

UNIVERSIDAD POLITÉCNICA DE PUEBLA

INGENIERÍA EN INFORMÁTICA



PROYECTO DE ESTADIA PROFESIONAL EN INFORMÁTICA

“Desarrollo de aplicación de tiempo de errores en mantenimiento por niveles”

PRESENTA

Luis Eduardo Pérez Ruíz

ASESOR TÉCNICO

Ing. Daniel Cruz Márquez

ASESOR ACADÉMICO

M.A.Nahir González Sosa



ÍNDICE

ÍNDICE DE FIGURAS	3
ÍNDICE DE TABLAS	6
1. CAPÍTULO I. INTRODUCCIÓN	7
1.1 Nombre del proyecto	8
1.2 Planteamiento del problema o necesidad	9
1.3 Justificación	9
1.4 Objetivo general	10
1.5 Objetivos específicos	10
2. CAPÍTULO II. METODOLOGÍA Y HERRAMIENTAS	10
2.3 Herramientas utilizadas	12
2.3.1 Android Studio	13
2.3.2 MySql	13
2.3.3 000webhost	14
2.3.4 Firebase	15
2.3.5 XAMPP	16
3. Capítulo III. RESULTADOS	17
3.1 Exploración	17
3.1.1 Establecimiento de actores:	17
3.1.2 Definición de alcance	18
3.1.3 Establecimiento de proyectos	19
3.1.4 Diseño arquitectónico - Modelo Cliente-Servidor	19
3.2 Iniciación	23
3.2.1 Diseño de interfaces	29
3.2.2 Diseño de base de datos	37
3.2.3 Diccionario de datos	37
3.3 Producto	40
3.3.1 Historia de usuario 1	41
3.3.2 Historia de usuario 2	43
3.3.3 Historia de usuario 3	49
3.3.4 Historia de usuario 4	55
3.3.5 Historia de usuario 5	70
3.3.6 Historia de usuario 6	74
3.3.7 Historia de usuario 7	81
3.3.8 Historia de usuario 8	86
3.3.9 Historia de usuario 9	93
3.4 Estabilización	98
3.4.1 Documentación de usuario	98

Proyecto de Estadía
Laboratorios Texzal de México



3.5 Pruebas	113
3.5.1 Alcance de las pruebas	114
3.5.2 Enfoque de pruebas (Estrategia)	114
3.5.3 Configurar laboratorio de pruebas	114
3.5.4 Casos de prueba aplicados	121
CONCLUSIONES Y RECOMENDACIONES	138
GLOSARIO	139
REFERENCIAS	<i>¡Error! Marcador no definido.</i>

ÍNDICE DE FIGURAS



Proyecto de Estadía Laboratorios Texzal de México

FIGURA 1 CICLO DE DESARROLLO MOBILE-D	11
FIGURA 2 ARQUITECTURA DE SOFTWARE CLIENTE-SERVIDOR.....	20
FIGURA 3 SDK ACTUALIZADO.....	24
FIGURA 4GOOGLE PLAY SERVICES ACTUALIZADO	24
FIGURA 5 SELECCIONAR DISPOSITIVO VIRTUAL	25
FIGURA 6 DISPOSITIVO FUNCIONANDO- HELLO WORLD	26
FIGURA 7 INICIAR SERVIDOR XAMPP.....	27
FIGURA 8 PÁGINA DE PHPMYADMIN.....	27
FIGURA 9 PÁGINA PRINCIPAL DE 000WEBHOST	28
FIGURA 10 MOCKUPLOGIN.....	29
FIGURA 11 MOCKUP REGISTRAR USUARIO.....	30
FIGURA 12 MOCKUP MENÚ PRINCIPAL.....	31
FIGURA 13 MOCKUP REGISTRAR FALLA.....	32
FIGURA 14 MOCKUP FALLAS EN CURSO	33
FIGURA 15 MOCKUP DETALLES FALLA EN CURSO	34
FIGURA 16 MOCKUP LISTA DE FALLAS GUARDADAS	35
FIGURA 17 MOCKUP PERFIL DE USUARIO	36
FIGURA 18 BASE DE DATOS FINAL	37
FIGURA 19 ITERACIÓN DE FASE: PRODUCTO	40
FIGURA 20 HISTORIA DE USUARIO “MANDAR NOTIFICACIÓN LOCAL”	41
FIGURA 21 ITERACIÓN “NOTIFICACIÓN LOCAL”	42
FIGURA 22 BASE DE DATOS LOCAL DE EJEMPLO CON MYSQL-XAMPP	43
FIGURA 23 ARCHIVOS .PHP PARA EL MANEJO DE DATOS LOCALES	44
FIGURA 24 ARCHIVO PARA CONSULTA DE DATOS.....	44
FIGURA 25 ARCHIVO PARA CONSULTA DE DATOS.....	45
FIGURA 26 ARCHIVO PARA CONSULTA DE DATOS.....	45
FIGURA 27 ARCHIVO PARA CONSULTA DE DATOS.....	45
FIGURA 28 EVENTOS DE LOS BOTONES DELA INTERFAZ.....	46
FIGURA 29 FUNCIÓN QUE CREA UNA CONEXIÓN ENTRE LA BASE DE DATOS Y LA APLICACIÓN	47
FIGURA 30 FUNCIÓN LEE LOS DATOS RECIBIDOS DE LA CONSULTA	47
FIGURA 31 ITERACIÓN “CRUD”	48
FIGURA 32LOGIN DE 000WEBHOST	50
FIGURA 33 000WEBHOST- APARTADO MANAGEDATABASES	51
FIGURA 34 000WEBHOST- APARTADO PHPMYADMIN.....	51
FIGURA 35 000WEBHOST- APARTADO FILE MANAGER	52
FIGURA 36 000WEBHOST- APARTADO FILE MANAGER	52
FIGURA 37 MODIFICAR ARCHIVO DE CONEXIÓN DE BASE DE DATOS.....	53
FIGURA38 MODIFICACIONES EN LA APLICACIÓN	53
FIGURA 39 ITERACIÓN “CRUD”	54
FIGURA 40 PÁGINA OFICIAL DE FIREBASE.....	56
FIGURA 41 PÁGINA DE BIENVENIDA DE FIREBASE	56
FIGURA 42 VENTANA PARA CREAR UN PROYECTO	57
FIGURA 43 VENTANA DE CONFIRMACIÓN DE CREACIÓN DE CUENTA.....	58
FIGURA 44 PÁGINA PRINCIPAL DE FIREBASE	59
FIGURA 45 VENTANA PARA AÑADIR UN PROYECTO ANDROID A FIREBASE	60
FIGURA 46 VENTANA PARA AGREGAR GOOGLESERVICES A NUESTRO PROYECTO	61
FIGURA 47 VENTANA PARA AGREGAR DEPENDENCIAS A NUESTRO PROYECTO.....	62
FIGURA 48 CAMBIOS EN DEPENDENCIAS BUILD.GRADLE(MODULE.APP).....	63
FIGURA 49 CAMBIOS EN EL ARCHIVO BUILD.GRADLE(PROJECT.NOTIFICAR)	63



Proyecto de Estadía Laboratorios Texzal de México

FIGURA 50 PÁGINA PRINCIPAL DE FIREBASE	64
FIGURA 51 PANEL LATERAL DE OPCIONES.....	64
FIGURA 52 VISTA PARA MANDAR UN MENSAJE DESDE CONSOLA EN FIREBASE	65
FIGURA 53 BOTÓN “ENVIAR MENSAJE”	65
FIGURA 54 MENSAJE RECIBIDO EN DISPOSITIVO	66
FIGURA 55 PÁGINA DE CONFIGURACIÓN DE FIREBASE CLOUD MESSAGING	66
FIGURA 56 CONFIGURACIÓN DE ARCHIVO “ANDROIDMANIFEST.XML”	67
FIGURA 57 CLASE “MIFIREBASESERVICIOMENSAJE.JAVA”	68
FIGURA 58 CLASE “MIFIREBASEINSTANCE.JAVA”.....	68
FIGURA 59 ITERACIÓN “MANDAR NOTIFICACIÓN A DIFERENTES DISPOSITIVOS”	69
FIGURA 60 MÉTODOS BÁSICOS DE LA CLASE ASYNCTASK.....	70
FIGURA 61 IMPLEMENTACIÓN DE ASYNCTASK 1	71
FIGURA 62 IMPLEMENTACIÓN DE ASYNCTASK 2	71
FIGURA 63IMPLEMENTACIÓN DE ASYNCTASK 3	72
FIGURA 64 ITERACIÓN “TIEMPO DE ESPERA EN NOTIFICACIONES”	73
FIGURA 65 PRIMER VERSIÓN DE BASE DE DATOS	74
FIGURA 66 “SPLASHSCREEN”	75
FIGURA 67 ACTIVIDAD “LOGIN”	76
FIGURA 68 ACTIVIDAD “AGREGAR USUARIO NUEVO”	77
FIGURA 69 ACTIVIDAD “MENÚ PRINCIPAL”	78
FIGURA 70 ACTIVIDAD “ENVÍO DE FALLAS”	79
FIGURA 71 ACTIVIDAD “LISTA DE FALLAS”	80
FIGURA 72 ITERACIÓN “LOGIN + AGREGAR USUARIO + DISEÑO DE INTERFAZ”	81
FIGURA 73 MODIFICAR ACTIVIDAD “LOGIN”.....	82
FIGURA 74 MODIFICAR ACTIVIDAD “ENVIAR FALLA” 1	83
FIGURA 75 MODIFICAR ACTIVIDAD “ENVIAR FALLA” 2.....	84
FIGURA 76 MODIFICAR ACTIVIDAD “MOSTRAR TODAS LAS FALLAS”	85
FIGURA 77 NUEVA BASE DE DATOS.....	87
FIGURA 78 ACTIVIDAD “MOSTRAR FALLA EN CURSO”	87
FIGURA 79 ACTIVIDAD “DETALLES DE FALLA EN CURSO”	88
FIGURA 80 BOTÓN “DETENER ESTA FALLA”	89
FIGURA 81 BOTÓN “CANCELAR ESTA FALLA”	90
FIGURA 82 PÁGINA OFICIAL DE FIREBASE – “SUSCRIBIR UN APP CLIENTE A UN TEMA”	91
FIGURA 83 CAMBIOS AL ARCHIVO PHP - TOPIC 1.....	91
FIGURA 84 CAMBIOS AL ARCHIVO PHP - TOPIC 2.....	92
FIGURA 85 ACTIVIDAD “PERFIL DE USUARIO”	94
FIGURA 86 BOTÓN “CERRAR ESTA SESIÓN”	95
FIGURA 87 BOTÓN “MODIFICAR ESTE USUARIO”	96
FIGURA 88 BOTÓN “ELIMINAR ESTE USUARIO”	97
FIGURA 89 EXPLORADOR DE ARCHIVOS	98
FIGURA 90PANTALLA DE INSTALACIÓN	99
FIGURA 91 PANTALLA DE INSTALACIÓN.....	99
FIGURA 92 PANTALLA DE CARGA DE INSTALACIÓN	100
FIGURA 93 PANTALLA DE INSTALACIÓN EXITOSA	100
FIGURA 94 ACTIVIDAD LOGIN	101
FIGURA 95LOGIN – INTENTO ERRÓNEO	102
FIGURA 96ACTIVIDAD “AGREGAR USUARIO”.....	103
FIGURA 97LOGIN - ACCESO CORRECTO	103
FIGURA 98ACTIVIDAD MENÚ PRINCIPAL.....	104



Proyecto de Estadía Laboratorios Texzal de México

FIGURA 99 ACTIVIDAD ENVIAR FALLA	105
FIGURA 100ACTIVIDAD FALLA EN CURSO Y SUS DETALLES.....	106
FIGURA 101BOTÓN “DETENER ESTA FALLA”.....	107
FIGURA 102BOTÓN “DETENER LA FALLA”	108
FIGURA 103FINALIZAR UNA FALLA.....	109
FIGURA 104CANCELAR FALLA	110
FIGURA 105ACTIVIDAD – PERFIL DE USUARIO, BOTÓN “CERRAR ESTA SESIÓN”	111
FIGURA 106BOTÓN “MODIFICAR ESTE USUARIO”	112
FIGURA 107 BOTÓN “ELIMINAR ESTE USUARIO”	113
FIGURA 108NESUS 5X	115
FIGURA 109 SAMSUNG GALAXY S3.....	116
FIGURA 110 SAMSUNG GALAXY S5.....	117
FIGURA 111 SAMSUNG GALAXY S4	118
FIGURA 112 MOTOROLA G.....	119
FIGURA 113 LG G3 STYLUS	120
FIGURA 114 MENSAJE DE ERROR AL INTENTAR COMPILAR EN VERSIÓN MENOR A 5.1.....	133
FIGURA 115 APLICACIÓN EN EJECUCIÓN EN SAMSUNG GALAXY S3.....	134
FIGURA 116 APLICACIÓN EN EJECUCIÓN EN SAMSUNG GALAXY S5.....	134
FIGURA 117 APLICACIÓN EN EJECUCIÓN EN NEXUS 5X.....	135
FIGURA 118 APLICACIÓN EN EJECUCIÓN EN SAMSUNG GALAXY S4 REAL	135
FIGURA 119 APLICACIÓN EN EJECUCIÓN EN MOTOROLA G REAL.....	136
FIGURA 120 APLICACIÓN EN EJECUCIÓN EN LG STYLE REAL.....	136
FIGURA 121 MANEJO DE ERROR EN FALTA DE CONEXIÓN A INTERNET	137

ÍNDICE DE TABLAS



Proyecto de Estadía Laboratorios Texzal de México

TABLA 1 ACTORES	17
TABLA 2 REQUERIMIENTOS FUNCIONALES	18
TABLA 3 TABLA DE REQUERIMIENTOS NO FUNCIONALES	19
TABLA 4 CRONOGRAMA DE ACTIVIDADES	22
TABLA 5 TABLA "FALLA ACTUAL"	38
TABLA 6 TABLA "FALLAS TABLA"	39
TABLA 7 TABLA "USUARIO"	40
TABLA 8 TABLA "EN LINEA FALLA"	40
TABLA 9 PRUEBA DE ACEPTACIÓN "MANDAR NOTIFICACIÓN LOCAL"	41
TABLA 10 HISTORIA DE USUARIO "CRUD LOCAL"	43
TABLA 11 PRUEBA DE ACEPTACIÓN "CRUD LOCAL"	48
TABLA 12 CRUD DE MANERA REMOTA	49
TABLA 13 PRUEBA DE ACEPTACIÓN "CRUD DE MANETA REMOTA"	54
TABLA 14 HISTORIA DE USUARIO "MANDAR NOTIFICACIÓN A DIFERENTES DISPOSITIVOS"	55
TABLA 15 PRUEBA DE ACEPTACIÓN "MANDAR NOTIFICACIÓN A DIFERENTES DISPOSITIVOS"	69
TABLA 16 HISTORIA DE USUARIO "TIEMPO DE ESPERA EN NOTIFICACIONES"	70
TABLA 17 PRUEBA DE ACEPTACIÓN "TIEMPO DE ESPERA EN NOTIFICACIONES"	72
TABLA 18 HISTORIA DE USUARIO "LOGIN + AGREGAR USUARIO + INTERFAZ DE USUARIO"	74
TABLA 19 PRUEBA DE ACEPTACIÓN "LOGIN + AGREGAR USUARIO + DISEÑO DE INTERFAZ"	80
TABLA 20 HISTORIA DE USUARIO "MODIFICACIONES A INTERFAZ DE USUARIO"	81
TABLA 21 PRUEBA DE ACEPTACIÓN "MODIFICAR INTERFAZ DE USUARIO"	85
TABLA 22 HISTORIA DE USUARIO "FALLAS EN CURSO"	86
TABLA 23 PRUEBA DE ACEPTACIÓN "FALLAS EN CURSO"	92
TABLA 24 HISTORIA DE USUARIO "PERFIL DE USUARIO"	93
TABLA 25 PRUEBA DE ACEPTACIÓN "PERFIL DE USUARIO"	97
TABLA 26 LISTA DE SISTEMAS OPERATIVOS A PROBAR	121
TABLA 27 CASO DE PRUEBA "LOGIN"	122
TABLA 28 MENÚ PRINCIPAL	125
TABLA 29 ENVIAR FALLA	126
TABLA 30 FALLAS EN CURSO	127
TABLA 31 DETALLES DE FALLA EN CURSO	129
TABLA 32 LISTA DE FALLAS	130
TABLA 33 PERFIL DE USUARIO	132

1. CAPÍTULO I. INTRODUCCIÓN



Proyecto de Estadía Laboratorios Texzal de México

Actualmente la gestión de datos y la automatización de procesos se han convertido en uno de los temas más importantes para organizaciones, empresas, escuelas, y todo establecimiento donde se manejen grandes cantidades de datos que, al ser numerosamente alto, sería complicado y costoso en cuanto horas-hombre se refiere, llevarlas a cabo, así como la automatización de procesos que liberan tiempo y mano de obra para la organización.

La finalidad de la Gestión de la información es ofrecer mecanismos que permitieran a la organización adquirir, producir y transmitir, al menor coste posible, datos e informaciones con una calidad, exactitud y actualidad suficientes para servir a los objetivos de la organización. En términos perfectamente entendibles sería conseguir la información adecuada, para la persona que lo necesita, en el momento que lo necesita, al mejor precio posible para toma la mejor de las decisiones. En el momento actual parece indiscutible que el éxito de la empresa no dependerá únicamente de cómo maneje sus activos materiales, sino también de la gestión de los recursos de información. La importancia de este recurso es tal que algunos autores estiman que las organizaciones deben ser consideradas como sistemas de información.

Laboratorios Texzal de México es una empresa pequeña la cual se dedica a la creación de productos químicos para el cuidado personal (cremas para la piel, polladas, etc). Actualmente el departamento de mantenimiento de la empresa tiene que notificar personalmente cuando una falla sucede en uno de los departamentos encargados del proceso de creación de los materiales que tiene a la venta, los cuales son Calidad, Producción y Desarrollo. Se requiere que este proceso de notificación entre de un error entra los usuarios sea de manera automatizada.

Por lo que este proyecto explica el diseño e implementación de una aplicación para dispositivos móviles para el envío de notificaciones remotas con tiempo de espera entre ellas a otros dispositivos con la aplicación instalada, así como guardar las fallas completadas en una base de datos remota. El envío entre notificaciones se realiza por medio del nivel de cada uno de los usuarios registrados en la aplicación. Se busca poder guardar todos los datos necesarios de las fallas que ocurren en la empresa así como poder notificar automáticamente a los demás usuarios cuando una falla es enviada para ser resuelta.

Por lo que en el este capítulo se hablará más a fondo del proyecto y la problemática, así como las razones por las que esta necesidad puede ser cubierta con el desarrollo de un software.

1.1 Nombre del proyecto

Desarrollo de aplicación para el manejo de errores en mantenimiento por niveles.



1.2 Planteamiento del problema o necesidad

Los laboratorios Texzal de México han encontrado dificultades a la hora de manejar distintas alarmas de errores entre los niveles del equipo de mantenimiento. Este departamento consiste en tres niveles los cuales están encargados de solucionar el problema, y funciona de la siguiente manera: cuando sucede un error en alguna maquinaria, lo primero que se hace es notificar personalmente al nivel 1 y 2 sobre el problema. Estos dos primeros niveles intentarán solucionar el problema que se suscitó en un límite de tiempo, ya que pasado ese límite de tiempo, tienen que notificar al nivel 3 para que éste venga a resolverlo. El realizar el proceso como actualmente se lleva a cabo causa los siguientes problemas:

- No existe un registro de los errores, así como información sobre el suceso: cuál es el problema, la fecha en que sucedió el problema, quién notifica ese problema al departamento de mantenimiento, cuánto tarda el departamento en resolver el problema y en qué área sucede el problema.
- Existe pérdida de tiempo al avisar entre niveles cuando un nivel inferior no puede resolver el problema.
- Algunos datos resultan obsoletos al no ser verídicos.

Esta problemática puede ser resuelta con la creación de un software que cubra todas las necesidades del cliente, por lo que se propone el desarrollo de una aplicación móvil orientada a smartphones con sistema operativo Android para aprovechar la versatilidad de un sistema que pueda ser llevado con el personal y registrar una falla al departamento en el instante que sucede.

1.3 Justificación

Actualmente en las empresas se requiere el uso de herramientas informáticas que automaticen ciertos procesos en sus actividades, así como la digitalización y almacenamiento verídica de información, hecho por el cual se decidió tomar este proyecto, ya que actualmente el proceso de notificación de fallas entre el departamento de mantenimiento se hace de manera presencial y no hay registro del evento. Esta situación puede ser resuelta con la implementación de un software que cubra las necesidades de los Laboratorios de Texzal de México.

Esta aplicación móvil que se va a desarrollar permitirá resolver los problemas planteados por el cliente como:

- La notificación no presencial a los distintos niveles de manejo de errores, para ser más específicos, la aplicación permitirá notificar automáticamente a todo personal que tenga la aplicación, además de la información completa del problema.
- Ahorro de tiempo al automatizar la notificación entre niveles con límites de tiempo, aprovechando también el almacenamiento de datos acerca de que errores se presentan junto con sus detalles.



Proyecto de Estadía Laboratorios Texzal de México

Se planea que esta aplicación no solamente vaya dirigida a la empresa Laboratorios Texzal de México, sino que también puede ser utilizada para distintas organizaciones que requieran la automatización de notificaciones a distintos niveles de manera no presencial que estén involucrados en el mismo proceso, así como el almacenamiento de datos en tiempo real.

Puesto que la herramienta va a ser desarrollada para la plataforma Android, podemos utilizar el IDE (Entorno de Desarrollo Integrado) más popular y completo aparte de gratuito en su paquete de herramientas básico. Aparte de eso se necesitará un servicio de almacenamiento en la nube el cual nos permitirá administrar los datos de la aplicación en tiempo real para todos los usuarios que utilicen la aplicación.

Además, esta misma plataforma nos proporcionará almacenamiento de archivos necesarios para el funcionamiento de la aplicación de manera remota. Las herramientas que se utilizarán para el desarrollo de esta aplicación son:

- IDE Android Studio usando programación orientada a objetos en Java
- 000webhost, servidor de hosting gratuito en línea de Apache que contiene MariaDB, PHP y Perl.

1.4 Objetivo general

Desarrollar un software a través de la metodología Mobile-D para mejorar el proceso de tiempo de notificación de errores entre los niveles de mantenimiento.

1.5 Objetivos específicos

1. Definir el alcance del proyecto a través de entrevistas con el cliente.
2. Instalar las herramientas adecuadas para el desarrollo de la aplicación.
3. Implementar la aplicación de manera iterativa usando las historias de usuario.
4. Realizar la documentación de "Manual de usuario".
5. Aplicar distintas pruebas para verificar el funcionamiento correcto de la aplicación.
6. Entrega del producto completo al cliente.

2. CAPÍTULO II. METODOLOGÍA Y HERRAMIENTAS



Resulta siempre de gran importancia tener definido un procedimiento o método para el desarrollo de sistemas, pues de no tenerlo puede ser difícil llevar a cabo con éxito las metas que se fijan, además de pasar por una serie de dificultades que normalmente podrían evitarse o resolverse fácilmente con una metodología adecuada, por lo que en este capítulo se hablará de la metodología de desarrollo de software que se eligió, sus ventajas, desventajas y porque se eligió, así como las herramientas elegidas para el proyecto.

2.1 Metodología a implementar

Para este proyecto se eligió seguir la metodología Mobile-D la cual tiene como objetivo conseguir ciclos de desarrollo muy rápidos en equipos muy pequeños. Fue creado en un proyecto finlandés en 2005, pero sigue estando vigente [5]. Basado en metodologías conocidas pero aplicadas de forma estricta como: Programación Extrema, Metodología Cristal y Proceso de Modelado Unificado [1].

2.1.1 Fases de la metodología

Se compone de distintas fases: exploración, inicialización, fase de producto, fase de estabilización y la fase de pruebas. Cada una tiene un día de planificación y otro de entrega. Se hablará de estas fases a continuación [5] [6]:

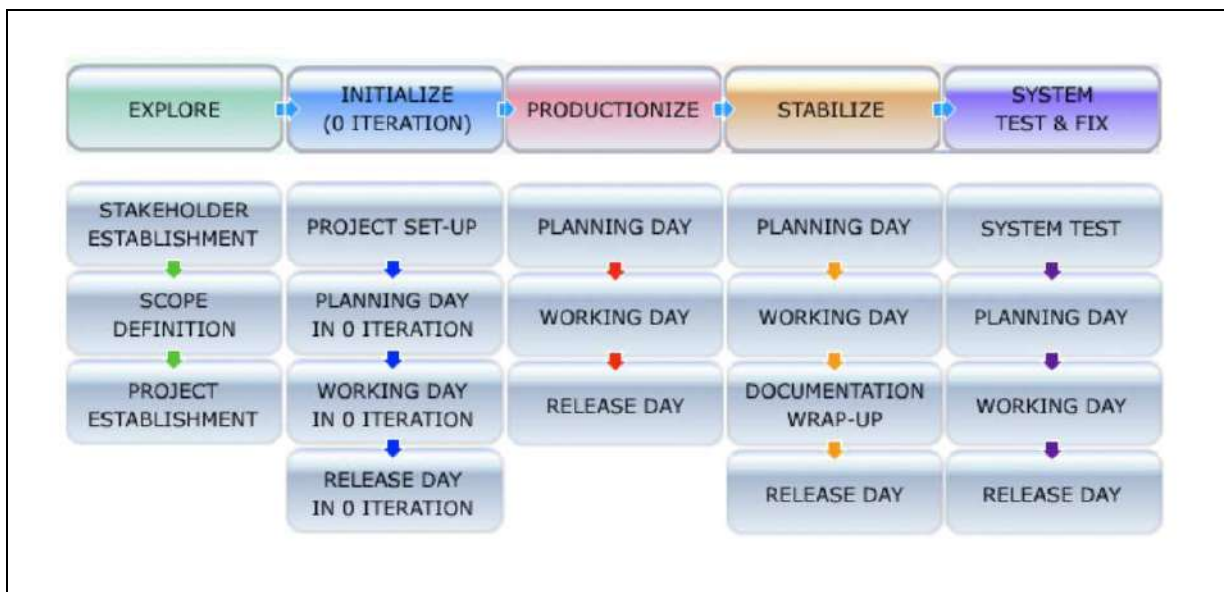


Figura 1 Ciclo de desarrollo Mobile-D



Exploración

En la fase de **exploración** se centra la atención en la planificación y a los conceptos básicos del proyecto. Aquí es donde hacemos una definición del alcance del proyecto y su establecimiento con las funcionalidades donde queremos llegar.

Iniciación

En la **iniciación** configuramos el proyecto identificando y preparando todos los recursos necesarios como hemos comentado anteriormente en esta fase la dedicaremos un día a la planificación y el resto al trabajo y publicación.

Producto

En la fase de **producto** se repiten iterativamente las subfases. Se usa el desarrollo dirigido por pruebas (TDD), antes de iniciar el desarrollo de una funcionalidad debe existir una prueba que verifique su funcionamiento. En esta fase podemos decir que se lleva a cabo toda la implementación.

Estabilización

Después de la fase de producto llega la fase de estabilización en la que se realizan las acciones de integración para enganchar los posibles módulos separados en una única aplicación. También se realiza la documentación de instrucciones para el usuario y especificaciones técnicas [12].

Pruebas

Fase de **pruebas**. Una vez parado totalmente el desarrollo se pasa a una fase de testeo hasta llegar a una versión estable según lo establecido en las primeras fases por el cliente. Si es necesario se reparan los errores, pero no se desarrolla nada nuevo.

2.3 Herramientas utilizadas

El sistema se instalará en Smartphones con sistema operativo Android en su versión 5.0.1 Lollipop con un almacenamiento interno de 16 gb y 1 gb de memoria Ram, suficientes para ejecutar la aplicación sin problema alguno. A continuación se hablará de las herramientas que se utilizarán para el desarrollo del sistema a desarrollar, así como las características que nos ofrece, razones por las cuales fueron seleccionados para este proyecto.



2.3.1 Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android, como las siguientes: [4].



- Un sistema de compilación basado en Gradlw flexible
- Un emulador rápido con varias funciones
- U entorno unificado en el que puedes realizar desarrollos para todos los dispositivos de Android
- InstantRun para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK
- Integración de plantillas de código y GitHub para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código
- Gran cantidad de herramientas y frameworks de prueba
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
- Compatibilidad con C++ y NDK

2.3.2 MySql



MySQL es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, a1 código de programación de MySQL. Todo el mundo puede contribuir para incluir elementos, arreglar problemas, realizar mejoras o sugerir optimizaciones. MySQL ha pasado de ser una "pequeña" base de datos a una completa

herramienta y ha conseguido superar a una gran cantidad de bases de datos comerciales (lo que ha asustado a la mayor parte de 10s proveedores comerciales de bases de datos). Por lo tanto, su rápido desarrollo se debe a la contribución de mucha gente al proyecto, así como a la dedicación del equipo de MySQL [2].

Fue inicialmente desarrollado por MySQL AB la cual fue adquirida por SunMicroSystems en 2008, y está a su vez fue comparada por Oracle Corporation en 2010, la cual ya era



duela desde 2005 de InnobaseOy, empresa filandesa desarrolladora del motor InnoDB para MySql [2].

Características de MySql [2]:

- Uso de multihilos mediante hilos del kernel.
- Completo soporte para operadores y funciones en cláusulas select y where.
- Completo soporte para cláusulas groupby y orderby, soporte de funciones de agrupación.
- Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- Soporta gran cantidad de datos. MySql Server tiene bases de datos de hasta 50 millones de registros.
- Se permiten hasta 64 índices por cada tabla. Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes.
- Disponibilidad en gran cantidad de plataformas y sistemas.

2.3.3 000webhost



000webhost es uno de los hosting web más grandes de la red, especialmente en su modalidad gratuita. Esta plataforma ofrece a los usuarios 1.5 GB de almacenamiento junto con 100 GB de tráfico al mes para alojar sus páginas web

de forma totalmente gratuita. No hay costos ocultos ni términos restrictivos. Se ofrece alojamiento web gratuito que es realmente mejor que algunos servicios pagados conocidos [3].

Características de 000webhost [3]:

- Ancho de banda de 10 GB (10,000 MB)
- Casi infinitos 1000 MB de espacio en disco
- Alojamiento de dominio gratuito
- Panel administrativo de hosting
- Cpanel gratis
- Constructor de sitios gratuito y fácil de usar
- Auto instalador (WordPress, Joomla...)
- Soporte para PHP & bases de datos MySQL
- Activación de cuenta instantánea sin costo



2.3.4 Firebase

Es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada por James Tamplin y Andrew Lee en 2011 y adquirida por Google en 2014. Esta plataforma ya viene instalada a la hora de instalar Android Studio, y nos ofrece distintos servicios que son realmente útiles a la hora de desarrollar aplicaciones móviles en Android. Algunos de estos son :



- **FirestoreAnalytics:** FirestoreAnalytics es una aplicación gratuita que proporciona una visión profunda sobre el uso de la aplicación por parte de los usuarios
- **Firestore Cloud Messaging:** Antiguamente conocido como Google Cloud Messaging (GCM), Firestore Cloud Messaging (FCM) es una plataforma para mensajes y notificaciones para Android, iOS, y aplicaciones web que actualmente puede ser usada de forma gratuita
- **FirestoreAuth:** FirestoreAuth es un servicio que puede autenticar los usuarios utilizando únicamente código del lado del cliente. Incluye la autenticación mediante Facebook, GitHub, Twitter y Google. Además, incluye un sistema de administración del usuario por el cual los desarrolladores pueden habilitar la autenticación de usuarios con email y contraseña que se almacenarán en Firestore
- **FirestoreDatabase:** Firestore proporciona una base de datos en tiempo real y back-end. El servicio proporciona a los desarrolladores de aplicaciones una API que permite que la información de las aplicaciones sea sincronizada y almacenada en la nube de Firestore
- **Firestore Storage:** Firestore Storage proporciona cargas y descargas seguras de archivos para aplicaciones Firestore, sin importar la calidad de la red. El desarrollador lo puede utilizar para almacenar imágenes, audio, vídeo, o cualquier otro contenido generado por el usuario.



2.3.5 XAMPP



XAMPP es un servidor web de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

A partir de la versión 5.6.15, XAMPP cambió la base de datos MySQL por MariaDB, un fork de MySQL con licencia GPL.

El programa se distribuye con la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. A esta fecha, XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.



3. Capítulo III. RESULTADOS

En esta parte del documento se hablará y mostrarán los resultados obtenidos a lo largo de este proyecto, así como la manera en que se siguió la metodología seleccionada para el desarrollo de la aplicación móvil.

3.1 Exploración

El equipo de desarrollo debe generar un plan y establecer las características y los conceptos básicos que están alrededor de todo el proyecto. El equipo de desarrollo se dio a la tarea de visitar al cliente personalmente para hablar acerca del problema y establecer los primeros lineamientos respecto a la metodología, lo cuales se mostrarán en las siguientes secciones.

3.1.1 Establecimiento de actores:

Para esta parte se definieron los actores que van a participar directamente con el uso de la aplicación móvil.

Usuario	Principal valor	Principal Interés del software	Restricciones
Producción	-Departamento encargado de la producción de nuevos productos.	-Notificar de forma automatizada a todo el departamento cuando una falla aparezca en su departamento.	-Ninguna
Calidad	-Departamento encargado de verificar la calidad de los productos que salen del departamento de producción.	-Notificar de forma automatizada a todo el departamento cuando una falla aparezca en su departamento.	-Ninguna
Desarrollo	-Departamento encargado de verificar que se sigan las normas establecidas en los demás departamentos.	-Notificar de forma automatizada a todo el departamento cuando una falla aparezca en su departamento.	-Ninguna

Tabla 1 Actores



3.1.2 Definición de alcance

El propósito de esta etapa es definir los objetos para el proyecto con respecto tanto a los contenidos, así como la línea de tiempo del proyecto, por lo que se habló directamente con el cliente para establecer las metas a lograr en el periodo de tiempo establecido.

Después de haber establecido las metas y objetivos, se empezó a hablar con el asesor técnico para hablar directamente de las necesidades que debía cubrir el producto de software, esto nos da las bases para establecer los requerimientos funcionales específicos y no funcionales específicos de la aplicación, los cuales se muestran a continuación:

ID-REQ	Requerimiento	Prioridad
RF01	La aplicación deberá tener una actividad que permita generar nuevos usuarios.	Obligatorio
RF02	La aplicación deberá tener una actividad que permita el acceso a los usuarios válidos y rechazar a los que no estén registrados.	Obligatorio
RF03	La aplicación permitirá registrar una "falla en curso" que podrá cancelarse o guardarse permanentemente al terminar la falla.	Obligatorio
RF04	La aplicación permitirá registrar fallas en la base de datos cuando la falla sea detenida por los niveles que hayan sido notificados (Nivel 1, Nivel 2 y Nivel 3) y sólo hasta entonces.	Obligatorio
RF05	La aplicación podrá cancelar una falla en curso por cualquier usuario en la aplicación.	Obligatorio
RF06	El sistema notificará después de un tiempo y automáticamente a los usuarios de Nivel 1 y Nivel 2 cuando una falla haya sido agregada a "Fallas en curso".	Obligatorio
RF07	La aplicación notificará 30 minutos después y automáticamente a los usuarios de Nivel 3 que hay una falla que no ha logrado resolver el Nivel 1 y Nivel 2.	Obligatorio
RF08	La aplicación deberá calcular al terminar una "Falla en curso" cuanto tiempo duro en total esa falla.	Obligatorio
RF09	La aplicación mostrará una lista de todas las fallas que se han registrado en el sistema como finalizadas.	Obligatorio
RF10	El sistema deberá mantener la sesión de usuario abierta.	Obligatorio
RF11	La aplicación deberá estar conectado a una base de datos en un servidor remoto.	Obligatorio

Tabla 2 Requerimientos funcionales



ID-REQ	TIPO	Prioridad	REQUERIMIENTO
RNF01	Interoperabilidad	Obligatorio	La aplicación será desarrollada para ser ejecutada en sistema Android 5.1 Lollipop en adelante.
RNF02	Interfaz	Deseable	Los colores de interfaz de usuario deben ser neutrales (no gris).
RNF03	Almacenamiento	Obligatorio	El sistema tendrá la capacidad de albergar la información de todas las fallas que se intenten registrar.
RNF04	Interoperabilidad	Obligatorio	Conectar la aplicación de manera remota para la accesibilidad de los datos desde cualquier dispositivo en cualquier lugar con acceso a la red.
RNF05	Interfaz	Obligatorio	Mensajes de retoralimentación en cada una de las acciones de la aplicación.
RNF06	Interfaz	Obligatorio	La fuente para el texto deberá ser Arial 14dp
RNF07	Interfaz	Obligatorio	La fuente para títulos deberá ser Arial 16dp
RNF08	Interfaz	Deseable	La aplicación mostrará solo un botón rojo y grande para agregar una nueva falla al sistema.
RNF09	Mantenimiento	Obligatorio	La aplicación deberá estar desarrollada en Android Studio.
RNF10	Mantenimiento	Obligatorio	La aplicación deberá estar desarrollada por módulos para una fácil modificación futura.

Tabla 3 Tabla de requerimientos no funcionales

3.1.3 Establecimiento de proyectos

Para esta etapa de la metodología el equipo de trabajo definió los recursos técnicos y humanos que se requieren para que se inicie el desarrollo del proyecto. También se decidió utilizar el diseño arquitectónico “Modelo cliente servidor” pues se acopla perfectamente a la funcionalidad de la aplicación móvil.

3.1.4 Diseño arquitectónico - Modelo Cliente-Servidor

Es un modelo de sistema en el que dicho sistema se organiza como un conjunto de servicios y servidores asociados, más unos clientes que acceden y usan los servicios. Los principales componentes de este modelo son:

- Un conjunto de servidores que ofrecen servicios a otros subsistemas.
- Un conjunto de clientes que llaman a los servicios ofrecidos por los servidores.

- Una red que permite a los clientes acceder a estos servicios. Esto no es estrictamente necesario ya que los clientes y los servidores podrían ejecutarse en una única máquina.

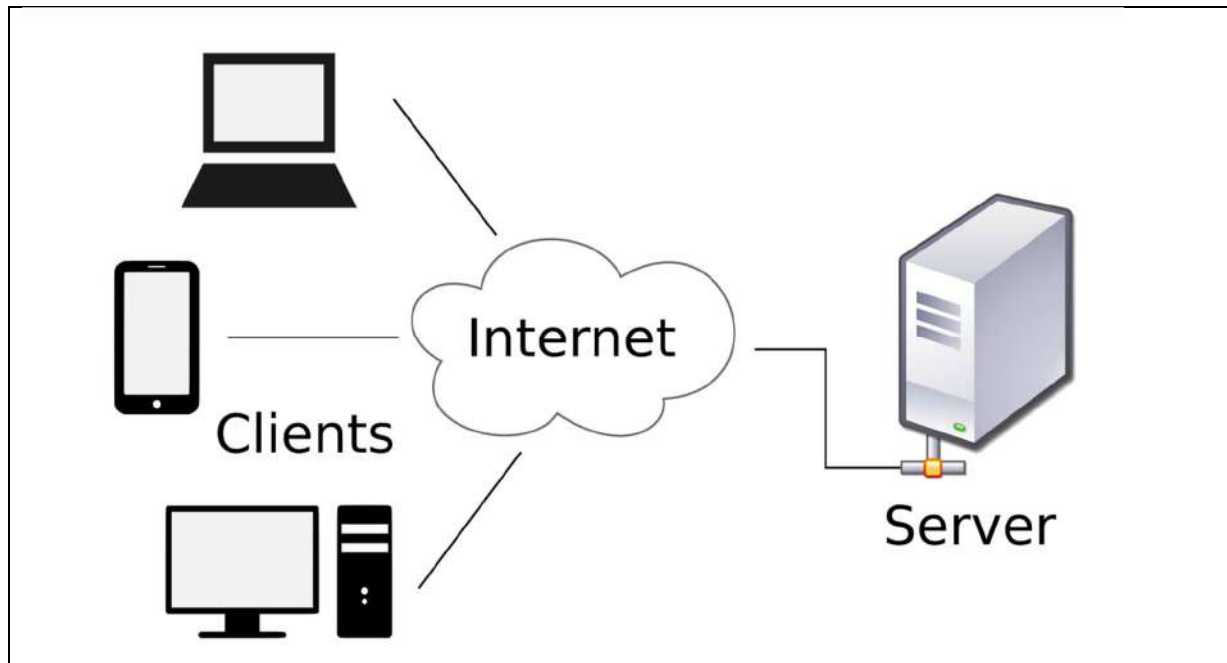


Figura 2 Arquitectura de software Cliente-Servidor

Fases	Actividades	Enero				Febrero				Marzo				Abril	A cargo
		1	2	3	4	1	2	3	4	1	2	3	4	1	
Exploración	Establecimiento de actores														Luis
	Definición de alcance														Luis
	Establecimiento de proyecto														Luis
	Diseño Arquitectónico														Luis
Iniciación	Configuración del entorno														Luis
	Establecer comunicación con el cliente														Luis
	Análisis de los requisitos iniciales														Luis
	Día de pruebas														Luis
Fase de producto -Día de planificación	Análisis de requisitos														Luis
	Revisión de pruebas de aceptación														Luis
	Generación de pruebas de aceptación														Luis
	Planificación de las iteraciones														Luis
-Día de trabajo	Programación en parejas														Luis
	Integración continua														Luis

Proyecto de Estadía
Laboratorios Texzal de México



	Informes al cliente															Luis
-Día de lanzamiento	Integración del sistema															Luis
	Prueba de aceptación															Luis
	Ceremonia de lanzamiento															Luis
Fase de estabilización															Luis	
Fase de pruebas															Luis	

Tabla 4 Cronograma de actividades

3.2 Iniciación

En esta fase se definieron las herramientas necesarias para el desarrollo de la Aplicación de notificación de mantenimiento por niveles, los cuales están descritos en la sección de “Herramientas” en el capítulo 2. Se buscó usar herramientas tecnológicas que tuvieran licencias gratuitas o de mínimo precio pues el presupuesto no es tan grande. Esto fue realizado tomando en cuenta los siguientes puntos:

- Se requiere de un equipo de programadores, en este caso el tamaño del proyecto hace posible que sea desarrollado por un único programador, el cual tomará capacitación de dos semanas para el uso del IDE Android Studio.
- Todos los componentes de software necesarios serán instalados en una laptop con las siguientes características necesarias:
 - Intel (R) Core(TM) i5-2520M CPU @ 2.50GHz 2.50 GHz
 - 8.00 GB Memoria RAM (Mínimo)
 - Sistema operativo de 64 bits Windows 10 Pro
 - Disco duro 1 TB con espacio libre mínimo de 20 GB
- El tipo de desarrollo de software: Se va a desarrollar una aplicación móvil, por lo que se decidió utilizar la plataforma Android Studio para programar las interfaces y funcionalidades de la aplicación.
- Se necesita usar una base de datos de manera remota, por lo que la primera opción de usar como gestor de base de datos gratuito (XAMPP) fue usada solamente para probar el funcionamiento de la conexión entre Adnroidstudio y una base de datos. Buscando, se encontró una página de host gratuito que nos permite tanto gestionar una base de datos con MySql de manera remota, tanto como servidor FTP el cual nos permitirá usar archivos PHP para el manejo de consultas entre la aplicación y la base de datos.
- También se necesitó de una API la cual nos permite mandar y administrar notificaciones en vivo entre dispositivos. Esta API viene preinstalada en el IDE de Android Studio.

Para la instalación de la herramienta Android Studio se realizaron los siguientes pasos:. El IDE Android Studio fue descargado de la página oficial, es una instalación como cualquier otro software.

La siguiente parte es actualizar el SDK (Kit de Desarrollo de Software) para poder utilizar la máquina virtual que nos permite emular dispositivos móviles para verificar el funcionamiento de la aplicación.

Se descargaron distintas versiones de Android para poder utilizarlas en el emulador (Figura 3) así como la versión más reciente de Google Play Services (Figura 4).

Proyecto de Estadía Laboratorios Texzal de México

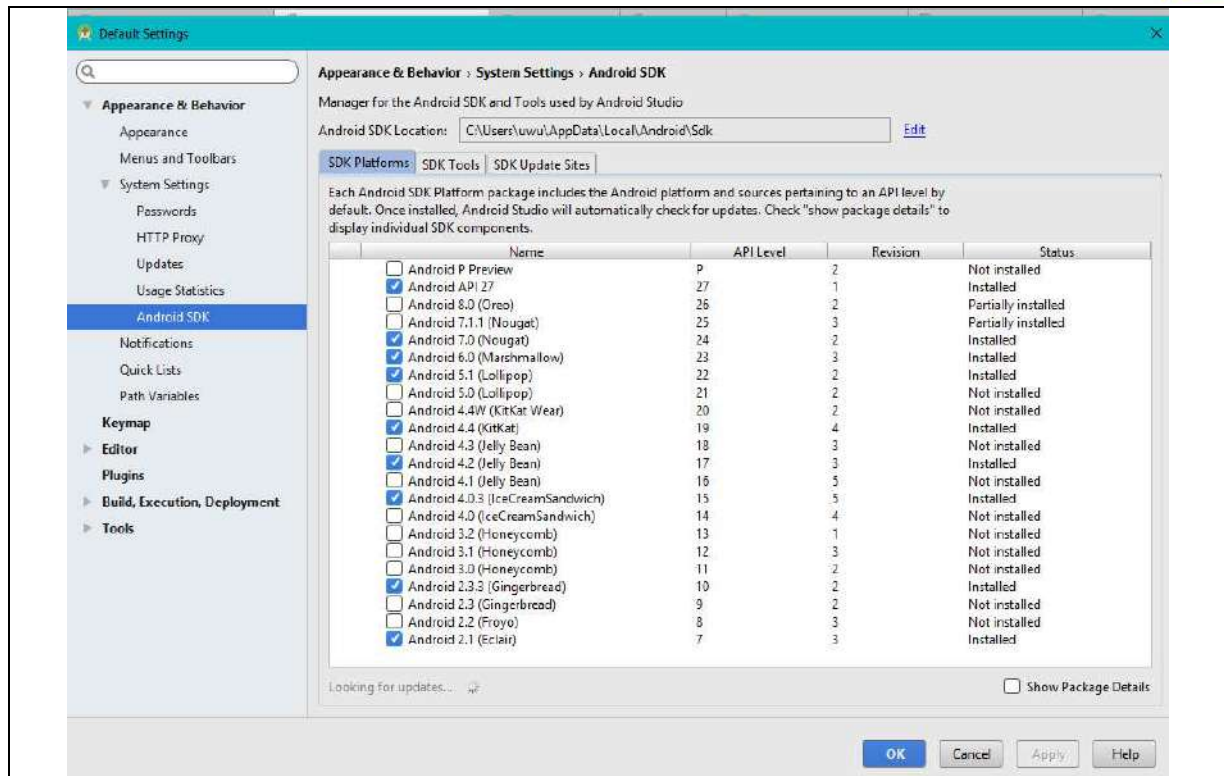


Figura 3 SDK actualizado

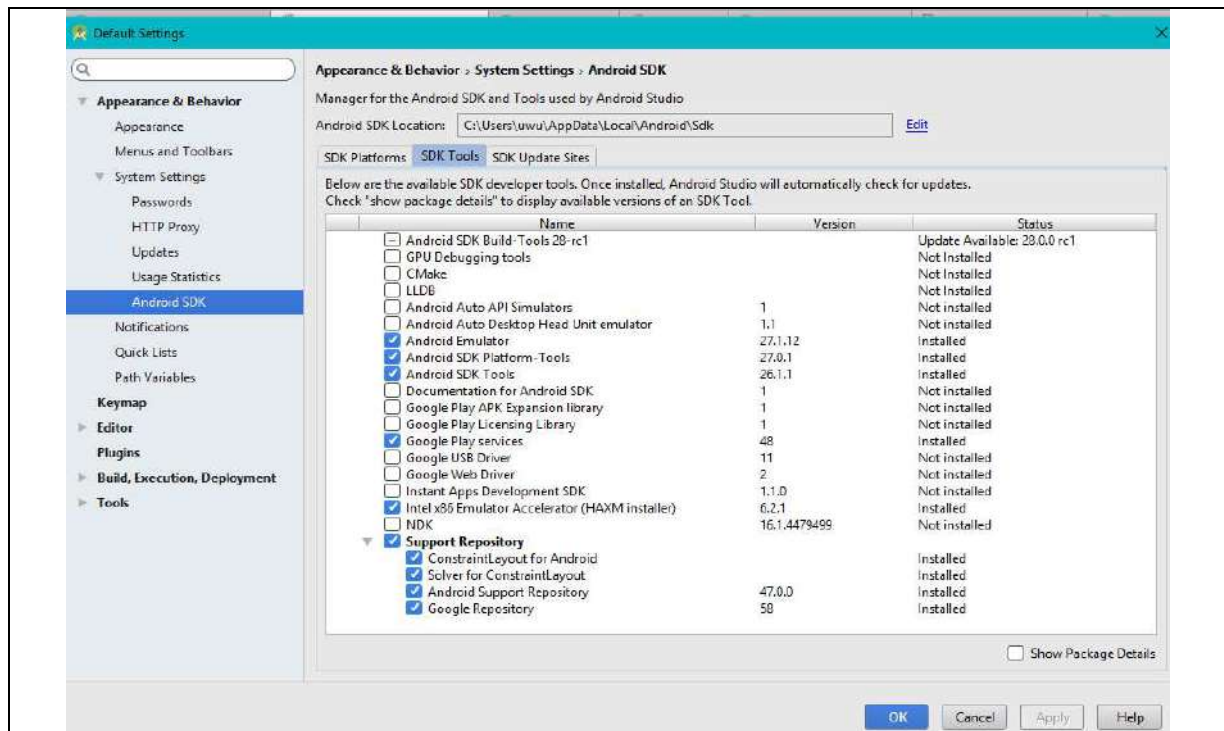


Figura 4 Google Play Services actualizado

El siguiente paso fue crear un dispositivo virtual en la opción AVD (Android Virtual Device) para poder usarlo con una de las versiones de Android descargadas en el SDK.

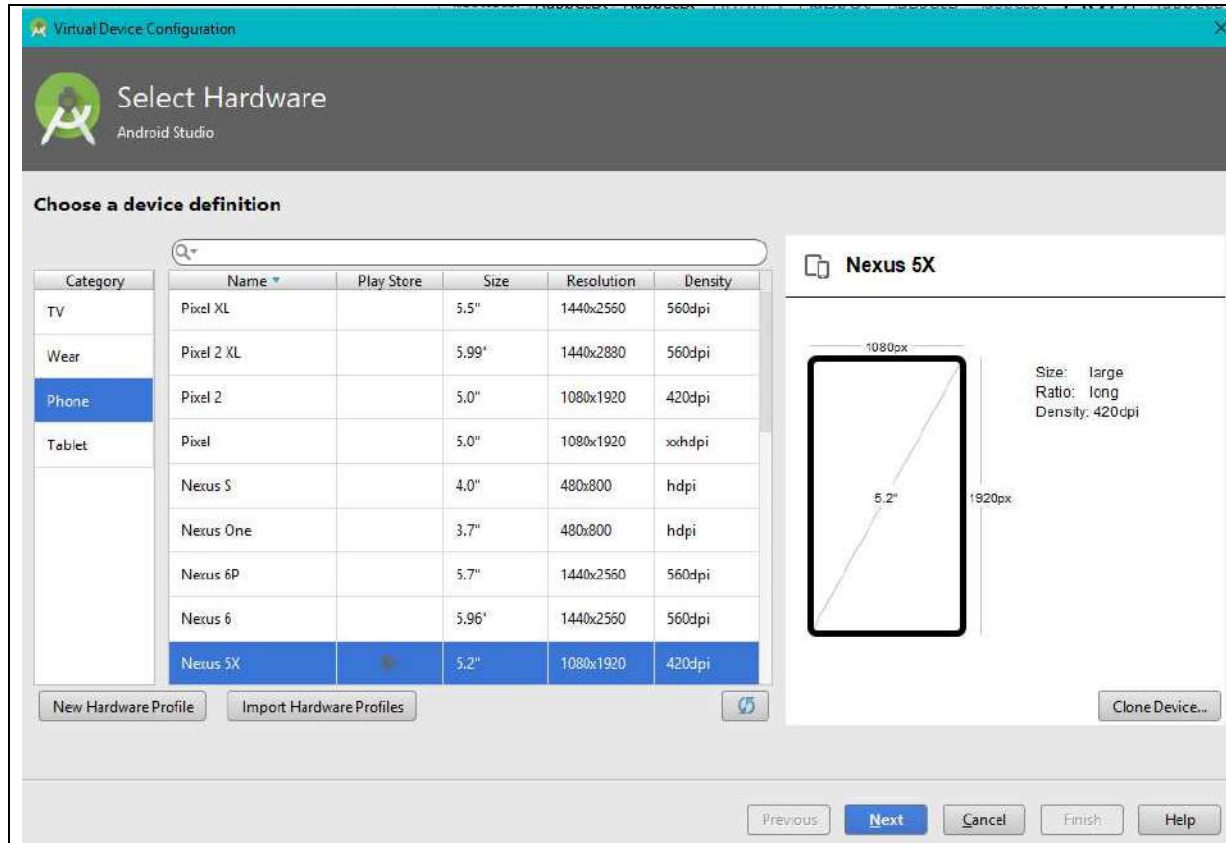


Figura 5 Seleccionar dispositivo virtual

En la Figura 5 se puede ver la variedad de dispositivos que el ADV te permite utilizar, para nuestro desarrollo decidimos utilizar el dispositivo "Nexus 5X" el cual soporta todas las versiones de Android y tiene un tamaño de pantalla normal para un Smartphone (5 pulgadas), aparte del hardware de especificaciones medias en el mercado. Terminado esto, se hizo la primera prueba de funcionamiento (Figura 6). Todo listo por parte del IDE.

Proyecto de Estadía Laboratorios Texzal de México

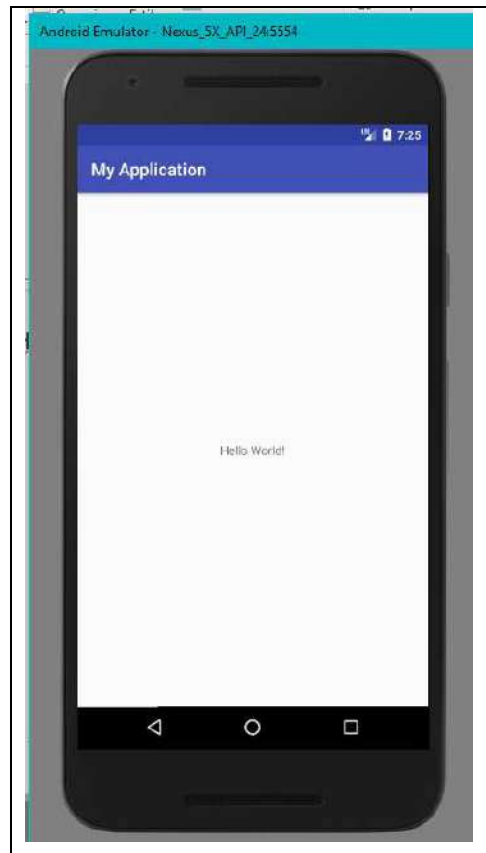


Figura 6 Dispositivo funcionando- hello world

Inicialmente se utilizó XAMPP como servidor, el cual su instalación es bastante sencilla y consta de simples pasos como descargar desde la página oficial e instalar. Para su uso debemos activar las casillas "Start" de Apache y MySQL (Figura 7).

Proyecto de Estadía
Laboratorios Texzal de México

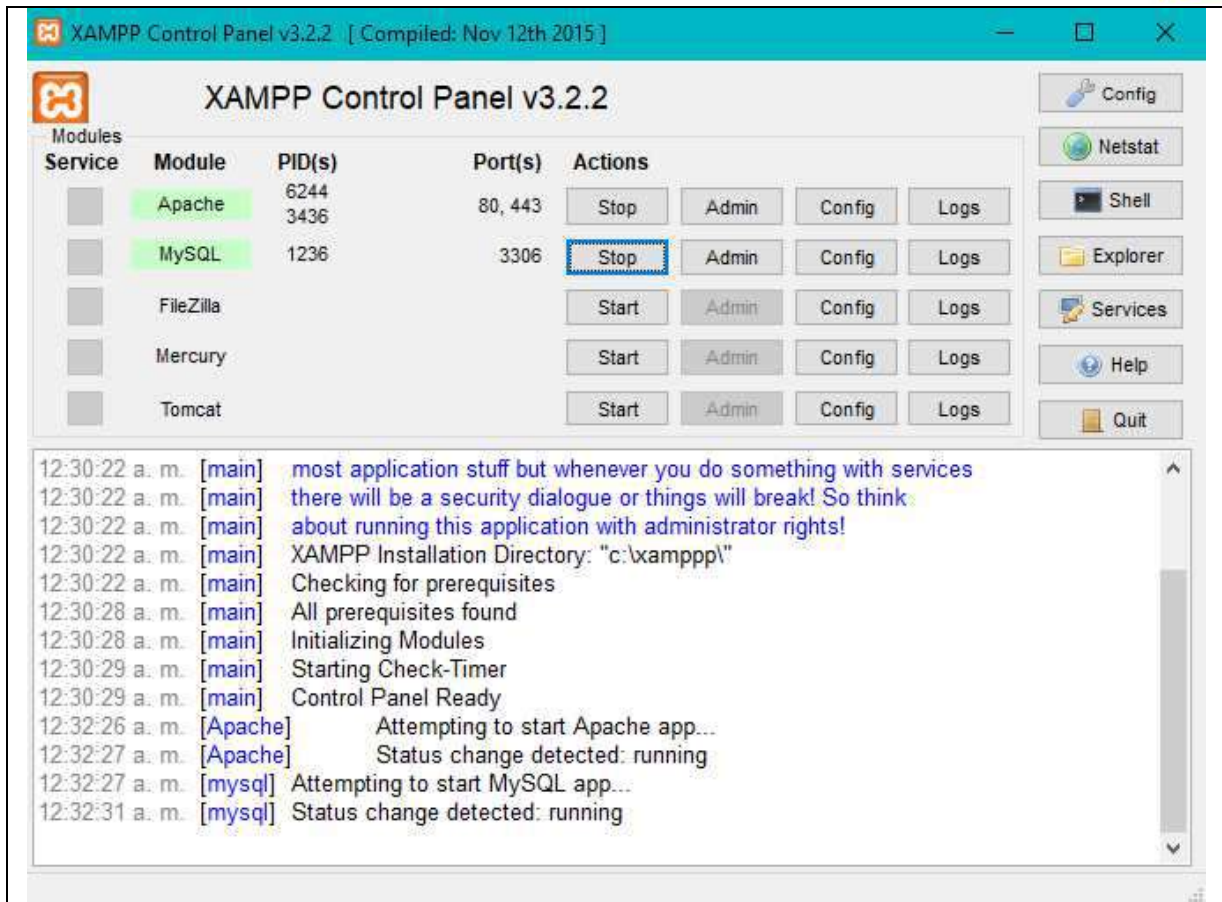


Figura 7 Iniciar servidor XAMPP

Y entrar a la página <http://localhost/phpmyadmin/> para comprobar que todo funciona (figura 8).

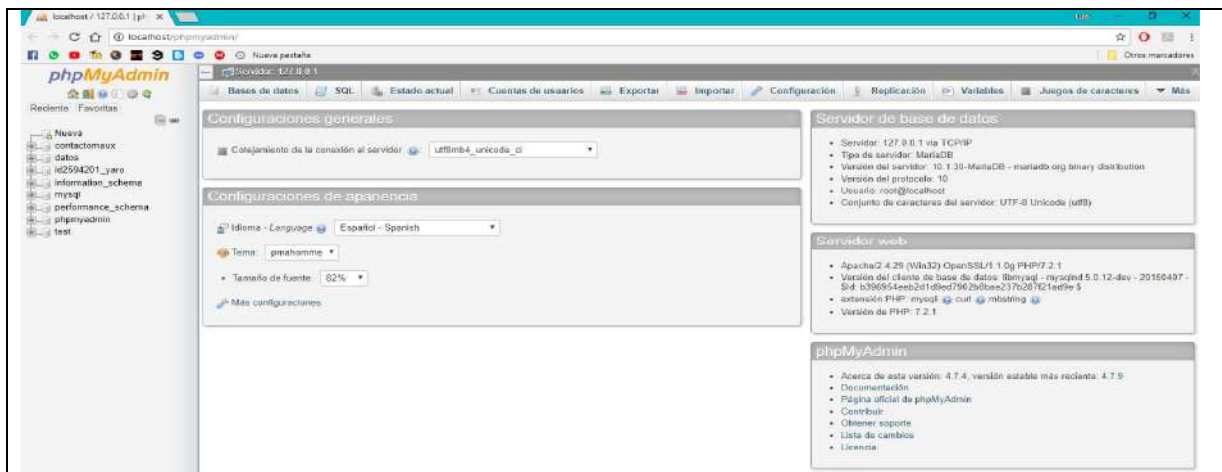


Figura 8 Página de phpmyadmin

Más tarde se empezó a utilizar el servicio de host remoto de la página 000webhost para utilizar un servicio de base de datos remoto.

Proyecto de Estadía Laboratorios Texzal de México



Figura 9 Página principal de 000webhost

Después de registrarnos e iniciar sesión debemos esperar un día para que nuestro servicio pueda ser utilizado. Una vez esto sucede, podemos iniciar.

Por último se dio a la tarea de establecer un modo de comunicación con el cliente, aparte de que inicialmente fueron visitas presenciales a la empresa, no siempre se podía acudir a las instalaciones, por lo que se decidió que nos comunicaríamos cada que necesitáramos algo por medio de la aplicación Whatsapp, así como mandar los avances semanales por medio de correo electrónico para ser presentada al personal de la empresa, dando retroalimentación y realizando las pruebas de aceptación.

En esta parte de la metodología también debemos realizar los prototipos de interfaces (mockup) de las pantallas que estarán disponibles en la aplicación. A continuación se mostrarán las pantallas seleccionadas para su desarrollo:



3.2.1 Diseño de interfaces

En esta parte de la metodología se muestran los diseños preliminares de las interfaces de usuario que comprenderán la aplicación.

MockupLogin: Pantalla encargada de permitir y denegar el acceso a usuarios registrados en la base de datos por medio de dos campos:

- Usuario
- Contraseña

También permite mantener la sesión iniciada o que se cierre al salir de la aplicación.

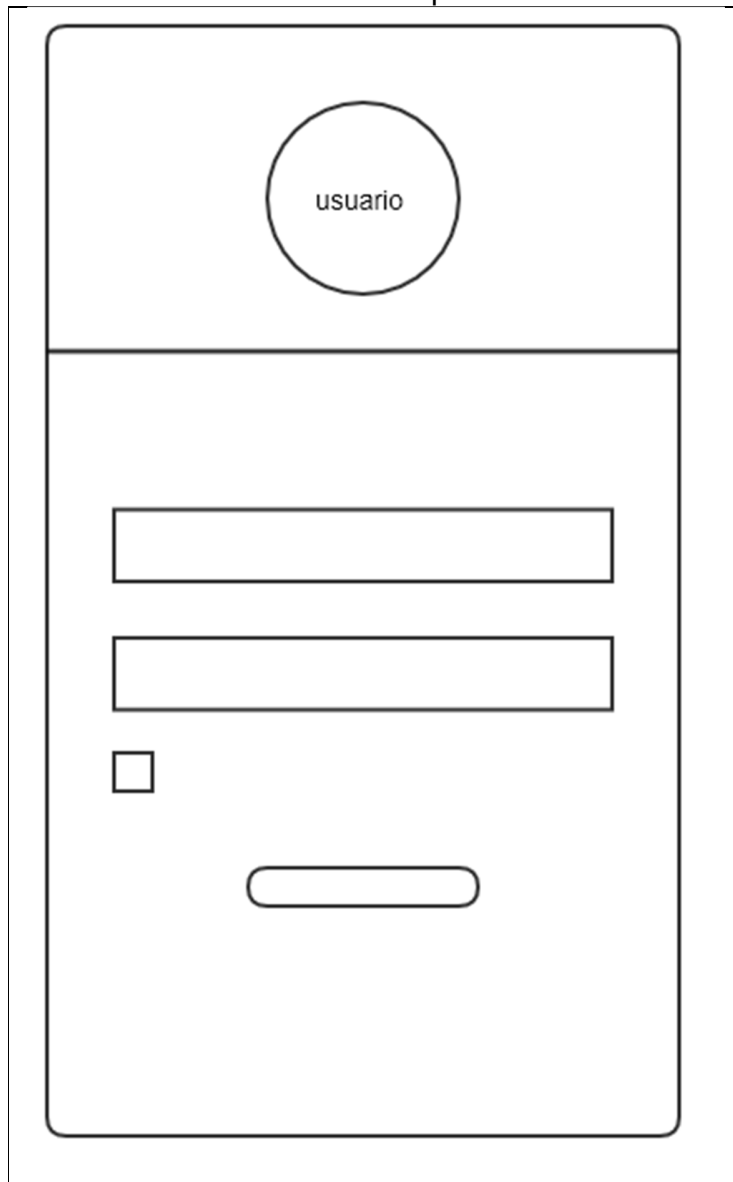


Figura 10 MockupLogin



Mockup Usuario Nuevo: Pantalla que permite agregar un usuario nuevo a la aplicación pidiendo cuatro datos necesarios acerca del usuario.

The mockup shows a vertical rectangular frame containing a registration form. At the top, there is a circular placeholder labeled 'imagen usuario png'. Below this, the text 'Campos de usuario nuevo' is centered. The form consists of four input fields: two simple rectangular boxes, followed by two dropdown menus (each with a downward arrow icon on the left), and a rounded rectangular button at the bottom labeled 'Botón'.

Figura 11 Mockup Registrar usuario



Mockup Menú Principal: Pantalla donde se muestran todas las funciones disponibles en la aplicación.

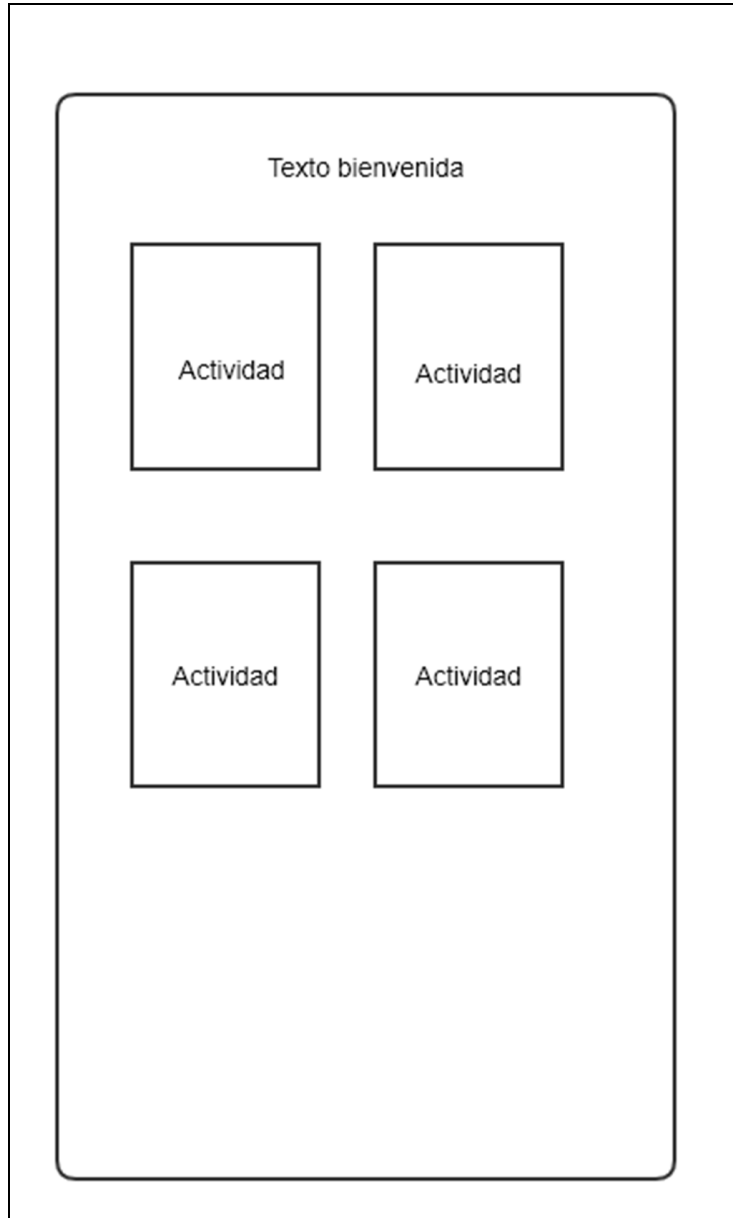


Figura 12 Mockup Menú principal



Mockup Agregar una falla: Pantalla que permite a un usuario agregar una falla nueva en la aplicación poniéndola en curso.



Figura 13 Mockup Registrar falla



Mockup Lista de fallas en curso: Pantalla encargada de mostrar una lista de todas las fallas que actualmente están en curso mostrando los datos relevantes en la vista de cada falla. También cuenta con un botón que permite ver los detalles completos de la falla en curso.

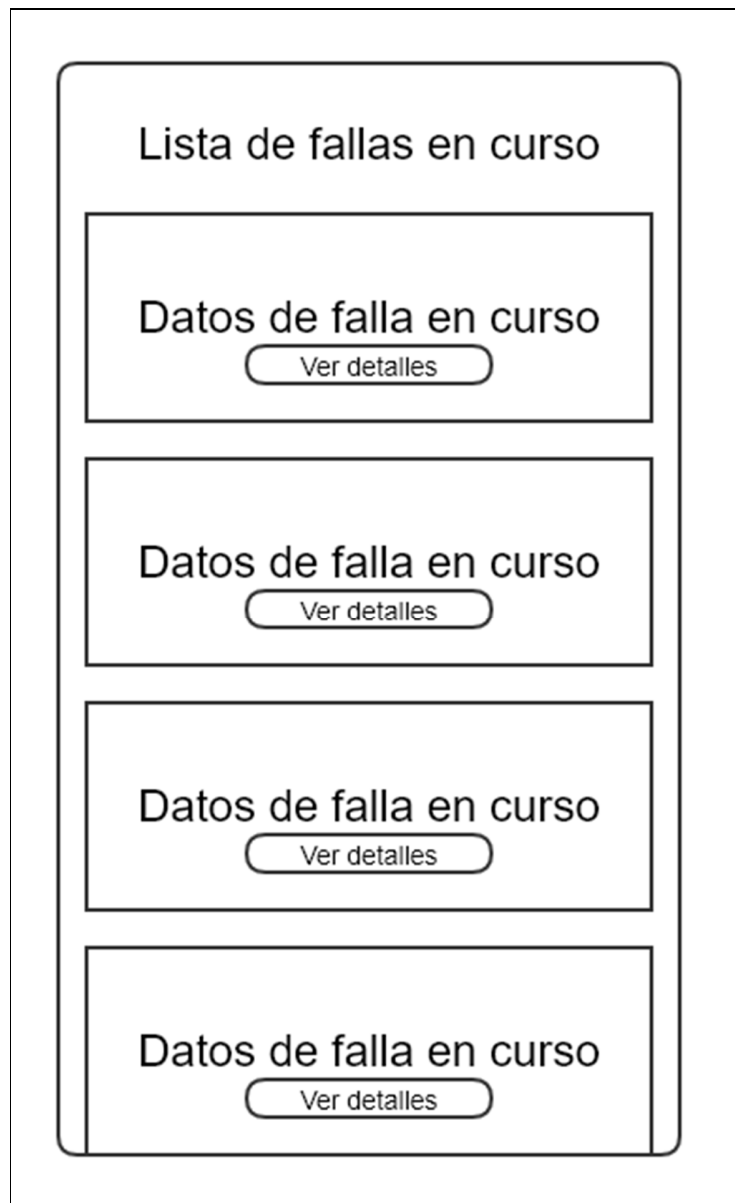


Figura 14 Mockup Fallas en curso



MockupDetalle de falla en curso: Pantalla que muestra todos los datos dela falla en curso incluyendo la hora en la que se dio de alta en el sistema. También habrá un botón que permite concluir la falla seleccionada y un botón para cancelar esa falla.



Figura 15 Mockup Detalles falla en curso



MockupLista de fallas: Pantalla encargada de mostrar una lista de todas las fallas que han sido completadas en la aplicación.

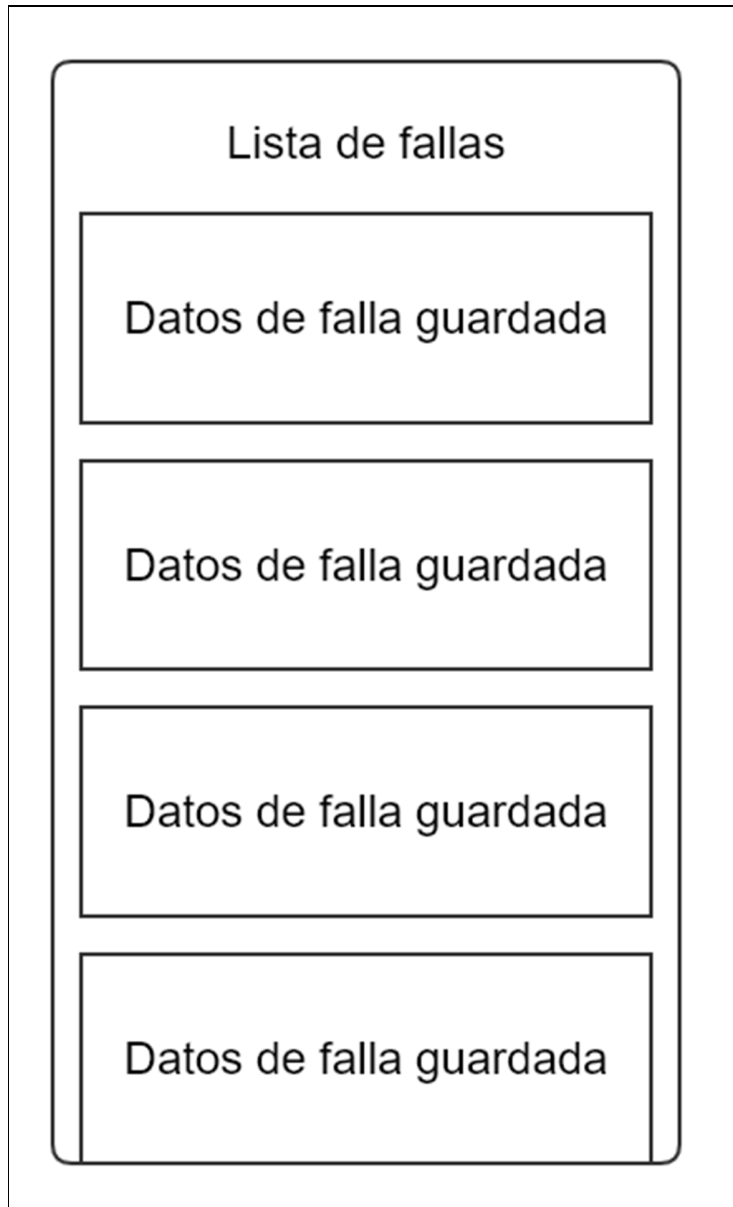


Figura 16 Mockup Lista de fallas guardadas



Mockup Perfil de usuario: Pantalla encargada de mostrar los datos del usuario actual en la aplicación así como las opciones de configuración de cuenta de usuario.

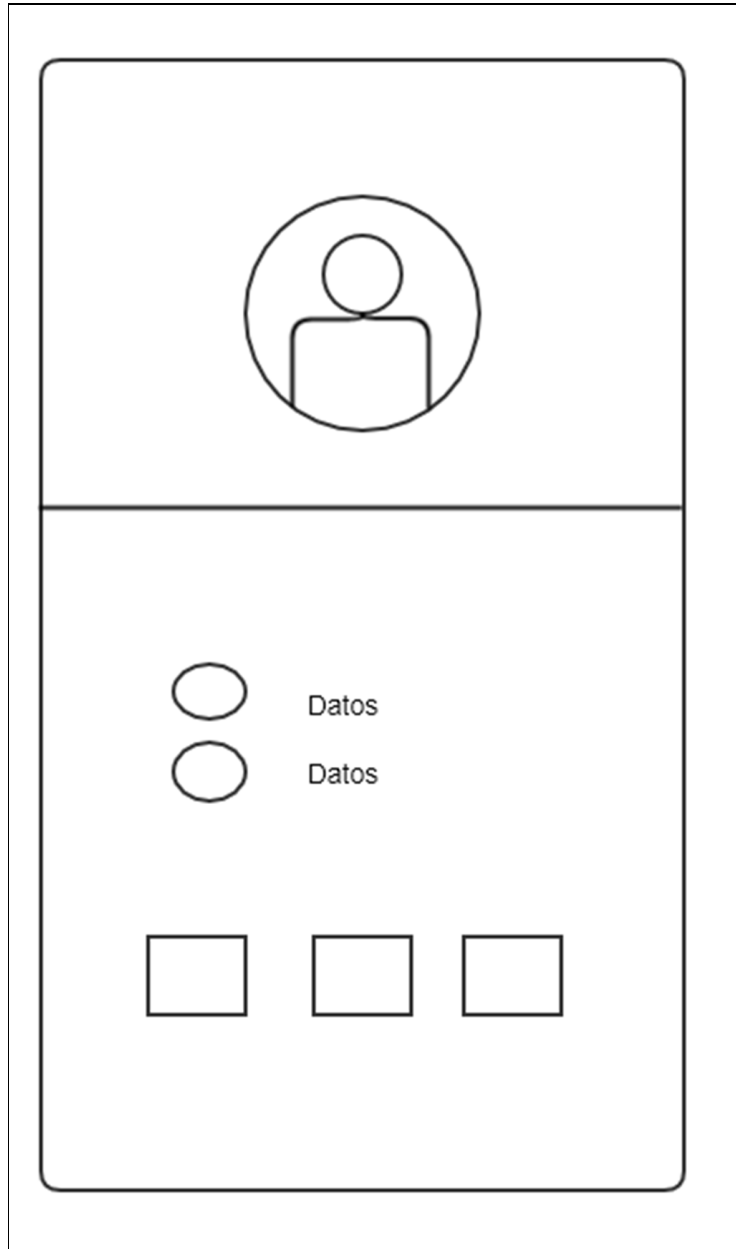


Figura 17 Mockup Perfil de usuario



3.2.2 Diseño de base de datos

La aplicación debe ser complementada por una base de datos que pueda almacenar todos los datos necesarios para su completo funcionamiento. A continuación se muestra la base de datos que se utilizó para el desarrollo de la aplicación.

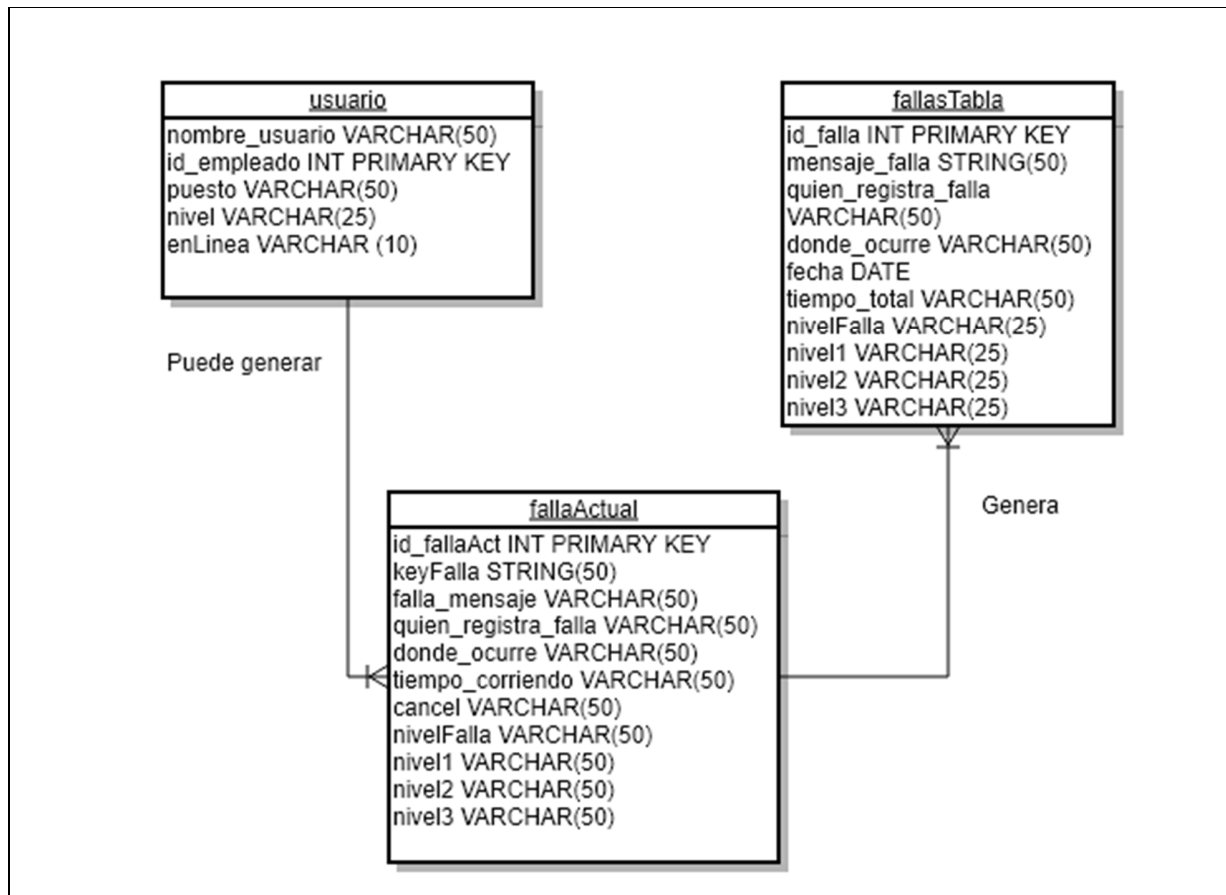


Figura 18 Base de datos Final

3.2.3 Diccionario de datos

También se presenta en esta parte el diccionario de datos de la base de datos implementada para el almacenamiento de los distintos datos del sistema. Se muestra a continuación.



Tabla "fallaActual"			
Campo	Tamaño	Tipo de dato	Descripción
Id_fallaAct	Autoincrementable	IntPrimarykey	Id único para cada falla en curso
keyFalla	50	String	Clave hecha por la aplicación para identificar cada una de las fallas desde la aplicación. Está conformada por: Mensaje de la falla + Nombre de usuario + Nivel de usuario + Puesto
falla_mensaje	50	varchar	Mensaje de la falla enviada por un usuario
quien_registra_falla	50	varchar	Nombre de usuario de quien envía la falla
donde_ocurre	50	varchar	Área en la que ocurre la falla
tiempo_total	50	varchar	Este campo se utiliza para guardar la hora y fecha en que la falla fue agregada
cancel	50	varchar	Este campo es utilizado en la aplicación para identificar cuando una falla en curso fue cancelada
nivelFalla	50	varchar	Este campo lo usa la aplicación para saber si una falla sigue siendo de nivel 2 o ha llegado hasta nivel 3
nivel1	50	varchar	Nombre de usuario nivel 1 que da por terminada la falla
nivel2	50	varchar	Nombre de usuario nivel 2 que da por terminada la falla
nivel3	50	varchar	Nombre de usuario nivel 3 que da por terminada la falla

Tabla 5 Tabla "fallaActual"



Tabla "fallasTabla"			
Campo	Tamaño	Tipo de dato	Descripción
Id_falla	Autoincrementable	IntPrimarykey	Id único para cada falla en curso
falla_mensaje	50	varchar	Mensaje de la falla enviada por un usuario
quien_registra_falla	50	varchar	Nombre de usuario de quien envía la falla
donde_ocurre	50	varchar	Área en la que ocurre la falla
fecha		Date	Campo que guarda la fecha de cuando fue terminada la falla
tiempo_corriendo	50	varchar	Este campo guarda el tiempo total dividido en días, horas, minutos y segundos que duró la falla en ser enviada hasta ser terminada
nivelFalla	25	varchar	Este campo lo usa la aplicación para saber si una falla sigue siendo de nivel 2 o ha llegado hasta nivel 3
nivel1	25	varchar	Nombre de usuario nivel 1 que da por terminada la falla
nivel2	25	varchar	Nombre de usuario nivel 2 que da por terminada la falla
nivel3	25	varchar	Nombre de usuario nivel 3 que da por terminada la falla

Tabla 6 Tabla "fallasTabla"



Tabla "usuario"			
Campo	Tamaño	Tipo de dato	Descripción
nombre_usuario	50	Varchar	Nombre de usuario para acceder al sisema
Id_empleado	Autoincrementable	IntPrimarykey	Id único para cada empleado
puesto	50	Varchar	Puesto del empleado
nivel	25	Varchar	Nivel del empleado

Tabla 7 Tabla "usuario"

Tabla "enLineaFalla"			
Campo	Tamaño	Tipo de dato	Descripción
Id_linea	Autoincrementable	IntPrimarykey	Id único para cada registro
usuario	50	Varchar	Campo que guarda cada uno de los usuarios que acceden a la aplicación.

Tabla 8 Tabla "enLineaFalla"

3.3 Producto

Llegamos a la tercera parte de la metodología seleccionada y su objetivo principal es implementar las funcionalidades a la aplicación.

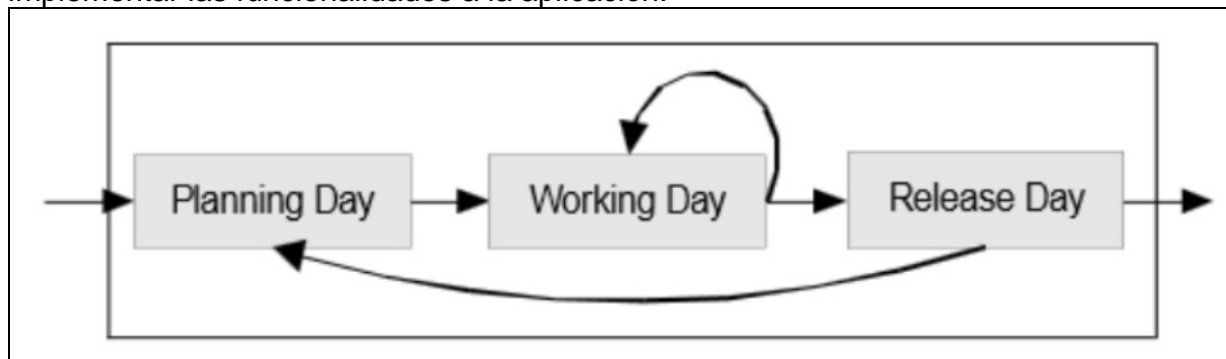


Figura 19 Iteración de fase: Producto

Empezamos por desarrollar historias de usuario la cual es una representación de un requisito de software escrito en una o dos frases utilizando el lenguaje común del usuario. A continuación se mostrarán las historias de usuario que fueron creadas en base a los requerimientos del cliente.



3.3.1 Historia de usuario 1

Historia N° 1	Tipo	Dificultad	Esfuerzo		Prioridad	Notas
Mandar notificación local	Nuevo	Fácil	Estima	Gastad	2	
	Arreglar Mejorar	Moderado Difícil	do 4 días	o 5 días		
Descripción						
Se necesita que la aplicación pueda mandar notificaciones a los dispositivos, inicialmente						
Fecha	Estado	Comentario				
22/01/2018	Definición					
22 - 27	Implementación					
27/01/2018	Hecho	Se implementó la funcionalidad con éxito				

Figura 20 Historia de usuario "Mandar notificación local"

Prueba de aceptación "Mandar notificación local"
Número: 1
Historia: Mandar notificación local
Fecha de historia: 22/01/2018
Fecha de término: 27/01/2018
Descripción: La aplicación necesita mandar notificaciones a distintos dispositivos, por lo que inicialmente se prueba con mandar una notificación de manera local.
Resultado esperado: Los resultados al implementar esta funcionalidad fueron los esperados.

Tabla 9 Prueba de aceptación "Mandar notificación local"

A continuación se muestra la interfaz terminada en esta iteración:

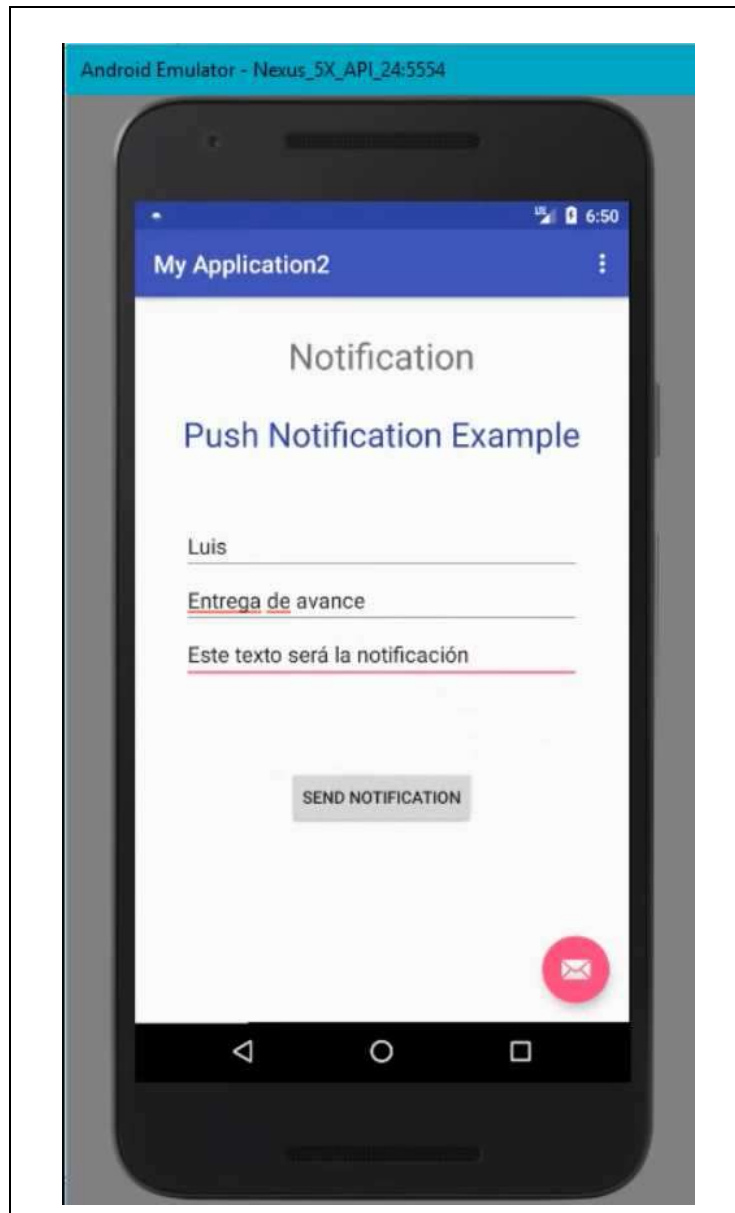


Figura 21 Iteración "Notificación local"

Prueba aceptación de historia de usuario "Mandar notificación local":

- ✓ La aplicación manda una notificación al pulsar un botón
- ✓ La aplicación manda la notificación con los datos escritos en los campos
- ✓ La notificación llega inmediatamente con los datos proporcionados



3.3.2 Historia de usuario 2

Historia N° 2	Tipo	Dificultad	Esfuerzo		Prioridad	Notas
CRUD (Create, Read, Update, Delete)	Nuevo	Fácil	Estimado	Gastado	2	
	Arreglar Mejorar	Moderado Difícil	4 días	5 días		
Descripción						
Se necesita que la aplicación pueda guardar los datos de la falla a dar de alta en una base de datos, primeramente de manera local						
Fecha	Estado	Comentario				
29/01/2018	Definición					
29 - 03	Implementación					
03/02/2018	Hecho	Se implementó la funcionalidad con éxito				

Tabla 10 Historia de usuario "CRUD local"

Para esta iteración se necesitó de XAMPP para el uso de datos MySQL de manera local. Se implementó una base de datos como ejemplo para comprobar la conectividad entre el dispositivo móvil mediante la máquina virtual de Android Studio y la base de datos local.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	id			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2	nombres	latin1_swedish_ci		No	Ninguna		
<input type="checkbox"/>	3	telefono			No	Ninguna		

Figura 22 Base de datos local de ejemplo con MySQL-XAMPP

Primero se crearon los archivos .php para el servidor local, los cuales nos van a permitir abrir una conexión con el servidor y hacer todos los movimientos en la base de datos (guardar, eliminar, consultar y actualizar datos).

Estos archivos van colocados en la carpeta C:\xampp\htdocs\CursoAndroid. El primero archivo nos permite realizar la conexión con la base de datos con los datos respectivos como servidor, usuario, contraseña y base de datos, el resto de los archivos nos permite hacer cada una de las funciones del CRUD (Crear registros, leer registros, actualizar registros y eliminar registros).

Proyecto de Estadía
Laboratorios Texzal de México



```
Area de trabajo
interfazSemantica.html
politicas.html
procedimientos.html
sitiosR.html
consulta.php
registro.php
eliminar.php
functions.php
actualizar.php

html +
  ▸ equipamiento
  ▸ fotos
  ▸ fotos1
  ▸ fotosCA
  ▸ fotosSlider
  ▸ icon
  ▸ Imagenes_s
  ▸ images
  ▸ imgUpp
  ▸ js
ayuda.html
colecciones.html
config.php
connection.php
contacto.html
contacto.php
dublin.html

1  <?php
2  //header( 'Content-Type: text/html;charset=utf-8' );
3
4  ▼ function ejecutarSqlCommand($commando){
5
6      $mysqli = new mysqli("localhost", "root", "", "datos");
7
8
9      // Revisar estado de conexión
10 ▼ if ($mysqli->connect_errno) {
11     printf("Connect failed: %s\n", $mysqli->connect_error);
12     exit();
13 }
14
15 if ( $mysqli->multi_query($commando)
16 ▼     if ($resultset = $mysqli->store_result()) {
17         while ($row = $resultset->fetch_array(MYSQLI_BOTH)) {
18             }
19             $resultset->free();
20         }
21     $mysqli->close();
22 }
23
24
25 ▼ function getSQLResultSet($commando){
26     $mysqli = new mysqli("localhost", "root", "", "datos");
27     // Revisar estado de conexión
28 ▼ if ($mysqli->connect_errno) {
29     printf("Connect failed: %s\n", $mysqli->connect_error);
30     exit();
31 }
32
33 ▼ if ( $mysqli->multi_query($commando)) {
34     return $mysqli->store_result();
35 }
36 }
37
38     $mysqli->close();
39 }
40
41 ?>
```

Figura 23 Archivos .php para el manejo de datos locales

```
Area de trabajo
interfazSemantica.html
politicas.html
procedimientos.html
sitiosR.html
consulta.php
registro.php
eliminar.php
functions.php

1  <?php
2  include('functions.php');
3  $id=$_GET["id"];
4
5
6 ▼ if($resultset=getSQLResultSet("SELECT * FROM `alumnos` WHERE id='$id'")){
7 ▼     while ($row = $resultset->fetch_array(MYSQLI_NUM)){
8         echo json_encode($row);
9     }
10 }
11
12 ?>
```

Figura 24 Archivo para consulta de datos

Proyecto de Estadía
Laboratorios Texzal de México



```
Area de trabajo
interfazSemantica.html
politicas.html
procedimientos.html
sitiosR.html
consulta.php
registro.php
eliminar.php
functions.php
actualizar.php

1 <?php include ('functions.php');
2 $nombres=$_GET['nombres'];
3 $telefono=$_GET['telefono'];
4
5
6 ejecutarSqlCommand("INSERT INTO `alumnos` (nombres, telefono)
7 VALUES (
8 '$nombres' ,
9 '$telefono')
10
11 ON DUPLICATE KEY UPDATE `nombres`= '$nombres',
12 `telefono`='$telefono';");
13
14 ?>
```

Figura 25 Archivo para consulta de datos

```
Area de trabajo
interfazSemantica.html
politicas.html
procedimientos.html
sitiosR.html
consulta.php
registro.php
eliminar.php
functions.php

1 <?php
2 include('functions.php');
3 $id=$_GET["id"];
4
5
6 if($resultset=getSQLResultSet("DELETE FROM `alumnos` WHERE id='$id'")){
7     while ($row = $resultset->fetch_array(MYSQLI_NUM)){
8         echo json_encode($row);
9     }
10 }
11
12 ?>
```

Figura 26 Archivo para consulta de datos

```
Area de trabajo
interfazSemantica.html
politicas.html
procedimientos.html
sitiosR.html
consulta.php
registro.php
eliminar.php
functions.php
actualizar.php

1 <?php
2 include('functions.php');
3 $id=$_GET["id"];
4 $nombres=$_GET['nombres'];
5 $telefono=$_GET['telefono'];
6
7 if($resultset=getSQLResultSet("UPDATE `alumnos` SET nombres='$nombres', telefono='$telefono' WHERE id='$id'")){
8     while ($row = $resultset->fetch_array(MYSQLI_NUM)){
9         echo json_encode($row);
10    }
11 }
12 ?>
```

Figura 27 Archivo para consulta de datos

Lo que concierne a la implementación en la aplicación para que esto funcione, bastó con agregar una función que nos permite cargar un archivo mediante un URL y una función que nos permite leer los datos recibidos. También eventos para los botones que activan dichas funciones.

Proyecto de Estadía
Laboratorios Texzal de México



```
btnconsultar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        new ConsultarDatos().execute("http://10.0.2.2/CursoAndroid/consulta.php?id="+editTextId.getText().toString());
    }
});

btnGuardar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        new CargarDatos().execute("http://10.0.2.2/CursoAndroid/registro.php?nombres="+editTextNombres.getText().toString()+
        "&telefono="+editTextTelefono.getText().toString());
    }
});

btnEliminar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        new EliminarDatos().execute("http://10.0.2.2/CursoAndroid/eliminar.php?id="+editTextId.getText().toString());
    }
});

btnActualizar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        new ActualizarDatos().execute("http://10.0.2.2/CursoAndroid/Actualizar.php?id="+editTextId.getText().toString()+
        "&nombres="+editTextNombres.getText().toString()+"&telefono="+editTextTelefono.getText().toString());
    }
});
});
```

Figura 28 Eventos de los botones de la interfaz



```
private String downloadUrl(String myurl) throws IOException {
    Log.i("URL", ""+myurl);

    //reemplazar espacios por "%20"
    myurl = myurl.replace(" ", "%20");
    InputStream is = null;
    // Solo mostrar los primeros 500 caracteres de la consulta.
    int len = 500;

    try {
        URL url = new URL(myurl); //convertir URL String a objeto URL
        HttpURLConnection conn = (HttpURLConnection) url.openConnection(); //abrir conexión
        conn.setReadTimeout(10000 /* milliseconds */);
        conn.setConnectTimeout(15000 /* milliseconds */);
        conn.setRequestMethod("GET");
        conn.setDoInput(true);
        // Realizar consulta
        conn.connect();
        int response = conn.getResponseCode();
        Log.d("respuesta", "The response is: " + response); //Repuesta en consola del estado de la consulta
        is = conn.getInputStream();

        // convertir InputStream a string
        String contentAsString = readIt(is, len);
        return contentAsString;

        // Cerrar el InputStream
    } finally {
        if (is != null) {
            is.close();
        }
    }
}
```

Figura 29 Función que crea una conexión entre la base de datos y la aplicación

```
public String readIt(InputStream stream, int len) throws IOException, UnsupportedEncodingException {

    //Leer un InputStream y convertirlo a String

    Reader reader = null;
    reader = new InputStreamReader(stream, "UTF-8");
    char[] buffer = new char[len];
    reader.read(buffer);
    return new String(buffer);
}
```

Figura 30 Función lee los datos recibidos de la consulta



Proyecto de Estadía
Laboratorios Texzal de México

Prueba de aceptación "CRUD local"
Número: 2
Historia: CRUD
Fecha de historia: 29/01/2018
Fecha de término: 03/02/2018
Descripción: La aplicación necesita guardar los datos de las fallas que sucedan en la empresa, por lo que inicialmente se prueba con una base de datos local de prueba.
Resultado esperado: Los resultados al implementar esta funcionalidad fueron los esperados.

Tabla 11 Prueba de aceptación "CRUD local"

A continuación se muestra la interfaz terminada en esta iteración:



Figura 31 Iteración "CRUD"



Prueba aceptación:

- ✓ La aplicación guarda los datos escritos en los campos
- ✓ La aplicación elimina los datos de un registro por medio de un id proporcionado por el usuario en el primer campo
- ✓ La aplicación actualiza los datos del registro que se solicite
- ✓ La aplicación muestra todos los datos del registro que coincide con el id escrito en el primer campo

3.3.3 Historia de usuario 3

Historia N° 3	Tipo	Dificultad	Esfuerzo		Prioridad	Notas
CRUD de manera remota	Nuevo Arreglar Mejorar	Fácil	Estimado	Gastado	1	
		Moderado	4 días	5 días		
Descripción						
Para este punto la aplicación puede realizar las funciones básicas con la base de datos pero de manera local, se necesita que pueda realizar lo mismo con una base de datos remota.						
Fecha	Estado	Comentario				
05/02/2018	Definición					
05 – 10	Implementación					
10/02/2018	Hecho	Se implementó la funcionalidad con éxito				

Tabla 12 CRUD de manera remota

En esta parte se empezó a hacer uso de la página 000webhost que nos permite tener una base de datos remota, así como el uso de archivos php para el manejo de los datos. Lo primero que se hizo fue crear una cuenta en la página la cual pide un día para que tu conexión remota esté disponible. Después de ese día se pudo acceder sin problema para nuestros propósitos.

Proyecto de Estadía Laboratorios Texzal de México

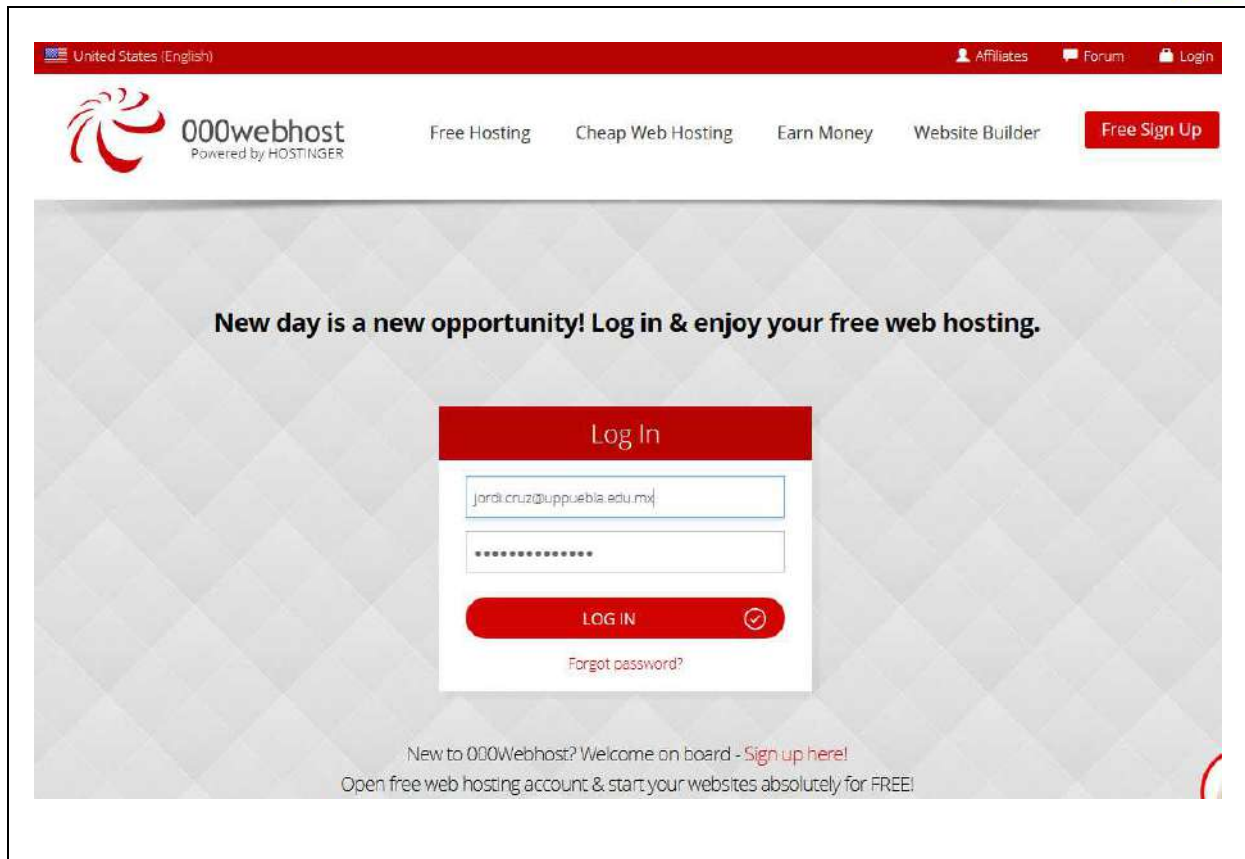


Figura 32 Login de 000webhost

Lo primero que se hará después de entrar al sitio con nuestro usuario y contraseña es ir a la sección de ManageDatabase y después entrar a la sección “Manage” y a PhpMyAdmin



DB Name	DB User	DB Host
id2594201_yaro	id2594201_yaro	localhost

Figura 33 000webhost- apartado Managedatabases

Lo siguiente es crear la tabla para guardar nuestros datos en la base de datos que nos da predeterminadamente la plataforma

	id	nombres	telefono
<input type="checkbox"/>	1	Luis Eduardo P.	2141307000
<input type="checkbox"/>	2	Alberto Figueroa G	44417292
<input type="checkbox"/>	3	Daniel German P.	2481994
<input type="checkbox"/>	6	Daniel hectorq	9000098

Figura 34 000webhost- apartado phpMyAdmin



Proyecto de Estadía Laboratorios Texzal de México

Ahora es necesario subir a la plataforma los archivos Php que nos permitirán realizar cualquier proceso desde nuestra aplicación a la base de datos

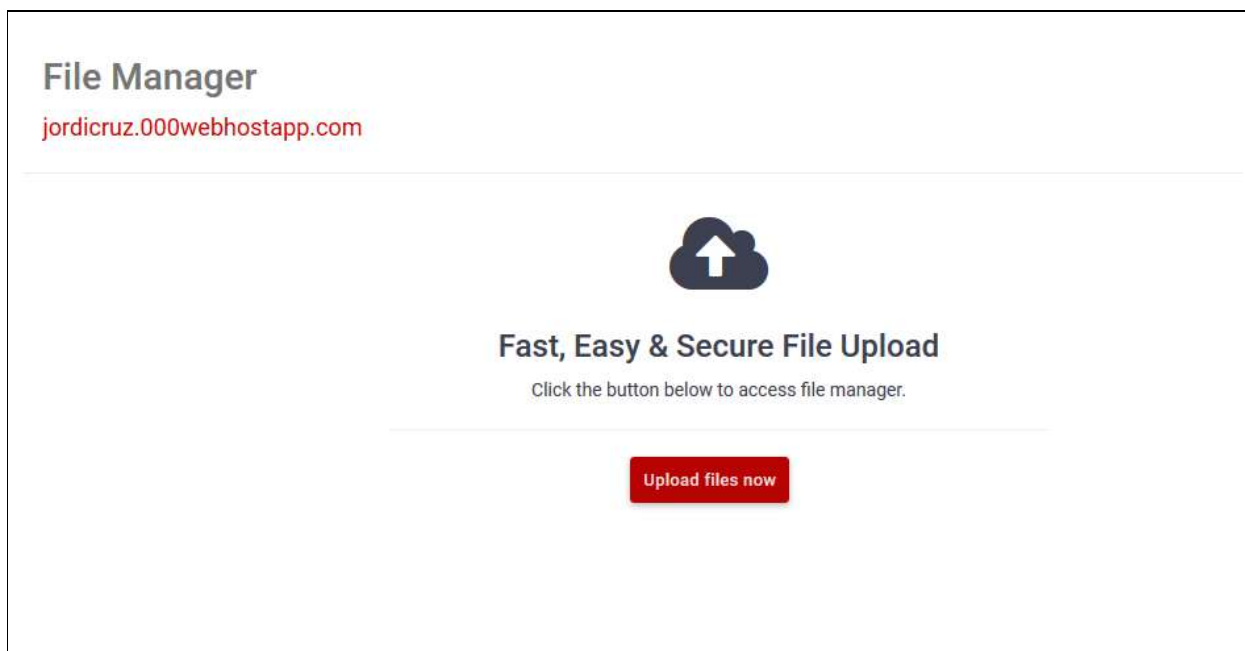


Figura 35 000webhost- apartado File manager

Vamos a subir o cargar la carpeta “CursoAndroid” la cual contiene nuestros archivos Php

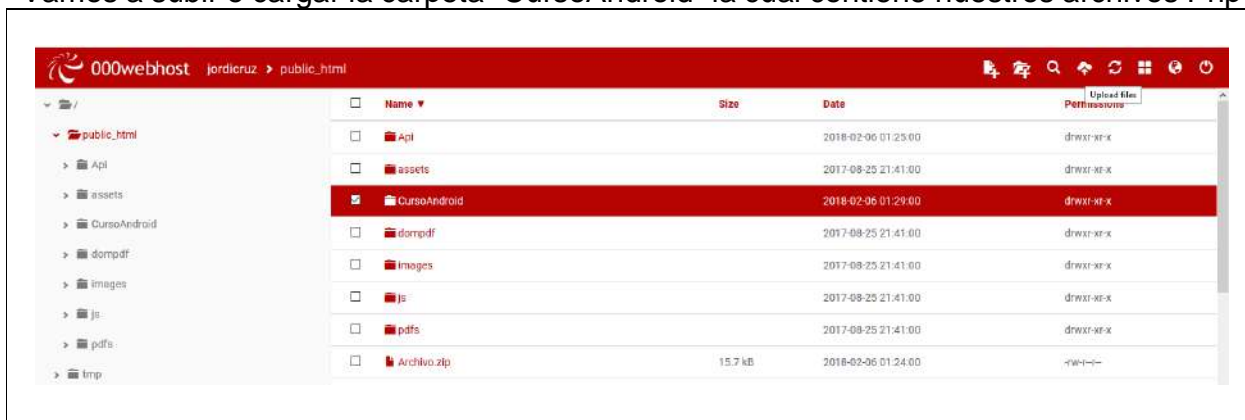


Figura 36 000webhost- apartado File manager

Una vez cargados los archivos entramos a la carpeta y abrimos el archivo “funtions.php” para modificar la dirección del servidor remoto. La página nos proporciona los datos para hacer una conexión remota, los datos son los siguientes:

- Servidor: localhost
- Base de datos: id2594201_yaro
- Usuario: id2594201_yaro
- Contraseña: Meli0das.35483



```
Edit file
```

```
/public_html/CursoAndroid/functions.php
```

```
1 <?php
2 //header( 'Content-Type: text/html;charset=utf-8' );
3
4 function ejecutarSqlCommand($commando){
5
6     $mysqli = new mysqli("localhost", "id2594201_yaro", "Meli0das.35483", "id2594201_yaro");
7
8
9     // Revisar estado de conexión
10 if ($mysqli->connect_errno) {
11     printf("Connect failed: %s\n", $mysqli->connect_error);
12     exit();
13 }
14
15 if ( $mysqli->multi_query($commando))
16     if ($resultset = $mysqli->store_result()) {
17         while ($row = $resultset->fetch_array(MYSQLI_BOTH)) {
18             '
19         }
20     }
```

Figura 37 Modificar archivo de conexión de base de datos

Hay pocos cambios que realizar en la aplicación el cual sólo es el cambio de dirección del servidor local al remoto.

```
btnconsultar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //new ConsultarDatos().execute("http://10.0.2.2/CursoAndroid/consulta.php?id="+editTextId.getText().toString());
        new ConsultarDatos().execute("http://jordicruz.000webhostapp.com/CursoAndroid/consulta.php?id="+editTextId.getText().toString());
    }
});

btnGuardar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //new CargarDatos().execute("http://10.0.2.2/CursoAndroid/registro.php?nombres="+editTextNombres.getText().toString()+"&telefono="+editTextTelefono.getText().toString());
        new CargarDatos().execute("http://jordicruz.000webhostapp.com/CursoAndroid/registro.php?nombres="+editTextNombres.getText().toString()+"&telefono="+editTextTelefono.getText().toString());
    }
});

btnEliminar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //new EliminarDatos().execute("http://10.0.2.2/CursoAndroid/eliminar.php?id="+editTextId.getText().toString());
        new EliminarDatos().execute("http://jordicruz.000webhostapp.com/CursoAndroid/eliminar.php?id="+editTextId.getText().toString());
    }
});

btnActualizar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //new ActualizarDatos().execute("http://10.0.2.2/CursoAndroid/Actualizar.php?id="+editTextId.getText().toString()+"&nombres="+editTextNombre.getText().toString());
        new ActualizarDatos().execute("http://jordicruz.000webhostapp.com/CursoAndroid/actualizar.php?id="+editTextId.getText().toString()+"&nombres="+editTextNombre.getText().toString());
    }
});
```

Figura38 Modificaciones en la aplicación

Prueba de aceptación “CRUD de maneta remota”
Número: 3
Historia: CRUD de manera remota
Fecha de historia: 05/02/2018
Fecha de término: 10/02/2018
Descripción: La aplicación actualmente puede almacenar datos en una base de datos de manera local mediante XAMPP, es tiempo de implementar una base de datos remota mediante la plataforma 000wehost.
Resultado esperado: Los resultados al implementar esta funcionalidad fueron los esperados.

Tabla 13 Prueba de aceptación “CRUD de maneta remota”

A continuación se muestra la interfaz terminada en esta iteración:

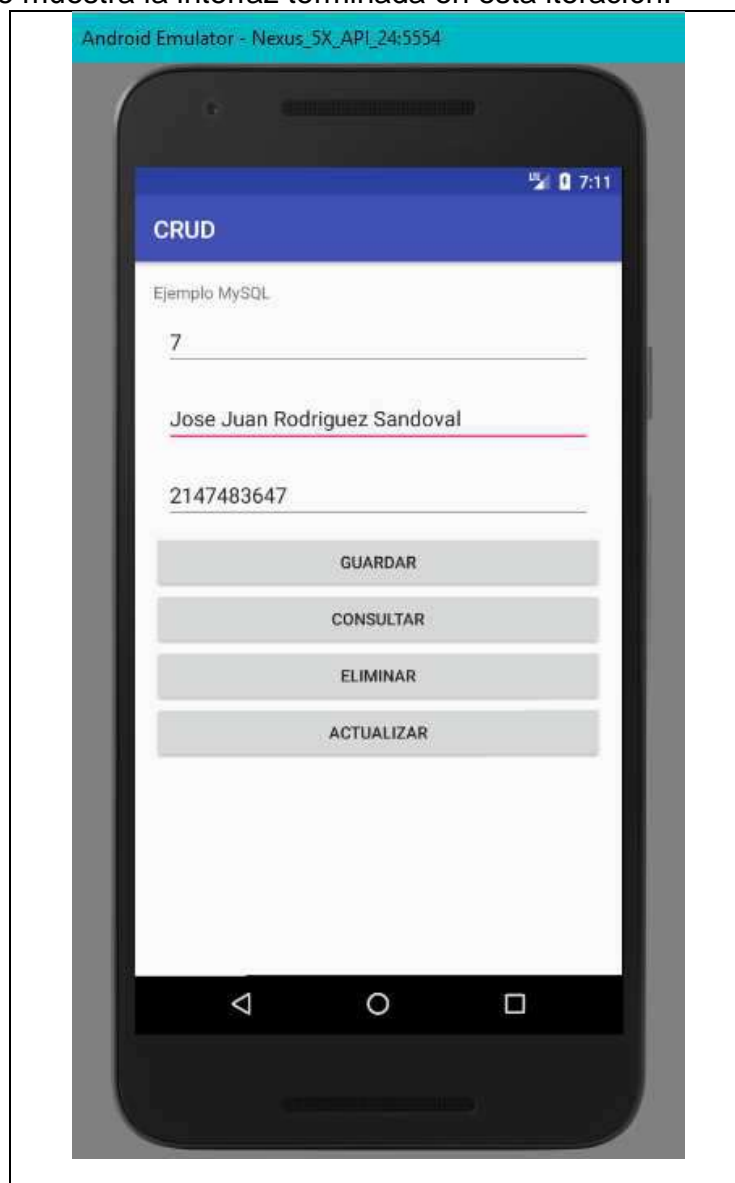


Figura 39 Iteración “CRUD”



Prueba aceptación:

- ✓ La aplicación guarda los datos escritos en los campos de manera remota
- ✓ La aplicación elimina los datos de un registro por medio de un id proporcionado por el usuario en el primer campo de manera remota
- ✓ La aplicación actualiza los datos del registro que se solicite de manera remota
- ✓ La aplicación muestra todos los datos del registro que coincide con el id escrito en el primer campo de manera remota

3.3.4 Historia de usuario 4

Historia N° 4	Tipo	Dificultad	Esfuerzo		Prioridad	Notas
Mandar notificación a diferentes dispositivos	Nuevo	Fácil	Estimado	Gastado	1	
	Arreglar Mejorar	Moderado Difícil	4 días	5 días		
Descripción						
Se busca la manera de implementar que nuestra aplicación pueda mandar una notificación a todos los dispositivos con la aplicación, así simulando el inicio de las notificaciones de errores por nivel en tiempo real						
Fecha	Estado	Comentario				
12/02/2018	Definición					
12 - 17	Implementación					
17/02/2018	Hecho	Se implementó la funcionalidad con éxito				

Tabla 14 Historia de usuario "Mandar notificación a diferentes dispositivos"

Para esta iteración se utilizó la plataforma Firebase, la cual nos permite mandar notificaciones entre los dispositivos que tengan instalada la aplicación. Para lograr esto se debe seguir una serie de pasos. Lo primero es acceder a la página web oficial de Firebase (Figura 40)(<https://firebase.google.com>) para agregar nuestro proyecto a su plataforma.

Proyecto de Estadía Laboratorios Texzal de México

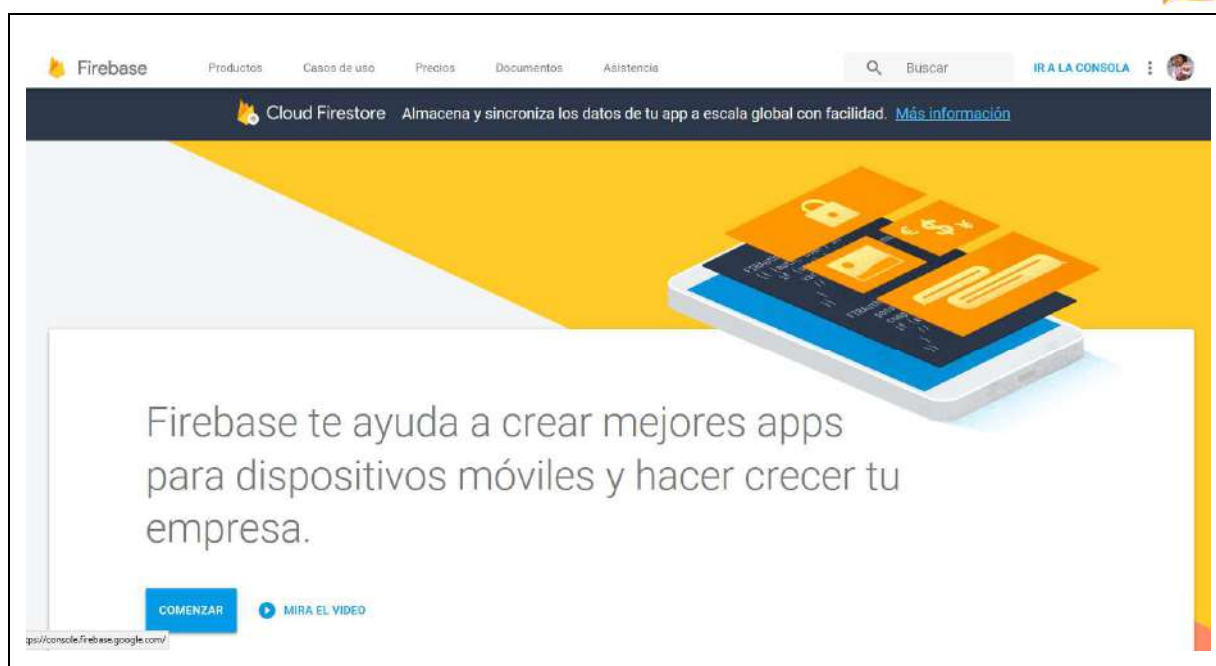


Figura 40 Página oficial de Firebase

Después de pulsar en “COMENZAR” nos dirigirá a la pantalla de bienvenida para añadir un nuevo proyecto.

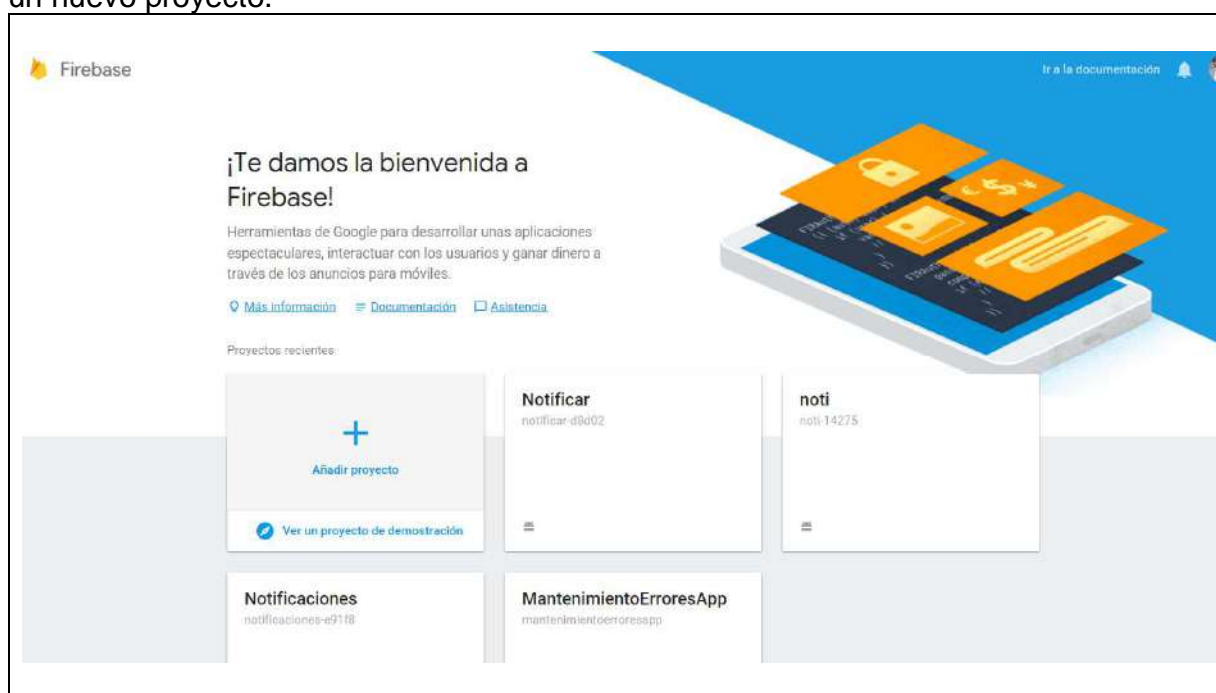


Figura 41 Página de bienvenida de Firebase

Proyecto de Estadía
Laboratorios Texzal de México



En la creación de la cuenta debemos seguir algunos sólo necesitamos poner un nombre a nuestro nuevo proyecto y seleccionar nuestro país.

A screenshot of the 'Añadir un proyecto' (Add a project) dialog box in the Firebase console. The dialog has a title bar with a close button (X) in the top right corner. It contains three main sections: 1. 'Nombre del proyecto' (Project name): A text input field containing 'LaboratoriosTexzalApp' with a dropdown arrow on the right. 2. 'ID del proyecto' (Project ID): A text input field containing 'laboratoriostexzalapp' with a pencil icon on the right. 3. 'País/Región' (Country/Region): A dropdown menu with 'México' selected. To the right of the project name field, there are icons for Android, iOS, and code, followed by a note: 'Nota: Los proyectos abarcan las aplicaciones de distintas plataformas' with a help icon. At the bottom, there are two buttons: 'CANCELAR' (grey) and 'CREAR PROYECTO' (blue). Below the form, there is a paragraph of text: 'De forma predeterminada, tus datos de Analytics se utilizarán para mejorar otras funciones de Firebase, así como otros productos de Google. En la configuración puedes controlar cómo se comparten dichos datos en cualquier momento. [Más información](#)'.

Añadir un proyecto ✕

Nombre del proyecto

LaboratoriosTexzalApp

ID del proyecto ⓘ

laboratoriostexzalapp

País/Región ⓘ

México

Nota: Los proyectos abarcan las aplicaciones de distintas plataformas ⓘ

De forma predeterminada, tus datos de Analytics se utilizarán para mejorar otras funciones de Firebase, así como otros productos de Google. En la configuración puedes controlar cómo se comparten dichos datos en cualquier momento. [Más información](#)

CANCELAR **CREAR PROYECTO**

Figura 42 Ventana para crear un proyecto

Proyecto de Estadía
Laboratorios Texzal de México



Hay que esperar unos segundos hasta que la página nos mande un mensaje de confirmación.

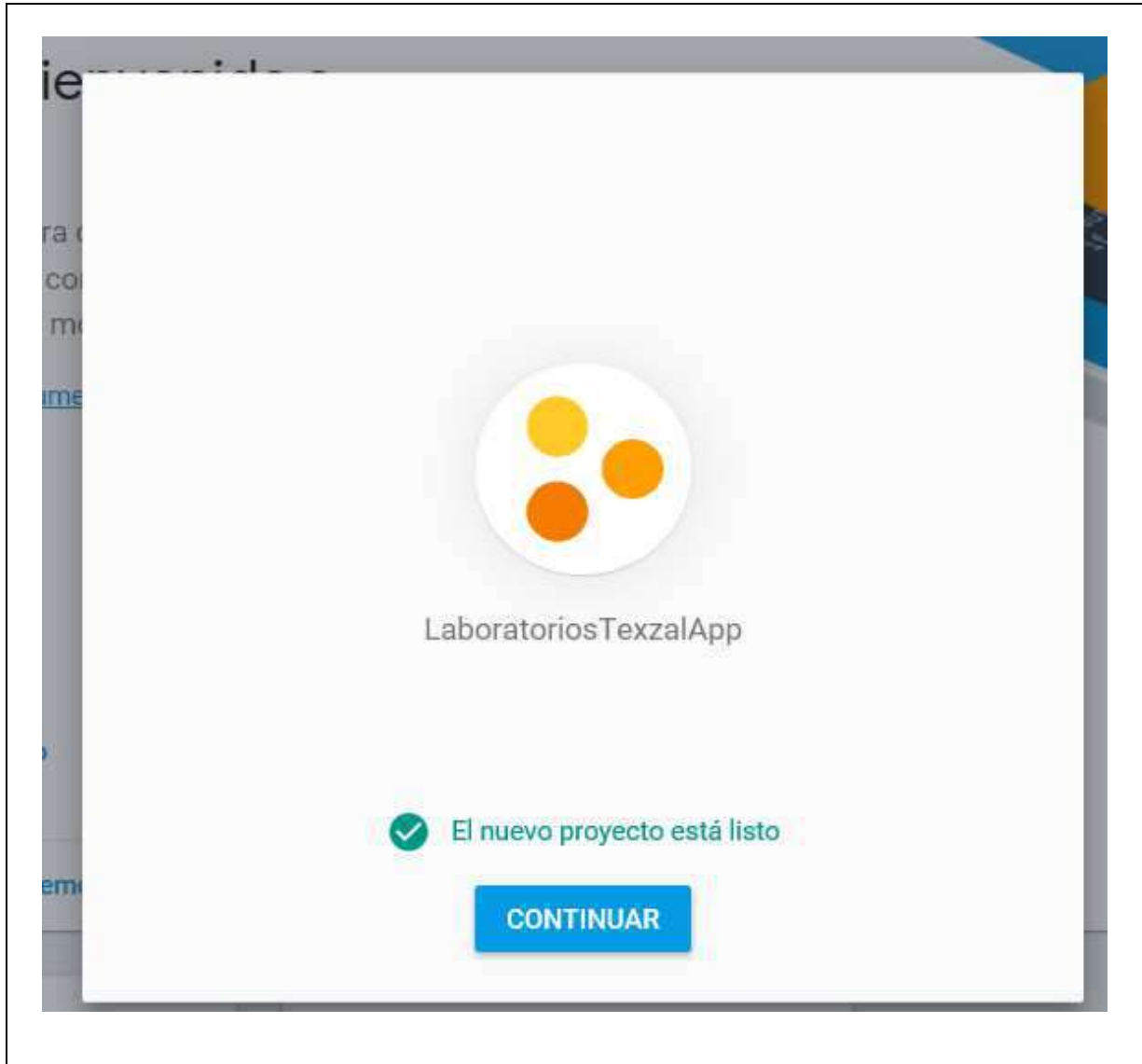


Figura 43 Ventana de confirmación de creación de cuenta



Al dar clic en continuar nos dirige a la página principal de Firebase donde nos da la opción de agregar un proyecto de Android a la plataforma.

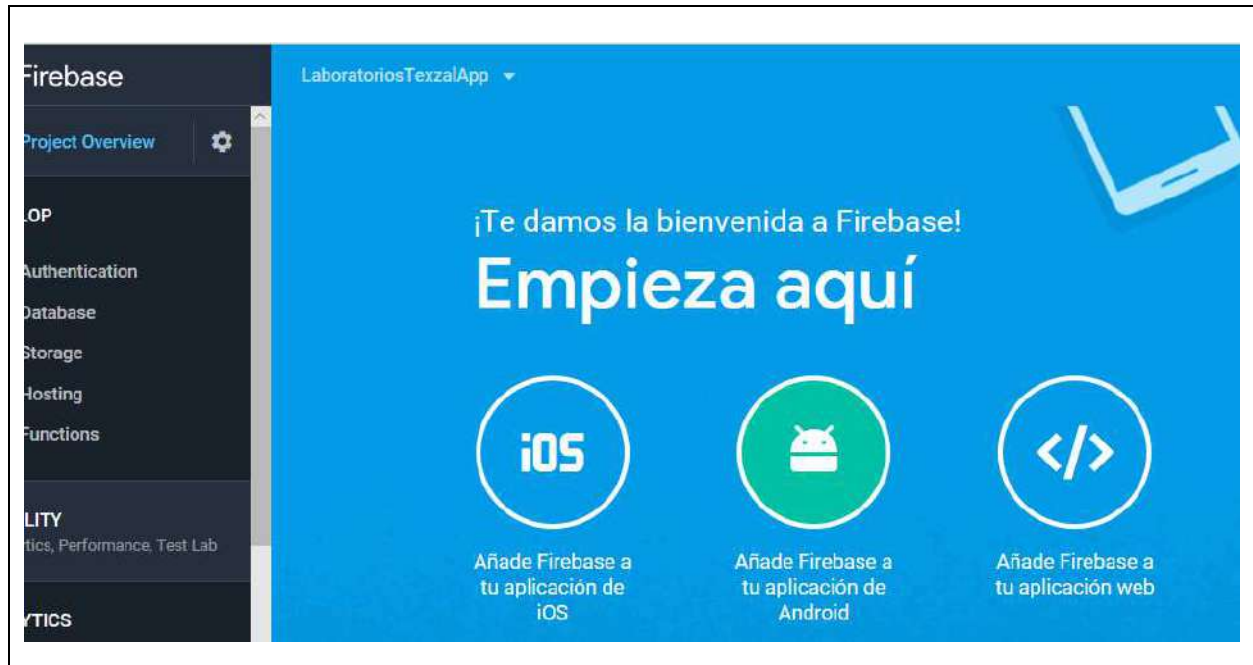


Figura 44 Página principal de Firebase

Nos abrirá la siguiente ventana en la que debemos escribir el paquete de la aplicación.

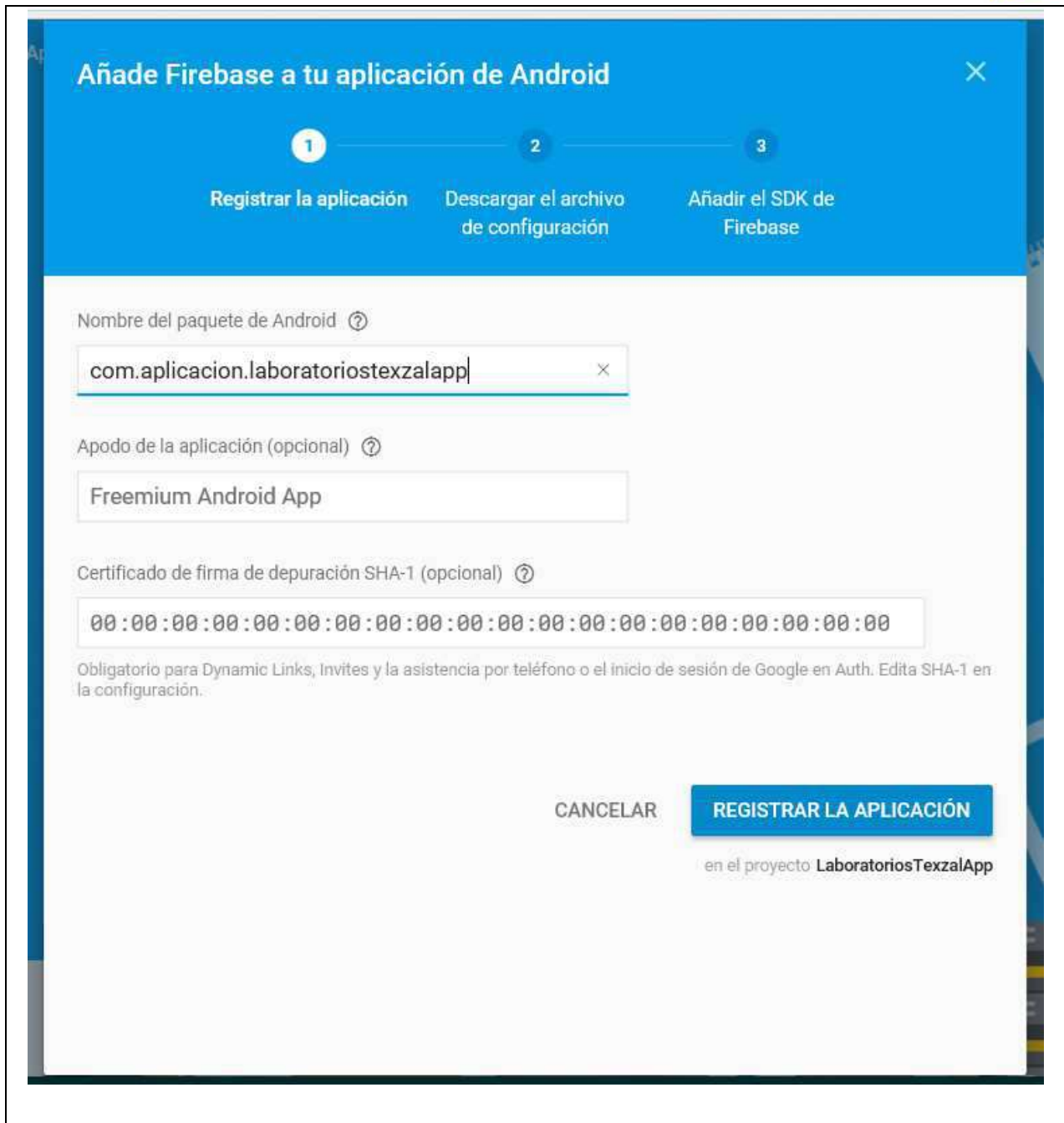


Figura 45 Ventana para añadir un proyecto Android a Firebase



Se pulsa el botón “Registrar la aplicación” y la página mostrará la siguiente pantalla la cual nos muestra instrucciones para implementar Firebase al proyecto en Android Studio.

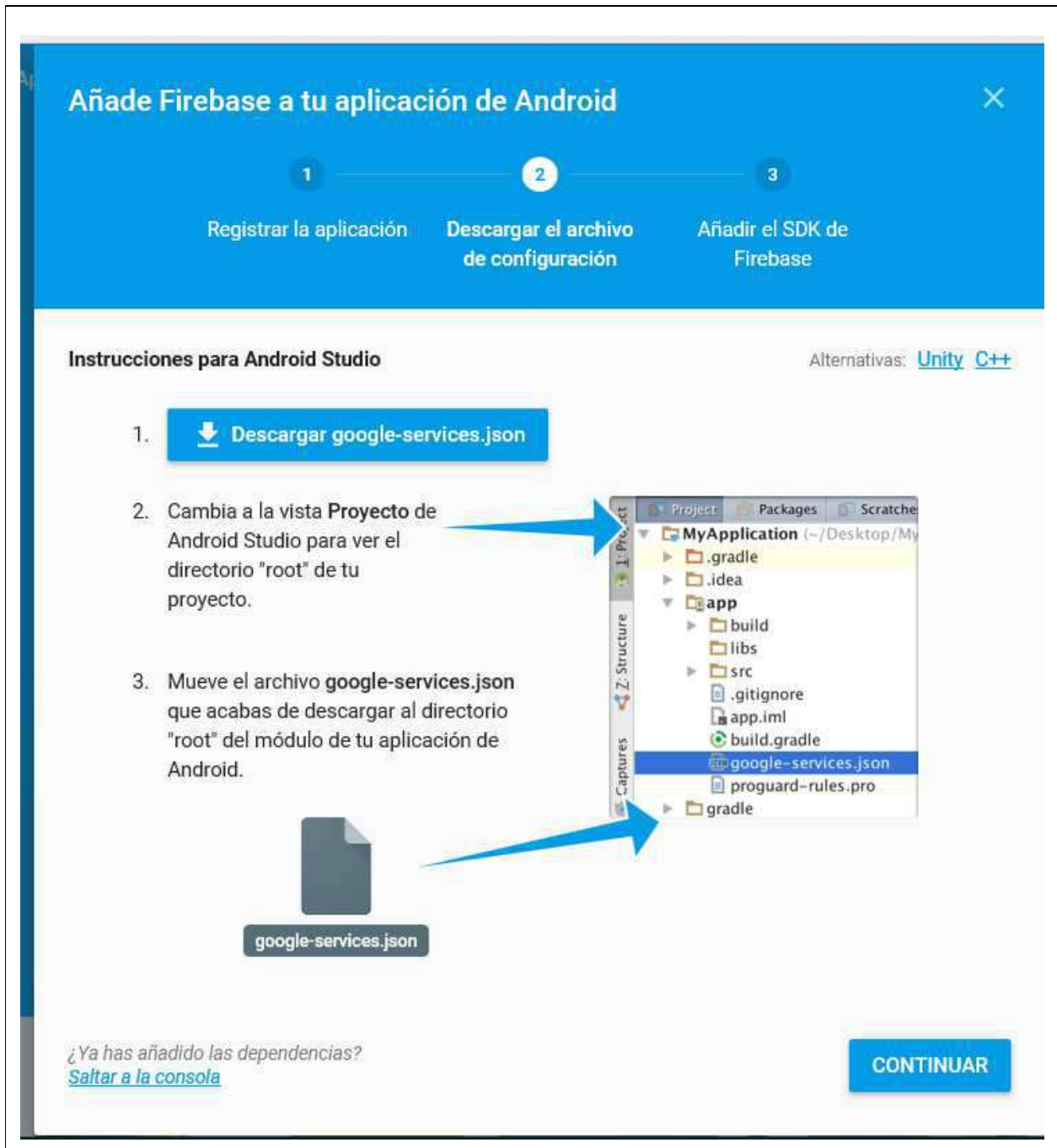


Figura 46 Ventana para agregar googleservices a nuestro proyecto

Se pulsa el botón continuar y nos manda a la última ventana de instrucciones. En esta ventana nos muestra las líneas de comando agregadas como dependencias a nuestro proyecto para que pueda utilizar los Servicios de Google y la plataforma Firebase.



Añade Firebase a tu aplicación de Android

1 Registrar la aplicación 2 Descargar el archivo de configuración 3 Añadir el SDK de Firebase

1. **build.gradle de proyecto** (<project>/build.gradle):

```
buildscript {
  dependencies {
    // Add this line
    classpath 'com.google.gms:google-services:3.2.0'
  }
}
```

2. **build.gradle de aplicación** (<project>/<app-module>/build.gradle):

```
dependencies {
  // Add this line
  compile 'com.google.firebase:firebase-core:12.0.0'
}
...
// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
```

incluye Analytics de forma predeterminada ?

3. Por último, pulsa **Sincronizar ahora** en la barra que aparece en el entorno IDE (entorno de desarrollo integrado):

Gradle files have changed since... **Sync now**

FINALIZAR

Figura 47 Ventana para agregar dependencias a nuestro proyecto

En las siguientes imágenes se muestra los cambios que se hicieron a nuestros archivos build.gradle(Module.app) según las instrucciones de Firebase.

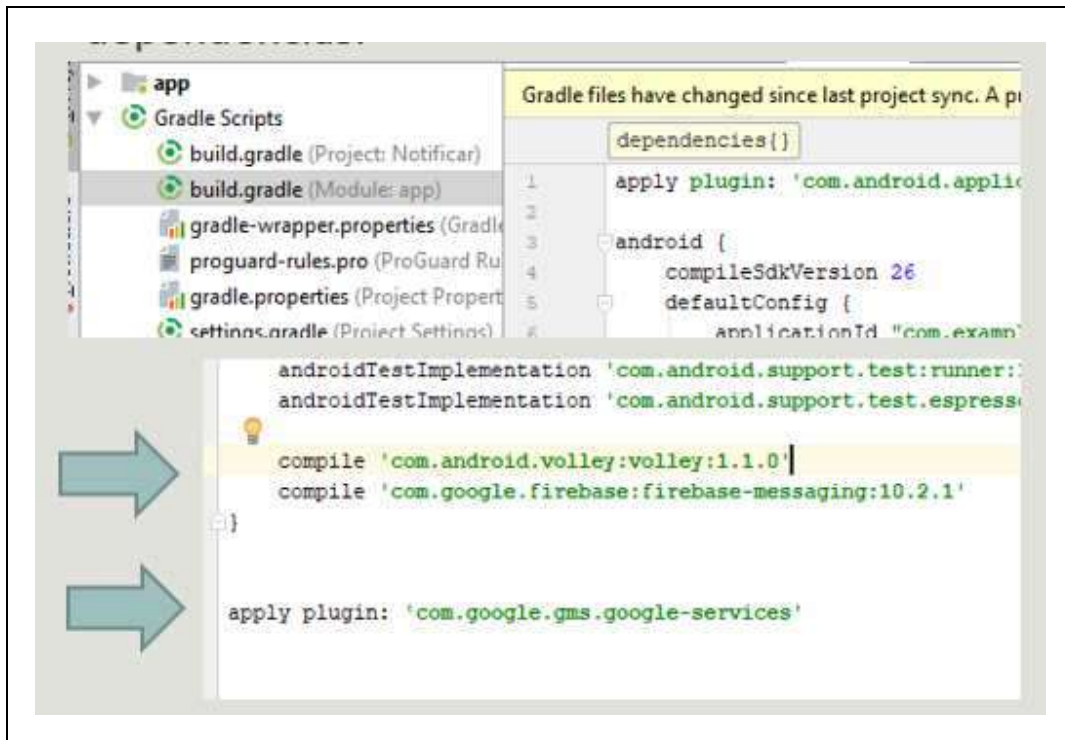


Figura 48 Cambios en dependencias build.gradle(Module.app)

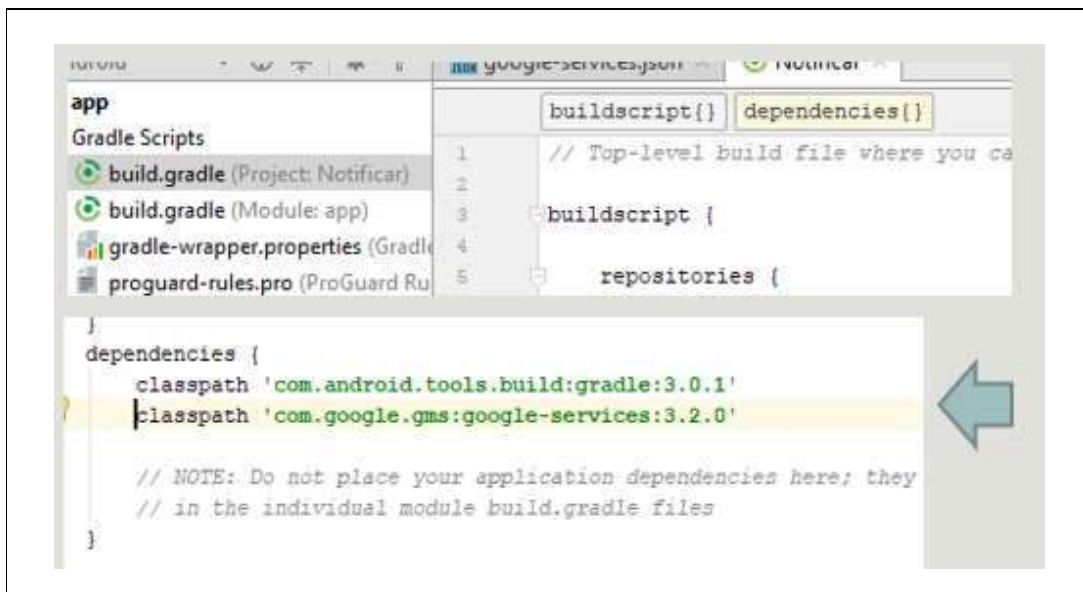


Figura 49 Cambios en el archivo build.gradle(Project.Notificar)

Eso es todo, una vez que la aplicación sincroniza los archivos necesarios nos dirigirá a esta página, donde podemos visualizar todas las opciones que la API nos ofrece (figura 50):

Proyecto de Estadía Laboratorios Texzal de México

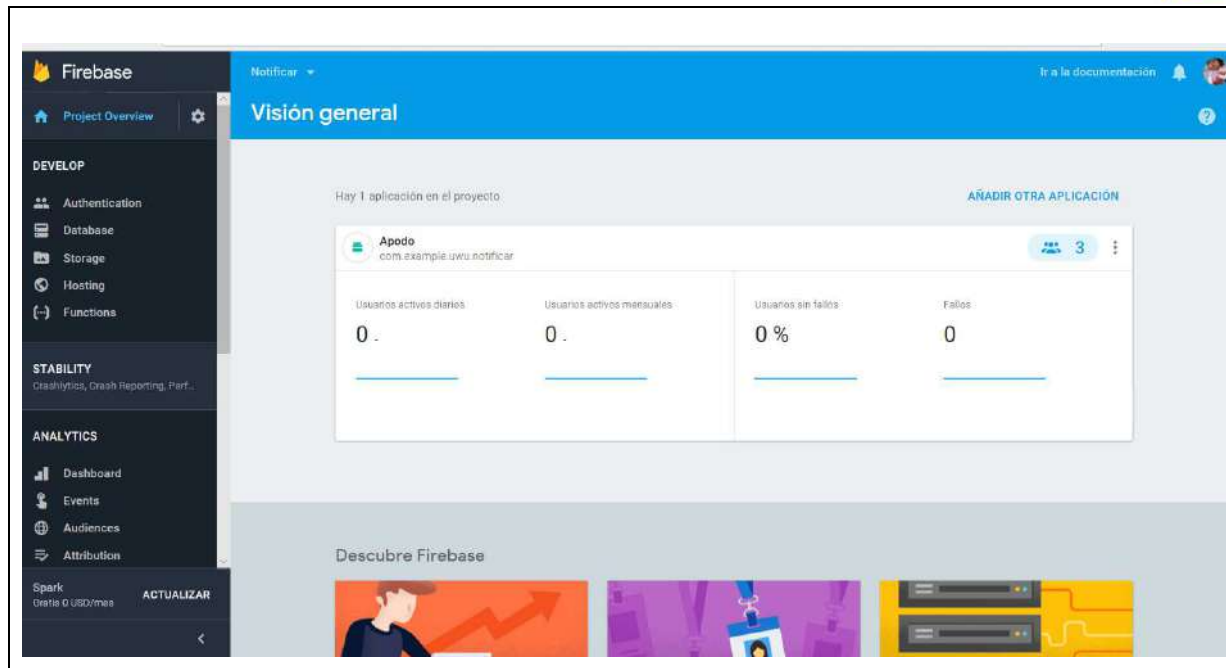


Figura 50 Página principal de Firebase

Por último se hizo una prueba para verificar que la conexión entre nuestra aplicación y la plataforma Firebase fuera exitosa. Para esto utilizamos la consola de mensajes de Firebase que nos permite enviar mensajes a un dispositivo único o por grupo, en este caso enviaremos la notificación al grupo de dispositivos que tengan instalada la aplicación. Accedemos a la consola a través del menú lateral izquierdo.

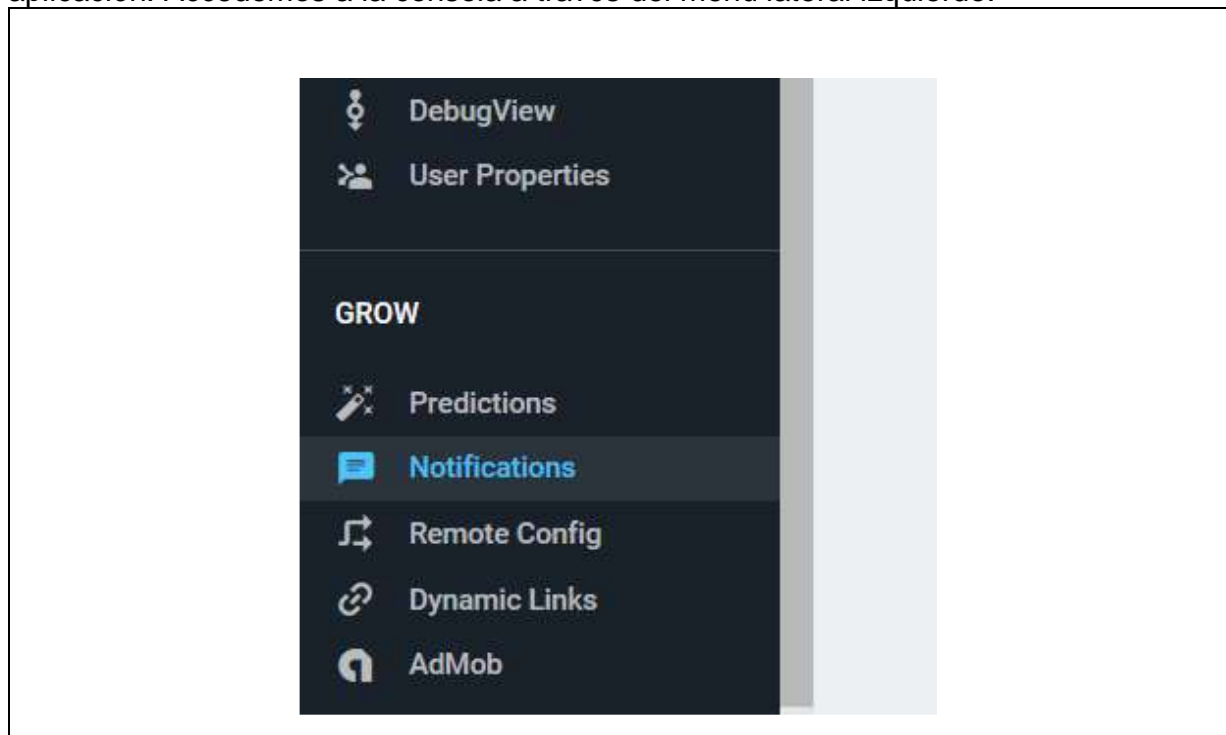


Figura 51 Panel lateral de opciones



Se abrirá la siguiente página en la que tenemos que llenar campos necesarios como el mensaje de la notificación y el destino de la notificación.

Figura 52 Vista para mandar un mensaje desde consola en Firebase

Después pulsaremos el botón “Enviar mensaje” y la notificación debería llegar a nuestro dispositivo en cuestión de par de segundos.

Figura 53 Botón “Enviar mensaje”

Proyecto de Estadía Laboratorios Texzal de México

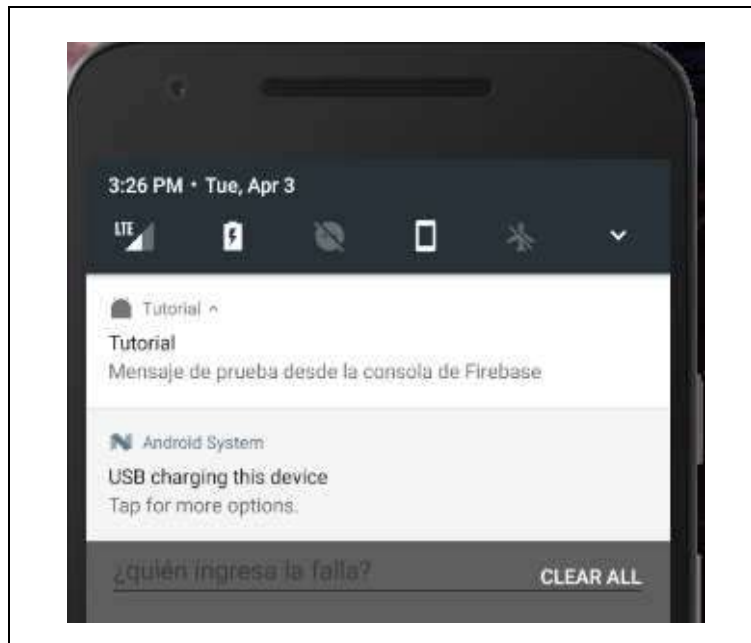


Figura 54 Mensaje recibido en dispositivo

El siguiente paso fue la búsqueda de información en las páginas oficiales de Android Studio y Firebase acerca de cómo enviar y recibir notificaciones en tiempo real entre dispositivos, inicialmente de dispositivo a dispositivo. Afortunadamente las páginas oficiales nos brindan información en nuestro idioma y existe mucha información en páginas externas. Se inició con la información de Firebase:



Figura 55 Página de configuración de Firebase Cloud Messaging

Lo primero es configurar el archivo AndroidManifest.xml que contiene todas las actividades y servicios que ocupará nuestra aplicación, por lo que hay que agregar una línea que nos permite usar la conexión a internet del dispositivo donde esté instalada la aplicación.



```
manifest application
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.teban.tutorial">
4
5     <uses-permission android:name="android.permission.INTERNET"></uses-permission>
6
7
8     <application
        ...
        <service android:name=".MiFirebaseServicioMensaje">
            <intent-filter>
                <action android:name="com.google.firebase.MESSAGING_EVENT"></action>
            </intent-filter>
        </service>
        <service android:name=".MiFirebaseInstance">
            <intent-filter>
                <action android:name="com.google.firebase.INSTANCE_ID_EVENT"></action>
            </intent-filter>
        </service>
    </application>
</manifest>
```

Figura 56 Configuración de archivo "AndroidManifest.xml"

En la parte de "android:name" se debe agregar el nombre de la clase que va a manejar ese servicio, por lo que se implementaron las siguientes clases.

MiFirebaseServicioMensaje.java: Esta clase nos permite recibir un mensaje que haya sido enviado desde la consola u otro dispositivo. Esta clase también contiene un método que nos permite mostrar la notificación en el dispositivo con los datos enviados como el título y el mensaje de la notificación.

MiFirebaseInstance.java: Esta clase nos permite recibir un id conocido como "Token" el cual es único y es asignado por Firebase al dispositivos donde se instala la aplicación. También contiene un método que guarda este Token en la base de datos para identificar a ese smarphone.

Proyecto de Estadía Laboratorios Texzal de México



```
MainActivity.java | AndroidManifest.xml | app | Tutorial | MiFirebaseServicioMensaje.java | MiFirebaseInstance.java | notificacion.xml | content_main.x
MiFirebaseServicioMensaje | onMessageReceived()
import com.google.firebase.messaging.FirebaseMessagingService;
import com.google.firebase.messaging.RemoteMessage;

/**
 * Created by teban on 23/01/17.
 */

public class MiFirebaseServicioMensaje extends FirebaseMessagingService {

    @Override
    public void onMessageReceived(RemoteMessage remote)
    {
        super.onMessageReceived(remote);
        if(remote.getNotification() != null)
        {
            Log.d( tag: "Notificacion", msg: "Notificacion: "+ remote.getNotification().getBody()+ ", titulo: "+ remote.getNotification().getTitle());
            verNotificacion(remote.getNotification().getTitle(),remote.getNotification().getBody());
        }
    }

    private void verNotificacion(String titulo, String cuerpo)
    {
        Intent i = new Intent( packageContext: this, MainActivity.class);
        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        PendingIntent pe = PendingIntent.getActivity( context: this, requestCode: 0, i, PendingIntent.FLAG_ONE_SHOT);
        Uri soundUri = RingtoneManager.getDefaultUri( RingtoneManager.TYPE_NOTIFICATION);

        NotificationCompat.Builder noti = new NotificationCompat.Builder( context: this)
            .setSmallIcon(R.mipmap.ic_launcher)
            .setContentTitle(titulo)
            .setContentText(cuerpo)
            .setAutoCancel(true)
            .setSound(soundUri)
            .setContentIntent(pe);
        NotificationManager notifi = (NotificationManager) getSystemService( Context.NOTIFICATION_SERVICE);
        notifi.notify( id: 0, noti.build());
    }
}
```

Figura 57 Clase “MiFirebaseServicioMensaje.java”

```
MainActivity.java | AndroidManifest.xml | app | Tutorial | MiFirebaseServicioMensaje.java | MiFirebaseInstance.java
1 package com.example.teban.tutorial;
2
3 import ...
16
17 /**
18  * Created by teban on 23/01/17.
19  */
20
21 public class MiFirebaseInstance extends FirebaseInstanceIdService {
22
23     @Override
24     public void onTokenRefresh()
25     {
26         super.onTokenRefresh();
27         Variables.TOKEN = FirebaseInstanceId.getInstance().getToken();
28         Log.d( tag: "Token", Variables.TOKEN);
29         enviarToken();
30     }
31
32     private void enviarToken() {...}
33
34 }
65
66
```

Figura 58 Clase “MiFirebaseInstance.java”

Con esto terminado, solo queda agregar más instrucciones para que la aplicación mande una notificación cuando se guarde un nuevo registro en la base de datos.

Prueba de aceptación “Mandar notificación a diferentes dispositivos”
Número: 4
Historia: Mandar notificación local
Fecha de historia: 12/02/2018
Fecha de término: 17/02/2018
Descripción: La aplicación necesita mandar notificaciones a distintos dispositivos en tiempo real cuando se agrega un registro a la base de datos .
Resultado esperado: Los resultados al implementar esta funcionalidad fueron los esperados.

Tabla 15 Prueba de aceptación “Mandar notificación a diferentes dispositivos”

A continuación se muestra la interfaz terminada en esta iteración:

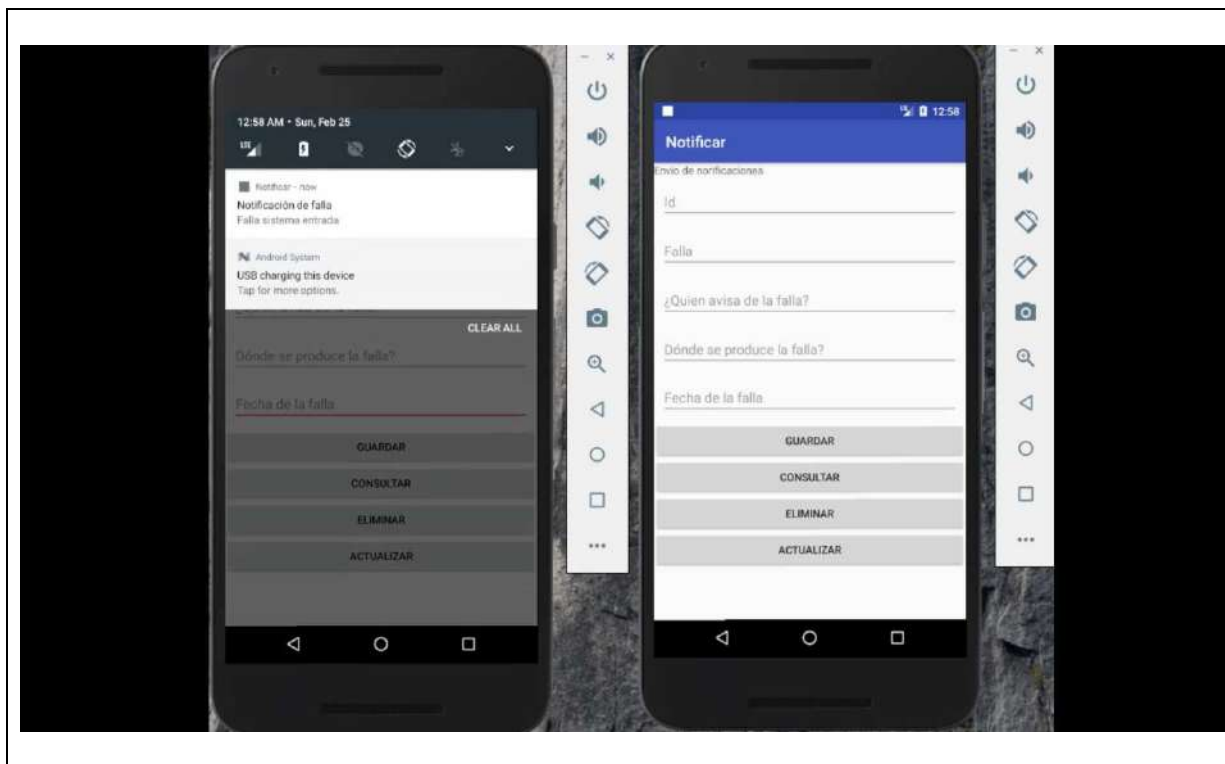


Figura 59 Iteración “Mandar notificación a diferentes dispositivos”

Prueba aceptación de historia de usuario “Mandar notificación a diferentes dispositivos”:

- ✓ La aplicación manda una notificación al guardar datos en la base de datos
- ✓ La aplicación manda la notificación a través del servicio de Firebase con los datos escritos en los campos
- ✓ La notificación llega inmediatamente con los datos proporcionados

3.3.5 Historia de usuario 5

Historia N° 5	Tipo	Dificultad	Esfuerzo		Prioridad	Notas
Tiempo de espera en notificaciones	Nuevo Arreglar Mejorar	Fácil	Estimado	Gastado	1	
		Moderado	4 días	5 días		
Descripción						
La aplicación debe mandarla notificación cuando se agrega una nueva falla a la base de datos pero no inmediatamente, se debe cumplir un tiempo para que se hace el envío.						
Fecha	Estado	Comentario				
19/02/2018	Definición					
19 - 24	Implementación					
24/02/2018	Hecho	Se implementó la funcionalidad con éxito				

Tabla 16 Historia de usuario "Tiempo de espera en notificaciones"

Para esta iteración lo primero fue investigar acerca de las tareas en segundo plano en Android Studio, y se encontró que se pueden realizar servicios en segundo plano y crear "AsyncTask". Una tarea asincrónica (**Asynctask**) se define por un cálculo que se ejecuta en un hilo secundario y cuyo resultado queremos que se publique en el hilo del interfaz de usuario. Sus métodos básicos son los siguientes:

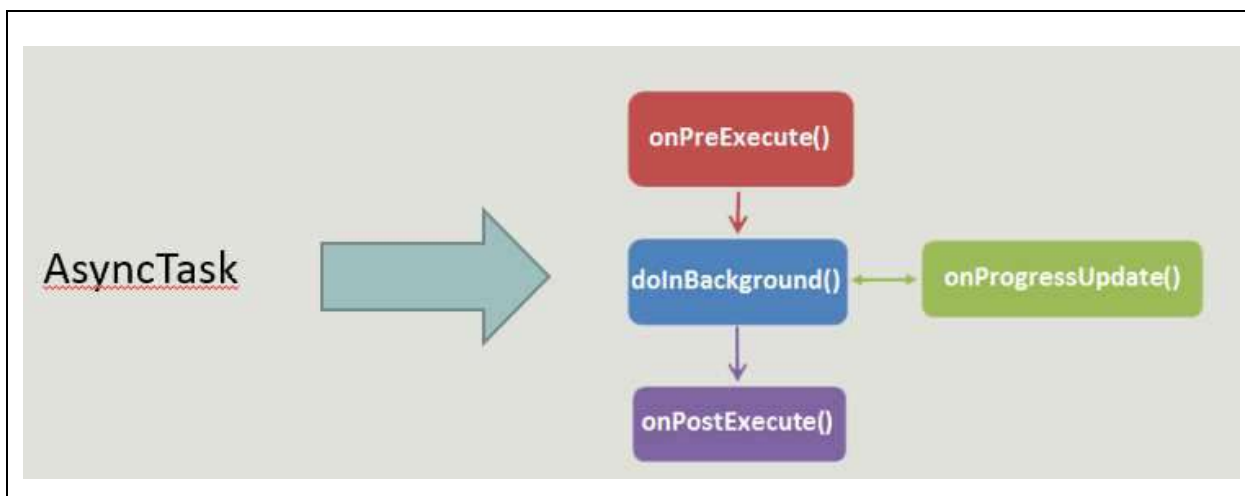


Figura 60 Métodos básicos de la clase AsyncTask

Declaramos una clase que extienda de AsyncTask para que nos proporcione los métodos necesarios, del cual se uso "OnPostExecute" y "doInBackground"



```
143      editTextId.setText("");
144      editTextFallaM.setText("");
145      editTextDóndeFalla.setText("");
146      editTextQuiénFalla.setText("");
147      editTextFechaFalla.setText("");
148
149    }
150  }
151
152  public class Time extends AsyncTask<Void, Integer, Boolean>{
153    @Override
154    protected void onPostExecute(Boolean aBoolean) {
155      //ejecutar():
156      enviarNotificacion();
157      Toast.makeText( context MainActivity.this, text "Notificación enviada a nivel 1 y 2", Toast.LENGTH_SHORT).show();
158    }
159  }
160
161  @Override
162  protected Boolean doInBackground(Void... voids) {
163    for (int i=0; i<3; i++)
164      hiloTiempoNotificacion();
165    return true;
166  }
167  }
```

Figura 61 Implementación de AsyncTask 1

OnPostExecute: Para mandar a ejecutar el método que manda la notificación por nivel.
doInBackground: Para mandar a ejecutar el método que cuenta el tiempo antes de mandar la notificación.

```
MainActivity
104
105 }
106 }):
107
108 }
109
110 public void hiloTiempoNotificacion () {
111   try {
112     Thread.sleep( millis: 10000);
113   }catch (InterruptedException e){
114
115   }
116 }
117
118
119
120
```

doInBackground →

Figura 62 Implementación de AsyncTask 2

Solo falta inicializar la clase cuando se mandan los datos de la falla para ser guardado en la base de datos, esto hará que el conteo empiece luego de guardar los datos correctamente.



```

123 private class CargarDatos extends AsyncTask<String, Void, String> {
124     @Override
125     protected String doInBackground(String... urls) {
126
127         //los parametros vienen de la llamada execute(): el parametro[0] es el url.
128         try {
129             return downloadUrl(urls[0]);
130         } catch (IOException e) {
131             return "No ha sido posible cargar los datos";
132         }
133     }
134     // onPostExecute muestra los resultados del AsyncTask.
135     @Override
136     protected void onPostExecute(String result) {
137         Toast.makeText(getApplicationContext(), text "Se almacenaron los datos correctamente", Toast.LENGTH_LONG).show()
138         //Log.d("LO TOKENS", "kdkdkd="+variables.cursor.getString(0));
139         Time tiempoDeEspera = new Time();
140         tiempoDeEspera.execute();
141
142
143         editTextId.setText("");
144         editTextFallaM.setText("");
145         editTextDóndeFalla.setText("");
146         editTextQuiénFalla.setText("");
147         editTextFechaFalla.setText("");
148     }
149 }
150

```

Figura 63 Implementación de AsyncTask 3

Se probó la implementación de esta clase a nuestro proyecto para verificar que la aplicación hacía lo que estábamos buscando, lo cual resultó exitosamente.

Prueba de aceptación “Tiempo de espera en notificaciones”
Número: 5
Historia: Tiempo de espera en notificaciones
Fecha de historia: 19/02/2018
Fecha de término: 24/02/2018
Descripción: La aplicación necesita mandar notificaciones a distintos dispositivos pero no inmediatamente, si no que la aplicación debe esperar un tiempo determinado para mandar la notificación a los niveles correspondientes.
Resultado esperado: Los resultados al implementar esta funcionalidad fueron los esperados.

Tabla 17 Prueba de aceptación “Tiempo de espera en notificaciones”

A continuación se muestra la interfaz terminada en esta iteración:

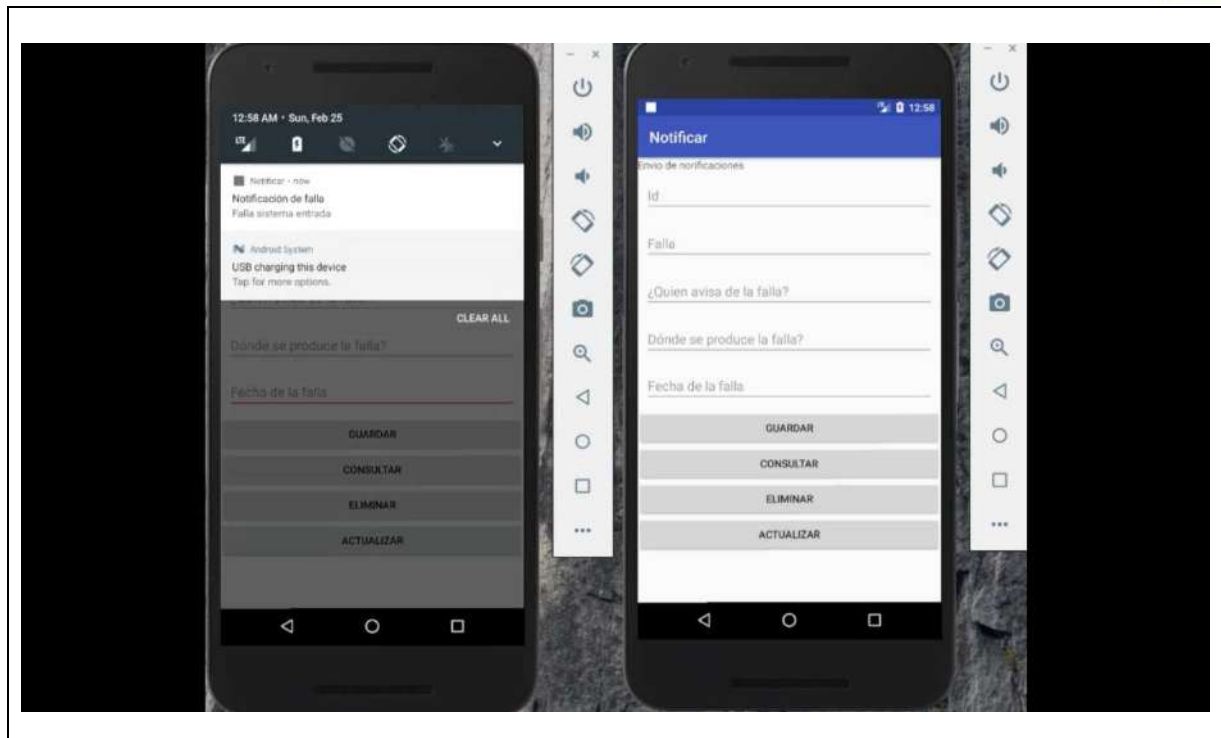


Figura 64 Iteración “Tiempo de espera en notificaciones”

Prueba aceptación de historia de usuario “Tiempo de espera entre notificaciones”:

- ✓ La aplicación inicia una tarea en segundo plano que permite pausar el envío de la notificación con la pausa de un hilo de ejecución dentro de la aplicación.
- ✓ La aplicación manda la notificación con los datos escritos en los campos a través de Firebase
- ✓ La notificación llega inmediatamente con los datos proporcionados a los dispositivos con la aplicación instalada.



Proyecto de Estadía
Laboratorios Texzal de México

3.3.6 Historia de usuario 6

Historia N° 6	Tipo	Dificultad	Esfuerzo		Prioridad	Notas
Login	+ Nuevo	Fácil	Estima	Gastad	1	
Agregar usuario	Arreglar	Moderado	do	o		
Interfaz de usuario	+ Mejorar	Difícil	4 días	5 días		
Descripción						
Se necesita que la aplicación pueda permitir y denegar su uso mediante un login, así como agregar usuarios nuevos al sistema.						
Fecha	Estado	Comentario				
26/02/2018	Definición					
26- 03	Implementación					
03/03/2018	Hecho	Se implementó la funcionalidad con éxito				

Tabla 18 Historia de usuario "Login + Agregar usuario + Interfaz de usuario"

En esta iteración el cliente le pidió al equipo de trabajo que se trabajara en el acceso al sistema, el agregado de un usuario nuevo y de la apariencia que tendrá finalmente la aplicación. Inicialmente necesitamos implementar una base de datos para nuestra aplicación la cual nos va a permitir almacenar los datos de todos los usuarios registrados en el sistema. Esta es la primera propuesta aceptada como base de datos:

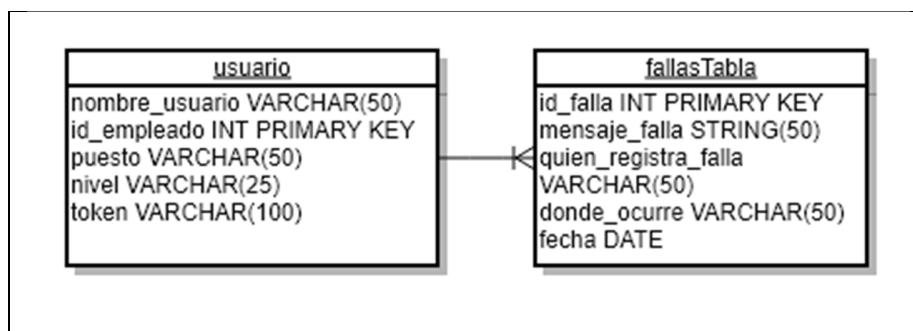


Figura 65 Primer versión de base de datos

Después se implementó un SplashScreen para dar inicio a la aplicación. Para la interfaz se utilizaron colores de acuerdo a los colores de los Laboratorios Texzal de México. En la siguiente imagen se muestra esta actividad.



Proyecto de Estadía Laboratorios Texzal de México



Figura 66 "SplashScreen"

Inmediatamente después del SplashScreen se colocó la actividad que hará el papel de Login en la aplicación para la administración de acceso al sistema.

Proyecto de Estadía Laboratorios Texzal de México



Figura 67 Actividad "Login"

A continuación se muestra la actividad que permite agregar usuarios nuevos a nuestra aplicación.

Proyecto de Estadía Laboratorios Texzal de México

A screenshot of a mobile application interface for adding a new user. The title bar at the top reads "Laboratorios Texzal de México" and shows the time as 17:47. Below the title bar is a dark green header with a white icon of a person and a plus sign. The main content area contains two text input fields: "Nombre de usuario" and "Contraseña nueva". Below these are two dropdown menus: "Producción" and "Nivel 1". At the bottom is a grey button labeled "AGREGAR USUARIO".

Figura 68 Actividad "Agregar usuario nuevo"

Después del inicio de sesión con un usuario válido desde la actividad Login, se accesa a la actividad "Menú principal" la cual nos va a permitir navegar por las opciones con las que contará la aplicación.

Proyecto de Estadía Laboratorios Texzal de México

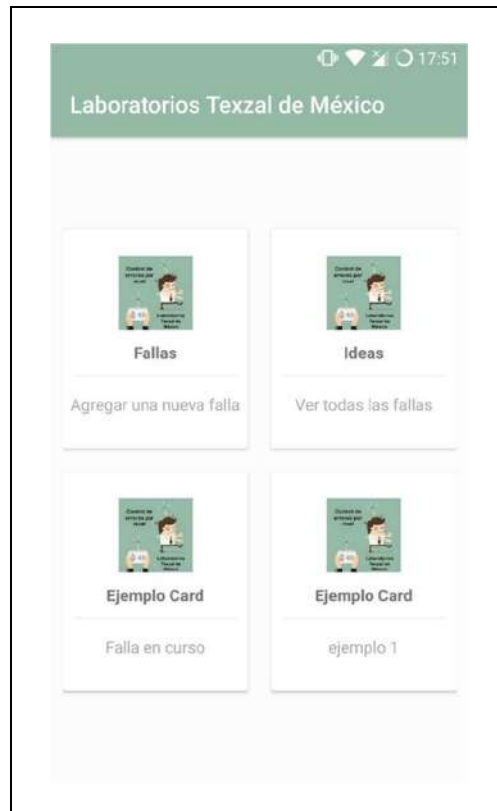


Figura 69 Actividad "Menú principal"

La primera opción en el menú principal es el Envío de nuevas fallas a la base de datos de la aplicación, el cual ha quedado con la siguiente apariencia hasta esta semana:

Proyecto de Estadía Laboratorios Texzal de México

A screenshot of a mobile application interface for reporting faults. The title bar is green and contains the text "Laboratorios Texzal de México". Below the title bar, the screen displays the heading "Envío de fallas". There are four input fields: the first contains the letter "d", the second is labeled "Falla", the third is labeled "Dónde se produce la falla?", and the fourth is labeled "Fecha de la falla". At the bottom of the form, there are four buttons: "GUARDAR", "ELIMINAR", "CONSULTAR", and "ACTUALIZAR". The status bar at the top shows signal strength, Wi-Fi, and the time 17:51.

Figura 70 Actividad "Envío de fallas"

La segunda opción en el menú principal nos permite visualizar la información de todas las fallas que se han registrado en la aplicación.

Proyecto de Estadía
Laboratorios Texzal de México



Figura 71 Actividad “Lista de fallas”

Prueba de aceptación “Login + Agregar usuario + Diseño de interfaz”
Número: 1
Historia: Mandar notificación local
Fecha de historia: 26/02/2018
Fecha de término: 03/03/2018
Descripción: La aplicación necesita permitir y denegar el acceso a sus funciones mediante un login, así como un apartado donde se puedan agregar usuarios nuevos con los datos correspondientes y únicamente necesarios para la aplicación.
Resultado esperado: Los resultados al implementar esta funcionalidad fueron los esperados. Solo se requieren modificaciones en algunas interfaces.

Tabla 19 Prueba de aceptación “Login + Agregar usuario + Diseño de interfaz”

A continuación se muestra la interfaz terminada en esta iteración:

Proyecto de Estadía Laboratorios Texzal de México

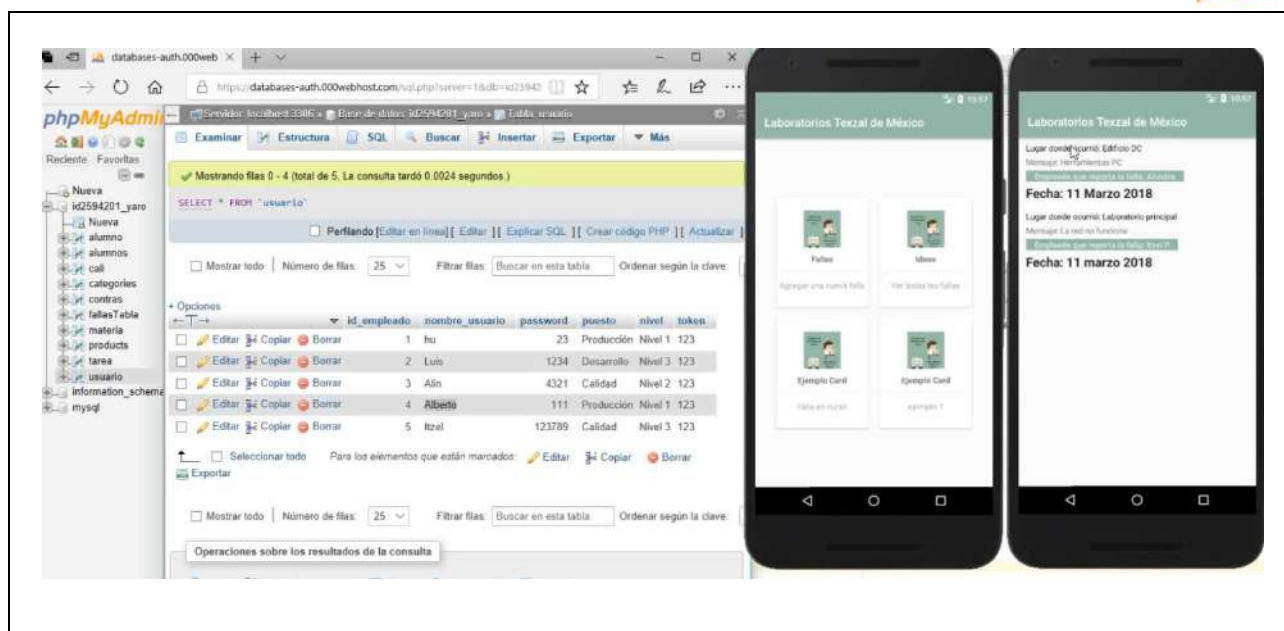


Figura 72 Iteración “Login + Agregar usuario + Diseño de interfaz”

Prueba aceptación de historia de usuario “Login + Agregar usuario + Diseño de interfaz”:

- ✓ La aplicación puede denegar y aceptar el acceso de los diferentes usuarios mediante un usuario y contraseña.
- ✓ La aplicación agrega satisfactoriamente nuevos usuarios listos para utilizar en la aplicación.
- ✓ La aplicación cumple con los colores institucionales y gusta al usuario final.
- ✓ La aplicación cuenta con un menú de opciones vistoso, legible y simple.
- ✓ La aplicación cuenta con una actividad que permite visualizar todas las fallas agregadas al sistema en manera de vista.

3.3.7 Historia de usuario 7

Historia N° 7	Tipo	Dificultad	Esfuerzo	Prioridad	Notas
Modificaciones a interfaz de usuario	Nuevo Arreglar Mejorar	Fácil	Estimado	1	
		Moderado	Gastado		
		Difícil	4 días	5 días	
Descripción					
Se necesita hacer algunos cambios a la interfaz de la aplicación.					
Fecha	Estado	Comentario			
05/03/2018	Definición				
05 - 10	Implementación				
10/03/2018	Hecho	Se implementó la funcionalidad con éxito			

Tabla 20 Historia de usuario “Modificaciones a interfaz de usuario”

Proyecto de Estadía Laboratorios Texzal de México

En esta iteración el cliente le pidió al equipo de trabajo que se mejoraran algunos aspectos gráficos de la interfaz, por ejemplo en la actividad de “Enviar una falla”, donde hay botones innecesarios. Lo primero fue desactivar la actividad SplashScreen ya que tomaba tiempo de inicio y dialogando con el cliente se llegó a la conclusión de que no era necesario, por lo que en esta iteración la aplicación ya no cuenta con SplashScreen y accede directamente a la actividad Login.

En la actividad Login se pidió que la aplicación pudiera mantener una sesión de usuario iniciada aun saliendo de la aplicación, cosa que no sucedía anteriormente, por lo que fue modificada haciendo uso de las herramientas disponibles en Android llamado “SharedPreferences”.

SharedPreferences es una clase que te permite almacenar datos de la aplicación de manera local en cada uno de los dispositivos, por lo que se usó para verificar cada vez que se inicia la aplicación si una sesión está ya abierta y se abra automáticamente el menú principal en lugar de la actividad Login.

Para esto, se agregó un campo para saber si el usuario requiere mantener la sesión abierta o no.

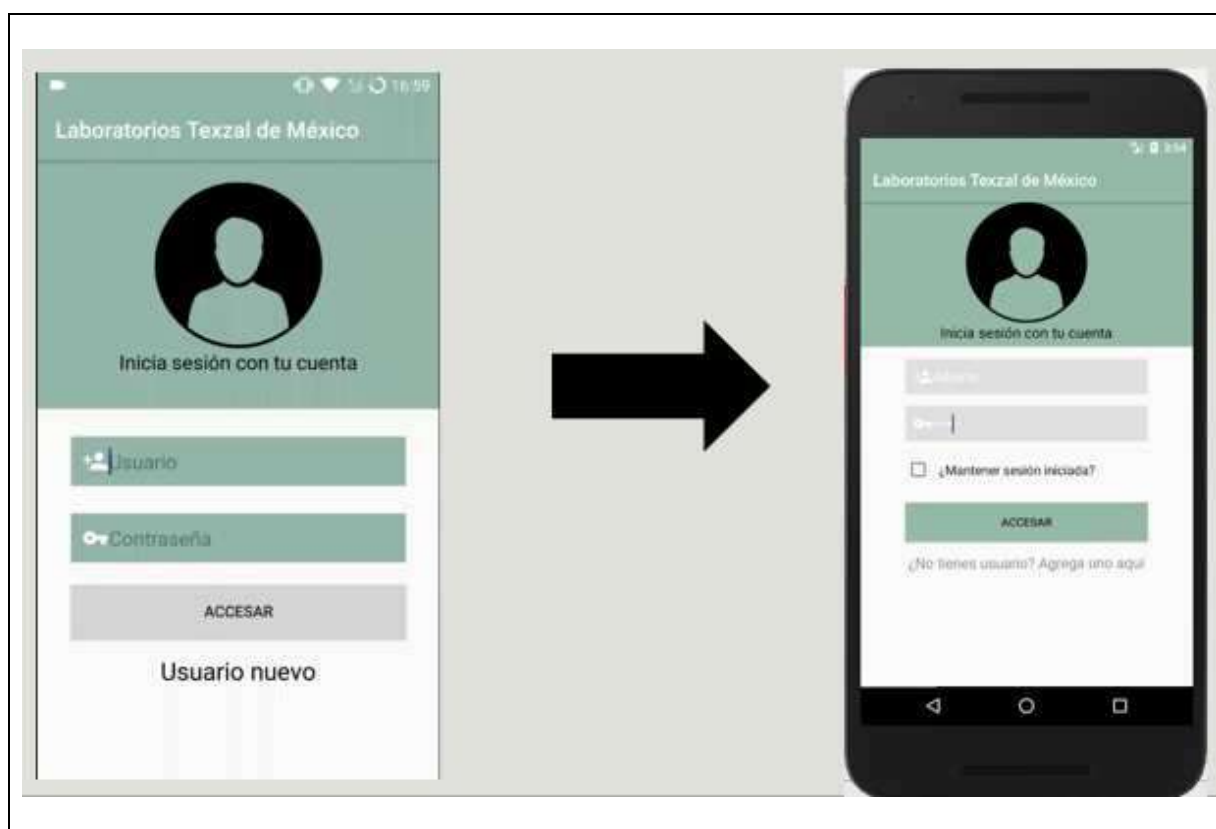


Figura 73 Modificar actividad “Login”

El siguiente cambio a la interfaz fue en la actividad “Enviar falla” pues tenía botones innecesarios y se pidió que se mostrara un botón rojo con el texto “Reportar falla”, lo cual se realizó.

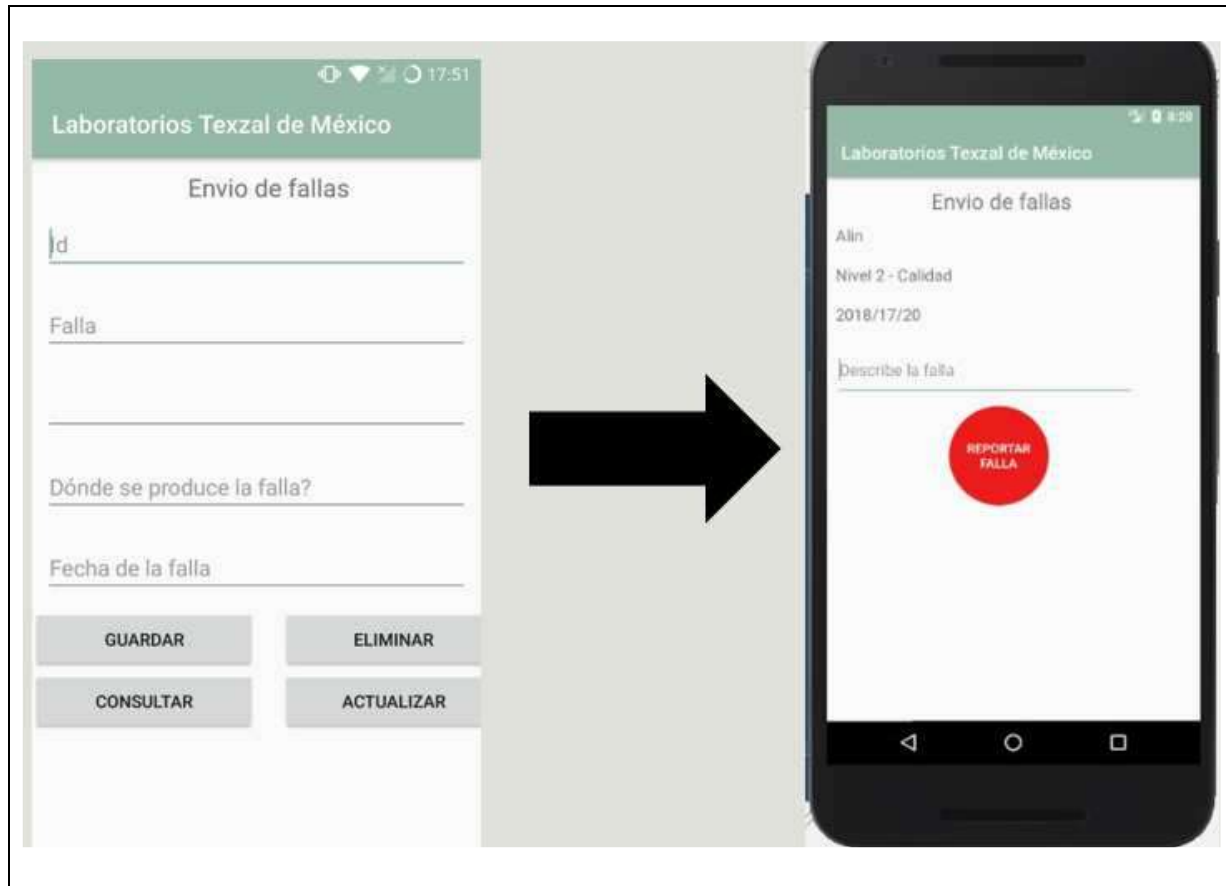


Figura 74 Modificar actividad “Enviar falla” 1

Se hizo entrega de estas interfaces y el cliente nos pidió solo una modificación más a la actividad “Enviar falla” pues ellos solo quieren que se muestre un botón rojo y grande que tenga el texto “Enviar falla” por lo que la siguiente interfaz fue la finalmente aceptada por el cliente.

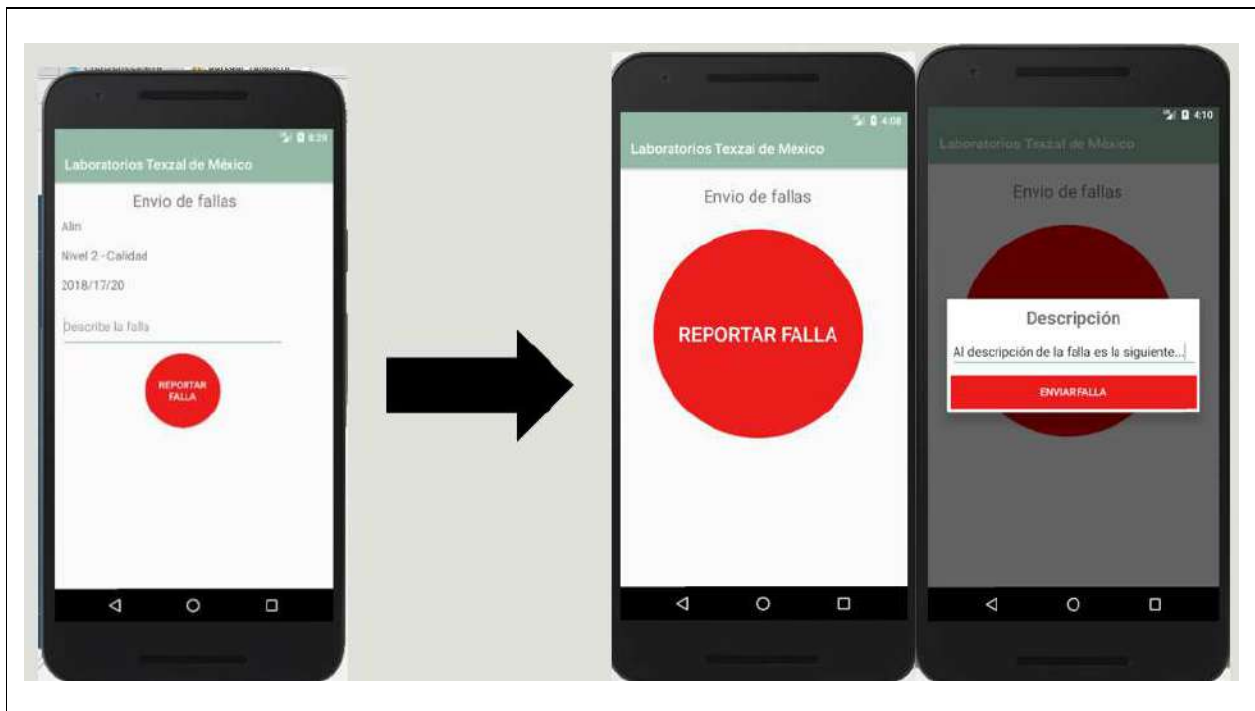


Figura 75 Modificar actividad "Enviar falla" 2

En esta semana también se realizó el cambio de vista a la actividad que muestra todas las fallas usando herramientas Material Design para separar cada una de las fallas en CardViews que son más vistosas que una simple lista en la pantalla.

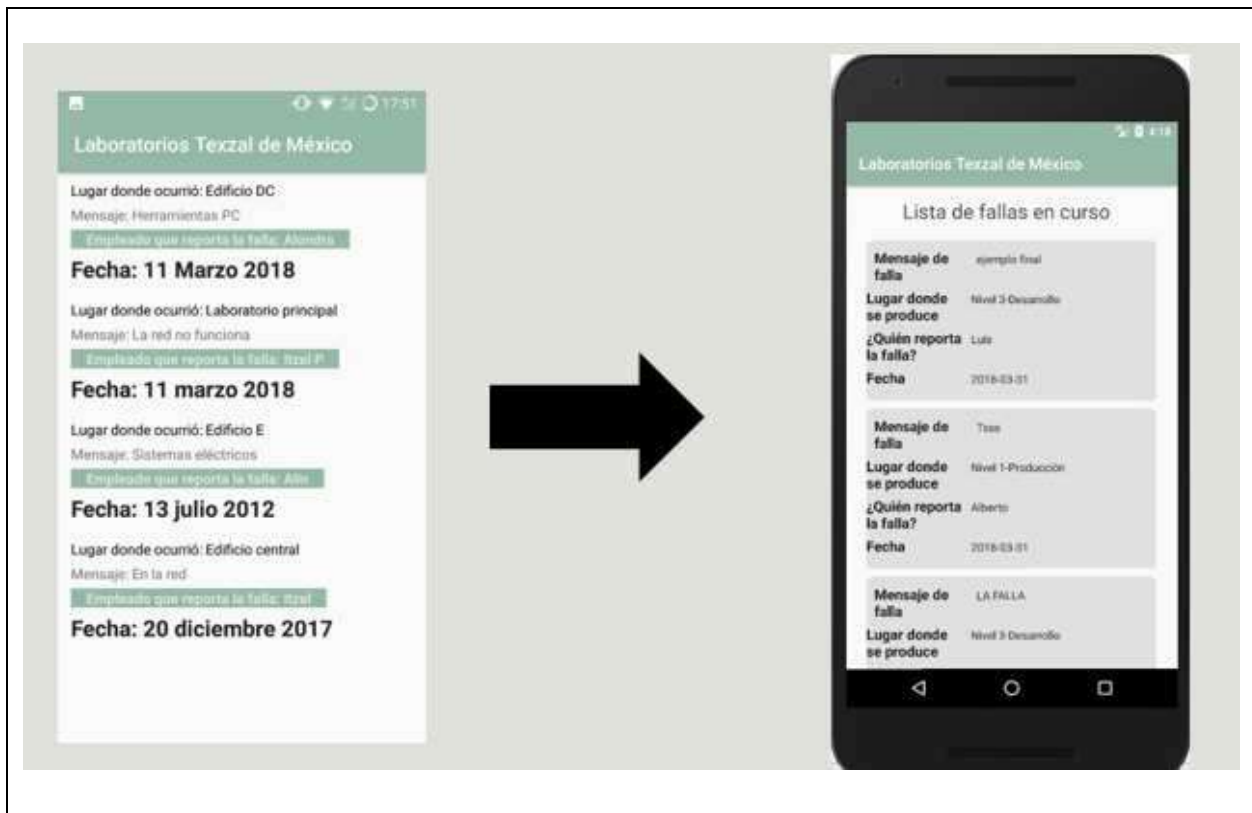


Figura 76 Modificar actividad “Mostrar todas las fallas”

Prueba de aceptación “Modificar interfaz de usuario”
Número: 1
Historia: Mandar notificación local
Fecha de historia: 05/03/2018
Fecha de término: 10/03/2018
En esta iteración se atienden distintas modificaciones en la interfaz de usuario según los requisitos y comentarios del cliente.
Resultado esperado: Los resultados al implementar esta funcionalidad fueron los esperados.

Tabla 21 Prueba de aceptación “Modificar interfaz de usuario”

Las interfaces que se mostraron anteriormente son la iteración terminada.

Prueba aceptación de historia de usuario “Modificaciones en interfaz de usuario”:

- ✓ La aplicación puede mantener una sesión de usuario activa aunque se cierre la aplicación y cierra una sesión al cerrar la aplicación si así se desea.
- ✓ La aplicación cumple con los colores institucionales y gusta al usuario final junto con las nuevas vistas en los apartados de Enviar falla y Mostrar todas las fallas.



Proyecto de Estadía
Laboratorios Texzal de México

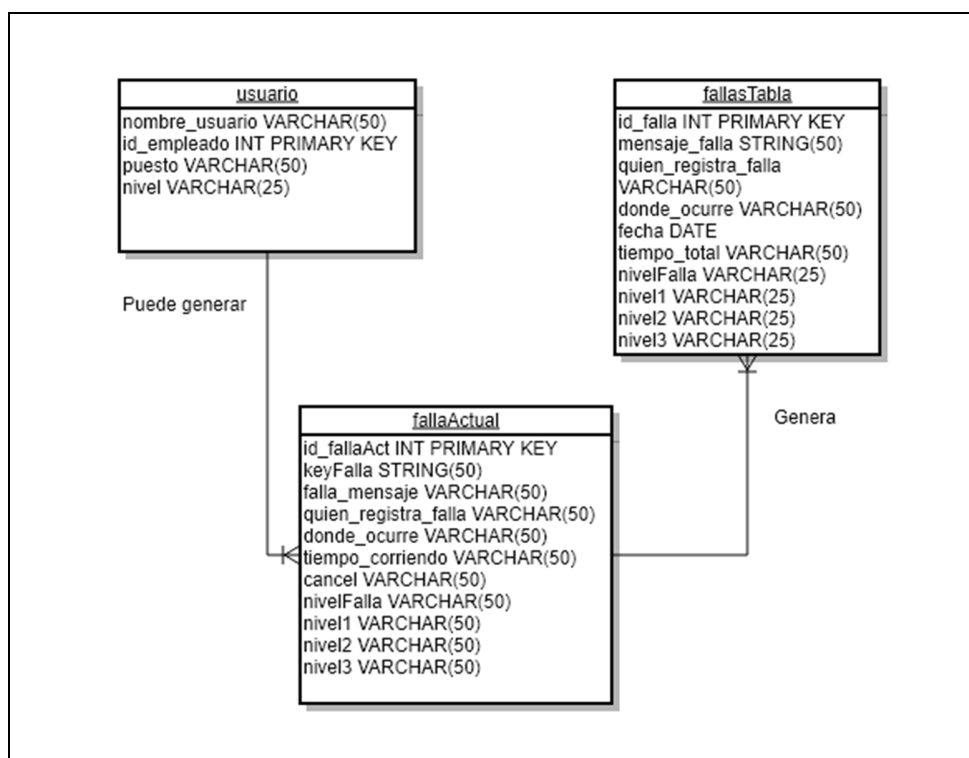
3.3.8 Historia de usuario 8

Historia N° 8	Tipo	Dificultad	Esfuerzo	Prioridad	Notas
Fallas en curso	Nuevo Arreglar Mejorar	Fácil Moderado Difícil	Estima do 10 días	Gastad o 12 días	1
Descripción					
La aplicación debe poder mantener una falla en curso en cuanto se da de alta para saber el tiempo total en que se fue resuelta la falla					
Fecha	Estado	Comentario			
12/03/2018	Definición				
12 - 24	Implementación				
24/03/2018	Hecho	Se implementó la funcionalidad con éxito			

Tabla 22 Historia de usuario "Fallas en curso"

La aplicación ya puede almacenar y mostrar los datos ingresados al registrar una falla nueva para la base de datos, pero se necesita que las fallas que se agreguen primero se pongan en curso pues deben ser detenidas por distintos usuarios para comprobar que la falla fue resuelta con éxito, así logrando cumplir los requerimientos funcionales **RF06**, **RF07** y **RF08**.

El primero paso fue el análisis del problema, por lo que la primera acción fue modificar la base de datos para guardar una falla activa y temporal, para cuando esta falla termine, se guarde en una tabla temporal de fallas terminadas con todos los datos del suceso. También se agregó una tabla únicamente. La siguiente imagen muestra los cambios en la base de datos:





Proyecto de Estadía Laboratorios Texzal de México

Figura 77 Nueva base de datos

Una vez realizado el cambio en la base de datos de nuestro servidor remoto, se empezó por desarrollar la actividad que mostrará las fallas en curso al ser dadas de alta en la actividad “Enviar falla”. El resultado fue el siguiente:



Figura 78 Actividad “Mostrar falla en curso”

Se agregó un botón a los CardView que genera cada una de las fallas en curso que lleva a una actividad donde se ven los detalles de la falla así como los botones que tan acción de cancelar esa falla en curso y a dar por terminada una falla en curso para ser almacenada temporalmente en la “Lista de fallas”. La siguiente imagen muestra la actividad encargada de esto:

Proyecto de Estadía Laboratorios Texzal de México



Figura 79 Actividad “Detalles de falla en curso”

Al agregar una falla, el sistema la registra como falla de nivel 2, lo cual hace que una falla pueda ser detenida por un usuario de nivel 1 y uno de nivel 2. Pasado un tiempo, se manda notificación a nivel 3, significa que la falla ha pasado a nivel 3 y se necesita de un usuario de cada nivel para detener la falla.

A continuación se mostrará una imagen del diálogo que aparece en la pantalla del usuario cuando se pulsa el botón “Detener esta falla”:

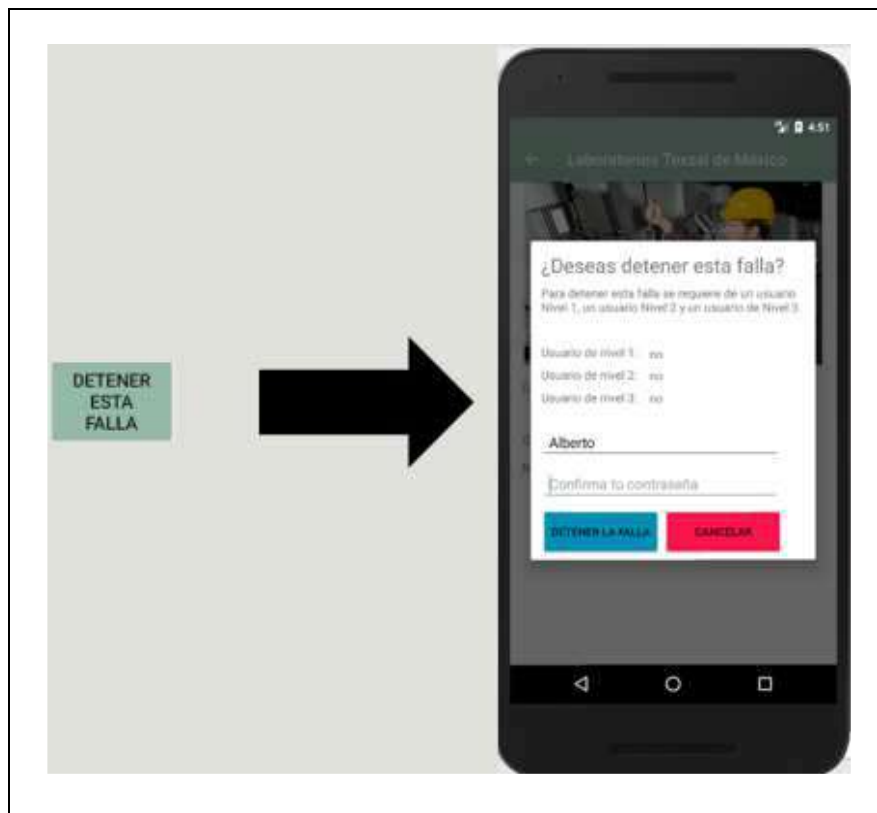


Figura 80 Botón “Detener esta falla”

En esta parte es donde cumplimos con el requerimiento principal de la aplicación, pues es donde se dan por terminadas las fallas solo cuando los usuarios, según el nivel de la falla, introducen sus contraseñas para confirmar que la falla fue resuelta satisfactoriamente.

Pero también puede suceder que se agregue una falla a “En curso” y se tenga la necesidad de cancelarla, esto puede hacerse en esta misma interfaz. El botón “Cancelar esta falla”, cancela la falla en curso en la que se esté y elimina todos sus datos, cancelando también el envío de notificaciones a los siguientes niveles. La siguiente imagen muestra el dialogo que la aplicación muestra al pulsar este botón.

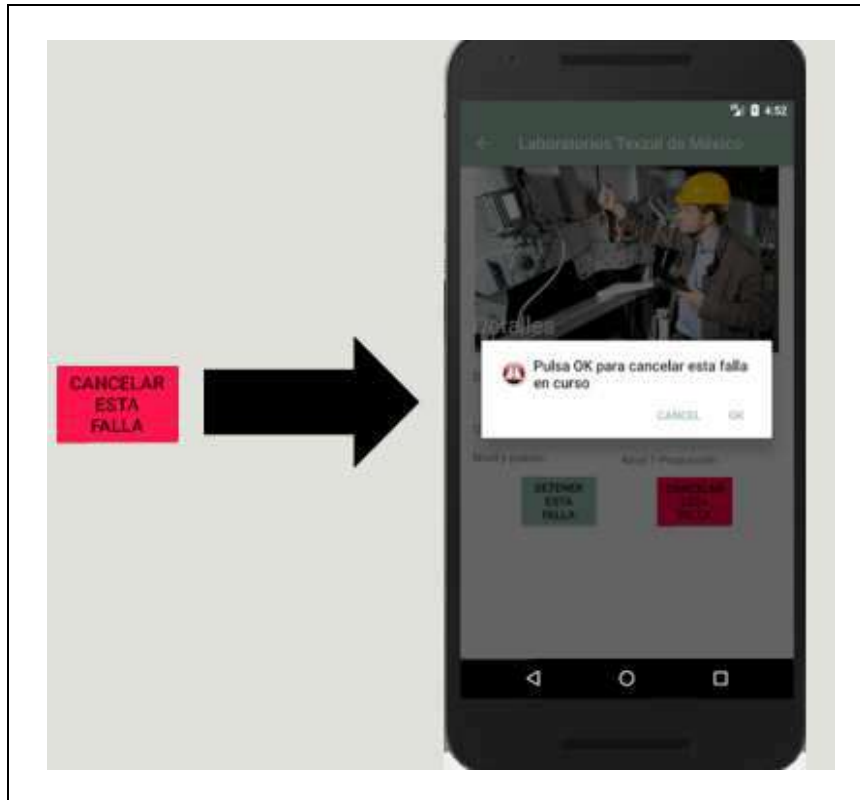


Figura 81 Botón "Cancelar esta falla"

Para la implementación de esta funcionalidad también se tuvo que agregar un segundo AsyncTask para manejar un nuevo tiempo de espera para mandar una nueva notificación a los usuarios del siguiente nivel. También se utilizaron los nuevos campos de la base de datos de la tabla "Falla en curso" en lo siguiente:

- Para saber ha sido cancelada y evitar mandar la siguiente notificación
- Para guardar el nombre de los usuarios que detienen esta falla
- Para almacenar la hora en que la falla fue agregada y finalmente hacer una operación y saber cuál fue la duración total de la falla para almacenar este dato en el registro de las fallas terminadas.

Para el desarrollo de esta actividad también se investigó como mandar notificaciones a un grupo de dispositivos y no a un dispositivo único desde nuestra aplicación. Afortunadamente la documentación en línea del IDE Android Studio y de la plataforma Firebase nos dio la solución tan solo leyendo varios de sus apartados.



Suscribe la app cliente a un tema

Las apps cliente se pueden suscribir a cualquier tema existente o pueden crear un tema nuevo. Cuando una app cliente se suscribe a un nombre de tema nuevo (uno que no existe aún para tu proyecto de Firebase), se crea un tema nuevo con ese nombre en FCM y cualquier cliente se puede suscribir a él posteriormente.

Para suscribirse a un tema, la app cliente llama a `subscribeToTopic()` de Firebase Cloud Messaging con el nombre del tema de FCM:

```
FirebaseMessaging.getInstance().subscribeToTopic("news");  
MainActivity.java
```

Para anular la suscripción, la app cliente invoca `unsubscribeFromTopic()` en Firebase Cloud Messaging con el nombre del tema.

Figura 82 Página oficial de Firebase – “Suscribir un app cliente a un tema”

Esta función que nos ofrece Firebase fue implementada en la actividad Login, pues cada que un usuario accede a la aplicación mediante su usuario y contraseña, en ese momento tenemos todos los datos del usuario, incluyendo el nivel al que pertenece, así suscribiendo el Smartphone a un nuevo tema o Topic.

También se tuvo que hacer una modificación al archivo php que nos permite realizar el envío de la notificación vía php y no por consola. El cambio es que ya no se va a mandar la notificación a un Token, sino al tema completo que se desea. En la siguientes imágenes se muestran esos cambios:

```
case 'enviarNoti':  
  
    $fallaT = $_POST["falla"];  
    $nivelN = $_POST["nivel"];  
  
    if($nivelN=="No hay datos")  
    {  
        echo "-1";  
    }else{  
        switch($nivelN){  
            case "Nivel 1":  
                $nivelN="/topics/nivel1";  
                break;  
            case "Nivel 2":  
                $nivelN="/topics/nivel1";  
                break;  
            case "Nivel 3":  
                $nivelN="/topics/nivel3";  
                break;  
        }  
        echo enviarNotificacion($nivelN,$fallaT);  
    }  
}
```

Figura 83 Cambios al archivo php - Topic 1



```
function enviarNotificacion($nivel, $mensaje)
{
    $url = 'https://fcm.googleapis.com/fcm/send';
    ignore_user_abort();
    ob_start();

    $fields = array('to' => $nivel ,
    'notification' => array('body' => $mensaje , 'title' => 'Notificación de falla'),
    'data' => array('Nick' => 'Mario', 'Room' => 'No fue mi culpa'));

    // $x++;

    define('GOOGLE_API_KEY', 'AIzaSyDoCNWfuyEOowfw01_JDOcb669pBH63ff8');

    $headers = array(
        'Authorization:key='.GOOGLE_API_KEY,
        'Content-Type: application/json'
    );

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($fields));

    $result = curl_exec($ch);
    if($result === false)
    die('Curl failed ' . curl_error());

    curl_close($ch);

    return $result;
}
```

Figura 84 Cambios al archivo php - Topic 2

Esta iteración fue de las más difíciles pues se tuvo que investigar mucho y mucho tiempo de prueba-error en busca del buen funcionamiento de la aplicación.

Prueba de aceptación “Fallas en curso”
Número: 1
Historia: Fallas en curso
Fecha de historia: 12/03/2018
Fecha de término: 24/03/2018
En esta iteración se desarrolla la actividad que muestra las fallas que se encuentran en curso. También se debe implementar el envío de la notificación el siguiente nivel cuando la falla rebasa el tiempo límite de ser una falla de segundo nivel. Se debe agregar una funcionalidad que permita a los usuarios cancelar una falla y también darla de terminarla según el nivel que haya alcanzado la falla en curso.
Resultado esperado: Los resultados al implementar esta funcionalidad fueron los esperados.

Tabla 23 Prueba de aceptación “Fallas en curso”



Las interfaces que se mostraron anteriormente son la iteración terminada.

Prueba aceptación de historia de usuario “Modificaciones en interfaz de usuario”:

- ✓ La aplicación abre la actividad de “Fallas en curso” en cuanto se agrega una nueva falla al servidor, comenzando el tiempo de espera para mandar la nueva notificación. Cumplido este tiempo, la aplicación llega a los dispositivos correspondientes correctamente.
- ✓ La aplicación permite cancelar una falla en curso evitando el envío de la siguiente notificación y eliminando todos los datos acerca del suceso.
- ✓ La aplicación identifica cuando se trata de una falla de Nivel 2 o Nivel 3 cuando un usuario la intenta detener, así dando la opción de terminarla por solo 2 usuarios o 3 usuarios respectivamente al nivel de la falla.
- ✓ Cuando una falla es terminada por el número de usuarios necesario, la aplicación guarda los datos de la falla en la base de datos permanentemente.
- ✓ La aplicación calcula el tiempo total que se tardó en terminar la falla.
- ✓ La aplicación almacena todos los datos necesarios acerca del suceso.
- ✓ La vista de la actividad “Fallas en curso” es vistosa y entendible aparte de funcional.

3.3.9 Historia de usuario 9

Historia N° 9	Tipo	Dificultad	Esfuerzo		Prioridad	Notas
Perfil de usuario	Nuevo Arreglar Mejorar	Fácil Moderado Difícil	Estimado	Gastado	2	
			4 días	5 días		
Descripción						
Se necesita una actividad donde se pueda visualizar los datos del usuario actual en la aplicación, así como las operaciones relacionadas con la sesión de usuario.						
Fecha	Estado	Comentario				
26/03/2018	Definición					
26 - 31	Implementación					
31/03/2018	Hecho	Se implementó la funcionalidad con éxito				

Tabla 24 Historia de usuario “Perfil de usuario”

Ya se tiene la mayoría de la funcionalidad principal de la aplicación para solucionar la problemática de los Laboratorios Texzal de México, solo queda implementar una actividad que permita visualizar los datos del usuario que está actualmente activo en la aplicación, así como las opciones de cerrar sesión, modificar los datos del usuario y eliminar la cuenta. La siguiente imagen muestra la actividad hecha para el perfil de usuario:

Proyecto de Estadía Laboratorios Texzal de México

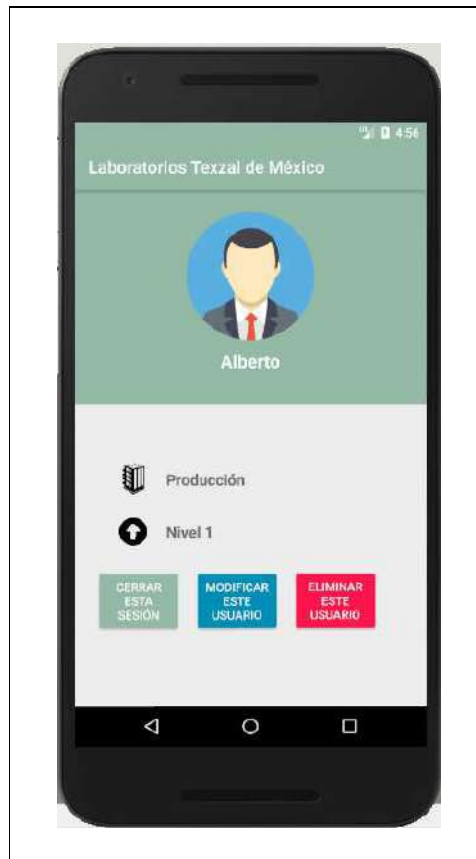


Figura 85 Actividad "Perfil de usuario"

A continuación se mostrarán en imágenes las funciones de cada uno de los botones. Primeramente el botón "Cerrar sesión" que como su nombre lo dice, cierra la sesión del usuario actual, también desuscribe al dispositivo del tema o Topic al que estaba registrado.

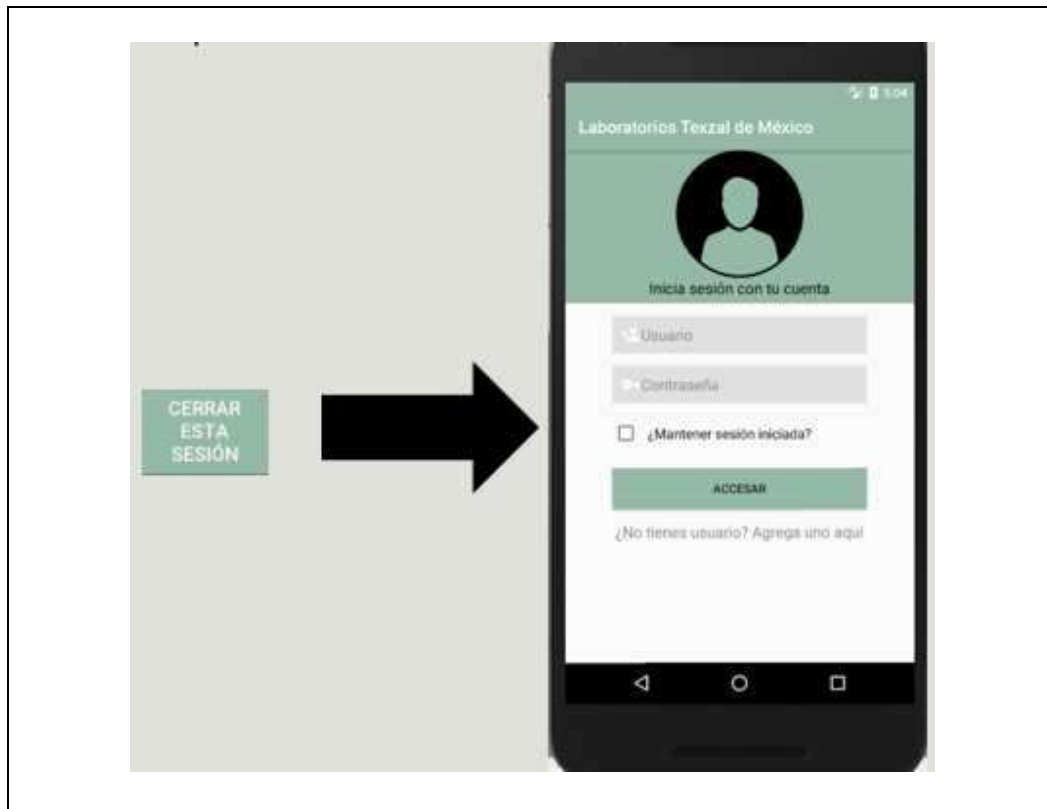


Figura 86 Botón “Cerrar esta sesión”

El botón de “Modificar este usuario” abre un dialogo donde se muestran los campos con los datos actuales del usuario donde podemos modificar todos esos datos y pulsando en “Actualizar cambios” para completar la acción en la base de datos. Cuando esto sucede, se actualiza automáticamente la actividad “Perfil de usuario” con los nuevos datos.

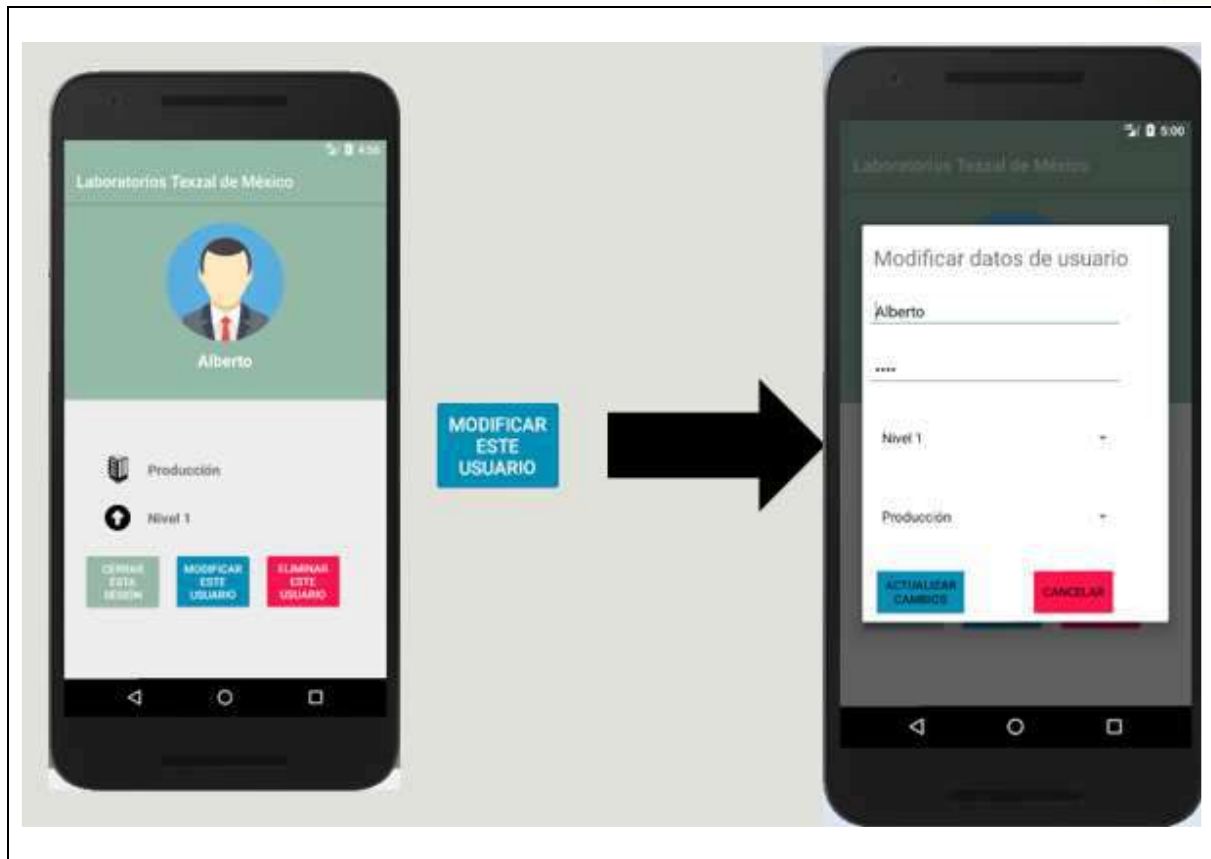


Figura 87 Botón “Modificar este usuario”

Por último el botón “Eliminar este usuario” el cual manda un mensaje de confirmación para completar esta acción pues no hay marcha atrás. Cuando se pulsa el botón “Ok” la cuenta es eliminada, el smartphone se desuscribe del Topic y devuelve la aplicación a la actividad “Login”.

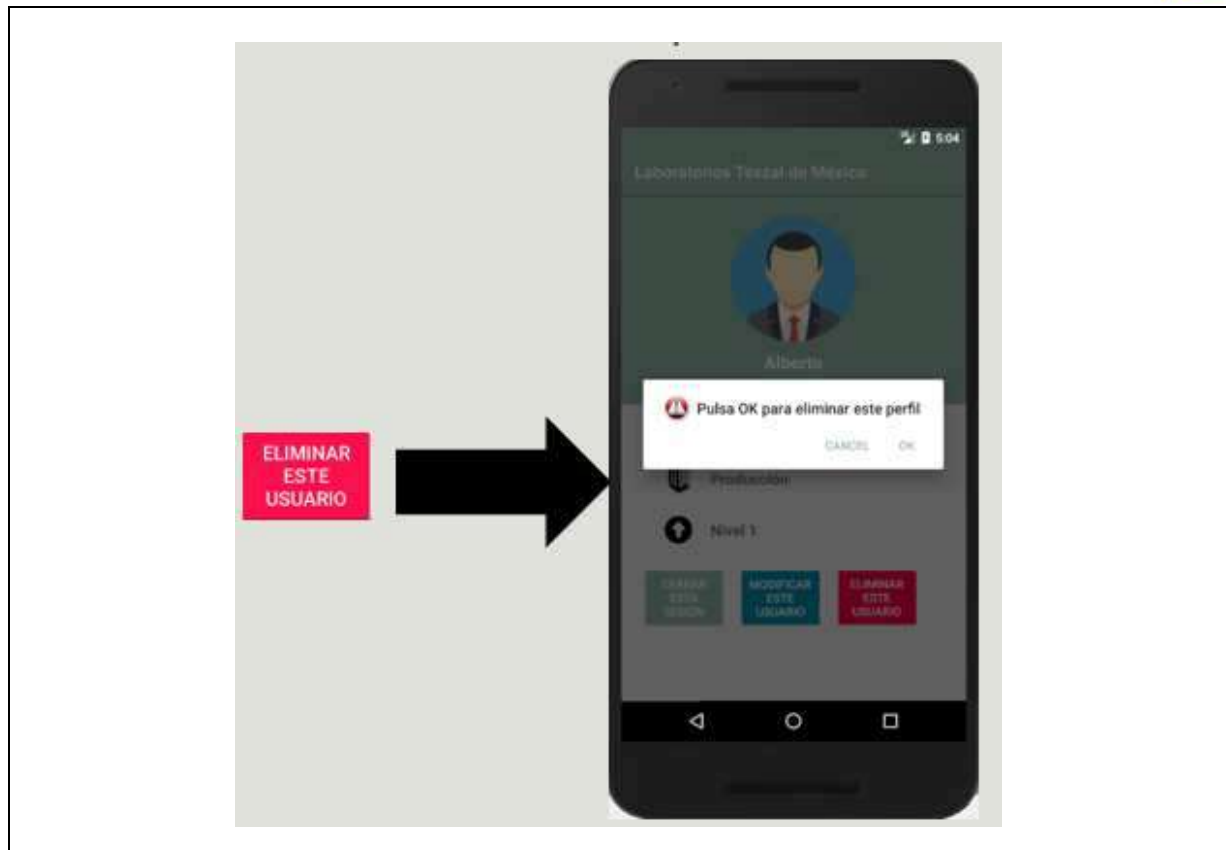


Figura 88 Botón "Eliminar este usuario"

Prueba de aceptación "Perfil de usuario"
Número: 1
Historia: Perfil de usuario
Fecha de historia: 26/03/2018
Fecha de término: 31/03/2018
Descripción: La aplicación requiere una actividad que mostrará los datos del usuario así como las opciones disponibles para su perfil.
Resultado esperado: Los resultados al implementar esta funcionalidad fueron los esperados.

Tabla 25 Prueba de aceptación "Perfil de usuario"

Las interfaces que se mostraron anteriormente son la iteración terminada.
Prueba aceptación de historia de usuario "Mandar notificación local":

- ✓ La aplicación muestra todos los datos del usuario.
- ✓ La aplicación cierra sesión correctamente.
- ✓ La aplicación permite modificar los datos del usuario actual.
- ✓ La aplicación permite eliminar satisfactoriamente una cuenta de usuario.



Proyecto de Estadía Laboratorios Texzal de México

3.4 Estabilización

Después de la fase de producto llega la fase de estabilización en la que se realizan las acciones de integración para enganchar los posibles módulos separados en una única aplicación. Como este proyecto fue realizado por una sola persona esta parte no fue realizada por completo, no se realizó una integración de módulos hechos por distintas personas en una sola aplicación, pero si se realizaron pruebas para verificar que las actividades independientes de la aplicación interactúen correctamente entre ellas realizando las tareas que deberían y como deberían, así como el intercambio de información entre ellas. Se mostrarán en la fase de pruebas.

En esta fase de la metodología se realiza la documentación de usuario, que se refiere a las instrucciones para usar la aplicación.

3.4.1 Documentación de usuario

A continuación se mostrarán las instrucciones con capturas de pantalla acerca de cómo usar la aplicación, desde la instalación hasta el recorrido por sus funcionalidades.

En primer lugar verifique que su dispositivo tiene la hora correcta.

Después, tenemos que instalar aplicación en nuestro dispositivo con la versión del sistema operativo Android 5.0 Lollipop. Basta con buscar el archivo .apk en tus archivos donde esté guardado. Hay que abrir el archivo con un toque sobre el ícono.

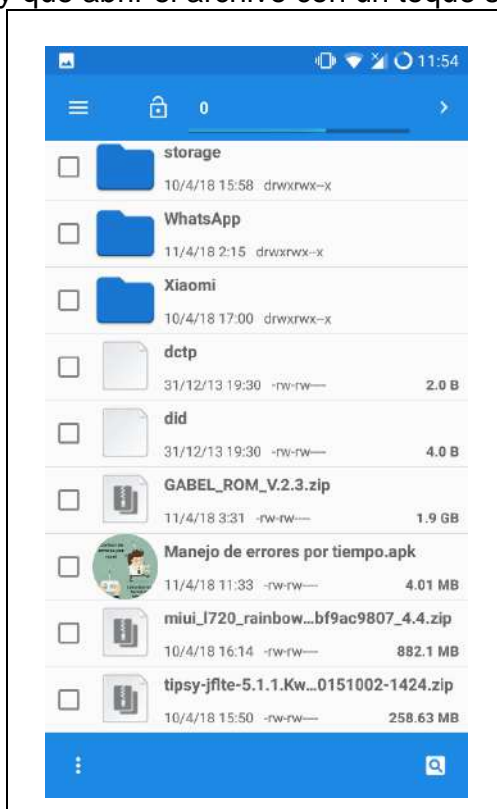


Figura 89 Explorador de archivos

Aparecerá la siguiente interfaz donde solo debemos pulsar “Instalar”.



Proyecto de Estadía Laboratorios Texzal de México

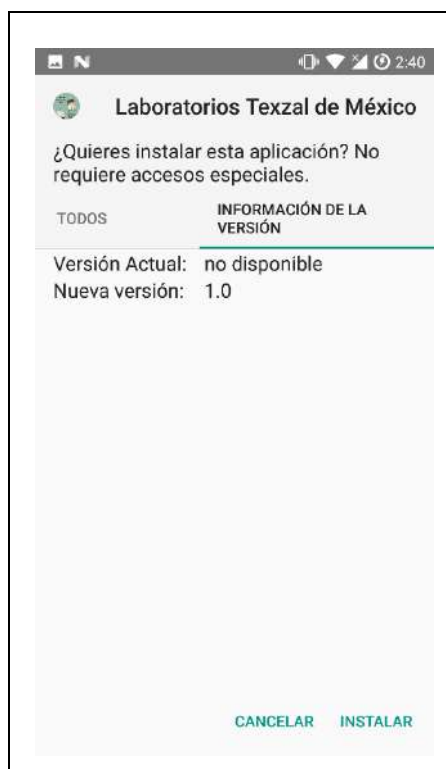


Figura 90 Pantalla de instalación

A continuación una pantalla de carga y la pantalla de confirmación de una instalación exitosa, se debe pulsar la opción “Abrir” para ir inmediatamente a la aplicación. Si recibe el siguiente error es porque no tiene una versión de Android 5.0 o superior:



Figura 91 Pantalla de instalación

Proyecto de Estadía
Laboratorios Texzal de México

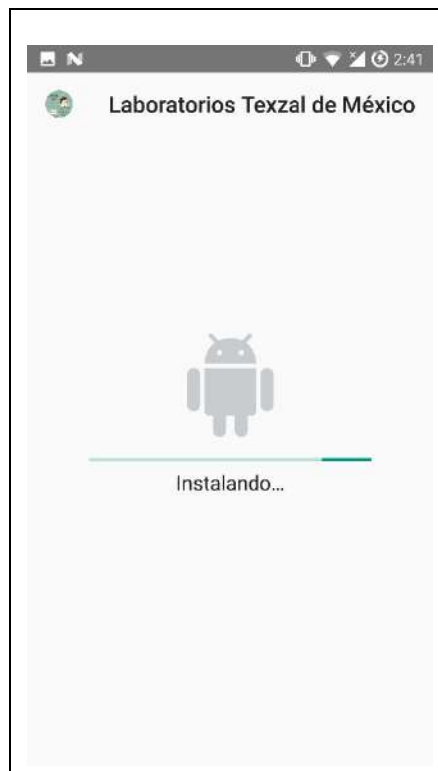


Figura 92 Pantalla de carga de instalación



Figura 93 Pantalla de instalación exitosa



Proyecto de Estadía Laboratorios Texzal de México

La aplicación tiene una pantalla para que tus usuarios escriban su Nombre de usuario y contraseña únicos para poder acceder a las todas funcionalidades que ofrece la aplicación.



Figura 94 Actividad Login

La aplicación manda al usuario mensajes de retroalimentación cada que el usuario interactúa con una función de la aplicación, como el siguiente:

Proyecto de Estadía
Laboratorios Texzal de México

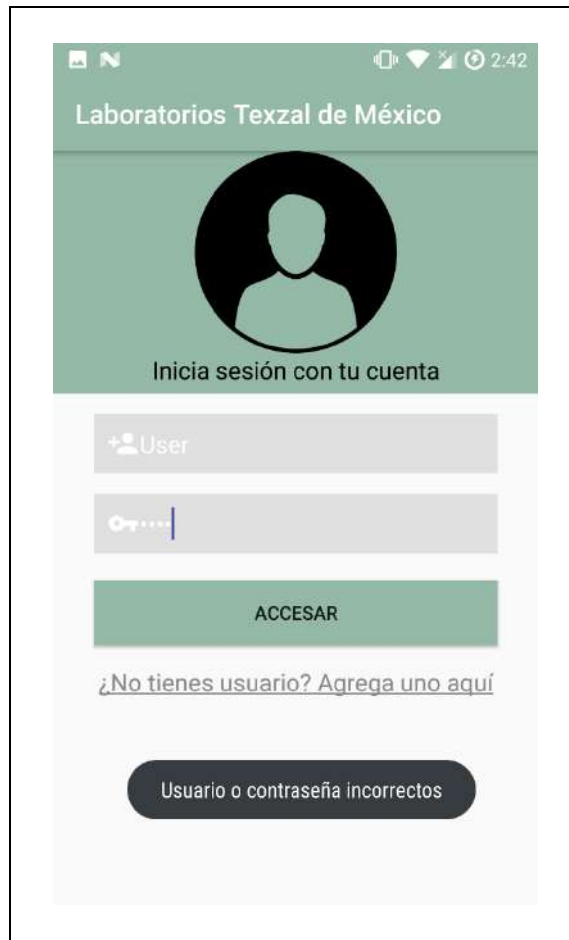


Figura 95Login – Intento erróneo

¿No tienes un usuario? Podrías empezar por agregar uno nuevo. A continuación se muestra la actividad que permite al personal agregar un “Usuario nuevo” a la aplicación.



Proyecto de Estadía Laboratorios Texzal de México

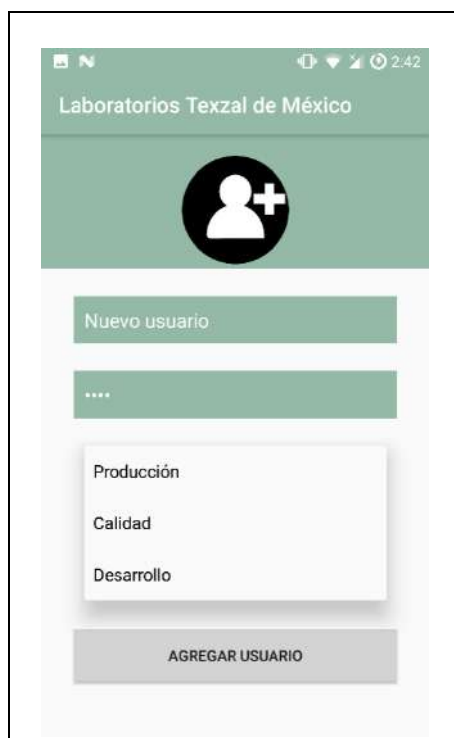


Figura 96 Actividad "Agregar usuario"

Una vez que tienes un usuario, puedes iniciar sesión en la aplicación.

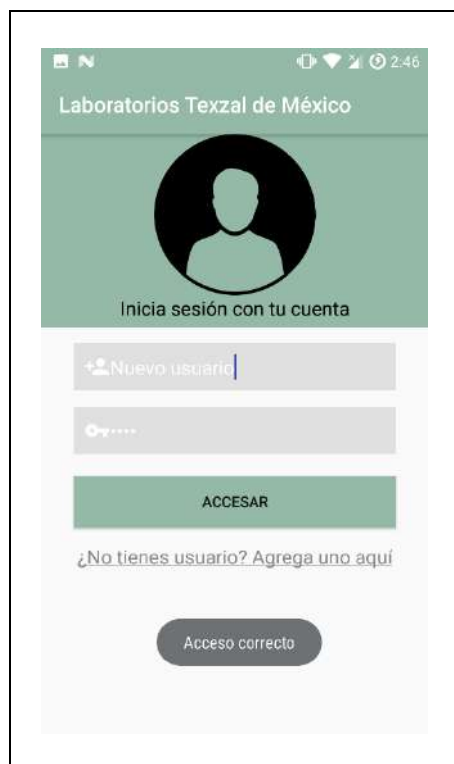


Figura 97 Login - Acceso correcto



Proyecto de Estadía Laboratorios Texzal de México

Una vez dentro, lo primer que se puede observar son las opciones disponibles en la aplicación, las cuales son

- Envío de fallas
- Fallas en curso
- Lista de fallas
- Perfil de usuario

Se empieza por lo primero, enviar una falla nueva. Basta con pulsar en el ícono de “Fallas”

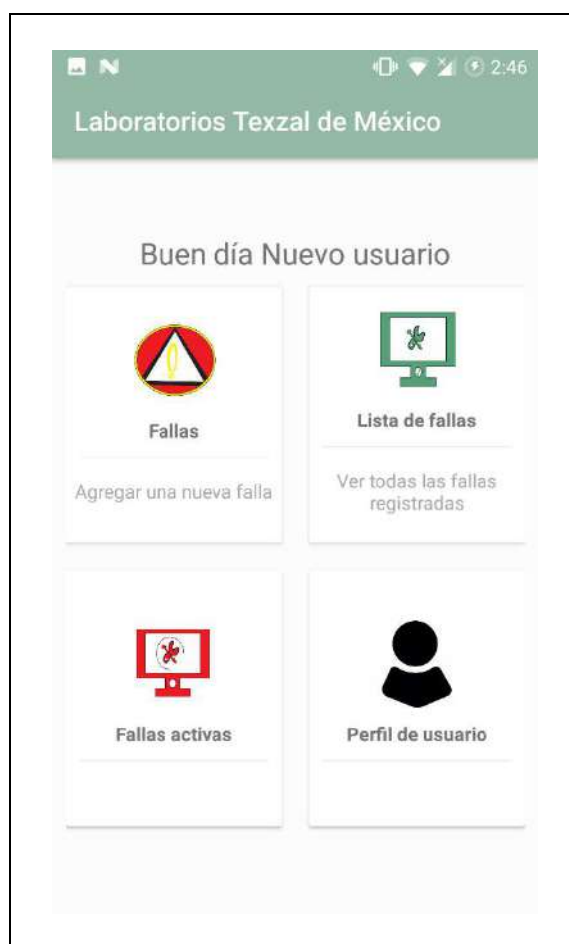


Figura 98 Actividad Menú principal

Se inicia por mostrar el envío de fallos. Basta con pulsar en el botón rojo y grande para que la aplicación nos muestre un dialogo donde el usuario va a describir la falla. Para después solo pulsar en “Enviar falla”



Figura 99 Actividad Enviar falla

Nos enviará a la actividad “Lista de fallas en curso” donde aparecerán todas las fallas que se den de alta por cada usuario. Si pulsamos en el botón “Ver detalles” podremos ver un poco más acerca de la falla. También tendremos disponibles los botones “Detener esta falla” y “Cancelar esta falla”



Figura 100 Actividad Falla en curso y sus detalles



Proyecto de Estadía
Laboratorios Texzal de México

Cuando una falla es enviada a “Fallas en curso” y, según el cliente, se requiere que la aplicación mande una notificación al nivel 1 y 2, un tiempo después que se envió. En este caso la falla puede ser terminada solamente por un usuario de nivel 1 y un usuario de nivel 2.

Para que un usuario pueda detener una falla debe confirmar su contraseña.

Si la falla no es resuelta antes que se manda notificación al Nivel 3, entonces la falla debe ser detenida con un usuario de cada nivel.

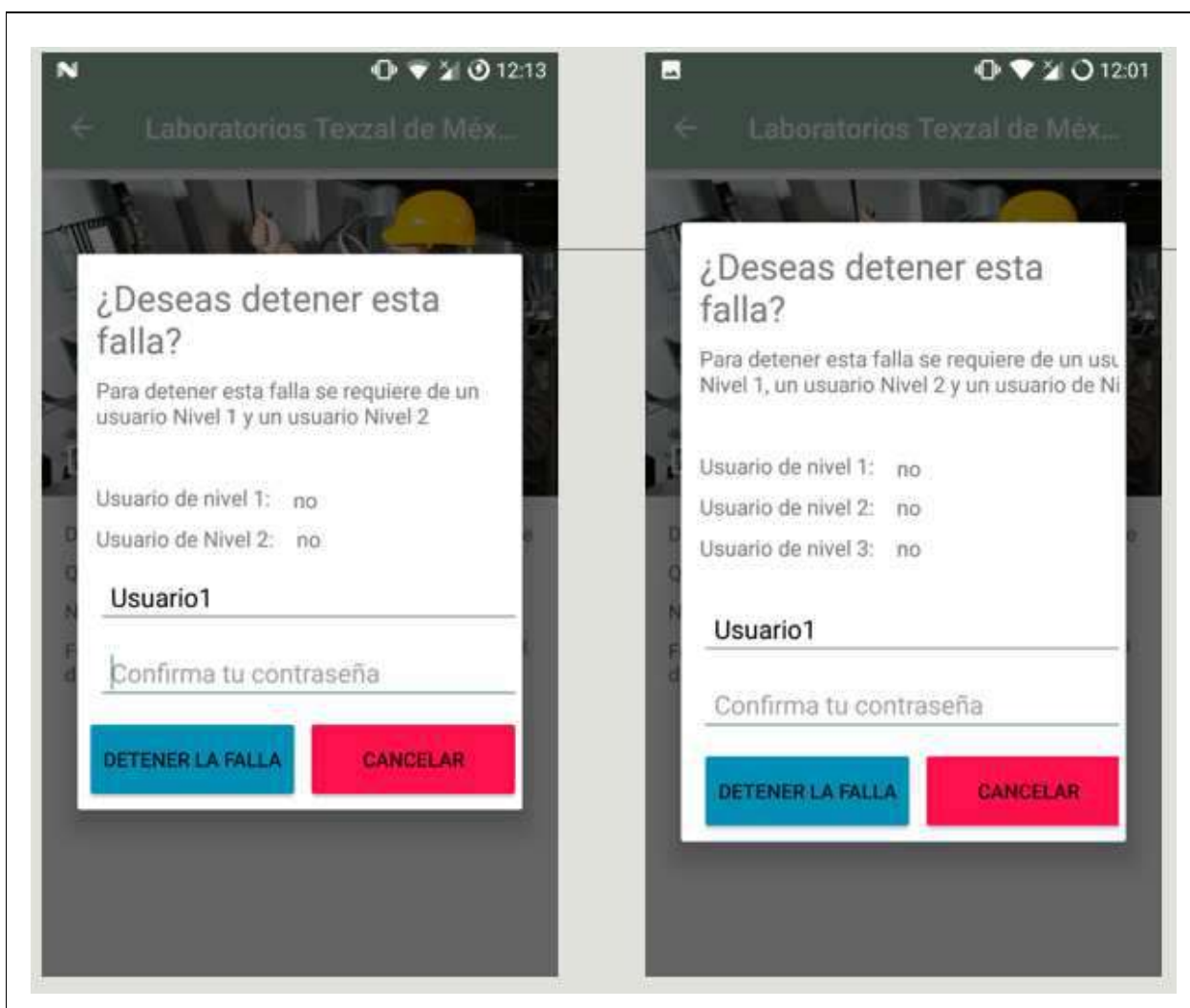


Figura 101 Botón “Detener esta falla”

Proyecto de Estadía Laboratorios Texzal de México

Cuando un usuario de cada nivel detiene la falla, esta es guardada permanentemente, lo que permite visualizar los datos de todas las fallas terminadas.

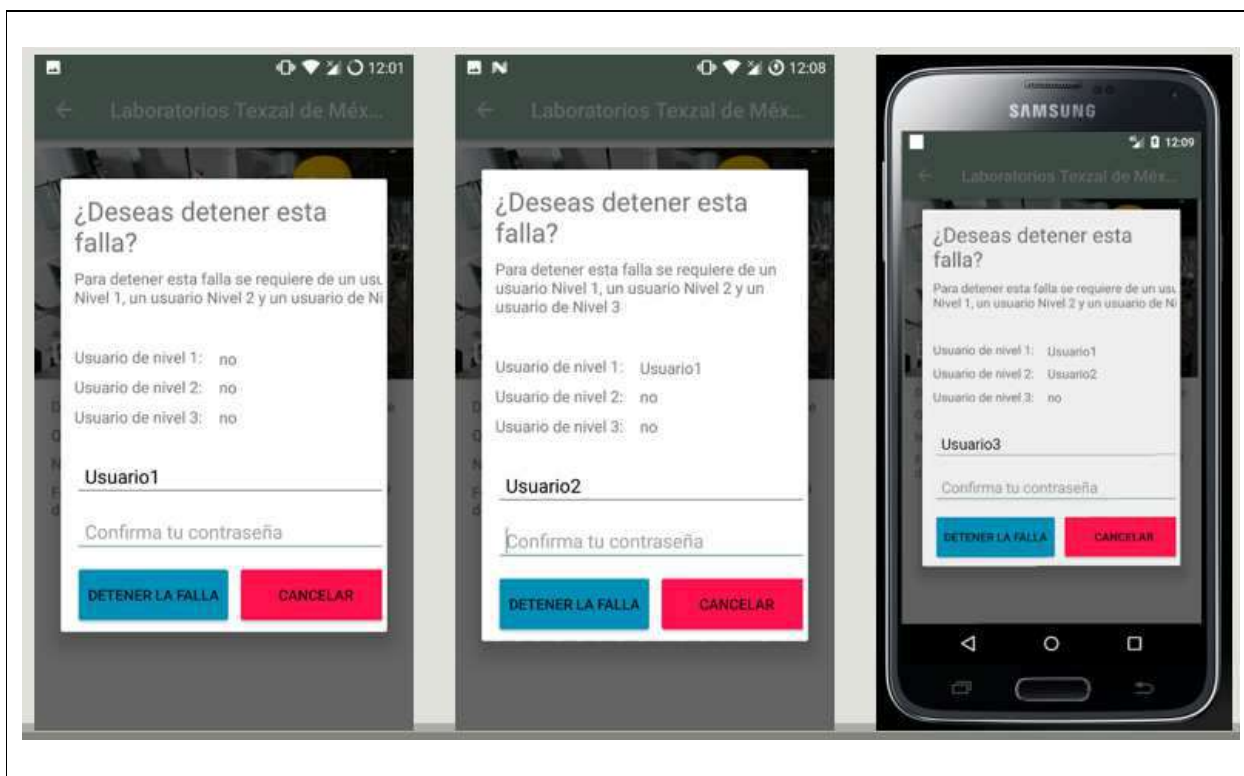


Figura 102 Botón "Detener la falla"

Proyecto de Estadía
Laboratorios Texzal de México

Cuando seas el último de terminar la falla, te aparecerá este mensaje que indica que la falla fue completada con éxito, solo basta volver al menú principal donde seleccionaremos la actividad “Lista de fallas” para visualizar las fallas terminadas.



Figura 103 Finalizar una falla



Proyecto de Estadía Laboratorios Texzal de México

Por otro lado, si usamos el botón “Cancelar esta falla”, nos mostrará un dialogo que nos permite continuar o cancelar la acción. Al confirmar, inmediatamente se cancela el envío de notificaciones a los demás niveles.

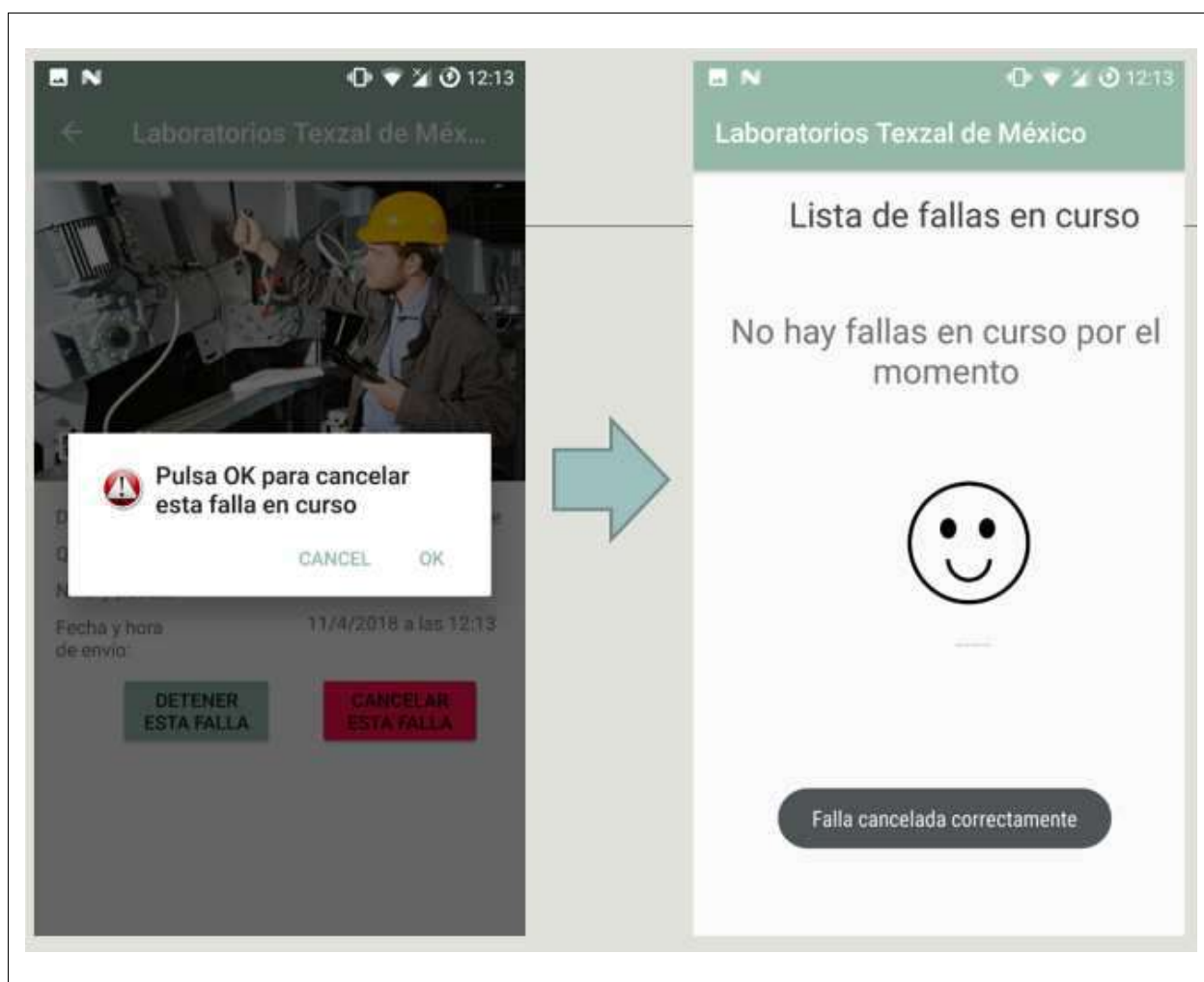


Figura 104 Cancelar falla

Por otro último, se muestra la actividad “Perfil de usuario” la cual permite visualizar los datos del usuario en actual en la aplicación, así como las acciones posibles respecto a la cuenta.

Al pulsar el botón “Cerrar esta sesión”, hace que se borren los datos del usuario en la aplicación y se devuelva a la actividad “Login”. Esto permite usar diferentes cuentas en un dispositivo a la vez.

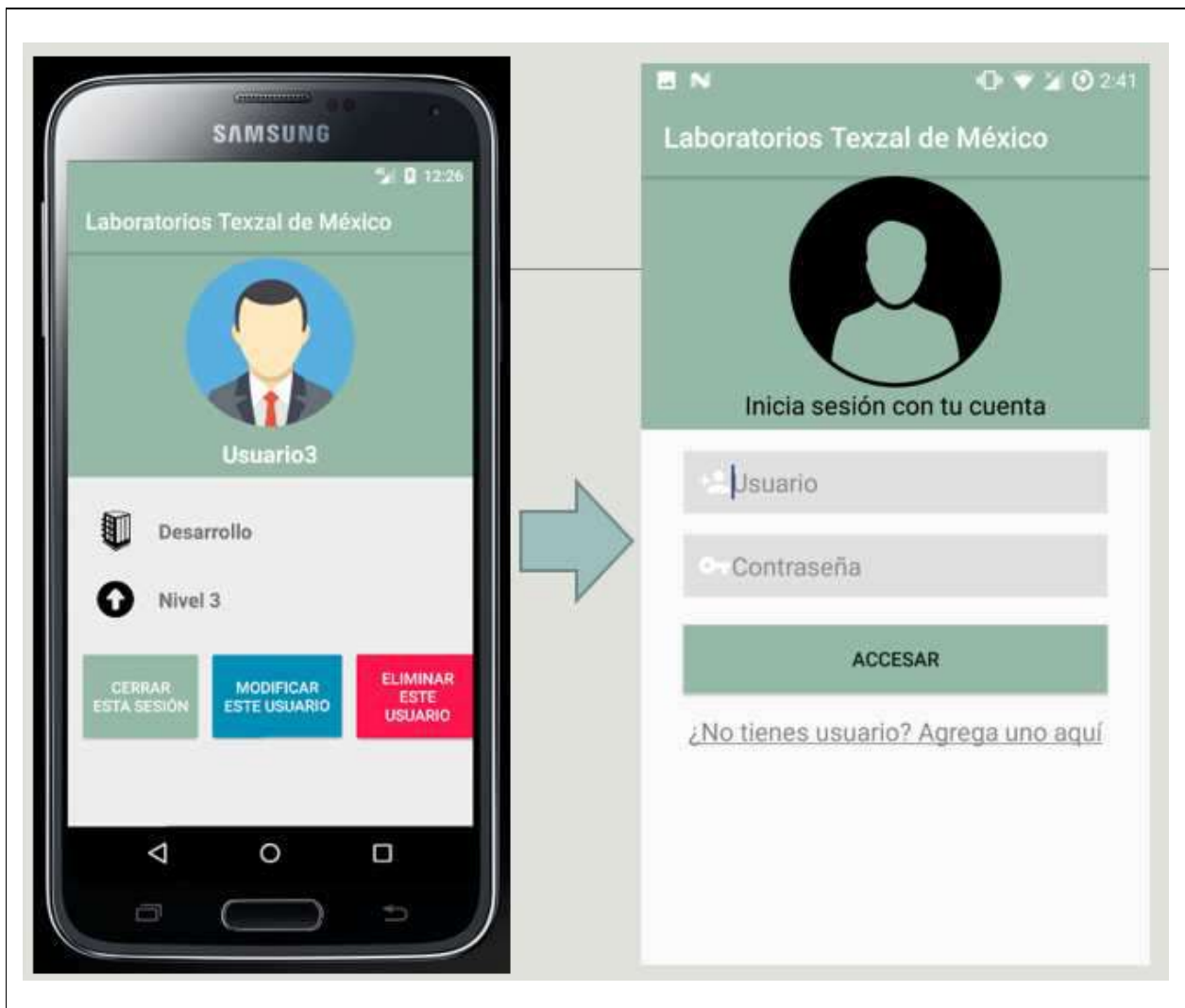


Figura 105 Actividad – Perfil de usuario, Botón “Cerrar esta sesión”

Proyecto de Estadía
Laboratorios Texzal de México



Al pulsar el botón “Modificar este usuario”, la aplicación nos muestra un dialogo con los datos actuales del usuario el cual nos permite actualizar cualquier dato del mismo.

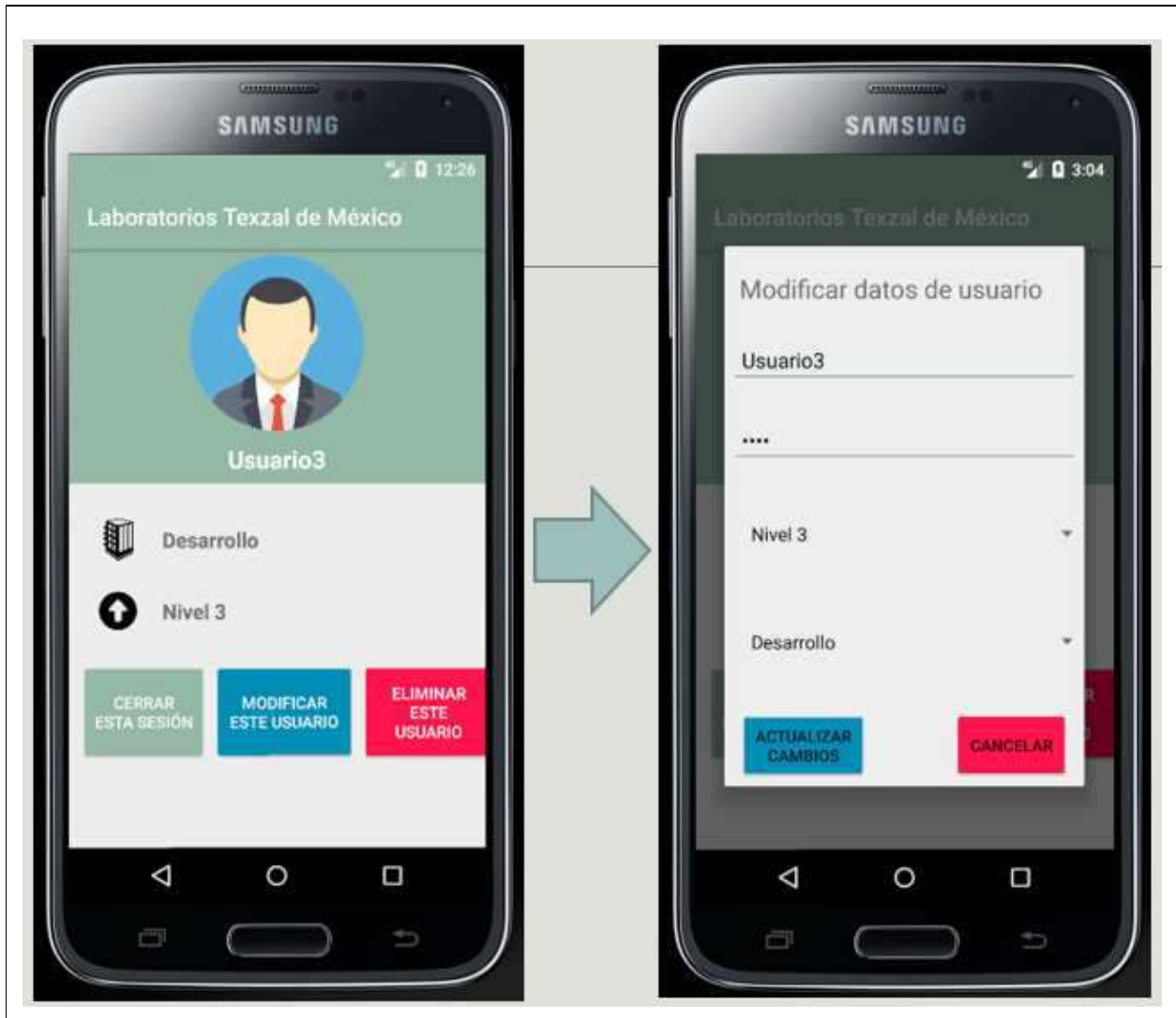


Figura 106 Botón “Modificar este usuario”

Al pulsar el botón “Eliminar este usuario”, la aplicación muestra un dialogo para confirmar la acción, al pulsar en “Ok” la cuenta del usuario es eliminada.

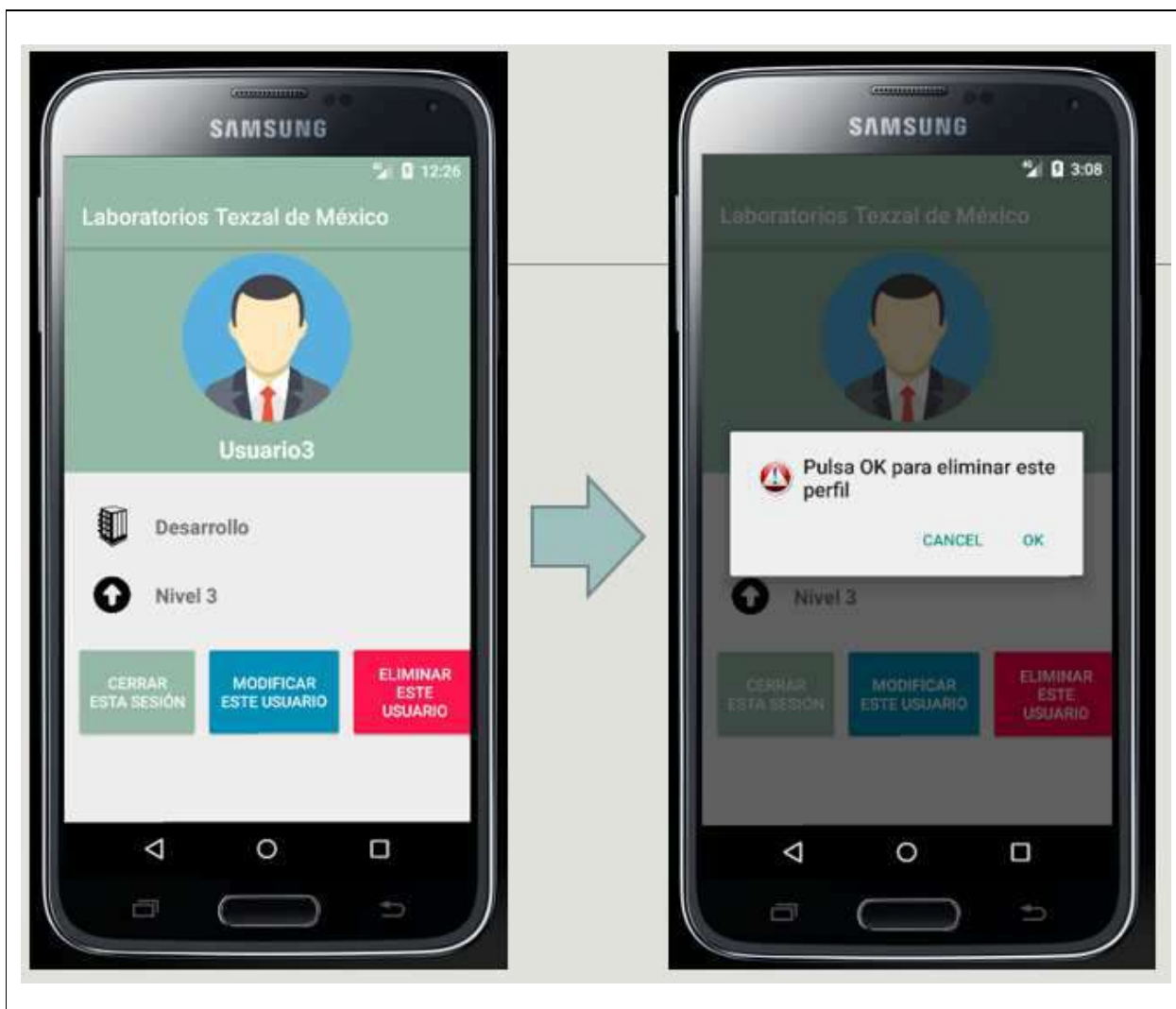


Figura 107 Botón “Eliminar este usuario”

3.5 Pruebas

Fase de **pruebas**. Una vez parado totalmente el desarrollo se pasa una fase de testeo hasta llegar a una versión estable según lo establecido en las primeras fases por el cliente. Si es necesario se reparan los errores, pero no se desarrolla nada nuevo.

Lo primero es definir el alcance de las pruebas así como los entornos y las herramientas de pruebas que se utilizarán para desarrollar y aplicar dichas pruebas a nuestra aplicación [11].



Proyecto de Estadía Laboratorios Texzal de México

3.5.1 Alcance de las pruebas

En esta parte se listarán todos los módulos, componentes o elementos que se van a probar. En nuestro caso se probarán todos los módulos para verificar su correcto funcionamiento. La lista de esos módulos es la siguiente:

- Login
- Agregar Usuario
- Menú principal
- Enviar falla
- Mostrar todas las fallas
- Fallas en curso
- Detalle de fallas en curso
- Perfil de usuario

3.5.2 Enfoque de pruebas (Estrategia)

La estrategia de pruebas puede definirse como un documento por separado o puede ser incluido dentro del plan de pruebas, el cual es el caso de este proyecto. Las pruebas que se realizarán serán:

- Testing de dispositivos
- Testing de usabilidad
- Testing en la red de comunicaciones

3.5.3 Configurar laboratorio de pruebas

La configuración del laboratorio se desarrollará siguiendo la combinación de servidores, computadores personales, emuladores y dispositivos reales. Los emuladores proporcionan velocidad y diversidad en la ejecución de las pruebas, al permitirnos verificar diversas condiciones, tipos de dispositivos y sistemas operativos desde un computador personal.

Sin embargo, tienen la desventaja de carecer de ciertas peculiaridades que podrían producir fallas en un ambiente real, además, no son sensibles a los factores ambientales que pueden afectar el comportamiento de los dispositivos. Por lo tanto la mejor manera de lograr el equilibrio es combinar el uso de dispositivos reales con emuladores. A continuación se definirán las versiones de software y hardware utilizadas para las pruebas:

Proyecto de Estadía
Laboratorios Texzal de México

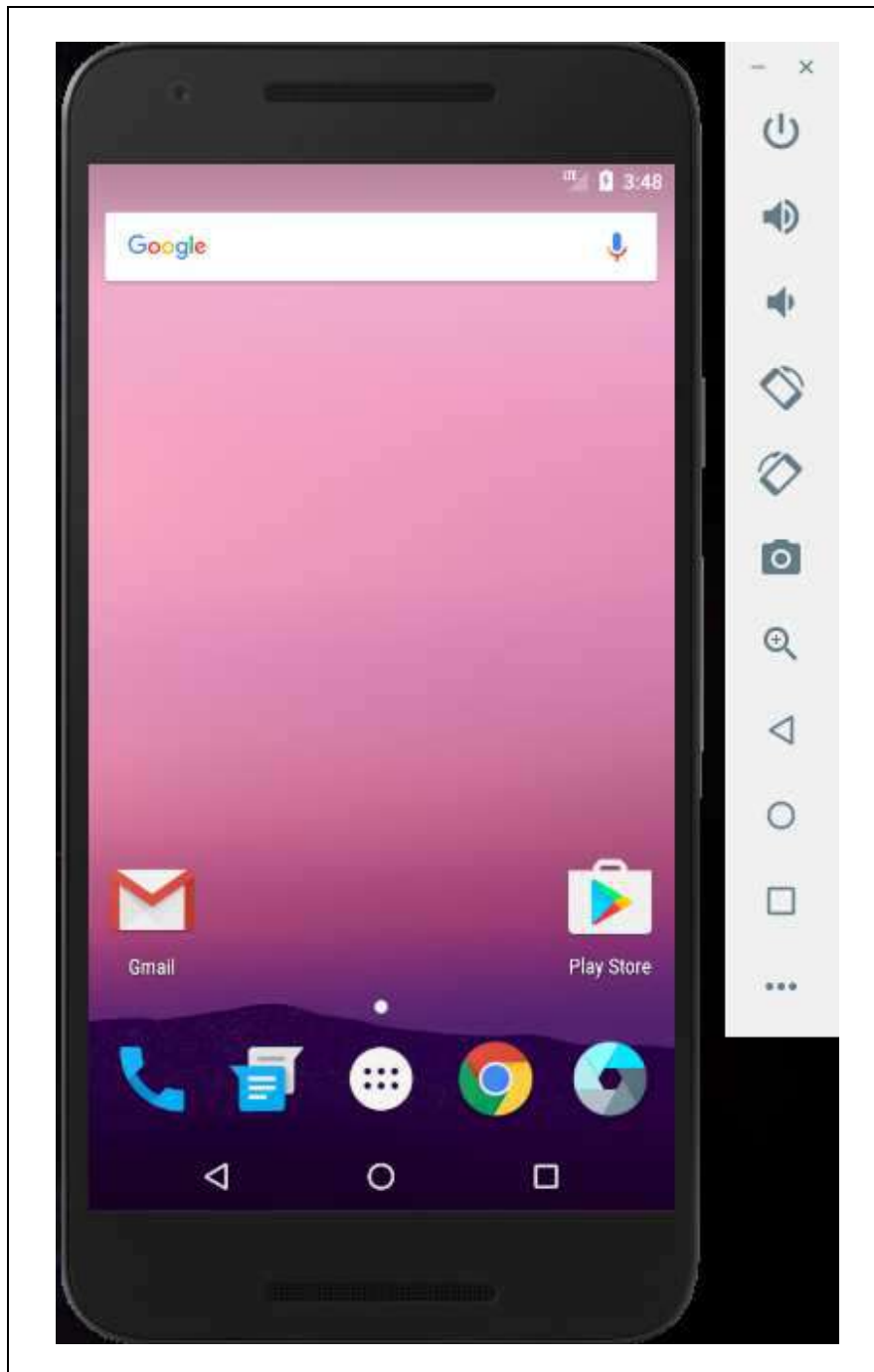


Figura 108Nesus 5X

Especificaciones de dispositivo Nexus5X:

- 1080 x 1920 pixels, 5.2 pulgadas
- Sistema Operativo: Android6.0 Marshmallow
- Procesador del sistema: Procesador hexa-coreSnapdragon 808
- Memoria (RAM): 2 GB.

- Almacenamiento: 16 GB}

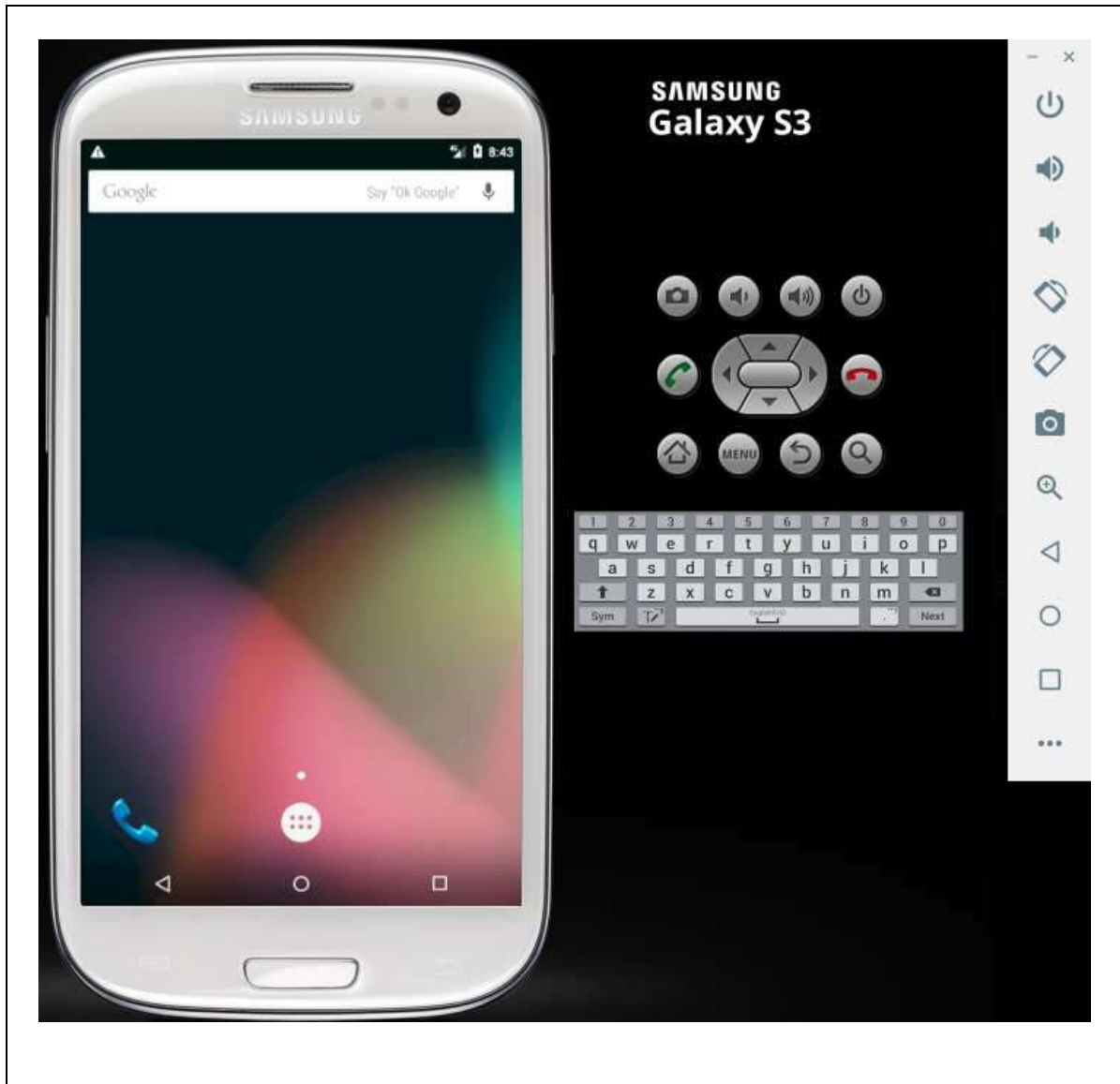


Figura 109 Samsung Galaxy S3

Especificaciones de dispositivo Galaxy S3:

- 720 x 1280 4.8 pulgadas
- Android OS, v4.0.4 Ice CreamSandwich
- Procesador del sistema: Procesador Exynos 4 Quadquad-core 1.4 GHz, GPU Mali 400MP
- Memoria (RAM): 1 GB.
- Almacenamiento: 16 GB

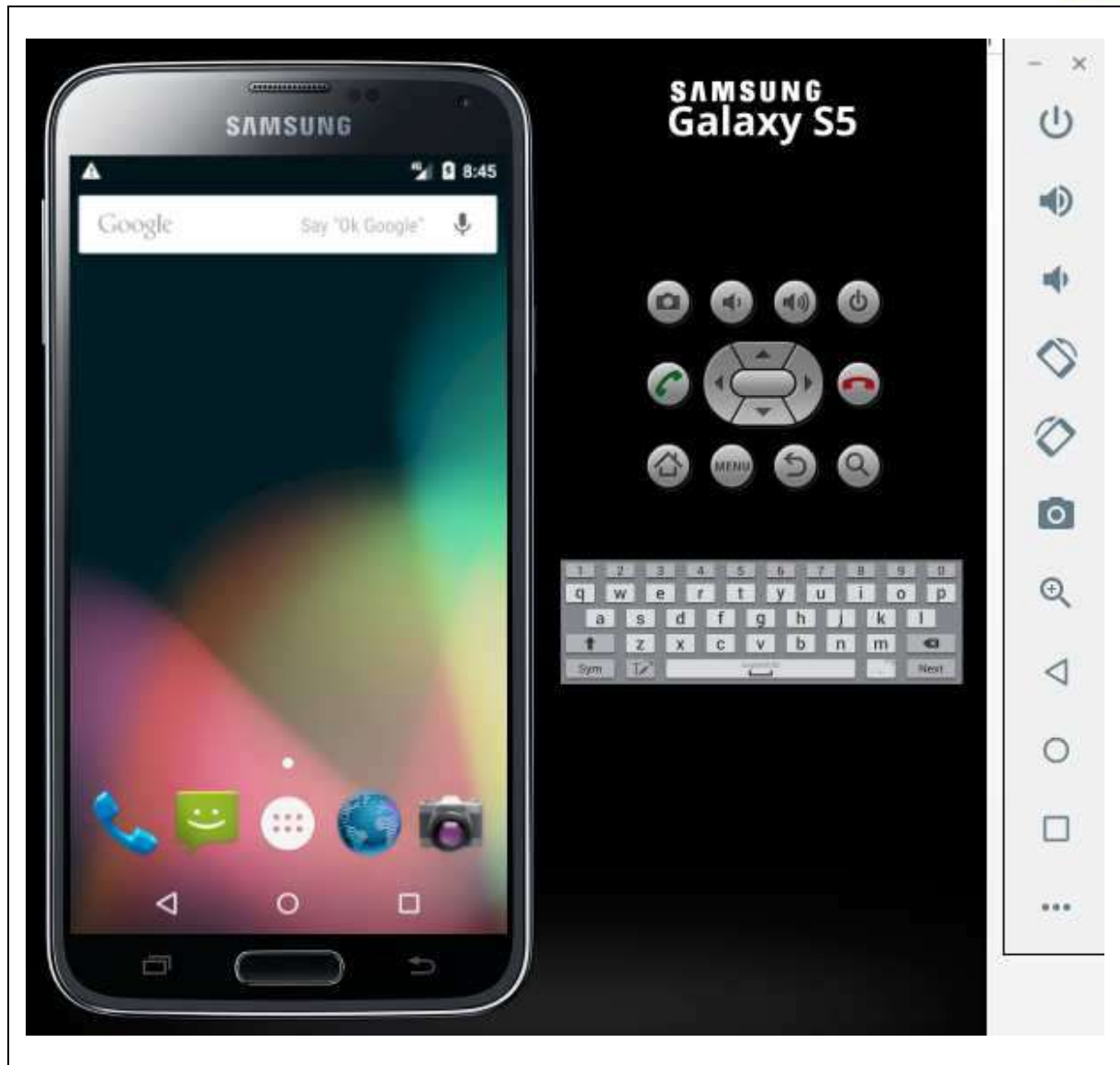


Figura 110 Samsung Galaxy S5

Especificaciones de dispositivo Galaxy S5:

- Tamaño de pantalla: 1080 x 1920 pixeles 5.1 pulgadas
- Sistema Operativo: Android 5,1 Lollipop®
- Procesador del sistema: Procesador Qualcomm MSM8974AC Snapdragon 801 quad-core 2.5 GHz, GPU Adreno 330
- Memoria (RAM): 3 GB.
- Almacenamiento: 32 GB

A continuación se muestran los dispositivos móviles reales que se utilizarán para las pruebas.



Figura 111 Samsung Galaxy S4

Especificaciones de dispositivo Galaxy S4:

- 1080 x 1920 pixels 5 pulgadas
- Sistema Operativo: Android 4.2.2 Jelly Bean Original, actualmente con 7.1.2 Nougat
- Procesador del sistema: Procesador Exynos 5 Octa octa-core 1.6 GHz
- Memoria (RAM): 2 GB.
- Almacenamiento: 16 GB



Figura 112 Motorola G

Especificaciones de dispositivo Motorola G:

- 720 x 1280 pixels 4.5 pulgadas
- Sistema Operativo: Android4.3 JellyBean Original, actualmente con 5.1 Lollipop
- Procesador del sistema: Procesador QualcommSnapdragon 400 quad-core 1.2 GHz, GPU Adreno 305
- Memoria (RAM): 1 GB.
- Almacenamiento: 16 GB



Figura 113 LG G3 Stylus

Especificaciones de dispositivo LG G3 Stylus:

- 540 x 960 pixels 5.5 pulgadas
- Sistema Operativo: Android4.3 JellyBean Original, actualmente con 5.0.2Lolipop
- Procesador del sistema: Procesador quad-core 1.3 GHzMemoria (RAM): 1 GB.
- Almacenamiento: 8 GB



Proyecto de Estadía
Laboratorios Texzal de México

A continuación se mostrará una tabla de los sistemas operativos seleccionados para las pruebas:

Tabla de Sistemas operativos
Lolipop 5.1
Nougat 7.1.2

Tabla 26 Lista de Sistemas operativos a probar

3.5.4 Casos de prueba aplicados

En esta sección se muestran los casos de uso que se aplicaron para probar las actividades que se describieron en la estrategia de pruebas.

Proyecto de Estadía
Laboratorios Texzal de México



Nombre del módulo o funcionalidad	Login			
Objetivo de la prueba	Verificar que la actividad permite el acceso únicamente con un usuario y contraseña válidos escritos en los campos. Verificar sus entradas y salidas.			
Caso de Prueba No 1.	Probar la validación los campos Usuario y Contraseña al intentar iniciar sesión			
Descripción	Un usuario debe acceder a las funciones de la aplicación con un usuario y contraseña registrados previamente en la aplicación.			
Entrada	El usuario deja vacío el campo Usuario. El usuario deja vacío el campo Contraseña. EL usuario pulsa el botón "Accesar"	El usuario escribe un usuario válido en el campo Usuario El usuario escribe una contraseña correcta conforme el usuario. EL usuario pulsa el botón "Accesar"	El usuario escribe un usuario no registrado en la aplicación. El usuario escribe una contraseña que no coincide con la del usuario.	EL usuario pulsa el texto "¿No tienes usuario? Agrega uno aquí"
Salida	La aplicación muestra un mensaje con el mensaje "Debes escribir un usuario y contraseña"	La aplicación muestra un mensaje con el mensaje "Acceso correcto" y lleva al usuario a la actividad "Menú principal"	La aplicación muestra un mensaje con el mensaje "Usuario o contraseña incorrectos"	La aplicación abre correctamente la actividad "Agregar usuario nuevo"
Resultados Finales	Los resultados son los esperados en esta prueba conforme al llenado de campos de la actividad Login			
Observaciones				
Resultados obtenidos	SATISFACTORIO Sin errores	ACEPTABLE De 1 a 2 errores	INSATISFACTORIO De 3 o más errores	
Caso 1	La aplicación reacciona ante las situaciones según lo esperado.			
Mejoras	Descripción			
	1.			
	2.			
Resultados Finales	Este módulo realiza su tarea satisfactoriamente.			

Tabla 27 Caso de prueba "Login"

Proyecto de Estadía
Laboratorios Texzal de México



Nombre del módulo o funcionalidad	Agregar usuario nuevo		
Objetivo de la prueba	Verificar que la actividad registre correctamente a un usuario y rechace la solicitud si ya hay un usuario registrado con el mismo nombre. Verificar sus entradas y salidas.		
Caso de Prueba No 1.	Datos correctos		
Descripción	Un usuario puede agregar un nuevo usuario y contraseña para acceder a las funciones de la aplicación.		
Entrada	El usuario deja vacíos los campos usuario y contraseña. EL usuario pulsa el botón "Registrar usuario"	El usuario escribe un nombre de usuario único y diferente, y una contraseña nueva. El usuario pulsa el botón "Registrar usuario"	El usuario escribe un nombre de usuario que ya está registrado en la aplicación, y una contraseña nueva. El usuario pulsa el botón "Registrar usuario"
Salida	La aplicación muestra un mensaje con el mensaje "Debes seleccionar un usuario y contraseña"	La aplicación muestra un mensaje con el mensaje "Debes escribir un usuario y contraseña"	La aplicación muestra un mensaje con el mensaje "Ese usuario ya ha sido registrado"
Resultados Finales	Los resultados fueron los esperados.		
Observaciones	Para comprobar la satisfacción de las mejoras, se solicitará que otro programador (pruebas) realice la prueba completa.		
Resultados obtenidos	SATISFACTORIO Sin errores	ACEPTABLE De 1 a 2 errores	INSATISFACTORIO De 3 o más errores
Caso 1	La aplicación reacciona según el comportamiento del usuario, evitando registros de usuarios repetidos y guardando los nuevos.		
Mejoras	Descripción		
	1.		
Resultados Finales	Este módulo realiza su tarea satisfactoriamente.		

Proyecto de Estadía
Laboratorios Texzal de México



Nombre del módulo o funcionalidad	Menú principal			
Objetivo de la prueba	Verificar que se abre cada uno de las actividades. Verificar que al entrar y salir de la aplicación se llegue directamente al Menú principal.			
Caso de Prueba No 1.	Entrar y salir de cada una de las actividades			
Descripción	Después de acceder con usuario y contraseña, la aplicación debe abrir todas las actividades disponibles.			
Entrada	El usuario teclea el botón "Fallas"	El usuario teclea el botón "Lista de fallas"	El usuario teclea el botón "Fallas activas"	El usuario teclea el botón "Perfil de usuario"
Salida	La aplicación abre correctamente la actividad "Enviar falla"	La aplicación abre correctamente la actividad "Enviar falla"	La aplicación abre correctamente la actividad "Enviar falla"	La aplicación abre correctamente la actividad "Enviar falla"
Entrada	El usuario teclea el botón "Atrás"	El usuario teclea el botón "Atrás"	El usuario teclea el botón "Atrás"	El usuario teclea el botón "Atrás"
Salida	La aplicación devuelve al usuario correctamente a la actividad "Menú principal"	La aplicación devuelve al usuario correctamente a la actividad "Menú principal"	La aplicación devuelve al usuario correctamente a la actividad "Menú principal"	La aplicación devuelve al usuario correctamente a la actividad "Menú principal"
Resultados Finales	Los resultados fueron los esperados.			
Caso de Prueba No 2.	Entrar y salir de la aplicación manteniendo la sesión abierta			
Entrada	El usuario accede correctamente al menú principal. El usuario pulsa el botón "Home". El usuario vuelve a entrar a la aplicación			
Salida	La aplicación direcciona al usuario a la actividad "Menú principal" correctamente.			
Observaciones	Para comprobar la satisfacción de las mejoras, se solicitará que otro programador (pruebas) realice la prueba completa.			
Resultados obtenidos	SATISFACTORIO Sin errores	ACEPTABLE De 1 a 2 errores	INSATISFACTORIO De 3 o más errores	
Caso 1	La aplicación y cierra todas las actividades desde la actividad "Menú principal"			

Proyecto de Estadía
Laboratorios Texzal de México



Caso 2	La aplicación direcciona al usuario a la actividad “Menú principal” al salir totalmente de la aplicación.		
	Descripción		
Mejoras	1.		
	2.		
Resultados Finales	Este módulo realiza su tarea satisfactoriamente.		

Tabla 28 Menú principal

Proyecto de Estadía
Laboratorios Texzal de México



Nombre del módulo o funcionalidad	Enviar falla		
Objetivo de la prueba	Verificar que la actividad registre la falla escrita como “En curso”		
Caso de Prueba No 1.	Enviar una falla		
Descripción	El usuario puede enviar una falla desde esta actividad y ponerla en estado “En curso”, después se abre la actividad “Fallas en curso”		
Entrada	El usuario pulsa el botón “Reportar falla” El usuario escribe el detalle de una falla. El usuario pulsa el botón “Enviar falla”		
Salida	La aplicación abre la actividad “Fallas en curso” con la falla recién agregada en la lista de la nueva actividad.		
Resultados Finales	Los resultados fueron los esperados.		
Observaciones	Para comprobar la satisfacción de las mejoras, se solicitará que otro programador (pruebas) realice la prueba completa.		
Resultados obtenidos	SATISFACTORIO Sin errores	ACEPTABLE De 1 a 2 errores	INSATISFACTORIO De 3 o más errores
Caso 1	La actividad registra correctamente la falla como “En curso” y abre la actividad “Fallas en curso”		
Mejoras	Descripción		
	1.		
	2.		
Resultados Finales	Este módulo realiza su tarea satisfactoriamente.		

Tabla 29 Enviar falla

Proyecto de Estadía
Laboratorios Texzal de México



Nombre del módulo o funcionalidad	Fallas en curso		
Objetivo de la prueba	Verificar que la actividad muestre todas las fallas en curso que no han sido canceladas con los datos básicos.		
Caso de Prueba No 1.	Visualizar todas las Fallas en curso		
Descripción	El usuario puede visualizar todas las fallas en curso que se han enviado.		
Entrada	El usuario entra a la actividad “Fallas en curso” desde “Menú principal”		
Salida	La aplicación muestra todas las fallas en curso dadas de alta si hay alguna. La aplicación muestra un mensaje con el siguiente texto “No hay fallas en curso por ahora”		
Resultados Finales	Los resultados fueron los esperados.		
Caso de Prueba No 2.	Acceder a los detalles		
Descripción	El usuario puede visualizar los detalles de cada una de las fallas en curso.		
Entrada	El usuario pulsa el botón “Detalles”		
Salida	La aplicación muestra la actividad “Detalles de falla en curso” correctamente.		
Observaciones	Para comprobar la satisfacción de las mejoras, se solicitará que otro programador (pruebas) realice la prueba completa.		
Resultados obtenidos	SATISFACTORIO Sin errores	ACEPTABLE De 1 a 2 errores	INSATISFACTORIO De 3 o más errores
Caso 1	La actividad muestra todas las fallas correctamente si es que hay alguna en curso, y muestra un mensaje de retoralimentacion cuando no hay fallas en curso. El botón “Detalles” dirige al usuario a la actividad “Detalles de falla en curso” correctamente		
Mejoras	Descripción		
	1.		
	2.		
Resultados Finales	Este módulo realiza su tarea satisfactoriamente.		

Tabla 30 Fallas en curso

Proyecto de Estadía
Laboratorios Texzal de México



Nombre del módulo o funcionalidad	Detalles de falla en curso		
Objetivo de la prueba	Verificar que la actividad muestra todos los datos de la falla en curso seleccionada. Verificar la funcionalidad de cada uno de los botones.		
Caso de Prueba No 1.	Visualizar los datos de la falla en curso correctamente		
Descripción	El usuario puede visualizar los detalles de cada falla en curso agregada.		
Entrada	El usuario entra a la actividad “Detalles de falla en curso” desde el botón “Detalles” de la actividad “Falla en curso”		
Salida	La aplicación muestra todos los datos correctamente según la falla en curso seleccionada		
Resultados Finales	Los resultados fueron los esperados.		
Caso de Prueba No 2.	Botón “Detener falla”		
Descripción	El usuario puede detener la falla si es que es del nivel correspondiente.		
Entrada	En la actividad Detalles de falla en curso, el usuario pulsa el botón “Detener esta falla”		
Salida	La aplicación muestra una ventana de dialogo que muestra donde se debe escribir la contraseña de usuario.		
Entrada	El usuario escribe la contraseña de manera errónea	El usuario escribe la contraseña correcta	El usuario deja el campo Contraseña vacío
	El usuario pulsa el botón “Detener la falla”	El usuario pulsa el botón “Detener la falla”	El usuario pulsa el botón “Detener la falla”
Salida	La aplicación muestra un mensaje con el texto “Tu contraseña es incorrecta”	La aplicación muestra un mensaje con el texto “Has detenido esta falla”	La aplicación muestra un mensaje con el texto “Debes confirmar tu contraseña para detener la falla”
Salida	Si se completa el número de confirmaciones por usuarios según el nivel de la falla, la falla se completa y se guarda permanentemente en la base de datos.		
Caso de Prueba No 3.	Botón “Cancelar falla”		
Descripción	El usuario puede cancelar las fallas que se encuentren en curso.		
Entrada	En la actividad Detalles de falla en curso, el usuario pulsa el botón “Cancelar esta falla”		
Salida	La aplicación muestra una ventana de dialogo de confirmación con el texto “Pulsa OK para cancelar esta falla en curso”		
Entrada	El usuario pulsa el botón “Cancelar”	El usuario pulsa el botón “OK”	
Salida	La aplicación cierra el dialogo y vuelve a la pantalla de Detalles de falla en curso	La aplicación muestra un mensaje con el texto “Falla en curso cancelada correctamente”	

Proyecto de Estadía
Laboratorios Texzal de México



Observaciones	La aplicación elimina la falla en curso y devuelve al usuario al Menú principal		
	Para comprobar la satisfacción de las mejoras, se solicitará que otro programador (pruebas) realice la prueba completa.		
Resultados obtenidos	SATISFACTORIO Sin errores	ACEPTABLE De 1 a 2 errores	INSATISFACTORIO De 3 o más errores
Caso 1	La actividad muestra todos los datos de la falla en curso que se selecciona.		
Caso 2	La actividad cumple su objetivo a detener una falla solamente cuando 2 o 3 personas (según el nivel que alcanzó la falla) confirman detener la falla con su contraseña en la interfaz de Detalla falla en curso.		
Caso 3	La actividad cancela la falla en curso si se pulsa el botón de confirmación.		
Mejoras	Descripción		
	1.		
	2.		
Resultados Finales	Este módulo realiza su tarea satisfactoriamente.		

Tabla 31 Detalles de falla en curso

Proyecto de Estadía
Laboratorios Texzal de México



Nombre del módulo o funcionalidad	Lista de fallas		
Objetivo de la prueba	Verificar que la actividad muestre todas las fallas completadas.		
Caso de Prueba No 1.	Visualizar todas las Fallas completadas		
Descripción	El usuario puede visualizar todas las fallas completadas por usuarios.		
Entrada	El usuario entra a la actividad “Lista de fallas” desde “Menú principal”		
Salida	La aplicación muestra todas las fallas completadas dadas de alta si hay alguna. La aplicación muestra un mensaje con el siguiente texto “No hay fallas completadas por ahora”		
Resultados Finales	Los resultados fueron los esperados.		
Observaciones	Para comprobar la satisfacción de las mejoras, se solicitará que otro programador (pruebas) realice la prueba completa.		
Resultados obtenidos	SATISFACTORIO Sin errores	ACEPTABLE De 1 a 2 errores	INSATISFACTORIO De 3 o más errores
Caso 1	La actividad muestra todas las fallas correctamente si es que hay alguna falla completada, y muestra un mensaje de retroalimentación cuando no hay fallas en curso.		
Mejoras	Descripción		
	1.		
	2.		
Resultados Finales	Este módulo realiza su tarea satisfactoriamente.		

Tabla 32 Lista de fallas

Proyecto de Estadía
Laboratorios Texzal de México



Nombre del módulo o funcionalidad	Perfil de usuario	
Objetivo de la prueba	Verificar que la actividad muestre los datos del usuario correctamente y las funciones de los botones cumplan sus objetivos.	
Caso de Prueba No 1.	Visualizar todas los datos de la actividad sobre el usuario	
Descripción	El usuario puede visualizar los datos de su cuenta al entrar a la actividad “Perfil de usuario”.	
Entrada	El usuario entra a la actividad “Perfil de usuario” desde “Menú principal”	
Salida	La actividad muestra correctamente los datos del usuario actual.	
Resultados Finales	Los resultados fueron los esperados.	
Caso de Prueba No 2.	Botón “Cerrar esta sesión”	
Descripción	El usuario puede cerrar su sesión para iniciar otra en el mismo dispositivo.	
Entrada	El usuario entra a la actividad “Perfil de usuario” desde “Menú principal” El usuario pulsa el botón “Cerrar esta sesión”	
Salida	La actividad se cierra, elimina los datos locales utilizados por la cuenta correctamente y se despliega la actividad “Login”	
Resultados Finales	Los resultados fueron los esperados.	
Caso de Prueba No 3.	Botón “Modificar este usuario”	
Descripción	El usuario puede modificar los datos de su cuenta con este botón.	
Entrada	El usuario entra a la actividad “Perfil de usuario” desde “Menú principal” El usuario pulsa el botón “Modificar este usuario”	
Salida	La actividad muestra un diálogo que con campos acerca de los datos del usuario.	
Entrada	El usuario modifica sus datos y pulsa el botón “Actualizar cambios”	El usuario pulsa el botón “Cancelar”
Salida	La actividad cierra el diálogo, muestra un mensaje con el texto “Datos actualizados correctamente”	La actividad cierra el diálogo y vuelve al perfil de usuario
Resultados Finales	Los resultados fueron los esperados.	
Caso de Prueba No 4.	Botón “Eliminar este usuario”	
Descripción	El usuario puede eliminar su cuenta con este botón.	
Entrada	El usuario entra a la actividad “Perfil de usuario” desde “Menú principal” El usuario pulsa el botón “Eliminar este usuario”	
Salida	La actividad muestra un diálogo de confirmación con el texto “Pulsa OK para eliminar este perfil”	
	El usuario pulsa el botón “OK”	El usuario pulsa el botón “Cancelar”

Proyecto de Estadía
Laboratorios Texzal de México



Entrada			
Salida	La actividad cierra el diálogo, elimina los datos del usuario de la aplicación y se inicia la actividad "Login"	La actividad cierra el diálogo y vuelve al perfil de usuario	
Resultados Finales	Los resultados fueron los esperados.		
Observaciones	Para comprobar la satisfacción de las mejoras, se solicitará que otro programador (pruebas) realice la prueba completa.		
Resultados obtenidos	SATISFACTORIO Sin errores	ACEPTABLE De 1 a 2 errores	INSATISFACTORIO De 3 o más errores
Caso 1	La actividad muestra todos los datos del usuario actual correctamente.		
Caso 2	La actividad cierra la sesión del usuario correctamente.		
Caso 3	La aplicación muestra los datos del usuario correctamente. Los cambios al perfil son realizados correctamente.		
Caso 4	La actividad elimina correctamente al usuario y devuelve a la pantalla de inicio.		
Mejoras	Descripción		
	1.		
	2.		
Resultados Finales	Este módulo realiza su tarea satisfactoriamente.		

Tabla 33 Perfil de usuario

Estos casos de uso se aplicaron para probar la funcionalidad de cada uno de las actividades que compone el sistema, tanto para los dispositivos virtuales como en los dispositivos reales.



Tomando en cuenta las pruebas podemos darnos cuenta que la aplicación no puede ser instalada en sistemas operativos menores a Android 5.1Lollipop, ya que las versiones anteriores no tienen las herramientas preinstaladas que necesitamos para el uso de la API firesabe (Figura 114). Afortunadamente la mayoría de los dispositivos ya cuentan con actualizaciones hasta esta versión de android y superior. Un ejemplo es con el dispositivo Motorola G, el cual fue lanzado al mercado desde Noviembre de 2013.

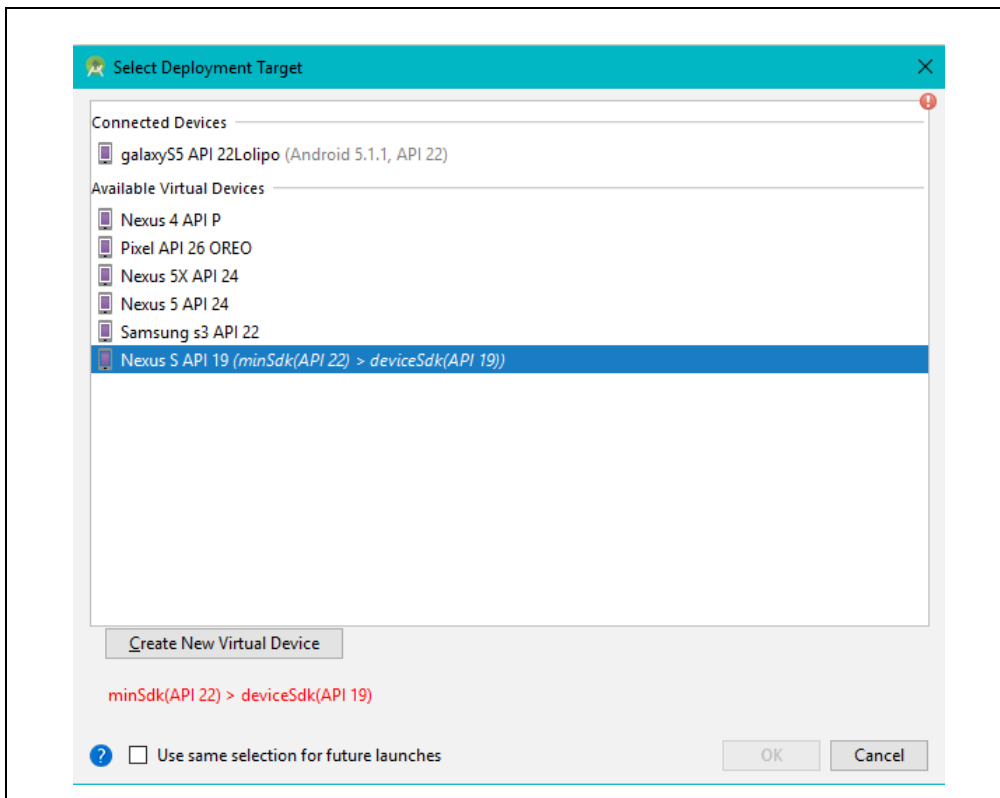


Figura 114 Mensaje de error al intentar compilar en versión menor a 5.1

También se puede notar que el cambio entre diferentes hardware's no es dificultad para usar la aplicación correctamente, la única diferencia es el cambio de posición de algunos elementos pues falta agregar archivos para que la aplicación soporte todas las resoluciones posibles en diferentes dispositivos. A continuación se muestran imágenes de la aplicación funcionando en cada uno de los dispositivos de prueba.

Proyecto de Estadía Laboratorios Texzal de México

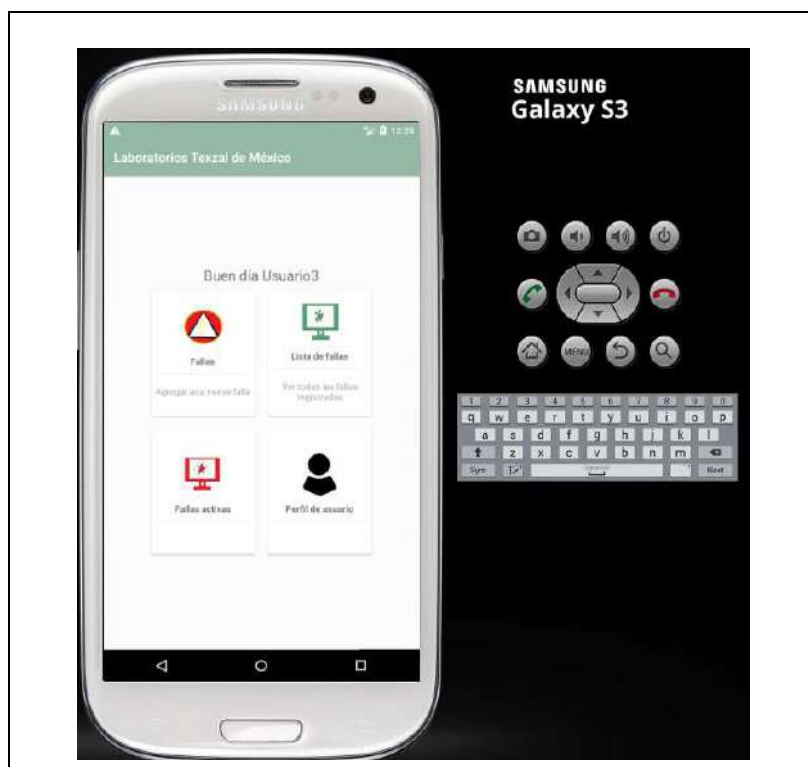


Figura 115 Aplicación en ejecución en SamsunsGalaxy S3

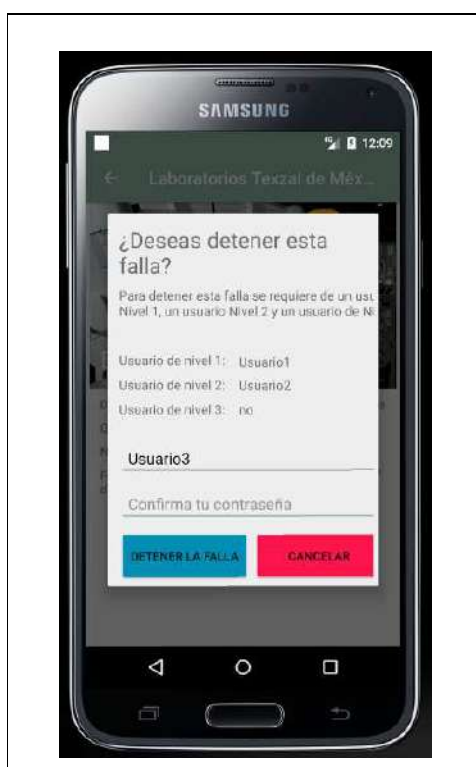


Figura 116 Aplicación en ejecución en SamsunsGalaxy S5

Proyecto de Estadía
Laboratorios Texzal de México



Figura 117 Aplicación en ejecución en Nexus 5X



Figura 118 Aplicación en ejecución en SamsunsGalaxy S4 Real



Proyecto de Estadía Laboratorios Texzal de México



Figura 119 Aplicación en ejecución en Motorola G Real



Figura 120 Aplicación en ejecución en LG Style Real



Proyecto de Estadía Laboratorios Texzal de México

La aplicación funciona igualmente bien tanto si se usa con el dispositivo conectado a una red Wi-Fi o a una red inalámbrica, solo en caso de que el usuario se encuentre en una zona donde la señal del móvil sea afectada, aunque en horario y lugar laboral, no debería haber problema.

A continuación se muestra una imagen del mensaje que recibe el usuario cuando no hay conexión a internet:



Figura 121 Manejo de error en falta de conexión a internet



CONCLUSIONES Y RECOMENDACIONES

El proyecto final realizado como Ingeniero de informática en la Universidad Politécnica de Puebla fue todo un éxito, por lo que es muy reconfortante pues he aplicado todos mis conocimientos adquiridos a lo largo de mi estancia en esta escuela y esta carrera universitaria. Cabe decir que la experiencia fue grata pues el proyecto que escogí realmente me llamaba la atención pues era un tipo de desarrollo nuevo para mí, ya que en los ciclos escolares solemos trabajar con software para equipos de cómputo portátiles o de escritorio, mas no con sistemas móviles. Desde el inicio hubo entusiasmo en realizar el proyecto y para aprender a usar un nuevo ambiente de trabajo con el IDE Android Studio.

Se puede decir que la experiencia con el cliente fue buena y satisfactoria pues se tuvo la disponibilidad de ambos lados para tratar los temas concernientes al desarrollo de la aplicación y que todo saliera como la empresa cliente desea, aparte de desarrollar mis habilidades como analista de sistemas al extraer correctamente los requerimientos funcionales y no funcionales a través de charlas con el cliente, así como el contacto semanal que se tenía por correos y la red social Whatsapp.

Por último, puedo expresar que he mejorado en el manejo de tiempos para desarrollar cada una de las etapas de la metodología que escogí (la cual también fue nueva para mi) pues me parece que ya tenía todos los conocimientos previos para adaptarme a algo nuevo y sacarlo adelante. La aplicación finalmente se terminó en tiempo y forma con la calidad esperada por el cliente y con la satisfacción personal y profesional de quien desarrolló este proyecto.

Actualmente la aplicación puede realizar todas las funciones que la empresa cliente requiere para cubrir su necesidad, aunque tiene capacidad para realizar otras funciones extras a éstas. La idea final del proyecto es que la aplicación pueda ser vendida a diferentes empresas ajustándola a las necesidades de cada una (cuando se requieran nuevas funcionalidades independientemente de las que ya ofrece). Se desea que el proyecto pueda ser desarrollado de tal manera que todo pueda ser configurado por la empresa cliente en un apartado de "Configuraciones", donde se tendrá la libertad de modificar el tiempo de notificación entre niveles, agregar niveles de mantenimiento, agregar departamentos de la empresa donde suelen ocurrir errores, los colores de la interfaz, etc.



GLOSARIO

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

IDE: Sigla de IntegratedDevelopmentEnvironment. Entorno de desarrollo que incorpora todas, o casi todas, las herramientas necesarias (herramientas de modelado o diseño, herramientas de debugging, etc.).



BIBLIOGRAFÍA

- [1] Sommerville, I. (2011). *Ingeniería del software*. Pearson Educación. (Metodologías de desarrollo rápido)
- [2] InaGilfillan. La Biblia de MySQL. Anaya Multimedia
- [3] URL: <https://www.000webhost.com/> (Información acerca de 000webhost. Consulta: 12 de febrero de 2018)
- [4] Android Studio web page, <https://developer.android.com/studio/intro/index.html?hl=es-419> (Información acerca de Android Studio. Consulta: 12 de febrero de 2018)
- [5] Mobile-D homepage, <http://agile.vtt.fi/mobiled.html> (Información acerca de Mobile-D methodology. Consulta: 12 de febrero de 2018)
- [6] Metodología de desarrollo ágil para sistemas móviles. Introducción al desarrollo con Android y iPhone (2009), (Información acerca de fases de Mobile-D methodology)
- [7] Firebase web page, <https://firebase.google.com> (Información acerca de Firebase Studio. (Consulta: Febrero de 2018)
- [8] VTT Electronics (2006). Página de Agile Software DevelopmentMethodologies. Retrievedfrom Mobile-D Method: <http://virtual.vtt.fi/virtual/agile/mobiled.html>
- [9] Android. RetrievedfromAndroidDevelopers: <http://developer.android.com/index.html> (Consulta: Marzo de 2018)
- [10] Using Mobile-D methodology in development of mobileapplications: Challenges and issues (Consulta: Marzo de 2018)
- [11] TestingBaires. Mobile Testing: Emuladores y dispositivos reales, ¿cuál elegir? (2015) Gustavo Terrera. Pruebas para móviles
- [12] Using Mobile-D methodology in development of mobileapplications: Challenges and issues (2013) ZlatkoStapic.