FACULTAD DE ARQUITECTURAS E INGENIERÍAS

Trabajo de fin de carrera titulado

"DISEÑO DE UN SISTEMA DE TRIPLE FACTOR DE AUTENTICACIÓN BASADO EN RECONOCIMIENTO DE SIMILITUD DE IMÁGENES"

Realizado por:

Juan Carlos Andrade Chávez

Director del proyecto:

Ing. Diego Fernando Riofrío Luzcano, PhD.

Como requisito para la obtención del título de MASTER EN TECNOLOGÍAS DE LA INFORMACIÓN

Quito, septiembre 2019

Diseño de un Sistema de Triple Factor de Autenticación basado en Reconocimiento de Similitud de Imágenes

DECLARACION JURAMENTADA

Yo, **JUAN CARLOS ANDRADE CHÁVEZ**, con cédula de identidad **171658711-6**, declaro bajo juramento que el trabajo aquí desarrollado es de mi autoría, que no ha sido previamente presentado para ningún grado a calificación profesional; y, que ha consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la UNIVERSIDAD INTERNACIONAL SEK, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

Juan Carlos Andrade Chávez

C.I: 1716587116

DECLARATORIA

El presente trabajo de investigación titulado:

“DISEÑO DE UN SISTEMA DE TRIPLE FACTOR DE AUTENTICACIÓN BASADO EN RECONOCIMIENTO DE SIMILITUD DE IMÁGENES”

Realizado por:

JUAN CARLOS ANDRADE CHÁVEZ

Como requisito para la Obtención del Título de:

MAGÍSTER EN TECNOLOGÍAS DE LA INFORMACIÓN

Ha sido dirigido por el profesor

ING. DIEGO FERNANDO RIOFRIO LUZCANO, PhD.

Quien considera que constituye un trabajo original de su autor

Ing. Diego Fernando Riofrio Luzcano, PhD.

DIRECTOR

PROFESORES INFORMANTES

Después de revisar el trabajo presentado, lo ha calificado como apto para su defensa oral ante el tribunal examinador.

MBA. Verónica Rodríguez Arboleda

MBA. Edison Estrella Mogollón

Quito, septiembre de 2019

DEDICATORIA

Dedico este trabajo a mi familia en especial a mis padres, Jorge y Guadalupe, a mis hermanos, Jorge y Anita, cuyo apoyo fue incondicional alentándome a no rendirme, y completar mis objetivos, a la universidad SEK que me brindó la oportunidad de culminar esta etapa de mi formación académica.

AGRADECIMIENTO

En primer lugar, a Dios que me dio la oportunidad de completar otra etapa en mi vida, a todos mis amigos que se mantuvieron pendientes de apoyarme con palabras de aliento para no rendirme, agradecer a toda mi familia, y mi director Diego Riofrío, que con paciencia y determinación me incentivo a la culminación de este trabajo, a todos mis demás maestros, que con su experiencia y sabiduría me aconsejaron como confrontar dificultades, finalmente a una persona muy especial que me brindo su tiempo y apoyo pero que lamentablemente no me pudo acompañar al final de esta etapa de mi vida.

RESUMEN

En estos últimos años ha existido una nueva tendencia de evolución en sistemas de autenticación, su meta es asegurar la información del usuario utilizando métodos de seguridad que respalden su identidad al momento de usar un sistema, no obstante, esto ha provocado que varios perpetradores busquen la manera de vulnerar dichas seguridades, aprovechando falencias aún no corregidas en los procesos de autentificación. Esta problemática demanda crear nuevos prototipos o diseños de seguridades. Es aquí donde este trabajo busca como objetivo primordial presentar una nueva forma de autenticar en un sistema usando un prototipo de un triple factor de seguridad basado en una interfaz de programación *API Restful* y una confirmación de similitud de imágenes, a través de una arquitectura de comunicación tipo cliente servidor entre un dispositivo móvil y el servidor. Para lo cual, se realizó un análisis de tipos de imágenes con el fin de encontrar cuáles tienen el menor error de similitud, al momento de comparar una imagen tomada con una cámara móvil con la original de la misma. Para luego utilizar un algoritmo de similitud de este tipo de imágenes como última capa de seguridad del prototipo.

ABSTRACT

In recent years there has been a new trend in the evolution of authentication systems, their goal is to secure user information using security methods that support their identity when using a system, however, this has caused several perpetrators to seek the way of violating said securities, taking advantage of flaws not yet corrected in the authentication processes. This problem demands the creation of new prototypes or safety designs. It is here that this work aims to present a new way to authenticate in a system using a prototype of a triple security factor based on a Restful API programming interface and a confirmation of similarity of images, through a communication architecture client server type between a mobile device and the server. For which, an analysis of types of images was performed in order to find which have the least similarity error, when comparing an image taken with a mobile camera with its original. To then use a similarity algorithm of this type of images as the last security layer of the prototype.

INDICE GENERAL DE CONTENIDOS

Índice General de Figuras

Índice General de Tablas

CAPÍTULO I
INTRODUCCIÓN

1.1 Problema de Investigación

1.1.1 Planteamiento del Problema

En un mundo globalizado y con tendencia a evolucionar, la tecnología juega un papel importante en el diario vivir, y por lo tanto la protección de información delicada del usuario final es un factor muy relevante (Chiriguayo, 2015).

Aunque en la actualidad los sistemas de autenticación han mejorado notablemente, todavía existen problemas de vulnerabilidad por usar métodos tradicionales con mecánicas y procesos muy conocidos (Tarazona, 2012). Con el fin de contrarrestar esto, se creó multifactor de autenticación (MFA), lo que ha impuesto una nueva forma de acceder a los portales.

Es así que alrededor de un 30% de empresas planean implementar MFA, mientras que un 51% que ya lo utilizan (Ometov et al., 2018). De esta forma, pequeñas y medianas empresas están continuamente implementando este servicio para mantener altos estándares de seguridad. Sin embargo, muchos de estos sistemas recaen en obsolescencia ya que utilizan métodos conocidos, por lo que deben mantenerse actualizados con nuevas mecánicas que puedan mejorar los métodos de autenticación.

Tal como lo dicen Nath, Asoke & Mondal (2016) “si bien la autenticación de tres factores puede parecer el remedio perfecto para proteger las redes y los recursos, existen muchos agujeros de seguridad contra los que este tipo de autenticación no protegerá.”

Por esta razón, se ha visto la necesidad de innovar algoritmos de autenticación, por el hecho de que recaen en la problemática de usar los procedimientos ya vulnerados que facilitan al intruso aplicar técnicas para forzar el acceso al portal (Alvarez, 2015).

La única forma de no recaer en el problema ya mencionado, es implementar nuevos procedimientos imperceptibles a los posibles intrusos, adaptando el sistema de acceso a estas innovadoras ideas. Utilizando para esto autenticaciones por medio de aplicativos del tipo API *Restful*[1] que usan *tokens* para validar las autorizaciones, del estilo llave pública y privada de cada usuario (Puerta, 2015).

1.1.2 Formulación del Problema

Del análisis anterior, el hecho de que los métodos de autenticación recaen en la problemática de usar procedimientos ya vulnerados, facilita a los intrusos el poder forzar el acceso disminuyendo así el nivel de confiabilidad de los sistemas de autenticación.

Esto se agrava debido a que los ingenieros aplican soluciones temporales a problemas de seguridad (parches), lo cual acarrea nuevas vulnerabilidades por donde los intrusos pueden encontrar otras debilidades en el sistema y así volver a comprometerlo (Saturnino, 2017).

1.1.3 Objetivo General

Implementar un sistema de autenticación de usuario que utilice un método de triple factor que utilice técnicas actuales de seguridad, y que facilite el acceso a un sistema además de que mejore el nivel de seguridad.

1.1.4 Objetivos Específicos

- Identificar las técnicas más adecuadas para que sean aplicadas como capas de seguridad en el prototipo de triple factor de autenticación, a través del análisis de metodologías usadas además de sus ventajas y desventajas.

[1] API Restful: Representational State Transfer está diseñado para aprovechar los protocolos existentes.

- Analizar los procedimientos de reconocimiento de similitudes en imágenes aplicables a un sistema de triple factor de autenticación mediante un estudio documental para la determinación del algoritmo más apropiado.
- Realizar una validación cuasi-experimental en varias pruebas de efectividad de similitud por pares de imágenes, que ayude en la determinación de el o los tipos de imágenes que mejor se acomodan a la solución.
- Desarrollar un prototipo de un sistema de triple factor de autenticación utilizando una arquitectura cliente servidor, validando así las técnicas de autenticación encontradas, entre esas la similitud de imágenes.
- Verificar la efectividad del sistema de autenticación con similitudes de imágenes como parte del proceso de autentificación de triple factor, implementando el algoritmo de similitud al prototipo se determinará la validez del mismo mediante pruebas de accesos.

1.1.5 Justificación

El uso de métodos y procedimientos ya conocidos respecto a la seguridad en la implementación de la autentificación y su constante vulneración hace que creación de nuevos paradigmas de seguridad aumente, y con ello se genere nuevos campos de investigación para el aseguramiento de la identidad del usuario.

Según datos del SENATICs (2015), se han incrementado considerablemente las formas para violar o romper contraseñas, ya que utilizan técnicas y herramientas de intrusión en sistemas que cada día son más sofisticadas

Por esta razón las empresas han optado por reforzar este aspecto utilizando sistemas más fuertes y completos, como son los sistemas de autenticación multifactor (MFA), cuyo objetivo es impedir intrusiones a usuarios no acreditados (Andalucía, 2016).

Con la utilización de MFA el aseguramiento de la identidad del usuario aumenta, ya que se aplica capas adicionales de protección además de los datos o rasgos que solo le pertenecen al individuo, de esta manera se aumenta barreras ante posibles ataques y dificulta al intruso corromper el sistema (NetIQ, 2016).

Para comunicaciones al servidor una arquitectura *API Restful* asegura un desarrollo ligero y funcional, pero que también es flexible de agregar una capa de seguridad robusta a la periferia del *API*, impidiendo amenazas de seguridad que han sido plagadas en la web (CA Technologies, 2015).

1.2 Marco Teórico

Este apartado inicialmente detalla conceptos de seguridad informática, para tener una visión amplia de los inicios, abordando temas como los sistemas de autenticación multi-factor, donde se particularizará el sistema de triple autenticación.

Debido a que la arquitectura propuesta es del tipo cliente-servidor, la principal idea es que ambas partes procesen funciones de autorización como: autenticación tradicional, verificación de credenciales usuario-servidor, etc. Estos factores deberán ser tratados a través del envió de datos encriptados, donde una metodología adecuada de desarrollo es *API Restful User Authentication*.

1.2.1 Seguridad Informática

"Se puede definir a la Seguridad Informática como cualquier medida que impida la ejecución de operaciones no autorizadas sobre un sistema o red informática, cuyos efectos pueden puedan conllevar daños sobre la información, comprometer su confidencialidad, autenticidad o integridad, disminuir el rendimiento de los equipos o bloquear el acceso de usuarios autorizados al sistema." (Vieites, 2011).

En palabras de Calvillo (2016) existen varias técnicas en la Seguridad Informática, donde su finalidad es garantizar la legibilidad de los datos manejados, utilizando procedimientos tales como:

- Codificación de la información
- Monitoreo de medios de comunicación
- Aplicación de tecnologías protectoras
- Mantenimiento de respaldos

De acuerdo a Rivero & Merida (2006) la seguridad Informática también se caracteriza por cumplir con: Integridad y confidencialidad de datos del usuario, mantener disponibilidad del sistema de información y autentificación de identidades, este último ítem es donde se enfoca el estudio.

1.2.2 Autentificación

Según Mateos (2005), un proceso de autenticación establece la identidad de alguna entidad bajo escrutinio. Por ejemplo, un viajero se autentica ante una guardia de fronteras presentando un pasaporte.

El pasaporte y el parecido con la fotografía adjunta se consideran pruebas suficientes de que el viajero es la persona identificada. Este proceso de validar el pasaporte (serie en una base de datos) y al evaluar la semejanza del viajero (fotografía), es una forma de autenticación (McDaniel, 2006).

Según Arencibia (2009) "autenticación es el proceso donde se verifica la identidad digital de un remitente por medio de un canal de comunicación el cual realiza una petición para conectarse a un sistema". Una vez confirmada la identidad se otorga la autorización la cual establece y delimita los recursos a los que puede acceder el usuario.

1.2.2.1 Sistemas de autenticación

En palabras de Perales (2011) los Sistemas de Autentificación se dividen en tres categorías relacionadas al usuario: Sistemas de conocimiento (password), posesión (tarjeta) y característica física (huella digital), básicamente lo que usan para verificar la identidad.

De acuerdo a Rouse (2014) un sistema donde se unifica a todos a los sistemas de autentificación mencionados anteriormente es MFA el cual combina estas categorías y las segrega creando una protección por capas.

Biométricos (Características Físicas)

Según Perales (2011) los sistemas basados en características del tipo físicas del usuario, son utilizados para la identificación mediante la inteligencia artificial o el

reconocimiento de formas. Existen de varios tipos reconocimiento, por ejemplo: iris, retina, huellas dactilares, geometría de la mano, escritura y voz.

Contraseñas (Conocimiento)

De acuerdo a Perales (2011) la autentificación basada en información es el aspecto secreto que solo el usuario dispone, su proceso se compone de una verificación de igualdad de un factor de conocimiento del individuo el cual puede ser una contraseña, un código o un número de identificación

Dispositivos (Posesión)

Perales (2011) menciona que un elemento o dispositivo físico que contiene datos de autenticación del usuario, por ejemplo, tarjetas, dispositivos usb, *dongle* criptográfico, etc. se establece como un aspecto de seguridad.

1.2.3 Multi-factor de autentificación (MFA)

Proskura (2017) define MFA, como un enfoque relacionado a la seguridad de autenticar un usuario, el cual proporciona más de un tipo de factor de identificación antes de que se pueda realizar una transacción. Los factores de identificación se refieren a algo que es de conocimiento del usuario (contraseña), un elemento que le pertenece (*token* de seguridad) o algo ligado al usuario (biométrica).

Para Rouse (2014) la finalidad de MFA es complicar la entrada de usuarios no autentificados, a un sistema sea este una base de datos, un dispositivo, una red o hasta una locación. MFA crea defensas por capas, que aun siendo comprometida una de estas las demás impiden al atacante completar su objetivo

1.2.3.1 Tipos de MFA

Una descripción más profunda de varios elementos de los sistemas de autentificación (Kon, 2019).

- **Código de acceso (*PassCode*):** Contraseña numérica como un número de identificación personal (*pin*).
- **Contraseña (*Password*):** Cadena de caracteres identificables para el usuario.
- **Desafío / Respuesta (*Challenge/Response*):** Respuestas a preguntas de desafío que pueden incluir información personal poco clara.
- **Doble Sistema de Autenticación:** Parte de un sistema MFA que posee una capa adicional de seguridad, en la mayoría de veces un token físico, como datos biométricos o *token* enviado al electrónico.
- **Tarjetas de banda magnética:** Tarjetas que contienen datos como números de identificación escritos en medios de almacenamiento magnético. Puede incluir otras características de seguridad, como una tarjeta de identificación de empleado con una foto en la parte frontal.
- **Códigos de seguridad de la tarjeta:** Códigos que están escritos físicamente en una tarjeta. Similar a las tarjetas de banda magnética con la diferencia que se solicita a los usuarios que ingresen el código para demostrar la validez de la tarjeta. En algunos casos, se escriben varios códigos en una cuadrícula y se pide a los usuarios que ingresen el código desde una fila y columna en particular.
- **Tarjetas inteligentes:** Son tarjetas con circuitos integrados que permiten la ejecución de cierta lógica programada, normalmente incluyen credenciales de autenticación, como certificados de clave pública.
- **Fichas de seguridad:** *Hardware*, como un dispositivo USB o teléfono móvil, que genera *tokens* sincronizados en el tiempo basados en una clave compartida con un servicio de autenticación. Dichos dispositivos pueden interactuar directamente con los servicios de autenticación. Alternativamente, pueden mostrar una contraseña única sincronizada en el tiempo para que los usuarios la ingresen.
- **Biometría (*Biometrics*):** Biometría como reconocimiento facial, de voz, escaneo de huellas dactilares, etc.

1.2.3.2 Sistema con Triple Factor de Autenticación

Un sistema del tipo triple factor de autenticación involucra dividir por tres capas elementos de autenticación referentes al usuario (Ometov et al., 2018). Dichos elementos pueden ser:

- Un factor que es conocimiento del usuario (contraseña)
- Un factor que es perteneciente al usuario (*token*, teléfono u otro dispositivo)
- Un factor inherente al usuario, puede ser una característica ligada al usuario (iris, huella dactilar, tono de voz)

Muchos de los sistemas de triple autenticación agregan varias capas de validación que son factores de conocimiento del usuario (contraseñas, señas particulares) o posesión (teléfono, *token*, tarjetas) (Muhammad & Tripathi, 2012).

En resumen, son medidas adicionales de seguridad para complementar una autenticación. Básicamente requerimientos adicionales a la solicitud de usuario y contraseña, la solicitud de un *token* numérico, una seña particular, escaneo de huella dactilar, añade mayor seguridad al sistema.

1.2.4 Desarrollo de Aplicativos Cliente Servidor para Autenticación

Arquitectura Cliente-Servidor modelo de comunicación flexible y escalable. Una arquitectura distribuida que provee al usuario acceso a la información sin importar su plataforma (Mariel, Abendaño, & Zulaica, 2004).

Un servidor del tipo cliente servidor provee una interoperabilidad posible de aplicar una interfaz de programación de aplicaciones del tipo *API REST* para el manejo de comunicaciones aprovechando el canal *HTML*.

1.2.4.1 API de autenticación de usuario (*API Restful User Authentication*)

Casi todas las *API REST* deben tener algún tipo de autenticación. Uno de los encabezados (*headers*) más comunes es llamado autorización (OWASP, 2017) .

Un *API* de autenticación debe diferenciar conceptos entre autenticación y autorización para conocer exactamente las funciones que van a cumplir.

Autentificación vs Autorización (*Authentication vs. Authorization*)

Es importante para comprender cómo funcionan las *API RESTful* y por qué se aceptan o rechazan los intentos de conexión (Rodríguez López, 2016). De esta forma:

- La autenticación es la verificación de las credenciales del intento de conexión. Este proceso consiste en enviar las credenciales del cliente al servidor de acceso remoto en forma de texto simple o cifrado mediante un protocolo de autenticación.
- La autorización es la verificación de que se permite al usuario. En pocas palabras vistas y permisos de los que puede hacer uso.

Una autorización se produce después de la autenticación exitosa.

Sánchez (2011) reafirma, la autenticación indica que usuario es y la autorización pregunta si tiene acceso a un determinado recurso.

Existen varios métodos comunes de autenticación en aplicativos del tipo *API* que se basan en algunos enfoques principales como autorización *Http*, *Oauth*, *Rest*, *API keys*, etc. Los cuales casi siempre se desarrollaron para resolver limitaciones en las comunicaciones tempranas y los sistemas de Internet, y como tales, generalmente utilizan arquitectura existente con implementaciones novedosas para permitir que se produzca la autenticación (CA Technologies, 2015).

Las *API REST* admiten la autenticación basada en *token* a través del encabezado de solicitud *Authtoken*. La *API* de inicio de sesión *POST* se utiliza para recuperar el *token* de autenticación. Una vez que se obtiene el *token* de autenticación, debe insertarse en el encabezado *Authtoken* para todas las solicitudes (CA Technologies, 2015).

A continuación, describir a profundidad a todos los enfoques que envuelven a un *API* revelará sus características y ventajas, así también como evolucionado.

HTTP Basic Authentication

En este enfoque, un agente de usuario *HTTP* proporciona un nombre de usuario y una contraseña para probar su autenticación. Los procedimientos como el uso de sesiones, cookies, páginas de inicio de sesión o alguna otra solución no son necesarias, y esto se debe al uso del encabezado *HTTP*, el cual no necesita métodos de ratificación de conexión o cualquier otro sistema de respuesta compleja (Evidian, 2015) .

API Keys

Las *API Keys* se crearon como una solución a los problemas de autenticación temprana de la autenticación básica *HTTP* y otros sistemas similares. En este enfoque, se asigna un valor único generado a cada primer usuario, lo que significa que el usuario es conocido. Cuando el usuario intenta volver a ingresar al sistema su clave única se usa para demostrar que son el mismo usuario que antes, contraseña que muchas veces es generada por su combinación de hardware y datos *IP* otras veces generada aleatoriamente por el servidor que los conoce (Dire, 2018).

OAuth

Es un método de autenticación y autorización, cuando este se utiliza únicamente para la autenticación, es lo que se conoce como "pseudo-autenticación" (Sánchez, 2011).

En este enfoque, el usuario inicia sesión en un sistema, el cual luego solicitará la autenticación, generalmente en forma de *token*. El usuario luego reenviará esta solicitud a un servidor de autenticación, que la rechazará o la permitirá. Desde aquí, el *token* se proporciona al usuario y luego al solicitante. Tal *token* puede luego ser verificado por el solicitante para su validación en cualquier momento, independientemente del usuario, y puede ser utilizado a lo largo del tiempo con un alcance y antigüedad de validez estrictamente limitados.

REST

Representational State Transfer (*REST*) Tipo de arquitectura de servicios web que es utilizada para desarrollos sencillos, rápidos y livianos usa como protocolos de comunicación *HTTP*. Los *API REST* admiten la autenticación basada en token a través del encabezado de solicitud *Authtoken*. La *API* de inicio de sesión *POST* se

utiliza para recuperar el *token* de autenticación. Una vez que se obtiene el *token* de autenticación, debe insertarse en el encabezado *Authtoken* para todas las solicitudes (CA Technologies, 2015).

CAPÍTULO II

ESTADO DEL ARTE

En los últimos años se han creado nuevas metodologías y procesos de seguridad, que tienen como finalidad resguardar información o datos importantes para el usuario, cada diferente método posee fortalezas y vulnerabilidades que han ayudado progresivamente a mejorar cada uno de ellos. Así como la creación de proyectos e investigaciones para fortalecer nuevas metodologías.

2.1. Métodos de Autenticación Actuales

2.1.1. Google Authenticator

Según Rouse (2014) este software es un nuevo método de autentificación lanzado en 2010, fue creado por el equipo de desarrollo de la empresa Google, y es un aplicativo de autenticación de dos factores (2FA), basado en la arquitectura *OTP* y actualizado con los algoritmos del tipo *TOTP* (*Time-based One-Time Password*) y *HTOTP* (*HMAC-based one-time password*), funcionalmente genera un *token* numérico temporalizado a un dispositivo móvil, dicho número es verificado por el sitio que soporta el autentificador (Developers, 2010).

2.1.2. Microsoft Authentificator

Simons (2016) lo define como un aplicativo de autentificación, desarrollado por la compañía *Microsoft*, fue liberado en el 2016, su arquitectura es del tipo Multifactor (MFA) disponible para dispositivos móviles, genera un código basado en tiempo el que es utilizado

para la verificación de dos pasos (2FA) maneja dos tipos notificaciones *push* o *SMS*, este software es catalogado como de alto nivel de seguridad, ligado a tecnologías de *Microsoft Windows* (LogmeIn, 2018).

2.1.3. Diccionario de Terminología

Varios términos y siglas utilizados vienen descritos a continuación para entendimiento completo del léxico complicado.

OTP (*One-Time Password*) Password Único: Sistema tipo calculadora especializada que proporciona bajo petición una contraseña única, dicha contraseña es validada y tiene una duración limitada así como una única utilización (Evidian, 2015b).

TOTP (*Time based One-Time Password*) Password Único basado en Tiempo: Variante del algoritmo *HOTP* especifica el cálculo de un valor de contraseña de un solo uso, basado en una representación del contador como un factor de tiempo (RFC-6238, 2011).

HTOTP (*HMAC-based One-Time Password*) Password Único basado en *HMAC*: Variación del *TOTP* cumple las misma funcionalidad solo que implementa criptografía *HMAC* también conocido como clave-hash (FIPS-198-1, 2008).

2.2. Multifactor de Autenticación

2.2.1. Autenticación Biométrica Multifactor Multimedia

Schultz (2012) patenta su invento, un dispositivo que puede recibir una solicitud para autenticar a un individuo, establece automáticamente una sesión multimedia con dicho usuario en respuesta a la recepción de la solicitud y captura un grupo de identificadores biométricos que lo relacionen desde la sesión multimedia.

El dispositivo puede realizar además un grupo de operaciones de autenticación biométrica utilizando los identificadores previamente analizados o capturados, con el fin de obtener un grupo de puntajes de autenticación para relacionarlos con los usuarios y determinar si se ha autenticado debidamente.

2.2.2. Sistema Multifactor de Autenticación

Singh, Rasansky, & Racho (2007) proponen la creación de sistemas y métodos para permitir la autenticación multifactorial de transacciones de cajeros automáticos (*ATM*) y transacciones en el punto de venta de un comerciante mediante la entrega de un *PIN* secundario a toda solicitud de transacción del usuario, dicho *PIN* es un numero generado aleatoriamente por única vez al teléfono móvil de un cliente vía *SMS*, el cliente debe responder dicho mensaje con su *PIN* de cliente o ingresando el *PIN* secundario del cajero automático para que la transacción pueda continuar.

Básicamente, el teléfono del cliente representa una terminal *PIN* móvil para varios dispositivos de pago utilizados en el sistema de punto de venta de un comerciante. Además, se puede permitir un nivel adicional de autenticación del cliente utilizando el teléfono móvil, lo que aumenta la seguridad de las transacciones en cajeros automáticos y los pagos sin efectivo.

2.2.3. Autenticación Multifactor para Interfaces Programáticas

Cavage, Behm, & Cabrera (2014) plantean involucrar la seguridad con la programación, un método implementado por computadora realiza una autenticación multifactor de un usuario antes de ejecutar una función de una interfaz programática. El método consiste en recibir en un servidor un código de usuario a través de una interfaz programática. El servidor calcula un código de servidor en respuesta al código de usuario, y compara el código de usuario con el código de servidor para verificar si el código de usuario corresponde al código de servidor. El servidor valida el código de usuario y ejecuta una función de la interfaz programática después de validar el código de usuario.

2.3.Empresas Populares que usan MFA

La importancia de ofrecer seguridad de datos a los clientes ha movido a grandes empresas a migrar a sistemas inteligentes que protejan datos sensibles del usuario final, este listado describe a compañías importantes que decidieron dar el gran paso a métodos más robustos de autentificación.

Blizzard: Compañía estadounidense fundada en 1994, desarrolladora de software de entretenimiento con populares videojuegos como *World of Warcraft, StarCraft, OverWatch*, entre otros (Entertainment, 2010), empleó varios sistemas de seguridad como los que se describen a continuación:

Token Autentificador: Un dispositivo electrónico con arquitectura *OTP*, imprimía códigos de seguridad de 6 dígitos para ser evaluados posteriormente por el servidor de autentificación, fue descartado tiempo después y remplazado por un nuevo aplicativo. El dispositivo se puede observar en la figura 1.

Figura 1 Token Autentificado Blizzard Fuente: https://www.cinemablend.com/games/Blizzard-Admits-Accounts-With-Authenticators-Have-Been-Hacked-42909.html

Autenticador *Blizzard*: Aplicativo con estándar 2FA, combina la autenticación de usuario y contraseña con dos modalidades, autorización mediante un mensaje de aprobación o generación del *token*. En la figura 2 se puede apreciar el aplicativo.

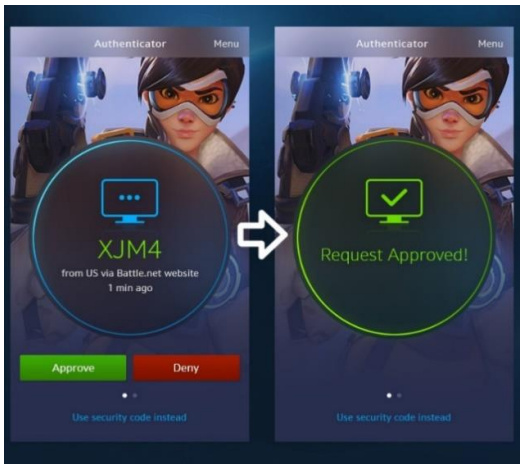

Figura 2 Autenticador Blizzard Fuente: https://worldofwarcraft.com/es-es/news/20815192/mantened-vuestras-cuentas-seguras-con-el-blizzard-authenticator

Facebook: Velasco (2018) señala que la famosa red social Facebook, implementó su nuevo método de doble autenticación basado en dos modalidades, el envío del *token* mediante *SMS* o el acoplamiento de aplicaciones como *Google Autheticador* o Duo *Mobile*, encargados de la generación de *tokens*. Las configuraciones de autorizar estos métodos se pueden ver en la figura 3.

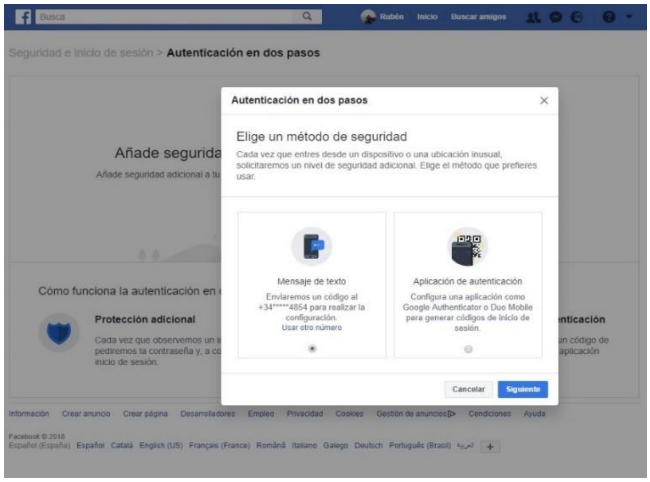

Figura 3 Configuración de 2FA en Facebook Fuente: https://www.redeszone.net/2018/05/24/activar-nueva-doble-autenticacion-facebook/

Google: La empresa americana dedicada a servicios de Internet, creó el aplicativo Autenticador de Google, software basado en la arquitectura *OTP*, generador de códigos de validación por tiempo, ofrece también un gestor de contraseñas para cuentas que usen el servicio autentificador de *Google* (Rouse, 2015). Una vista al aplicativo se encuentra en la figura 4.

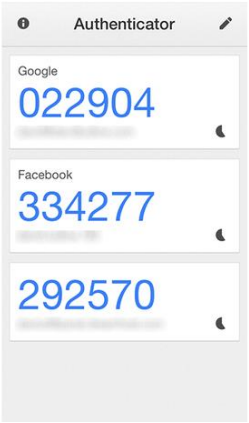

Figura 4 Aplicativo Google Authenticator Fuente: https://www.expertreviews.co.uk/software/internet-security/1400496/how-to-use-google-2-step-verification-and-authenticator-to

2.4. Procesamiento de Imágenes

2.4.1. Marcas de Agua para la Autentificación de Imágenes

Wu & Liu (2018) proponen un método de incrustación de datos para la autenticación de imágenes basado en la búsqueda de tablas en el dominio de frecuencia. Una marca de agua visualmente significativa y un conjunto de características simples se incrustan invisiblemente en la imagen marcada, las cuales almacenan la información legítima de la imagen de forma comprimida. El esquema puede detectar y localizar alteraciones con respecto a la imagen original.

2.4.2. Algoritmos de Análisis de Similitud de Imágenes

La tabla 1 es un análisis de investigaciones de algoritmos de similitud de imágenes, segmentados en las funciones que son necesarias para el prototipo. Los ítems marcados con X son las características que cumple el algoritmo.

Nombre (Autor)	Modificable	Código Abierto	Multiplataforma	Lenguajes	Observaciones
Librería GD (Pete Cooper)		X	X	C# y PHP	Genera Base de Datos
Libpuzzle (Chi Wong)		X		PHP	
ImageComparison (Vivek Moyal)	X	X	X	C# y PHP	
blink-diff (Empresa Yahoo)		X		C#	Modulo Pre construido

Tabla *1 Algoritmos de Similitud de Imágenes Fuente: Autor*

De acuerdo a los requerimientos del prototipo, para la selección del algoritmo es necesario que cumpla con todas las características detalladas en la tabla 1, adicional debe ser adaptable y flexible, razón por la cual se ha escogido el algoritmo *ImageComparison*, ya que gracias a su versatilidad de lenguaje C# y PHP nos permite migrar con facilidad.

2.4.3. Algoritmo *ImageComparison*

Vivek (2017) creó la clase *ImageComparison*, una técnica para determinar el grado de igualdad entre dos imágenes, por medio de una librería de funciones que busca adaptar y comparar similitudes entre éstas.

Sus funciones básicamente cumplen con, verificar el tipo de imagen, recortar a un bloque de 8 x8, extraer el valor medio de color, transformarlo a una matriz de bits y finalmente realizar una comparación de distancias en la matriz, entregando el coeficiente de similitud (*Hamming Distance*).

Una consideración final es, si el coeficiente de similitud calculado es menor o igual a 10 se puede deducir que la imagen es similar, de lo contrario, hay alguna diferencia en la imagen.

2.4.4. Recuperación de la Imagen con *Binary Hamming Distance*

La tesis de Landre & Truchetet (2007) refiere a *Hamming Distance* como la distancia obtenida al realizar una operación *XOR* un dato de binarización. Dicha distancia es conocida como coeficiente de similitud en el algoritmo de *ImageComparison*.

Esta investigación utilizó la técnica *Hamming Distance* en una colección de imágenes naturales reales que albergaban 10.000 imágenes y en una colección virtual de un millón de imágenes. Los resultados fueron óptimos tanto en términos de velocidad como de precisión.

2.5. Sistema Central

2.5.1. *Secure Quick Reliable Login* (*SQRL*)

Breeding & Smith (2013) definen a *SQRL* como un sistema criptográfico de autenticación, parte de un proyecto de *software* libre creado por Steve Gibson un ingeniero de software que se especializó en temas de seguridad, su principal propósito fue el de identificar usuarios y permitirles realizar autenticaciones, actualmente puede usarse como una capa adicional a sistemas tradicionales de seguridad.

SQRL funciona en base al uso de claves públicas y privadas obtenidas a través de una única clave maestra, a cada usuario se le asigna una identidad (clave pública) mientras que cada dominio una clave privada, dichas claves son cadenas de caracteres encriptados, cada clave pública tiene encadenado una clave privada, es así como verifica identidades de usuario. Se puede observar en la figura 5 el proceso que realiza *SQRL*.

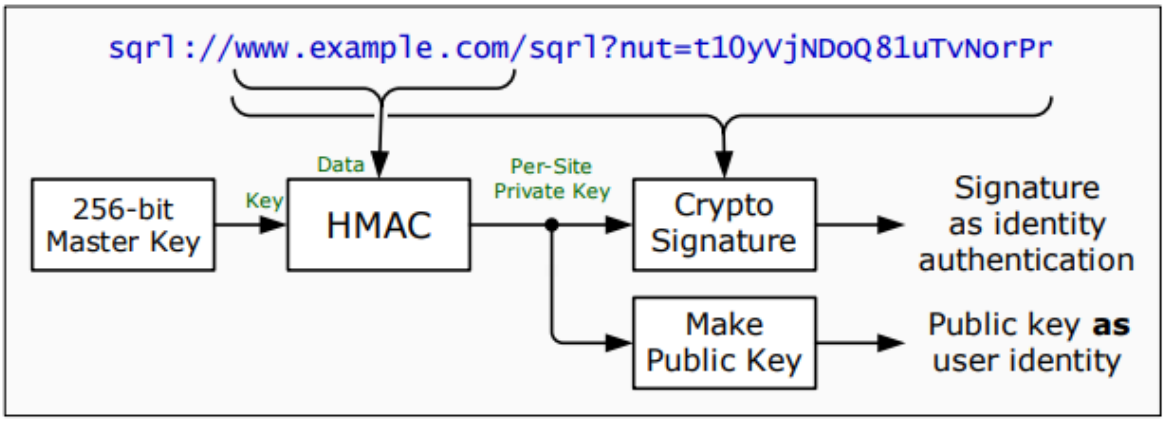

Figura 5 Procesamiento de Generación de Claves SQRL Fuente: (Breeding & Smith, 2013)

Un código *QR* en el sitio contiene la clave maestra (*SQRL code*) lo que es el dominio más la clave privada del sitio, el aplicativo móvil al escanearlo lo traduce como una sentencia que alberga la clave privada, procesa toda la cadena para obtener la paridad con la clave pública del usuario, cuando ambas claves son validadas una alerta pregunta al usuario si desea ingresar al sitio, aceptando el aviso autoriza la autenticación. El proceso se puede visualizar en la figura 6.

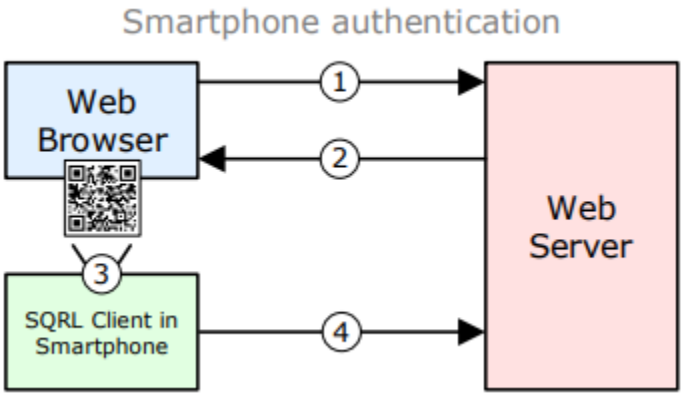

Figura 6 Procedimiento de validación SQRL Fuente: (Breeding & Smith, 2013)

2.6. API RestFul

2.6.1. Mecanismo de Autenticación de Usuario basado en *Tokens* para el intercambio de datos en *RestFul API*

Xiang-Wen, Chin-Yun, Cheng, & YuChin (2015) proponen un nuevo mecanismo llamado *token* desechable, que se basa en la autenticación de *token API RestFul* en el protocolo *HTTP*. Este mecanismo pide a un cliente que almacene un par de llaves pública y privada calculado por el servidor. En cada comunicación, el cliente utiliza la llave pública almacenada, la llave privada y la marca de tiempo actual para producir un *token* desechable, que posteriormente recibe el servidor para su verificación. Con este mecanismo, cada comunicación será válida solo en un período de tiempo fijo, reduciendo así los riesgos de identidad robada.

CAPÍTULO III

DISEÑO DE LA SOLUCIÓN

La autenticación de tres factores es muy similar a la de contraseña basada en tarjetas inteligentes, con la única diferencia que requiere características biométricas como un adicional factor de autenticación (Xiang-Wen et al., 2015).

En este estudio dicho factor adicional se basa en la comparación de la similitud de una imagen generada por un servidor web contra la imagen fotografiada en un dispositivo celular.

3.1. Arquitectura del Sistema

La solución para la problemática planteada es un nuevo sistema para la autenticación en tres factores, para lo que se desarrolló un prototipo que innova con un nuevo tercer factor, reconocimiento de similitud de imágenes. Su objetivo consiste en simplificar y ahorrar el tiempo de validar credenciales en portales *webs*.

De esta forma, se pretende solventar el principal problema que tienen en la actualidad los sistemas con métodos de autenticación tradicional (contraseña), que al ser un método ya muy conocido es fácilmente vulnerable, ya que es cuestión de tiempo y de recursos llegar a vulnerar los métodos de autenticación simple “Contraseñas” (Catoria, 2012). Para ello se ideo agregar un nuevo paradigma a la autentificación en tres pasos.

3.1.1. Definición de Requisitos

En base al análisis del problema descrito en el primer capítulo, se concluyó que el prototipo debe usar un triple factor de autenticación, representados en el siguiente diagrama de casos de uso (figura 7). Dentro de este proceso, con el fin de comprobar la autenticidad del usuario, entre la aplicación cliente y el servidor se intercambia las credenciales por medio de llaves o *tokens*.

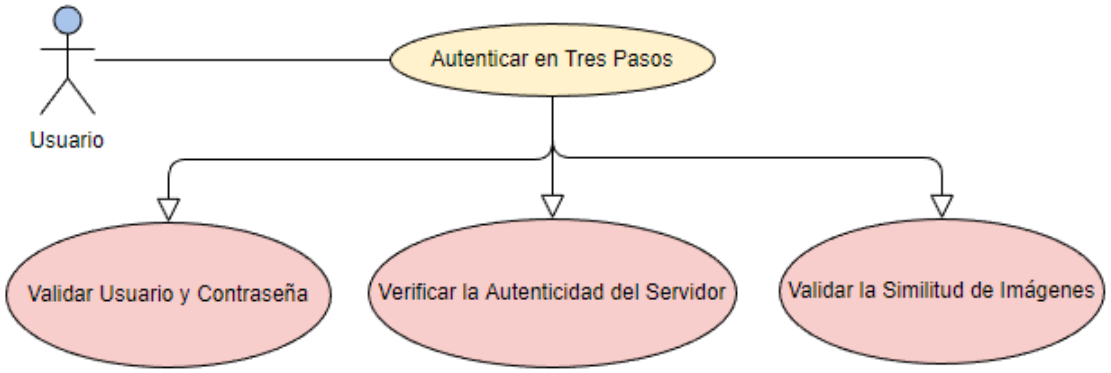

Figura 7 Diagrama de Casos de Uso Autenticación Tres Pasos Fuente: Autor

Como se observa en la figura 7, los requisitos se dividen en tres factores de autenticación como:

1. **Validar Usuario y Contraseña (Factor 1):** Este procedimiento de autenticación tradicional se realiza a través de un usuario y una contraseña, cada usuario posee también un *token* de usuario (llave privada), éste es una cadena única generada para cada usuario.

2. **Verificación del servidor (Factor 2):** Se elaboró un proceso para escanear un código *QR*, dicho proceso valida al servidor que se debe comunicar, cada servidor genera periódicamente un código *qr* que posee el *token* de servidor (llave pública), éste es una cadena única generada para cada servidor de autentificación y se encuentra encriptado dentro del código *QR*, adicional a otros datos encriptados para generar la sesión en el navegador cuando se ratifique la autenticación.

3. **Validación de similitud de imágenes (Factor 3):** Se estableció el método para determinar si la imagen generada por el servidor es semejante a la que fotografió el usuario.

El factor 3 del prototipo demanda la necesidad de un algoritmo de similitud de imágenes, lo que involucró algunos pasos adicionales:

- Seleccionar las mejores imágenes ajustables al método de comparación.
- Registrar el coeficiente de similitud, este coeficiente es el valor entregado por el algoritmo en base a la comparación de la imagen original contra la que se fotografía, la descripción de cómo se obtiene este valor se encuentra en el apartado 3.1.3.
- Calcular el rango de tolerancia de aceptación en los resultados obtenidos en el coeficiente de similitud.
- Realizar pruebas para sustentar la validez de aplicar el rango tolerable a este factor de autenticación.

Para una descripción más profunda de estos últimos pasos, referirse al capítulo 4.

3.1.2. Arquitectura Aplicativo / Servidor

La arquitectura seleccionada fue la de cliente-servidor, modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios (Marini, 2012), de esta forma las tareas son repartidas entre el proveedor de servicio llamado servidor (servidor *web*), y el demandante llamado cliente (aplicativo móvil). La principal razón de selección de este tipo de arquitectura fue para no centralizar las tareas de seguridad en un solo punto, impidiendo que en los ataques de intrusión al servidor obtengan todo el procedimiento de autorización de accesos.

Se optó por seleccionar un aplicativo móvil ya que se ha transformado en una herramienta de trabajo y vida diaria (Ruiz, Sánchez, & Trujillo, 2015), adicional que seccionar el procedimiento de seguridad entre el servidor y el aplicativo móvil garantiza que no se exponga toda la mecánica de solicitud de acceso.

Un servidor del tipo cliente servidor permite ajustar una interfaz de programación de aplicaciones del tipo *API REST* centrada en el manejo de comunicaciones del tipo *HTML*, dicho aplicativo del servidor se complementó como una de tipo *API Restful User Autentication*, lo que involucra el siguiente concepto:

La Arquitectura *Representational State Transfer* (*REST*) es un estilo arquitectónico, que proporciona dirección para construir servicios web distribuidos y acoplados libremente. Usualmente utilizado para desarrollos rápidos, livianos, escalables y fáciles de mantener suele utilizar protocolos *HTTP* de comunicación (Evidian, 2015).

Las ventajas que se obtienen al usar está arquitectura son los siguientes:

- Soporta diferentes formatos (*XML*, *Json*, etc)
- Los mensajes que utiliza son más pequeños en tamaño y ocupan menos ancho de banda.
- En términos de rendimiento es superior tiene mejor soporte de almacenamiento cache.
- En términos de seguridad es más robusto y su versatilidad para el manejo de *APIS*

Por otro lado, se eligió un servidor web para ofrecer un servicio estable, sencillo y accesible de diferentes locaciones, adicional a poder agregar una capa más de seguridad contra posibles ataques.

Es así que esta arquitectura involucra varios procedimientos y funciones en el servidor y el cliente las que se describen en el siguiente diagrama de procesos.

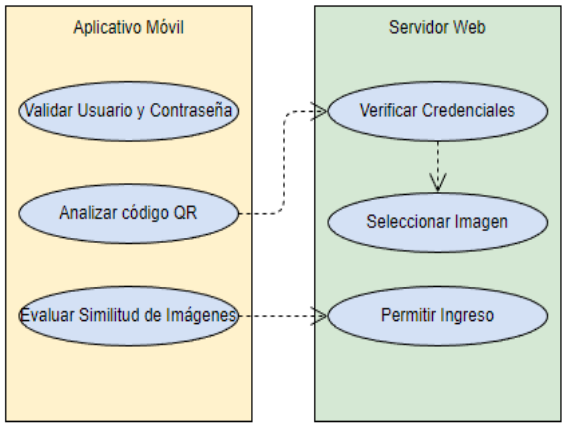

Figura 8 Diagrama de procesos internos Fuente: Autor

En la **¡Error! No se encuentra el origen de la referencia.**, los procedimientos son listados como funciones que realiza cada elemento tecnológico y como se comunican entre ellos al momento de presentarse una autenticación. La secuencia que recorre toda la arquitectura propuesta se puede listar de la siguiente forma:

- Validar Usuario y Contraseña. Proceso interno del aplicativo móvil que comprueba las credenciales del usuario, de presentarse un acceso autorizado se crea la variable que contiene el *token* del usuario (llave privada). Este último es un requerimiento para la verificación de credenciales que realiza el servidor.

- Analizar código *QR*. El escaneo del código dispara la función de decodificar la información encriptada en el *token* servidor (llave publica) la cual apunta al servidor con el que se debe comunicar para enviar el *token* de usuario, esta última llave es única para cada cliente.

- Verificar credenciales. Al recibir el *token* de usuario, se inicia en el servidor la función de comprobación, para ratificar que dicho usuario conste dentro de la base de datos de acceso al servidor.

- Seleccionar imagen. Este procedimiento está encadenado a la verificación de credenciales, se encarga de generar una imagen aleatoria que se despliega en la página web, para dar paso al siguiente factor de autenticación.

- Enviar similitud de imagen. Tercer factor de autenticación, al fotografiar la imagen generada por el servidor el aplicativo evalúa la similitud de imágenes la fotografiada contra la que genero el servidor, si el resultado entra dentro del rango de tolerancia se permite el acceso a la administración del sitio.

- Presentar fallo autenticación. De no presentarse un éxito en el acceso al portal, todo el proceso volverá a la presentación del código *QR* para otro intento de autenticación.

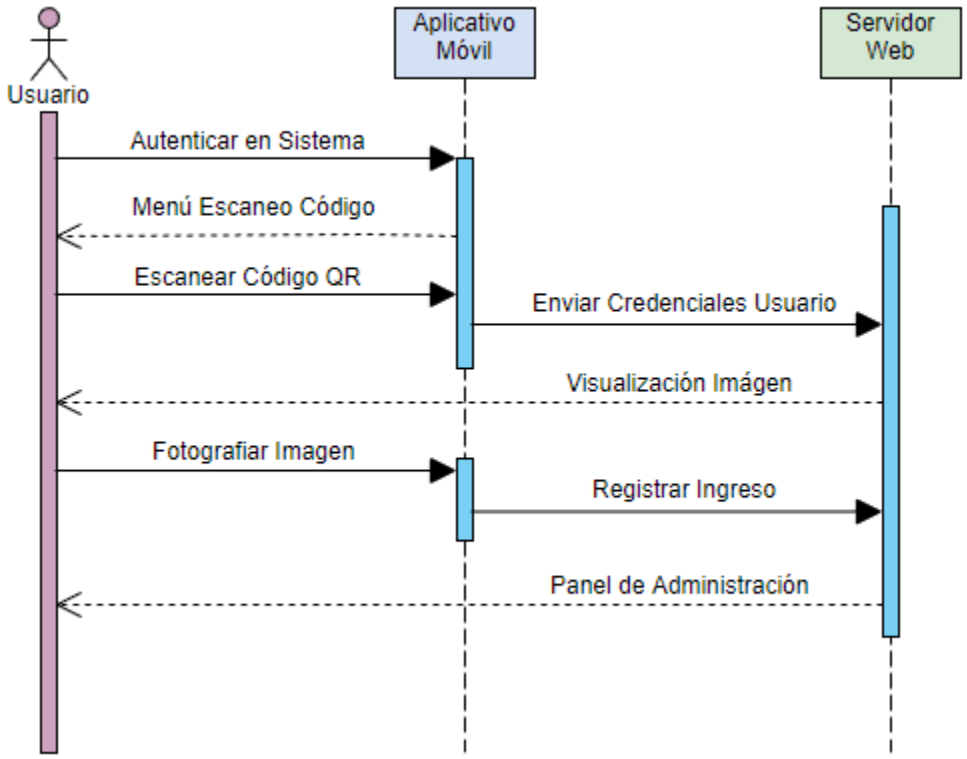

Figura 9 Diagrama de Secuencia del prototipo Fuente: Autor

La **¡Error! No se encuentra el origen de la referencia.** describe la secuencia de pasos existentes entre la entidad usuario y el prototipo de autenticación. A continuación, se describe progresivamente el proceso que inicia con la intención de ingresar al sistema del usuario y continúa con cada factor de autenticación:

1 **Autenticar en Sistema.** Este proceso forma parte del factor 1 (detallado en el apartado 3.1.1) ingresando usuario y contraseña el aplicativo móvil se encarga de validar, si es exitoso la respuesta del aplicativo es el menú para escanear el código *QR*.

2 **Escanear código *QR*:** El usuario escanea el *QR* entregado por el sitio para ratificar el servidor al que debe comunicarse, siendo este el factor 2 de autenticación (detallado en el apartado 3.1.1).

3 **Enviar credenciales de usuario:** Este es un proceso es realizado por el aplicativo móvil, de existir una comunicación exitosa con el servidor se inicia un proceso donde

se analiza el *token* del usuario contra el *token* del servidor, cuya respuesta del servidor es la visualización de la imagen en el portal *web*.

4 **Fotografiar imagen:** El usuario fotografía la imagen que fue desplegada usando el aplicativo móvil, con lo que se genera un proceso de verificación de similitud de imágenes entre la que se fotografió y la que generó el servidor, este proceso utiliza un algoritmo de similitud (detallado en el apartado 3.1.3).

5 **Registrar Ingreso:** Al recibir una confirmación del anterior paso, se registrará el usuario y la fecha con hora de ingreso dentro del servidor, lo que devuelve como respuesta el ingreso al panel de administración.

3.1.3. Algoritmo de Similitud de Imágenes

Como se ha dicho ya, el algoritmo de similitud cumple con realizar varios procesos y tareas para determinar el coeficiente de similitud verificando si dos imágenes tienen igualdad entre sí, mediante el análisis de pixeles a través de las distancias entre escalas de grises.

Para cumplir con este objetivo, este algoritmo sigue la secuencia de pasos:

1) Inicializar variables img1 e img2 mediante el proceso *MimeType*
2) Redimensionar ambas imágenes usando la función ResizeImage
3) Aplicación filtro grises utilizando función *FilterGrayscale*
4) Encontrar la media de color con el procedimiento *ColorMeanValue*
5) Transformar a *bits* las medias de color por medio de función bits
6) Determinar las distancias *hammer* adicionando más uno, si los valores *bits* no son iguales.
7) Retornar variable distancia de *hammer* (valor entero) considerado coeficiente de similitud.

Varios de los procesos detallados anteriormente utilizan la siguiente librería de funciones, las cuales forman parte del núcleo del algoritmo de similitud:

a) Función *MimeType*: Determina si la imagen es jpg, jpeg o png y devuelve su tipo, de lo contrario devolverá false.

b) Función *ResizeImage*: Proceso que iguala las dimensiones, el tamaño que adapta es un cuadrado de 8 * 8.

c) Función *FilterGrayScale*: Convierte el color de un elemento en un tono de gris.

d) Función *ColorMeanValue*: Se encarga de eliminar la media del color de la imagen y almacenarlos en un arreglo.

e) Función *Bits*: Usando el arreglo de media de color se convierte a unidad de bit el arreglo.

El Algoritmo arroja valores enteros que representan la similitud de comparación de imágenes, estos valores se sitúan desde 0 en adelante. Los rangos más aceptables son entre 0 a 10.

Todas estas tareas necesitan de imágenes que sean aplicables al algoritmo, cuestión que es tratada en el capítulo 4.

3.2. Detalle de la Implementación

Este apartado describe todos los elementos involucrados en la construcción del prototipo, lenguajes, dispositivos y comunicaciones que conforman todos los procesos necesarios para implementar el sistema, como se puede apreciar en la figura 10.

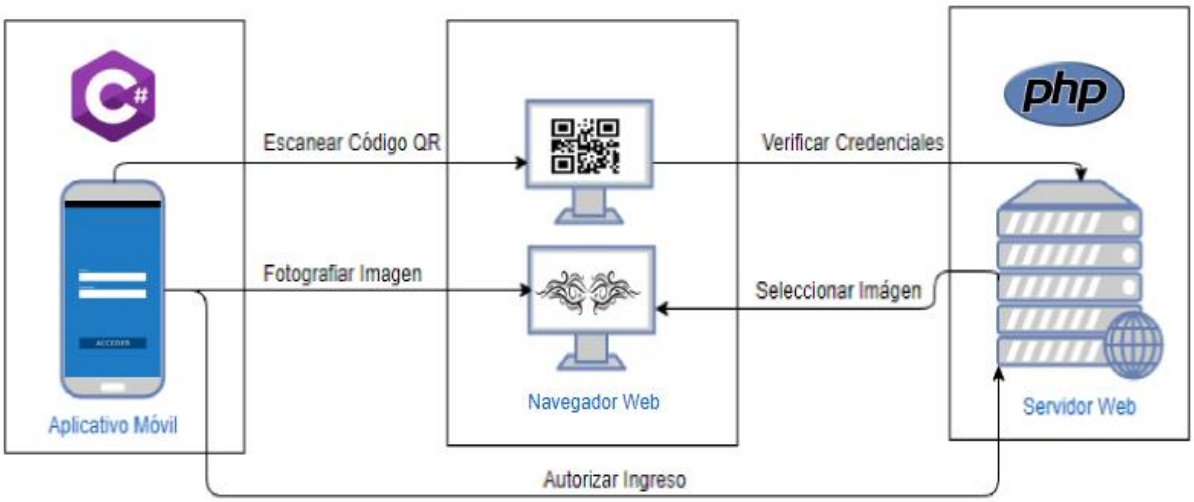

Figura 10 Diagrama de Arquitectura y Tecnologías Fuente: Autor

Para describir lo involucrado en cada elemento de la arquitectura del sistema, continuación se lista los lenguajes y herramientas para el desarrollo del prototipo, abordados por cada componente de la solución, el aplicativo móvil y el servidor:

Aplicativo móvil

- El lenguaje seleccionado fue C# en donde todas sus funciones son escritas en *scripts*.
- *Plugin* para el reconocimiento del código *QR*.
- Librerías para comunicaciones externas con el servidor.
- Proceso de des encriptación de códigos en base 64.
- Algoritmo de similitud de imágenes.

Servidor

- El lenguaje seleccionado fue PHP todos sus procesos delimitados en archivos.
- *Plugin* para creación de códigos *QR*
- Proceso de encriptación de códigos en base 64
- Servidor web apache *tomcat*
- Base de datos MySQL.

3.2.1. Manejo de Interfaces de Usuario

A continuación, se presenta un ejemplo de las pantallas que constituyen la implementación del prototipo. Descritas estas en sus elementos centrales, aplicativo móvil y el servidor web.

Aplicativo Móvil

- Menú de Acceso
- Área de Fotografía

Servidor Web

- Generación de código *QR*
- Evaluación de intento de conexión
- Generación imagen aleatoria
- Acceso al Sistema de Administración

Menú de Acceso

La figura 11 enseña lo que es la pantalla inicial del prototipo donde se procesa la autorización de ingreso al aplicativo móvil mediante un usuario y contraseña, lo cual es el primer filtro de acceso del prototipo, como ya se ha explicado anteriormente.

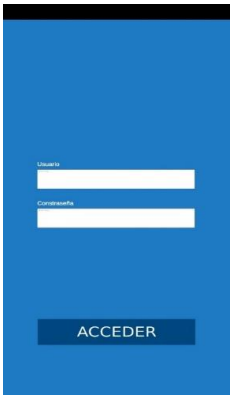

Figura 11 Interfaz de Ingreso Fuente: Autor

Una vez validadas las credenciales del usuario se enciende la cámara para que el usuario escanee un código *QR*. Al validar dicho código el dispositivo se comunica con el servidor web y envía encriptado el *token* del usuario. La figura 12 muestra la vista de este procedimiento de comprobación.

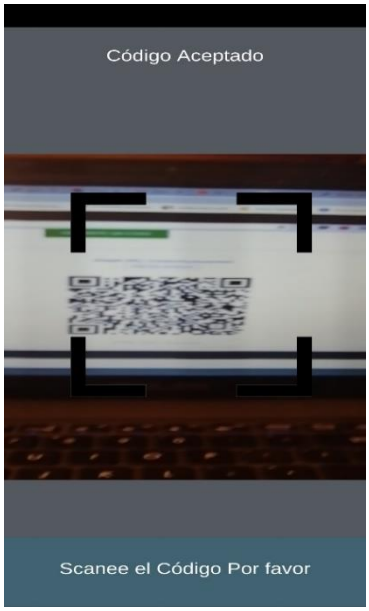

Figura 12 Interfaz de comprobación de código QR Fuente: Autor

Área de Fotografía

Una vez comprobadas las credenciales de conexión, el servidor responde con una imagen que es desplegada en el sitio web, la cual debe ser fotografiada por el usuario para continuar con el siguiente paso de validación. Como se muestra en el ejemplo de la figura 13.

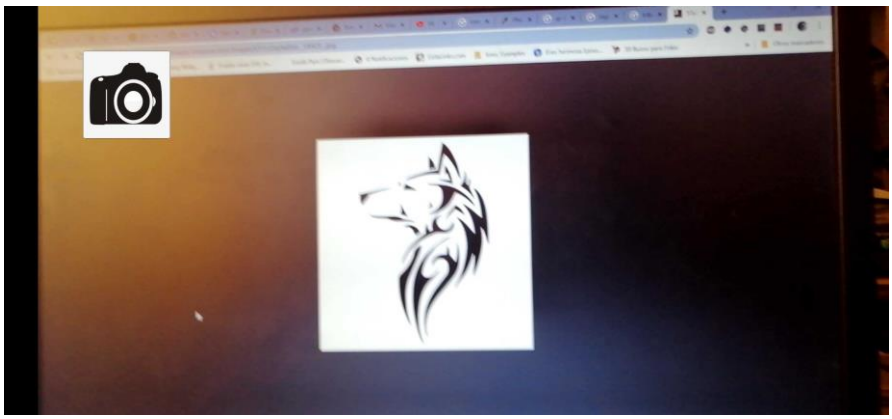

Figura 13 Gráfico de Interfaz de Fotografía Fuente: Autor

En este punto el aplicativo celular procesa la similitud entre la imagen del servidor y la imagen fotografiada, tomando en cuenta que, si el resultado se encuentra entre el rango de tolerancia (calculado en el capítulo 4 apartado 4.2.7), se concluye el factor 3 de autenticidad validar similitud de imágenes. Este proceso es parte fundamental en el estudio, por ser el método nuevo de los tres factores de autenticación.

3.2.2. Servidor Web

Generación código *QR*

El servidor crea continuamente códigos *QR* que mantienen información encriptada y serializada del sitio para quien atente una conexión con el mismo.

Dentro del portal web la sección para realizar accesos mantiene una generación continua de códigos *QR* el segundo paso para el factor de autenticación del prototipo, el código contiene información encriptada, el *token* del sitio (llave pública) necesario para verificar la autenticidad del servidor.

La figura 14 indica un ejemplo de generación de código *QR* en el navegador web.

Figura 14 Código QR generado en el navegador web Fuente: Autor

Evaluación de intento de conexión

Al presentarse una solicitud de conexión validada, se genera nuevamente una comunicación entre el aplicativo del dispositivo móvil y el servidor, donde empieza el proceso de verificación de credenciales, el dato provisto por el aplicativo del celular es el *token* del usuario, este dato es procesado para ratificar si el usuario tiene privilegios de ingresar al sitio.

Generación imagen aleatoria

De existir una aprobación en el proceso anterior el servidor desplegará una imagen aleatoria para que sea fotografiada y evaluada por el aplicativo del dispositivo celular. La figura 15 representa la imagen aleatoria desplegada en el portal *web*.

Figura 15 Imagen aleatoria portal web Fuente: Autor

Acceso al sistema de Administración

Una vez recorrido todos los procesos anteriores finalmente el usuario puede ingresar al portal de administración. La figura 16 representa la imagen de ingreso a la administración del portal *web* al que se deseaba conectar.

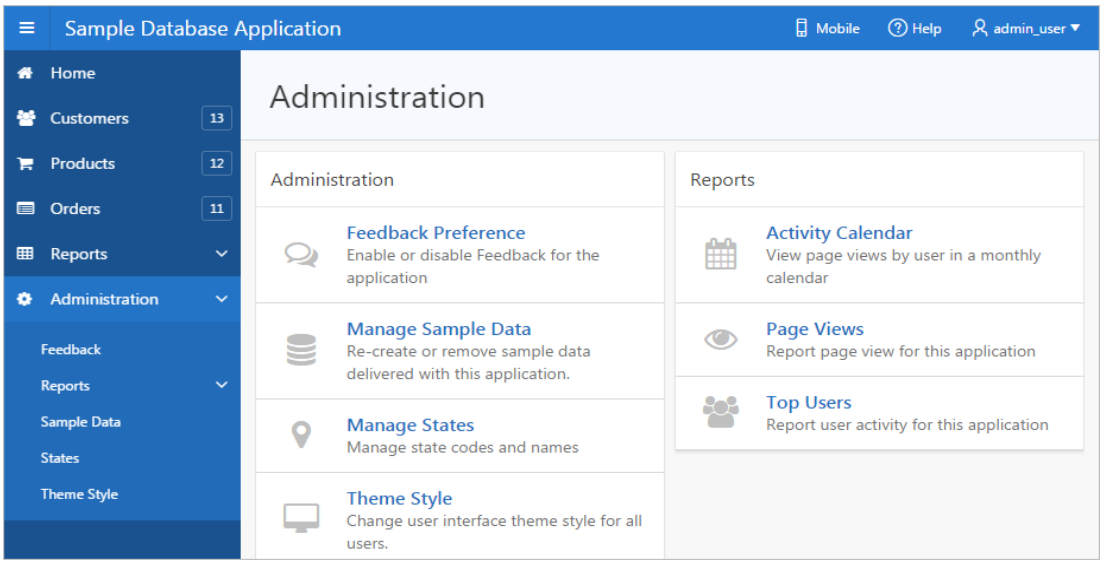

Figura 16 Imagen acceso administrativo del portal web Fuente: Autor

Esta es la pantalla final del proceso de autenticación, demuestra que se produjo una autenticación exitosa por parte del usuario.

CAPÍTULO IV

RESULTADOS

4.1. Método

Una vez definida la arquitectura para solucionar la problemática planteada y al identificar que el último factor de autenticación necesita de imágenes para analizar su similitud, dentro del presente estudio se planteó un análisis para determinar el tipo de imágenes más adecuado. Es así que, con este objetivo se obtuvieron imágenes de tres tipos:

- **Tribales:** Consisten en imágenes del tipo tribal consideradas como imágenes simples con patrones geométricos definidos (Porto & Gardey, 2013), ya que sus formas y colores no son complejas y al no poseer un fondo de imagen ayuda a simplificar el análisis del algoritmo.

- **Animales:** Son imágenes de similares características a las anteriores, sus formas son de animales con tinte tribal. Poseen las mismas ventajas, aunque se observan mejor distribuidas geométricamente.

- **Normales:** Son imágenes de tipo paisaje, representadas por un conjunto de retratos y fondos de pantalla que poseen colores y características más complejas para su análisis.

Una vez identificados estos tipos de imágenes, la evaluación fue realizada siguiendo los siguientes pasos:

I. Se obtuvo una muestra de 100 imágenes por cada tipo de los antes mencionados.

II. Se tomaron fotografías de las 100 imágenes intercalando dos posiciones diferentes de la cámara, 90 (vertical) y 180 (horizontal) grados.

III. Se calculó el coeficiente de similitud de cada imagen utilizando el algoritmo definido en el apartado 3.1.4. Cabe recalcar que este coeficiente es la medida de similitud de las distancias entre bits de las imágenes comparadas.

IV. Por cada valor del coeficiente de similitud se calculó la frecuencia de incidencias de fotografías, mediante la contabilización de las imágenes que obtenían el mismo coeficiente. Cabe recalcar que este valor de similitud representa la efectividad del algoritmo en determinar similitudes.

V. Además, se determinó los rangos de valores del coeficiente que tienen más frecuencias acumuladas. Para esto se observó los resultados más altos de frecuencias acumuladas y comparándolos con los cuartiles superiores e inferiores que señala el diagrama de cajas.

VI. Finalmente, con los datos obtenidos de las frecuencias acumuladas y la ayuda visual de los cuartiles de dispersión en el diagrama de cajas se determinó el rango de tolerancia aplicable al algoritmo de similitud y se realizaron pruebas de validez que corroboren la efectividad de usar este rango para autenticar a un usuario.

4.2. Análisis de tipos de imágenes

Este apartado muestra los resultados obtenidos de la validación de los tipos de imágenes, según la secuencia de pasos descrita anteriormente. Los datos se detallan por cada uno de los tipos de imágenes y por cada inclinación de la cámara, como se describe a continuación:

- Gráfica de resultados imágenes Tribales Normales en 180 Grados

- Gráfica de resultados imágenes Tribales Normales en 90 Grados
- Gráfica de resultados imágenes Tribales Animales en 180 Grados
- Gráfica de resultados imágenes Tribales Normales en 90 Grados
- Gráfica de resultados imágenes Normales en 180 Grados
- Diagramas de Caja Imágenes en 180 y 90 Grados

Para las primeras gráficas se seleccionó diagramas de tipo barras que permitan la comparación entre valores obtenidos del coeficiente de similitud en todos los lotes de imágenes. En el caso de los dos últimos gráficos, se optó por el de cajas con la finalidad de determinar la dispersión de datos en y así poder comparar las frecuencias de las similitudes de las imágenes por cada tipo.

4.2.1. Análisis Tribales 180 Grados

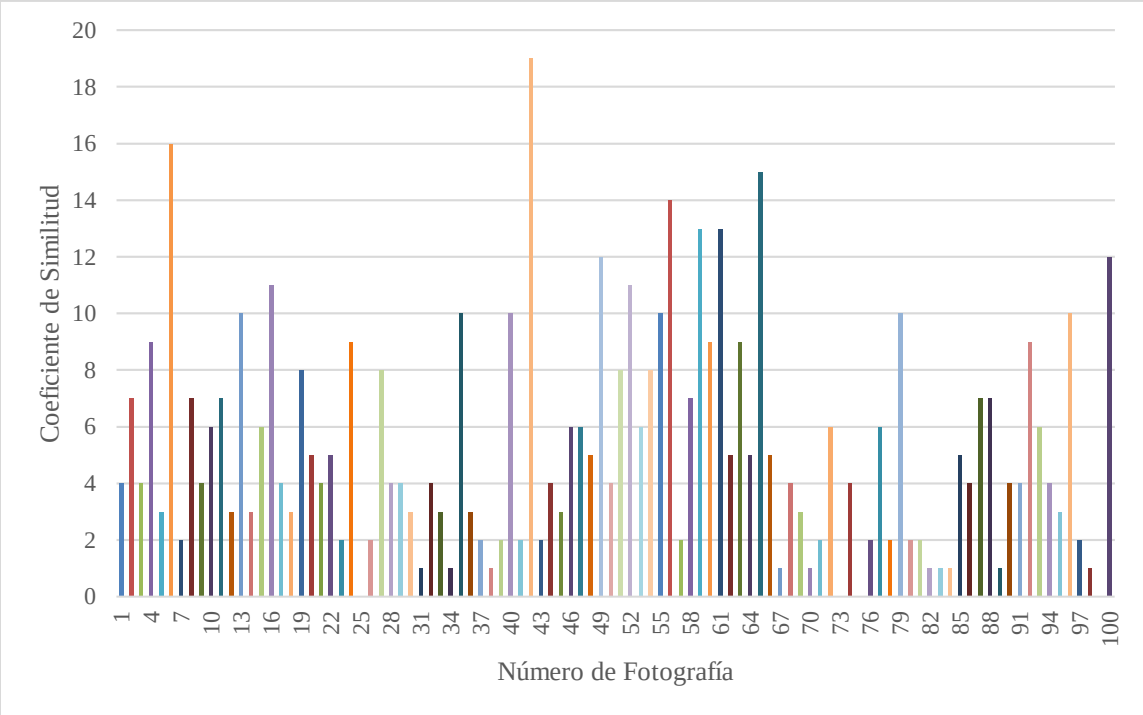

Figura 17 Gráfico de Análisis Imágenes Tribales 180 Grados Fuente: Autor

Como se puede apreciar en la figura 17, el mayor valor alcanzado por el algoritmo es de 19 mientras el menor es de 0.

Coeficiente de Similitud	Frecuencia
0	4
1	11
2	14
3	10
4	16
5	7
6	8
7	5
8	4
9	5
10	6
11	2
12	2
13	2
14	1
15	1
16	1
17	0
18	0
19	1
	100

Tabla 2 Frecuencias del Coeficiente de Similitud Fuente: Autor

Como muestra la tabla 2, la variación de valores es constante, no obstante, la frecuencia más alta de aciertos está entre el rango del 0 al 8 (80 frecuencia acumulada), indica un comportamiento persistente y muy aceptable para lo recomendable según el algoritmo.

A continuación, en el análisis estadístico se puede observar que la varianza 15.42 es el margen de errores que rodea al promedio de 5.30, todo este comportamiento se registra en la tabla 3.

Estadístico	Valor
Valor Inferior	0
Valor Superior	19
Promedio	5,30
Varianza	15,42

Tabla 3 Cálculos Estadísticos de los Resultados Obtenidos Fuente: Autor

4.2.2. Análisis Tribales 90 Grados

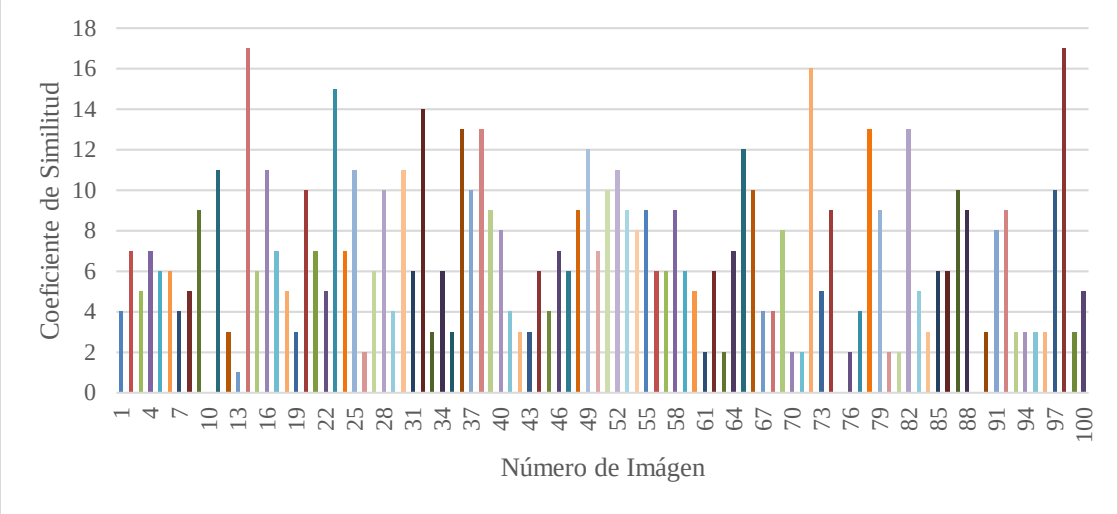

Figura 18 Gráfico de Análisis Imágenes Tribales 90 Grados Fuente: Autor

Como se puede apreciar en la figura 18, el valor mayor arrojado por el algoritmo es de 17 mientras el menor es de 0.

Coeficiente de Similitud	Frecuencia
0	3
1	1
2	8
3	13

4	8
5	8
6	14
7	8
8	4
9	10
10	7
11	5
12	2
13	4
14	1
15	1
16	1
17	2
	100

Tabla 4 Frecuencias del Coeficiente de Similitud Fuente: Autor

Como muestra la tabla 4, existe variación de valores constante y aceptable, su frecuencia más alta de aciertos está en el rango del 0 al 9 (77 frecuencia acumulada), a comparación del análisis a 180 se observa menos frecuencias de aciertos.

A continuación, en el análisis estadístico se puede observar que la varianza 14.90 es el margen de errores que rodea al promedio de 6.7, todo este comportamiento se registra en la tabla 5.

Estadístico	Valor
Valor Inferior	0
Valor Superior	17
Promedio	6,7
Varianza	14,90

Tabla 5 Cálculos Estadísticos de los Resultados Obtenidos Fuente: Autor

4.2.3. Análisis Animales 180 Grados

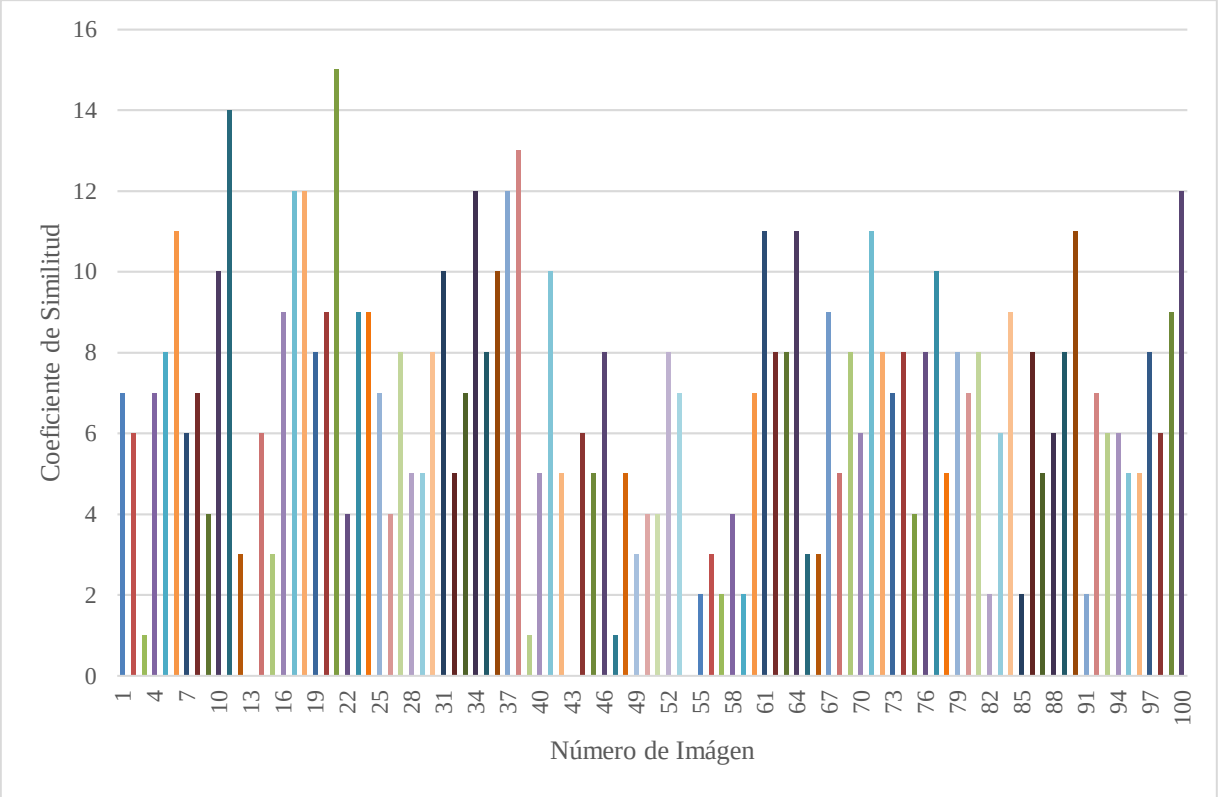

Figura 19 Gráfico de Análisis Imágenes Animales 180 Grados Fuente: Autor

Como se puede apreciar en la figura 19, el valor mayor arrojado por el algoritmo es de 15 mientras el menor es de 0.

Coeficiente de Similitud	Frecuencia
0	3
1	3
2	6
3	8
4	7
5	12
6	10
7	10
8	18

9	5
10	5
11	5
12	5
13	1
14	1
15	1
	100

Tabla 6 Frecuencias del Coeficiente de Similitud Fuente: Autor

Como muestra la tabla 6, la variación de valores es constante, no obstante, la frecuencia más alta de aciertos está entre el rango del 0 al 8 (77 frecuencia acumulada), indica un comportamiento persistente y muy aceptable para lo recomendable según el algoritmo.

A continuación, en el análisis estadístico se puede observar que la varianza 10.65 es el margen de errores que rodea al promedio de 6.7, todo este comportamiento se registra en la tabla 7.

Estadístico	Valor
Valor Inferior	0
Valor Superior	15
Promedio	6,65
Varianza	10,65

Tabla 7 Cálculos Estadísticos de los Resultados Obtenidos Fuente: Autor

4.2.4. Análisis Animales 90 Grados

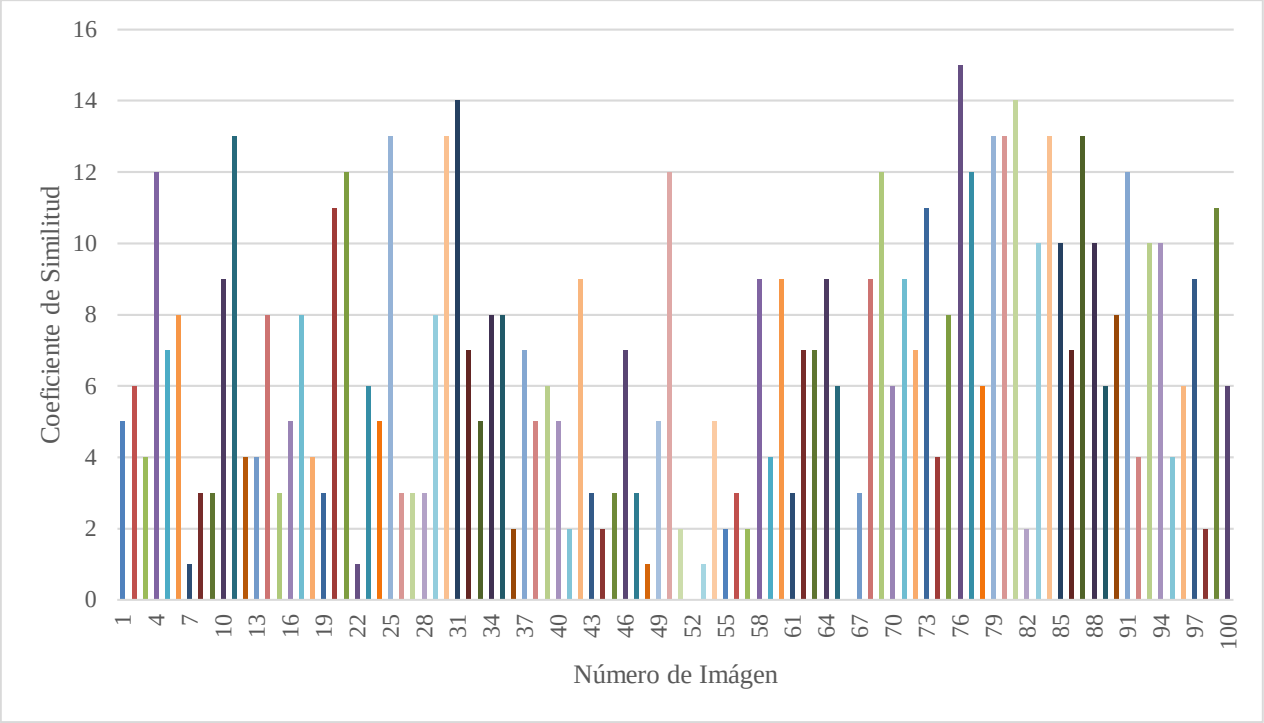

Figura 20 Gráfico de Análisis Imágenes Animales 90 Grados Fuente: Autor

Como se puede apreciar en la figura 20, el valor mayor arrojado por el algoritmo es de 15 mientras el menor es de 0.

Coeficiente de Similitud	Frecuencia
0	2
1	4
2	8
3	13
4	8
5	8
6	9
7	8
8	8
9	8

10	5
11	3
12	6
13	7
14	2
15	1
	100

Tabla 8 Frecuencias del Coeficiente de Similitud Fuente: Autor

Como muestra la tabla 8, existe variación de valores constante y aceptable, su frecuencia más alta de aciertos está en el rango del 0 al 9 (76 frecuencia acumulada), a comparación del análisis a 180 se observa menos frecuencias de aciertos.

A continuación, en el análisis estadístico se puede observar que la varianza 13.20 es el margen de errores que rodea al promedio de 6.7, todo este comportamiento se registra en la Tabla 9.

Estadístico	Valor
Valor Inferior	0
Valor Superior	15
Promedio	6,66
Varianza	13,20

Tabla 9 Cálculos Estadísticos de los Resultados Obtenidos Fuente: Autor

4.2.5. Análisis Normales 180 Grados

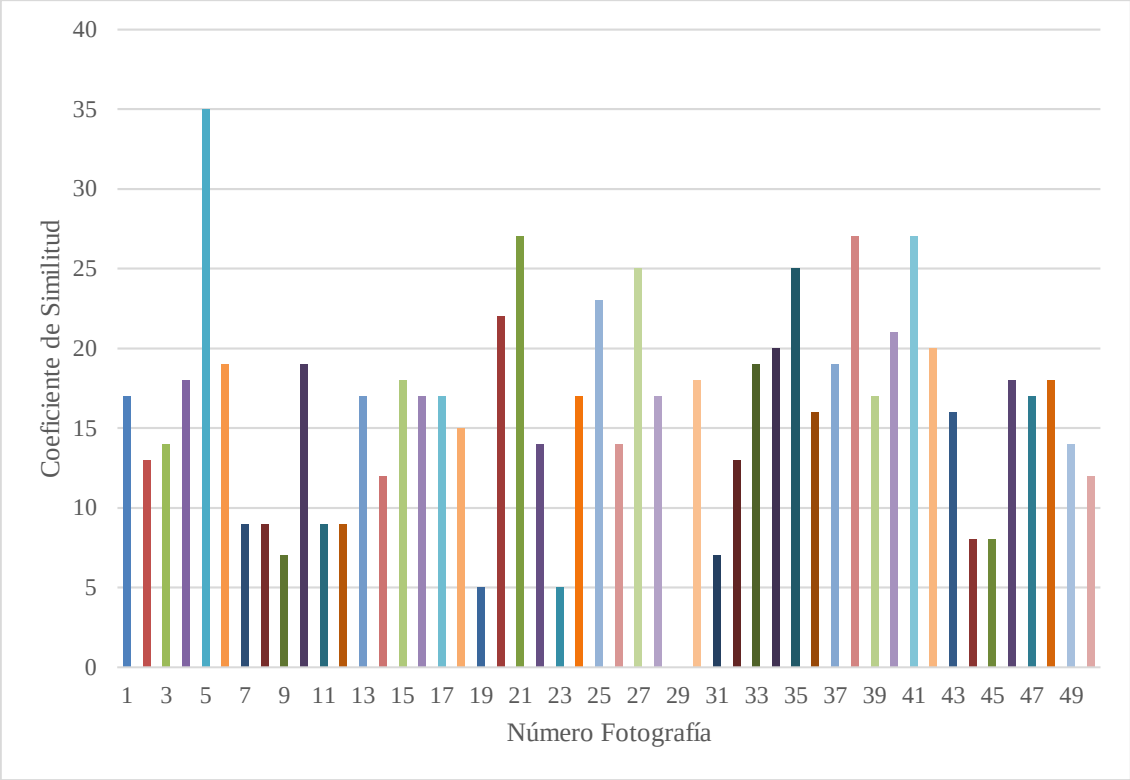

Figura 21 Gráfico de Análisis Imágenes Normales 180 Grados Fuente: Autor

Como se puede apreciar en la figura 21, el valor mayor arrojado por el algoritmo es de 35 mientras el menor es de 0.

Coeficiente de Similitud	Frecuencia
0	1
1	0
2	0
3	0
4	0
5	2
6	0
7	2
8	2

9	4
10	0
11	0
12	2
13	2
14	4
15	1
16	2
17	8
18	5
19	4
20	2
21	1
22	1
23	1
24	0
25	2
26	0
27	3
28	0
29	0
30	0
31	0
32	0
33	0
34	0
35	1
	50

Tabla 10 Frecuencias del Coeficiente de Similitud Fuente: Autor

Como muestra la tabla 10, la variación de valores es variable, existen varios datos dentro del rango que no aparecen, la frecuencia más alta de aciertos está entre el rango del 5 al 19 (39 frecuencia acumulada), indica un comportamiento persistente pero no aceptable para lo recomendable según el algoritmo.

A continuación, en el análisis estadístico se puede observar que la varianza 47.91 es el margen de errores que rodea al promedio de 15.75, la diferencia entre ambos alberga un considerable porcentaje de error que se puede producir, todo este comportamiento se registra en la tabla 11.

Estadístico	Valor
Valor Inferior	0
Valor Superior	35
Promedio	15,75
Varianza	47,91

Tabla 11 Cálculos Estadísticos de los Resultados Obtenidos Fuente: Autor

4.2.6. Diagramas de Caja

Las figuras 22 y 23, describen características más visibles de dispersión y simetría entre los tres tipos de imagen. Esto ratifica que las imágenes de Tribales y las imágenes de Animales distan considerablemente en valores de las imágenes normales, demostrando naturalezas diferentes en los coeficientes de similitud.

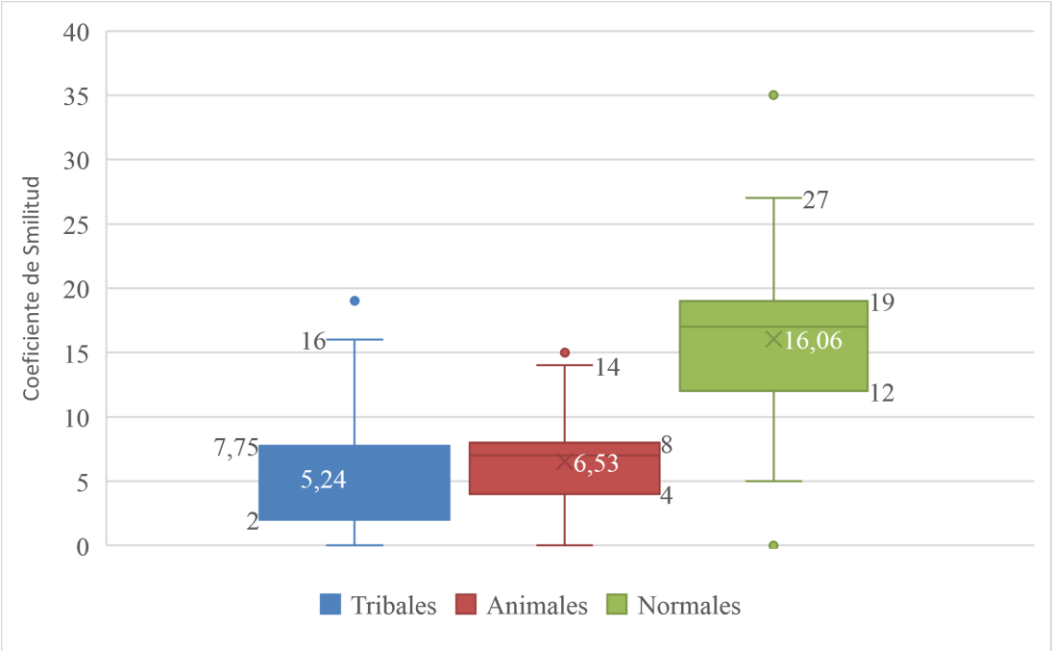

Figura 22 Gráfico de Análisis Diagramas de Caja a 180 Grados Fuente: Autor

Estudiando el diagrama de caja en la figura XX, se puede puntualizar las diferencias que existen entre los análisis de las imágenes. Las medianas de tribales no distan mucho entre sí con 5.24 y 6.53, a diferencia de la mediana de normales que asciende a 16.06. Las mayores frecuencias de similitudes en imágenes tribales puntúan valores del coeficiente entre 8 y 2 mientras que las concentraciones para las imágenes normales se varía con 19 a 12, recordar que estos datos son de imágenes en inclinación de 180 grados.

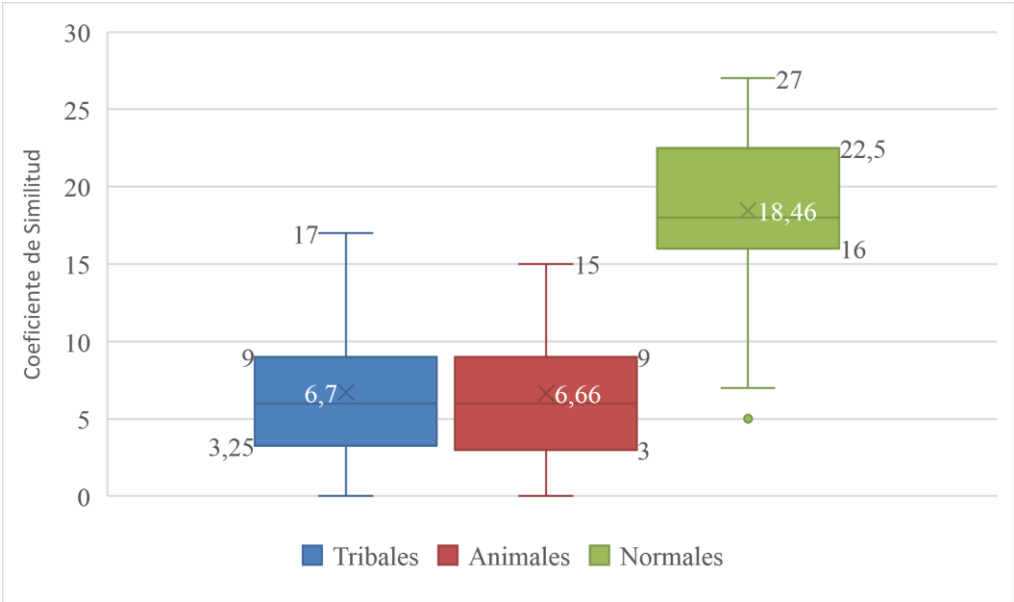

Figura 23 Gráfico de Análisis Diagramas de Caja a 90 Grados Fuente: Autor

De igual manera que en el análisis anterior, se puede observar que las imágenes tribales tienen mayor frecuencia de similitudes en valores del coeficiente entre 9 y 3 mientras que las concentraciones del coeficiente para las imágenes normales se dispara en valores de 22.5 a 16. Los valores de las medianas de tribales no distan mucho entre sí con valores de 6.7 y 6.66 en cambio la mediana de normales sube a un valor de 18.46, considerar que este análisis es para una rotación de 90 grados.

Cabe recalcar que los cuartiles superior e inferior, definidos por los bordes de cada caja en la figura, son los que delimitan la mayor concentración de datos, lo que determina que las imágenes en un ángulo de 180 son las más aptas ya que expresan cuartiles más pequeños y adaptables a los rangos sugeridos por el algoritmo que calcula el coeficiente de similitud.

4.2.7. Validación de rango de tolerancia

Este punto tiene como objetivo determinar el rango tolerable para aplicar al algoritmo de similitud, y así poder adaptar de mejor manera el algoritmo al prototipo de autenticación. Con este objetivo, se agrupó los resultados obtenidos (tabla 12) del análisis de frecuencias y los diagramas de cajas.

	Frecuencias	Diagrama de Cajas
Imágenes	Rangos Aceptables	Cuartiles Superior/Inferior
Tribales 180	0 - 8	2 - 7,75
Tribales 90	0 - 9	3,25 - 9
Animales 180	0 - 8	4 - 8
Animales 90	0 - 9	3 - 9
Normales 180	0 – 19	12 -19

Tabla 12 Rangos y Cuartiles de las Imágenes Fuente: Autor

Como se puede apreciar, se consideró como base de rango de tolerancia el cero ya que los autores del algoritmo (Moyal, 2017) aconsejan utilizarlo, ya que simboliza que la imagen es totalmente igual a la que se evaluó, por lo cual los rangos aceptables van desde este valor (0).

Para delimitar el rango de tolerancia se consideró la mayor frecuencia acumulada de coeficientes de similitud en rangos de 0 a 8, además de los cuartiles superiores con valores aproximados a los de los rangos, lo que determino decidir por un rango de tolerancia entre 0 a 8.

Una vez definido el rango de tolerancia, se aplicó éste en el proceso de autenticación por medio de imágenes, es decir, para que se permita la autenticación al usuario la imagen fotografiada debe obtener un coeficiente de similitud dentro de este rango. Esta prueba se la realizó para fotografías tomadas en 180 debido a que presentan un comportamiento más constante en el análisis de imágenes a diferencia de las fotografías en 90 grados, adicional a factores externos como el enfoque de la cámara que se abordan en el apartado 4.3.

Los resultados para cinco pruebas con los tipos de imágenes tribales y cinco con los de animales, son presentados en la tabla 13, se puede observar que existieron cuatro autenticaciones aceptadas de cinco intentos para cada una de los tipos de imágenes, esto equivale a un 80% de efectividad del rango de tolerancia aplicado al algoritmo de similitud de imágenes. Considerar que la columna de autorización es la que describe si el escenario presento una autorización valida o no.

Rotación	Tipo Imagen	Prueba	Valor Obtenido	Autorización
180	Tribal	1	7	Si
		2	6	Si
		3	3	Si
		4	9	No
		5	4	Si
	Animales	6	4	Si
		7	12	No
		8	2	Si
		9	3	Si
		10	5	Si

Tabla 13 Resultados de las 5 pruebas Fuente: Autor

4.3. Discusión

El seleccionar imágenes en rotación de 180 se debe a varios factores favorables que se encontraron, entre esos el enfocar y cuadrar una imagen en un espacio tipo paisaje es mucho más sencillo que en un ángulo de 90 grados. Esto sucede por el hecho de que la cámara del dispositivo móvil está más cerca del monitor que indica la imagen de validación y facilita el enfocar la captura.

Por otro lado, con respecto a la elección de imágenes tribal y tribal animal sobre las imágenes normales, se fundamentan en el análisis de los diagramas de cajas, donde existe una marcada diferencia en distancias de dispersión de datos entre cajas, mientras que tribales y tribales animales se encuentran en rangos similares.

Otro hecho para no seleccionar las imágenes normales se basa en que sus características reúnen una variedad de coloración de pixeles, lo que dificulta al algoritmo la diferenciación en escalas de grises, contrario a lo que ocurre con los tribales por su forma simple y de dos colores bien marcados.

Sin embargo, existen factores externos que afectaron a este análisis en los tres tipos de imágenes, por el hecho de usar fotografías para el determinar el coeficiente de similitud los brillos y reflejos en el monitor dificultan obtener un resultado favorable. Como es conocido, las entradas de luz o haces de luz afectan a la coloración del pixel fotografiado (Platero, 2009), lo que provoca que el algoritmo determine valores diferentes. También el hecho de fotografiar con un dispositivo móvil al monitor revela las franjas de refrescamiento que al coincidir con la captura se visualiza una línea negra del cambio de *frame* (LedandGo, 2018), lo que afecta radicalmente a la fotografía de manera similar a los reflejos.

Al realizar la validación del rango aplicado en el algoritmo de similitud, en dos de las 10 pruebas se obtuvieron accesos no permitidos, esto se produjo ya que los valores del coeficiente de similitud excedieron al permisible. Analizando ambos fallos se ratificó que las dos imágenes presentaron problemas de enfoque, básicamente un contraste bajo, posiblemente por la falta de suficiente luz que modificó a las fotografías definiendo pixeles de coloración diferente lo que el algoritmo traduce como diferencias notables.

Las pruebas realizadas para evaluar la validez de aplicar el rango de tolerancia en el algoritmo fueron exitosas y reflejaron un 80% de efectividad que fue calculado en base a 10 pruebas de intentos de conexión con imágenes del tipo tribal y animal, 8 de los 10 intentos se presentaron favorables lo que equivale en porcentaje al 80% ya descrito.

CAPÍTULO V

CONCLUSIONES Y TRABAJOS FUTUROS

5.1. Conclusiones

El presente trabajo de investigación es un aporte de programación y lógica de diseño, para el desarrollo de sistemas de seguridad web, el cual posibilita un acceso rápido y seguro a un sistema de seguridad, utilizando un método de autenticación que utiliza similitud de imágenes.

El objetivo principal en este trabajo fue alcanzado, completándose el diseño y desarrollo del sistema de triple autenticación, cabe recalcar que las mayores dificultades se presentaron en la etapa de similitud de imágenes, la cual requirió un análisis previo de las imágenes más adaptables a usarse al momento de ser fotografiadas, fuera de ello no existió mayor complicación.

Grandes ventajas de adaptación de las capas de seguridad seleccionadas no fueron opacadas con la desventaja que se observó en la aplicación del algoritmo de similitud, varios factores no manejables explicados más adelante dificultaron parcialmente el desarrollo, pero los mismos se pudieron ser controlados con soluciones satisfactorias. Adicional que el utilizar factores de seguridad frescos y nuevos aumentan niveles de seguridad ya que son procedimientos que no han sido perpetrados.

En la búsqueda de algoritmos de reconocimiento de similitudes en imágenes se encontró una gran variedad, sin embargo, algunos de ellos no prestaban las características

requeridas para el prototipo, tales como la manipulación de procedimientos en su funcionalidad o no permitir migrar su lenguaje de programación, por lo que varios fueron descartados. El algoritmo seleccionado es un proyecto de código abierto que ofrece la flexibilidad de modificar funciones a lo pretendido.

El análisis de imágenes ayudó a determinar aquellas que son aptas y que presentan menos problemas con el uso de la cámara del dispositivo móvil, aumentando la probabilidad de obtener resultados idóneos. La selección de imágenes tribales y animales se marcó favorable en el análisis, sus colores y formas son simples, lo que facilita al algoritmo de similitud determinar resultados positivos.

El uso de imágenes complejas toma más tiempo de análisis y menos exactitud, esto se produce porque aquellas como las de paisajes poseen variedad de colores, que al ser fotografiados con la cámara del dispositivo, pueden cambiar factores como sobre exposición de luz, la tonalidades de los pixeles, provocando que el algoritmo determine resultados diferentes a los esperados.

Varios factores que no favorecen en la calidad de la captura de una imagen al fotografíar desde un monitor, tales como los reflejos de luz o la aparición de la franja de *frames* de refrescamiento, tienden a provocar una alteración en el análisis de similitud de imágenes. El contraste bajo o falta de enfoque son otros elementos que son provocados por el uso de la cámara del dispositivo móvil. Estas circunstancias desembocan en imágenes fotografiadas con cambios de tonalidades, lo que el algoritmo determina como una desigualdad.

Una vez completado el desarrollo del prototipo según lo previsto y realizadas las pruebas de funcionalidad, se reveló un buen índice de efectividad, aunque debe considerarse que las prestaciones de la cámara del dispositivo móvil también juegan un papel importante, ya que la nitidez de la fotografía es un factor que también ayuda a determinar la similitud de imágenes.

Referente a las modificaciones realizadas al algoritmo de similitud, se determinó que el rango de tolerancia a aplicarse fuera en valores entre 0 a 8, lo que demostró un 80% de

efectividad en 10 pruebas de autenticación, que implica un 20% de margen de error reflejando una pequeña posibilidad de equivocación en la autenticación. No obstante, de presentarse un error, el prototipo permite volver a intentar un acceso.

5.2.Trabajos Futuros

Abordando inicialmente la metodología utilizada en el presente trabajo demuestra una gran solidez, no obstante, la actualización o variación en las capas planteadas es posible, el agregar nuevos elementos de reconocimiento de seguridad o la selección de otros lenguajes de programación utilizados es del todo plausible, es una versatilidad en los sistemas multifactor lo que pueden seguir siendo escalables.

El estudio realizado en este trabajo también brinda pautas para que surjan nuevas iniciativas para el uso de nuevas técnicas de autenticación. De esta forma, varios trabajos pueden surgir como consecuencia del desarrollo de esta tesis:

- Adaptar los dispositivos biométricos que poseen los artefactos móviles para mejorar el tiempo de autenticación del primer factor, que se encarga de validar usuario y contraseña; como reconocimiento facial, de iris o huella digital presente en algunos dispositivos móviles.
- Validar que método de encriptación para *tokens* es el mejor, basado en un análisis de vulnerabilidades usando la ISO 27001, esta ISO determina riesgos de seguridad y estableciendo ocurrencia e impacto al sistema.
- Indagar otros métodos de reconocimiento de similitud de imágenes aplicables al prototipo, así como nuevos procedimientos que relacionan inteligencia artificial o reconocimiento de patrones que pueden resultar ser algoritmos más precisos o que disminuyan el porcentaje de error que actualmente se maneja.
- Levantar un análisis de las vulnerabilidades que se producen al utilizar códigos *QR*, este tipo de códigos presentan algunas desventajas, es pertinente realizar un estudio que determine y presente las mejores formas de mitigar estos inconvenientes.

- Adicionar o actualizar nuevas capas de seguridad dentro del prototipo, puesto que la versatilidad de haber usado MFA posibilita el agregar más componentes de seguridad transparentes para el usuario, pero que aumentará los niveles de seguridad.
- Realizar análisis comparativos del uso de códigos *QR* contra algún otro método de lectura, posiblemente el utilizar otro tipo de código muestre variación de resultados, una compilación de otros procedimientos que establezcan ventajas y desventajas de aplicarlos.

BIBLIOGRAFÍA

Alvarez, L. (2015). *Metodología para la Gestión de la Seguridad Informática*. Cuba.

Andalucía, C. (2016). *Informe de divulgaciónUso de autenticación multi-factor en sistemas y aplicaciones II*. Andalucía.

Arencibia, A. (2009). *Sistema de autenticación, autorización y auditoría (AAA) para aplicaciones basadas en servicios Web XML Authentication, Authorization and Accounting (AAA) system to applications based on XML Web services*.

Breeding, P., & Smith, W. (2013). *Welcome to SQRL*. *248*(5), 70–75.

CA Technologies. (2015). *Libro de estrategias para la administración de API Entender las soluciones para*.

Calvillo, F. (2016). *Tecnicas de Seguridad Informatica*. Revista GTI, 3(7), 79-86

Catoria, F. (2012). Aplicación de técnicas de fuerza bruta con diccionario de datos, para vulnerar servicios con métodos de autenticación simple "Contraseñas", pruebas de concepto con software libre y su remediación *Retrieved from WeLiveSecurity*.

Cavage, M., Behm, B., & Cabrera, L. (2014). *Multifactor Authentication for Programmatic Interfaces*. https://patentimages.storage.googleapis.com/5d/1a/98/bc9b21eef82a8/US8776190.pdf

Chiriguayo, S. J. (2015). *Comercio Electrónico: Importancia de la Seguridad en las Transacciones Electrónicas, Amenazas y Soluciones a Implementar*.

Developers, G. (2010). Google Authenticator. Retrieved from https://github.com/google/google-authenticator/wiki

Dire. (2018). *Manual de API de Consumo para integradores*. Madrid.

Entertainment, B. (2010). Acerca de Blizzard Entertainment. Retrieved from https://www.blizzard.com/es-es/company/about/

Evidian. (2015). *Los 7 métodos de Autentificación más utilizados*. 25. Retrieved from https://www.evidian.com/pdf/wp-strongauth-es.pdf

FIPS-198-1, N. (2008). The Keyed-Hash Message Authentication Code. *Federal Information Processing Standard Publication*, *198*(July), 1–20. Retrieved from http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

Kon, M. (2019). *Implementación de la Autentificación Multifactorial*. RECAI Revista de Estudios en Contaduría, Administración e Informática, 47-71.

Landre, J., & Truchetet, F. (2007). Image retrieval with binary hamming distance. *VISAPP 2007 - 2nd International Conference on Computer Vision Theory and Applications, Proceedings*, *IU*(MTSV/-), 237–240.

LogmeIn, S. (2018). Use the Microsoft Authenticator. Retrieved from

https://support.logmeininc.com/lastpass/help/microsoft-authenticator-lp030022

Mariel, B., Abendaño, M., & Zulaica, J. (2004). Inplementacion de un reconocedor de voz gratuito a el sistema de ayuda a invidentes Dos-Vox en español. *Universidad de Las Americas Puebla*, *Capitulo 5*, 1.

Marini, E. (2012). *El modelo cliente/servidor*. Madrid: linuxito.

Mateos, J. S. P. M. T. (2005). Tecnologías biométricas aplicadas a la seguridad. *Ra-Ma Editorial*, 456. Retrieved from http://www.ra-ma.es/libros/TECNOLOGIAS-BIOMETRICAS-APLICADAS-A-LA-SEGURIDAD/143/978-84-7897-636-2

McDaniel, P. (2006). *Authentication*. Pennsylvania. U.S. Patent No. 6,424,981. Washington, DC: U.S. Patent and Trademark Office.

Muhammad, A., & Tripathi, N. (2012). Evaluation of OpenID-Based Double-Factor Authentication for Preventing Session Hijacking. *JOURNAL OF COMPUTERS*, p. 235.

Nath, Asoke; Mondal, T. (2016). *Issues and Challenges in Two Factor Authentication Algorithms*. 10.

NetIQ. (2016). *Multifactor Authentication Removing Risk While Simplifying Processes*.

Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., & Koucheryavy, Y. (2018). Multi-Factor Authentication: A Survey. *Cryptography*, *2*(1), 1. https://doi.org/10.3390/cryptography2010001

OWASP, F. (2017). *Los diez riesgos más críticos en Aplicaciones Web*. Los Angeles.

Perales, R. (2011). *Sistemas de Autenticación*. Retrieved from zonalibre.org

Porto, J., & Gardey, A. (2013). Definición D.E. Retrieved from https://definicion.de/tribal/

Proskura, A. (2017). *Las nuevas funcionalidades hacen que la MFA y el cifrado sean accesibles para las pymes*. Praha.

Puerta, J. M. (2015). *Desarrollo de una API para la descripción y gestión de Servicios Web REST*. Madrid.

RFC-6238. (2011). *TOTP: Time-Based One-Time Password Algorithm*. Retrieved from https://tools.ietf.org/html/rfc6238

Rivero, T. S. U. J., & Merida, F. (2006). *Introducción a La Seguridad Informática*. 1–12.

Rodríguez López, M. L. I. de L. R. (2016). Metodología para describir servicios RESTful consumidos automáticamente por clientes máquina. *Metodología Para Describir Servicios RESTful Consumidos Automáticamente Por Clientes Máquina Traba*, 48.

Rouse, M. (2014). *Techtarget*. Journal of Technology Research, 5, 1.

Rouse, M. (2015). Google 2-Step Verification. Retrieved from https://support.google.com

Ruiz, J., Sánchez, J., & Trujillo, J. (2015). Utilización de Internet y dependencia a teléfonos. *Latinoamericana de Ciencias Sociales y Niñez*.

Sánchez, J. (2011). *Sistemas de Autenticación y Autorización en Internet*. Lleida.

Saturnino, M. (2017). *Falta de talento en ciberseguridad: la nueva lucha de las empresas*.

Schultz, P. (2012). *Multifactor Multimedia Biometric Authentication*. Retrieved from https://patentimages.storage.googleapis.com/0d/51/1d/d872e9ffba224/US8189878.pdf

SENATICs. (2015). *Autenticación Doble Factor*. Asunción.

Simons, A. (2016). Microsoft Authenticator. Retrieved from https://techcommunity.microsoft.com/t5/Azure-Active-Directory-Identity/Microsoft-Authenticator-8211-Coming-August-15th-Supports-AzureAD/ba-p/245008

Singh, M., Rasansky, R., & Racho, J. (2007). *Multifactor Authentication System*.

Tarazona, C. (2012). *Amenazas Informáticas y Seguridad de la Información*.

Velasco, R. (2018). Doble autenticación para proteger tu cuenta de Facebook. Retrieved from https://www.redeszone.net/2018/05/24/activar-nueva-doble-autenticacion-facebook/

Vieites, Á. G. (2011). *Enciclopedia de la Seguridad Informática. 2ª edición*. Grupo Editorial RA-MA.

Vivek, M. (2017). *Compare Two Images for Similarity*. Retrieved from https://www.phpclasses.org/blog/post/584-How-Can-PHP-Compare-Two-Images-for-Similarity.html

Wu, M., & Liu, B. (2018). Digital watermarking for image authentication. *Signals and Communication Technology*, 33–37. https://doi.org/10.1007/978-3-319-78942-2_4

Xiang-Wen, H., Chin-Yun, H., Cheng, H. W., & YuChin, C. (2015). *A Token-Based User Authentication Mechanism for Data Exchange in RESTful API*.