



Instituto Tecnológico Superior de Martínez de la Torre

Desarrollo e implementación de un sistema integral que permita la gestión de la información de dispositivos de control de plagas urbanas, aplicado a la empresa de Servicios Especializados en Fitosanidad y Calidad Agroalimentaria (SEFICA).

TESIS

PARA OBTENER EL GRADO:

**Ingeniería en Sistemas
Computacionales**

PRESENTA:

Luis Fernando Carmona González.

ASESOR:

MRYSI. Ángel Salas Martínez

Martínez de la Torre, Ver.

SEPTIEMBRE, 2019



Nombre del Documento:
Carta de Autorización de Impresión

No. Pág.
1/1

Nº. Oficio DET/ITSMT/ISC/070/2019

ASUNTO: Autorización de Impresión TESIS.

Martínez de la Torre, Ver., a 11 de Septiembre de 2019

C. LUIS FERNANDO CARMONA GONZÁLEZ
NO. DE CONTROL: 140I0026
EGRESADO(A) DE LA CARRERA
INGENIERÍA EN SISTEMAS COMPUTACIONALES
P R E S E N T E

Por medio de la presente hago constar que ha cumplido satisfactoriamente con lo estipulado por el Lineamiento para la Titulación Integral.

Por tal motivo **Se Autoriza** la impresión de la Tesis titulada:

"Desarrollo e implementación de un sistema integral que permita la gestión de la información de dispositivos de control de plagas urbanas, aplicada a la empresa de Servicios Especializados en Fitosanidad Agroalimentaria (SEFICA)."

Dándose un plazo máximo de 30 días naturales a partir de la fecha de la expedición de la presente para realizar la solicitud del Acto de Recepción para la obtención del Título Profesional.

A T E N T A M E N T E


MRSI. ÁNGEL SALAS MARTÍNEZ

Nombre y Firma
Presidente de Academia



JEFATURA DE CARRERA
ING. SISTEMAS
COMPUTACIONALES


MCA. YANAHUI BARRERA CASTELLANOS

Nombre y Firma
Jefe(a) de Carrera de Ingeniería

C.c.p. División de Estudios Profesionales.
C.c.p. Archivo

CONTENIDO

Resumen	1
1. Introducción	2
1.1 Planteamiento del problema	2
1.2 Hipótesis	3
1.3 Objetivos	3
1.3.1 Objetivo General	3
1.3.2 Objetivos específicos	4
1.4 Justificación	5
1.5 Alcances y Limitaciones	6
1.6 Organización del Documento	7
2. Estado del Arte	8
2.1 Aplicación SisMoFrutal	9
2.2 Aplicación Sisa Monitor	10
2.3 Software Hesperides	11
3. Marco teórico	14
3.1 Aplicaciones móviles vs Aplicaciones web	14
3.1.1 Aplicaciones web	15
3.1.2 Aplicaciones nativas	15
3.1.3 Aplicaciones híbridas	16
3.2 Sistemas operativos para móviles	18
3.2.1 IOS	18
3.2.2 Android	19
3.3 Entorno para desarrollo de aplicación	22
3.3.1 Entorno de desarrollo para aplicaciones web	23
3.3.2 Entornos de desarrollo para aplicaciones móviles	25
3.4 Base de datos para dispositivos móviles	28
3.4.1 Firebase	29
3.4.2 SQLite	30
3.4.3 MySQL	31
4. Metodología	33
4.1 Línea de Investigación	33
4.2 Tipo de investigación	33

4.3 Alcance de investigación	34
4.4 Cronograma de investigación	36
4.5 Población Y Muestra.....	36
4.5.1 Selección de la Muestra	37
4.6 Recolección de Datos.....	37
5. Desarrollo e Implementación de la App.....	38
5.1 Metodología de desarrollo.....	38
5.2 Fase de Planificación.....	40
5.2.1 Requerimientos Funcionales y No funcionales de la App.....	40
5.3 Fase de Diseño	42
5.3.1 Arquitectura de la App.....	42
5.3.2 Diagrama de casos de uso	43
5.3.3 Diagrama de clases	45
5.3.4 Diseño de vistas de la aplicación móvil y wireframe de la web	46
5.2 Fase de Codificación	53
5.2.1 Desarrollo de base de datos	53
5.2.2 Codificación de la aplicación móvil.....	55
5.2.3 Codificación del sistema Web.....	68
5.3 Fase de Pruebas.....	77
5.4 Elaboración y ejecución de pruebas	78
Conclusion	80
Recomendaciones	80
Referencias	82

AGRADECIMIENTOS

En este presente trabajo de titulación hare los siguientes agradecimientos.

En primera instancia, agradezco a mis docentes, especialmente a los ingenieros Ángel Salas y Víctor Guillermo Asad por haber depositado toda su confianza en mí como su alumno y asesorado, ellos han sido uno de los principales apoyos para no desistir en este corto, pero arduo camino.

A mis compañeros y amigos, Félix, Juan, Ivette, Ana y Jorge que sin este gran equipo no se hubiera podido llegar hasta este punto.

A mi padre porque siempre me ha enseñado buenos valores ante la sociedad convirtiéndome en un joven de buenos principios.

A mi hermana menor que a pesar de su edad me ha sido el mejor ejemplo de que con constancia y disciplina todo se puede lograr.

A mi novia por acompañarme en los buenos y malos momentos siendo siempre una gran compañera.

Quiero agradecer también a mis tíos o como yo los llamo mis segundos padres Leonardo y Marco que fueron la principal fuente de apoyo a lo largo de toda mi carrera como estudiante.

A mi maestra de secundaria Aurora que cuando pregunto si estudiaría le respondí que sería un Ing. en sistemas mientras ella solo se burló diciendo que con suerte saldría la secundaria.

También sin dejar atrás a mi más grande ángel en el cielo, mi abuelita Ángela que no puedo cumplir su promesa de estaría conmigo en ese gran día y que siempre está en mi corazón.

Y por último a mi madre que con tanto amor me ha acompañado y que a pesar de todo siempre me ha motivado a salir adelante, la que siempre me guio por el camino del bien, la que me apoyo hasta el último momento, que nunca dejo de creer en mí y que siempre encuentra las palabras correctas para lograr no rendirme.

“Recuerda, Sin sacrificio no hay victoria”

-Gracias, mama.

RESUMEN

En la actualidad la mayoría de las personas cuentan con acceso a un dispositivo móvil con conexión a internet, prácticamente todas las personas cuentan con dispositivos móviles personales con los que interactúan en sus tareas cotidianas.

La gran mayoría de tareas que se realizan en el día a día se pueden organizar y hacer desde un dispositivo móvil: comprar boletos de avión, pagar en un restaurante, postear en redes sociales, incluso adquirir productos sin tener que salir de casa, solo con el simple hecho de dar un clic.

La tecnología está cambiando las formas de hacer las cosas y las empresas deben adaptarse a eso o dejar lugar a quien sepa hacerlo.

Es por ello que la implementación de una aplicación móvil que pueda agilizar un proceso o una tarea en el ámbito empresarial ayuda al crecimiento de las empresas al poder contar con avances tecnológicos.

Desarrollar una aplicación móvil para un uso específico no es algo fácil, primero se tiene que hacer un estudio previo con la intención de buscar alguna otra aplicación existente que ya cumpla con funciones similares o incluso que realice el objetivo específico que se ha planteado y poder así saber si es factible seguir con la idea o abandonarla, de seguir con la idea se tiene que buscar el grado de innovación que la app ofrecerá.

Una vez que se toma la decisión de realizar la app se tiene que crear un plan de trabajo, saber bajo que metodología se trabajara, cual serán los alcances, las limitaciones y muchas cosas más.

La presente tiene como objetivo documentar el estudio y la realización de una aplicación móvil a la que se le integra una aplicación web que se ha desarrollado con el fin de agilizar la gestión de la información de los dispositivos de control de plagas urbanas que implementan la empresa de Servicios Especializados en Fitosanidad y Calidad Agroalimentaria (SEFICA).

1. INTRODUCCIÓN

La presente tesis es una investigación que tiene como objetivo observar cómo se pueden agilizar la gestión de la información de los dispositivos de control de plagas urbanas que implementa la empresa SEFICA mediante el uso de un sistema integral usando principalmente una aplicación móvil y así ver el impacto que esta genera en la empresa.

Durante el desarrollo se hablará de la metodología de desarrollo de software que se utilizará ya que esta permitirá una buena estructura de la investigación. También se expondrán diversos lenguajes de programación, así como los sistemas operativos que existen ya que se tiene que tener información general de todas las vertientes que existen al desarrollar una aplicación para dispositivos móviles.

Se hablará de los SGBD (sistemas gestores de base datos) y de los tipos de base de datos que existen, mostrando sus características y hablando de porque se eligió Firebase como SGBD y las ventajas que este ofrecen.

En el último capítulo de esta tesis se mostrará toda la fase de diseño y codificación de las dos aplicaciones concluyendo con el manual de usuario y las respectivas referencias.

1.1 Planteamiento del problema.

La región de Martínez de la Torre es una de las principales zonas del estado de Veracruz que cuenta con las características necesarias para producir diversos frutos y vegetales destacando con mayor fuerza la producción de limón persa y por ser el cítrico con mayor producción existe un gran número de empresas exportadoras del limón persa y para ello se requiere de una exhaustiva vigilancia en cada una de ellas enfocada en todas las áreas que involucran el proceso de empaque y exportación de limón persa, es por esta razón que empresas como Servicios Especializados en Fitosanidad y Calidad Agroalimentaria, a quien a partir de ahora por sus siglas nombraremos SEFICA, se encarga de ofrecer dispositivos para el control de plagas urbanas, no obstante la empresa no cuenta con una herramienta tecnológica que ayude con la administración de la información que proporcionan cada uno de sus

dispositivos al momento de realizar las revisiones semanales y de esta forma cubrir con las necesidades que tiene la empresa como contar con una bitácora digital para poder así automatizar la búsqueda de dicha información al momento de generar reportes.

“Lety es encargada del departamento de control de plagas urbanas de la empresa SEFICA, al estar ella al frente del departamento lleva un control de las empresas y los dispositivos de control instaladas en las mismas, pero solo lo realiza a papel y lápiz lo que hace muy difícil para ella consultar información de manera rápida y eficiente cuando el cliente lo solicita, hasta que conoce a Fernando que le propone ayudarla creando una app que realice todo lo que ella requiere, logrando así agilizar todo el proceso desde un dispositivo móvil con Android logrando también una significativa reducción de costos en insumos en papelería.”

1.2 Hipótesis

- El control de las plagas, mediante un sistema integral ayudara a evitar un ambiente inocuo en la industria.
- El registro de cada empresa, así el control de los dispositivos implementados en cada una agilizará la respuesta ante una plaga en la industria.
- Mantener la información actualizada en tiempo y forma sobre los dispositivos empleados para el control de plagas ayuda a lograr un buen manejo y aportan a la sustentabilidad económica, social y ambiental de la producción.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar e implementar un sistema integral que permita a la empresa de Servicios Especializados en Fitosanidad y Calidad Agroalimentaria (SEFICA), gestionar la información recolectada de las empresas asociadas y los dispositivos empleados por cada una de ellas para el control de plagas urbanas.

1.3.2 Objetivos específicos

- Estudio de las tecnologías de mayor impacto para el desarrollo de aplicaciones móviles.
- Erradicar el uso de hojas de cálculo para almacenar información del registro y seguimiento brindado en cada uno de los dispositivos empleados para el control de plagas.
- Desarrollar un prototipo funcional que permita cubrir la necesidad de obtención de información en menor tiempo.
- Brindar una alternativa para las empresas del giro de control de plagas, en función de almacenamiento y consulta de información.

1.4 Justificación.

La evolución de las tecnologías de información y el incremento considerable en los últimos años del uso de los dispositivos móviles, ha colocado las Apps dentro de las preferencias del usuario, considerando el acceso a información con o sin conexión a internet. En la actualidad, los dispositivos móviles forman un grupo sumamente heterogéneo y pueden incorporar casi cualquier componente de hardware y software que amplíe y diversifique su función inicial.

La creación de una aplicación para la gestión de la información que es recolectada de empresas asociadas y dispositivos empleados en cada una de ellas para el control de plagas urbanas utilizando como plataforma dispositivos móviles que cuenten con un sistema operativo Android con una versión superior a la 7.0. El desarrollo de este proyecto es de suma importancia tanto para las empresas asociadas como para SEFICA que es la empresa a la que se desarrollara el proyecto trabajando en conjunto con el departamento de Control de plagas urbanas de la misma.

Uno de los principales beneficios es que el manejo de la información será más preciso en las revisiones, también la aplicación móvil estará interactuando con la aplicación web en tiempo real, esto nos permitirá saber que equipos ya están revisados, y que empresa está siendo inspeccionada justo después de añadir la información.

Otro de los beneficios es la reducción de costos, ya que al dejar a un lado el uso del lápiz y papel se estaría ahorrando los costos de impresión y de papel, ya que se estarían enviando los reportes de revisión en un formato digital por medio de correo electrónico y no en uno físico como se está realizando hasta ahora, apoyando también a evitar la tala de árboles cuidando el medio ambiente.

Tener la información guardada en una base de datos alojada en la web para poder así ser mostrada en una aplicación contribuiría a evitar traspapelemos en la empresa SEFICA al momento de enviar la información a sus clientes.

Por el lado de la seguridad, utilizar Firebase como nuestro servidor de base de datos es garantía de que la información estará totalmente segura logrando así

poder tener la confianza de la empresa con la que se esa trabajando y así poder dar ese gran salto tecnológico al que a muchas empresas les resulta difícil dar por la resistencia al cambio ya que tiene un proceso y sistema establecido, algo rústicos y que en ocasiones suele ser obsoletos. Utilizar Firebase también ayudara a SEFICA a guardar la información de sus revisiones en caso de que su dispositivo móvil pierda la conexión a internet tomando en cuenta que ciertos lugares o áreas por cuestiones de aislamiento e infraestructura la señal se cancela.

1.5 Alcances y Limitaciones

El presente trabajo se enfoca en el desarrollo de una App para dispositivos móviles que presenten sistemas operativos Android, enfocándose en específico en la versión 7.0 denominada “*Nougat*”, donde se ofrezca la privacidad de los datos tanto de las empresas que solicitan el servicio, así como de los datos de cada uno de los dispositivos inspeccionados. En la primera versión de la App se considerará incluir la promoción y publicidad de la empresa, sin embargo, queda abierta la posibilidad de que la aplicación no se pueda alojar en el Google Play debido a que esto dependerá de los recursos con los que se cuente la empresa para su alojamiento. El contexto de la investigación y su análisis se desarrolla en Martínez de la Torre y sus alrededores a través de la empresa SEFICA.

1.6 Organización del Documento

El presente documento se encuentra conformado por cinco capítulos mismos que se describen a continuación: *Capítulo I: Introducción*; En este capítulo se describe el planteamiento del problema además de la inclusión de la hipótesis, así como los objetivos que se pretenden alcanzar y de igual manera los alcances que se quieren obtener con la realización del proyecto sin dejar a un lado sus limitaciones. En el *Capítulo II: Estado del arte*; Se presenta una breve descripción de algunas aplicaciones o sistemas que realizan actividades similares, o se asemejan en funcionamiento a la que se pretende desarrollar con la investigación realizada en este proyecto. Para el *Capítulo III: Marco teórico*; Se habla de todas las ventajas y desventajas de realizar una aplicación móvil y se describe de todos los tipos de aplicaciones que existen exponiendo los puntos fuertes de cada una, pero destacando por qué se tomó la decisión de utilizar una aplicación móvil. También se habla de los sistemas operativos para dispositivos móviles haciendo un comparativo de Android y IOS de Apple inc., abordando también los SGBD que existen y las diferencias entre ellos que los hacen únicos. En el *Capítulo IV: Metodología*; En este capítulo se aborda, la metodología sobre la que se trabajó la investigación, la línea sobre la que se rige la investigación la cual está avalado por el tecnológico de Martínez, también la selección de la población y la muestra. Para finalizar con el *Capítulo V: Desarrollo e implantación de la app*; En el que se expresa la descripción de las fases que incluye la metodología de desarrollo como lo son: la planeación, el diseño, la codificación al igual que la ejecución de las pruebas en los diferentes dispositivos con los que cuenta la empresa sefica.

2. ESTADO DEL ARTE

El objetivo de realizar una investigación y recopilación de los proyectos, aplicaciones, sistemas y demás herramientas tecnológicas que tomen como punto de partida llevar el control de los dispositivos, revisiones o talvez solo el registro de empresas, podrá abrirle paso a un mundo de ideas para poder llevar a cabo el sistema integral para el control y registro de la información de los dispositivos, los cuales se encarga de monitorear la empresa SEFICA, a esto lo podemos llamar estado del arte o marco referencial, ya que se crea una recopilación de información sobre todo lo ya antes creado en el ámbito tecnológico e implementado en el ramo de la inocuidad para poder cumplir el objetivo principal que es realizar un sistema integral que solucione la problemática y a su vez que muestre características innovadoras que causen impacto en la industria. En este estado del arte se presentarán algunas aplicaciones o sistemas que fueron desarrolladas por empresas o institutos de investigación que fueron en su ramo aplicado una novedad tecnológica, pero se debe tomar en cuenta que el objetivo es implementarlo en otro sector de la inocuidad industrial; Se debe de tomar en cuenta que se extraerán algunas características de cada una de ellas no significando que se realizara una aplicación similar, sino una muy diferente pero con características que se le asemejaran.

Las aplicaciones y sistemas que aquí se presentaran, se han obtenido de artículos de divulgación y de algunos portales de internet en los cuales se muestran las respectivas descripciones y videos del funcionamiento de ellas, también permiten poder descargar sus aplicaciones desde la Play Store, ya que son proyectos ya puestos en funcionamiento. Resaltando también que algunas de las aplicaciones que se presentan están sus servicios apegados a una suscripción por lo cual solo se obtuvieron pequeñas descripciones.

2.1 Aplicación SisMoFrutal

Según el diario (Diario Jornada, 2017), SisMoFrutal se trata de una aplicación para la toma de datos de trampas y estaciones de monitoreo para el Sistema de Monitoreo de Frutales del Instituto Nacional de Tecnología Agropecuaria (**INTA**), es un sistema para el monitoreo de plagas en frutales. Está conformado por dos partes: por un lado, es una app para dispositivos móviles que permite el ingreso de datos a campo y, por otro, un sistema de administración y consultas mediante el que es posible visualizar los datos, desde la web. También permite asignar distintos permisos a usuarios para administración la información, de acuerdo con las responsabilidades en la toma de decisiones ver figura 1. Permite configurar si se descargarán en el dispositivo las fotos de muestra. Si se deshabilita la descarga, se borrarán las fotos almacenadas en el dispositivo y se descargarán en el momento de la visualización (sin almacenarse en el dispositivo), la visualización será más lenta y requerirá el uso de datos, con la descarga habilitada, las fotos se descargarán todas juntas y se almacenarán en el dispositivo, siendo la visualización de forma rápida.



Figura 1. Aplicación móvil SISMOFRUTAL. Fuente: Diario Jornada, 2017

Dentro de los beneficios que ofrece la aplicación se pueden identificar: la comunicación entre una aplicación móvil y un sistema web, el empleo de sensores de temperatura para medir el desarrollo de la larva, la creación de graficas a partir de la recolección de la información, ubicar geográficamente la zona en la que se está teniendo mayor índice de población de la larva.

2.2 Aplicación Sisa Monitor

La aplicación fue desarrollada para registrar el monitoreo de plagas, enfermedades y reportes en la Web, que favorecen a la toma de decisiones de manera más rápida e informada en diversos cultivos, generando ahorro. Entregar información exacta y confiable del estado fitosanitario de la finca (Ruiz, 2016). El monitoreo de plagas y enfermedades es un gran problema en los programas de protección de cultivos. Un registro continuo de la presencia de las plagas, y de sus enemigos naturales, monitoreadas por medio de trampas, distribuidas a través del campo, y/o por la revisión cuidadosa de una determinada población de plantas; es una actividad laboriosa que toma tiempo, pero muy necesaria para tomar la decisión de aplicar plaguicidas. Sería de gran ventaja para los agricultores tener un sistema tecnológico asequible que reduzca significativamente el tiempo del monitoreo en el campo, y que de forma automática calcule porcentualmente, la incidencia y severidad de plagas y enfermedades, así como su relacionamiento frente a los umbrales económicos establecidos para los cultivos. Es un sistema tecnológico basado en una aplicación móvil para dispositivo Android que permite recolectar datos sin necesidad de conexión a Internet, así como la generación de reportes en la web donde se visualizan fácilmente los resultados, en tiempo real, planos, gráficos y tendencias, exponiendo la ubicación en el campo de la plaga o enfermedad, analizando la dinámica poblacional de las plagas, para poder planificar una estrategia de control. Con esta información se logra reducir el número de aspersiones con respecto a la costosa programación de aplicaciones calendario acostumbrada; y, además del impacto en la reducción de costos de producción, se promueve la acción de fauna benéfica natural o inducida sobre las poblaciones del insecto-plaga ver Figura 2.

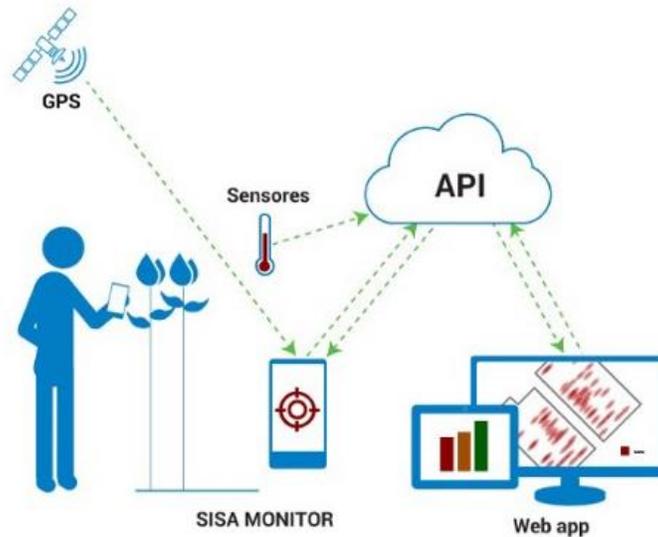


Figura 2. Esquema de funcionamiento de Sisa Monitor. Fuente: Ruiz, 2016.

La aplicación más amigable en el mercado para control fitosanitario. Recolecta datos en cultivos sin necesidad de conectividad. Se complementa con el servicio de reportes en web. Es la herramienta eficiente para el monitoreo de plagas y enfermedades en rosas, adaptable a otros cultivos. Dentro de sus beneficios se pueden destacar: fácil uso, ahorro de tiempo, optimización de recursos, focalización.

2.3 Software Hesperides.

Hesperides es un software agrícola diseñado para llevar un control de las actividades y la trazabilidad de la producción en una explotación agraria, de acuerdo a lo estipulado por el Ministerio de Agricultura, que obliga a mantener actualizado el denominado cuaderno de campo o cuaderno de explotación. También sirve para poner en contacto a equipos técnicos y asesores con productores. Cuenta además como herramienta extra con una calculadora que permite conocer cuáles son los principales indicadores de sostenibilidad del cultivo (huella hídrica, huella de carbono y eficiencia en uso del agua), aunque este módulo no está disponible en la versión de prueba.

La aplicación web, que es el principal recurso de la herramienta (las aplicaciones móviles son básicamente para la recogida de información en campo), presenta las siguientes opciones:

- *Administración del cuaderno de campo*, desde donde se crean los denominados *Planes de control y Observaciones*, que permiten definir durante las visitas a campo si el cultivo está listo para su cosecha, por ejemplo.
- *Gestión de trampas del técnico*, sección vinculada al control biológico de plagas sin utilizar sustancias químicas. Cada trampa se identifica con un código alfanumérico, el tipo al que corresponde, la especie a capturar, la ubicación o el fabricante de la feromona. La introducción del dato relativo a las capturas se puede hacer desde la app móvil o desde la propia interfaz web.
- *Cuaderno de la explotación agraria*, apartado en el que los productores introducen los aspectos relativos a la gestión diaria de sus explotaciones: registros, parcelario, personal, maquinaria, equipos, precios de productos e informes.
- *Efectivos productivos*, sección en la que las explotaciones se pueden organizar o agrupar por clústeres (por ejemplo, todos los frutos con hueso). Esta función resulta útil para el posterior envío de noticias segmentadas.
- *Cuaderno de los técnicos*, apartado desde donde los asesores llevan un control de las explotaciones a su cargo, pudiendo gestionar noticias (por ejemplo, aviso de plagas a los productores afectados), estrategias a desarrollar (aplicación de fitosanitarios o fertilizantes), planificación de visitas, revisiones (información introducida en campo, por ejemplo, que exige una revisión), etc.
- *Administración del cliente*, sección desde la cual se pueden crear nuevas explotaciones, cultivos o campañas.

Asimismo, en la parte superior izquierda, junto al avatar ver figura 3, están los accesos al apartado de información, que permite cambiar el idioma y rellenar datos accesorios acerca del usuario, y el acceso a los video-tutoriales de la aplicación.

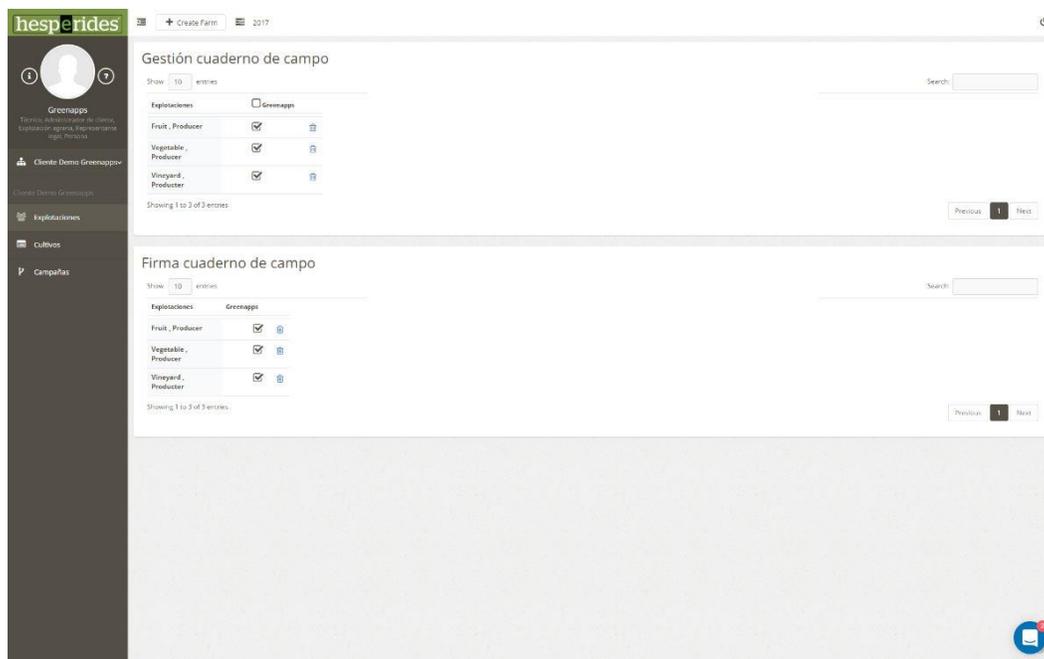


Figura 3. Accesos del apartado de información. Fuente: eFoodPrint, 2017

Por otro lado, la app reduce los tiempos de registro de información y mejora la capacidad de toma de decisiones gracias a la disponibilidad y envío de información desde cualquier lugar.

Para determinar la factibilidad y el impacto que puede presentar el desarrollo de la aplicación para SEFICA, depende en gran medida de los estudios previos realizados, ya que se pueden encontrar fortalezas de los existentes y trabajar en las debilidades para conseguir un resultado que sea más eficiente. Una vez realizado el análisis de las aplicaciones y sistemas encontrados, cuyas características son semejantes al desarrollo propuesto en esta investigación, se puede destacar que no se encontraron estudios o desarrollos enfocados al control de plagas urbanas, además de destacar que las antes mencionadas funcionan sin la presencia de una conexión a internet existente para poder almacenar la información, situación que se

traduce como a una fortaleza fundamental dentro del desarrollo a realizar, considerando que la mayor parte del tiempo las personas que usan la aplicación móvil se encontraran propensas a no contar con una conexión a internet disponible para efectuar la carga de la información, además de lo anterior, en el desarrollo propuesto se podrá emplear el control de plagas urbanas en forma general, por lo que no está limitada solo a frutos o solo a plantas, abriendo una brecha para su implementación en diversos tipos de cultivos.

3. MARCO TEÓRICO.

3.1 Aplicaciones móviles vs Aplicaciones web

En función del objetivo que se persiga, puede ser más interesante una aplicación móvil o una aplicación web, todo depende de la estrategia del cliente y cada opción tiene sus ventajas e inconvenientes ver figura 4. Pero, además, deben de tomar en cuenta para qué está utilizando el usuario las aplicaciones móviles y la web móvil: por ejemplo, los usuarios prefieren las Apps para las aplicaciones más habituales para las que disponen de Apps específicas. Sin embargo, la web móvil es la opción preferida para buscar información, sobre todo si es de carácter esporádico, y para realizar compras.

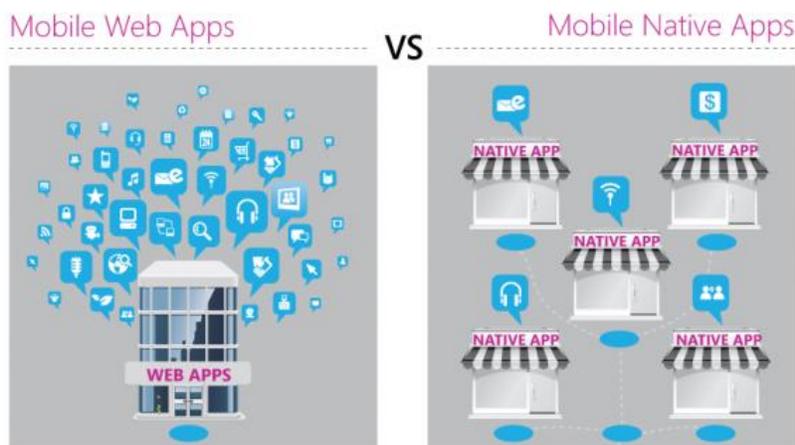


Figura. aplicaciones nativas vs web. Fuente: Elliot Lemenager,2014.

3.1.1 Aplicaciones web

Las aplicaciones web reciben este nombre porque se ejecutan en internet. Es decir que los datos o los archivos en los que trabajan son procesados y almacenados dentro de la web. Estas aplicaciones, por lo general, no necesitan ser instaladas en una computadora o dispositivo móvil. El concepto de aplicaciones web está relacionado con el almacenamiento en la nube. Toda la información se guarda de forma permanente en grandes servidores de internet y envían a los dispositivos o equipos datos que se requieren en ese momento, quedando una copia temporal dentro del dispositivo. En cualquier momento, lugar y desde cualquier dispositivo se puede tener acceso a este servicio, sólo se necesita contar con una conexión a internet y todos los datos de acceso, que por lo general son el nombre de usuario y contraseña.

Estos grandes servidores de internet que prestan el servicio de alojamiento están ubicados alrededor de todo el mundo, así hacen que el servicio prestado no sea tan costoso o gratuito en la mayoría de los casos y extremadamente seguro. [1]

Función de una aplicación Web

Cuando se utiliza una aplicación web desde una computadora o dispositivo móvil, pero la mayor parte del procesamiento se hace dentro de una red de servidores. Estos servidores pueden unir todo su poder de procesamiento con el fin de tramitar solicitudes de todo el mundo, y a su vez, utilizan servidores especializados para almacenar los datos con los que estás trabajando, así como los datos de los demás usuarios. Como todo esto sucede sin problema ni demora alguna, pareciera que la aplicación se está ejecutando dentro del equipo.

3.1.2 Aplicaciones nativas

Las aplicaciones nativas se denominan así porque se desarrollan en el lenguaje nativo del propio terminal. Dependiendo de la plataforma para la que se requiera una aplicación será desarrollada en un lenguaje específico. Por ejemplo, para desarrollar en Android se utiliza Java, en iOS Objective C, y ahora también el nuevo Swift, y en Windows Phone se usa C# y Visual Basic. NET ver figura 5.

Sistema operativo	Fabricante	Lenguaje de programación
Android	Google	Java
IOS	Apple	Objetive C, Swift
Windows Phone	Microsoft	C#, Visual Basic. NET
BlackBerry OS	RIM	C/C++

Figura 5. Tabla de aplicaciones híbridas. Fuente: qode.pro, 2014.

Estas aplicaciones se alimentarán de los recursos del propio Smartphone, teniendo acceso a diferentes características como la cámara, el GPS, entre otras. Además de esto, tienen muchas ventajas ya que, al estar diseñadas directamente para el software del terminal, tendrán un rendimiento optimizado, así como una interfaz mucho más adaptada al sistema operativo al cual el usuario está acostumbrado. Es por esto que son las favoritas del mercado debido a que ofrecen resultados más potentes en cuanto a diseño, usabilidad y eficiencia se refiere. Su distribución se hace a través de las tiendas oficiales de cada sistema operativo, lo que garantiza una visibilidad y seguridad plena.

Otra de sus ventajas es que permiten su uso sin necesidad de conexión a internet, aunque esto no quita que en alguna de sus partes la requiera. Las notificaciones push son otro de su fuerte, así como la creación de un acceso directo en pantalla principal después de la instalación. [2]

3.1.3 Aplicaciones híbridas

Una aplicación híbrida es independiente de los sistemas operativos iOS y Android. Se crea con un Core HTML5(*HyperText Markup Language*) y una única interfaz. Posteriormente se compila dentro de un contenedor nativo para lanzarse a través de todas las plataformas. Esto permite reducir la inversión inicial en el proyecto y a medio-largo plazo no requiere mantener dos bases de código diferentes.

Las aplicaciones móviles híbridas se construyen de manera similar a las páginas web, con una combinación de HTML5, CSS(*Cascading Style Sheets*) y JavaScript.

Sin embargo, en lugar de verse en un navegador móvil (como se ve una aplicación web normal) éstas aplicaciones web se conciben para usarse a través de un WebView, es decir, un navegador integrado dentro de una aplicación nativa. De esta forma la mayor parte de la información se carga a medida que el usuario navega a través de la app.

Este desarrollo híbrido permite, por un lado, hacer uso de algunas funciones de hardware del dispositivo móvil, como el acelerómetro, la cámara o los contactos, que son funciones que a menudo tienen restringido el acceso desde navegadores móviles ver Figura 6. Y, por otro lado, simular de cierta manera la experiencia del usuario, ya que una app híbrida aparecerá como cualquier otra aplicación que encuentres en un Smartphone: se instalan en el dispositivo y puedes encontrar en las tiendas de aplicaciones. [3]

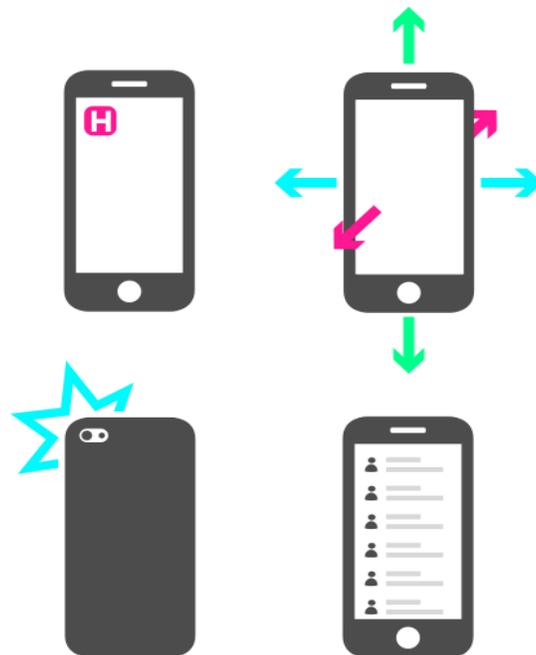


Figura 6. Funciones de una app híbrida. Fuente: cuatrochenta.com 2017.

3.2 Sistemas operativos para móviles

Un sistema operativo móvil o SO móvil es un conjunto de programas de bajo nivel que permite la abstracción de las peculiaridades del hardware específico del teléfono móvil y provee servicios a las aplicaciones móviles, que se ejecutan sobre él. Al igual que los PC que utilizan Windows, Linux o Mac OS, los dispositivos móviles tienen sus sistemas operativos como Android, iOS, Windows Phone o BlackBerry OS, entre otros. Los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos. Los Sistemas Operativos para teléfonos móviles se vuelven cada día más importantes pues la tecnología avanza y en materia de comunicaciones aún más, la telefonía celular cada vez se convierte más en una parte importante en la vida de la sociedad que exige más y más, es importante diseñar sistemas que soporten las aplicaciones que se demandan, que sean fluidos, fáciles, accesibles y hasta divertidos.

Es por eso que las compañías móviles han desarrollado una competencia bastante reñida en cuanto al desarrollo de SO se refiere, desde los inicios en los años 90 con las versiones de EPOC32 para PDA's hasta los más avanzados y sofisticados como Android, IOS que además de ser eficientes y estables son multiplataforma, lo que hace que cualquier persona tenga acceso a ellos desde un celular básico hasta un Smartphone. Además, cada vez más usuarios les agradan la idea de manipular y estilizar sus equipos y es lo que los nuevos SO están ofreciendo y esto implica más retos de programación e incluso en el Hardware. [4]

3.2.1 IOS

iOS es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad. No permite la instalación de iOS en hardware de terceros. Actualmente es el segundo sistema operativo móvil más utilizado del mundo, detrás de Android, con una cuota de mercado de entre 10-15% al año 2017. La última versión del sistema operativo es iOS 12 aparecida en junio de 2018 sustituye a iOS 11 con el objetivo principal de mejorar la experiencia del usuario.

Los elementos de control consisten de deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y provee una interfaz fluida. La interacción con el sistema operativo incluye gestos como deslices, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz. Se utilizan acelerómetros internos para hacer que algunas aplicaciones respondan a sacudir el dispositivo (por ejemplo, para el comando deshacer) o rotarlo en tres dimensiones (un resultado común es cambiar de modo vertical al apaisado u horizontal).

En el marco de las filtraciones acerca de los programas de vigilancia mundial de 2013-2014 de Edward Snowden, Der Spiegel publicó que la NSA estadounidense tiene grupos de trabajo dedicados a descifrar los sistemas de seguridad de iOS; además tiene pequeños programas conocidos como scripts que permiten a la agencia vigilar a los usuarios de las distintas versiones del sistema iOS su geolocalización, notas de voz, fotos y otras aplicaciones como Google Earth, Facebook o Yahoo! Messenger.

iOS se deriva de macOS, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo Tipo Unix. iOS cuenta con cuatro capas de abstracción: la capa del núcleo del sistema operativo, la capa de "Servicios Principales", la capa de "Medios" y la capa de "Cocoa Touch".[5]

3.2.2 Android

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró. Android fue presentado en 2007 junto la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. Android es el sistema operativo móvil

más utilizado del mundo, con una cuota de mercado superior al 80 % al año 2017, muy por encima de IOS. [6]

El éxito del sistema operativo lo ha convertido en objeto de litigios sobre patentes en el marco de las llamadas guerras de patentes entre las empresas de teléfonos inteligentes. Según los documentos secretos filtrados en 2013 y 2014, el sistema operativo es uno de los objetivos de las agencias de inteligencia internacionales.

La versión básica de Android es conocida como Android Open Source Project (AOSP).

El 25 de junio de 2014 en la Conferencia de Desarrolladores Google I/O, Google mostró una evolución de la marca Android, con el fin de unificar tanto el hardware como el software y ampliar mercados.

El 17 de mayo de 2017, se presentó Android Go. Una versión más ligera del sistema operativo para ayudar a que la mitad del mundo sin Smartphone consiga uno en menos de cinco años. Incluye versiones especiales de sus aplicaciones donde el consumo de datos se reduce al máximo.

A pesar que IOS sea un sistema operativo exclusivo de Apple y esto genere un alto estatus social de quien use un Smartphone de tan prestigiada marca, Apple solo domina los mercados anglosajones, Japón, y algunos países nórdicos europeos. El resto es territorio Android como se puede observar en la siguiente figura 7.



Figura 7. Sistemas operativos por países. Fuente: Computerhoy.com, 2018

En cuestión de números, la cifra exacta de que sistema operativo se vende más es un poco relativo, pero lo que sí se puede contar o tomar como estadística son los móviles que usan iOS (el sistema operativo de Apple), y Android. Este dato es fácil de medir a través de las apps o las páginas webs ya que el sistema operativo que usa un terminal es un dato público y hace referencia a un Smartphone real y en por supuesto en actividad.

La siguiente gráfica ver figura 8 muestra la información recabada estadísticamente por el portal de internet **Statista.com** durante los últimos 7 años: A la izquierda se pueden apreciar los datos de 2010, y a la derecha, del 2017. [7]

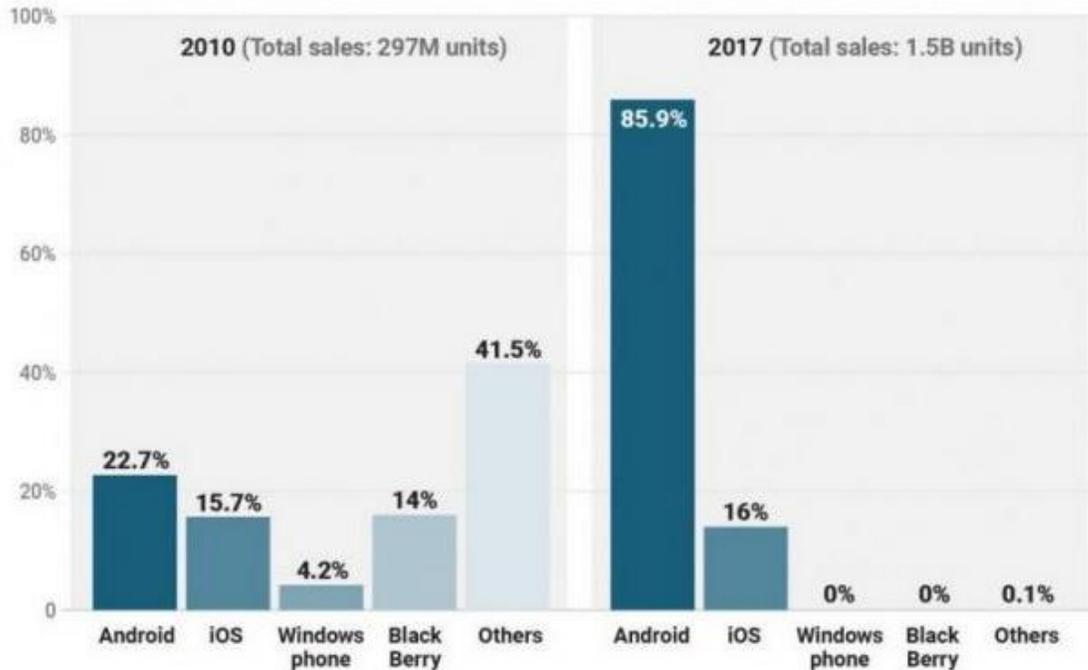


Figura 8. Sistemas operativos activos en apps e internet. Fuente: Statista. 2017

3.3 Entorno para desarrollo de aplicación

Un entorno de desarrollo integrado o entorno de desarrollo interactivo (*Integrated Development Environment- IDE*), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Normalmente, un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de los IDE tienen auto-completado inteligente de código (*IntelliSense*). Algunos IDE contienen un compilador, un intérprete, o ambos, tales como NetBeans y Eclipse; otros no, tales como *SharpDevelop* y *Lazarus*.

El límite entre un IDE y otras partes del entorno de desarrollo de software más amplio no está bien definido. Muchas veces, a los efectos de simplificar la construcción de la interfaz gráfica de usuario (GUI) se integran un sistema controlador de versión y varias herramientas. Muchos IDE modernos también

cuentan con un navegador de clases, un buscador de objetos y un diagrama de jerarquía de clases, para su uso con el desarrollo de software orientado a objetos.

Los IDE están diseñados para maximizar la productividad del programador proporcionando componentes muy unidos con interfaces de usuario similares. Los IDE presentan un único programa en el que se lleva a cabo todo el desarrollo. Generalmente, este programa suele ofrecer muchas características para la creación, modificación, compilación, implementación y depuración de software [8]

Para el trabajo que se realizó fueron utilizados 2 tipos de IDE: Entornos de desarrollo web y Móvil. a continuación, se presentará una descripción de los mismos.

3.3.1 Entorno de desarrollo para aplicaciones web.

Los entornos de desarrollo o editores web son aplicaciones de software que sirven para desarrollar páginas web. Si bien una página web (que son escritas en lenguaje HTML) puede ser creada a partir de un simple procesador de texto, existen programas especializados que facilitan la tarea de los desarrolladores web (webmasters).[9]

En general, existen tres tipos de editores web:

- Editores web de texto: estos ayudan al desarrollador en la escritura del código HTML (y otros códigos de lenguajes web), proveyendo distintos colores al texto de forma automática, acceso a códigos ya definidos, ayuda de asistentes, etc.
- Editores web de objetos: estos alternan entre la edición de texto y el uso de objetos para el desarrollo de una página web.
- Editores web gráficos o WYSIWYG (“Lo que ves es lo que obtienes” por sus siglas en inglés *What You See Is What You Get*) estos proveen una interfaz gráfica, donde es posible modificar y ver la página web tal cual se verá en un navegador. Son los editores más fáciles de utilizar.

Lo primero que se debe de hacer a la hora de elegir un entorno de desarrollo web es tener claro si vamos a utilizar HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) y JavaScript (*Frontend*). En la parte de servidor (*Backend*) existen varias opciones, las más comunes son PHP, Java y ASP.NET. Dependiendo del Backend se elija ver figura 9. [10]



Figura 9. Expiacion grafica sobre Front-end y Back-end Fuente: devcode.la, 2017

Si se toma como base el lenguaje de programación del lado del servidor se pueden agrupar los IDE´S en las siguientes categorías:

PHP

- Netbeans
- Eclipse
- Sublime Text
- Aptana
- Visual Studio Code
- WebMatrix
- Atom

Java

- Netbeans

- Eclipse
- IntelliJ

.NET

- Visual Studio Community

Cuando se habla de los editores WYSIWYG hace referencia a aquellos que permiten crear páginas web de forma gráfica prácticamente sin intervenir en el código HTML, pues el programa lo genera automáticamente como, por ejemplo:

- Dreamweaver
- FrontPage
- NVU
- Amaya
- Adobe Golive
- Writer (de OpenOffice.org)

3.3.2 Entornos de desarrollo para aplicaciones móviles.

Para el desarrollo de aplicaciones móviles se encuentran con muchas opciones de herramientas, lenguajes y entornos para la programación de las mismas. [11]

para empezar el desarrollo de las aplicaciones se tiene que tomar encuesta las posibilidades que existen para el desarrollo de las mimas como, por ejemplo:

- Java ME
- Android
- Windows Mobile

Java ME

En 1999, Sun desarrolló una versión de Java especialmente diseñada para dispositivos móviles, Java 2 Micro Edition, basada en una máquina virtual llamada KVM. Esta primera versión solo contenía una única máquina virtual y un único API, hecho que puso de manifiesto la insuficiencia de esta solución para la gran variedad de dispositivos diferentes que existían. De esta forma, en el 2000 nació la primera

versión de una configuración, el Java 2 Platform, *Micro Edition, connected limited device configuration* (J2ME CLDC 1.0). Una configuración ofrece el API básico para programar dispositivos, aunque no aporta todas las clases necesarias para desarrollar una aplicación completa. Por lo tanto, la primera configuración no tenía las herramientas necesarias para permitir a los desarrolladores escribir programas para el dispositivo Palm. En julio del 2000 nació la primera implementación de un perfil, concretamente el llamado *mobile information device profile* (MIDP), que no estaba destinado a PDA, sino a teléfonos móviles y a paginadores. A partir de este primer perfil, J2ME fue ampliamente aceptado por la comunidad de desarrolladores de dispositivos móviles y se ha ido expandiendo a una gran velocidad hasta la actualidad.

Java ME (Java Micro Edition) (anteriormente conocida como J2ME) es, por lo tanto, una edición de Java orientada a dispositivos pequeños. Es una versión recortada del Java SE con ciertas extensiones enfocadas a las necesidades particulares de este tipo de dispositivos. Esta tecnología consiste en una máquina virtual y en un conjunto de API5 adecuados para estos dispositivos.

Esta plataforma produce normalmente aplicaciones portables, aunque algunas veces existen librerías específicas de cada dispositivo (comúnmente usadas para juegos), que las hacen no portables. A pesar de ello, Java ME se ha convertido en una buena opción para crear aplicaciones para teléfonos móviles, ya que se puede emular en un PC durante la fase de desarrollo y luego se pueden cargar fácilmente las aplicaciones en el móvil. Aunque el proceso no sea directo, resulta bastante económico portarlas a otros dispositivos al utilizar tecnologías Java para el desarrollo.

Se usa muchas veces para proporcionar aplicaciones simples en teléfonos móviles de gama baja. Por lo tanto, las aplicaciones (incluyendo sus datos) no pueden ocupar demasiada memoria si se tienen que ejecutar en la mayoría de estos teléfonos. Además, tienen que estar firmadas criptográficamente para poder usar APIs (*application programming interface*) como la de acceso al sistema de ficheros. Esto es relativamente caro y raramente se hace, incluso para aplicaciones

comerciales. Java ME se ejecuta sobre una máquina virtual que permite un acceso razonable, pero no completo, a las funcionalidades del dispositivo sobre el que se ejecuta la aplicación. El proceso JSR (*java specification requests*) sirve para incrementar gradualmente la funcionalidad disponible para JavaME, mientras proporciona a los operadores y a los fabricantes la capacidad de prevenir o limitar el acceso al software disponible. [8]

Android

Android es una plataforma basada en Linux de la Open Handset Alliance, entre cuyos treinta y cuatro miembros se encuentran Google, HTC, Motorola, Qualcomm y T-Mobile. Por lo tanto, treinta y cuatro de las principales compañías de software, hardware y telecomunicaciones dan soporte a esta plataforma.

El kernel de Linux se usa como HAL (*hardware abstraction layer*). La programación de aplicaciones se hace básicamente en Java. Es necesario el SDK (*software development kit*) específico de Android para desarrollar, aunque se puede usar cualquier IDE (*integrated development environment*) Java. El código que sea crítico en cuanto a rendimiento se puede escribir en C, C++ u otros lenguajes de código nativo usando el NDK (*native development kit*) de Android. [8]

Windows Mobile

La plataforma Windows Mobile estuvo disponible en una gran variedad de dispositivos de diferentes operadores inalámbricos. Se podía encontrar el software Windows Mobile en productos de Dell, HP, Motorola, Palm e i-mate. Los dispositivos con Windows Mobile estaban disponibles para redes GSM o CDMA.

Windows Mobile es una variante de Windows CE para teléfonos móviles. Originalmente, Windows CE se desarrolló para ordenadores de bolsillo y PDA con pantallas táctiles que funcionaban con un stylus, y se adaptó posteriormente para su uso en Smartphone equipados con un teclado. Por lo tanto, los teléfonos se convirtieron en la mayor base de dispositivos instalados con CE, aunque la cuota de mercado ha caído dramáticamente desde la aparición de Android y iPhone. Windows Mobile soporta un subconjunto de la interfaz de programación de Win32 y

una GUI simplificada con una ventana en la pantalla a la vez. Las aplicaciones se pueden usar en .NET Compact Framework. Windows Mobile 6.5 introdujo las interfaces estilo iPhone basadas en el contacto con los dedos, mientras que Windows Phone 7 es un rediseño sustancial que usa Silverlight y XNA para interfaces de usuario más ricas. [8]

En base a las características que hacen que un sistema operativo móvil sea más rentable, se tomó la decisión de trabajar el proyecto para su empleo exclusivo en dispositivos que cuenten con alguna versión de sistema operativo Android, y en específico contemplando la versión 7.0 Nougat para el desarrollo del proyecto, dando pauta a que la aplicación pueda ser instalada en la versión 7.0 o superiores.

3.4 Base de datos para dispositivos móviles.

Una base de datos móviles es aquella que es portable y posee una independencia del servidor corporativo de bases de datos, pero puede comunicarse con el servidor desde cualquier punto remoto para compartir datos corporativos.

Las bases de datos móviles se consideran las bases de datos de nueva generación, ya que van dirigidas a sistemas informáticos portátiles como computadoras personales y dispositivos móviles que se conectan con las estaciones base (servidores) mediante redes de comunicación digitales inalámbricas, además deben funcionar incluyendo cuando no estén conectadas a una red a diferencia de los sistemas de bases de datos fijos.

Las bases de datos móviles utilizan técnicas especiales para la administración de memoria debido a la capacidad de almacenamiento limitada con la que cuentan los dispositivos móviles. Además, el hecho de que se almacenen en un dispositivo móvil o computadora personal permite a los usuarios realicen cambios en la información sin necesidad de estar conectados a una red a esto se le suma la posibilidad de sincronizar o consultar datos de los servidores al instante en que el dispositivo utilizado se conecte a internet. [8]

Todas las funcionalidades de una base de datos y las comunicaciones inalámbricas las provee un Sistemas de Gestión de base de datos DBMS (*Data Base*

Management System) son representadas en la Figura 10, que se muestra a continuación.

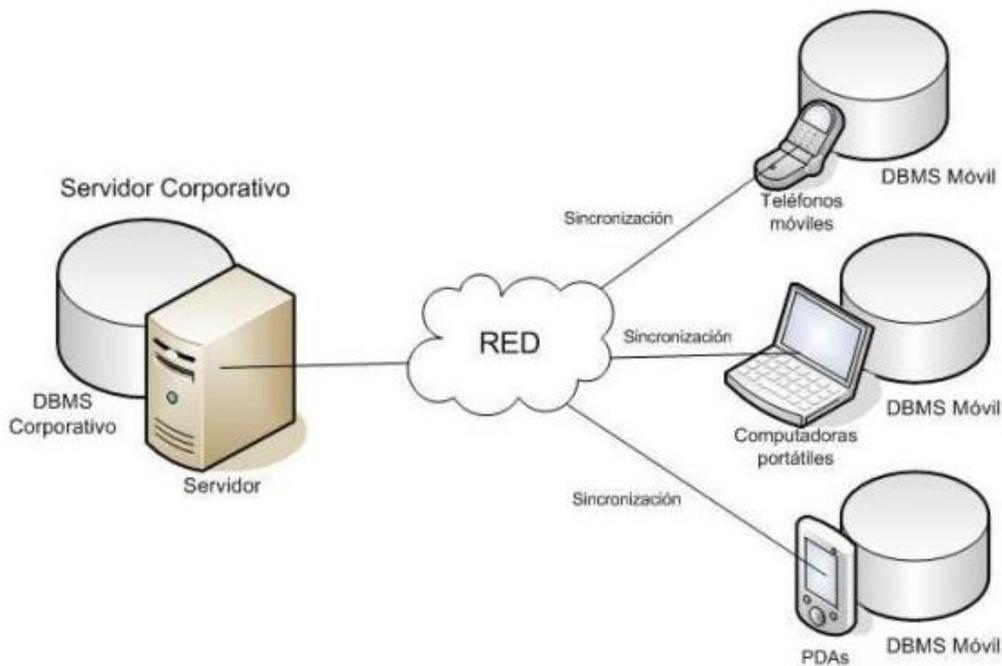


Figura 10. Estructura de una base de datos móviles. Fuente: Thomas M. Connolly, 2005

Dentro de DBMS que existen se mencionaran los más utilizados para después concluir con el porque se utilizó cierto DBMS en el proyecto.

3.4.1 Firebase

Firestore Realtime Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Cuando se compilan apps multiplataforma con los SDK de iOS, Android y JavaScript, todos los clientes comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes.

Firestore Realtime Database permite compilar aplicaciones ricas y colaborativas, ya que permite el acceso seguro a la base de datos directamente desde el código del cliente. Los datos persisten de forma local. Además, incluso cuando no hay conexión, se siguen activando los eventos en tiempo real, lo que proporciona una

experiencia adaptable al usuario final. Cuando el dispositivo vuelve a conectarse, Realtime Database sincroniza los cambios de los datos locales con las actualizaciones remotas que ocurrieron mientras el cliente estuvo sin conexión, lo que combina los conflictos de forma automática.

Realtime Database proporciona un lenguaje flexible de reglas basadas en expresiones, llamado reglas de seguridad de Firebase Realtime Database, para definir cómo se deberían estructurar los datos y en qué momento se pueden leer o escribir. Integrar Firebase Authentication permite que los desarrolladores definan quién tiene acceso a qué datos y cómo acceden a ellos.

Realtime Database es una base de datos NoSQL y, como tal, tiene diferentes optimizaciones y funcionalidades en comparación con una base de datos relacional. La API de Realtime Database está diseñada para permitir solo operaciones que se puedan ejecutar rápidamente. Eso permite crear una excelente experiencia de tiempo real que puede servir a millones de usuarios sin afectar la capacidad de respuesta. Es importante pensar cómo deben acceder a los datos los usuarios y estructurarlos según corresponda. [12]

3.4.2 SQLite

Es una biblioteca en lenguaje C que implementa un motor de base de datos SQL pequeño, rápido, autónomo, de alta confiabilidad y con todas las funciones. SQLite es el motor de base de datos más utilizado en el mundo. Está integrado en todos los teléfonos móviles y en la mayoría de las computadoras, y se incluye en muchas otras aplicaciones que la gente usa todos los días.

El código fuente de SQLite está en el dominio público y es de uso gratuito para todos para cualquier propósito. [13]

Características.

- Las transacciones son atómicas, consistentes, aisladas y duraderas (ACID) incluso después de fallas del sistema y fallas de energía.
- Configuración cero : no se necesita configuración ni administración.
- Implementación SQL completa con capacidades avanzadas como índices parciales , índices de expresiones , JSON y expresiones de tabla comunes .

- Una base de datos completa se almacena en un único archivo de disco multiplataforma . Ideal para usar como formato de archivo de aplicación .
- Admite bases de datos de tamaño terabyte y cadenas y blobs de tamaño gigabyte.
- Huella de código pequeño: menos de 600Ki totalmente configurados o mucho menos con funciones opcionales omitidas.

3.4.3 MySQL

MySQL es un sistema de gestión de bases de datos (SGBD) multiusuario, multiplataforma y de código abierto bajo licencia GNU como también bajo una variedad de acuerdos propietarios.

MySQL es muy utilizado en bases de datos para aplicaciones web, en especial en la combinación LAMP (Linux, Apache, MySQL, Perl/PHP/Python) y otras como MAMP, WAMP, etc. Probablemente la combinación más popular sea MySQL y el lenguaje PHP.

Algunas aplicaciones populares que emplean base de datos MySQL son: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal, entre otras. [14]

Características de MySQL

- MySQL está escrito en C y C++
- Emplea el lenguaje SQL para consultas a la base de datos.
- MySQL Server está disponible como freeware bajo licencia GPL.
- MySQL Enterprise es la versión por suscripción para empresas, con soporte las 24 horas.
- Trabaja en múltiples plataformas como: AIX, BSDi, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Microsoft Windows.

Para el proyecto se seleccionó eh implemento Firebase.

Firestore es un el sistema fue más que intuitiva por la ayuda de su documentación, además que no requirió de la implementación de un web-service que utilizan los datos SQL, también utilizar Firestore fue de mucha ayuda al momento de guardar la

información, al poder almacenar los datos en cache se puede seguir utilizando la aplicación y generando reportes en caso de que allá una pérdida de perdida de señal y no se cuente con una conexión a internet activa con el fin de que cuando se restablezca la conexión este envíe los datos en segundo plano, sin causar alguna interrupción al usuario del dispositivo.

4. METODOLOGÍA

4.1 Línea de Investigación

La presente investigación se ubica dentro de la línea de investigación de Tecnologías de desarrollo web y móvil con clave LGAC-2017-SMTZ-ISCO-08, misma que forma parte de la carrera de Ingeniería en Sistemas Computacionales, como parte de los trabajos de desarrollo y descubrimiento de nuevos conocimientos sobre tecnologías emergentes.

4.2 Tipo de investigación

La investigación es un conjunto de procesos sistemáticos, críticos y empíricos que se aplican al estudio de un fenómeno. [15]

Diferentes autores han dado sus distintas perspectivas de las clasificaciones de las metodologías de investigación más, sin embargo, se realizó una recopilación de información concluyendo en la siguiente clasificación:

- según los paradigmas de la investigación
- según la finalidad de la investigación
- según la dimensión cronológica.

cada uno de estos tres criterios de clasificación de las metodologías de investigación incluye un tipo de investigación que se puede aplicar a los estudios las cuales serán presentadas en la siguiente figura 11.

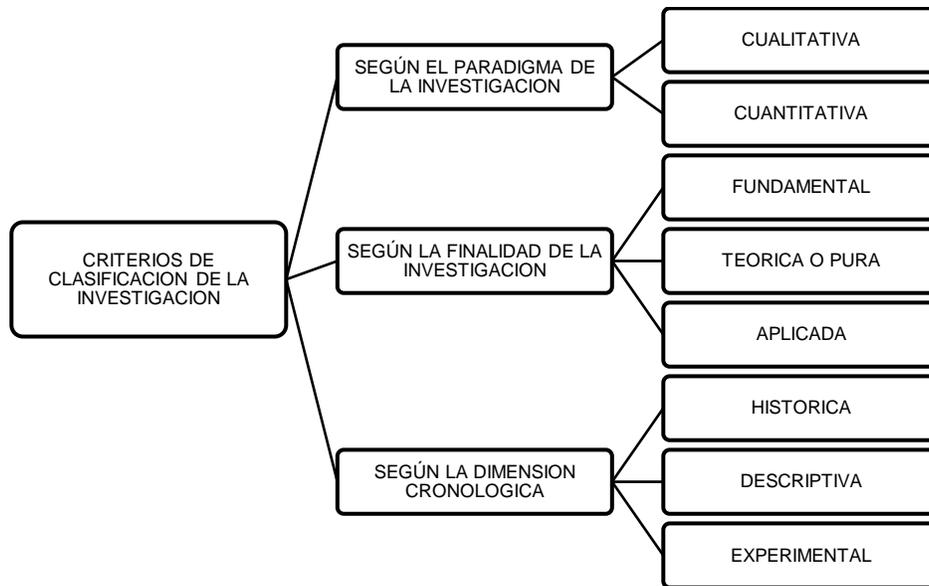


Figura 11. Criterios de la investigación. Fuente: Recopilación propia.

En esta investigación se empleará la investigación pura-aplicada ya que durante la primera etapa se realizó una recolección y estudios de toda la información que se recabó con todos los datos más importantes que tuvieran una relación directa o indirecta con el tema a investigar para poder así ampliar el conocimiento para poder dar solución a un problema, una vez obtenido todo el análisis de la información, poder proceder con el siguiente paso que es la aplicación que es donde fue utilizada la investigación aplicada ya que el producto generará nuevos conocimientos estos están dirigidos al sector productivo de bienes o servicios que se busca obtener un nuevo producto al igual que poder ser competitivo en un sector en específico, como le es el seguimiento a control de plagas.

4.3 Alcance de investigación

Definida la investigación como una del tipo pura-aplicada se puede determinar el alcance de dicha investigación. Según Sampieri en su libro Metodología de la Investigación [15] establece 4 tipos de alcances para diferentes tipos de investigaciones, los cuales son: *Exploratorio*, *Descriptivo*, *Correlacional* y *Explicativo*. Cada uno de estos alcances tienen un fin común que depende de los objetivos que se quieran cumplir con la investigación como si muestra en la siguiente tabla ver figura 12.

Exploratorios	Descriptivo	Correlacional	Explicativo
<ul style="list-style-type: none"> • Investigan problemas poco estudiados • Indagan desde una perspectiva innovadora • Ayudan a identificar conceptos promisorios • Preparan el terreno para nuevos estudios 	<ul style="list-style-type: none"> • Consideran al fenómeno estudiado y sus componentes • Miden conceptos • Definen variables 	<ul style="list-style-type: none"> • Ofrecen predicciones • Explican la relación entre variables • Cuantifican relaciones entre variables 	<ul style="list-style-type: none"> • Determinan las causas de los fenómenos • Generan un sentido de entendimiento • Son sumamente estructurados

Figura 12. Tabla comparativa de los alcances de investigación, Fuente: Sampieri, 2010

Con lo anterior de acuerdo al tipo de investigación tendrá alcance exploratorio, ya que buscando en diferentes fuentes tanto como físicas y digitales no se tiene dato de alguna investigación sobre el uso de un sistema integral para llevar el control de plagas urbanas dentro de la industria que se dedicada a la exportación de los cítricos, lo cual servirá esta investigación para poder ofrecer más información para poder dar solución a un problema que es de suma importancia en el ámbito industrial. Por otro lado, se implementará un sistema integrar para el control de plagas dentro del ámbito industrial el cual tendrá un grado de innovación ya que los usuarios pasaran de utilizar papel y lápiz a utilizar un dispositivo móvil, dando esto un mayor porcentaje de eficacia al momento de consultar la información al ofrecer datos más precisos y seguros.

4.4 Cronograma de investigación

Para mostrar de manera gráfica las actividades que se realizaron durante la investigación en la Figura 13 se muestra el cronograma que define las actividades y tiempo específico en que se realizaron de acuerdo al proceso de desarrollo de la investigación.

ACTIVIDADES		Julio				Agosto		Septiembre				Octubre					Noviembre				Diciembre					Enero			
		1	2	3	4	1	2	1	2	3	4	1	2	3	4	5	1	2	3	4	1	2	3	4	5	1	2	3	4
Generación de la idea y elección del tema	P	■	■	■																									
	R																												
Planteamiento del problema	P		■	■	■	■																							
	R																												
Elaboración de Hipotesis	P					■	■	■	■																				
	R																												
Justificación	P							■	■																				
	R																												
Antecedentes y estado del arte	P									■	■	■	■	■															
	R																												
Revisión de la literatura y desarrollo del marco teórico	P											■	■	■	■	■	■												
	R																												
Selección y desarrollo del diseño de investigación	P																■	■	■	■	■								
	R																												
Definición y selección de la muestra.	P																				■	■	■	■					
	R																												
Recolección y análisis de datos	P																												
	R																												
Elaboración del reporte de resultados	P																												
	R																												

Figura 13. Cronograma de actividades. Fuente: Elaboración propia

4.5 Población Y Muestra

La definición de población es la totalidad de unidades de análisis del conjunto a estudiar a lo que corresponde, de acuerdo a la investigación la población será todas las empresas que están dentro del listado de servicios de “control de plagas urbanas” mismo que ofrece SEFICA en este listado se tiene una población total de 50 clientes.

Para llevar a cabo la muestra se utilizará la muestra probabilística ya que todas las empresas que se encuentran dentro del listado tienen la misma probabilidad de ser parte de la muestra. Debido a que se cuenta con el tamaño de la población se utilizará la fórmula siguiente:

$$n = \frac{N \cdot Z_a^2 \cdot p \cdot q}{d^2 \cdot (N - 1) + Z_a^2 \cdot p \cdot q}$$

En donde, N = tamaño de la población Z = nivel de confianza, P = probabilidad de éxito, o proporción esperada Q = probabilidad de fracaso D = precisión (Error máximo admisible en términos de proporción) [13]

4.5.1 Selección de la Muestra

Todas las empresas tienen la misma oportunidad de ser parte de la muestra, contando que el universo se expande a varios municipios del estado de Veracruz, para la investigación han sido seleccionadas 5 de las empresas que se encuentran dentro del municipio de Martínez de la Torre donde se realiza dicha investigación.

4.6 Recolección de Datos.

La Recolectar los datos implica elaborar un plan detallado de procedimientos que nos conduzcan a reunir datos con un propósito específico. [15]

Para la recolección, las técnicas que se utilizaron fueron los Cuestionarios y observación.

Los cuestionarios fueron realizados por el investigador con base a las dudas que surgieron de acuerdo a la observación del CheckList proporcionado por la empresa SEFICA con el fin de automatizar dicho proceso.

La observación se realizó de igual manera por parte del investigador, esta técnica se llevó a cabo asistiendo a las revisiones de los dispositivos instalados en las empresas seleccionadas como muestra.

5. DESARROLLO E IMPLEMENTACIÓN DE LA APP.

5.1 Metodología de desarrollo

Para poder desarrollar un software se necesita una metodología en la cual se base el proyecto, esto quiere decir que necesitamos una estructura paso a paso de cómo será realizando; Es frecuente que en un ambiente laboral moderno sea difícil o imposible predecir la forma en la que evolucionará un sistema (por ejemplo, una aplicación en web). Las condiciones del mercado cambian con rapidez, las necesidades de los usuarios finales se transforman y emergen nuevas amenazas competitivas sin previo aviso. En muchas situaciones no será posible definir los requerimientos por completo antes de que el proyecto comience. Se debe ser suficientemente ágil para responder a lo fluido que se presenta el ambiente de negocios. [15] Con el fin de evitar que los cambios de interés repentinos en el cliente afecten e desarrollo del proyecto será necesario aplicar la ingeniería de desarrollo de software Ágil.

La ingeniería de software ágil combina una filosofía con un conjunto de lineamientos de desarrollo. La filosofía pone el énfasis en: la satisfacción del cliente y en la entrega rápida de software incremental, los equipos pequeños y muy motivados para efectuar el proyecto, los métodos informales, los productos del trabajo con mínima ingeniería de software y la sencillez general en el desarrollo.

Los lineamientos de desarrollo enfatizan la entrega sobre el análisis y el diseño (aunque estas actividades no se desalientan) y la comunicación activa y continua entre desarrolladores y clientes. [15]

Dentro del panorama ágil existen diferentes opciones. XP (Extreme programming), Scrum o Kanban y cada una de ellas cuenta con una forma diferente de entender la flexibilidad. Así, cada una de las **metodologías ágiles de gestión de proyectos** se caracteriza por:

- **Extreme Programming: también conocida como XP**, se centra en el vínculo que se crea entre el cliente y el equipo de proyecto. Fomenta la interacción permanente entre ambos, que facilita la introducción de cambios y minimiza las posibilidades de error. Requiere de la organización de los

equipos en pequeñas células, con un número de integrantes limitado, no demasiado amplio; por lo que no es recomendable para proyectos de larga duración.

- **Scrum:** es el orden dentro del caos, aceptando la naturaleza cambiante de un proyecto, trata de proponer directrices que simplifiquen su gestión. Ésta es, de todas las metodologías ágiles de gestión de proyectos, la que con mayor eficacia facilita el hallazgo de soluciones específicas para los problemas que van surgiendo durante el desarrollo del proyecto. Pese a sus ventajas, en especial en lo referente a la coordinación de personas, entre sus inconvenientes se encuentra el fomentar un entorno donde los niveles de estrés son difíciles de controlar, en parte debido a la necesidad de trabajar bajo presión para poder realizar las entregas parciales a tiempo.
- **Kanban:** clasifica las tareas según su estatus ayudando a determinar los niveles de productividad en cada fase del proyecto. Su sencillez de aplicación simplifica también la planificación y la designación de responsabilidades. No obstante, su efectividad no es a prueba de todo, ya que la aplicación de ésta, la más famosa de las metodologías ágiles de gestión de proyectos, resulta excesivamente compleja cuando el volumen de trabajo es elevado o la dificultad de las tareas a realizar va en aumento. [16]

La metodología que se utilizará para la investigación será **SCRUM**.

SCRUM es, precisamente, una evolución del Agile Management. Se ha seleccionado ya que por el lado del mercado demanda calidad, rapidez en la entrega y bajos costes, para lo cual se debe ser muy ágil y flexible en el desarrollo del producto para conseguir ciclos de desarrollo cortos que puedan satisfacer la demanda del cliente sin menoscabar por ello la calidad del resultado final.

Otro de los puntos es que se permite mucha flexibilidad a la hora de acometer cambios a mitad del proyecto ya que tras cada fase se replantea las tareas y los objetivos.

Otra de las ventajas de utilizar SCRUM es que funciona para equipos pequeños, para equipos de trabajo que no superan las 8 personas

5.2 Fase de Planificación

Se realizaron reuniones por separado con los asesores tanto interno como externo para hablar sobre los temas más importantes con respecto al proyecto. En la reunión con el asesor externo se realizó una lluvia de ideas sobre las necesidades de la empresa y así definir el objetivo del proyecto al igual que los alcances que tendrá la aplicación. Se llevó un seguimiento personal durante 2 días al proceso de recolección de la información actual para poder así brindar una mayor experiencia con la app a los que por cuestiones de políticas de la empresa no se puede agregar dicha evidencia.

5.2.1 Requerimientos Funcionales y No funcionales de la App

Requerimientos Funcionales.

Los requerimientos funcionales de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones.

Por lo general, estos deben incluir funciones desempeñadas por pantallas específicas, descripciones de los flujos de trabajo a ser desempeñados por el sistema y otros requerimientos de negocio, cumplimiento, seguridad u otra índole.

[17]

Para el sistema se muestran los siguientes requerimientos funcionales:

- El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados. Los usuarios deben ingresar al sistema con un nombre de usuario y contraseña.
- La aplicación podrá ser utilizada en Tablet y Smartphone.
- La aplicación permitirá a los usuarios autorizados visualizar la lista de empresas asociadas a la app.
- Los usuarios pueden crear nuevas empresas asociadas, pero no pueden borrarlas.
- Los usuarios pueden crear un nuevo reporte, pero no pueden eliminarlos.

- En el módulo Reporte de inspección, el campo empresa solo cargara el listado de las empresas Asociadas
- Los campos de “Observaciones” aceptan caracteres alfabéticos únicamente.
- La aplicación debe poder utilizarse con los navegadores web Chrome, Firefox e Internet Explorer.

Requerimientos no Funcionales de la App móvil.

Un requisito no funcional o atributo de calidad es, un requisito que sabe bien y especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento, por eso suelen denominarse Atributos de calidad de un sistema. Queda entonces el requisito no funcional, que son las restricciones o condiciones que impone el cliente al programa que necesita. [18]

Para el sistema se muestran los siguientes requerimientos funcionales:

- La App debe ser capaz de guardar N cantidad de datos.
- Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.
- Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
- El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.
- El sistema debe contar con manuales de usuario estructurados adecuadamente.
- El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final
- El sistema debe poseer interfaces gráficas bien formadas
- El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente accederlo.
- La probabilidad de falla del Sistema no podrá ser mayor a 0,05.

5.3 Fase de Diseño

5.3.1 Arquitectura de la App.

La arquitectura de la aplicación sirve para proyectar como es su funcionamiento por ejemplo la arquitectura cliente/servidor funciona en 2 capas • Front/End • Back/End que es la arquitectura que más se muestra ya que del lado de la capa Front/End se muestra todo lo que el usuario puede apreciar visualmente del sistema y en la capa de Back/End nos describe como es el funcionamiento interno del sistema el que se está trabajando.

En la figura 14 se muestra como está desarrollada la arquitectura del sistema integral.

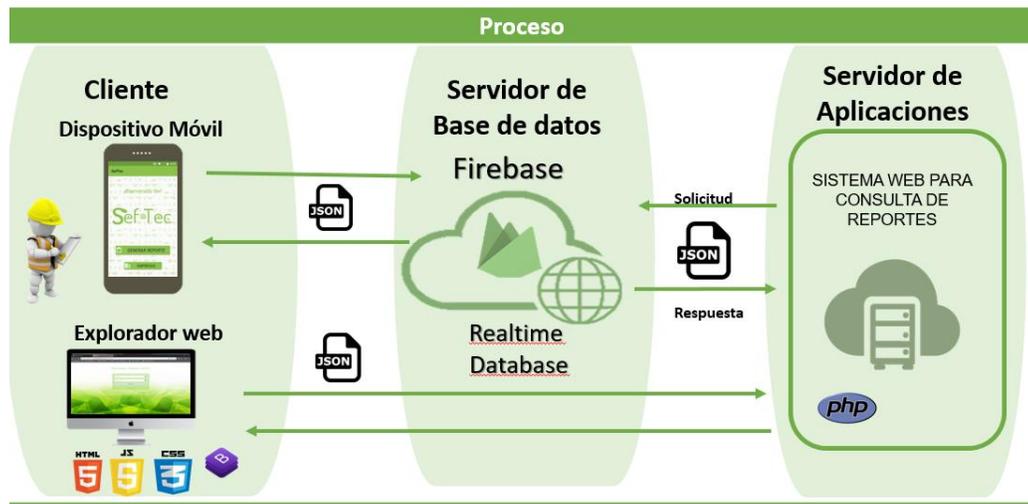


Figura 14. Arquitectura de la app Fuente: Edición Propia

5.3.2 Diagrama de casos de uso

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso). Un diagrama de casos de uso consta de los siguientes elementos:

- Actor.
- Casos de Uso.
- Relaciones de Uso, Herencia y Comunicación.

Un **Actor** es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra **rol**, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

El **caso de uso** es una operación o tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso. [19]

Para la investigación y desarrollo del sistema solo habrá 2 actores, el primero que es el que cumple el rol de utilizar la aplicación móvil y el segundo que es el que se encarga de utilizar la aplicación web ver figura 15.

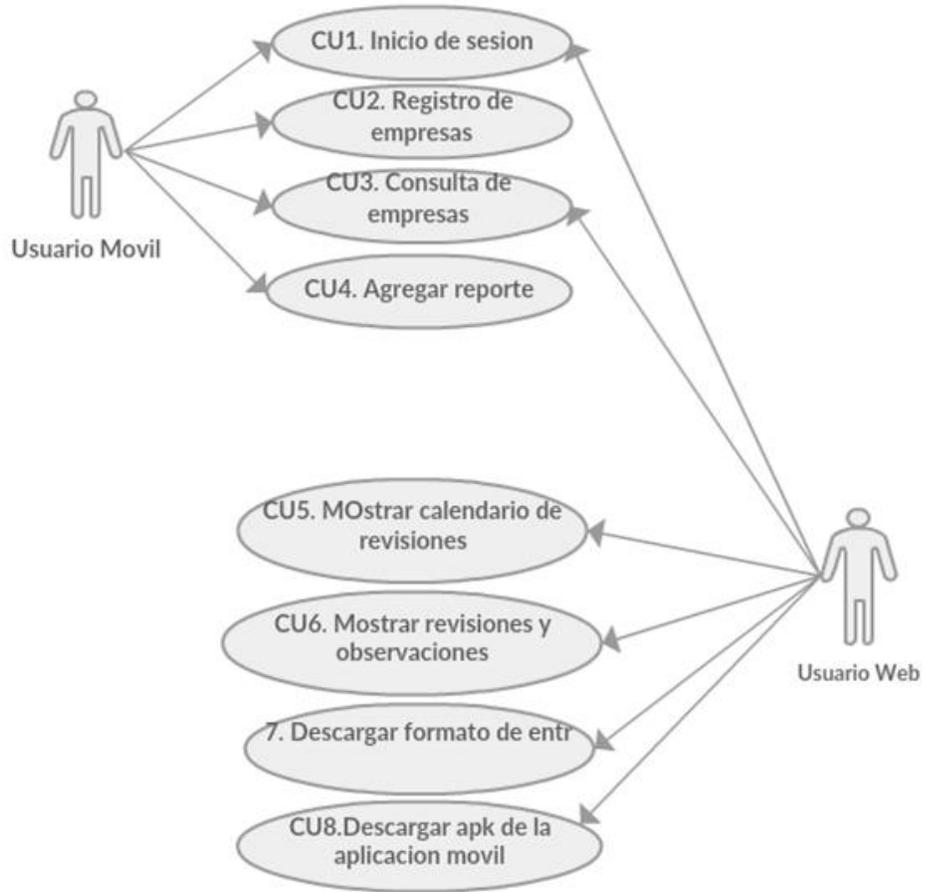


Figura 15. Diagrama de caso de uso Fuente: Edición Propia

5.3.3 Diagrama de clases

Los diagramas de clase describen los tipos de objetos de un sistema, así como los distintos tipos de relaciones que pueden existir entre ellos. Los diagramas de clase se convierten así en la técnica más potente para el modelado conceptual de un sistema software, la cual suele recoger los conceptos clave del modelo de objetos subyacente al método orientado a objetos que la incorpora, en este caso UML 1.1[20]

En la figura 16 se muestra el diagrama UML del proyecto.

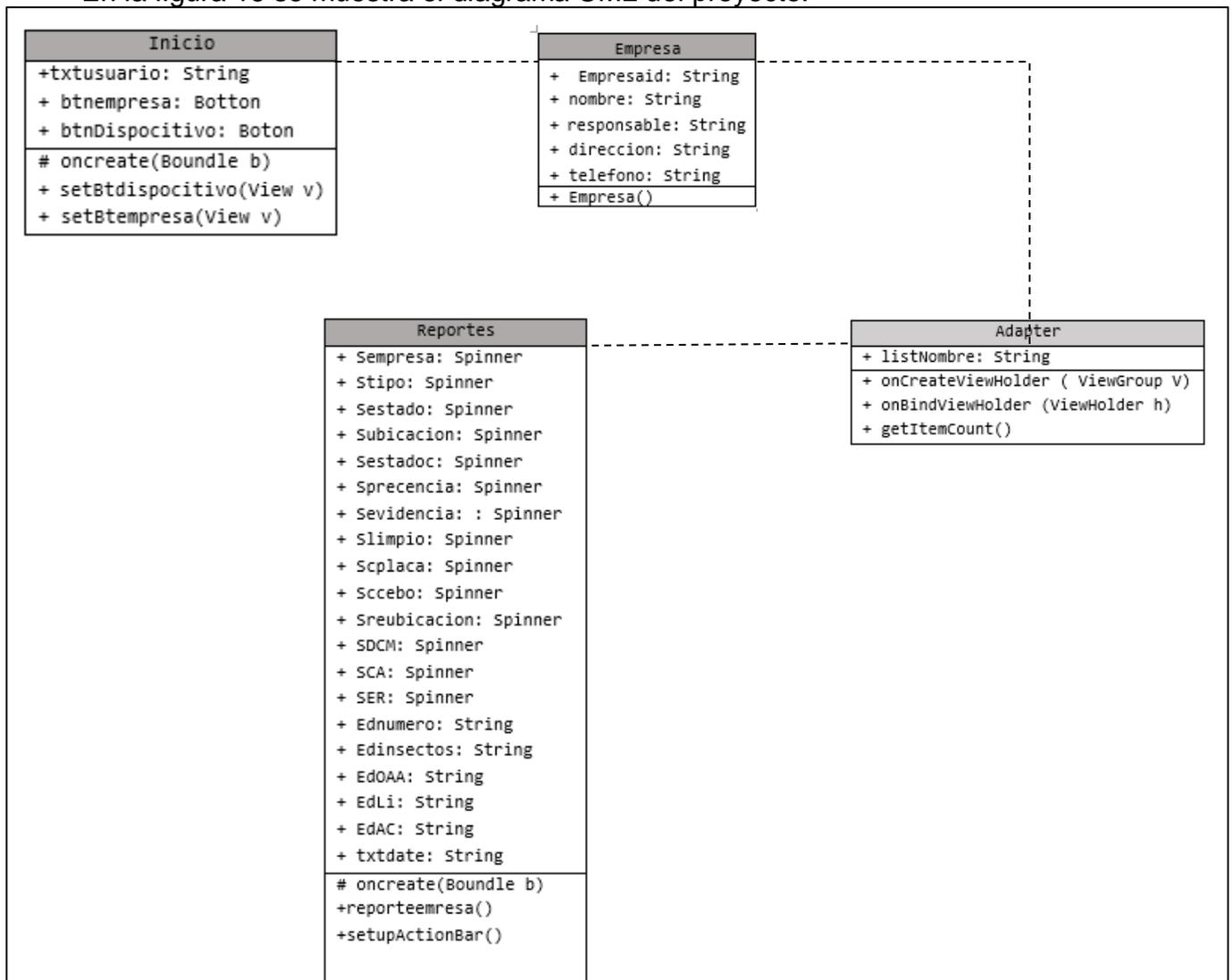


Figura 16. Diagrama UML. Fuente: edición propia.

5.3.4 Diseño de vistas de la aplicación móvil y wireframe de la web

Diseño de las vistas de la Aplicación móvil.

Se realizó el bosquejo de los 4 primeros módulos de la app basándose en las ideas y acuerdos, incluyendo en el diseño los colores que identifican a la empresa.

El primer módulo ver figura 17, hace referencia al inicio de sesión, en él se agregó el logotipo de la empresa (SEFICA) seguido de los apartados **Usuario/Contraseña** al cual solo tendrán acceso 2 usuarios previamente definidos y recalando que no se incluirá la opción de registrarse a petición del gerente para así poder tener el control de quien puede tener acceso a la app por ser de índole privado



Figura 17. Pantallas de aplicación Inicio de sesión Fuente: Elaboración Propia

El segundo módulo realizado es el principal, en él se incluye un mensaje de bienvenida con el nombre de la aplicación seguido de la imagen del usuario que accede a la app, seguido de ello estarán los apartados de “Empresa” y “Dispositivos de control” ver figura 18.



Figura 18. Pantalla principal de la aplicación Fuente: Elaboración Propia

En el apartado de Empresas se incluyen las empresas asociadas que llevan su control de plagas urbanas y en Dispositivos de control se lleva un listado de los dispositivos con los que cuenta la empresa para prestar sus servicios ver figura 19 y 20.

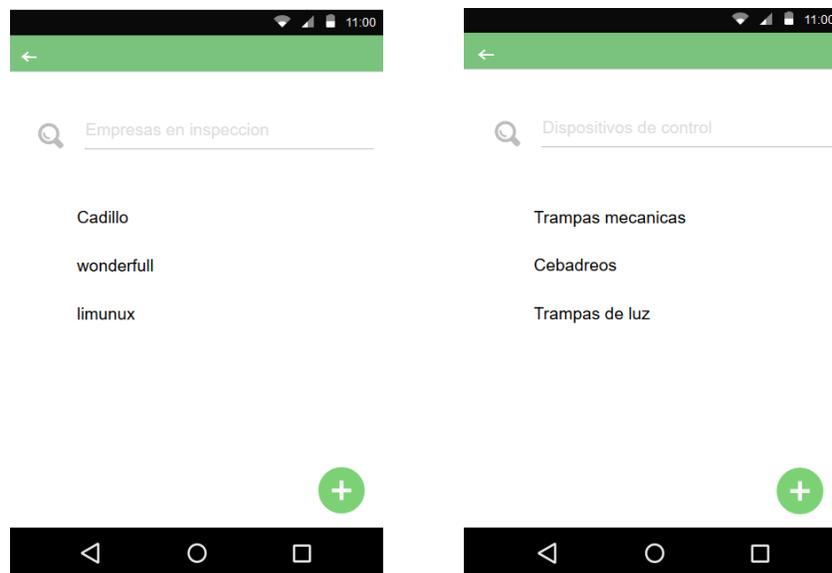


Figura 19 y 20. Pantallas para selección de la empresa y dispositivos. Fuente: Elaboración propia.

En el tercer módulo seguido de hacer clic en la opción de empresas, incluye un listado de las empresas asociadas y podrá ser filtrado por la barra de búsqueda, dentro del módulo se incluirá un botón para poder seguir agregando ya que la

empresa está en constante vinculación con para poder ofrecer sus servicios a nuevas empresas.

El cuarto módulo es el contenido de la opción **Dispositivos de control** al igual que el anterior incluirá una barra de búsqueda el cual nos permitirá buscar directamente una empresa que este en la lista, también tendrá un botón donde se podrán ir agregando más dispositivos ya que la empresa no se limita a actualizaciones y la inclusión de nuevas técnicas para el control de plagas y esto incluye dispositivos más avanzados y nuevos que en el futuro se podrían implementar al momento de prestar sus servicios.

La fase de Diseño de la app concluyo como resultado final lo siguiente:

El producto final del diseño consta de 6 pantallas.

Splash: Se definió un *Activity* de inicio que es la que mostrara el emblema de la aplicación para poder así darle un mejor tiempo de respuesta a la aplicación. Para poder agregar el emblema distintivo de la app se tomó en cuenta 2 factores: El nombre y la acción que se está realizando, así que se decidió utilizar un escudo ya que los dispositivos de los cuales se llevara el registro son los encargados de proteger a las empresas de todo tipo de plagas. Y dentro del escudo la S del SEFICA que es la empresa para que se estará desarrollando la app.

LogIn: Quedo establecido que la manera de acceso al contenido principal de la aplicación será solo se está registrado desde la consola de Firebase ya que la app no tendrá una opción para realizar el registro de usuarios por cuestiones de seguridad previamente establecidas.

Inicio: En esta pantalla se mostrará un mensaje de Bienvenida seguido del nombre del usuario que será tomado del nombre de usuario del correo, también el servidor de correo será creado solo para el LogIn del usuario ya que este no podrá ser utilizado para otras actividades relacionadas con el correo. Seguido de ello se muestra el nombre/logotipo de la aplicación “Sef-Tec” que es la conjugación de las dos palabras que asocian la app, SEFICA que es la empresa que implementará la aplicación y Tec de tecnología, ya que se hará un salto tecnología al pasar de utilizar

lápiz y papel a un teléfono móvil para su realizar todo su proceso de recolección de datos. Esta pantalla integrara 2 botones el primero que nos enviara a la pantalla de reportes y el segundo a la pantalla de mostrar empresas.

Mostar empresa: En esta pantalla se mostrará la lista y una barra de búsqueda de todas las empresas que están asociadas con SEFICA para realizar las revisiones de sus dispositivos de captura, esta también incluye un *Floating Botton* que nos enviará a nuestra quinta pantalla que será “Agregar empresa”.

Agregar empresa: En esta pantalla se defino la información específica de cada empresa que se debe de tener para poder así al momento de realizar los reportes seleccionar la empresa donde se están realizando, dicha información incluye: Nombre, Responsable de la inspección, Dirección de la empresa y numero.

Reporte de inspección: Esta última pantalla incluirá toda la información del check list que antes se llenaba a mano, en el apartado de Nombre de la empresa se programará un *spinner* en el que se descargará la lista de empresas previamente agregadas. Y así poder llenar toda la información de acuerdo con la empresa.

Aplicación web

Esta aplicación se busca desarrollar para que el jefe del departamento pueda revisar la información capturada en las revisiones y a su vez poder realizar los reportes para poder enviárselos a los responsables de cada empresa para el LogIn de la aplicación web se utiliza una hoja de estilos Bootstrap que son estilos css predefinidos utilizados para poder realizar toda la parte de diseño y la manera en la que se obtiene la conexión a la base de datos es mediante php creando un archivo que incluye la URL que da acceso a la base de datos y así poder hacer uso de una de las herramientas de Firebase, *FirebaseAuth*.

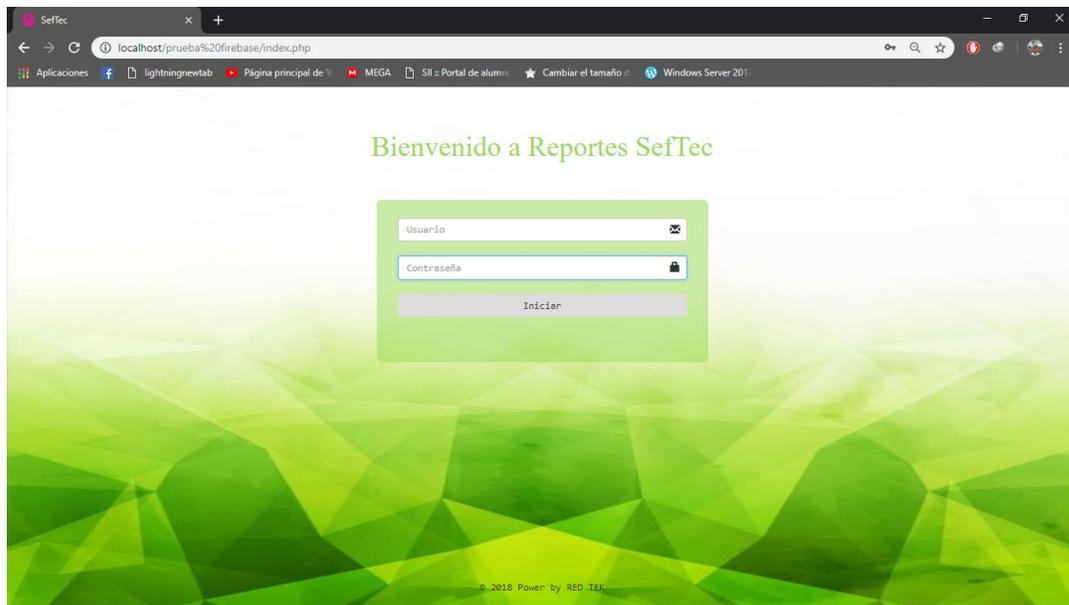


Figura 21. LogIn de la aplicación web. Fuente: Elaboración propia.

Una de las inclusiones de la aplicación es la implementación de la API de Google Calendar que muestra los eventos guardados en la agenda del usuario ofreciendo la posibilidad de poder sincronizarse con la aplicación móvil de Google Calendar ver figura 22.

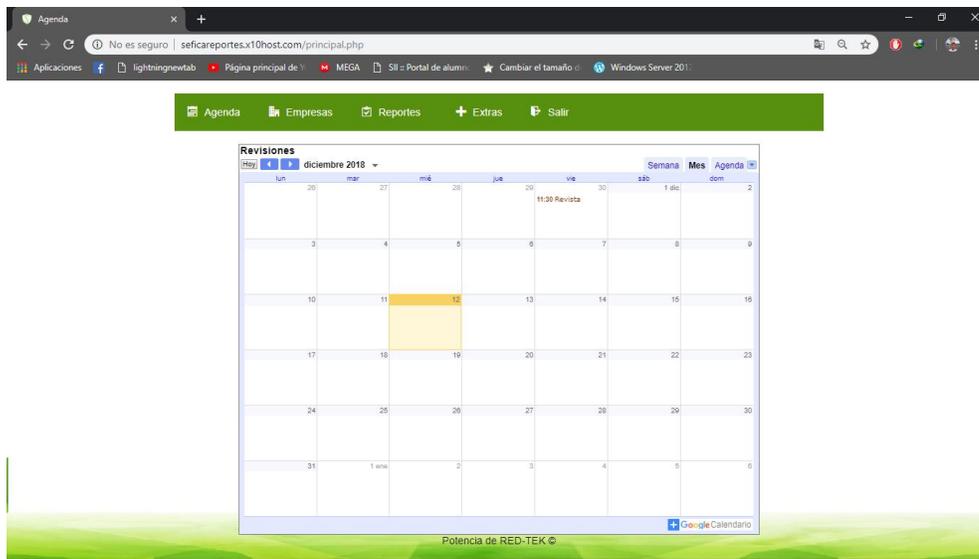


Figura 22. Implementación de Google Calendar. Fuente: Elaboración propia.

Para poder mostrar la información en forma de una tabla se implementó DataTables, un plug de JQuery este plug sirve de gran ayuda para poder darle un formato y estilo a todos los elementos que contiene una tabla dinámica también incluyendo un scrib que sirve como apoyo al momento de filtrar la información. Este plug también permite añadir una selección de botones que permitan hacer diferentes acciones, imprimir PDF, copiar información de la tabla y descargar Excel e vcf.

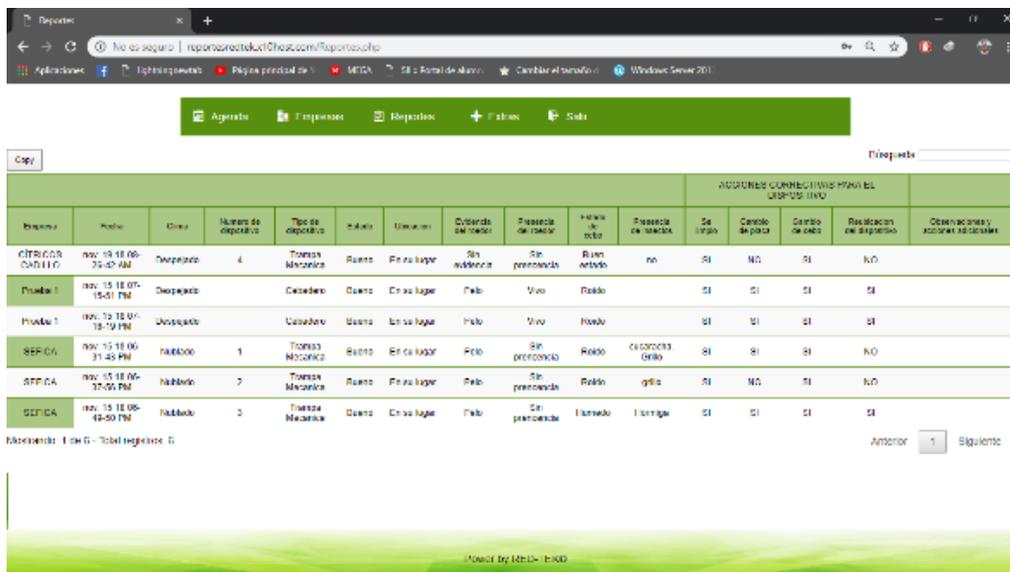


Figura 23. Implementación del plug DataTables by JQuery. Fuente: Elaboración propia.

En el apartado de extras se agregaron 2 botones de descarga uno es la plantilla con el cual el usuario envia la informacion en forma de reporte y otro el es para descargar el APK de la aplicación movil ya que de esta manera solo los usuarios registrados podran descargarla y de igual forma ahorrar el costo del servicio de Google play y asi concervar el carácter privado de nuestra aplicación ver figura 24.

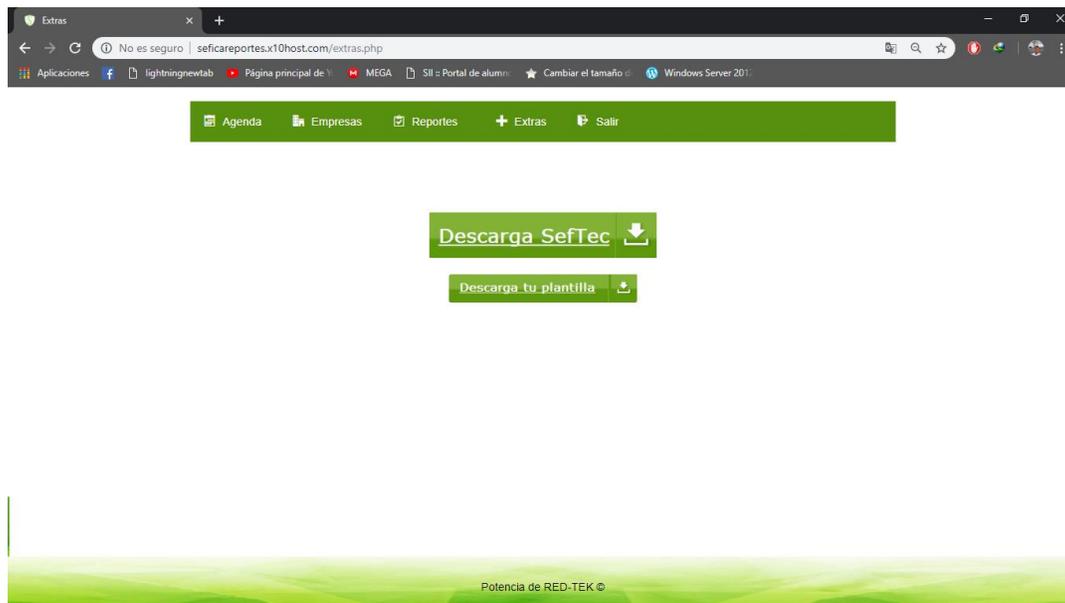


Figura 24. Panta del módulo “Extras”. Fuente: Elaboración propia.

5.2 Fase de Codificación

5.2.1 Desarrollo de base de datos

La base de datos que se implementó para el desarrollo del sistema no es una base de datos relacional, también llamada NoSQL. La base de datos puede conceptualizarse como un árbol JSON alojado en la nube. A diferencia de una base de datos de SQL, no hay tablas ni registros.

Cuando se agregan datos al árbol JSON, la información se convierte en un nodo de la estructura JSON existente al cual se le asigna una clave de identificación, como un ID de usuario o nombres semánticos.

Para la creación del árbol JSON del sistema se tomó en cuenta para el nodo principal nombrarlo “Empresas” tomando como referencia que las empresas son el organismo al cual se le realizarán las revisiones, como nodo hijo se nombró “Datos de la empresa” en el que se recolectaran los datos de las empresas (Nombre, Dirección, Responsable y teléfono)

Seguido de ello se definió el siguiente nodo hijo, que contiene toda la información que se requiere para llenar los datos de los reportes de cada revisión por dispositivo al cual se le asignó el nombre de “Reportes”, concluyendo con el JSON como se muestra en la figura 25.

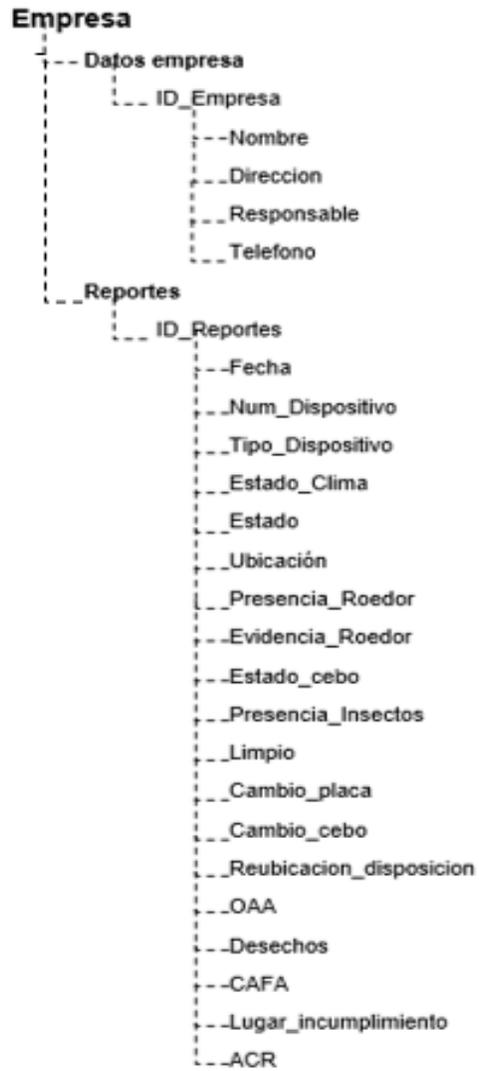


Figura 25. rbol JSON, Fuente: Edici3n propia.

5.2.2 Codificación de la aplicación móvil.

En esta etapa se inició la codificación de la app en el IDE de desarrollo Android Studio en la versión 3.0.

Primero se inició con la creación del proyecto llamado *SefTec* seguido de ello se seleccionó la API 27 que va de acuerdo a los sistemas operativos que tiene los equipos de la empresa.

Splash

Para la creación de la app se inició con una *Empty Activity* la cual se nombró splash ya que de manera lineal al momento de la programación. Agregándole en el archivo .xml el emblema y el fondo para poder así cumplir con la primera pantalla, en la clase *main* de splash se programó de la manera siguiente: Ver figura 26, 27 y 28.

```
1 package comsefica1.example.fernadocarmona.proyectoseftec;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.os.Handler;
7
8
9
10 public class SPLASH extends Activity{
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_splash);
15
16         new Handler().postDelayed(new Runnable() {
17             @Override
18             public void run() {
19                 // si creo un intent que nos mandara a la siguiente activity que sera la de LogIn
20                 Intent intencion = new Intent(SPLASH.this,LOGIN.class);
21                 startActivity(intencion);
22             }
23         },2000);
24         // Asignacion de la duracion en la que se mostrara el activity(en mili-segundos)
25     }
26 }
27
28 @Override
29 protected void onPause() {
30     super.onPause();
31     finish();
32 }
33 }
34
```

Figura 26. Código Java Splash. Fuente: Elaboración Propia.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".SPLASH"
8     android:background="@drawable/fd">
9
10
11     <ImageView
12         android:layout_width="match_parent"
13         android:layout_height="match_parent"
14         android:src="@drawable/logo1"/>
15
16 </android.support.constraint.ConstraintLayout>
17

```

Figura 27. Código XML Splash. Fuente: Elaboración Propia.



Figura 28. Resultado Final. Fuente: Elaboración Propia.

LogIn

Después de terminar con la creación la pantalla *splash*, se continuó con la programación de la *activity LogIn* que es donde el usuario escribe sus datos, lo primero que se hizo fue agregar el *imageView* donde se cargaría el logotipo de la empresa Sefica, después los dos *EditTex* siendo el segundo con el tipo de entrada *password* para que al momento de escribir la contraseña aparezcan los caracteres ocultos y por último el botón agregar que es el que nos mandara una vez validada la información a el *activity* inicio.

En el *ImageView* está insertado un imagen con extensión *.GIF* y para ello se tuvo que agregar la *dependencia implementation 'pl,droidsonroids.gif:Android-gif.drawable:1.2.7'* al Gradle.

Para poder implementar Firebase en la app, se dirigen desde el navegador a Firebase iniciando sesión por medio de una cuenta *google* y se crea un nuevo proyecto llamado Sefica, para así después vincularlo desde el IDE. Una vez ya creado el proyecto se dirigen a Android Studio y se vincula el proyecto de Android con Firebase para ello se necesita iniciar sesión desde el propio IDE y seguido de ello se debe de entrar a *Tool>Firebase* y una vez seleccionado el Asistente dar clic a *Authentication* y conectarse a la app a de Firebase para después agregar las dependencias. La activity *LogIn* quedó como se presenta en la figura 29, 30,31 y 32.

```
19 public class LOGIN extends AppCompatActivity implements View.OnClickListener{
20
21     //Definicion de los objetos que utilizaremos
22     private EditText TextEmail;
23     private EditText TextPassword;
24     private Button btnRegistrar, btnLogin;
25     private ProgressDialog progressDialog;
26
27     //Declaramos un objeto firebaseAuth
28     private FirebaseAuth firebaseAuth;
29
30     @Override
31     protected void onCreate(Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33         setContentView(R.layout.activity_login);
34
43         btnLogin = (Button) findViewById(R.id.botonLogin);
44         progressDialog = new ProgressDialog(this);
45
46         //asociamos un listener al evento clic del botón
47         btnLogin.setOnClickListener(this);
48     }
49
50     private void loguearUsuario() {
51         //Obtenemos el email y la contraseña desde las cajas de texto
52         final String email = TextEmail.getText().toString().trim();
53         String password = TextPassword.getText().toString().trim();
54
55         //Validaciones
56         if (TextUtils.isEmpty(email)) {
57             Toast.makeText(this, "Ingresar un email", Toast.LENGTH_LONG).show();
58             return;
59         }
60
61         if (TextUtils.isEmpty(password)) {
62             Toast.makeText(this, "Ingresa la contraseña", Toast.LENGTH_LONG).show();
63             return;
64         }
65     }
66
67     progressDialog.setMessage("Buscando en la base de datos...");
68     progressDialog.show();
69
70     //loguear usuario
71     firebaseAuth.signInWithEmailAndPassword(email, password)
72         .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
73         @Override
74         public void onComplete(@NonNull Task<AuthResult> task) {
75             // IF para verificar y enviar el nombre del usuario
76             if (task.isSuccessful()) {
77                 int pos = email.indexOf("@");
78                 String user = email.substring(0, pos);
79                 Toast.makeText(LOGIN.this, "Bienvenido a SefTec", Toast.LENGTH_LONG).show();
80                 Intent intencion = new Intent(getApplicationContext(), INICIO.class);
81                 intencion.putExtra(INICIO.user, user);
82                 startActivity(intencion);

```

Figura 29. Código Java LogIn parte1. Fuente: Elaboración propia.

```

82         startActivity(intencion);
83     } else {
84         Toast.makeText(LOGIN.this, "Usuario no encontrado ", Toast.LENGTH_LONG).show();
85     }
86     progressDialog.dismiss();
87 }
88 ));
89 }
90 }
91
92 @Override
93 public void onClick(View view) {
94     switch (view.getId()) {
95         case R.id.botonLogin:
96             loguearUsuario();
97             break;
98     }
99 }
100 }
101

```

Figura 30. Código Java LogIn parte 2. Fuente: Elaboración propia.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".LOGIN"
8     android:background="@drawable/fd">
9
10     <pl.droidsonroids.gif.GifImageView
11         android:id="@+id/gifImageView"
12         android:layout_width="331dp"
13         android:layout_height="198dp"
14         android:layout_marginBottom="8dp"
15         android:layout_marginEnd="8dp"
16         android:layout_marginStart="8dp"
17         android:layout_marginTop="8dp"
18         android:src="@drawable/cargar"
19         app:layout_constraintBottom_toTopOf="@+id/btempresas"
20         app:layout_constraintEnd_toEndOf="parent"
21         app:layout_constraintStart_toStartOf="parent"
22         app:layout_constraintTop_toTopOf="parent"
23         app:layout_constraintVertical_bias="0.547" />
24
25     <EditText
26         android:id="@+id/TxtEmail"
27         android:layout_width="316dp"
28         android:layout_height="wrap_content"
29         android:layout_margin="15dp"
30         android:layout_marginEnd="8dp"
31         android:layout_marginStart="8dp"
32         android:layout_marginTop="8dp"
33         android:drawableLeft="@drawable/ic_correo"
34         android:hint="@string/enter_email"
35         android:inputType="textEmailAddress"
36         app:layout_constraintEnd_toEndOf="parent"
37         app:layout_constraintStart_toStartOf="parent"
38         app:layout_constraintTop_toBottomOf="@+id/gifImageView" />
39

```

Figura 31. Código XML LogIn. Fuente: Elaboración propia.



Figura 32. Resultado Final. Fuente: Elaboración Propia.

Inicio

Para la tercer *activity* que es la de inicio se implementó primero el mensaje de bienvenida que añadirá el nombre del usuario que manda a llamar desde la información de correo seguido de ello inserta un *imageView* para poder mostrar el logotipo de la empresa, para después añadir los 2 botones que envían a las pantallas 4 y 6 Ver figura 33.

El primer botón Generar reporte que enviara a la pantalla donde llenar la información más importante y el segundo es el que envía a la pantalla 4 que es Empresas asociadas en el cual se mostrara la lista de las empresas asociadas y un botón flotante donde se podrá dar clic para pasar al formulario de agregar nuevas empresas.



Figura 33. Resultado Final. Fuente: Elaboración propia.

```

1 package comsefical.example.fernadocarmona.proyectoseftec;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.Button;
8 import android.widget.TextView;
9
10 public class INICIO extends AppCompatActivity {
11     public static final String user="names";
12     TextView txtUser;
13     Button btempresa;
14     Button btadispositivo;
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_inicio);
20
21
22         txtUser =(TextView)findViewById(R.id.textser);
23         btempresa=(Button)findViewById(R.id.btempresas);
24         btadispositivo=(Button)findViewById(R.id.btadispositivos);
25
26         String user = getIntent().getStringExtra("names");
27         txtUser.setText("¡Bienvenido "+ user +"!");
28
29     }
30     public void setBtdispositivo (View vista){
31         Intent intent= new Intent (INICIO.this, Reportes.class);
32         startActivity(intent);
33     }
34
35     public void setBtempresa (View vista){
36         Intent intent= new Intent (INICIO.this, M_EMPRESAS.class);
37         startActivity(intent);
38     }
39 }
40

```

Figura 34. Código Java Inicio. Fuente: Elaboración propia.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".INICIO"
9     android:background="@drawable/fd">
10
11     <TextView
12         android:id="@+id/textser"
13         android:layout_width="wrap_content"
14         android:layout_height="wrap_content"
15         android:layout_marginBottom="8dp"
16         android:layout_marginTop="8dp"
17         android:textColor="#8bc34a"
18         android:textSize="38sp"
19         android:textStyle="italic"
20         app:layout_constraintBottom_toBottomOf="parent"
21         app:layout_constraintEnd_toEndOf="parent"
22         app:layout_constraintLeft_toLeftOf="parent"
23         app:layout_constraintRight_toRightOf="parent"
24         app:layout_constraintStart_toStartOf="parent"
25         app:layout_constraintTop_toTopOf="parent"
26         app:layout_constraintVertical_bias="0.107" />
27
28     <Button
29         android:id="@+id/btempresas"
30         android:layout_width="197dp"
31         android:layout_height="41dp"
32         android:layout_margin="15dp"
33         android:layout_marginBottom="16dp"
34         android:layout_marginEnd="8dp"
35         android:layout_marginStart="8dp"
36         android:layout_marginTop="8dp"
37         android:background="@drawable/boton_inicio"
38         android:fontFamily="sans-serif-condensed"
39         android:drawableLeft="@drawable/ic_empresa"
40         android:onClick="setBtempresa"
41         android:text="@string/empresa"
42         android:textAllCaps="false"
43         android:textColor="#323232"
44         android:textStyle="italic"
45         android:textSize="22sp"
46         app:layout_constraintBottom_toBottomOf="parent"
47         app:layout_constraintEnd_toEndOf="parent"
48         app:layout_constraintHorizontal_bias="0.584"
49         app:layout_constraintStart_toStartOf="parent"
50         app:layout_constraintTop_toBottomOf="@+id/btdispositivos" />

```

```

52 <Button
53     android:id="@+id/btdispositivos"
54     android:layout_width="284dp"
55     android:layout_height="58dp"
56     android:layout_margin="15dp"
57     android:layout_marginBottom="8dp"
58     android:layout_marginEnd="8dp"
59     android:layout_marginStart="8dp"
60     android:layout_marginTop="8dp"
61     android:background="@drawable/boton_inicio"
62     android:fontFamily="sans-serif-condensed"
63     android:onClick="setBtdispositivo"
64     android:text="@string/greporte"
65     android:drawableLeft="@drawable/ic_reporte"
66     android:textStyle="italic"
67     android:textAllCaps="false"
68     android:textColor="#323232"
69     android:textSize="26sp"
70     app:layout_constraintBottom_toBottomOf="parent"
71     app:layout_constraintEnd_toEndOf="parent"
72     app:layout_constraintHorizontal_bias="0.584"
73     app:layout_constraintStart_toStartOf="parent"
74     app:layout_constraintTop_toTopOf="parent"
75     app:layout_constraintVertical_bias="0.776" />
76
77 <ImageView
78     android:layout_width="match_parent"
79     android:layout_height="399dp"
80     android:layout_marginBottom="108dp"
81     android:layout_marginEnd="8dp"
82     android:layout_marginStart="8dp"
83     android:layout_marginTop="8dp"
84     android:src="@drawable/seftec"
85     app:layout_constraintBottom_toBottomOf="parent"
86     app:layout_constraintEnd_toEndOf="parent"
87     app:layout_constraintStart_toStartOf="parent"
88     app:layout_constraintTop_toTopOf="parent" />
91 </android.support.constraint.ConstraintLayout>

```

Figura 35. Código XML Inicio 1.0. Fuente: Elaboración propia.

Agregar Empresa

En este módulo se crearon los 5 campos que se tiene que llenar con la información precisa de la empresa ya que esta es necesaria para poder seleccionar de que empresa vamos a realizar la inspección. Se le añadió al botón de guardado la acción que al instante de pulsarlo nos envíe a la *activity* de mostrar empresa, para poder visualizar el catálogo de las empresas añadidas y así continuar. ver figura 36.



The screenshot shows a mobile application interface for adding a new company. The title bar is green and contains a back arrow and the text "Agrega una nueva empresa". Below the title bar, the text "Ingresa los datos de la empresa" is displayed. There are four input fields: "Nombre de la empresa" (with "Fernando Carmona" entered), "responsable de la inspeccion", "Direccion", and "Telefono". At the bottom, there is a green button labeled "GUARDAR".

Figura 36. Pantalla agregar nueva empresa. Fuente: Elaboración propia.

```

19 public class N_EMPRESAS extends AppCompatActivity {
20     EditText Nombre_empresa, Res_Empresa, Dirc_Empresa, Tel_empresa;
21     Button Guardar;
22     private DatabaseReference EMPRESAS;
23
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_empresas);
28         setupActionBar();
29         EMPRESAS = FirebaseDatabase.getInstance().getReference("EMPRESAS");
30
31         Nombre_empresa = (EditText) findViewById(R.id.Nempresa);
32         Res_Empresa = (EditText) findViewById(R.id.Rempresa);
33         Dirc_Empresa = (EditText) findViewById(R.id.Dempresa);
34         Tel_empresa = (EditText) findViewById(R.id.Tempresa);
35         Guardar = (Button) findViewById(R.id.btnguardar);
36
37         Guardar.setOnClickListener(new View.OnClickListener() {
38             @Override
39             public void onClick(View v) {
40                 guardarEmpresa();
41             }
42         });
43
44     }
45
46     private void guardarEmpresa() {
47         String nombre = Nombre_empresa.getText().toString();
48         String responsabe = Res_Empresa.getText().toString();
49         String direccion = Dirc_Empresa.getText().toString();
50         String telefono = Tel_empresa.getText().toString();
51
52
53         if (!TextUtils.isEmpty(telefono)) {
54             String id = EMPRESAS.push().getKey();
55             EMPRESAS.datos = new EMPRESAS(id, nombre, responsabe, direccion, telefono);
56             EMPRESAS.child("Datos de la Empresa").child("Id: "+id).setValue(datos);
57
58             Toast.makeText(this, "Empresa añadida", Toast.LENGTH_LONG).show();
59             Intent intent = new Intent(N_EMPRESAS.this, M_EMPRESAS.class);
60             startActivity(intent);
61         } if (TextUtils.isEmpty(nombre)) {
62             Toast.makeText(this, "Debes ingresar un nombre", Toast.LENGTH_LONG).show();
63             return;
64         } if (TextUtils.isEmpty(responsabe)) {
65             Toast.makeText(this, "ingresa el responsable", Toast.LENGTH_LONG).show();
66             return;
67         } if (TextUtils.isEmpty(direccion)) {
68             Toast.makeText(this, "Ingresa la contraseña", Toast.LENGTH_LONG).show();
69             return;
70         } if (TextUtils.isEmpty(telefono)) {
71             Toast.makeText(this, "Debes de ingresar el telefono", Toast.LENGTH_LONG).show();
72             return;
73         }
74     }
75
76     private void setupActionBar() {
77         ActionBar actionBar = getSupportActionBar();
78         if (actionBar != null) {
79             actionBar.setDisplayHomeAsUpEnabled(true);
80             actionBar.setTitle("Agrega una nueva empresa");
81         }
82     }
83
84 }

```

Figura 37. Código java N_Empresa. Fuente: Elaboración propia.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".N_EMPRESAS"
9     android:background="@drawable/fd">
10
11     <TextView
12         android:id="@+id/textView"
13         android:layout_width="225dp"
14         android:layout_height="26dp"
15         android:layout_marginBottom="8dp"
16         android:layout_marginEnd="8dp"
17         android:layout_marginStart="8dp"
18         android:layout_marginTop="8dp"
19         android:text="@string/Datos_empresa"
20         android:textAlignment="center"
21         android:textColor="#329032"
22         android:textSize="18sp"
23         android:textStyle="bold"
24         app:layout_constraintBottom_toBottomOf="parent"
25         app:layout_constraintEnd_toEndOf="parent"
26         app:layout_constraintHorizontal_bias="0.506"
27         app:layout_constraintStart_toStartOf="parent"
28         app:layout_constraintTop_toTopOf="parent"
29         app:layout_constraintVertical_bias="0.838" />
30
31     <EditText
32         android:id="@+id/Nempresa"
33         android:layout_width="263dp"
34         android:layout_height="45dp"
35         android:layout_marginEnd="8dp"
36         android:layout_marginStart="8dp"
37         android:layout_marginTop="88dp"
38         android:hint="@string/Nempresa"
39         android:inputType="textPersonName"
40         app:layout_constraintEnd_toEndOf="parent"
41         app:layout_constraintStart_toStartOf="parent"
42         app:layout_constraintTop_toTopOf="parent" />
43
44     <EditText
45         android:id="@+id/Dempresa"
46         android:layout_width="263dp"
47         android:layout_height="45dp"
48         android:layout_marginEnd="8dp"
49         android:layout_marginStart="8dp"
50         android:layout_marginTop="24dp"
51         android:hint="@string/Rempresa"
52         android:inputType="textPersonName"
53         app:layout_constraintEnd_toEndOf="parent"
54         app:layout_constraintHorizontal_bias="0.503"
55         app:layout_constraintStart_toStartOf="parent"
56         app:layout_constraintTop_toBottomOf="@+id/Nempresa" />
57
58     <EditText
59         android:id="@+id/Dempresa"
60         android:layout_width="263dp"
61         android:layout_height="45dp"
62         android:layout_marginEnd="8dp"
63         android:layout_marginStart="8dp"
64         android:layout_marginTop="28dp"
65         android:hint="@string/Dempresa"
66         android:inputType="textPersonName"
67         app:layout_constraintEnd_toEndOf="parent"
68         app:layout_constraintHorizontal_bias="0.504"
69         app:layout_constraintStart_toStartOf="parent"
70         app:layout_constraintTop_toBottomOf="@+id/Rempresa" />
71
72     <EditText
73         android:id="@+id/Tempresa"
74         android:layout_width="263dp"
75         android:layout_height="42dp"
76         android:layout_marginEnd="8dp"
77         android:layout_marginStart="8dp"
78         android:layout_marginTop="44dp"
79         android:hint="@string/Tempresa"
80         android:inputType="phone"
81         android:maxLength="10"
82         app:layout_constraintEnd_toEndOf="parent"
83         app:layout_constraintHorizontal_bias="0.504"
84         app:layout_constraintStart_toStartOf="parent"
85         app:layout_constraintTop_toBottomOf="@+id/Dempresa" />
86
87     <Button
88         android:id="@+id/Bempresa"
89         android:layout_width="263dp"
90         android:layout_height="42dp"
91         android:layout_marginEnd="8dp"
92         android:layout_marginStart="8dp"
93         android:layout_marginTop="44dp"
94         android:hint="@string/Bempresa"
95         android:inputType="text"
96         android:maxLength="10"
97         app:layout_constraintEnd_toEndOf="parent"
98         app:layout_constraintHorizontal_bias="0.504"
99         app:layout_constraintStart_toStartOf="parent"
100        app:layout_constraintTop_toBottomOf="@+id/Tempresa" />

```

Figura 38. Código XML N: Empresa.

Mostar empresa

Este módulo se diseñó primero que el anterior pero solo de manera gráfica sin agregarle acciones, una vez concluido el módulo “Agregar empresa” se inició con la programación de acciones ya que este se muestra una lista con la información de las empresas agregadas.

Este módulo contara con un *recyclerView* el cual al conectarse con la base de datos mostrara la informacion más precisa de las empresas con el nombre, responsable y telefono. Ver figura 39.



Figura 39. Mostrar empresa. Fuente: Elaboración propia.

```

22
23 public class M_EMPRESAS extends AppCompatActivity {
24
25     RecyclerView rv;
26     List<EMPRESAS> empre;
27
28     Adapter adapter;
29     private DatabaseReference Rf1;
30
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.activity_m_empresas);
35         setupActionBar();
36
37         rv=findViewById(R.id.recycler);
38         rv.setLayoutManager(new LinearLayoutManager(this));
39
40         empre= new ArrayList<>();
41
42         Rf1=FirebaseDatabase.getInstance().getReference("EMPRESAS");
43
44         adapter=new Adapter(empre);
45
46         rv.setAdapter(adapter);
47         Rf1.child("DatosEmpresa").addValueEventListener(new ValueEventListener() {
48             @Override
49             public void onDataChange(DataSnapshot dataSnapshot) {
50                 for (DataSnapshot snapshot:
51                     dataSnapshot.getChildren()) {
52                     EMPRESAS empresaSnapshot= snapshot.getValue(EMPRESAS.class);
53                     empre.add(empresa);
54                 }
55                 adapter.notifyDataSetChanged();
56             }
57         });
58
59     @Override
60     public void onCancelled(DatabaseError databaseError) {
61
62     }
63
64 });
65
66
67
68 FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
69 fab.setOnClickListener(new View.OnClickListener() {
70     @Override
71     public void onClick(View view) {
72         Intent intent= new Intent (M_EMPRESAS.this, M_EMPRESAS.class);
73         startActivity(intent);
74     }
75 });
76
77 private void setupActionBar() {
78     ActionBar actionBar= getSupportActionBar();
79     if (actionBar !=null){
80         actionBar.setDisplayHomeAsUpEnabled(true);
81         actionBar.setTitle("Empresas Asociadas");
82     }
83 }
84 }

```

Figura 40. Código java M:Empresa. Fuente: Elaboración propia.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".M_EMPRESAS"
8     android:background="@drawable/fd">
9
10
11     <android.support.design.widget.FloatingActionButton
12         android:id="@+id/fab"
13         android:layout_width="wrap_content"
14         android:layout_height="wrap_content"
15         android:layout_alignParentBottom="true"
16         android:layout_alignParentEnd="true"
17         android:layout_gravity="bottom|end"
18         android:layout_marginBottom="11dp"
19         android:layout_marginEnd="21dp"
20         android:src="@drawable/ic_empresa" />
21
22
23     <android.support.v7.widget.RecyclerView
24         android:id="@+id/recycler"
25         android:layout_width="match_parent"
26         android:layout_height="match_parent"
27         android:layout_alignParentStart="true"
28         android:layout_alignParentTop="true"
29         tools:layout_editor_absoluteX="347dp"
30         tools:layout_editor_absoluteY="473dp" />
31 </RelativeLayout>

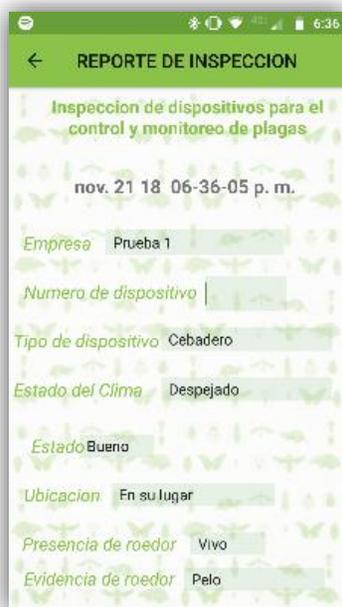
```

Figura 41. Código java M:Empresa. Fuente: Elaboración propia.

Añadir reporte.

En este último módulo ver figura 42 se creó todo el formulario donde se realiza el llenado de la información de los dispositivos la información se llena de 2 maneras por texto y *spinner*'s, cabe de destacar que es el módulo más importante de la aplicación ya que en él se guarda la información principal de la aplicación, la información de las inspecciones.

Una de las opciones que ofrece este módulo es que mediante un *spinner* se manda a llamar los nombres de las empresas para poder agregar cada reporte de los dispositivos de acuerdo a la empresa correspondiente. También mostrara la fecha y la hora en la cual se realizará el reporte, esta opción fue añadida para poder llevar un control de seguridad para que no se pueda agregar un reporte fuera de las fechas/horas destinadas para las inspecciones.



The screenshot shows a mobile application interface for reporting an inspection. The title bar is green and contains a back arrow and the text 'REPORTE DE INSPECCION'. Below the title bar, the main content area has a light green background with a pattern of small green leaves. The form consists of the following fields:

- Inspeccion de dispositivos para el control y monitoreo de plagas** (Title)
- nov. 21 18 06-36-05 p. m.** (Date and Time)
- Empresa** Prueba 1 (Text input)
- Numero de dispositivo** (Text input)
- Tipo de dispositivo** Cebadero (Text input)
- Estado del Clima** Despejado (Text input)
- Estado** Bueno (Text input)
- Ubicacion** En su lugar (Text input)
- Presencia de roedor** Vivo (Text input)
- Evidencia de roedor** Pelo (Text input)

Figura 42. Reporte de inspección Fuente: Edición propias.

```

25 import java.util.List;
26
27 public class Reportes extends AppCompatActivity {
28     private DatabaseReference NombreEmpresa;
29     private DatabaseReference reporte;
30
31     Spinner Semp,STP,SE,SCLIMA,SU,SEC,Spres,SER,R2,R3,R4,R5,SDCM,SCA,SERA;
32     EditText ednumerod,edInsectos,edAAA,edLI,edACR;
33     Button btnReporte;
34     TextView date1;
35
36
37
38
39     @Override
40     protected void onCreate(Bundle savedInstanceState) {
41         super.onCreate(savedInstanceState);
42         setContentView(R.layout.activity_reportes);
43         setUpActionBar();
44         NombreEmpresa= FirebaseDatabase.getInstance().getReference("EMPRESAS");
45
46         Thread t= new Thread(){
47             @Override
48             public void run(){
49                 try {
50                     while (!isInterrupted()){
51                         Thread.sleep(1000);
52                         runOnUiThread(new Runnable() {
53                             @Override
54                             public void run() {
55                                 TextView tdate= (TextView)findViewById(R.id.date);
56                                 long date= System.currentTimeMillis();
57                                 SimpleDateFormat sdf=new SimpleDateFormat("MM dd yy hh-mm-ss a");
58                                 String dateString = sdf.format(date);
59                                 tdate.setText(dateString);
60                             }
61                         });
62                     }
63                 }catch (InterruptedException e){}
64             }
65         };
66         t.start();
67
68
69

```

```

70 //Spinner's
71 Semp=(Spinner)findViewById(R.id.Semp);
72 STP=(Spinner)findViewById(R.id.STP);
73 SCLIMA=(Spinner)findViewById(R.id.SCLIMA);
74 SE=(Spinner)findViewById(R.id.SE);
75 SU=(Spinner)findViewById(R.id.SU);
76 SEC=(Spinner)findViewById(R.id.SEC);
77 Spres=(Spinner)findViewById(R.id.Spres);
78 SER=(Spinner)findViewById(R.id.SER);
79 R2=(Spinner)findViewById(R.id.R2);
80 R3=(Spinner)findViewById(R.id.R3);
81 R4=(Spinner)findViewById(R.id.R4);
82 R5=(Spinner)findViewById(R.id.R5);
83 SDCM=(Spinner)findViewById(R.id.SDCM);
84 SCA=(Spinner)findViewById(R.id.SCA);
85 SERA=(Spinner)findViewById(R.id.SERA);
86
87 //EditText
88 ednumerod=(EditText)findViewById(R.id.ednumerod);
89 edInsectos=(EditText)findViewById(R.id.edInsectos);
90 edAAA=(EditText)findViewById(R.id.edAAA);
91 edLI=(EditText)findViewById(R.id.edLI);
92 edACR=(EditText)findViewById(R.id.edACR);
93
94 //Fecha
95 date1=(TextView) findViewById(R.id.date);
96
97
98
99 //Button
100 btnReporte=(Button)findViewById(R.id.btnReport);
101
102 btnReporte.setOnClickListener(new View.OnClickListener() {
103     @Override
104     public void onClick(View v) {
105         reporteEmpresa();
106     }
107 });
108
109
110 NombreEmpresa.child("DatosEmpresa").addValueEventListener(new ValueEventListener() {
111     @Override
112     public void onDataChange(DataSnapshot dataSnapshot) {
113         final List<String>nombre2= new ArrayList<String>();
114

```

Figura 43. Código java Reportes. Elaboración propia

5.2.3 Codificación del sistema Web

El sistema web es el complemento de la app móvil, está pensado para el gerente, para un manejo meramente de oficina.

Una vez iniciada sesión, se puede tener acceso a la agenda que se maneja en el dispositivo móvil que utiliza el usuario, también el gerente, que es el usuario para el que está pensada puede tener acceso a la lista de las empresas asociadas, los reportes y observaciones que se van agregando desde la aplicación, ya que está vinculada con Firebase, el administrador de la base de datos. Otro de los módulos muestra 2 botones, el primero tiene la etiqueta de “Descargar formato” el cual ofrece el formato de impresión para plasmar la información de las revisiones con forme sea necesario y el otro botón es para descargar el archivo APK de la aplicación móvil, por ser de índole privado no está en la Play Store.

Ya explicados a groso modo los 5 módulos a continuación a continuación se muestra el código y su resultado.

LogIn

El inicio de sesión es realizado por Firebase Auth en el código principal que se muestra en la figura [#] se hace un llamado al archivo de conexión. Php que se muestra en la figura 44 que es el que contiene la URL Uniform Resource Locator (Localizador Uniforme de Recursos por sus siglas en ingles) que permite la conexión de la base de datos en Firebase.

The image shows a code editor window titled 'conexion.php'. The code is as follows:

```
1 <script src="https://www.gstatic.com/firebasejs/5.5.5/firebase.js"></script>
2 <script>
3 // Inicializacion de Firebase
4 var config = {
5   apiKey: "AIzaSyBhox3EV1CxzXdRvIJD0gTJ9IJadhU1bFI", //----- Llave de acceso a la Base de datos
6   authDomain: "sefica-e73fd.firebaseio.com", //-----Dominio para poder autentificar el usuario
7   databaseURL: "https://sefica-e73fd.firebaseio.com", //----Url de la conexion a Firebase
8   projectId: "sefica-e73fd",
9   storageBucket: "sefica-e73fd.appspot.com",
10  messagingSenderId: "507809215190"
11 };
12 firebase.initializeApp(config);
13 </script>
```

Figura 44. Conexión a Firebase.

El estilo que se eligió para toda información hace referencia a todos los colores que se utilizaron en la aplicación a gusto del cliente ese incluye el icono de la aplicación

en las pestañas, un fondo con un degradado verde ya que el verde es el color definido por el cliente, ver figura 45.



Figura 45. Pantalla de inicio de sesión.

Como un tema de seguridad se establecieron reglas de seguridad, en caso de que el usuario no este validado se enviara un *Modal* que es una ventana emergente que notifique al usuario que este incorrecto, en la figura 46 se muestra el fragmento de código que se insertó en la página index que es la que se muestra en la figura 45.

```
50
51
52 <!-- Mensaje-->
53 <div id="errorEntrar" class="modal fade" role="dialog">
54   <div class="modal-dialog">
55
56     <!-- Contenido del modal de validacion de usuario-->
57     <div class="modal-content">
58       <div class="modal-header">
59         <button type="button" class="close" data-dismiss="modal">&times;</button>
60         <h4 class="modal-title">Usuario Incorrecto</h4>
61       </div>
62       <div class="modal-body">
63         <p>Vuelve a escribir usuario y/o contraseña de nuevo</p>
64       </div>
65       <div class="modal-footer">
66         <button type="button" class="btn btn-default" data-dismiss="modal">Cerrar</button>
67       </div>
68     </div>
69   </div>
70 </div>
```

Figura 46 Código de la implementación del Modal

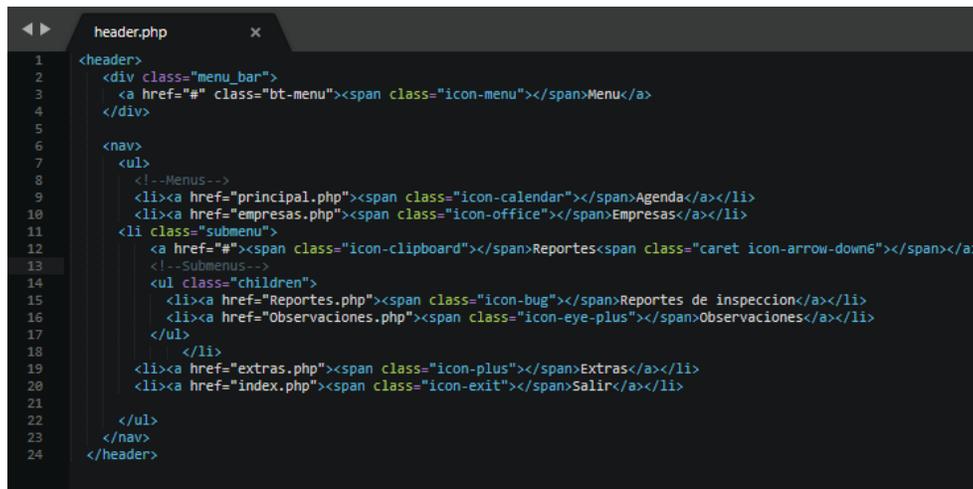
Una vez que el usuario es validado podrá acceder a la aplicación.

El menú por el que navegara el usuario en la aplicación web es un menú fijo el cual se le agrego a todos los módulos en el apartado de header haciendo una inclusión

por medio de una línea de código a un archivo .php mediante la siguiente línea de código:

```
<?php include('menu.php'); ?>
```

El menú que se está utilizando se programó mediante un archivo .php ver figura 47 al cual se le dio forma mediante una hoja de estilo CSS que se creó con base a los gustos antes mencionados. Otra de las funciones de este menú que este pudiera visualizarse a manera *responsive*, ya eso permitirá que los usuarios puedan acceder al sistema desde sus dispositivos móvil sin tener problemas de compatibilidad visual, para ello se programó un archivo JS(JavaScript) ver figura 48, en el cual mediante un evento clic nos permite expandir o mostrar los menús desde el dispositivo que estemos utilizando.



```
1 <header>
2   <div class="menu_bar">
3     <a href="#" class="bt-menu"><span class="icon-menu"></span>Menu</a>
4   </div>
5
6   <nav>
7     <ul>
8       <!--Menus-->
9       <li><a href="principal.php"><span class="icon-calendar"></span>Agenda</a></li>
10      <li><a href="empresas.php"><span class="icon-office"></span>Empresas</a></li>
11      <li class="submenu">
12        <a href="#"><span class="icon-clipboard"></span>Reportes<span class="caret icon-arrow-down6"></span></a>
13        <!--Submenus-->
14        <ul class="children">
15          <li><a href="Reportes.php"><span class="icon-bug"></span>Reportes de inspeccion</a></li>
16          <li><a href="Observaciones.php"><span class="icon-eye-plus"></span>Observaciones</a></li>
17        </ul>
18      </li>
19      <li><a href="extras.php"><span class="icon-plus"></span>Extras</a></li>
20      <li><a href="index.php"><span class="icon-exit"></span>Salir</a></li>
21    </ul>
22  </nav>
23 </header>
```

Figura 47. Código del archivo header.php

```
menu2.js x
1 $(document).ready(main);
2
3 var contador = 1;
4
5 function main () {
6     $('.menu_bar').click(function(){
7         if (contador == 1) {
8             $('nav').animate({
9                 left: '0'
10            });
11            contador = 0;
12        } else {
13            contador = 1;
14            $('nav').animate({
15                left: '-100%'
16            });
17        }
18    });
19
20    // Mostramos y ocultamos submenus
21    $('.submenu').click(function(){
22        $(this).children('.children').slideToggle();
23    });
24 }
```

Figura 48. Código del archivo menu.js

Agenda

Como se tenía planificado, la codificación del módulo *agenda* incluye la implementación de google calendar en sincronización con Calendar de google que es la aplicación que se tiene en los Smartphone de hoy en día.

Para poder integrar el calendario a la app web fue necesario obtener el código del mismo el cual se obtiene ingresando desde un navegador a Google calendar y ya una vez creado y seleccionado un calendario se dirigirán a configuración en el apartado de implantación en donde se mostrara información como en la figura 49, a la cual para editar y seleccionar formato se debe de dar clic en personalizar lo cual te dirigirá a otra pantalla ver figura 50, donde ofrece la URL que se agregó al archivo Agenda. php como se muestra en la figura 51 y en la figura 52 podremos apreciar el resultado final.

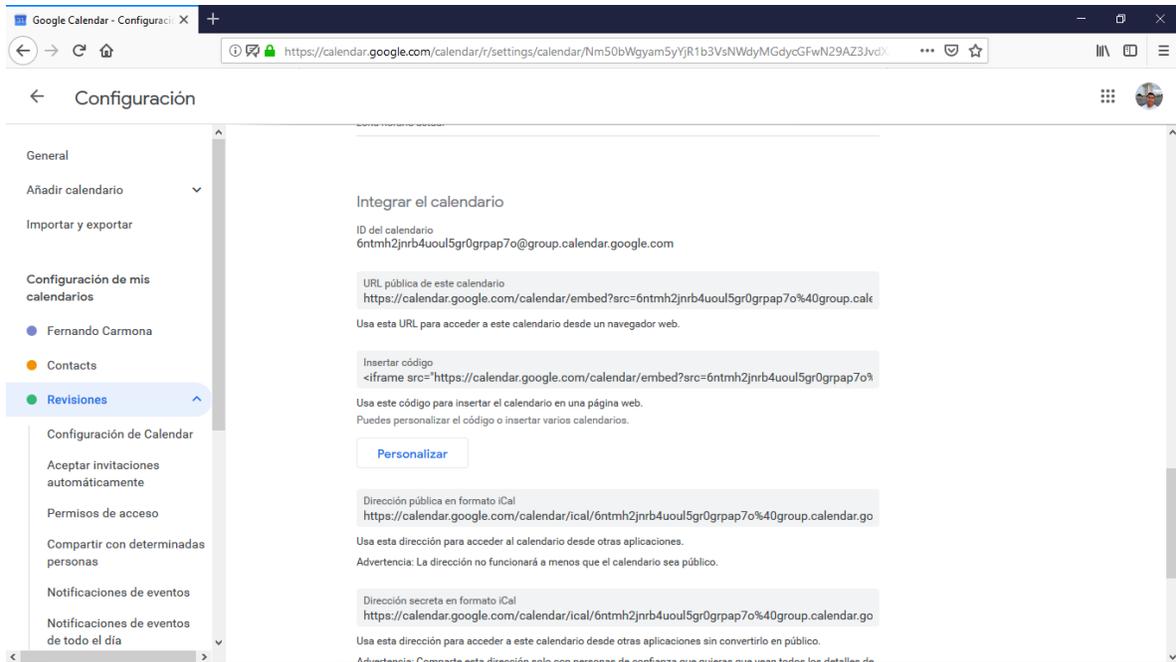


Figura 49. pantalla de configuracion de Calendar by Google.

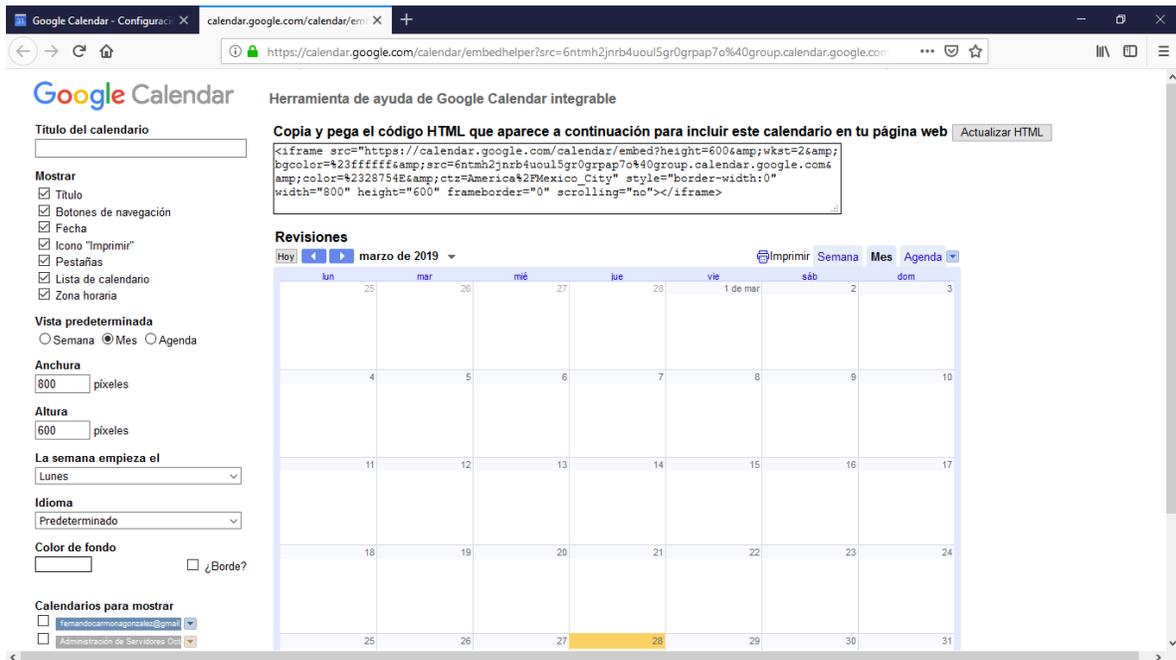


Figura 50. Herramienta de ayuda para dar estilo a la integración de Calendar.

```

principal.php
1 <!DOCTYPE html>
2 <html>
3 <html lang="en" dir="ltr">
4 <head>
5 <meta charset="utf-8">
6 <title>Agenda</title>
7
8 <?php include('menu.php'); ?>
9
10 </head>
11 <body>
12
13 <?php include('header.php'); ?>
14
15
16 <!--Fragmento en el cual se implanta calendar-->
17 <div align="center">
18
19 <iframe src="https://calendar.google.com/calendar/embed?title=Revisiones&showPrint=0&showTz=0&height=768&wkst=2&
20 hl=es_419&bgcolor=%23ffffff&src=6ntmh2jnrb4uou15gr0grrpap7o%40group.calendar.google.com&color=%23853104&
21 ctz=America%2FMexico_City" style="border:solid 1px #777" width="800" height="600" frameborder="0" scrolling="yes"></iframe>
22 </div>
23 </body>
24 <?php include('footer.php'); ?>
25 </html>

```

Figura 51. implantación de Calendar en nuestro código.

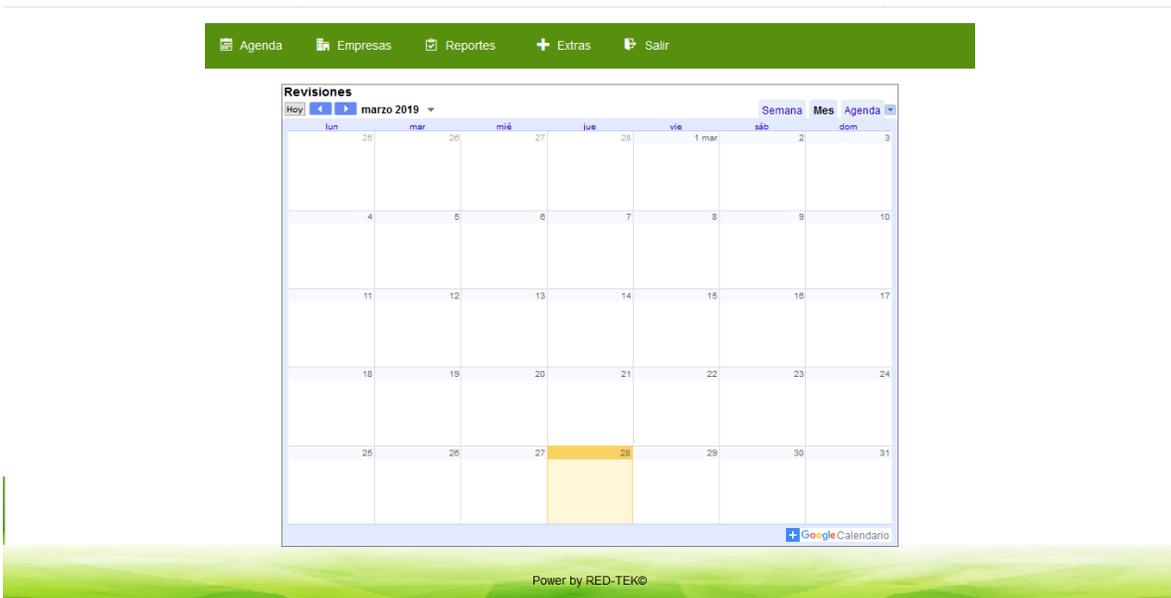


Figura 52. Resultado final del módulo Agenda

Empresas

Este módulo fue codificado con la intención que el usuario pueda visualizar la información de las empresas añadidas desde la aplicación móvil.

Para poder mostrar en una tabla la información guardada en la base de datos se creó un archivo .php donde de manera individual en el cual mediante la librería curl (Curl URL Request Library) se manda a llamar la información campo por campo como se muestra en la figura 53.

```

mostrar.php
1 <?php
2 $url = "https://sefica-e73fd.firebaseio.com/EMPRESAS/DatosEmpresa.json"; //url de nuestra base de datos.
3
4 $ch = curl_init();
5 curl_setopt($ch, CURLOPT_URL, $url);
6 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
7
8
9 $response = curl_exec($ch);
10 curl_close($ch);
11
12 //Campos de la tabla
13 echo "<table id='Tempresa' class='display' style='width:100%'>";
14 echo " <thead>
15 <tr>
16 <th>Nombre</th>
17 <th>Responsable</th>
18 <th>Direccion</th>
19 <th>Telefono</th></tr>
20 </thead><tbody>";
21 $data = json_decode($response, true);
22 foreach ($data as $key => $value) {
23 //Identificadores de los datos en la base de datos.
24 echo "<tr>";
25 echo "<td>". $data[$key]["nombre"]."</td>";
26 echo "<td>". $data[$key]["responsable"]."</td>";
27 echo "<td>". $data[$key]["direccion"]."</td>";
28 echo "<td>". $data[$key]["telefono"]."</td>";
29 echo "</tr>";
30 }
31
32 echo "</tbody>";
33 echo "</table>";
34
35 ?>

```

Figura 53. Código para obtener los datos del SGBD

Una vez obteniendo la información que llena la tabla de empresas se procedió a darle estilo y funcionalidad con el plug de DataTables by JQuery, quedando de esta manera:

En el archivo empresas.php se hacen solo las referencias del *header*, *body* y *footer*. como se muestra en la figura 54.

En el body se manda a llamar al archivo tabla.php que es el que tiene toda la información de la tabla de empresas en la cual está el plug y los estilos incluyendo una traducción ya que JQuery viene en su defecto en inglés como se muestra en la figura 55 y así poder concluir con el resultado final del módulo. ver figura 56.

```

empresas.php
1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3 <head>
4 <meta charset="utf-8">
5 <title>Empresas</title>
6
7 <?php include('menu.php'); ?>
8
9
10
11 </head>
12 <body>
13
14 <?php include('header.php'); ?>
15 <section>
16 <?php include('tabla.php'); ?>
17 </section>
18 </body>
19 <?php include('footer.php'); ?>
20 </html>
21

```

Figura 54. Código del archivo empresas.php

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <link rel="stylesheet" type="text/css" href="estilos/datatables3.css">
5 <link rel="stylesheet" type="text/css" href="https://cdn.datatables.net/buttons/1.5.2/css/buttons.dataTables.min.css">
6 <script type="text/javascript" language="javascript" src="https://code.jquery.com/jquery-3.3.1.js"></script>
7 <script type="text/javascript" language="javascript" src="https://cdn.datatables.net/1.10.19/js/jquery.dataTables.min.js"></script>
8 <script type="text/javascript" language="javascript" src="https://cdn.datatables.net/buttons/1.5.2/js/dataTables.buttons.min.js"></script>
9 <script type="text/javascript" language="javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>
10 <script type="text/javascript" language="javascript" src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.36/pdfmake.min.js"></script>
11 <script type="text/javascript" language="javascript" src="https://cdn.datatables.net/buttons/1.5.2/js/buttons.flash.min.js"></script>
12 <script type="text/javascript" language="javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>
13 <script type="text/javascript" language="javascript" src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.36/pdfmake.min.js"></script>
14 <script type="text/javascript" language="javascript" src="https://cdn.datatables.net/buttons/1.5.2/js/buttons.html5.min.js"></script>
15 <script type="text/javascript" class="init">
16 $(document).ready(function() {
17     $('#Tempresa').DataTable( {
18         "language": {
19             "lengthMenu": "Mostrar _MENU_ registros por página.",
20             "zeroRecords": "Lo sentimos. No se encontraron registros.",
21             "info": "Mostrando: _START_ de _END_ - Total registros: _TOTAL_ ",
22             "infoEmpty": "No hay registros aún.",
23             "infoFiltered": "(filtrados de un total de _MAX_ registros)",
24             "search": "Búsqueda",
25             "loadingRecords": "Cargando ...",
26             "processing": "Procesando...",
27             "searchPlaceholder": "Comience a teclear...",
28             "paginate": {
29                 "previous": "Anterior",
30                 "next": "Siguiente",
31             }
32         }
33     });
34 });
35 </script>
36 </head>
37 <body>
38 <body>
39 <?php include('mostrar.php'); ?>
40 </body>
41 </html>

```

Figura 55. Código del archivo tabla.php

Empresas

https://reportes-sefica.000webhostapp.com/empresas.php#

Aplicaciones lightningnewsb Página principal de... MEGA SII : Portal de alum... Cambiar el tamaño... Windows Server 20... Ver Harry Potter pa...

Agenda Empresas Reportes Extras Salir

Mostrar 10 registros por página. Búsqueda

Nombre	Responsable	Direccion	Telefono
CÍTRICOS CADILLO	LIC.ROSSANA GONZÁLEZ	CALLE GREGORIO TORRES QUINTERO S/N COM. EL CAÑIZO	2323245400
fer	tec	direccion	2323232323
Prueba 1	Fernando Carmona	Villa Nueva mtz	2321605491
SEFICA	ING. NARCISO MARTÍNEZ RIVERA	AV.CENTRAL #600 COL. LIBERTAD	2293305422

Mostrando: 1 de 4 - Total registros: 4 Anterior 1 Siguiente

Power by RED-TEK© Powered by 000webhost

Figura 56. Resultado final del módulo Empresas.

De igual manera se realizó el mismo procedimiento para los módulos de Observaciones y reportes de inspección ya que se basan en la misma estructura: mandar a llamar la información, transformarla en una tabla, darle estilo y funcionalidad a la tabla y mostrarla en pantalla. A diferencia del módulo anterior a estos 2 se les agrego un fragmento de código

que nos permite copiar la información de la tabla ya sea completa o filtrada agregándola en el documento tabla1.php así como se muestra en la figura 57.

```
10 <script type="text/javascript" language="javascript" src="https://cdn.datatables.net/buttons/1.5.2/js/dataTables.buttons.min.js"></script>
11
12
13 <script type="text/javascript" language="javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>
14 <script type="text/javascript" language="javascript" src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.36/pdfmake.min.js"></script>
15
16 <script type="text/javascript" language="javascript" src="https://cdn.datatables.net/buttons/1.5.2/js/buttons.flash.min.js"></script>
17 <script type="text/javascript" language="javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>
18
19 <script type="text/javascript" language="javascript" src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.36/vfs_fonts.js"></script>
20
21 <script type="text/javascript" language="javascript" src="https://cdn.datatables.net/buttons/1.5.2/js/buttons.html5.min.js"></script>
22
23
24
25 <script type="text/javascript" class="init">
26 $(document).ready(function() {
27   $('#Tempresa').DataTable( {
28
29
30     dom: 'Bfrtip',
31     buttons: [
32       {
33         extend: 'copyHtml5',
34
35       }
36     ],
37
38   },
39
40
```

Figura 57. Implantación de los botones de copiar.

Extras

En este módulo se incluyó como se había planificado 2 botones el primero para descargar la aplicación móvil y el segundo para descargar el archivo que servirá para llenar el documento de reporte por parte del usuario con la información de las tablas.

El primer botón llamado descarga SefTec fue insertado como una imagen .jpg al cual se programó de manera que al dar clic procediera a descargar el archivo APK, esto se realizó utilizando la siguiente línea de código:

```
<a href="dw/1.apk" download="SefTec"> </a>
```

donde href nos sirve para hacer la referencia de donde se encuentra nuestro archivo que será descargado y img src es la referencia de la imagen a la cual se le dará la opción de descargar y de igual forma se realizó para el botón de plantilla que es el que se utiliza para descargar el formato, dándonos un resultado final como se muestra en la figura 58.

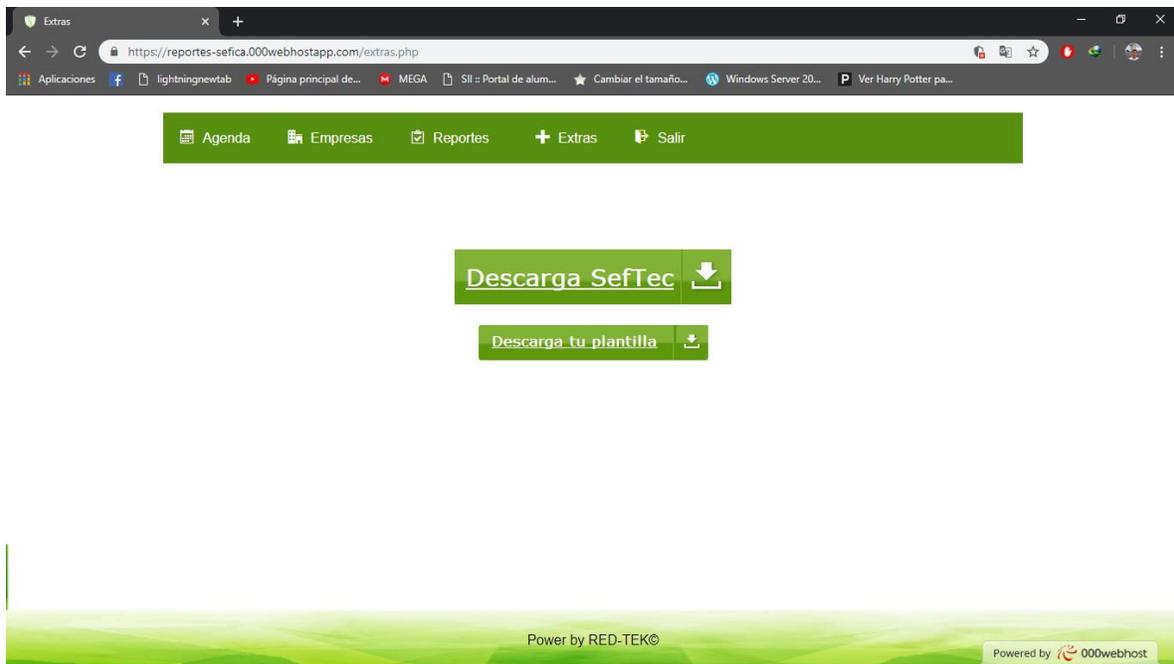


Figura 58. Resultado final del módulo Extras

5.3 Fase de Pruebas

Las pruebas de software (en inglés software testing) son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada o stakeholder. Es una actividad más en el proceso de control de calidad.

Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. Dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento de dicho proceso de desarrollo. Existen distintos modelos de desarrollo de software, así como modelos de pruebas. A cada uno corresponde un nivel distinto de involucramiento en las actividades de desarrollo. Las pruebas se pueden clasificar según lo que verifican, por ejemplo:

Pruebas funcionales

Una prueba funcional es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software (requisitos funcionales).

Niveles de prueba

Podemos considerar el proceso de pruebas funcionales como un proceso donde se va probando inicialmente lo de más bajo nivel y se van integrando y probando paulatinamente componentes hasta lograr un sistema completo totalmente probado. Por eso se dice que hay distintos niveles de prueba. Se empieza por las pruebas unitarias, luego las pruebas de Integración, luego las de pruebas de sistema, las de humo, las alpha, las beta y finalmente las de pruebas de aceptación.

Las pruebas de regresión se pueden considerar como la ejecución (normalmente automática) de las pruebas ya realizadas hasta el momento.

Pruebas no funcionales

Una prueba no funcional es una prueba cuyo objetivo es la verificación de un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema (requisitos no funcionales) como por ejemplo la disponibilidad, accesibilidad, usabilidad, mantenibilidad, seguridad, rendimiento. Pruebas de mantenibilidad

5.4 Elaboración y ejecución de pruebas

Elaboración

Para poder saber que la implementación del sistema es funcional, como primer paso se realizó una prueba de compatibilidad que consta de probar primero la aplicación móvil con los 4 dispositivos móviles con los que cuenta la empresa tomando en cuenta que son los únicos dispositivos que contaban con el principal requisito que era el sistema operativo igual o superior a Android 7.0 Nougat, además de ello se incluyó una Tablet que cuanta las características necesarias con la cual se realizaron las pruebas a manera de comprobar si es viable adquirir un dispositivo para darle como fin específico la manipulación de la aplicación móvil y así comprobar que es funcional para los trabajadores que lo utilizaran.

Para la aplicación web se realizará también las pruebas de compatibilidad con 2 diferentes exploradores web, los más utilizados por el jefe del departamento de control de plagas que será el encargado de poder administrar dicha aplicación.

Ejecución

Para la ejecución de las pruebas se tomaron en cuenta a modo de cuadro comparativo el funcionamiento y comportamiento del sistema tanto en móvil como en web tal como se muestra en las figuras 59 y 60.

Aplicación Móvil					
Modelo del dispositivo	Se ajusta 100% a la pantalla sin deformar algún elemento	Tiene algún problema relacionado con el Hardware/Software	Presento problemas de funcionalidad al ejecutar la aplicación(Congelación de pantalla, sin visualizan de datos, Tarda para responder)	Sistema operativo Android	Observaciones
Motorola G5 Plus	Si	No	No	8.1	
Samsung Galaxy A9	Si	No	No	8.1	
Samsung Galaxy J7	Si	No	No	7.1	
LG K10	Si	No	Si	7.1	El dispositivo tarda en cargar el listado de empresas y no muestra respuesta después de guardar un reporte.
Lenovo Tab 4 10 - Slate Black	Si	Si	No	7.0	El dispositivo no cuenta con ranura para una tarjeta SIM pero con wifi se puede cargar toda la información en cache lo que resulta apto para utilizar la aplicación.

Figura 59. Tabla de ejecución de pruebas móvil. Edición: Propia.

Aplicación Web				
Tipo de Explorador	Se ajusta 100% a la pantalla sin deformar algún elemento	Tiene algún problema relacionado con el Hardware/Software	Presento problemas de funcionalidad al ejecutar la aplicación(Congelación de pantalla, sin visualizan de datos, Tarda para responder)	Observaciones
Google Chrome	Si	No	No	
INTERNET EXPLORER	Si	No	Si	No muestra el calendario de revisiones

Figura 60. Tabla de ejecución de pruebas web. Edición: Propia.

CONCLUSION

En conclusión, la aplicación tuvo un cambio significativo a lo largo del periodo de realización, cuando se comenzó la aplicación solo se conformaba de 4 pantallas simples y no estaba designado que fuera una aplicación integral ya que se planteó que se descargara en formato PDF en automático después de cada revisión, pero con su evolución quedo descartado esa opción y solo se quedó la idea principal “llenar un formulario con la información de cada dispositivo” con esas bases empezamos codificando hasta que se tuvo la segunda reunión con el cliente que nos expresó que quería también visualizar la información desde su computadora y el mismo poder enviar los reportes en un documento de Word para que la empresa revisada pueda regularizar la información, lo cual cambio por completo la perspectiva del proyecto convirtiéndolo así en el proyecto que se presentó como resultado final.

Poder desarrollar una aplicación integral con un una base de datos no relacional me ayudo a explorar una nueva manera de guardar la información ya que es una tecnología innovadora, segura y utilizada por las empresas más reconocidas, utilizar Firebase como SGBD aporta un extra al proyecto.

RECOMENDACIONES

- La principal recomendación para poder utilizar la aplicación móvil es tener un Smartphone con sistema operativo Android versión superior a 7.0 Nougat.
- Para poder ingresar a la aplicación se debe de tener conexión a internet para poder validar la información que se va a ingresar.
- Para que la información se guarde en condiciones de perdida de conexión se recomienda ingresar a la aplicación por lo menos una vez y verificar que la información de las empresas asociadas aparezca ya que de ser lo contrario la aplicación se reiniciara y se perderá la información del reporte y se necesitara volver a hacer el llenado.

- El navegador que se recomienda utilizar para abrir la aplicación web es Mozilla Firefox ya que por ser más ligero hace que la fluidez de la navegación por la aplicación ofrezca la mejor experiencia.
- Para poder llenar la plantilla de reporte se recomienda agregar 4 celdas extras a la tabla ya que el formato del copiado así lo requiere.
- Una de las recomendaciones para las acciones futuras en el proyecto, sería modificar el botón de copiar por un botón que genere un documento que cumpla los lineamientos para enviar los reportes en formato .docx

REFERENCIAS

- [1] Goodwill Community Foundation (2018) . ¿Qué es una aplicación web? Recuperado de: <https://edu.gcfglobal.org/es/informatica-basica/que-es-una-aplicacion-web/1/>
- [2] Oqode.pro (2014) ¿Qué es una App Nativa? Recuperado de: <https://www.qode.pro/blog/que-es-una-app-nativa/>
- [3] Alfonso Martínez (6/06/2017). Cuatroochenta.com: ¿APP HIBRIDA O APP NATIVA? Recuperado de: <https://cuatroochenta.com/app-hibrida-o-app-nativa-segun-para-que/>
- [4] Wikipedia.org (2019). Sistema Operativo móvil. Recuperado de: https://es.wikipedia.org/wiki/Sistema_operativo_m%C3%B3vil
- [5] Wikipedia.org (2019). IOS. Recuperado de: <https://es.wikipedia.org/wiki/IOS>
- [6] ictea.com (2019). ¿Qué es el Sistema Operativo Android? Recuperado de: <http://www.ictea.com/cs/index.php?rp=/knowledgebase/8974/iQue-es-el-Sistema-Operativo-Android.html>
- [7] Juan Antonio Pascual (07/07/2018). computerhoy.com: Android vs iPhone: la guerra de los smartphones en cifras. Recuperado de: <https://computerhoy.com/reportajes/industria/android-vs-iphone-guerra-smartphones-cifras-271447>
- [8] Wikipedia.org (2019). Entorno de desarrollo integrado. Recuperado de: https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado
- [9] Alegsa.com.ar (2009). Definición de Editor web - ALEGSA © 2009-07-05 Recuperado de: http://www.alegsa.com.ar/Dic/editor_web.php
- [10] Luis del Valle Hernández(2015).ProgramarFacil.com: Cómo elegir el entorno de desarrollo web? recuperado de: <https://programarfacil.com/podcast/36-como-elegir-el-entorno-de-desarrollo-web/>
- [11] Connolly T.M. y Begg C. E. Sistemas de bases de datos, un enfoque practico. 4ed. Ed. Addison Wesley.
- [12] Firebase Realtime Database. Enero 14, 2019 Recuperado de: <https://firebase.google.com/docs/database/?hl=es-419>
- [13] SQLite.org (2019) Recuperado de:<https://www.sqlite.org/about.html>
- [14] Leandro Alegsa (2016-07-19) alegsa.com: Definición de MySQL (SGBD) <http://www.alegsa.com.ar/Dic/mysql.php>

- [15] Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. (2010). Metodología de la Investigación (5ed.). México: McGraw-Hill.
- [16] OBisnesSchool.com (2019) Metodologías ágiles de gestión de proyectos: elige la mejor. Recuperada de: <https://www.obs-edu.com/int/blog-project-management/agile-project-management-2/metodologias-agiles-de-gestion-de-proyectos-elige-la-mejor>
- [17] PMOinformatica.com (06/02/2017) Requerimientos funcionales. Recuperado de: <http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>
- [18] PMOinformatica.com (06/05/2015) Requerimientos no funcionales. Recuperado de: <http://www.pmoinformatica.com/2017/02/requerimientos-no-funcionales-ejemplos.html>
- [19] Universidad nacional abierta y a distancia (2018) Recuperado de: http://stadium.unad.edu.co/ovas/10596_9839/diagramas_de_casos_de_uso.html
- [20] Rational Software Corporation et al. "UML 1.1 Documentation Set". Recuperado de: <http://www.rational.com/uml>

ANEXOS





Manual de instalación de la aplicación
SEF-TEC

Introducción

En este manual se realizará una breve explicación de cómo utilizar el sistema integral SEF-TEC que está conformada por una aplicación móvil y una aplicación Web

Aplicación Móvil

Lo primero que se debe hacer es instalar la aplicación web para poder descargar el archivo **APK** de la aplicación

Paso 1

Desde el dispositivo móvil, ingresar a la liga <https://reportes-sefica.000webhostapp.com>, y entrar con el usuario y contraseña correspondiente como se muestra en la figura [1]

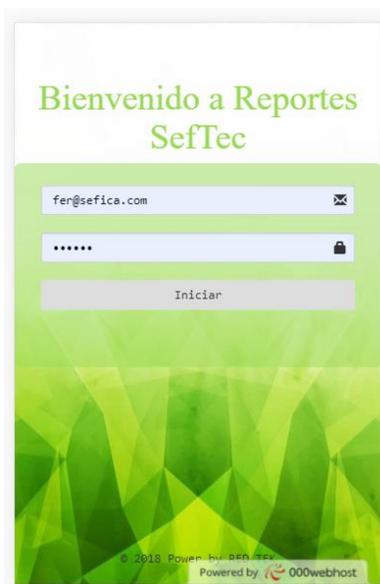


Figura 1. Iniciar sesión.

Paso 2

Dar clic en el apartado de extras como se muestra en la figura [2] y seguido de ello dar clic en el botón de **Descargar SefTec** descargar para poder bajar el APK instalador de la app he instalarla. Figura [3]

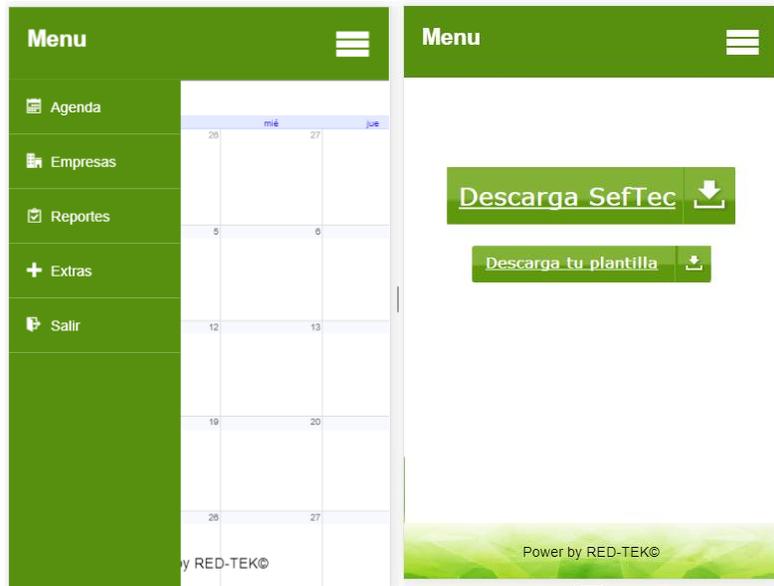


Figura 2. Menú

Figura 3. Extras

Paso 3

Una vez descargada e instalada la aplicación se procede a abrirla buscando el icono desde el menú de la terminal, ver figura [4]



Figura 4. Icono de la Aplicación

Paso 4

Lo primero que se nos mostrara es una pantalla con el logo de la aplicación para proseguir con la pantalla de usuario y contraseña para ingresar a la aplicación



Paso 5

Después de ingresar a la aplicación se observan dos botones y el nombre de usuario en la parte superior.





Manual de usuario de la aplicación SEF-TEC

Una vez ingresando a la aplicación, lo primordial es agregar una empresa para utilizar esa información más adelante. Dando clic en empresas se mostrara una lista de empresas previamente cargadas como prueba

Primero el botón con el icono de empresa envía a la siguiente página cumple su función llenar la información de las empresas para poder agregarla a la lista..



Después de agregar una empresa se puede dirigir al apartado de generar un reporte y así poder seleccionar una empresa a la cual se le realizara la revisión correspondiente. El primer paso es seleccionar la empresa la cual se cargarán un spinner y después se empezara a llenar con base a la información de la revisión.



Después de llenar la información del dispositivo se mostrará el área de acuerdo a observaciones adicionales que es la información que se recaba al final de todas las revisiones de acuerdo a las condiciones de la empresa.

y así es como se utiliza la aplicación móvil.

Nota: Para poder utilizar la aplicación en modo off-line primero debemos de ingresar teniendo conexión a internet y es indispensable no cerrar sesión seguido de ello revisar que el apartado de empresas haya cargado toda la lista de empresas. Para que la información se guarde en la memoria cache del dispositivo.

The image shows a mobile application interface for reporting an inspection. The title is 'REPORTE DE INSPECCION'. The form includes several sections: 'Reubicacion del dispositivo' with a 'SI' radio button; 'Observaciones y acciones adicionales' with a text input field; 'Observaciones durante la inspeccion' with a text input field; 'Desechos manejados' with a 'NO' radio button; 'Consumo de alimentos de forma adecuada' with a 'SI' radio button; 'evidencias de residuos de alimentos' with a 'NO' radio button; and 'Lugar/Area donde se observó el incumplimiento' with a text input field. At the bottom, there is a section for 'ACCIONES CORRECTIVAS RECOMENDADAS' and a green 'GUARDAR' button.

Preguntas frecuentes.

En este apartado se incluyen una serie de preguntas creadas por mí (el desarrollador) claro con sus respectivas respuestas para que puedas resolver algunas de tus dudas, antes y después de la instalación de la app SefTec espero poder ayudarte.

¿Con que dispositivos móviles la aplicación SefTec es compatible?

La aplicación móvil SefTec es compatible con el 99% de los dispositivos que cuenten con sistema operativo Android y a su vez con la versión de SO 7.0 Nougat.

¿Puedo descargar la aplicación SefTec desde la Play Store?

SefTec al ser una aplicación móvil de uso privado solo se puede descargar accediendo a la aplicación web, puedes acceder desde el navegador de tu dispositivo móvil, entrar a la aplicación con tu usuario y contraseña seguido de ello dirigirse al apartado de extras en el cual se mostrará un botón con la etiqueta de DESCARGAR SEFTEC y lo seguido será solo instalarlo.

¿Por qué necesito una cuenta para usar SefTec?

Como SefTec utiliza tanto datos personales como información de empresas solo usuarios específicos pueden acceder a ella.

¿Qué hacer si olvide mi usuario y contraseña?

Por motivos de seguridad la aplicación no muestra una manera de restablecer la información de usuario o crear otro, en esta caso solo envía un correo a oficialredtek@gmail.com identificate con nosotros y se te responderá con tu información de tu usuario y contraseña.

¿Porque no puedo acceder con mi información a la aplicación SefTec?

Primero debes de confirmar que tienes una conexión a internet y después que tu información sea correcta, si ya has realizado los 2 pasos anteriores y aun así no consigues entrar comunícate con nosotros.

Tengo un problema con la aplicación o me gustaría informar un error.

Agradecemos sus comentarios y nos gustaría utilizarlo para mejorar el rendimiento de nuestra aplicación. Si experimenta algún problema con nuestra aplicación móvil o desea informar un error, contáctenos al correo antes mencionado y proporciónenos la siguiente información:

- Una captura de pantalla que muestra el mensaje de error o el problema técnico que está experimentando
- ¿Cuánto tiempo llevas experimentando este problema?
- Una explicación de los pasos que ha seguido que le dio como resultado el mensaje de error o problema técnico

¿Puedo utilizar la misma cuenta en varios dispositivos?

Al estar toda la información cargada en la nube se puede utilizar cualquier dispositivo, claro teniendo en cuenta la versión del SO, solo basta con instalar la aplicación insertar tus datos y listo.

¿Se necesita contar siempre con una conexión a internet?

La respuesta es sí y no. Si se necesita tener internet para poder acceder al inicio de a la aplicación y visualizar el listado de empresas. Y no, una vez entrando a la aplicación y cargada la lista de empresas esta se guarda en la memoria cache y se pueden guardar los reportes teniendo en cuenta que la información se subirá tan presto se tenga una conexión a internet.

¿Cómo puedo visualizar la información que eh recolectado con la aplicación?

Para visualizar la información puedes entrar a la aplicación web desde tu computadora entrando al siguiente link: <https://reportes-sefica.000webhostapp.com/>