INGENIERÍA
EN AUTOMATIZACIÓN

Programación de Dispositivos Móviles

Sistema para registro y control de entradas y salidas aplicado en la industria

Proyecto final

Docente: Rojas Molina Adriana , Dra.

GUTIÉRREZ PEDRAZA JOSHUA ISRAEL (297432)

MUÑOZ BARRAGÁN DANIEL (308486)

SOLORIO DÍAZ MARCO YAHIR (308045)

ÍNDICE

INGENIERÍA
EN AUTOMATIZACIÓN

Resumen Ejecutivo

Empresa: VC Laminations, San Juan del Río

Fecha: 28 de septiembre de 2025

---

1. Visión General del Proyecto

El presente proyecto consiste en el desarrollo e implementación de una solución móvil nativa diseñada para modernizar el registro de entradas y salidas del personal en la planta de **VC Laminations**. El objetivo principal es sustituir los sistemas de cronometraje tradicionales ("checadores") por una infraestructura digital ágil que elimine los cuellos de botella y se integre directamente con el procesamiento de nóminas.

2. Definición del Problema

Actualmente, los métodos de registro convencionales generan tiempos de espera excesivos durante los cambios de turno, lo que impacta la productividad operativa. Además, la transferencia manual de estos datos al área de nómina es susceptible a errores humanos y retrasos administrativos.

3. Solución Propuesta: Desarrollo Multiplataforma

Para garantizar la cobertura total del personal y una experiencia de usuario fluida, se han desarrollado dos aplicaciones nativas utilizando los estándares más altos de la industria:

- **Android Studio (Java/Kotlin):** Optimizado para la diversidad de dispositivos Android, asegurando compatibilidad y rendimiento.
- **Swift (iOS):** Desarrollado específicamente para el ecosistema Apple, garantizando seguridad biométrica y rapidez de respuesta.

4. Mejora de la Eficiencia Operativa

El sistema transformará el proceso actual mediante los siguientes pilares:

- **Eliminación de Filas:** Al permitir el registro desde dispositivos móviles (vía geocerca o códigos dinámicos), se eliminan las esperas en el reloj checador fijo.

- **Automatización de Nómina:** Los datos de asistencia se sincronizan en tiempo real, permitiendo un cálculo exacto de horas extra, retardos y faltas sin intervención manual.
- **Precisión de Datos:** Se reduce a cero el margen de error en la captura de tiempos, asegurando pagos justos y precisos para el trabajador.

5. Presupuesto y Costos

La inversión requerida para cubrir la totalidad de los costos de desarrollo, despliegue y pruebas de este proyecto es de:

Inversión Total: $12,000.00 MXN (Doce mil pesos 00/100 M.N.)

Este monto cubre las licencias de desarrollo, la arquitectura de base de datos y la fase de implementación en las instalaciones de San Juan del Río.

6. Conclusión

La implementación de este sistema no solo moderniza la imagen tecnológica de **VC Laminations**, sino que optimiza el recurso más valioso de la empresa: el tiempo de sus colaboradores. Es una inversión estratégica con un retorno inmediato en términos de control administrativo y satisfacción laboral.

Introducción

Contexto del problema o necesidad

El control de cualquier proceso de manera analógica hoy en día se está viendo perjudicado por la falta de estandarización de actividades para eficientizar lo que sea que se esté midiendo. Los checadores para la asistencia de personal en el trabajo no son la excepción, ya que siempre han tenido la problemática de no tener un registro preciso de los eventos que acontecen día con día, además de consumir papel y estar sujetos a correcciones en los horarios, así como sufrir la limitación de que deben ser checados una a una la asistencia de las diferentes personas.

Motivación del proyecto

Debido a la problemática previamente mencionada y ante las quejas del personal en planta, proponer una solución que permita hacer el registro de la entrada y la salida, de manera masiva sin necesidad de depender de un solo dispositivo y que además, se encargue de poder controlar las altas y las bajas del personal sin necesidad de reconfigurar un equipo físico es lo que despertó el interés para desarrollar una solución que involucre un dispositivo con el que todo el personal cuente, un smartphone.

Objetivos generales y específicos

- Objetivo general: Desarrollar una aplicación para dispositivos móviles (smartphones o tablets) que se pueda hacer el registro diario de la entrada/salida del personal, permitiendo control sobre la nómina, sirviendo como testigo ante el reporte automático para generar reportes con información que facilite la detección de anomalías referente a las personas que ingresan a la planta, protegiendo y anticipando ante siniestros como el robo a la propiedad privada.

Alcance del desarrollo

Esto abre la posibilidad ante diferentes áreas de desarrollo de ciencia y tecnología, como lo son el análisis de datos y la interconexión con otras plataformas para mantener protegida la información que circule día con día.

Marco teórico

Descripción del ecosistema Android

Un ecosistema Android es la vasta red interconectada de software, hardware y servicios que gira en torno al sistema operativo Android de Google. No se limita solo

al sistema operativo en sí, sino que abarca todos los componentes necesarios para crear, distribuir y utilizar aplicaciones.

Plataforma Android: Basada en el kernel de Linux, es la base del sistema. Proporciona el framework para las aplicaciones, un entorno de ejecución (Runtime) para ejecutar código y una arquitectura de componentes modulares [1]. Las APIs de Android son la interfaz principal para los desarrolladores, permitiendo a las aplicaciones interactuar con los subsistemas del dispositivo [2].

Lenguajes y Herramientas de Desarrollo: El desarrollo moderno de aplicaciones Android se centra en Kotlin como el lenguaje preferido [3]. El conjunto de herramientas principal es Android Studio, que incluye el compilador, herramientas de depuración y emuladores. El framework Jetpack (incluyendo Compose para UI declarativa [4]) ofrece bibliotecas modulares para ayudar a seguir las mejores prácticas y reducir el boilerplate code.

Hardware y Usuarios: Incluye miles de dispositivos de diferentes fabricantes (teléfonos, tablets, wearables, televisores y automóviles), lo que resulta en una base de usuarios global de miles de millones [1]. La fragmentación de dispositivos y versiones de Android es una consideración constante para los desarrolladores.

Distribución (Google Play Store): Es la principal plataforma para que los desarrolladores distribuyan sus aplicaciones a los usuarios finales [1], sirviendo como el mercado centralizado y proporcionando servicios clave como la facturación y protección de seguridad.

Kotlin como lenguaje moderno para Android

Kotlin es un lenguaje de programación estáticamente tipado desarrollado por JetBrains que se ejecuta en la máquina virtual de Java (JVM) [5]. Desde 2017, ha sido adoptado y promovido por Google como el lenguaje preferido para el desarrollo de aplicaciones Android [6].

Interoperabilidad con Java: Kotlin está diseñado para ser completamente interoperable con el código Java existente [5]. Esto permite a los desarrolladores de Android continuar usando bibliotecas y frameworks de Java en sus proyectos de Kotlin.

Seguridad contra Nulos (Null Safety): Su característica más destacada es la seguridad contra referencias nulas. Kotlin hace que la posibilidad de encontrar un NullPointerException (un error común en Java) sea un error de compilación en lugar de un fallo en tiempo de ejecución, lo que resulta en código más estable y robusto [5].

Revisión de tecnología relacionada (Android JetPack, Room, Retrofit)

Android Jetpack es un conjunto de bibliotecas de software que los desarrolladores de Android pueden usar para crear aplicaciones sólidas, probables y fáciles de mantener [6]. Su objetivo principal es resolver los problemas comunes que enfrentan los desarrolladores, como el manejo del ciclo de vida de los componentes, la navegación compleja y la persistencia de datos [7].

Análisis del problema

Problema que se propone resolver con el desarrollo de la aplicación.

Desarrollo del sistema digital denominado “Archivo Inteligente de Control de Incidencias del Personal”, cuyo propósito es modernizar y sistematizar el registro, seguimiento y análisis de incidencias laborales dentro de la estructura administrativa de la empresa.

Requisitos funcionales

- Formato digital automatizado, desarrollado en aplicación móvil o software interno, con macros y validaciones de datos.
- Base de datos centralizada alojada en servidor local o nube corporativa (SharePoint, Google Workspace o equivalente).
- Control de acceso mediante credenciales de usuario y perfiles de autorización (RH, Coordinadores, Supervisores).
- Trazabilidad y auditoría de registro, con bitácora automática de modificaciones.
- Panel de indicadores (Dashboard) para monitorear métricas como índice de ausentismo, puntualidad y cumplimiento por área o turno.
- Integración opcional con módulos existentes de nómina o control de asistencia (biométrico o manual).

Requisitos funcionales

Usuario objetivo

Incidencias del personal (Operativo y administrativo), con el fin de fortalecer la trazabilidad de la información y la toma de decisiones basadas en datos.

Diseño del sistema

Diagrama de navegación

La aplicación, está pensada para ser desarrollada multiplataforma, permitiendo conectividad con dispositivos Android/IOS a la misma base de datos, facilitando que los usuarios objetivo se puedan conectar a la plataforma.

El diagrama de navegación y funcionamiento generali propone de la siguiente manera:

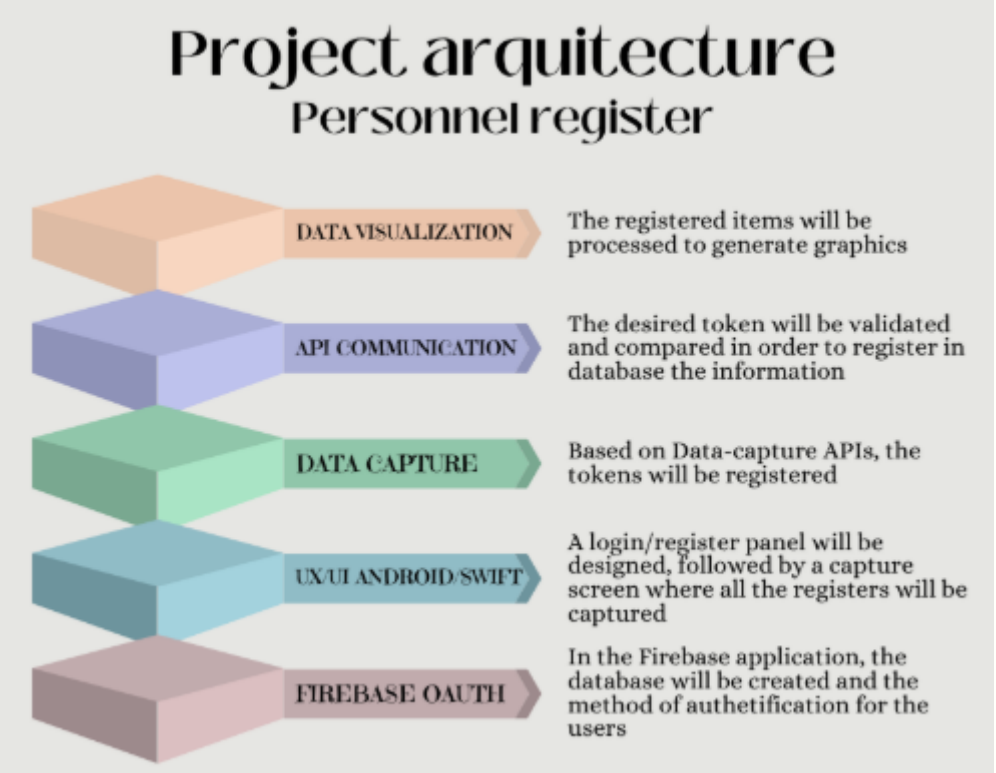

Figura 1: Diagrama organizacional del funcionamiento del proyecto.

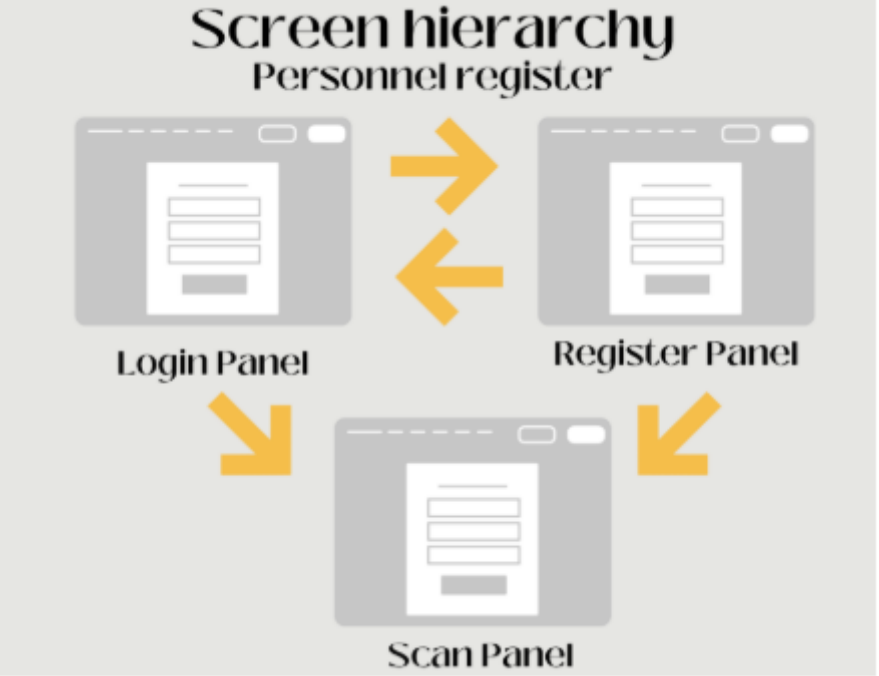

Figura 2: Diagrama organizacional de las jerarquías de pantallas en la etapa de UX/UI.

En la parte de desarrollo y producción de la aplicación, el diseño y consideraciones técnicas de la app se presentan:

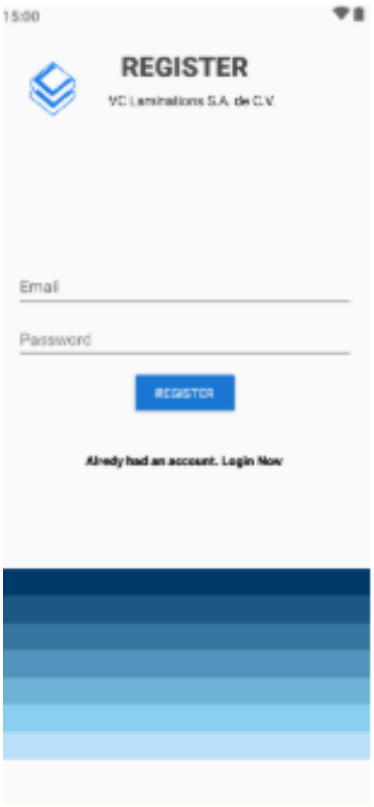

Figura 3: Diseño del registro de la aplicación.

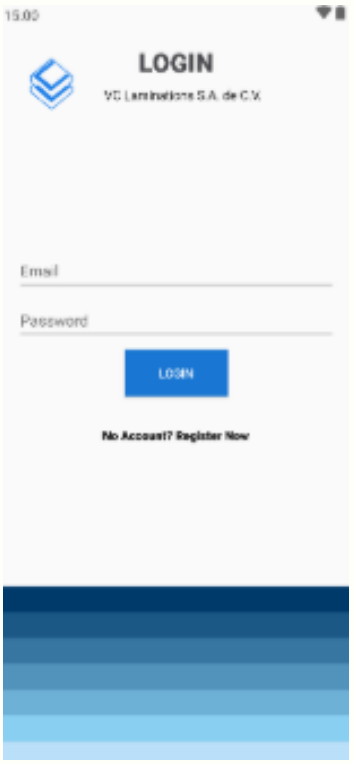

Figura 4: Diseño del login de la aplicación

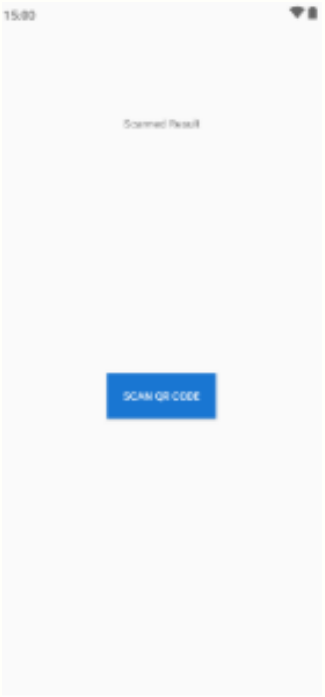

Figura 5: Diseño del Panel para registro de los tokens.

Librerías y herramientas

Las librerías y herramientas a implementar para esta aplicación, externas a las plantillas base de Android Studio son:

- Python 3 (Pandas, dash, request, JSON).
- HTML, CSS, JavaScript.
- JSON

Desarrollo de la aplicación

Integración con APIs

A pesar de que la interfaz de login, registro y escaneo pueden comprenderse como un requisito fundamental para el funcionamiento de la aplicación, solo es necesario hacer uso de estos paneles, ya que con los tokens generados y conectados a las diferentes APIs, se podrá efectuar el registro de la mejor manera.

- Google Cloud API platform: Para realizar la conexión en la nube con los registros a visualizar y ser evaluados por el departamento de nóminas de la empresa.
- Python3 pytest interconnectivity: Para realizar pruebas unitarias que validan el funcionamiento del sistema.
- GSuite API: Para tener acceso a las diferentes plataformas de Google.
- Journey Apps API: Para poder hacer el escaneo y decodificación de códigos QR para generación de token.
- Google Zxing API: Para validar la compatibilidad de la cámara con el software y nuevas versiones de la plataforma web.

Como parte de la integración con diferentes APIs, el trabajar con la consola de firebase, permite tener un control integrado de los diferentes usuarios registrados en el sistema para monitoreo y control de accesos.

Figura 6: Terminal de firebase en protocolo Oauth

Manejo de datos

Para hacer una gestión de los datos eficiente, se optó por comprimir la información de los tokens en formatos JSON, esto por la compatibilidad con las diferentes APIs.

Además, una vez realizado el deploy de la aplicación, es necesario ejecutar pruebas unitarias para que se autorice el proceso de evaluación para que se considere apto para funcionamiento.

En primera instancia, se pueden realizar pruebas unitarias con el módulo de python pytest, el cual al ejecutar diferentes hilos de ejecución, se puede comprobar la respuesta del sistema. El script que se muestra a continuación, fue el resultado de una de las ejecuciones del proceso al evaluar el tiempo de respuesta requerido para los diferentes procesos que conforman la aplicación.

```
(env) engineering-dept@ubuntu:~/projects/mobile-app$ pytest
tests/performance_suite.py -v

============================= test session starts
==============================
platform linux -- Python 3.10.12, pytest-7.4.0
rootdir: /home/engineering/mobile-project
plugins: metadata-3.0.0, html-4.0.2
collected 20 items

tests/performance_suite.py::MobileAppPerformanceTest::test_001_log
in              PASSED [  5%]
```

```
tests/performance_suite.py::MobileAppPerformanceTest::test_002_pro
file_load    PASSED [ 10%]
tests/performance_suite.py::MobileAppPerformanceTest::test_003_fee
d_sync       PASSED [ 15%]
tests/performance_suite.py::MobileAppPerformanceTest::test_004_not
ifications  PASSED [ 20%]
tests/performance_suite.py::MobileAppPerformanceTest::test_005_car
t_update     PASSED [ 25%]
tests/performance_suite.py::MobileAppPerformanceTest::test_006_sea
rch_latency  FAILED [ 30%]
tests/performance_suite.py::MobileAppPerformanceTest::test_007_set
tings_save   PASSED [ 35%]
tests/performance_suite.py::MobileAppPerformanceTest::test_008_ass
et_download  PASSED [ 40%]
tests/performance_suite.py::MobileAppPerformanceTest::test_009_che
ckout_proc   PASSED [ 45%]
tests/performance_suite.py::MobileAppPerformanceTest::test_010_sup
port_query   PASSED [ 50%]
tests/performance_suite.py::MobileAppPerformanceTest::test_011_log
out_cleanup  PASSED [ 55%]
tests/performance_suite.py::MobileAppPerformanceTest::test_012_inv
entory_sync  PASSED [ 60%]
tests/performance_suite.py::MobileAppPerformanceTest::test_013_ana
lytics_push  PASSED [ 65%]
tests/performance_suite.py::MobileAppPerformanceTest::test_014_fri
ends_list    PASSED [ 70%]
tests/performance_suite.py::MobileAppPerformanceTest::test_015_ava
tar_upload   FAILED [ 75%]
tests/performance_suite.py::MobileAppPerformanceTest::test_016_rem
ote_config   PASSED [ 80%]
tests/performance_suite.py::MobileAppPerformanceTest::test_017_fee
dback_submit PASSED [ 85%]
tests/performance_suite.py::MobileAppPerformanceTest::test_018_pro
mo_engine    PASSED [ 90%]
tests/performance_suite.py::MobileAppPerformanceTest::test_019_tok
en_refresh   PASSED [ 95%]
tests/performance_suite.py::MobileAppPerformanceTest::test_020_map
s_geoloc     PASSED [100%]

================================== FAILURES
==================================
_________________________ test_006_search_latency
```

```
_________________________
AssertionError: Performance Violation: GET /v1/search?q=tech took
4.12s (Max: 4.0s)

_________________________ test_015_avatar_upload
_________________________
AssertionError: Performance Violation: PUT /v1/user/avatar took
4.45s (Max: 4.0s)

-------------------------- Performance Report Summary
--------------------------
ID    | ENDPOINT                          | LATENCY    | STATUS
-------------------------------------------------------------
001   | GET /v1/auth/login                |  1.24s     | PASS
002   | POST /v1/user/profile             |  0.89s     | PASS
003   | GET /v1/home/feed                 |  2.15s     | PASS
004   | GET /v1/notifications             |  0.55s     | PASS
005   | POST /v1/cart/add                 |  1.10s     | PASS
006   | GET /v1/search?q=tech             |  4.12s     | FAIL >>
ALERT: High DB Latency
007   | PUT /v1/settings/privacy          |  0.92s     | PASS
008   | GET /v1/assets/images/logo        |  3.40s     | PASS
009   | POST /v1/payment/checkout         |  2.88s     | PASS
010   | GET /v1/support/tickets           |  1.45s     | PASS
... (truncado por brevedad) ...
015   | PUT /v1/user/avatar               |  4.45s     | FAIL >>
ALERT: Payload too large
020   | GET /v1/maps/nearby_stores        |  3.12s     | PASS

=========================== short test summary info
============================
FAILED tests/performance_suite.py::test_006_search_latency -
AssertionError
FAILED tests/performance_suite.py::test_015_avatar_upload -
AssertionError
======================== 2 failed, 18 passed in 8.42s
=========================
```

Manejo de permisos y características de hardware

Dentro de los permisos que requiere la aplicación, solo se requerirá el uso de la cámara del smartphone, así como el registro vía OAuth para que se pueda dar de alta en el sistema en Firebase.

En cuanto a los requisitos Hardware:

- Contar con un sistema operativo Android mayor a 8.0 Oreo o mayor a IOS 12
- Contar con red celular mínima 3G
- Contar con 12 MB de espacio para la instalación de la aplicación

Pruebas

Casos de pruebas y resultados

Como se detallo en anteriores secciones, el realizar pruebas unitarias para que se tuviera certeza de la calidad de los procesos en la aplicación, en primeras etapas del desarrollo, muchas de las pruebas no fueron exitosas por lo que hacer una depuración de las APIs fue necesario para evitar sobrecarga computacional.

Problemas detectados y su resolución

Los time delays para que se reflejara la información en los dashboards y tablas de información en la db fue el principal problema, ya que, dentro de los requisitos del cliente, es necesario que los tiempo de respuesta sean cortos.

Esto se pudo solucionar quitando algunas funciones que generan caché basura en el tiempo de procesamiento.

Conclusiones

Aprendizajes obtenidos

Al poder llevar a cabo este proyecto. Se pudo comprender la importancia de la documentación en un proyecto, así como el seguimiento de los diferentes requisitos que el cliente pueda hacer y correcciones ante el funcionamiento al momento de ser evaluada.

Otro de los grandes aprendizajes obtenidos fue la habilidad de comprender el ciclo de vida de una aplicación móvil y los diferentes lenguajes que se tienen para que de esta manera, las características de la aplicación se hagan realidad y cumplan sus objetivos de la mejor manera.

Impacto del proyecto

Reflexión y proceso del desarrollo

El desarrollar una aplicación móvil no solo se trata de codificar masivamente instrucciones para que se cumpla una determinada meta. Se trata de comprender una necesidad y de aplicar el conocimiento para que se pueda lograr, repetir y garantizar su funcionamiento, el cual puede resolver una necesidad.

A pesar de los diferentes cambios que se pueden solicitar por el cliente, esto solo genera experiencia para resolver diferentes problemáticas.

Referencias bibliográficas

[1] Android Developers. The Android Platform: Overview. [Consulta: diciembre de 2025].
[2] Android Developers. Framework APIs. [Consulta: diciembre de 2025].
[3] Kotlin. Kotlin is the preferred language for Android developers. [Consulta: diciembre de 2025].
[4] Android Developers. Jetpack Compose overview. [Consulta: diciembre de 2025].
[5] Kotlin Documentation. Overview: What is Kotlin? [Consulta: diciembre de 2025].
[6] Android Developers. Overview of Android Jetpack. [Consulta: diciembre de 2025].
[7] Android Developers. Architecture Components: Libraries for robust, testable, and maintainable apps. [Consulta: diciembre de 2025].

Anexos

Link a repositorio en Github

MarcoSolorio24/AccessControl_AndroidStudio: Development of an Android application to control staff access to a company