UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

TEMA:

“SISTEMA DE AUTENTICACIÓN BASADO EN CÓDIGOS QR PARA ACCESO A SERVICIOS PROPORCIONADOS POR LA UTN”

AUTOR: JEAN CARLOS RODRÍGUEZ VÁSQUEZ

DIRECTOR: FABIÁN GEOVANNY CUZME RODRÍGUEZ

ASESOR: LUIS EDILBERTO SUAREZ ZAMBRANO

IBARRA – ECUADOR

2023

UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004231278		
APELLIDOS Y NOMBRES:	Rodríguez Vásquez Jean Carlos		
DIRECCIÓN:	Ibarra – Tobías Mena y Río Quinindé		
EMAIL:	jcrodriguezv@utn.edu.ec		
TELÉFONO FIJO:	062640372	**TELÉFONO MÓVIL:**	0968401188

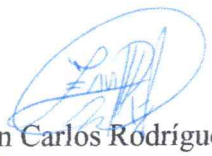
DATOS DE LA OBRA	
TÍTULO:	SISTEMA DE AUTENTICACIÓN BASADO EN CÓDIGOS QR PARA ACCESO A SERVICIOS PROPORCIONADOS POR LA UTN
AUTOR (ES):	Rodríguez Vásquez Jean Carlos
FECHA: DD/MM/AAAA	13 de junio de 2023
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	X **PREGRADO** ☐ **POSGRADO**
TITULO POR EL QUE OPTA:	Ingeniero en Electronica Redes de Comunnicacion
ASESOR /DIRECTOR:	**MSC. Fabián Cuzme**

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 13 días del mes de junio de 2023.

EL AUTOR:



Jean Carlos Rodríguez Vásquez

C.I.: 1004231278

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCAS APLICADAS

CERTIFICACIÓN:

MAGÍSTER CUZME RODRÍGUEZ FABIÁN GEOVANNY, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN CERTIFICA:

Que el presente trabajo de Titulación SISTEMA DE AUTENTICACIÓN BASADO EN CÓDIGOS QR PARA ACCESO A SERVICIOS PROPORCIONADOS POR LA UTN, ha sido desarrollado por el señor Jean Carlos Rodríguez Vásquez bajo mi supervisión.

Es todo cuanto puedo certificar en honor a la verdad.

MsC. Fabián G. Cuzme Rodríguez

C.I: 1311527012

DIRECTOR

DEDICATORIA

A mi padre Galo Rodríguez y a mi madre Jaqueline Vásquez, no puedo dejar de agradecerles su amor, dedicación y esfuerzo constante en mi formación como persona y en mi educación. Han sido un pilar fundamental en mi vida, siempre dispuestos a brindarme su apoyo y guía en cada paso que he dado. Gracias por enseñarme el valor del trabajo, la perseverancia y la honestidad, y por inculcarme los principios y valores que hoy me definen como persona. Este logro no solo es mío, sino también de ustedes, quienes me permitieron alcanzarlo.

Jean Carlos Rodríguez Vásquez

AGRADECIMIENTOS

Agradezco a Dios, Él que siempre ha guiado mis pasos y me ha dado fuerzas para nunca rendirme, hoy culmino una etapa muy importante de mi vida. Sin su amor y protección, no hubiera sido posible alcanzar este logro.

También quiero expresar mi agradecimiento a mi familia, quienes han sido mi mayor apoyo y fuente de motivación durante todo este tiempo. A mis padres, en particular, por su ejemplo de sacrificio, dedicación y perseverancia. Gracias por estar siempre a mi lado y brindarme su amor incondicional.

Asimismo, deseo expresar mi agradecimiento a la Universidad Técnica del Norte y a todos los docentes que me han acompañado a lo largo de estos años de estudio. Gracias por transmitirme sus conocimientos, experiencias y valores, y por ayudarme a desarrollar mis habilidades y potencialidades.

Finalmente, quiero agradecer a mis amigos, quienes han sido mi compañía y apoyo en todo momento. Gracias por estar presentes en mi vida, por sus palabras de aliento, sus consejos y su amistad sincera. Han sido una parte importante de mi proceso de crecimiento y desarrollo.

Jean Carlos Rodríguez Vásquez

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

En cualquier sistema que requiera acceso restringido a información o recursos, la autenticación de usuarios es un proceso crucial. Su función principal es verificar la identidad del usuario que está solicitando el acceso y asegurarse de que tenga los permisos adecuados para realizar la acción solicitada. Si la autenticación no es adecuada, cualquier persona podría ingresar al sistema y realizar actividades maliciosas o no autorizadas, lo que podría comprometer la seguridad y la privacidad de la información. Por lo tanto, la autenticación de usuarios es una herramienta importante para garantizar la seguridad y la privacidad de la información y evitar posibles brechas de seguridad.

El objetivo principal de este proyecto es optimizar el acceso a recursos mediante la autenticación por medio de códigos QR, una solución altamente efectiva que permite compartir información segura entre dispositivos y garantizar que solo los usuarios autorizados puedan acceder a los recursos requeridos. Durante el desarrollo del proyecto se llevó a cabo el diseño, las pruebas de funcionamiento y la implementación de esta tecnología, lo que permitió constatar que la gestión de accesos es un aspecto crítico para la seguridad de las organizaciones y que la implementación de métodos de seguridad puede ayudar a mitigar riesgos, mejorar el cumplimiento y aumentar la eficiencia en toda la institución.

Después de realizar pruebas, se pudo observar que el sistema es capaz de identificar y gestionar las cuentas de manera eficiente, lo que mejora significativamente la gestión y seguimiento de las acciones de los usuarios en los servicios de préstamo de computadores y registro de ingreso. Esto, a su vez, permite tener un método seguro de autenticación, lo que garantiza la confidencialidad de los datos y proporciona una mejor experiencia de usuario.

ABSTRACT

In any system that requires limited access to information or resources, user authentication is a crucial process. Its main function is to verify the identity of the user requesting access and ensure that they have the appropriate authorization to perform the requested action. If authentication is not adequate, anyone could enter the system and perform malicious or unauthorized activities, compromising the security and privacy of the information. Therefore, user authentication is an important tool to ensure the security and privacy of information and avoid possible security breaches.

The main objective of this project is to optimize access to resources through authentication using QR codes, a highly effective solution that allows secure information exchange between devices and ensures that only authorized users can access the required resources. During the project's development, the design, functionality testing, and implementation of this technology were carried out, which allowed for the verification that access management is a critical aspect for organizational security, and that the implementation of security methods can help mitigate risks, improve regulatory compliance, and increase efficiency throughout the institution.

After conducting tests, it was observed that the system is capable of efficiently identifying and managing accounts, which significantly improves the management and monitoring of user actions in computer loan and access registration services. This, in turn, allows for a secure authentication method, ensuring the confidentiality of data, and providing a better user experience.

1. ANTECEDENTES

1.1. Problema

Según Kocausta (2020) “El control de acceso es una técnica de seguridad que regula quién o qué puede ver, usar o acceder a un lugar u otros recursos”.

Este define los siguientes conceptos:

Autenticación: Se debe considerar que el ente a autenticar debe tener previamente una identificación, que se validara en el proceso de acceso a un servicio o inicio de sesión.

Autorización: Este proceso permite determinar los derechos y privilegios de acceso a ciertos recursos, como la revisión de del registro de empleados en un sistema informático autorizado a cierto departamento.

Es la función para determinar los derechos o privilegios de acceso a los recursos. Por ejemplo, el personal de recursos humanos tiene la autoridad para acceder a los registros de los empleados y este protocolo a menudo se formaliza como reglas de control de acceso en un sistema informático.

En la UTN existen varios servicios disponibles para los estudiantes, a los cuales se requiere la verificación de la identidad de este para acceder a ellos, como se menciona en el trabajo de investigación de Orozco & Cerezo (2019), el uso de los códigos QR en carnets estudiantiles permiten mejorar el control de acceso de los estudiantes a la vez que mejora la gestión de la información y su autenticación. (Orozco Toledo & Cerezo Castelo, 2019)

En referencia a dicho por Castro Acuña (2019) menciona que, aun cuando este es un elemento físico que se encuentra expuesto al deterior en la calidad de la imagen códigos QR

tienen incorporada funciones para corrección de errores ante la presencia de ruido o daño parcial. Estos “han ganado popularidad como una alternativa para acceder a diferentes contenidos y realizar trámites sin contacto. Menús en restaurantes, tiquetes de avión, recibos de servicios públicos, boletos de cine, entre otros, ya cuentan con esta tecnología” (el Tiempo, 2021). Los códigos QR se han popularizado para mejorar la experiencia de usuario utilizándolos como enlaces web, pero estas URL también pueden ser modificadas para redirigir a otras páginas web con fines maliciosos.

En el artículo tratado por El Tiempo (2021) se habla también de una de las modalidades de cibercrímenes que ha cobrado fuerza como lo es el QRLjacking, que principalmente se realiza a base de ataques de ingeniería social, vulnerando la seguridad de varios usuarios, como: Ataques de tipo phishing (Qrishing), donde el escanear el código QR abre una página web que suplanta a una legítima, solicitando información confidencial; QRLJacking o secuestro de sesión, en este tipo de ataques se engaña a un usuario de un servicio que use códigos QR para el inicio de sesión, en donde, la victima debe escanear un código QR modificado por los ciberdelincuentes para capturar sus datos personales; Otra manera de vulnerar la seguridad por medio de esta tecnología lo menciona Bardají (2022), que es por medio de la descarga de malware o inyección de código malicioso, en este ataque se realiza la descarga de un software malicioso de manera forzada, dirigiéndose a una página web diseñada para explotar las vulnerabilidades del dispositivo o brechas de seguridad de este, de esta manera, puede realizar múltiples acciones maliciosas; carga e inicio automáticamente de una llamada telefónica a un número que ya está predefinido; Rastreo, con estos códigos es posible saber por dónde navega el usuario en internet y conocer su geolocalización.

“Si los usuarios pudieran verificar que los símbolos QR decodificados son auténticos, es decir, que no han sido generados por terceras partes que están suplantando al emisor real, entonces, se podrían detectar todos estos ataques” (Hernández Encinas & Peinado Domínguez, 2012).

Conociendo el contexto de los problemas de seguridad que se pueden dar con el uso de la tecnología de códigos QR, la solución se plantea a través de la verificación de cierta información contenida en el código que permita identificar si es legítimo y seguro. Una forma de lograrlo es por medio del cifrado de datos con los que posteriormente se pueda autenticar su validez. Este método de autenticación se busca implementar en la UTN para identificar a los estudiantes en el ingreso o adquisición de servicios, sin embargo, considerando que en la UTN cada estudiante tendrá un código QR de identificación, y siendo alrededor de 12500 personas, como menciona Penna & de León (2016) surge la necesidad de al menos un repositorio de identidades con sus correspondientes credenciales. Para el fin de este proyecto se considera la implementación de un sistema que pueda crear códigos QR con información cifrada, que a la vez pueda ser autenticada por el mismo sistema, y con esto brindar a la comunidad universitaria un método de identificación seguro. (Penna & de León, 2016)

1.2. Objetivos

1.2.1. Objetivo General

Implementación de un sistema de autenticación basado en códigos QR seguros para acceso a servicios ofrecidos en la biblioteca de la UTN

1.2.2. Objetivos Específicos

- Analizar las técnicas de criptografía implementadas en códigos QR para credenciales de identificación, forma de almacenar información de códigos QR cifrados y gestores de autenticación seguros para códigos QR.
- Establecer requerimientos de diseño, para un sistema de cifrado y gestores autenticación aplicados a códigos QR, basados en la metodología PHVA.
- Diseñar un sistema para generación de códigos QR cifrados y gestores de autenticación para corroborar el funcionamiento de este en la validación del código QR.
- Realizar pruebas de funcionamiento a los códigos QR generados, induciendo a pruebas que permitan comprobar la autenticación del usuario.

1.3. Alcance

El planteamiento general del proyecto busca corroborar la identidad de los estudiantes por medio de un sistema autenticador de credenciales basado en códigos QR, por medio de encriptación de claves personales en los códigos QR, para lo cual se plantean objetivos específicos que se darán cumplimiento en base a la metodología PHVA, con el fin de dar gestión y control de los procesos que se detallan a continuación.

La primera fase de la metodología determinada para el proyecto es la planificación, en donde se realiza la recopilación de información sobre las técnicas de criptografía, creación de códigos QR, cifrado de códigos QR, almacenamiento, recuperación de la información y establecimiento de un sistema gestor de cogidos para la autenticación de usuario basados en códigos QR; con la consulta y análisis de bibliografía relacionada en artículos científicos y libros.

Teniendo la referencia de las técnicas y de algoritmos para cifrar códigos QR, y determinando el indicado de acuerdo a sus características en la segunda fase que es “Hacer”, se lleva a cabo los cálculos para el cifrado y la cantidad de información cifrada, de acuerdo al tipo de código implementar, que se determina por el tamaño de su matriz y por ende la cantidad de datos que puede almacenar, teniendo en consideración que la cantidad de información y el proceso requerido para descifrar los datos sea los más eficiente al momento de recuperar el mensaje original. Teniendo verificada la parte de diseño teórico, se implementa de forma práctica el cifrado al código, para corroborar obtención del código y la autenticación. La realización de este proyecto tiene como objetivo la implementación del cifrado de códigos QR para credenciales de acceso, en donde, para propósitos administrativos, se requiere de un sistema de autenticación que almacene y valide las credenciales, que permitan el acceso a un servicio.

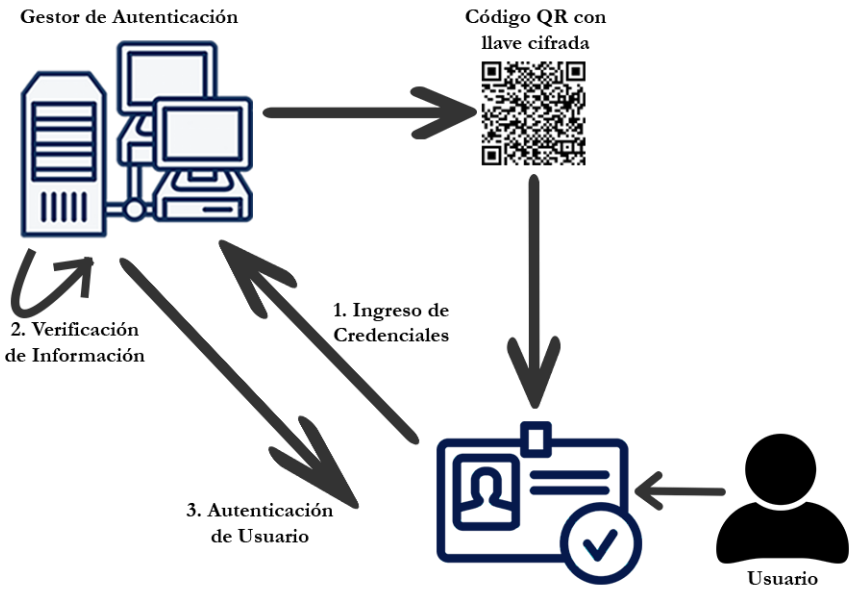
Para determinar que el cifrado es seguro estando ya en la tercera dase de la metodología que es “Verificar”, se requiere comprobar la capacidad que tiene el código QR de resguardar la información y evitar la recuperación por terceros, para lo cual se va a montar un escenario controlado en donde los códigos una vez generados, se verificaran la lectura y validación del código por parte de dispositivos autorizados con ayuda de un sistema de autenticación. Se plantea además el uso de un ataque por fuerza bruta, que requerirá de la obtención de los datos cifrados y un proceso de comprobación de claves a base de prueba y error, para obtener el mensaje original, a partir de los resultados obtenidos del ataque se analiza la confiabilidad de este método de encriptación y el efecto que tiene su uso para resguardar información confidencial y permitir el acceso en un servicio de autenticación seguro.

Para la cuarta y última fase que es “Actuar”, se realizaran pruebas de funcionamiento de toda la arquitectura, generación de códigos seguros y el proceso de autenticación para el

acceso a un servicio, validando funcionamiento óptimo del sistema de acuerdo con parámetros de tiempos de respuesta, tiempo de acceso, lectura, cifrado, autenticación. En caso de encontrar algún inconveniente en el funcionamiento se harán los ajustes necesarios para darle viabilidad al proyecto.

Figura 1

Arquitectura



1.4. Justificación

MINTEL (2021) En su programa Ecuador eficiente y ciberseguro tiene como objetivo proteger a la sociedad frente a las amenazas cibernéticas, generar confianza en el uso del internet y fomentar el desarrollo económico y social basado en el uso de las Tecnologías de la Información y de la Comunicación (TIC). En el Ecuador se entiende a la ciberseguridad como la capacidad del Estado para proteger a las personas, sus bienes activos de información y servicios esenciales ante riesgos y amenazas que se identifican en el ciberespacio.

Este proyecto influye directamente en la ciberseguridad, ya que, como menciona Paredes Valderrama A. (2019) por medio de la autenticación de usuarios, a través de diferentes

mecanismos, podrá demostrar que un individuo especifico es en efecto quien asegura ser al momento en que intenta acceder a un servicio o recurso protegido por un sistema electrónico. Considerando que, la autenticación de los usuarios al momento de querer acceder a un servicio es necesario, la creación de códigos QR con llaves cifradas, permite no solo dar este acceso, sino que da la facultad de asegurar que no hay una suplantación de identidad usando una identificación falsa.

La implementación de esta tecnología genera un impacto positivo en la administración institucional, pues, con el sistema de autenticación se puede llevar un control de acceso a lugares o servicios de los usuarios sin que se pueda dar una suplantación de identidad, permitiendo la optimización de procesos, a la vez que, la comunidad de la UTN se beneficia de la administración que pasa de métodos convencionales donde interviene el humano, a la del sistema gestor de autenticación como lo menciona IBM España (2018) con: organizaciones de todo el mundo están adoptando los nuevos procesos empresariales digitalizados y avances tecnológicos para implementar soluciones de automatización capaces de repetir las acciones humanas. De esta forma, se eliminan tareas rutinarias y las tareas de los empleados evolucionan para ofrecer resultados de mayor valor.

Otro campo donde la implementación de este proyecto impacta es en lo económico, si bien, la creación e implementación de un código QR no representa un gasto elevado para la organización. Esta tecnología ya está en uso en la UTN, pero no tiene la función de autenticación, por lo que se requeriría de la implementación de un nuevo código para los carnets estudiantiles existentes, que, si bien representan un costo adicional, la aplicación del nuevo sistema presenta mayores beneficios.

2. MARCO TEÓRICO

El siguiente capitulo abarca los temas y conceptos necesarios para la comprensión y desarrollo del trabajo de investigación, como son: la ciberseguridad y su importancia, algoritmos de cifrado, códigos QR con sus características y métodos de seguridad, la gestión de acceso con la autenticación de usuarios y un breve desarrollo teórico de lenguajes de programación.

2.1. Ciberseguridad

Este es un término que es cada vez más común de escuchar, siendo de gran importancia para el estilo de vida actual en donde información sensible se maneja por medios tecnológicos, "es el área de las ciencias de la computación encargada del desarrollo y la implementación de los mecanismos de protección de la información y de la infraestructura tecnológica" (Urcuqui et al., 2018).

Urcuqui et al. (2018) hace referencia a lo determinante que es su aplicación para garantizar la confidencialidad, integridad y disponibilidad de los datos, generando prácticas, herramientas, sistemas entre otros, para contrarrestar peligros como los cibercriminales y software malicioso. La ciberseguridad debe estar en constante evolución dado el gran desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) junto a la expansión de las redes IoT y el Big Data, las vulnerabilidades de hardware y software, además, los ataques de día cero.

2.1.1. Delito informático

Postigo Palacios (2020) habla sobre estos delitos en su trabajo y expresa que estos se dan atreves de medios tecnológicos dada la creciente expansión de la tecnología y la

dependencia a ellos, con el fin de vulnerar la confidencialidad, la integridad, la disponibilidad y dar un mal uso de los sistemas informáticos, en donde, el atacante tiene interés malicioso sobre la estructura física y lógica de los sistemas dado el valor de los datos que se almacenan o procesan. Los delitos cibernéticos son difíciles de comprobar, pues, los atacantes están en constante evolución y las técnicas desarrolladas para estos fines se difunden rápido y siendo cada vez más complejo los ataques.

Este accionar mal intencionado por parte de los cibercriminales está lleno de nuevos avances en técnicas de vulneración de la seguridad informática, haciendo uso de tendencias modernas, herramientas novedosas y el desconocimiento de las personas para llevar a cabo su cometido. Surgiendo la necesidad de primero capacitar a las personas a poder distinguir los ataques y aplicando métodos de seguridad en todo el manejo de información digital.

2.1.2. Triangulo CIA

Antes de pretender implementar algún método para brindar seguridad, se debe tener en consideración que es lo que se va a resguardar, Freato (2015) menciona que los principios comúnmente más aceptados en seguridad de TICs son la confidencialidad, la integridad y la disponibilidad de los datos, al cual se le pueden añadir más principios dependiendo de lo que se resguarde o el experto, en cualquier caso, la vulneración de alguno de los principios mencionados significa una violación de la seguridad.

Los principios de confidencialidad (confidentiality), integridad (integrity) y disponibilidad (availability), conformación lo que es el triángulo CIA.

- **Confidencialidad**

Según (Freato, 2015) “una violación de la confidencialidad significa que, en algún lugar, alguna información crítica y confidencial ha sido revelada inesperadamente”.

Un criterio importante para considerar es que, una información a la que puede acceder un grupo de personas puede ser confidencial para otras, esto depende del tipo de información y el nivel de confidencialidad que le otorgue la empresa, algo de suma importancia a considerar para crear estrategias de seguridad.

- **Integridad**

Según (Freato, 2015) es "una violación de la integridad significa que la información se ha corrompido o, alternativamente, el significado de la información se ha alterado inesperadamente".

Este enfoque en la ciberseguridad debe considerar desde el almacenamiento de la información, su manejo y él envió de la misma, significando que en ningún punto de la existencia de la información esta debe ser modificada.

- **Disponibilidad**

Según (Freato, 2015) es "una brecha de disponibilidad significa que el acceso a la información se niega inesperadamente".

Este criterio de seguridad abarca desde los posibles ataques de un cibercriminal para borrar o encriptar información, a los problemas de hardware, en los que, los medios de almacenamiento pueden tener fallos y se requieran respaldos para no perder acceso a la información.

2.1.3. Criptografía

"La criptografía es un método de protección de la información y las comunicaciones mediante el uso de códigos que permite que solo aquellos a quienes está destinada la información puedan leerla y procesarla" (VIU, 2021), esta se puede clasificar tres tipos:

criptografía clave secreta, clave pública y función hash, en la Tabla 1 se describe sus características.

Tabla 1

Tipos de Criptografía

Tipo	Descripción
Criptografía de clave secreta (simétrica)	Utiliza una única clave tanto para el cifrado como para el descifrado; también llamado cifrado simétrico. Se utiliza principalmente para la privacidad y la confidencialidad.
Criptografía de clave pública (asimétrica)	Utiliza una clave para el cifrado y otra para el descifrado. A este método se le conoce como cifrado asimétrico. Se utiliza principalmente para autenticación e intercambio de claves
Funciones hash	Emplea una transformación matemática para "cifrar" la información de forma irreversible, proporcionando una huella digital. Se utiliza principalmente para la integridad de los mensajes

Nota: Adaptado de (VIU, 2021)

2.2. Algoritmos de Cifrado

Según Postigo Palacios (2020) los métodos dirigidos a brindar la privacidad de la información están en continua evolución. En la actualidad, el uso de algoritmos de cifrado y sistemas centralizados de autenticación permiten crear un entorno de seguridad fuerte.

Kaspersky (2022) menciona que los algoritmos de cifrado se emplean para convertir datos en textos cifrado, en donde un algoritmo utiliza una clave que permite modificar los datos originales de una forma predecible; logrando así, que el mensaje generado aun cuando parecen dígitos aleatorios, se puede obtener el mensaje original en texto sin formato mediante la clave de descifrado. En donde los dos métodos de cifrado más comunes son el cifrado simétrico y el cifrado asimétrico.

2.2.1. Cifrado Simétrico

Three Points (2022) desarrolla en su trabajo que los cifrados simétricos o también conocido como cifrado de clave secreta, emplea una única clave que permite cifrar y descifrar los datos, siendo necesario que el receptor tenga dicha clave para poder recuperar el mensaje. El cifrado simétrico es un buen método de autenticación, ya que, el mensaje cifrado no puede recuperarse de ninguna otra forma, siempre y cuando la clave sea secreta. En el caso de que una persona no autorizada obtuviera la clave, significaría un grave problema de seguridad afectando a la confidencialidad como a la autenticación. Esta persona además de tener la capacidad de descifrar cualquier mensaje también puede crear y cifrar nuevos mensajes, considerando que tanto el transmisor como el receptor deben conocer la clave, el mayor problema para este tipo de cifrado es el intercambio seguro de esta, a su vez mientras más participantes conozcan el clave más inseguro se torna el encriptado. La seguridad de los cifrados simétricos recaen principalmente en la clave, pues el algoritmo es de libre conocimiento.

Figura 2

Esquema Criptografía Simétrica

Nota: Fuente (Lewis, 2022)

2.2.1.1. Cifrado en Flujo

Postigo Palacios (2020) presenta en su trabajo que, en este cifrado, los mensajes vienen en un flujo constante y creciente, y se emplea principalmente cuando se desconoce el tamaño del mensaje. En el proceso de cifrado, el mensaje es cambiado bit a bit, que crea un flujo de clave, cifrando el mensaje al aplicar el operador XOR entre los bits del flujo de datos y los de la clave.

2.2.1.2. Cifrado en Bloque

Postigo Palacios (2020) en su libro establece que esta técnica divide el mensaje en fragmentos del mismo tamaño o bloques, cifrando cada uno de ellos. Para conseguir un tamaño igual entre los bloques, el último de estos contiene información que no es útil. Se emplea una clave de cifrado en los extremos de la comunicación, emisor y receptor. Se emplean cuando se tiene un flujo constante de bits, por lo que, es necesario tener todo el mensaje antes del cifrado de la información para usar esta técnica.

- **DES**

Jalca Regalado et al. (2018) en su libro Redes de Computadoras detalla el proceso del cifrado DES, especificando que trabaja con bloques de 64 bits. Comienza haciendo una permutación en donde los bits del bloque de entrada son intercambiados con una tabla para el resultado ser almacenado en 2 registros de 32 bits. Hecho este proceso se empieza a realizar las funciones de cifrado en un ciclo de 16 rondas. El siguiente y último paso es unir los dos bloques y realizar una permutación inversa a la inicial. Dentro de cada ronda los bits de paridad son eliminados con cada permutación almacenando estos resultados en bloques de 28 bits. Como paso final, de la cadena total de 64 bits se seleccionan 48 aplicando una permutación de compresión. En el año 1998 utilizando ataques de fuerza bruta se demostró la brecha de

seguridad de este método de cifrado y se determinó que el problema se debía al reducido tamaño de la longitud de la clave.

Serrato Losada (2019) menciona que después de comprobar que el algoritmo DES no brinda una seguridad eficaz, se desarrollaron variantes, pues, el algoritmo no era malo, sino que su debilidad recaía en la clave, estas variantes utilizan DES, pero asignan claves más largas. Ejemplos de estos nuevos métodos de cifrado son: 3-DES con una longitud de clave de 168 bits y el algoritmo IDEA con una longitud de clave de 128.

- **AES**

Jalca Regalado et al. (2018) menciona que AES a partir del año 2000 fue el algoritmo que reemplazo al en ese entonces usado DES, en el proceso de cifrado AES denomina Estado a el resultado intermedio del cifrado, un array de bytes de cuatro filas. En cada iteración del cifrado se da lugar a una transformación que a su vez está conformada por cuatro adicionales que son: sustitución de bytes no lineal, desplazamientos diferentes en las filas del Estado, mezcla de columnas, adición de la clave. En base al esquema de la clave se obtiene una clave de cifrado en cada iteración. Este esquema consiste en dos operaciones: expansión de clave y selección de clave de vuelta de cifrado.

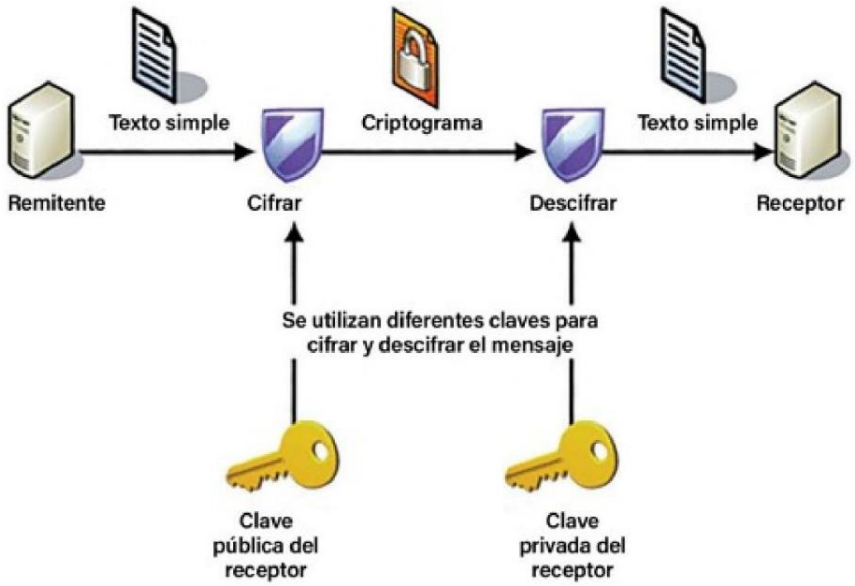
2.2.2. Cifrado Asimétrico

Three Points (2022) menciona en su trabajo que el cifrado asimétrico emplea dos claves para cifrar y descifrar los mensajes, una clave pública y una privada. Estas se relacionan entre sí, pero no son iguales, la clave pública puede ser empleada por cualquiera para cifrar un mensaje, pero este solo puede ser recuperado por un receptor que tenga la clave privada. Una ventaja de este tipo de cifrados es que cada participante tiene su propia clave, lo que permite que el nivel de seguridad se mantenga, aun cuando existan varios participantes en el

intercambio de información. Este tipo de cifrado se emplea en correos electrónicos, firmas digitales y protocolos criptográficos.

Figura 3

Esquema Criptografía Asimétrica



Nota: Fuente (Lewis, 2022)

2.2.2.1. Necesidad de Clave Pública

"La manera más rápida de acceder a la comunicación cifrada no es a través del criptoanálisis; sino hacerse con la clave, sustrayéndola o sobornando a alguien que la conozca o tenga acceso a ella" (Triguero Ortega et al., 2005).

Continuando las ideas de los autores, en su obra abordan los problemas que se desarrollaban entorno a una clave privada para el cifrado de la información, en donde, la solución que surgía era cambiar la clave frecuentemente. Para la comunicación entre las partes, una de estas determina la clave a usar en la comunicación y a su vez debe notificar a las demás partes. Se debe generar un entorno seguro para el intercambio de las claves que idealmente sería un encuentro privado en donde no habría dificultad alguna. En otro caso, cuando una de las partes determina la clave privada y tiene que transferirla a la otra parte, debe ser por un

medio seguro, dando principio al problema, pues, los medios de comunicación no suelen ser tan seguros como se suponen. Se plantea así un problema de distribución de claves, esto junto a problemas de autenticidad de la información.

La solución desarrollada a estos problemas es la criptografía de clave pública, que para realizarse requiere de una clave pública, que es conocida por todos; y otra privada que es conocida únicamente por una persona. La clave pública se usa para cifrarlos mensajes y la privada para descifrarlos, permitiendo de esta manera que cualquiera pueda cifrar y enviar un mensaje y que solo el receptor dueño de la clave privada puede entender el mensaje.

2.2.2.2. Cifrado RSA

"El RSA es un sistema criptográfico que permite enviar mensajes cifrados sin tener que intercambiar una clave privada y es el más utilizado para este fin. También permite realizar firmas digitales" (KeepCoding, 2022).

- **Funciones de Trampilla**

Lake (2018) desarrolla en su trabajo que el cifrado RSA tiene la característica de que su algoritmo tiene una forma de resolución sencilla en una dirección, y requiere grandes cálculos y procesamiento exorbitante en el sentido inverso. Esto debido principalmente a que usa números muy grandes producto de números primos. Al momento de intentar descifrar los mensajes sin el conocimiento de las llaves, hay demasiadas variables, por lo que no se puede calcular una solución en un tiempo razonable.

- **Generar Claves**

Para poder cifrar un mensaje con RSA Lake (2018) menciona que se necesitan 2 claves, las cuales tienen que ser números primos que se establecen previo a una prueba de primalidad, que son algoritmos diseñados para obtener números primos. Los números que se elijan deben

ser muy grandes y no deben ser muy cercanos, pues esta condición permite que sea más fácil de descifrarlos.

Serrato Losada (2019) resume el proceso como:

El servidor genera dos números que son públicos n y e.

El cliente cifra el mensaje M usando la siguiente operación:

$$C = (M^e) mod\ n \quad (1)$$

El servidor recibe el mensaje cifrado C y lo descifra usando la siguiente operación:

$$M = (C^d) mod\ n \quad (2)$$

El servidor genera dos números primos grandes p y q. En la actualidad cada uno tiene 1024 bits, es decir, 309 dígitos decimales. Los multiplica y obtiene el número n.

$$n = p * q \quad (3)$$

Por otro lado, obtiene el indicador de phi de Euler

$$F = (p - 1)(q - 1) \quad (4)$$

Genera la clave pública e que es primo relativo de F. Esto significa que e y F no tienen divisores comunes más allá del 1. Además, el número e tiene que ser mayor que 1 y menor que F, es decir $1 < e < F$.

Siempre se elige el valor para $e = 65537$, con el fin de obtener más eficiencia y seguridad, este es un número primo y es el número 4 de Fermat. Con esta selección el servidor no envía dos números, sólo la clave pública.

Ahora, se genera la clave privada d que cumpla que

$$e * d \; mod \; F = 1 \qquad (5)$$

En las operaciones pasadas se aplicó la aritmética modular. En la cual se prioriza el resto de la división, no el cociente. En la operación $a = b \; mod \; n$, a es el resto de dividir b por n. Por ejemplo, $58 \; mod \; = 4$.

Con todos estos procesos y cálculos para obtener la clave privada y pública, el recurso computacional necesario es considerablemente alto, a la vez que factorizar un número de esta dimensión y poder hacer un ataque por fuerza bruta es demasiado complicado, que, si bien no es imposible, conlleva un tiempo para lograrlo contraproducente.

2.2.2.3. Cifrado ElGamal

"Este método de cifrado se basa en la función unidireccional exponencial discreta. Su creación ayudo al desarrollo de los algoritmos RSA y DSA. Este consta de tres elementos: el generador de claves, el algoritmo y el de descifrado" (Serrato Losada, 2019).

Serrato Losada (2019) también menciona que este algoritmo se lo emplea para firmas digitales y para el cifrado y descifrado de datos. Su seguridad se basa en que se tiene un único sentido y la dificultad de cómputo de su algoritmo discreto. Con la capacidad de cómputo adecuado se puede romper el cifrado, sin embargo, actualmente se es incapaz de romper el cifrado de números grandes en un tiempo razonable.

2.2.3. Funciones de Resumen o hash

Postigo Palacios (2020) desarrolla en su libro que las funciones hash se basan en que una función matemática transforma un valor inicial de entrada a un valor de salida. Permitiendo de esta manera poder hacer un proceso inverso y conocer el valor de entrada al conocer el valor

de salida. La complejidad de transformación a la función de resumen depende de la función matemática. Estas funciones matemáticas son complejas para este cifrado, además n se tiene una clave secreta como entrada.

Una característica importante de este tipo de cifrado es que, siempre se obtiene un tamaño fijo en la salida, independientemente del tamaño del mensaje en la entrada. Se usa comúnmente en la autenticación de mensajes, almacenamiento de contraseñas y para las firmas digitales.

Los métodos empleados para vulnerar este cifrado es a través del análisis del algoritmo matemático, en busca de alguna debilidad lógica y explotarla (criptoanálisis), o a través de un ataque de fuerza bruta con el mensaje codificado producido por el algoritmo.

2.2.3.1. Algoritmo SHA (Secure Hash Algorithm)

"Es una familia de funciones hash publicadas por el instituto Naciones de Normas y Tecnología de Estados Unidos. Se considera un algoritmo muy estable y se utiliza mucho en los sistemas de criptografía" (Postigo Palacios, 2020).

2.2.3.2. Algoritmo MD5 (Message-Digest 5)

Este algoritmo, como lo menciona Postigo Palacios (2020) representa la información en 128 bits, con 32 símbolos hexadecimales. Si bien este algoritmo es bastante utilizado, actualmente se escoge más SHA para hacer las funciones de resumen. MD5 se usa para comprobar que un archivo ha sido descargado sin alteraciones, además de brindar protección contra software mal intencionado. Otro uso que se le ha dado es la de calcular el hash de las claves de los usuarios en sistemas Linux.

2.2.4. Comparación de Métodos Criptográficos

Según Postigo Palacios (2020) los diferentes tipos de cifrado tienen como fin, brindar seguridad a la información que se transmite por medios digitales, cada uno de estos tienen sus propios procesos, algoritmos y otras consideraciones para lograr este objetivo, que, de acuerdo con los medios y necesidades del caso será más beneficioso uno u otro tipo de cifrado. En la Tabla 2 se desarrolla una comparativa breve entre el cifrado simétrico y asimétrico.

Tabla 2

Comparativa del Cifrado Simétrico y el Cifrado Asimétrico

Cifrado Simétrico	Cifrado Asimétrico
Siempre emplea una única clave criptográfica para cifrar y descifrar el mensaje	El cifrado asimétrico emplea dos claves criptográficas. Las claves se conocen como clave pública y clave privada. La clave pública se utiliza para el cifrado, por otro lado, la clave privada se usa para el descifrado.
El espacio de claves se incrementa enormemente conforme aumentan los interlocutores.	El espacio de claves es más manejable cuando los interlocutores son muchos.
AES, DES, 3DES, QUAD	RSA, Diffie-Hellman, El Gamal, DSA
El cifrado simétrico es una técnica sencilla en comparación con el cifrado asimétrico.	El cifrado asimétrico es de naturaleza relativamente complicada, porque se emplean claves criptográficas independientes para realizar las dos operaciones.
Requiere menor tiempo de proceso que el cifrado asimétrico.	Requiere mayor tiempo de proceso que el cifrado simétrico.
El cifrado simétrico es una técnica antigua de cifrado.	El cifrado asimétrico es una técnica de cifrado radicalmente nueva.
Apropiados para el cifrado de grandes cantidades de datos.	El cifrado asimétrico se emplea para intercambiar claves secretas
El cifrado simétrico es una técnica antigua	El cifrado asimétrico es una técnica relativamente nueva

Son vulnerables a ataques de fuerza bruta, siendo fundamental la fortaleza de la clave.	Requiere claves de mayor tamaño para garantizar la seguridad.
No permite autenticar al emisor ya que una misma clave la utilizan dos personas	Permite autenticar a quien utilice la clave privada.

Nota. Adaptado de (Escrivá et al., 2013)

2.3. Códigos Quick Response

Tiwari (2017) habla en su artículo sobre los códigos QR, y menciona, que el significado se traduce como, código "respuesta rápida", a simple vista es matriz 2D, pero cuyo propósito es almacenar una gran cantidad de datos, considerando a los códigos de barras 1D, siendo otro propósito, que este debe decodificarse a alta velocidad sin tener que usar un dispositivo específico para lograrlo, consiguiendo que estos sean leídos por la mayoría de los dispositivos modernos con cámara como portátiles o teléfonos. Estos códigos brindan más ventajas como la interpretación del código en cualquier dirección, corrección de errores (del código dañado se puede recuperar la información original) y diferentes versiones que con diferentes características pueden ser usados dependiendo de la necesidad.

Figura 4

Ejemplo de código QR

Fuente: (Luque Ordóñez, 2012)

2.3.1. Tipos de Códigos Quick Response

La ISO/IEC como parte de su trabajo de estandarización su norma 18004:2015(E) menciona 4 tipos de códigos QR que se muestran en la Tabla 3.

Tabla 3

Tipos de Códigos Quick Response

Tipo	Descripción
Código QR Modelo 1	Fue la especificación original para Código QR y se describe en AIM ITS 97-001 Especificación de simbología internacional-Código QR.
Código QR Modelo 2	Era una forma mejorada de la simbología con características adicionales (principalmente la adición de patrones de alineación para ayudar a la navegación en símbolos más grandes) y fue la base de la primera edición de ISO/IEC 18004.
Código QR	Es muy similar al Código QR Modelo 2, su formato de Código QR difiere solo en la adición de la facilidad para que los símbolos aparezcan en una orientación de imagen especular para la inversión de la reflectancia (símbolos claros sobre fondos oscuros) y la opción de especificar conjuntos de caracteres alternativos al predeterminado.
Micro QR Code	Es una variante de QR Code con un número reducido de módulos superiores y una gama restringida de tamaños, lo que permite almacenar una cantidad pequeña o moderada de datos. representado en un símbolo pequeño, particularmente adecuado para el marcado directo en partes y componentes, y para aplicaciones donde el espacio disponible para el símbolo está severamente restringido.

Nota: Adaptado de (ISO/IEC 18004, 2015)

2.3.2. Estructura Códigos Quick Response

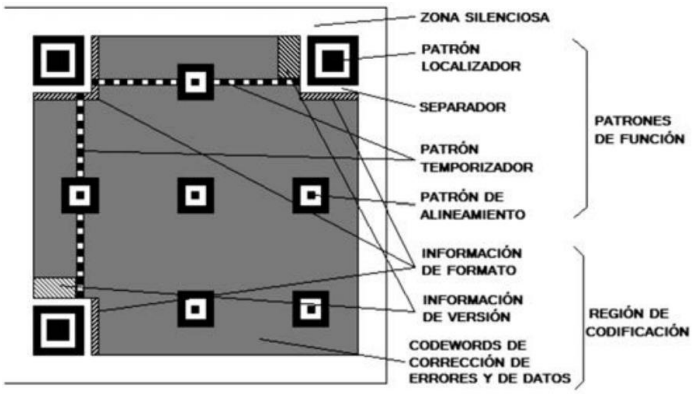
Luque Ordoñez (2012) trata en su documento la estructura que tiene el código QR y las partes principales que este tiene y se muestra en el siguiente texto.

Estos códigos son símbolos bidimensionales formados por cuadros negros y blancos llamados módulos, representando información binaria de ceros y unos. Cada módulo se encuentra en una posición específica cumpliendo una estructura cuadrada, en donde, se distinguen dos bloques de módulos que son: los patrones de función y la región de codificación. No todo el símbolo lleva datos codificados, sino que, también tiene información requerida para decodificar, denominados patrones de función, de los cuales se distinguen los siguientes.

- Patrón de localización: existe por triplicado en el símbolo, ubicados en las esquinas superiores y la inferior izquierda. Sirven para calcular la orientación rotacional del símbolo.
- Patrón de alineamiento: secuencia alternada de módulos blancos y negros que ayuda a calcular las coordenadas de los módulos del símbolo.
- Patrón temporizador: patrón de función que permite resincronizar las coordenadas de mapeo del símbolo ante posibles distorsiones moderadas.
- Separador: patrón de función formado por módulos blancos, cuyo ancho es de un módulo y que separa los patrones localizadores del resto del símbolo.

Figura 5

Estructura genérica de un símbolo.



Nota: Adaptado de Estructura de un símbolo (Luque Ordóñez, 2012)

2.3.3. Corrección de Errores

"Los códigos QR emplean codificación de errores basada en algoritmos de Reed-Solomon, generando un conjunto de codewords de corrección de errores (ECC, Error Correction Codewords) que se añaden a los de datos aportando redundancia" (Luque Ordóñez, 2012).

Castro Acuña et al. (2019) desarrollan en su trabajo la característica de los códigos QR de la corrección de errores, existiendo varios niveles de corrección de errores que depende de la capacidad del código de almacenar información. Estos niveles son L, M, Q y H, donde, el más común es el M con un 15% de capacidad de corrección.

En la normativa ISO/IEC 18004 menciona que existen cuatro niveles para la corrección de errores en base al algoritmo Reed-Solomon, que son nominados como: L, M, Q y H, estos permiten la recuperación de información del símbolo en mayor o menor medida que se presenta en la Tabla 4.

Tabla 4

Capacidad de corrección de error de los códigos QR

Nombre del nivel	Nivel	Corrección de error (aproximado)
Low	Nivel L	7%
Medium	Nivel M	15%
Quality	Nivel Q	25%
High	Nivel H	30%

Nota. Adaptado de (ISO/IEC 18004, 2015)

2.3.3.1. Reed-Solomon

Las características básicas de los códigos Reed Solomon se presentan en la Tabla número 5.

Tabla 5

Características de los códigos Reed Solomon

Características de los códigos Reed Solomon
1. Son un tipo de código cíclico no binario
2. Especialmente útiles en presencia de errores a ráfagas.
3. Presentan una redundancia moderada y una gran capacidad de corrección (se pueden llegar a corregir hasta t símbolos erróneos en un bloque, simplemente añadiéndole 2t símbolos adicionales).
4. Muy utilizados en telecomunicación juntamente con otras técnicas que sean más robustas frente a errores aislados.

Nota. Extracto del trabajo realizado por (Flores Asenjo, 2019)

2.4. Seguridad en Códigos QR

Castro Acuña et al. (2019) expone en su trabajo sobre la seguridad de los códigos QR, si bien, son una herramienta en donde se aplican métodos que permiten una gran corrección de errores y en cierta medida asegurar la correcta llegada del mensaje, estos no cuentan con una forma nativa de seguridad en la información que contienen, para lo cual se aplican otros métodos de seguridad como la criptografía o esteganografía.

2.4.1. Seguridad aplicada al código QR

2.4.1.1. Técnicas de corrección de errores

Castro Acuña et al. (2019) en su trabajo indica que los códigos QR en base al algoritmo Reed-Solomon, a partir del cual se puede establecer el nivel de corrección que brindara el código, en base a la cantidad de información a almacenar y el tipo de código.

2.4.1.2. Técnicas de colores

Por medio de color y texturas que se utilizan en la creación del símbolo se puede lograr que la información sea de acceso más limitado como afirma Castro Acuña et al. (2019) en su

trabajo, además, otra forma de brindar seguridad es con la aplicación de marcas de agua, con las cuales se requiere un lector infrarrojo para obtener el mensaje oculto.

2.4.2. Seguridad de la información contenida en el código QR

2.4.2.1. Método de cifrado AES

"AES por las siglas en inglés de Advanced Encryption Standard. Se caracteriza por su esquema de rondas de bloques de cifrado con claves de mínimo de 128 bits a 256 bits" (Castro Acuña et al., 2019).

2.4.2.2. Método de cifrado RSA

"Algoritmo basado en la factorización de enteros y es asimétrico, el cual utiliza dos claves, una pública y otra privada. Es computacionalmente intensivo para implementar ya que implica cálculos pesados, dado que el tamaño de la clave es 1024 bits" (Castro Acuña et al., 2019).

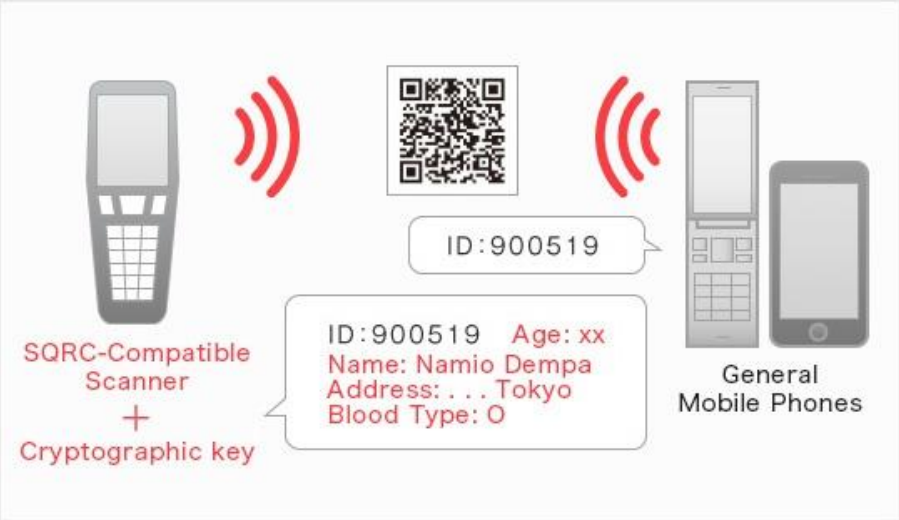
2.4.2.3. SQRC

Por sus siglas en ingles SQRC (Security Quick Response Code), es una variación de código QR que transporta dos tipos de datos: públicos y privados.

Saranya et al. (2016) en su trabajo menciona que, dentro de los códigos QR se maneja varios tipos de información, que pueden llegar a ser de carácter confidencial, estos códigos al ser fáciles de crear y leer se busca la manera de hacerlos más seguros, surgiendo de esta manera una variante de estos códigos como son los códigos de respuesta rápida segura (SQRC), desarrollada por la empresa DENSO WAVE, ayudando a prevenir la falsificación.

Figura 6

Funcionamiento de un Código SQRC



Nota: Fuente (DENSO WAVE, 2022)

2.4.2.4. Esteganografía visual y digital

Se busca ocultar el mensaje a plena vista como indica Castro Acuña et al. (2019), emitiendo un mensaje secreto en forma de QR que se encuentra inmerso en una imagen, la cual, por medio de ciertos procesos para generar la imagen camufla el código haciéndolo imperceptible.

2.4.3. Seguridad Híbrida

2.4.3.1. Algoritmo de Verificación Inteligente

Castro Acuña et al. (2019) menciona que SVA (Smart Verification Algorithm) por sus siglas en inglés, es un algoritmo busca aumentar la privacidad de la información para el acceso de usuarios a sistemas inteligentes IoT, autenticando a las personas por medio de un sistema que genera unas llaves privadas.

2.5. Gestión de Acceso

Según FORTRA (2022) lo que en inglés se conoce como IAM o Identity and Access Management (Gestión de Identidades y Accesos) es muy importante en la seguridad de TI, ya que permite la gestión de identidades digitales y el acceso de usuarios a la información, sistemas y recursos de la entidad. Para este fin, se emplean políticas, programas y tecnologías para reducir los riesgos de autenticación, inicio de sesión y autorización en el acceso a servicios de la organización.

IBM (2022) menciona que la gestión de identidades y accesos es parte indispensable para la seguridad de una empresa. Permite proteger a la entidad contra credenciales de usuario comprometidas y contraseñas no seguras, que son una brecha de seguridad aprovechados por criminales informáticos.

"Un enfoque sólido de la IAM permite a las organizaciones mitigar los riesgos, mejorar el cumplimiento y aumentar la eficiencia en toda la empresa" (FORTRA, 2022).

2.6. Métodos de Autenticación

Evidian (2021) hace una aclaración entre lo que es identificar y autenticar, diferenciando la una de la otra. La identificación de un usuario es un proceso más simple, que se realiza con un número de serie, un ID, entre otros, que no suponen un secreto, siendo un tipo de información pública, en cambio, la autenticación da un mayor nivel de confianza de la identidad de un usuario al requerir la compartición de un secreto.

Entendiendo que una institución puede potenciar el orden, así como la seguridad por medio del control de acceso y la autenticación de usuarios, cabe la duda de cómo realizar este proceso, para lo cual, existen varios métodos, en la Tabla 6 se ejemplifican algunos de ellos.

Tabla 6

Métodos de Autenticación

Métodos	Descripción
Identificador y contraseña	No requiere ninguna modificación de la infraestructura, solo una forma de ingresar la contraseña.
Identificador y OTP (One Time Password)	El usuario posee un "calculador" específico que le proporciona un código de acceso durante un período limitado. Requiere previamente una contraseña.
El Token USB	Los Token USB (con chip) no necesitan lector y pueden conectarse directamente a un puerto para dar acceso.
Tarjeta Inteligente	Las tarjetas inteligentes almacenan información cifrada, requieren un lector para dar acceso.
Tarjeta inteligente con identificador y contraseña	Tarjeta inteligente que almacena el identificador y la contraseña que se complementa con LDAP.
Soluciones biométricas	Se utilizan lectores biométricos para controlar los accesos físicos.
RFID activo	Se aplica el protocolo RFID para identificar al usuario sin contacto físico, incluso a algunos metros de distancia.

Nota. Adaptado de (Evidian, 2021)

2.6.1. Gestión de Claves de Cifrado

"Las claves son análogas a la combinación de una caja fuerte. Si un adversario conoce una combinación segura, la caja fuerte más fuerte no proporciona seguridad contra la penetración" (ATICO34, 2020).

ATICO34 (2020) también menciona que una mala gestión de claves puede comprometer algoritmos fuertes, para lo cual, la gestión de claves de cifrado implica controlar todo el ciclo de vida de las claves criptográficas, desde que son creadas, siguiendo por su uso, almacenamiento, archivo y eliminación.

Esta gestión depende del tipo de cifrado que se emplea en las claves. Como se trató en el apartado 2.2 Algoritmos de Cifrado, se conocen los cifrados simétricos y asimétricos, que, dado su diferencia en el uso de claves, también influye en los procesos de gestión.

Para la mejor comprensión los temas siguientes, existe cierta terminología la cual presenta ATICO34 (2020) que se usa en estos sistemas de llaves, estos se muestran en la Tabla 7.

Tabla 7

Terminología para Sistemas de Llaves

Termino	Definición
Clave de cifrado de datos (DEK)	Es una clave de cifrado que permite cifrar y descifrar los datos.
Clave de cifrado de clave (KEK)	Es una clave de cifrado que permite cifrar y descifrar el DEK.
Interfaz de programa de aplicación de administración de claves (KM API)	Es una interfaz de aplicación que está diseñada para recuperar y pasar de manera segura las claves de cifrado de un servidor de administración de claves al cliente que solicita las claves.
Autoridad de certificación (CA)	Es una entidad que crea claves públicas y privadas, crea certificados, verifica certificados y realiza otras funciones PKI.
Seguridad de la capa de transporte (TLS)	Es un protocolo criptográfico que proporciona seguridad, mediante autenticación mutua, para datos en movimiento a través de una red informática.

Sistema de gestión de claves (KMS)	Es el sistema que aloja el software de gestión de claves.

Nota. Adaptado de (ATICO34, 2020)

2.6.1.1. Gestión de Llaves Simétricas

ATICO34 (2020) menciona que la gestión de las claves simétricas se da desde el momento en que un usuario solicita una clave de acceso, todo este proceso tendría nueve pasos a seguir que se muestran a continuación.

- Un usuario hace una solicitud para acceder a datos cifrados.
- La base de datos, la aplicación, el sistema de archivos o el almacenamiento envían una solicitud de recuperación DEK al cliente (KM API).
- Luego, el cliente (KM API) y KM verifican los certificados de cada uno:
  - El cliente (KM API) envía un certificado al KM para su verificación.
  - El KM verifica el certificado contra la CA para la autenticación.
  - Una vez que el certificado del cliente (KM API) ha sido verificado, el KM envía su certificado a la API de KM para su autenticación y aceptación.
- Una vez que los certificados han sido aceptados, se establece una conexión TLS segura entre el cliente (KM API) y el KM.
- El KM luego descifra el DEK solicitado con el KEK
- KM envía el DEK al cliente (KM API) a través de la sesión TLS cifrada.
- El KM API envía el DEK a la base de datos, aplicación, sistema de archivos o almacenamiento.
- La base de datos (puede) almacenar en caché el DEK en una memoria segura temporal.

- La base de datos, la aplicación, el sistema de archivos o el almacenamiento envían la información de texto sin formato al usuario.

2.6.1.2. Gestión de Claves Asimétricas

Para este tipo de claves ATICO34 (2020) presenta que se tienen cinco pasos, que se muestran a continuación.

- El remitente y el destinatario verifican los certificados del otro:
  - El remitente envía un certificado al destinatario para su verificación.
  - El destinatario luego verifica el certificado con su Autoridad de certificación (CA) o una Autoridad de validación externa (VA) para la autenticación.
  - Una vez que el certificado del remitente ha sido verificado, el destinatario envía su certificado al remitente para su autenticación y aceptación.
- Una vez que el remitente y el destinatario tienen aceptación mutua:
  - El remitente solicita la clave pública del destinatario.
  - El destinatario envía su clave pública al remitente.
- El remitente crea una clave simétrica efímera y cifra el archivo que se enviará. (una clave simétrica efímera es una clave de cifrado simétrica utilizada solo para una sesión)
- El remitente encripta la clave simétrica con la clave pública y envía los datos cifrados con la clave simétrica cifrada.
- El destinatario recibe el paquete y descifra la clave simétrica con la clave privada. Luego, descifra los datos con la clave simétrica.

2.7. Bases de Datos

Pulido Romero et al. (2019) define a las bases de datos como elementos de gran importancia en la informática, llegando a formar parte de muchas áreas dado sus aplicaciones que permiten mejorar la gestión de datos en el mundo moderno en donde se maneja grandes volúmenes de información digital.

2.7.1. Características

Pulido Romero et al. (2019) destaca tres características que son:

Independencia de los datos: En donde, los datos no forman parte de un programa y no dependen alguno para manejar la información, lo que ocasiona que cualquier aplicación puede hacer uso de la información.

Reducción de la redundancia: La redundancia es la duplicidad de los datos. Al reducir la redundancia el espacio de almacenamiento puede ser aprovechado mejor y se soluciona un problema que es la inconsistencia en datos duplicados.

Seguridad: La base de datos puede gestionar el acceso a la información, permitiendo autorizar a ciertos usuarios y restringiendo a otros.

2.7.2. Tipos de Base de Datos

Las bases de datos al ser usadas para diferentes aplicaciones requieren seguir cierta estructura o funcionamiento, estas pueden ser clasificadas según la variabilidad de los datos, según el contenido y por su modelo. En la Tabla 8 se muestra la clasificación por su modelo.

Tabla 8

Clasificación de las Bases de Datos por su Modelo

Modelo	Descripción
Base de datos jerárquica	Este modelo tiene una organización en forma de árbol invertido, en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz y a los nodos que no tienen hijos se les conoce como hojas.
Base de datos de red	En este modelo un mismo nodo puede tener varios padres. Mejora el problema de redundancia de datos del modelo jerárquico.
Base de datos transaccionales	Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas, por lo general, al entorno de análisis de calidad, datos de producción e industrial. Ya que su único propósito es recolectar y recuperar datos a la mayor velocidad posible, la redundancia y la duplicación no representan un problema.
Base de datos relacionales	Usadas para representar problemas reales y administrar datos de manera dinámica. Su principio es el uso de relaciones, que pueden considerarse en forma lógica como conjuntos de datos llamados tuplas.
Base de datos multidimensionales	Son bases de datos diseñadas para desarrollar aplicaciones muy concretas como Cubos OLAP. Técnicamente son muy similares a las bases de datos relacionales. Su diferencia se basa en los campos o atributos de una tabla que pueden ser de dos tipos, o bien, pueden representar dimensiones de la tabla, o incluso pueden representar métricas que se desea aprender.
Base de datos orientada a objetos	Este modelo de reciente creación, propio de los modelos informáticos orientados a objetos, tiene como propósito la tarea de almacenar en la base de datos los objetos completos (estado y comportamiento).
Base de datos documentales	Permiten la indexación a texto completo y, en líneas generales, realizar búsquedas más potentes. Sirven para almacenar grandes volúmenes de información como registros históricos.

Base de datos deductivas	Permite hacer deducciones a través de inferencias y se basa principalmente en reglas y hechos que son almacenados en la base de datos. Mejorando a las bases datos relacionales tanto para responder a consultas recursivas como para deducir relaciones indirectas de los datos almacenados en la base de datos.

Nota. Adaptado de (Pulido Romero et al., 2019)

2.7.3. Sistemas de Gestión de Base de Datos

También conocido como DBMS (Data Base Management System), son una herramienta de software que se utiliza para almacenar, manipular y administrar datos en una base de datos, Naeem (2022) menciona que, permiten diseñar un motor de base de datos de acuerdo las necesidades de análisis e informes. En general un DBMS permiten a los usuarios crear bases de datos, almacenamiento de datos y la actualización de datos a través de consultas SQL.

En la Tabla 9 se recopila información de varios DBMS, para un posterior análisis y selección para implementación en el sistema.

Tabla 9

Características de DBMS

DBMS	Características	Ventajas	Desventajas
MySQL	Propietaria y publica Portabilidad	Fácil de aprender y utilizar multiplataforma Código abierto Fácil configuración Veloz a realizar operaciones	El soporte para disparadores es muy básico No soporta algunas conversiones de datos Los privilegios de las tablas no se borrar de forma automática
Oracle	Propietaria Portable Compatible Alto rendimiento	DBMS popular Oracle ofrece porte técnico	Una mala configuración ofrece resultados desfavorables

		Permite la gestión de múltiples bases de datos	
Postgre SQL	Incluye herencia entre las tablas Incorpora estructuras de arrays	Ahora en costos Instalación limitada Estabilidad Gran capacidad de almacenamiento	Lento en inserciones y actualizaciones Ofrece soporte en línea
SQLite	Dominio público DBMS relacional Algunos lenguajes de programación lo incluyen en sus módulos o bibliotecas	Multiplataforma muchos lenguajes de programación tiene soporte o módulos para sqlite Pequeño tamaño	Su límite es de 2 terabytes su base de datos En ocasiones no permite se exporte a otras bases de datos.
InterBase	Propietario Arquitectura única El lenguaje de procedimientos y trigger es muy potente	Para Microsoft Windows y Linux Permite hacer copias de seguridad en caliente Tiene cercanía al estándar SQL	No permite realizar particiones No es popular
Microsoft SQL Server	Propietario Integra nuevas herramientas Recuperación de datos eficaz y rápida Portabilidad	Para Windows Soporte de transacciones Estabilidad Seguridad Soporte de procedimientos almacenados Entorno grafico	Utiliza muchos recursos computaciones como memoria RAM Es de paga

Nota. Adaptado de (Rosado, 2015)

2.8. Lenguajes de Programación

OpenWebinars (2020) define a un lenguaje de programación como el conjunto de instrucciones a través del cual los humanos interactúan con las computadoras. Los lenguajes de programación interactuar con el computador a través de algoritmos e instrucciones escritas en una sintaxis que la computadora convertirá en lenguaje de máquina.

Moreno Pérez (2014) comenta en su libro que, en la actualidad, es común el usar programas, desde editores de texto, juegos, reproductores multimedia entre muchos otros, sin

embargo, estas piezas de software no funcionan con el lenguaje que manejamos los humanos, sino que, tienen un a naturaleza binaria.

Se distinguen de esta manera tres clases de lenguajes de programación según rockcontent (2018):

El lenguaje de máquina que es el más primitivo de los códigos y se basa en el código binari0, todo en 0 y 1. Este lenguaje es utilizado directamente por máquinas o computadora

Lenguaje de programación de bajo nivel que es un lenguaje un poco más fácil de interpretar, pero puede variar de acuerdo con la máquina o computadora que se esté programando.

Lenguajes de programación de alto nivel, en esta categoría se encuentran los más utilizados. Se usan palabras del inglés lo cual facilita que una persona pueda intervenir más fácil que en los dos anteriores.

Los lenguajes de programación más usados son: Java, lenguaje C, Pyton, C++, C#, Visual Basic. NET, SQL, PHP, Ruby, lenguaje de programación en R, Rust, TypeScript, Swift, Perl, lenguaje de programación en GO, Kotlin, Scheme, Erlang, Elixir, Pascal, Haskell, Objetive-C, Scala, Lava.

2.9. Android Studio

Talently (2022) menciona que es un entorno de desarrollo integrado (IDE) para el desarrollo de herramientas y aplicaciones de sistema operativo Android, usando para la creación de código el lenguaje de programación java, pero soportando lenguajes como: Kotlin, NDK y C++.

Este editor de código brinda un gran impulso al desarrollo de aplicaciones, existiendo grandes comunidades, repositorios y bibliotecas que se pueden aprovechar para generar un

proyecto, impulsado a su vez por la característica del sistema operativo Android que es libre, gratuito y multiplataforma.

2.10. Trabajos Relacionados

La facilidad para crear y leer códigos QR induce a que sean usados para diversos fines, de entre los cuales, la encriptación de información confidencial se ha venido desarrollando y usando en diferentes entornos. Los siguientes trabajos son casos en los que los códigos QR tienen una aplicación de seguridad por medio de encriptación.

- A Novel Approach for the Detection of OMR Sheet Tampering Using Encrypted QR Code

Un enfoque novedoso para la detección de la manipulación de hojas OMR mediante un código QR cifrado, es la traducción ha español, en donde, Tiwari & Sahu (2014) mencionan en su trabajo que en los exámenes que usan OMR (sistema de reconocimiento óptico de marcas) no son totalmente seguros en la manipulación de los datos, que es un problema que afecta exámenes competitivos de alto nivel, en donde, una tercera persona puede manipular los resultados para favores ilícitamente a él examinado. Con el fin de mejorar la seguridad del sistema OMR se plantea el uso de códigos QR encriptados, de manera que cualquier alteración de datos pueda ser identificada. (Tiwari & Sahu, 2014)

- A Secure QR Code System for Sharing Personal Confidential Information

Un sistema seguro de códigos QR para compartir información personal confidencial, es la traducción ha español, en donde, Ahamed & Asiful Mustafa (2019) indican en su trabajo que la confidencialidad de la información es un reto cada vez más complicado de lograr, los autores hacen especial énfasis en la falsificación de información confidencial. Estos datos idealmente deberían poder ser autenticados, para lo cual se plantea el uso de un sistema de códigos QR

seguros (SQRC) para que un usuario pueda mantener la información oculta y segura, encriptando información en el código por medio de algoritmo RSA que a su vez permite autorizar información mediante la validación del código. (Ahamed & Asiful Mustafa, 2019)

- Diseño de herramientas computacionales asociadas al modelo algorítmico AES, para la encriptación de datos en módulos Quick Response (QR) para plataformas Android

Hipólito Martínez (2015) destaca en su trabajo el valor de la información, mucha de esta debe permanecer oculta o con exceso restringido, por lo que, se requiere un mecanismo de seguridad. Dado las grandes ventajas de los códigos QR, plantea la encriptación de datos dentro del código y que puede ser interpretada por una aplicación para sistema operativo Android.

La idea central de los trabajos anteriormente citados es el manejo seguro de información confidencial por medio de una herramienta muy útil y sencilla como son los códigos QR, teniendo en cuenta esto, se plantea para el presente trabajo aplicar este concepto para la administración de usuarios en la UTN, en donde, se pueda autenticar la identidad de las personas y de esta manera gestionar de manera ágil y segura los recursos de esta entidad.

3. DISEÑO

El presente capitulo describe los pasos a seguir para el diseño del sistema de autenticación, partiendo de la definición del funcionamiento para un posterior análisis de los requerimientos, en donde, se establecerá el software a utilizar para la base de datos, la programación y la creación de la aplicación móvil.

3.1. Descripción del Sistema

El sistema que se plantea se enfoca en la autenticación del usuario por medio de una aplicación móvil, usando para esto credenciales, además, de generar claves para dar acceso a un servicio, la cual, se logra con la lectura de un código QR que contiene la clave cifrada.

3.1.1. Situación Actual

El escenario en donde se lleva a cabo las pruebas del sistema sera un ambiente controlado para comprobar su funcionamiento, el cual, pretende mejorar la administración y gestión de los usuarios de la biblioteca.

Para que un estudiante haga uso de los servicios brindados por la universidad, este debe validar su identidad, para lo que, se suele usar como método la presentación y retención del carnet del estudiante, teniendo gran parte del control de acceso el personal administrativo.

El manejo de la información personal en muchos casos es manual en una hoja de registro, que no permite asegurar la integridad, confidencialidad y disponibilidad de los datos, lo que, en si es un problema de seguridad.

El carnet físico para la validación de la identidad de una persona se ha usado durante muchos años, y si bien, cumple su función está sujeta a problemas de la portabilidad y la limitada información a la que se puede acceder, que se ve resuelto por métodos digitales que

son prácticos, permiten centralizar la información, gestionarla de forma más eficaz y manejar más información.

En la biblioteca universitaria se existen servicios tecnológicos en los cuales pueden ser implementados los servicios, en una entrevista previa con la Directora de Biblioteca se conocieron los servicios que esta brinda a la comunidad universitaria, de los cuales, se plantea el funcionamiento del sistema para el registro de los estudiantes en el ingreso de la biblioteca y la activación de computadores.

3.1.2. Descripción de Funcionamiento

- Desde el punto de vista del usuario

El usuario debe contar con un dispositivo móvil en el que se encuentre instalada la aplicación del sistema, la misma aplicación requiere un previo inicio de sesión con usuario y contraseña.

La aplicación hace una solicitud de un código SQRC para poder acceder a un servicio, el código se colocará frente a un lector que validara la información, y permitirá o denegara el uso del servicio requerido.

- Desde el punto de vista del administrador

El sistema consta de una aplicación móvil que establece una conexión con la base de datos para crear y almacenar las credenciales de inicio de sesión.

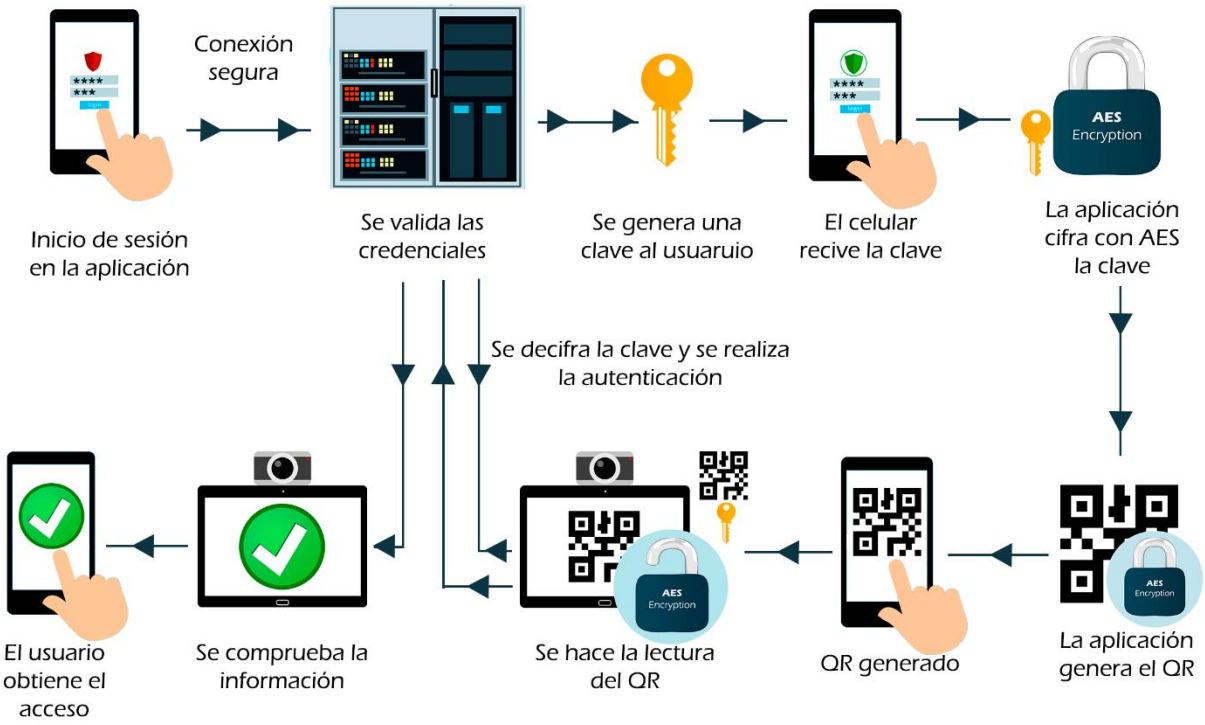
Cuando se hace la solicitud desde la aplicación, un programa crea una clave que permitirá el acceso a un servicio, esta clave se almacena en la base de datos asociada a el usuario, la cual, se envía a la aplicación por medio de un canal seguro, esta clave se cifra y se

genera un código QR, el cual se escanea en un lector que se comunica con el programa de gestión de claves para descifrar el clave dentro del QR y dar o denegar el acceso al servicio.

3.1.3. Diagrama del Sistema

Figura 7

Diagrama del Sistema



3.2. Análisis de Requerimientos

Para establecer los requerimientos del sistema se hace uso de un estudio de investigación en trabajos relacionados que fundamenten los parámetros a cumplir en el proyecto.

La recolección y tratamiento de información se basa en el estándar ISO/IEC/IEEE 29148 Ingeniería de sistemas y software - Procesos del ciclo de vida – Requisitos ingeniería, para determinar requerimientos de acuerdo con su prioridad y necesidad en el proyecto, aspectos a considerar para el correcto desarrollo y el cumplimiento de los objetivos planteados.

3.2.1. Nomenclatura de Requerimientos

Para una distinción más rápida y ágil de los diferentes requerimientos se empleará nomenclaturas que se presentan en la Tabla 10 y que hacen referencia a diferentes secciones a considerar en el proyecto.

Tabla 10

Nomenclatura de Requerimientos

Descripción	Abreviatura
Requerimientos de Stakeholders	StSR
Requerimientos de Sistema	SySR
Requerimientos de Arquitectura	SrSH

3.2.2. Requerimientos de Stakeholders

En este apartado se definen los requisitos del sistema, tanto operacionales como de funcionamiento, a partir de los cuales, se fundamenta el diseño y se establecen los medios para cumplir con los objetivos del proyecto. En la Tabla 11 se evalúa los requerimientos de stakeholders.

Tabla 11

Requerimientos de Stakeholders

Requerimientos de Stakeholders (StSR)				
Número	**Descripción**	**Prioridad**		
		Alta	**Media**	**Baja**
Requerimientos Operacionales				
StRS1	Conexión segura para envío de datos	X		
StRS2	Autenticación de Usuarios	X		
StRS3	El sistema mejorara la seguridad en la administración de las claves por medio del cifrado	X		
StRS4	El sistema permite gestionar las claves de cifrado	X		

StRS5	El sistema tiene la capacidad de leer códigos QR y administrar su información	X		
Requerimientos de Usuarios				
StRS6	Los usuarios manejaran el acceso por medio de una aplicación móvil	X		
StRS7	La aplicación permitirá administrar la clave y contraseña	X		

3.2.3. Requerimientos del Sistema

Los requerimientos del sistema se obtienen en base a las funciones que debe desempeñar, así como, las limitaciones que tendrá. Entre estos se analiza requerimientos de performance, interfaces, estados y físicos.

Tabla 12

Requerimientos del Sistema

Requerimientos del Sistema (SySR)				
Número	**Descripción**	**Prioridad**		
		Alta	**Media**	**Baja**
Requerimientos Performance				
SyRS1	Comprobación de información en un tiempo corto		X	
SyRS2	Se debe dar acceso durante toda la jornada de apertura de la biblioteca – 10 horas		X	
SyRS3	El sistema permite gestionar a los usuarios identificándolos por medio de códigos QR	X		
Requerimientos de Interfaz				
SyRS4	Cámara para lectura del código QR	X		
SyRS5	Aplicación móvil con la que se puede identificar al usuario	X		
SyRS6	Protocolo de comunicación segura entre la aplicación y la base de datos	X		
SyRS7	Conexión a Internet	X		
Requerimientos de Uso				
SyRS8	Facilidad de uso de la aplicación móvil		X	
SyRS9	Debe permitir la gestión de varios usuarios	X		
Requerimientos de Seguridad				

SyRS10	Algoritmo de cifrado para encriptar las claves	X
SyRS11	Seguridad en la comunicación entre la aplicación móvil y la base de datos	X
SyRS12	Inicio de sesión en la aplicación móvil con ingreso de usuario y contraseña	X
SyRS13	Acceso a la base de datos mediante credenciales	X
Requerimientos de Modo y Estado		
SyRS14	El sistema diferencia dos estados que son: la lectura de los datos y la comprobación de los mismos	X

3.2.4. Requerimientos de Arquitectura

Los requerimientos de arquitectura establecen las necesidades para cumplir con los requerimientos del sistema, entre los cuales, se analizan requerimientos de software, diseño y lógicos que se presentan en la Tabla 13.

Tabla 13

Requerimientos de Arquitectura

Requerimientos de Arquitectura (SrSH)				
Número	**Descripción**	**Prioridad**		
		Alta	**Media**	**Baja**
Requerimientos de Software				
SrSH1	Uso de software libre			X
SrSH2	Base de datos rápida y de fácil gestión		X	
SrSH3	Aplicación compatible con versiones antiguas del S.O. móvil			X
SrSH4	Lenguaje orientado a objetos		X	
Requerimientos de Diseño				
SrSH5	Almacenamiento y actualización de credenciales	X		
SrSH6	Creación, almacenamiento y eliminación de las claves de acceso	X		
SrSH7	Creación del código QR en la aplicación		X	

SrSH8	El sistema debe permitir la lectura de la información del código QR y la recuperación del mensaje al descifrar la información	X	
	Requerimientos Lógicos		
SrSH9	La información se manejará de forma segura almacenando información en la base de datos, a la que se hará consultas por medio de la aplicación móvil		X

3.3. Elección de Software

La elección de software se basará en los requerimientos establecidos, buscando la herramienta o método que permita cubrir esta necesidad de todas las partes de software implicadas en el sistema.

- **Base de Datos**

Existen diversas bases de datos para realizar consultas y operaciones en los datos almacenados. Entre ellas se analizará cinco que son bastante usadas en la actualidad las cuales son: MySQL, MongoDB, Oracle, PostgreSQL y Microsoft SQL Server. En la tabla 14, se presenta una comparación entre ellas en cuanto a los requerimientos mencionados.

Tabla 14

Elección de la Bases de Datos

Bases de Datos	**Requerimientos**						**Valoración**
	SyRS1	StRS1	SrSH2	SyRS2	SyRS6	SrSH9	
MySQL	X	X	X	X		X	5
MongoDB	X	X				X	3
Oracle					X	X	2
PostgreSQL			X	X		X	3
Microsoft SQL Server					X	X	2
X = cumple el requerimiento							

Una parte importante en el sistema es la base de datos, que considerando los requerimientos que se plantearon en la sección anterior y el análisis expuesto en la tabla 14 se decide hacer uso de MySQL dado su facilidad de uso, robustes y comunidad que permite un trabajo óptimo. Esta base de datos estará alojada en un hosting gratuito en internet y sera administrada por medio de phpMyAdmin.

- **Servidor de Hosting**

Para la realización de este sistema se requiere de un hosting que cuenta con cierto nivel de seguridad, base de datos con su gestor y un servidor web. En general, los hostings gratuitos en línea suelen tener algunas limitaciones en comparación con los planes de pago, como menor cantidad de espacio en disco, ancho de banda limitado, y menos características avanzadas. Sin embargo, pueden ser una buena opción para proyectos pequeños o para pruebas y experimentos. En la tabla 15 se muestra una recopilación de información de hostings gratuitos.

Tabla 15

Elección del Servidor de Hosting

Servidores Hosting	**Requerimientos**											**Valoración**
	SyRS1	SyRS2	StRS4	StRS7	SyRS1	SyRS6	SyRS7	SyRS13	SrSH5	SrSH6	SrSH9	
InfinityFree		X						X	X			3
AwardSpace	X	X		X	X	X	X	X	X	X	X	10
Byet.host		X	X				X	X				4
Freehostia		X					X	X				3
000webhost	X	X	X	X	X	X	X	X	X	X	X	11
LucusHost	X	X	X	X	X	X	X	X	X	X	X	11
X = cumple el requerimiento												

La elección del servidor de hosting es LucusHost que proporciona una versión de gratuita con características avanzadas, como mayor cantidad de espacio en disco, ancho de

banda ilimitado, soporte para múltiples dominios, SSL gratuito, entre otros, además, cuenta con un panel de control fácil de usar y una interfaz intuitiva que permite una gestión rápida y sencilla del sitio web y ofrece soporte para tecnologías como PHP, MySQL, y cPanel.

- **IDE para la programación de la Aplicación Móvil**

Previo a la elección del entorno en el que se realizara la aplicación móvil se realiza un análisis entre diferentes IDEs para el desarrollo de aplicaciones Android, estos serán: Android Studio, Eclipse, Visual Studio que se muestran en la tabla 16.

Tabla 16

Elección del IDE para el Desarrollo de la Aplicación Móvil

Bases de Datos	**Requerimientos**						**Valoración**
	StRS5	SyRS6	SyRS8	SyRS11	SrSH1	SrSH3	
Android Studio	X	X	X	X	X	X	6
Eclipse		X	X	X	X	X	5
Visual Studio		X	X	X			3
X = cumple el requerimiento							

El IDE Android Studio está especialmente diseñado para el desarrollo de aplicaciones de Android, proporciona una gran cantidad de herramientas y características para el desarrollo, incluyendo un editor de código fuente, un depurador, una herramienta de construcción de proyectos, y una interfaz de usuario para diseñar y previsualizar las pantallas de la aplicación, ofrece una gran cantidad de recursos y documentación para el desarrollo, es gratuito y de código abierto y con documentación que se puede usar en el desarrollo de la aplicación.

- **Lenguaje de Programación**

El lenguaje de programación sera necesario para el desarrollo de la aplicación móvil, el lector QR, este lenguaje deberá aportar al cumplimiento de los requerimientos planteados, la elección del lenguaje de programación se dará entre Java, C / C++, HTML / CSS / JavaScript, Python, Ruby y Groovy. El resultado del análisis se muestra en la tabla 17.

Tabla 17

Elección del Lenguaje de Programación

Bases de Datos	**Requerimientos**						**Valoración**
	SyRS10	SrSH1	SrSH3	SrSH4	SrSH7	SrSH8	
Java	X	X	X	X	X	X	6
C / C++	X	X	X	X			4
HTML/CSS/JavaScript		X	X				2
Python	X	X	X	X		X	5
Ruby	X	X	X	X			4
Groovy	X	X	X	X			4
X = cumple el requerimiento							

El lenguaje de programación escogido para realizar el trabajo es JAVA, considerando su portabilidad que permite que el código Java se ejecute en diferentes plataformas sin la necesidad de recompilarlo. Además, Java es un lenguaje seguro que cuenta con características de seguridad integradas (Cifrado AES). Es fácil de aprender y utilizar, y cuenta con una gran comunidad de desarrolladores y un amplio soporte. Por último, la posibilidad de desarrollar aplicaciones para varias plataformas, Java es un lenguaje de programación muy versátil, y los IDEs elegidos lo soportan y proporcionan información suficiente para la programación. Este lenguaje se utilizará en la programación de la aplicación móvil y en la aplicación que permitirá la lectura del código QR.

Otro lenguaje que se empleara es php, ya que, este es una característica que tiene en funcionamiento el hosting escogido, además de que se usara la herramienta phpMyAdmin para gestionar la base de datos y ejecutar sentencias que permitan consultar, escribir y leer datos necesarios para la gestión de credenciales.

- **Método de Seguridad**

El sistema plantea poder dar seguridad a los datos por medio de la encriptación de la claves, por lo que es indispensable hacer una elección que cumpa este cometido, para tener una elección acertada se toma en cuenta los requerimientos planteados anteriormente. En la tabla 18 se muestra el resultado del análisis.

Tabla 18

Elección del Algoritmo de Cifrado

Bases de Datos	Requerimientos				Valoración
	SyRS1	SyRS10	SrSH3	SrSH8	
AES	X	X	X	X	4
RSA	X	X	X	X	4
ChaCha20	X	X			2
Triple DES (3DES)	X	X	X	X	4
Curva Elíptica	X	X			2
Blowfish	X	X	X	X	4
X = cumple el requerimiento					

AES (Advanced Encryption Standard) es un algoritmo de cifrado simétrico que ha demostrado ser seguro y eficiente en una amplia variedad de aplicaciones de cifrado. Es ampliamente utilizado en la actualidad y es compatible con la mayoría de los sistemas operativos y dispositivos. AES utiliza bloques de cifrado de 128 bits, 192 bits o 256 bits, lo que lo hace muy difícil de romper mediante ataques de fuerza bruta. Además, AES es rápido y

eficiente, lo que lo hace adecuado para aplicaciones que requieren cifrado en tiempo real. Por lo tanto, la elección de AES es una elección sólida para la mayoría de las aplicaciones de cifrado y proporciona un alto nivel de seguridad y eficiencia.

3.4. Elección de Hardware

En este apartado se muestra las consideraciones que se tiene para cubrir la necesidad de hardware para el funcionamiento del sistema, esta elección permitirá la lectura del código QR y la comprobación de su información.

- **Lector de QR**

El procesamiento y comprobación del código QR se dará en el servidor, por lo cual, es necesario contar con un medio para capturar el código y enviarlo al servidor para su análisis. En la tabla 19 se muestra el análisis de los requerimientos aplicables al lector QR.

Tabla 19

Elección del Lector QR

Bases de Datos	Requerimientos			Valoración
	StRS5	SyRS1	SyRS4	
Escáner Portátil	X	X	X	4
Escáner de Sobremesa	X	X	X	4
Escáner de Mano	X	X	X	4
Cámara Web	X	X	X	4
X = cumple el requerimiento				

Si se trata de un proyecto que requiere la lectura de códigos QR en un entorno de escritorio, una cámara web puede ser una opción adecuada y conveniente. Las cámaras web son fáciles de usar, no requieren software o hardware adicional y pueden ser utilizadas en una variedad de dispositivos. Sin embargo, es importante tener en cuenta que la calidad de la

cámara web puede afectar la capacidad de lectura de los códigos QR y puede ser necesario ajustar la iluminación y el enfoque para obtener mejores resultados.

3.5. Diseño

En este apartado se presenta las características que tendrá el sistema, en base a los requerimientos establecidos y la decisión tomada en la elección de hardware y software, además, el diseño permitirá tener una visión clara del funcionamiento del sistema para su posterior implementación.

3.5.1. Diagrama de Casos de Uso

El diagrama de casos de uso es una herramienta utilizada para representar visualmente cómo las entidades de un sistema interactúan y qué funciones realiza el sistema en respuesta a esas interacciones. Permite comprender mejor el comportamiento del sistema y cómo interactúa. Además, el diagrama de casos de uso también se utiliza para documentar los requisitos funcionales del sistema, lo que ayuda a definir los requisitos del sistema y a garantizar que se cumplan los objetivos del proyecto.

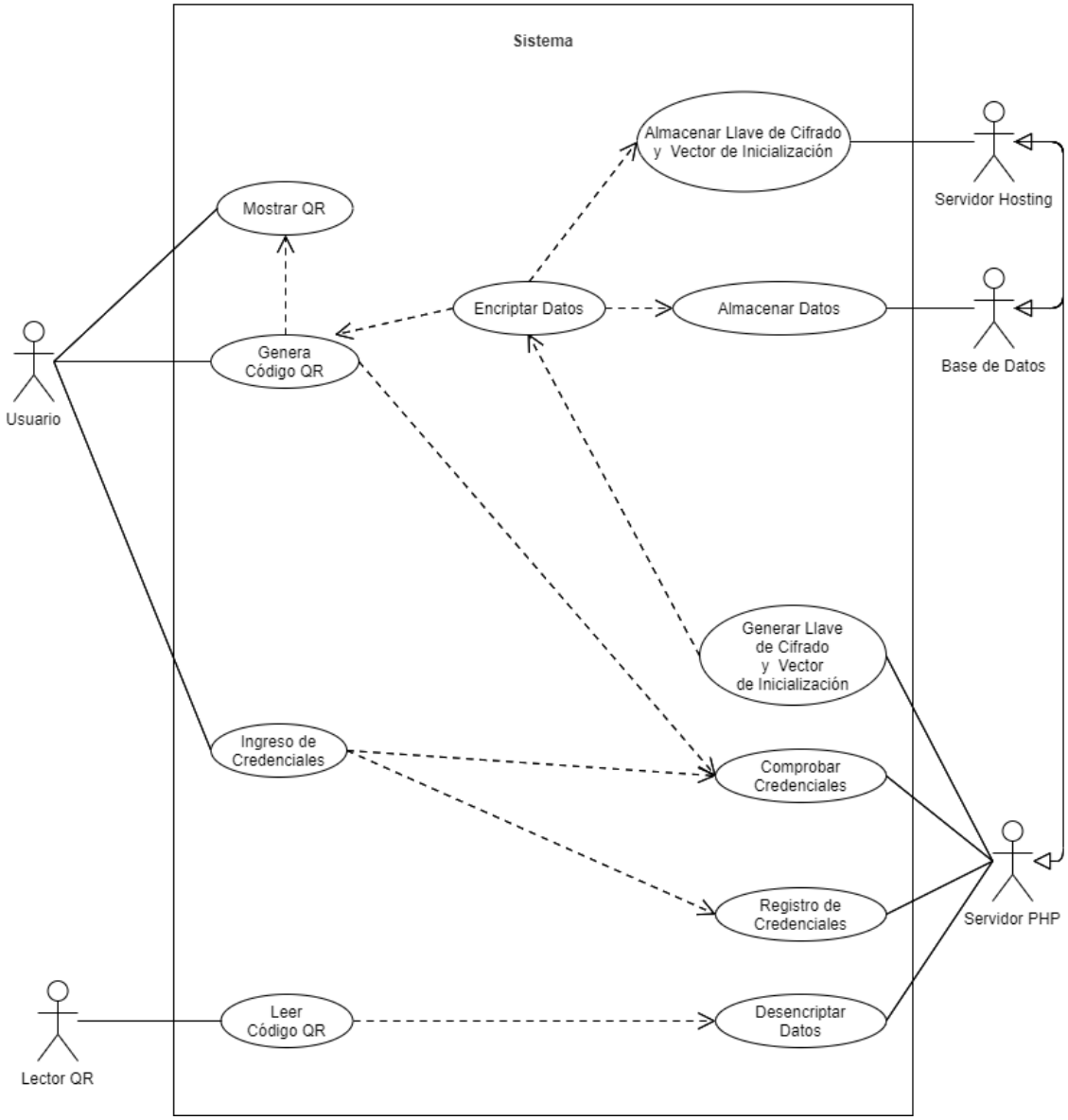
Este sistema se cuenta con cinco actores identificados que son:

- Usuario: son las personas que interaccionan con el sistema y los beneficiaros del servicio.
- Lector QR: herramienta empleada para verificar la información del código QR.
- Servidor Hosting: sistema externo que permite la interacción con la base de datos y el servidor php.
- Base de Datos: interviene en el almacenamiento y recuperación de la información.
- Servidor PHP: ejecuta tareas programadas de acuerdo con peticiones hechas por el usuario a través del sistema.

Estos cinco actores interaccionan entre sí para cumplir con la finalidad del sistema, en la figura 8 se muestra estos actores y sus casos de uso.

Figura 8

Diagrama de Casos de Uso del Sistema



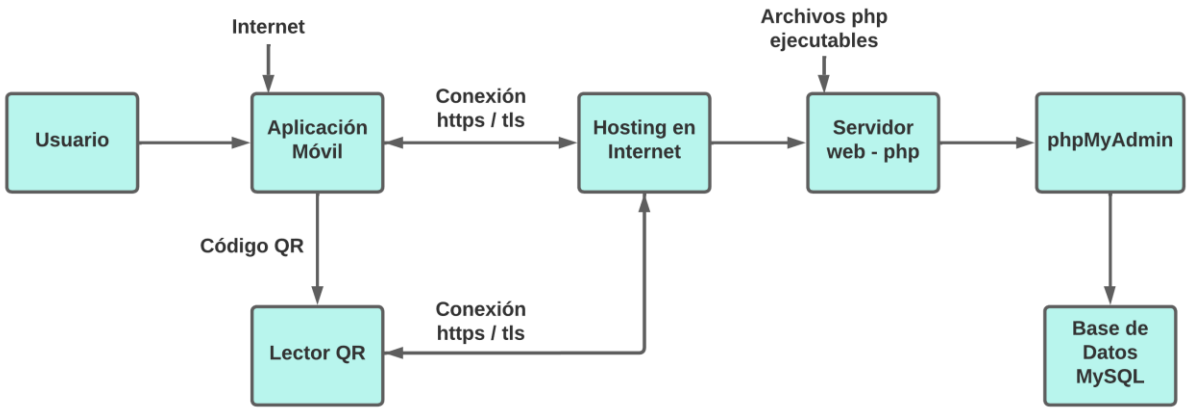
3.5.2. Diagrama de Bloques

En este diagrama representado en la figura 10, se muestra los componentes del sistema, que son: el usuario, la aplicación móvil, el lector QR, el servidor de hosting, servidor web que

soporte php, phpMyAdmin y la base de datos MySQL. Estos elementos a su vez se relacionan entre sí y requieren otros elementos para cumplir un propósito.

Figura 9

Diagrama de Bloques del Sistema



3.5.3. Base de Datos

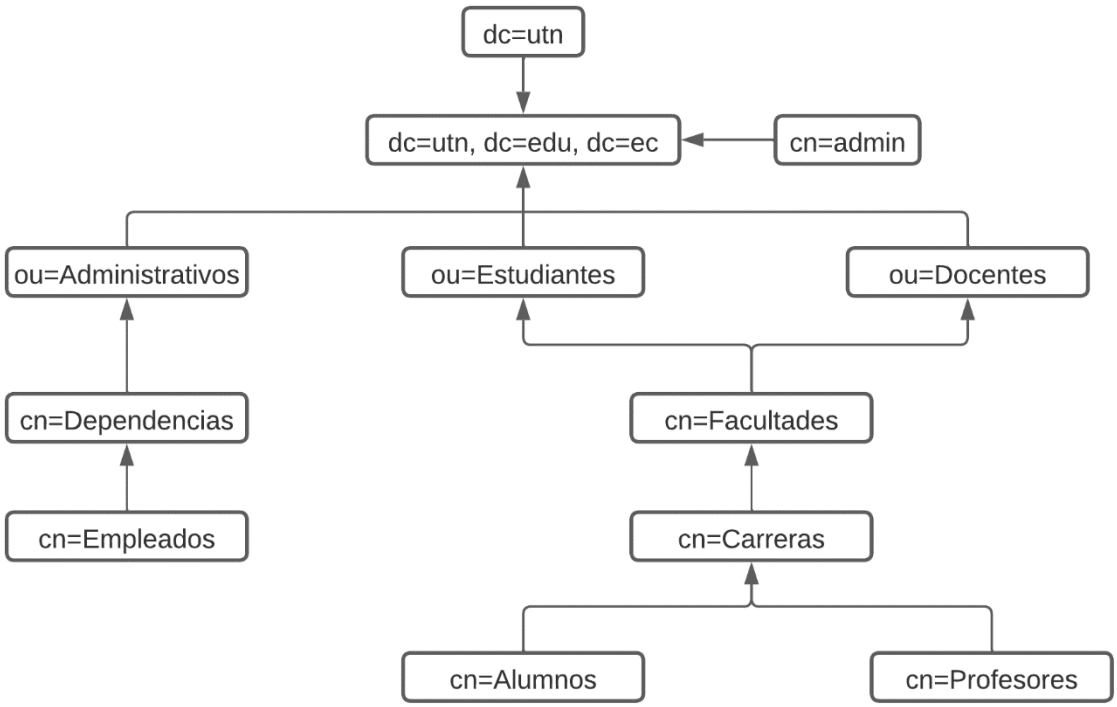
La base de datos permitirá registrar contraseñas y claves para para gestionar el acceso o el uso de servicios a usuarios que existan en la base de datos, esto implica que el usuario debe estar registrado en la universidad para poder usar el sistema.

Un papel importante de la base de datos es almacenar la información de las claves de acceso que serán generadas por un programa y que a su vez son corroboradas para el acceso de los distintos usuarios.

La estructura del árbol de la LDAP se creó siguiendo el mismo diseño utilizado en la Universidad Técnica del Norte, de modo que se copió su estructura tal como se ilustra en la figura 10.

Figura 10

Estructura jerárquica de la Base de Datos LDAP UTN



Nota. Adaptado DDTI

La base de datos consta de las siguientes partes

- dc.- Componente de Dominio, el nombre de la LDAP de la institución, es decir, dc=UTN,dc=edu,dc=ec.
- ou. - Unidad Organizativa, subdominios de estructura LDAP, para este caso Administrativos, Docentes y Estudiantes.
- cn.- Nombre Común, grupos de Dependencias, Facultades y Carreras en las que se subdividen las unidades organizativas.

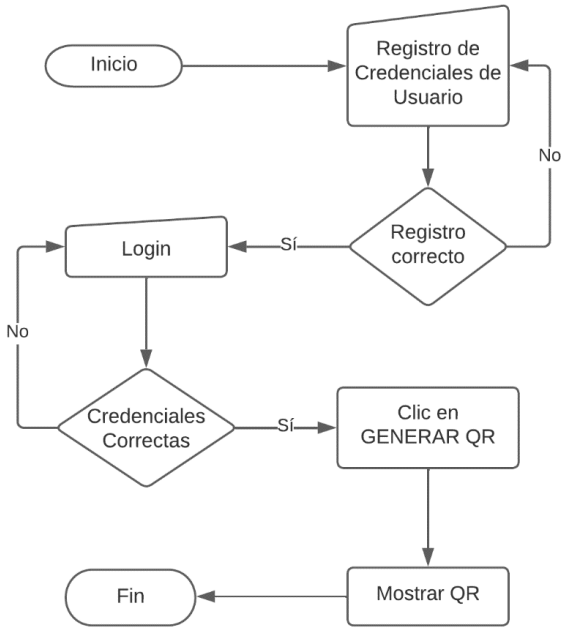
3.5.4. Aplicación Móvil

La aplicación móvil como se determinó en la elección del software estará programada con el lenguaje de java en el entorno de Android Studio.

La aplicación constara de tres actividades, una en donde los usuarios existentes en la base de datos deberán registrar una contraseña, que también se registrara en la base de datos. En la siguiente actividad el usuario que ya haya registrado una contraseña ingresará esta junto con el correo personal asociado, después, comprobando el correcto ingreso de las credenciales, se tendrá la actividad principal en donde se podrá generar el código QR con la información de una clave cifrada que le permitirá identificar al usuario. En la figura11 se muestran los procesos que sigue la aplicación.

Figura 11

Diagrama de Flujo de la Aplicación Móvil

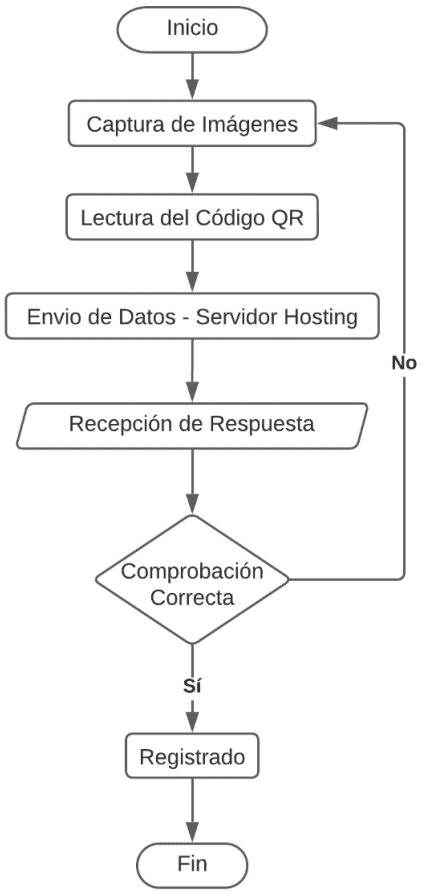


3.5.5. Lector de Códigos QR

Esta aplicación estará programada con el lenguaje de java y desarrollada en NetBeans, contará con un JFrame Form (camera.java) que presentará información al usuario y un java class (conexión.java) que interactuará con el servidor web. En la figura 12 se muestra la información correspondiente a las variables y métodos utilizados.

Figura 12

Diagrama de Flujo del Lector de Códigos QR



3.5.6. Gestión de Claves de Cifrado

En la investigación realizada se encontró que las prácticas más empleadas para guardar las llaves de cifrado son baúles de llaves (key vault), un servicio prestado por terceros, siendo otra opción el almacenamiento en una base de datos segura o el almacenamiento de las llaves en un fichero de acceso restringido, siendo este la opción que se aplicara en el sistema, al momento de generar la clave de acceso esta es cifrada y se almacena en la base de datos, en tanto que, la llave de cifrado y el vector de inicialización se almacena en el directorio de acceso restringido. Para obtener la llave de cifrado original se debe hacer la consulta a la clave cifrada

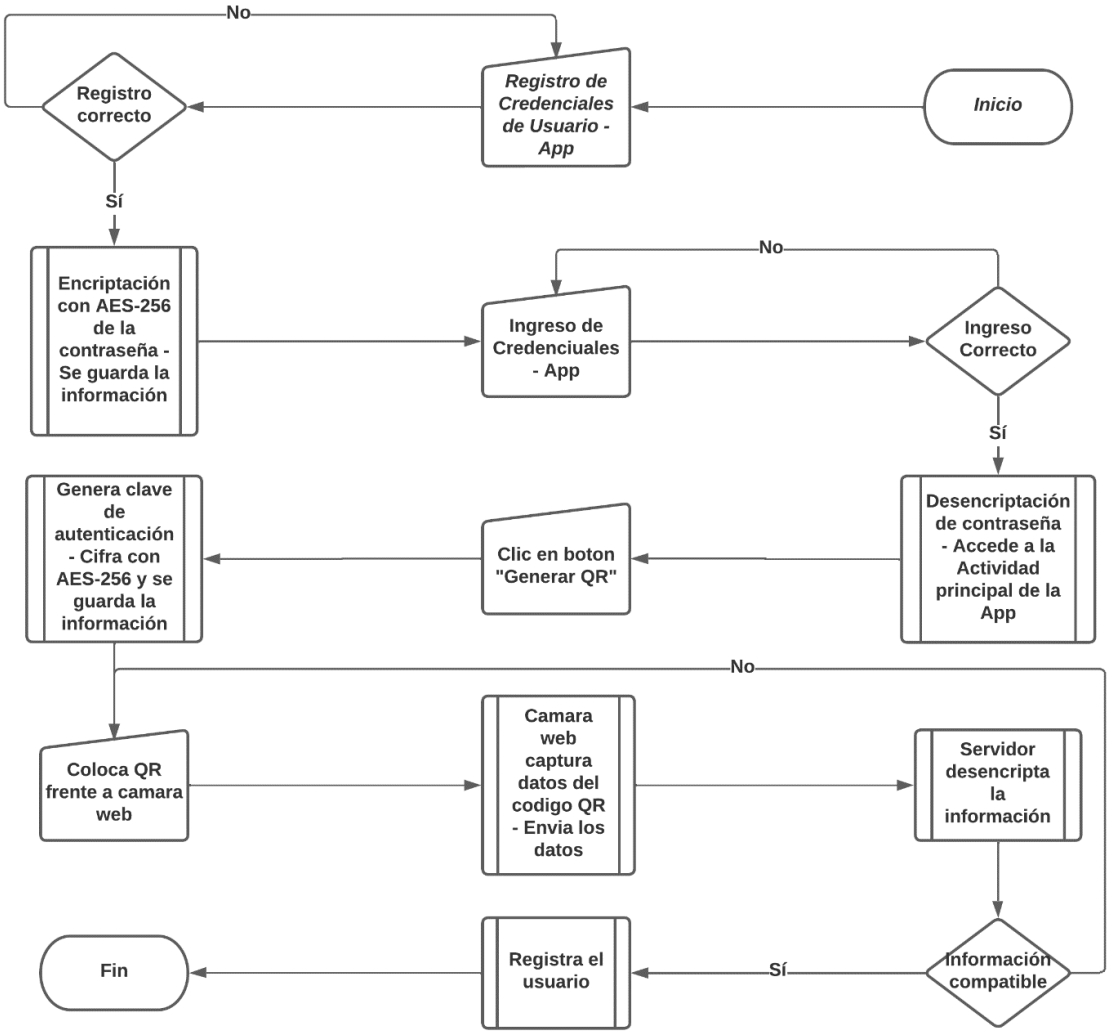
en la base de datos, además de, la llave de cifrado y el vector de inicialización guardados en el fichero de acceso restringido.

3.5.7. Diagrama de Flujo del sistema

En la figura 9 se muestra el proceso explicado de cómo funciona el sistema, desde, la creación de credenciales, pasando por los procesos establecidos en la programación hasta la autenticación del usuario.

Figura 13

Diagrama de Flujo del Sistema



4. IMPLEMENTACIÓN Y PRUEBAS

En este capítulo se presenta el proceso de desarrollo de todas las partes del sistema, la creación de la base de datos, de la aplicación móvil y del lector de códigos QR, así como, la elaboración de los archivos ejecutables que permitirán la administración de la base de datos a través de phpMyAdmin. También se abarca las pruebas de funcionamiento que corroboren los requerimientos planteados en el capítulo tres.

4.1. Implementación

En este apartado se muestra el desarrollo y la solución de problemas en el proceso de programación de todas las partes del sistema, así como, la explicación de la estructura y las partes requeridas en el código.

4.1.1. Hosting en Internet

El servicio de hosting es una versión gratuita brindada por LucusHost, este servidor tiene las siguientes características: se puede registrar hasta 2 dominios, 15 GB de espacio SSD NVMe,1536 MB de RAM, 125% de CPU (1,25 vCore), sin límite de tráfico, cuentas de correo ilimitadas, cPanel y SSL gratuito. Características que permiten comprobar el funcionamiento del sistema y que cuenta con un parámetro importante de seguridad como lo es el SSL.

Para contar con este servicio de hosting se debe llenar un formulario de registro y dar la aceptación de términos y condiciones, sin más, se dará el acceso al servidor hosting que aloja el servidor web, los ficheros de configuración y almacenamiento, phpMyAdmin y la base de datos MySQL.

Figura 14

Funciones de Hosting Empleadas en el Sistema

4.1.2. Base de Datos

El hosting tiene el apartado de la base de datos, en donde, al ser un servicio gratuito, se cuenta con ciertas condiciones en la configuración de la base de datos, como es un prefijo para identificar al usuario sin posibilidad de configurar este apartado.

La creación de la base de datos es sencilla, se debe registrar un nombre para la base de datos, crear un usuario asociado a la base de datos al cual se debe configurar una contraseña y dar los permisos requeridos para el manejo de la base de datos. En la figura 15 se puede ver la base de datos que esta creada en el hosting.

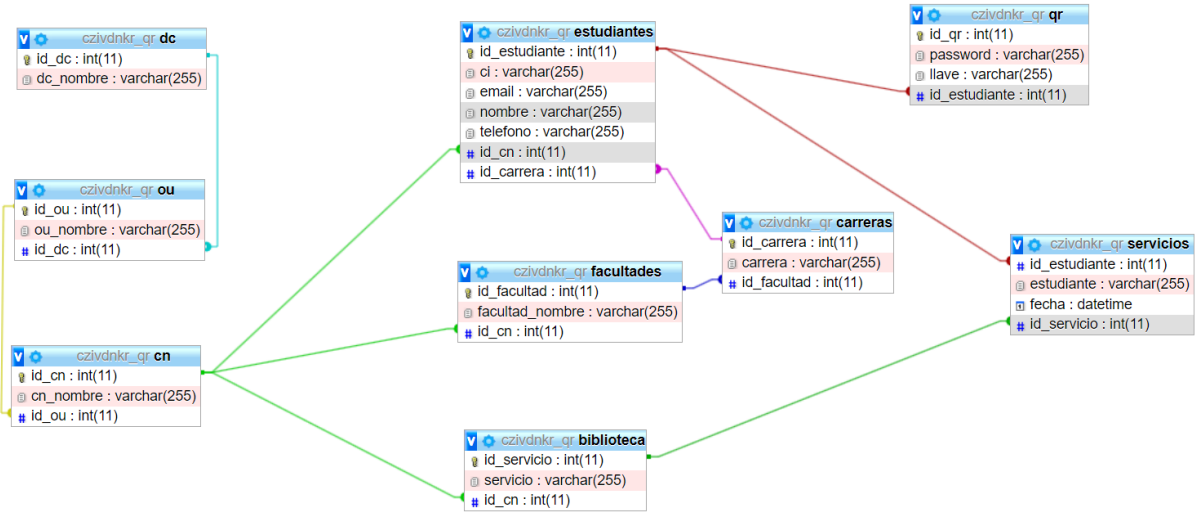
Figura 15

Base de Datos Creada

La administración de la base de datos se hace a través de phpMyAdmin y el ingreso de datos se hace con la interfaz gráfica, después de ingresar los datos planteados en el diseño en la figura 16 se muestra la estructura de la tabla una vez configurada.

Figura 16

Estructura de la Base de Datos



En la tabla usuario, los datos ingresados son generados por software especial capaz de brindar números y nombres aleatorios que sirven para hacer la prueba de funcionamiento del sistema, esto en consideración a que los datos requeridos son confidenciales y propios de cada usuario y la institución que los maneja. Los datos utilizados se muestran en la figura 17.

Figura 17

Datos de la Tabla Estudiantes

id_estudiante	ci	email	nombre	telefono	id_cn
1	1004567890	sigarcia@utn.edu.ec	Sofía Isabella García	0987654321	4
2	0345678901	ajperez@utn.edu.ec	Alejandro José Pérez	0987654321	4
3	0456789012	vlramirez@utn.edu.ec	Valentina Lucía Ramírez	0987654321	4
4	1004403227	mecastro@utn.edu.ec	Martín Eduardo Castro	902890804	4
5	1007303422	vctorres@utn.edu.ec	Victoria Camila Torres	0993642707	4
6	1001921317	magomez@utn.edu.ec	Mateo Andrés Gómez	0926920088	4
7	0463516619	isrodriguez@utn.edu.ec	Isabella Sofía Rodríguez	0987969028	4
8	1005995217	sdortiz@utn.edu.ec	Santiago David Ortiz	0929233947	4
9	1006430994	nacastro@utn.edu.ec	Natalia Andrea Castro	0936630797	4
10	1006700063	semartinez@utn.edu.ec	Samuel Esteban Martínez	0977302163	4

4.1.3. Aplicación Móvil

La aplicación móvil se programa en Android Studio, se diferencia la parte lógica y la parte gráfica en la creación de una aplicación, las cuales se programan en java y xml respectivamente.

La parte gráfica de la aplicación se divide en tres activitys que son:

- **activity_resgistro.xml**

En esta actividad se presenta un formulario en donde se debe ingresar el email y una contraseña que en posterior sera almacenada, y tiene un botón para ejecutar una acción programada en la parte lógica. En la figura 18 se muestra la forma del activity registro.

Figura 18

Activity Registro

En la parte del código se debe tomar especial cuidado en el nombre que se le asigna a los cuadros de texto para el ingreso del correo (*remail*), la contraseña (*rpassword*) y del botón de registro (*bregistrar*).

- **activity_login.xml**

Este activity es similar al de registro en su estructura, de igual manera se requiere el ingreso del correo y contraseña y se tiene un botón al cual se asocia una acción. En la figura 19 se muestra el activity login.

De igual manera se debe recordar el nombre que se le asigna a los cuadros de texto para el ingreso del correo (*lemail*), la contraseña (*lpassword*) y del botón de registro (*blogin*).

Figura 19

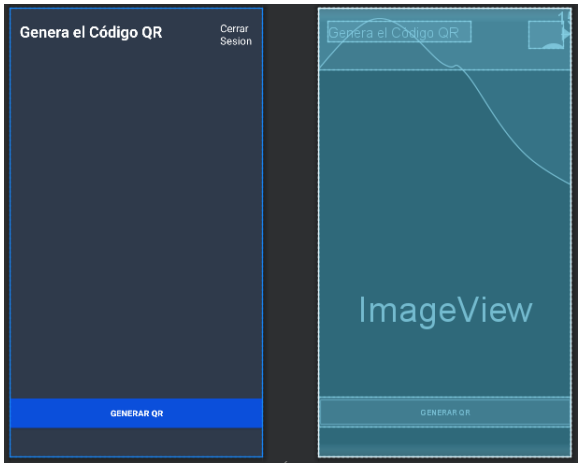
Activity Registro

- **activity_main.xml**

La estructura del main activity cuenta con un cuadro de texto que presentara información del usuario, otro que al ser seleccionado permitirá borrar las credenciales ingresadas en al aplicación, también cuenta con un cuadro de vista de imagen en donde se mostrara el código QR generado y un botón que permitirá esta acción. En la figura 20 se muestra la estructura del main activity.

Figura 20

Acitvity Main



Para que el código cumpla con las funciones planteadas se deben añadir las librerías necesarias, esto se realiza añadiendo directamente en el archivo *build.gradle* el llamado de las librerías. Para esta aplicación se hace uso de zxing que es una biblioteca de código abierto escrita en Java para leer códigos de barras y códigos QR desarrollada por Google, que en total serán tres versiones de esta librería para poder tener compatibilidad con distintas versiones del sistema operativo de Android.

Otras librerías requeridas son: cardview:1.0.0: esta librería proporciona la vista de tarjeta (CardView) que se utiliza comúnmente en la interfaz de usuario de Android. La vista de tarjeta es una forma de mostrar información en una tarjeta de estilo similar a una tarjeta física; Material 1.4.0: esta librería proporciona componentes de diseño de materiales de Google para la interfaz de usuario de Android. Los componentes incluyen botones, barras de herramientas, tarjetas, menús, iconos y más; Volley:1.2.1: esta librería proporciona una biblioteca para realizar solicitudes de red de manera sencilla y eficiente en aplicaciones de Android. Volley gestiona automáticamente las solicitudes de red en segundo plano y ofrece un manejo sencillo de la caché de respuestas; Jaxb-api:2.4.0-b180725.0427: esta librería proporciona una API para convertir objetos Java en XML y viceversa. JAXB (Java Architecture for XML Binding) es

una especificación que define cómo se pueden mapear objetos Java a XML y viceversa; Jaxb-api:2.3.1: esta es otra versión de la librería JAXB; Commons-codec:1.15: esta librería proporciona implementaciones de codificación y decodificación de varios formatos, como Base64, hex y URL-safe. Estas implementaciones se utilizan comúnmente en aplicaciones web para la manipulación de datos codificados. En la figura 21 se muestran las librerías implementadas.

Figura 21

Librerías Implementadas en la Aplicación

```
implementation 'com.journeyapps:zxing-android-embedded:4.3.0'
implementation('com.journeyapps:zxing-android-embedded:4.3.0') { transitive = false }
implementation 'com.google.zxing:core:3.3.0'

implementation 'androidx.appcompat:appcompat:1.5.1'
implementation 'com.google.android.material:material:1.7.0'
implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
testImplementation 'junit:junit:4.13.2'
androidTestImplementation 'androidx.test.ext:junit:1.1.4'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.0'

//librerias web service
implementation 'androidx.cardview:cardview:1.0.0'
implementation 'com.google.android.material:material:1.4.0'
implementation 'com.android.volley:volley:1.2.1'

implementation 'javax.xml.bind:jaxb-api:2.4.0-b180725.0427'
implementation 'javax.xml.bind:jaxb-api:2.3.1'
implementation 'commons-codec:commons-codec:1.15'
```

En la parte lógica del programa se tiene tres archivos .java donde se desarrolla todo el proceso de la aplicación, estos archivos son:

- **registro.java**

La actividad se inicia en el método "*onCreate*", donde se establece el diseño de la actividad utilizando el método "*setContentView*" y se configuran los elementos de la interfaz

de usuario (EditText, Button) mediante *findViewById*. Luego se establece un *OnClickListener* en el botón de registro, que llama al método "*insertarDatos*" cuando se hace clic en él.

El método "*insertarDatos*" recupera el texto ingresado en los campos de texto (correo electrónico y contraseña) y comprueba si están vacíos. Si ambos campos tienen datos, se realiza una solicitud POST al servidor especificado en la URL. Se crea un objeto de tipo StringRequest que envía la información del correo electrónico y la contraseña al servidor, y luego espera una respuesta. Si el servidor responde "Registrado correctamente", se muestra un mensaje de confirmación y se inicia una nueva actividad (login). Si la respuesta no es correcta, se muestra un mensaje de error.

Figura 22

Comprobación de Campos y Envió de Datos

```
//se comprueba que los campos no esten vacios
if (email.isEmpty()){
    t_email.setError("Complete los Campos");
    return;
}else if (password.isEmpty()){
    t_password.setError("Complete los Campos");
    return;
}else{
    //estando completos los campos se realiza la peticion al servidor phpmyadmin para registrar la contrasña
    progressDialog.show();
    StringRequest request = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
```

Finalmente, hay un método adicional llamado "*login*" que se activa cuando se hace clic en un botón en la actividad de registro, lo que permite al usuario cambiar a la actividad de inicio de sesión.

- **login.java**

Este código implementa la funcionalidad de inicio de sesión de la aplicación. La actividad principal se define en el archivo activity_login.xml. En la actividad, se definen dos campos de texto EditText, uno para el correo electrónico y otro para la contraseña, que se inicializan en el método onCreate().

En el método onCreate(), también se comprueba si ya se han registrado datos de inicio de sesión anteriormente utilizando la interfaz de preferencias compartidas de Android. Si ya se ha iniciado sesión, se redirige al usuario a la actividad principal.

El método de inicio de sesión se llama login() y se ejecuta cuando se hace clic en el botón de inicio de sesión. En el método login(), se comprueba si los campos de correo electrónico y contraseña están vacíos y se muestra un mensaje de error si es necesario. Si no hay errores, se realiza una solicitud POST a una URL específica (url: que establece la conexión con el servidor web del hosting) utilizando la biblioteca Volley de Android. Se proporcionan los valores del correo electrónico y la contraseña en la solicitud POST. Si la solicitud tiene éxito, se inicia la actividad principal y se guarda el estado de inicio de sesión utilizando la interfaz de preferencias compartidas de Android. Si la solicitud no tiene éxito, se muestra un mensaje de error.

También hay un método de registro() que se ejecuta cuando se hace clic en el botón de registro. Este método simplemente cambia a la actividad de registro.

Figura 23

Método Login

```
public void login(View view){
    //se comprueba que los campos no esten vacios
    if (t_email.getText().toString().equals("")){
        Toast.makeText( context: this, text: "Ingrese Correo",Toast.LENGTH_SHORT).show();
    }else if (t_pass.getText().toString().equals("")){
        Toast.makeText( context: this, text: "Ingrese Contraseña",Toast.LENGTH_LONG).show();
    }else {
        final ProgressDialog progressDialog=new ProgressDialog( context: this);
        progressDialog.setMessage("Espere");
        progressDialog.show();

        //se recibe los valores ingresados y se guarda como una variable string
        str_email=t_email.getText().toString().trim();
        str_password=t_pass.getText().toString().trim();

        //se hace requerimientos a phpmyadmin
        StringRequest request =new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
```

- **home.java**

Esta actividad genera un código QR en base a la información de inicio de sesión de un usuario en una base de datos.

Primero, se busca en las preferencias compartidas del usuario el correo electrónico de inicio de sesión. Si el correo electrónico existe, se realiza una solicitud HTTP POST a un servidor PHP en la URL proporcionada en la variable *url*. El parámetro de la solicitud es el correo electrónico del usuario.

Si la respuesta de la solicitud no es "Error al Generar", se procesa la respuesta y se crea el código QR con la información codificada. La información se codifica en hexadecimal y se separa con un carácter "g". La primera parte es el correo electrónico codificado y la segunda parte es la clave cifrada del código QR.

Finalmente, se muestra el código QR en una *ImageView*. La aplicación también utiliza una ventana emergente *ProgressDialog* para mostrar el progreso de la generación del código QR. El código también incluye una función para impedir que se tomen capturas de pantalla.

Figura 24

Cifrado de la Llave

```
//se establece la llave para el cifrado
SecretKeySpec secretKeySpec = new SecretKeySpec(llave_llave_qr, algorithm: "AES");
//se establece el vector de inicializacion
IvParameterSpec ivParameterSpec = new IvParameterSpec(iv);
//se configura el cifrado
Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
// Se cambia la longitud de la llave de 16 bytes (128 bits) a 32 bytes (256 bits)
byte[] llave_llave_qr_256 = Arrays.copyOf(llave_llave_qr, newLength: 32);
secretKeySpec = new SecretKeySpec(llave_llave_qr_256, algorithm: "AES");
//se encripta
cipher.init(Cipher.ENCRYPT_MODE, secretKeySpec, ivParameterSpec);
//se guarda el resultado en un array
byte[] mensajeCifrado = cipher.doFinal(llave_qr);
String mensajeCifradoHex = new BigInteger( signum: 1, mensajeCifrado).toString( radix: 16);
```

Figura 25

Representación del Mensaje en el Código QR

```
String mensajeqr =emailHex + "g" + mensajeCifradoHex;

//en el QR se muestra el email codificado en hexadecimal
//se separa por una "g" para diferenciar las dos partes en el reconocimiento del QR
//se encuentra tambien la clave cifrada convertida a formato hexadecimal
BarcodeEncoder barcodeEncoder = new BarcodeEncoder();
Bitmap bitmap = barcodeEncoder.encodeBitmap(mensajeqr, BarcodeFormat.QR_CODE,  width: 750,  height: 750);

ivCodigoQR.setImageBitmap(bitmap);
```

4.1.4. Lector de Códigos QR

Este programa de Java utiliza una librería llamada "Webcam-capture" para acceder a la cámara web de un dispositivo y poder escanear códigos QR. La interfaz gráfica del programa es una ventana que muestra el video en vivo de la cámara web.

El programa utiliza la librería ZXing para decodificar el código QR en la imagen capturada por la cámara web. Si se detecta un código QR, el programa envía una solicitud HTTP a un servidor web utilizando una clase *conexion*, que se encarga de manejar la comunicación HTTP. El contenido del código QR se envía como un parámetro en la solicitud HTTP.

La respuesta del servidor web se almacena en una cadena de caracteres llamada "mensaje". Si el mensaje es "Acceso Permitido", el programa cierra la conexión de la cámara web y termina su ejecución. Si el mensaje no es "Acceso Permitido", el programa continúa escaneando códigos QR hasta que se detecte uno con el mensaje correcto.

El programa utiliza un hilo para ejecutar la tarea de escanear continuamente la imagen de la cámara web en busca de códigos QR. El hilo se ejecuta en segundo plano mientras la ventana del programa está abierta. Cuando se detecta un código QR, el hilo se detiene temporalmente mientras se envía la solicitud HTTP y se espera la respuesta del servidor web.

Si el mensaje es "Acceso Permitido", el hilo se detiene permanentemente y la conexión de la cámara web se cierra.

4.1.5. Archivos Ejecutables PHP

Estos archivos se encuentran alojados en el servidor hosting y son accesibles por medio del servidor web, cada archivo realiza una operación específica. En total son 5 archivos php que intervienen en varios procesos requeridos por el sistema, estos son:

- **login.php**

El código es un archivo PHP que se conecta a la base de datos MySQL para autenticar un usuario a través de información recibida por la aplicación. Primero se establece la conexión con la base de datos y se verifica si la conexión fue exitosa. Luego se recibe la información de correo electrónico y contraseña enviada desde la aplicación y se almacena en variables.

El código busca en la tabla "usuario" de la base de datos el registro que coincida con el correo electrónico proporcionado. A continuación, recorre los resultados de la consulta para obtener la contraseña cifrada almacenada en la base de datos.

Luego se desencripta la contraseña cifrada utilizando una clave de cifrado y un vector de inicialización que se almacenan en un archivo que se lee desde el servidor. Finalmente, se verifica que la contraseña ingresada por el usuario desde la aplicación coincide con la contraseña desencriptada almacenada en la base de datos. Si coinciden, se muestra un mensaje de "Ingreso Exitoso" que es el mensaje que permite pasar entre la actividad login a la actividad principal de la aplicación, de lo contrario se muestra un mensaje de "Credenciales Incorrectas".

Figura 26

Comprobación de Credenciales

```
//Se comprueba que la variable email enviado desde la aplicacion sea igual al email de la base de datos
//Y que el password enviado desde la aplicacion es igual al password almacendo en la base de datos
if($email==$email_bd && $password==$password_decrypt){
    echo "Ingreso Exitoso";
}else{
echo " Credenciales Incorrectas";
}
```

- **insertar.php**

Este código PHP se encarga de recibir datos de un formulario mediante el método POST, cifrar la contraseña usando el algoritmo AES-256-CBC, guardar la contraseña cifrada en la base de datos y crear un archivo en el servidor para almacenar la clave de cifrado y el vector de inicialización utilizados.

Primero, se establece una conexión con la base de datos y se reciben los datos de correo electrónico y contraseña del formulario. Luego, se genera una clave de cifrado aleatoria y un vector de inicialización aleatorio. La contraseña se cifra utilizando estos valores y se guarda en la base de datos. A continuación, se crea un archivo en el servidor para almacenar la clave de cifrado y el vector de inicialización, utilizando una función hash para el correo electrónico como nombre del archivo.

Si la operación de guardado es exitosa, se escribe en el archivo la clave de cifrado y el vector de inicialización, se confirma el registro y se cierra el archivo. Si no es exitosa, se indica que el correo electrónico es incorrecto.

Figura 27

Registro de Credenciales

```php
if($resultado->affected_rows > 0){
//se crea el archivo donde se guarda la llave de cifrado y el vector de inicializacion
//Para evitar posibles problemas de seguridad, se hace un hash del correo utilizando la función md5()
    $archivo_pass = fopen("/home/tcrcwxrg/pass/" . md5($email) . ".txt","w+b");
    if( $archivo_pass == false ) {
        echo "Error al Crear Archivo";
      }
      else
      {
            // Mensaje de confiramcion
            echo "Registrado Correctamente";
            // Se escribe en el archivo los datos con un salto de linea
            fwrite($archivo_pass, $llave_pass_hex . PHP_EOL . $iv_hex);
            // Fuerza a que se escriban los datos pendientes en el buffer:
            fflush($archivo_pass);
      }
      // Cierra el archivo
      fclose($archivo_pass);

}else{
    echo "Email Incorrecto";
}
```

-
- **qr.php**

Este código PHP realiza la conexión a la base de datos y permite actualizar el atributo llave de un usuario. Se recogen los datos del usuario a través de un formulario POST. A continuación, se genera una llave aleatoria mediante la función random_bytes(32), la cual es codificada en formato hexadecimal para un manejo más sencillo. Se genera un vector de inicialización y se cifra la contraseña mediante la función openssl_encrypt().

Luego, se guarda la llave cifrada en la base de datos. Si la operación de actualización de la base de datos es exitosa, se crea un archivo de texto que contiene la llave de cifrado y el vector de inicialización utilizando la función fopen(). En caso contrario, se muestra un mensaje de error.

En un segundo código, se realiza la conexión a la misma base de datos y se actualiza la llave de un usuario. Se recoge el correo electrónico del usuario a través de un formulario POST.

A continuación, se genera un vector de inicialización y se cifra la llave aleatoria mediante la función openssl_encrypt(). Luego, se guarda la llave cifrada en la base de datos. Si la operación de actualización de la base de datos es exitosa, se crea un archivo de texto que contiene la segunda llave aleatoria y el vector de inicialización utilizando la función fopen(). En caso contrario, se muestra un mensaje de error.

Se envían la información a la aplicación para que pueda cifrar la información y generar el código QR.

Figura 28

Envió de Credenciales

```
if( $archivo_llave == false ) {
    echo "Error al crear el archivo";
}
else
{
    // Escribir en el archivo:
     fwrite($archivo_llave, $llave_llave_qr_hex . PHP_EOL . $iv_hex);
    // Fuerza a que se escriban los datos pendientes en el buffer:
     fflush($archivo_llave);
     fclose($archivo_llave);
}


}else{
    echo "Error al Generar";}
```

- **acceso.php**

Este es un código escrito en PHP que recibe una petición POST enviada por la aplicación de Java. Primero, establece una conexión con una base de datos MySQL utilizando la función mysqli_connect(). Luego, comprueba si la petición es de tipo POST y recupera los datos enviados mediante el método file_get_contents(), para separar y decodificar los valores hexadecimales recibidos.

El código luego busca la clave pública correspondiente al correo electrónico que se ha recibido en la base de datos y la desencripta. Luego, se comparan las claves recibidas desde la

aplicación de Java con las claves desencriptadas almacenadas en la base de datos. Si coinciden, se muestra un mensaje de "Acceso Permitido", de lo contrario se muestra un mensaje de "Acceso Denegado".

Para desencriptar las claves, el código utiliza la función openssl_decrypt(), que utiliza un algoritmo de cifrado AES-256-CBC y la clave de cifrado obtenida de la base de datos y un vector de inicialización (IV) que se encuentra también en el archivo de clave privada en formato hexadecimal. Además, la función bin2hex() se utiliza para convertir las claves desencriptadas en formato binario a formato hexadecimal para facilitar su comparación y autenticar al usuario.

Figura 29

Comprobación de Credenciales en el Código QR

```
//Se comprueba que la variable email enviada desde el escaner sea igual al email de la base de datos
//Y que la llave enviada desde el escaner es igual a la llave desencriptada almacenda en la base de datos
if($email==$email_bd && $llave_qr_comprobar==$llave_bd_comprobar && $open_key){
    echo "Acceso Permitido";
}else{
echo "Acceso Denegado";
}
```

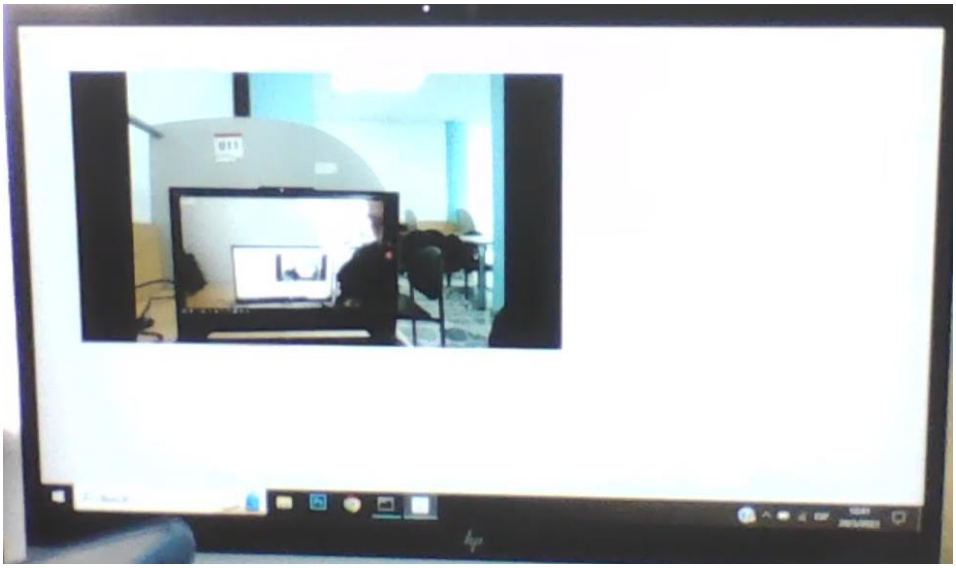
4.1.6. Integración en Servicios de la Biblioteca

Con el objetivo de verificar el correcto funcionamiento del sistema desde una perspectiva diferente, se estableció que el funcionamiento del sistema como prueba de funcionamiento se daría para registrar al estudiante que usa un computador. Para ello, se logró obtener acceso a un ordenador en la biblioteca de la universidad y se procedió a instalar el programa necesario para ejecutar el sistema en dicho equipo. Esto para comprobar si el funcionamiento del sistema se mantenía estable y sin problemas en otro escenario.

La parte del sistema que funcionara en este dispositivo es la de lector QR, el procesamiento que realiza es la de capturar imágenes por la cámara web, enviar la información para la comprobación al servidor y recibir la respuesta de validez de la información.

Figura 30

Lector QR en Funcionamiento



- **Servicio 1**

El primer servicio en el que se plantea el uso del sistema es con los préstamos de los computadores de la biblioteca, donde, un estudiante puede requerir un computador previo a un registro.

Figura 31

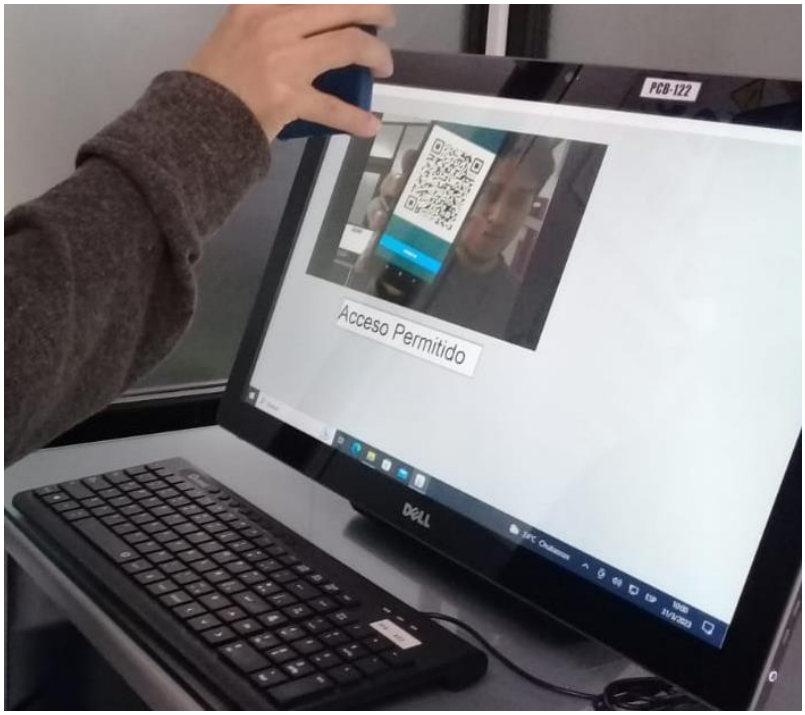
Computador Prestado por la Biblioteca – Servicio 1

- **Servicio 2**

Se realiza también la prueba de funcionamiento del sistema en los computadores de registro ubicados en la entrada de la biblioteca, donde los estudiantes ingresan sus credenciales para realizar el registro correspondiente.

Figura 32

Computador de Registro del Estudiante - Servicio 2



Este sistema al permitir la autenticación de usuarios se puede implementar en cualquier ambiente en el que se requiera identificar o registrar a un estudiante, como: al requerir el uso de la Videoteca y la Sala de Audio y Video en donde un estudiante puede reservar este espacio previo a un registro, al ingreso de la biblioteca como un método de registro diferente al de ingreso de cédula.

4.2. Pruebas

En esta sección se muestra el funcionamiento del sistema en todas las etapas, así como, las pruebas planteadas para corroborar la seguridad de la información que se maneja en el proceso.

Las pruebas que se realizan buscan mostrar el funcionamiento del sistema en las diferentes etapas que esta presenta y los requerimientos que cumplen, que son:

Prueba 1: Funcionamiento del Sistema.

Prueba 2: Cifrado AES-256 para Encriptar la Información y Manejo de Llaves de cifrado.

Prueba 3: Seguridad en la Comunicación del Sistema.

Prueba 4: Integración en los servicios de la biblioteca de la UTN.

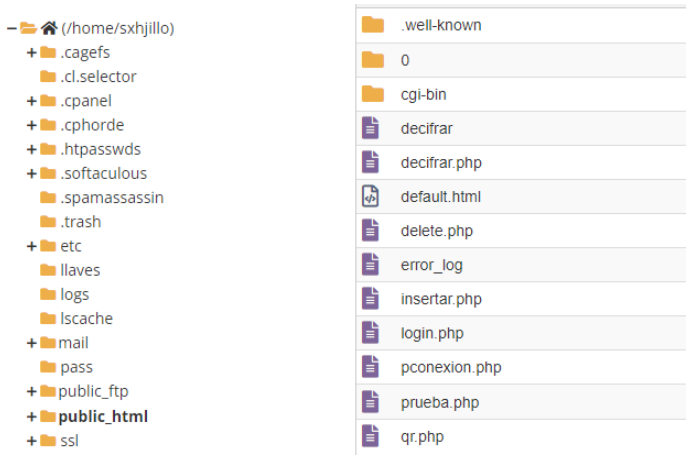
4.2.1. Prueba 1

- **Archivos del Sistema**

Los archivos de configuración se encuentran alojados en el servidor de hosting en ficheros que pueden ser accedidos desde internet, lo que permite realizar la consulta desde la aplicación, al momento que se realiza las peticiones desde la aplicación estos ejecutan el código que contiene y generan la acción establecida.

Figura 33

Archivos de Configuración PHP



El sistema accede a estos archivos por medio de un *url*, que especifica el archivo específico para el proceso, esta dirección se encuentra dentro del código de funcionamiento del sistema como se muestra en la figura 35.

Figura 34

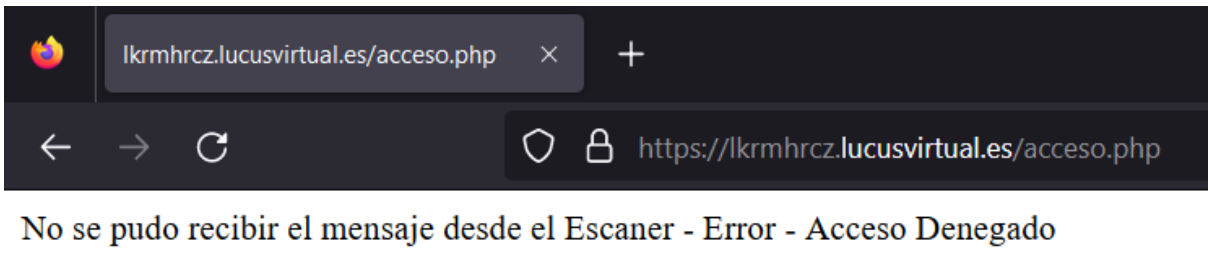
Dirección URL del Archivo acceso.php

```
public String url = "https://lkrmhrcz.lucusvirtual.es/acceso.php";
```

Cuando se cumplen los parámetros establecidos en el código del archivo de configuración, este devolverá los datos requeridos por el sistema, de otra manera se obtendrá un mensaje indicando el error, en la figura 36 se muestra el resultado de una petición realizada desde un navegador a la dirección url del archivo de configuración para el acceso.

Figura 35

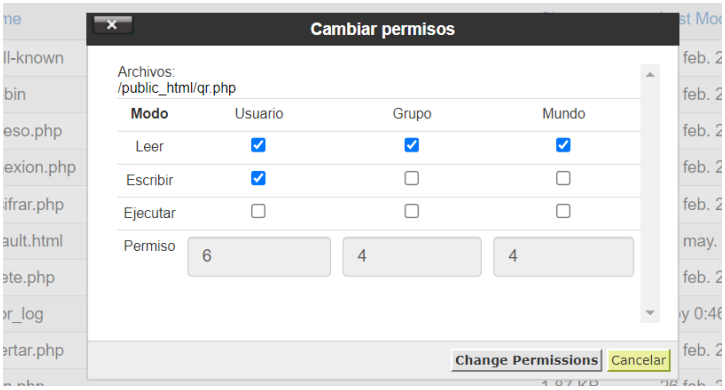
Acceso al Archivo de Configuración desde un Navegador



La utilización de permisos establecidos en los sistemas permite controlar el nivel de acceso a los recursos disponibles, limitando la capacidad para ejecutar o modificar código. En el caso específico de las consultas desde internet, este enfoque resulta especialmente útil ya que evita la posibilidad de que terceros puedan inyectar código malicioso o realizar acciones no autorizadas en el sistema. De esta forma, se garantiza una mayor seguridad en el manejo de información y se minimizan los riesgos de vulnerabilidad ante posibles ataques. En la figura 37 se muestran los permisos establecidos para los archivos de configuración del sistema.

Figura 36

Permisos de los Archivos PHP

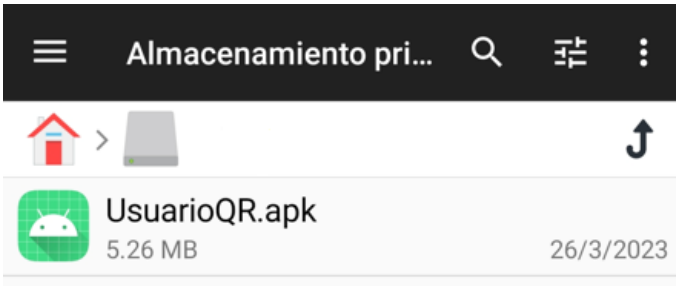


- **Instalación de la Aplicación**

Esta aplicación se puede instalar en el dispositivo con el archivo “UsuarioQR.apk” un ejecutable diseñado para Android, este documento se debe disponer en el almacenamiento del dispositivo para realizar el proceso de instalación.

Figura 37

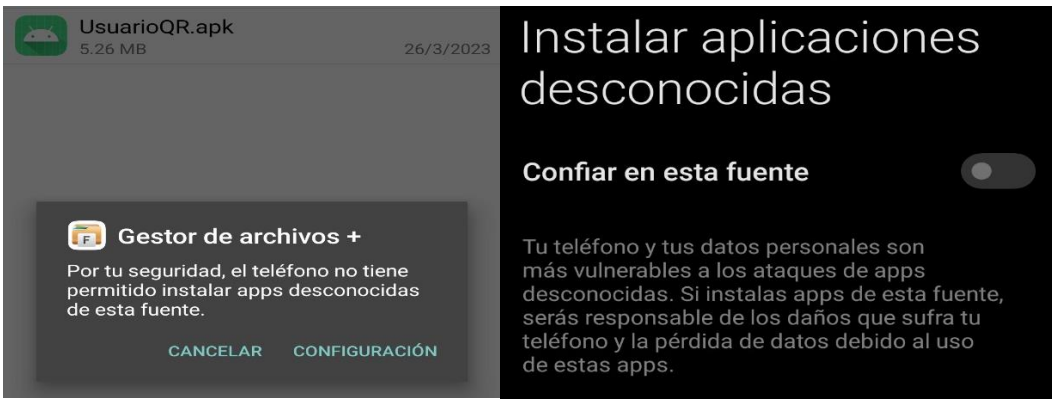
Archivo APK de la Aplicación



Cuando el usuario de un clic en el archivo se mostrarán avisos de seguridad, y para poder instalar esta aplicación es necesario dar permisos en la configuración del celular de forma manual.

Figura 38

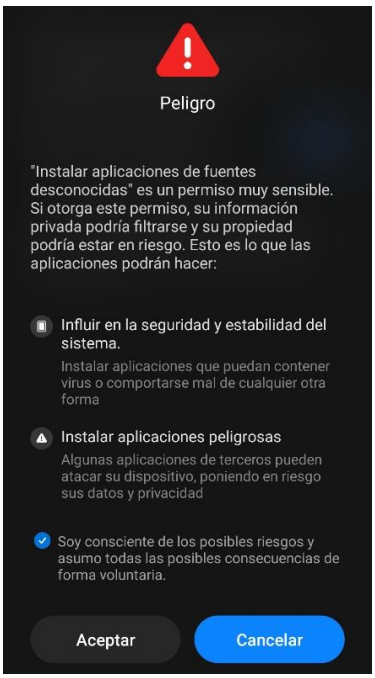
Permisos de Instalación



Al configurar en el dispositivo que se puede confiar en el archivo que se instala se presentan avisos y consideraciones de seguridad más detalladas, esto se da por que la aplicación no se encuentra en una tienda certificada de aplicaciones.

Figura 39

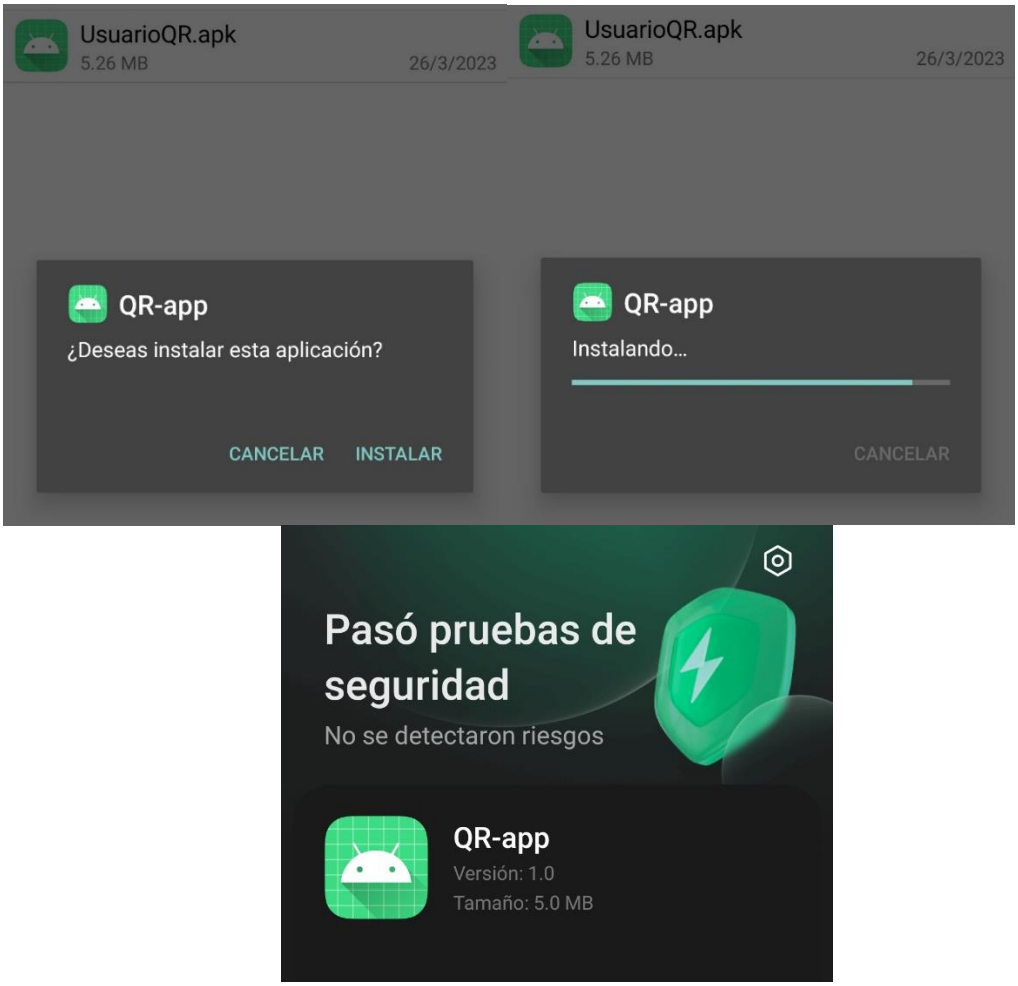
Alerta de Seguridad



Después de dar los permisos necesarios y leer las advertencias de seguridad, el celular permite hacer la instalación de la aplicación.

Figura 40

Instalación de la Aplicación



- **Uso de la Aplicación**

En principio se debe tener instalada la aplicación en el dispositivo, partiendo de esto, el usuario debe acceder a la parte de registro de la aplicación, donde se muestra el texto “Crea una Contraseña”, aquí el estudiante debe ingresar el correo institucional, pues es el único que se acepta, además, se debe ingresar una contraseña. Teniendo estos dos campos llenos se puede dar clic en el botón registrar, en caso de que no se cumpla las condiciones de esta actividad no se realiza el registro.

Figura 41

Registro del Estudiante

En la segunda actividad que es el ingreso de las credenciales el usuario debe ingresar el correo institucional y la contraseña previamente registrada, al dar un clic en el botón login la aplicación comprueba los datos y pasa a la siguiente actividad para generar el código QR. De darse un error en la comprobación de la información o el ingreso de credenciales se muestra un mensaje de error.

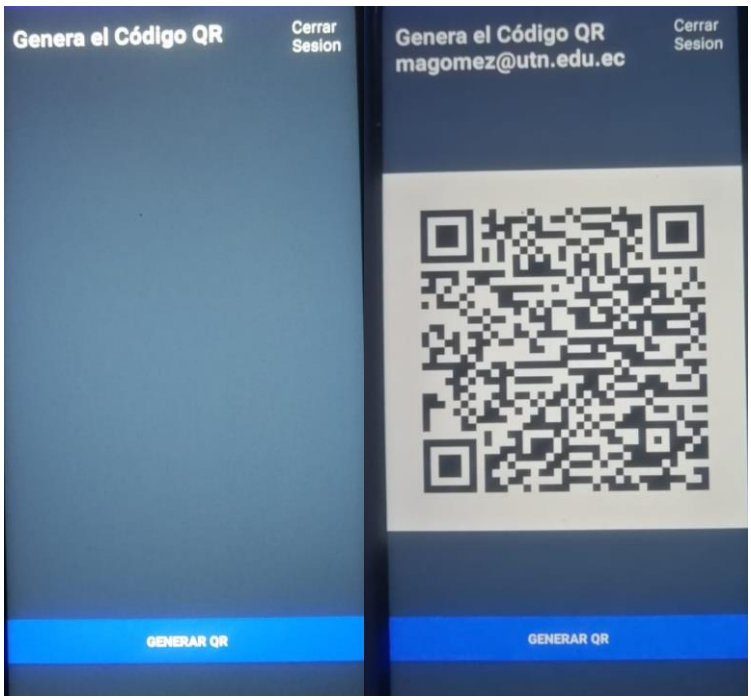
Figura 42

Ingreso de Credenciales

Una vez que las credenciales se han comprobado correctamente se accede a la actividad principal en donde se genera el QR con el que el estudiante puede autenticarse, la primera vista de la actividad es un fondo vacío sin el QR, para obtenerlo el usuario debe dar un clic en el botón “GENERAR QR”, momento en el cual la aplicación mostrar la imagen del código.

Figura 43

Generación del Código QR



4.2.2. Prueba 2

La información de las llaves de cifrado no se encuentra almacenada en texto plano en ningún lado, tanto las claves de inicio de sesión como las llaves de autenticación son cifradas en pasos previos para posteriormente ser almacenadas en la base de datos como se puede ver en la figura 38.

Figura 44

Credenciales Cifradas

id_qr	password	llave	id_estudiante
1	7î'□®□x□è.ð^-á Êmó| 'šžÌl0¼□/Ê´ê	½7N¿ƒ(èERlôréD□D-(.ÖîO.k fó÷è9d	1
2	+è□0œm½š·□□€ã□Ë{		8
3	\¥–µú□®®±_Ä□□¥½µ	òÃp□Ë□²ÿ–¯À¤1□#□$û»□ë;ÿ&›'$f□©Ž□	5
4	QË□_niÑÙèt¢□|ió@8,Jƒ¾/Ø□úßï'1•ŸÄ	I<ˆ—Çˆèó5+BàÄð+•i6bflíŸ%Ë"*Œ□ÜÞÖ	10
5	IDf□—lÅî¨öÒµw](%$4©¦–©P©ôO¯¹□â×Ì	õuèµŒÙ□ñ.ñÁ·þ‹¹SX·plðMëDú□dûE‰ó4	4
6	#Z*&4b¾□ú□8µáš§.ªâEÖ[ôtŠi¡□±€7 □+	6u¯"òß□□l›ëìZ□~ã ™½†©…□}B†œeyþèÞ	6

En el proceso de cifrado se generan las llaves de cifrado y un vector de inicialización tanto para cifrar las claves de inicio de sesión, como para cifrar las llaves de autenticación, estas son almacenadas en ficheros con acceso restringido, haciendo que estos archivos no sean accesibles desde internet, lo que permite tener mayor seguridad en el manejo de estos datos sensibles, los ficheros utilizados y los permisos que manejan se muestran en las figuras 39 y 40.

Figura 45

Ficheros de Almacenamiento de Llaves de Cifrado y Vectores de Inicialización

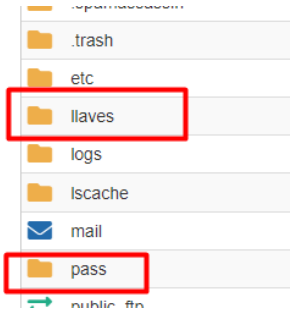


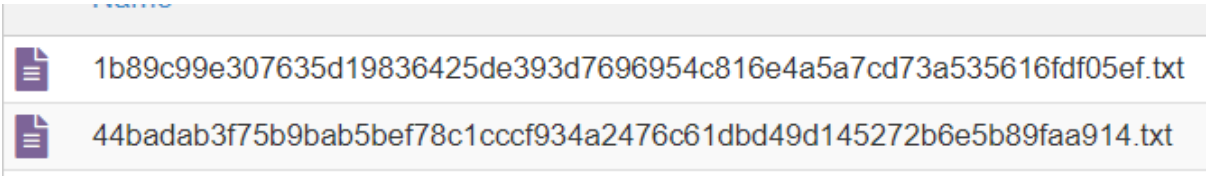
Figura 46

Permisos de los Ficheros

Cuando se crea un archivo en el sistema, es importante contar con un mecanismo que permita identificarlo de manera unívoca, para poder realizar búsquedas y tener una mejor gestión. Una forma de lograr esto es mediante el uso del hash sha-256, una función criptográfica que genera una secuencia de caracteres única a partir de cualquier dato de entrada. En este caso, se utiliza el email del usuario como dato de entrada para generar el nombre del archivo, lo que garantiza que cada archivo creado tenga un identificador único y fácilmente manejable a nivel de programación. Además, al ser una codificación unidireccional, no es posible recuperar el email original a partir del nombre del archivo, lo que brinda una capa adicional de seguridad en el manejo de la información.

Figura 47

Archivos que Contienen Información del Cifrado



La codificación en hexadecimal es una forma de representar la información almacenada en un archivo mediante una secuencia de caracteres que utiliza una base numérica de 16 dígitos. Esta forma de codificación se utiliza ampliamente en informática, ya que permite manejar de manera más sencilla los datos al presentarlos en una forma más compacta y fácilmente legible. Al almacenar la información de un archivo en formato hexadecimal, se puede acceder y manipular los datos de manera más eficiente y con menor probabilidad de errores, lo que resulta especialmente útil en aplicaciones que requieren un manejo preciso y seguro de la información. En la figura 42 se muestra la información almacenada en un archivo.

Figura 48

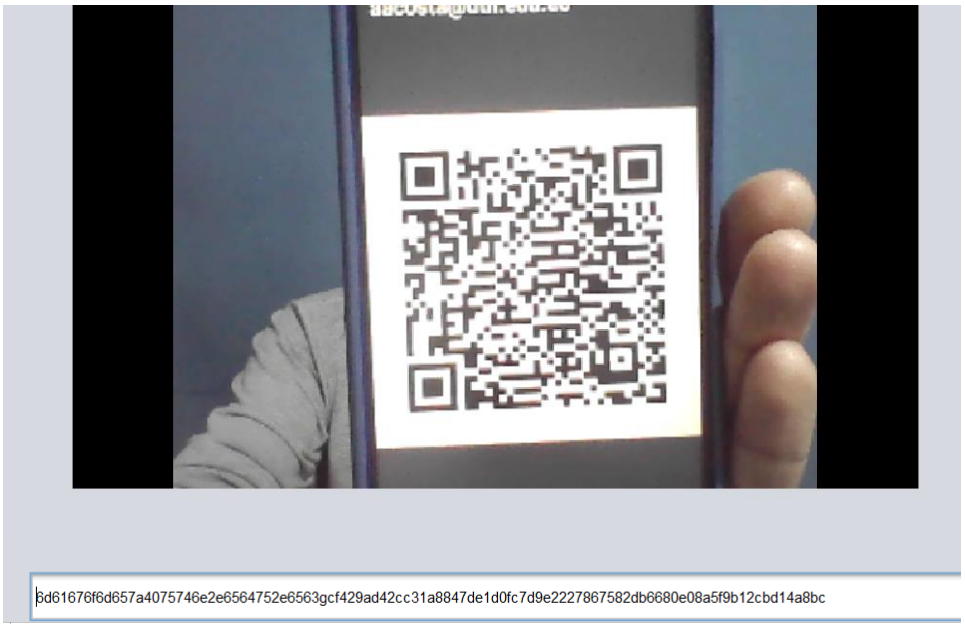
Llave de Cifrado y Vector de Inicialización

- **Comprobación de la Información Almacenada en el Código QR**

Para efecto de prueba, en la parte inferior se muestra el mensaje que se extrae del código QR, que en una versión final sería quitada esta sección; Al ejecutar el programa la cámara web se activa y empieza a capturar imágenes en búsqueda de un código QR, una vez identificado captura la información y envía los datos para la respectiva inspección y validación de los mismos, dependiendo de la respuesta actuara de una manera u otra.

Figura 49

Lectura del Código QR



Usando un conversor de texto a hexadecimal se puede comprobar la primera parte del mensaje que contiene el código QR en donde se muestra el email codificado en hexadecimal, que se observa en la figura 44.

Figura 50

Correo Codificado en Hexadecimal

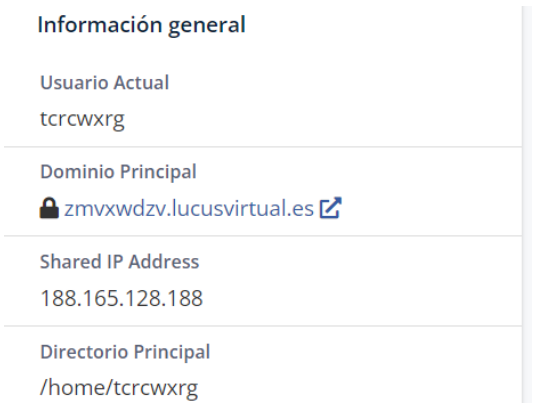
Por cuestiones prácticas se opta por ingresar un carácter que no existe en formato hexadecimal que es la letra “g”, la cual permitirá distinguir las partes del mensaje dentro del QR, la siguiente parte vendría a ser la clave de acceso que a primera vista se encuentra codificada en hexadecimal, pero es un texto previamente cifrado con AES-256.

4.2.3. Prueba 3

El servidor de hosting tiene un URL y una dirección IP como se muestra en la figura 35, con esta información es posible realizar una captura de paquetes filtrando la información con estos datos.

Figura 51

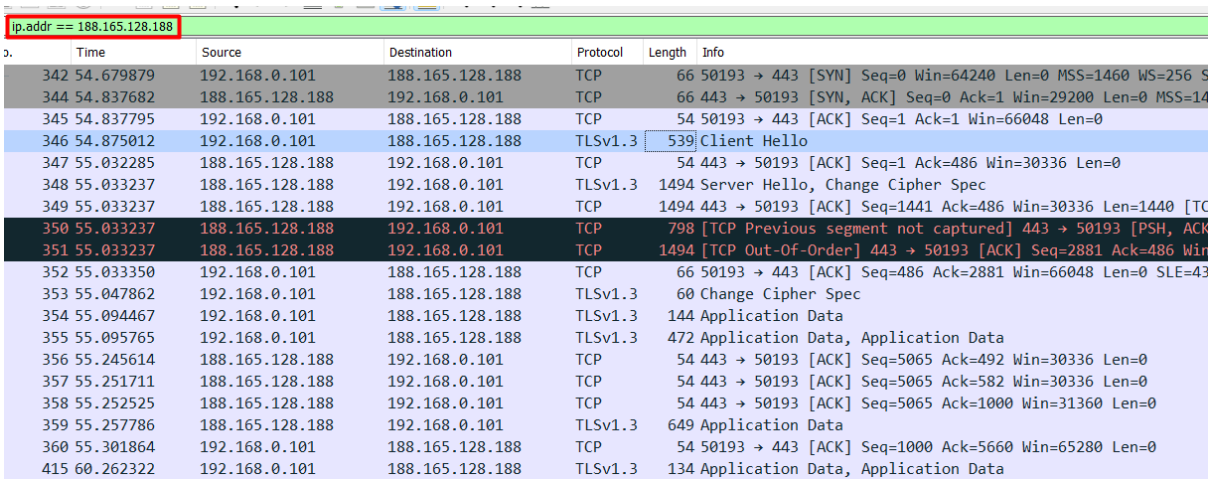
Información General - Servidor Hosting



Usando la herramienta de WireShark se captura los paquetes al momento de realizar el acceso al servidor de hosting y filtrando los paquetes con la dirección IP 188.165.128.188 y se muestra que se maneja el protocolo TLS, protocolo de cifrado que protege la transferencia de datos e información. La captura de paquetes se observa en la figura 36.

Figura 52

Captura de Paquetes - WireShark

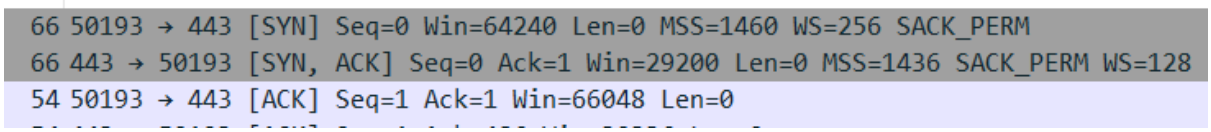


- **Apretón de Manos de Tres Vías**

O three-way handshake, este proceso implica la negociación de parámetros entre dos dispositivos para establecer una conexión confiable. El cliente envía un paquete SYN al servidor, el servidor responde con un paquete SYN-ACK, y luego el cliente envía un paquete ACK para confirmar la conexión. Una vez completado el proceso de tres vías, se establece una conexión TCP bidireccional entre los dispositivos y pueden comenzar a intercambiar datos.

Figura 53

Three-Way Handshake



- **Protocolo TLS en la Comunicación**

TLS v1.3 tiene la capacidad de establecer una conexión segura sin requerir intercambio de información de identidad del cliente en la fase inicial de la conexión, utilizando un cifrado más fuerte y eficiente para proteger la comunicación, así como un conjunto de algoritmos de autenticación más seguros. Esto mejora la privacidad de los usuarios y la velocidad de la conexión.

Figura 54

Datos Encriptados con TLS

```
TLSv1.3 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
    Opaque Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 85
    Encrypted Application Data: 074206b87aa2fb3671d4f78c21e5a3a73aa7b8ea94526f777c8a0e228a26a45ec4e9088c…
    [Application Data Protocol: Hypertext Transfer Protocol]
```

4.2.4. Prueba 4

La aplicación tiene interfaces simples, que buscan ser intuitivas para que el usuario no tenga dificultad en usarlas, además, la aplicación maneja mensajes de error y de confirmación de las acciones realizadas para una mejor comprensión del proceso.

Figura 55

Mensaje de la Aplicación

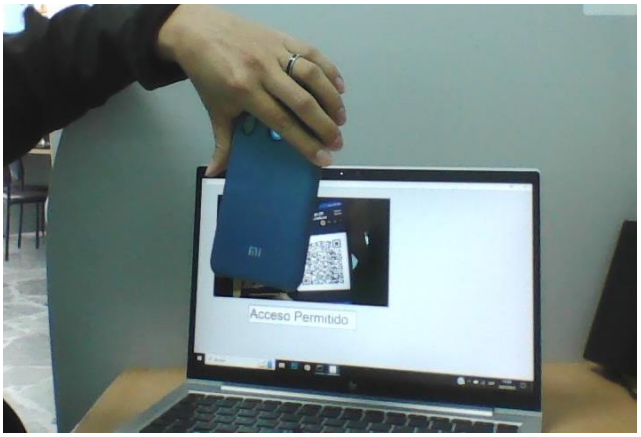
Figura 56

Interfaces de la Aplicación

El funcionamiento del sistema fuera de un entorno de pruebas se desempeña sin problemas en el dispositivo proporcionado, en la figura 49 se puede observar el funcionamiento.

Figura 57

Funcionamiento del Sistema



Las pruebas se realizaron para registrar el acceso de varias cuentas de usuarios, lo que se puede comprobar en la tabla biblioteca en donde se registra el momento en el que se registra el código QR por el computador.

Figura 58

Registro de Usuarios

- **Servicio 1**

El servicio que se usaría son los computadores a los que tienen acceso los estudiantes en la biblioteca, para llevar a cabo la prueba de funcionamiento en un estudiante que no esté relacionado con este proyecto, una estudiante universitaria utilizó el sistema, con una nueva cuanta y registro desde el principio.

Figura 59

Registro en la Aplicación – Servicio 1

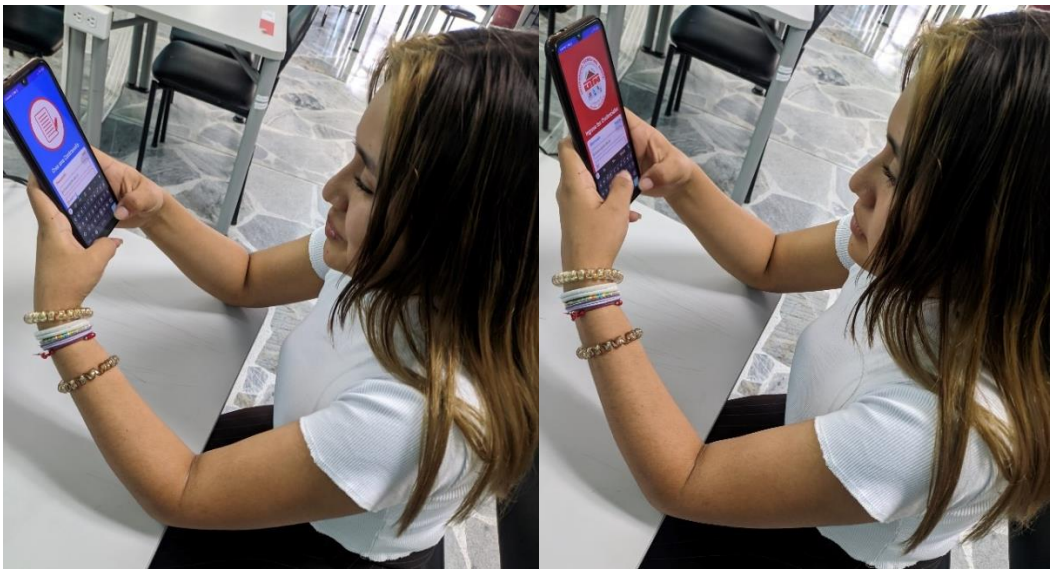
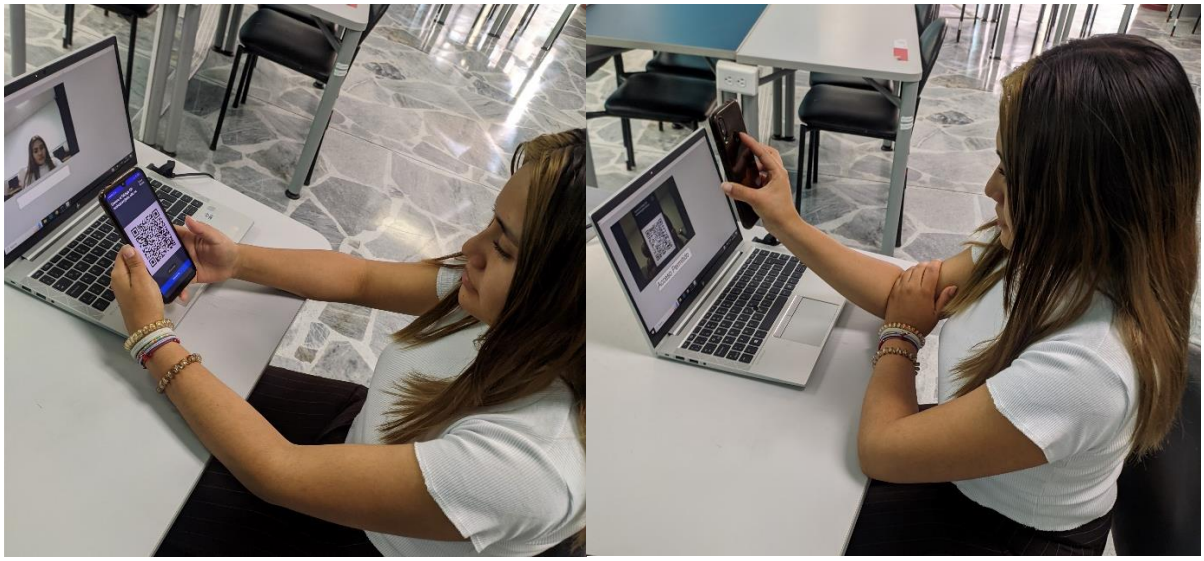


Figura 60

Autenticación en del Estudiante – Servicio 1



Realizar pruebas de funcionamiento con usuarios externos y registrar sus acciones en la base de datos permite evaluar el desarrollo de la aplicación en cuanto a funcionalidad y mejorar la experiencia del usuario.

Figura 61

Registro del Estudiante – Servicio 1

+ Opciones

estudiante	fecha	id_estudiante
Mateo Andrés Gómez	2023-03-25 00:15:39	6
Mateo Andrés Gómez	2023-03-25 00:15:48	6
Mateo Andrés Gómez	2023-03-27 02:27:01	6
Mateo Andrés Gómez	2023-03-27 02:47:46	6
Mateo Andrés Gómez	2023-03-27 02:48:32	6
Mateo Andrés Gómez	2023-03-28 22:30:12	6
Evelin Mishelle Huera Vargas	2023-03-28 22:54:10	11
Evelin Mishelle Huera Vargas	2023-03-28 22:54:14	11

El registro detallado de las acciones del usuario, incluyendo su nombre, fecha, hora e ID correspondiente, es útil para comprender mejor cómo se está utilizando la aplicación y tomar medidas para mejorar la seguridad y proteger la información de los usuarios.

- **Servicio 2**

El segundo servicio en el que se realizó la prueba de funcionamiento fue en la zona de ingreso a la biblioteca, donde se encuentran los computadores para el registro del estudiante al ingresar a la biblioteca, de igual manera un estudiante hizo uso del servicio.

Figura 62

Autenticación del Estudiante - Servicio 2

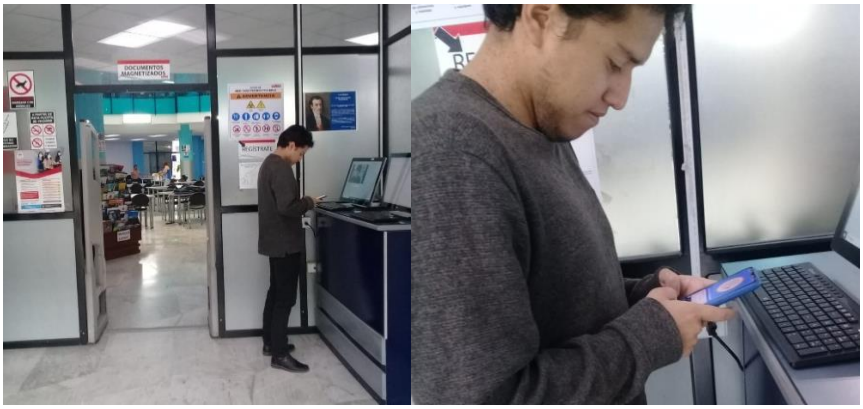


Figura 63

Autenticación en del Estudiante – Servicio 2

De igual manera se puede corroborar el registro del estudiante en este servicio, observando en la base de datos el nuevo nombre y la fecha en la que se registra como se puede ver en la figura 54.

Figura 64

Registro del Estudiante – Servicio 1

Evelin Mishelle Huera Vargas	2023-03-28 22:56:30	11
Evelin Mishelle Huera Vargas	2023-03-28 22:56:33	11
Mateo Andrés Gómez	2023-03-31 15:17:39	6
Mateo Andrés Gómez	2023-03-31 15:20:11	6
Mateo Andrés Gómez	2023-03-31 15:31:38	6
Christopher Alejandro Ortega Chulde	2023-03-31 17:00:04	12
Christopher Alejandro Ortega Chulde	2023-03-31 17:00:21	12

4.2.5. Resumen de las Pruebas

La aplicación presenta interfaces simples e intuitivas con una sólida protección de seguridad en el manejo de llaves de cifrado, archivos de configuración y datos de usuario. La utilización de codificación hexadecimal y el protocolo TLS para la transferencia de datos aseguran una eficiente manipulación y transmisión de información, minimizando la probabilidad de errores y vulnerabilidades. Las pruebas realizadas demostraron un buen funcionamiento de la aplicación, lo que sugiere una experiencia positiva para el usuario, los resultados obtenidos se muestran en la tabla 21.

Tabla 20

Resumen de las Pruebas Realizadas

Nº	Pruebas	Resultados	Desempeño
1	Funcionamiento del Sistema	La prueba de archivos de configuración muestra que están alojados en un servidor de hosting y pueden ser accedidos desde la aplicación. La consulta de peticiones ejecuta el código del archivo y devuelve los datos requeridos o un mensaje de error si no se cumplen los parámetros. Los permisos establecidos en el sistema garantizan un mayor control de acceso y seguridad, minimizando los riesgos de vulnerabilidad ante posibles ataques.	Satisfactorio

2	Cifrado AES-256 para Encriptar la Información y Manejo de Llaves de cifrado.	Las llaves de cifrado y las claves de inicio de sesión son cifradas y almacenadas en ficheros con acceso restringido para mayor seguridad. Se utiliza el hash sha-256 para generar nombres de archivo únicos a partir del email del usuario, lo que brinda una capa adicional de seguridad en el manejo de la información. La codificación en hexadecimal permite acceder y manipular los datos de manera más eficiente y con menor probabilidad de errores.	Satisfactorio
3	Seguridad en la Comunicación del Sistema	En la prueba se realizó una captura de paquetes del servidor de hosting utilizando la herramienta WireShark. Se filtró la información con la dirección IP y se observó que se utiliza el protocolo TLS para proteger la transferencia de datos.	Satisfactorio
4	Integración en los servicios de la biblioteca de la UTN	Se realizaron pruebas de registro de acceso de varias cuentas de usuarios y se registró la información en la base de datos. Además, se probó el sistema con una estudiante universitaria ajena al proyecto para evaluar su funcionalidad y mejorar la experiencia del usuario. El registro detallado del usuario se realiza efectivamente.	Satisfactorio

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

La implementación de un método de criptografía eficaz requiere la consideración de diversos factores, como el nivel de seguridad requerido, el procesamiento necesario y la gestión de las llaves de cifrado. Cada uno de estos aspectos es fundamental para seleccionar el método de cifrado más adecuado a las necesidades de cada situación y garantizar la protección de la información que se desea cifrar.

Para delimitar los requerimientos del sistema es necesario considerar los objetivos, la identificación de los usuarios y sus necesidades que son aspectos fundamentales. Estos aspectos permiten tener una visión clara y precisa del sistema a implementar, garantizando su efectividad y eficiencia en su uso.,

El registro de este proceso en la base de datos con información detallada del estudiante y la fecha y hora en la que se realizó, permite una mejor gestión y seguimiento de las acciones realizadas por los usuarios en la aplicación.

Cifrar el contenido de un código QR proporciona una capa adicional de seguridad al proteger la información sensible que se maneja en él. De esta manera, se asegura la confidencialidad de los datos y se evita que terceros no autorizados tengan acceso a ellos.

Establecer parámetros claros para la creación, almacenamiento y eliminación de las claves de cifrado, permiten minimizar los riesgos de exposición y asegurar que la confidencialidad de los datos se mantenga intacta.

El uso de códigos QR cifrados se presenta como una alternativa viable y eficaz para lograr una autenticación segura de los usuarios en diversos servicios. Este método ofrece la

ventaja de requerir pocos recursos. Es importante estar al tanto de las opciones de seguridad disponibles y considerar aquellas que mejor se adapten a las necesidades de cada situación.

Para que una aplicación móvil sea instalada de forma segura requiere que esta sea distribuida por una tienda oficial, además, de contar con una firma digital que respalde su integridad, pasos necesarios a seguir cuando se requiera la distribución de la aplicación a los usuarios.

Las pruebas de integración de la aplicación con los servicios de la biblioteca no solo garantizan que esta pueda ser utilizar de manera eficiente por cualquier persona, sino que también puede ayudar a identificar cualquier problema o error en el proceso de registro y creación de cuentas.

Recomendaciones

Antes de implementar este sistema el entorno universitario, se requiere realizar una evaluación exhaustiva de los requisitos de configuración necesarios para garantizar un funcionamiento óptimo, se deben considerar factores adicionales, como la necesidad de acceso a la base de datos institucional y la posible complejidad del entorno, para garantizar el correcto funcionamiento del sistema y evitar problemas posteriores. Además, se recomienda que se realicen pruebas en un ambiente controlado antes de realizar la implementación en producción, para detectar y corregir cualquier problema o incompatibilidad antes de que afecte al funcionamiento del sistema en el entorno real.

Se recomienda manejar las llaves de cifrado con permisos de modificación, lectura y ejecución para reducir las brechas de seguridad. De esta manera, se garantiza un control adecuado sobre el acceso a la información cifrada y se minimiza el riesgo de exposición de información confidencial. Además, es importante tener en cuenta que la gestión de las llaves de cifrado debe ser realizada por personal autorizado y capacitado en temas de seguridad informática.

Para los interesados en el código del sistema, así como usuarios que deseen realizar un aporte a la aplicación, visitar el Repositorio de GitHub de Sistema QR.

Usar la codificación hexadecimal para un mejor procesamiento y transmisión de datos y obtener mayor precisión en el procesamiento de la información.

REFERENCIAS

Ahamed, S., & Asiful Mustafa, H. (2019). *A Secure QR Code System for Sharing Personal Confidential Information*. https://doi.org/10.1109/IC4ME247184.2019.9036521

Arimetrics. (2022). *Qué es el Código QR - Definición, significado y ejemplos*. https://www.arimetrics.com/glosario-digital/codigo-qr

ATICO34. (2020, julio 8). *¿Qué es la gestión de claves de cifrado? | Grupo Atico34*. https://protecciondatos-lopd.com/empresas/gestion-claves/

Bardají, E. (2022). *Cuando el malware se encuentra en la carta del restaurante (cartas por código QR)*. https://www.esedsl.com/blog/cuando-el-malware-se-encuentra-en-la-carta-del-restaurante-cartas-por-codigo-qr

Bradshaw, S., Brazil, E., & Chodorow, K. (2020). *Definitive guide: Powerful and scalable data storage*.

Castro Acuña, N., Leguizamón Páez, M., & Mora Lancheros, A. (2019). *Análisis de métodos y técnicas existentes para minimizar agujeros de seguridad al usar códigos QR | Revista UIS Ingenierías*. *18*. https://revistas.uis.edu.co/index.php/revistauisingenierias/article/view/9610/10012

Comeau, Andrew. (2015). *MySQL explained : your step-by-step guide to database design* (CreateSpace).

DENSO WAVE. (2022, octubre 9). *QR Code Solution - DENSO WAVE*. https://www.denso-wave.com/en/system/qr/product/sqrc.html

developers. (2020, junio 3). *Configuración de seguridad de la red - Android Developers*. https://developer.android.com/training/articles/security-config?hl=es-419

el Tiempo. (2021, diciembre 19). *Códigos QR: amenazas en ciberseguridad y cómo protegerse*. https://www.eltiempo.com/tecnosfera/novedades-tecnologia/codigos-qr-amenazas-en-ciberseguridad-y-como-protegerse-640054

Enriquez, L., Gabriel, J., Casas, D., & Isabel, S. (2013). *USABILIDAD EN APLICACIONES MÓVILES*.

Escrivá, G., Rosa, G., Romero, M., David, S., Ramada, J., & Onrubia Pérez, R. (2013). *Seguridad informática*. MACMILLAN.

Espinosa Herrera, E., & Vesga Arias, E. (2006). *La importancia de la confiabilidad, la disponibilidad y la mantenibilidad en una base de datos*.

Evidian. (2021). *Los 7 métodos de Autentificación más utilizados*. https://www.evidian.com/

Flores Asenjo, S. J. (2019). *Introducción a los códigos Reed Solomon*.

FORTRA. (2022). *Gestión de identidades y accesos | HelpSystems*. https://www.fortra.com/es/soluciones/seguridad-informatica/gestion-de-identificacion-y-accesos

Freato, R. (2015). *Microsoft Azure Security*. https://books.google.com.ec/books?id=kurnBwAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Hernández Encinas, L., & Peinado Domínguez, A. (2012). *Una propuesta para el uso de códigos QR en la autenticación de usuarios*. http://recsi2012.mondragon.edu/es/programa/recsi2012_submission_17.pdf

Hipólito Martínez, J. Y. (2015). *Diseño de herramientas computacionales asociadas al modelo algorítmico AES, para la encriptación de datos en módulos Quick Response (QR) para plataformas Android*.

IBM. (2022). *¿Qué es la gestión de identidades y accesos? Definiciones IAM, SSO, MFA e IDaaS | IBM*. https://www.ibm.com/ar-es/topics/identity-access-management

IBM España. (2018). *La evolución de la automatización de procesos*. https://www.ibm.com/downloads/cas/RJGDMZ2D

ISO/IEC 18004. (2015). *Information technology - Automatic identification and data capture techniques - QR Code bar code symbology specification*.

Jalca Regalado, J. J., Castro Romero, V. F., Menéndez Azúa, M. D. J., Quimiz Murillo, L. R., Anzúles Parrales, G. R., Pilay Campozano, Y. H., & Pin Pin, Á. L. (2018). *REDES DE COMPUTADORAS*. 3ciencias. https://books.google.com.ec/books?id=rElVDwAAQBAJ&pg=PA107&dq=Cifrado+DES+aes&hl=es-419&sa=X&ved=2ahUKEwiA68W_lp37AhW8mIQIHbZjBCAQ6AF6BAgGEAI#v=onepage&q=Cifrado%20DES%20aes&f=false

Kaspersky. (2022). *Cifrado de datos y cómo hacerlo*. https://latam.kaspersky.com/resource-center/definitions/encryption

KeepCoding. (2022). *¿Qué es RSA en criptografía?* https://keepcoding.io/blog/que-es-rsa-en-criptografia/

Kocausta, D. (2020, junio 23). *¿Qué es el control de acceso?* https://dkocausta.medium.com/access-control-eri%C5%9Fim-kontrol%C3%BC-nedir-7a58405b3f73

Lake, J. (2018, diciembre 10). *What is RSA encryption and how does it work?* https://www.comparitech.com/blog/information-security/rsa-encryption/

Lewis, A. (2022). *Conceptos básicos de Bitcoins y Blockchains: una introducción a las criptomonedas y a la tecnología que las impulsa*. Mango. https://books.google.com.ec/books?id=sBRtEAAAQBAJ&pg=PT88&dq=esquema+criptografia+simetrica+y+asimetrica&hl=es-419&sa=X&ved=2ahUKEwiImvyFjqX7AhV1STABHSGED3oQ6AF6BAgGEAI#v=onepage&q=esquema%20criptografia%20simetrica%20y%20asimetrica&f=true

Lizárraga, C., Sara, C., & Díaz Martínez, L. (2007). *USO DE SOFTWARE LIBRE Y DE INTERNET COMO HERRAMIENTAS DE APOYO PARA EL APRENDIZAJE (USE OF FREE SOFTWARE AND INTERNET LIKE TOOLS OF SUPPORT FOR LEARNING)*. *10*(1), 83–100. http://www.gnu.org

Luque Ordóñez, J. (2012). *Códigos QR*. https://www.acta.es/medios/articulos/comunicacion_e_informacion/063009.pdf

Maldonado Cuautenco, S. J. (2016). *Implementación del criptosistema AES con permutación variable para el cifrado de imágenes en hardware*. IPN.

Maquen Nino, G. L., Teran Santa Cruz, F. E., del Castillo Castro, C. I., & Villon Prieto, R. D. (2022). Best Practices for Relational Database Optimization using Microsoft SQL. *Universidad Ciencia y Tecnología*, *26*(114), 29–38. https://doi.org/10.47460/uct.v26i114.588

Marqués, M. (2009). *UNIVERSITAT JAUME I DE CASTELLÓ Bases de Datos*.

Mendoza, A. G., Bolaños Burgos, F., Cedeño Sarmiento, C., & Saltos Rivas, W. R. (2020). La importancia de la autenticación multifactor para el usuario final en un entorno financiero.

Informática y Sistemas: Revista de Tecnologías de la Informática y las Comunicaciones, 4(1), 42. https://doi.org/10.33936/isrtic.v4i1.2347

MINTEL. (2021). *Política de Ciberseguridad - Acuerdo Ministerial 006-2021*. https://www.telecomunicaciones.gob.ec/wp-content/uploads/2021/06/Acuerdo-No.-006-2021-Politica-de-Ciberseguridad.pdf

Moreno Pérez, J. Carlos. (2014). *Programación*. 301.

Naeem, T. (2022, enero 31). *Software de gestión de bases de datos: características, tipos y usos*. https://www.astera.com/es/type/blog/database-management-software/

Olivares, C. (2019). *Elementos para una Metodología de Gestión de Identidad Digital en la Empresa*.

OpenWebinars. (2020, julio 16). *Qué es un lenguaje de programación*. https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/

Orozco Toledo, M., & Cerezo Castelo, S. (2019). *Propuesta de mejora para el control de acceso de los estudiantes al CRAI de la Universidad Estatal de Milagro por medio de la lectura de códigos QR en carnets estudiantiles*. http://repositorio.unemi.edu.ec/bitstream/123456789/4808/3/REVISION%20ANTIPLAGIO.pdf

Ortiz, N., Duarte, D., Mora, M., & Caicedo, F. (2013). *ARQUITECTURA Y DISEÑO DE BASES DE DATOS MÓVILES ARCHITECTURE AND DESIGN OF A MOBILE DATABASES*.

Otero, C. (2021, agosto 10). *Si tienes esta versión de Android no podrás acceder a Gmail, YouTube y otras apps de Google*. https://as.com/meristation/2021/08/10/betech/1628615650_670628.html

Paredes Valderrama, A. J. (2019). *ESQUEMA DE AUTENTICACIÓN MEDIANTE CÓDIGOS QR PARA INGRESO A INSTALACIONES RESTRINGIDAS* [Universidad de los Andes]. https://repositorio.uniandes.edu.co/bitstream/handle/1992/44790/u831014.pdf?sequence=1

Penna, E., & de León, M. (2016). *Implementación de un servicio de autenticación centralizado y gestión de identidades en la Universidad de la República*.

Pliego García, C. (2013). *Desarrollo de una aplicación generadora y lectora de códigos QR en Android*. Universidad Carlos III de Madrid.

Postigo Palacios, A. (2020). *Seguridad informática* (1a ed.). Paraninfo. https://books.google.com.ec/books?id=UCjnDwAAQBAJ&pg=PA133&dq=algoritmos+de+cifrado&hl=es-419&sa=X&ved=2ahUKEwjMy7HNqJz7AhWXRTABHVaJDuIQ6AF6BAgIEAI#v=onepage&q=algoritmos%20de%20cifrado&f=false

Pulido Romero, E., Callejas, J. E., Escobar Domínguez, O., Núñez Pérez, J. A., Rodríguez Zamora, R., & Zamorategui Berber, A. (2019). *Base de datos*. Grupo Editorial Patria. https://elibro-net.bdbiblioteca.universidadean.edu.co/es/ereader/bibliotecaean/121283?page=40

Rivero Hernández, D., Pérez Vázquez, R., & Vila Labrada, J. (2013). BASES DE DATOS MÓVILES. *TLATEMOANI*, *14*. http://www.eumed.net/rev/tlatemoani/index.htm

rockcontent. (2018, septiembre 27). *25 tipos de lenguaje de programación más usados en la actualidad*. https://rockcontent.com/es/blog/tipos-de-lenguaje-de-programacion/

Rosado, S. (2015, febrero 8). *Tabla comparativa de los sistemas gestores de base de datos*. http://desarrollowebydesarrolloweb.blogspot.com/2015/02/tabla-comparativa-de-los-sistemas.html

Saranya, K., Student, V. B., & Student, P. A. (2016). SQRC-based Vehicle and ID-address Proof Verification System. *ijartet.com International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)*, *3*(3). https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwicz_W0pqT7AhUeSjABHVl7AXUQFnoECBMQAQ&url=https%3A%2F%2Fijartet.com%2F1308%2Fv3s3sriramanathan%2Fconference&usg=AOvVaw3HYTH9g5Alyt5eCWRhac4K

Serrato Losada, H. D. (2019). *Comparación de Métodos Criptográficos para la Seguridad Informática*. https://core.ac.uk/download/pdf/344723797.pdf

Talently. (2022, abril 26). *¿Qué es Android Studio?* https://talently.tech/blog/que-es-android-studio/

Three Points. (2022). *Cifrado simétrico y asimétrico: ¿Cuál es mejor?* https://www.threepoints.com/blog/cifrado-asimetrico-y-simetrico

Tiwari, S. (2017). An introduction to QR code technology. *Proceedings - 2016 15th International Conference on Information Technology, ICIT 2016*, 39–44. https://doi.org/10.1109/ICIT.2016.38

Tiwari, S., & Sahu, S. (2014). *A Novel Approach for the Detection of OMR Sheet Tampering Using Encrypted QR Code*. https://doi.org/10.1109/ICCIC.2014.7238430

Triguero Ortega, J., Guerrero López, M., & Crespo García del Castillo, E. (2005). *Introducción a la criptografía. Historia y actualidad*. Universidad Castilla La Mancha. https://books.google.com.ec/books?id=ZH-IAgAAQBAJ&printsec=frontcover&dq=clave+publica+cifrado&hl=es-419&sa=X&ved=2ahUKEwjru5D_0aL7AhVzRTABHfWcDTAQ6AF6BAgMEAI#v=onepage&q=clave%20publica%20cifrado&f=true

Universidad de Oviedo. (2005). *Redes*.

Universidad Veracruzana. (2021). *Contraseñas y medidas complementarias*.

Urcuqui, L., García, P., & Osorio, Q. (2018). *Ciberseguridad: un enfoque desde la ciencia de datos*. Universidad ICESI. https://elibro.net/es/ereader/utnorte/120435?page=22

Velasteguí López, E. (2019). Las ventajas y desventajas del internet en la sociedad. *ConcienciaDigital*, *2*(1), 35–45. https://doi.org/10.33262/concienciadigital.v2i1.928

Viteri Ojeda, J. C., Andrade Álvarez, C. E., Valencia Ortiz, N. P., & Castro Viteri, C. A. (2020). Usos y beneficios de las aplicaciones móviles en las empresas de la ciudad de Riobamba. *ConcienciaDigital*, *3*(1.2), 6–19. https://doi.org/10.33262/concienciadigital.v3i1.2.1165

VIU. (2021). *Qué es la criptografía y cuáles son sus usos*. https://www.universidadviu.com/es/actualidad/nuestros-expertos/que-es-la-criptografia-y-cuales-son-sus-usos

Zúñiga Sáenz, R. (2005). *Operaciones: concepto, sistema, estrategia y simulación*.

ANEXOS

ANEXO 1. Ficha de Requerimientos

Proyecto de titulación: "Sistema de Autenticación Basados en Códigos QR para Acceso a Servicios Proporcionados por la UTN"

Objetivo de análisis: Definir de forma clara todos los requerimientos que requiere el sistema para brindar seguridad y cumplir con los objetivos planteados dentro del trabajo de investigación.

Fecha de realización: 25 de enero del 2023

Lista de Stakeholders

Descripción	Abreviatura
Requerimientos de Stakeholders	StSR
Requerimientos de Sistema	SySR
Requerimientos de Arquitectura	SrSH

Requerimiento de Stakeholders (StSR)			
Requerimientos Operacionales			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
StRS1	Conexión segura para envío de datos	Aunque los usuarios de las computadoras que son extremo de una comunicación puedan estar tranquilos en cuanto a la seguridad de estas computadoras, la red de comunicaciones siempre es un punto de desconfianza. La prevención ante los ataques a la red suele pasar siempre por el uso de alguna u otra manera de técnicas de criptografía (Universidad de Oviedo, 2005).	**Alta**
StRS2	Autenticación de Usuarios	La autenticación es el mecanismo principal de un modelo estándar de seguridad, es importante diferenciar entre la autenticación y la autorización, siendo la autorización otro elemento importante en una estrategia de seguridad de la información (Mendoza et al., 2020).	**Alta**
StRS3	El sistema mejorara la seguridad en la administración de las claves por medio del cifrado	El uso de algoritmos de cifrado y sistemas centralizados de autenticación permiten crear un entorno de seguridad fuerte (Postigo Palacios, 2020).	**Alta**
StRS4	El sistema permite gestionar las claves de cifrado	La gestión de claves de cifrado es administrar el ciclo de vida completo de las claves criptográficas. Esto incluye: generar, usar, almacenar, archivar y eliminar claves. La protección de las claves de cifrado incluye limitar el acceso a las claves física, lógica y mediante el acceso de usuario / rol (ATICO34, 2020).	**Alta**
StRS5	El sistema tiene la capacidad de leer códigos QR y administrar su información	Tienen ventajas como la decodificación de alta velocidad, interpretación del código en cualquier dirección, corrección de errores y diferentes versiones (Tiwari, 2017)	**Alta**
Requerimientos de Usuarios			
StRS6	Los usuarios se identificarán y manejarán el acceso por medio de una aplicación móvil	En el mundo en línea se presenta una gran cantidad de oportunidades para hacer uso de la Identidad Digital como las transacciones de datos, la identidad es un componente clave para garantizar la seguridad, control de acceso, personalización e incluso el cumplimiento normativo (Olivares, 2019).	**Alta**
StRS7	La aplicación permitirá administrar la clave y contraseña	El uso de la contraseña es el método más utilizado, esto significa que su gestión es uno de los aspectos más importantes para asegurarlos sistemas. Las contraseñas	**Alta**

		deficientes o mal custodiadas pueden favorecer el acceso y el uso no autorizado de los datos y servicios (Universidad Veracruzana, 2021).	
Requerimiento de Sistema (SySR)			
Requerimientos de Performance			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
SySR1	Comprobación de información en un tiempo corto	Se puede medir el rendimiento de una base de datos en términos de tiempo, haciendo referencia al tiempo que toma al servidor de base de datos responder una solicitud o requerimiento de información solicitada. En tal sentido, es crucial que el tiempo de respuesta a los datos que se requieren de la base de datos sea el mínimo posible para que puedan atender solicitudes en menor tiempo lo que se traduce en mayor cantidad de clientes atendidos (Maquen Nino et al., 2022).	**Media**
SySR2	Se debe dar acceso durante toda la jornada de apertura de la biblioteca – 10 horas	En la fase de diseño de equipos o sistemas, se debe buscar el equilibrio entre la disponibilidad y el costo. La disponibilidad se expresa como el porcentaje de tiempo en que el sistema está listo para operar o producir (Espinosa Herrera & Vesga Arias, 2006).	**Media**
SySR3	El sistema permite gestionar a los usuarios identificándolos por medio de códigos QR	La gestión de identidades y accesos es parte indispensable para la seguridad de una empresa. Permite proteger a la entidad contra credenciales de usuario comprometidas y contraseñas no seguras, que son una brecha de seguridad aprovechados por criminales informáticos (IBM, 2022).	**Alta**
Requisitos de Interfaz			
SySR4	Cámara para lectura del código QR	Para descifrar un código QR solamente se necesita un dispositivo con cámara de fotos y un lector compatible (Pliego García, 2013).	**Alta**
SySR5	Aplicación móvil con la que se puede identificar al usuario	La apps, son un software que se instala en dispositivos móviles o tablets para ayudar al usuario en una tarea definida; es decir, servir como asistente en operaciones y gestiones del día a día ya sea de carácter profesional, de ocio o entretenimiento (Viteri Ojeda et al., 2020).	**Alta**
SySR6	Protocolo de comunicación segura entre la aplicación y la base de datos	El internet es una de las herramientas que ha marcado tendencia en la actualidad, gracias a todos los beneficios que brinda su aplicación, que ha permitido que muchos artefactos electrónicos de uso cotidiano, tales como celulares, tablets, computadoras, etc., puedan comunicarse con la nube (Velasteguí López, 2019).	**Alta**

SySR7	Conexión a Internet	Se hace uso de Internet y la Web para acceder a la información en tiempo real, aprovechando que los dispositivos móviles (Ortiz et al., 2013).	**Alta**
Requerimientos de Uso			
SySR8	Facilidad de uso de la aplicación móvil	Las aplicaciones se espera que tengan cierto grado de aceptación entre los usuarios, que depende de las características de la aplicación. Dentro de estos atributos uno de los considerados más importantes es la usabilidad, que indica la facilidad con la que un usuario puede usar una aplicación de software (Enriquez et al., 2013).	**Medio**
SySR9	Debe permitir la gestión de varios usuarios	La eficiencia en la gestión de acceso es la optimización en cada caso de uso, tanto para, la creación de roles, solicitudes, aprobaciones eliminaciones y certificación en toda la organización (FORTRA, 2022).	**Alta**
Requerimientos de Seguridad			
SySR10	Cifrado AES para encriptar las claves	AES (Advanced Encryption Standard), es uno de los algoritmos que garantiza los tres aspectos importantes en la criptografía (confiabilidad, integridad y disponibilidad), siendo hoy en día el más utilizado (Maldonado Cuautenco, 2016).	**Medio**
SySR11	Seguridad en la comunicación entre la aplicación móvil y la base de datos	La comunicación entre una aplicación Android y phpMyAdmin se realiza a través del protocolo HTTPS utilizando solicitudes y respuestas HTTP (developers, 2020).	**Alta**
SySR12	Inicio de sesión en la aplicación móvil con ingreso de usuario y contraseña	El identificador y la contraseña son el par de autenticación más conocido. Simple, robusto, incluso rústico (Evidian, 2021).	**Alta**
SySR13	Acceso a la base de datos mediante credenciales	La base de datos puede gestionar el acceso a la información, permitiendo autorizar a ciertos usuarios y restringiendo a otros (Pulido Romero et al., 2019).	**Alta**
Requerimientos Modo/Estado			
SySR14	El sistema diferencia dos estados que son: la lectura de los datos y la comprobación de los mismos	No es posible mejorar un sistema si primero no se cuenta con una definición de su estado (Zúñiga Sáenz, 2005).	**Alta**

Requerimiento de Arquitectura (SrSH)			
Requerimientos de Software			
Nomenclatura	**Requerimiento**	**Descripción**	**Prioridad**
SrSH1	Uso de software libre	Está basado en el principio de colaboración comunitaria, no hay costo por licencias ni actualizaciones, las herramientas son independientes de las plataformas, no desaparecen, se mejoran con el tiempo y en especial son adaptables y configurables a las necesidades del usuario (Lizárraga et al., 2007).	**Baja**
SrSH2	Base de datos rápida y de fácil gestión	La primera cosa para tener en cuenta al elegir un sistema de gestión de los elementos de una base de datos para tu organización es la facilidad de uso, la seguridad de datos, la funcionalidad, la capacidad de integración, el soporte y desarrollo, la escalabilidad y el costo (Marqués, 2009).	**Media**
SrSH3	Aplicación compatible con versiones antiguas del S.O. móvil	Siendo Android la versión de SO más extendido y con los cientos de millones de smartphones que hay en el mundo, son literalmente millones de usuarios y usuarias los/as que usan versiones más antiguas de Android (Otero, 2021).	**Baja**
SrSH4	Lenguaje orientado a objetos	Smith, (2011) afirma que, este lenguaje crea una arquitectura de software, que permite flexibilidad a través del diseño modular, además de que ayuda a prevenir el código inmanejable, por lo cual es un lenguaje muy útil para el esquema diseñado.	**Media**
Requerimientos de Diseño			
SrSH5	Almacenamiento y actualización de credenciales y claves	La Gestión de Identidades y Accesos es muy importante en la seguridad, ya que permite la gestión de identidades digitales y el acceso de usuarios a la información, sistemas y recursos de la entidad. Para este fin, se emplean políticas, programas y tecnologías para reducir los riesgos de autenticación, inicio de sesión y autorización en el acceso a servicios de la organización (FORTRA, 2022).	**Alta**
SrSH6	Creación, almacenamiento y eliminación de las claves de acceso	Las claves son análogas a la combinación de una caja fuerte. Si un adversario conoce una combinación segura, la caja fuerte más fuerte no proporciona seguridad contra la penetración (ATICO34, 2020).	**Alta**
SrSH7	Creación del código QR en la aplicación	Los códigos QR son fáciles de usar y se pueden incluir en infinidad de sitios, desde una página web, una app o en cualquier soporte que admita una impresión. Esto genera múltiples ventajas y una gran versatilidad (Arimetrics, 2022)**.**	**Media**

SrSH8	El sistema debe permitir la lectura de la información del código QR y la recuperación del mensaje al descifrar la información	Saranya et al. (2016) en su trabajo menciona que, dentro de los códigos QR se maneja varios tipos de información, que pueden llegar a ser de carácter confidencial, estos códigos al ser fáciles de crear y leer se busca la manera de hacerlos más seguros, surgiendo de esta manera una variante de estos códigos como son los códigos de respuesta rápida segura (SQRC)	**Alta**
Requerimientos Lógicos			
SrSH9	La información se manejará de forma segura almacenando información en la base de datos, a la que se hará consultas por medio de la aplicación móvil	El usuario requeriría poder acceder y actualizar la información de los archivos en los directorios de inicio de un servidor o cliente de registros de una base de datos (Rivero Hernández et al., 2013).	**Alta**

Bibliografía.

Ahamed, S., & Asiful Mustafa, H. (2019). *A Secure QR Code System for Sharing Personal Confidential Information.* https://doi.org/10.1109/IC4ME247184.2019.9036521

Arimetrics. (2022). *Qué es el Código QR - Definición, significado y ejemplos*. https://www.arimetrics.com/glosario-digital/codigo-qr

ATICO34. (2020, julio 8). *¿Qué es la gestión de claves de cifrado? | Grupo Atico34*. https://protecciondatos-lopd.com/empresas/gestion-claves/

Bardají, E. (2022). *Cuando el malware se encuentra en la carta del restaurante (cartas por código QR)*. https://www.esedsl.com/blog/cuando-el-malware-se-encuentra-en-la-carta-del-restaurante-cartas-por-codigo-qr

Bradshaw, S., Brazil, E., & Chodorow, K. (2020). *Definitive guide: Powerful and scalable data storage*.

Castro Acuña, N., Leguizamón Páez, M., & Mora Lancheros, A. (2019). *Análisis de métodos y técnicas existentes para minimizar agujeros de seguridad al usar códigos QR | Revista UIS Ingenierías*. *18*. https://revistas.uis.edu.co/index.php/revistauisingenierias/article/view/9610/10012

Comeau, Andrew. (2015). *MySQL explained : your step-by-step guide to database design* (CreateSpace).

DENSO WAVE. (2022, octubre 9). *QR Code Solution - DENSO WAVE*. https://www.denso-wave.com/en/system/qr/product/sqrc.html

developers. (2020, junio 3). *Configuración de seguridad de la red - Android Developers*. https://developer.android.com/training/articles/security-config?hl=es-419

el Tiempo. (2021, diciembre 19). *Códigos QR: amenazas en ciberseguridad y cómo protegerse*. https://www.eltiempo.com/tecnosfera/novedades-tecnologia/codigos-qr-amenazas-en-ciberseguridad-y-como-protegerse-640054

Enriquez, L., Gabriel, J., Casas, D., & Isabel, S. (2013). *USABILIDAD EN APLICACIONES MÓVILES*.

Escrivá, G., Rosa, G., Romero, M., David, S., Ramada, J., & Onrubia Pérez, R. (2013). *Seguridad informática*. MACMILLAN.

Espinosa Herrera, E., & Vesga Arias, E. (2006). *La importancia de la confiabilidad, la disponibilidad y la mantenibilidad en una base de datos*.

Evidian. (2021). *Los 7 métodos de Autentificación más utilizados*. https://www.evidian.com/

Flores Asenjo, S. J. (2019). *Introducción a los códigos Reed Solomon*.

FORTRA. (2022). *Gestión de identidades y accesos | HelpSystems*. https://www.fortra.com/es/soluciones/seguridad-informatica/gestion-de-identificacion-y-accesos

Freato, R. (2015). *Microsoft Azure Security*. https://books.google.com.ec/books?id=kurnBwAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Hernández Encinas, L., & Peinado Domínguez, A. (2012). *Una propuesta para el uso de códigos QR en la autenticación de usuarios*. http://recsi2012.mondragon.edu/es/programa/recsi2012_submission_17.pdf

Hipólito Martínez, J. Y. (2015). *Diseño de herramientas computacionales asociadas al modelo algorítmico AES, para la encriptación de datos en módulos Quick Response (QR) para plataformas Android*.

IBM. (2022). *¿Qué es la gestión de identidades y accesos? Definiciones IAM, SSO, MFA e IDaaS | IBM*. https://www.ibm.com/ar-es/topics/identity-access-management

IBM España. (2018). *La evolución de la automatización de procesos*. https://www.ibm.com/downloads/cas/RJGDMZ2D

ISO/IEC 18004. (2015). *Information technology - Automatic identification and data capture techniques - QR Code bar code symbology specification*.

Jalca Regalado, J. J., Castro Romero, V. F., Menéndez Azúa, M. D. J., Quimiz Murillo, L. R., Anzúles Parrales, G. R., Pilay Campozano, Y. H., & Pin Pin, Á. L. (2018). *REDES DE COMPUTADORAS*. 3ciencias. https://books.google.com.ec/books?id=rElVDwAAQBAJ&pg=PA107&dq=Cifrado+DES+aes&hl=es-419&sa=X&ved=2ahUKEwiA68W_lp37AhW8mIQIHbZjBCAQ6AF6BAgGEAI#v=onepage&q=Cifrado%20DES%20aes&f=false

Kaspersky. (2022). *Cifrado de datos y cómo hacerlo*. https://latam.kaspersky.com/resource-center/definitions/encryption

KeepCoding. (2022). *¿Qué es RSA en criptografía?* https://keepcoding.io/blog/que-es-rsa-en-criptografia/

Kocausta, D. (2020, junio 23). *¿Qué es el control de acceso?* https://dkocausta.medium.com/access-control-eri%C5%9Fim-kontrol%C3%BC-nedir-7a58405b3f73

Lake, J. (2018, diciembre 10). *What is RSA encryption and how does it work?* https://www.comparitech.com/blog/information-security/rsa-encryption/

Lewis, A. (2022). *Conceptos básicos de Bitcoins y Blockchains: una introducción a las criptomonedas y a la tecnología que las impulsa*. Mango. https://books.google.com.ec/books?id=sBRtEAAAQBAJ&pg=PT88&dq=esquema+criptografia+simetrica+y+asimetrica&hl=es-

419&sa=X&ved=2ahUKEwiImvyFjqX7AhV1STABHSGED3oQ6AF6BAgGEAI#v=onepage&q=esquema%20criptografia%20simetrica%20y%20asimetrica&f=true

Lizárraga, C., Sara, C., & Díaz Martínez, L. (2007). *USO DE SOFTWARE LIBRE Y DE INTERNET COMO HERRAMIENTAS DE APOYO PARA EL APRENDIZAJE (USE OF FREE SOFTWARE AND INTERNET LIKE TOOLS OF SUPPORT FOR LEARNING)*. *10*(1), 83–100. http://www.gnu.org

Luque Ordóñez, J. (2012). *Códigos QR*. https://www.acta.es/medios/articulos/comunicacion_e_informacion/063009.pdf

Maldonado Cuautenco, S. J. (2016). *Implementación del criptosistema AES con permutación variable para el cifrado de imágenes en hardware*. IPN.

Maquen Nino, G. L., Teran Santa Cruz, F. E., del Castillo Castro, C. I., & Villon Prieto, R. D. (2022). Best Practices for Relational Database Optimization using Microsoft SQL. *Universidad Ciencia y Tecnología*, *26*(114), 29–38. https://doi.org/10.47460/uct.v26i114.588

Marqués, M. (2009). *UNIVERSITAT JAUME I DE CASTELLÓ Bases de Datos*.

Mendoza, A. G., Bolaños Burgos, F., Cedeño Sarmiento, C., & Saltos Rivas, W. R. (2020). La importancia de la autenticación multifactor para el usuario final en un entorno financiero. *Informática y Sistemas: Revista de Tecnologías de la Informática y las Comunicaciones*, *4*(1), 42. https://doi.org/10.33936/isrtic.v4i1.2347

MINTEL. (2021). *Política de Ciberseguridad - Acuerdo Ministerial 006-2021*. https://www.telecomunicaciones.gob.ec/wp-content/uploads/2021/06/Acuerdo-No.-006-2021-Politica-de-Ciberseguridad.pdf

Moreno Pérez, J. Carlos. (2014). *Programación*. 301.

Naeem, T. (2022, enero 31). *Software de gestión de bases de datos: características, tipos y usos*. https://www.astera.com/es/type/blog/database-management-software/

Olivares, C. (2019). *Elementos para una Metodología de Gestión de Identidad Digital en la Empresa*.

OpenWebinars. (2020, julio 16). *Qué es un lenguaje de programación*. https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/

Orozco Toledo, M., & Cerezo Castelo, S. (2019). *Propuesta de mejora para el control de acceso de los estudiantes al CRAI de la Universidad Estatal de Milagro por medio de la lectura de códigos QR en carnets estudiantiles*. http://repositorio.unemi.edu.ec/bitstream/123456789/4808/3/REVISION%20ANTIPLAGIO.pdf

Ortiz, N., Duarte, D., Mora, M., & Caicedo, F. (2013). *ARQUITECTURA Y DISEÑO DE BASES DE DATOS MÓVILES ARCHITECTURE AND DESIGN OF A MOBILE DATABASES*.

Otero, C. (2021, agosto 10). *Si tienes esta versión de Android no podrás acceder a Gmail, YouTube y otras apps de Google*. https://as.com/meristation/2021/08/10/betech/1628615650_670628.html

Paredes Valderrama, A. J. (2019). *ESQUEMA DE AUTENTICACIÓN MEDIANTE CÓDIGOS QR PARA INGRESO A INSTALACIONES RESTRINGIDAS* [Universidad de los Andes]. https://repositorio.uniandes.edu.co/bitstream/handle/1992/44790/u831014.pdf?sequence=1

Penna, E., & de León, M. (2016). *Implementación de un servicio de autenticación centralizado y gestión de identidades en la Universidad de la República*.

Pliego García, C. (2013). *Desarrollo de una aplicación generadora y lectora de códigos QR en Android*. Universidad Carlos III de Madrid.

Postigo Palacios, A. (2020). *Seguridad informática* (1a ed.). Paraninfo. https://books.google.com.ec/books?id=UCjnDwAAQBAJ&pg=PA133&dq=algoritmos+de+cifrado&hl=es-419&sa=X&ved=2ahUKEwjMy7HNqJz7AhWXRTABHVaJDuIQ6AF6BAgIEAI#v=onepage&q=algoritmos%20de%20cifrado&f=false

Pulido Romero, E., Callejas, J. E., Escobar Domínguez, O., Núñez Pérez, J. A., Rodríguez Zamora, R., & Zamorategui Berber, A. (2019). *Base de datos*. Grupo Editorial Patria. https://elibro-net.bdbiblioteca.universidadean.edu.co/es/ereader/bibliotecaean/121283?page=40

Rivero Hernández, D., Pérez Vázquez, R., & Vila Labrada, J. (2013). BASES DE DATOS MÓVILES. *TLATEMOANI*, *14*. http://www.eumed.net/rev/tlatemoani/index.htm

rockcontent. (2018, septiembre 27). *25 tipos de lenguaje de programación más usados en la actualidad*. https://rockcontent.com/es/blog/tipos-de-lenguaje-de-programacion/

Rosado, S. (2015, febrero 8). *Tabla comparativa de los sistemas gestores de base de datos*. http://desarrollowebydesarrolloweb.blogspot.com/2015/02/tabla-comparativa-de-los-sistemas.html

Saranya, K., Student, V. B., & Student, P. A. (2016). SQRC-based Vehicle and ID-address Proof Verification System. *ijartet.com International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)*, *3*(3). https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwicz_W0pqT7AhUeSjABHVl7AXUQFnoECBMQAQ&url=https%3A%2F%2Fijartet.com%2F1308%2Fv3s3sriramanathan%2Fconference&usg=AOvVaw3HYTH9g5Alyt5eCWRhac4K

Serrato Losada, H. D. (2019). *Comparación de Métodos Criptográficos para la Seguridad Informática*. https://core.ac.uk/download/pdf/344723797.pdf

Talently. (2022, abril 26). *¿Qué es Android Studio?* https://talently.tech/blog/que-es-android-studio/

Three Points. (2022). *Cifrado simétrico y asimétrico: ¿Cuál es mejor?* https://www.threepoints.com/blog/cifrado-asimetrico-y-simetrico

Tiwari, S. (2017). An introduction to QR code technology. *Proceedings - 2016 15th International Conference on Information Technology, ICIT 2016*, 39–44. https://doi.org/10.1109/ICIT.2016.38

Tiwari, S., & Sahu, S. (2014). *A Novel Approach for the Detection of OMR Sheet Tampering Using Encrypted QR Code*. https://doi.org/10.1109/ICCIC.2014.7238430

Triguero Ortega, J., Guerrero López, M., & Crespo García del Castillo, E. (2005). *Introducción a la criptografía. Historia y actualidad*. Universidad Castilla La Mancha. https://books.google.com.ec/books?id=ZH-IAgAAQBAJ&printsec=frontcover&dq=clave+publica+cifrado&hl=es-419&sa=X&ved=2ahUKEwjru5D_0aL7AhVzRTABHfWcDTAQ6AF6BAgMEAI#v=onepage&q=clave%20publica%20cifrado&f=true

Universidad de Oviedo. (2005). *Redes*.

Universidad Veracruzana. (2021). *Contraseñas y medidas complementarias*.

Urcuqui, L., García, P., & Osorio, Q. (2018). *Ciberseguridad: un enfoque desde la ciencia de datos*. Universidad ICESI. https://elibro.net/es/ereader/utnorte/120435?page=22

Velasteguí López, E. (2019). Las ventajas y desventajas del internet en la sociedad. *ConcienciaDigital*, *2*(1), 35–45. https://doi.org/10.33262/concienciadigital.v2i1.928

Viteri Ojeda, J. C., Andrade Álvarez, C. E., Valencia Ortiz, N. P., & Castro Viteri, C. A. (2020). Usos y beneficios de las aplicaciones móviles en las empresas de la ciudad de Riobamba. *ConcienciaDigital*, *3*(1.2), 6–19. https://doi.org/10.33262/concienciadigital.v3i1.2.1165

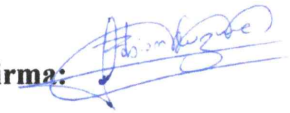
VIU. (2021). *Qué es la criptografía y cuáles son sus usos*. https://www.universidadviu.com/es/actualidad/nuestros-expertos/que-es-la-criptografia-y-cuales-son-sus-usos

Zúñiga Sáenz, R. (2005). *Operaciones: concepto, sistema, estrategia y simulación.*

Realizado por: Jean Rodríguez

Firma: 

Revisado por: Fabián Cuzme (Director)

Firma: 

ANEXO 2. Casos de Usos

El anexo que se presenta a continuación tiene como objetivo describir detalladamente los casos de uso del sistema. Estos casos de uso representan las diferentes interacciones que los usuarios pueden tener con el sistema para lograr sus objetivos. Cada caso de uso se describe en términos de su nombre, descripción, actores involucrados, precondiciones, flujos de eventos principales y alternativos, y postcondiciones. Además, se proporciona una breve explicación sobre la importancia de cada caso de uso en el contexto general del sistema. Con esta información, se busca brindar una guía detallada y completa sobre el funcionamiento del sistema para los usuarios y cualquier otra persona interesada en su desarrollo y uso.

Tabla 21

Ingreso de Credenciales

Casos de Uso	**Ingreso de Credenciales**
ID. del Caso de Uso	**01 Estado Registro / Lógin**
Actor	Usuario
Descripción	El usuario se registra en la aplicación o accede con credenciales almacenadas
Precondiciones	Si el usuario requiere el login a la aplicación se debe haber registrado previamente sus credenciales
Flujo Principal	1. El usuario ingresa las credenciales en los cuadros de texto de la aplicación. 2. El usuario da clic en el botón de la aplicación. 3. La aplicación envía los datos al servidor PHP. 4. El servidor PHP comprueba la validez de los datos. 5. El servidor PHP envía un mensaje de confirmación. 6. La aplicación avanza a la siguiente actividad.
Flujos Alternos	Punto 5 en el flujo principal A. El mensaje enviado desde el servidor PHP es un mensaje de “Error” o “Credenciales Incorrectas”. 1. El usuario debe volver a ingresar las credenciales.

	2. El usuario da clic en el botón de la aplicación. 3. La aplicación retorna al punto 6.
Flujos de Excepción	Punto 2 en el flujo principal A. Las credenciales ingresadas son incorrectas 1. La aplicación cancela el envío de credenciales al servidor. 2. La aplicación muestra un mensaje de error al usuario. 3. La aplicación retorna al punto 1 del flujo principal.
Post- condiciones	Credenciales comprobadas

Tabla 22

Generar Código QR

Casos de Uso	**Generar Código QR**
ID. del Caso de Uso	**02 Estado Generar / Mostrar**
Actor	Usuario
Descripción	El usuario da clic a el botón “Generar QR” para requerir un código de autenticación.
Precondiciones	Se debe haber ingresado las credenciales correctamente.
Flujo Principal	1. El usuario da clic a el botón “Generar QR” 2. La aplicación recopila la información de credenciales de usuario. 3. La aplicación envía los datos al servidor PHP. 4. El servidor comprueba la credenciales. 5. La aplicación recibe la información para la autenticación. 6. La aplicación genera el código QR. 7. La aplicación muestra el código QR en pantalla.
Flujos Alternos	Punto 5 en el flujo principal A. La aplicación recibe un mensaje de “Error al Generar” 1. La aplicación muestra un mensaje de error 2. La aplicación retorna al punto 1.
Flujos de Excepción	Punto 4 en el flujo principal A. El servidor muestra obtiene un inconsistencia en los datos 1. El usuario debe volver al caso de uso 01 Estado Registro / Lógin

Post- condiciones	Código QR Generado

Tabla 23

Leer Código QR

Casos de Uso	**Leer Código QR**
ID. del Caso de Uso	**03 Estado Captura / Lectura**
Actor	Lector QR
Descripción	El lector está en búsqueda de un código QR, en cuanto detecta uno envía los datos para su comprobación.
Precondiciones	El lector debe estar activo.
Flujo Principal	1. El lector recepta imágenes por la cámara web. 2. El lector captura un frame donde se encuentra un código QR. 3. El lector transforma la información del código en texto. 4. El texto receptado de almacena en una variable para comprobar su información. 5. El lector envía la información al servidor PHP. 6. El servidor desencripta la información. 7. La información es comprobada con la base de datos. 8. Se recibe un mensaje de “Acceso Permitido”.
Flujos Alternos	Punto 5 en el flujo principal A. El mensaje recibido es de “Acceso Denegado” 1. El lector retorna al punto 1.
Post- condiciones	Recibe la autenticación de la información.

Tabla 24

Encriptar Datos

Casos de Uso	**Encriptar Datos**
ID. del Caso de Uso	**04 Estado Generar / Encriptar / Guardar**
Actor	Servidor PHP
Descripción	El servidor PHP recibe datos a encriptar, crear la llave y vector de inicialización para el cifrado, una vez cifrada la información, se comparte el mensaje cifrado y se guarda la llave y el vector de inicialización.
Precondiciones	Recibir la orden desde la aplicación.
Flujo Principal	1. El servidor PHP recibe la petición junto a los datos desde la aplicación. 2. El servidor crea bytes aleatorios para la llave de cifrado y el vector de inicialización. 3. Se emplea el algoritmo de cifrado AES-256-CBC junto con la llave de cifrado y el vector de inicialización para encriptar los datos. 4. El servidor crea un archivo con el nombre del email del usuario codificado con sha-256. 5. El servidor escribe dentro del archivo la llave de cifrado y el vector de inicialización. 6. El servidor presenta la información encriptada con AES.
Flujos Alternos	Punto 4 en el flujo principal A. El servidor reescribe un archivo existente. 1. El lector continua al punto 5.
Post- condiciones	Se obtiene la información encriptada.

ANEXO 3. Descripción del Código de la Aplicación

El código de la aplicación se encuentra en el repositorio de GitHub. La aplicación establece varios atributos por actividad, los cuales son explicados en las siguientes tablas,

Tabla 25

Atributos de la Actividad Registro

Actividad Registro		
Atributo	**Tipo**	**Descripción**
t_email	EditText	Cuadro de texto donde se ingresa el email
t_password	EditText	Cuadro de texto donde se ingresa la contraseña
b_registrar	Button	Botón que permite el registro de la información en la base de datos con la función *setOnClickListener*
url	String	Indica la dirección url del servidor, para hacer la solicitud de registro
response	String	Guarda la respuesta del servidor después de realizar la petición de registro de datos
onCreate	Método	Inicia el llamado de las variables de ingresados en la parte gráfica y ejecuta el método onClick
onClick	Método	Se ejecuta al momento de hacer clic en el botón registro y ejecuta el método *insertarDatos*
insertarDatos	Método	Guarda los datos ingresados en los cuadros de texto, hace la petición al servidor php utilizando *POST* y ejecuta el método *onResponse*
onResponse	Método	Permite identificar la información recibida del servidor he indicar si hubo un error en el proceso
getParams	Método	Crea las variables que serán enviadas por método *POST* en la petición al servidor web
login	Método	Permite el paso de la actividad registro a la actividad login

Tabla 26

Atributos de la Actividad Login

Actividad Login		
Atributo	**Tipo**	**Descripción**
t_email	EditText	Cuadro de texto donde se ingresa el email
t_password	EditText	Cuadro de texto donde se ingresa la contraseña
str_email	String	Guarda la información de email en formato string
str_password	String	Guarda la información del password en formato string
url	String	Indica la dirección url del servidor, para hacer la solicitud de login
response	String	Guarda la respuesta del servidor después de realizar la petición de registro de datos
onCreate	Método	Inicia el llamado de las variables de ingresados en la parte gráfica, busca las preferencias de inicio de sesión con *sharedPreferences* y permita dar acceso a la actividad Main de forma más rápida
login	Método	Comprueba que los campos no estén vacíos, guarda las variables ingresadas y los enviar al servidor web usando *POST* Ejecuta el método onResponse
onResponse	Método	Recepta la respuesta del servidor web para dar acceso a la actividad principal o mostrar un mensaje de error y guarda los datos de inicio de sesión con *SharedPreferences* usando un modo privado
getParams	Método	Crea las variables que serán enviadas por método *POST* en la petición al servidor web
registro	Método	Permite el paso de la actividad login a la actividad registro

Tabla 27

Atributos de la Actividad Main

Actividad Main		
Atributo	**Tipo**	**Descripción**
ivCodigoQR	ImageView	Muestra el código QR una vez que se haya generado
btnGenerar	Button	Botón que permite iniciar el método para crear el código QR con la función *setOnClickListener*
username	String	Recupera el email almacenado en *sharedPreferences*
url	String	Indica la dirección url del servidor
response	String	Guarda la respuesta del servidor después de realizar la petición de registro de datos
onResponse	Método	Recibe la llave de cifrado enviada desde el servidor web para transformarla en una variable de tipo *secretKeySpec*. Después de cifrar el mensaje usando *cipher*, para un manejo más fácil se transforma los datos en formato hexadecimal con *BigInteger*, se transforma resultado en un mapa binario con *encodeBitmap* y lo muestra en el *ImageView* ivCodigoQR.
getParams	Método	Crea las variables que serán enviadas por método *POST* en la petición al servidor web
cerrar_sesion	Método	Permite borrar los datos almacenados de inicio de sesión
onBackPressed	Método	Evita que se tomen capturas de pantalla desde esta actividad

JFrame Form - camera.java		
Atributo	**Tipo**	**Descripción**
jPanel1	jPanel	Presenta información en la interfaz gráfica
jPanel2	jPanel	Muestra la imagen capturada por la cámara web
image	BufferedImage	Almacena una imagen en la memoria en forma de matriz de píxeles

mensajeEviar	String	Almacena la información extraída del código QR
panel	WebcamPanel	Mostrará un stream de video capturado por una cámara web
webcam	Webcam	Proporciona métodos para acceder a características de la cámara web, como resolución, tasa de fotogramas, ajustes de brillo y contraste, etc.
camera()	Constructor	Llama al método *initWebcam()*
initWebcam()	Método	Ejecuta los procesos para el funcionamiento y representación de la cámara web
run()	Método	Controla la captura de imágenes de la cámara, las procesa y decodifica códigos QR. Si se detecta un código QR válido, se envía una solicitud HTTP para verificar el código y si es correcto, se cierra la cámara

Java Class- conexion.java		
Atributo	**Tipo**	**Descripción**
url	String	Indica la dirección url del servidor, para enviar los datos receptados del código QR
mensajeqr	String	Muestra la imagen capturada por la cámara web
image	StringBuilder	Almacenar una cadena modificable de caracteres, en este caso el código QR
response	StringBuilder	Almacenar una cadena modificable de caracteres, en este caso la respuesta del servidor
conexión()	Constructor	Establece una conexión HTTP con un servidor y envía un mensaje de texto *mensajeEnviar* a través de una solicitud *POST*. Luego, lee la respuesta del servidor y almacena el resultado en una variable response.
getMiVariable()	Método	Devuelve el valor de la variable de instancia *conexion.mensajeqr*