL

Para la evaluación de los costos asociados al trabajo del desarrollador – en el TFM:

TOTAL HORAS A INVERTIR EN EL TFM	408
PAGO DESARROLLADOR POR HORA EN EUROS (€)	5,0 €
TOTAL EUROS (€)	2.040,0 €
VALOR CONVERSIÓN 1 EUR (€) - COP (\$)	\$ 3.555,9
TOTAL COP (\$)	\$ 7.446.000

Tabla 7. Costos de desarrollo de Lift IL App

Al no explotar la funcionalidad de datos, Lift IL invierte gastos mensuales no utilizados. Con el desarrollo de la aplicación, se aprovecharían tanto los datos, como la información generada a partir de la aplicación (costos intangibles), con lo cual se podría estimar un valor de retorno de inversión, aproximado a partir de los costos totales de operación dividido el costo de inversión del desarrollo:

TIEMPO DE RETORNO DE INVERSIÓN APROXIMADO (MESES)	6,1
--	------------

Tabla 8. Tiempo de retorno de inversión aproximado

El tiempo de retorno es bastante corto, la empresa Lift IL, estaría de acuerdo en implementar el desarrollo teniendo en cuenta las condiciones expuestas, otorgando una valoración positiva al proyecto.

7. Implementación de base de datos y entorno de desarrollo en Drupal¹⁴ - servidor

7.1. Implementación de base de datos en Dreamhost:

La base de datos se aloja en un servidor (Dreamhost), cuyo acceso se realiza a través de:

[Servidor: tfmdb.mariopastas.com](https://tfmdb.mariopastas.com)

Database Credentials:

User: maupastasdb

Database Name: tfmdb

Se crea un repositorio con la instalación de drupal y herramientas en Gitlab:

git@gitlab.com:opastas/tfm.git o <https://gitlab.com/opastas/tfm.git>

Usuario para subir archivos en servidor dreamhost y acceder como administrador:

The screenshot shows the 'Editing User: maupastas (on nelson)' page. It contains several configuration sections:

- Full Name:** maupastas
- Home Directory:** /home/maupastas
- Enhanced security?**
- User Type:** Shell user - allows login via SSH (secure shell) for command-line access, as well as SFTP. Shell Type: /bin/bash. Disallow FTP?
- CPU Reporting:**
- New Password:** leave blank for no change (8-31 characters)
- New Password Again:** (empty field)
- Pick a password for me:**

A blue 'Save Changes' button is located at the bottom right of the form.

Ilustración 21. Edición de usuario y configuración de acceso

Una vez añadido el usuario, se crea el sistema de archivos para el dominio / subdominio seleccionado dándole acceso para la modificación/creación de archivos:

¹⁴ Ref.: Drupal (2019) [en línea]. [Fecha de consulta: 16 de mayo de 2019]<
<https://www.drupal.org/>>

ssh maupastas@homiemail-sub0.dreamhost.com

El acceso se realizará a través de una llave ssh.

Dentro del servidor se ha configurado el uso de PHP v. 7.2

The screenshot shows the DreamPress control panel for the domain `tfm.mariopastas.com`. The interface is titled "Fully Hosted" and includes a sub-domain configuration section. Under "Domain name", the domain is set to `tfm.mariopastas.com`. The "Do you want the www in your URL?" section has three options: "Leave it alone" (unselected), "Add WWW" (unselected), and "Remove WWW" (selected). The "Users, Files, and Paths" section shows the user `maupastas (on netson)` and the web directory `/home/username/tfm.mariopastas.com`. The "Web Options" section shows "PHP mode" set to "PHP 7.2 FastCGI (Default)", "HTTPS" disabled, "PHP Priority Upgrade" checked, and "Extra Web Security?" checked. At the bottom, there are links for "DNS | Visit | WebFTP | UniqueIP", "Shared Hosting with PHP 7.2 (User: maupastas)", "Not Secure", "Add Malware Remover", "0 Addresses", and buttons for "Restore" and "Delete".

Ilustración 22. Usuario y selección de herramientas, configuración de dominio.

7.2. Instalación de Drupal¹⁵

Después de la configuración de base de datos, y PHP, Apache en las versiones adecuadas, se procede a instalar Drupal CMS:

<https://github.com/drupal-composer/drupal-project>

Se accede a la instalación de composer, una herramienta para manejar la instalación de librerías y dependencias.

```
$ composer install
```

7.3. Herramienta JSON – API de Drupal:

Finalmente la herramienta empleada para enviar los datos desde el servidor hacia la app, es el módulo JSON API, configurado en Drupal, tanto para recibir como enviar los datos hacia clientes. Con este módulo se puede realizar el

¹⁵ <https://github.com/drupal-composer/drupal-project>

consumo de REST API, es decir, desde Lift IL APP se pueden enviar operaciones CRUD hacia el servidor donde se encuentra instalado Drupal y alojada la base de datos.

Virtualización del entorno de desarrollo

Para posterior desarrollo, y facilidad de virtualización de entorno, se emplearon las herramientas vagrant y box scotch (con el fin de evitar tener que crear el entorno de desarrollo nuevamente instalando PHP y Apache):

```
git clone https://github.com/scotch-io/scotch-box tfm
cd tfm
vagrant up
```

7.4. Creación de tablas y administración de contenido

A través de drupal y su interfaz gráfica web, se crearon las tablas correspondientes al diseño de la app.

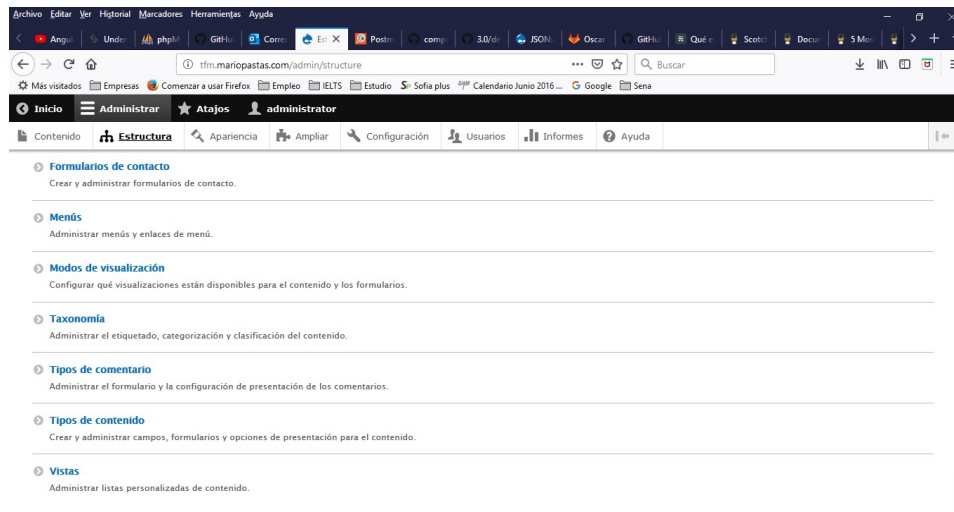


Ilustración 23. Estructura definida a partir de tipos de contenido

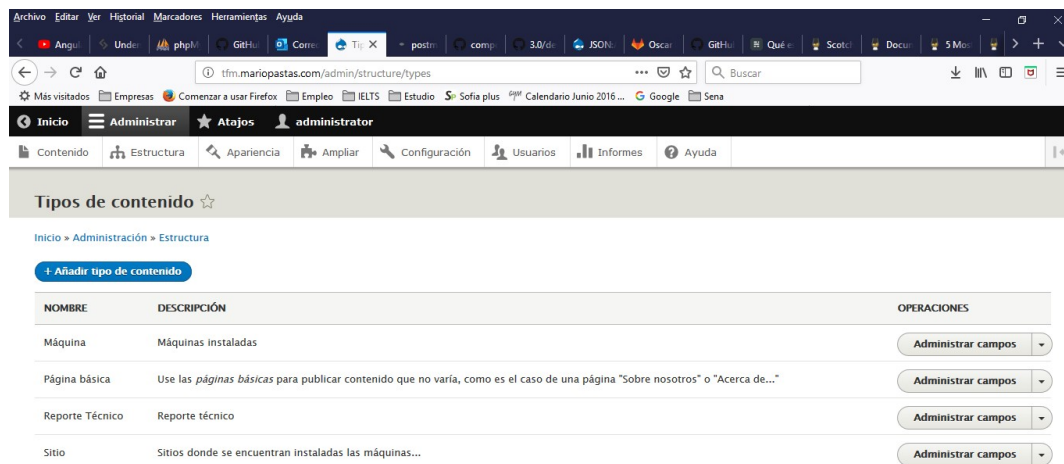


Ilustración 24. Tablas creadas dentro del administrador de contenido.

Inicio » Administración » Informes

Esta lista muestra todos los campos actualmente en uso como referencia rápida.

NOMBRE DEL CAMPO	TIPO DE ENTIDAD	TIPO DE CAMPO	USADO EN
body	block_content	Texto (con formato, largo, con resumen) (módulo: text)	Bloque básico
comment_body	comment	Texto (largo con formato) (módulo: text)	Comentarios por defecto
body	node	Texto (con formato, largo, con resumen) (módulo: text)	Página básica, Reporte Técnico
field_address	node	Texto (sin formato) (módulo: core)	Sitio
field_contact_name	node	Texto (sin formato) (módulo: core)	Sitio
field_contact_phone	node	Texto (sin formato) (módulo: core)	Sitio
field_email	node	Correo electrónico (módulo: core)	Sitio
field_failuretype	node	Lista (de texto) (módulo: options)	Reporte Técnico
field_image	node	Imagen (módulo: image)	Reporte Técnico
field_installation_site	node	Referencia de entidad (módulo: core)	Máquina

Ilustración 25. Entidades creadas en table. Consultadas a través de: Inicio – Administración – Informes.

7.4.1. Tabla reporte técnico:

Dentro de la tabla de reporte se incluyen los campos: comentario, estado, fecha de programación, imagen, máquina, recursos, tipo de falla, tipo de mantenimiento, técnico o usuario que ejecuta.

Administrar un campo

ETIQUETA	NOMBRE DE SISTEMA	TIPO DE CAMPO	OPERACIONES
Comentario	body	Texto (con formato, largo, con resumen)	Editar
Estado	field_state	Lista (de texto)	Editar
Fecha Programación	field_scheduled_date	Fecha	Editar
Imagen	field_image	Imagen	Editar
Maquina	field_machine_ref	Referencia de entidad	Editar
Recursos	field_resources	Texto (largo con formato)	Editar
Tipo de falla	field_failuretype	Lista (de texto)	Editar
Tipo Mantenimiento	field_mantenimiento	Lista (de texto)	Editar
Usuario ejecutante	field_user_job	Referencia de entidad	Editar

Ilustración 26. Tabla de Reporte técnico – campos configurados

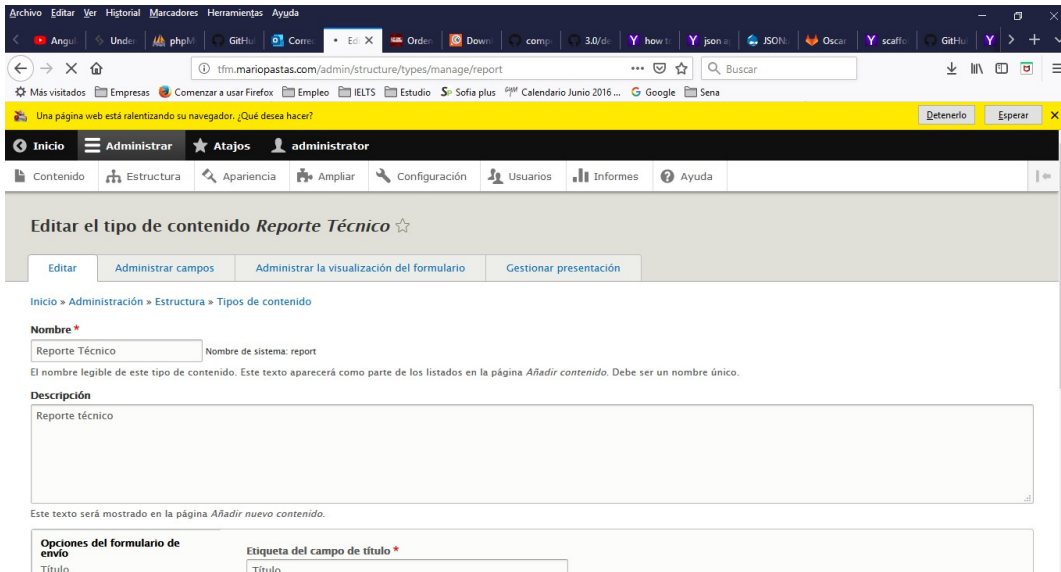


Ilustración 27. Ejemplo de creación de reporte técnico

7.4.2. Tabla de usuarios:

El sistema de administración de contenido permite crear, modificar, actualizar usuarios. Se han creado usuarios de prueba para la aplicación cliente en Android.

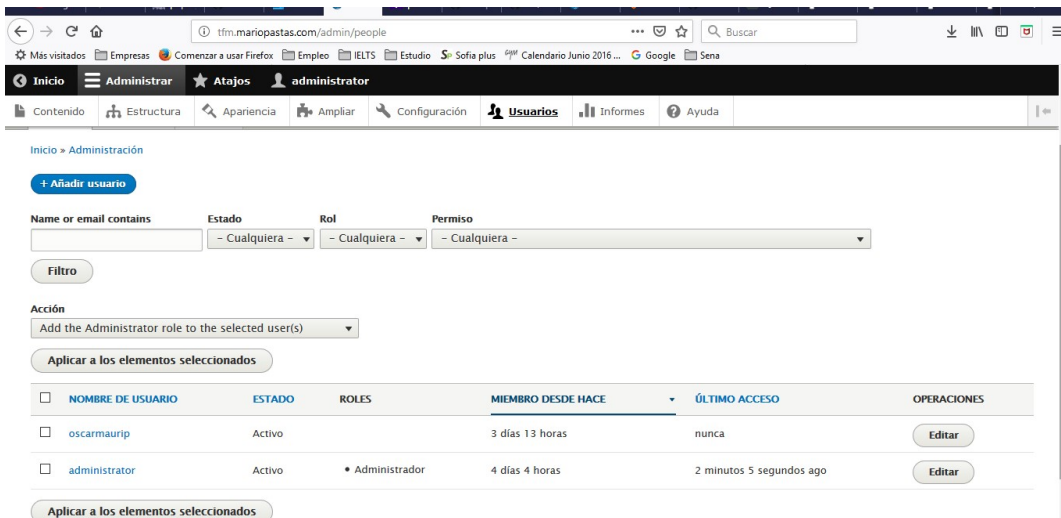


Ilustración 28. Usuarios – Configuración y creación

7.4.3 Tabla Máquina:

Dentro de la tabla máquina se incluye de acuerdo al diseño: Serial, sitio de instalación, y tipo de máquina. Se crearon máquinas de prueba para la aplicación.

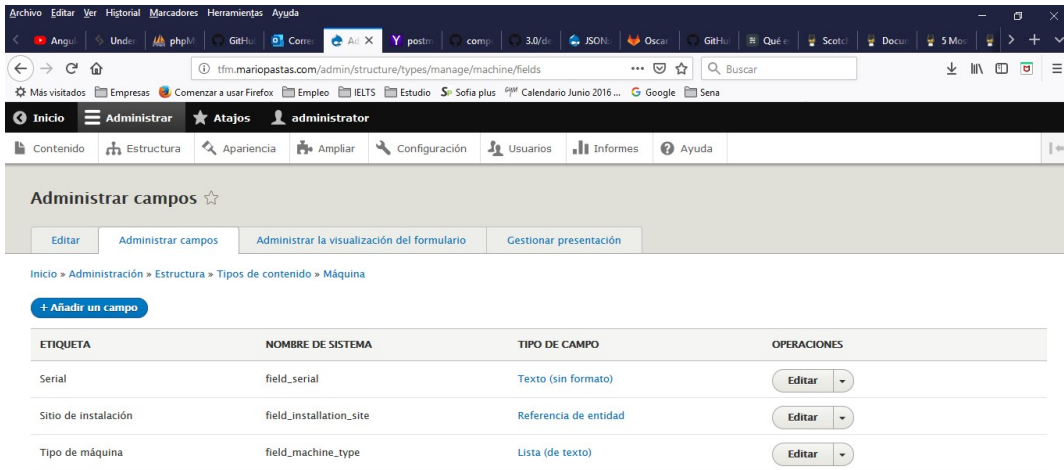


Ilustración 29. Configuración de campos en tabla máquina

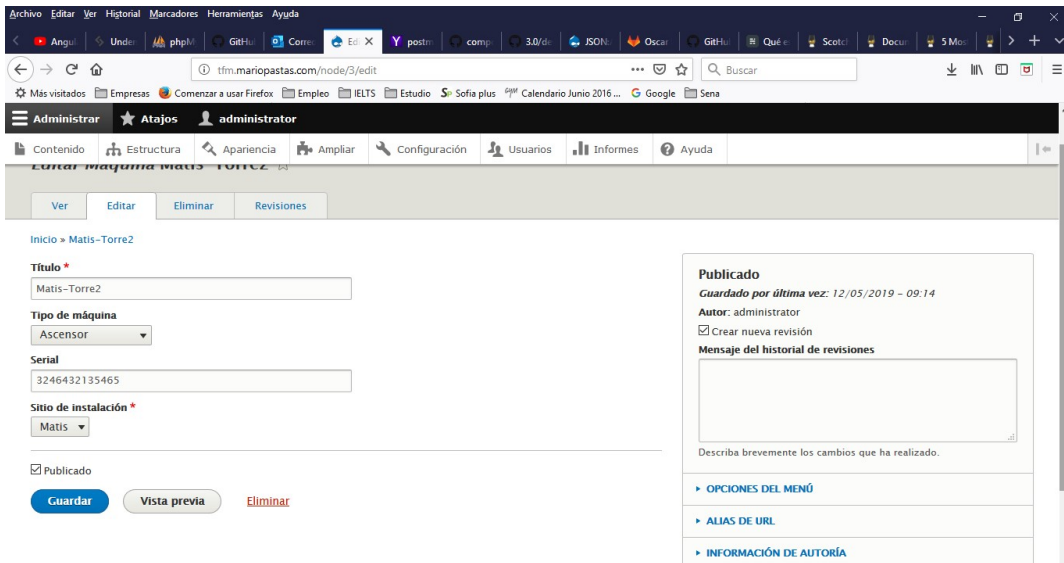


Ilustración 30. Ejemplo de creación de máquina

7.4.4. Tabla de sitios:

Dentro de la tabla sitios se incluye de acuerdo al diseño: Nombre y apellido de persona de contacto, dirección, teléfono, correo electrónico.

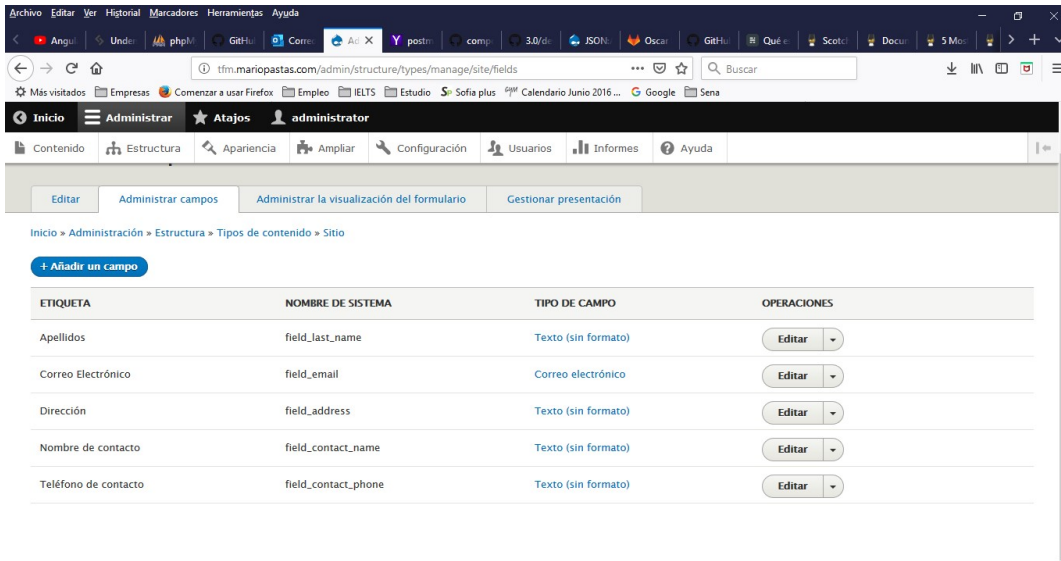


Ilustración 31. Creación y configuración de campos en la tabla Sitio.

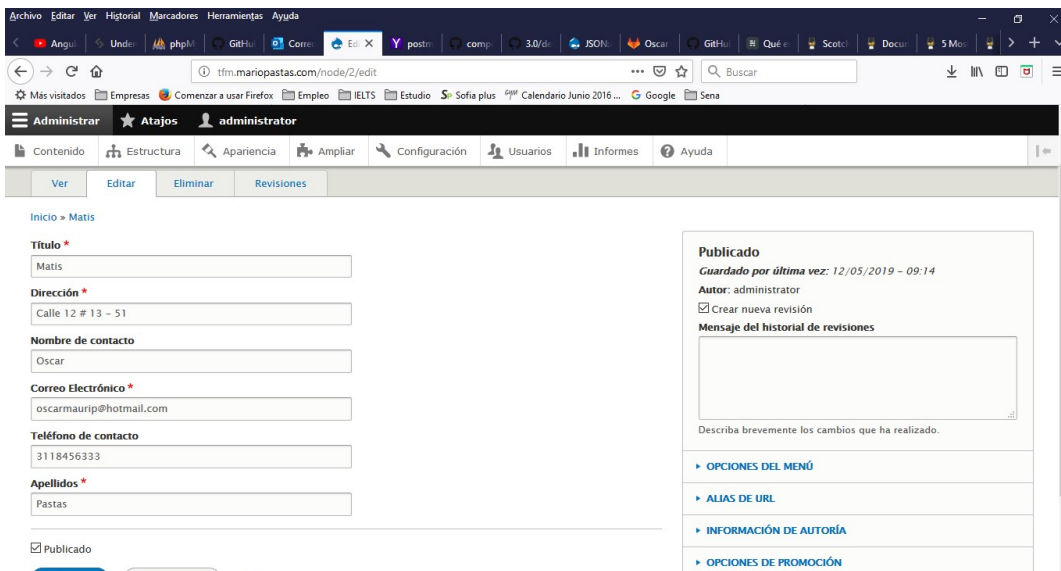


Ilustración 32. Ejemplo de sitio creado

7.5. Contenido de prueba para la aplicación

Se ingresaron diferentes valores para realizar la simulación de los datos dentro de la aplicación, a través del usuario administrador de contenido desde la aplicación web:

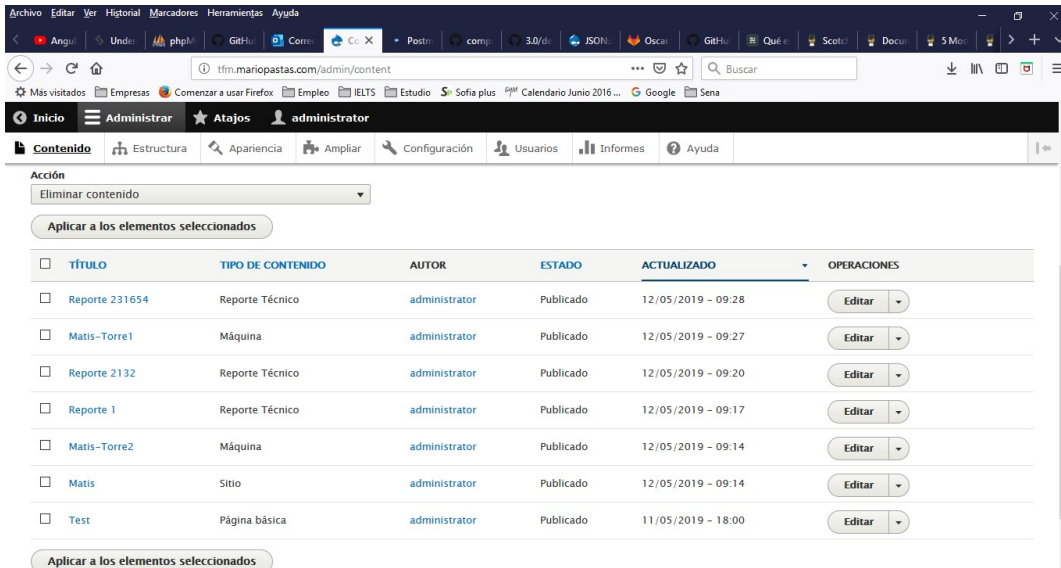


Ilustración 33. Ejemplos de creación de reportes, sitios y máquinas

7.6. Servicios REST y JSON API

Los servicios REST y JSON API se configuraron a partir del menú presente, luego de realizar la verificación por consola de la instalación.

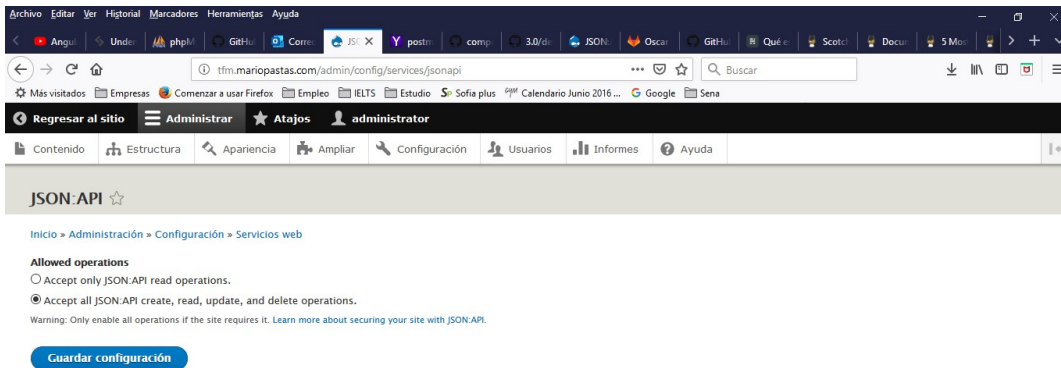


Ilustración 34. JSON API

RESOURCE NAME	RUTA	DESCRIPCIÓN	OPERACIONES
Acción (read-only)	/entity/action/{action}: GET		Activar
Almacenamiento de campo (read-only)	/entity/field_storage_config/{field_storage_config}: GET		Activar
Archivo	/entity/file/{file}: GET, PATCH, DELETE /entity/file: POST		Activar
Bloque (read-only)	/entity/block/{block}: GET		Activar
Bloque personalizado	/block/{block_content}: GET, PATCH, DELETE /block: POST		Activar
Campo (read-only)	/entity/field_config/{field_config}: GET		Activar
Campo base sobrescrito (read-only)	/entity/base_field_override/{base_field_override}: GET		Activar
Cargar archivos	/file/upload/{entity_type_id}/{bundle}/{field_name}: POST		Activar
Clave (read-only)	/entity/key/{key}: GET		Activar

Ilustración 35. Habilitación de servicios REST¹⁶

7.7. Herramienta Postman:¹⁷

Para simulación de las operaciones de CRUD, se empleó una herramienta denominada Postman. Basta con conectar a la aplicación web (mediante el dominio) y la herramienta permite revisar la respuesta a las diferentes peticiones realizadas, para análisis del archivo JSON y configuración de la aplicación realizada en Android.

Ejemplo de petición GET de prueba realizada:

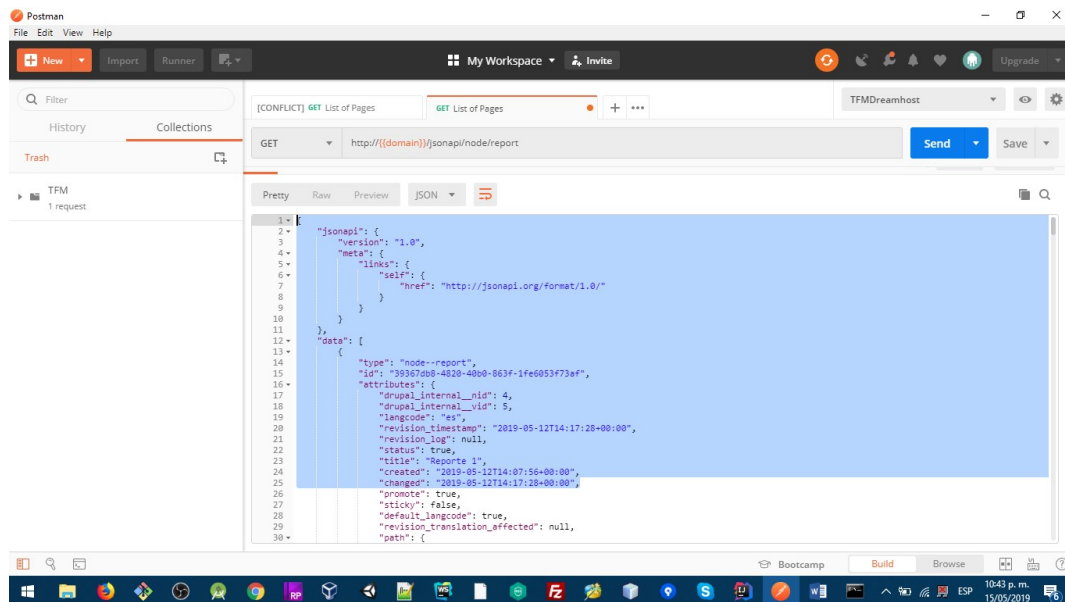


Ilustración 36. Petición GET en Postman

¹⁶ Ref.: Restful Web Services (2019) [en línea]. [Fecha de consulta: 18 de mayo de 2019]<
<https://www.drupal.org/docs/8/core/modules/rest/overview>>

¹⁷ Ref.: Postman (2019) [en línea]. [Fecha de consulta: 20 de mayo de 2019]<
<https://www.getpostman.com/>>

En el análisis de la petición realizada para la estructuración de servicios REST, se puede observar que el archivo JSON, contiene datos, agrupados con atributos id, tipo, “attributes”, dentro del cual se encuentra el título de reporte de prueba: “Reporte 1”. El campo body, presenta un valor de: “p>La máquina fue atendida, y se dejó operando correctamente. Se restablece energía y se programa nuevamente.</p>” con etiquetas HTML.

La herramienta permite realizar operaciones de tipo POST, GET, PUT, PATCH, DELETE, etc, útiles en el desarrollo de la aplicación:

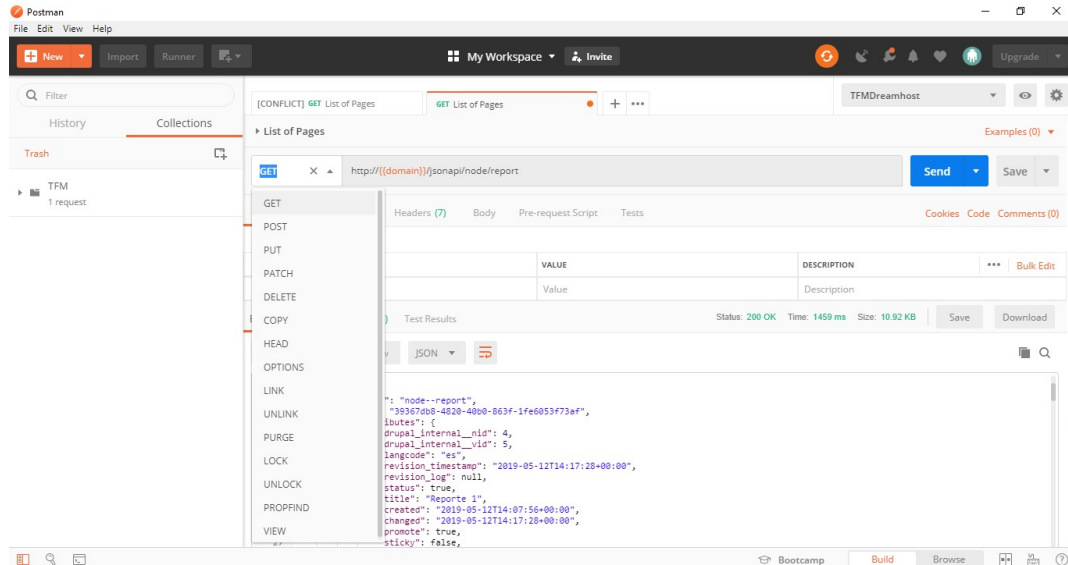


Ilustración 37. Simulación de operaciones GET, POST, etc. en Postman

8. Implementación de aplicación en Android Studio

7.1. Configuración de Android Studio:

Para la configuración y prueba en Android Studio se utilizó la consola de Firebase con el fin de simular una base de datos, muy parecida a la que se encuentra en el servidor web con CMS Drupal.

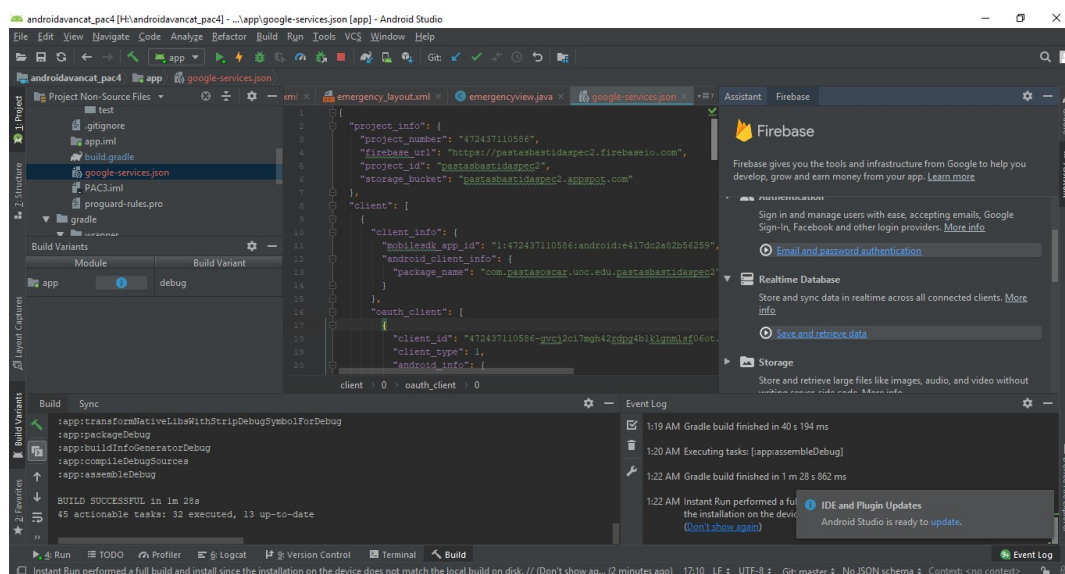


Ilustración 38. Configuración de Android Studio – simulación con Firebase Auth y RealtimeDatabase¹⁸

Las pruebas iniciales fueron realizadas en un dispositivo Moto G4, API24

7.2. Pantalla de ingreso

La programación de pantalla de ingreso se realizó mediante la verificación de campos user-email y password ingresados por el usuario. El programa revisa si el e-mail contiene "@" y si el password es de más de 4 dígitos.

Como usuario de prueba se utilizó:

User-e-mail: opastas@uoc.edu

Password: uoc1234

¹⁸ Ref.: *Firebase Console* (2019) [en línea]. [Fecha de consulta: 17 de mayo de 2019]<
<https://firebase.google.com/?hl=es>>

Solamente de esta forma se concede el acceso a la aplicación desde previa solicitud de datos a un servidor remoto, en este caso Firebase Authentication de Google.

Lo anterior se realiza con fines de prueba. Finalmente dentro de las modificaciones realizadas, se obtiene el acceso mediante “Autenticación básica” en la url:

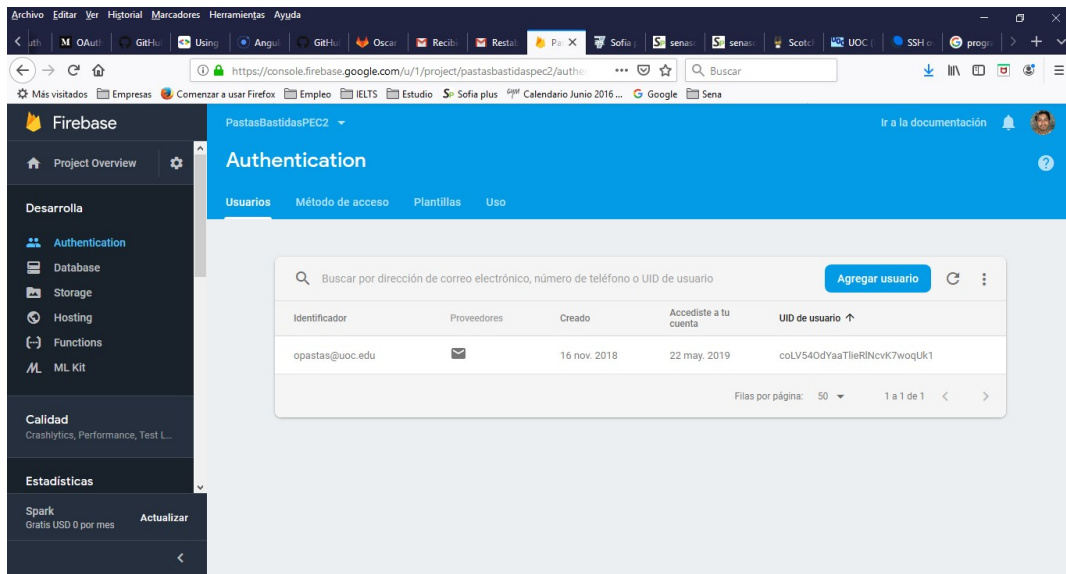


Ilustración 39. Configuración de Firebase – Authentication con usuario opastas@uoc.edu

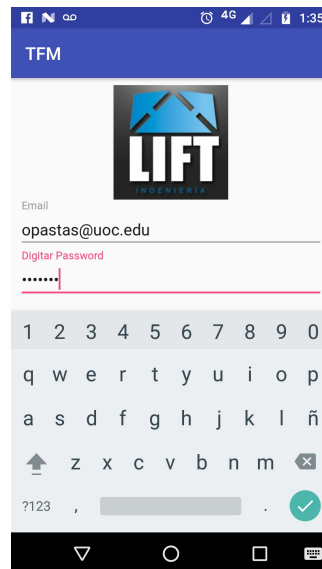


Ilustración 40. Pantalla de ingreso

Dentro de esta pantalla al estar en el campo e-mail, el teclado muestra el símbolo “@” y al oprimir la tecla “enter” pasa automáticamente al campo de digitar password, que una vez ingresado, con la tecla de símbolo “√” permite continuar con la validación.

7.2. Pantalla de menú principal

Al ingresar adecuadamente desde la pantalla de ingreso, la app conduce al usuario hasta el menú principal, en este caso para navegar dentro de las diferentes alternativas:

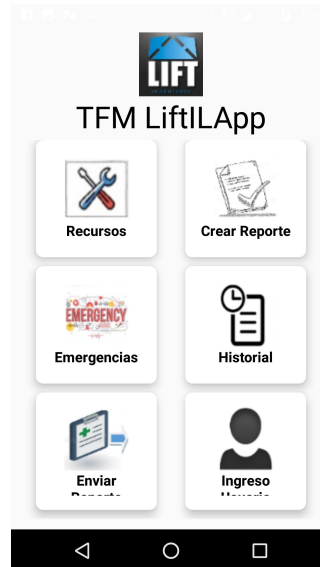


Ilustración 41. Pantalla de Menú Principal

7.3. Pantalla de recursos

A través de ésta sección, se verifican los recursos asignados a cada uno de los usuarios en las diferentes máquinas donde se ha realizado alguna actividad.

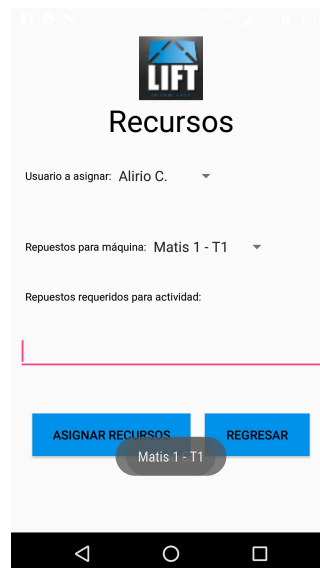


Ilustración 42. Pantalla de recursos

Muestra los campos de usuario/técnico, nombre de máquina, y los recursos empleados en la última tarea. Las opciones seleccionadas son resaltadas mediante un Toast.

7.4. Pantalla de Crear Reporte

Mediante esta pantalla se programará una actividad para el personal técnico. Los datos de prueba fueron simulados desde Android.



Ilustración 43. Pantalla de crear reporte

Una vez creado el reporte, la app confirma con un Toast la creación del evento.

7.5. Pantalla de emergencias

Al seleccionar emergencias, el usuario observará las diferentes fallas que ha presentado una máquina en diferentes ocasiones, con lo cual se podrá realizar un seguimiento adecuado.

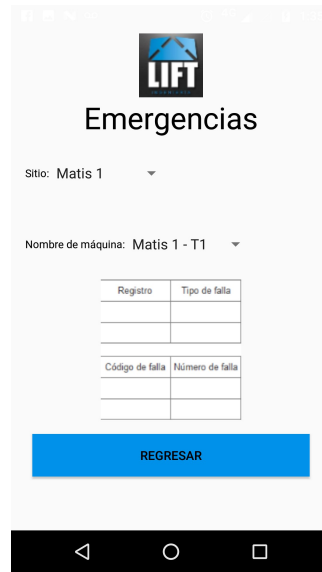


Ilustración 44. Pantalla de emergencias

7.6. Pantalla de Historial

El historial constituye una forma de verificar las actividades realizadas en las diferentes máquinas,

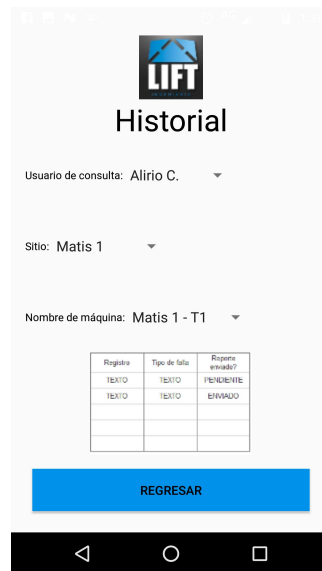


Ilustración 45. Pantalla de historial

7.7. Icono de Ingreso

Mediante la tecla “back” de Android OS o utilizando el icono de ingreso, se regresa al inicio de la aplicación, para realizar el proceso de autenticación nuevamente con otro usuario.

7.8. Uso de librería Retrofit¹⁹

Esta librería se utiliza para convertir la API HTTP en una interfaz Java. En este caso se utilizó con solicitudes y peticiones mediante un servicio al servidor con Drupal.

La declaración de la interfaz se realiza a través de anotaciones, por ejemplo:

```
@GET("users/list")
```

¹⁹ Ref.: *RETROFIT* (2019) [en línea]. [Fecha de consulta: 17 de mayo de 2019]<
<https://square.github.io/retrofit/>>

9. Conclusiones

- Ha sido fundamental realizar la implementación mediante un CMS que tuviera opción de manejar REST API y JSON API para lograr la comunicación adecuada con un cliente como la aplicación diseñada. El proceso de autenticación se ha logrado mediante la opción básica, pudiéndose mejorar la seguridad, modificando el proceso a opción OAuth.
- Durante el desarrollo del trabajo se analizan diferentes alternativas de librerías para serialización/deserialización JSON, siendo Retrofit, adecuada al proyecto consiguiendo los resultados esperados para las peticiones/envíos de datos entre cliente (aplicación móvil) y servidor (base de datos + web + CMS).
- Se cumple con el objetivo de mejora de gestión del mantenimiento, se capacita a los usuarios finales. Mediante un seguimiento al sistema, la gestión del mantenimiento permitirá ser mejorada y medida luego de incluir gráficas de seguimiento, así como mejoras para los diferentes usuarios. La aplicación ha tenido gran aceptación en el contexto empresarial, actualmente se encuentra en fase de ejecución y mejoras.
- El sistema/arquitectura puede considerarse multiplataforma, para lo cual a futuro existe perspectiva de crecimiento de la arquitectura con un desarrollo nativo para iOS ya sea híbrido o web y/o para soporte de otros dispositivos. Actualmente el desarrollo se logró para PC/laptop y dispositivos Android. Adicionalmente puede llevarse a cabo un desarrollo front-end sobre laptop/desktop para un mejor servicio y alternativa al cliente Lift IL.
- A través de esta arquitectura, se puede implementar el trabajo llevado a cabo en otras empresas relacionadas con las modificaciones requeridas. Por supuesto, se llevaría a cabo de la misma forma en que se ha desarrollado el TFM, teniendo en cuenta un perfil de usuario, escalabilidad del sistema, número de usuarios, requerimientos, plataforma disponible, viabilidad económica.
- Se cumplen los diferentes objetivos específicos mencionados. La implementación de usuario normal y administrador, se ha dejado dentro de una sola interfaz de app, por facilidad en la capacitación del usuario. Sin embargo para el futuro desarrollo se tendrá un usuario administrador y un usuario normal. Las funcionalidades que pueden ser implementadas son: Administración y creación de elementos mediante fragments y envío de datos al servidor, registro de nuevos usuarios. Al tener una gran cantidad de datos, se debe modificar las peticiones y respuestas REST dado que se tiene la posibilidad de utilizar filtros

adecuados para optimizar el diseño. Lo anterior es parte de la fase de mejoramiento de la aplicación.

- El desarrollar el curso de Master y específicamente el TFM me ha permitido un crecimiento personal y profesional enorme, teniendo una gran oportunidad de cambio de carrera, gracias a los docentes y UOC.

10. Glosario

API: f. Sigla de Application Program Interface. Interfaz expuesta para ser utilizada dentro de una aplicación con el objetivo de dar acceso a librerías o funciones externas a la aplicación.²⁰

App, Aplicación: Software desarrollado para ser utilizado por un dispositivo móvil.

DCU: m. Sigla de Diseño centrado en usuario: Metodología de diseño.

IDE: m. Sigla de Integrated Development Environment. Entorno de desarrollo que incorpora todas, o casi todas, las herramientas necesarias (herramientas de modelado o diseño, herramientas de debugging, etc.)²¹

TFM: Sigla de Trabajo final de Master.

MVC: Modelo vista controlador

²⁰ Tomado de: FLAMARICH, Jordi, Conceptualización PID_00245395 (2018) [en línea]. [Fecha de consulta: 20 de febrero de 2019] <http://materials.cv.uoc.edu/daisy/Materials/PID_00245395/pdf/PID_00245395.pdf>

²¹ Idem

11. Bibliografía

- [1] *Convenciones Formales Referencias Bibliográficas - UOC* (2018) [en línea]. [Fecha de consulta: 21 de febrero de 2019] <<https://www.uoc.edu/portal/ca/servei-linguistic/convencions/referencies-bibliografiques/criteris/index.html>>
- [2] **SUAREZ F., VILLARRAGA J.** (2017), IMPLEMENTACIÓN DE UNA APLICACIÓN PARA LA GESTIÓN DEL MANTENIMIENTO EN LA CAR <http://repository.udistrital.edu.co/bitstream/11349/7642/1/Su%C3%A1rezPac hecoGerm%C3%A1nFelipe2018.pdf>, Universidad Distrital Francisco J. de Caldas, Bogotá.
- [3] *Protecnus Software de mantenimiento preventivo y correctivo* [en línea]. [Fecha de consulta: 21 de febrero de 2019] <<https://www.protecnus.com/>>
- [4] *App para gestión de órdenes y mantenimiento por industrias* [en línea]. [Fecha de consulta: 21 de febrero de 2019] <<http://www.marianobacastorni.com/tendencias/app-para-gestion-de-ordenes-y-mantenimiento-por-industrias/>>
- [5] **FLAMARICH, J.** (2018), *Conceptualización PID_00245395*, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 21 de febrero de 2019] <http://materials.cv.uoc.edu/daisy/Materials/PID_00245395/pdf/PID_00245395.pdf>
- [6] **FLAMARICH, J.** (2018), *Diseño y prototipado PID_00245397*, Universitat Oberta de Catalunya.
- [7] **SAENZ, N.; VIDAL R.**, Redacción de textos científico-técnicos P08/89018/00445, Universitat Oberta de Catalunya.
- [8] **ROSALES J.** (2017), APLICACIÓN MÓVIL PARA LA GESTIÓN DE MANTENIMIENTOS, Revista Iberoamericana de producción académica y gestión educativa (<http://www.pag.org.mx/index.php/PAG/article/view/679> - Vol. 4, Número 8, ISSN: 2007-8412), Universidad Tecnológica General Mariano Escobedo.
- [9] **CHAVARRI, M.**, (2013), Formación e-learning dirigida a profesionales de la instalación y el mantenimiento, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 21 de febrero de 2019] <<http://hdl.handle.net/10609/24103>>
- [10] **CANALES, J.**, (2017), Wiki para planes de calidad en servicios de mantenimiento, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 21 de febrero de 2019] <<http://hdl.handle.net/10609/59786>>
- [11] **DÍAZ, E.**, (2019), Aplicación de página única basada en diseño centrado en el usuario y XP, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 21 de febrero de 2019] <<http://hdl.handle.net/10609/88466>>

- [12] **GÓMIZ, J.**, (2019), Desarrollo de aplicación con PhoneGap: PlaneaTe, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 21 de febrero de 2019] < <http://hdl.handle.net/10609/89165>>
- [13] **GUTIERREZ, S.**, (2012), Aplicación web para la gestión de facturación de una empresa, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 21 de febrero de 2019] < <http://hdl.handle.net/10609/14686>>
- [14] **MARTORELL, J.**, (2010), Gestió del Manteniment, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 21 de febrero de 2019] < <http://hdl.handle.net/10609/426>>
- [15] **BLÁZQUEZ, O.**, (2013), Mantenimiento preventivo, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 21 de febrero de 2019] <http://hdl.handle.net/10609/23043>
- [16] **GONZÁLEZ, P.**, (2015), Implantación de un sistema ERP en una PyME, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 21 de febrero de 2019] < <http://hdl.handle.net/10609/42924>>
- [17] **MARTÍ, C.** (2015), Desenvolupament d'una aplicació per a dispositius mòbils per visualitzar l'agenda d'activitats de la web www.festamajor.biz, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 21 de febrero de 2019] < <http://hdl.handle.net/10609/39882>>
- [18] **LÓPEZ, M.** (2018), ENERGY.home. Control i monitorització del pla de manteniment preventiu de legionel·losis i instal·lacions d'energia solar tèrmica als edificis esportius, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 21 de febrero de 2019] < <http://hdl.handle.net/10609/81985>>
- [19] **ROSELL, D.** (2018), QualityControl, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 22 de febrero de 2019] < <http://hdl.handle.net/10609/81545>>
- [20] **PERALES, A.** (2017), Aplicació web per a un taller mecànic, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 22 de febrero de 2019] < <http://hdl.handle.net/10609/58665>>
- [21] **CAMPOS, J.; JANTUNEN, E.; PRAKASH, O.** (2009), A web and mobile device architecture for mobile e-maintenance, Springer-Verlag (<https://doi.org/10.1007/s00170-009-1942-x>, ISSN: 0268-3768), London.
- [22] **EMMANOULIDIS, C.; KATSIKAS, S.; GIORDAMLIS, C.** (2008), WIRELESS CONDITION MONITORING AND MAINTENANCE MANAGEMENT: A REVIEW AND A NOVEL APPLICATION DEVELOPMENT PLATFORM, CETI Institute (<http://www.ipet.gr/~chrism/files/emmanouilidiswceam2008.pdf>), Greece.

- [23] **LEGNER, C.; NOLTE, C.; URBACH, N.** (2011), EVALUATING MOBILE BUSINESS APPLICATIONS IN SERVICE AND MAINTENANCE PROCESSES: RESULTS OF A QUANTITATIVE-EMPIRICAL STUDY, ECIS 2011 Proceedings. 247 (<http://aisel.aisnet.org/ecis2011/247>)
- [24] **YOUNG-DO J.** (2013), Implementation of Facility Maintenance Management System using Smart Phones, JIIBC 2013 (<http://dx.doi.org/10.7236/JIIBC.2013.13.1.191>, Vol. 13, No. 1, pp. 191-197, ISSN: 2289 - 0246)
- [25] **MACCHI, M.; CRESPO, A.; HOLGADO M.; FUMAGALLI, L.; BARBERA, L.** (2014), Value-driven engineering of E-maintenance platforms, Journal of Manufacturing Technology Management (<https://doi.org/10.1108/JMTM-04-2013-0039>, Vol. 25 Issue: 4, pp.568-598)
- [26] **CAMPOS, J.** (2014), Current and prospective information and communication technologies for the emaintenance applications, Journal of Quality in Maintenance Engineering (<https://doi.org/10.1108/JQME-05-2014-0029>, Vol. 20 Issue: 3, pp.233-248)
- [27] **DHILLON B. (2002)**, Engineering Maintenance – a modern approach (2da edición), ISBN: 978-1-4200-3184-3, CRC Press.
- [28] *Análisis DAFO* (2019) [en línea]. [Fecha de consulta: 25 de febrero de 2019] <https://es.wikipedia.org/wiki/An%C3%A1lisis_DAFO>
- [29] *Matriz DOFA* (2019) [en línea]. [Fecha de consulta: 25 de febrero de 2019] <<https://es.slideshare.net/remyor09/cmo-construir-una-matriz-dofa>>
- [30] *Gerencia de Mercadeo* (2019) [en línea]. [Fecha de consulta: 25 de febrero de 2019] <<http://3w3search.com/Edu/Merc/Es/GMerc065.htm>>
- [31] *Lift Maintenance* (2019) [en línea]. [Fecha de consulta: 22 de febrero de 2019]<<https://play.google.com/store/apps/details?id=com.radioknit.liftmaintenance>>
- [32] *Happy Service* (2019) [en línea]. [Fecha de consulta: 22 de febrero de 2019] <<https://play.google.com/store/apps/details?id=com.happyservice>>
- [33] *AppSheet* (2019) [en línea]. [Fecha de consulta: 22 de febrero de 2019] <<https://www.appsheet.com/>>
- [34] *Easy Maintenance Reminder* (2019) [en línea]. [Fecha de consulta: 22 de febrero de 2019] <<https://play.google.com/store/apps/details?id=com.easy.maintenancereminder>>

- [35] *EMS Maintenance* (2019) [en línea]. [Fecha de consulta: 23 de febrero de 2019] <<https://play.google.com/store/apps/details?id=com.maintenancesolution.ems>>
- [36] **RAMIREZ, R.; BOLTÀ H.** (2016), Métodos para el desarrollo de aplicaciones móviles PID_ 00246016, Universitat Oberta de Catalunya - UOC.
- [37] **BAUMEISTER, H.; LICHTER, H.; RIEBISCH, M.** (2017), *Agile Processes in Software Engineering and Extreme Programming*, ISBN 978-3-319-57632-9, Springer Open.
- [38] **Redding, L.; Roy R.** (2015), *Through-life Engineering Services*, ISBN: 978-3-319-12110-9, Springer.
- [39] **LUQUE, I.**; (2011), Bases de datos.
- [40] **Silberschatz, A.; Korth, S.; Sudarshan, S.** (2006), *Fundamentos de Bases de datos*, Ed. Mc Graw-Hill, 5ta ed.
- [41] **PARDO, S.; SERNA S.** (2017), *Diseño de Interfaces de aplicaciones móviles*, ISBN: 9789587626889, Ra – Ma.
- [42] *JSON API - Drupal* (2019) [en línea]. [Fecha de consulta: 20 de abril de 2019] < <https://www.drupal.org/docs/8/modules/jsonapi/jsonapi> >
- [43] *BOX Scotch* (2019) [en línea]. [Fecha de consulta: 20 de abril de 2019] < <https://box.scotch.io/> >
- [44] *Vagrant - Aplicaciones* (2019) [en línea]. [Fecha de consulta: 22 de abril de 2019] < <https://codeandoando.com/que-es-vagrant-y-como-usarlo/> >
- [45] *Composer* (2019) [en línea]. [Fecha de consulta: 23 de abril de 2019] < <https://styde.net/que-es-composer-y-como-usarlo/>>
- [46] **JULIA, D.**, (2012), *Sistemas de gestión de contenidos*, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 20 de abril de 2019] < <http://hdl.handle.net/10609/63046>>
- [47] **VENDRELL, V.**, (2013), *Potenciació de la línia de negoci de www.guiacomerciolocal.es mitjançant una aplicació Android*, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 20 de abril de 2019] < <http://hdl.handle.net/10609/22786>>
- [48] **CIVIT, A.**, (2013), *Creació d'un portal web per a un hotel rural usant el CMS Drupal*, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 20 de abril de 2019] < <http://hdl.handle.net/10609/22364>>

- [49] **SERRANO, J.; RICO, O.**, (2013), Creació d'un portal web per a un hotel rural usant el CMS Drupal, Universitat Oberta de Catalunya [en línea]. [Fecha de consulta: 20 de abril de 2019] < <http://hdl.handle.net/10609/1706>>
- [50] *Postman* (2019) [en línea]. [Fecha de consulta: 20 de mayo de 2019]< <https://www.getpostman.com/>>
- [51] *Firestore Console* (2019) [en línea]. [Fecha de consulta: 17 de mayo de 2019]< <https://firebase.google.com/?hl=es>>
- [52] *Restful Web Services* (2019) [en línea]. [Fecha de consulta: 18 de mayo de 2019]< <https://www.drupal.org/docs/8/core/modules/rest/overview>>
- [53] *Drupal* (2019) [en línea]. [Fecha de consulta: 18 de mayo de 2019]< <https://www.drupal.org/>>
- [54] *JSON Api Specification* (2019) [en línea]. [Fecha de consulta: 21 de mayo de 2019]< <https://dev.acquia.com/blog/decoupling-drupal-8-with-json-api/15/05/2018/19596>>
- [55] *Retrofit Tutorial — Send Objects In Request Body* (2019) [en línea]. [Fecha de consulta: 21 de mayo de 2019]< https://www.youtube.com/watch?v=j7IRiTJ_-cl&t=380s>
- [56] *Online JSON Viewer* (2019) [en línea]. [Fecha de consulta: 22 de mayo de 2019] < <http://jsonviewer.stack.hu/>>
- [57] *Generate Plain Old Java Objects from JSON* (2019) [en línea]. [Fecha de consulta: 22 de mayo de 2019] < <http://www.jsonschema2pojo.org/>>
- [58] *Creating new resources (POST)* (2019) [en línea]. [Fecha de consulta: 22 de mayo de 2019] < <https://www.drupal.org/node/2806633>>
- [59] *Retrofit — Add Custom Request Header* (2019) [en línea]. [Fecha de consulta: 25 de mayo de 2019] < <https://futurestud.io/tutorials/retrofit-add-custom-request-header>>
- [60] *How to POST to a RESTfull endpoint in Drupal 8 with basic authentication* (2019) [en línea]. [Fecha de consulta: 27 de mayo de 2019] < <https://blog.karmacomputing.co.uk/how-to-post-to-a-restfull-endpoint-in-drupal-8-with-basic-authentication/>>
- [61] *How to parse json using Retrofit 2, Android* (2019) [en línea]. [Fecha de consulta: 25 de mayo de 2019] < <https://www.freshbytelabs.com/2018/04/how-to-parse-json-using-retrofit-2.html>>
- [62] *Edición gratuita de vídeos – VSDC* (2019) [en línea]. [Fecha de consulta: 05 de junio de 2019] < <https://www.youtube.com/watch?v=woxc4L95mfg>>

12. Anexos

A. PERFILES DE USUARIO:



Ilustración 46. Perfil 1. Fuente: Propia



Ilustración 47. Perfil 2. Fuente: Propia

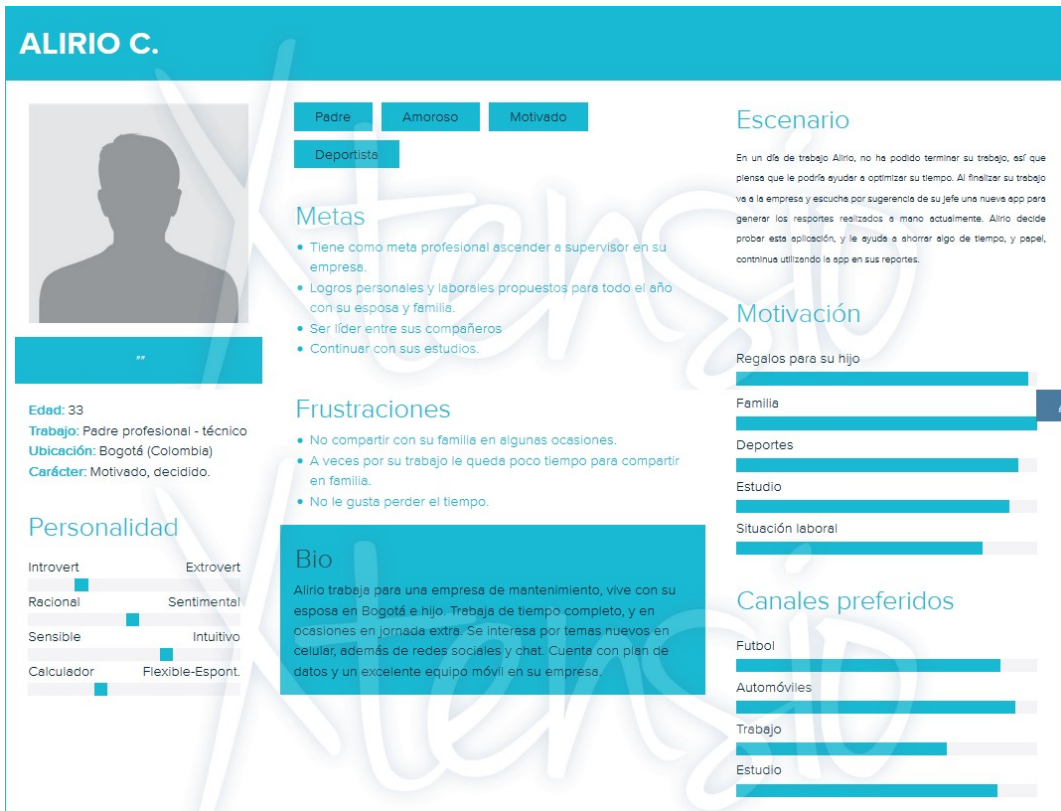


Ilustración 48. Perfil 3. Fuente: Propia

B. ENTREVISTA, ENCUESTA Y RESULTADOS

B. 1. FICHA ENCUESTA APLICADA Y RESULTADOS:

Preguntas realizadas en la encuesta de aplicaciones móviles de mantenimiento:

1. Qué aspectos le interesaría en una aplicación móvil de mantenimiento?
 - a. Reportes
 - b. Programación de recursos
 - c. Facilidad de manejo y experiencia de usuario
 - d. Estadísticas que pueda suministrar la aplicación
 - e. Todas las anteriores características
 - f. No está interesado

2. Piensa que una aplicación móvil le ayudaría a agilizar la entrega de reportes a clientes de su empresa?
 - a. Si
 - b. No

3. Considera que las aplicaciones móviles podrían ser una herramienta eficaz de mantenimiento y programación de recursos para su empresa?
 - a. Si
 - b. No

4. Piensa que una aplicación de este tipo es adecuada para personas en el rango de edad:
 - a. 15-20
 - b. 20-40
 - c. 40 en adelante
 - d. todas las anteriores

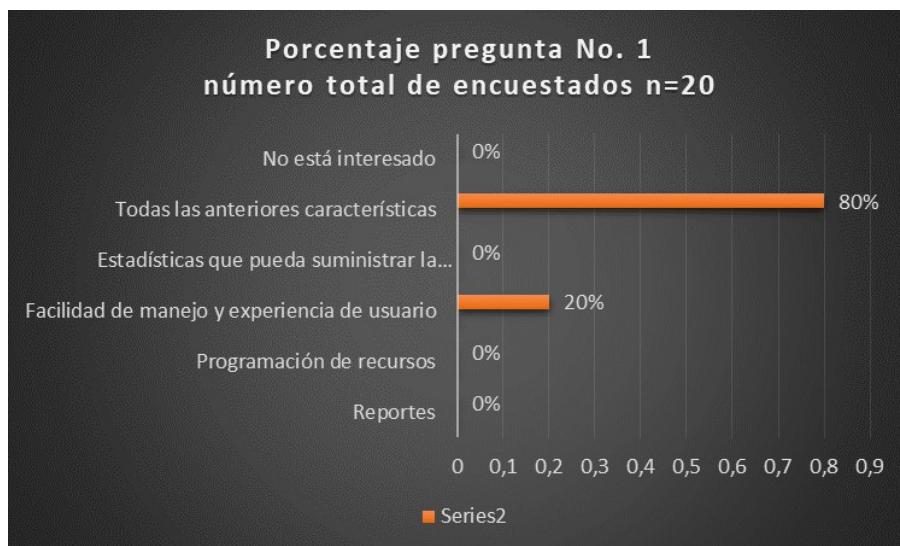
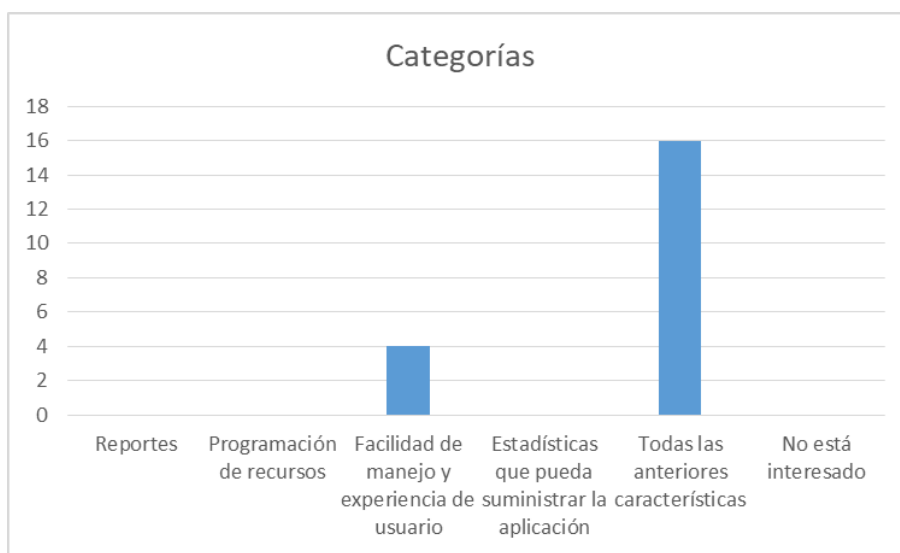
5. Cuenta con un dispositivo móvil y acceso de datos fijo en su empresa:
 - a. Si
 - b. No

6. ¿En qué medio consulta cuando tiene inquietudes de mantenimiento?
 - a. Medio especializado en internet
 - b. Busca una aplicación móvil y la instala en su teléfono móvil
 - c. No consulta en ningún sitio

7. Cuanto tiempo dedica diariamente a navegación en redes sociales y aplicaciones instaladas en su teléfono?
 - a. Menos de 2 horas diarias.
 - b. Entre 2 - 4 horas diarias
 - c. De 4 – 6 horas diarias
 - d. Más de 6 horas diarias

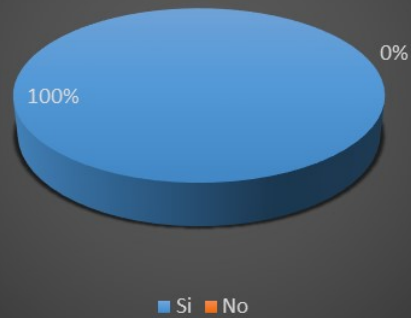
RESULTADOS:

Categoría	Frecuencia
Reportes	0
Programación de recursos	0
Facilidad de manejo y experiencia de usuario	4
Estadísticas que pueda suministrar la aplicación	0
Todas las anteriores características	16
No está interesado	0



App móvil agiliza entrega de reportes?	Frecuencia	Porcentaje
Si	20	100%
No	0	0%

App móvil agiliza entrega de reportes?



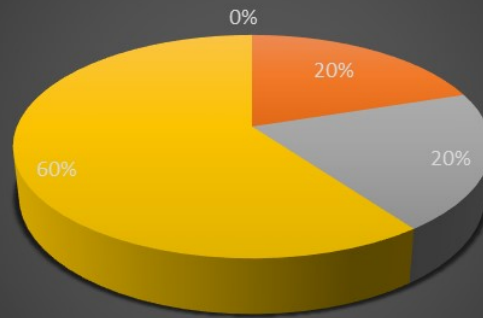
Aplicaciones móviles podrían ser una herramienta eficaz de mantenimiento?	Frecuencia	
Si	20	100%
No	0	0%

Aplicaciones móviles podrían ser una herramienta eficaz de mantenimiento?



Rango de edad adecuado para usar aplicación de mantenimiento	Frecuencia	
15-20	0	0%
20-40	4	20%
40 o mayor	4	20%
Todas las anteriores	12	60%

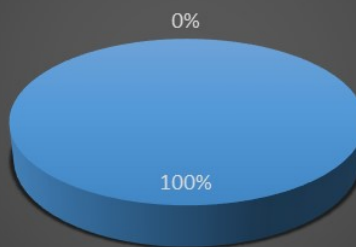
Rango de edad adecuado para usar aplicación de mantenimiento



■ 15-20 ■ 20-40 ■ 40 o mayor ■ Todas las anteriores

Cuenta con un dispositivo móvil y acceso de datos fijo en su empresa?	Frecuencia	
Si	20	100%
No	0	0%

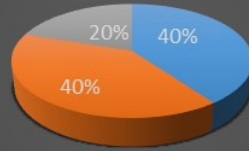
Cuenta con un dispositivo móvil y acceso de datos fijo en su empresa? Frecuencia



■ Si ■ No

¿En qué medio consulta cuando tiene inquietudes de mantenimiento?	Frecuencia	
Medio especializado en internet	8	40%
Busca una aplicación móvil y la instala en su teléfono móvil	8	40%
No consulta en ningún sitio	4	20%

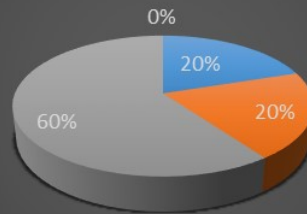
¿En qué medio consulta cuando tiene inquietudes de mantenimiento?



- Medio especializado en internet
- Busca una aplicación móvil y la instala en su teléfono móvil
- No consulta en ningún sitio

Tiempo que se dedica diariamente a navegación en redes sociales y aplicaciones	Frecuencia	
Menos de 2 horas diarias.	4	20%
Entre 2 - 4 horas diarias	4	20%
De 4 – 6 horas diarias	12	60%
Mas de 6 horas diarias	0	0%

Tiempo que se dedica diariamente a navegación en redes sociales y aplicaciones



- Menos de 2 horas diarias.
- Entre 2 - 4 horas diarias
- De 4 – 6 horas diarias
- Mas de 6 horas diarias

B. 2. FICHA ENTREVISTA APLICADA Y RESULTADOS:

1. Conoce apps que permitan el envío de reportes de mantenimiento a través de internet? Le gustaría conocerlas?
2. ¿Crees que se puede desarrollar reportes de mantenimiento a través de una aplicación móvil?

3. ¿Dónde buscarías información para mantenimiento a través de aplicaciones móviles?

4. ¿Invertirías recursos económicos en una aplicación móvil para manejo de mantenimiento?

5- ¿Qué características te interesarían en una aplicación móvil de mantenimiento?

RESULTADOS

PERSONA 1:

1. HAS VISTO EL FUNCIONAMIENTO DE APPS
2. SI, TE INTERESA
3. TIENDA PLAY STORE, INTERNET
4. SI, NO MUCHO
5. FUNCIONALIDAD EN REPORTE: REPORTE SALGA LA ACTIVIDAD A REALIZAR

PERSONA 2:

1. NO CONOCE APLICACIONES ESPECIALES, CORREO ELECTRÓNICO.
2. SI
3. EN INTERNET, PORTALES WEB ESPECIALIZADOS
4. SI, PORQUE LOS DATOS SE ACCEDEN FÁCILMENTE, Y SE ORGANIZAN AUTOMÁTICAMENTE, PERMITIENDO LLEVAR ESTADÍSTICAS.
5. FÁCILMENTE NAVEGABLE, USABLE, PERMITIR AHORRO DE TIEMPO EN DIGITACIÓN DE DATOS. LOS DATOS ESTARÍAN DISPONIBLES Y COMPILADOS EN CUALQUIER MOMENTO. NO SE PERDERÍAN. AHORRO DE PAPEL Y DE ARCHIVOS FÍSICOS.

PERSONA 3:

1. NO CONOCE, PERO ESTARÍA INTERESADO EN CONOCER APLICACIONES MÓVILES
2. SI
3. GOOGLE, PORTALES WEB
4. NO
5. RECORDATORIO DE FECHA, TIPO DE MANTENIMIENTO, RESULTADO DEL MANTENIMIENTO, TIPO DE FALLA.

PERSONA 4:

1. ME GUSTARÍA, NO LAS CONOZCO
2. SI
3. EN TIENDA APP STORE, PLAY STORE, EN INTERNET
4. SI, PARA UNA EMPRESA POR SUPUESTO.
5. FÁCIL USO, GENERACIÓN DE INDICADORES/ESTADÍSTICAS, INGRESO Y CONSULTA FÁCIL DE DATOS.