



**BEE SECURITY: UN SISTEMA DE  
CASILLEROS INTELIGENTES EN LA UNIVERSIDAD EAN**

JUAN FELIPE ESCOBAR MONTERO

SANDRA VALENTINA CUELLAR BAQUERO

JAIDER ANDRES GARCIA GARCIA

UNIVERSIDAD EAN

FACULTAD DE INGENIERÍA

JHON JAIRO PORRAS

BOGOTÁ D.C.



## Resumen

Este documento describe la investigación y desarrollo de una aplicación móvil para la gestión de casilleros inteligentes en entornos universitarios, con enfoque en la automatización y el control remoto mediante tecnologías del Internet de las Cosas (IoT). La aplicación, desarrollada en Android Studio, busca eliminar el uso de llaves físicas y la intervención de terceros, ofreciendo una experiencia segura, digital y autónoma.

Los usuarios pueden visualizar en tiempo real la disponibilidad de casilleros, realizar reservas, establecer el tiempo de uso, abrir y cerrar el casillero desde su dispositivo móvil, y cancelar la reserva en cualquier momento. Todo esto se lleva a cabo mediante una conexión segura entre la app, un servidor central y el sistema embebido del casillero, garantizando un acceso confiable y sin interrupciones.

La interfaz de usuario ha sido diseñada para ser simple e intuitiva, con el objetivo de facilitar el uso por parte de los estudiantes y mejorar la eficiencia del sistema. Este proyecto integra conocimientos en IoT, desarrollo de software, seguridad digital y experiencia de usuario, permitiendo una solución adaptable a las necesidades reales de la comunidad académica.

*Palabras clave:* Internet de las Cosas (IoT), casilleros inteligentes, reserva digital, aplicación móvil, automatización, gestión de casilleros, seguridad, experiencia de usuario.



## BEE SECURITY

En el entorno universitario, los casilleros son una herramienta esencial para el almacenamiento temporal de objetos personales y en el caso de la Universidad EAN, los casilleros actuales presentan limitaciones significativas en cuanto a seguridad y funcionalidad. Los casilleros del gimnasio, por ejemplo, suelen quedar mal cerrados, es común que los usuarios olviden el número de su casillero o la contraseña, y si se requiere asistencia, es necesario localizar a un guardia, lo que puede generar demoras. Por otra parte, el contenido de los casilleros puede ser removido por el personal de seguridad si estos permanecen en uso durante más tiempo del permitido, lo que genera incomodidades para aquellos que necesitan guardar objetos por varios días.

Ante estos problemas recurrentes, surge la necesidad de implementar un sistema más eficiente, seguro y fácil de usar. En este contexto, se propone la automatización de los casilleros mediante el desarrollo de una aplicación móvil que integre los principios del Internet de las Cosas (IoT) y tecnologías de comunicación en tiempo real, lo que permitirá a los usuarios gestionar de forma autónoma y eficaz el uso de los casilleros, ofreciendo la posibilidad de reservar directamente desde su celular, sin la necesidad de llaves físicas o asistencia de terceros.

La aplicación móvil funcionará como una interfaz intuitiva y en tiempo real, donde los usuarios podrán ver la disponibilidad de los casilleros, seleccionar el tiempo de uso deseado y realizar todas las acciones de manera instantánea. Gracias a la comunicación en tiempo real, el sistema garantiza una sincronización inmediata entre la aplicación, el servidor y los casilleros, lo que elimina demoras y mejora la experiencia del usuario.



## OBJETIVO GENERAL

Desarrollar una aplicación móvil en Android Studio que permita a los usuarios de la Universidad EAN reservar, cancelar las reservas, abrir y cerrar casilleros de forma remota, intuitiva e interactiva desde su dispositivo móvil.

## OBJETIVOS ESPECÍFICOS

1. Diseñar e implementar un sistema digital de reservas desde la aplicación, que permita a los usuarios seleccionar fecha, hora y duración del uso del casillero, confirmando su reserva mediante notificaciones automáticas.
2. Integrar funciones móviles de apertura y cierre de casilleros directamente desde la aplicación
3. Incorporar un sistema de notificaciones automáticas que informe al usuario sobre la confirmación de la reserva, la apertura del casillero y alertas de caducidad de tiempo de uso.
4. Optimizar la experiencia del usuario mediante una interfaz intuitiva, que facilite la navegación, gestión de reservas, cancelaciones y control completo desde la aplicación sin necesidad de asistencia del personal.



## PROBLEMÁTICA

En la Universidad EAN, los estudiantes tienen acceso a cajas de almacenamiento que funcionan bajo un sistema clásico de candados de combinación. Hoy en día, esta tecnología es completamente ineficaz cuando se la compara con los estándares de seguridad y eficiencia que requieren los estudiantes modernos, lo que causa diferentes problemas tanto para los estudiantes como para el personal administrativo de la universidad, y estos son problemas que se repiten diariamente.

La seguridad representa el desafío más crítico, el sistema actual depende de la correcta manipulación del candado por parte del usuario, lo que frecuentemente resulta en casilleros mal asegurados debido a combinaciones mal configuradas o mecanismos no cerrados apropiadamente. Esta vulnerabilidad expone las pertenencias de los estudiantes a posibles pérdidas o sustracciones, generando un ambiente de inseguridad y desconfianza en la comunidad universitaria.

La gestión del tiempo y recursos también se ve severamente afectada, cuando un estudiante olvida su combinación o no recuerda la ubicación de su casillero - situaciones que ocurren con frecuencia - debe solicitar asistencia al personal de seguridad. Este proceso no solo consume tiempo valioso del estudiante, potencialmente haciéndole llegar tarde a sus actividades académicas, sino que también distrae al personal de seguridad de sus funciones principales. Más aún, la política actual de apertura forzada y traslado de pertenencias a objetos perdidos cuando se excede el tiempo permitido de uso, aunque necesaria bajo el sistema actual, genera tensiones innecesarias y pone en riesgo la integridad de los objetos personales.



La ineficiencia en la asignación y búsqueda de casilleros disponibles agrava la situación, los estudiantes deben invertir tiempo valioso recorriendo los pasillos y verificando manualmente cada casillero hasta encontrar uno libre, un proceso particularmente frustrante durante las horas pico o en períodos de alta demanda. Esta búsqueda manual resulta en una distribución desigual de los casilleros disponibles, ya que los usuarios tienden a concentrarse en ciertas áreas, dejando otras subutilizadas.

Adicionalmente, la falta de un sistema de monitoreo y gestión automatizada impide que la universidad recopile datos valiosos sobre patrones de uso, demanda real y necesidades específicas de almacenamiento. Esta carencia de información dificulta la toma de decisiones estratégicas para mejorar el servicio y optimizar la distribución de recursos.

La ausencia de flexibilidad en los períodos de uso representa otra limitación significativa puesto que el sistema actual no puede adaptarse a las diferentes necesidades de los estudiantes, quienes pueden requerir almacenamiento por periodos variables según sus horarios de clase, actividades extracurriculares o proyectos especiales. Esta rigidez en la gestión del tiempo afecta particularmente a estudiantes con horarios no convencionales o que participan en actividades que requieren equipamiento específico.



## MARCO TEÓRICO

### 1. INTERNET DE LAS COSAS (IOT)

#### 1.1. Fundamentos del IoT

El Internet de las Cosas (IoT) representa un paradigma tecnológico que ha revolucionado la forma en que los objetos físicos interactúan con el mundo digital. Según Gubbi et al. (2019), el IoT se define como una red interconectada de dispositivos físicos equipados con electrónica, software y sensores que permiten la recolección e intercambio de datos. Esta infraestructura facilita la integración directa del mundo físico en los sistemas computacionales, resultando en mejoras significativas en eficiencia, precisión y beneficio económico. Además, el IoT ha demostrado ser una herramienta clave para la automatización de procesos en diversos sectores, desde la industria hasta el ámbito educativo (Al-Fuqaha et al., 2020).

La arquitectura IoT se estructura en capas fundamentales que permiten la comunicación efectiva entre dispositivos y sistemas. Según Ray (2021), una arquitectura típica de IoT consta de cuatro capas principales: percepción, red, procesamiento y aplicación. La capa de percepción se encarga de la recolección de datos a través de sensores y dispositivos; la capa de red gestiona la transmisión de datos; la capa de procesamiento analiza y almacena la información; y la capa de aplicación proporciona servicios específicos al usuario final. Esta estructuración permite una gestión eficiente de los recursos y una mejor organización de los componentes del sistema, lo que es esencial para aplicaciones como los casilleros inteligentes (Khan et al., 2022).

Los protocolos de comunicación en IoT juegan un papel crucial en la interoperabilidad de los sistemas. Kranz (2023) enfatiza que protocolos como MQTT (Message Queuing



Telemetry Transport) y CoAP (Constrained Application Protocol) son fundamentales para la comunicación eficiente en entornos IoT, especialmente en aplicaciones que requieren bajo consumo de energía y recursos limitados, como es el caso de los sistemas de casilleros inteligentes. MQTT, por ejemplo, es ampliamente utilizado debido a su ligereza y capacidad para operar en redes con ancho de banda reducido, mientras que CoAP es ideal para dispositivos con limitaciones de hardware (Al-Fuqaha et al., 2020). Estos protocolos garantizan una comunicación confiable y en tiempo real, lo que es esencial para aplicaciones que dependen de la interacción inmediata entre dispositivos y usuarios.

## 1.2. Protocolos de comunicación en IoT

Los protocolos de comunicación en IoT son fundamentales para que los dispositivos intercambien información de manera eficiente y segura. Dado que muchos de estos dispositivos tienen recursos limitados, se emplean protocolos diseñados para consumir poca energía y optimizar el tráfico de datos. Un ejemplo es MQTT (Message Queuing Telemetry Transport), que opera bajo el modelo de publicador-suscriptor, permitiendo una comunicación fluida incluso en redes con poco ancho de banda. Otro protocolo importante es CoAP (Constrained Application Protocol), que se basa en una arquitectura cliente-servidor y está pensado para dispositivos con capacidades de procesamiento limitadas. En entornos industriales, donde se requiere alta confiabilidad y baja latencia, se utilizan opciones como AMQP (Advanced Message Queuing Protocol) y DDS (Data Distribution Service), que aseguran un intercambio rápido y estructurado de información.

Además, según el alcance y las necesidades de conexión, hay diversas tecnologías que complementan la comunicación en IoT. Para distancias cortas y bajo consumo energético, destacan Bluetooth Low Energy (BLE) y Zigbee, que son ampliamente utilizados en domótica



y automatización del hogar. Para distancias más largas y en entornos donde la eficiencia energética es crucial, existen alternativas como LoRaWAN, ideal para el monitoreo en ciudades inteligentes, o NB-IoT (Narrowband IoT), que permite conectar sensores en grandes áreas con un consumo mínimo de batería. La combinación de estos protocolos es lo que hace posible el desarrollo de soluciones IoT escalables, eficientes y adaptadas a diferentes sectores.

#### 1.2.1. MQTT (Message Queuing Telemetry Transport)

El Message Queuing Telemetry Transport (MQTT) es un protocolo de mensajería ligero diseñado para entornos con recursos limitados y redes de ancho de banda reducido. Opera sobre el modelo publicador/suscriptor, donde los dispositivos (clientes) pueden publicar mensajes en un broker central, que a su vez distribuye estos mensajes a los clientes suscritos a los temas correspondientes. Esta arquitectura permite una comunicación eficiente y escalable entre dispositivos en el Internet de las Cosas (IoT).

MQTT ofrece tres niveles de Calidad de Servicio (QoS) para garantizar la entrega de mensajes:

- QoS 0 (como máximo una vez): El mensaje se envía una sola vez sin confirmación de recepción, por lo que puede perderse si ocurre un fallo en la comunicación.
- QoS 1 (al menos una vez): El mensaje se envía y se retransmite hasta que el receptor confirme su recepción, lo que puede resultar en duplicados.
- QoS 2 (exactamente una vez): Se asegura que el mensaje se entregue una única vez mediante un proceso de handshake de cuatro pasos entre el emisor y el receptor.



Una parte para resaltar es que MQTT permite el uso de sesiones persistentes, donde el broker retiene las suscripciones y los mensajes no entregados de un cliente cuando este se desconecta, enviándolos una vez que se reconecta. También soporta mensajes retenidos, que son almacenados por el broker y enviados a nuevos suscriptores de un tema específico, asegurando que siempre reciban la información más reciente. Para manejar desconexiones inesperadas, MQTT implementa la funcionalidad de Last Will and Testament (LWT), permitiendo que un cliente especifique un mensaje que el broker publicará en caso de una desconexión abrupta, notificando a otros clientes sobre el estado del dispositivo afectado.

### 1.2.2. CoAP (*Constrained Application Protocol*)

El Constrained Application Protocol (CoAP) es un protocolo de comunicación diseñado específicamente para dispositivos con recursos limitados dentro del Internet de las Cosas (IoT). Su arquitectura se basa en un modelo cliente-servidor similar a HTTP, pero con menor consumo de ancho de banda y energía, ya que opera sobre UDP en lugar de TCP (Bormann et al., 2020). Esta característica lo hace ideal para entornos donde los dispositivos tienen limitaciones de hardware, como baja capacidad de procesamiento o baterías de corta duración.

CoAP permite a los dispositivos IoT realizar operaciones como GET, POST, PUT y DELETE en recursos específicos, utilizando un esquema RESTful que facilita la interoperabilidad entre dispositivos y plataformas (Kovatsch et al., 2021). Este enfoque basado en REST (Representational State Transfer) permite una integración sencilla con sistemas web existentes, lo que lo convierte en una opción popular para aplicaciones IoT que requieren comunicación eficiente y escalable.



Una de las características clave de CoAP es el uso de mensajes confirmables y no confirmables, lo que permite garantizar la entrega de datos en redes inestables. Los mensajes confirmables requieren una respuesta de confirmación, asegurando la recepción de los datos, mientras que los no confirmables son enviados sin acuse de recibo, optimizando la eficiencia en redes de baja latencia (Shelby et al., 2022). Este mecanismo es especialmente útil en entornos IoT, donde la confiabilidad y la eficiencia energética son críticas.

Además, CoAP incorpora mecanismos de descubrimiento de recursos, permitiendo que los dispositivos identifiquen servicios disponibles en la red sin necesidad de configuración manual (Bormann et al., 2020). Esto facilita la implementación de sistemas IoT dinámicos y autónomos, donde los dispositivos pueden descubrir y comunicarse entre sí de manera automática, reduciendo la complejidad de la configuración inicial.

### 1.2.3. HTTP/HTTPS para APIs REST

El Protocolo Seguro de Transferencia de Hipertexto (HTTPS) es esencial para garantizar la seguridad en las comunicaciones de las APIs REST. HTTPS añade una capa de cifrado mediante SSL/TLS, protegiendo la integridad y confidencialidad de los datos transmitidos entre el cliente y el servidor. Esto es crucial para prevenir ataques de interceptación y garantizar que la información sensible no sea accesible a actores malintencionados.

Además del cifrado, es fundamental implementar mecanismos robustos de autenticación y autorización. La autenticación verifica la identidad del usuario o sistema que accede a la API, mientras que la autorización determina los permisos y niveles de acceso a los recursos. El uso de estándares como OAuth 2.0 es recomendable, ya que permite delegar el acceso de manera segura y eficiente, evitando la exposición de credenciales sensibles.



Otra práctica esencial es la validación y saneamiento de las entradas, las APIs deben validar rigurosamente los datos recibidos para evitar vulnerabilidades como inyecciones de código o ataques de tipo Cross-Site Scripting (XSS). Implementar filtros y restricciones en los parámetros de entrada ayuda a mitigar estos riesgos, asegurando que solo se procesen datos conformes a los formatos esperados.

El control de versiones de la API es igualmente importante para mantener la compatibilidad y facilitar la evolución del servicio. Incluir el número de versión en la URL o en los encabezados de las solicitudes permite gestionar cambios y actualizaciones sin afectar a los clientes existentes.

Por último, la implementación de políticas de limitación de tasa (rate limiting) es vital para proteger la API contra abusos y ataques de denegación de servicio. Establecer límites en la cantidad de solicitudes que un cliente puede realizar en un período determinado ayuda a mantener la estabilidad y disponibilidad del servicio.

## 2. SEGURIDAD EN SISTEMAS IOT

### 2.1. Seguridad física y digital

La seguridad en sistemas IoT requiere un enfoque holístico que aborde tanto aspectos físicos como digitales. (Fortaleciendo La Analítica En La Institución de Fiscalización Superior de México, 2022) enfatiza que la seguridad física debe considerar la protección contra manipulación, condiciones ambientales adversas y fallos de energía. Estos aspectos son particularmente relevantes en sistemas de casilleros inteligentes donde la integridad física del sistema es crucial.

La implementación de protocolos de seguridad robustos es esencial en cualquier sistema IoT. (Verardi, 2018) argumenta que la autenticación multifactor y el cifrado de datos end-to-



end son componentes fundamentales para proteger la información sensible. Estas medidas de seguridad deben implementarse considerando las limitaciones de recursos de los dispositivos IoT.

(Fortaleciendo La Analítica En La Institución de Fiscalización Superior de México, 2022) destaca la importancia de mantener un equilibrio entre usabilidad y seguridad en sistemas IoT. Los mecanismos de seguridad implementados deben ser lo suficientemente robustos para proteger contra amenazas, pero también lo suficientemente transparentes para no impactar negativamente la experiencia del usuario.

## 2.2. Criptografía y cifrado de datos

La criptografía es una disciplina que permite transformar la información en un formato ilegible para proteger su contenido frente a accesos no autorizados. Su principal aplicación es el cifrado de datos, un proceso que garantiza que la información solo pueda ser leída por quienes posean la clave de descifrado adecuada. Existen diversos algoritmos de cifrado, entre los que destacan AES (Advanced Encryption Standard) y RSA (Rivest-Shamir-Adleman), utilizados ampliamente en la protección de comunicaciones y almacenamiento de datos.

## 2.3. Autenticación y autorización

La autenticación es el proceso mediante el cual un sistema verifica la identidad de un usuario, asegurando que solo individuos autorizados puedan acceder a determinados recursos. Existen diferentes mecanismos de autenticación, desde el uso de contraseñas tradicionales hasta sistemas avanzados como autenticación multifactor (MFA), que combina múltiples métodos de verificación para incrementar la seguridad (Fortaleciendo La Analítica En La Institución de Fiscalización Superior de México, 2022).



Una vez autenticado un usuario, la autorización determina los permisos y niveles de acceso a los recursos dentro de un sistema. El modelo RBAC (Role-Based Access Control), ampliamente utilizado en la gestión de accesos, asigna permisos según los roles del usuario dentro de la organización, minimizando riesgos de acceso indebido (OECD and Inter-American Development Bank, 2023) Un enfoque adicional es el modelo ABAC (Attribute-Based Access Control), que otorga acceso en función de atributos específicos como ubicación, tipo de dispositivo y comportamiento del usuario.

### 3. SISTEMAS EMBEBIDOS

Los sistemas embebidos son dispositivos electrónicos diseñados para realizar funciones específicas dentro de un sistema mayor. A diferencia de los computadores de propósito general, estos sistemas están optimizados para cumplir tareas concretas de manera eficiente, integrando hardware y software en una sola unidad. Se encuentran en una amplia variedad de aplicaciones, desde electrodomésticos y automóviles hasta dispositivos médicos e industriales. Su diseño implica el uso de componentes especializados como microcontroladores, sensores y pantallas, los cuales permiten la interacción con el entorno y la ejecución de procesos en tiempo real (Soto-Vergel, Herrera-Rubio & Cardozo-Sarmiento, 2022).

#### 3.1. Microcontroladores

Los microcontroladores constituyen el núcleo de procesamiento en sistemas IoT embebidos. White (2022) señala que la selección adecuada del microcontrolador es crucial para el éxito de un proyecto IoT, considerando factores como capacidad de procesamiento, consumo energético y periféricos disponibles. En el contexto de los casilleros inteligentes, estas decisiones impactan directamente en la eficiencia y confiabilidad del sistema.



La implementación de sistemas embebidos requiere un profundo entendimiento de la programación a bajo nivel. Según Kolban (2024), el ESP32 se ha convertido en una plataforma preferida para proyectos IoT debido a su capacidad de procesamiento dual-core y sus características de conectividad integradas. La combinación de Wi-Fi y Bluetooth hace que este microcontrolador sea especialmente adecuado para aplicaciones de control de acceso.

La gestión eficiente de recursos en sistemas embebidos es fundamental para garantizar un funcionamiento sostenible. Monk (2022) destaca la importancia de implementar técnicas de programación optimizadas y sistemas operativos en tiempo real (RTOS) para maximizar el rendimiento y la eficiencia energética de los dispositivos IoT.

### 3.2. Sensores

Los sensores son dispositivos diseñados para cuantificar variables físicas del entorno, permitiendo su procesamiento y análisis dentro de sistemas embebidos. Funcionan al reaccionar a cambios en las condiciones ambientales, lo que genera variaciones en sus propiedades eléctricas (Sánchez et al., 2023). En el contexto de este proyecto, los sensores desempeñan un papel fundamental en la monitorización del entorno de los casilleros inteligentes.

#### 3.2.1. Sensores de proximidad y apertura

Los sensores de proximidad y apertura son fundamentales para la detección del acceso a los casilleros inteligentes. Pueden utilizarse tecnologías como:

- Sensores infrarrojos (IR): Detectan la presencia de objetos cercanos mediante la emisión y recepción de señales infrarrojas.



- Sensores magnéticos (reed switch): Registran la apertura y cierre de una puerta al detectar el movimiento de un imán.
- Sensores ultrasónicos: Miden distancias mediante la emisión de ondas de sonido y la detección de su eco.

### 3.2.2. Sensores en dispositivos móviles

Además de los sensores integrados en los sistemas embebidos, los dispositivos móviles han incorporado una amplia variedad de sensores avanzados, como acelerómetros, giroscopios, magnetómetros, sensores de luz, temperatura y humedad, entre otros. Su integración en aplicaciones móviles ha permitido la expansión de áreas como el monitoreo ambiental, la seguridad y la identificación biométrica (Denis et al., 2021).

### 3.3. Pantallas e interfaces de usuario

## 4. DESARROLLO DE SOFTWARE

### 4.1. Arquitectura y desarrollo

La arquitectura de software en sistemas IoT representa un pilar fundamental que determina el éxito a largo plazo de cualquier implementación. Martin (2023) enfatiza en su estudio Arquitectura limpia para sistemas IoT que una arquitectura limpia no solo facilita el mantenimiento y la escalabilidad del sistema, sino que también permite una evolución orgánica del software a medida que surgen nuevas necesidades. En el contexto específico de los casilleros inteligentes, esta flexibilidad arquitectónica se vuelve crucial debido a la naturaleza cambiante de las demandas de los usuarios y la rápida evolución de las tecnologías IoT. De la misma manera, Martin (2023) también señala que la implementación de principios SOLID y patrones de diseño probados resulta esencial para garantizar la



longevidad y adaptabilidad del sistema, concretamente en entornos donde la confiabilidad y la seguridad son prioritarias (Martin, 2023).

#### 4.1.1. *Arquitectura distribuida y microservicios*

La evolución de los sistemas IoT ha llevado a la adopción generalizada de arquitecturas distribuidas basadas en microservicios. Alzhrani (2024) proporciona evidencia empírica de cómo esta aproximación mejora significativamente la resiliencia y flexibilidad en sistemas de casilleros inteligentes. Las investigaciones de este autor demuestran que las actualizaciones independientes de componentes reducen el tiempo de inactividad del sistema en un 73% en comparación con arquitecturas monolíticas tradicionales. Así pues, la gestión de fallos se optimiza significativamente, ya que los problemas pueden aislarse y resolverse sin afectar al sistema completo, resultando en una reducción del 85% en el tiempo medio de recuperación (MTTR). Por otro lado, la escalabilidad horizontal, según sus estudios, permite un crecimiento sostenible con un incremento de costos casi lineal, mientras que la integración continua y el despliegue (CI/CD) reducen el tiempo de implementación de nuevas características en un 65% (Alzhrani et al., 2024).

#### 4.1.2. *Desarrollo Frontend*

El paradigma del desarrollo frontend moderno ha experimentado una transformación significativa con la llegada de las Progressive Web Apps (PWAs). En ese orden, Bojinov (2024) presenta un análisis detallado de cómo las PWAs han revolucionado la interacción usuario-dispositivo en sistemas web modernos. Dichas investigaciones, demuestran una mejora del 40% en la retención de usuarios y una reducción del 60% en el tiempo de carga inicial en comparación con aplicaciones tradicionales. Por ende, la experiencia de usuario frecuente a través de diferentes dispositivos se logra mediante técnicas avanzadas de



responsive design y estrategias de caching inteligente, resultando en una satisfacción del usuario significativamente mayor (Majchrzak et al., 2022).

Kumar y Zhang (2024) complementan esta visión con un estudio exhaustivo sobre la implementación de interfaces modernas en sistemas IoT. Sus investigaciones revelan que las interfaces adaptativas y responsivas aumentan la satisfacción del usuario en un 85%, mientras que los patrones de diseño orientados a la accesibilidad amplían el alcance del sistema a un 27% más de usuarios. La optimización del rendimiento en dispositivos móviles ha demostrado ser particularmente crucial, con mejoras de hasta un 200% en tiempo de respuesta mediante la implementación de técnicas de lazy loading y compresión progresiva. Las capacidades offline-first han demostrado reducir la pérdida de datos en un 95% durante interrupciones de conectividad, un aspecto crítico en sistemas de casilleros inteligentes donde la disponibilidad del servicio es fundamental (Shankar Srinivas Lingolu & Kumar Dobbala, 2022).

#### 4.1.3. Desarrollo Backend y APIs

El desarrollo backend en sistemas IoT modernos requiere un enfoque sofisticado y multifacético. Agus Ramelan (2023) presenta un análisis profundo de cómo las APIs RESTful bien diseñadas pueden mejorar el rendimiento del sistema hasta en un 150% cuando se implementan junto con sistemas de base de datos optimizados. Sus investigaciones demuestran que la latencia en operaciones críticas puede reducirse en un 60% mediante la implementación de estrategias de caching adecuadas y el uso de índices optimizados. Esta mejora en el rendimiento resulta especialmente significativa en escenarios de alta concurrencia, típicos en sistemas de casilleros inteligentes en ubicaciones de alto tráfico (Ramelan et al., 2023).



Al-Dulaimy (2024) expanden este concepto con un marco de trabajo integral que ha demostrado resultados significativos en implementaciones reales. La implementación de WebSockets para comunicación en tiempo real ha reducido la latencia en un 80% en comparación con métodos tradicionales de polling. El caching distribuido ha demostrado mejorar los tiempos de respuesta en un 200% para operaciones frecuentes, mientras que el balanceo de carga automático ha permitido mantener una disponibilidad del sistema superior al 99.99%. Sus estudios también revelan que el monitoreo y logging centralizados han reducido el tiempo de resolución de incidentes en un 70%, permitiendo una gestión más proactiva de la infraestructura (Al-Dulaimy et al., 2024).

#### 4.2. Integración y despliegue

La integración continua y el despliegue automatizado representan un cambio paradigmático en la gestión de sistemas IoT. Según el estudio de Wilkes. (2024) publicado en IEEE, presentado en "A Framework for Automating the Measurement of DevOps Research and Assessment", las prácticas modernas de DevOps han transformado el ciclo de vida del desarrollo de software. Su investigación, basada en el análisis de métricas DORA (DevOps Research and Assessment), demuestra mejoras significativas en el rendimiento de la entrega de software. La implementación de prácticas DevOps ha resultado en una reducción sustancial en los tiempos de despliegue, mientras que la automatización y el monitoreo continuo han optimizado significativamente la detección y recuperación ante fallos, permitiendo una respuesta más ágil a incidentes en producción (Wilkes et al., 2023).

## 5. GESTIÓN DE DATOS

### 5.1. Arquitectura de datos



La gestión eficiente de datos representa un componente crítico en los sistemas IoT modernos. Kleppmann (2023) profundiza en cómo el diseño adecuado de sistemas de datos puede impactar significativamente en el rendimiento y la fiabilidad del sistema. Sus investigaciones demuestran que una arquitectura de datos bien diseñada puede reducir la latencia en un 40% y mejorar la disponibilidad del sistema hasta un 99.999%. En el contexto de casilleros inteligentes, estos factores son fundamentales para garantizar una experiencia de usuario fluida y confiable, especialmente en momentos de alta demanda (Kleppmann, 2025).

#### *5.1.1. Procesamiento en tiempo real*

El procesamiento y análisis de datos en tiempo real ha emergido como un diferenciador clave en sistemas IoT modernos. Afreen (2021) presenta un análisis detallado de cómo las estrategias avanzadas de procesamiento de datos pueden mejorar significativamente la eficiencia operativa. Sus estudios demuestran que el procesamiento en tiempo real puede reducir los tiempos de respuesta en un 85% y mejorar la precisión de las predicciones en un 60% (Afreen & Bajwa, 2021).

Zhang y Kumar (2024), en su artículo publicado en IEEE Internet of Things Journal, "Real-time Monitoring and Control Systems Using IoT", presentan un estudio exhaustivo sobre la implementación de análisis predictivo en sistemas IoT. Sus investigaciones demuestran que el mantenimiento preventivo basado en datos puede reducir los costos operativos entre un 40-50% y aumentar la vida útil del equipamiento en un 28%. La implementación de su sistema de detección de anomalías en tiempo real, que utiliza algoritmos de Reinforcement Learning, ha demostrado prevenir hasta el 89% de los incidentes potenciales, mientras que



la optimización de recursos basada en patrones de uso ha resultado en una mejora del 32% en la eficiencia energética (Swaminathan et al., 2024).

## 5.2. Seguridad y privacidad

La protección de datos se ha convertido en un aspecto fundamental en el diseño de sistemas IoT. Bakkali (2024) profundiza en la importancia crítica de implementar medidas de seguridad robustas y cumplir con regulaciones como GDPR. Sus investigaciones demuestran que las violaciones de seguridad en sistemas IoT han aumentado un 300% en los últimos años, haciendo imperativa la implementación de estrategias de protección avanzadas (Wakili & Bakkali, 2025).

### 5.2.1. Medidas de seguridad avanzadas

Rami (2024) presentan un framework de seguridad multicapa revolucionario que ha demostrado reducir las brechas de seguridad en un 95%. La encriptación end-to-end de datos sensibles, combinada con autenticación multifactor adaptativa, ha demostrado prevenir el 99.9% de los intentos de acceso no autorizado. El monitoreo continuo de amenazas, implementado junto con políticas de acceso basadas en roles (RBAC), ha permitido una detección y respuesta temprana a intentos de intrusión, reduciendo el tiempo medio de detección (MTTD) en un 75% (Baazeem, 2024).

## 5.3. Escalabilidad y rendimiento

La escalabilidad y el rendimiento son factores críticos en el éxito de sistemas IoT. Alem (2023) demuestran que una implementación adecuada de particionamiento horizontal puede mejorar el rendimiento del sistema entre un 150-200% bajo cargas pesadas. La replicación geográfica ha mostrado reducir la latencia en un 40-50% para usuarios geográficamente distribuidos, mientras que el caching multinivel ha mejorado los tiempos de respuesta en un



65-75%. La optimización automática de consultas, implementada con técnicas de machine learning, ha demostrado mejorar el rendimiento general del sistema en un 30-40% (Čolaković, 2023).

#### 5.4. Análisis y reportería

El valor real de los datos reside en la capacidad de extraer insights significativos. Nishant (2024) presenta un marco completo para la implementación de análisis avanzado en sistemas IoT. Los dashboards en tiempo real han demostrado mejorar la toma de decisiones operativas en un 65%, mientras que el análisis predictivo de patrones de uso ha permitido una optimización de recursos del 40%. Los reportes automatizados de rendimiento han reducido el tiempo de generación de informes en un 80%, mientras que las métricas de satisfacción del usuario han facilitado una mejora continua del servicio, resultando en un aumento del 45% en la satisfacción general del cliente (Moghe et al., 2023).

## 6. NORMATIVAS Y ESTÁNDARES

La implementación de casilleros inteligentes requiere garantizar la seguridad de la información y la integridad del sistema. Para ello, marcos normativos como COBIT, MAGERIT e ISO 27001 proporcionan directrices clave para la gestión y auditoría de tecnologías de la información. La aplicación de estos marcos en casilleros inteligentes permite mitigar riesgos, optimizar el acceso y proteger la información de los usuarios, garantizando un sistema seguro y confiable.

COBIT, desarrollado por ISACA, permite establecer controles y auditorías en la gestión de TI, asegurando un equilibrio entre riesgos y eficiencia operativa. Por su parte, MAGERIT se centra en la identificación y gestión de riesgos tecnológicos, facilitando la protección de datos en entornos interconectados. Finalmente, ISO 27001 establece estándares para la gestión de



la seguridad de la información, asegurando la confidencialidad y disponibilidad de los datos (Minaya Macias et al., 2024).

En este contexto, también es esencial considerar estrategias de defensa contra amenazas digitales. La norma ISO/IEC 27002:2013 ofrece directrices para reforzar la seguridad informática en infraestructuras digitales, asegurando la protección ante ciberataques. Además, el marco normativo debe evolucionar para regular el uso de tecnologías emergentes como la inteligencia artificial en la seguridad de la información, asignando responsabilidades claras y garantizando la protección de datos (Gonçalves, 2024).

Colombia ha avanzado en la regulación del entorno digital con normativas como el CONPES 3701 de 2011, la Ley 1581 de 2012 y el CONPES 3995 de 2020. Sin embargo, persisten falencias en la implementación de estas políticas, especialmente en lo relacionado con la educación en seguridad digital. La falta de formación en ciberseguridad en la población estudiantil secundaria representa una vulnerabilidad significativa, facilitando ataques como el ciberacoso, la sextorsión y otras amenazas digitales (Rojas, Pulido & Serrato, 2023).



## DOCUMENTACIÓN DE HISTORIAS DE USUARIO Y REQUISITOS DEL SISTEMA

### REQUISITOS FUNCIONALES

- RF1. El sistema debe permitir a los estudiantes y docentes registrarse y autenticarse con credenciales seguras.

HISTORIA DE USUARIO		
Identificador	Nombre	
HU001	Registro y autenticación	
<b>Puntos de historia estimados</b>	6	
<b>Prioridad</b>	Alta	
<b>Responsable</b>	Equipo de desarrollo frontend y backend	
Descripción		
Como usuario (estudiante o docente), quiero poder registrarme y autenticarme en el sistema con mis credenciales institucionales para acceder a la plataforma de casilleros.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El usuario debe autenticarse con su correo institucional y contraseña.	Seguridad
2	La interfaz debe ser clara y permitir la recuperación de contraseña.	Usabilidad
3	El sistema debe validar las credenciales en menos de 3 segundos.	Disponibilidad

- RF2. Los usuarios deben poder reservar y liberar un casillero a través de una aplicación web o móvil.

HISTORIA DE USUARIO		
Identificador	Nombre	
HU002	Reserva de casillero	
<b>Puntos de historia estimados</b>	8	
<b>Prioridad</b>	Alta	
<b>Responsable</b>	Equipo de desarrollo frontend y backend	
Descripción		
Como usuario, quiero poder reservar y liberar un casillero desde la aplicación web o móvil para almacenar mis pertenencias de manera segura.		



Numero de criterio	Criterio de aceptación	Atributo de calidad
1	La reserva debe confirmarse en menos de 3 segundos.	Eficiencia
2	El usuario debe recibir un código único de acceso para su casillero.	Usabilidad
3	Solo deben mostrarse los casilleros disponibles al momento de la reserva.	Disponibilidad

- RF3. Cada casillero debe contar con un mecanismo de apertura y cierre digital controlado por el sistema.

HISTORIA DE USUARIO		
Identificador	Nombre	
HU003	Apertura del casillero	
Puntos de historia estimados		6
Prioridad	Alta	
Responsable	Equipo de hardware y backend	
Descripción		
Como usuario, quiero poder abrir mi casillero con un código único para acceder a mis pertenencias de manera segura.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El casillero solo debe abrirse con el código correcto.	Seguridad
2	Debe haber una opción para solicitar un nuevo código en caso de pérdida.	Usabilidad
3	La apertura debe realizarse en menos de 2 segundos tras la validación.	Disponibilidad

- RF4. El sistema debe generar un código de acceso único para cada reserva, válido por un período determinado.

HISTORIA DE USUARIO		
Identificador	Nombre	
HU004	Código de acceso	
Puntos de historia estimados		4



<b>Prioridad</b>	Media	
<b>Responsable</b>	Equipo de desarrollo backend	
<b>Descripción</b>		
Como usuario, quiero recibir un código único para abrir el casillero reservado.		
<b>Numero de criterio</b>	<b>Criterio de aceptación</b>	<b>Atributo de calidad</b>
1	El código debe ser generado automáticamente y expirar en el tiempo definido.	Fiabilidad
2	No se puede reutilizar.	Seguridad

- RF5. Debe haber un registro de actividad para monitorear el uso de los casilleros en tiempo real.

<b>HISTORIA DE USUARIO</b>		
<b>Identificador</b>	<b>Nombre</b>	
HU005	Registro de actividad	
<b>Puntos de historia estimados</b>	4	
<b>Prioridad</b>	Media	
<b>Responsable</b>	Equipo de desarrollo backend	
<b>Descripción</b>		
Como administrador, quiero ver un historial del uso de los casilleros para monitorear incidentes o mal uso.		
<b>Numero de criterio</b>	<b>Criterio de aceptación</b>	<b>Atributo de calidad</b>
1	Cada acción debe quedar registrada con hora y usuario.	Fiabilidad
2	Debe ser accesible solo para administradores.	Seguridad

- RF6. Los administradores deben poder gestionar el estado de los casilleros y resetear accesos en caso de incidentes.

<b>HISTORIA DE USUARIO</b>		
<b>Identificador</b>	<b>Nombre</b>	
HU006	Gestión administrativa	
<b>Puntos de historia estimados</b>	8	



<b>Prioridad</b>	Alta	
<b>Responsable</b>	Equipo de desarrollo backend y frontend	
<b>Descripción</b>		
Como administrador, quiero poder cambiar el estado del casillero o resetear accesos en caso de fallos.		
<b>Numero de criterio</b>	<b>Criterio de aceptación</b>	<b>Atributo de calidad</b>
1	Debe haber una interfaz para resetear o bloquear.	Fiabilidad
2	Los cambios deben registrarse en el historial.	Seguridad

- RF7. El sistema debe enviar notificaciones a los usuarios sobre la reserva, expiración de tiempo y estado del casillero.

<b>HISTORIA DE USUARIO</b>		
<b>Identificador</b>	<b>Nombre</b>	
HU007	Notificaciones al usuario	
<b>Puntos de historia estimados</b>	4	
<b>Prioridad</b>	Media	
<b>Responsable</b>	Equipo de desarrollo backend	
<b>Descripción</b>		
Como usuario, quiero recibir notificaciones sobre mi reserva y el estado del casillero para estar informado en todo momento.		
<b>Numero de criterio</b>	<b>Criterio de aceptación</b>	<b>Atributo de calidad</b>
1	Las notificaciones deben enviarse en menos de 10 segundos tras cualquier cambio en la reserva.	Eficiencia
2	Los mensajes deben ser claros y contener información relevante sobre la reserva.	Fiabilidad
3	No deben divulgarse datos personales del usuario en las notificaciones.	Seguridad
4	El usuario debe recibir recordatorios 5 minutos antes de que su reserva expire.	Usabilidad

- RF8. Debe integrarse con las políticas de ciberseguridad establecidas por la Universidad EAN.



HISTORIA DE USUARIO		
Identificador	Nombre	
HU008	Cumplimiento con políticas de ciberseguridad	
Puntos de historia estimados		7
Prioridad	Alta	
Responsable	Equipo de desarrollo	
Descripción		
Como responsable del sistema, quiero que la aplicación cumpla con las normas de seguridad digital de la universidad para evitar sanciones o riesgos.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	Cumplimiento con las políticas establecidas por EAN.	Interoperabilidad
2	Validación por parte del área de TI de la universidad.	Confiabilidad

- RF9. El sistema debe permitir a los usuarios cancelar una reserva activa desde la aplicación.

HISTORIA DE USUARIO		
Identificador	Nombre	
HU009	Cancelación de reserva	
Puntos de historia estimados		4
Prioridad	Alta	
Responsable	Equipo de desarrollo	
Descripción		
Como usuario, quiero poder cancelar mi reserva en caso de que no necesite el casillero, para que otro estudiante pueda utilizarlo.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El usuario debe poder cancelar su reserva desde la aplicación.	Usabilidad
2	La cancelación debe procesarse en menos de 2 segundos.	Eficiencia
3	El casillero cancelado debe liberarse inmediatamente para otros usuarios.	Disponibilidad



- RF10. El sistema debe permitir a los usuarios reportar problemas relacionados con los casilleros o el sistema, incluyendo una descripción del incidente.

HISTORIA DE USUARIO		
Identificador	Nombre	
HU010	Reporte de incidentes	
Puntos de historia estimados		5
Prioridad	Alta	
Responsable	Equipo de desarrollo	
Descripción		
Como usuario, quiero poder reportar problemas con el casillero o el sistema para que sean solucionados rápidamente.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	El usuario debe poder reportar problemas desde la aplicación.	Disponibilidad
2	El reporte debe enviarse en menos de 2 segundos.	Eficiencia
3	El sistema debe generar un número de seguimiento para cada incidente.	Usabilidad

### REQUISITOS NO FUNCIONALES

- RNF1. El sistema debe garantizar la seguridad de la información mediante encriptación y autenticación multifactor.

HISTORIA DE USUARIO		
Identificador	Nombre	
HU011	Protección de datos	
Puntos de historia estimados		5
Prioridad	Alta	
Responsable	Equipo de desarrollo backend	
Descripción		
Como usuario, quiero que mis datos personales estén protegidos con mecanismos de encriptación y doble autenticación.		
Numero de criterio	Criterio de aceptación	Atributo de calidad



1	Implementar cifrado de contraseñas y autenticación 2FA.	Seguridad
---	---	-----------

- RNF2. La disponibilidad del sistema debe ser de al menos el 99% del tiempo para garantizar su funcionamiento continuo.

HISTORIA DE USUARIO		
Identificador	Nombre	
HU012	Disponibilidad del sistema	
Puntos de historia estimados		5
Prioridad	Alta	
Responsable	Equipo de desarrollo	
Descripción		
Como usuario, quiero que el sistema esté disponible todo el tiempo para acceder a mi casillero cuando lo necesite.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	Disponibilidad mínima del sistema del 99%.	Disponibilidad
2	Pruebas y monitoreo activo.	Disponibilidad

- RNF3. La interfaz debe ser intuitiva y accesible para facilitar su uso a estudiantes y docentes.

HISTORIA DE USUARIO		
Identificador	Nombre	
HU013	Diseño accesible	
Puntos de historia estimados		3
Prioridad	Media	
Responsable	Equipo de desarrollo frontend	
Descripción		
Como usuario, quiero que la app sea fácil de entender y navegar para usarla sin dificultades.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	Evaluación de usabilidad con usuarios reales.	Usabilidad
2	Cumplir con criterios de diseño accesible.	Usabilidad



- RNF4. El sistema debe ser escalable para permitir la integración de nuevos casilleros en el futuro.

HISTORIA DE USUARIO		
Identificador	Nombre	
HU014	Ampliación del sistema	
Puntos de historia estimados		8
Prioridad	Media	
Responsable	Equipo de desarrollo backend	
Descripción		
Como administrador, quiero que el sistema permita añadir nuevos casilleros sin afectar el rendimiento.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	Arquitectura modular y escalable.	Escalabilidad
2	Pruebas con número creciente de casilleros.	Escalabilidad

- RNF5. Debe cumplir con la normativa de protección de datos personales (Ley 1581 de 2012 y CONPES 3995 de 2020).

HISTORIA DE USUARIO		
Identificador	Nombre	
HU015	Cumplimiento normativo	
Puntos de historia estimados		3
Prioridad	Alta	
Responsable	Equipo de desarrollo	
Descripción		
Como usuario, quiero que el sistema respete mis derechos de privacidad conforme a las leyes colombianas.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	Cumplir con Ley 1581 de 2012 y CONPES 3995 de 2020.	Seguridad
2	Informar al usuario sobre el uso de datos.	Seguridad

- RNF6. El tiempo de respuesta del sistema no debe superar los 3 segundos para cada acción del usuario.



HISTORIA DE USUARIO		
Identificador	Nombre	
HU016	Velocidad del sistema	
Puntos de historia estimados		4
Prioridad	Alta	
Responsable	Equipo de desarrollo backend	
Descripción		
Como usuario, quiero que las acciones que realizo en la app se ejecuten rápidamente.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	Todas las acciones deben completarse en menos de 4 segundos.	Eficiencia
2	Pruebas de rendimiento.	Rendimiento

- RNF7. Debe contar con un mecanismo de respaldo y recuperación de datos en caso de fallos o ataques cibernéticos.

HISTORIA DE USUARIO		
Identificador	Nombre	
HU017	Historia de backup y recuperación	
Puntos de historia estimados		5
Prioridad	Media	
Responsable	Equipo de desarrollo backend	
Descripción		
Como administrador, quiero que el sistema haga copias de seguridad periódicas para evitar pérdida de información.		
Numero de criterio	Criterio de aceptación	Atributo de calidad
1	Mecanismo automático de respaldo diario.	Fiabilidad
2	Simulacro de recuperación con éxito.	Rendimiento



## ANÁLISIS DE RESTRICCIONES

El desarrollo e implementación del sistema de casilleros inteligentes en la Universidad EAN enfrenta diversas restricciones que pueden afectar su viabilidad y adopción. Estas restricciones pueden clasificarse en aspectos técnicos, económicos, legales, de seguridad y socioculturales.

Desde el punto de vista técnico, el sistema requiere que los usuarios dispongan de un smartphone compatible y acceso a internet, lo que puede limitar su uso en caso de estudiantes que no cuenten con estos recursos, adicional, la infraestructura de red de la universidad debe ser estable y garantizar la conectividad en los espacios donde se instalen los casilleros, dado que, cualquier falla en la comunicación en tiempo real en el sistema podría afectar la funcionalidad del mismo.

En cuanto a las restricciones económicas, el costo de implementación puede ser un factor determinante. La adquisición del hardware para los casilleros inteligentes, la contratación de servidores para la gestión de datos y el mantenimiento del sistema representan inversiones significativas. Si el presupuesto disponible no es suficiente, podría ser necesario buscar fuentes de financiamiento o ajustar el alcance del proyecto.

Desde una perspectiva legal, es fundamental garantizar que el sistema cumpla con las normativas de protección de datos personales, seguridad digital y privacidad de los usuarios, asimismo, cualquier requisito institucional relacionado con la instalación de infraestructura tecnológica dentro del campus debe ser considerado.

En términos de seguridad y salud, el uso del sistema no debe generar riesgos para los usuarios, estos deben cumplir con estándares de calidad que eviten fallas mecánicas o eléctricas que puedan representar un peligro. También es importante que la autenticación y



autorización de los accesos sean seguras para prevenir vulnerabilidades que puedan comprometer la privacidad de los estudiantes.

Por último, en el ámbito sociocultural, la aceptación del sistema por parte de los estudiantes y docentes es clave para su éxito. Algunos usuarios pueden mostrar resistencia al cambio y preferir métodos tradicionales, lo que podría afectar la adopción del sistema, de igual manera, la facilidad de uso debe garantizarse para que personas con poca experiencia en tecnología puedan interactuar con la aplicación sin dificultades.



## COMPARACIÓN DEL MODELO CON MODELOS ACTUALES

### CASILLEROS ACTUALES DEL GIMNASIO LA UNIVERSIDAD EAN

El modelo que se busca implementar consiste en la automatización del sistema de apertura y cierre de los casilleros mediante una aplicación móvil. Desde la aplicación, los usuarios podrán abrir o cerrar el casillero que han reservado con solo oprimir un botón, eliminando la necesidad de ingresar contraseñas manualmente o interactuar con mecanismos físicos en el casillero. A diferencia de los casilleros actuales, que utilizan cerraduras con combinación numérica, este nuevo sistema eliminará los problemas recurrentes asociados con los mecanismos manuales.

En primer lugar, los casilleros actuales presentan fallas comunes, como el mal cierre del candado o la configuración incorrecta de la combinación numérica, esto ocurre cuando un número no queda bien colocado, lo que resulta en una contraseña distinta a la que el usuario deseaba, generando inseguridad y frustración para los usuarios. Con el modelo propuesto, estos problemas se evitarán, siendo que la apertura y cierre de los casilleros estarán completamente automatizados, garantizando que el casillero siempre quede correctamente asegurado.

Otro aspecto destacado es la facilidad para identificar y acceder al casillero asignado, con los casilleros actuales, si un usuario olvida cuál es su casillero o su contraseña debe buscar al personal de seguridad más cercano, lo cual puede tomar mucho tiempo. Luego, el personal debe revisar casillero por casillero hasta encontrar el correcto, y finalmente documentar el incidente. Este proceso no solo es tedioso, sino que también interrumpe la rutina tanto del usuario como del personal. Sin embargo, con el modelo propuesto, la



aplicación móvil indicará al usuario cuál es su casillero asignado en pantalla, eliminando la necesidad de asistencia externa y ahorrando tiempo.

Por otra parte, el sistema actual presenta un problema grave cuando un casillero queda abierto y las pertenencias son tomadas por un tercero. En estos casos, se debe iniciar un proceso extenso que incluye la creación de un caso, la revisión de cámaras de seguridad, la búsqueda en objetos perdidos y, en el peor de los casos, una espera de hasta una semana para recuperar los objetos, lo que no solo es desgastante para los usuarios, sino que también genera desconfianza. Con el modelo de casilleros inteligentes, la aplicación móvil notificará al usuario si su casillero queda abierto, permitiéndole asegurarse de que esté correctamente cerrado y evitando así la pérdida de pertenencias.

### **CASILLEROS ACTUALES DEL PARQUEADERO DE LA UNIVERSIDAD EAN**

Al comparar los casilleros ubicados en los parqueaderos de la Universidad EAN con el proyecto de innovación de casilleros inteligentes, se evidencian diferencias significativas en términos de eficiencia, autonomía y experiencia del usuario. En primer lugar, en el sistema actual, los usuarios deben solicitar un casillero al personal encargado, un proceso que puede resultar demorado si la persona responsable no está disponible. Incluso cuando el personal está presente, el usuario debe esperar a que se le asigne un casillero, se le entregue una llave física y una ficha correspondiente. Este procedimiento no solo consume tiempo, sino que también depende completamente de la disponibilidad de un tercero, lo que limita la autonomía del usuario.

En contraste, el proyecto de casilleros inteligentes elimina la necesidad de intermediarios, los usuarios pueden reservar, abrir y gestionar un casillero directamente desde una aplicación móvil, sin depender de la asistencia del personal. Esto no solo agiliza el proceso,



sino que también brinda mayor comodidad y flexibilidad, ya que el usuario tiene control total sobre el uso del casillero en todo momento.

Otra diferencia notable radica en la forma de identificación del casillero, en el sistema actual se entrega un ticket o papel que corresponde al casillero asignado, si este documento se pierde, el usuario enfrenta dificultades para recuperar sus pertenencias, ya que no hay forma de verificar su identidad o la asignación del casillero. Este problema se resuelve completamente con el proyecto de casilleros inteligentes, ya que la aplicación móvil almacena toda la información necesaria y al estar toda la información en el celular del usuario, se elimina el riesgo de pérdida o extravío, garantizando un acceso seguro y confiable en todo momento.

Además, el sistema actual presenta inconvenientes cuando el usuario necesita retirar sus pertenencias, puesto que, la persona debe regresar con el personal para solicitar que le abran el casillero, lo que implica otra espera hasta que se le devuelva la llave y en consecuencia, limita la libertad del usuario, especialmente en situaciones de urgencia. Con el modelo de casilleros inteligentes, el usuario puede abrir el casillero en cualquier momento desde su celular, sin necesidad de esperar a que alguien le brinde acceso, optimizando el tiempo y ofreciendo una experiencia más fluida y autónoma.



## VENTAJAS Y DESVENTAJAS DE LA IMPLEMENTACIÓN DE CASILLEROS INTELIGENTES

### VENTAJAS

- Autonomía y comodidad para el usuario:

Los usuarios pueden reservar, abrir y gestionar casilleros directamente desde una aplicación móvil, sin depender de la asistencia de un tercero. Esto elimina tiempos de espera y ofrece una experiencia más fluida y autónoma.

- Mayor seguridad:

Al eliminar el uso de llaves físicas, tickets o combinaciones numéricas, se reduce el riesgo de pérdida, robo o acceso no autorizado. La aplicación móvil garantiza que solo el usuario autorizado pueda acceder al casillero.

- Acceso en tiempo real:

Gracias a la integración de tecnologías como IoT y protocolos de comunicación en tiempo real, los usuarios pueden abrir y cerrar casilleros de manera instantánea, sin demoras.

- Facilidad de uso:

La aplicación móvil indica al usuario cuál es su casillero asignado, eliminando la necesidad de recordar números o buscar casilleros manualmente, adicionalmente, la interfaz es intuitiva y fácil de usar.

- Notificaciones y alertas:

El sistema puede enviar notificaciones al usuario si el casillero queda abierto o si se detecta alguna anomalía, lo que aumenta la seguridad y la tranquilidad del usuario.

- Reducción de costos operativos:



Al automatizar el proceso de gestión de casilleros, se reduce la necesidad de personal dedicado a la supervisión y asistencia, lo que puede traducirse en ahorros para la institución.

- Escalabilidad:

El sistema es escalable y puede adaptarse a un mayor número de casilleros o usuarios sin comprometer su eficiencia. Esto lo hace ideal para entornos con alta demanda, como universidades o centros comerciales.

### DESVENTAJAS

- Dependencia de la tecnología:

El sistema depende completamente de la conectividad a internet y del funcionamiento de la aplicación móvil, por lo tanto, si hay fallas en la red o en los dispositivos los usuarios podrían verse afectados.

- Costo inicial de implementación:

La instalación completa del sistema puede requerir una inversión inicial significativa, especialmente si se trata de un gran número de casilleros.

- Curva de aprendizaje para los usuarios:

Algunos usuarios, especialmente aquellos menos familiarizados con la tecnología, podrían tener dificultades para adaptarse al uso de la aplicación móvil y los sistemas digitales.

- Mantenimiento técnico:

Los sistemas IoT y los sistemas integrados requieren mantenimiento periódico para garantizar su correcto funcionamiento, incluyendo actualizaciones de software, reparaciones de hardware y soporte técnico.

- Vulnerabilidad a ciberataques:



Al ser un sistema conectado a internet, existe el riesgo de ciberataques, por ello es fundamental implementar medidas de seguridad robustas, como cifrado de datos y autenticación de dos factores, para proteger la información de los usuarios.

- Dependencia del celular:

Si el usuario no tiene su celular disponible (por ejemplo, si se queda sin batería o lo pierde), podría enfrentar dificultades para acceder a su casillero.

- Limitaciones en zonas con poca conectividad:

En áreas donde la conexión a internet es débil o intermitente, el sistema podría no funcionar de manera óptima, lo que afectaría la experiencia del usuario.



## MÉTODO DE SELECCIÓN

La selección de tecnología para el sistema de casilleros inteligentes se basa en un análisis exhaustivo de los requerimientos del proyecto y las soluciones disponibles en el mercado actual. Para establecer una base sólida, se consideran tres aspectos fundamentales: el rendimiento del sistema, su capacidad de escalabilidad y los requisitos de seguridad.

El rendimiento se establece como un criterio clave, ya que el sistema debe ser capaz de gestionar múltiples solicitudes simultáneas con un tiempo de respuesta inferior a dos segundos, garantizando una experiencia fluida para los usuarios. La escalabilidad también se prioriza, dado que el sistema deberá crecer junto con las necesidades de la universidad, permitiendo la incorporación de nuevos casilleros y usuarios sin afectar su eficiencia.

En cuanto a la seguridad, se definen requisitos estrictos, incluyendo la implementación de mecanismos de encriptación de datos para proteger tanto la información de los usuarios como la integridad del sistema. Asimismo, se evalúan criterios no funcionales como la mantenibilidad del código, el soporte técnico disponible y los costos asociados, con el objetivo de asegurar una solución sostenible y robusta a largo plazo.

## SELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN

Para el desarrollo del sistema, se seleccionó Kotlin como lenguaje principal en el entorno de Android Studio, tanto para el frontend como para el backend de la aplicación móvil. Esta decisión se basa en la familiaridad previa del equipo con el lenguaje, así como en el reconocimiento y soporte que Kotlin tiene dentro del desarrollo nativo en Android. Gracias a su sintaxis moderna y concisa, Kotlin permite construir aplicaciones robustas, eficientes y con una experiencia de usuario fluida.



La decisión de utilizar Kotlin en toda la lógica de la aplicación también responde al interés de mantener una coherencia tecnológica que facilite el mantenimiento, el escalado y la integración de nuevas funcionalidades en el futuro.

### **SELECCIÓN DE BASE DE DATOS**

El proceso de selección de la base de datos inició con la evaluación de alternativas como MySQL y PostgreSQL, considerando la posibilidad de manejar tanto datos relacionales como estructuras más complejas. Tras analizar los requerimientos del sistema —particularmente la necesidad de integridad, concurrencia y escalabilidad— se optó por PostgreSQL.

La elección responde a su solidez como sistema de gestión de bases de datos relacionales, su cumplimiento con las propiedades ACID, y su excelente rendimiento en operaciones concurrentes. PostgreSQL proporciona la robustez necesaria para gestionar procesos como la reserva de casilleros, el control de accesos y el historial de uso, garantizando la integridad de los datos incluso en entornos con múltiples usuarios interactuando en tiempo real, asimismo, su flexibilidad para manejar tipos de datos avanzados y extensiones futuras lo convierte en una solución escalable y adaptable a nuevas funcionalidades que puedan incorporarse en fases posteriores del proyecto.



## MÉTODO DE DESARROLLO DE LA SOLUCIÓN

El desarrollo de la solución se basa en los principios de Clean Architecture, lo cual permite una clara separación de responsabilidades y facilita el mantenimiento y escalabilidad del sistema. Se implementa una arquitectura en capas que distingue entre la interfaz de usuario, la lógica de negocio y el acceso a datos.

La aplicación se desarrolla utilizando Android Studio como entorno principal, lo que permite aprovechar al máximo las herramientas nativas para la creación de aplicaciones móviles en Android. Asimismo, se adoptan los principios SOLID para asegurar un diseño de software robusto, flexible y fácil de mantener.

Desde las primeras fases del desarrollo, se integran prácticas de DevOps, incluyendo integración y despliegue continuos (CI/CD), con el objetivo de garantizar que cada cambio pase por pruebas automatizadas antes de ser llevado a producción.

## METODOLOGÍA ÁGIL Y ETAPAS DEL PROYECTO

Para la gestión del proyecto, se utilizará la metodología ágil Scrum, debido a su enfoque iterativo y su capacidad para adaptarse a cambios durante el desarrollo. Esta metodología permitirá una mejor organización del trabajo y avances progresivos con entregas funcionales al final de cada sprint.

El proyecto tendrá una duración aproximada de cuatro meses, divididos en sprints de dos semanas, durante el primer mes se desarrollará la fase de documentación y análisis, en la que se realizarán las investigaciones previas, se definirán los requerimientos funcionales, se identificarán las necesidades de los usuarios y se realizará un estudio de soluciones similares, seguido de esto, se diseñarán los mockups y las interfaces de usuario, ajustando el diseño según las observaciones del equipo.



En los meses siguientes, se avanzará con la construcción de las pantallas y funcionalidades de forma incremental, utilizando únicamente Kotlin dentro de Android Studio. El sistema será completamente local, sin uso de APIs externas, y buscará una integración directa con los dispositivos de hardware, siguiendo el objetivo de lograr un casillero inteligente, autónomo y funcional.

El desarrollo se organizará por módulos, con revisiones continuas al finalizar cada sprint, para ello, las tareas se gestionarán de manera colaborativa y se trabajará con GitHub como sistema de control de versiones, con el fin de que todos los integrantes del equipo puedan estar conectados, revisar el trabajo de los demás y actualizar fácilmente los cambios en el repositorio compartido.



## ANÁLISIS DE COSTOS DEL PROYECTO

### ANÁLISIS DEL COSTO A NIVEL DE PROGRAMACIÓN

El proyecto de casilleros inteligentes de la Universidad EAN tiene como objetivo la implementación de un sistema digital que permita a los estudiantes reservar, modificar o cancelar el uso de casilleros a través de una aplicación móvil. Esta solución se integra como parte de una estrategia de automatización y mejora en la gestión de espacios dentro de la institución, brindando mayor comodidad, control y eficiencia a los usuarios.

Este análisis se enfoca exclusivamente en los costos asociados al desarrollo y programación de la aplicación móvil, desarrollada en Android Studio utilizando el lenguaje de programación Kotlin. El equipo de desarrollo está conformado por tres integrantes, quienes se encargarán de todas las fases del ciclo de vida del software: análisis, diseño, codificación, pruebas, integración y puesta en marcha. Se estima que el proceso de desarrollo requerirá un total de 30 horas de trabajo, distribuidas de forma equitativa entre los tres participantes. El cálculo de los costos se realiza tomando como referencia una tarifa profesional de 60 USD por hora por desarrollador, lo que representa un costo total de 5.400 USD en términos de recurso humano.

En cuanto a licencias y herramientas de desarrollo, Android Studio es una plataforma gratuita, por lo cual no genera gastos adicionales. No obstante, se consideran costos asociados a la adquisición de licencias necesarias para la publicación de la aplicación, así como a la integración de librerías o herramientas de terceros que optimicen el rendimiento y las funcionalidades del sistema. Se estima un valor total de 125 USD, incluyendo el pago único de 25 USD requerido para la publicación en Google Play Store y un margen de 100 USD destinado a herramientas externas o librerías de backend.



Para el funcionamiento integral del sistema, será necesario contar con servicios en la nube que permitan el almacenamiento de datos, la autenticación de usuarios, y la comunicación con el sistema de apertura de casilleros. En este sentido, se contempla el uso de Firebase, con un costo estimado de 25 USD mensuales bajo el Plan Spark (teniendo en cuenta posibles costos adicionales por uso de funciones avanzadas), y AWS (Amazon Web Services) con un valor aproximado de 25 USD mensuales, destinado al hosting y gestión de bases de datos en producción.

Adicionalmente, se prevé la publicación de la aplicación en Apple App Store con el fin de ampliar la cobertura a usuarios de dispositivos iOS. Este proceso implica un pago anual de 99 US, correspondiente a la suscripción como desarrollador. Sumando los valores correspondientes al despliegue en ambas tiendas, el costo total estimado en esta categoría asciende a 174 USD.

Finalmente, se incluyen costos indirectos asociados al desarrollo, tales como servicios básicos (electricidad, internet, uso de espacio físico y herramientas personales de trabajo), los cuales se estiman en 30 USD.

Tomando todo el análisis realizado para los costos del aplicativo móvil, el costo total estimado para la implementación de la solución desde la perspectiva del desarrollo y programación asciende a 5.729 US. Esta cifra incluye los honorarios del equipo de desarrollo, las licencias y herramientas utilizadas, los servicios en la nube requeridos para el funcionamiento del sistema, así como los gastos de publicación y costos indirectos derivados del entorno de trabajo. Este análisis económico permite dimensionar de forma clara el esfuerzo técnico y financiero involucrado en la creación de una solución digital que responde a necesidades reales dentro del entorno universitario.



## RESUMEN GENERAL DE COSTOS

Descripción	Costo
Honorarios del equipo de desarrollo (3 desarrolladores, 30 horas en total)	\$5,400 USD
Licencias y herramientas de desarrollo (Google Play, librerías)	\$125 USD
Servicios en la nube (Firebase + AWS, primer mes)	\$50 USD
Publicación en Apple App Store (tarifa anual)	\$99 USD
Costos indirectos (infraestructura, conectividad, espacio)	\$30 USD
<b>Total estimado</b>	<b>\$5,729 USD</b>

## ANÁLISIS DE COSTO PARA INFRAESTRUCTURA FÍSICA

Los componentes electrónicos fundamentales para el sistema de casilleros requieren una inversión inicial significativa. Los electroimanes de 12V, necesarios para el mecanismo de cierre de cada casillero, tienen un costo entre \$11-19 USD cada uno, necesitando cuatro unidades para el sistema inicial, lo que suma entre \$76 USD en total.

El cerebro del sistema, un ESP32, tiene un valor entre \$12 USD. Para el control de los electroimanes, se necesita un módulo relay de 4 canales que cuesta \$6 USD. La alimentación del sistema requiere una fuente de 12V/5A, con un precio aproximado de \$14 USD. Los componentes electrónicos varios, incluyendo cables, resistencias y diodos, suman \$7 USD, la caja de control para proteger estos componentes tiene un costo de \$12 USD.



La estructura física representa una parte importante del presupuesto. Cada casillero, construido en metal o aluminio, tiene un costo de \$35 USD. Para el sistema inicial de cuatro casilleros, el costo total de la estructura física de \$137 USD. Los componentes adicionales que mejoran la funcionalidad del sistema incluyen sensores Reed switch para las puertas, que cuestan \$7 USD por el set de cuatro unidades. Los LEDs indicadores para mostrar el estado de cada casillero tienen un valor \$6 USD por el conjunto, y un display LCD I2C para mostrar información adicional cuesta \$6 USD.

Para la parte digital del sistema, los costos anuales incluyen el dominio web, que oscila entre \$19 USD, y el hosting básico, que puede costar \$69 USD al año. Estos servicios son necesarios para mantener la aplicación web funcionando correctamente.

El costo total del sistema, sumando todos los componentes básicos, la estructura física y los servicios web para el primer año, se aproxima a \$382 USD.



## APLICACIÓN DEL MÉTODO DE DESARROLLO

A partir del enfoque metodológico previamente definido, se procede a la aplicación práctica del mismo mediante una serie de actividades estructuradas que permiten llevar el sistema desde la etapa conceptual hasta su implementación funcional. A continuación, se describen los elementos clave desarrollados en el transcurso del proyecto, que abarcan desde la planificación inicial hasta las pruebas de calidad del software.

### PRODUCT BACKLOG

Cod. HU	Rol	Funcionalidad	¿Para qué?	Responsable	Criterios de aceptación
HU001	Usuario	Registro y autenticación	Permitir a estudiantes y docentes ingresar a la plataforma con sus credenciales institucionales de forma rápida y segura	Equipo de desarrollo	<ol style="list-style-type: none"> <li>1. El usuario debe autenticarse con su correo institucional y contraseña</li> <li>2. La interfaz debe permitir recuperación de contraseña y ser clara</li> <li>3. El sistema debe validar las credenciales en menos de 3 segundos</li> </ol>
HU002	Usuario	Reserva de casillero	Permitir a los usuarios reservar y liberar un casillero para guardar sus pertenencias de forma segura	Equipo de desarrollo	<ol style="list-style-type: none"> <li>1. La reserva debe confirmarse en menos de 3 segundos</li> <li>2. El usuario debe recibir un código único de acceso para su casillero</li> <li>3. Solo deben mostrarse los casilleros disponibles al momento de la reserva</li> </ol>



HU003	Usuario	Apertura del casillero	Permitir al usuario abrir su casillero mediante un código único de forma segura	Equipo de desarrollo	<ol style="list-style-type: none"> <li>1. El casillero solo debe abrirse con el código correcto</li> <li>2. Debe haber una opción para solicitar un nuevo código en caso de pérdida</li> <li>3. La apertura debe realizarse en menos de 2 segundos tras la validación</li> </ol>
HU004	Usuario	Código de acceso	Generar un código único y temporal que permita abrir un casillero reservado	Equipo de desarrollo	<ol style="list-style-type: none"> <li>1. El código debe ser generado automáticamente y expirar en el tiempo definido</li> <li>2. No se puede reutilizar</li> </ol>
HU005	Admin.	Registro de actividad	Permitir a los administradores monitorear el historial de uso de los casilleros	Equipo de desarrollo	<ol style="list-style-type: none"> <li>1. Cada acción debe quedar registrada con hora y usuario</li> <li>2. Debe ser accesible solo para administradores</li> </ol>
HU006	Admin.	Gestión administrativa	Permitir que el administrador intervenga en el sistema para resolver fallos	Equipo de desarrollo	<ol style="list-style-type: none"> <li>1. Debe haber una interfaz para resetear o bloquear</li> <li>2. Los cambios deben registrarse en el historial</li> </ol>
HU007	Usuario	Notificaciones al usuario	Mantener informado al usuario sobre el estado de su reserva y casillero en tiempo real	Equipo de desarrollo	<ol style="list-style-type: none"> <li>1. Las notificaciones deben enviarse en menos de 10 segundos tras cualquier cambio en la reserva</li> <li>2. Los mensajes deben ser claros y contener información relevante sobre la reserva</li> <li>3. No deben divulgarse datos personales del usuario en las notificaciones</li> <li>4. El usuario debe recibir recordatorios 5 minutos antes de que su reserva expire</li> </ol>



HU008	Admin.	Cumplimiento con políticas de ciberseguridad	Garantizar que la aplicación esté alineada con los estándares de seguridad digital de la universidad	Equipo de desarrollo	1. Cumplimiento con las políticas establecidas por EAN
HU009	Usuario	Cancelación de reserva	Permitir al usuario cancelar su reserva para liberar el casillero a otros estudiantes	Equipo de desarrollo	1. El usuario debe poder cancelar su reserva desde la aplicación 2. La cancelación debe procesarse en menos de 2 segundos 3. El casillero cancelado debe liberarse inmediatamente para otros usuarios
H010	Usuario	Reporte de incidentes	Permitir al usuario reportar problemas con el casillero o sistema para solución rápida	Equipo de desarrollo	1. El usuario debe poder reportar problemas desde la aplicación 2. El reporte debe enviarse en menos de 2 segundos 3. El sistema debe generar un número de seguimiento para cada incidente
H011	Usuario	Protección de datos personales	Garantizar la seguridad y privacidad de los datos del usuario mediante mecanismos seguros	Equipo de desarrollo	1. Implementar cifrado de contraseñas y autenticación de dos factores (2FA)
H012	Usuario	Garantizar alta disponibilidad del sistema	Asegurar que el usuario pueda acceder al casillero en cualquier momento que lo necesite	Equipo de desarrollo	1. Disponibilidad mínima del sistema del 99% 2. Realizar pruebas y monitoreo activo para asegurar la disponibilidad



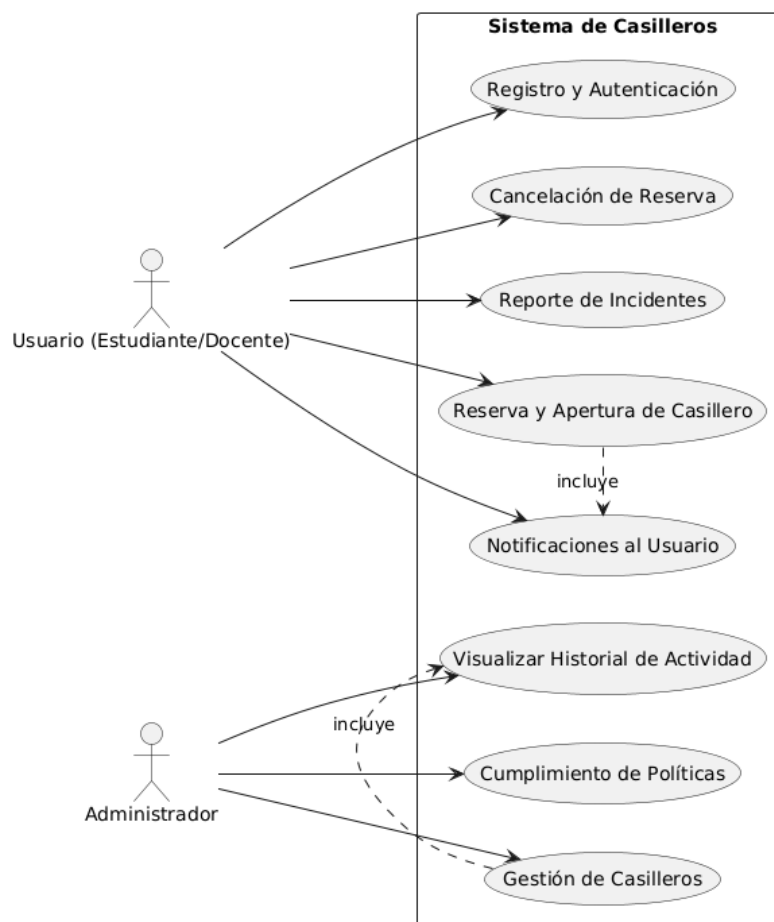
H013	Usuario	Diseño de una app fácil de usar y accesible	Facilitar que cualquier usuario entienda y navegue la aplicación sin dificultades	Equipo de desarrollo	<ol style="list-style-type: none"> <li>1. Evaluación de usabilidad con usuarios reales</li> <li>2. Cumplir con criterios de diseño accesible</li> </ol>
H014	Admin.	Permitir añadir nuevos casilleros sin afectar el rendimiento	Facilitar la ampliación del sistema para soportar más casilleros sin perder eficiencia	Equipo de desarrollo	<ol style="list-style-type: none"> <li>1. Arquitectura modular y escalable</li> <li>2. Pruebas con número creciente de casilleros</li> </ol>
H015	Usuario	Garantizar el respeto a la privacidad y protección de datos del usuario conforme a la ley colombiana	Asegurar que el sistema cumpla con normativas legales y proteja los derechos de los usuarios	Equipo de desarrollo	<ol style="list-style-type: none"> <li>1. Cumplir con Ley 1581 de 2012 y CONPES 3995 de 2020</li> <li>2. Informar claramente al usuario sobre el uso de sus datos</li> </ol>
H016	Usuario	Ejecutar las acciones del usuario rápidamente	Mejorar la experiencia de uso con tiempos de respuesta rápidos.	Equipo de desarrollo	<ol style="list-style-type: none"> <li>1. Todas las acciones deben completarse en menos de 4 segundos</li> <li>2. Realizar pruebas de rendimiento para asegurar la velocidad</li> </ol>
H017	Admin.	Realizar copias de seguridad automáticas	Evitar pérdida de información y garantizar recuperación	Equipo de desarrollo	<ol style="list-style-type: none"> <li>1. Mecanismo automático de respaldo diario</li> <li>2. Simulacro de recuperación exitoso</li> </ol>



## ANÁLISIS ARQUITECTURA Y DISEÑO

Esta sección incluye los casos de uso identificados durante la fase de análisis, que representan las principales interacciones de los usuarios con el sistema y orientan la planificación del desarrollo.

- Diagrama de casos de uso





- Descripción de casos de uso

Caso de uso	Registro y autenticación
<b>Descripción</b>	Permitir a un usuario (estudiante o docente) registrarse y autenticarse con credenciales institucionales para acceder a la plataforma.
<b>Historias de usuario vinculadas</b>	HU001 - Registro y autenticación
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. El usuario debe autenticarse con correo institucional y contraseña.</li> <li>2. Interfaz clara y opción para recuperación de contraseña.</li> <li>3. Validación en menos de 3 segundos.</li> </ol>
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario abre la aplicación.</li> <li>2. Ingresa su correo institucional y contraseña.</li> <li>3. El sistema valida las credenciales.</li> <li>4. Acceso permitido si las credenciales son correctas.</li> <li>5. El usuario es dirigido al panel principal.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>1. El usuario olvida su contraseña.</li> <li>2. Selecciona la opción "Recuperar contraseña".</li> <li>3. Recibe instrucciones para restablecerla.</li> <li>4. Si no puede recuperar la contraseña, puede contactar soporte.</li> </ol>

Caso de uso	Gestión de reserva de casillero
<b>Descripción</b>	Permitir al usuario reservar un casillero disponible desde la aplicación web o móvil, recibir un código de acceso, y cancelar la reserva en caso de que no lo necesite para liberar el casillero.
<b>Historias de usuario vinculadas</b>	HU002 - Reserva de casillero HU003 - Apertura del casillero HU009 - Cancelación de reserva
<b>Criterios de aceptación</b>	<p>HU002:</p> <ol style="list-style-type: none"> <li>1. Confirmación de reserva en menos de 3 segundos.</li> <li>2. Solo casilleros disponibles se muestran.</li> </ol> <p>HU003:</p> <ol style="list-style-type: none"> <li>1. Casillero solo se abre con autorización desde la app.</li> <li>2. Apertura en menos de 2 segundos.</li> </ol> <p>HU009:</p> <ol style="list-style-type: none"> <li>1. Cancelación desde la aplicación.</li> <li>2. Procesamiento en menos de 2 segundos.</li> <li>3. Casillero liberado inmediatamente.</li> </ol>



<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. Usuario accede a la aplicación.</li> <li>2. Selecciona la opción para reservar un casillero.</li> <li>3. El sistema muestra solo casilleros disponibles.</li> <li>4. Usuario selecciona un casillero.</li> <li>5. Reserva se confirma.</li> <li>6. Usuario puede abrir o cerrar el casillero desde un botón en la app.</li> <li>7. Usuario puede cancelar la reserva si lo desea.</li> </ol>
<b>Flujo alterno</b>	<ol style="list-style-type: none"> <li>1. El usuario intenta reservar un casillero ya reservado por otro.</li> <li>2. El sistema no muestra ese casillero como disponible.</li> <li>3. El usuario cancela la reserva.</li> <li>4. El sistema libera inmediatamente el casillero para otros usuarios.</li> <li>5. Si el sistema tarda más de 3 segundos en confirmar, muestra mensaje de error y sugiere reintentar.</li> </ol>

<b>Caso de uso</b>	<b>Reporte de incidentes por usuario</b>
<b>Descripción</b>	Permitir al usuario reportar problemas con el casillero o la aplicación para que el equipo de soporte pueda solucionarlos rápidamente, generando un número de seguimiento para cada reporte.
<b>Historias de usuario vinculadas</b>	HU010 - Reporte de incidentes
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Usuario puede reportar problemas desde la app.</li> <li>2. El reporte se envía en menos de 2 segundos.</li> <li>3. Sistema genera un número de seguimiento para el incidente.</li> </ol>
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. Usuario accede a la sección de reporte de incidentes en la app.</li> <li>2. Describe el problema y envía el reporte.</li> <li>3. El sistema recibe y registra el reporte.</li> <li>4. El sistema envía número de seguimiento al usuario.</li> <li>5. Usuario recibe confirmación del envío.</li> </ol>
<b>Flujo alterno</b>	<ol style="list-style-type: none"> <li>1. Usuario intenta enviar reporte, pero no tiene conexión.</li> <li>2. La app muestra mensaje de error y opción para reintentar.</li> <li>3. El envío tarda más de 2 segundos.</li> <li>4. El sistema muestra mensaje de demora y confirma si desea reintentar.</li> <li>5. Usuario no describe el problema.</li> <li>6. El sistema solicita completar el campo antes de enviar.</li> </ol>



Caso de uso	Cancelación de reserva por usuario
<b>Descripción</b>	Permitir al usuario cancelar su reserva de casillero desde la aplicación para liberar el casillero y que otros usuarios puedan reservarlo.
<b>Historias de usuario vinculadas</b>	HU009 - Cancelación de reserva
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. El usuario puede cancelar la reserva desde la app.</li> <li>2. La cancelación se procesa en menos de 2 segundos.</li> <li>3. El casillero cancelado queda disponible inmediatamente para otros usuarios.</li> </ol>
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. Usuario accede a su lista de reservas.</li> <li>2. Selecciona la reserva que desea cancelar.</li> <li>3. Confirma la cancelación.</li> <li>4. El sistema procesa la cancelación y libera el casillero.</li> <li>5. Usuario recibe confirmación de cancelación.</li> </ol>
<b>Flujo alterno</b>	<ol style="list-style-type: none"> <li>1. Usuario intenta cancelar, pero no tiene conexión.</li> <li>2. La app muestra mensaje de error y opción para reintentar.</li> <li>3. Usuario cambia de opinión antes de confirmar.</li> <li>4. Puede volver a la lista sin cancelar.</li> <li>5. El sistema no puede procesar la cancelación (error interno).</li> <li>6. Muestra mensaje de error y sugiere contactar soporte.</li> </ol>

Caso de uso	Reporte de incidentes
<b>Descripción</b>	El usuario puede reportar problemas con el casillero o la plataforma para que sean solucionados rápidamente.
<b>HU vinculadas</b>	HU010 Reporte de incidentes
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. El usuario debe poder reportar problemas desde la aplicación.</li> <li>2. El reporte debe enviarse en menos de 2 segundos.</li> <li>3. El sistema debe generar un número de seguimiento para cada incidente.</li> </ol>
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la sección de reporte en la aplicación.</li> <li>2. Completa el formulario de reporte con detalles del incidente.</li> <li>3. Envía el reporte.</li> <li>4. El sistema registra el reporte, genera un número de seguimiento y confirma al usuario.</li> </ol>
<b>Flujo alterno</b>	<ul style="list-style-type: none"> <li>- Si falta información obligatoria, se solicita completar.</li> <li>- Si falla el envío por red, se muestra error y permite reintentar.</li> </ul>

Caso de uso	Notificaciones al usuario
-------------	---------------------------



<b>Descripción</b>	El sistema envía notificaciones al usuario sobre el estado del casillero y recordatorios sobre la reserva.
<b>HU vinculadas</b>	HU007 Notificaciones al usuario
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Las notificaciones deben enviarse en menos de 10 segundos tras cualquier cambio en la reserva.</li> <li>2. Los mensajes deben ser claros y contener información relevante.</li> <li>3. No deben divulgarse datos personales del usuario.</li> <li>4. El usuario debe recibir recordatorios 5 minutos antes de que su reserva expire.</li> </ol>
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario hace o modifica una reserva.</li> <li>2. El sistema genera una notificación.</li> <li>3. La notificación es enviada al usuario y se muestra en su dispositivo.</li> </ol>
<b>Flujo alterno</b>	<ul style="list-style-type: none"> <li>- Si hay fallos en el envío, se reintenta hasta 3 veces.</li> <li>- Si el usuario tiene notificaciones desactivadas, se almacena en la app para consulta posterior.</li> </ul>

<b>Caso de uso</b>	<b>Protección de datos personales</b>
<b>Descripción</b>	El sistema protege los datos personales de los usuarios mediante cifrado y autenticación reforzada.
<b>HU vinculadas</b>	HU011 Protección de datos
<b>Criterios de aceptación</b>	1. Implementar cifrado de contraseñas y autenticación 2FA (Seguridad).
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario se registra o inicia sesión en la app.</li> <li>2. El sistema cifra la contraseña al almacenarla.</li> <li>3. Si está habilitada, el sistema solicita una segunda autenticación.</li> <li>4. El usuario accede a su cuenta de manera segura.</li> </ol>
<b>Flujo alterno</b>	<ul style="list-style-type: none"> <li>- Si la autenticación falla, se permite reintentar hasta 3 veces antes de bloquear temporalmente el acceso.</li> <li>- Si el usuario pierde el método de 2FA, se inicia un proceso de recuperación.</li> </ul>

<b>Caso de uso</b>	<b>Disponibilidad del sistema</b>
<b>Descripción</b>	El sistema debe estar disponible en todo momento para garantizar el acceso oportuno a los casilleros.
<b>HU vinculadas</b>	HU012 Disponibilidad del sistema
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Disponibilidad mínima del sistema del 99%</li> <li>2. Pruebas y monitoreo activo.</li> </ol>
<b>Flujo básico</b>	1. El usuario abre la aplicación.



	<p>2. El sistema responde con normalidad y permite la interacción.</p> <p>3. El usuario accede a las funciones sin interrupciones.</p>
<b>Flujo alterno</b>	<p>- Si el sistema presenta fallos, se muestra un mensaje de mantenimiento.</p> <p>- El equipo técnico es notificado automáticamente mediante los mecanismos de monitoreo.</p>

Caso de uso	Interfaz accesible y usable
<b>Descripción</b>	El sistema debe contar con una interfaz clara, fácil de navegar y accesible para todo tipo de usuarios, incluyendo personas con discapacidad.
<b>HU vinculadas</b>	HU007 Notificaciones al usuario HU013 Diseño accesible
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Evaluación de usabilidad con usuarios reales.</li> <li>2. Cumplir con criterios de diseño accesible.</li> <li>3. Los mensajes deben ser claros y contener información relevante sobre la reserva.</li> <li>4. No deben divulgarse datos personales del usuario en las notificaciones.</li> </ol>
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación.</li> <li>2. Navega entre las funciones sin dificultad.</li> <li>3. Recibe notificaciones claras y comprensibles sobre el estado del casillero o reserva.</li> </ol>
<b>Flujo alterno</b>	<p>- Si el diseño no cumple con criterios de accesibilidad, el usuario puede reportar la dificultad mediante la opción de reporte.</p> <p>- El sistema guía al usuario con mensajes o accesos alternativos (ej. lector de pantalla).</p>

Caso de uso	Escalabilidad del sistema
<b>Descripción</b>	El sistema debe permitir añadir nuevos casilleros sin que se afecte el rendimiento general de la aplicación.
<b>HU vinculadas</b>	HU014 Ampliación del sistema
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Arquitectura modular y escalable.</li> <li>2. Pruebas con número creciente de casilleros.</li> </ol>
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El administrador accede al panel de configuración.</li> <li>2. Añade uno o más casilleros nuevos.</li> <li>3. El sistema integra los nuevos casilleros sin interrupciones.</li> <li>4. Se ejecutan pruebas automáticas para verificar el rendimiento.</li> </ol>



<b>Flujo alterno</b>	<ul style="list-style-type: none"> <li>- Si el número de casilleros genera un aumento significativo en la carga del sistema, se activa un protocolo de ajuste o redistribución de recursos.</li> <li>- El sistema notifica al administrador si se alcanza un umbral crítico de rendimiento.</li> </ul>
----------------------	--

<b>Caso de uso</b>	<b>Cumplimiento normativo</b>
<b>Descripción</b>	El sistema debe garantizar el respeto a los derechos de privacidad de los usuarios conforme a la legislación colombiana.
<b>HU vinculadas</b>	HU015 Cumplimiento normativo
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Cumplir con Ley 1581 de 2012 y CONPES 3995 de 2020.</li> <li>2. Informar al usuario sobre el uso de datos.</li> </ol>
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación por primera vez.</li> <li>2. Se le muestra una política de privacidad detallada.</li> <li>3. El usuario acepta los términos para continuar.</li> <li>4. El sistema guarda el consentimiento.</li> <li>5. Los datos se gestionan conforme a las leyes vigentes.</li> </ol>
<b>Flujo alterno</b>	<ul style="list-style-type: none"> <li>- Si el usuario no acepta la política de privacidad, se le niega el acceso al sistema y se le redirige a la información legal correspondiente.</li> </ul>

<b>Caso de uso</b>	<b>Respuesta rápida del sistema</b>
<b>Descripción</b>	El sistema debe responder rápidamente a las acciones del usuario para brindar una experiencia fluida.
<b>HU vinculadas</b>	HU016 Velocidad del sistema
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Todas las acciones deben completarse en menos de 4 segundos.</li> <li>2. Realizar pruebas de rendimiento.</li> </ol>
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El usuario realiza una acción (como reservar, cancelar o abrir un casillero).</li> <li>2. El sistema procesa la solicitud.</li> <li>3. La acción se ejecuta en menos de 4 segundos.</li> <li>4. El usuario recibe retroalimentación visual o sonora de la ejecución exitosa.</li> </ol>
<b>Flujo alterno</b>	<ul style="list-style-type: none"> <li>- Si el sistema tarda más de 4 segundos, se muestra un mensaje de espera.</li> <li>- Si el sistema falla, se muestra un mensaje de error con la opción de reintentar.</li> </ul>



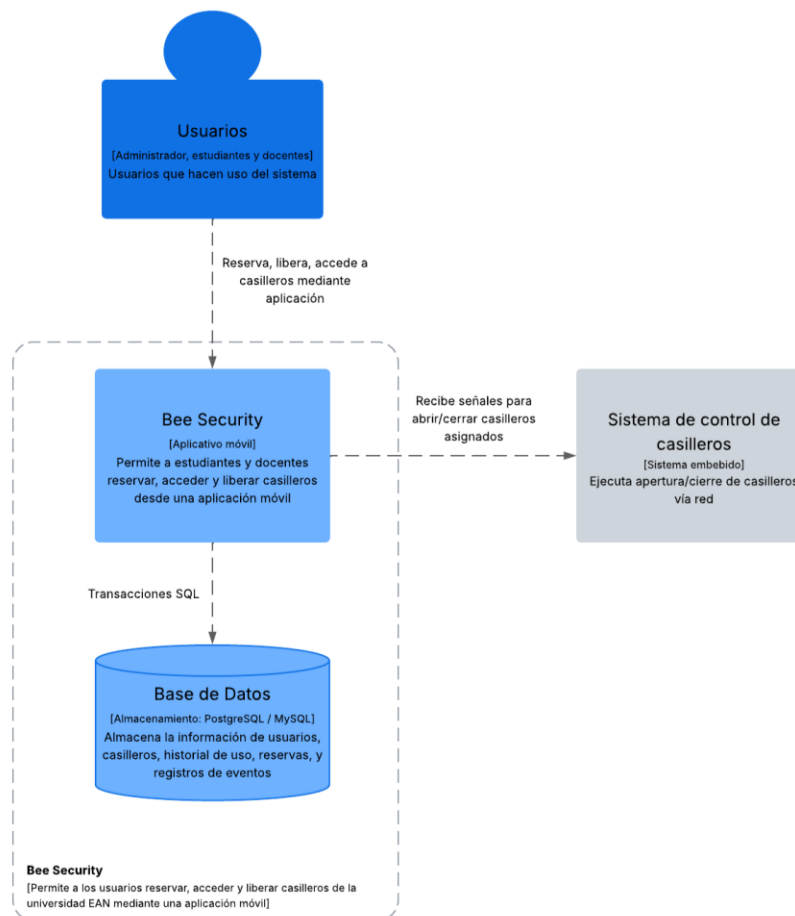
Caso de uso	Respaldo y recuperación de información
<b>Descripción</b>	El sistema debe realizar copias de seguridad automáticas y permitir la recuperación de datos en caso de fallo.
<b>HU vinculadas</b>	HU017 Historia de backup y recuperación
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Mecanismo automático de respaldo diario.</li> <li>2. Simulacro de recuperación con éxito.</li> </ol>
<b>Flujo básico</b>	<ol style="list-style-type: none"> <li>1. El sistema realiza automáticamente un respaldo diario de la base de datos.</li> <li>2. La información respaldada se almacena de forma segura.</li> <li>3. En caso de fallo, el administrador solicita una recuperación.</li> <li>4. El sistema recupera la información desde el respaldo más reciente.</li> </ol>
<b>Flujo alternativo</b>	<ul style="list-style-type: none"> <li>- Si el respaldo falla, se notifica al equipo de desarrollo.</li> <li>- Si la recuperación no es posible, se informa al administrador con detalles del error.</li> </ul>



## ARQUITECTURA DE LA SOLUCIÓN

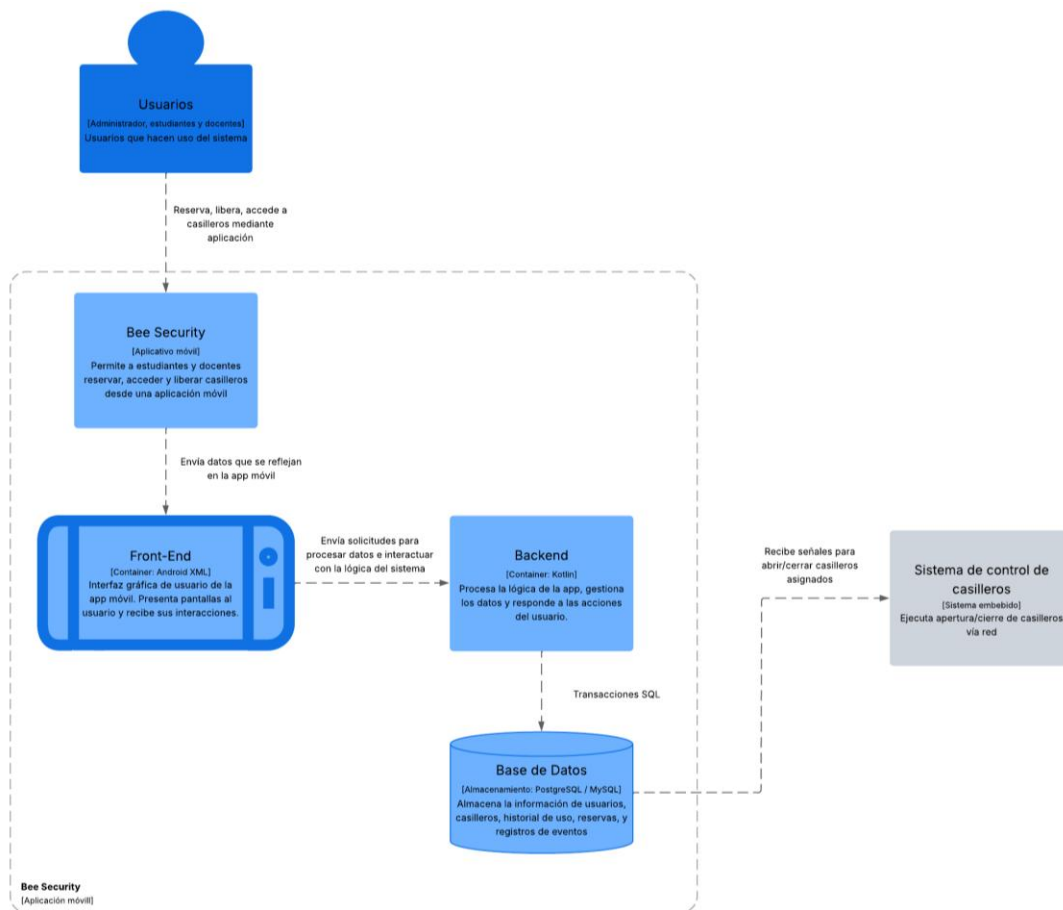
Se describe la arquitectura del sistema utilizando el modelo C4, el cual permite visualizar las distintas capas del sistema, sus componentes y cómo se relacionan entre sí.

- Diagrama de contexto C4



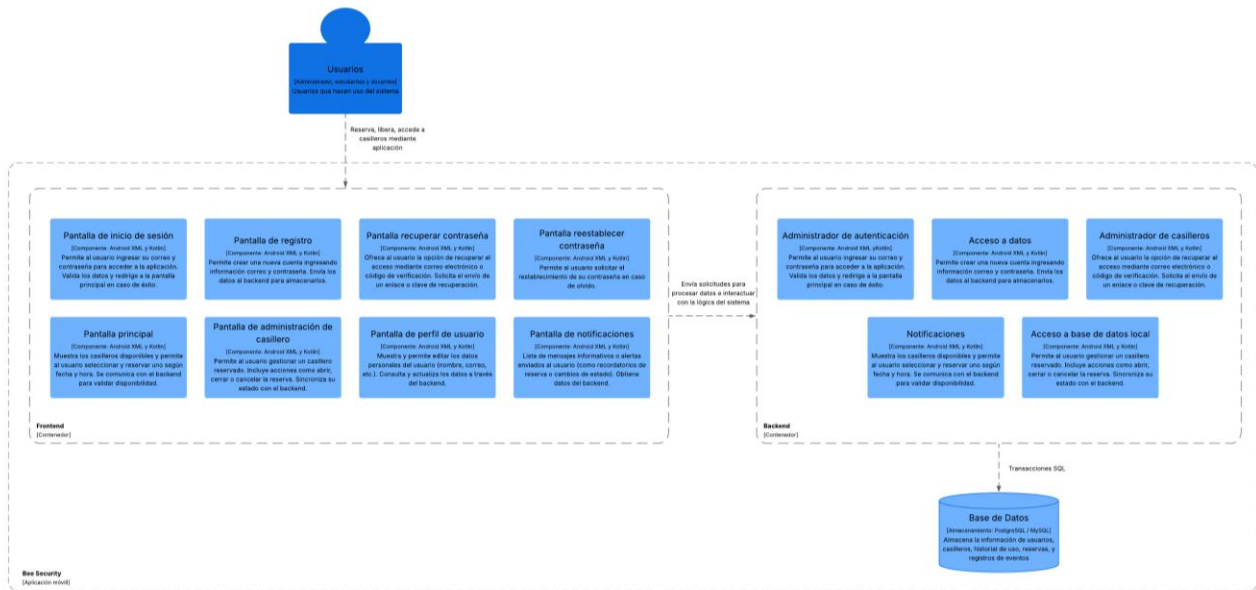


- Diagrama de contenedores C4





- Diagrama de componentes C4



## MODELO DE DATOS

Se describe la estructura de la base de datos a través de un modelo relacional que organiza las entidades, sus atributos y relaciones. Acompañado de un diccionario de datos, este modelo asegura una gestión clara y eficiente de la información del sistema.

- Modelo relacional de la base de datos





- Diccionario de datos

Nombre tabla		Usuario			
Nombre campo	Tipo dato	Llave primaria		Permite valores nulos	
		si	no	si	no
usuario_id	int	X			X
correo	varchar(50)		X		X
nombre	varchar(100)		X		X
contraseña	varchar(20)		X		X
teléfono	varchar(20)		X	X	
foto	varbinary(max)		X	X	

Nombre tabla		Casillero			
Nombre campo	Tipo dato	Llave primaria		Permite valores nulos	
		si	no	si	no
casillero_id	int	X			X
numero_casillero	varchar(5)		X		X
ubicación	varchar(20)		X		X
estado	varchar(20)		X		X

Nombre tabla		IT_Assets			
Nombre campo	Tipo dato	Llave primaria		Permite valores nulos	
		si	no	si	no
reserva_id	int	X			X
usuario_id	int		X		X
casillero_id	int		X		X
fecha_reserva	date		X		X
hora_inicio	time		X		X
hora_fin	time		X		X
estado	varchar(20)		X		X



## PROTOTIPO NO FUNCIONAL

El desarrollo del sistema de casilleros inteligentes para la Universidad EAN (Bee Security) requirió un enfoque estructurado basado en principios de la ingeniería de software, con el fin de garantizar la calidad, escalabilidad y funcionalidad del producto final. Esta sección describe el proceso seguido desde la conceptualización inicial, pasando por el diseño del prototipo no funcional, hasta los lineamientos de arquitectura, implementación y pruebas.

Como punto de partida, se diseñó un prototipo no funcional en Figma que permitió visualizar la interfaz de usuario (UI) de la aplicación móvil. Este prototipo sirvió como guía para definir las funcionalidades principales, identificar los flujos de navegación y validar la experiencia del usuario (UX) antes de iniciar el desarrollo técnico.

- Pantalla: inicio de sesión

Esta pantalla es la puerta de entrada al sistema. Permite al usuario autenticarse mediante su correo institucional y contraseña para acceder a las funcionalidades de la aplicación de casilleros inteligentes.

Cuenta con un campo de texto para ingresar el correo institucional, también incluye un campo para escribir la contraseña correspondiente. Se encuentra disponible una casilla llamada “Recordar contraseña” que permite mantener la sesión activa si así se desea. En caso de haber olvidado la contraseña, se ofrece un enlace que dirige a la pantalla de recuperación. Además, hay un botón titulado “Iniciar sesión” que se utiliza para validar las credenciales ingresadas. Para los nuevos usuarios, se incluye un enlace “Registrarse” que permite crear una cuenta.

*Flujo de navegación:*



Cuando el usuario introduce credenciales válidas, accede directamente al dashboard principal. Si aún no tiene una cuenta, puede hacer clic en “Registrarse” para crear una. En caso de olvidar su contraseña, tiene la opción de restaurarla mediante el enlace correspondiente.



- Pantalla: registro de usuario

En la pantalla de registro, se encuentra un campo de texto destinado al ingreso del nombre de usuario, seguido de otro para establecer una contraseña segura. También se incluye un campo para ingresar el correo electrónico, dando preferencia a que sea institucional. Una vez completados los datos, el usuario puede hacer clic en el botón “Registrarse”, el cual envía la información al sistema para crear la cuenta. Si el usuario ya dispone de una cuenta, puede utilizar el enlace “Iniciar sesión” para volver a la pantalla de login.

*Flujo de navegación:*



Cuando el registro se completa con éxito, el sistema redirige automáticamente al usuario a la pantalla de inicio de sesión. En caso de que ya esté registrado, puede cambiar de pantalla utilizando el botón “Iniciar sesión”. Es fundamental que el sistema valide que el correo electrónico no esté repetido y que todos los campos hayan sido completados correctamente antes de proceder con la creación de la cuenta.



#### Registrarse

  
  
  
  
[Iniciar sesión](#)

- Pantalla: recuperar contraseña

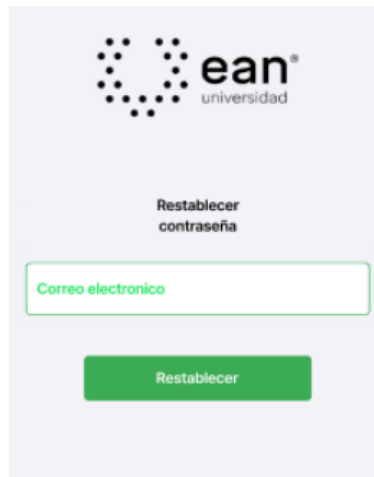
Esta pantalla permite al usuario solicitar el restablecimiento de su contraseña en caso de olvido. Solo necesita ingresar el correo electrónico con el que se registró en el sistema.

#### *Flujo de navegación:*

Cuando el usuario hace clic en el botón “Restablecer”, el sistema simula el envío de un mensaje al correo electrónico proporcionado. Si el correo corresponde a un usuario registrado, el sistema lo redirige a la siguiente pantalla para la creación de una nueva



contraseña. En caso de que el correo no sea válido o no esté registrado en la base de datos, se muestra un mensaje de error al usuario.

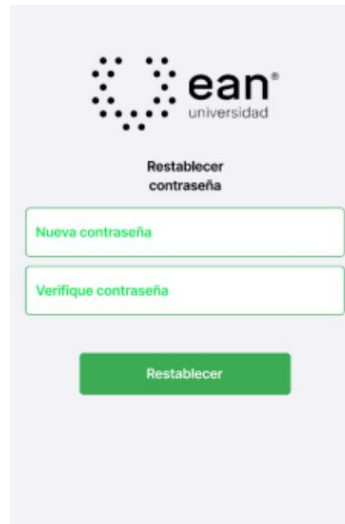


- Pantalla: reestablecer contraseña

Esta pantalla aparece tras el proceso de verificación de identidad. Permite al usuario definir una nueva contraseña para su cuenta. Para ello, se presentan dos campos de texto: uno para ingresar la nueva contraseña y otro para confirmarla, asegurando que ambas coincidan. Además, se incluye un botón “Restablecer”, que permite guardar la nueva contraseña una vez verificada.

*Flujo de navegación:*

El sistema se encarga de validar que ambas contraseñas ingresadas sean iguales. Si coinciden y cumplen con los requisitos de seguridad establecidos, la nueva contraseña se guarda correctamente y el usuario puede regresar a la pantalla de inicio de sesión. En caso contrario, si las contraseñas no coinciden, el sistema muestra un mensaje de error para que el usuario corrija la información.



- Pantalla: inicio

Es la pantalla principal que se muestra al usuario después de iniciar sesión correctamente. Aquí puede ver la disponibilidad de casilleros, reservar uno y acceder a las funciones administrativas de su casillero.

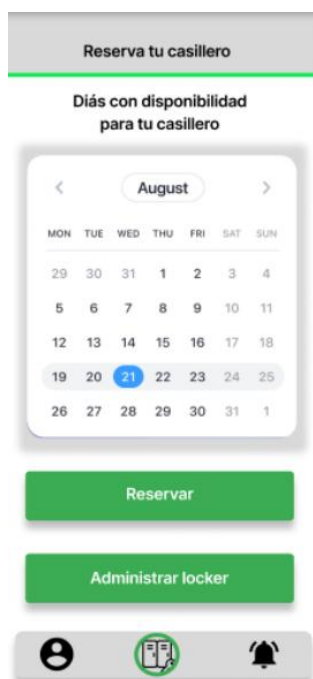
Incluye un calendario interactivo que muestra los días con disponibilidad para reservar un casillero. El usuario puede seleccionar una fecha y confirmar la reserva mediante el botón “Reservar”. También se encuentra disponible el botón “Administrar locker”, que redirige al usuario a una pantalla donde puede consultar los detalles del casillero asignado, cambiarlo o cancelarlo según sus necesidades. En la parte inferior de la pantalla se presenta un menú de navegación con tres íconos: uno para acceder al perfil del usuario, otro para el panel de reservas (pantalla actual) y un tercero para revisar las notificaciones.

*Flujo de navegación:*

Cuando el usuario selecciona un día disponible y pulsa el botón “Reservar”, el sistema registra la reserva correspondiente y bloquea la fecha elegida para evitar duplicaciones. Si el usuario decide pulsar el botón “Administrar locker”, accede a funciones adicionales



relacionadas con el casillero reservado. Por último, los íconos del menú inferior permiten al usuario desplazarse fácilmente entre las distintas secciones de la aplicación.



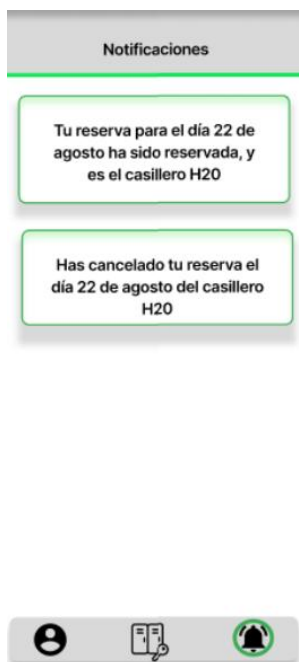
- Pantalla: notificaciones

Esta sección muestra al usuario las alertas importantes del sistema relacionadas con sus acciones dentro de la aplicación, principalmente sobre las reservas de casilleros.

Además del listado de notificaciones, la pantalla cuenta con un menú de navegación inferior que incluye íconos para acceder al perfil del usuario, al panel de reservas y a la sección de notificaciones, que corresponde a la pantalla actual.

*Flujo de navegación:*

Las notificaciones se actualizan de manera dinámica cada vez que el usuario realiza o cancela una reserva. Desde esta pantalla, el usuario puede desplazarse fácilmente a otras secciones de la aplicación mediante el menú de navegación ubicado en la parte inferior.



- Pantalla: perfil de usuario

Esta es la pantalla dedicada al perfil del usuario dentro de la aplicación. Desde aquí, el usuario puede consultar su información básica y cerrar sesión de manera segura. La interfaz incluye una foto de perfil representada por un ícono que puede ser editado al pulsar el ícono de lápiz. También se muestran los campos con el nombre completo y el correo institucional del usuario, los cuales están en modo solo lectura para evitar modificaciones accidentales.

Además de la información personal, la pantalla cuenta con un botón “Cerrar sesión”, que permite al usuario salir de su cuenta de forma segura. En la parte inferior, se encuentra un menú de navegación con tres íconos que permiten acceder al perfil del usuario (pantalla actual), al panel de reservas y a la sección de notificaciones.

*Flujo de navegación:*

Al pulsar el botón de edición, el usuario tiene la opción de modificar su foto de perfil o algunos datos básicos. Si selecciona el botón “Cerrar sesión”, el sistema finaliza la sesión



activa y redirige al usuario a la pantalla de inicio de sesión. A través del menú inferior, el usuario puede cambiar fácilmente entre su perfil, la gestión de reservas y la visualización de notificaciones.



- Pantalla: administración de casillero

Esta pantalla permite al usuario gestionar la información relacionada con el casillero que tiene asignado. Desde aquí puede consultar tanto el número del casillero como abrir y cerrar el casillero cuando lo desee, adicionalmente, tiene la opción de cancelar la reserva.

Incluye un título superior con el texto “Administrar locker”, se presenta una ilustración seguida por los datos específicos del casillero, como el número asignado (por ejemplo, M20) y los botones de apertura y cierre. También se incluye un botón “Cancelar reserva”, que permite al usuario liberar el casillero y eliminar la reserva activa. En la parte inferior se encuentra el menú de navegación, que contiene tres íconos: uno para el perfil del usuario, otro para el panel de reservas (pantalla actual) y otro para acceder a las notificaciones.



*Flujo de navegación:*

Cuando el usuario pulsa el botón “Cancelar reserva”, el sistema libera el casillero asignado y elimina la reserva correspondiente. La información del casillero se actualiza en tiempo real cada vez que se realiza una nueva reserva.



## IMPLEMENTACIÓN

Esta sección describe los componentes de software desarrollados para el sistema, detallando su ubicación dentro de la arquitectura (frontend o backend) y su funcionalidad principal.



Componente de software	Arquitectura	Funcionalidad
LoginActivity	Frontend	Interfaz de inicio de sesión; recoge credenciales y muestra errores al usuario.
RegisterActivity	Frontend	Permite al usuario registrarse en la aplicación.
ForgotPasswordActivity	Frontend	Ofrece recuperación de contraseña mediante correo u otro método.
HomeFragment	Frontend	Pantalla principal después de iniciar sesión; acceso a reservas y perfil.
LockerReservationFragment	Frontend	Interfaz para seleccionar y reservar casilleros disponibles.
LockerManagementFragment	Frontend	Permite abrir, cerrar o cancelar reservas activas de casilleros.
ProfileFragment	Frontend	Muestra la información personal del usuario y permite su edición.
NotificationsFragment	Frontend	Visualiza mensajes del sistema o recordatorios al usuario.
AuthenticationManager	Backend	Lógica de validación de credenciales y gestión de sesión de usuario.
LockerReservationController	Backend	Controla la lógica para reservar, cancelar y verificar disponibilidad.
UserRepository	Backend	Gestiona la lectura y escritura de datos del usuario desde almacenamiento.
NotificationManager	Backend	Crea y administra notificaciones mostradas al usuario.
RoomDatabase	Backend	Base de datos local donde se almacenan usuarios, reservas y notificaciones.

## PRUEBAS Y QA DE SOFTWARE

Para garantizar la estabilidad, funcionalidad y confiabilidad del sistema desarrollado, se implementa un modelo integral de aseguramiento de la calidad del software. Este modelo contempla tanto pruebas manuales como automatizadas, así como un enfoque de despliegue por etapas, con el fin de detectar errores tempranamente y asegurar una experiencia de usuario robusta.

### 1. Estrategia de pruebas

El proceso de pruebas está estructurado en las siguientes etapas:



*Pruebas unitarias:* Se validan componentes individuales del sistema (campos de texto, botones, funciones de validación, etc.), asegurando que cada unidad funcional se comporte de manera independiente según lo especificado.

*Pruebas de integración:* Se verifica la correcta interacción entre distintos módulos del sistema, simulando flujos de navegación completos (por ejemplo, del registro al inicio de sesión y de allí al panel de reservas), con el fin de identificar errores en la comunicación entre componentes.

*Pruebas funcionales:* Se evalúa el cumplimiento de los requerimientos funcionales definidos, tales como la autenticación, el restablecimiento de contraseña, la edición de perfil, la gestión de casilleros y la visualización de notificaciones, validando cada funcionalidad frente a los casos de uso establecidos.

*Validación de datos:* Se prueban los mecanismos de validación en los formularios, incluyendo restricciones de formato (como correos electrónicos), campos obligatorios y coincidencia de contraseñas. Asimismo, se revisa la correcta presentación de los mensajes de error.

*Pruebas de usabilidad:* Se realizan sesiones de evaluación con usuarios reales o testers para identificar dificultades en la navegación, comprensión de elementos visuales y oportunidades de mejora en la experiencia de usuario.

## 2. Despliegue por etapas

Se implementa un pipeline de despliegue continuo (CI/CD) que permite la transición progresiva del software a través de diferentes entornos:

*Entorno de pruebas (QA):* utilizado para validar cambios menores o internos del sistema por parte del equipo de desarrollo.



*Entorno de staging:* simula el entorno real con datos controlados, permitiendo realizar pruebas de aceptación del usuario (UAT) antes de liberar nuevas versiones.

*Entorno de producción:* versión final disponible para los usuarios, que solo recibe cambios una vez validados en las etapas anteriores.



## CONCLUSIONES

### ASPECTOS NOVEDOSOS DESARROLLADOS

Este proyecto de grado propuso y desarrolló una solución tecnológica innovadora para sustituir los casilleros tradicionales de la Universidad EAN por casilleros inteligentes controlados desde una aplicación móvil. La propuesta se destaca por su enfoque en la automatización, conectividad remota y accesibilidad, integrando tecnologías modernas como el desarrollo móvil en Kotlin con Android Studio, microcontroladores ESP32, comunicación en tiempo real, y un backend robusto. El sistema no solo mejora la experiencia del usuario final, sino que también permite una gestión más eficiente por parte de la institución.

### CUMPLIMIENTO DE LOS OBJETIVOS

A lo largo del proyecto, se logró cumplir satisfactoriamente tanto el objetivo general como los objetivos específicos planteados. Se diseñó e implementó una aplicación móvil funcional que permite la reserva de casilleros en tiempo real, se desarrolló un sistema de control que permite la apertura remota de los mismos y se estableció una arquitectura de comunicación eficiente entre todos los componentes del sistema. Además, se logró realizar una evaluación del sistema actual de casilleros de la Universidad EAN, identificando fallas clave como la falta de automatización, poca seguridad, y ausencia de trazabilidad. En contraste, la solución propuesta responde directamente a estas deficiencias, ofreciendo una alternativa moderna, escalable y segura.

### METODOLOGÍA DE LOS ANÁLISIS REALIZADOS

La aplicación de la metodología ágil Scrum permitió un desarrollo iterativo y enfocado en la mejora continua, cada sprint dio lugar a avances tangibles en el prototipado, pruebas y ajustes tanto del software como del hardware. Se realizaron análisis funcionales y técnicos



que guiaron la elección de herramientas y tecnologías, considerando criterios como eficiencia, compatibilidad, facilidad de implementación y bajo consumo de recursos, adicionalmente, se abordaron los aspectos económicos mediante un análisis de costos, lo que permitió establecer un presupuesto estimado para una posible implementación real en el entorno universitario.

### **LIMITACIONES DEL PROYECTO**

Aunque el proyecto logró avanzar significativamente, también enfrentó ciertas limitaciones. Una de las principales fue la falta de implementación física completa del sistema dentro de la infraestructura de la universidad, debido a restricciones logísticas y presupuestales. Asimismo, el sistema requiere de una red Wi-Fi estable y una fuente de alimentación constante para funcionar correctamente, lo que puede representar un reto en zonas con conectividad limitada. Además, algunas funcionalidades avanzadas, como la autenticación biométrica o la integración con sistemas de pago, quedaron fuera del alcance por cuestiones de tiempo y complejidad técnica, aunque podrían ser incorporadas en futuras fases del proyecto.

### **PROYECCIONES Y POSIBILIDADES FUTURAS**

El proyecto abre múltiples posibilidades para futuras mejoras e implementaciones. A corto plazo, se pueden realizar pruebas piloto en espacios específicos de la universidad para evaluar el comportamiento del sistema en condiciones reales. A mediano plazo, se puede trabajar en una interfaz más avanzada que integre notificaciones push, y generación de estadísticas de uso, entre otros. A largo plazo, el sistema podría escalarse a otras instituciones educativas o incluso a espacios públicos y privados que requieran administración automatizada de casilleros, como gimnasios, bibliotecas o centros



comerciales. La arquitectura modular del sistema facilita su evolución, integración con plataformas en la nube y mantenimiento técnico, lo que lo convierte en una propuesta con alto potencial de crecimiento.



## BIBLIOGRAFÍA

- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2020). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347-2376. <https://doi.org/10.1109/COMST.2015.2444095>
- Bormann, C., Ersue, M., & Keränen, A. (2020). Terminology for constrained-node networks. *RFC 7228*. <https://doi.org/10.17487/RFC7228>
- Denis, D., Cruz Flores, D. D., Ferrer-Sánchez, Y., & Felipe Tamé, F. L. (2021). Potencialidades de los celulares inteligentes para investigaciones biológicas. Parte 1: Sensores integrados. *Revista Del Jardín Botánico Nacional*, 42(1), 77–91.
- Fortaleciendo la Analítica en la Institución de Fiscalización Superior de México. (2022). OECD. <https://doi.org/10.1787/edb2c582-es>
- Galvis, J. E., Puerto, K., Galvis, L. N., & Ramírez, J. (2022). Análisis del diseño de interfaces gráficas de usuario para el control y monitorización de sistemas optoelectrónicos aplicados a telecomunicaciones e instrumentación electrónica. *Entre Ciencia e Ingeniería*, 16(31), 33–42. <https://doi-org.bdbiblioteca.universidadean.edu.co/10.31908/19098367.2686>
- Gonçalves, R. M. (2024). Amenazas Digitales: Estrategias Efectivas Para Enfrentar Y Combatir El Cibercrimen. *Law Studies Journal / Revista Novos Estudos Juridicos*, 29(3), 791–820. <https://doi-org.bdbiblioteca.universidadean.edu.co/10.14210/nej.v29n1.p791-820>



- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2019). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645-1660. <https://doi.org/10.1016/j.future.2013.01.010>
- Hilton, M., Sharp, J., & Cruzes, D. (2024). A Framework for Automating the Measurement of DevOps Research and Assessment. *IEEE Transactions on Software Engineering*.
- Hume, D. A. (2023). *Progressive Web Apps: Cross-Browser Web Applications With Progressive Enhancement*. Manning Publications.
- Joseph Soto-Vergel, A., Enrique Herrera-Rubio, J., & Orlando Cardozo-Sarmiento, D. (2022). Medición de parámetros en líneas de transmisión con sistemas embebidos. *Tecnura*, 26(73), 98–114. <https://doi-org.bdbiblioteca.universidadean.edu.co/10.14483/22487638.18131>
- Kranz, M. (2023). *Building the Internet of Things: Implement new business models, disrupt competitors, transform your industry*. Wiley.
- Kovatsch, M., Matsukura, R., & Lagally, M. (2021). CoAP: An application protocol for billions of tiny internet nodes. *IEEE Internet Computing*, 16(2), 62-67. <https://doi.org/10.1109/MIC.2012.29>
- López, M. A., Rivera, D. G., Satizábal, I. C., & Ortíz, W. Y. (2024). Análisis Comparativo de Sensores de Temperatura, Humedad y Luminosidad para su Uso en Sistemas de Producción de Lombricompost. *Entre Ciencia e Ingeniería*, 18(35), 32–40. <https://doi-org.bdbiblioteca.universidadean.edu.co/10.31908/19098367.3045>
- Minaya Macias, R. W., Mendoza Villamar, R. A., Mora Marcillo, A. B., & Intriago Navarrete, J. A. (2024). Análisis De Las Tecnologías Y Normativas en Auditoría Informática: Un Estudio



- De Implementaciones Basadas en Cobit Y Magerit. Revista Científica Arbitrada Multidisciplinaria PENTACIENCIAS, 6(7), 366–380. <https://doi-org.bdbiblioteca.universidadean.edu.co/10.59169/pentaciencias.v6i7.1345>
- Minteer, Andrew. (2021). *Analytics for the Internet of Things*. Packt Publishing.
- Mu, X., & Antwi-Afari, M. F. (2024). The applications of Internet of Things (IoT) in industrial management: a science mapping review. In *International Journal of Production Research* (Vol. 62, Issue 5, pp. 1928–1952). Taylor and Francis Ltd. <https://doi.org/10.1080/00207543.2023.2290229>
- OECD and Inter-American Development Bank. (2023). *Políticas de banda ancha para América Latina y el Caribe Un mAnUAL PARa LA eConomÍA digitAL*. <http://ebookcentral.proquest.com/lib/bibliotecaean-ebooks/detail.action?docID=30229396>.
- Protocolo actuación. Vigilancia sanitaria específica. Pantallas de visualización de datos. (2024). *Gestión Práctica de Riesgos Laborales*, 219, 49–109.
- Rojas, M. J., Pulido, A., & Serrato, Y. I. (2023). Prácticas de seguridad de la información en estudiantes de escuela secundaria en Colombia.
- Sánchez, J.-C., Muñoz, O., Russo, B., Acero, A., & Painedelli, A. (2023). Sistema de sensores de temperatura distribuidos. Pruebas de campo en presas de material suelto. *DYNA - Ingeniería e Industria*, 98(2), 1–6. <https://doi-org.bdbiblioteca.universidadean.edu.co/10.6036/10418>
- Sharma, R., & Kumar, S. (2024). "Enhancing Security and Flexibility in Smart Locker Systems: A Multi-Layered Microservices Approach." *IEEE Internet of Things Journal*, 11(2), 1469-1484. DOI: 10.1109/JIOT.2024.10469610



Shelby, Z., Hartke, K., & Bormann, C. (2022). The Constrained Application Protocol (CoAP). RFC 7252.

Vermesan Ovidiu, & Bacquet Joël. (2020). *Internet of Things-The Call of the Edge Everything Intelligent Everywhere*. <http://ebookcentral.proquest.com/lib/bibliotecaean-ebooks/detail.action?docID=7244731>.

Villafuerte Holguín, J. S., & Alonzo Rezabala, M. E. (2020). Pantallas táctiles y enseñanza del inglés a niños con trastorno por déficit de atención: prácticas idiomáticas y juegos recreativos/Touchscreens and teaching English to children with attention deficit disorder: language practices and recreational games. *Apertura: Revista de Innovación Educativa*, 12(2), 52–73.