

**UNIVERSIDAD POLITÉCNICA DE PUEBLA**  
**Ingeniería en Informática**



**Proyecto de Estadía Profesional**

“Aplicación de supervisión de créditos para etapa inicial, intermedia y final del proceso de crédito.”

Área temática del CONACYT: VII  
Ingenierías y tecnologías

**Presenta:**

**Laura Zacatzontetl Hernandez**

**Asesor técnico**

Lic. Germán Ortega Benítez

**Asesor académico**

M.C. Rebeca Rodríguez Huesca

# Resumen

---

En este documento se describirá el problema al que se enfrenta la financiera Maestros Unidos debido a la falta de un software que ayude a la verificación de la aplicación del crédito, también se mencionarán los objetivos generales y específicos, así como la metodología usada para el desarrollo del proyecto, la cual está conformada por seis etapas las cuales son detalladas a continuación: la primer etapa es product backlog en ella se definen las historias de usuario, la segunda etapa es sprint planning donde se establece la asignación de roles, priorización de requerimientos y funcionalidades detalladas de la aplicación, la tercera etapa es sprint durante la cual se muestra el conjunto de tareas que se han logrado implementar, la cuarta etapa es sprint backlog en ella se listan las tareas que se realizaron para desarrollar las historias de usuario, en la etapa daily sprint meeting se detallan las reuniones diarias que se llevaron a cabo durante el desarrollo del proyecto y por último la etapa demo y retrospectiva hace mención de las pruebas que se realizaron y de las cuales se recibió retroalimentación para la mejora de la aplicación.

VERIFICAPP es un software que ayudará al proceso de supervisión y verificación de la aplicación del crédito de la financiera Maestros Unidos. Este software permitirá agilizar la recopilación de los diferentes tipos de evidencias y automatizar la generación de reportes en pdf que ayuden a tomar medidas preventivas para la recuperación del monto del crédito y así evitar problemas al momento de la recuperación de la cartera para ambas partes, de este modo la empresa podrá obtener una cartera crediticia sana y así obtener beneficios a mediano y corto plazo con instituciones que suelen otorgar un fondeo para operaciones futuras de la financiera.

# Índice

---

1. Introducción.....	7
1.1 Descripción del problema o necesidad .....	7
1.2 Justificación.....	8
1.3 Objetivo General y Específicos .....	9
2. Metodología y herramientas .....	10
2.1 SCRUM.....	10
2.2 Herramientas tecnológicas utilizadas.....	12
3. Resultados .....	15
3.1 Product backlog .....	15
3.2 Sprint Planning.....	18
3.3 Sprint.....	24
3.3.1 Sprint 1 .....	24
3.3.2 Sprint 2.....	31
3.3.3 Sprint 3.....	33
3.3.5 Sprint 4.....	46
3.3.7 Sprint 5.....	52
3.4 Sprint Backlog .....	59
3.5 Daily sprint meeting.....	61
3.6 Demo y Retrospectiva.....	66
4. Conclusiones y recomendaciones.....	71
5. Anexos .....	72
4. Referencias bibliográficas .....	78

# Índice de figuras

---

Figura 1. Etapas metodología SCRUM .....	11
Figura 2. Historias de usuario.....	15
Figura 3. Historias de usuario.....	16
Figura 4. Historias de usuario.....	16
Figura 5. Historias de usuario. ....	16
Figura 6. Base de datos VERIFICAPP .....	24
Figura 7. Action bar de la aplicación VERIFICAPP .....	24
Figura 8. Registro de VERIFICAPP .....	25
Figura 9. Registro de VERIFICAPP con campos llenos.....	26
Figura 10. Inicio de Sesión de VERIFICAPP .....	27
Figura 11. Inicio de Sesión de VERIFICAPP .....	28
Figura 12. Búsqueda de cliente. ....	29
Figura 13. Buscador de cliente.....	30
Figura 14. Cliente seleccionado con sus respectivos datos.....	31
Figura 15. Pantalla Verificación.....	32
Figura 16. Pantalla Verificación.....	33
Figura 17. Pregunta de tipo texto abierto .....	34
Figura 18. Pregunta de tipo respuesta seleccionable .....	35
Figura 19. Pregunta de tipo texto abierto .....	36
Figura 20. Pregunta de tipo foto.....	37
Figura 21. Cámara de la pregunta tipo foto.....	38
Figura 22. Fotografía tomada dentro de la ampliación VERIFICAPP .....	39
Figura 23. Mensaje para volver a tomar otra foto.....	40
Figura 24. Pregunta de tipo georreferencia.....	41
Figura 25. Pregunta de tipo texto abierto .....	42
Figura 26. Pregunta de tipo foto.....	43
Figura 27. Pregunta de tipo decimal .....	44
Figura 28. Pregunta de tipo texto abierto .....	45
Figura 29. Visualización de respuestas .....	46

Figura 30. Pantalla firmar documento .....	47
Figura 31. Pantalla firmar documento con firma.....	48
Figura 32. Pantalla con el botón finalizar reporte habilitado.....	49
Figura 33. Mensaje de precaución .....	50
Figura 34. Mensaje de precaución. ....	51
Figura 35. Botones visualizar reporte y generar reporte habilitados .....	52
Figura 36. Barra de progreso al generar el pdf. ....	53
Figura 37. Pdf dentro de la aplicación VERIFICA .....	54
Figura 38. Pdf dentro de la aplicación VERIFICAPP .....	55
Figura 39. Pdf dentro de la aplicación VERIFICAPP.....	56
Figura 40. Reporte pdf en aplicación externa .....	57
Figura 41. Reporte pdf en aplicación externa .....	58
Figura 42. Reporte pdf en aplicación externa .....	59
Figura 43. Búsqueda de cliente.....	66
Figura 44. Visualización de los datos búsqueda de cliente .....	67
Figura 45. Pantalla Verificación.....	68
Figura 46. Pantalla cuestionario dinámico .....	69
Figura 47. Reporte de verificación .....	70

# Índice de Tablas

---

Tabla 1. Requerimientos funcionales.....	17
Tabla 2. Requerimientos no funcionales.....	18
Tabla 3. Asignación de roles.....	18
Tabla 4. Priorización de requerimientos.....	19
Tabla 5. Tareas de cada Sprint.....	23
Tabla 6. Tareas para implementar las historias.....	60
Tabla 7. Descripción del Daily Sprint meeting.....	62
Tabla 8. Descripción del Daily Sprint meeting.....	63
Tabla 9. Descripción del Daily Sprint meeting.....	65

# 1. Introducción

---

En este capítulo se describe el problema que tiene el proceso de supervisión y verificación de la aplicación del crédito de la financiera Maestros Unidos, así como también se describe la solución propuesta, las herramientas con las que se cuenta para realizar el proyecto, el objetivo general y los objetivos específicos que ayudarán durante el desarrollo del proyecto.

## 1.1 Descripción del problema o necesidad

La empresa Consultas Nacionales de Información y Asesoría SA DE CV, está conformada por pequeñas y varias empresas, una de ellas es la financiera Maestros Unidos, que se encarga de realizar créditos y préstamos con descuento vía nomina únicamente ofertados a empleados del Sindicato Nacional de Trabajadores de la Educación (SNTE).

Actualmente la financiera tiene software especializado para realizar las consultas en buró de crédito y también cuenta con acceso a las bases de datos de los trabajadores de la SNTE para la validación de los datos de los futuros clientes y así poder dar seguimiento al crédito o al préstamo. Maestros Unidos ofrece cinco tipos de crédito los cuales son: crédito de habilitación o avío, refaccionario, prendario, simple o personal y cuenta corriente, conforme al crédito seleccionado por el cliente es como se supervisará dicho proceso, para el crédito de avío y refaccionario es necesario realizar la supervisión de campo de forma presencial y detallada para verificar si el crédito ha sido aplicado conforme lo que se estipuló en el contrato, este tipo de verificación se realiza para evitar el mal uso del crédito en proyectos que puedan llegar a infringir la ley, también se realiza para evitar desvíos del crédito a proyectos no sustentables que puedan poner en riesgo la recuperación del monto del crédito.

Debido a estos motivos se planea dar un seguimiento puntual mediante una aplicación que permita verificar de una manera más sencilla y tecnológica el proceso.

Se pretende desarrollar una aplicación en Android que permita documentar la supervisión y verificación de la aplicación del crédito, almacenando en una base de datos diferentes tipos de evidencias de la supervisión de campo, posteriormente con la obtención de evidencias la aplicación nos permitirá generar un reporte detallado en pdf y así poder tomar las medidas necesarias para la recuperación de la cartera.

## 1.2 Justificación

Actualmente la aplicación de los créditos es supervisada o verificada por el promotor de forma presencial, pero debido a la falta de tiempo y de una herramienta tecnológica que ayude a la recolección de evidencias, la mayoría de las aplicaciones del crédito no suelen ser supervisadas correctamente e incluso algunas de ellas no son supervisadas, lo que es un error grave porque puede llegar a ocasionar un gran conflicto en la recuperación del crédito, debido a que no se tiene el seguimiento correcto de la supervisión de éste, pues no se sabe si la aplicación del crédito ha sido exitosa o no, y por lo tanto no se sabe si se obtendrá una recuperación de la cartera exitosamente.

Es por ello que el dar un seguimiento a la supervisión y verificación de la aplicación del crédito es altamente importante para la financiera porque si se verifica de manera correcta la aplicación del crédito nos permitirá tener una cartera crediticia sana.

Se desarrollará una aplicación en Android que permita documentar la supervisión y verificación de la aplicación del crédito, almacenando en una base de datos diferentes tipos de evidencias de la supervisión de campo, como lo son características principales del inmueble, ganado o materia prima, fotografías con geolocalización del inmueble, ganado o lugar del almacenamiento de materias primas, ubicación, superficie, medidas, colindancias, nombre del predio o finca, cantidad, razas entre otras características. Estos datos a su vez nos permitirán generar un reporte detallado en pdf sobre cómo ha sido aplicado el crédito y así poder determinar si el monto del crédito será recuperado exitosamente o no y así tomar las medidas necesarias para la recuperación de éste.

La empresa tendrá mayor eficacia en la supervisión y verificación de créditos puesto que la aplicación agilizará el proceso de recopilación de evidencia y la promotora o el encargado de dicho proceso tendrá un mejor control del seguimiento de la aplicación de los créditos. La relevancia del estudio para el usuario es que debe contar con conocimientos previos en tecnologías de la información, así como la facilidad de adaptarse a nuevos entornos y software de trabajo.

El proyecto impacta tanto en los clientes como en la empresa puesto que el tener una verificación eficaz de la aplicación del crédito evitará problemas al momento de la recuperación de la cartera para ambas partes y la empresa al poder obtener una cartera crediticia sana podrá obtener beneficios a mediano y corto plazo con instituciones que suelen otorgar un fondeo para operaciones futuras de la financiera.



Las herramientas con las que se cuenta para el desarrollo son una computadora de escritorio con software especializado en desarrollo de aplicaciones en Android (Android Studio), así como también el manejador de BD Firebase y acceso a internet.

### **1.3 Objetivo General y Específicos**

Desarrollar una aplicación en Android que ayude a el proceso de supervisión y verificación de la aplicación del crédito de la financiera Maestros Unidos generando reportes detallados en pdf para mejorar el proceso.

Objetivos específicos:

- Analizar el proceso de verificación de campo actual
- Diseñar las pantallas de la aplicación, así como también los cuestionarios para cada tipo de crédito y el reporte de pdf
- Programar cada una de las pantallas del proyecto, las validaciones de cada uno de los campos y el reporte detallado en pdf
- Realizar pruebas de funcionalidad y validación para cada uno de los campos de las pantallas

## 2. Metodología y herramientas

---

En este capítulo se describen las definiciones, características, ventajas y desventajas de la metodología SCRUM al igual que las herramientas tecnológicas Android Studio y el manejador de BD Firebase para el desarrollo del proyecto.

### 2.1 SCRUM

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa. Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, autogestión e innovación. [1]

#### 2.1.1 Etapas de la metodología [2]:

Las etapas de la metodología SCRUM se muestran en la figura 1, dicha metodología está conformada por seis etapas las cuales son:

- **Product Backlog:** Conjunto de requisitos denominados historias descritos en un lenguaje no técnico y priorizados por valor de negocio, o lo que es lo mismo, por retorno de inversión considerando su beneficio y coste.
- **Sprint Planning:** Durante una reunión el product owner presenta las historias del backlog por orden de prioridad, se determina la cantidad de historias a completar en ese sprint, en la segunda parte de la reunión se decide y organiza cómo se va a conseguir.
- **Sprint:** Iteración de duración prefijada durante la cual el equipo trabaja para convertir las historias del product backlog a las que se ha comprometido, en una nueva versión del software totalmente operativo.
- **Sprint Backlog:** Lista de tareas necesarias para llevar a cabo las historias del sprint.
- **Daily sprint meeting:** Reunión diaria de máximo quince minutos en la que el equipo se sincroniza para trabajar de forma coordinada. Cada miembro comenta que hizo el día anterior, que hará hoy y si existen impedimentos.
- **Demo y retrospectiva:** Reunión que se celebra al final del sprint y en la que el equipo presenta las historias logradas mediante la demostración del producto, en la retrospectiva el equipo analiza qué se hizo bien, las mejoras y discusiones que pueden ser perfeccionadas.

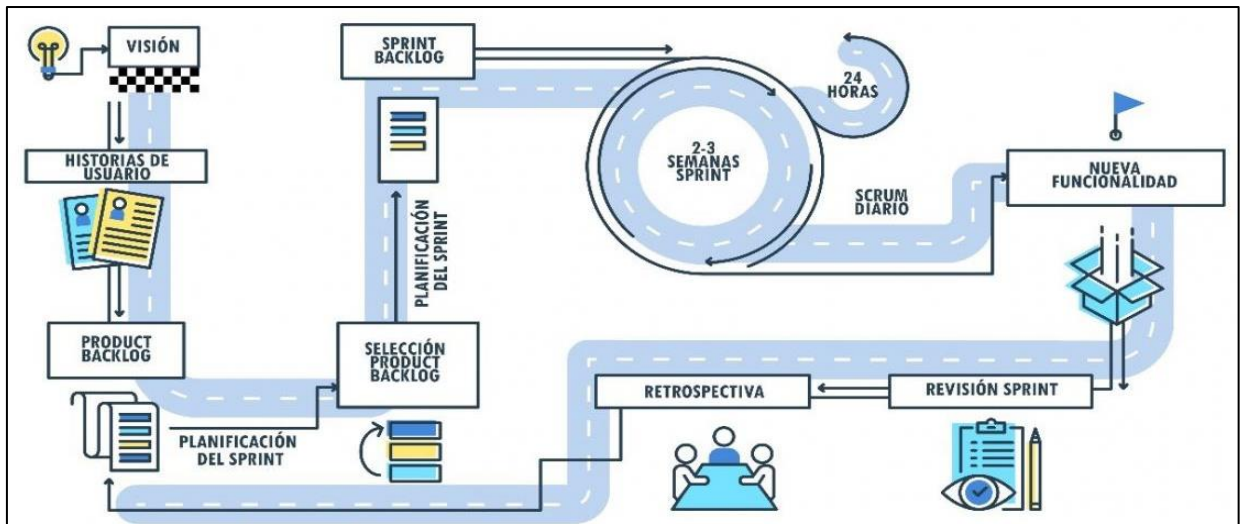


Figura 1. Etapas de la metodología SCRUM

### 2.1.2 Roles de la metodología [2]:

SCRUM está formado por los siguientes roles:

- Scrum master: Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el product owner para maximizar el rol.
- Product owner: Representante de los accionistas y clientes que usan el software, persona enfocada en el negocio y quien traslada la visión del proyecto al equipo.
- Team: Grupo de profesionales con conocimientos técnicos que desarrollan el proyecto, los cuales llevan a cabo las historias a las que se comprometieron al inicio de cada sprint.

### 2.1.3 Ventajas de SCRUM [3]:

- Entrega de un producto funcional al finalizar cada Sprint.
- Posibilidad de ajustar la funcionalidad en base a la necesidad de negocio del cliente.
- Visualización del proyecto día a día.
- Alcance acotado y viable.
- Equipos integrados y comprometidos con el proyecto, toda vez que ellos de ellos definieron el alcance y se auto administran
- Alta flexibilidad a cambios
- Mayor calidad del software
- Maximiza el retorno de la inversión
- Reducción de riesgos

### **2.1.4 Desventajas de SCRUM [3]:**

- No genera toda la evidencia o documentación.
- No es apto para todos los proyectos.
- Alto nivel de estrés de miembros del equipo
- La necesidad de cumplir con un sprint puede no ser de la mejor calidad

### **Razones por las cuales se utilizará la metodología SCRUM:**

- Al ser SCRUM una metodología ágil y flexible para gestionar el desarrollo de software permitirá maximizar el retorno de la inversión.
- Una de las principales razones es que nos permite construir la funcionalidad de mayor valor que en este caso son los cuestionarios dinámicos de la aplicación
- La inspección continua nos permite realizar mejoras en la aplicación
- La metodología nos permite innovar en el comienzo de cada nuevo sprint lo que beneficia al proyecto, si surgieran nuevas ideas
- Permite priorizar nuestras funcionalidades que aporten mayor valor.

## **2.2 Herramientas tecnológicas utilizadas**

Las herramientas tecnológicas son importantes en el desarrollo de proyectos informáticos a continuación se describen las herramientas que se utilizaron en el desarrollo de este proyecto.

### **2.2.1 Android Studio [4]:**

Android Studio es un entorno de desarrollo integrado IDE, basado en IntelliJ IDEA de la compañía JetBrains. Android Studio utiliza una licencia de software libre Apache 2.0, está programado en Java y es multiplataforma.

#### **2.2.1.1 Características [4]:**

Tiene una excelente estructura simple y organizada para llevar a cabo la elaboración de proyectos de cualquier tamaño, sumamente intuitiva, simple y fácil de usar, tiene una excelente visualización, cuenta con plantillas para el diseño de diversos elementos, facilita la creación de proyecto, así como también permite ejecutar los proyectos en cualquier dispositivo que cuente con el sistema operativo de Android y también permite usar emuladores.

Utiliza ProGuard para optimizar y reducir el código del proyecto al exportar a APK, cuenta con la herramienta Gradle que es la encargada de gestionar y automatizar la construcción de proyectos, como pueden ser las tareas de testing, compilación o empaquetado, permite la integración con Google Cloud Platform, para el acceso a los diferentes servicios que proporciona Google en la nube.

Requerimientos del sistema:

- Microsoft Windows 8/7/Vista/2003 (32 o 64 bit)
- Mínimo de 2 GB de RAM, recomendado 4 GB de RAM
- 400 MB de espacio en disco
- 1 GB para Android SDK, emulador de imágenes del sistema, y cachés
- Resolución mínima de pantalla de 1280 x 800
- Java Development Kit (JDK) 7 o superior

#### **2.2.1.2 Ventajas [4]:**

- Al descargar Android Studio se disponen de todas las herramientas necesarias para el desarrollo de aplicaciones
- Construye los paquetes .apk, mediante el uso de Gradle
- Facilita la distribución de código, y por lo tanto el trabajo en equipo.
- Reutilización de código y recursos.
- Permite compilar desde línea de comandos, para aquellas situaciones en las que no esté disponible un entorno de desarrolló.
- Mayor facilidad para la creación de diferentes versiones de la misma aplicación

#### **2.2.1.3 Desventajas [4]:**

- Curva de aprendizaje más lenta para nuevos desarrolladores de Android.
- El sistema de construcción de proyectos Gradle puede resultar complicado inicialmente.
- En comparativa con Eclipse, tiene menor número de plugins.

#### **2.2.2 Firebase [5]:**

Firebase es la nueva y mejorada plataforma de desarrolló móvil en la nube de Google. Se trata de una plataforma disponible para diferentes plataformas Android, iOS, web, con lo que de esta forma presentan una alternativa seria a otras opciones para ahorro de tiempo en el desarrolló. Su principal finalidad es el desarrollo de aplicaciones web y aplicaciones móviles.

##### **2.2.2.1 Características [5]:**

Provee una solución gratuita para tener todo tipo de medidas hasta 500 tipos de eventos, para gestionarlo todo desde un único panel, también permite construir mejores apps, permitiendo poder ahorrar tiempo, evitar bugs y obtener un aceptable nivel de calidad, cuenta con testeo, configuración remota, mensajería en la nube y autenticación.

### **2.2.2.2 Ventajas [6]:**

- Sincroniza fácilmente los datos de sus proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja.
- Usa un conjunto de herramientas multiplataforma: se integra fácilmente para plataformas web como en aplicaciones móviles. Es compatible con grandes plataformas, como IOS, Android, aplicaciones web, Node.js, Unity y C++.
- Usa la infraestructura de Google y escala automáticamente para cualquier tipo de aplicación, desde las más pequeñas hasta las más potentes.
- Crea proyectos sin necesidad de un servidor: Las herramientas se incluyen en los SDK para los dispositivos móviles y web, por lo que no es necesario la creación de un servidor para el proyecto.

### **2.2.2.3 Desventajas [6]:**

- Límite de conexiones simultáneas como máximo, 100 conexiones simultáneas. Para conexiones simultáneas ilimitadas deberás pagar por una cuenta con más capacidad y funcionalidades.
- Algunas funcionalidades no disponibles
- El servicio de testing del apk en la nube sólo puede realizarse quince pruebas en dispositivos por día, de los cuales, sólo cinco dispositivos son físicos, los demás son virtualizados.

# 3. Resultados

---

En este capítulo se describe como se llevaron a cabo cada una de las etapas de la metodología SCRUM, también contiene imágenes con sus respectivas descripciones que sirven de ayuda para comprender el funcionamiento y validación de la aplicación.

## 3.1 Product backlog

Durante esta etapa se realizó una reunión con la financiera Maestros Unidos en la cual se definieron las historias de usuario como se muestran en las figuras 2, 3, 4 y 5.

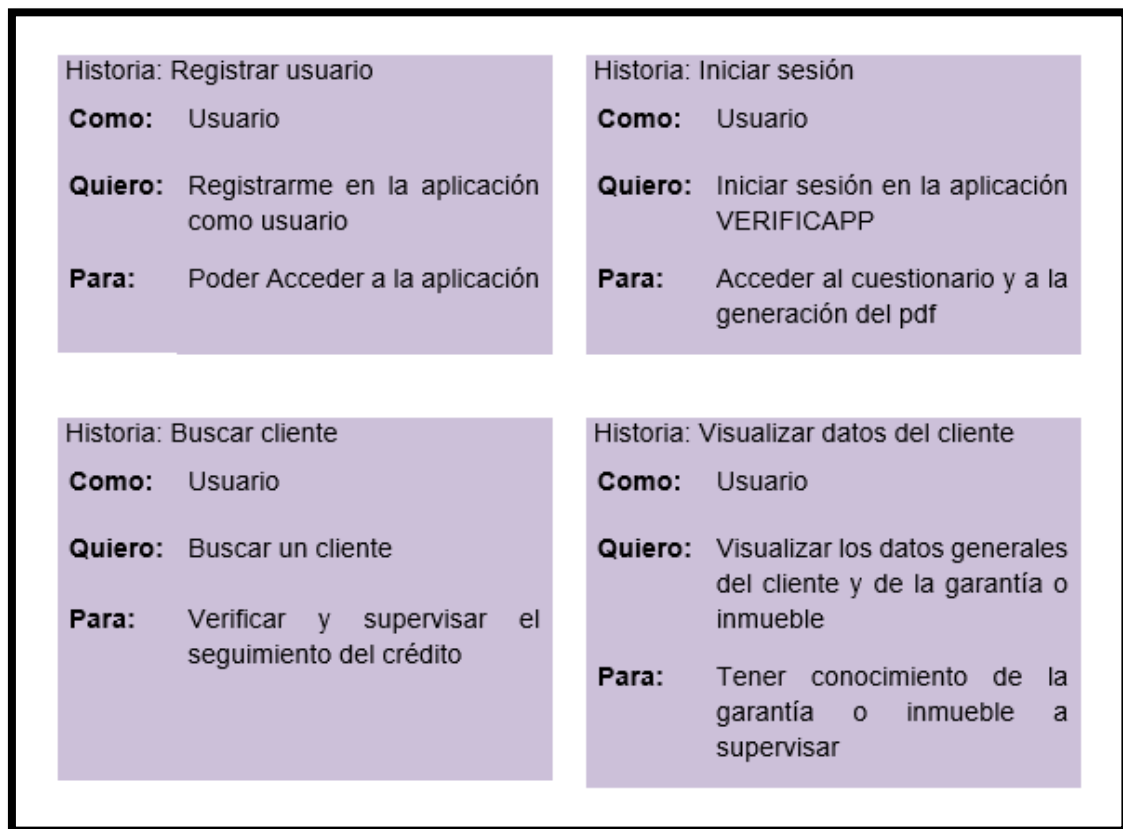


Figura 2. Historias de usuario.

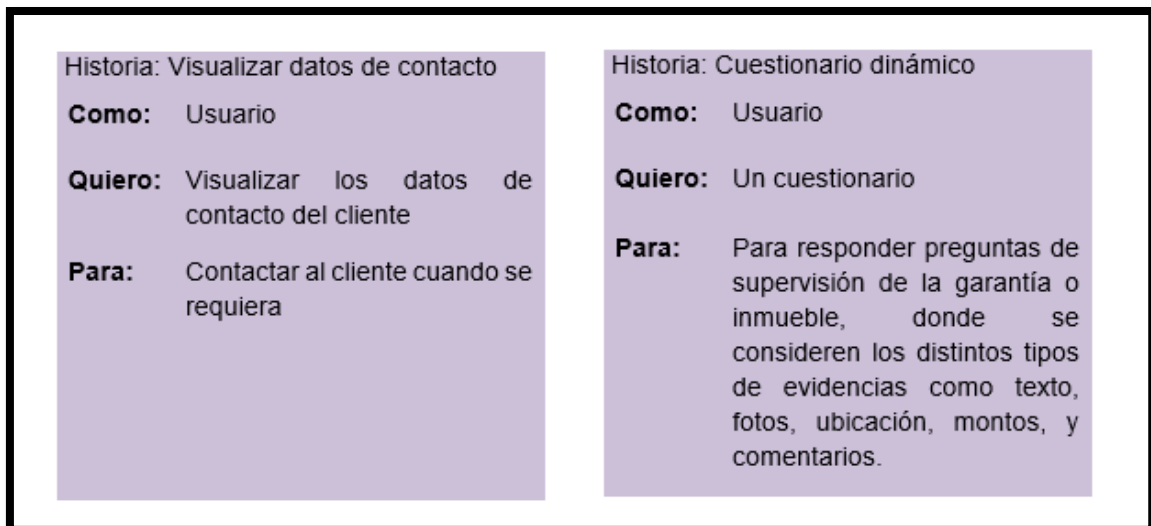


Figura 3. Historias de usuario.

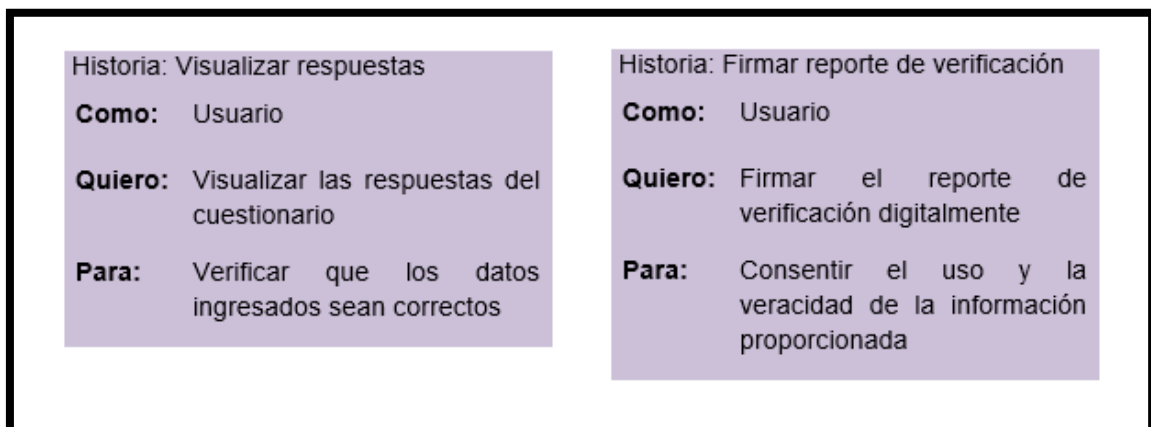


Figura 4. Historias de usuario.

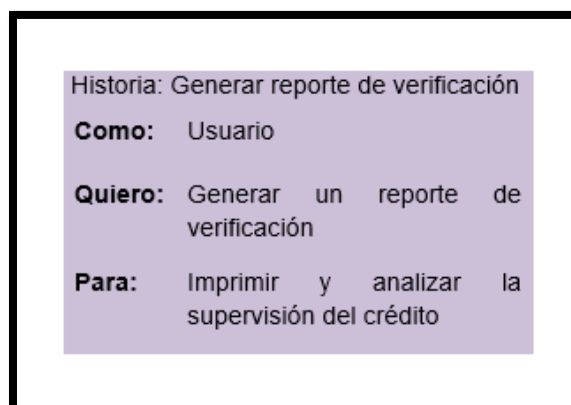


Figura 5. Historias de usuario.



Durante esta etapa también se detallan en la tabla 1 y 2 los requerimientos funcionales y no funcionales, que fueron detallados por el Product Owner.

### Requerimientos funcionales

Id	Descripción del requerimiento
RF1	La aplicación deberá contar con una pantalla de registro.
RF2	La aplicación deberá contar con un inicio de sesión.
RF3	La aplicación validará que el usuario y contraseña se encuentren registrados.
RF4	La aplicación validará el número de teléfono, el email y contraseña introducidas en el formulario de registro.
RF5	La aplicación permitirá realizar la búsqueda de cliente por folio o nombre.
RF6	La aplicación deberá mostrar los datos generales del cliente y de la garantía.
RF7	La aplicación permitirá visualizar el cuestionario dinámico.
RF8	La aplicación permitirá contestar el cuestionario dinámico.
RF9	La aplicación permitirá tomar fotografías al contestar el cuestionario.
RF10	La aplicación permitirá obtener la georreferencia de la garantía que se está verificando al obtener su latitud y longitud para posteriormente mostrarla en formato de dirección.
RF11	La aplicación permitirá tomar fotografías del inmueble o de la evidencia que sea proporcionada por el cliente para verificar que los datos sean verídicos.
RF12	La aplicación permitirá seleccionar una respuesta del spinner al contestar el cuestionario.
RF13	La aplicación permitirá introducir números decimales al contestar el cuestionario.
RF14	La aplicación permitirá introducir texto plano al contestar el cuestionario.
RF15	La aplicación permitirá mostrar las respuestas del cuestionario.
RF16	La aplicación permitirá firmar el documento.
RF17	La aplicación permitirá finalizar el cuestionario.
RF18	La aplicación permitirá visualizar el pdf dentro de la aplicación.
RF19	La aplicación permitirá generar el pdf y visualizarlo en una aplicación externa
RF20	La aplicación debe estar conectada a internet para almacenar la información.

**Tabla 1.** Requerimientos funcionales.

## Requerimientos no funcionales

Id	Descripción del requerimiento
<b>RNF1</b>	Las pantallas de la aplicación tendrán el logo de la empresa y un fondo blanco.
<b>RNF2</b>	Las interfaces de la aplicación serán amigables para el usuario.
<b>RNF3</b>	Los formularios serán simples para el usuario.
<b>RNF4</b>	La aplicación deberá ser fácil de usar para el usuario teniendo contenido ordenado y clara.
<b>RNF5</b>	Los datos serán almacenados en Firebase y posteriormente consultada durante toda la aplicación.
<b>RNF6</b>	Los datos serán almacenados en tiempo real.
<b>RNF7</b>	La aplicación se desarrollada en el lenguaje de programación Java.

**Tabla 2.** Requerimientos no funcionales.

## 3.2 Sprint Planning

La metodología SCRUM establece tres tipos de roles los cuales desempeñan distintas funciones y responsabilidades como se detalla a continuación:

1. Product Owner: gestionar el producto backlog y optimizar el producto.
2. Scrum Master: gestiona el proceso scrum y elimina impedimentos.
3. Team: desarrolla incrementos terminados y se autogestiona.

La asignación de roles para este proyecto se muestra en la Tabla 4:

Roles	Nombre	Iniciales
Product Owner	Alejandro Atala Layun	ATL
Scrum Master	Germán Ortega Benítez	GOB
Team	Laura Zacatzontetl Hernandez	LZH

**Tabla 3.** Asignación de roles

La tabla 4 surge a partir de las historias de usuario antes mencionadas, en ella se muestra información detallada sobre las funcionalidades de VERIFICAPP, como la descripción de la tarea, prioridad, requisito y por quien será realizado cada sprint.

Id	Historia de Usuario	Tarea	Prioridad	Requisito	Realizado por
1	HU1. Registrar usuario	Diseñar la base de datos en Firebase	Alta	Realizar el diagrama de la base de datos en Firebase	LZH
		Crear la base de datos en Firebase	Alta	Crear la base de datos en Firebase tal y como se estableció en el diagrama	LZH
		Diseñar el action bar de la aplicación	Baja	Realizar el diseño del action bar de la aplicación, la cual contenga el icono de la financiera	LZH
		Programar el action bar	Media	Programar el action bar en lenguaje Android con su respectivos colores e icono	LZH
		Diseñar el formulario de registro de usuario	Media	Realizar el diseño del formulario en Android Studio con los campos nombre completo, teléfono, email, password y puesto.	LZH
		Programar formulario registrar usuario	Media	Programar el formulario registrar en lenguaje Android con sus respectivos campos	LZH
2	HU2. Iniciar sesión	Diseñar el formulario de inicio de sesión	Media	Realizar el diseño del formulario en Android Studio con los campos email y password	LZH
		Programar formulario iniciar sesión	Media	Programar el formulario iniciar sesión en lenguaje Android con sus respectivos campos	LZH

Id	Historia de Usuario	Tarea	Prioridad	Requisito	Realizado por
3	HU3. Buscar cliente	Crear las consultas de información de los clientes y de las garantías	Media	Crear las consultas de Firebase en el lenguaje Android para consultar la información que requiera la aplicación	LZH
		Diseñar la interfaz búsqueda cliente	Media	Realizar el diseño de la interfaz búsqueda de cliente la cual contenga: un buscador del cliente, nombre, dirección, dirección fiscal y teléfono del cliente, tipo de garantía y dirección de la garantía y por último un botón continuar.	LZH
		Programar la búsqueda del cliente	Media	Programar la búsqueda en lenguaje Android con búsqueda por letra y folio	LZH
4	HU4. Visualizar datos del cliente	Programar la visualización de datos del cliente	Media	Programar la visualización de datos del cliente en Android el cual debe mostrar: datos generales del cliente como lo son nombre, dirección, dirección fiscal y teléfono, así como también los datos generales de la garantía como tipo de garantía y dirección.	LZH
5	HU5. Visualizar datos de contacto	Programar la visualización de datos de contacto del cliente	Media	Programar la visualización de datos de contacto del cliente en Android el cual debe mostrar: datos generales del cliente nombre, dirección y teléfono.	LZH

Id	Historia de Usuario	Tarea	Prioridad	Requisito	Realizado por
6	HU6. Cuestionario dinámico	Diseñar la interfaz Verificación de la aplicación del crédito	Alta	Realizar el diseño de la interfaz verificación de la aplicación del crédito la cual debe contener: nombre, dirección y teléfono del cliente, cuestionario dinámico, botón firmar documento, botón visualizar reporte, botón generar reporte y finalizar reporte.	LZH
		Programar la interfaz verificación de la aplicación del crédito	Media	Programar la interfaz en lenguaje Android con los campos: datos del cliente	LZH
		Programar el cuestionario dinámico	Alta	Programar el cuestionario dinámico en lenguaje Android el cual debe contener campos como: fotografía, ubicación, texto y opciones seleccionables	LZH
7	HU7. Visualizar respuestas	Diseñar la interfaz firmar documento	Media	Realizar el diseño de la interfaz firmar documento la cual debe contener: indicaciones, cuadro de dibujo, botón limpiar y botón continuar	LZH
		Programar la firma digital del documento	Media	Programar la función firmar en lenguaje Android, así como también el crear la imagen firma, guardarla y agregarla al pdf	LZH
		Programar el botón finalizar reporte	Media	Programar las funciones de modificar el estado de las preguntas para procesarlas como finalizadas y terminar la verificación de la ampliación del crédito	LZH

Id	Historia de Usuario	Tarea	Prioridad	Requisito	Realizado por
8	HU8. Firmar reporte de verificación	Diseñar la interfaz para visualizar el reporte	Media	Realizar el diseño de la interfaz para visualizar el reporte la cual debe contener un PDF viewer.	LZH
		Programar la visualización del pdf	Media	Programar la función para visualizar pdf en lenguaje Android para mostrar los datos previos del pdf	LZH
		Diseñar y programar el reporte de VERIFICAPP	Alta	Realizar el diseño y la programación del PDF la cual debe mostrar la información del supervisor y del cliente, así como también el cuestionario en forma de tabla, con fotografías y firma digital	LZH
9	HU9. Generar reporte de verificación	Programar la generación del pdf	Alta	Programar las funciones para generar el pdf en lenguaje Android	LZH
		Diseñar el botón cerrar Sesión	Media	Realizar el diseño del botón cerrar sesión en el action bar y agregar su respectivo icono	LZH
		Programar la función cerrar sesión	Media	Programar la función para cerrar sesión en lenguaje Android	LZH

**Tabla 4.** Priorización de requerimientos.

También en esta etapa se agruparon las historias de usuario que conformarán cada sprint, así como también el periodo del sprint, el daily sprint meeting, demo y retrospectiva y por último el nombre del desarrollador tal y como se muestra en la tabla 5, es importante mencionar que el desarrollo del proyecto es de dos meses y medio, las fechas de inicio y terminó son del 13 de septiembre de 2019 al 29 de noviembre de 2019.

Sprint	Historia de Usuario	Periodo del Sprint	Daily Sprint Meeting	Demo y Retrospectiva	Desarrollado por
1	HU1. Registrar usuario	13/09/2019 al 4/10/2019	9:30 a 9:45 hrs	4/10/2019	LZH
	HU2. Iniciar sesión				
	HU3. Buscar cliente				
2	HU4. Visualizar datos del cliente	7/10/2019 al 18/10/2019	9:30 a 9:45 hrs	18/10/2019	LZH
	HU5. Visualizar datos de contacto del cliente				
3	HU6. Cuestionario dinámico	21/10/2019 al 1/11/2019	9:30 a 9:45 hrs	1/11/2019	LZH
4	HU7. Visualizar respuestas del cuestionario	4/11/2019 al 15/11/2019	9:30 a 9:45 hrs	15/11/2019	LZH
	HU8. Firmar reporte de verificación				
5	HU9. Generar reporte de verificación	18/11/2019 al 29/11/2019	9:30 a 9:45 hrs	29/11/2019	LZH

**Tabla 5.** Tareas de cada Sprint

### 3.3 Sprint

#### 3.3.1 Sprint 1

El primer sprint dio inicio el 13 de septiembre del 2019 en el cual se diseñó y creó la base de datos VERIFICAPP en Firebase se muestra la estructura del árbol en la figura 6, también se diseñó y programó el action bar el cual contiene toda la aplicación tal cual se aprecia en la figura 7 y por último se diseñaron y programaron las pantallas registrar e iniciar sesión y buscar cliente como se observan en las figuras 8 a 10.

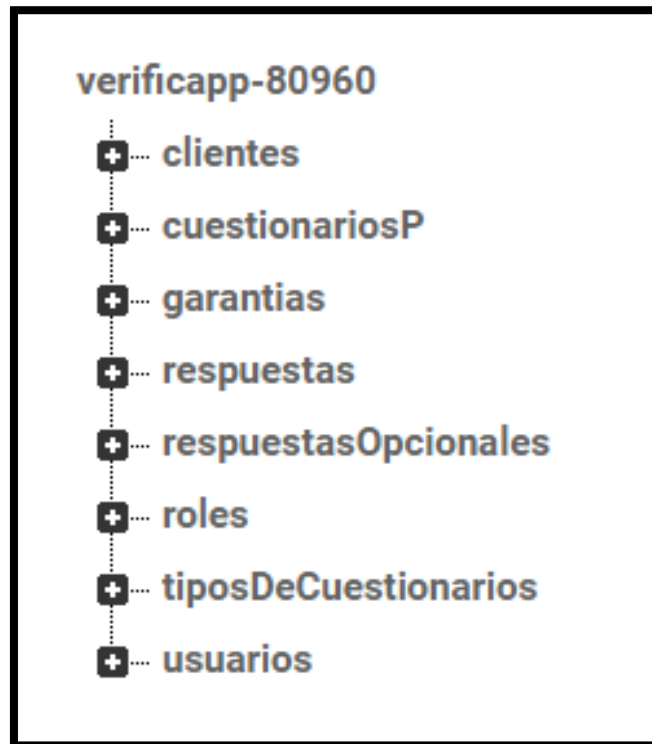


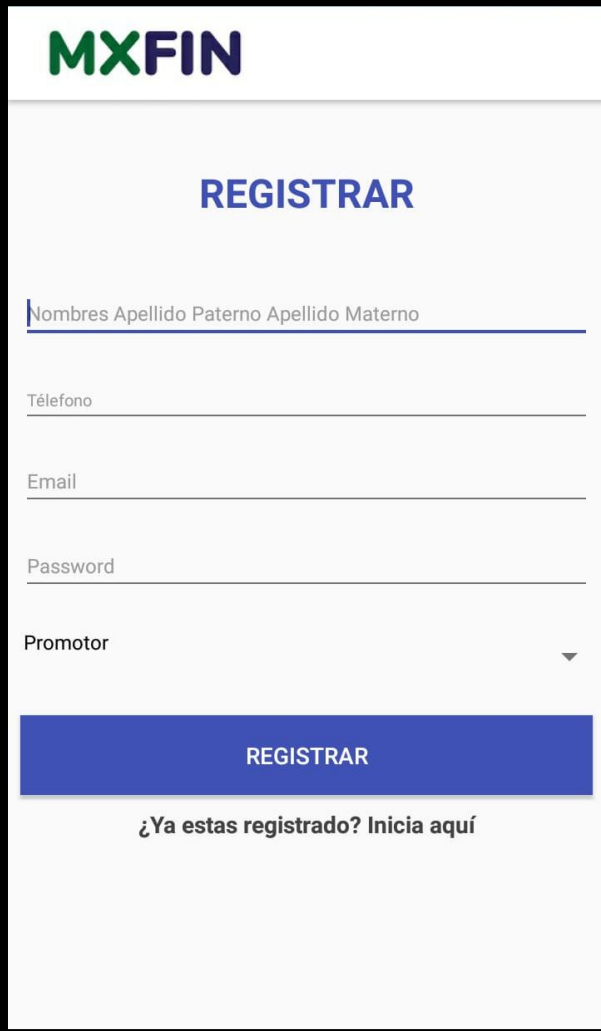
Figura 6. Base de datos VERIFICAPP



Figura 7. Action bar de la aplicación VERIFICAPP



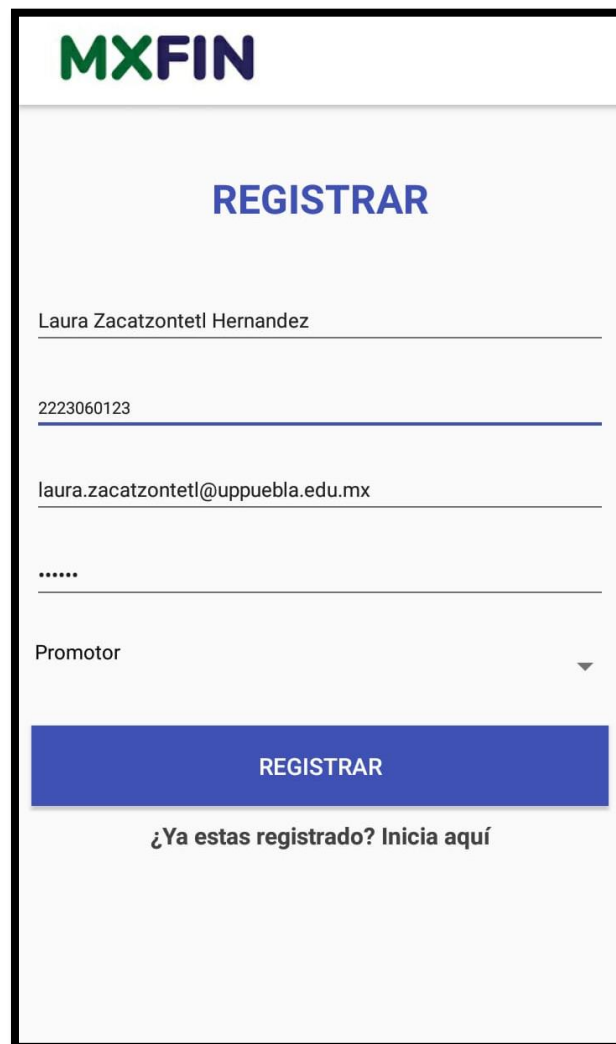
En la pantalla registrar como se muestra en la figura 8 se debe introducir el nombre completo, el número, email y finalmente introducir un password mayor a 6 caracteres debido a que todos los campos son validados y si el dato no es correcto no permitirá registrar al usuario.



The image shows a registration form for MXFIN. At the top left is the MXFIN logo. The title 'REGISTRAR' is centered in blue. Below the title are five input fields: 'Nombres Apellido Paterno Apellido Materno', 'Teléfono', 'Email', 'Password', and 'Promotor' (a dropdown menu). A blue 'REGISTRAR' button is positioned below the fields. At the bottom, there is a link that says '¿Ya estas registrado? Inicia aquí'.

**Figura 8.** Registro de VERIFICAPP

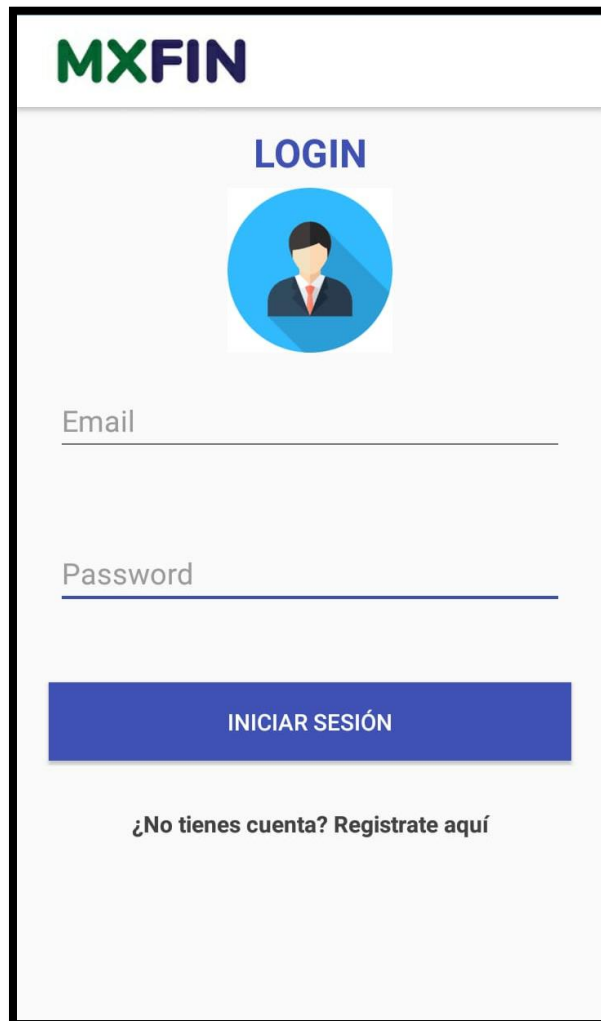
Para ingresar a la aplicación es necesario registrarse llenando el formulario que se muestra en la Figura 9, como se mencionó en la figura 8 los campos deben ser introducidos correctamente de no ser así no permitirá el registro del usuario.



The image shows a registration form for MXFIN. At the top left is the MXFIN logo. The title 'REGISTRAR' is centered in blue. Below the title are five input fields, each with a horizontal line underneath. The first field contains 'Laura Zacatzontetl Hernandez'. The second field contains '2223060123'. The third field contains 'laura.zacatzontetl@uppuebla.edu.mx'. The fourth field contains six dots. The fifth field is a dropdown menu labeled 'Promotor' with a downward arrow. Below the fields is a blue button with the text 'REGISTRAR' in white. At the bottom, there is a link that says '¿Ya estas registrado? Inicia aquí'.


**Figura 9.** Registro de VERIFICAPP con campos llenos.

La figura 10 muestra los campos a llenar para el inicio de sesión el cual solicita introducir el correo y contraseña ya registrados ambos datos son validados, de no introducir los datos correctamente no permitirá iniciar sesión.



**MXFIN**

**LOGIN**



Email

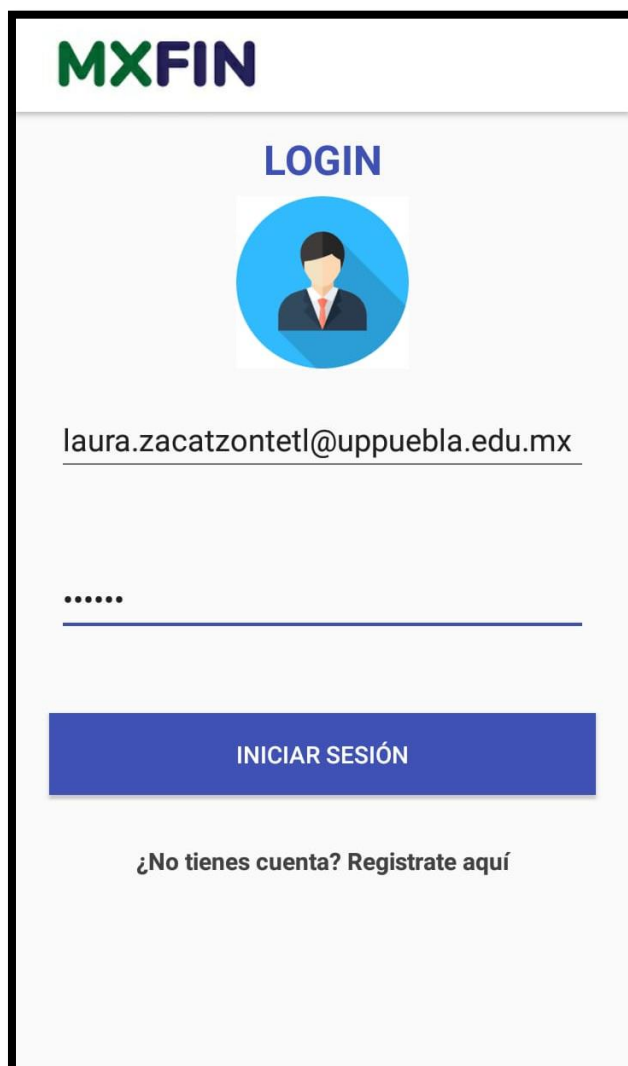
Password

**INICIAR SESIÓN**

[¿No tienes cuenta? Regístrate aquí](#)


**Figura 10.** Inicio de Sesión de VERIFICAPP.

Por ello es necesario introducir el usuario y contraseña correctamente en la pantalla de inicio de sesión de la aplicación VERIFICAPP como se aprecia en la figura 11.



**MXFIN**

**LOGIN**



laura.zacatzontetl@uppuebla.edu.mx

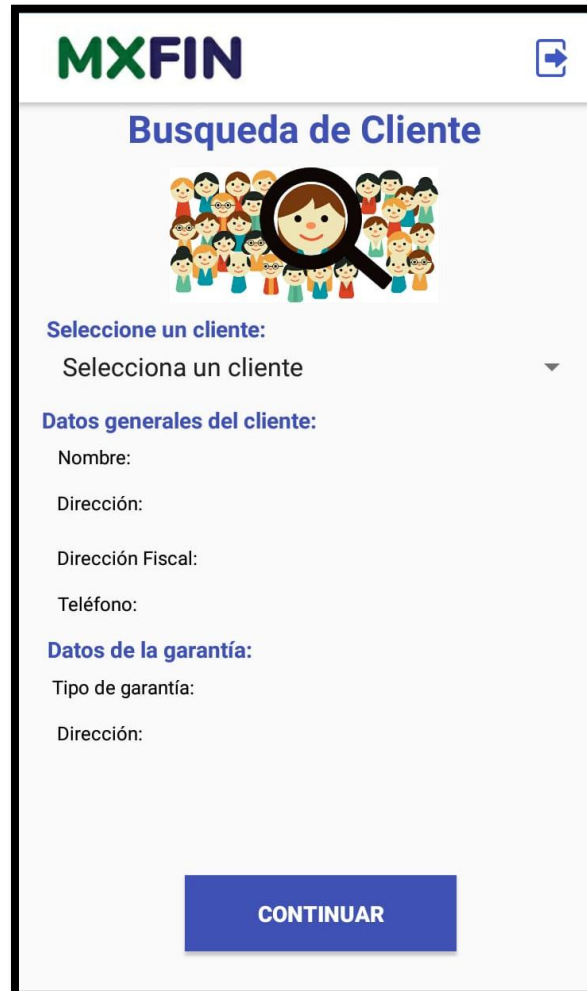
.....

**INICIAR SESIÓN**

[¿No tienes cuenta? Regístrate aquí](#)

**Figura 11.** Inicio de Sesión de VERIFICAPP

La pantalla búsqueda de cliente que se observa en la figura 12 permite buscar por folio o nombre para posteriormente seleccionar el botón continuar y responder el cuestionario dinámico.



**MXFIN**

**Busqueda de Cliente**

Seleccione un cliente:  
Selecciona un cliente

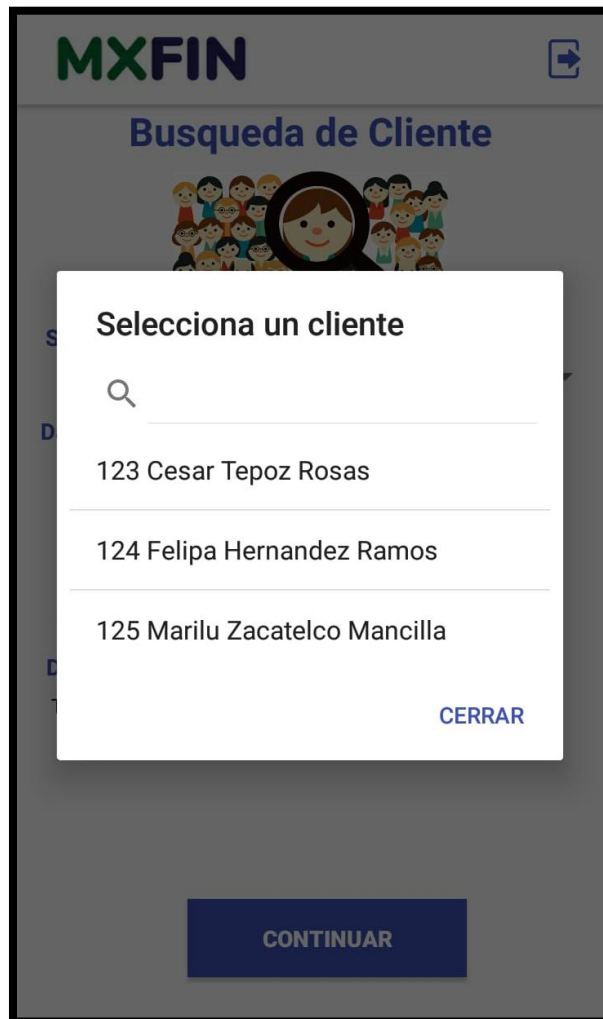
**Datos generales del cliente:**  
Nombre:  
Dirección:  
Dirección Fiscal:  
Teléfono:

**Datos de la garantía:**  
Tipo de garantía:  
Dirección:

**CONTINUAR**

**Figura 12.** Búsqueda de cliente.

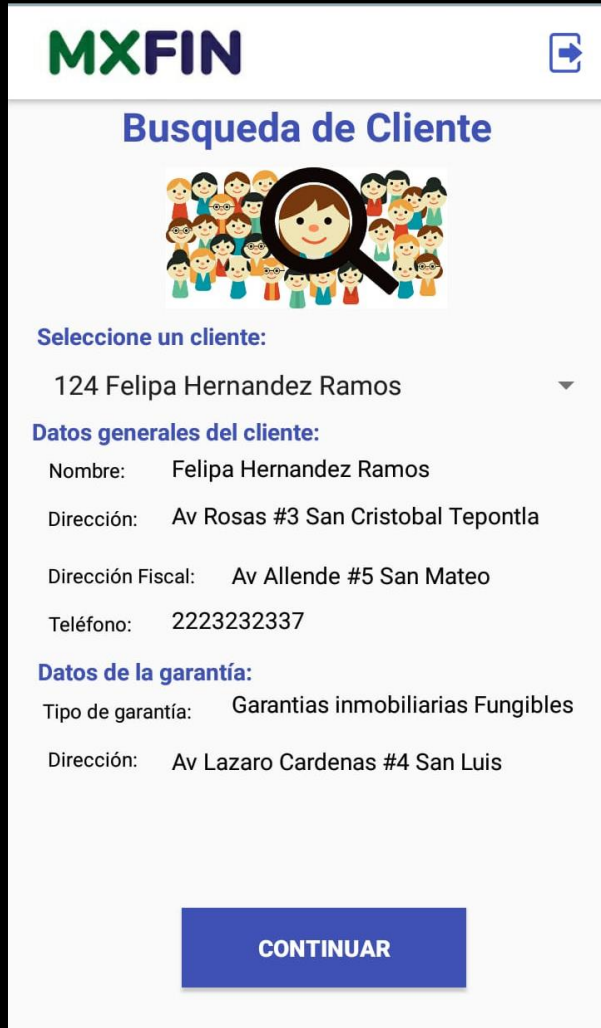
Como muestra en la figura 13, al tocar el buscador nos muestra la lista de clientes que tienen una verificación pendiente al elegir el cliente se visualizarán los datos generales del cliente así como también los datos de la garantía figura 14.




**Figura 13.** Buscador de cliente


### 3.3.2 Sprint 2

Este sprint comenzó el 7 de Octubre del 2019, durante el cual se programaron los módulos visualizar datos del cliente y visualizar datos de contacto. En la figura 14 se muestran los datos del cliente seleccionado como lo son datos generales del cliente y los datos de la garantía.



**MXFIN** 

## Busqueda de Cliente



**Seleccione un cliente:**

124 Felipa Hernandez Ramos ▼

**Datos generales del cliente:**

Nombre: Felipa Hernandez Ramos

Dirección: Av Rosas #3 San Cristobal Tepontla

Dirección Fiscal: Av Allende #5 San Mateo

Teléfono: 2223232337

**Datos de la garantía:**

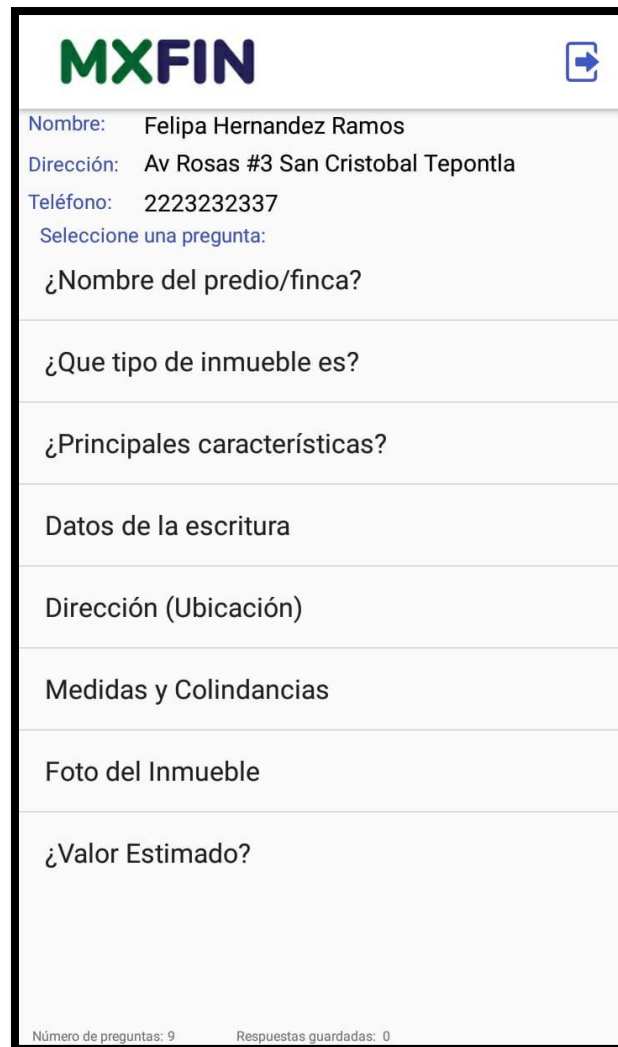
Tipo de garantía: Garantias inmobiliarias Fungibles

Dirección: Av Lazaro Cardenas #4 San Luis

**CONTINUAR**

**Figura 14.** Cliente seleccionado con sus respectivos datos generales y datos de la garantía.

La pantalla verificación que se muestra en la figura 15 permite visualizar datos de contacto del cliente, así como también muestra la interfaz de el cuestionario dinámico correspondiente al cliente, en la parte inferior se visualiza el número de preguntas y las respuestas guardadas.



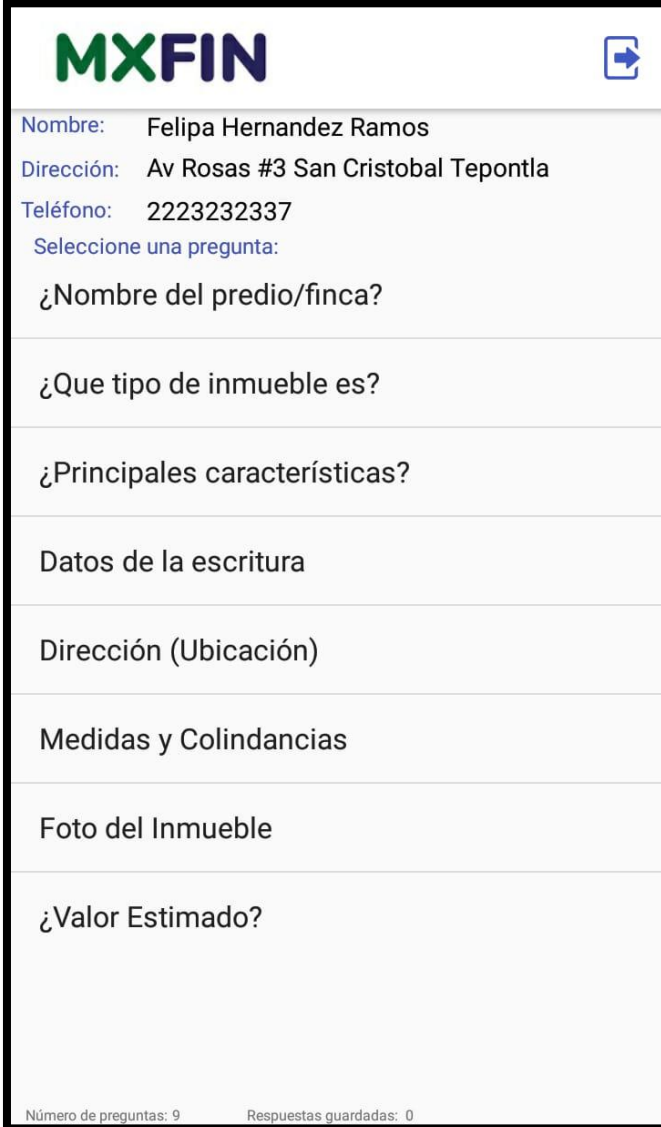
The screenshot displays the MXFIN application interface. At the top left is the MXFIN logo, and at the top right is a share icon. Below the header, the client's contact information is listed: Nombre: Felipa Hernandez Ramos, Dirección: Av Rosas #3 San Cristobal Tepontla, and Teléfono: 2223232337. A link 'Seleccione una pregunta:' is followed by a list of questions: '¿Nombre del predio/finca?', '¿Que tipo de inmueble es?', and '¿Principales características?'. Below these are several menu items: 'Datos de la escritura', 'Dirección (Ubicación)', 'Medidas y Colindancias', 'Foto del Inmueble', and '¿Valor Estimado?'. At the bottom, a status bar shows 'Número de preguntas: 9' and 'Respuestas guardadas: 0'.

**Figura 15.** Pantalla Verificación



### 3.3.3 Sprint 3

El sprint 3 dio inicio el 21 de octubre del 2019, en el cuál se desarrolló el módulo cuestionario dinámico esta pantalla que se observa en la figura 16 muestra los datos de contacto del cliente, cabe mencionar que se denomina cuestionario dinámico debido a que el cuestionario que se muestra debe corresponder al tipo de garantía a supervisar, así como también cada campo debe corresponder al tipo de pregunta, existen diversos tipos de preguntas los cuales son: campo de texto, respuesta seleccionable, georeferencia y numérico, estos campos se muestran conforme cambia el cuestionario y conforme al tipo de pregunta. En el anexo A se muestra el código que permite el funcionamiento del módulo cuestionario dinámico.



The screenshot displays the MXFIN application interface. At the top left is the MXFIN logo, and at the top right is a share icon. Below the header, the client's contact information is listed: Nombre: Felipa Hernandez Ramos, Dirección: Av Rosas #3 San Cristobal Tepontla, and Teléfono: 2223232337. A link 'Seleccione una pregunta:' is provided. The main content area contains a list of questions: '¿Nombre del predio/finca?', '¿Que tipo de inmueble es?', and '¿Principales características?'. Below these are several menu items: 'Datos de la escritura', 'Dirección (Ubicación)', 'Medidas y Colindancias', 'Foto del Inmueble', and '¿Valor Estimado?'. At the bottom, the status 'Número de preguntas: 9' and 'Respuestas guardadas: 0' is shown.

Figura 16. Pantalla Verificación.

La pantalla verificación consiste en un cuestionario dinámico que al seleccionar una pregunta lanzará una ventana emergente que permitirá introducir la respuesta correspondiente al campo, en este caso texto plano como lo muestra la figura 17.

The image shows a mobile application interface for MXFIN. At the top, the logo 'MXFIN' is displayed in green and blue. Below the logo, there is a list of user information: 'Nombre: Felipa Hernandez Ramos', 'Dirección: Av Rosas #3 San Cristobal Tepontla', and 'Teléfono: 2223232337'. A prompt 'Seleccione una pregunta:' is followed by a list of questions, with '¿Nombre del predio/finca?' selected. A modal dialog box is open over this question, containing the same question text, a text input field with the value 'Finca Los Blanca', and two buttons: 'CANCELAR' and 'OK'. At the bottom of the screen, there is a status bar showing 'Número de preguntas: 9' and 'Respuestas guardadas: 0'.

**Figura 17.** Pregunta de tipo texto abierto

El cuestionario también contiene un menú con respuestas seleccionables como se observa en la figura 18 que podrán ser seleccionadas y posteriormente presionar la opción ok para almacenarlas.

**MXFIN**

Nombre: Felipa Hernandez Ramos  
Dirección: Av Rosas #3 San Cristobal Tepontla

¿Que tipo de inmueble es?

- Bienes por accesión
- Bienes por analogía
- Bienes por destino
- Bienes por incorporación
- Bienes por naturaleza
- Bienes por representación

CANCELAR OK

Número de preguntas: 9 Respuestas guardadas: 1

**Figura 18.** Pregunta de tipo respuesta seleccionable.

En la figura 19 se muestra un campo de tipo texto plano del cuestionario dinámico que al presionar ok el dato será almacenado en la base de datos.

The image shows a mobile application interface for 'MXFIN'. At the top, the logo 'MXFIN' is displayed in green and blue, with a share icon to its right. Below the logo, the user's information is shown: 'Nombre: Felipa Hernandez Ramos', 'Dirección: Av Rosas #3 San Cristobal Tepontla', and 'Teléfono: 2223232337'. The main content area contains a questionnaire with the following questions and answers:

- Seleccione una pregunta:
  - ¿Nombre del predio/finca?  
Finca Los Blanca
  - ¿Que tipo de inmueble es?  
B
  - ¿Principales características?  
Finca perfectamente delimitada
  - ¿
  - D
  - D
- Medidas y Conducciones
- Foto del Inmueble
- ¿Valor Estimado?

At the bottom of the screen, it shows 'Número de preguntas: 9' and 'Respuestas guardadas: 2'. A modal dialog box is overlaid on the questionnaire, titled '¿Principales características?' and contains the text 'Finca perfectamente delimitada' with 'CANCELAR' and 'OK' buttons.

**Figura 19.** Pregunta de tipo texto abierto.

Al responder una pregunta de tipo foto como se muestra en la figura 20 se mostrará una ventana emergente que permitirá elegir la aplicación con la que se desea tomar la foto.

The screenshot shows the MXFIN mobile application interface. At the top, the logo 'MXFIN' is displayed in green and blue, with a share icon to its right. Below the logo, the user's personal information is listed: 'Nombre: Felipa Hernandez Ramos', 'Dirección: Av Rosas #3 San Cristobal Tepontla', and 'Teléfono: 2223232337'. A section titled 'Seleccione una pregunta:' contains three questions with their respective answers: '¿Nombre del predio/finca? Finca Los Blanca', '¿Que tipo de inmueble es? Bienes por naturaleza', and '¿Principales características? Finca perfectamente delimitada'. Below these questions is a section titled 'Datos de la escritura'. A camera action dialog is overlaid on the bottom half of the screen, featuring a camera icon and the text 'Completar acción con Cámara'. Two options are available: 'SOLO UNA VEZ' and 'SIEMPRE', both in green text. Below the dialog, there is a section titled 'Utiliza otra aplicación' with a search bar and a result for 'B612' with its icon.

**Figura 20.** Pregunta de tipo foto.

Al tomar la foto se abrirá la cámara y permitirá modificar las configuraciones de la cámara si así se desea como se observa en la figura 21.

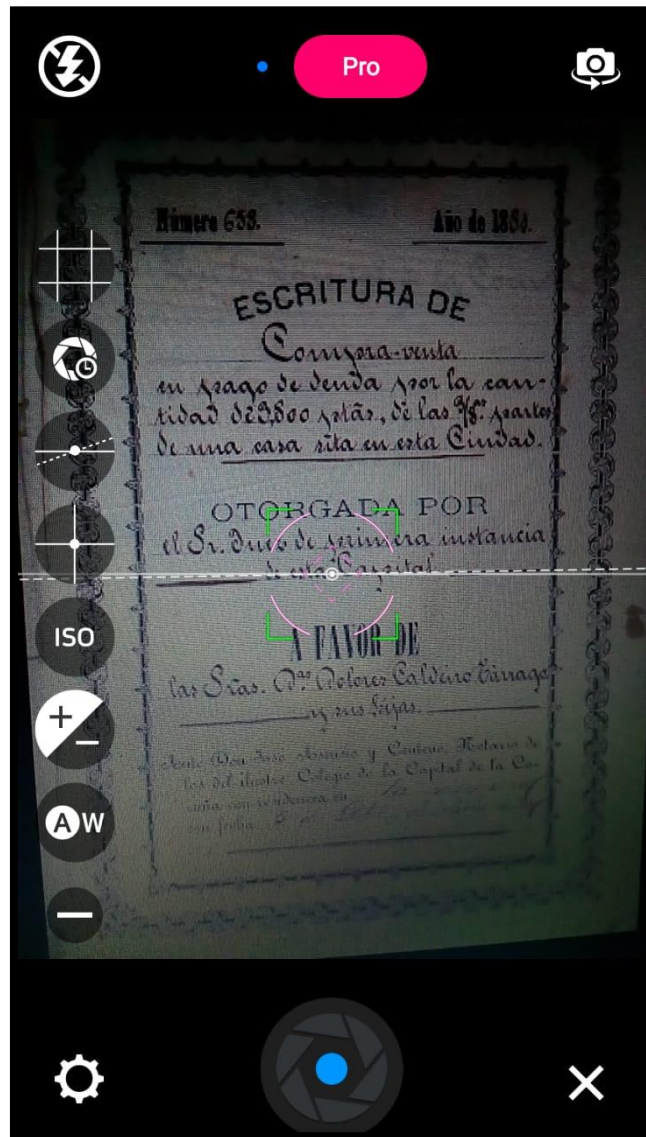


Figura 21. Cámara de la pregunta tipo foto.

Para confirmar la foto capturada que se desea agregar al reporte de verificación se debe presionar el botón aceptar que se aprecia a visualizar en la figura 22 para ser almacenada en la base de datos.

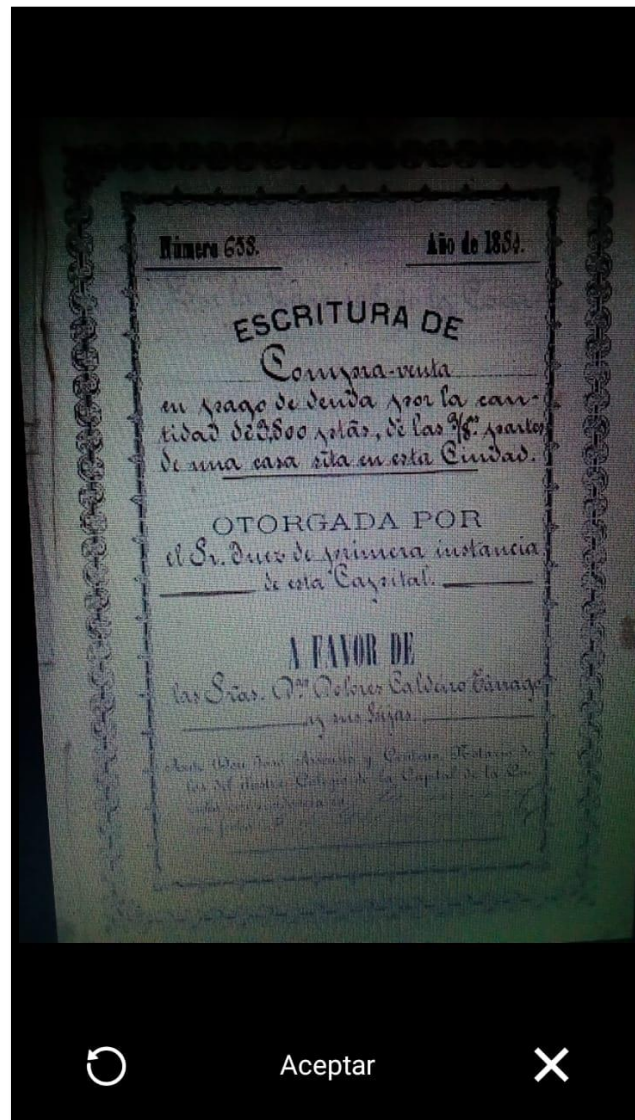
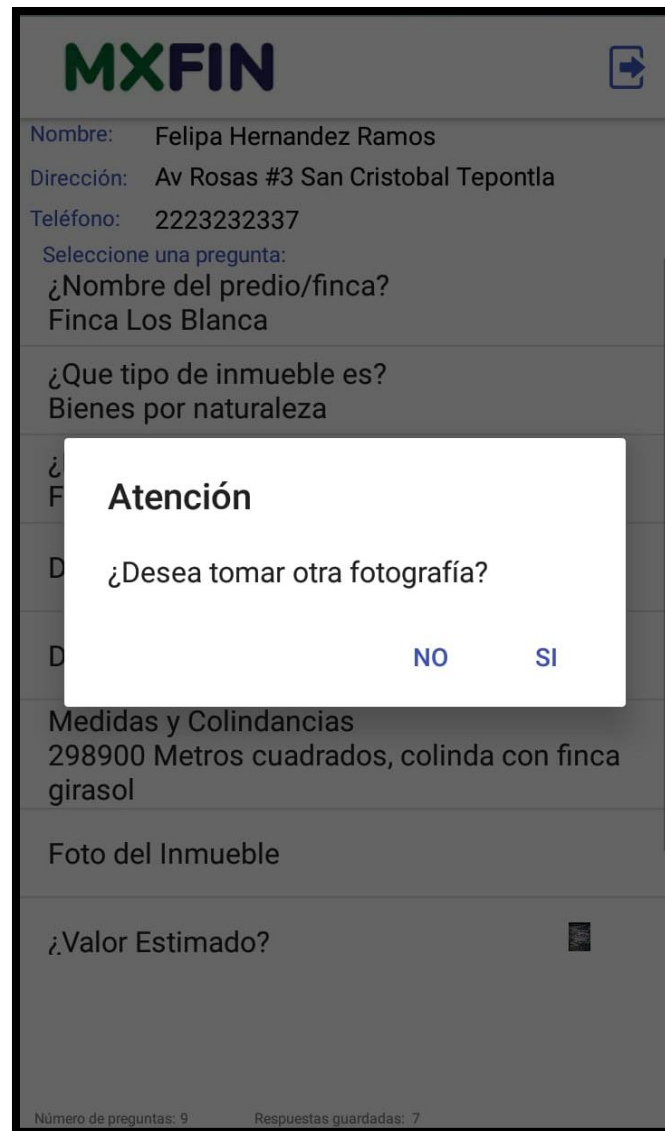


Figura 22. Fotografía tomada dentro de la ampliación VERIFICAPP.

Cuando se haya finalizado de tomar la fotografía nos mostrará un mensaje de alerta como se muestra en la figura 23 preguntando si se desea tomar otra fotografía, si se selecciona si se abrirá nuevamente la cámara para capturar la nueva foto y si se selecciona no, se cerrará la pequeña ventana.



**Figura 23.** Mensaje para volver a tomar otra foto.



Al responder una pregunta de tipo georeferencia, se mostrará la dirección completa así como también la latitud y longitud como se visualizar en la figura 24 en donde se encuentra el inmueble, en esta pregunta se permite agregar comentarios si así se desea.



The image shows a mobile application interface for MXFIN. At the top, the logo 'MXFIN' is displayed in green and blue. Below the logo, there is a form with the following fields: 'Nombre: Felipa Hernandez Ramos', 'Dirección: Av Rosas #3 San Cristobal Tepontla', and 'Teléfono: 2223232337'. A question is partially visible: '¿Nombre del predio (finca)?'. A white dialog box titled 'Ubicación' is overlaid on the screen. It features a red location pin icon on a map. Below the icon, the address is listed: 'Av. Miguel Alemán 116, Centro, 72760 Cholula, Pue., México Cholula Puebla México 72760'. The coordinates '19.061511666666668,-98.30717666666666' are shown below the address. There is a text input field labeled 'Comentario extra' with a blue underline. At the bottom of the dialog box are two buttons: 'CANCELAR' and 'OK'. Below the dialog box, the question '¿Valor Estimado?' is visible. At the very bottom of the screen, it says 'Número de preguntas: 9' and 'Respuestas guardadas: 4'.

**Figura 24.** Pregunta de tipo georreferencia.

También existe el tipo de dato alfanérico que consiste en la mezcla de números y letras como se muestra en la figura 25.

The image shows a mobile application interface for 'MXFIN'. At the top, the logo 'MXFIN' is displayed in green and blue. Below the logo, there is a form with the following fields: 'Nombre: Felipa Hernandez Ramos', 'Dirección: Av Rosas #3 San Cristobal Tepontla', and 'Teléfono: 2223232337'. A section titled 'Seleccione una pregunta:' contains a question '¿Nombre del predio/finca?' with the answer 'Finca Los Blanca'. Below this is another question '¿Que tipo de inmueble es?' with a partially visible answer 'B'. A modal dialog box is open in the center, titled 'Medidas y Colindancias', with the text '98900 Metros cuadrados, colinda con' and a blue underline. At the bottom of the dialog are two buttons: 'CANCELAR' and 'OK'. Below the dialog, the form continues with 'Foto del Inmueble' and '¿Valor Estimado?'. At the very bottom, there is a footer with 'Número de preguntas: 9' and 'Respuestas guardadas: 6'.

**Figura 25.** Pregunta de tipo texto abierto.

Durante la la verificación se pueden llegar a tener varias preguntas que sean del tipo foto por lo que se tomarán el número de fotos que se desee tanto del inmueble como de cualquier otra evidencia observar la toma de una nueva fotografía en la figura 26.



Figura 26. Pregunta de tipo foto.

La aplicación también permite introducir números con punto decimal, como se puede observar en la figura 27.

The image shows a mobile application interface for 'MXFIN'. The background is a form with the following visible text: 'Nombre: Felipa Hernandez Ramos', 'Dirección: Av Rosas #3 San Cristobal Tepontla', 'Teléfono: 2223232337', 'Seleccione una pregunta: Bienes por naturaleza', '¿Principales características? Finca perfectamente delimitada', and '¿Valor Estimado?'. A white dialog box is overlaid on the screen, titled '¿Valor Estimado?'. It contains a text input field with the number '20000000' and a blue cursor. Below the input field are two buttons: 'CANCELAR' and 'OK'. At the bottom of the application screen, there are two buttons: 'MOSTRAR RESPUESTAS' and 'FIRMAR DOCUMENTO'. At the very bottom, it says 'Número de preguntas: 9' and 'Respuestas guardadas: 10'.

**Figura 27.** Pregunta de tipo decimal.

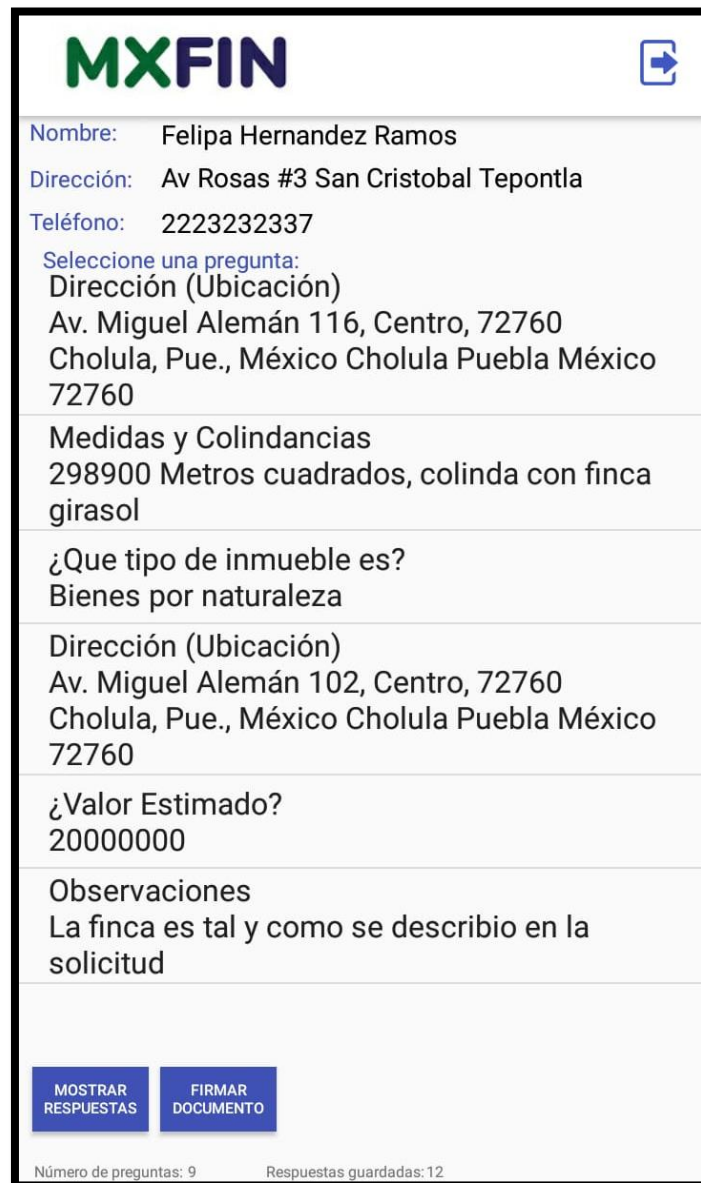
Como ya se ha mencionado con anterioridad pueden llegar a existir preguntas del mismo tipo en este caso se observa en la figura 28 otra pregunta de texto plano.

The image shows a mobile application interface for 'MXFIN'. At the top, the logo 'MXFIN' is displayed in green and blue. Below the logo, there is a form with the following fields: 'Nombre: Felipa Hernandez Ramos', 'Dirección: Av Rosas #3 San Cristobal Tepontla', and 'Teléfono: 2223232337'. A dropdown menu is open, showing 'Selecciona una pregunta:' with the selected option 'Bienes por naturaleza'. Below this, there is a question: '¿Principales características?' with the answer 'Finca perfectamente delimitada'. A white modal dialog box titled 'Observaciones' is overlaid on the screen, containing a text input field with the text 'La finca es tal y como se describió en' and two buttons: 'CANCELAR' and 'OK'. Below the modal, there is another question: '¿Valor Estimado?' with the answer '20000000'. At the bottom of the screen, there are two buttons: 'MOSTRAR RESPUESTAS' and 'FIRMAR DOCUMENTO'. At the very bottom, it says 'Número de preguntas: 9' and 'Respuestas guardadas: 11'.

Figura 28. Pregunta de tipo texto abierto

### 3.3.5 Sprint 4

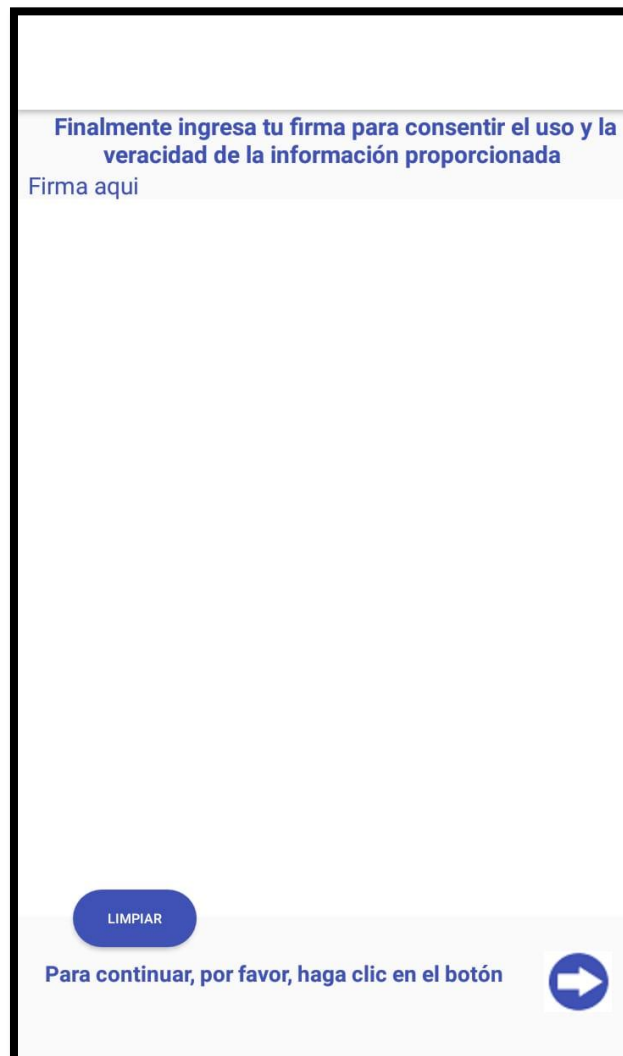
Este sprint comenzó 4 de noviembre del 2019, durante el cual se programó la funcionalidad visualizar respuestas y firmar reporte de verificación. Conforme el usuario vaya respondiendo el cuestionario las respuestas se mostrarán debajo de cada pregunta como se visualiza en la figura 29, al tener almacenadas el mismo número de preguntas como de respuestas guardadas, se nos habilitará el botón mostrar respuestas y firmar documento.



The screenshot displays the MXFIN application interface. At the top left is the MXFIN logo, and at the top right is a share icon. Below the header, the user's information is listed: Nombre: Felipa Hernandez Ramos, Dirección: Av Rosas #3 San Cristobal Tepontla, and Teléfono: 2223232337. A section titled 'Seleccione una pregunta:' contains a list of questions and their corresponding answers. The first question is 'Dirección (Ubicación)' with the answer 'Av. Miguel Alemán 116, Centro, 72760 Cholula, Pue., México Cholula Puebla México 72760'. The second question is 'Medidas y Colindancias' with the answer '298900 Metros cuadrados, colinda con finca girasol'. The third question is '¿Que tipo de inmueble es?' with the answer 'Bienes por naturaleza'. The fourth question is 'Dirección (Ubicación)' with the answer 'Av. Miguel Alemán 102, Centro, 72760 Cholula, Pue., México Cholula Puebla México 72760'. The fifth question is '¿Valor Estimado?' with the answer '20000000'. The final section is 'Observaciones' with the answer 'La finca es tal y como se describio en la solicitud'. At the bottom, there are two blue buttons: 'MOSTRAR RESPUESTAS' and 'FIRMAR DOCUMENTO'. Below the buttons, the status 'Número de preguntas: 9' and 'Respuestas guardadas: 12' is displayed.

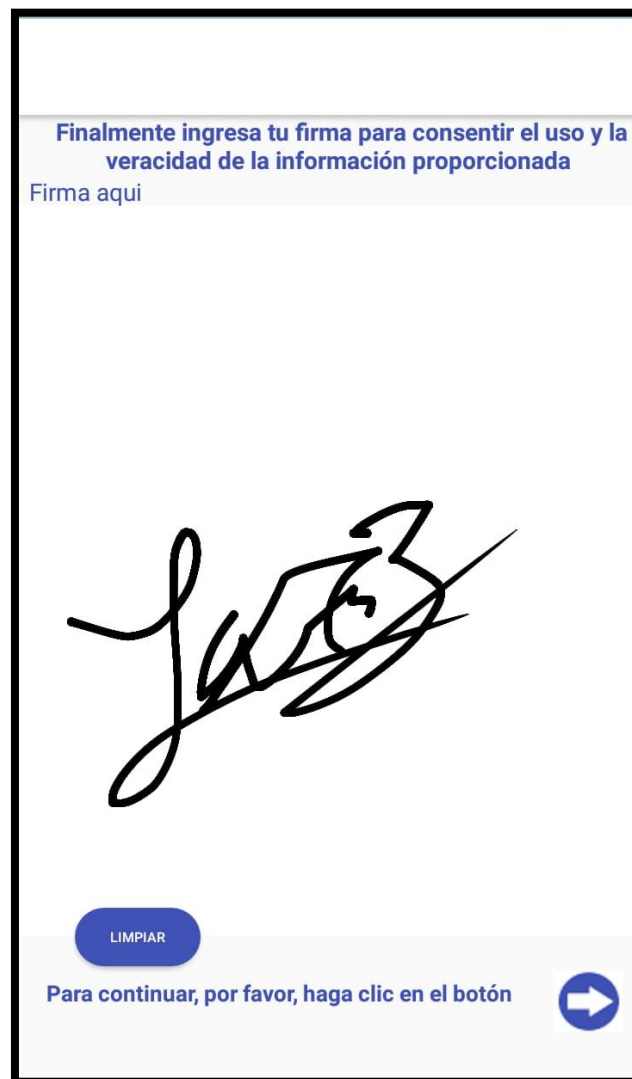
Figura 29. Visualización de respuestas.

Al presionar el botón firmar documento se mostrará la pantalla de la figura 30 en la cual se podrá firmar al dibujar con el dedo índice u cualquier otro dedo, existe un botón limpiar por si se desea repetir la firma, se requiere la firma para consentir la veracidad de los datos y para continuar con la generación del reporte de verificación, es necesario presionar la fecha azul para continuar



**Figura 30.** Pantalla firmar documento

La firma puede intentarse todas las veces que lo deseen, pues al presionar el botón limpiar que se observa en la figura 31 eliminará todo el contenido de la pantalla para reintentar la firma.



**Figura 31.** Pantalla firmar documento con firma



Durante este sprint también se desarrolló el módulo finalizar, el botón finalizar reporte se visualiza en la figura 32 cambia el estado del cuestionario a cuestionario finalizado, no permitiendo volver a ser contestado por ningún usuario.

**MXFIN**

Nombre: Felipa Hernandez Ramos  
Dirección: Av Rosas #3 San Cristobal Tepontla  
Teléfono: 2223232337

Seleccione una pregunta:  
Dirección (Ubicación)  
Av. Miguel Alemán 116, Centro, 72760  
Cholula, Pue., México Cholula Puebla México  
72760

Medidas y Colindancias  
298900 Metros cuadrados, colinda con finca girasol

¿Que tipo de inmueble es?  
Bienes por naturaleza

Dirección (Ubicación)  
Av. Miguel Alemán 102, Centro, 72760  
Cholula, Pue., México Cholula Puebla México  
72760

¿Valor Estimado?  
20000000

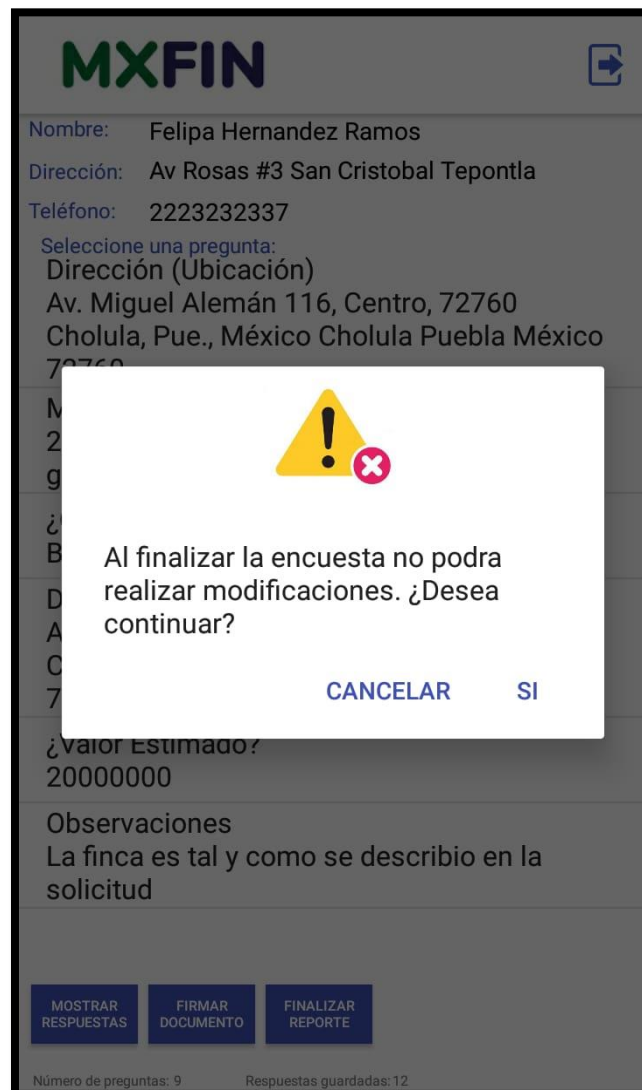
Observaciones  
La finca es tal y como se describio en la solicitud

MOSTRAR RESPUESTAS   FIRMAR DOCUMENTO   FINALIZAR REPORTE

Número de preguntas: 9   Respuestas guardadas: 12

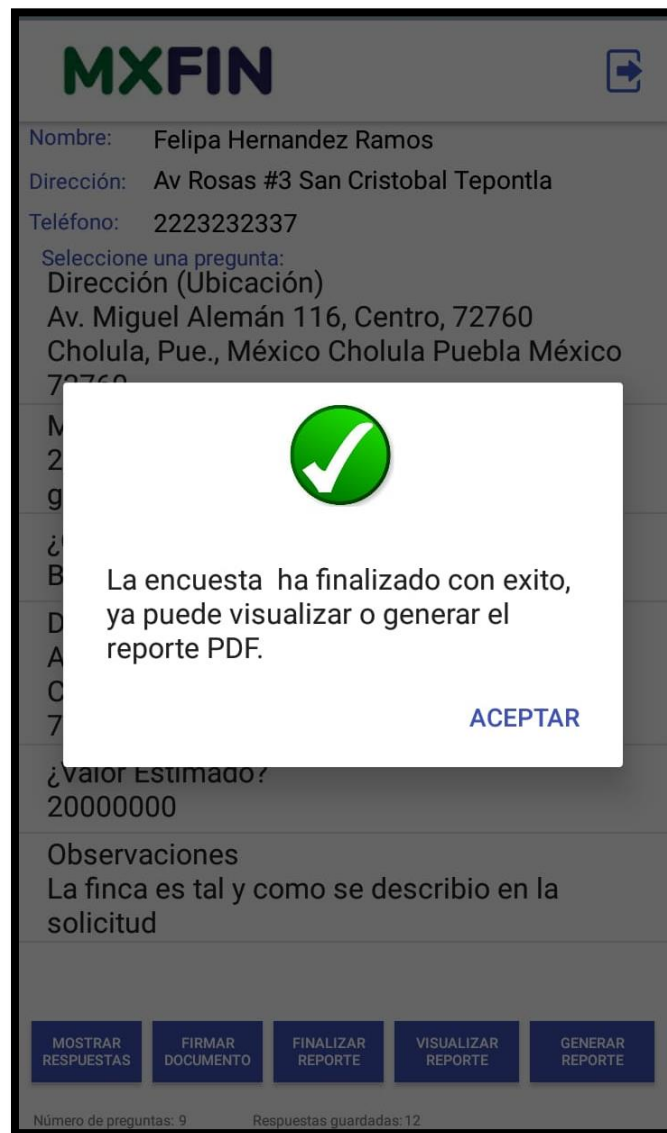
**Figura 32.** Pantalla con el botón finalizar reporte habilitado

En la figura 33 se muestra un mensaje que indica que no podrá modificar una respuesta al finalizar la encuesta.



**Figura 33.** Mensaje de precaución.

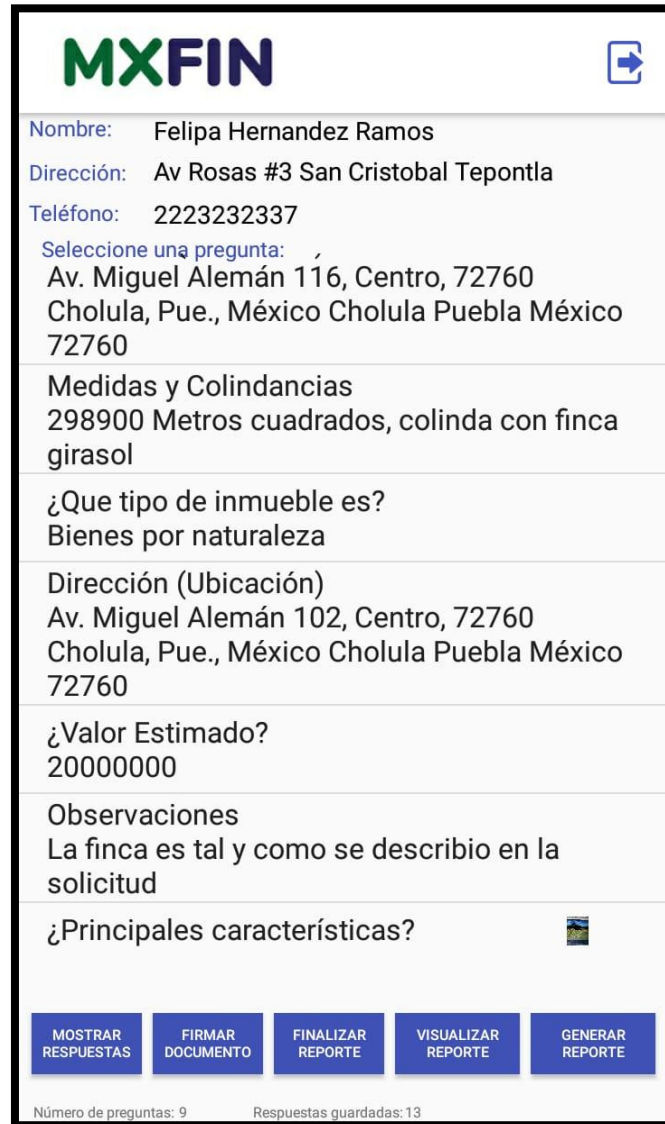
Al dar clic en la opción si los botones de visualizar reporte y generar reporte se habilitarán mostrando un mensaje como se observa en la figura 34.



**Figura 34.** Mensaje de precaución.

### 3.3.7 Sprint 5

El sprint siete comenzó el 18 de noviembre del 2019, en el cual se desarrolló el módulo visualizar pdf y generar reporte en pdf, el cual consiste en que al presionar el botón finalizar reporte nos habilitará el botón visualizar reporte y generar reporte como se muestra en la figura 35.



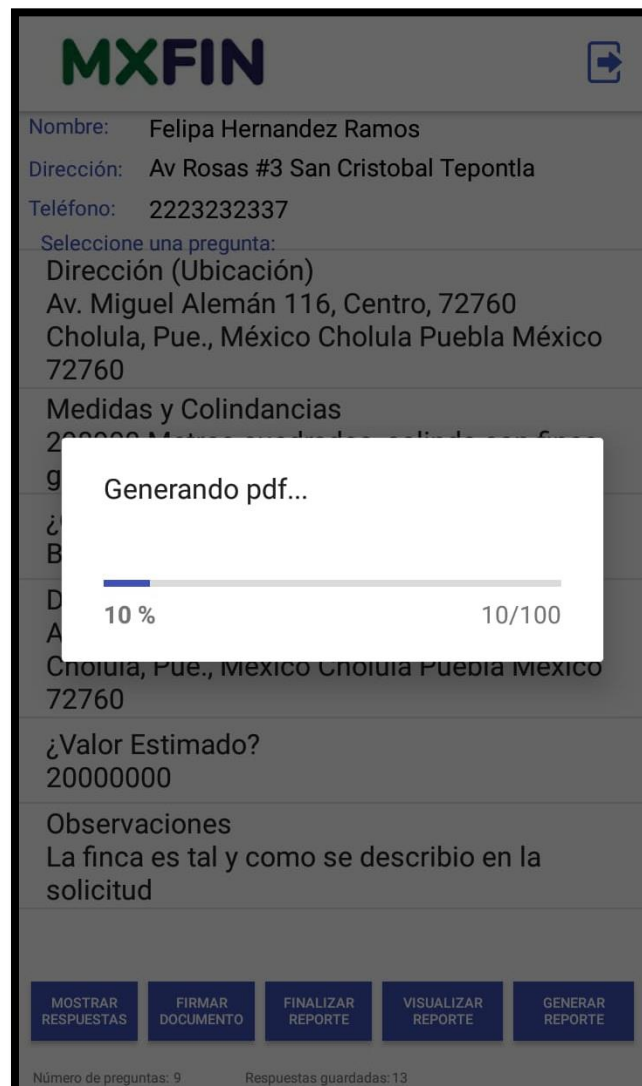
The screenshot displays the MXFIN application interface. At the top left is the MXFIN logo, and at the top right is a share icon. The form contains the following information:

- Nombre:** Felipa Hernandez Ramos
- Dirección:** Av Rosas #3 San Cristobal Tepontla
- Teléfono:** 2223232337
- Seleccione una pregunta:** Av. Miguel Alemán 116, Centro, 72760 Cholula, Pue., México Cholula Puebla México 72760
- Medidas y Colindancias:** 298900 Metros cuadrados, colinda con finca girasol
- ¿Que tipo de inmueble es?** Bienes por naturaleza
- Dirección (Ubicación):** Av. Miguel Alemán 102, Centro, 72760 Cholula, Pue., México Cholula Puebla México 72760
- ¿Valor Estimado?** 20000000
- Observaciones:** La finca es tal y como se describio en la solicitud
- ¿Principales características?** (with a small image icon)

At the bottom, there are five blue buttons: MOSTRAR RESPUESTAS, FIRMAR DOCUMENTO, FINALIZAR REPORTE, VISUALIZAR REPORTE, and GENERAR REPORTE. Below the buttons, it shows "Número de preguntas: 9" and "Respuestas guardadas: 13".

Figura 35. Botones visualizar reporte y generar reporte habilitados.

Al presionar el botón visualizar reporte nos cargará inicialmente una barra de progreso como se observa en la figura 36.



**Figura 36.** Barra de progreso al generar el pdf.

Para posteriormente mostrar el pdf dentro de la aplicación tal y como se visualiza en las figuras 37, 38 y 39. Al generar el pdf se visualizarán los datos del verificador, los datos del cliente y los datos de la garantía, así como también en una tabla las preguntas y respuestas contestadas como se observa en la figura 37.

**MXFIN**

**ABASTECEDORA DE SERVICIOS DE PUEBLA, S.A. DE C.V., SOFOM, E.N.R**  
**Reporte de Campo**  
**Generado: 13-11-2019**

El propósito de la visita de campo es confirmar y corroborar las características y estado de la garantía o el inmueble.

**Datos del Verificador:**  
Nombre: Laura Zacatzontel Hernandez  
Teléfono: 2223060126

**Datos del Cliente**  
Nombre: Felipa Hernandez Ramos  
Dirección: Av Rosas #3 San Cristobal Tepontla  
Teléfono: 2223232337

**Datos de la Garantía o Inmueble:**  
Tipo de Garantía: Garantías inmobiliarias Fungibles  
Dirección: Av Lazaro Cardenas #4 San Luis

Pregunta	Respuesta
¿Principales características?	Finca perfectamente delimitada
¿Que tipo de inmueble es?	Bienes por naturaleza
Observaciones	La finca es tal y como de describo en la solicitud
¿Valor Estimado?	20000000
¿Nombre del predio/finca?	Finca Los Blanca
Dirección (Ubicación)	Av Benito Juarez 55a, San Cristobal Tepontla, 72765 Cholula, Pue., México Cholula Puebla México 72765
Medidas y Colindancias	298900 Metros cuadrados, colinda con finca girasol

Enlaces

**Figura 37.** Pdf dentro de la aplicación VERIFICA.

En la figura 38 se puede visualizar las fotos que se capturaron durante el cuestionario dinámico, dichas fotografías han sido agregadas al reporte de verificación.

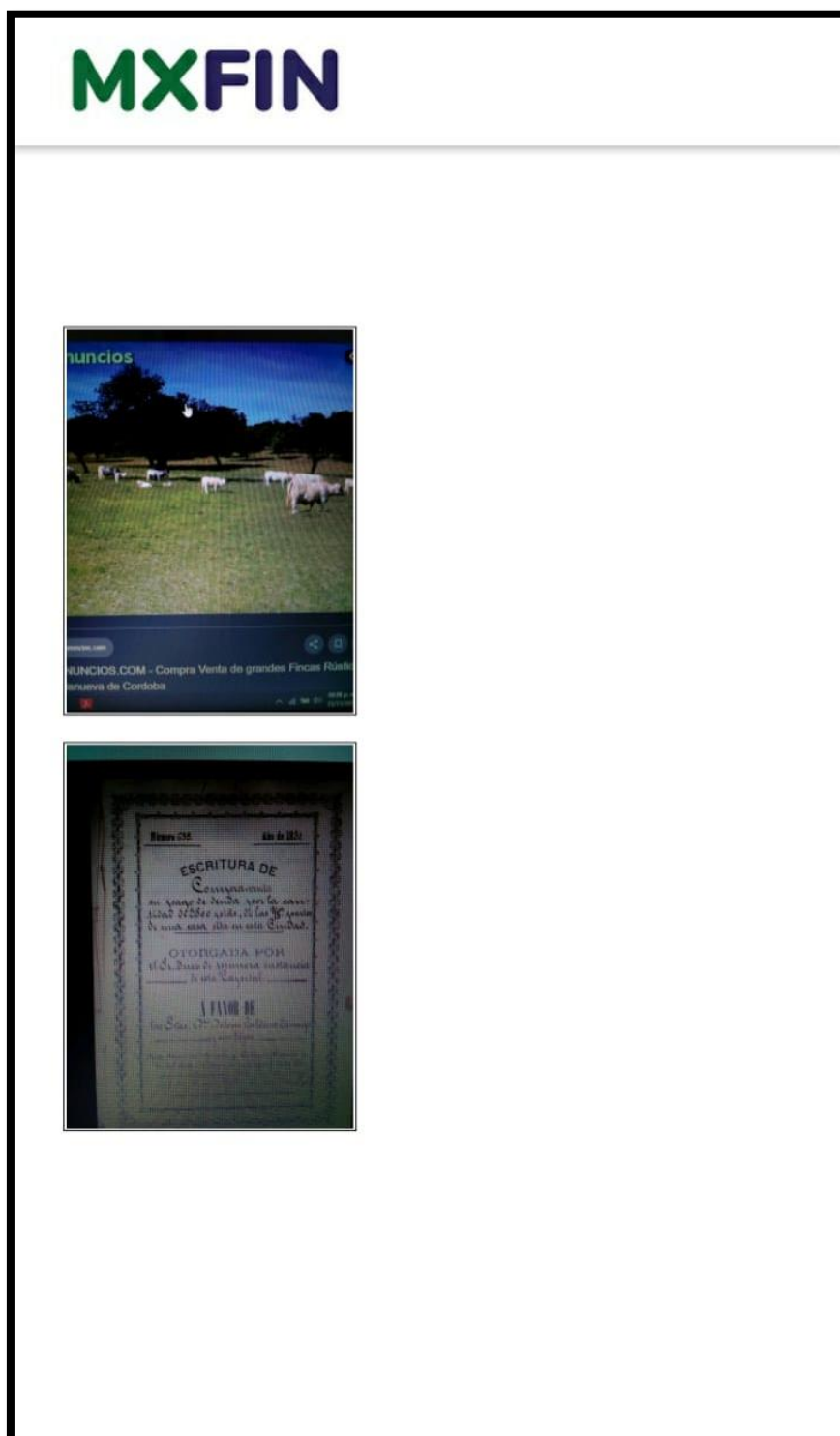
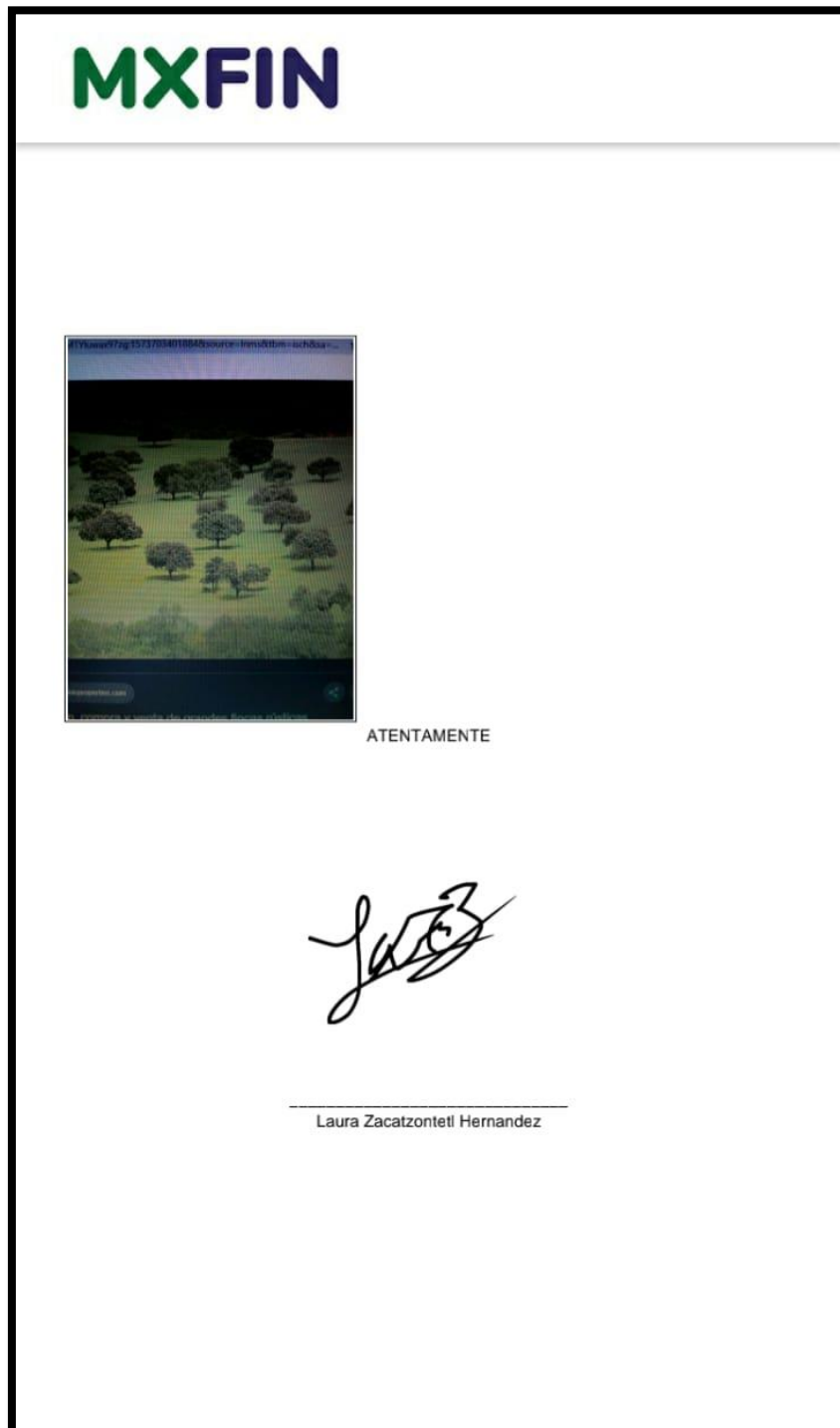


Figura 38. Pdf dentro de la aplicación VERIFICAPP.

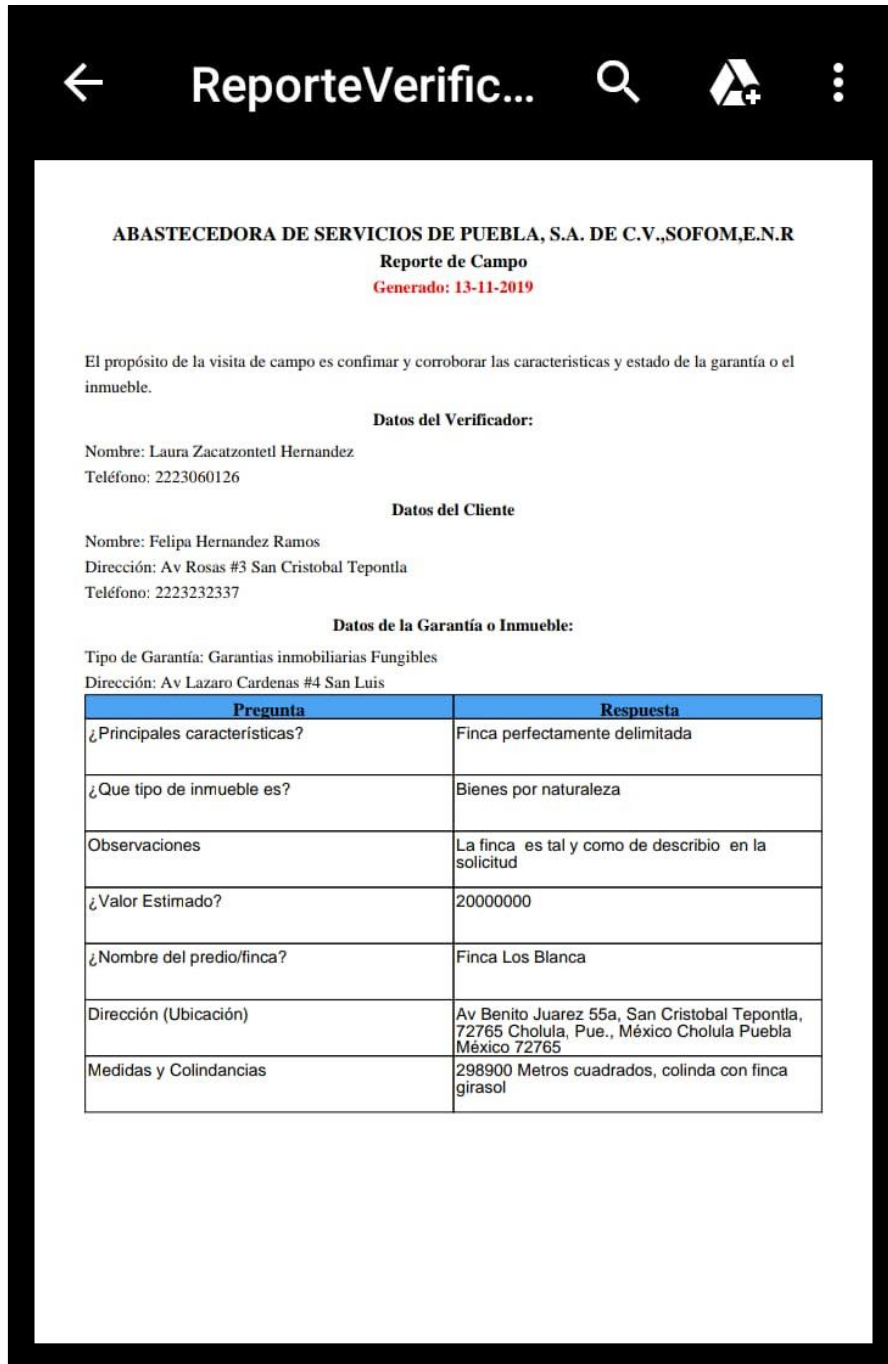
Por último se muestra la firma del verificador del inmueble para dar veracidad al reporte generado como se observa en la figura 39.



**Figura 39.** Pdf dentro de la aplicación VERIFICAPP.

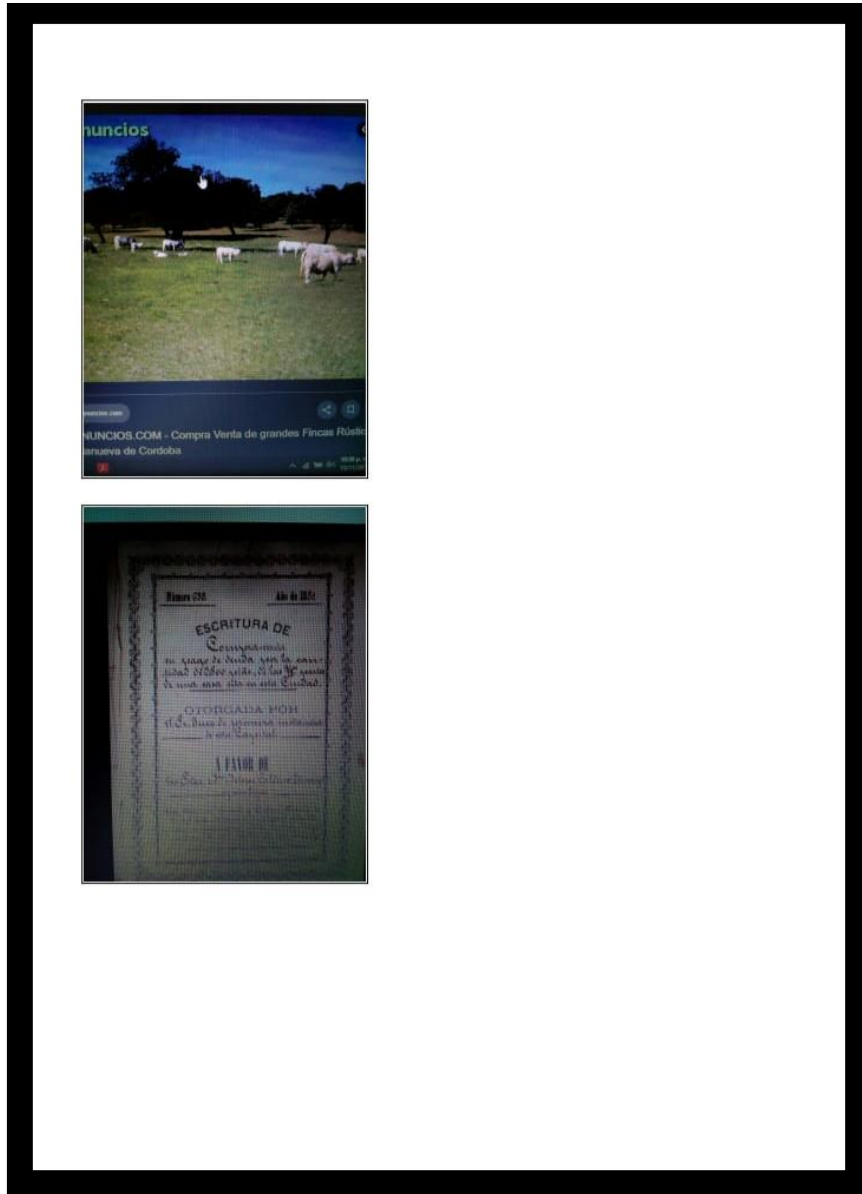


Al presionar el botón generar reporte nos abrirá automáticamente un lector de pdf con el recién pdf creado como se visualiza en la figura 40 el cual puede ser descargado nuevamente si así se desea con las opciones que ofrece la tercera aplicación.



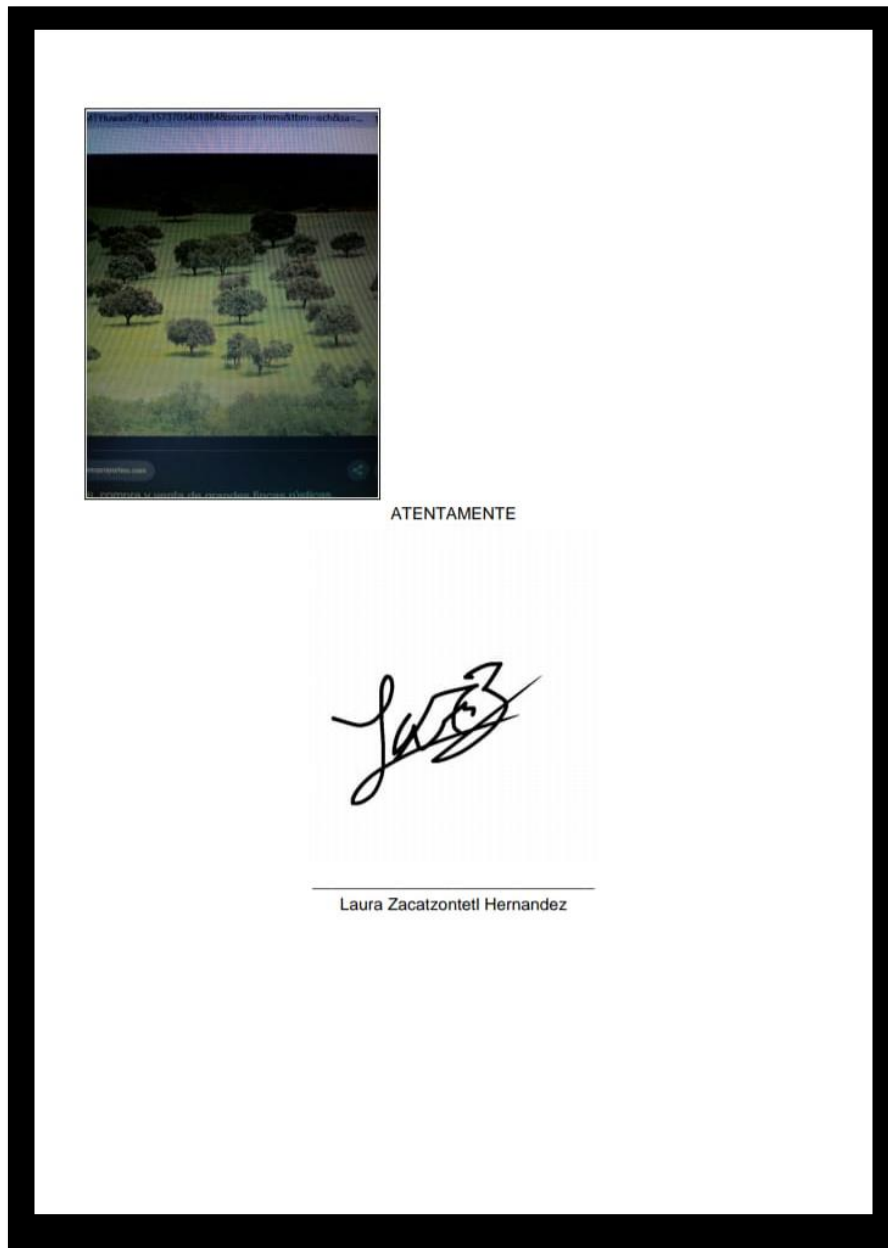
**Figura 40.** Reporte pdf en aplicación externa.

Cabe mencionar que dicho pdf es guardado automáticamente en la carpeta documentos de la memoria interna del dispositivo, el pdf generado también incluye todas las evidencias que se muestran en la figura 41 almacenadas durante la contestación del cuestionario dinámico.



**Figura 41.** Reporte pdf en aplicación externa.

Finalmente se muestra la firma del verificador así como su nombre, esto se aprecia en la figura 42.



**Figura 42.** Reporte pdf en aplicación externa.

### 3.4 Sprint Backlog

En la etapa de sprint backlog se listan las tareas en la tabla 5 que se llevaron a cabo para implementar las historias. Cabe mencionar que estas historias fueron mencionadas en el punto 3.1. para mayor información a detalle.

Id	Historia de Usuario	Tarea
1	HU1. Registrar usuario	Diseñar la base de datos en Firebase
		Crear la base de datos en Firebase
		Diseñar el action bar de la aplicación
		Programar el action bar
		Diseñar el formulario de registro de usuario
		Programar formulario registrar usuario
2	HU2. Iniciar sesión	Diseñar el formulario de inicio de sesión
		Programar formulario iniciar sesión
3	HU3. Buscar cliente	Crear las consultas de información de los clientes y de las garantías
		Diseñar la interfaz búsqueda cliente
		Programar la búsqueda del cliente
4	HU4. Visualizar datos del cliente	Programar la visualización datos del cliente
5	HU5. Visualizar datos de contacto	Programar la visualización datos de contacto del cliente
6	HU6. Cuestionario dinámico	Diseñar la interfaz Verificación de la aplicación del crédito
		Programar la interfaz verificación de la aplicación del crédito
		Programar el cuestionario dinámico
7	HU7. Visualizar respuestas	Diseñar la interfaz firmar documento
		Programar la firma digital del documento
		Programar el botón finalizar reporte
8	HU8. Firmar reporte de verificación	Diseñar la interfaz visualizar reporte
		Programar la visualización del pdf
		Diseñar y programar el reporte de VERIFICAPP
9	HU9. Generar reporte de verificación	Programar la generación del pdf
		Diseñar el botón cerrar Sesión
		Programar la función cerrar sesión

**Tabla 6.** Tareas para implementar las historias.

### 3.5 Daily sprint meeting

Durante esta etapa se realizaron reuniones de un lapso de tiempo menor a 15 minutos para analizar el avance del proyecto y aclarar cualquier duda que pueda surgir durante el desarrollo de la aplicación. La tabla 7 muestra a detalle las reuniones diarias que se realizaron durante el mes de septiembre.

SEPTIEMBRE				
ID	Fecha	Horario	Description del Daily Sprint meeting	Asistió
1	13/09/2019	9:30 a 9:45 hrs	Se especificaron más a detalle los requerimientos de la aplicación y se mostró el diseño de las pantallas.	ATL, GOB, LZH
2	17/09/2019	9:30 a 9:45 hrs	Se diseño la base de datos junto con al Scrum master durante esta reunión también se habló sobre la cuenta en donde se manejaría la base de datos en Firebase.	LZH, GOB
3	18/09/2019		Durante esta reunión se hizo entrega del logotipo y de los colores de diseño con las que se deseaba implementar la aplicación.	LZH, GOB
4	19/09/2019	9:30 a 9:45 hrs	Se dio a conocer la nueva base de datos generada en una cuenta externa ala de la empresa por cuestiones de prueba	LZH, GOB
5	20/09/2019		Se habló sobre la necesidad de un equipo más sofisticado y de mayor calidad de procesador que una laptop.	LZH, GOB
6	23/09/2019	9:30 a 9:45 hrs	El scrum master hizo entrega de una computadora con mayor capacidad de procesador para el desarrollo de la aplicación	LZH, GOB
7	24/09/2019	9:30 a 9:45 hrs	Durante esta reunión se mostró el diseño de las pantallas iniciar sesión, registrar y búsqueda de cliente.	LZH, GOB
8	25/09/2019	9:30 a 9:45 hrs	Se hizo la demostración de funcionalidad de la pantalla registrar usuario, durante la reunión el scrum master solicitó agregar la validación de cada uno de los campos.	LZH, GOB
9	26/09/2019	9:30 a 9:45 hrs	Se mostró la funcionalidad de la pantalla iniciar sesión, durante la reunión el scrum master solicitó agregar la validación de cada uno de los campos.	LZH, GOB
10	27/09/2019	9:30 a 9:45 hrs	Se mostró la funcionalidad de la pantalla buscar cliente.	LZH, GOB

11	30/09/2019	9:30 a 9:45 hrs	El scrum master pidió se realizarán pruebas con diversos usuarios de las pantallas ya desarrolladas.	LZH, GOB
----	------------	-----------------	--	----------

**Tabla 7.** Description del Daily Sprint meeting.

La tabla 8 muestra a detalle las reuniones diarias que se realizaron durante el mes de octubre.

<b>OCTUBRE</b>				
<b>ID</b>	<b>Fecha</b>	<b>Horario</b>	<b>Description del Daily Sprint meeting</b>	<b>Asistió</b>
1	1/10/2019	9:30 a 9:45 hrs	Se mencionaron la lista de errores que surgieron durante las pruebas, el scrum master solicitó corregir los errores.	LZH, GOB
2	2/10/2019	9:30 a 9:45 hrs	Se mostraron las nuevas validaciones implementadas en la pantalla registrar.	LZH, GOB
3	3/10/2019	9:30 a 9:45 hrs	Se mostraron las nuevas validaciones implementadas en la pantalla registrar.	LZH, GOB
4	4/10/2019	9:30 a 9:45 hrs	El scrum master solicitó agregar nuevos campos a mostrar en la pantalla búsqueda de cliente.	LZH, GOB
5	7/10/2019	9:30 a 9:45 hrs	Se mostró la modificación de diseño de la pantalla búsqueda de cliente al agregar los nuevos campos solicitados.	LZH, GOB
6	8/10/2019	9:30 a 9:45 hrs	El scrum master estableció los campos de contacto a mostrar en la pantalla cuestionario dinámico.	LZH, GOB
7	9/10/2019	9:30 a 9:45 hrs	Se mostró el primer diseño de la pantalla cuestionario dinámico.	LZH, GOB
8	10/10/2019	9:30 a 9:45 hrs	El scrum master solicitó modificaciones en el diseño de la pantalla cuestionario dinámico.	LZH, GOB
9	11/10/2019	9:30 a 9:45 hrs	Se mostró la pantalla ya modificada de cuestionario dinámico con los datos de contacto ya visualizándose.	LZH, GOB
10	14/10/2019	9:30 a 9:45 hrs	Durante esta reunión se estableció el diseño completo de la pantalla cuestionario dinámico, incluyendo los nuevos botones y funcionalidades que contendrá el cuestionario.	LZH, GOB
11	15/10/2019	9:30 a 9:45 hrs	Se mostró el diseño completo de la pantalla cuestionario dinámico.	LZH, GOB

12	16/10/2019	9:30 a 9:45 hrs	El scrum master decidió establecer los diseños de las pantallas ya realizadas compatibles desde Android 6 en adelante.	LZH, GOB
13	17/10/2019	9:30 a 9:45 hrs	Durante esta reunión se habló acerca de la falta de dispositivos Android para realizar las pruebas de la aplicación.	LZH, GOB
14	18/10/2019	9:30 a 9:45 hrs	El scrum master hizo entrega de una de las tabletas de la empresa para realizar las pruebas de la aplicación.	LZH, GOB
15	21/10/2019	9:30 a 9:45 hrs	Durante esta reunión se establecieron los tipos de garantías e inmuebles que se manejarían.	LZH, GOB
16	22/10/2019	9:30 a 9:45 hrs	Se establecieron las preguntas de todos los tipos de cuestionarios.	LZH, GOB
17	23/10/2019	9:30 a 9:45 hrs	Se establecieron los tipos de preguntas de los cuestionarios.	LZH, GOB
18	24/10/2019	9:30 a 9:45 hrs	Se mostró la base de datos con los cuestionarios y sus respectivas preguntas y atributos.	LZH, GOB
19	25/10/2019	9:30 a 9:45 hrs	Se mostró la pantalla de cuestionario dinámico con las preguntas del cuestionario garantías inmobiliarias fungibles.	LZH, GOB
20	28/10/2019	9:30 a 9:45 hrs	Se mostró la pantalla cuestionario dinámico con las preguntas y los campos dinámicos.	LZH, GOB
21	29/10/2019	9:30 a 9:45 hrs	El scrum master solicitó realizar pruebas de usuario de la pantalla cuestionario dinámico.	LZH, GOB
22	30/10/2019	9:30 a 9:45 hrs	Se mostró la lista de errores de la pantalla cuestionario dinámico y el scrum master solicitó corregir dichos errores.	LZH, GOB
23	31/10/2019	9:30 a 9:45 hrs	Se mostró la pantalla de cuestionario dinámico con todas las validaciones y el scrum master solicitó nuevamente pruebas de usuario.	LZH, GOB

**Tabla 8.** Description del Daily Sprint meeting.

La tabla 9 muestra a detalle las reuniones diarias que se realizaron durante el mes de noviembre.

NOVIEMBRE				
ID	Fecha	Horario	Description del Daily Sprint meeting	Asistió
1	1/11/2019	9:30 a 9:45 hrs	Se mostraron las pruebas de los usuarios y se determinó que la pantalla cuestionario dinámico estaba en perfecto funcionamiento.	LZH, GOB
2	4/11/2019	9:30 a 9:45 hrs	Se estableció que el botón mostrar respuestas mostraría sólo las preguntas con respuestas contestadas.	LZH, GOB
3	5/11/2019	9:30 a 9:45 hrs	Se mostró el funcionamiento del botón mostrar respuestas.	LZH, GOB
4	6/11/2019	9:30 a 9:45 hrs	El scrum master solicitó las pruebas de consistencia de la función mostrar respuestas.	LZH, GOB
5	7/11/2019	9:30 a 9:45 hrs	Se mostró el resultado de las pruebas de consistencia.	LZH, GOB
6	8/11/2019	9:30 a 9:45 hrs	Se determinó que la función mostrar estaba lista para ser habilitada para el usuario.	LZH, GOB
7	11/11/2019	9:30 a 9:45 hrs	Durante esta reunión se diseñó la pantalla firmar documentos.	LZH, GOB
8	12/11/2019	9:30 a 9:45 hrs	Se mostró el funcionamiento de la pantalla firmar documento.	LZH, GOB
9	13/11/2019	9:30 a 9:45 hrs	Se realizó una retroalimentación de la pantalla firmar documento para modificar la pantalla de la ampliación, solicitando agregar un nuevo botón llamado limpiar en la pantalla firmar documento y también solicitó las pruebas de dicha pantalla.	LZH, GOB
10	14/11/2019	9:30 a 9:45 hrs	Se mostró la pantalla firmar documentos con los respectivos campos ya realizados.	LZH, GOB
11	15/11/2019	9:30 a 9:45 hrs	Se aprobó la pantalla firmar documento para ser habilitada al usuario.	LZH, GOB
12	18/11/2019	9:30 a 9:45 hrs	Se mostró la funcionalidad del botón finalizar reporte.	LZH, GOB
13	19/11/2019	9:30 a 9:45 hrs	Se realizaron pruebas al módulo finalizar reporte por parte del Scrum Master.	LZH, GOB
14	20/11/2019	9:30 a 9:45 hrs	Se aprobó el botón finalizar reporte y se estableció el diseño para la generación del pdf del botón	LZH, GOB



			visualizar reporte dentro de la aplicación.	
15	21/11/2019	9:30 a 9:45 hrs	Se mostró la funcionalidad del botón visualizar reporte de verificación y se solicitó por parte del scrum master pruebas del funcionamiento de dicho botón.	LZH, GOB
16	22/11/2019	9:30 a 9:45 hrs	Se presentaron las pruebas y el scrum master aprobó dicho botón.	LZH, GOB
17	25/11/2019	9:30 a 9:45 hrs	El scrum master solicitó realizar el botón generar reporte de verificación en pdf el cual se podrá visualizar desde una aplicación externa.	LZH, GOB
18	26/11/2019	9:30 a 9:45 hrs	Se mostró el funcionamiento del botón generar reporte, así como también el scrum master solicitó realizar las pruebas al funcionamiento de este botón.	LZH, GOB
19	27/11/2019	9:30 a 9:45 hrs	Se mostraron las pruebas realizadas al módulo visualizar reporte, así como también se recibió retroalimentación de los errores de la generación del pdf.	LZH, GOB
20	28/11/2019	9:30 a 9:45 hrs	Se mostró el funcionamiento del botón generar reporte con las modificaciones ya realizadas.	LZH, GOB
21	29/11/2019	9:30 a 9:45 hrs	Se aprobó el módulo generar reporte, así como el demo de la aplicación VERIFICAPP.	LZH, GOB

**Tabla 9.** Description del Daily Sprint meeting.

### 3.6 Demo y Retrospectiva

En esta etapa se realizaron pruebas de consistencia, fiabilidad y funcionalidad de todos los módulos de la aplicación VERIFICAPP, para comprobar el correcto funcionamiento de la aplicación y así lanzar la aplicación a los usuarios destinados que harán uso de la aplicación para verificar los inmuebles o garantías. A continuación, se presentan los resultados de cada sprint y las propuestas de mejoras que se propusieron durante las pruebas.

Durante las pruebas del sprint 1 se determinó que en la pantalla búsqueda que se muestra en la figura 43 de cliente, se asignen los clientes a cada uno de los verificadores y respecto a los clientes que se le asignaron al verificador deben ser los que les aparezcan en el buscador de clientes. Cabe mencionar que los clientes son registrados en una base de datos externa de un software que actualmente está siendo desarrollado es por ello que como una mejora futura se pretende consultar en esta base de datos externa los clientes ya asignados a los verificadores.

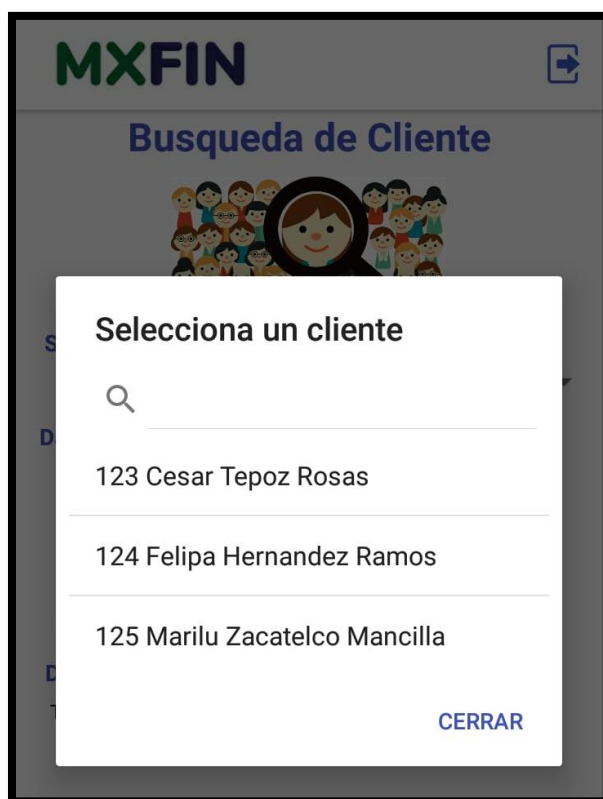




Figura 43. Búsqueda de cliente.

Durante las pruebas del sprint 2 se determinó que en la pantalla búsqueda de cliente como se visualiza en la figura 67 se agreguen nuevos campos de visualización de los datos de la garantía como, por ejemplo: características o nombre del lugar para tener mejor información de la garantía.

**MXFIN** 

## Busqueda de Cliente



**Seleccione un cliente:**

124 Felipa Hernandez Ramos ▼

**Datos generales del cliente:**

Nombre: Felipa Hernandez Ramos

Dirección: Av Rosas #3 San Cristobal Tepontla

Dirección Fiscal: Av Allende #5 San Mateo

Teléfono: 2223232337

**Datos de la garantía:**

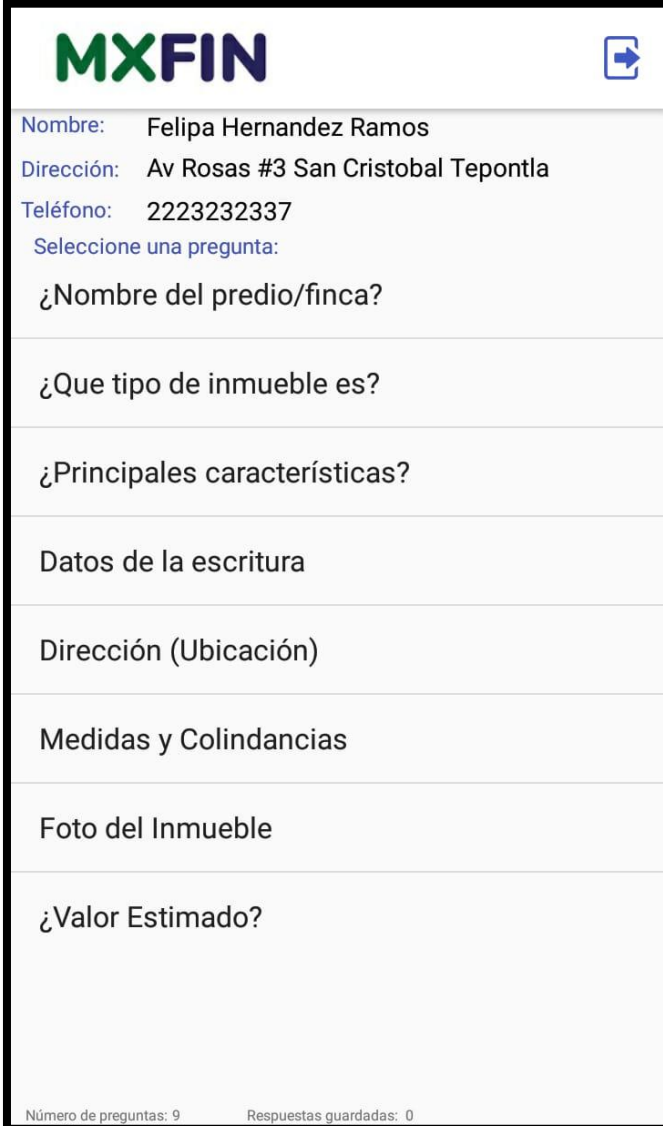
Tipo de garantía: Garantias inmobiliarias Fungibles

Dirección: Av Lazaro Cardenas #4 San Luis

**CONTINUAR**

**Figura 44.** Visualización de los datos búsqueda de cliente.

Durante las pruebas del sprint 3 de la pantalla cuestionario dinámico como se muestra en la figura 45 se propuso como objetivo de mejora, mostrar las imágenes capturadas debajo de la pregunta que haya solicitado las fotografías.



The screenshot shows a mobile application interface for MXFIN. At the top left is the MXFIN logo, and at the top right is a share icon. Below the header, the user's information is displayed: Name: Felipa Hernandez Ramos, Address: Av Rosas #3 San Cristobal Tepontla, and Phone: 2223232337. A prompt 'Seleccione una pregunta:' is followed by a list of questions: '¿Nombre del predio/finca?', '¿Que tipo de inmueble es?', '¿Principales características?', 'Datos de la escritura', 'Dirección (Ubicación)', 'Medidas y Colindancias', 'Foto del Inmueble', and '¿Valor Estimado?'. At the bottom, it shows 'Número de preguntas: 9' and 'Respuestas guardadas: 0'.

Nombre:	Felipa Hernandez Ramos
Dirección:	Av Rosas #3 San Cristobal Tepontla
Teléfono:	2223232337
Seleccione una pregunta:	
¿Nombre del predio/finca?	
¿Que tipo de inmueble es?	
¿Principales características?	
Datos de la escritura	
Dirección (Ubicación)	
Medidas y Colindancias	
Foto del Inmueble	
¿Valor Estimado?	
Número de preguntas:	9
Respuestas guardadas:	0

**Figura 45.** Pantalla Verificación.

Durante las pruebas del sprint 4 en la pantalla cuestionario dinámico la cual se muestra en la figura 46 se propuso que al presionar el botón mostrar respuestas las imágenes también puedan ser visualizadas de bajo de la pregunta que contiene dicha evidencia, así como también se propuso que al presionar la preguntas con su respuesta lance un mensaje que pregunte si desea modificar la respuesta, en caso de ser si mostrar una nueva y pequeña ventana que permita modificar la respuesta.

**MXFIN**

Nombre: Felipa Hernandez Ramos  
Dirección: Av Rosas #3 San Cristobal Tepontla  
Teléfono: 2223232337

Seleccione una pregunta:  
Dirección (Ubicación)  
Av. Miguel Alemán 116, Centro, 72760  
Cholula, Pue., México Cholula Puebla México  
72760

Medidas y Colindancias  
298900 Metros cuadrados, colinda con finca girasol

¿Que tipo de inmueble es?  
Bienes por naturaleza

Dirección (Ubicación)  
Av. Miguel Alemán 102, Centro, 72760  
Cholula, Pue., México Cholula Puebla México  
72760

¿Valor Estimado?  
20000000

Observaciones  
La finca es tal y como se describio en la solicitud

**MOSTRAR RESPUESTAS** **FIRMAR DOCUMENTO**

Número de preguntas: 9 Respuestas guardadas: 12

**Figura 46.** Pantalla cuestionario dinámico.

Durante las pruebas del sprint 5 en el botón generar reporte se propuso que al generar el pdf como se muestra en la figura 47 se agregue la pregunta antes de la imagen insertada en el pdf, para tener la información más precisa y clara.

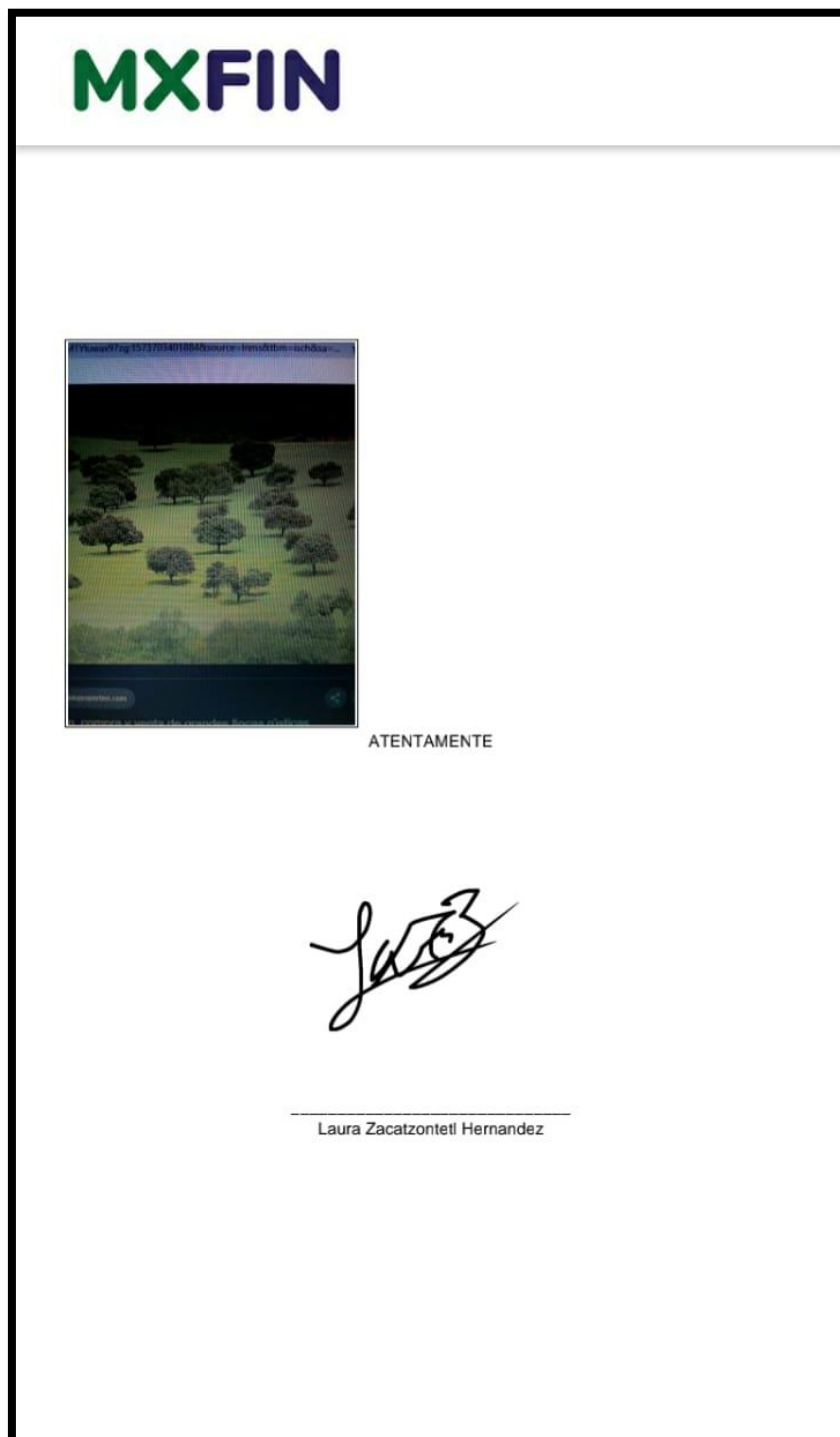


Figura 47. Reporte de verificación.

# 4. Conclusiones y recomendaciones

---

El desarrollar una aplicación en Android ha sido un gran reto personal, pues implica aprender un nuevo entorno de trabajo como lo es Android Studio y Firebase para el desarrollo de la base de datos, así como también requiere un gran análisis de requerimientos y un diseño previo de la aplicación para evitar problemas o dificultades durante la implementación de la aplicación.

La aplicación VERIFICAPP ayuda a el proceso de supervisión y verificación de la aplicación del crédito, sin embargo, es necesario hacer buen uso del software puesto que si los verificadores o supervisores alteran la información no se podrá tener una supervisión eficaz de la aplicación del crédito y la cartera podría verse afectada.

Durante las pruebas de este proyecto se detectaron nuevas funcionalidades que podrían desarrollarse como por ejemplo visualizar las imágenes tomadas debajo de la pregunta que solicitó dicha evidencia o sólo permitirle buscar los clientes que han sido asignadas a tal supervisor, cabe mencionar que durante el desarrollo de una aplicación siempre surgirán mejoras y procesos que automatizar.

Al desarrollar este proyecto he adquirido nuevos conocimientos, puesto que yo no tenía conocimientos en desarrollar aplicaciones y tuve que aprender por mi parte a desarrollar aplicaciones en el entorno de programación en Android, así como también tuve que tomar varios cursos en línea para aprender a desarrollar base de datos en Firebase y así implementar las consultas y funciones para el funcionamiento de la aplicación.

En este proyecto desarrollé mi primera aplicación en Android por lo que he adquirido más experiencia en el análisis diseño e implementación de software, así como también conocí un nuevo entorno laboral de personas que se desempeñan en el desarrollo de software y esto a su vez me motivo a querer desarrollarme profesionalmente como programadora de software.

# 5. Anexos

---

## Anexo A

En este anexo se muestra el código en lenguaje java de las consultas en Firebase que realizan el funcionamiento de la aplicación, específicamente las consultas y código que permite el funcionamiento del cuestionario dinámico.

```
databaseReference.child("cuestionariosP").child(seccion).orderByChild("ordende
lapregunta").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        for (DataSnapshot objSnapshot:dataSnapshot.getChildren())
        {
            if(pregunta.equals((String)
objSnapshot.child("pregunta").getValue().toString())){
                tipoo = (String)
objSnapshot.child("tipo").getValue().toString();

                break;
            }
        }
    }
}

if (tipoo.equals("numero")){
    final AlertDialog.Builder builderN = new
AlertDialog.Builder(verificacionFinal.this);
    final int indice=i;
    builderN.setTitle(pregunta);
    builderN.setMessage(" ");
    inputNumeros = new EditText(verificacionFinal.this);
    inputNumeros.setInputType(InputType.TYPE_CLASS_NUMBER);
    builderN.setView(inputNumeros);
    builderN.setPositiveButton("OK", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            if(inputNumeros.getText().toString().length()>0) {
                String cadena = pregunta + "\n" +
inputNumeros.getText().toString();
                listAsk.set(indice, cadena);
                arrayAdapterAsk = new
ArrayAdapter<String>(verificacionFinal.this,
android.R.layout.simple_list_item_1, listAsk);
                listView_preguntas.setAdapter(arrayAdapterAsk);
                String id = UUID.randomUUID().toString();
                databaseReference.child("respuestas/" +seccion+ "/"
+fechaActual+ "/" +nombre+ "/" +seccion+ "/" +id).child("id").setValue(id);
                databaseReference.child("respuestas/" +seccion+ "/"
+fechaActual+ "/" +nombre+ "/" +seccion+
"/" +id).child("nombrePersona").setValue(nombre.toString());
                databaseReference.child("respuestas/" +seccion+ "/"
+fechaActual+ "/" +nombre+ "/" +seccion+
"/" +id).child("pregunta").setValue(pregunta.toString());
                databaseReference.child("respuestas/" +seccion+ "/"
+fechaActual+ "/" +nombre+ "/" +seccion+
"/" +id).child("respuesta").setValue(inputNumeros.getText().toString());
                databaseReference.child("respuestas/" +seccion+ "/"
```



```

+fechaActual+ "/" + nombre + "/" + seccion +
"/" + id).child("tipoDeCuestionario").setValue("numero");
        databaseReference.child("respuestas/" + seccion + "/"
+fechaActual+ "/" + nombre + "/" + seccion +
"/" + id).child("status").setValue("inicializado")    } } });
        builderN.setNegativeButton("Cancelar", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) { } });
        AlertDialog alertN = builderN.create();
        alertN.show();
        if (tipoo.equals("decimal")) {
            final AlertDialog.Builder builderN = new
AlertDialog.Builder(verificacionFinal.this);
            final int indice=i;
            builderN.setTitle(pregunta);
            builderN.setMessage(" ");
            inputNumeros = new EditText(verificacionFinal.this);
            inputNumeros.setInputType(InputType.TYPE_CLASS_NUMBER |
InputType.TYPE_NUMBER_FLAG_DECIMAL);
            builderN.setView(inputNumeros);
            builderN.setPositiveButton("OK", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {
                    if(inputNumeros.getText().toString().length()>0) {
                        String cadena = pregunta + "\n" +
inputNumeros.getText().toString();
                        listAsk.set(indice, cadena);
                        arrayAdapterAsk = new
ArrayAdapter<String>(verificacionFinal.this,
android.R.layout.simple_list_item_1, listAsk);
                        listView_preguntas.setAdapter(arrayAdapterAsk);
                        String id = UUID.randomUUID().toString();
                        databaseReference.child("respuestas/" + seccion + "/"
+fechaActual+ "/" + nombre + "/" + seccion + "/" + id).child("id").setValue(id);
                        databaseReference.child("respuestas/" + seccion + "/"
+fechaActual+ "/" + nombre + "/" + seccion +
"/" + id).child("nombrePersona").setValue(nombre.toString());
                        databaseReference.child("respuestas/" + seccion + "/"
+fechaActual+ "/" + nombre + "/" + seccion +
"/" + id).child("pregunta").setValue(pregunta.toString());
                        databaseReference.child("respuestas/" + seccion + "/"
+fechaActual+ "/" + nombre + "/" + seccion +
"/" + id).child("respuesta").setValue(inputNumeros.getText().toString());
                        databaseReference.child("respuestas/" + seccion + "/"
+fechaActual+ "/" + nombre + "/" + seccion +
"/" + id).child("tipoDeCuestionario").setValue("decimal");
                        databaseReference.child("respuestas/" + seccion + "/"
+fechaActual+ "/" + nombre + "/" + seccion +
"/" + id).child("status").setValue("inicializado");
                    } } });
            builderN.setNegativeButton("Cancelar", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int i) {} });
            AlertDialog alertN = builderN.create();
            alertN.show();
        }
}

```

```

if (tipoo.equals("texto abierto")){
    final AlertDialog.Builder builder2 = new
AlertDialog.Builder(verificacionFinal.this);
    final int indice=i;
    builder2.setTitle(pregunta);
    builder2.setMessage(" ");
    input = new EditText(verificacionFinal.this);
    input.setInputType(InputType.TYPE_CLASS_TEXT);
    builder2.setView(input);
    builder2.setPositiveButton("OK", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            String id = UUID.randomUUID().toString();
            if(input.getText().toString().length()>0) {
                String cadena = pregunta + "\n" + input.getText().toString();
                listAsk.set(indice, cadena);
                arrayAdapterAsk = new
ArrayAdapter<String>(verificacionFinal.this,
android.R.layout.simple_list_item_1, listAsk);
                listView_preguntas.setAdapter(arrayAdapterAsk);
                databaseReference.child("respuestas/" +seccion+ "/"
+fechaActual+ "/" +nombre+ "/" +seccion+ "/" +id).child("id").setValue(id);
                databaseReference.child("respuestas/" +seccion+ "/"
+fechaActual+ "/" +nombre+ "/" +seccion+
"/" +id).child("nombrePersona").setValue(nombre.toString());
                databaseReference.child("respuestas/" +seccion+ "/"
+fechaActual+ "/" +nombre+ "/" +seccion+
"/" +id).child("pregunta").setValue(pregunta.toString());
                databaseReference.child("respuestas/" +seccion+ "/"
+fechaActual+ "/" +nombre+ "/" +seccion+
"/" +id).child("respuesta").setValue(input.getText().toString());
                databaseReference.child("respuestas/" +seccion+ "/"
+fechaActual+ "/" +nombre+ "/" +seccion+
"/" +id).child("tipoDeCuestionario").setValue("texto abierto");
                databaseReference.child("respuestas/" +seccion+ "/"
+fechaActual+ "/" +nombre+ "/" +seccion+
"/" +id).child("status").setValue("inicializado");
                input=null;
            }
        }
    });
    builder2.setNegativeButton("Cancelar", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
        }
    });
    AlertDialog alert2 = builder2.create();
    alert2.show();
}
if (tipoo.equals("foto")){
    llamarIntent();
}
if (tipoo.equals("georeferencia")){
    LayoutInflater li2 = LayoutInflater.from(context);
    View promptsView = li2.inflate(R.layout.dialog_georeferencia, null);
    AlertDialog.Builder builderGeo = new
AlertDialog.Builder(verificacionFinal.this);

```

```

        builderGeo.setView(promptststView);
        final EditText longt = (EditText)
promptststView.findViewById(R.id.editLong);
        final TextView direccionC = (TextView)
promptststView.findViewById(R.id.direccionC);
        final EditText comentario = (EditText)
promptststView.findViewById(R.id.editComen);
        longt.setText(TextL + "," + TextLa);
        direccionC.setText(TextDireccion);
        longt.setEnabled(false);
        builderGeo.setNegativeButton("SALIR", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) { } });
        builderGeo.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                Date ahora = new Date();
                SimpleDateFormat formateador = new SimpleDateFormat("hh:mm");
                SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd",
Locale.getDefault());
                Date date = new Date();
                firebaseDatabase = FirebaseDatabase.getInstance();
                databaseReference = firebaseDatabase.getReference();
                String seccion = mSpinnerseccion.getSelectedItem().toString();
                String id = UUID.randomUUID().toString();
                databaseReference.child("respuestas/" +seccion+ "/" +fechaActual+
"/"+nombre+ "/" +seccion+ "/" +id).child("id").setValue(id);
                databaseReference.child("respuestas/" +seccion+ "/" +fechaActual+
"/"+nombre+ "/" +seccion+
"/"+id).child("nombrePersona").setValue(nombre.toString());
                databaseReference.child("respuestas/" +seccion+ "/" +fechaActual+
"/"+nombre+ "/" +seccion+
"/"+id).child("pregunta").setValue(pregunta.toString());
                databaseReference.child("respuestas/" +seccion+ "/" +fechaActual+
"/"+nombre+ "/" +seccion+
"/"+id).child("respuestaDireccion").setValue(longt.getText().toString());
                databaseReference.child("respuestas/" +seccion+ "/" +fechaActual+
"/"+nombre+ "/" +seccion+
"/"+id).child("respuestaComentario").setValue(comentario.getText().toString());
                ;
                databaseReference.child("respuestas/" +seccion+ "/" +fechaActual+
"/"+nombre+ "/" +seccion+
"/"+id).child("respuesta").setValue(direccionC.getText().toString());
                databaseReference.child("respuestas/" +seccion+ "/" +fechaActual+
"/"+nombre+ "/" +seccion+
"/"+id).child("tipoDeCuestionario").setValue("georeferencia");
                databaseReference.child("respuestas/" +seccion+ "/" +fechaActual+
"/"+nombre+ "/" +seccion+ "/" +id).child("status").setValue("inicializado");
                Toast.makeText(verificacionFinal.this, "Respuesta guardada",
Toast.LENGTH_LONG).show(); } });
        builderGeo.setNegativeButton("Cancelar", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) { }
        });
        AlertDialog alertGeo = builderGeo.create();
        alertGeo.show();
    }
    if(tipoo.equals("respuesta seleccionable")){

```

```

databaseReference.child("respuestasOpcionales").child(seccion).child(pregunta)
.orderByChild("respuesta0").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        listUnica.clear();
        for (DataSnapshot objSnapshot : dataSnapshot.getChildren()) {
            unica = (String)
objSnapshot.child("respuesta0").getValue(String.class);
            listUnica.add(unica);
        }
        if(listUnica.size() != 0){
            final boolean[] array1;
            array1 = new boolean[listUnica.size()];
            Arrays.fill(array1, Boolean.FALSE);

            final CharSequence charSequence2 [] = new
String[listUnica.size()];
            for (int k=0; k<listUnica.size(); k++){
                charSequence2[k] = String.valueOf(listUnica.get(k));
            }
        }
        final CharSequence charSequence2[] = new String[listUnica.size()];
        for (int i=0; i<listUnica.size(); i++) {
            charSequence2[i] = String.valueOf(listUnica.get(i));
        }
        resUnica = list2.toArray(new CharSequence[list.size()]);
        final int indice=i;
        final AlertDialog.Builder builderRU = new
AlertDialog.Builder(verificacionFinal.this);
        builderRU.setTitle(pregunta);
        builderRU.setSingleChoiceItems(charSequence2, -1, new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                respuestaUnica = listUnica.get(i);
            }
        });
        builderRU.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                if(respuestaUnica!=null) {
                    String id = UUID.randomUUID().toString();
                    databaseReference.child("respuestas/" + seccion + "/"
+ fechaActual + "/" + nombre + "/" + seccion + "/" +
id).child("id").setValue(id);
                    databaseReference.child("respuestas/" + seccion + "/"
+ fechaActual + "/" + nombre + "/" + seccion + "/" +
id).child("nombrePersona").setValue(nombre.toString());
                    databaseReference.child("respuestas/" + seccion + "/"
+ fechaActual + "/" + nombre + "/" + seccion + "/" +
id).child("pregunta").setValue(pregunta.toString());
                    databaseReference.child("respuestas/" + seccion + "/"
+ fechaActual + "/" + nombre + "/" + seccion + "/" +
id).child("respuesta").setValue(respuestaUnica.toString().trim());
                    databaseReference.child("respuestas/" + seccion + "/"
+ fechaActual + "/" + nombre + "/" + seccion + "/" +

```

```

id).child("tipoDeCuestionario").setValue("respuesta seleccionable");
        databaseReference.child("respuestas/" + seccion + "/"
+ fechaActual + "/" + nombre + "/" + seccion + "/" +
id).child("status").setValue("inicializado");
    }
    else {
        LayoutInflater imagenadvertencia_alert
=LayoutInflater.from(verificacionFinal.this);final View vista =
imagenadvertencia_alert.inflate(R.layout.imagenadvertencia, null);
        AlertDialog.Builder alerta = new
AlertDialog.Builder(verificacionFinal.this);
        alerta.setMessage("No seleccionó alguna
opción.").setCancelable(true).setCustomTitle(vista).setPositiveButton("Aceptar
",null) ;alerta.show();
    }
    if(respuestaUnica!=null) {
        String cadena = pregunta + "\n" + respuestaUnica;
        listAsk.set(indice, cadena);
        arrayAdapterAsk = new
ArrayAdapter<String>(verificacionFinal.this,
android.R.layout.simple_list_item_1, listAsk);
        listView_preguntas.setAdapter(arrayAdapterAsk);
        respuestaUnica=null;
    } } };
    builderRU.setNegativeButton("Cancelar", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
}
        });
    AlertDialog alertRU = builderRU.create();
    alertRU.show(); }
    @Override public void onCancelled(DatabaseError databaseError) { }
} });
}

```

## 4. Referencias bibliográficas

---

[1] URL: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html> Página oficial de Softeng la cual es una consultoría e ingeniería de software, cuya misión es ayudar a las empresas a incrementar su ventaja competitiva optimizando sus sistemas. Fecha de consulta: 18/Octubre/2019

[2] Eugenia Bahit. “Scrum & eXtreme programming para programadores” Editorial safeCreative, Argentina, 2012.

[3] Adell, Leo. “Benefits & Pitfalls of using Scrum software development methodology” Editorial Belatrix Software Development Blog, Belatrix, Buenos Aires. 2013, Pag 1-2.

[4] URL: <https://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/> Página oficial de academia de Android en ella se puede consultar información relacionada con Android. Fecha de consulta: 19/Octubre/2019

[5] Neil Smyth, “Firebase Essentials” Editorial eBookFrenzy, USA. 2017

[6] URL: <https://obux.wordpress.com/2017/03/09/5-ventajas-y-desventajas-de-firebase-en-2017/> Página oficial de Obux en ella se puede consultar información relacionada a Ubuntu entre otros artículos referentes a la tecnología. Fecha de consulta: 22/Octubre/2019



Universidad Politécnica de Puebla  
Ingeniería en Informática

*Laura Zacatzontetl Hernandez  
Germán Ortega Benítez.  
Rebeca Rodríguez Huesca.*

Este documento se distribuye para los términos de la  
Licencia 2.5 Creative Commons (CC-BC-NC-ND 2.5 MX)