



UNIVERSIDAD SURCOLOMBIANA
GESTIÓN DE BIBLIOTECAS



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

1 de 2

Neiva, 16 de diciembre _____

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

Jhon Edinson Tafur Mosquera,

con C.C. No. 1.075.312.432,

_____, con C.C. No. _____,

_____, con C.C. No. _____,

_____, con C.C. No. _____,

Autor(es) de la tesis y/o trabajo de grado o _Proyecto de Pasantías_ titulado _Desarrollo de aplicación móvil dirigida al proyecto Claro pruebas y cruzadas en sitios conectantes -IE MINTIC- 7K_ presentado y aprobado en el año _2022_ como requisito para optar al título de _Ingeniero electrónico_;

Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales "open access" y en las redes de información con las cuales tenga convenio la Institución.
- Permita la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.
- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.



UNIVERSIDAD SURCOLOMBIANA GESTIÓN DE BIBLIOTECAS



CARTA DE AUTORIZACIÓN

CÓDIGO**AP-BIB-FO-06****VERSIÓN****1****VIGENCIA****2014****PÁGINA****2 de 2**

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “Los derechos morales sobre el trabajo son propiedad de los autores”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

EL AUTOR/ESTUDIANTE: Jhon Edinson Tafur Mosquera

Firma:

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.

	UNIVERSIDAD SURCOLOMBIANA GESTIÓN DE BIBLIOTECAS				   	
	DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO					
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA 1 de 3

TÍTULO COMPLETO DEL TRABAJO:

Desarrollo de aplicación móvil dirigida al proyecto Claro pruebas y cruzadas en sitios conectantes -IE MINTIC- 7k

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Tafur Mosquera	Jhon Edinson

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
Molina Mosquera	Johan Julián

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre

PARA OPTAR AL TÍTULO DE: Ingeniero Electrónico

FACULTAD: Ingeniería

PROGRAMA O POSGRADO: Ingeniería Electrónica

CIUDAD: Neiva

AÑO DE PRESENTACIÓN: 2022

NÚMERO DE PÁGINAS:40

TIPO DE ILUSTRACIONES (Marcar con una X):

Diagramas_X_ Fotografías_X_ Grabaciones en discos___ Ilustraciones en general___ Grabados___ Láminas___ Litografías___ Mapas___ Música impresa___ Planos___ Retratos___ Sin ilustraciones___ Tablas o Cuadros_X_

Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.

	UNIVERSIDAD SURCOLOMBIANA GESTIÓN DE BIBLIOTECAS					   	
	DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO						
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 3

SOFTWARE requerido y/o especializado para la lectura del documento: Android Studio, Firebase

MATERIAL ANEXO:

PREMIO O DISTINCIÓN (*En caso de ser LAUREADAS o Meritoria*):

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

<u>Español</u>	<u>Inglés</u>	<u>Español</u>	<u>Inglés</u>
1. _Android Studio_	Android Studio	6. _____	_____
2. _Base de datos_	Data base	7. _____	_____
3. __Cruzada_	Crusade	8. _____	_____
4. __Firebase_	Firebase	9. _____	_____
5. _____	_____	10. _____	_____

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

Ingelec S.A.S actualmente trabaja de la mano con Comcel S.A (Claro) en el Proyecto Claro pruebas y cruzadas en sitios conectantes -IE- Mintic 7K el cual tiene un tiempo de desarrollo de 10 años. Ingelec ha venido desarrollando una especie de formato en donde se registra toda la información de cada actividad realizada con el fin de tener evidencias para posteriormente evitar inconvenientes con la facturación mensual de dicho proyecto. Sin embargo, el diligenciamiento de este formato no se realiza de manera correcta (o ni siquiera hace) ya que es algo tedioso y demorado para los técnicos en zona quienes son los encargados de realizarlo. Puesto que, generalmente los recorridos que deben hacer hasta las BTS (Estación base-transmisión) son largos y peligrosos. Por lo tanto, para facilitar, agilizar y asegurar el registro actividades con evidencias fotográficas por parte de los técnicos se propone implementar una aplicación móvil con modelo cliente-servidor (para el ingreso y retención de datos) a través de Android Studio mediante el lenguaje Java, enlazada a Firebase como base de datos cuya interfaz sea fácil de entender por el usuario y le permita registrar los datos de la actividad en el menor tiempo posible.

	UNIVERSIDAD SURCOLOMBIANA GESTIÓN DE BIBLIOTECAS						
	DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO						
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	3 de 3

ABSTRACT: (Máximo 250 palabras)

Ingelec S.A.S is currently working hand in hand with Comcel S.A (Claro) in Claro pruebas y cruzadas en sitios conectantes -IE- Mintic 7K Project wich has a development time of ten years. Ingelec has been developing a kind of format where all the information of each activity performed is recorded in order to have evidence to later avoid inconveniences with the monthly billing of the project. However, the filling out of this form is not being done correctly (or not even done) because it is tedious and time-consuming for the technicians in zone who are in charge of fill it, since, trips they must take to the BTS (Base-Transmission Station) use to be long and dangerous. Therefore, to facilitate, streamline and ensure the recording of activities including photographic evidence by the technicians, it is proposed to implement a mobile application with client-server model (for data entry and retention) through Android Studio using Java language, linked to Firebase as a database whose interface is easy to understand by the user and allows him to record the activity data in the shortest possible time.

APROBACION DE LA TESIS

Nombre Presidente Jurado:

Firma:

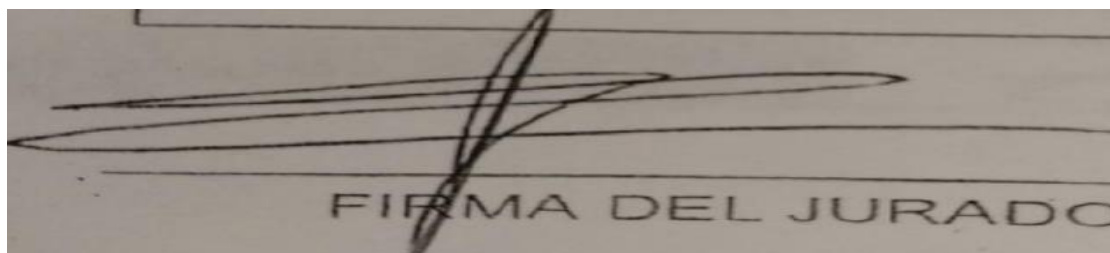
Nombre Jurado: Martín Diomedes Bravo Obando

Firma:



Nombre Jurado: Jesús David Quintero Polanco

Firma:



Vigilada Mineducación

La versión vigente y controlada de este documento, solo podrá ser consultada a través del sitio web Institucional www.usco.edu.co, link Sistema Gestión de Calidad. La copia o impresión diferente a la publicada, será considerada como documento no controlado y su uso indebido no es de responsabilidad de la Universidad Surcolombiana.

**DESARROLLO DE APLICACIÓN MÓVIL DIRIGIDA AL PROYECTO CLARO
PRUEBAS Y CRUZADAS EN SITIOS CONECTANTES -IE- MINTIC 7K.**

JHON EDINSON TAFUR MOSQUERA

COD: 20162152620

UNIVERSIDAD SURCOLOMBIANA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA ELECTRÓNICA

NEIVA – HUILA

2022

**DESARROLLO DE APLICACIÓN MÓVIL DIRIGIDA AL PROYECTO CLARO
PRUEBAS Y CRUZADAS EN SITIOS CONECTANTES -IE- MINTIC 7K.**

JHON EDINSON TAFUR MOSQUERA

COD: 20162152620

**TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO
ELECTRÓNICO**

Director

ING. JOHAN JULIAN MOLINA MOSQUERA

Docente Programa Ingeniería Electrónica

UNIVERSIDAD SURCOLOMBIANA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA ELECTRÓNICA

NEIVA – HUILA

2022

DEDICATORIA

A mis padres, María del Pilar Mosquera y Royber René Tafur por su constante e incondicional apoyo y motivación que sin importar cuantas veces erre, nunca dejan de creer en mí.

A mis hermanos. Pues son ellos el motor que me impulsa a seguir creciendo cada día más para alcanzar mis metas.

A mis amigos y demás personas que de una u otra manera fueron de gran ayuda en el desarrollo de este proyecto.

AGRADECIMIENTOS

A mis padres por brindarme su amor y ser un apoyo incondicional en cada momento de mi vida aun cuando he caído en completa frustración; y a mis hermanos que de una u otra manera han contribuido a mi formación.

A cada uno de los profesores e ingenieros de la Universidad Surcolombiana, que fueron guías y apoyo en mi vida como universitario, en especial al ingeniero Julián Molina, quien fue el encargado de dirigir este proyecto de grado.

A los ingenieros Martín Diomedes Bravo y Jesús David Quintero por haber aportado su juicio para el mejoramiento de este trabajo de grado.

A mis compañeros de universidad y amigos externos, que de alguna manera contribuyeron con sus consejos y experiencias para que pudiera llegar hasta este punto.

Por último, pero no menos importante, agradecerme a mí mismo por no desertar y continuar avanzando y mejorando tanto personal como profesionalmente a pesar de todos los problemas que se presentaron en el camino.

CONTENIDO

	Pág.
INTRODUCCIÓN	
1. OBJETIVOS	11
1.1 OBJETIVO GENERAL	11
1.2 OBJETIVOS ESPECÍFICOS	11
2. MARCO TEÓRICO	12
2.1 ANDROID STUDIO	12
2.1.1 ¿QUÉ ES ANDROID STUDIO?	12
2.1.2 LENGUAJES UTILIZADOS POR ANDROID STUDIO	12
2.1.3 CARACTERÍSTICAS DE ANDROID STUDIO	13
2.2 FIREBASE	13
2.2.1 ¿QUÉ ES FIREBASE?	13
2.2.2 CARACTERÍSTICAS DE FIREBASE	14
2.3 FIREBASE AUTHENTICATION	14
2.3.1 VENTAJAS DE FIREBASE AUTHENTICATION	15
2.3.2 CARACTERÍSTICAS DE FIREBASES AUTHENTICATION	15
2.4 FIREBASE CLOUD FIRESTORE	16
2.4.1 CARACTERÍSTICAS DE CLOUD FIRESTORE	16
2.5 FIREBASE CLOUD STORAGE	17
2.5.1 VENTAJAS DE FIREBASE CLOUD STORAGE	17
3. ENLACE ANDROID STUDIO – FIREBASE	18
4. PANTALLA DE REGISTRO DE USUARIO	21
4.1 DISEÑO DE PANTALLA DE REGISTRO	21
4.2 PROGRAMACIÓN DE PANTALLA DE REGISTRO	24
5. PANTALLA DE LOGUEO	27
5.1 DISEÑO DE PANTALLA DE LOGUEO	27
5.2 PROGRAMACIÓN DE PANTALLA DE LOGUEO	29
6. PANTALLA DE REGISTRO DE DATOS	31
6.1 DISEÑO DE PANTALLA DE REGISTRO DE DATOS	31
6.2 PROGRAMACIÓN DE PANTALLA DE REGISTRO DE DATOS	34
7. CONCLUSIONES	39
8. BIBLIOGRAFÍA	40

LISTA DE FIGURAS

	Pag.
Figura 1. Paso a paso enlace Firebase Authentication – Android Studio	18
Figura 2. Paso a paso enlace Cloud Firestore – Android Studio	19
Figura 3. Paso a paso enlace Cloud Storage – Android Studio	19
Figura 4. Indicativo de enlace exitoso con Firebase Authentication	20
Figura 5. Línea de código modificada en archivo manifest	22
Figura 6. Diseño de pantalla de registro	23
Figura 7. Condición de ingreso y seguridad de contraseña	24
Figura 8. Datos de usuarios registrados en Cloud Firestore	26
Figura 9. Usuarios registrados en Firebase Authentication	26
Figura 10. Diseño pantalla de logueo	28
Figura 11. Servicio de restablecimiento de contraseña	30
Figura 12. Diseño de sección de registro de datos	33
Figura 13. Diseño de sección de registro fotográfico	33
Figura 14. Ventana de selección de fecha	35
Figura 15. Datos escritos almacenados Firebase	37
Figura 16. Datos fotográficos almacenados en Firebase	37

ACRÓNIMOS

MINTIC: Ministerio de las Tecnologías de la Información y las Comunicaciones.

BTS: (Base Transceiver Station). Estación Base transmisora.

NDK: (Native Development Kit). Conjunto de herramientas que implementan fragmentos de una app en código nativo.

SDK: (Software Development Kit). Kit de desarrollo de software; herramientas o programas para crear aplicaciones para plataformas explícitas.

PHP: (Pre Hypertext -processor). Lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

NoSQL: Base de datos no relacionales

ISO: (International Organization for Standardization). Organización internacional de Normalización. Organización mundial cuya función es la elaboración de normas técnicas internacionales.

UID: (Unique ID). Código identificador estándar empleado en la construcción de software.

AdMob: Plataforma de Google que permite recaudar dinero con aplicaciones móviles a través de publicidad integrada.

IDE: Entorno de desarrollo integrado. Software que ayuda a los programadores a código de software de manera eficiente.

RESUMEN

La empresa Ingeniería electrónica del Huila (Ingelec S.A.S) actualmente trabaja de la mano con Comcel S.A (Claro) en el Proyecto Claro pruebas y cruzadas en sitios conectantes -IE- Mintic 7K el cual tiene un tiempo de desarrollo de 10 años. Ingelec ha venido desarrollando una especie de formato en donde se registra toda la información de cada actividad realizada con el fin de tener evidencias para posteriormente evitar inconvenientes con la facturación mensual de dicho proyecto. Sin embargo, el diligenciamiento de este formato no se realiza de manera correcta (o ni siquiera hace) ya que es algo tedioso y demorado para los técnicos en zona quienes son los encargados de realizarlo. Puesto que, generalmente los recorridos que deben hacer hasta las BTS (Estación base-transmisión) son largos y peligrosos. Por lo tanto, para facilitar, agilizar y asegurar el registro actividades con evidencias fotográficas por parte de los técnicos se propone implementar una aplicación móvil con modelo cliente-servidor (para el ingreso y retención de datos) a través de Android Studio mediante el lenguaje Java, enlazada a Firebase como base de datos cuya interfaz sea fácil de entender por el usuario y le permita registrar los datos de la actividad en el menor tiempo posible.

PALABRAS CLAVE:

Android Studio, Firebase, Java, BTS, Cruzada, Modelo Cliente-Servidor, Base de datos.

ABSTRACT

The company Ingeniería Electrónica del Huila (Ingelec S.A.S) is currently working hand in hand with Comcel S.A (Claro) in Claro pruebas y cruzadas en sitios conectantes - IE- Mintic 7K Project wich has a development time of ten years. Ingelec has been developing a kind of format where all the information of each activity performed is recorded in order to have evidence to later avoid inconveniences with the monthly billing of the project. However, the filling out of this form is not being done correctly (or not even done) because it is tedious and time-consuming for the technicians in zone who are in charge of fill it, since, trips they must take to the BTS (Base-Transmission Station) use to be long and dangerous. Therefore, to facilitate, streamline and ensure the recording of activities including photographic evidence by the technicians, it is proposed to implement a mobile application with client-server model (for data entry and retention) through Android Studio using Java language, linked to Firebase as a database whose interface is easy to understand by the user and allows him to record the activity data in the shortest possible time.

KEYWORDS:

Android Studio, Firebase, Java, BTS, Cruzada, Client-Server Model, Data base.

INTRODUCCIÓN

Android Studio es el IDE oficial para el desarrollo de aplicaciones móviles de Android y se basa en IntelliJ IDEA. Además del editor de código eficaz y las herramientas para programadores que ofrece IntelliJ, Android Studio brinda más funciones para aumentar la productividad al desarrollar aplicaciones de Android como:

- Un sistema de compilación que se basa en Gradle
- Un emulador veloz y eficiente que cuenta con múltiples funciones
- Un entorno unificado en el que se puede desarrollar para cualquier dispositivo android
- Compatibilidad con C++ y NDK ¹

En esta ocasión, el desarrollo de este proyecto se hace a través del lenguaje Java ya que se encontró que es más cómodo para el desarrollo de apps.

Por otro lado, Firebase es una plataforma móvil lanzada en el 2011 y adquirida por Google en 2014, cuya función principal es desarrollar y facilitar la creación de aplicaciones de alta calidad rápidamente, con el fin de aumentar la base de usuarios y hacerla más rentable. La plataforma está enlazada a la nube y se encuentra disponible para diferentes plataformas, ya sea iOS, Android y la web. Posee múltiples funciones para que el desarrollador pueda mezclar y ajustar la plataforma según sus necesidades. ²

En este proyecto, se realiza un enlace directo entre el IDE Android Studio, y Firebase para retención y clasificación de evidencia tanto escrita como fotográfica del proyecto Claro pruebas y cruzadas en sitios conectantes -ie- mintic 7k haciendo uso del Authentication, Storage y el Firestore Database de Firebase en donde se archivan los datos ingresados.

¹ “Introducción a Android Studio”, [En línea] [Citado el 16 de agosto de 2022.] Disponible en la web: <https://developer.android.com/studio/intro?hl=es-419>

² “Un poco de historia de Firebase”, [En línea] [Citado el 16 de agosto de 2022] Disponible en la web: <https://openwebinars.net/blog/que-es-firebase-de-google/>

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Desarrollo de una aplicación móvil a través de un software libre dirigida al proyecto Claro pruebas y cruzadas en sitios conectantes -IE- Mintic 7K que permita el ingreso de información (datos e imágenes) y almacenamiento de esta en una base de datos de Ingelec.

1.2 OBJETIVOS ESPECIFICOS

- Realizar una aplicación confiable y de interfaz cómoda para el usuario a través de un software libre.
- Hacer uso de la plataforma Firebase para salvaguardar la información ingresada en la aplicación.
- Implementar en la aplicación un sistema de registro y logueo de usuarios.
- Añadir el servicio de recuperación de contraseña en el cual se envíe un correo electrónico con un link de restauración de clave al email previamente registrado por el usuario.

2. MARCO TEÓRICO

2.1 ANDROID STUDIO

2.1.1 ¿QUÉ ES ANDROID STUDIO?

Android Studio es un IDE, un software, que posee una gran variedad de herramientas que permiten a los desarrolladores crear nuevas aplicaciones para dispositivos Android y/o iOS. Nos ofrece las herramientas necesarias para la generación del código, que se denomina como lógica de la aplicación. Pero también los mecanismos para diseñar la interfaz de usuario que mostrará el desarrollo final.

2.1.2 ¿QUÉ LENGUAJE SE UTILIZA PARA PROGRAMAR EN ANDROID STUDIO?

Android Studio fue diseñado con Java, por lo que desde hace mucho tiempo ha sido el lenguaje de programación por defecto para el desarrollo de apps en este IDE. Sin embargo, Kotlin es el lenguaje que utilizan la mayoría de los desarrolladores para la programación en android debido a los beneficios que ofrece, como; un código sencillo e intuitivo sin renunciar a la eficiencia.

Además de los dos lenguajes previamente mencionados, Android Studio también es compatible con otros lenguajes de programación como Javascript, C#, y PHP.³

³ “Los 5 lenguajes más usados para la programación en Android” [En línea] [Citado el 16 de agosto de 2022.] Disponible en la web: <https://keepcoding.io/blog/programacion-en-android/>

2.1.3 CARACTERÍSTICAS DE ANDROID STUDIO

- El sistema de compilación es permisivo, encima es compatible con Gradle, lo que permite la automatización de compilaciones de forma flexible y con gran rendimiento.
- El entorno permite al usuario trabajar de forma fluida y con gran variedad de funciones prácticas y útiles.
- Posee compatibilidad con servicios en la nube como Google Cloud Plataform
- Permite Modificar fragmentos de código y recursos de una app, sin necesidad de reiniciarla.
- Tiene la función de integración con GitHub y plantillas de código que ayudan a compilar funciones de apps comunes y también importar códigos de muestra.⁴

2.2 FIREBASE

2.2.1 ¿QUÉ ES FIREBASE?

Firebase es una plataforma móvil que tiene como función principal desarrollar y facilitar la creación de aplicaciones móviles de alta calidad. Lo anterior con el fin de incrementar la base de datos y de esta manera incrementar la monetización de dicha app.

Esta plataforma fue fundada en 2011 por Firebase, como SDK de chat para móviles que tenía por nombre “Envolv”, lo cual no tuvo el éxito esperado. Por lo tanto, más adelante se transformaría en otro producto llamado “Firebase Realtime Database” el cual recaudo una gran cantidad de dólares durante 2012-2013.

Debido al éxito que tuvo, en el año 2014 salió al mercado “Firebase Hosting” y “Firebase Authentication”, y en este mismo año la compañía fue comprada

⁴ “¿Qué características tiene Android Studio?” [En línea] [Citado el 16 de agosto de 2022.] Disponible en la web: <https://talently.tech/blog/que-es-android-studio/>

por Google. En el 2016 logran convertir Firebase en una plataforma unificada con una variedad de productos que buscan cubrir el objetivo principal de apoyar y facilitar sus labores a los desarrolladores.

2.2.2 CARACTERÍSTICAS DE FIREBASE

- Es una multiplataforma soportada por Android, iOS, y Web, que permite la monetización a través de AdMob con anuncios y publicidad.
- Brinda el desarrollo y gestión de apps multiplataforma gracias a sus APIs integradas a SDK para Android, iOS y Javascript, lo que permite gestionar diferentes aplicaciones sin salir de la plataforma.
- Posee gran poder de crecimiento. Gracias a la fácil gestión que tienen los usuarios sobre las aplicaciones es posible tener un gran crecimiento acorde a los objetivos planteados.⁵

2.3 FIREBASE AUTHENTICATION

Firebase Authentication es un sistema de autenticación que se basa en tokens proporcionando una fácil integración con casi cualquier plataforma. Algunas de sus características principales incluyen la autenticación por redes sociales y SMS. Este sistema posee gran variedad de ventajas, entre las principales están la facilidad de uso e integración con otros servicios de Firebase.

Este sistema permite implementar un inicio de sesión para ofrecer una experiencia más personalizada y a su vez brindar una mejor seguridad a los datos que los usuarios manejen en la app.

Al implementar este sistema de autenticación en la app, Firebase se encarga de administrar todo en cuanto a la sesión del usuario. Por ejemplo, se cerciora de que esta no se cierre en el momento de reiniciar la app o el navegador. También, es compatible con iOS, Android y la Web.

⁵ “¿Qué es Firebase? ¿Qué ventajas ofrece a nuestras apps?” [En línea] [Citado el 16 de agosto de 2022.] Disponible en la web: <https://www.nts-solutions.com/blog/firebase-que-es.html>

2.3.1 VENTAJAS DE FIREBASE AUTHENTICATION

- Ahorra tiempo en el desarrollo de métodos de servicios web para la autenticación. Ya que, evita la implementación de servicios, la toma de credenciales y otros medios de autenticación. Con Firebase no se debe programar demasiado. La información que debe resguardarse lo hará en el sistema de Firebase, y este administrará cada tipo de servicio de autenticación de manera eficaz.
- Brinda un análisis detallado sobre la información e interacción de los usuarios en la plataforma. Estos datos demográficos y análisis de usuarios lo convierten en un sistema demasiado eficiente para fines de mejora.

2.3.2 CARACTERÍSTICAS PRINCIPALES DE FIREBASE AUTHENTICATION

- Solución de autenticación directa: Se basa en un sistema de inicio de sesión completo para su desarrollo. Este sistema emplea direcciones de correo electrónico (E-mail) y contraseñas para el inicio de sesión y trabaja en todo tipo de aplicación web y móvil.
- Integración de proveedores de identidad federada: En adición a la autenticación de credenciales, uno de los mejores sistemas de autenticación (y quizás el más popular) es a través del uso de proveedores de identidad federada. Emplear los servicios de Firebase Authentication permite que los usuarios inicien sesión con las siguientes plataformas.
 - Google
 - Facebook
 - Twitter
 - GitHub
- Autenticación con número de teléfono: Firebase también permite realizar la autenticación por medio de sus números de teléfono a través de un SMS. ⁶

⁶ “¿Qué es Firebase Authentication?” [En línea] [Citado el 27 de agosto de 2022.] Disponible en la web: <https://blog.back4app.com/es/que-es-firebase-authentication/>

2.4 FIREBASE CLOUD FIRESTORE

Cloud Firestore es una base de datos de documentos NoSQL que se distingue por su permisividad y escalabilidad en la nube y que, además, permite guardar y sincronizar los datos necesarios para el desarrollo del sistema cliente-servidor, así como dispositivos móviles y otros.

Una de las mayores ventajas es que este producto está hecho para manejar cargas de trabajo sumamente exigentes de aplicaciones muy populares o usadas por muchos usuarios, lo que permite que pueda escalar automáticamente para soportar una app que se desee desarrollar

2.4.1 CARACTERÍSTICAS DE CLOUD FIRESTORE

- No requiere servidores: Es una base de datos completamente gestionada por Google, por lo cual, no depende del uso alguno de un servidor externo, lo que agiliza el proceso de escala de recursos con el fin de acoplarse a las demandas que requiera el cliente.
- Sincronización en tiempo real: Cloud Firestore ofrece los servicios de actualización y sincronización en tiempo real, gestionando y monitoreando la actividad que se lleve a cabo en la plataforma, por ejemplo, informes analíticos en tiempo real, contenidos, catálogos de productos, entre otros.
- Herramientas adicionales: Además de que puede integrarse a Firebase Authentication, Cloud Firestore proporciona otra serie de herramientas que facilitan el desarrollo de aplicaciones, como, por ejemplo, un modelo de recopilación y documento de datos, bibliotecas clientes web y móviles, y una capa de almacenamiento muy consistentes en comparación a otras bases de datos NoSQL.⁷

⁷ “¿Qué es Cloud Firestore?” [En línea] [Citado el 28 de agosto de 2022.] Disponible en la web: <https://keepcoding.io/blog/que-es-cloud-firestore/>

2.5 FIREBASE CLOUD STORAGE

Cloud Storage o también conocido como Firebase Storage ofrece la función de almacenamiento de objetos en Google Cloud Plataform y permite el uso compartido de archivos tipo multimedia dentro de una app. Al enlazar Google Cloud Plataform a los servicios de Firebase, se obtiene disposición de la seguridad Google lo que protege cualquier carga o descarga en su aplicación.

2.5.1 VENTAJAS DE FIREBASE CLOUD STORAGE

La ventaja fundamental radica en que este producto permite subir archivos a la nube, guardarlos y compartirlos con otros usuarios. Además, garantiza al usuario el uso compartido de datos multimedia de forma segura.

Otras ventajas o características que destacan de este servicio son:

- Transferencia de datos sencilla a Cloud Storage.
- Permitividad de clasificación de archivos.
- Las clases de almacenamiento para cualquier carga de trabajo.
- La capacidad de almacenamiento es escalable, por lo tanto, no hay preocupación en llegar a un límite o tope de almacenamiento.⁸

⁸ “Firebase Storage: qué es y cómo funciona” [En línea] [Citado el 29 de agosto de 2022.] Disponible en la web: <https://keepcoding.io/blog/que-es-cloud-firestore/>

3. ENLACE ANDROID STUDIO – FIREBASE

Antes de comenzar con el diseño y programación de la aplicación móvil, se debe crear el proyecto (en donde se resguardarán los datos de la app) en Firebase y realizar el enlace entre el proyecto en Firebase y la app en Android Studio.

Una vez se haya hecho el registro y logueo en Firebase se busca la opción “Agregar proyecto”, luego, se siguen las instrucciones que muestra Firebase tal cual como las menciona. Estas instrucciones incluyen la adición de los repositorios de Firebase; como líneas de código en el archivo “build.gradle (module: app)” y en el “build.gradle (Project)” de la aplicación en Android Studio.

Realizar los enlaces de los servicios de Firebase con Android Studio es más sencillo de lo que parece debido a las constantes actualizaciones de este software. A continuación, se muestran los pasos a seguir para realizar el enlace entre el Firebase Authentication – Android Studio desde este último.

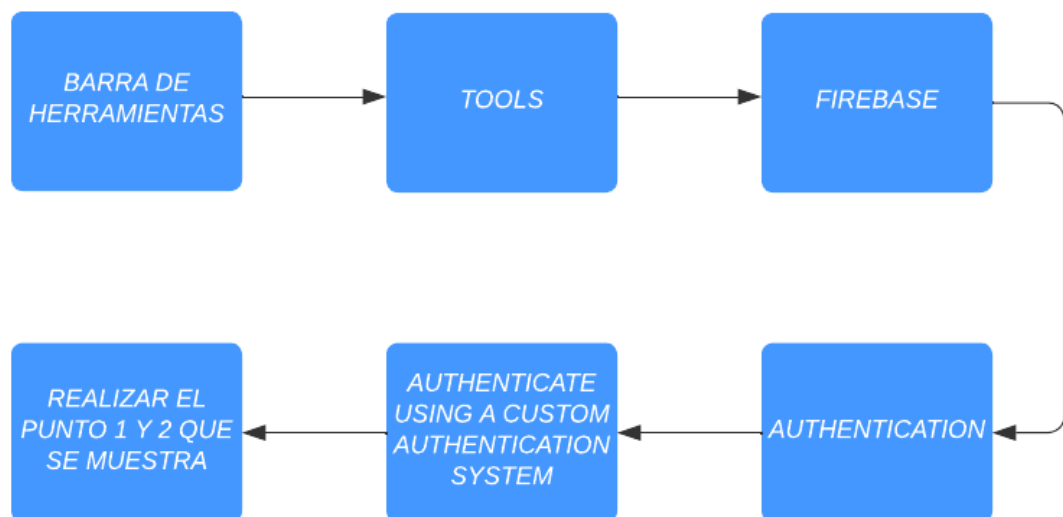


Figura 1. Paso a paso enlace Firebase Authentication – Android Studio
Fuente: Autor

De igual manera, el enlace entre los demás servicios como Cloud Storage y Cloud Firestore a Android Studio son bastante similares. En las siguientes figuras se muestra el paso a paso para realizar dichos enlaces.

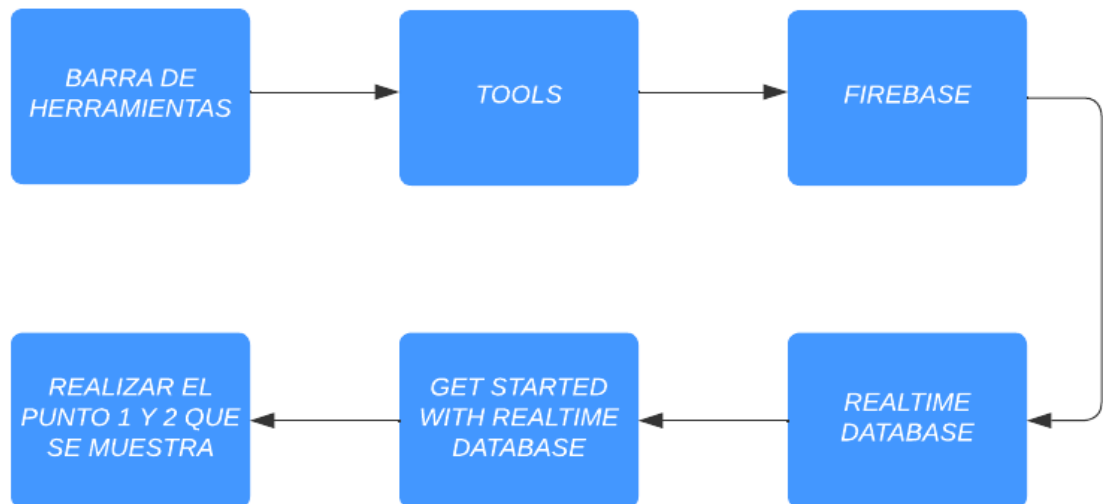


Figura 2. Paso a paso enlace Cloud Firestore – Android Studio

Fuente: Autor

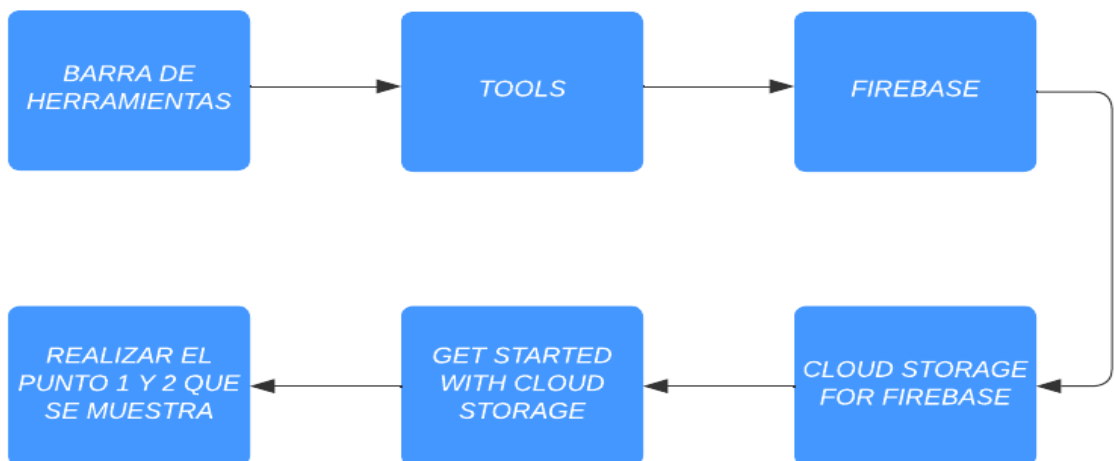


Figura 3. Paso a paso enlace Cloud Storage – Android Studio

Fuente: Autor

Se puede verificar que los enlaces fueron exitosos ya que Android Studio indica si lo fue. En la figura siguiente se muestra el indicador del software de Android cuando se encuentra correctamente enlazado a los servicios de Firebase, más exactamente al servicio de autenticación.

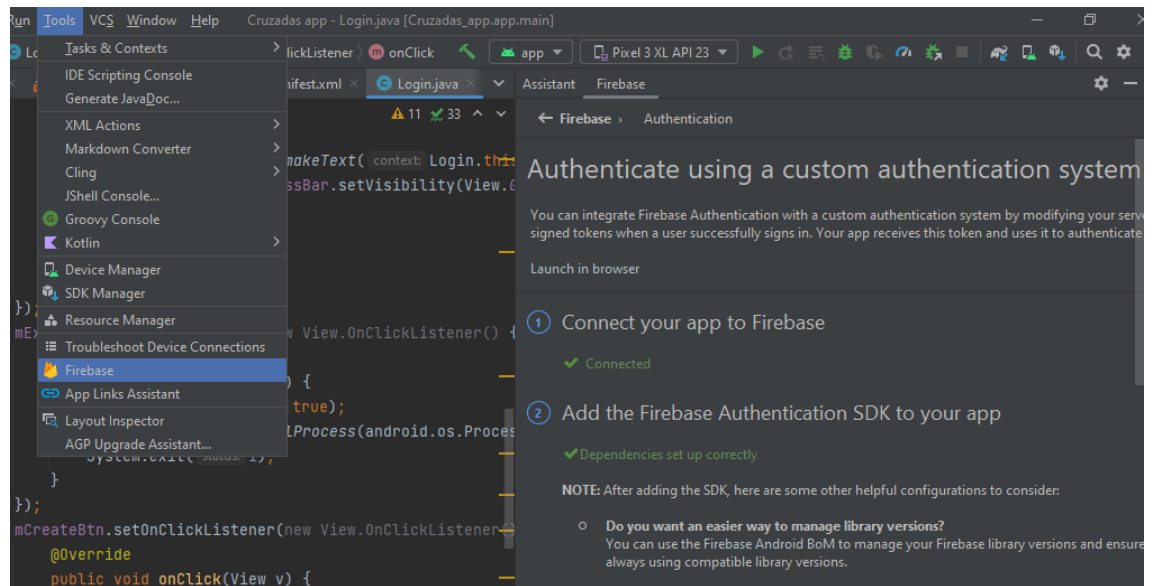


Figura 4. Indicativo de enlace exitoso con Firebase Authentication
Fuente: Autor

En síntesis, ya se tienen enlazados al software de desarrollo los servicios que se requieren para el funcionamiento de la aplicación. También, se tiene creado el proyecto en Firebase en donde Ingelec administrará la información que se ingrese, a su conveniencia.

4. PANTALLA DE REGISTRO DE USUARIO

Esta pantalla como su nombre lo indica, tiene por función el registro de usuarios a través del diligenciamiento de una serie de datos como lo son Nombre del usuario, Correo electrónico, Contraseña y Número de teléfono. El registro se realiza de una manera sencilla e inmediata, sin la necesidad de una verificación por correo electrónico o SMS para hacer del registro algo rápido y eficaz.

4.1 DISEÑO DE PANTALLA DE REGISTRO DE USUARIO

El diseño de esta pantalla y de la aplicación en general cuenta con unos colores allegados a los que maneja la empresa Ingelec en su logo y un fondo referente en cuanto a tecnología. Estos colores son una mezcla entre azul, blanco y naranja.

Esta ventana de la aplicación cuenta con una serie de elementos pensados para una interfaz de usuario cómoda, y fácil de entender para el usuario.

En primera instancia se genera y posiciona un ImageView en la parte superior de la ventana y en este elemento se adjunta el logo de la empresa Ingelec S.A.S. Luego, bajo el objeto anterior se agrega un TextView el cual le indica al usuario que debe crear una cuenta nueva.

Posteriormente, se crean una serie de EditText en los cuales se solicita por medio de hints (mensaje que se borra a pulsar el elemento) al usuario ingresar una información puntual; esta información incluye Nombre del usuario, E-mail, Número de teléfono y Contraseña. Estos EditText tienen cualidades particulares, por ejemplo; El cuadro para E-mail es un EditText de Input Type "E-mail" para que en su llenado aparezca en el teclado el símbolo "@". También, el EditText Teléfono es de Input Type "Number" para que en su diligenciamiento aparezcan sólo valores numéricos. Por otra parte, el EditText Contraseña tiene un Input Type "password" para que a medida en que este campo se llena, los caracteres que aparezcan en pantalla se muestren ocultos. Así, "*****".

Una vez se tiene la información requerida para el registro, se crea un elemento Button (botón) con el mensaje “Registrarse” El cual al ser oprimido por el usuario enviará la información a la base de datos de Firebase en donde quedará guardada. Durante el ajuste del elemento anterior se presentó un inconveniente, el cual no permitía que el botón cambiara de color al que se le asignaba. Por lo tanto, hubo que hacer una corrección en el archivo AndroidManifest.xml. En este archivo se realiza un cambio en una de las primeras líneas de código ya que al parecer tenía un bug que no permitía cambiar el color de ningún elemento Button. A continuación, se muestra la línea que se modificó para solventar el problema ya mencionado.

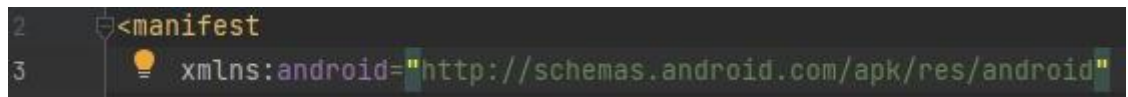
A screenshot of a code editor showing the first two lines of an AndroidManifest.xml file. Line 2 contains the opening tag for the manifest: `<manifest`. Line 3 contains the namespace declaration for Android: `xmlns:android="http://schemas.android.com/apk/res/android"`. The text is highlighted in a dark theme with syntax coloring.

Figura 5. Línea de código modificada en archivo manifest

Fuente: Autor

Como opción extra, se crea un TextView con el mensaje “¿Ya tienes una cuenta? Ingresa aquí”. El fragmento del mensaje que dice “Ingresa Aquí” tendrá propiedades de un elemento Button. Esto para que el usuario en caso de ya estar registrado pueda transportarse a la pantalla de logueo presionando dicho mensaje. Y se agrega un Progressbar que se muestra en pantalla únicamente durante el tiempo que tarde la ejecución del registro.

Una vez configurados, con sus respectivos ID y ajustados los elementos previamente mencionados a la pantalla de registro, se obtiene como resultado la interfaz que visualizará el usuario al momento de realizar el registro en el aplicativo. En la siguiente figura se muestra el diseño final de la pantalla de registro y sus elementos señalados.



Figura 6. Diseño de pantalla de registro
Fuente: Autor

Ya con el diseño de la ventana de registro listo se procede a realizar la programación debida para que cada elemento ejecute la acción que le corresponda de manera rápida y eficaz.

4.2 PROGRAMACIÓN DE PANTALLA DE REGISTRO DE USUARIO

Se inicia el proceso de programación de la aplicación enlazando los elementos del archivo xml al archivo java. Primero, se ubica dentro de la clase en la que se va a realizar todas las tareas de esta pantalla. Inicialmente, se establecen las variables acordes a su elemento (EditText, Button, entre otros), y posteriormente se enlazan dichas variables a los elementos del archivo xml a través de sus respectivos ID's con el comando findViewById.

El trabajo principal que se realiza en esta pantalla se hace al momento de oprimir el botón de "REGISTRARSE". Por lo tanto, se cita este elemento con su variable previamente asignada seguido de un setOnClickListener para generar el método (public void) que pone en ejecución las funciones de los demás elementos al momento de oprimir el botón previamente mencionado.

Una vez ubicados dentro del método del botón de registro, se procede a establecer una serie de condiciones para los elementos EditText, las cuales se activan cuando hay un elemento sin diligenciar. Estos condicionales se hacen con el fin de evitar campos vacíos al momento de registrarse. También, se agrega un condicional extra para el EditText de la contraseña, esta condición obliga al usuario a ingresar una contraseña que tenga más de 6 caracteres. Además, se le ordena a la Progressbar hacerse visible durante el tiempo que tome subir los datos digitados a Firebase.

La siguiente figura muestra cómo se establece la condición para evitar que un campo quede vacío y que muestre un mensaje solicitando llenar el campo y que contenga una cantidad de caracteres mayor a 6 en el EditText de la contraseña.

```
if (TextUtils.isEmpty(password)){  
    mPassword.setError("Ingrese la contraseña");  
    return;}  
if (password.length() < 6){  
    mPassword.setError("La contraseña debe tener al menos 6 caracteres");}
```

Figura 7. Condición de ingreso y seguridad de contraseña

Fuente: Autor

Ya que el usuario está obligado a llenar todos los datos que se solicitan para el registro se procede a implementar el servicio de Firebase Authentication y Cloud Firestore para la adquisición y salvaguardar los datos del usuario.

Una vez ingresados los datos, se envían a la base de datos para crear el usuario en la parte de autenticación de Firebase. A esto, se le agrega una condición a partir de los resultados de la autenticación. Si la condición se cumple, se creará una carpeta nueva en Cloud Firestore con el nombre de “Usuarios” donde quedará guardada la información de este (nombre, teléfono, email) y los organizará por UID para evitar sobreescrituras en caso de nombres similares. También, mostrará un mensaje informando que el usuario ha sido creado y lo transportará a la pantalla de logueo. De lo contrario (si la autenticación no es exitosa), se mostrará en pantalla un anuncio mencionando un error en el registro y la Progressbar se dejará de mostrar.

Para finalizar con el desarrollo de la pantalla de Registro, se modifica el TextView “¿Ya tienes una cuenta? Ingresa aquí” de la parte inferior para que éste al ser presionado envíe al usuario nuevamente a la pantalla de logueo.

Como el funcionamiento general de la aplicación móvil que se está desarrollando requiere el uso de red wifi para la conexión cliente-servidor se solicita el permiso al dispositivo para el uso de internet en el archivo AndroidManifest.xml de la siguiente manera: `<uses-permission android:name="android.permission.INTERNET"/>`

Tan pronto como el usuario se registre los datos ingresados se suben al servidor (Firebase) en donde únicamente la empresa Ingelec tiene acceso para visualizar y administrar la información que se deposite allí.

Se realizan cuatro pruebas en el proceso de registro para verificar el funcionamiento de la aplicación móvil en la pantalla ya desarrollada junto con el enlace entre la app y la base de datos. También se comprueba desde el servidor que los datos ingresados sean los correctos y estén a disposición de modificaciones de ser necesarias.

En las siguientes figuras se muestran los resultados de las pruebas de registro realizadas. También, se visualiza cómo Cloud Firestore y Firebase Authentication organizan la información ingresada de manera sencilla para que el administrador de la aplicación (Ingelec) pueda eliminar o modificar cualquier elemento.

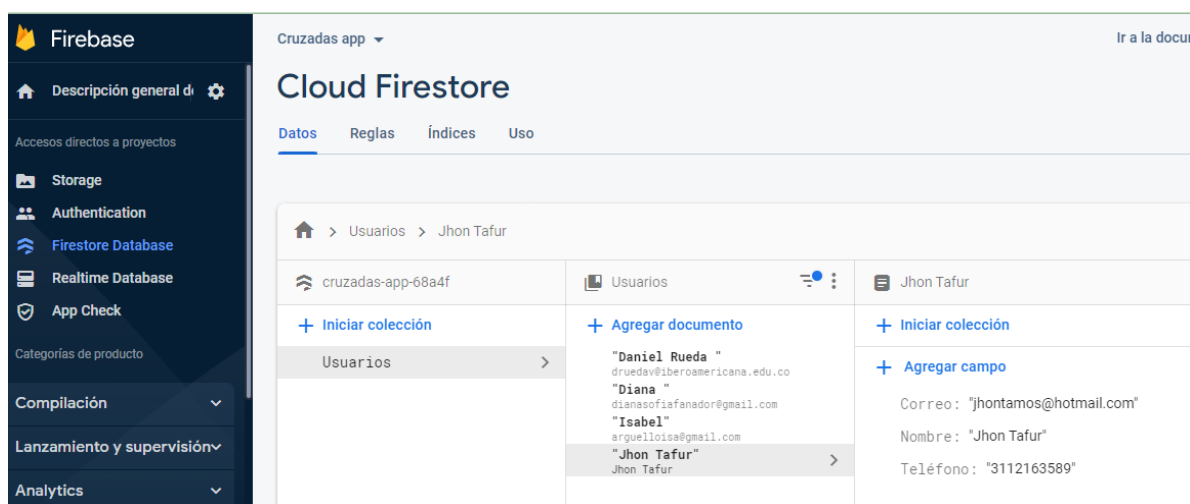


Figura 8. Datos de usuarios registrados en Cloud Firestore
Fuente: Autor

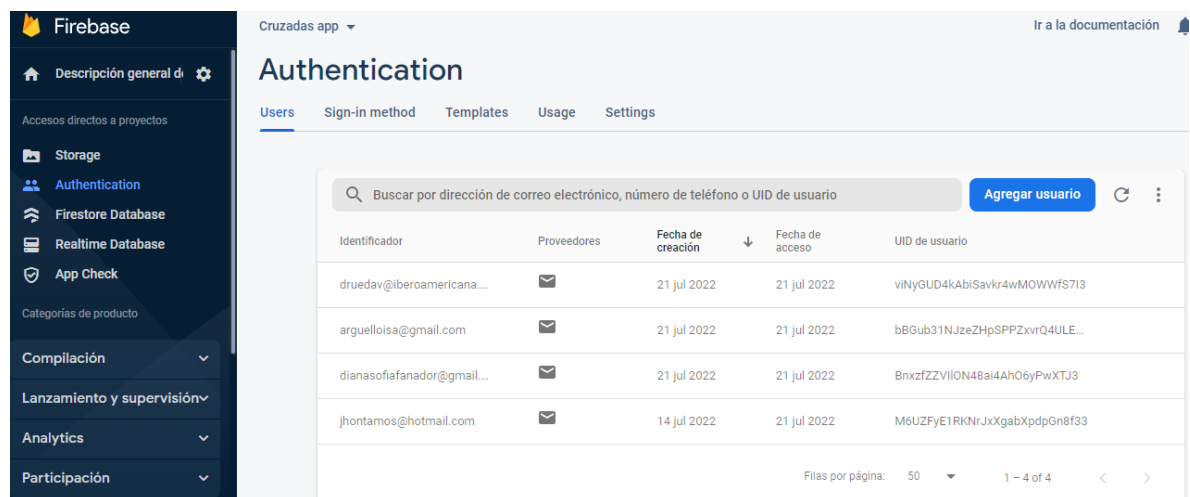


Figura 9. Usuarios registrados en Firebase Authentication
Fuente: Autor

Firebase Authentication genera un Unique ID (UID) al momento de realizar la autenticación en pro de diferenciar los usuarios con este ID para distinguir un usuario de otro en caso de requerirse para alguna tarea. Para identificar los usuarios de manera más fácil, Firebase permite organizar y mostrar los usuarios por cualquiera de sus propiedades, ya sea Nombre, E-mail o Teléfono como se ve en la figura 8.

5. PANTALLA DE LOGUEO

La ventana de logueo es la pantalla que el usuario visualiza al momento de ingresar en la aplicación. Esta ventana solicita digitar el Email y contraseña previamente registrados para poder ingresar en la sección de ingreso de datos (la cual se menciona más adelante). También, cuenta con una serie de imágenes en las que se pueden ver nuevamente el logo de Ingelec, y las certificaciones ISO con las que cuenta la empresa. Además, se incluye el servicio de restablecimiento de contraseña a través de correo electrónico en caso de ser olvidada por el usuario.

5.1 DISEÑO DE PANTALLA DE LOGUEO

Al igual que la pantalla de registro, esta ventana está diseñada con colores que tienen relación al logo de Ingelec. Además, de una estructura relativamente sencilla y fácil de entender por el usuario.

En la parte superior se tiene el logotipo de Ingelec en un elemento ImageView, seguido del título del proyecto al que va dirigida la aplicación móvil como TextView. Bajo el título se ubican dos EditText (uno tipo E-mail y el otro tipo password) en los cuales se debe digitar el correo electrónico y contraseña para poder realizar el logueo.

Se añaden dos botones; uno para salir de la aplicación (“SALIR”) y otro para verificar si el correo y contraseña ingresados coinciden y existen en la base de datos, y de ser así, permite realizar el logueo (“INGRESAR”).

Bajo los botones anteriores se ubican dos mensajes con propiedades Button. En el primero se encuentra el mensaje “¿No tienes cuenta? Regístrate Aquí” en donde al ser presionado se direcciona al usuario a la pantalla de registro. Y en el segundo se observa el mensaje “¿Olvidaste tu contraseña? El cual al seleccionarse solicita al usuario el correo electrónico que registró para posteriormente enviarle un Email en donde podrá reestablecer la contraseña en caso de haberla olvidado.

Se finaliza la parte visual de esta pantalla agregando al final de este una imagen con las certificaciones ISO con las que Ingelec cuenta. A continuación, se muestra el resultado final del diseño de la pantalla de logueo.



Figura 10. Diseño de pantalla de logueo
Fuente: Autor

5.2 PROGRAMACIÓN DE PANTALLA DE LOGUEO

Como ya se tiene realizada la parte de registro del usuario de manera funcional y el diseño de la pantalla de logueo, se procede a realizar la programación del funcionamiento de esta pantalla.

Se inicia creando las variables acordes a sus atributos y posteriormente se relacionan con el ID de su item en el archivo xml. Después, se convierten las variables de email y contraseña en String para posteriormente dar instrucciones que se ejecutan al momento de presionar el botón de “INGRESAR”.

Dentro del funcionamiento de dicho botón se agregan unos condicionales para obligar al usuario a ingresar el correo electrónico y contraseña previamente registrados (para no dejar campos vacíos). Y como añadidura, se agrega un progressbar que se mostrará mientras se verifican los datos digitados.

Así, al momento de oprimir “INGRESAR” se le ordena a la variable que enlaza Android Studio con Firebase realizar el inicio de sesión a partir de un correo y contraseña. Esta se crea junto a un `addOnCompleteListener` listener como una nueva tarea (con un `if task.isSuccessful`) para que de ser correcta la información digitada el usuario se ejecute correctamente el logueo y sea enviado a la pantalla de diligenciamientos de datos de la actividad en campo. En caso de no coincidir el correo y/o contraseña ingresados con las registradas en la base de datos, se mostrará un enunciado con el mensaje “Email y/o contraseña incorrecta” y se desactivará la visibilidad del progressbar.

Por otro lado, se programa el botón “SALIR” con la sencilla tarea de cerrar la aplicación y enviar al usuario de vuelta al escritorio de su dispositivo al momento de presionar este botón.

Continuando con los EditText tipo Button se le ordena al primer mensaje (¿No tienes cuenta? Regístrate Aquí”) la sencilla tarea de enviar al usuario a la pantalla de registro.

Para finalizar con el desarrollo de esta ventana de logueo, se agrega el servicio de cambio de contraseña.

En la parte inferior de la pantalla se muestra un TextView con el mensaje “¿Olvidaste tu contraseña?”, a este TextView se le agregan las propiedades de un Button para que al oprimirse realice lo siguiente:

Mostrará un anuncio preguntando al usuario si ha olvidado la contraseña. Esto seguido un cuadro de texto solicitando el correo de la cuenta de la cual ha olvidado la contraseña. Se observan también dos opciones. La opción “Sí” genera una nueva variable que será la del email que ingrese el usuario. A este email se le enviará un correo con un enlace en el cual podrá cambiar la contraseña de cuenta. El usuario al abrir el correo electrónico de restablecimiento deberá ingresar al enlace adjunto el cual le solicitará digitar una nueva contraseña.

Se crea la función onSuccess para que de ser exitoso el envío del correo se muestre un cuadro de dialogo informando que el correo se ha enviado con éxito. De lo contrario, también se mostrará un anuncio informando que se presentó un error en el envío del correo. Si en el anuncio del servicio de reestablecer contraseña se presiona la opción “No” este simplemente se cerrará. La siguiente imagen muestra cómo se visualiza el servicio de restablecimiento de contraseña en la aplicación.

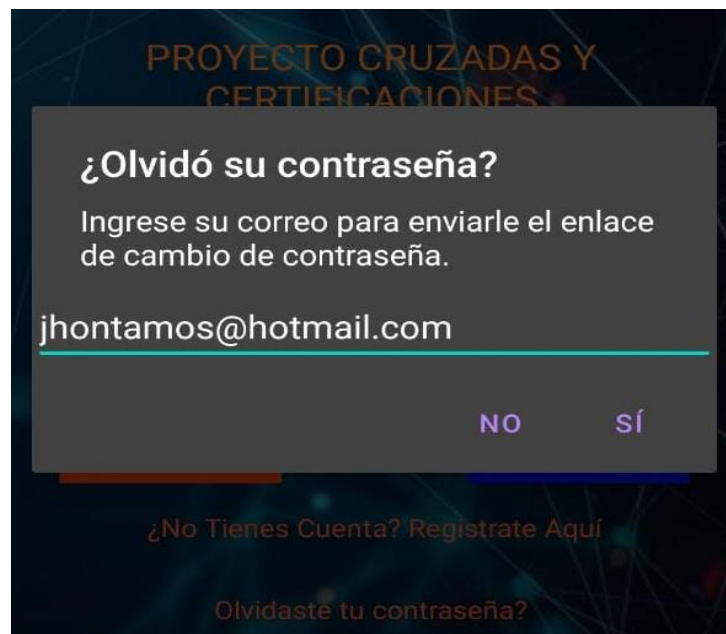


Figura 11. Servicio de restablecimiento de contraseña
Fuente: Autor

6. PANTALLA DE REGISTRO DE DATOS

Ya con el sistema de registro y logueo funcionando correctamente se inicia el desarrollo de lo que sería la parte principal de la aplicación. Cabe mencionar que en un principio los datos se iban a ubicar en Google Drive con la ayuda de Google Sheets. Sin embargo, a medida en que se desarrollaba e investigaba sobre bases de datos y sistemas cliente-servidor se encontró que Firebase ofrece todos estos servicios de manera gratuita, confiable y eficaz. Por ello, se optó por salvaguardar todos los datos en Firebase.

Esta ventana de la aplicación es donde el técnico que realiza labores en campo debe digitar toda la información que se le solicita sobre la actividad que realice ya sea cruzada o certificación, por lo que, el diligenciamiento de datos es de carácter obligatorio.

6.1 DISEÑO DE PANTALLA DE REGISTRO DE DATOS

La estructura visual de esta pantalla es la más compleja de todas debido a que cuenta con muchos más elementos que las anteriores. Por ello, antes de agregar cualquier elemento en esta, se debe incorporar un Scrollview que abarque toda esta pantalla y permita realizar un desplazamiento vertical para poder adjuntar todos los demás elementos.

Aquí, la información se divide en dos partes; registro de datos por escrito y registro fotográfico. Para el registro de datos escritos se usan en su mayoría EditText sencillos similares a los que se agregaron en la pantalla de registro de usuario. Las características de llenado de estos cuadros de texto dependen de la información que se deposite en ellos. Por ejemplo: Si un cuadro solicita información alfabética sencilla como el municipio en el que se realiza la actividad, el input type será normal. Por su parte, si lo que se solicita es información que sólo se da en valores numéricos como un ID, el input type será "number". Por otro lado, el EditText que solicita la fecha se modifica con características Date para que el usuario tenga la opción de seleccionar la fecha de manera más rápida en lugar de escribirla digito por digito.

Algunos de los elementos de esta pantalla que registran datos son los spinner. Estos elementos despliegan la cantidad de datos que se les programe. Para este caso, se utilizan en la selección de departamento, de actividad realizada y de equipo instalado.

Para el registro fotográfico se añaden cuatro ImageView enlazados a cuatro Buttons. Estos botones llevan al usuario a su galería de imágenes para que elija cual foto desea adjuntar. Las fotos que escogidas se mostrarán en los ImageView para que el usuario tenga la certeza de la foto que se subirá al servidor. Cabe resaltar que el llenado de las dos primeras imágenes es de carácter obligatorio.

Para finalizar, se agregan dos botones al final de la pantalla. El primero para cerrar sesión y volver a la pantalla de logueo. Y el segundo para subir los datos previamente ingresados al servidor. Las siguientes figuras muestran el diseño final de la pantalla de diligenciamiento de datos.

Cruzadas app

INGELEC GROUP TECHNOLOGIES

FORMATO DE EJECUCION PROYECTO CRUZADAS Y CERTIFICACIONES

INFORMACION GENERAL

Actividad realizada

Fecha de Visita

Departamento

Municipio

Bts a Visitar

BTS Conectante

Nombre del Técnico

ID (últimos 4 números)

OT

Hora de Llegada Hora de Salida

ACTIVIDADES EJECUTADAS

Equipo Instalado

Vlans probadas (si aplica)

Nemónico de Origen

Nemónico de Destino

¿Actividad Exitosa?

¿Actividad Exitosa?

CARGAR IMAGEN

CARGAR IMAGEN

CARGAR IMAGEN

CERRAR SESION **GUARDAR DATOS**

Figura 12 y 13. Diseño de sección de registro de datos y registro fotográfico
Fuente: Autor

6.2 PROGRAMACION DE PANTALLA DE REGISTRO DE DATOS

Como se ha venido haciendo para las demás pantallas, se inicia declarando variables y enlazándolas a sus respectivos elementos en el archivo xml a excepción de los spinner. Para el desarrollo de esta ventana se agregan unas variables extras para realizar y diferenciar el proceso de carga de cada imagen en los ImageView. Ahora, se procede a programar elemento por elemento.

Primero, se asignan instrucciones a los cuatro elementos spinner existentes, y el contenido que deben mostrar al desplegarse. Para ello, se crea la lista de elementos con un Array de String en la que se ingresan los valores que mostrará el spinner al ser desplegado. Dentro de esta lista, como primera opción se escribe el requerimiento que se le solicita al usuario seleccionar, seguido de las opciones entre las que debe elegir. Esto, debido a que el spinner muestra automáticamente la primera opción a elegir por defecto. Más adelante con la ayuda de unos condicionales se obliga al usuario a seleccionar una opción válida.

Después, se realiza el enlace a través de ID entre la variable que tendrá el spinner y el elemento spinner en el archivo de diseño xml para proseguir generando la vista de la lista de opciones con la ayuda del ArrayAdapter en el item spinner, a partir de los datos almacenados previamente en el Array de String. Se configura para que dicha lista se muestre de manera desplegable y que el spinner muestre y mantenga a la vista la opción que se seleccione.

Como los spinners muestran por defecto el texto en color negro, y para la comodidad visual del usuario se requiere que este sea blanco, se crea un `onItemSelectedListener` para darle atributos de `TextView` al spinner y así poder cambiar su color de texto.

Continuando con la programación, se toma el EditText en el que se debe ingresar la fecha de visita a la BTS y dentro de un setOnClickListener se genera un método onClick en donde se agrega el servicio de DatePickerDialog (Cuadro de selección de fecha) con las variables de año, mes y día. Así, el usuario puede elegir de manera rápida y dinámica la fecha de visita. Por último, se toma la selección tomada por el usuario y se muestra en el TextView en el orden Día/Mes/Año. La siguiente figura muestra cómo se visualiza la selección de fecha dentro de la aplicación.

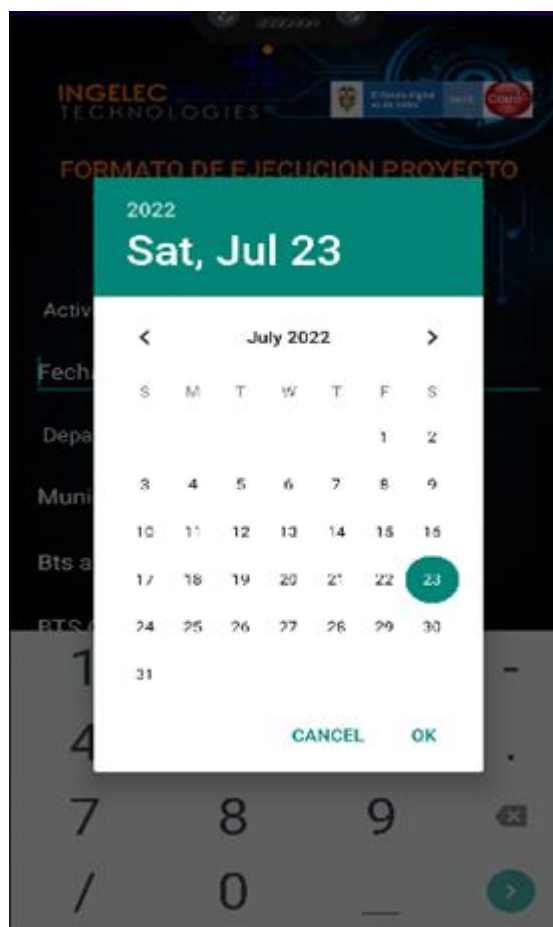


Figura 14. Ventana de selección de fecha
Fuente: Autor

Por otra parte, la adquisición de datos de los EditText sencillos (en donde el usuario sólo debe digitar de manera tradicional lo que se le solicite) se realiza en un método privado de ingreso de datos en donde la información digitada se convierte en datos string y por medio de una serie de condiciones el técnico es obligado a llenar cada uno de estos espacios o de lo contrario, si queda algún espacio en blanco, cada vez que se oprima el botón de guardar datos, se generará un mensaje notificando al usuario que “x” espacio está en blanco y no

podrá guardar la información hasta llenar cada dato. Cuando ya se tienen toda la información requerida diligenciada, se crea un Hashmap dentro de la carpeta “Actividad” en la base de datos de Firestore en donde quedan guardados y a disposición de Ingelec todos los datos diligenciados por el técnico en campo.

Prosiguiendo con el desarrollo de los demás elementos de la pantalla de ingreso de datos, se toman ahora los ImageView juntos a los Button que dan la opción de cargar la imagen desde multimedia. Para ello, se crean los métodos (uno para cada imagen) que realizan la tarea de dirigir al usuario a multimedia; en estos métodos se generan intents nuevos, los cuales se configuran para que sean de tipo imagen y se les asigna la tarea de obtener contenido a partir de un método de selección que ubica elementos del tipo de intent previamente configurado. Estos métodos se citan dentro de sus respectivos botones de cargar imagen, para que al ser presionado el botón se ejecute su método asignado.

Para que la imagen que se seleccione en multimedia se ajuste y muestre en el ImageView se genera un protected void en donde se toma la imagen seleccionada y se procesa en un bitmap para que esta se ajuste y visualice en el cuadro de imagen que le corresponda.

Por último, para salvaguardar en el storage de Firebase las imágenes cargadas se agregan unas líneas de código en el método de ingreso de datos mencionado con anterioridad. Aquí, se ordena que las imágenes cargadas se guarden en el Storage de Firebase en una carpeta de nombre “Actividades”. Seguido a esto, se generan una secuencia de carpetas extras, la primera con el nombre de la BTS que se visitó, seguido de otra con la fecha de la visita y dentro de ésta última, se ubicarán las imágenes las cuales tendrán por título un UID (unique ID). Lo anterior se hace con el propósito de evitar sobre escrituras y pérdidas de información. También, a medida en que cargan las imágenes se muestra un cuadro de texto con el porcentaje de carga de la imagen y otro cuadro adicional notificando si la foto fue subida a la base de datos de manera exitosa o no.

Se realiza una prueba para verificar que la aplicación esté funcionando correctamente en su totalidad. En figuras anteriores se muestra el correcto funcionamiento y guardado de los usuarios a realizar la autenticación. Por lo que, ahora procede a testear el diligenciamiento de datos. En las siguientes figuras se muestra cómo se guarda y clasifica la información escrita y fotográfica en la base de datos de Firebase con la prueba realizada.

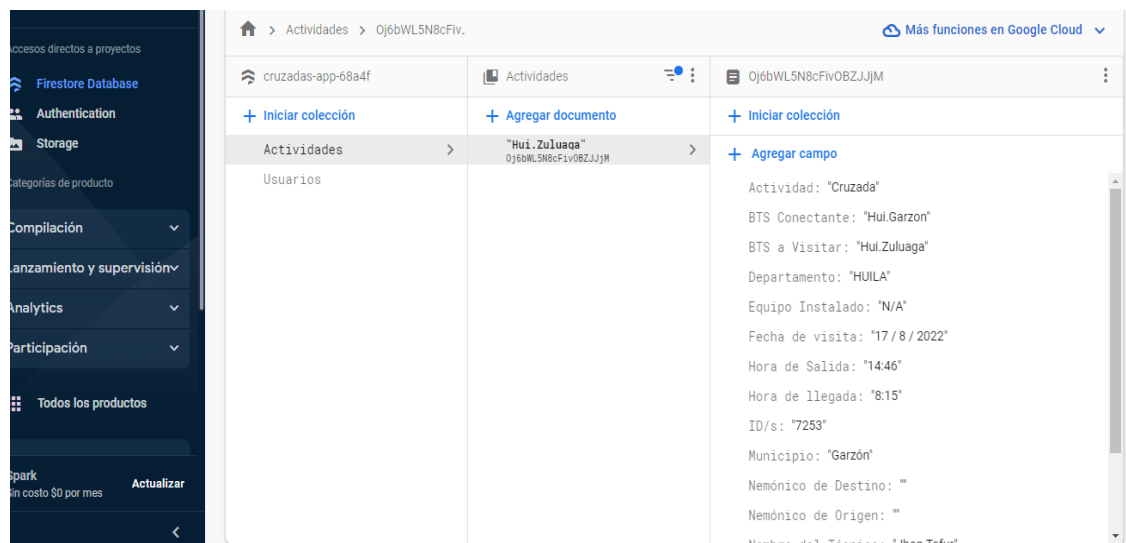


Figura 15. Datos escritos almacenados en Firebase
Fuente: Autor

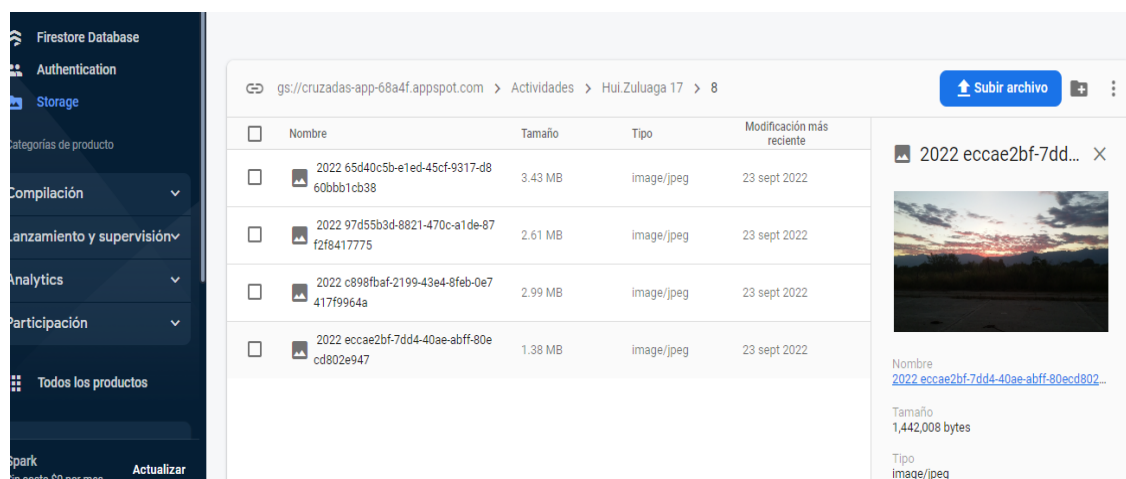


Figura 15. Datos fotográficos almacenados en Firebase
Fuente: Autor

Finalmente, se valida que la aplicación móvil queda funcional y lista para ser empleada por la empresa Ingelec en el proyecto Claro pruebas y cruzadas en sitios conectantes -ie- mintic 7k con el fin agilizar y tener como respaldo registro escrito y fotográfico de toda actividad realizada en dicho proyecto.

7. CONCLUSIONES

- Firebase ofrece una gran variedad de servicios para el desarrollo de aplicaciones móviles de manera gratuita y sencilla. Además, garantiza la seguridad de la información que se deposite en esta base de datos. Por lo anterior, para el desarrollo de esta aplicación se decidió hacer uso de los servicios de Firebase en lugar de los de Google Sheets.
- Para el desarrollo de esta aplicación móvil se utilizó el software libre para desarrollo de aplicaciones móviles Android Studio como se tenía previsto en los objetivos de este proyecto, ya que es un software que además de ser gratuito, incluye una gran cantidad de herramientas eficaces que facilitan el desarrollo de aplicaciones para dispositivos iOS y Android.
- El proyecto Claro pruebas y cruzadas en sitios conectantes -ie- mintic 7k generalmente se realiza en zonas muy retiradas del casco urbano, siendo el tiempo un factor esencial para el personal en campo. Ingelec tradicionalmente maneja un formato de actividades en excel el cual ocupa demasiado tiempo en ser diligenciado. Por lo que, para seguridad de ellos, a través de este aplicativo pueden entregar de manera mucho más rápida los formatos de las actividades que realicen, incluso tienen la facilidad de diligenciar el formato una vez hayan llegado a casco urbano.
- Uno de los detalles más importantes en el desarrollo de la aplicación fue la implementación del UID en los usuarios, actividades e imágenes para su archivamiento. Ya que, garantizan que no se vayan a generar sobreescrituras que puedan ocasionar pérdidas de información.
- El servicio de restablecimiento de contraseña puede parecer muy simple, ya que, se realiza de manera muy rápida. Esto es con el fin de que el técnico en campo ocupe la menor cantidad de tiempo cuando utilice la aplicación.

8. BIBLIOGRAFIA

“Introducción a Android Studio”, [En línea] [Citado el 16 de agosto de 2022.] Disponible en la web: <https://developer.android.com/studio/intro?hl=es-419>

“Un poco de historia de Firebase”, [En línea] [Citado el 16 de agosto de 2022] Disponible en la web: <https://openwebinars.net/blog/que-es-firebase-de-google/>

“Los 5 lenguajes más usados para la programación en Android” [En línea] [Citado el 16 de agosto de 2022.] Disponible en la web: <https://keepcoding.io/blog/programacion-en-android/>

“¿Qué características tiene Android Studio?” [En línea] [Citado el 16 de agosto de 2022.] Disponible en la web: <https://talently.tech/blog/que-es-android-studio/>

“¿Qué es Firebase? ¿Qué ventajas ofrece a nuestras apps?” [En línea] [Citado el 16 de agosto de 2022.] Disponible en la web: <https://www.nts-solutions.com/blog/firebase-que-es.html>

“¿Qué es Firebase Authentication?” [En línea] [Citado el 27 de agosto de 2022.] Disponible en la web: <https://blog.back4app.com/es/que-es-firebase-authentication/>

“¿Qué es Cloud Firestore?” [En línea] [Citado el 28 de agosto de 2022.] Disponible en la web: <https://keepcoding.io/blog/que-es-cloud-firestore/>

“Firebase Storage: qué es y cómo funciona” [En línea] [Citado el 29 de agosto de 2022.] Disponible en la web: <https://keepcoding.io/blog/que-es-cloud-firestore/>