BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS
DE LA COMPUTACIÓN

Diseño y desarrollo de una apliación móvil para un sistema completo para la gestión de laboratorios.

TESIS PARA OBTENER EL
GRADO EN LICENCIATURA EN
INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA
Eric Senén Mora López

ASESOR:
Dr. Mariano Larios Gómez

Puebla, Puebla
Noviembre 2024

Dedicatoria

Quiero expresar mi más profundo agradecimiento a las personas que han sido fundamentales en mi vida y en este proceso:

A mis padres, Nora y Camilo, quienes con su esfuerzo y dedicación me han apoyado incondicionalmente para alcanzar mis metas y sueños.

A mis segundos padres, Rebeca y Víctor, que me criaron y cuidaron, guiándome para ser un gran profesional.

A mis hermanos, Nain y Rebeca, con quienes crecí y aprendí a guiar, cuidar y ser una mejor persona.

A mi tía Aracely y a mis primas Macarena y Claudia, quienes siempre me han apoyado y acompañado. Su cariño y aliento han sido fundamentales en este viaje, y agradezco profundamente su presencia en mi vida.

A mis amigos: a Rubén, quien me brindó la oportunidad y creyó en mí al iniciar mi camino profesional; a Alfonso y Julio, que siempre han estado a mi lado, compartiendo sus experiencias y brindando su apoyo en momentos de necesidad; a Moisés, quien se ha convertido en uno de mis amigos más leales y me ayuda a razonar mis palabras y acciones; y a Ariana, que a pesar de la distancia, siempre ha sido una amiga fiel y me ha dado el empujón que necesitaba.

A Berenice, el amor de mi vida, quien ha sido una compañera incondicional y un apoyo constante. Su amor y su luz han transformado mi camino, haciendo que cada paso sea más especial y significativo.

Finalmente, agradezco a todas las personas que han dejado su huella en mi vida al cruzarse en mi camino.

Eric Senén Mora López

Agradecimientos

Quiero agradecer a la Benemérita Universidad Autónoma de Puebla y a la
Facultad de Ciencias de la Computación, de igual manera, quisiera reiterar que esta tesis no hubiera sido posible sin la guia y apoyo de mis profesores y compañeros que estuvieron junto a mi en esta etapa. A aquellos que siempre me brindaron tanto sus conocimientos como una palabra de aliento. En especial, quiero agradecer al Dr. Mariano Larios Gómez, cuya guía y apoyo incondicional a lo largo de mi carrera han sido fundamentales para alcanzar este logro. No solo lo considero un excelente mentor, sino también un verdadero amigo. Asimismo, quiero extender mi agradecimiento al M.C. Carlos Zamora Lima, por su dedicación y compromiso en mi formación. Su enseñanza y apoyo han dejado una huella importante en mi trayectoria académica.

Tabla de contenido

Resumen

En esta tesis se presenta la planeacion, desarrollo e implementacion de una aplicación móvil diseñada para optimizar la dministracion de laboratorios universitarios, enfocandose en tres áreas clave: el control de prestamo de herramientas y equipos de los laboratorios, la gestion de entradas y salidas de los usuarios y la administracion del inventario. Esta aplicación esta dirigida a facilitar las labores de los responsables del laboratorio, estudiantes y docentes, enfocado a mejorar los procesos mediante el uso de nuevas tecnologias que atomtaiza tareas manuales, permite el acceso remoto a la informacion y reduce la posibilidad de errores.

El sistema permite a los usuarios el solicitar y registrar préstamos de equipos, asi como a los encargados gestionar su devolucion. Además, integra un módulo centrado en el control de accesos que registra la entrada y salida de miembros de laboratorio mediante identificaciones unicas, esto nos garantiza la seguridad del espacio y facilita el seguimiento de las personas que hacen uso del laboratorio. La gestion del inventario es otra funcionalidad central de la aplicación, nos permite agregar, editar o eliminar las herramientas y equipos de laboratorio.

La apliacion se desarrolló utlizando la tecnologia llamada React Native para la interfaz móvil , se implemento Firebase para la gestion de una base de datos en tiempo real y TypeScript para garantizar el tipado estatico. La apliacion móvil se implemento en las plataformas iOS y Android, proporcionando una solucion multiplataforma que facilita el acceso a todos los usuarios de laboratorio.

Introduccion.

En el ámbito universitario, la administracion eficaz de recursos, en especial los laboratorios de las facultades, es crucial para la gestión, uso y distribución de las instalaciones e inventario. Sin embargo, la administración manual de los accesos a estos, y el préstamo del inventario resulta complicado y es propenso a errores humanos, esto afecta negativamente la experiencia y el desarrollo academico de los usuarios. La necesidad de desarrollar una solucion utilizando la tecnología que tenemos al alcance es evidente, estos problemas pueden ser vistos en los laboratorios de hardware, donde al hacer prestamos de inventarios, la fila para la adquisicion de material, y tambien el no saber la disponibilidad de estos pueden generar que existan perdidas de tiempo a demas de afectar negativamente el desarrollo de los alumnos, maestros o incluso los responsables. Otra comparativa de esta problemática es en los laboratorios de química, donde los insumos necesarios los cuales son limitados, hablando de substancias o materiales, pueden estar agotados y los encargados de estos mismos no tienen un control o no pueden llevar consigo este mismo, generando que haya pérdidas en tiempo y esfuerzo para los alumnos que los requieran. Igualmente cuando en los laboratorios o simuladores de estomatología, estos ya están apartados y el cupo es limitado, los alumnos necesitan una opción para poder hacer un registro más sencillo y tener conocimiento para cuando estos estén disponibles. Esto es una problemática general cuando los inventarios, cupos, insumos y lugares no pueden ser consultados correctamente ni tener un control. En esta época, y con las nuevas tecnologías es más sencillo y necesario hacer una migración correcta y fácil a las tecnologías móviles y en la nube para tener una, casi, perfecta gestión y control de los laboratorios.

La finalidad de este proyecto es diseñar y desarrollar una aplicación móvil para gestionar el acceso de alumnos a laboratorios universitarios y controlar préstamos de inventario. Se emplearán tecnologías modernas como React Native, diseño Atómico,

Jest, TypeScript, lenguajes nativos como Swift y Kotlin para lograr una solución robusta y escalable que optimice la gestión de recursos y Firebase para implementar una base de datos en tiempo real.

La implementación de una aplicación móvil dedicada a la gestión de acceso y préstamos en laboratorios universitarios no solo mejorará la experiencia de los usuarios, sino que también optimizará la administración de recursos y promoverá un ambiente académico más eficiente y colaborativo. Esto puede contribuir significativamente a la calidad educativa y operativa de los laboratorios universitarios.

Además de proporcionar una solución práctica a un problema común en entornos universitarios, este estudio integrará tecnologías modernas y enfoques de ingeniería de software. La implementación del diseño Atómico permitirá una arquitectura modular y flexible, mientras que TypeScript mejorará la calidad y mantenibilidad del código. El uso de Swift y Kotlin, en Xcode y Android Studio, ayudarán a las implementaciones nativas del hardware de los dispositivos, fomentando el estudio e implementación propia de la tecnología. Estas contribuciones tendrán un impacto duradero en el campo de la ingeniería en ciencias de la computación.

Si bien este trabajo se enfoca en el diseño y desarrollo de la aplicación móvil, existen áreas relacionadas que podrían abordarse en futuras investigaciones, como la integración de sistemas específicos para el uso de diversos laboratorios que existen en la universidad. Este estudio sienta las bases para futuros desarrollos y exploraciones en el campo de la gestión de recursos en los laboratorios universitarios.

Justificación

Esta investigacion se realizo con el fin de satisfacer la necesidad de abordar los desafios inherentes a la gestion de laboratorio universitarios, llevar un mejor control

en el inventrario asi como en la oportunidad de aplicar tecnologías modernas y enfoques innovadores en el desarrollo para una solucion practica y efectiva que beneficiará a la comunidad académica y contribuirá al avance del campo de la ingeniería en ciencias de la computación.

Objetivos del Estudio

Objetivo General

Diseñar y desarrollar una aplicación móvil utilizando tecnologías modernas como React Native, para la aquitectura diseño Atómico, Jest para las pruebas unitarias, TypeScript para mantener un codigo legible, Swift, Kotlin para las implementaciones nativas y Firebase como la base de datos, con el fin de gestionar el acceso de los alumnos a laboratorios universitarios y facilitar el control de préstamos de inventario.

Objetivos Específicos

1. Realizar una revisión exhaustiva de la literatura existente sobre la gestión de laboratorios universitarios, el control de inventario y el desarrollo de aplicaciones móviles utilizando las tecnologías mencionadas.

2. Explorar en detalle los conceptos de diseño atomico , analizando su utilidad y aplicabilidad en el desarrollo de la aplicación móvil propuesta.

3. Desarrollar una interfaz de usuario intuitiva y eficiente utilizando React Native, aprovechando la capacidad de la plataforma para generar aplicaciones nativas multiplataforma.

4. Implementar pruebas automatizadas utilizando Jest para garantizar la calidad y estabilidad del código a lo largo del ciclo de desarrollo de la aplicación.

5. Integrar TypeScript en el desarrollo de la aplicación para mejorar la calidad del código, detectar errores de manera temprana y facilitar su mantenimiento y evolución a largo plazo.

6. Evaluar la aplicación desarrollada en términos de usabilidad, rendimiento y satisfacción del usuario, identificando posibles áreas de mejora y optimización.

7. Usar Firebase como la base de datos confiable y de facil implementacion para el desarrollo de la aplicación movil.

Estos objetivos guiarán el proceso de investigación y desarrollo de la aplicación móvil, asegurando que se alcancen los resultados esperados y se cumplan los requisitos del proyecto.

Hipótesis

Se espera desarrollar una aplicación móvil funcional y efectiva para la gestión de laboratorios universitarios, que mejore la eficiencia operativa y la experiencia del usuario. Se espera también generar conocimiento sobre el uso de tecnologías modernas en entornos académicos y contribuir al avance del campo de la ingeniería en ciencias de la computación.

Estado del Arte

En esta premisa de la implementación de una aplicación móvil, junto con el uso de tecnologías modernas, puede generar mejoras significativas en la gestión de recursos

en laboratorios universitarios, así mismo, la entrada y salida de estos. Esto se deriva de varias perspectivas clave:

1. Interfaz de Usuario Intuitiva y Eficiente:

La importancia de una interfaz de usuario (UI) intuitiva no puede ser subestimada. Estudios han demostrado que una UI bien diseñada puede mejorar significativamente la satisfacción del usuario y su eficiencia al interactuar con el sistema. En el contexto de laboratorios universitarios, una aplicación móvil con una UI intuitiva permitirá a los estudiantes:

- **Acceder y reservar equipos rápidamente:** Esto elimina la necesidad de procesos manuales y reduce el tiempo de espera, facilitando un uso más óptimo de los recursos.
- **Entrar y salir fácilmente:** Al poder llevar un mejor control de accesos y tiempo de estadía en los laboratorios, se lleva un mejor control del espacio de estos. Freixas Ramírez, Y., & Noa Pérez, V. (2015) nos hablan sobre como EUA, Francia, Inglaterra y China se destacan en la produccion y utilizacion de controladores de acceso automatizados.
- **Navegar fácilmente por las funciones de la aplicación:** Una UI bien organizada y fácil de usar reduce la curva de aprendizaje, permitiendo que los estudiantes y el personal administrativo adopten rápidamente la nueva tecnología sin necesidad de extensas capacitaciones. Acosta, A & Zambrano N. (2006) nos hablan sobre la importancia de contar con una interfaz de usuario que sea intuitiva.

2. Pruebas Automatizadas Exhaustivas:

Las pruebas automatizadas son fundamentales para asegurar la estabilidad y calidad del software. La adopción de Jest, un marco de pruebas popular, aporta varios beneficios:

- **Detección temprana de errores:** Las pruebas automatizadas permiten identificar problemas en las primeras etapas del desarrollo, lo que facilita su corrección antes de que se conviertan en problemas graves.
- **Garantía de funcionalidad continua:** Las pruebas repetidas aseguran que las nuevas actualizaciones no introduzcan errores, manteniendo la fiabilidad del sistema.
- **Reducción de costos a largo plazo:** Al detectar y corregir errores de manera temprana, se reducen los costos asociados con el mantenimiento y soporte del software a largo plazo.

Velásquez, S. M., Monsalve Sossa, D. E., Zapata, M. E., Gómez Adasme, M. E., & Ríos, J. P. (2019) nos explican sobre la necesidad e importancia de las pruebas en las aplicaciones moviles en los años actuales, el tener cada dia mas dispositivos y tecnologias a nuestra disposicion.

3. Código Legible y Mantenible con TypeScript:

La integración de TypeScript en el desarrollo de la aplicación mejora la calidad del código de varias maneras:

- **Tipado estático:** TypeScript ayuda a detectar errores de tipo en tiempo de desarrollo, mejorando la precisión y robustez del código. Boris Cherny nos comenta en Programming Typescript: Makin your JavaScritpt Aplications Sacle la importancia de un tipado estatico.
- **Documentación implícita:** El uso de tipos y interfaces actúa como una forma de documentación, facilitando la comprensión y el mantenimiento del código por parte de los desarrolladores actuales y futuros.
- **Escalabilidad:** Un código más legible y estructurado permite una evolución y escalabilidad más fluidas, lo que es crucial para adaptarse a las necesidades cambiantes de los usuarios y la institución.

4. Autosuficiencia y Rendimiento Optimizado con Código Nativo:

El desarrollo de la aplicación utilizando Swift para iOS y Kotlin para Android ofrece varias ventajas:

- **Rendimiento superior:** Las aplicaciones nativas suelen tener un mejor rendimiento en comparación con aquellas que dependen de librerías externas, proporcionando una experiencia de usuario más fluida y rápida.
- **Acceso completo a las capacidades del dispositivo:** El uso de código nativo permite aprovechar al máximo las características y funcionalidades del sistema operativo, mejorando la funcionalidad y eficiencia de la aplicación.
- **Independencia de librerías externas:** Desarrollar librerías propias evita problemas legales relacionados con derechos de autor y reduce la dependencia de terceros, promoviendo la autosuficiencia y el control total sobre la aplicación.

5. Evidencia de Beneficios en Entornos Académicos y Empresariales:

Numerosas investigaciones han demostrado que la adopción de tecnologías modernas puede mejorar significativamente la eficiencia operativa y la experiencia del usuario en diversos entornos:

- **Estudios en entornos académicos:** Investigaciones han mostrado que el uso de aplicaciones móviles y tecnologías avanzadas en instituciones educativas mejora la gestión de recursos, facilita el aprendizaje y aumenta la satisfacción de los estudiantes.
- **Experiencias en el sector empresarial:** Las empresas que han adoptado tecnologías como la automatización de pruebas, el uso de TypeScript y el desarrollo de aplicaciones con código nativo han reportado mejoras en la calidad del software, la eficiencia operativa y la satisfacción del cliente.

5. **Sistemas Tradicionales vs. Modernos:** La gestión tradicional de laboratorios universitarios ha dependido en gran medida de métodos manuales y sistemas de registro en papel. Sin embargo, la creciente complejidad y demanda de recursos en las universidades ha impulsado la adopción de sistemas automatizados.

Investigaciones recientes destacan la implementación de software especializado para la gestión de laboratorios, que facilita la reserva de equipos y el seguimiento del inventario en tiempo real.

6. Estudios de Caso en Universidades Referencias: Varias universidades han implementado sistemas de gestión de laboratorios con resultados positivos. Por ejemplo, la Universidad de California desarrolló un sistema automatizado que permite reducir los tiempos de espera para la reserva de equipos y mejorar la precisión del inventario. Otro caso relevante es el del Instituto Tecnológico de Massachusetts (MIT), que implementó una plataforma de gestión de recursos que integra funcionalidades de mantenimiento preventivo y gestión de accesos.

7. **Sistemas Automatizados de Control de Inventario:** La literatura muestra una tendencia creciente hacia el uso de sistemas automatizados de control de inventario en entornos académicos y empresariales. Estos sistemas utilizan tecnologías como RFID (Identificación por Radiofrecuencia) y códigos de barras para rastrear y gestionar el inventario de manera eficiente. Los estudios han demostrado que la automatización en el control de inventario reduce significativamente los errores humanos y mejora la precisión en la gestión de recursos.

8.Aplicaciones en el Sector Educativo Referencias: Un ejemplo notable es el proyecto de la Universidad de Stanford, que implementó un sistema de control de inventario basado en RFID para sus laboratorios de ingeniería. Este sistema permite una visibilidad en tiempo real del inventario, optimizando el uso de recursos y reduciendo las pérdidas de equipos.

Marco teorico

Para el desarrollo de este proyecto se utlizaran tecnologias modernas y altamente populares para el desarrollo. Para el lado del front end se utilizara el popular framework de Meta, React native.

React Native: React Native es una tecnología popular para el desarrollo de aplicaciones móviles multiplataforma. Permite a los desarrolladores escribir código en JavaScript y renderizarlo en aplicaciones nativas tanto para iOS como para Android. Como nos explican Lazcano Calixto, Ricardo Neftali, Valencia González, Luis Ángel, Baena Díaz, Daniel Esteban y Venegas Guzmán, Ricardo (2019), React Native cambia el juego en el desarrollo de aplicaciones moviles, al nacer la necesidad de mejorar los tiempos de desarrollo tanto como los costos, se creo una alternativa llamada cordova, que es un empaquetador de codigo HMLT, CSS y JavaScript que renderiza el contenido dentro de un WebView. Sin embargo esto genera que el rendimiento de la aplicación sea sacrificado, ademas de que el contenido tiende a ser de mala calidad y de no tan buen aspecto. Es cuando React Native llega a cambiar el juego, creando aplicaciones realmente nativas, no generando un WebView, si no que transpila JavaScript a los idiomas nativos. Mediante una serie de componentes preconstruidos que se traducen directamente a los componentes nativos de iOS y Android, tales como <View>, <Text> o <Button>. React Native los convierte en sus equivalentes nativos:

- iOS: Utiliza componentes de UIKit.
- Android: Utiliza componentes de Android View

Estudios han mostrado que React Native reduce los costos y el tiempo de desarrollo, manteniendo un rendimiento cercano al de las aplicaciones nativas.

La arquitectura de diseño que utilizaremos para la aplicación sera el **diseño atómico**. Frost, B. (2016) fue el creador de esta arquitectura revisar si ya hable de el, donde nos explica mediante una comparacion a la tabla periodica de elementos, como podemos organizar una aplicación para generar codigo mantenible y reutilizable, esto nos

ayudara tambien a que sea un proyecto escalable. El diseño atómico tiene 5 niveles importantes, los cuales son:
Agregar ejemplos de cada uno con imágenes

- Atomos (Atom): Estos son los niveles basicos de la creacion de la materia, en nuestro modelo simbolizan a los compnoentes mas basicos del proyecto, como son textos, imágenes, inputs o botones.
- Moleculas (Molecules): Estos son arreglos de atomos, donde se utilizan componentes basicos para crear un componente mas sofisticado, como podrian ser formularios, un logo con texto, etc.
- Organizmos (Organisms): Estos son componentes complejos, que utilizan moleculas para mostrar una interfaz al usuario, como pueden ser headers que utilizan un logo con texto junto con un menu desplegable, etc.
- Templates: Un template es un grupo de organizmos que se unen para formar paginas, pero contienen informacion real y no una simulacion.
- Pantallas (Pages): Son instancias de los templates, aquí el contenido que muestra es el contenido final dirigido hacia el usuario.

Jest es un framework de pruebas automatizadas ampliamente utilizada en el desarrollo de aplicaciones JavaScript. Este fue creado por Meta y en 2022 fue tranferido a la fundacion OpenJS para que la comunidad se hiciera cargo del proyecto. El uso de jest en nuestro proyecto es fundamental para realizar pruebas unitarias y de integracion. Facil de usar, no necesita ser configurado dado que React Native viene por defecto cuando se inicializa el proyecto. Cuenta con una documentacion exaustiva y facilita la detección temprana de errores y mejora la confiabilidad del software.

TypeScript, un superconjunto de JavaScript, ha ganado popularidad por su capacidad para mejorar la legibilidad y mantenibilidad del código. Utilizado para crear un tipado

estatico que permite declarar tipos de datos, esto nos ayuda a detectar errores en tiempo de compilacion evitando problemas en la ejecucion. Contando con una gran documentacion es la mejor opccion para mejorar la calidad de nuestro codigo.

Swift y Kotlin son lenguajes de programación nativos para iOS y Android, respectivamente. Ambos lenguajes ofrecen ventajas significativas en términos de rendimiento y acceso a las funcionalidades del sistema operativo. Las aplicaciones desarrolladas con Swift y Kotlin tienen un rendimiento superior y una integración más profunda con las características nativas del dispositivo en comparación con las aplicaciones multiplataforma. Gracias al uso de React Native, podemos generar los dos proyectos de iOS y Android sin necesidad de crear dos proyectos, con un solo codigo podemos crear las aplicaciones para ambas plataformas.

Finalmente, tenemos a Firebase, una plataforma de desarrollo de aplicaciones moviles y web creada por Google en 2011. Para nuestro proyecto utilizaremos las funcionalidades:
Agregar imágenes

- Base de datos en tiempo real: Realtime Database una base NoSQL que nos permite alamcenar y sincronizar datos en tiempo real.
- Autenticacion: Nos proporciona servicios de autenticacion que permiten a los usuarios registrarse y acceder a la aplicación por los multiples y conocidos metodos de autenticacion, como son: correo electronico y contraseña, este sera el metodo utilizado para el proyecto, Google, Facebook, Apple, etc.
- Notificaciones push: Firebase Cloud Messagign (FCM) es un servicio que permite enviar notificaciones y mensajes a los usuarios de la apliacaion en tiempo real, este nos ayudara a notificar prestamos y entregas de herramientas de los laboratorios a los encargados y docentes.

Firebase nos entrega una interaccion sensilla, ofrece SDKs para las plataformas Android e iOS, ademas de tener soporte para React Native por parte de la comunidad.

Proceso de Desarrollo de la Aplicación

Análisis de Requisitos

En este capitulo se abordara el analizis de requisitos para el desarrollo de la aplicación movi. Este análisis abarca la gestion de entradas y salidas de usuarios, la gestion de inventario y el sistema de prestamos. El objetivo es establecer de manera clara las necesidades de sistema y las interacciones entre los diferentes actores.

Identificacion de los actores.

Primeramente se identificaran a los actores que interactuan con el sistema. Los actores, los usuarios o sistemas, que participaran en los diferentes procesos de la aplicacion. Para este sistema, se identificaron los siguientes actores clave:

- Responsable del laboratorio: Este es el usuario encargado de gestionar el laboratorio, siendo el responsable de registrar, editar y eliminar el inventario. Ademas de supervisar las solicitudes de préstamos y el registro de entradas y salidas de los usuarios.
- Docente: Los docentes tendran acceso a los registros del sistema, como son las entradas y salidas de los estudiantes y otros usuarios, asi como a los detalles de los prestamos y del inventario.
- Estudiante/Usuario: Los estudiantes registrados podran solicitar el prestamo de herramientas atravez de la aplicación. Ademas de regisrar su entrada y salida del laboratorio cuando accedan o salgan de las intalaciones.

Requisitos funcionales

Los requisitos funcionales nos describen las fucnionalidades que debe cumplir este sistema para satisfacer las necesidades de los diferentes actores. Acontinuacion dse

describiran los principales modulos de la aplicación y los requisitos funcionales asociados.

Registro de usuarios.

En este modulo se hara el registro de los usuarios mediante la implementacion de registro en firebase usando un correo electronico y una contraseña, el usuario debe agregar su matricula y nombre para poder ser identificado.

- RF01. El sistema debe permitir al usuario ingresar a la pantalla de registro, donde ingresara su nombre, matricula, correo y una contraseña donde estara la opccion de repetirla para asegurar la experiencia de usuario correcta al evitar que la contraseña sea olvidada o mal escrita.
- RF02. El sistema debe retornar un aviso al usuario si fue correcto el registro y llevarlo a la pagina principal o si hubo algun error al ingresar su informacion en Firebase.
- RF03. El sistema debe permitir al usuario iniciar sesion con sus credenciales, llevandolo a la pagina principal si se autentifico, o mostrando si exisitio algun error al iniciar sesion.

Gestion de entradas y salidas de usuarios.

En este modulo se permitira a los usuarios registrar su entrada y salida del laboratorio, algo escencial para controlar el acceso a las intalaciones y maneter un registro histoico del uso de estas.

- RF04. El sistema debe permitir a los usuarios registrar su entrada y salida del laboratorio utlizando un identificador unico, en este caso un codigo QR.
- RF05. La aplicación debe registrar la hora de entrada y salida del laboratorio, asi como la fecha y almacenar esta informacion en Firebase.

Gestion de inventario

En este modulo se podra revisar la existencia de los insumos del laboratorio, asi mismo si fueron prestados y a quien fueron prestados.

- RF06. EL sistema debe permitir a los encargados y docentes revisar el estado del inventario, mostrando la cantidad de estos y a quienes fueron prestados si fue el caso
- RF07. El sistema debe permitir buscar objectos del inventario mediante una busqueda.

Modificacion del inventario

En este modulo se hara el registro, modificacion y eliminacion del inventario del laboratorio.

- RF08. El sistema debe permitir hacer el registro de herramientas al inventario, mediante su nombre, cantidad, tiempo limite de prestamo y un codigo de identificacion para las herramientas.
- RF09. El sistema debe permitir hacer la modificacion de herramientas al inventario, tanto en nombre y cantidad.
- RF10. El sistema debe permitir hacer la eliminacion de herramientas del inventario.

Prestamos de herramientas.

En este modulo se hara el prestamo de herramientas del laboratorio.

- RF11. El sistema debe permitir el prestamo de herramientas mediante la lectura de un codigo QR.

- RF12. El sistema debe avisar al responsable y al docente, que una herramienta fue prestada y los datos del estudiante.
- RF13. El sistema debe permitir el retorno de la herramienta, lo que actualizara automaticamente el inventario.
- RF14. El sistema debe notificar al usuario la fecha de devolucion de la herramienta.

Funcionalidades para docentes.

En este modulo se proporciona a los docentes herramientas para supervisar las actividades del laboratorio y el uso de herramientas por parte de los estudiantes.

- RF15. Los docentes deben poder acceder a un panel de control donde puedan revisar el historial de entradas y salidas por usuario, permitiendo filtrar por fecha.
- RF16. EL sistema debe permitir a los docentes consultar el historial de prestamos, pudiendo filtrar por fechas, usuarios o herramientas.

Requisitos no funcionales.

Ademas de los requisitos funcionales, el sistema de cumplir con una serie de requisitos no funcionales que garanticen su funcionalidad, cumpliendo con estandares de calidad, seguridad y rendimiento. Los principales requisitos no funcionales son los siguientes:

- RNF01. Seguridad. El acceso a la aplicación debe estar protegido mediante un sistema de autenticacion seguro. Los usuarios deben tener roles bien definidos (responsable, docente y estudiante) y el acceso debe estar restringido en funcion de estos roles.

- RNF02. Usabilidad. La interfaz de usuario debe ser intuitiva y facil de utilizar, con una navegacion sensilla, y pantallas accesisbles, mostrando la informacion de manera clara.
- RNF03. Rendimiento. El sistema debe ser capaz de gestionar multiples usuario de manera concurrente, sin afectar el rendimiento de la aplicación.
- RNF04. Portabilidad. El sistema debe ser compatible con dispositivos Android e iOS, asi asegurando que los usuarios puedan acceder al sistema desde cualquier dispositivo movil.
- RNF05. Escalabilidad. El sistema debe ser capaz de expandirse para gestionar multiples laboratorios si fuera necesario en el furuto, ademas de ser facilmente escalado con nuevas funcionalidades para las necesidades diferentes de cada laboratorio, asi como con otros sistemas de la universidad.

Casos de uso

Una vez definidos los requisitos, procederemos a la creacion de los diagramas de caso de uso para visualizar como interactuan los diferentes actores con el sistema. Estos nos describiran las interacciones y nos aseguraremos que todas las fucnionalidades esten cubiertas.

1) Registro de Usuarios e inicio de sesion.
    a) Caso de uso 1: Registros de usuario.
        i) Actor principal: Usuario.
        ii) Precondiciones: EL usuario tiene la aplicación instalada y ejecutandose.
        iii) Flujo principal:
            (1) El usuario selecciona la opcion “Registro”
            (2) El sistema muestra un formulario para ingresar nombre, matrícula, correo y contraseña.
            (3) El usuario completa el formulario y confirma la contraseña.
            (4) El sistema valida la informacion y registra al usuario en Firebase.

(5) EL sistema muestra un mensaje de éxito o error.

(6) Si el registro es exitoso, el sistema redirige al usuario a la página principal.

(7) Si el registro no es exitoso, el sistema indica el error al usuario.

Como se muestra en la figura 1, el flujo del caso de uso de registro de usuario es esencial para establecer la base de usuarios del sistema, permitiendo el acceso a las funciones del sistema y mejorando la experiencia general del usuario.

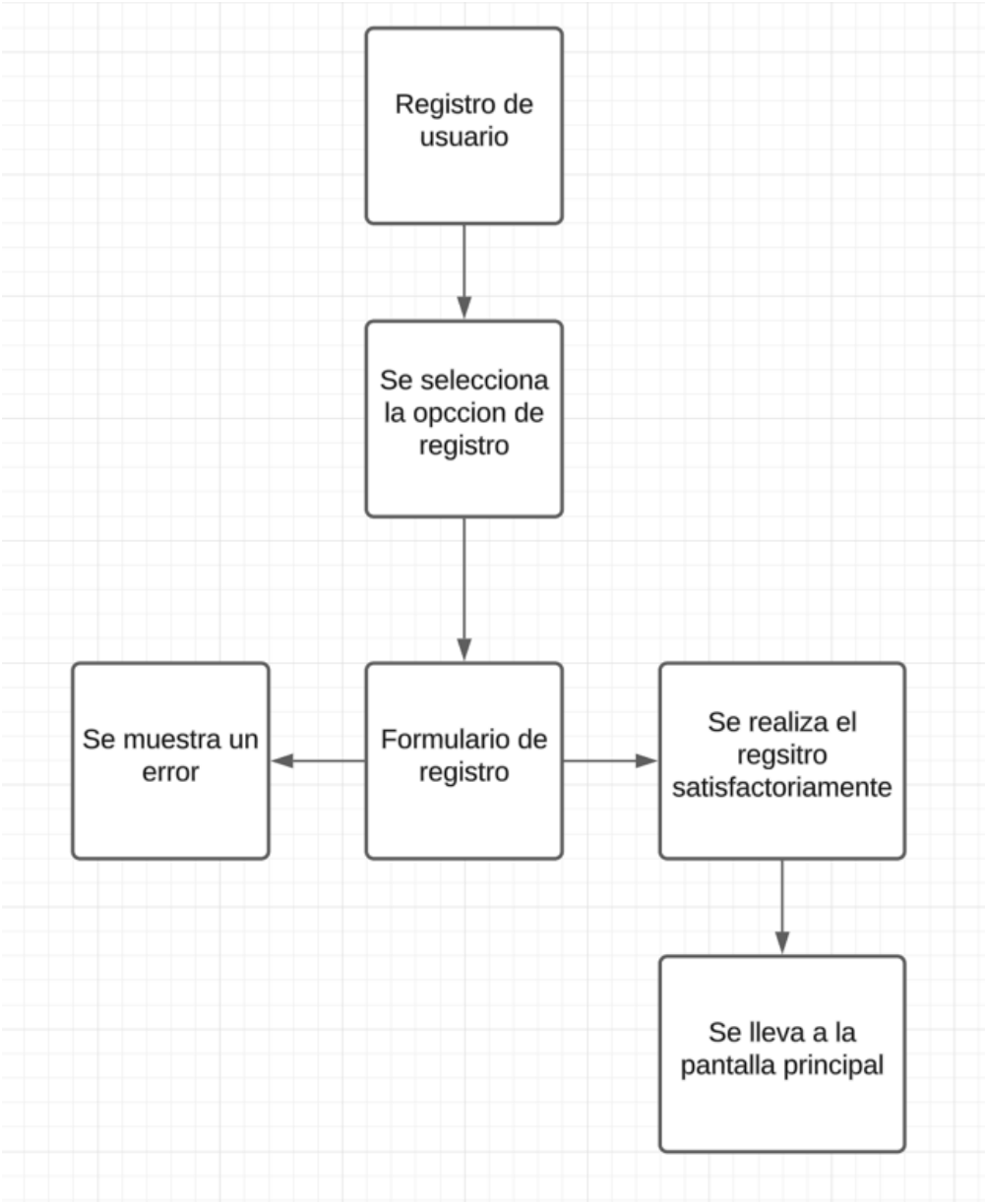

Figura 1 Flujo de registro de usuario

b) Caso de Uso 2: Inicio de sesion del usuario.

i) Actor principal: Usuario

ii) Precondiciones: El usuario se a regsitrado en el sistema satisfactoriamente.

iii) Flujo principal:

(1) El usuario selecciona la opccion “Inico de sesion”

(2) El usuario debe llenar el formulario con su correo y contraseña.

(3) El sistema debe validar los datos y encaso de éxito mandar a la pantalla principal, en cambio, debe mandar un aviso del error al usuario.

El flujo del caso de uso para el inicio de sesión, representado en la figura 2, detalla las etapas que un usuario debe seguir, así como las interacciones necesarias con el sistema para autenticarse y acceder a la plataforma.

Figura 2 Flujo de inicio de sesion

2) Registro de acceso y salida del usuario.
   a) Caso de uso 3: Acceso al laboratorio por el estudiante.
      i) Actor principal: Estudiante
      ii) Precondicones: El estudiante debe haber iniciado sesion y entrado a la pagina principal.

iii) Flujo principal:

(1) El estudiante selecciónа la opcion acceso a laboratorio.

(2) El estudiante debe escanear un qr con la informacion del laboratorio.

(3) El sistema debe registrar los datos de entrada, hora, fecha y matricula del usuario.

(4) El sistema debe avisar al estudiante si el acceso fue exitoso o si existio una falla.

En la figura 3 se refleja el flujo de caso de uso del acceso al laboratorio por el estudiante, lo que permite entender mejor cómo se desarrollan las interacciones en este proceso.

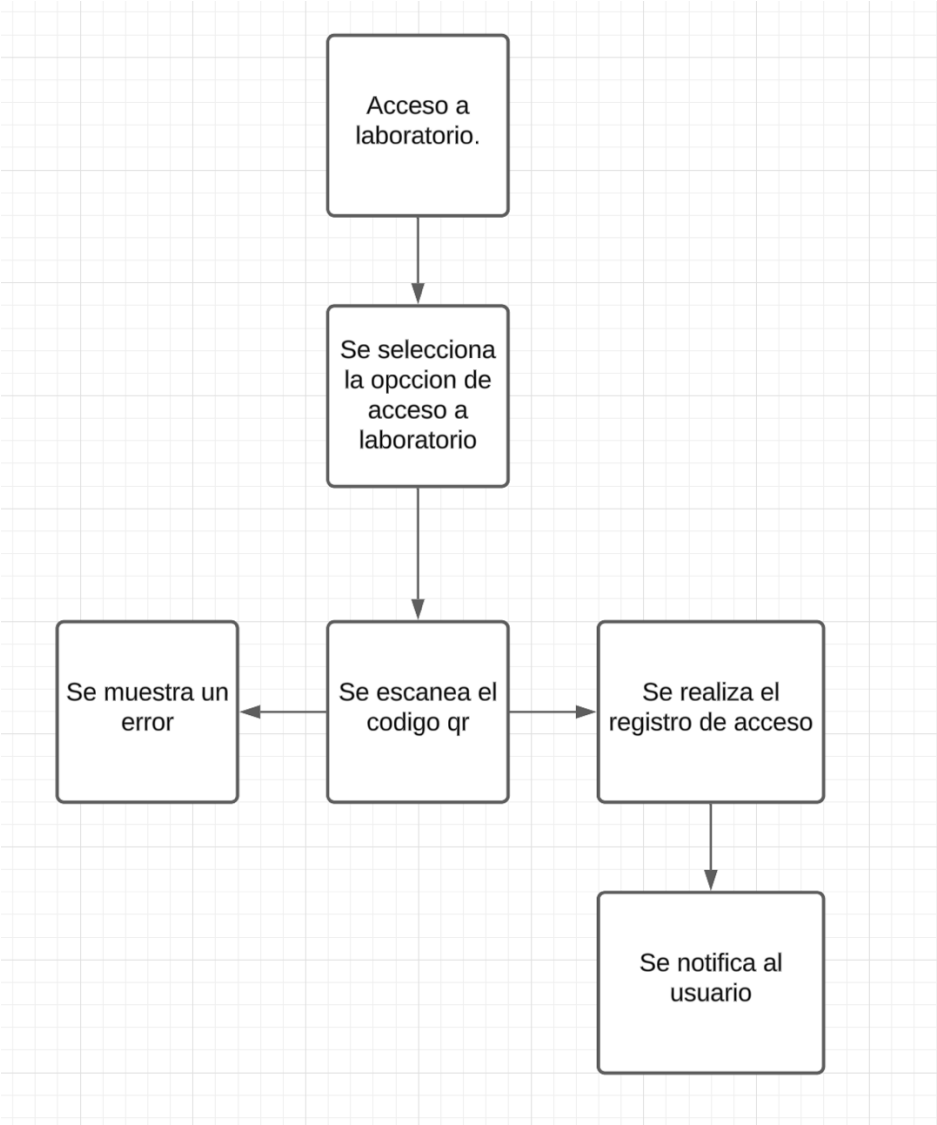

Figura 3. Flujo de acceso a laboratorio.

b) Caso de uso 4: Salida del laboratorio por el estudiante.
   i) Actor principal: Estudiante.
   ii) Precondiciones: El estudiante debe haber accedido al laboratorio por medio de el escaneo del qr.
   iii) Flujo principal:
      (1) El estudiante slecciona la opcion de salida del laboratorio.
      (2) El estudiante debe escanear un qr con la infrmacion del laboratorio.
      (3) El sistema debe actualizar con los datos de salida, hora y fecha el anterior acceso.

(4) El sistema debe avisar al estudiante si la salida fue exitosa o si existio una falla.

En la figura 4 se representa el flujo de salida del laboratorio por el estudiante, un procedimiento garantiza el cumplimiento de las normativas de seguridad y control.

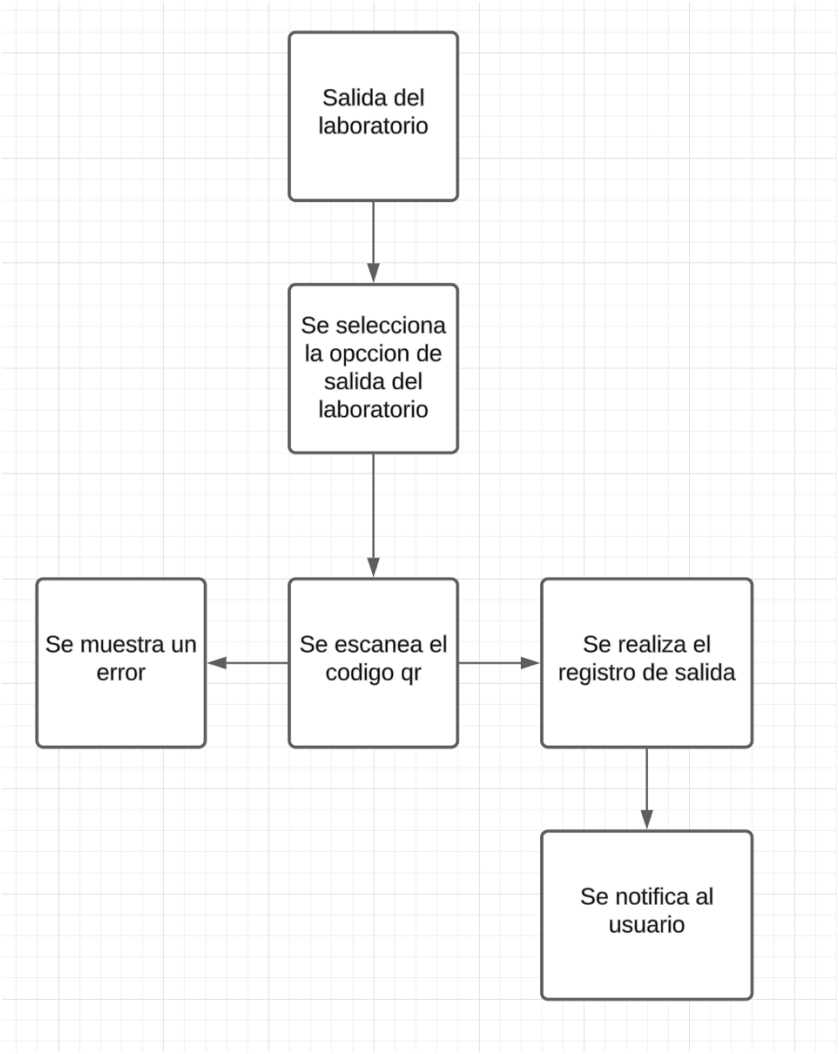

Figura 4. Flujo de salida del laboratorio.

3) Gestion del inventario
   a) Caso de uso 5: Revision del inventario.
      i) Actor principal: Encargados y docentes.
      ii) Precondiciones: El usuario debe haber accedido a la aplicación y estar en la pantalla principal.
      iii) Flujo pincipal:

(1) El usuario selecciona la opcion de revision de inventario.

(2) El usuario ingresa el nombre de la herramienta.

(3) El sistema valida la informacion y busca las existencias.

(4) El sistema retorna la informacion o avisa si no encontro una herramienta.

En la figura 5 se presenta el flujo de revisión del inventario, que incluye desde la selección de la opción correspondiente hasta la validación del nombre de la herramienta por parte del sistema, lo que asegura una gestión precisa de los recursos disponibles.

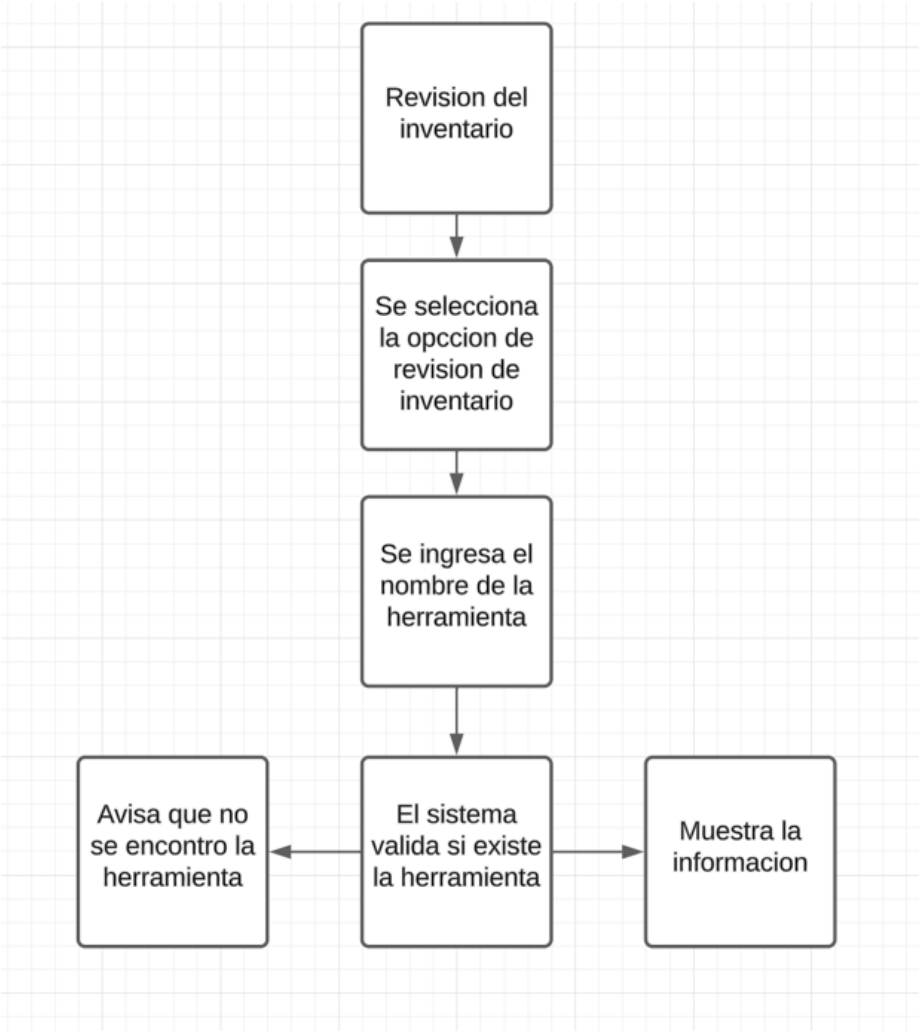

Figura 5. Flujo de revision de inventario.

b) Caso de uso 6: Añadir inventario

i) Actor principal: Encargados y docentes.

ii) Precondiciones: El usuario debe haber accedido a la aplicación y estar en la pantalla principal.

iii) Flujo pincipal:

(1) El usuario selecciona la opcion de añadir de inventario.

(2) El usuario ingresa el nombre, cantidad y tiempo de prestamos de la herramienta.

(3) El sistema valida la informacion.

(4) El sistema agrega la herramienta o avisa si se encontro algun error.

El caso de uso 6, presentado en la figura 6, detalla el proceso de añadir inventario, permitiendo a los actores principales, encargados y docentes, incorporar nuevas herramientas al sistema de gestión.

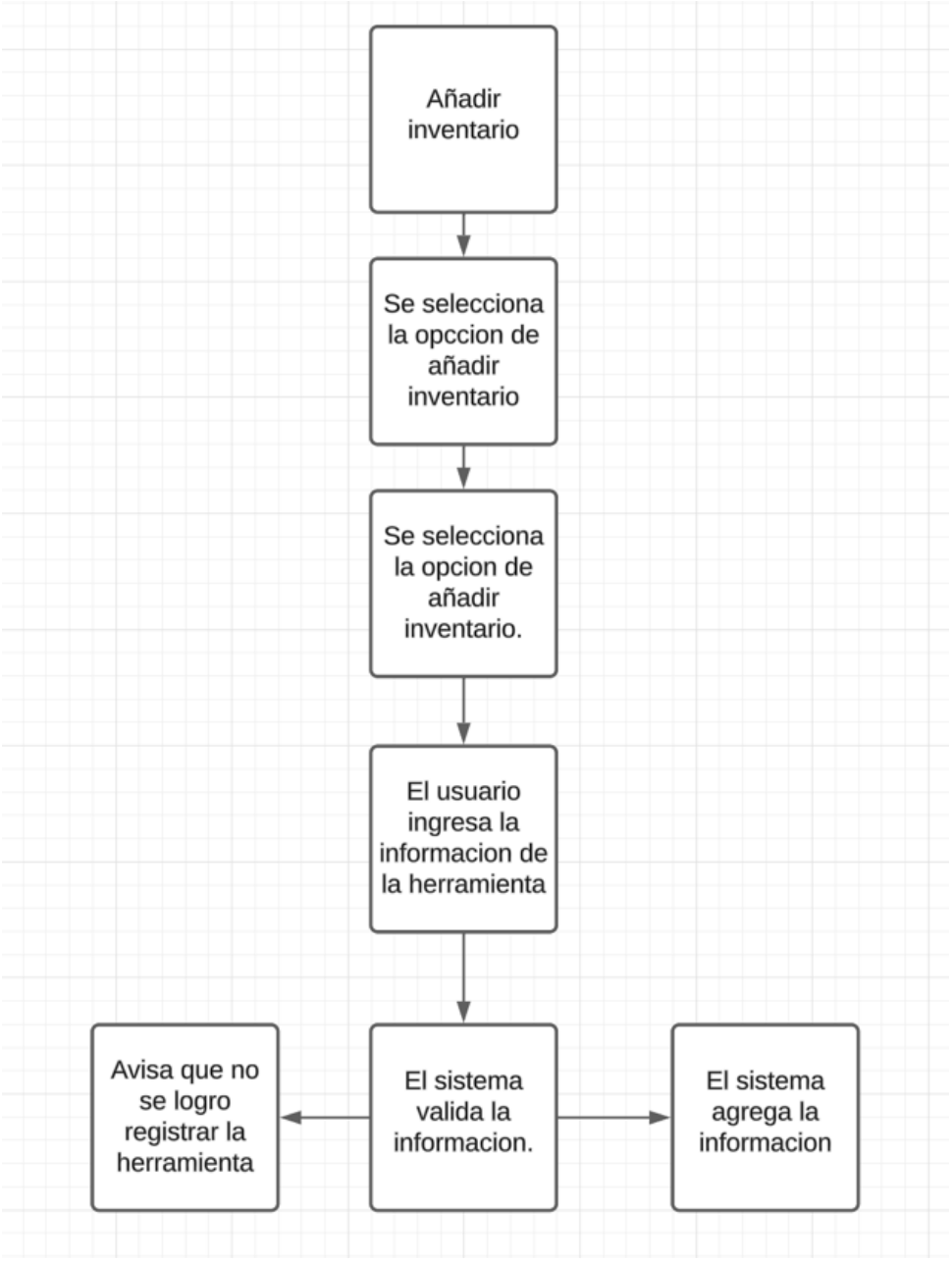

Figura 6. Flujo de añadir inventario.

c) Caso de uso 7. Modificacion de la cantidad inventario.

i) Actor principal: Encargados y docentes.

ii) Precondiciones: El usuario debe haber accedido a la aplicación y estar en la pantalla principal.

iii) Flujo pricipal:

(1) El usuario selecciónа la opccion de modificacion del inventario.

(2) El usuario selecciona el checkbox para solo editar la cantida del inventario.
(3) El sistema habilita el input de cantidad para modificar el inventario.
(4) El usuario ingresa el nombre de la herramienta y la cantidad nueva.
(5) El sistema valida la informacion.
(6) El sistema retorna el éxito o avisa si no se encontro una herramienta.

El caso de uso 7, mostrado en la figura 7, detalla el proceso de modificación de la cantidad de inventario, permitiendo a los encargados y docentes, actualizar las existencias de herramientas en el sistema.

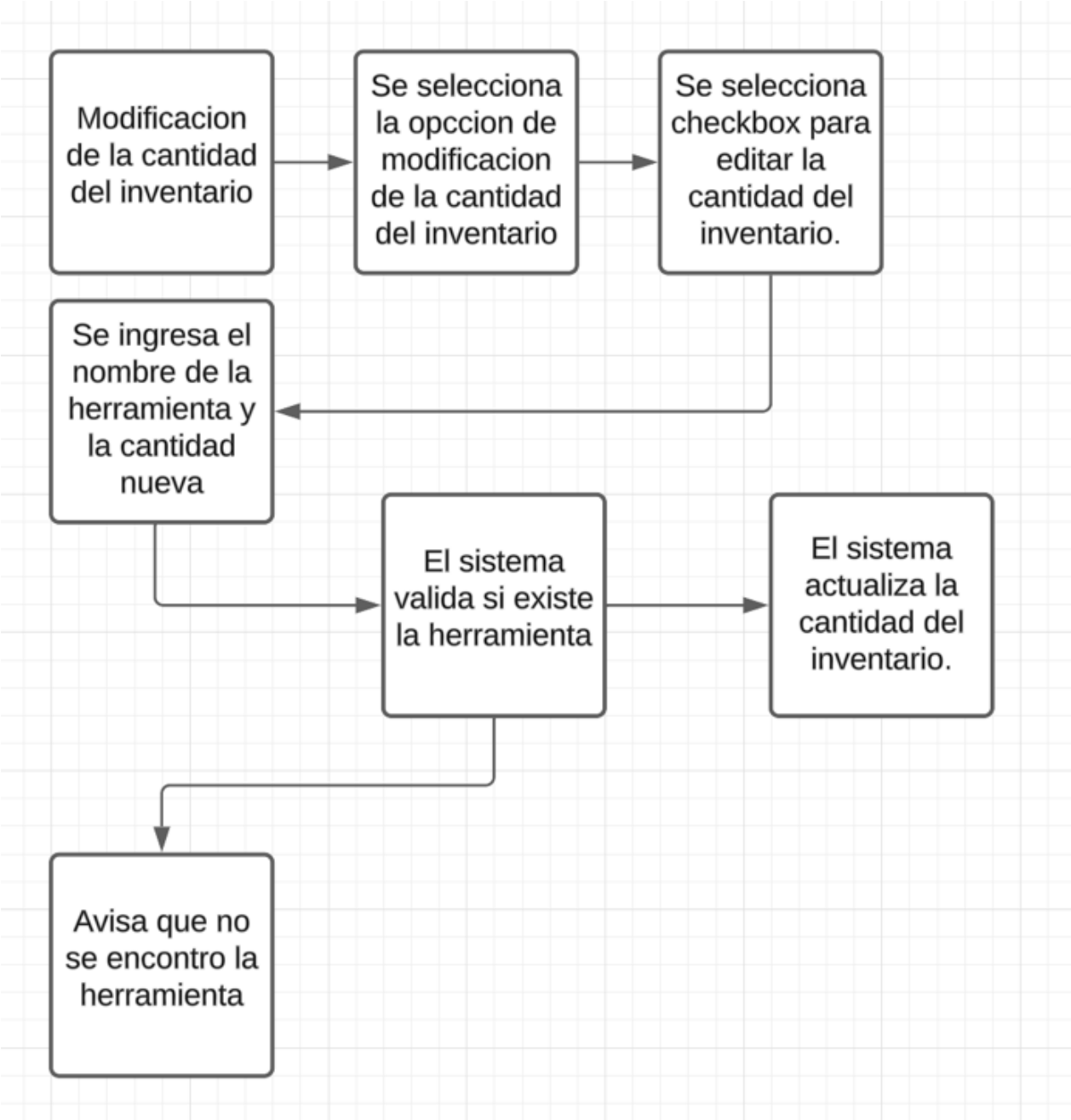

Figura 7. Flujo de modificacion de la cantidad del inventario.

d) Caso de uso 8. Modificacion de atributos del inventario.

   i) Actor principal: Encargados y docentes.

   ii) Precondiciones: El usuario debe haber accedido a la aplicaion y estar en la pantalla principal.

   iii) Flujo principal:

(1) El usuario selecciona la opccion de modificacion del inventario.

(2) El usuario ingresa los nuevos parametros para el inventario.

(3) El sistema valida la informacion.

(4) El sistema retorna el éxito o avisa si no se encontro una herramienta.

Como se muestra en la figura 8, los encargados y docentes pueden iniciar el flujo para modificar los atributos del inventario, lo que es fundamental para mantener datos precisos y actualizados.

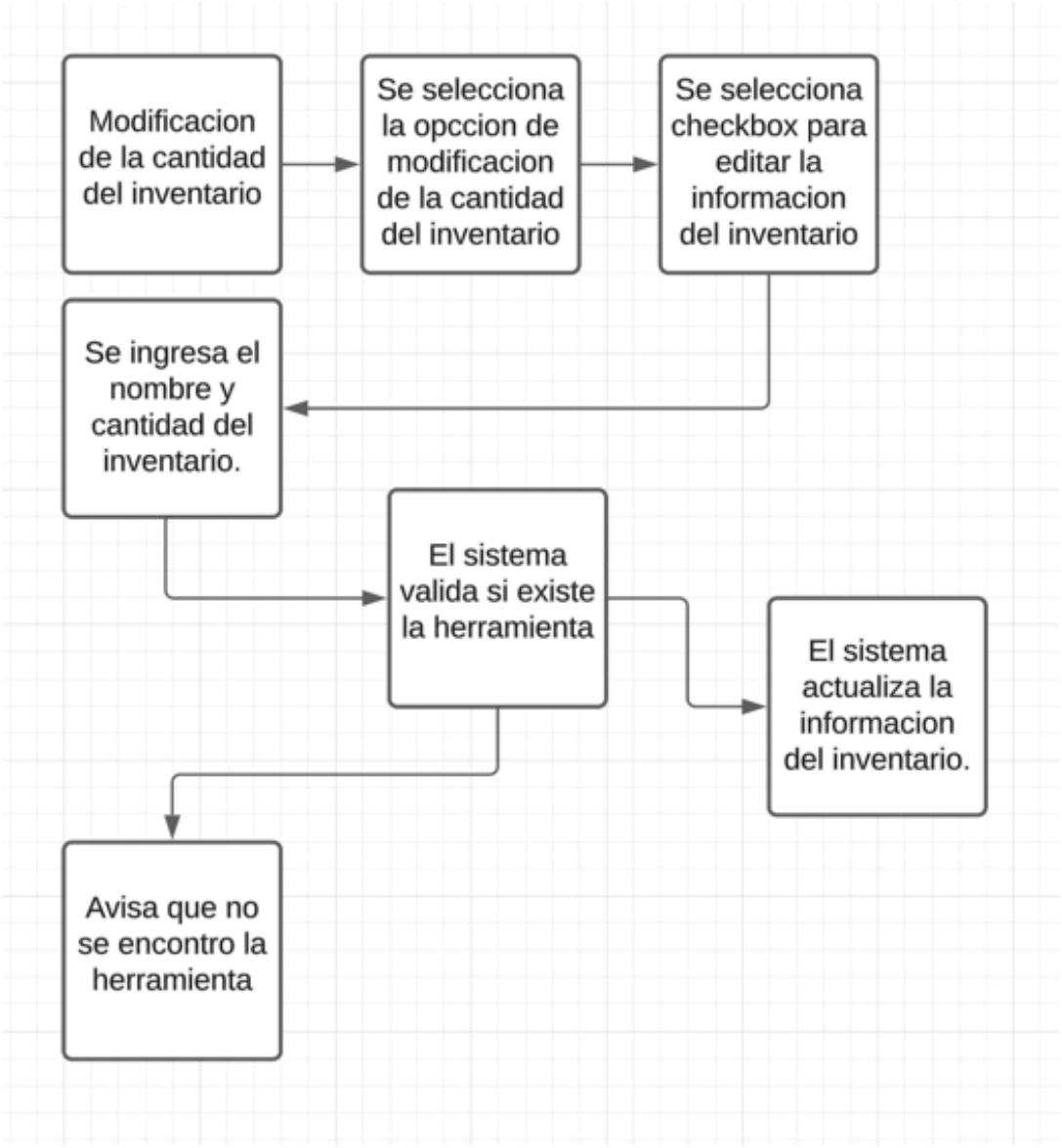

Figura 8. Flujo de modificacion de los atributos del inventario.

e) Caso de uso 9. Eliminacion del inventario,

i) Actor principal: Encargados y docentes.
ii) Precondiciones: El usuario debe haber accedido a la aplicaion y estar en la pantalla principal.
iii) Flujo principal:
    (1) El usuario selecciona la opccion de modificacion del inventario.
    (2) El usuario selecciona el chekbox para eliminar el usuario.
    (3) El sistema desbloquea el boton elminar inventario.
    (4) El usuario ingresa el nombre de la herramienta..
    (5) El sistema valida la informacion.
    (6) El sistema retorna el éxito y avisa al encargado.
    (7) El sistema avisa si no se encontro una herramienta.

Como se muestra en la figura 9, los encargados y docentes, una vez que han accedido a la aplicación y están en la pantalla principal, pueden iniciar el flujo para eliminar elementos del inventario, lo cual es vital para mantener un registro actualizado y eficiente.

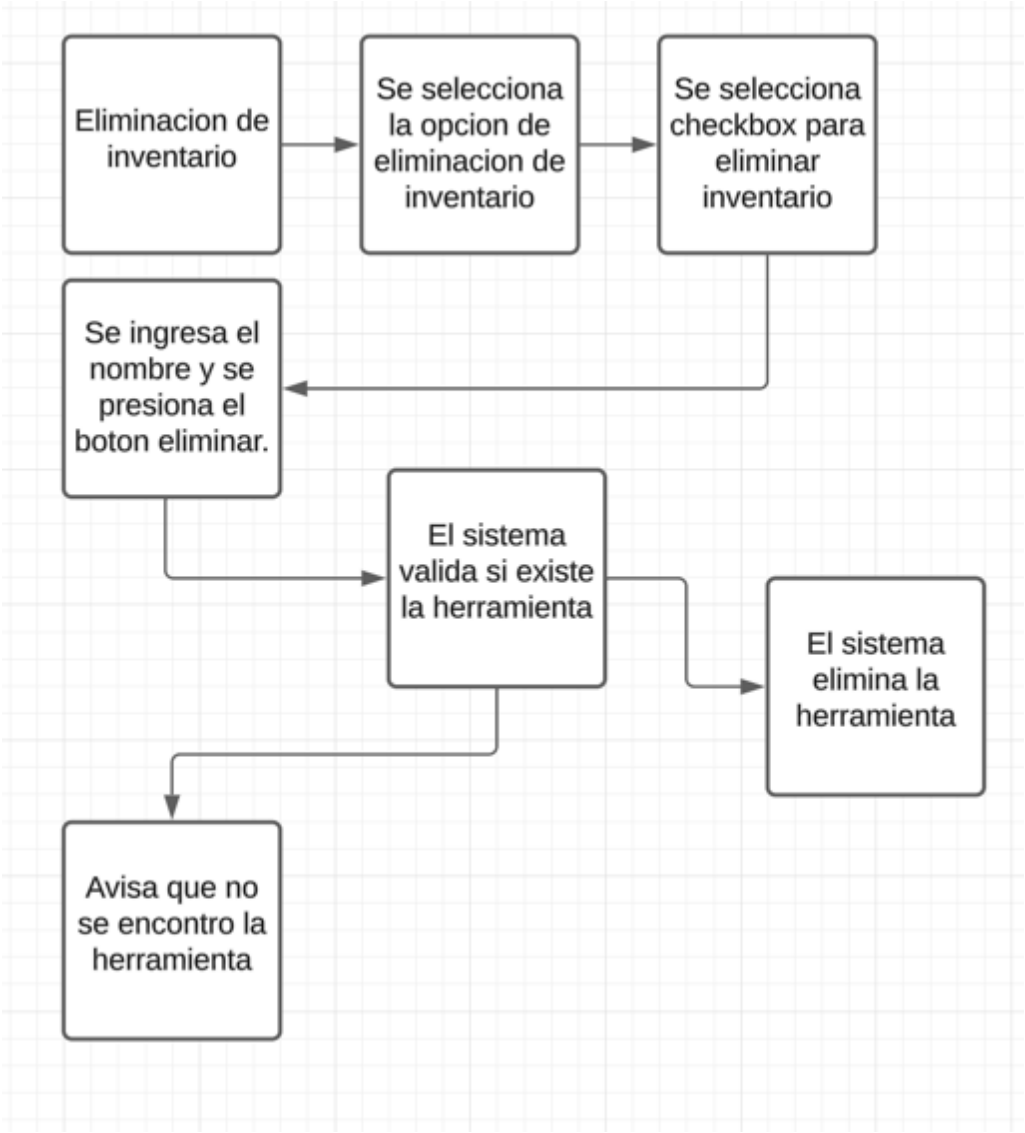

Figura 9. Flujo de eliminacion de herramienta.

4) Prestamo de herramienta del laboratorio.
   a) Caso de uso 10. Prestamo de inventario.
      i) Actor principal. Estudiante.
      ii) Precondiciones: El usuairo debe haber accedido a la aplicación y estar en la pantalla principal.
      iii) Flujo principal:
         (1) El usuario selecciona la opccion de prestamo de herramientas.
         (2) El usuario escanea el qr de la herramienta.
         (3) El sistema valida la informacion.

(4) El sistema agrega a sus prestamos la herramienta y avisa al encargado y al docente.

(5) El sistema avisa si hubo un error.

En la figura 10, el flujo principal del caso de uso prestamo de inventario se describe desde la selección de la opción de préstamo de herramientas hasta la validación de la información por parte del sistema, asegurando que todos los pasos se ejecuten correctamente.

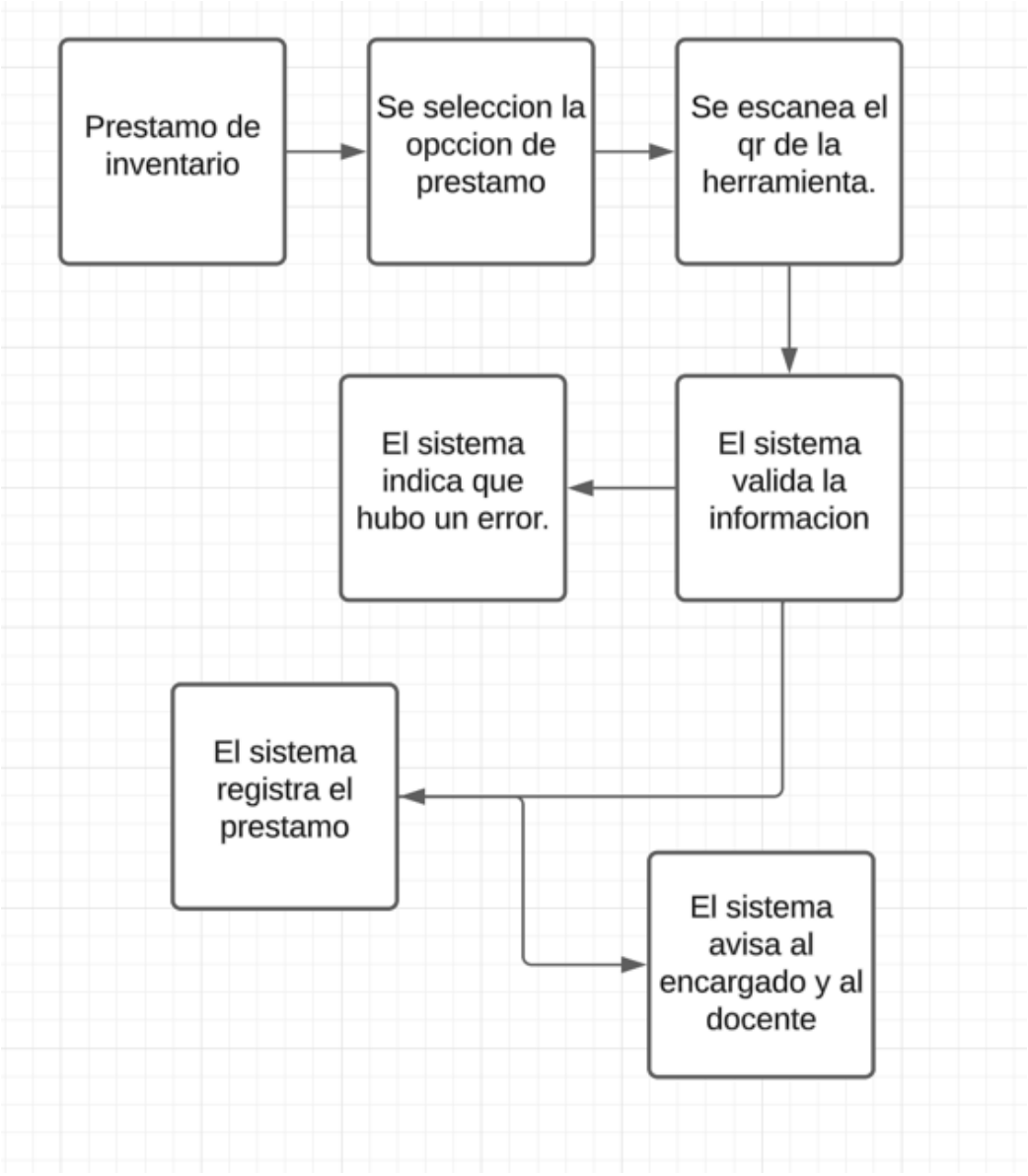

Figura 10. Flujo prestamo de equipo.

b) Caso de uso 11. Retorno de inventario

i) Actor principal. Estudiante.

ii) Precondiciones: El usuairo debe haber accedido a la aplicación y estar en la pantalla principal.

iii) Flujo principal:

(1) El usuario selecciona la opccion de regreso de herramientas.

(2) El usuario escanea el qr de la herramienta.

(3) El sistema valida la informacion.

(4) El sistema eliminala de sus prestamos la herramienta y avisa al encargado y al docente.

(5) El avisa si hubo un error.

El caso de uso de retorno de inventario, ilustrado en la figura 11, describe el proceso de retorno de inventario, donde el actor principal, el estudiante, gestiona la devolución de herramientas que ha utilizado durante sus actividades.

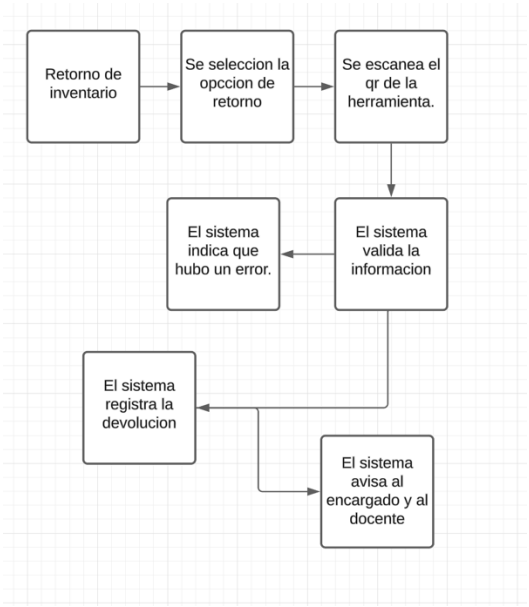

Figura 11. Flujo de retorno de inventario.

c) Caso de uso 12. Estado del prestamo.

i) Actor principal. Estudiante.

ii) Precondiciones: El usuairo debe haber accedido a la aplicación y estar en la pantalla principal.

iii) Flujo principal:

(1) El usuario selecciona la opccion de mis prestamos.

(2) El sistema pide la informacion sobre los prestamos del usuario.

(3) El sistema muestra los prestamos o avisa si no existe alguno.

(4) El sistema muestra la herramienta y el limite de tiempo del prestamo.

(5) El sistema muestra una advertencia si el limite del tiempo del prestamo fue superado y avisa al docente y al encargado.

Como se muestra en la figura 12, el estudiante, después de acceder a la aplicación y estar en la pantalla principal, puede seleccionar la opción para revisar sus préstamos, facilitando el seguimiento de los recursos utilizados.

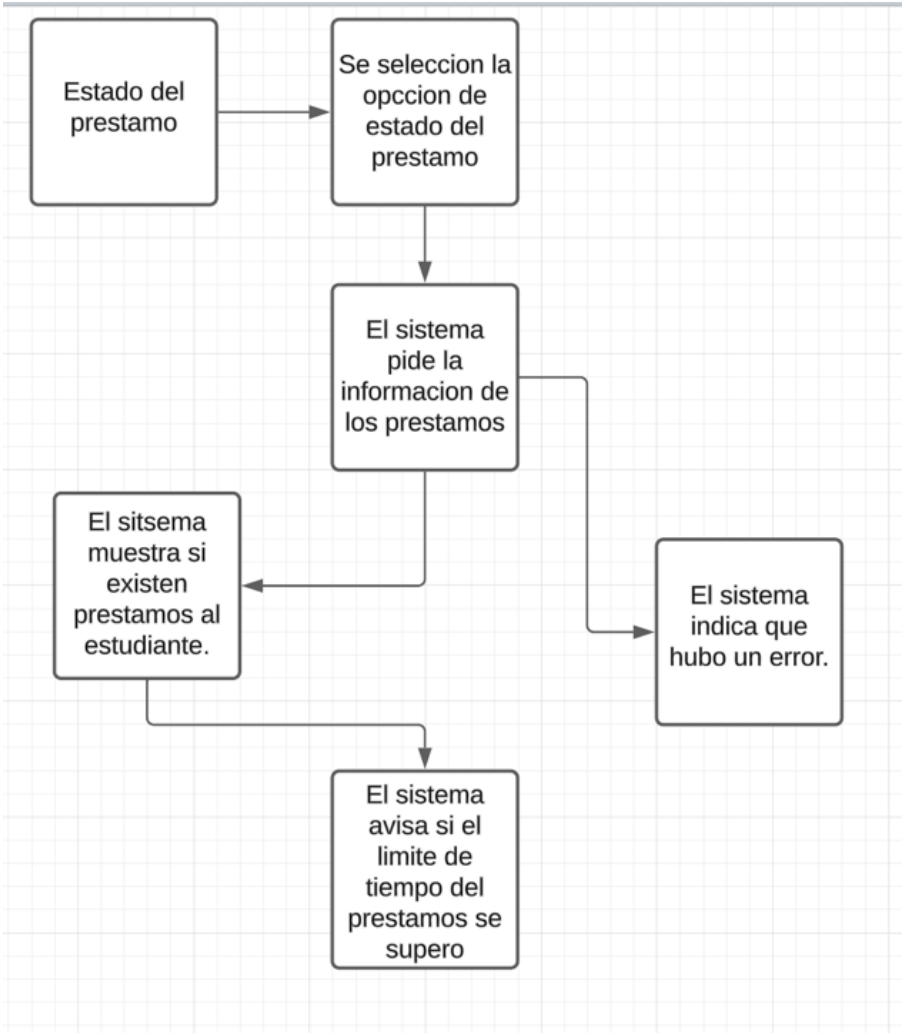

Figura 12. Flujo del estado del prestamo.

5) Funcionalidades para docentes.
   a) Caso de uso 13. Revisar historial de entradas y salidas por usuario.
      i) Actor principal: Docente y encargado
      ii) Precondicones: EL usuario debe haber accedido a al aplicación y estar en la pantalla principal.
      iii) Flujo principal:
         (1) El usuairo selecciona la opccion verificar entradas y salidas.
         (2) El usuario selecicona la fecha a consultar.
         (3) El sistema obtiene la informacion sobre los usuarios activos y salidas.

(4) El sistema muesrra los usuarios en el laboratorio, los que ya han hecho su salida o si no existe alguno.

En la figura 13, se detalla el flujo principal del caso de uso revisar historial de entradas y salidas por usuario, desde la selección de la opción para verificar entradas y salidas hasta la consulta de la fecha, asegurando que el sistema pueda proporcionar datos precisos sobre los usuarios activos.

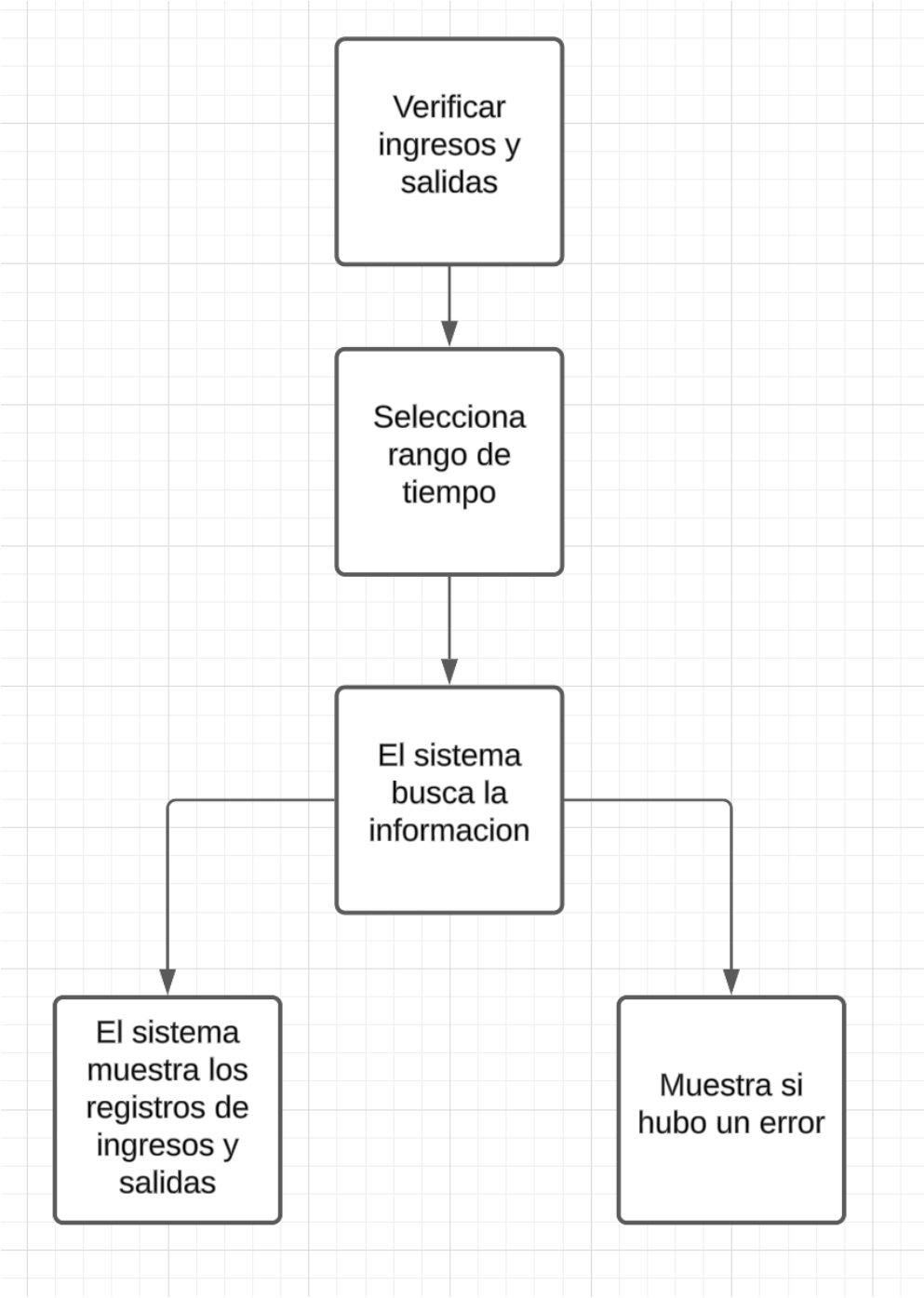

Figura 13. Flujo de verificacion de ingresos.

b) Caso de uso 14: Revisar el historial de prestamos.

   i) Actor principal: Encargado y docente.

   ii) Precondicones: EL usuario debe haber accedido a al aplicación y estar en la pantalla principal.

iii) Flujo principal:

(1) El usuario selecciona la opccion verificar prestamos.

(2) El usuario selecciona el rango de fechas de los prestamos, asi como activos, devuletos, retrasdos.

(3) El sistema obtiene la informacion sobre los los prestamos con los parametros del docente.

(4) El sistema muestra los prestamos de herramientas filtrados.

Como se muestra en la figura 14, tanto el encargado como el docente, pueden iniciar el flujo para verificar los préstamos, asegurando un seguimiento preciso de las herramientas utilizadas.

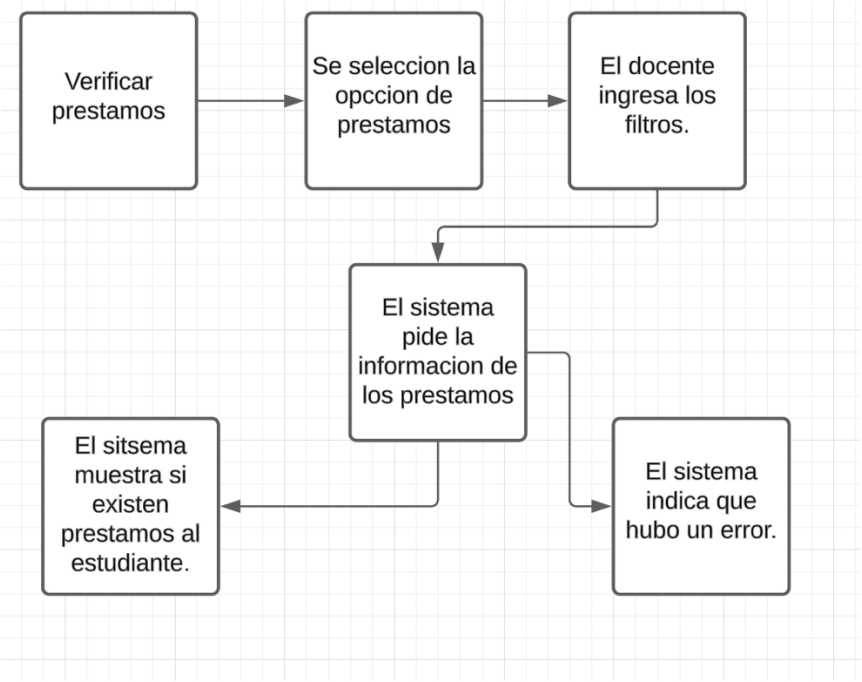

Figura 14. Flujo de verificacion de prestamos.

6) Funcionalidades filtradas

a) Caso de uso 15: Filtrar las opcciones mostradas por nivel.

i) Actor principal: Usuario

ii) Actor secundario: Sistema.

iii) Precondiciones: El usuario debe hacer el inicio de sesion.

iv) Flujo principal:

(1) El sistema valida el nivel del usuario y muestra las opcciones.
(2) Para el estudiante se mostraran Prestamo y devolucion de herramientas, entrada y salida del laboratorio y mis prestamos.
(3) Para los docentes y encargados prestamo de equipo, mis prestamos e informacion de prestamos y accesos a laboratorio.

Como se muestra en la figura 15, el usuario, al iniciar sesión, permite que el sistema valide su nivel y determine las opciones que se le mostrarán, optimizando así la experiencia de uso.

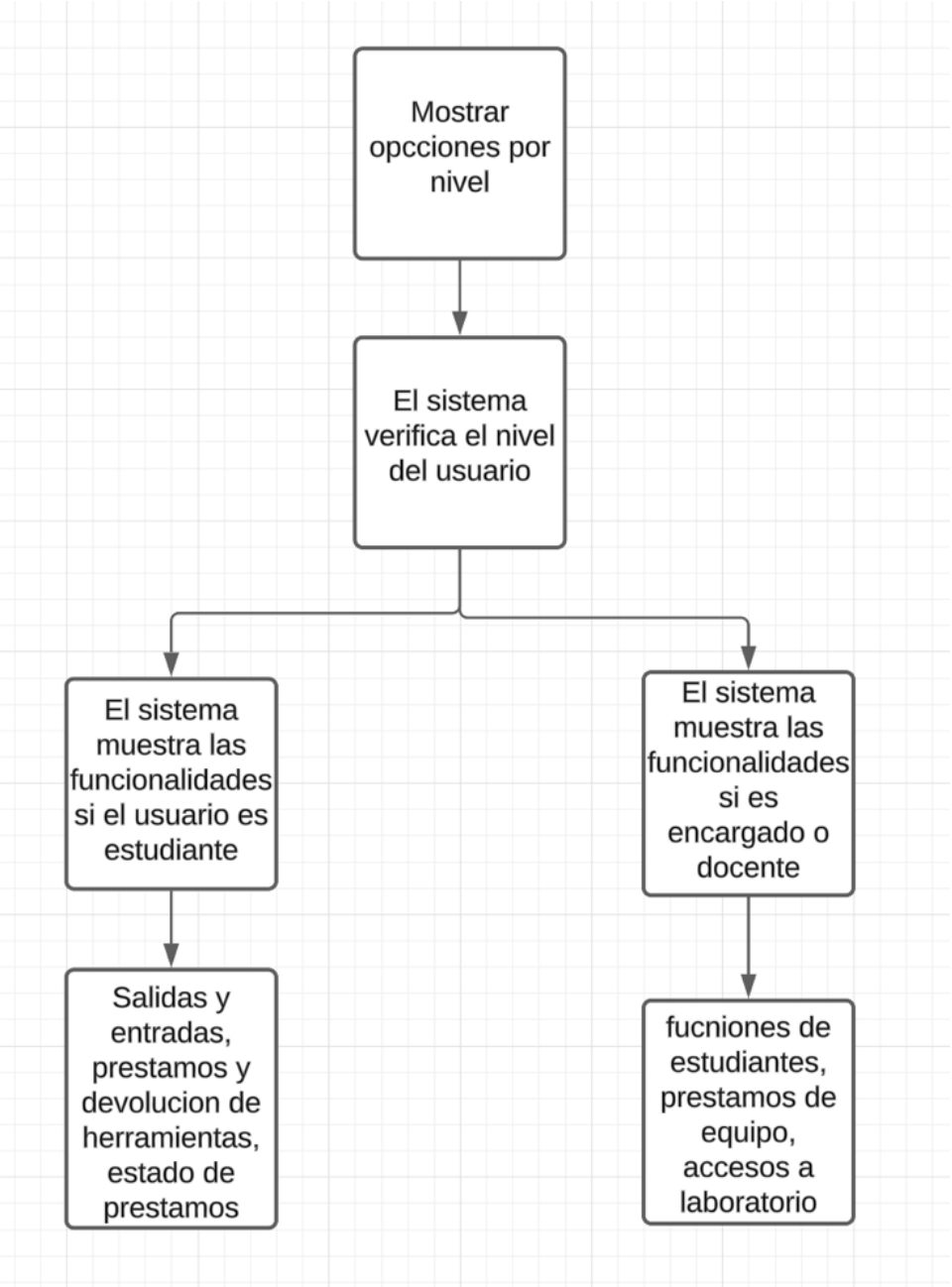

Figura 15. Flujo de opcciones por nivel de atorizacion.

Implementación y Pruebas

La aplicación sera probada en el laboratorio de desarrollo tecnologico, se pondra a prueba contra un sistema analogico donde los ingresos, prestamos, e inventario estan

escritos en papel, se medira el tiempo en el cual un prestamo es realizado y documentado, ademas de la facilidad para transportar esta informacion y que el encargado y el docente sean notificados, ademas de la necesidad de la presencia de uno de estos.

Aplicación

Los modulos que se implementaron en este proyecto son:

Modulo de registro: Permite la creacion de usuarios, mediante un formulario que recopila la informacion y la regista en una colección de usuarios. El sistema tomara el correo y la contraseña del usuario para crear un registro, este nos retornada un UID (Unique ID) que nos servira como su identificador del documento. Cada usuario sera un documento y contendra la informacion basica:

- Nombre
- Matricula
- Nivel

La consola de firebase nos da una imagen como esta de las colecciones:

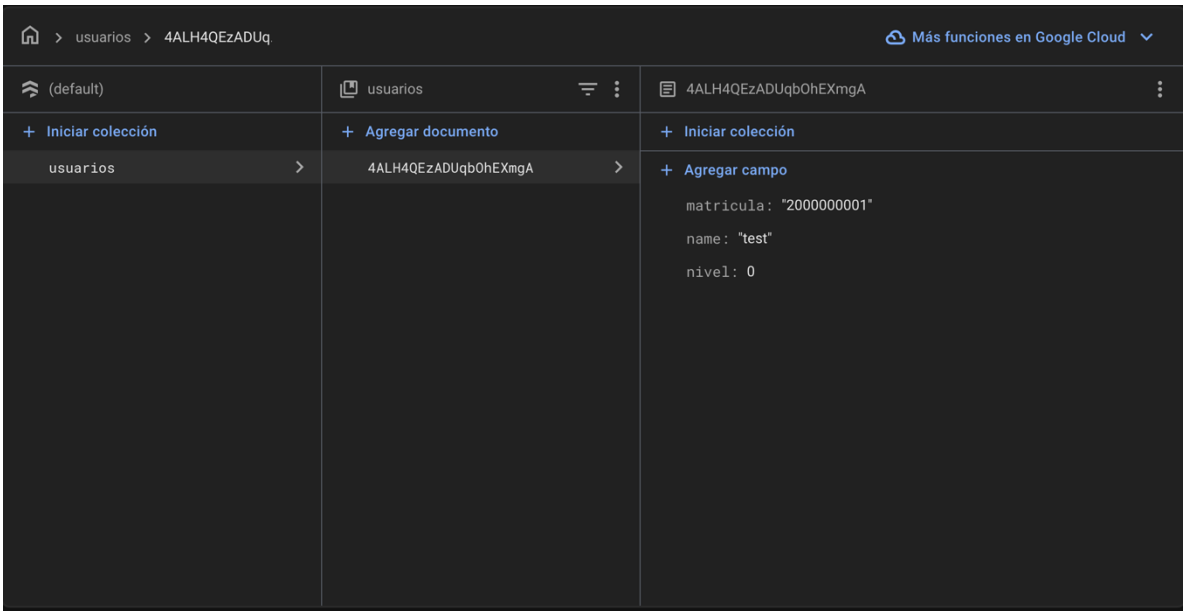

Figura16. Consola de firebase mostrando la coleccion de usuarios.

Nosotros podemos modificar directamente los datos del usuario al tener acceso a la consola.
Cuando este proceso sea concluido satisfactoriamente, nos dirigira a la pantalla de home. En la figura 17 podemos ver la interfaz de usuario, donde solo estan los requerimientos basicos presentados de una forma limpia.
Componentes utilizados:

1) Atomos:
    a) <TextInput />
    b) <Text />
    c) <Button />
2) Moleculas:
    a) <RegisterForm />
    b) <SafeAreaView/>
    c) <LabAlert />
3) Organismos:
    a) <Register />
4) Template
    a) Ninguno
5) Pantalla
    a) <RegisterScreen />

Librerias utilizadas:

1) react-native-firebase/app
2) react-native-firebase/auth
3) **react-native-firebase/firestore**

Utilidades

1) Alert
2) Navegacion

Cobertura de test: 100%

Figura 17. Pantalla de registro

Modulo de autenticacion: Permite al usuario usar sus credenciales para autenticarse, con esto podemos recuperar su UID y podra acceder a las funcionalidades de la

aplicación. Esto lo mandara a la pantalla principal. En la figura 18 podemos ver una pantalla de inicio de sesion limpio y facilmente entendible.

Componentes utilizados:

1) Atomos:
    a) <TextInput />
    b) <Text />
    c) <Button />
2) Moleculas:
    a) <LoginForm />
    b) <SafeAreaView/>
    c) <LabAlert />
3) Organismos:
    a) <Login />
4) Template
    a) Ninguno
5) Pantalla
    a) <LoginScreen />

Librerias utilizadas:

1) react-native-firebase/app
2) eact-native-firebase/auth

Utilidades:

1) Alert
2) Navegacion

Cobertura de test: 100%

Figura 18. Pantalla de inicio de sesion

Modulo principal: Permite al usuario visualizar las opcciones a las que puede acceder, este modulo se vuelve la raiz elminando el stack despues de hacer un registro o inicio de sesion correctos. Asi se evita poder regresar a la pantalla anterior

de login, evitando errores de navegacion. En la figura 19 se observa el menu principal de la aplicación, mostrando las opcciones basicas de esta.

Componentes utilizados:

1) Atomos:
    a) <Text />
    b) <Button />
2) Moleculas:
    a) <SafeAreaView/>
3) Organismos:
    a) <Home />
4) Template
    a) Ninguno
5) Pantalla
    a) <HomeScreen />

Librerias utilizadas:

Ninguna.

Utilidades.

Navegacion

Cobertura de test: 100%

Figura 19. Pantalla principal

Modulo de acceso a laboratorio: Permite al usuario generar un registro de acceso al laboratorio, generando un registro en la base de datos. Finalmente si el registro fue exitoso se mandara una alerta el responsable del laboratorio. En la figura 20 se

observa como se toma captura del qr con ayuda de la camara para permitir el acceso al labortario.

Componentes utilizados:

1) Atomos:
    a. <Text/>
2) Moleculas:
    a. <SafeAreaView/>
    b. <LabAlert />
3) Organismos:
    a. <QRScanner />
4) Template:
    a. <QRScannerTemplate />
5) Pantalla:
    a. <AccessScreen />

Librerias utilizadas:

1) **react-native-qrcode-scanner**
2) **react-native-firebase/messaging**
3) **react-native-firebase/firestore**

Utilidades:

1) **Alert**
2) **Notifications**
3) Navegacion

Cobertura de test: 100%

Figura 20. Pantalla de lectura de qr para acceder

Modulo de salida del laboratorio: Permite al usuaio registrar su salida del laboratorio, actualizando su registro de acceso en la base de datos. En la figura 21 se observa

como se toma captura del qr con ayuda de la camara para registrar la salida del labortario

Finalmente si la actualizacion fue exitosa se mandara una alerta el encargado del laboratorio.

Componentes utilizados:

1) Atomos:
    a. <Text/>
2) Moleculas:
    a. <SafeAreaView/>
3) Organismos:
    a. <QRScanner />
4) Template:
    a. <QRScannerTemplate />
5) Pantalla:
    a. <OutScreen />

Librerias utilizadas:

1) **react-native-qrcode-scanner**
2) **react-native-firebase/messaging**
3) **react-native-firebase/firestore**

Utilidades:

1) **Alert**
2) **Notifications**
3) Navegacion

Cobertura de test: 100%

Figura 21. Pantalla de lectura de qr para salida

Modulo de inventario: Permite al usuairo revisar el estado del inventario, este consta de un input donde puede ingresar una herramienta, el sistema hace una busqueda y este retorna la informacion o avisa si no existe. En la figura 22 podemos ver como se

pide al usuario ingresar los datos de la herramienta a buscar y en la figura 23 el resultado de la busqueda.

Componentes utilizados:

1) Atomos:
    a. <Text/>
    b. <TextInput/>
    c. <Button />
    d. <FlatList />
    e. <Alert />
2) Moleculas:
    a. <SafeAreaView />
    b. <InventorySearchForm />
    c. <UserItem />
    d. <InventoryItem />
    e. <LabAlert />
3) Organismos:
    a. <Inventory />
    b. <UserList />
4) Template:
    a. Ninguno
5) Pantallas:
    a. <InventroyScreen />

Libreias utliizadas:

1) react-native-firebase/firestore

Utilidades:

1) Alert
2) Navegacion

Cobertura de test: 100%

Figura 22. Busqueda de inventario

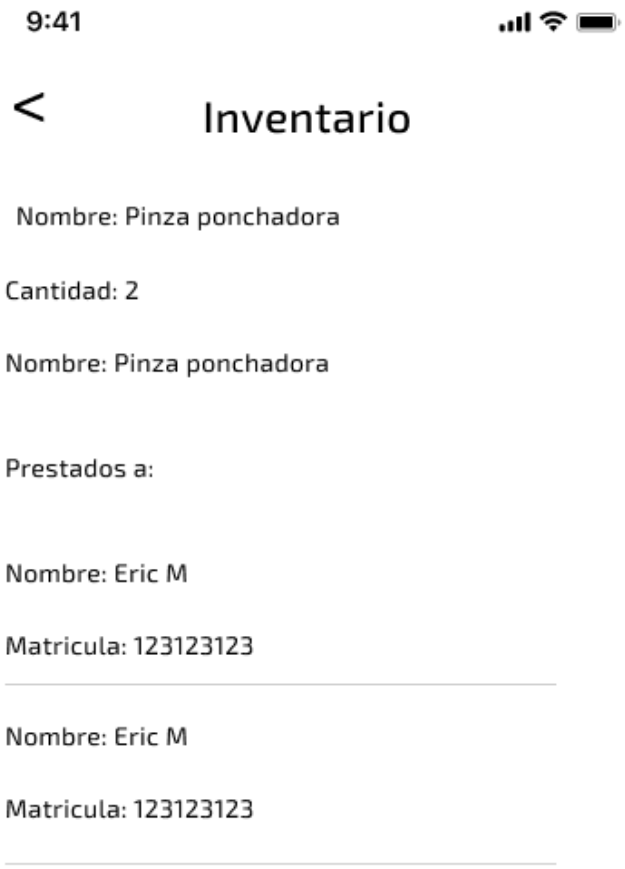

Figura 23. Informacion del inventario

Modulo de modificacion de inventario: Permite al usuario modificar el inventario, su cantidad, nombre o eliminando un registro. En la figura 24 se observa el formulario para que se actualize una herramienta del inventario, para la figura 25 tenemos una

alerta para hacer una doble confirmacion para evitar eliminar una herramienta por error.

Componentes utilizados:

1) Atomos:
    a) <Text />
    b) <TextInput />
    c) <Button />
    d) <Alert />
2) Moleculas:
    a) <SafeAreaView />
    b) <InventroySearchForm />
    c) <InventoryItem />
    d) <LabAlert />
3) Organismos:
    a) <ModifyInventroy />
4) Template:
    a) Ninguno
5) Pantallas:
    a) <ModifyInventoryScreen />

Librerias utilizadas:

1) **react-native-firebase/firestore**

Utilidades:

1) Alert
2) Navegacion

Cobertura de test: 100%

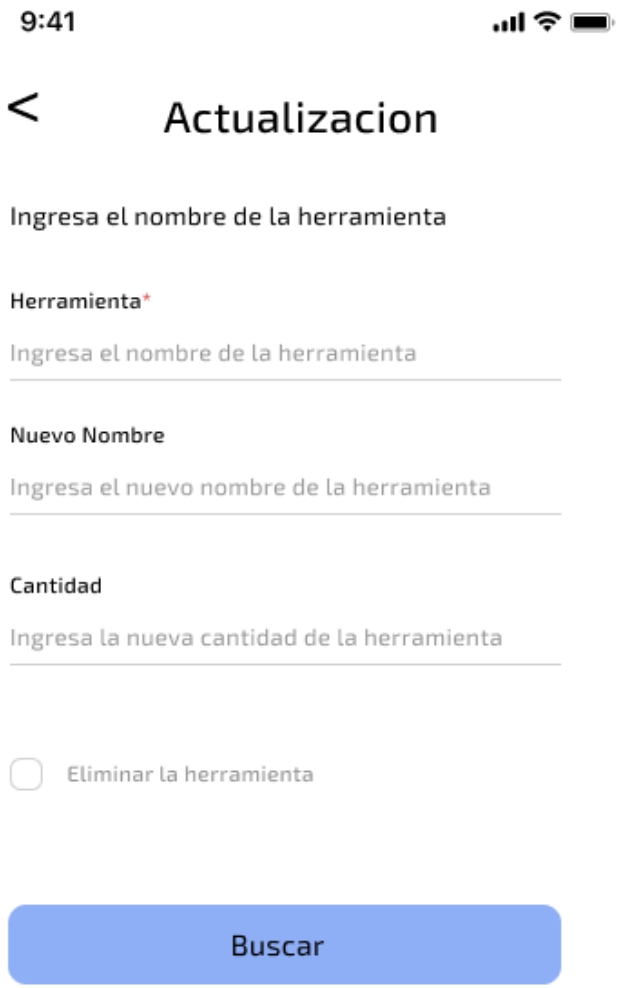

Figura 24. Actualizacion del inventario

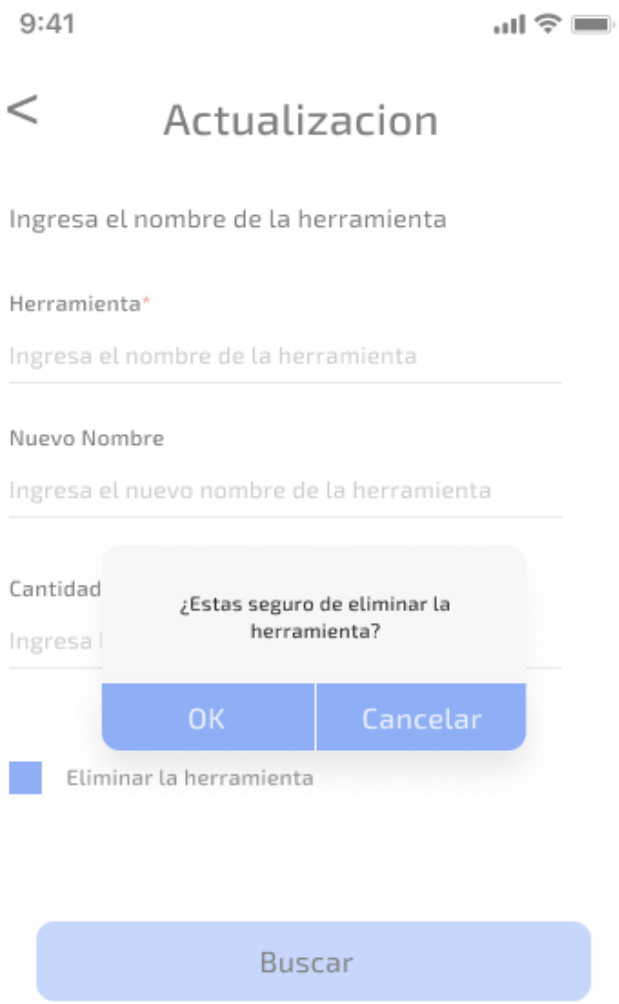

Figura 25. Confirmacion de eliminacion del inventario

Modulo de prestamo de inventario: Permite al usuario realizar un prestamo del inventario. En la figura 26 podemos ver como se captura el qr para el prestamo de

inventario, al igual que las figuras 20, 21 y 27 podemos observar que se reutiliza el template de captura de qr, solo cambiando algunos datos como el nombre.

Componentes utilizados:

1) Atomos:
   a) <Text />
2) Moleculas:
   a) <SafeAreaView />
3) Organismos:
   a) <QRScanner />
4) Template:
   a) <QRScannerTemplate />
5) Pantalla:
   a) <LoanScreen />

Librerias utilizadas:

1) **react-native-qrcode-scanner**
2) **react-native-firebase/messaging**
3) **react-native-firebase/firestore**

Utilidades:

1) **Alert**
2) **Notifications**
3) Navegacion

Cobertura de test: 100%

Figura 26. Lectura de qr para prestamo

Modulo de devolucion de inventario: Permite al usuario devolver la herramienta al inventario. En la figura 27 se ve como se reutiliza el template de QRScannerTemplate, este solo cambia el titulo y la funcion de actualizacion que realiza.

Componentes utilizados:

1) Atomos:
   b) <Text />
2) Moleculas:
   b) <SafeAreaView />
3) Organismos:
   a) <QRScanner />
4) Template:
   a) <QRScannerTemplate />
5) Pantalla:
   a) <ReturnScreen />

Librerias utilizadas:

1) **react-native-qrcode-scanner**
2) **react-native-firebase/messaging**
3) **react-native-firebase/firestore**

Utilidades:

1) **Alert**
2) **Notifications**
3) Navegacion

Cobertura de test: 100%

Figura 27. Lectura de qr para devolucion

Modulo de revision del prestamo: Permite la usuario revisar el estado de su prestamo.

Componentes utilizados:

1) Atomos:
    a. <Text />
2) Moleculas:
    a. <SafeAreaView />
    b. <InventoryItem />
3) Organimos:
    a. <InventoryList />
4) Template:
    a. Ninguno
5) Pantalla:
    a. <UserInventory />

Librerias utilizadas:

1) react-native-firebase/firestore

Utilidades:

1) Alert
2) Navegacion

Cobertura de test: 100%

Modulo de historial de entras y salidas: Permite al usuario revisar el historial de entradas y salidas del laboratorio mediante un rango de tiempo.

En la figura 28 se observa como se escoje el rango de tiempo para hacer la busqueda del hisotrial de entradas y salidas. Y en la figura 29 el resultado de esta.

Componentes utlizados:

1) Atomos:
    a. <Text />
    b. <TextInput />
    c. <Button />
    d. <FlatList />
2) Moleculas:
    a. <SafeAreaView/>
    b. <HistoryRange />
    c. <UserItem />
3) Organismos:
    a. <UserList />
4) Templates:
    a. <HistoryListTemplate />
5) Pantallas:
    a. <UserAccessScreen />

Librerias utilizadas:

1) React-native-firebase/firestore

Utilidades:

1) Navegacion

Cobertura de test: 100%

Figura 28. Seleccion de rango de busqueda para entradas

Figura 29. Registro de entradas

Modulo de historial de prestamos: Permite al usuario revisar el historial de prestamos del laboratorio mediante un rango de tiempo. En la figura 30 se observa como se reutilza el template HistoryListTemplate para ecoger el rango de tiempo para hacer la

busqueda del hisotrial de entradas y salidas. Y en la figura 31 el resultado de esta tambien reutilizando la pantalla para evitar generar codigo duplicado.

Componentes utilzados:

1) Atomos:
    a. <Text />
    b. <TextInput />
    c. <Button />
    d. <FlatList />
2) Moleculas:
    a. <SafeAreaView />
    b. <HistoryRange />
    c. <InventroyItem />
3) Organismos:
    a. <ItemList />
4) Templates:
    a. <HistoryListTemplate />
5) Pantallas:
    a. <InventoryListScrren/>

Librerias utilizadas:

1) React-native-firebase/firestore

Utilidades:

1) Navegacion

Cobertura de test: 100%

Figura 30. Seleccion de rango de tiempo para prestamos

Figura 31. Listado de prestamos

Resultados

Resultados del Proyecto: Pruebas y Testeo de la Aplicación

La aplicación móvil para la gestión de acceso y préstamos en laboratorios universitarios fue sometida a una serie de pruebas rigurosas para garantizar su funcionamiento óptimo y la precisión en la gestión de datos. A continuación se detallan los resultados de estas pruebas.

Registro de Ingreso y Salida

Prueba Realizada:

1. Se realizaron registros de ingreso y salida en múltiples laboratorios utilizando códigos QR.
2. Se verificó la sincronización de estos datos con Firebase en tiempo real.
3. Se realizaron pruebas con diferentes usuarios y en distintos momentos del día para evaluar el comportamiento en condiciones diversas.

Resultado:

1. Precisión del Registro: La aplicación mostró una precisión del 100% en el registro de tiempos de entrada y salida. Todos los datos fueron correctamente registrados en la base de datos de Firebase, y no se observaron discrepancias entre los datos mostrados en la aplicación y los almacenados en Firebase.
2. Sincronización en Tiempo Real: Los registros se sincronizaron en tiempo real, sin retrasos significativos. Las actualizaciones se reflejaron instantáneamente en la interfaz de usuario y en la base de datos.
3. Gestión de Errores: La aplicación manejó correctamente los errores de escaneo y las inconsistencias en los códigos QR. Los mensajes de error fueron claros y

orientaron al usuario para resolver problemas como escaneos fallidos o códigos QR inválidos.

Préstamo de Equipos

Prueba Realizada:

Se realizaron pruebas de préstamo de equipos utilizando códigos QR para registrar y devolver equipos.

Se verificó la disponibilidad de equipos en la base de datos antes y después de cada préstamo.

Se registraron múltiples transacciones de préstamo y devolución para evaluar el rendimiento y la exactitud.

Resultado:

1. Exactitud en la Disponibilidad: La aplicación gestionó correctamente la disponibilidad de los equipos. No se permitieron préstamos cuando no había unidades disponibles, y las actualizaciones en la base de datos fueron precisas.
2. Registro de Préstamos: Todos los registros de préstamo se crearon correctamente en la colección `prestamos` en Firebase, incluyendo los detalles del equipo, la fecha de préstamo, y el ID del alumno. La información fue precisa y coherente.
3. Actualización de Cantidad Prestada: La cantidad de equipos prestados se actualizó adecuadamente en la base de datos. Los informes mostraron que la cantidad de equipos disponibles se ajustó de acuerdo con los préstamos y devoluciones realizados.

Consulta del Historial de Actividad

Prueba Realizada:

1. Se consultaron historiales de actividad para varios alumnos, que incluyeron préstamos, registros de ingreso y salida.
2. Se verificó la precisión del tiempo total acumulado en el laboratorio.
3. Se realizaron pruebas con diferentes perfiles de usuario y niveles de actividad.

Resultado:

1. Visualización del Historial: La aplicación mostró un historial detallado de préstamos, ingresos y salidas de manera clara y organizada. Los datos se presentaron en un formato fácil de leer, facilitando la comprensión del historial del alumno.
2. Cálculo del Tiempo Total: El cálculo del tiempo total que el alumno ha pasado en el laboratorio fue preciso. La aplicación sumó correctamente las diferencias entre las fechas de ingreso y salida, y mostró el tiempo total de forma correcta en formato legible para el usuario.
3. Rendimiento y Escalabilidad: La aplicación mantuvo un buen rendimiento incluso con un gran volumen de datos históricos. La interfaz respondió rápidamente a las consultas, y no se observaron problemas de lentitud o congelamientos.

Pruebas de Rendimiento y Usabilidad

Prueba Realizada:

1. Se evaluó el rendimiento de la aplicación bajo diferentes condiciones de carga, incluyendo múltiples usuarios simultáneos y grandes volúmenes de datos.
2. Se realizaron pruebas de usabilidad con un grupo de usuarios finales para evaluar la facilidad de uso y la navegación.

Resultado:

1. Rendimiento: La aplicación demostró una capacidad de respuesta rápida y estable, incluso bajo condiciones de carga pesada. Los tiempos de respuesta

fueron consistentes y no se encontraron problemas significativos de rendimiento.

2. Usabilidad: Los usuarios finales encontraron la interfaz intuitiva y fácil de usar. La navegación entre diferentes secciones (registro de ingresos, préstamos, consulta de historial) fue fluida, y los usuarios pudieron realizar las tareas deseadas sin dificultades.

Pruebas de Seguridad

Prueba Realizada:

1. Se evaluaron las medidas de seguridad implementadas, incluyendo la autenticación de usuarios y el acceso a los datos en Firebase.
2. Se realizaron pruebas para verificar la protección de datos sensibles y la prevención de accesos no autorizados.

Resultado:

1. Seguridad de Autenticación: La autenticación de usuarios a través de Firebase Authentication funcionó de manera robusta. Solo los usuarios autenticados pudieron acceder a las funcionalidades protegidas de la aplicación.
2. Protección de Datos: Los datos en Firebase se protegieron adecuadamente. Las reglas de seguridad de Firestore fueron configuradas para garantizar que los datos de cada usuario fueran accesibles solo por los usuarios autorizados.
3. Prevención de Accesos No Autorizados: No se encontraron vulnerabilidades significativas en el sistema de seguridad. Las pruebas confirmaron que los datos eran accesibles solo para los usuarios con los permisos adecuados.

Conclusiones

En este estudio se a demostrado el potencial de las tecnologías modernas para mejorar la gestión de laboratorios universitarios. La aplicación móvil desarrollada ha proporcionado una solución eficiente y escalable para la gestión de recursos en entornos académicos, con impactos positivos en la experiencia del usuario y la eficiencia operativa.

A pesar de los avances logrados, quedan inquietudes por abordar, como la escalabilidad de la aplicación para instituciones con diferentes niveles de infraestructura tecnológica y la adaptación a cambios en los procesos y políticas de la institución. Se sugieren futuras investigaciones para explorar estos temas y mejorar aún más la aplicación móvil desarrollada. Como son:

1. Efectividad de la Aplicación Móvil: La implementación de la aplicación móvil ha demostrado ser efectiva en mejorar la eficiencia operativa y la experiencia del usuario en la gestión de laboratorios universitarios. Los usuarios han encontrado la aplicación fácil de usar y han reportado beneficios significativos en términos de acceso a recursos y seguimiento de inventario.

2. Importancia de las Tecnologías Modernas: La utilización de tecnologías modernas como React Native, Compound Components, Jest y TypeScript ha sido fundamental para el éxito del proyecto. Estas tecnologías han permitido desarrollar una aplicación móvil robusta y escalable, con una interfaz de usuario intuitiva y una arquitectura flexible.

3. Respuesta a las Necesidades del Usuario: La aplicación móvil ha abordado eficazmente las necesidades y preocupaciones de los usuarios en la gestión de laboratorios universitarios. La capacidad de reservar espacios y equipos, realizar seguimiento de préstamos y recibir notificaciones ha mejorado la experiencia general de los usuarios y ha optimizado los procesos administrativos.

4. Contribución al Campo de la Ingeniería en Ciencias de la Computación: Este estudio ha contribuido al avance del campo de la ingeniería en ciencias de la computación al demostrar el potencial de las tecnologías móviles para mejorar la gestión de recursos en entornos académicos. La aplicación desarrollada sirve como un ejemplo de cómo la innovación tecnológica puede impulsar la eficiencia y la colaboración en la educación superior.

La investigación ha proporcionado evidencia sólida de los beneficios de utilizar una aplicación móvil basada en tecnologías modernas para la gestión de laboratorios universitarios. La aplicación ha demostrado ser una herramienta valiosa para mejorar la eficiencia operativa y la experiencia del usuario en entornos académicos, y representa un paso adelante en la aplicación práctica de la ingeniería en ciencias de la computación.

Bibliografía

Cogsy. (2024). The importance of inventory accuracy for DTC brands. Recuperado de Inventory Accuracy: How to Calculate & Improve it | Cogsy Logistics | Free Full-Text | Impact of Internet of Things (IoT) on Inventory Management: A Literature Survey (mdpi.com)

Mousavi, Y., Yang, X.-M., Bakhshi, A., & Xu, A. (2022). The impact of IoT on inventory management in supply chains: A comprehensive review12. Journal of Manufacturing and Materials Processing, 6(2), 33. https://doi.org/10.3390/jmmp6020033

Miles, S. B., Sarma, S. E., & Williams, J. R. (Eds.). (2008)1. RFID technology and applications. Cambridge University Press.Compound Pattern (patterns.dev)

Meta. (2024). Re-watch the React Native Keynote @ React Conf 2024. https://reactnative.dev

TypeScript. (2024). TypeScript: JavaScript with syntax for types. Recuperado de TypeScript: JavaScript With Syntax For Types. (typescriptlang.org)

Google. (2024). Kotlin Docs. Kotlin Documentation. Recuperado de Kotlin Docs | Kotlin Documentation (kotlinlang.org)

Apple (2024). Swift. Documentation. Recuperado de Swift.org - Documentation

Freixas Ramírez, Y., & Noa Pérez, V. (2015). Sistema para el control de acceso a los laboratorios del Centro de Tecnologías para la Formación. CUBA.

Acosta, A., & Zambrano, N. (2006)Importancia, problemas y soluciones en el diseño de la interfaz de usuario SABER. Revista Multidisciplinaria del Consejo de Investigación de la Universidad de Oriente, 18(2), 174-182. Disponible en: http://www.redalyc.org/articulo.oa?id=427739430010

Velásquez, S. M., Monsalve Sossa, D. E., Zapata, M. E., Gómez Adasme, M. E., & Ríos, J. P. (2019). Pruebas a aplicaciones móviles: avances y retos. Lámpsakos, (21), 39-50. https://doi.org/10.21501/21454086.2983

Cherny, B. (2019). Programming TypeScript: Making Your JavaScript Applications Scale. O'Reilly Media, Inc.

Wansbrough, L. (n.d.)A QR code scanner component for React Native built on top of react-native-camera1. Retrieved October 10, 2024, from react-native-qrcode-scanner - npm (npmjs.com)

Invertase Limited. (2020)React Native Firebase Documentation1. Recuperado de React Native Firebase | React Native Firebase (rnfirebase.io)

Lazcano Calixto, Ricardo Neftali, Valencia González, Luis Ángel, Baena Díaz, Daniel Esteban y Venegas Guzmán, Ricardo (2019). React Native: acortando las distancias entre desarrollo y diseño móvil multiplataforma.Revista Digital Universitaria (RDU). Vol. 20, núm. 5 septiembre-octubre. DOI:http://doi.org/10.22201/codeic.16076079e.2019.v20n5.a5

Frost, B. (2016). Atomic Design.