APLICACIÓN MÓVIL DE CONTROL DE ACCESO POR MEDIO DE RECONOCIMIENTO FACIAL

Paula Andrea Gómez Alzate

Ingeniera de Sistemas

Tutor

Danny Alexando Múnera Ramirez , Ingeniero electrónico

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería de Sistemas

Medellín, Antioquia, Colombia

2022

Tabla de contenido

LISTA DE TABLAS

LISTA DE FIGURAS

Siglas, acrónimos y abreviaturas

SDK	Kit de desarrollo de software

Introducción

En los últimos años la tecnología ha ido avanzando a pasos agigantados, trayendo consigo muchos beneficios para la sociedad en diferentes campos, la medicina, la ingeniería, la ciencia, entre otros. Esto ha impactado también a los sistemas de control de acceso, los cuales incorporan tecnologías de punta, estos permiten automatizar la entrada y salida de personal o vehículos en una empresa, con la finalidad de agilizar el registro de personal, garantizar la seguridad y tener una mejor administración (Bito Storage, 2022).

Existen empresas que poco a poco han incorporado estos sistemas, debido a que sus necesidades lo requieren. Los métodos de control de acceso tradicionales retrasan el ingreso a las empresas, causando largos tiempos de espera, ya que se debe llenar información relevante de forma manual. La empresa debe garantizar que sus empleados no inicien su jornada laboral sino tienen sus afiliaciones al día y además debe asegurar que las personas que ingresaban a las instalaciones sí pertenezcan a la empresa.

El control de acceso empezó a ser una problemática muy común en las empresas, lo cual dió iniciativa para crear diferentes tipos de sistemas de control de acceso, ya sean biométricos, de proximidad u otros. Para este semestre de industria, se propone la creación de una solución biométrica y de lectura de cédula colombiana que se desarrolla con la empresa Idelity, la cual es una empresa enfocada en la investigación y la prestación de servicios de reconocimiento biométrico. La biometría se puede definir como: “un sistema automático basado en la inteligencia artificial y el reconocimiento de patrones, que permite la identificación y/o verificación de la identidad de personas a partir de características morfológicas o de comportamiento, propias y únicas del individuo, conocidas como autentificadores” (Lisa Institute, 2021). Idelity ha liderado el desarrollo de múltiples proyectos en esta área los cuales han sido llevados de manera satisfactoria al mercado local

Con base en lo anterior y ante la necesidad que es latente dentro de las empresas, Idelity dentro de su visión de crecimiento busca generar un mecanismo que permita registrar, de manera

automática, la entrada y salida de personal. En este caso se plantea desarrollar una aplicación, utilizando la biometría como medio para asegurar un reconocimiento facial automático del empleador o visitante, evitando así la entrada de personal no autorizado, con información vencida o no actualizada.

1. Marco teórico

En este capítulo se va realizar una revisión de los elementos teóricos necesarios para la comprensión del presente manuscrito. Inicialmente se presentan los conceptos teóricos para desarrollo de aplicaciones móviles en android, seguido las tecnologías para la implementación de sistemas de control de acceso, por último la revisión de aplicaciones para el control de acceso.

1.1 Desarrollo de aplicaciones móviles en Android

Para comprender android es necesario entender algunos conceptos básicos como: el ciclo de vida de una actividad y la arquitectura. A continuación se da una breve explicación de cada uno de estos conceptos.

Ciclo de vida de una actividad: Una actividad es una pantalla de una aplicación, la cual permite interacción con el usuario. A lo largo de su vida útil pasa por unos estados que se usan con las siguientes funciones: onCreate, onStart, onResume, onPause, onStop y onDestroy. El método onCreate, se encarga de crear la actividad e inicializar las variables, es el primer estado del ciclo de vida, onResume es la segunda función que se ejecuta cuando se crea una actividad, además se pueden recuperar los datos del usuario que necesita persistir. Por último, mencionando solo las funciones más importantes, está OnDestroy, la cual se ejecuta cuando se destruye la actividad, es decir se cierra la aplicación o se elimina la actividad.

Patrón arquitectónico MVVM: Es un patrón de diseño que fue creado para separar la interfaz de usuario con la lógica, logrando así la independencia de la vista. El modelo representa la capa de datos y/o lógica de negocio. La vista presenta la información y es activa, reaccionando a cambios en el modelo. El modelo de vista es un actor intermediario entre el modelo y la vista y contiene toda la lógica de presentación (Izertis, 2019) .

1.2 Tecnologías para la implementación de sistemas de control de acceso

En esta sección se revisan las principales tecnologías para el desarrollo de un sistema de control de acceso. Inicialmente se aborda la identificación por lectura de código de barras y luego el uso de reconocimiento facial.

1.2.1 Barcode PDF417

Barcode se usa para el scanner del código de barras de las cédulas colombianas, permitiendo así entregar todos los caracteres que contiene la misma. Barcode es "un código de barras apiladas que se puede utilizar para codificar grandes cantidades de información a través de múltiples códigos. Cada patrón en un código de barras individual consiste exactamente de 4 barras y 4 espacios, y cada patrón es de 17 unidades de longitud" (Cognex, 2022). Existe una librería llamada Zxing que contiene Barcode, esta retorna una decodificación de todos los caracteres que se encuentran en el código de barras.

1.2.2 Sistemas de Reconocimiento Facial

El reconocimiento facial es una tecnología que se encarga de identificar o verificar la identidad de un sujeto por medio de una imagen. El reconocimiento facial es un método de identificación biométrica (como el uso de huellas dactilares), es decir, un método que utiliza las medidas corporales, en concreto de la cara, para identificar o verificar la identidad de una persona. (kaspersky, 2022)

Para el reconocimiento facial se siguen los siguientes pasos:

1. Detección de rostro
2. Captura de información y traducción de rasgos
3. Comparación y verificación

1.2.2.1 SDK - IDKit Android

Es un conjunto de herramientas que permiten el reconocimiento facial para android, por medio de un algoritmo ya implementado que contiene los modelos para el reconocimiento. Este SDK

funciona con una base de datos local que almacena inicialmente. Para su funcionamiento requiere al menos de una foto de una persona como registro. Luego, el reconocimiento se realiza al comparar la foto de evidencia con la registrada en el sistema. La función retorna un resultado que varía en un rango de 0 a 100, siendo 0 ninguna coincidencia y 100 un acierto total (es decir, la foto de evidencia coincide con una persona en la base de datos).

1.2.2.2 Detalles del SDK

Para hacer uso de este **SDK** es necesario integrarlo al android, ubicándolo en la carpeta raíz de la aplicación, luego se debe agregar en las dependencias estos archivos (modelo, librerías). Una vez se tengan sincronizadas estas dependencias, se crea un objeto **idKit** el cuál da acceso a todas las funciones del **SDK,** que permiten registrar y comparar las imágenes de los usuarios. Las funciones más usadas son **idkit.connect(),** se utiliza para hacer la conexión a la base de datos. El método **addFace()** y **setCustomData()** registra la imagen en la base de datos del **SDK**.

1.3 Revisión de Aplicaciones para el Control de Acceso

Con el objetivo de tener una visión acerca de las aplicaciones que hay en el mercado y de ponerle un valor agregado a la que se desea desarrollar, se investigaron cuáles son los sistemas de control de acceso que hay actualmente y cuáles son sus características. A continuación se presenta la descripción detallada de algunas aplicaciones relevantes.

Onboarding : Es una aplicación móvil que tiene las siguientes características: biometría facial y prueba de vida, validación de documento avanzada. Compara la imagen impresa en el documento de identidad con el selfie capturado durante el proceso y otorga una puntuación de similitud para determinar si es la misma persona (Veri.das, 2022).

UBio X Face: Este aplicativo móvil cuenta con reconocimiento facial en movimiento a una distancia de hasta 3 metros, anti-spoofing contra ataques, fotos, videos y máscara 3D, autenticación facial bajo iluminación hasta 25.000 lux, detección de rostro vivo, detección de temperatura, combinada con el reconocimiento facial. Enrolamiento y registro facial fácil (IdeasControl, 2019).

Falcon Cloud: Es una aplicación web que permite monitorear en tiempo real las entradas y salidas de una empresa" (Falcon Cloud, 2022). Se mencionan a continuación algunas de sus principales características:

- Plataforma Cloud de fácil acceso.
- Marcación por huella, rostro, tarjeta o clave.
- Marcación móvil con georeferenciación.
- Centralización de información de diferentes sucursales.
- Sincronización automática de dispositivos biométricos.
- Procesamiento rápido y automático de la información.

Discusión

Durante la revisión de aplicaciones para el control de acceso se encontró que la mayoría usan sistema de reconocimiento facial y enrolamientos de usuarios, algunas implementan validación de identidad por medio de tarjetas, claves y documentos. Sin embargo, en ninguna de las aplicaciones se encuentra un software a la medida, es decir que se acomode a las necesidades del cliente.

2. Metodología

Durante la ejecución de un proyecto es necesario llevar un orden o gestionar de manera adecuada los tiempos y tareas para el desarrollo, por esta razón existen las metodologías que suplen está necesidad. Entendiendo esto, para llevar a cabo esta práctica se utilizaron una combinación de metodologías clásicas y ágiles. Se tuvieron en cuenta las etapas de análisis, diseño, implementación y por último de validación. Además, inspirado en las metodologías ágiles, se optó por desarrollar reuniones diarias (*SCRUM daily meeting*), las cuales ayudaban a una mejor comunicación entre el equipo. También se definieron tres entregas intermedias (*sprints*). A continuación se muestra en qué consiste cada una de las anteriores etapas mencionadas.

- **Análisis**: Se trata de entender qué quiere el cliente, también se hace una recolección de requisitos o requerimientos del sistema.
- **Diseño**: Se diseña la estructura interna del software, se crean las entidades que van a estar relacionadas y el modelo de los datos, todo esto se logra luego de hacer la etapa de análisis.
- **Implementación**: En esta fase hay que elegir las herramientas adecuadas para el desarrollo de la aplicación. Un entorno de desarrollo que facilite el trabajo y un lenguaje de programación apropiado para el tipo de software a construir (Ungoti, 2022)
- **Validación:** En esta etapa se busca detectar los fallos cometidos en las etapas anteriores para corregirlos. Es una de las etapas más importantes, ya que se hacen todas las pruebas necesarias, tanto lógicas como visuales, para poner en producción (Ungoti, 2022)

3. Análisis y Diseño

Todo proyecto en desarrollo de software tiene un análisis y diseño previo a la implementación para garantizar un buen producto final. En esta sección se presentan los requerimientos funcionales, no funcionales, reglas de negocio y la arquitectura del sistema, con el fin de dar un contexto general de todo el sistema y el alcance del prototipo final.

3.1 Requisitos funcionales

El objetivo principal es lograr un prototipo final con todas las especificaciones del usuario final, para crear la aplicación de control de acceso se hace una recolección de requerimientos funcionales que se presentan a continuación:

RF1. La aplicación debe tener una pantalla de login, donde se permita ingresar usuario y contraseña para tener un control de clientes.
RF2. La aplicación debe tener un menú con las siguientes opciones: reconocimiento facial, lector código de barras, historial de registros y digitar cédula.
RF3. El sistema debe permitir crear visitantes con los datos: nombre, apellido, cédula, RH, tipo de persona, (empleado o visitante) fecha de ARL, seguridad social y fecha de la última dosis de la vacuna contra el COVID-19.
RF4. La aplicación debe validar que las fechas de ARL y seguridad social estén al día y si no es así, mostrar un color rojo en la pantalla, el cual permita avisar a la persona que manipule la aplicación, que tiene el acceso restringido.
RF5. La aplicación debe permitir subir información al servidor de los nuevos visitantes.
RF6. La aplicación debe guardar información sobre el acceso o salida de los empleados o visitantes de manera local. Se deben almacenar los siguientes datos: hora, fecha, confirmación de entrada, vehículo, placas, contenido, síntomas de covid y temperatura.
RF7. La aplicación debe subir la información de los registros locales cada día, con el fin de asegurar los datos en el servidor.
RF8 La aplicación debe permitir descargar trabajadores y guardarlos en una base de datos local, la cual permita la persistencia de estos.

RF9. La aplicación debe integrar el SDK de innovatrics para el control de acceso por medio de reconocimiento facial

RF10. La aplicación debe asociar a cada trabajador una foto para el acceso por reconocimiento facial.

RF11. La aplicación debe hacer un reconocimiento facial a los trabajadores al momento del ingreso o salida del lugar de trabajo.

RF12. La aplicación debe subir las fotos tomadas de todos los trabajadores al servidor, con el fin de que sea un recurso compartido con otros dispositivos.

RF13. La aplicación debe descargar las fotos de los trabajadores guardadas en el servidor.

RF14. La aplicación debe tener la opción de digitar la cédula para registrar un ingreso o salida, así validar que el trabajador o visitante existen.

RF15. La aplicación debe permitir escanear el código de barras de cédulas colombianas y al hacerlo mostrar la información: nombre, apellido y cédula.

RF16. La aplicación debe tener un botón que permita cerrar sesión y antes de hacerlo enviar la información registrada localmente.

RF17. Si me salgo de la aplicación o la cierro, al abrirla no debo ingresar otra vez usuario y contraseña.

RF18. La aplicación debe usar solo los colores de la empresa.

RF19. La aplicación debe trabajar sin conexión a internet, es decir, debe permitir el control de acceso ya sea por medio de reconocimiento facial, escáner de cédula o digitación.

3.2 Requisitos no funcionales

Existen ciertas condiciones con las cuales la aplicación debe garantizar su funcionamiento. Son requerimientos que están relacionados indirectamente con lo que el usuario espera, a continuación se presentan:

RFN1 Se debe garantizar una buena conexión a internet para hacer la descarga de la información de los empleados.

RFN2 La aplicación debe ser instalada en celulares de marca Samsumg de la serie A,

ya que este cumple con todos los requisitos básicos, esto se puede afirmar debido a que se realizaron pruebas sobre estos celulares.

RNF3 La aplicación debe tener actualizada la licencia del SDK, esta licencia se debe comprar una sola vez.

RNF4 El SDK para el reconocimiento facial solo permite asociar hasta 5000 usuarios por celular.

3.3 Reglas de negocio

La aplicación captura y almacena datos personales confidenciales como la imagen del rostro, nombre, cédula, datos de eps y arl. Por esta razón es necesario solicitar autorización a los empleados de la empresa. Para el cumplimiento de lo anterior a través del proceso de autenticación electrónica, en cumplimiento de lo definido en la Ley 1581 de 2012, en los capítulos 25 y 26 del Decreto Único 1074 de 2015.

3.4 Arquitectura

En el desarrollo de una aplicación es muy importante tener claro cuales son los componentes del sistema, el flujo y qué herramientas son las más adecuadas. Para esto se crean dos diagramas, el primero muestra una arquitectura general del sistema y en el segundo una más específica de la aplicación android.

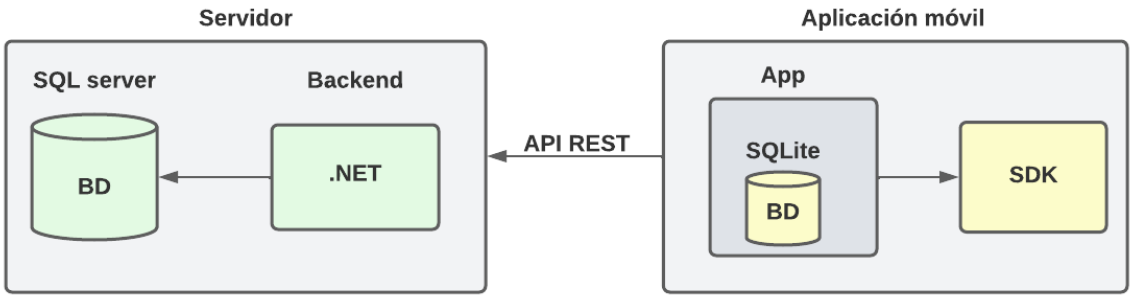

Figura 1. Arquitectura general del sistema

En la Figura 1 se presenta la arquitectura general del sistema, la cual contiene dos módulos principales, el primero representa el servidor donde se almacena la información suministrada por la aplicación. Este módulo se desarrolla con las herramientas SQL server, .Net y está alojado en una instancia de AWS. El servidor se encarga de proveer una API REST, la cual se consume desde la aplicación móvil. El segundo módulo representa la aplicación móvil, la cual incluye el SDK con la integración de la aplicación. Está desarrollado con el lenguaje Java y una base de datos local, SQLite, en este se incluye el desarrollo de las funcionalidades, la creación de base de datos, el consumo de la API y la integración con el SDK.

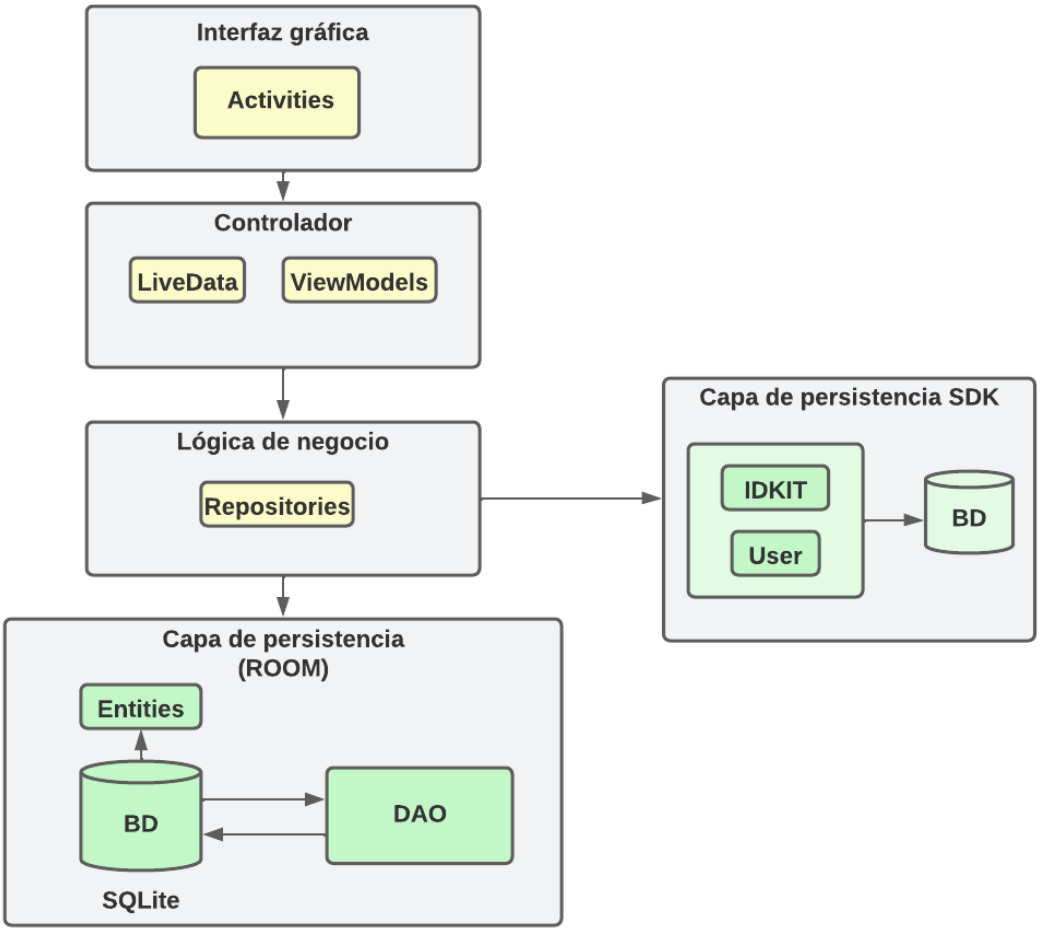

Figura 2. Arquitectura aplicación móvil

La Figura 2 presenta específicamente la arquitectura de la aplicación móvil MVVM (Modelo,Vista, Modelo de vista), separa la lógica de presentación de datos (vistas o interfaz de usuario) de la parte lógica de la aplicación (Android, 2022). Este patrón de diseño permite que en la vista solo haya inicialización de variables de interfaz de usuario, animación y diseño. En esta parte se hace referencia a las actividades o fragmentos, donde va la lógica y también el xml, que incluye todo el diseño visual de las actividades o fragmentos.

Por otro lado, hay una capa especial entre la Vista y el Modelo que se llama ViewModel, es el modelo de vista que proporciona un conjunto de interfaces, cada una de las cuales representa un componente de la interfaz de usuario en la vista (Android, 2022). El ViewModel posee una característica esencial para las aplicaciones, un ejemplo es cuando un celular es girado de manera horizontal y la información se pierde, porque los datos no persisten debido al ciclo de vida de las actividades. El ViewModel permite que se conserve la información, es decir que está sujeto a cambios de configuración. Adicional al ViewModel se usa una clase llamada LiveData, es un observador que constantemente actualiza la vista con los datos del modelo.

El repositorio, como se observa en la Figura 2, está conectado con dos módulos, ambos contienen base de datos, por esto sus funciones principales son entregar información y almacenar la lógica de la aplicación. Dado lo anterior el repositorio contiene la lógica de la integración con el SDK, el consumo de servicios y el modelo del negocio.

El módulo del SDK contiene una base de datos y unas clases (IDKIT y User) que permiten la conexión e integración con la lógica de la aplicación. Por último, el módulo que se ve al final contiene el modelo de datos, las consultas a la base de datos, las entidades y la base de datos local implementada en SQLite con el ORM Room, el cual se encarga del mapeo de datos y facilitar consultar a la base de datos.

En resumen, la arquitectura usada para la aplicación se basó en las recomendaciones de Google para el desarrollo de aplicaciones móviles, con el fin de desarrollar un prototipo modular y escalable, además de implementar las herramientas más actuales.

4.Implementación

Luego de elaborar un análisis y diseño, tener claro cuales son los objetivos o alcances del sistema y definir una arquitectura, se lleva a cabo la implementación. Esta sección presenta los detalles para la implementación del sistema en 4 partes. La primera tiene un diagrama de casos de uso necesarios para el desarrollo de la aplicación, la segunda aborda el modelo de la base de datos local de la aplicación, la tercera presenta una breve explicación del SDK y, por último, se presentan las vistas de la interfaz gráfica de usuario con su respectiva explicación.

4.1 Diagrama Casos de Uso

A continuación se presenta en la Figura 3 un diagrama de casos de uso con las principales funciones que pueden tener los usuarios para interactuar con la aplicación.

En la figura 3, se muestra un actor guarda que es el único usuario que interactúa con la aplicación, este puede realizar funciones como, ingresar a la aplicación por medio de login, registrar visitantes, sincronizar datos y registrar ingreso o salida, ya sea por medio de digitar cédula, escaneo de cédula o reconocimiento facial.

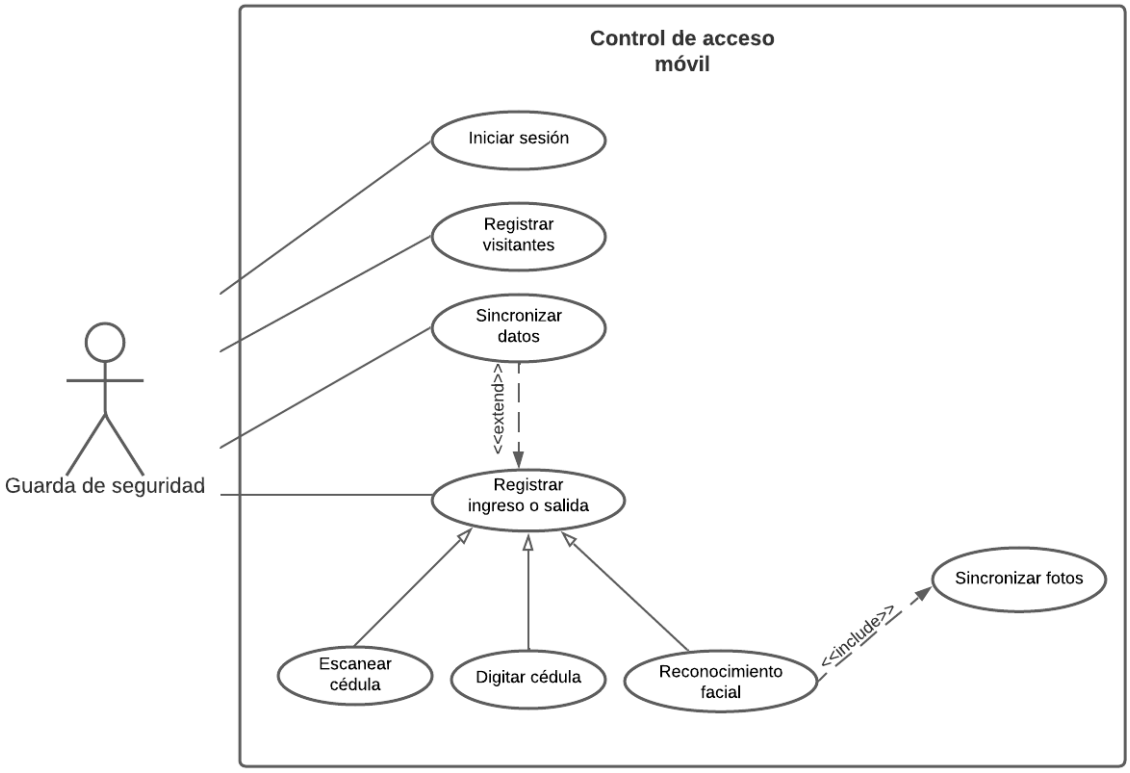

Figura 3. Diagrama de casos de uso

4.2 Modelo base de datos de la aplicación

En la Figura 4 se presenta un modelo simple de la base de datos de la aplicación, es decir, solo se mencionan las entidades más importantes. El modelo consta de 5 entidades: *person*, *image, accesType, subContractor* y *record*.

Person: Almacena los datos de los empleados, nombre, cédula, rh, fechas de vencimiento, entre otros. Esta es una de las entidades más importantes del modelo, ya que provee casi toda la información para guardar en el servidor.

Image: Guarda la información para el acceso a los datos de la base de datos de cada persona: ID único de la foto guardada que retorna el SDK y ID que identifica la persona, así solo debe haber un registro por cada trabajador.

AccesType: Modela los diferentes tipos de acceso, reconocimiento facial, digitación de cédula y escáner de cédula con su respectivo identificador.

SubContractor: Esta entidad contiene los campos nombre e ID del contratista que es poblada desde el servidor.

Record: Guarda los registros de control de acceso de todas las personas, tanto empleados como visitantes. Es una de las entidades más importantes ya que contiene la fecha y hora de entrada o salida, control de síntomas y si la persona tiene asociado un vehículo.

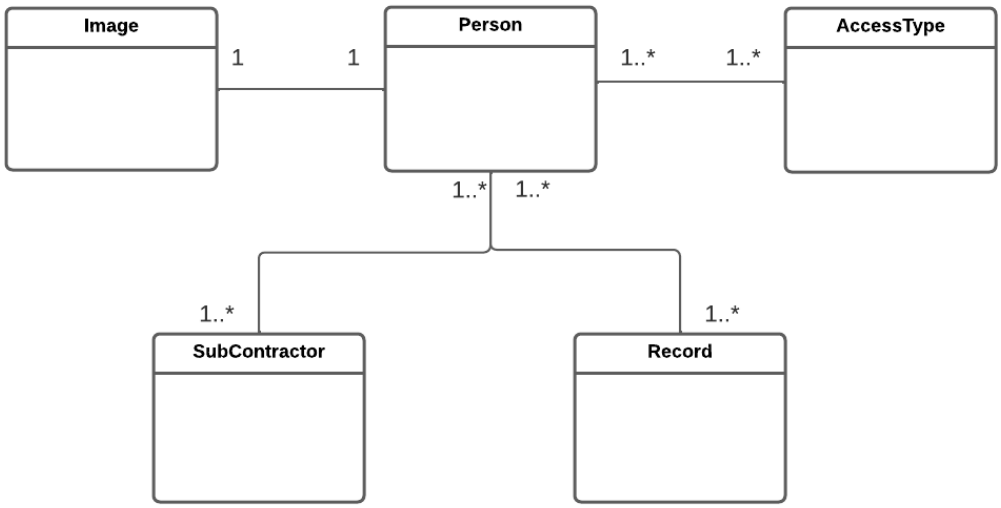

Figura 4. Modelo simplificado base de datos local

5.3 Uso del SDK IDKIT

Para agregar un reconocimiento facial a la aplicación se integra el SDK IDKIT, el cual permite suplir esta necesidad. El SDK contiene una base de datos donde guarda todas las imágenes con un único ID. Este ID se guarda en otra base de datos que es la propia de la aplicación junto con el identificador de cada persona. De esta manera, cada que se desee consultar la imagen de la persona, se podrá obtener. Como se mencionó, existe una segunda base de datos, donde se guarda la información de las personas, nombre, fechas de vencimiento e información

relevante para almacenar un registro. A continuación se mencionan y se explican algunos temas importantes:

Para empezar se aborda el tema del SDK. Contiene una base de datos propia, que crea inmediatamente se inicializa la instancia del IDKIT con la conexión a la base de datos. El código de la Figura 5 muestra cómo se desarrolla este proceso.

```
IDKit.initWithLicense(license);
idkit.connect(IDKIT_CONNECTION_STRING);
```

Figura 5. Pantallazo líneas de código inicialización IDKIT

La Figura 5 contiene la validación de la licencia y la conexión a la base de datos, la variable constante IDKIT_CONNECTION_STRING, es el nombre que se da a la base de datos. Luego de tener la conexión se comienzan a usar las funciones y los objetos que contiene el IDKIT:

User: es el objeto que se guarda en la base de datos, a su vez posee funciones para crear (*user.addFace()*), borrar, y obtener data de la foto del usuario (getCustomData()).

Idkit: esta clase contiene métodos que permiten gestionar al User, además contiene una de las funciones más importantes: *idkit.matchUsers(userEvidencia, userOrigin)* la cual permite hacer un comparativo entre cada uno de los usuarios almacenados en la base de datos y retorna un score. Y por último se tiene la función *idkit.registerUser(user[contador])* que retorna el identificador de la foto guardada en la base de datos.

5.4 Vistas de la aplicación

En esta sección está determinada por unas actividades que tienen la mayor parte de las vistas de la aplicación, a continuación se pueden ver en la Figura 6.

Figura 6. Listado de vistas de la aplicación

Cómo se observa en la Figura 6 hay nueve *activities.* Cada una de ellas contiene la relación entre el xml y la lógica que hace algún elemento luego de determinado evento. Existe la actividad NewPersonManualActivity, la cual le permite llenar al usuario la información para las personas que deseen acceder como visitantes, por medio de opción digitar cédula. A su vez está NewPersonScanActivity, la cual cumple la misma función pero se accede por medio de la opción del menú escanear cédula.

5. Resultados

En esta sección se presenta el prototipo final obtenido para la práctica, dividido en dos partes. En la primera se encuentran las pruebas para evaluar el desempeño del SDK y la segunda se muestran las vistas de interfaz de usuario.

5.1 Pruebas del SDK

Se realizaron pruebas de desempeño para el reconocimiento facial del SDK, con el fin de evidenciar qué tan efectiva es la aplicación reconociendo las personas. Para esto se plantea un experimento en el cual, a 5 personas se les tomó una foto para asociarla a la base de datos del SDK. Luego, se hicieron pruebas en 5 escenarios: Luz solar, bombillo led, accesorios (gafas, gorras y sombrero), foto a una distancia cerca y lejos. Para cada uno de los escenarios se tomaron 6 muestras (fotos) a cada persona.

A continuación, en las tablas, se presentan los resultados de las pruebas donde la primera columna contiene el nombre del escenario, luego el número de la persona y por último la iteración de la foto. Dentro del campo de iteraciones solo se ven dos resultados, un “si” y un “no”. El “sí”, significa que la aplicación reconoce a la persona correctamente; el “no”, que no la reconoce.

A continuación se muestra la Tabla 1, la cual reporta los datos para tomar la foto bajo el escenario de luz solar:

Escenario	Personas	1	2	3	4	5	6
Luz del día	1	si	si	si	si	si	si
	2	si	si	si	si	si	si
	3	si	si	si	si	si	si
	4	si	si	si	si	si	si
	5	si	si	si	si	si	si

Tabla 1. Escenario luz solar

La Tabla 2 representa las muestras para el escenario bombillo led.

Escenarios	Personas	1	2	3	4	5	6
Bombillo led	1	si	si	si	si	si	si
	2	si	si	si	si	si	si
	3	si	si	si	si	si	si
	4	si	si	si	si	si	si

Tabla 2. Escenario Bombillo led

La Tabla 3 presenta las muestras para el escenario de accesorios, las pruebas se realizaron con sombreros, gafas, gafas de sol y gorras

Escenarios	Personas	1	2	3	4	5	6
Accesorios/gafas/gorras	1	si	no	si	si	si	si
	2	no	no	si	si	si	si
	3	si	no	si	si	si	si
	4	no	no	si	no	si	si
	5	no	no	si	no	si	si

Tabla 3. Escenario Accesorios

La Tabla 4 presenta las muestras para el escenario donde se toman las fotos a una distancia de menos de un metro entre la persona que se le toma la foto y la que la toma.

Escenarios	Personas	1	2	3	4	5	6
Cerca	1	si	si	si	si	si	si
	2	si	si	si	si	si	si
	3	si	si	si	si	si	si
	4	si	si	si	si	si	si
	5	si	si	si	si	si	si

Tabla 4. Escenario distancia de menos de un metro

La Tabla 5 representa las muestras para el escenario donde se toman las fotos a una distancia de uno, dos y tres metros de la persona a la cual se le toma la foto. La mayoría que ilustra no, son tomadas a más de dos metros.

Escenarios	Personas	1	2	3	4	5	6
Lejos	1	no	no	no	no	si	si
	2	no	no	no	no	no	si
	3	no	no	no	no	no	no
	4	no	no	no	no	no	no
	5	no	no	no	no	si	no

Tabla 5. Escenario distancia más de un metro

Para evaluar el desempeño se aplicó la fórmula de exactitud (*accuracy*) en cada uno de los escenarios planteados inicialmente, donde se divide el número de verdaderos positivos sobre el total de los casos (Dagnechaw, 2020).

$$Exactitud = \frac{Número\ de\ predicciones\ correctas}{Número\ total\ de\ predicciones}$$

Luego de aplicar la fórmula se presentan los resultados en la Tabla 6, obtenidos para cada uno de los escenarios:

Escenarios	Exactitud
Luz solar	100%
Bombillo led	100%
Accesorios	66,70%
Cerca	100%
Lejos	13,30%

Tabla 6. Resultados finales prueba desempeño SDK

Después de obtener los resultados de la prueba de desempeño para el SDK se puede evidenciar que posee un 100% de exactitud para los escenarios de luz solar, bombilla led y la toma de fotos en una distancia de menos de un metro. Sin embargo, el SDK no tiene muy buena exactitud cuando la persona cuenta con

algún accesorio que cubra la cabeza o el rostro, los resultados son del 66,70%. Finalmente se pudo validar que el escenario donde menos exactitud posee el SDK es cuando la persona se encuentra lejos de la cámara, puesto que el resultado es de sólo 13,30% de exactitud.

5.2 Interfaz de usuario

Para mostrar evidencia de la funcionalidad de la aplicación, en esta sección, se presentan figuras con la respectiva explicación, con el fin de abarcar todas las funcionalidades principales.

En la Figura 7 se presenta la primera funcionalidad que cumple la aplicación, es decir el login. En la primera pantalla, se digita el usuario y contraseña. Al presionar el botón ingresar se consume el servicio que entrega toda información de los trabajadores y se continúa a la siguiente pantalla donde se presenta el menú, el cual contiene todas las opciones de la aplicación. En la segunda pantalla, en la parte superior derecha, se encuentra un botón, este permite desplegar otro pequeño menú con todas las opciones que necesitan acceso a internet, puesto que permiten descargar y cargar información en el servidor.

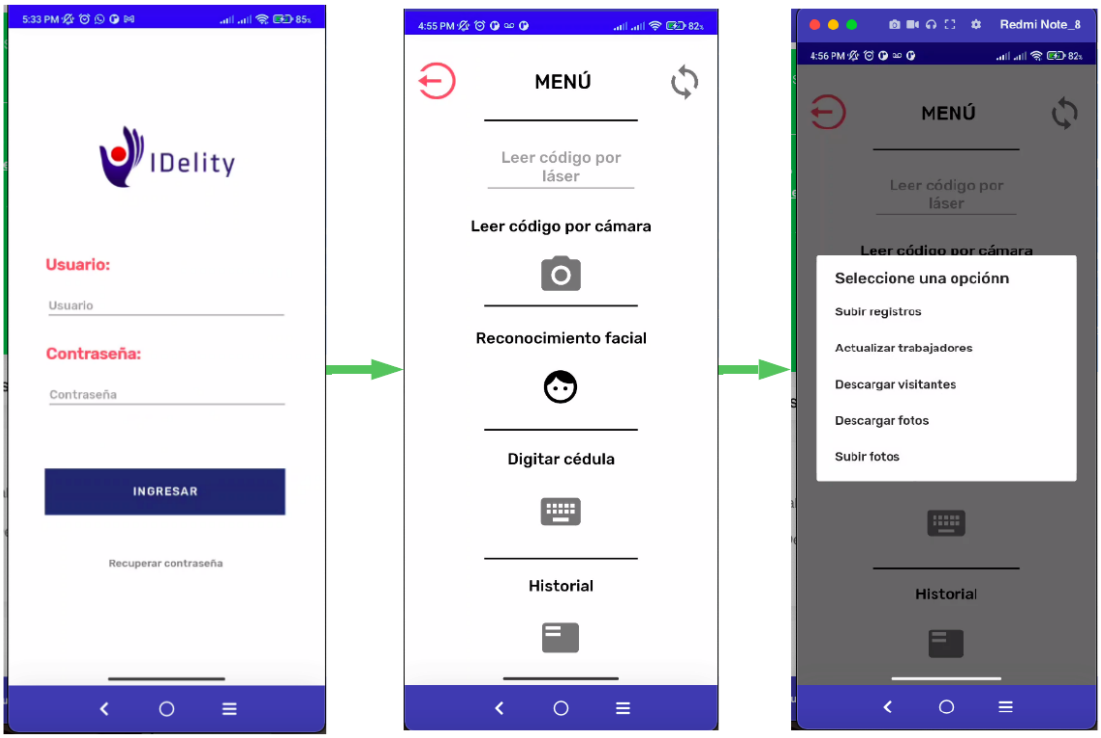

Figura 7. Funcionalidad login y despliegue de opciones

La Figura 8 presenta un registro de control de acceso para un trabajador, por medio de la opción escaneo de cédula. En la primera pantalla se puede observar que la cámara abre para escanear el código de barras, seguido muestra el nombre, cédula del trabajador y foto si hay en la base de datos, además se muestra el fondo de color rojo o verde. El rojo significa que la persona no cumple con alguno de los requisitos que la empresa requiere; el verde, que la persona puede acceder sin problema. A continuación la siguiente pantalla permite al usuario registrar si es una entrada o salida, si incluye vehículos o necesita tomar síntomas de covid.

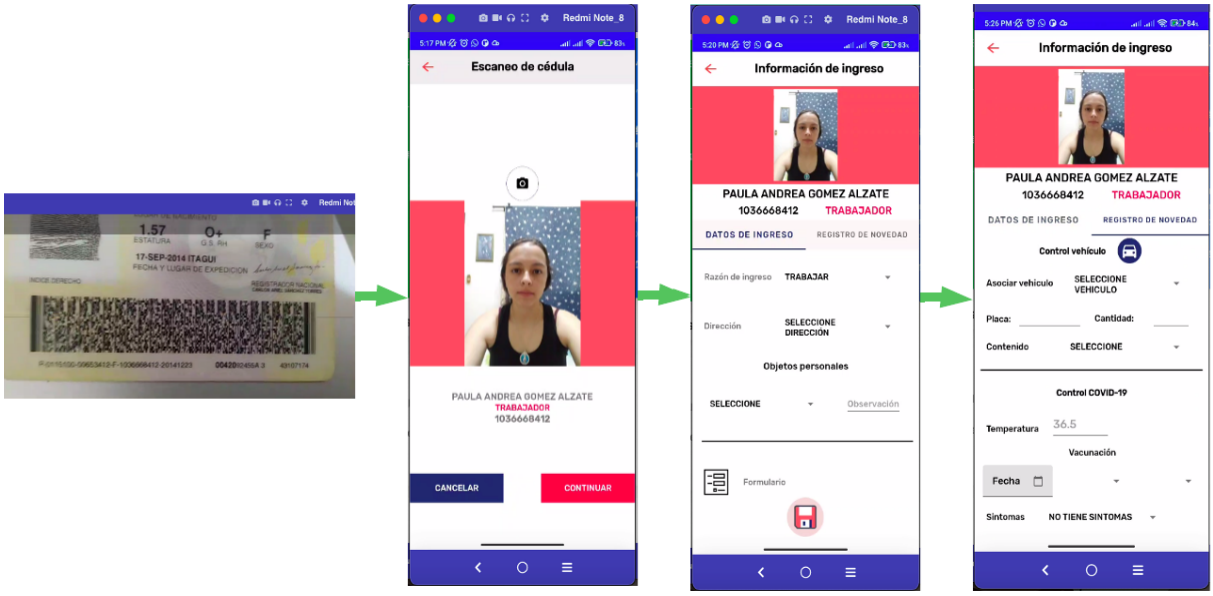

Figura 8. Control de acceso por medio de escaneo de cédula

La Figura 9 presenta la funcionalidad principal de la aplicación, el reconocimiento facial. En la primera pantalla se debe ingresar el número de cédula de la persona a la cual se le desea asociar una foto. Luego, se abre la cámara y se procede a tomar la foto. Una vez se obtenga la foto se puede observar en la segunda pantalla, donde para confirmar, se presiona en el botón guardar. Después de realizar los pasos anteriores se puede lograr un reconocimiento facial presionando el botón capturar y en la misma pantalla se evidenciará la foto que mayor coincidencia presentó con las demás de la base de datos. Por último, la tercera pantalla muestra un listado de registros (historial) de las personas que han ingresado o salido el presente día, con su respectiva información.

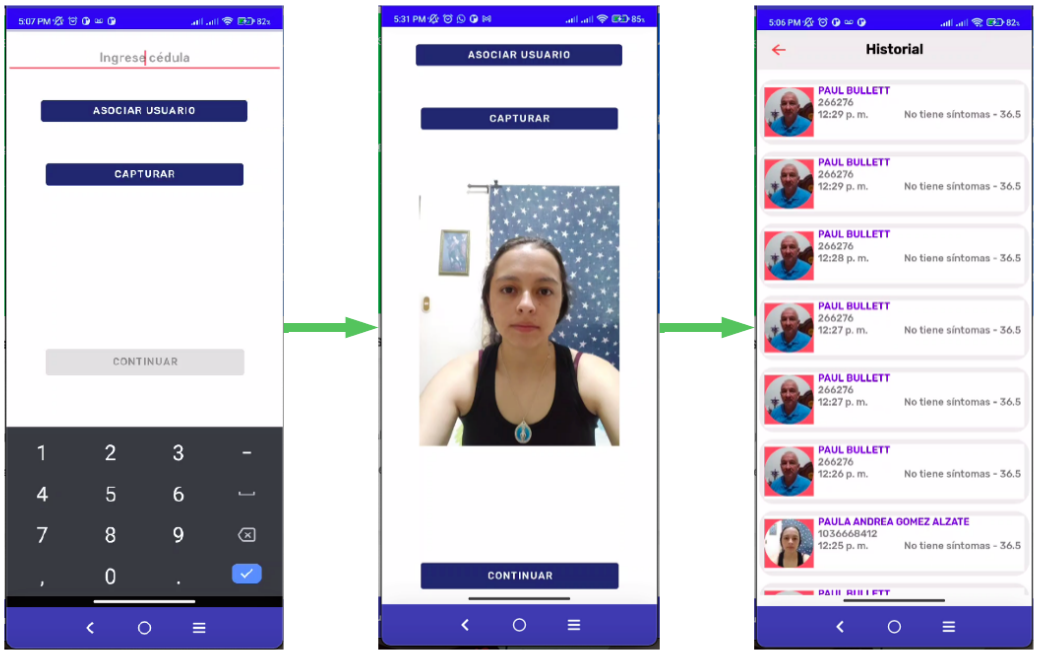

Figura 9. Control de acceso por reconocimiento facial e historial de registros

La Figura 10 contiene el flujo del control de acceso por medio de la digitalización de cédula. La primera vista presenta un cuadro de texto, donde se ingresa la cédula y se valida la existencia del empleado o visitante. Si la persona existe se observan las vistas dos y tres de la Figura 10. La validación de los requerimientos que la empresa solicita se realizan y por medio del fondo rojo o verde se da acceso a la empresa.

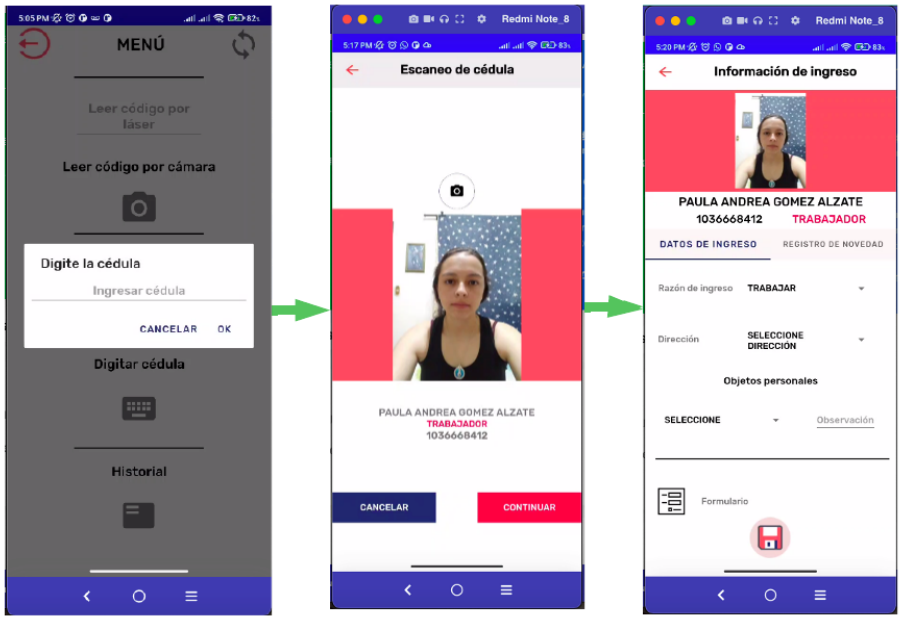

Figura 10. Control de acceso por medio de digitalización de cédula.

Para finalizar en la Figura 11 se presentan dos vistas, la primera se utiliza para ingresar datos de una persona que no está registrada en la base de datos y la segunda sigue realizando la misma funcionalidad de las vistas presentadas en las figuras anteriormente.

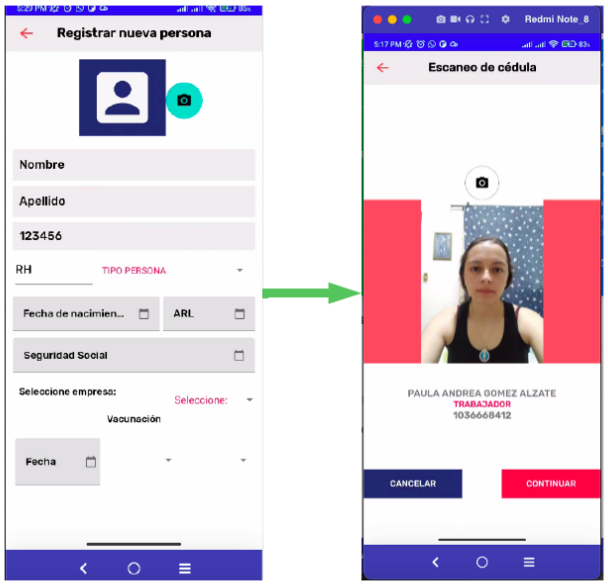

Figura 11. Interfaz gráfica para crear un visitante.

6. Conclusiones

En este informe se presenta el desarrollo de una aplicación móvil de reconocimiento facial para la gestión del control de acceso. La aplicación se desarrolló en el sistema Android, utilizando el lenguaje de programación Java, la integración del SDK de reconocimiento facial y algunas herramientas que se usan actualmente.

Para la realización de este informe fue de vital importancia dividir por secciones los temas trabajados, con el fin de lograr un mayor entendimiento a lo largo de la lectura, por esta razón la sección siguiente concluye en general todo el proyecto.

Para el desarrollo del proyecto de práctica, fue muy importante comprender la necesidad y los requerimientos del usuario, para lograr un prototipo funcional acorde al problema planteado. En general existen términos del cual no se conoce significado alguno, por esta razón se realiza una búsqueda exhaustiva de cada uno, con el fin de dar contexto y hacer más amena la lectura, todo esto se reúne en la sección de marco teórico. Por otra parte, para el desarrollo de la aplicación móvil, se implementó una mezcla de metodologías ágiles y clásicas, las cuales permitieron un buen manejo de los tiempos y comunicación.

Se destacan las etapas de análisis y diseño, los cuales abarcan los requerimientos funcionales y no funcionales, reglas de negocio y arquitectura, presentando así un panorama general de las herramientas necesarias para el desarrollo del prototipo.

Una vez se tuvieron claros los objetivos, necesidades, arquitectura y demás partes, se realizó la implementación, en la cual se integró el SDK con la aplicación, se crearon funcionalidades e interfaz gráfica adecuada a los requisitos del usuario, dándole un valor agregado de las demás aplicaciones que hay en el mercado. Por último, se realizaron pruebas de desempeño con el fin de validar con qué exactitud el SDK identificó a las personas y se lograron resultados del 100% en algunos escenarios. Sin embargo se logró identificar también que, el SDK no genera buenos resultados para un escenario donde la foto es tomada a una distancia de más de un metro, puesto que la exactitud fue de un 13,30%. Las pruebas permiten garantizar al usuario en qué escenario la aplicación alcanza un mayor nivel de potencia y así tener un ambiente controlado de esta.

También se realizaron pruebas de funcionalidades, registro de personas, conexión a servidor y se logró cumplir con el alcance planteado inicialmente.

Referencias

Institute, L. (2021). *Reconocimiento facial: Descubre cómo funciona y quién (y para qué) lo utiliza.*https://www.lisainstitute.com/blogs/blog/reconocimiento-facial-como-funciona-quien-utiliza

Bito. (2022) *Access control in the company.* https://www.bito.com/en-gb/expert-knowledge/article/access-control-in-the-company/

Cognex.(2022). *Código de barras PDF 417.* https://www.cognex.com/es-co/resources/symbologies/stacked-linear-barcodes/pdf417-bar-codes

Dagnechaw, S. (2020). *What is Accuracy, Precision, and Recall? And Why are they Important?* https://shiffdag.medium.com/what-is-accuracy-precision-and-recall-and-why-are-they-important-ebfcb5a10df2

Android. (2022) *ViewModel Overview.* https://developer.android.com/

IdeasControl. (2019) *Biometría.*https://www.ideascontrol.com/soluciones-biometria/

Icertis. (2019) *Componentes de arquitectura de Android, de MVC a MVVM.*

https://ahorasomos.izertis.com/solidgear/componentes-de-arquitectura-de-android-de-mvc-a-mvvm/

Falcon Cloud. (2022) *Solución Biométrica #1 para el Control de Horarios y Asistencia.*

https://www.proware.com.co/

Ungoti. (2022) *Ciclo de vida del desarrollo de software.*
https://ungoti.com/es/soluciones/desarrollo-de-software/sdlc/