UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMATICA

TESIS DE GRADO

“MÉTODO DE AUTENTICACIÓN BASADO EN LA DINÁMICA DE TECLEO”

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMATICA
MENCIÓN: INGENIERIA DE SISTEMAS INFORMÁTICOS

POSTULANTE: SIDNEY EYLEEN UGARTE TEJERINA
TUTOR METODOLOGICO: LIC. FREDDY MIGUEL TOLEDO PAZ
ASESOR: Ph. D. JOSE MARIA TAPIA BALTASAR

LA PAZ – BOLIVIA

2018

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

DEDICATORIA

Con todo mi cariño y mi amor para las personas que hicieron todo en la vida para que yo pudiera lograr mis sueños, siempre estuvieron ahí para motivarme y confiando en mi día tras día , agradecerles todo el sacrificio que hicieron por mí.

Mi padre Alberto Ugarte y mi madre Helen Tejerina.

A mis hermanos

Con mucho cariño a todos Cinthia, Israel, Rosario, Luz y Diego que me supieron dar el apoyo, cada uno a su manera brindándome su alegría y buen humor asiendo que mis caídas sean menos duras.

Los quiero mucho….

Para ti con mucho amor y cariño que fuiste mi modelo a seguir mi maestro y guía en este trabajo que por ti lo pude terminar gracias por impulsarme a que pueda llegar a mi meta.

A mi amor Jesus Callisaya

AGRADECIMIENTO

Al Ph. D. Jose Maria Tapia Baltazar, por su apoyo, comprensión, paciencia y dedicación incondicional, por sus sugerencias, observaciones, seguimiento constante, conocimientos y experiencias brindadas en el transcurso del presente trabajo y por todo el tiempo dedicado en cada una de sus revisiones

Al Lic. Freddy Miguel Toledo Paz quién me dedico su tiempo, paciencia y comprensión al realizar el seguimiento de esta Tesis, orientándome a través de su conocimiento y sus sugerencias para la culminación del trabajo.

Un Trabajo de Tesis de Grado, constituye un gran esfuerzo personal para alcanzar los objetivos propuestos, dicho trabajo no sería posible de realizar sin la colaboración, apoyo, orientación, consejos, sugerencias, conocimientos y experiencias de mis docentes y amigos.

También agradezco a todo el plantel Docente que durante todo este tiempo instruyó y capacito, transmitiéndome sus conocimientos y experiencias, de igual forma al plantel Administrativo, Jefatura de Carrera, Kárdex, Laboratorio (Lasin) de Informática y Biblioteca por la documentación y apoyo brindado.

A la Universidad Mayor de San Andres, por brindarme la mejor educación en estos años de mi carrera.

RESUMEN

El análisis de la dinámica del tecleo es una técnica usada para verificar la identidad de usuarios de sistemas informáticos. La presente investigación trata sobre la dinámica de tecleo y sus aplicaciones en nuestro medio. La biometría informática, actualmente, es una de las ciencias más importantes. Últimamente existen aplicaciones y estudios de investigación, pero sin embargo todavía hay mucho por investigar por cuanto existen muchas aplicaciones relacionadas, una de ellas es la seguridad. En los últimos años la demanda de los sistemas biométricos se ha incrementado debido a que la reputación y seguridad en las organizaciones de la sociedad del conocimiento se hace cada vez más vulnerable ocasionando pérdidas económicas cuantiosas, así como en otros aspectos del ser humano.

Por tanto en esta investigación se ha creado un método, el cual está compuesto por etapas, las cuales describen como realizar el reconocimiento de patrones biométricos, basados en la dinámica de tecleo de los usuarios, que se autentifiquen en una aplicación la cual fue implementada para demostrar el uso del método propuesto. Es importante destacar que la finalidad del método es generar información única a partir del comportamiento del usuario. Esta información es lo suficientemente diferente para cada uno, de tal manera que lo distinga de los demás usuarios.

ABSTRACT

The analysis of the keystroke is a technique used to verify the identity of users of computer systems. This research deals with the keystroke dynamics and its applications in our environment. Computing biometrics currently is one of the most important sciences. Lately there are applications and research studies, yet there is still much research because there are many related applications, one of which is safety. In recent years the demand for biometric systems has increased because the reputation and security organizations the knowledge society becomes increasingly vulnerable causing economic losses, as well as other aspects of the human being Hal. Therefore this research has created a method which this compound by stages, which describe how to perform biometric pattern recognition, based on keystroke dynamics of users who authenticate to an application which was implemented to demonstrate the use of proposed method.

Importantly, the purpose of the method is to generate unique information from the behavior user. This information is different enough for everyone, in such a way that distinguishes it from other users.

INDICE

INDICE DE FIGURAS

INDICE DE FORMULAS

INDICE DE TABLAS

CAPITULO I

1.1 Introducción

En la actualidad el uso de los sistemas informáticos es cada vez mayor, debido al fácil acceso a la información a través de ellos. Tareas comunes como el control del personal en una empresa, la vigilancia en negocios, la contabilidad, entre otros, son automatizadas en sistemas que se utilizan únicamente por personas autorizadas, en el mejor de los casos.

Es común que se utilice un mecanismo de identificación, como un nombre de sesión y una contraseña, para acceder a estos sistemas.

Uno de los problemas que ocasiona este tipo de identificación, es el descuido por parte de los usuarios, quienes muestran u ofrecen su información confidencial (usuario y la contraseña) sin saber que mantenerla en secreto es toda la seguridad con la que cuenta el sistema.

Así, cualquier persona que conozca esa información puede acceder al sistema, usurpando la identidad de un usuario.

Afortunadamente, existen mecanismos robustos para la identificación de personas como la autenticación y la biométrica informática que se pueden definir, de la siguiente manera:

La autenticación es un servicio de seguridad relacionado con la identificación de entidades o de datos, es decir, una entidad tiene que comprobar que es quien dice ser o se debe corroborar la validez de los datos, lo que implica la integridad de los mismos.

La biometría informática es un conjunto de técnicas utilizadas para la identificación de personas basadas en uno o más aspectos físicos, así como en aspectos conductuales, es decir, se miden características de las personas de manera directa o indirecta, aplicando técnicas de clasificación, selección, agrupamiento o detección de casos anómalos, ect.

En la mayoría de los casos, la autenticación y la biometría han sido aplicadas de forma independiente. Ejemplo de ello son los sistemas de voto electrónico, donde se han propuesto protocolos seguros de votación utilizando firmas digitales o sistemas biométricos como la huella digital para la identificación de los votantes.

No obstante, si ambas técnicas fueran utilizadas tanto para autenticar como para generar información única de los usuarios, es posible mejorar la seguridad de los sistemas, con una probabilidad insignificante de vulnerabilidad en ataques de usurpación de la identidad. Por tal

motivo en el presente trabajo se realizará la generación de información del usuario a través de la dinámica de tecleo, para generar información por cada usuario autenticado y evitar usurpación de contraseñas.

El método propuesto será realizado para aplicaciones móviles para mejorar la seguridad de autentificación.

1.2. Estado del Arte

Generación de información única del usuario basada en biometría dinámica de tecleo implementado en Android.

Es importante destacar que la finalidad de la técnica es generar información única a partir del comportamiento del usuario. Esta información debe de ser lo suficientemente diferente para cada uno, de tal manera que lo distinga de los demás usuarios, pero que no necesariamente lo identifiqué, ya que lo esencial en la técnica propuesta es producir información única. Esto es diferente a lo propuesto en otros trabajos o sistemas del estado del arte. (Reynoso Coronel 2014) cabe mencionar para generar información única generar un numero semilla para cada usuario.

La biometría es la ciencia que se dedica a la identificación de individuos a partir de una característica anatómica o un rasgo de su comportamiento.

El siguiente trabajo contiene una descripción general de algunas técnicas biométricas así como el desarrollo de una aplicación de control de acceso de empleados mediante la huella digital. El presente trabajo consta de cuatro capítulos pero el ultimo capitulo se enfoca en el desarrollo de una aplicación de control de acceso de empleados, utilizando como dispositivo biométrico un lector de huella digital. Se describe la aplicación, se realiza el análisis y diseño de la misma 20 incluyendo descripción de las estructuras, descripción de algoritmos y diagramas de flujo de los procedimientos utilizados en la aplicación, y se finaliza con la descripción del funcionamiento del sistema. (Castillo Marroquín 2005).

1.3 Planteamiento del Problema

- La usurpación de cuentas es utilizada con el fin de perjudicar a una persona, es decir, difamarla o manchar su nombre con diversos fines que el usurpador busque (secuestro).

- Los casos más comunes son robo de identidad en páginas sociales también en páginas de tiendas online donde las personas son afectadas monetariamente estos casos de usurpación también se debe a que los usuarios hacen uso de sus datos personal como contraseñas asiendo más fácil el acceso a sus cuentas.
- Los usuarios deben tomar en cuenta que para crear una contraseña también podrían intercalar las letras poniendo mayúsculas con minúsculas y símbolos entre sí para que sea más seguro y mejor la autenticación.

También es sabido que hoy en día las personas se están acostumbrando a rebelar las contraseñas de sus teléfonos móviles más que todo se ve ese caso en las parejas ya que rebelando las contraseñas uno puede ver sus cuentas de páginas sociales y revisarlas sin ningún problema.

Los usuarios suelen ser distraídos ya que algunos escriben sus contraseñas en un papel para poder recordar sus contraseñas y usuarios esto en una empresa llevara a problemas como pérdida de imagen de la compañía ya que el usurpador puede manipular a su disposición toda la información que pueda obtener gracias al descuido de un empleado de la empresa.

1.3.1 Problema General

¿Genera la dinámica de tecleo, patrones únicos por cada usuario autenticado?

1.3.2 Problemas Específicos

Para el presente trabajo se estableció los siguientes 3 problemas específicos

Problema Específico 1

Los problemas de seguridad informática han dejado de pertenecer solo a los equipos de cómputo y han comenzado a reflejarse cada vez más en los teléfonos móviles .los ataques están enfocados a robar información personal o corporativa.

Problema Específico 2

El uso de letras especiales ejemplo (ñ, Ñ) para elaborar la contraseña.

Problema Específico 3

El problema también es la generación de patrones de autenticación usando la dinámica de tecleo.

1.4 Hipótesis

La dinámica de tecleo genera patrones únicos para verificar la autentificación de usuarios.

1.5 Objetivos

1.5.1 Objetivo General

El presente trabajo tiene como objetivo general:

"Crear un método de autentificación basada en dinámica de tecleo, que contribuya a la verificación de la autentificación del usuario, aplicando los patrones de comportamiento por cada usuario".

1.5.2 Objetivos Específicos

Para el presente trabajo se establecieron los siguientes 3 objetivos específicos detallados a continuación.

El primer objetivo específico es:

Objetivo Específico 1

Elaborar algoritmo multiplataforma de reconocimiento de dinámica de tecleo para aquellos usuarios que utilicen el sistema de autentificación, los cuales generan patrones de comportamiento únicos para que posteriormente sean almacenados en una base de datos para realizar el análisis de muestras.

El segundo objetivo específico es:

Objetivo Específico 2

Analizar los datos del usuario con el método de autenticación planteado en este trabajo y así obtener muestras de patrones de cada usuario autenticado.

El tercer objetivo específico es:

Objetivo Específico 3

Delimitar el rango de palabras promedio por cada usuario registrado en el sistema para que el algoritmo de análisis sea eficiente en el tiempo de respuesta.

1.6 Justificación

1.6.1 Justificación Social

Hoy en día la mayoría de las personas cuentan con un teléfono celular inteligente, debido a que la tecnología está avanzando día a día las personas realizan todo tipo de actividades a través de sus teléfonos móviles como compras en línea, transferencias bancarias revisan sus correos, etc.
Por tal motivo el método de autenticación basado en la dinámica de tecleo ayudara a mejorar la seguridad y evitar suplantación del usuario.

1.6.2 Justificación Técnica

A pesar de que los sistemas de identificación o autentificación biométricos son sistemas que ofrecen gran seguridad en todas sus aplicaciones.

Hoy en día, nuestras vidas se encuentran rodeadas de sistemas electrónicos cuyo acceso se protege habitualmente con métodos de autenticación simples: uso de contraseñas, pin o patrones. Estos mecanismos tradicionales de autenticación son en general vulnerables a ataques de fuerza bruta y diccionario que han forzado el desarrollo de métodos más robustos.
Utilizando la biometría dinámica con el método propuesto, la autenticación del usuario no se limitara exclusivamente al momento puntual de la solicitud de acceso o login, sino que el comportamiento reflejado durante la sesión será contrastado continuamente con los patrones esperados y almacenados para detectar cualquier discrepancia que delate una suplantación del usuario legítimo.

1.6.3 Justificación Económica

Para la implementación del prototipo se utilizó phonegap y cordova que es software libre por tal motivo el costo de la investigación es cero.

1.7 Límites y Alcances

1.7.1 Delimitación Temática

El área de investigación está basada en la biometría dinámica, se desarrollara un método de aprendizaje el cual reconozca patrones de dinámica de tecleo para la autentificación de usuario. La biometría dinámica está relacionada con la Seguridad de Tecnologías de Información.

1.7.2 Delimitación Espacial

El método propuesto puede ser aplicado a cualquier tipo de sistema que tenga un módulo de autenticación, es decir usuario y contraseña por tal motivo este método será para sistemas en internet, que tenga plataformas web y aplicaciones móviles

1.7.3 Delimitación Temporal

La seguridad de autenticación para los móviles será puesta en práctica a partir de la implementación del método de autenticación basada en la dinámica de tecleo y el mismo se realizara en el año 2018.

1.8 Metodología de investigación

Una de las etapas para aplicar la dinámica de tecleo, es aplicar un algoritmo de aprendizaje para verificar el nivel de confianza de la muestra, entre los algoritmos que ayudan a realizar esta tarea se puede clasificar en algoritmos: Geométricos, Redes Neuronales, Lógica Difusa y Algoritmos Evolutivos. Vincenzi (2014). Por tal motivo para el presente trabajo se plantea un nuevo algoritmo basado en el paradigma positivista y en el cual se utilizará el método científico cuantitativa que se apoya en las técnicas estadísticas para el proceso de la información y generar patrones de comportamiento lo cual proporcionara un umbral de reconocimiento aceptable en cuanto a tiempo de respuesta por el usuario.

"Se debe considerar también que existen problemas técnicos o de estado como: Cansancio, Estado de ánimo, Estado de salud, Características del teclado, Iluminación ambiental, Disposición del equipo. Por tal motivo el método propuesto también comprende de una etapa de autentificación para evitar el problema de tomar muestras con problemas técnicos". Vincenzi (2014).

1.8.1 Metodología del desarrollo de la aplicación

1.8.1.1 Mobile-D

Mobile-D está pensado para grupos de no más de 10 desarrolladores colaborando en un mismo espacio físico. Si trabajan con el ciclo de desarrollo propuesto, los proyectos deberían finalizar con el lanzamiento de productos completamente funcionales en menos de diez semanas.

La aproximación de Mobile-D se ha apoyado en muchas otras soluciones bien conocidas y consolidadas: eXtreme Programming (XP), Crystal methodologies y Rational Unified Process (RUP). Los principios de programación extrema se han reutilizado en lo que se refiere a las prácticas de desarrollo, las metodologías Crystal proporcionaron un input muy valiosos en términos de la escalabilidad de los métodos y el RUP es la base para el diseño completo del ciclo de vida.

El ciclo del proyecto se divide en cinco fases vea la figura 1.1: exploración, inicialización, producción, estabilización y prueba del sistema. En general, todas las fases (con la excepción de la primera fase exploratoria) contienen tres días de desarrollo distintos:
Planificación, trabajo y liberación. Se añadirán días para acciones adicionales en casos particulares.

Figura 1.1 Las cinco fases del metodo Mobile-D

Fuente: Marchenko, 2008

1.8.2 Población y muestra

El universo está constituido por los sistemas que requieran un nivel de seguridad de nivel medio-alto en la autentificación de los usuarios.

Muestra: La muestra está constituida aplicando el prototipo de autenticación basado en la dinámica de tecleo a un los sistemas operativos android, ISO.

1.9 Aporte

Se verifico que el estudio de la dinámica de tecleo por cada usuario genera patrones de identificación únicos, lo que conlleva a crear el método de autenticación basada en la dinámica de tecleo.

CAPITULO II

MARCO TEORICO

2.1 Introducción

Para un mejor entendimiento de los conceptos que se utiliza, definiremos los conceptos de autenticación , autentificación, dinámica de tecleo, seguridad de información, metodología mobile-d, sistema operativo android para un claro entendimiento del capítulo 3 de la metodología para el prototipo, como se resolvió el problema principal, y como se llegó a una hipótesis.

2.2 Autenticación

Permite certificar que la identidad de las entidades participantes en la comunicación es verdadera. Se logra verificando dichas entidades usando mecanismos como firmas digitales, certificados digitales o características biométricas, de acuerdo a los siguientes tres factores de (Reynoso 2014) son:

a) **Algo que el usuario tiene:** puede ser un dispositivo difícil de clonar, al mismo tiempo tiene que ser común y sencillo para los usuarios. Por ejemplo un teléfono inteligente ya que tiene poder de cómputo y así es posible almacenar información protegida contra manipulaciones.

b) **Algo que el usuario sabe:** como una contraseña lo suficientemente compleja, mezclando mayúsculas, minúsculas y dígitos, además no debe tener ninguna relación con el usuario respecto a sus asuntos personales, profesionales o familiares.

c) **Algo que el usuario es:** como las huellas dactilares, el tono de voz, la forma de escribir o patrones oculares.

2.2.1 Autentificación Continua

La autenticación continua de usuarios se sirve de las mismas estructuras y técnicas que la verificación de claves estáticas (parámetros, clasificadores, métricas de distancia, vectores característicos) aunque agrega algunas limitaciones propias; entre ellas, la necesidad de

fragmentar el texto de entrada para evitar considerar las pausas en el ingreso de datos como mediciones efectivas de parámetros y la imposibilidad de utilizar métricas que tengan en cuenta la correlación ya que es esperable que el texto a verificar difiera sustancialmente del texto de entrenamiento (Zhong 2012). Este último fenómeno fuerza a utilizar modelos de menor precisión para evitar que la tasa de falsos rechazos crezca excesivamente.

2.2.2 Autorización

Para la informática, la autorización es la parte de un sistema operativo que protege los recursos de sistema, de modo tal que sólo puedan ser utilizados por los usuarios que cuentan con permiso para eso.

La autorización, por lo tanto, es una especie de permiso. Consiste en dar consentimiento para que otros hagan o dejen de hacer algo.

En este sentido, la autorización puede consistir en que una persona en concreto, que por determinados motivos no pueda realizar una acción necesaria para ella, establezca mediante el correspondiente documento acreditativo que autoriza a un familiar o amigo para que lleve a cabo aquella por él (Porto, Merino 2009).

2.2.3 Patrones

Es un archivo comparativamente pequeño que se deriva de las características de una muestra o muestras del usuario, que se utiliza para obtener las correspondencias biométricas en el proceso de la comparación.

El patrón se crea por medio de un complejo proceso algorítmico que transforma las características diferenciales de la muestra. El concepto de patrón es uno de los elementos que definen la tecnología biométrica, a pesar de que no todos los sistemas biométricos utilizan patrones para realizar el proceso de comparación, puesto que algún sistema de reconocimiento de la voz utiliza la muestra original para realizar la comparación biométrica.

Dependiendo de cuándo hayan sido generados, los patrones pueden referirse a patrones de registro o de verificación (Marroquín 2005) tiene dos puntos.

- **Patrones de registro.** Se crean en la primera interacción del usuario con el sistema biométrico, y se almacenan para ser utilizados en futuras comparaciones.

- **Patrones de verificación.** Se generan durante los siguientes intentos de verificación, al comparar la característica con la almacenada en el patrón.

Se pueden utilizar múltiples muestras para generar el patrón de registro, el reconocimiento facial, por ejemplo, utilizará varias imágenes de la cara para generar el patrón de registro.
El patrón de verificación se deriva normalmente de una única muestra. Un patrón procedente de una única imagen facial se puede comparar con el patrón de registro para determinar el grado de similitud.

2.3 ¿Que es biometría?

"Los datos biométricos se refieren a las características físicas (y conductuales) más propias de cada uno, que pueden ser detectadas por dispositivos e interpretadas por computadoras de modo que puedan usarse como nuestros representantes en el ámbito digital. De este modo podemos vincular nuestros datos digitales y nuestra identidad de forma permanente, consistente y sin ambigüedad y recuperarlos rápida y automáticamente recurriendo a las computadoras. "
(sales@Aware.com)

2.3.1 Tipos de tecnología biométrica

Ya que la biometría se basa tanto en características físicas como en el comportamiento, podemos diferenciar dos tecnologías de este tipo escrito por (Reynoso 2014).

2.3.2 Biometría estática

Mide la anatomía del usuario, se basa en medidas y datos derivados de la medición directa de una parte del ser humano. Dentro de esta clasificación, podemos encontrar:

- Huellas digitales
- Geometría de la mano.
- Análisis del iris.
- Análisis de la retina
- Reconocimiento facial.

2.3.3 Biometría dinámica

Mide el comportamiento del usuario. Son sistemas orientados al reconocimiento o autentificación del usuario basados en la utilización de factores asociados al comportamiento del usuario: cómo se mueve, cómo articula los sonidos y, lo que es más importante, cómo interactúa con el sistema en sí que lo está intentando reconocer (Reynoso 2014).

Dentro de esta clasificación, encontramos:

- Patrón de voz.
- Firma manuscrita o verificación de escritura.
- Dinámica del tecleo.
- Análisis gestual.

2.3.4 Aplicaciones de la biometría

La primera modalidad biométrica utilizada fueron las huellas dactilares usadas para identificar a un sospechoso en una investigación criminal. Con el auxilio de las nuevas tecnologías de captura de imágenes y el poder de las computadoras, ese proceso que usaba el papel y era trabajoso se hizo mayormente digital (aunque no totalmente) y automatizado. Nuevas tecnologías permiten emplear la identificación biométrica en otras aplicaciones, a saber, en el proceso de "autenticación" física y lógica de las personas en los controles de acceso, así como en el reconocimiento casi en tiempo real de sospechosos en un control fronterizo, y en otras aplicaciones en las cuales es necesario acceder a los datos muy rápidamente.

Las aplicaciones biométricas pueden agruparse según tres objetivos: 1) verificación, 2) identificación y 3) control de duplicados:

La verificación requiere que se realice una comparación biométrica "uno-a-uno" a fin de asegurar el acceso ya sea a un activo físico o digital, tal como una aplicación informática o base de datos. En este tipo de aplicación, los datos biométricos se usan como contraseña o PIN para reforzar el control de acceso al comparar la muestra biométrica de un individuo con una muestra única confiable almacenada. La muestra almacenada puede estar en una base de datos central, un teléfono inteligente, o ser una clave contenida en una credencial, tal como un documento de identidad inteligente. De esta manera podemos "autenticar" la declaración de identidad de una persona mediante la pregunta siguiente: "¿Es usted la persona a la que se le adjudicó esta clave?"

y utilizar el resultado de la comparación para franquear o impedir su acceso. El empleo de datos biométricos para controlar el acceso es especialmente interesante en las aplicaciones destinadas a la seguridad comercial y personal. La verificación biométrica puede utilizarse como una alternativa muy conveniente o como un refuerzo del PIN o contraseña, en cuyo caso, el usuario puede optar por usar el PIN o la contraseña si le resulta más conveniente. Por ejemplo, el iPhone 5S de Apple ofrece esta alternativa.

La identificación es un proceso distinto y más exigente (en términos de algoritmos biométricos y desempeño informático), que sirve para confirmar si los datos biométricos de un individuo se hallan presentes en una base de datos o "galería". Una galería puede contener miles de millones de plantillas e incluso más. En el proceso de identificación, se capturan los datos biométricos vitales de un individuo y se los envía a un sistema de búsqueda biométrico para una comparación de "uno-a-muchos". El sistema compara matemáticamente la plantilla de la muestra biométrica de sondeo con todas las plantillas existentes en la galería. Al hacerlo, los datos biométricos ayudan a identificar a un individuo aun cuando mienta acerca de su identidad. En general, las aplicaciones para la identificación se utilizan en el sector público donde es vital confiar en la identidad por un problema de seguridad pública, por ejemplo en la investigación criminal y en la aplicación de la ley, la emisión de visas y la gestión de fronteras, la verificación de antecedentes antes de contratar nuevos empleados en áreas como defensa e inteligencia.

El control de duplicados es el proceso biométrico que permite determinar si un individuo figura más de una vez en una base de datos. Este recurso posibilita detectar fraudes tales como cuando un individuo se inscribe varias veces en un programa de beneficios sociales. El proceso implica comparar la plantilla biométrica de cada registro presente en la base de datos con cada uno de los otros en un proceso denominado "des duplicación biométrica".

2.3.5 Biometría Dinámica de Tecleo

La biometría del tecleo se encuentra dentro del área de la biometría dinámica, sistemas basados en la utilización de factores no estáticos, y factores asociados al comportamiento del usuario.

El principal mecanismo de interacción de un humano con una computadora es el teclado, aunque existen otros medios de interacción muy comunes hoy día, como lo es ratón, o incluso el micrófono, pero a pesar de todo el mayor porcentaje de información del usuario a la

computadora viene del teclado, y además es un elemento de hardware que viene de fábrica con todos los ordenadores. Esto, como veremos, es una ventaja fundamental para un sistema de seguridad sobre Internet.

Así pues, aparece una rama de la biometría dedicada al estudio del reconocimiento del patrón de tecleo de un usuario, la biometría del tecleo, la cual se centra en las técnicas necesarias para identificar en qué medida existe una cierta regularidad en el modo de teclear de un usuario de un sistema informático.

El proceso de tecleo es un proceso realmente complejo y que trasciende el aspecto meramente físico, en tanto es una capacidad emergente que surge de la propia dinámica cerebral en su origen. Desde el cerebro generamos los estímulos necesarios que se transmiten por el sistema nervioso periférico hasta nuestros músculos que efectúan complejas contracciones y distensiones para presionar un centenar de teclas de una computadora, plasmando la información verbal que el cerebro está procesando en un momento determinado.

En este tipo de tecnología no se hace necesario tener hardware adicional para el muestreo de patrones, y esto lo hace ideal para aplicaciones sobre Internet (Reinoso 2014).

2.3.5.1 La Frase de Control

Para que el sistema de identificación resulte efectivo, la frase de control utilizada debe caracterizarse por:

- Componerse de letras y números sin sentido. En este caso el cerebro ha de pensar qué teclas pulsar y mover las manos de su posición natural. Si se utilizara una frase con sentido la suplantación de identidad sería posible, ya que las manos se encontrarían en una posición natural y la cadencia sería más fácil de imitar.

- La fase de entrenamiento del sistema, o de inscripción de un nuevo usuario, ha de ser extensa para disponer de muchas muestras y realizar correctamente el patrón de comportamiento del individuo frente al tecleo.

2.3.5.2 Características de la Dinámica de Tecleo

La dinámica del tecleo resulta una buena técnica de control, para la identificación del usuario, y más si es complementaria a otras, como la autenticación mediante usuario y contraseña.

Resulta una aplicación sencilla y práctica, que puede ser integrada sin complicaciones en numerosos entornos informáticos. El coste de los dispositivos que capturan los datos, los teclados, es muy reducido y muchos entornos informáticos ya disponen de ellos.

La fase de inscripción y obtención de las muestras necesarias para determinar el patrón puede realizarse de forma continua mientras el usuario se encuentra tecleando. Eso sí, el sistema ha de tener en cuenta que los distintos teclados pueden variar el ritmo con el que usuario teclea. Otros factores externos, como el estado de concentración del usuario o posibles lesiones físicas pueden influir en el ratio de falsos negativos. Es por ello que este tipo de sistemas suelen ser complementarios a otros sistemas de control de acceso.

2.3.5.3 Reconocimiento Biométrico de la Dinámica de Tecleo

La dinámica de pulsación es el proceso de analizar el modo en el que un cierto usuario escribe en un teclado mediante un constante monitoreo de las entradas que se hacen a través de un teclado, miles de veces por segundo con el objetivo de identificar el comportamiento de este usuario y su patrón de ritmo de tecleo. A través de diversos estudios se ha demostrado que la forma o el patrón de tecleo de un usuario es un rasgo biométrico con una enorme garantía, lo bueno este sistema es que tan solo requiere de un teclado convencional y del software necesario (Banerjee, pág. 2012).

Las técnicas de verificación de usuarios pueden definirse como estáticas o continuas.

La verificación estática, lleva a cabo un análisis de las pulsaciones solo en momentos temporales específicos, un por ejemplo es cuando se realiza la secuencia de entrada de la contraseña en un ordenador. Este método permite obtener una mayor seguridad que la que da tan solo el uso de una clave de acceso, pero tiene el inconveniente de no permitir la detección de una substitución del usuario una vez finalizada la verificación inicial.

Verificación continúa, monitoriza el comportamiento del usuario a lo largo de todo el proceso de interacción del usuario con el teclado.

A pesar de que es posible aplicarlo a la identificación de usuarios, el análisis de dinámica de tecleo se aplica fundamentalmente a la verificación. En este último caso, se sabe quién es el usuario que demanda ser autenticado y la función del sistema biométrico es verificar que su identidad no está siendo usurpada. Para el primer caso el sistema no cuenta con información adicional más que la dinámica de tecleo, la que debe bastarle para determinar quién está tecleando (Siguenza, 2010).

El modelado de la dinámica de tecleo utiliza mediciones de parámetros presentes en la escritura continua, como ser:

- Latencia entre presión de teclas consecutivas (wait)
- Tiempo de retención de la tecla (hold)
- Velocidad general de tecleo Probabilidad de error (frecuencia de uso de las teclas backspace y delete)
- Hábitos de sectorización del teclado (tendencias en el uso de teclas repetidas, como los números)
- Orden de liberación de teclas al escribir mayúsculas

Se ha reportado también la medición de las diferencias en la presión al teclear Tang, Yang y Wu. (2010), aunque la necesidad de contar con un teclado especialmente diseñado para tal fin hace que dicha técnica no sea de uso extendido. También se ha propuesto una ingeniosa medición indirecta de los parámetros de tecleo mencionados y extracción de otros parámetros significativos por medio del sonido realizado por el usuario al teclear Roth, Liu, Ross y Metadas (2013).

Las métricas generalmente utilizadas para determinar la calidad de los sistemas de seguridad biométrica, incluyendo la autenticación por modelado de cadencias de tecleo, son el FAR (False Acceptance Rate o tasa de falsos positivos) y FRR (False Rejection Rate o tasa de falsos negativos). Valores del orden del 5% o menos son usuales para las implementaciones mencionadas en la literatura.

2.4 Verificación de Claves Estáticas

La verificación de claves estáticas, usualmente entre ocho y veinte caracteres de longitud, se realiza mediante la formación de un vector característico con la secuencia de valores de uno o más de los parámetros enumerados anteriormente, capturados en el momento en que el usuario ingresa su clave. Dicho vector es comparado con un vector patrón que es el resultado de un proceso de entrenamiento en donde la misma clave es ingresada repetidamente. La cantidad de repeticiones requeridas para el entrenamiento de la herramienta dependerá de la velocidad de convergencia del modelo utilizado. El vector patrón generado contiene información de las medias, varianzas y, eventualmente, de la función de ajuste inferida de todos los parámetros utilizados para cada carácter de la clave.

Entre las métricas de distancia utilizadas para comparar el vector característico para la autenticación actual y el vector patrón se encuentran la distancia euclídea Messerman (2011), L1 o de Manhattan Joyce (2011) y la distancia de Mahalanobis Bleha, (1990). En (Zhong 2012) se describe una novedosa métrica de distancia específica para aplicación en verificación de cadencias de tecleo, que combina ciertas propiedades de la distancia de Manhattan y de Mahalanobis y se beneficia simultáneamente tanto de la tolerancia a valores fuera de rango de la primera como de la sensibilidad a las correlaciones entre parámetros de la segunda.

2.5 Capas de Seguridad en Dispositivos Móviles

Para poder entender la importancia de la seguridad en los dispositivos móviles hay que conocer las capacidades de estos dispositivos y cómo se ha llegado a ellas. Principalmente, los dispositivos móviles han vivido una importante revolución en cuanto a las aplicaciones que pueden ejecutar. Esta revolución ha estado marcada por tres motivos (Domingo Prieto, S.f.):

a) Un hardware potente, con muchos sensores.

b) Un sistema operativo complejo que facilita un SDK1 sencillo y potente para los desarrolladores.

c) Un mercado de aplicaciones completamente integrado en el sistema y muy intuitivo, lo que facilita las transacciones tanto a los usuarios como a los desarrolladores.

Debido a estas nuevas funcionalidades que ofrecen los sistemas operativos para móviles y a las aplicaciones que se han creado sobre ellos, los dispositivos móviles acaban almacenando gran cantidad de datos, generalmente confidenciales.

Ya no únicamente guardamos los números de teléfono de nuestros contactos, el registro de llamadas o los SMS, sino que también almacenamos una gran cantidad de información personal, como pueden ser cuentas bancarias, documentos o imágenes

Este aumento de la información personal almacenada provoca que más personas puedan estar interesadas en obtenerla. Además, la complejidad actual de los sistemas operativos para móviles ha incrementado los agujeros de seguridad expuestos. Por lo tanto, cuando utilizamos dispositivos móviles, es recomendable seguir unas prácticas de seguridad, que serán parecidas a las utilizadas en los ordenadores.

Para poder analizar la seguridad de los dispositivos móviles de manera eficiente, se ha organizado este módulo en cuatro apartados: la comunicación inalámbrica, el sistema operativo, la aplicación y el usuario. Cada apartado contendrá una pequeña introducción, una breve descripción tanto de los principales ataques como de los principales mecanismos de prevención y un caso de estudio (Domingo Prieto, S.f).

2.5.1 Ataques Informáticos

Un ataque informático es un intento organizado e intencionado causada por una o más personas para causar daño o problemas a un sistema informático o red. Los ataques en grupo suelen ser hechos por bandas llamados "piratas informáticos" que suelen atacar para causar daño, por buenas intenciones, por espionaje, para ganar dinero, entre otras. Los ataques suelen pasar en corporaciones.

Un ataque informático consiste en aprovechar alguna debilidad o falla en el software, en el hardware, e incluso, en las personas que forman parte de un ambiente informático; para obtener un beneficio, por lo general de condición económica, causando un efecto negativo en la

seguridad del sistema, que luego pasa directamente en los activos de la organización (EcuRed, 2017).

2.5.2 Tipos de Ataques

Ataques lógicos (EcuRed, 2017):

- **Trashing** (cartoneo): Este ocurre generalmente cuando un usuario anota su login y password en un papelito y luego, cuando lo recuerda, lo arroja a la basura. Esto por más inocente que parezca es el que puede aprovechar un atacante para hacerse de una llave para entrar al sistema.
- **Monitorización:** Este tipo de ataque se realiza para observar a la víctima y su sistema, con el objetivo de establecer sus vulnerabilidades y posibles formas de acceso futuro.
- **Ataques de autenticación:** Este tipo de ataque tiene como objetivo engañar al sistema de la víctima para ingresar al mismo. Generalmente este engaño se realiza tomando las sesiones ya establecidas por la víctima u obteniendo su nombre de usuario y password.
- **Denial of Service(DoS):** Los protocolos existentes actualmente fueron diseñados para ser hechos en una comunidad abierta y con una relación de confianza mutua. La realidad indica que es más fácil desorganizar el funcionamiento de un sistema que acceder al mismo; así los ataques de Negación de Servicio tienen como objetivo saturar los recursos de la víctima de forma tal que se inhabilita los servicios brindados por la misma.
- **Tampering o Data Diddling:** Esta categoría se refiere a la modificación desautorizada de los datos o el software instalado en el sistema víctima (incluyendo borrado de archivos). Borrado de Huellas: El borrado de huellas es una de las tareas más importantes que debe realizar el intruso después de ingresar en un sistema, ya que, si se detecta su ingreso, el administrador buscará como conseguir "tapar el hueco" de seguridad, evitar ataques futuros e incluso rastrear al atacante.

2.5.3 Mecanismos de Prevención

Como ya hemos visto, los recursos y la información que gestiona el sistema operativo pueden estar en riesgo. Por lo tanto, es importante ver de qué mecanismos de seguridad disponen los sistemas operativos para dispositivos móviles. Los mecanismos de seguridad más importantes son los privilegios de usuarios, el aislamiento de procesos y las actualizaciones (Prieto, s.f.) .

2.5.4 Prácticas de Seguridad

Como hemos visto en este módulo, los dispositivos móviles ya deben ser tratados como un ordenador en cuanto a la seguridad se refiere, puesto que han heredado muchas de sus características. Por lo tanto, muchas de las prácticas de seguridad que aquí veremos serán similares a las que utilizamos cuando estamos delante de un ordenador, pero, como todavía lo vemos como un dispositivo inferior, tenemos una falsa sensación de seguridad. Por lo tanto, es importante seguir estas prácticas de seguridad cuando se utiliza un dispositivo móvil (Prieto, s.f.) Que son:

- **Activar el control de acceso inicial.** Este acceso puede ser mediante el código PIN o usuario y contraseña.
- **Configurar el bloqueo automático.** Después de un tiempo de inactividad es conveniente que el dispositivo se bloquee.
- **Activar autenticación para desbloquear.** Esta autenticación para desbloquear el dispositivo puede ser más simple y rápida que la inicial, como reconociendo un patrón dibujado en la pantalla.
- **Controlar las aplicaciones que se instalan.** Se deben tratar con precaución las aplicaciones que se instalen en el sistema, intentando bajarlas de fuentes de confianza y con una reputación positiva. También hay que revisar los permisos que estas aplicaciones requieren para su funcionamiento (en caso de que el sistema operativo limite las acciones de las aplicaciones por medio de permisos).
- **Mantener todo el software actualizado**. Con el fin de corregir lo mejor posible los problemas de seguridad, es importante mantener tanto las aplicaciones como el sistema operativo actualizados. Además, si es posible, hay que configurarlos para que realicen la actualización automáticamente.
- **Realizar copias de seguridad.** Es muy importante que periódicamente se realicen copias de la información importante que se almacena en el dispositivo. Además, los datos copiados tendrían que encontrarse fuera del dispositivo, por ejemplo en la web.
- **Cifrar la información delicada.** Este cifrado puede realizarse tanto utilizando los servicios que ofrece el sistema operativo como mediante aplicaciones de terceros.

- **Monitorizar el uso de recursos.** Se pueden detectar anomalías realizando un control de la utilización de los recursos del dispositivo móvil por parte de las aplicaciones. Esto incluye revisión de la factura telefónica para detectar posibles usos fraudulentos.
- **Deshabilitar los sistemas de comunicación cuando no se utilicen**. Además de reducir el consumo energético, deshabilitar los sistemas de comunicación cuando no se utilizan puede evitar ataques. Los sistemas de comunicación únicamente se deben utilizar en redes de confianza.
- **Permitir control remoto.** En caso de robo, es importante tener una aplicación en el dispositivo móvil que permita controlarlo remotamente. Así, se puede localizar el dispositivo, recuperar sus datos almacenados o borrar los datos confidenciales para que no sean comprometidos. También se puede tener una aplicación que borre los datos automáticamente después de varios intentos de acceso fallidos.
- **Contactar con el proveedor de servicios en caso de pérdida.** En caso de pérdida del dispositivo, lo primero que hay que hacer es informar al proveedor de servicios para que efectúe el bloqueo del dispositivo.
- **Eliminar la información confidencial antes de desechar el dispositivo.** Al deshacerse del dispositivo, no sabemos en qué manos puede caer, por lo tanto es importante eliminar toda la información que contiene.
- **Tener sentido común.** Se deben seguir las mismas precauciones que se tienen con los ordenadores cuando tratamos con archivos adjuntos a correos electrónicos, enlaces desde SMS y, en general, navegación por Internet.

2.6 Metodología Ágil Mobile D

Se compone de distintas fases: exploración, inicialización, fase de producto, fase de estabilización y la fase de pruebas donde cada etapa posee un día de planeamiento y un día de entregas de las tareas asignadas. Al concluir todas las fases se tiene una aplicación publicable y entregable al cliente (Rodríguez, 2011).

La metodología de desarrollo de aplicaciones móviles, parte como creación del proyecto "ICARUS" en el 2004, posee cualidades de muchas otras metodologías como ser eXtremeProgramming, Crystal Methodologies y Rational Unified Process (Salo & Hulkko, 2008).

- Las ventajas de esta metodología son las siguientes:
- Un costo bajo al realizar un cambio en el proyecto.
- Entrega resultados de manera rápida.
- Asegura el software adecuado en el momento adecuado.
- La metodología también cuenta con las siguientes desventajas:
- No sirve para grupos de desarrollos grandes y segmentados.
- Depende de buena comunicación entre los miembros del equipo.

Mobile-D tiene el objetivo de ser una metodología de resultados rápidos, con mira a grupos de pocas personas o pequeños grupos, los integrantes del grupo deben poseer una habilidad y capacidad similar entre todos. Se compone de varias fases: exploración, inicialización, fase de producto, fase de estabilización y la fase de pruebas; cada una posee un día de planificación y un día de entregas. Posee iteraciones en la fase de producto donde la entrada a la segunda iteración de la fase de producto es el resultado de la iteración 0 y todo está controlado bajo un control de versión para el proyecto (Salo & Hulkko, 2008).

2.7 Fases de la Metodología Mobile D

La metodología cuenta con 5 fases por las cuales pasa el producto a realizarse, la línea de producción empieza con la fase de exploración, después pasa a la fase de Iniciación, luego pasa a la fase de producto posteriormente a la fase de estabilización y la fase de pruebas (Salo & Hulkko, 2008).

2.7.1 Fase de Exploración

Se centra la atención a la planificación y a los conceptos básicos del proyecto. Se realizan los alcances del proyecto y su establecimiento con las funcionalidades donde se va a llegar. Tipo de patrón: Patrón de fase. El propósito de esta fase es la planificación y establecimiento de una buena planificación *"A well planned is half done"*, esta fase es muy importante para establecer las bases para una implementación bien controlada de software, la arquitectura del producto, el proceso de desarrollo y la selección del medio ambiente (Salo & Hulkko, 2008).

Se necesita diferentes grupos diferentes puntos de vista de partes interesadas en el producto para ofrecer una mejor experiencia en la fase de exploración véase la figura 2.1.

Los objetivos de la fase de exploración son:
Establecer los grupos de actores necesarios en la planificación y el seguimiento del proyecto de desarrollo de software.

Definir los alcances y límites del proyecto de desarrollo de software.

Planificar el proyecto respecto al entorno, el personal y los problemas del proceso.
Las entradas de la fase de exploración son:

- La propuesta del producto.
- Biblioteca de procesos de Mobile D.
- Contrato.
- Documento de requisitos iniciales.

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la especificación de requisitos de software. Deberían aplicarse los siguientes principios:

- El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.
- Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.
- Todo requisito debería ser unívocamente identificable mediante algún código o sistema de numeración adecuado.
- Plan de proyecto y descripción del proceso base.

Establecer el proceso de línea de base es un paso donde el proceso de Mobile-D, así como los modelos de desarrollo de software de organización existentes se adaptan a construir un proceso de línea de base que es adecuado para el proyecto incipiente. Cada uno de los patrones de

Mobile-D proporciona apoyo a la adaptación al proporcionar clasificación para la importancia de los patrones en cuanto a su necesidad de la aplicación del producto de software. Sin embargo, también deben considerarse cuidadosamente otros patrones que no sean "esenciales", ya que a menudo pueden ser muy valiosos en, por ejemplo, aspectos a más largo plazo. La adaptación se puede realizar junto con especialistas en procesos de organización, así como los principales grupos de interés del proyecto de desarrollo de software, si está disponible (Salo, 2008).

Para la especificación se puede usar diagramas de casos de uso los cuales pueden ser descritos con UML.

- Normas y restricciones en caso de que existan.

Las salidas de esta fase son:

- El documento de requisitos iniciales donde se ha definido los requerimientos iniciales del desarrollo del producto.
- Plan de proyecto incluyendo línea de tiempo, el ritmo, las terminaciones, los recursos del proyecto, los actores y sus responsabilidades.
- Descripción base del proceso que incluye la línea de base, las actividades de seguimiento de calidad, documentación, puntos de integración el hardware a llegar las salidas.
- Plan de Medición y plan de Formación, descripción de la línea de la arquitectura.

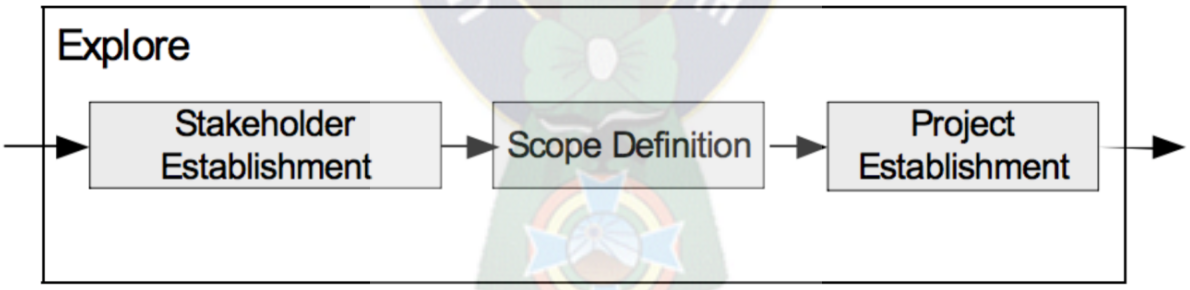

Figura 2.1 Fase de exploracion

Fuente: Koskela, 2004

Las funciones del proyecto en la etapa de exploración son:

- Equipo del proyecto.
- Grupo de apoyo.
- Grupo del cliente y el cliente.
- Grupo directivo.

- Grupo de exploración.

2.7.1.2 UML (Unified Modeling Language)

UML son las siglas de “Unified Modeling Language” o “Lenguaje Unificado de Modelado”. Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software. UML es una herramienta propia de personas que tienen conocimientos relativamente avanzados de programación y es frecuentemente usada por analistas funcionales y analistas-programadores. (Krall, 2014)

2.7.1.3 Casos de Uso

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos:

- **Actor**

  Una definición previa, es que un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema, ver símbolo en Fig. 2.2.

- **Casos de Uso**.

  Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso, ver símbolo en Fig. 2.2.

Relaciones de Uso, Herencia y Comunicación:

- **Asociación**

  Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple, ver símbolo en Fig. 2.2.

Dependencia o Instanciación

Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada, ver símbolo en Fig. 2.2.

Generalización

Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de **Uso** (<<uses>>) o de **Herencia** (<<extends>>).

Este tipo de relación está orientado exclusivamente para casos de uso.

Extends ver símbolo en Fig. 2.2: Se recomienda utilizar cuando un caso de uso es similar a otro.

Uses ver símbolo en Fig. 2.2: Se recomienda utilizar cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

De lo anterior cabe mencionar que tiene el mismo paradigma en diseño y modelamiento de clases, en donde está la duda clásica de usar o heredar (salinas, 2010)

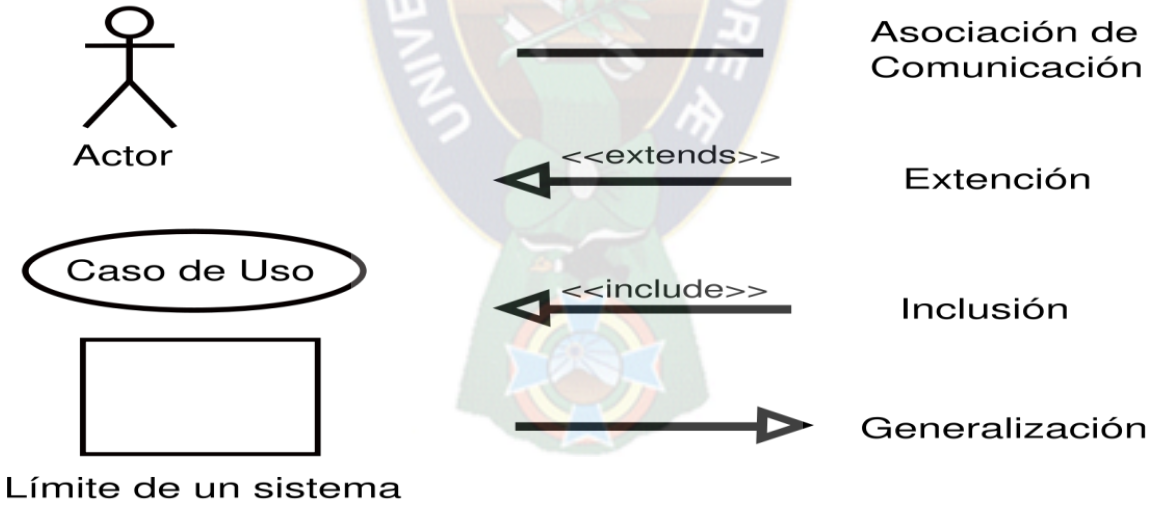

Figura 2.2 Simbologia UML

Fuente: Wikipedia

2.7.2 Fase de Iniciación

En la iniciación se configura el proyecto y se preparan todos los recursos necesarios, se le dedica un día a la planificación y el resto al trabajo y publicación, véase la fig. 2.3.

Tipo de patrón: Patrón de fase

Clasificación de patrón: Esencial

El propósito de esta fase es permitir el éxito de las siguientes fases del proyecto mediante la preparación y verificación de todas las cuestiones fundamentales del desarrollo a fin de que todos están en plena disposición de la aplicación de los requisitos seleccionados por el cliente (Salo & Hulkko, 2008).

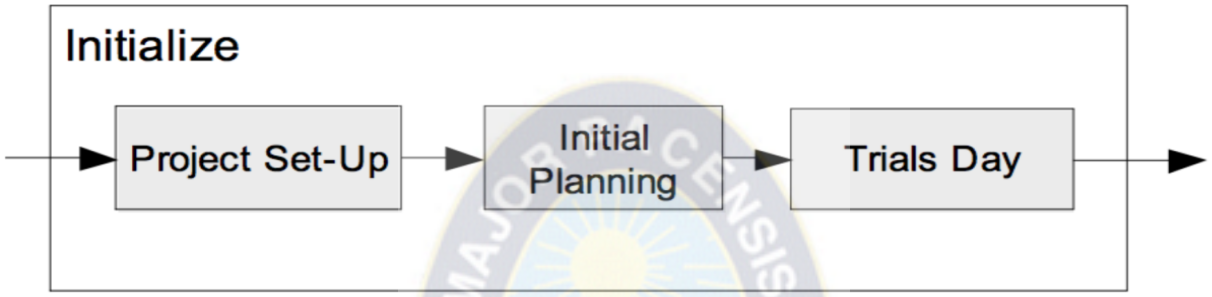

Figura 2.3 Proceso fase de inicialización

Fuente: Koskela, 2005

Los objetivos de esta fase son:

Obtener una buena compresión global del producto para el equipo de desarrollo del proyecto, sobre los requisitos iniciales y la línea de la arquitectura.

Preparar los requisitos físicos, técnicos y humanos, así como la comunicación con el cliente, los planes del proyecto y todas las cuestiones fundamentales de desarrollo a fin de que todo esté en plena disposición para la implementación.

Las entradas de esta fase son:

- Documento de requisitos Iniciales.
- Plan de medición y plan de formación.
- Descripción de la línea de arquitectura.

Las salidas de la fase son:

- Plan de proyecto actualizado
- La 1ra versión del diseño de software.
- Documento con descripción del diseño.
- Funcionalidad implementada.
- Documento de requisitos iniciales actualizados.
- Desarrollo de notas y la interfaz de usuario.

En la tecnología de la información, la interfaz de usuario (UI) es todo diseño en un dispositivo de información con el cual un ser humano puede interactuar - incluyendo pantalla, teclado, ratón, lápiz óptico, la apariencia de un escritorio, personajes iluminados, mensajes de ayuda y como un programa de aplicación o un sitio Web invita a la interacción y responde a él(Rouse, 2014).
Ilustración de cada requisito.

Pruebas aceptadas de cada requisito.

En la etapa de iniciación los roles son los siguientes:

- Grupo del proyecto.
- Jefe del proyecto.
- Arquitectos del proyecto.
- Grupo de apoyo.
- Grupo del cliente.

2.7.3 Fase de Producto

Antes de iniciar el desarrollo de una funcionalidad debe existir una prueba que verifique su funcionamiento, en esta fase se lleva a cabo toda la implementación de los módulos.
Tipo de patrón: Patrón de fase
El propósito en la fase de producción es implementar la funcionalidad requerida en el producto mediante la aplicación del ciclo de desarrollo iterativo e incremental (Salo & Hulkko, 2008). Véase la fig. 2.4.

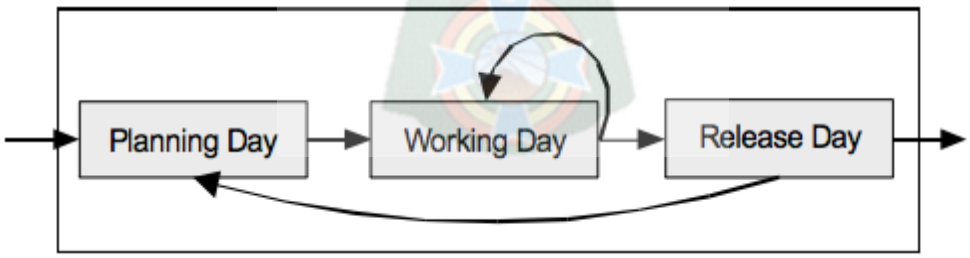

Figura 2.4 Proceso fase de producción

Fuente: Koskela, 2004

Los objetivos de esta fase son:

- Implementar la funcionalidad del producto priorizando los requerimientos del cliente.

- Centrarse en la funcionalidad básica fundamental para permitir múltiples ciclos de mejora.

Las entradas de esta fase son:

- Actualizado plan de proyecto y plan de la línea de la arquitectura.
- La 1ra versión de la arquitectura de software y descripción del diseño.
- Planes para la comprobación de los elementos críticos del desarrollo.
- Funcionalidad implementada.
- Métrica de datos.
- Experiencia del equipo de proyecto.
- Historia y tarjetas de tareas.

Hay tres momentos comunes en los que se trabajarán las historias durante un proyecto ágil:
Comienzo: A menudo se crea una pila de historias, durante el inicio como parte de sus necesidades de visualización de actividades para identificar el alcance de su sistema. Construcción: Durante las iteraciones de construcción, se identifica nuevas historias, se divide las historias existentes cuando se dé cuenta de que son demasiado grandes para implementarse en una sola iteración, redefinir las historias existentes o eliminar historias que ya no se consideran de alcance. El punto es que sus historias evolucionan con el tiempo al igual que otros tipos de modelos de requisitos evolucionan. Además, las solicitudes de mejoramiento pueden ser identificadas por su personal de soporte durante la fase de producción y luego enviadas a un equipo de desarrollo mientras están trabajando en un próximo lanzamiento. Estas solicitudes de mejora son efectivamente nuevas historias.
Transición: A veces se identifican nuevas historias durante la fase de transición, aunque esto no es muy común ya que el enfoque de la liberación es endurecer el sistema y no la nueva funcionalidad. Pero sucede, y estas historias serían priorizadas y colocadas en la pila en orden de prioridad (Ambler, 2003).

- Datos sobre los recursos gastados.
- Manuales, especificaciones API y material de apoyo.
- Pruebas unitarias.

Después de cada Iteración la entrada de la siguiente es:
Los resultados de la iteración anterior.
Los elementos de salida de esta fase son:

- Funcionalidad Implementada.
- Documento de aceptación de pruebas.

Una prueba de aceptación es una descripción formal del comportamiento de un producto de software, generalmente expresado como un ejemplo o un escenario de uso. Se han propuesto una serie de diferentes notaciones y enfoques para tales ejemplos o escenarios. En muchos casos, el objetivo es que sea posible automatizar la ejecución de tales pruebas mediante una herramienta de software, ya sea ad-hoc para el equipo de desarrollo o fuera de la plataforma.
Similar a una prueba unitaria, una prueba de aceptación generalmente tiene un resultado binario, pasa o falla. Un fallo sugiere, aunque no prueba, la presencia de un defecto en el producto.
Los equipos maduran en su práctica de pruebas de aceptación de uso ágil como la principal forma de especificación funcional y la única expresión formal de las necesidades del negocio. Otros equipos usan pruebas de aceptación como complemento a los documentos de especificación que contienen casos de uso o más texto narrativo (Martin, 2003).

- Notas de desarrollo.
- Ilustraciones de Interfaz de Usuario

La ilustración utilizada como parte de la interfaz debe convertirse en un elemento funcional de trabajo. Tomando la decisión a favor de utilizar la ilustración de cualquier tipo en la pantalla o una página web, el diseñador tiene que pensar a fondo cómo tomar todo lo posible de su amplio potencial. La ilustración en la mayoría de los casos se convierte en la manera eficiente de proporcionar al usuario una información más rápida y sencilla de lo que podría ocurrir con el texto. Usando la ilustración en la disposición, es posible satisfacer las necesidades múltiples del usuario que es porqué es tan popular en interfaces del usuario de diversas clases (Yalanska, 2016).

- Lista de puntos de acción.
- Actualizado plan del proyecto.
- Historias y tarjetas de tareas.
- Conocimiento de los requisitos del sistema y pruebas de aceptación.

Una vez definido el documento de requisitos iniciales y el documento de aceptación de pruebas podemos decir que ya se tiene el conocimiento de requisitos del sistema y las pruebas de aceptación para el software.

- Lista de defectos.
- Documento de requisitos iniciales.
- Informe de estado diario.

La fase de producto usa los mismos roles que las anteriores fases, sin embargo, la comunicación con el cliente se debe enfatizar con retroalimentación rápida durante la ejecución de esta fase para lograr resultados satisfactorios.

Los roles son:

- Equipo del proyecto.
- Grupo de apoyo.
- Grupo del cliente.
- Grupo directivo.

2.7.4 Fase de Estabilización

En esta fase se llega la integración para vincular los módulos separados en una única aplicación.

Tipo de patrón: Patrón de fase.

Clasificación de patrón: Esencial

El propósito de la fase de estabilización es asegurar la calidad de la implementación del proyecto (Salo & Hulkko, 2008). Véase la fig. 2.5.

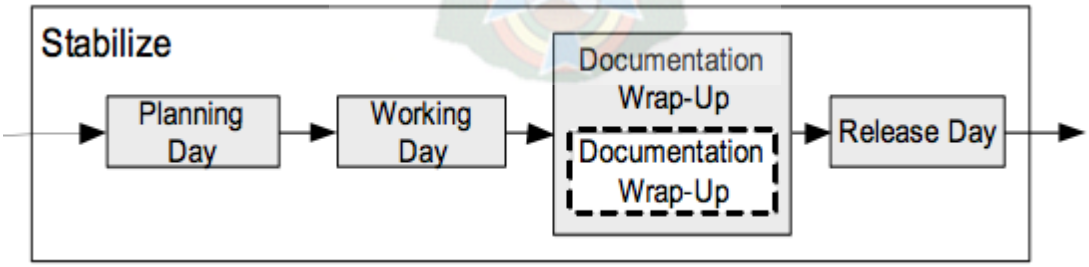

Figura 2.5 Proceso fase de estabilización

Fuente: Ihme, 2004

Los objetivos de la fase de estabilización son:

- Finalizar la implementación del producto.
- Mejorar y garantizar la calidad del producto.
- Finalizar la documentación del proyecto.

Las entradas de la fase de estabilización son:

- La funcionalidad implementada del producto.
- Los artefactos de desarrollo relacionado.

Las salidas de esta fase son:

- La funcionalidad implementada de todo el proyecto de todo el software.
- La documentación del producto finalizado.

El propósito de un documento es transmitir información sobre el sistema de software de una manera persistente. Normalmente, los modelos se crean antes que los documentos del proceso Mobile-D. En la primera fase del proceso, la justificación de las decisiones de diseño sobre la estructura del sistema se registrará a menudo en un documento de arquitectura y diseño de software. En la fase de Estabilización, algunos modelos serán incluidos o refinados como parte de los documentos de software, aunque muchos simplemente serán descartados una vez que hayan cumplido su propósito. El término documentación incluye también comentarios en código fuente. Aunque los comentarios en código fuente no forman parte de los documentos de software, ambos deben ser coherentes (Salo & Hulkko, 2008).

En la fase de estabilización se tiene las siguientes funciones o roles del equipo de trabajo:

- Equipo del proyecto.
- Jefe del proyecto.

Arquitectos del proyecto.

- Grupo de apoyo.
- Grupo del cliente.

  Grupo directivo.

2.7.5 Fase de pruebas

Se pasa al testeo hasta tener una versión estable del producto según lo establecido por el cliente. Si es necesario se reparan errores pero no se desarrolla nada nuevo. Una vez terminado todas las fases se debería contar con una aplicación publicable y entregable al cliente.
Tipo de patrón: Patrón de fase
Clasificación de patrón: Esencial

El propósito de la fase de pruebas es ver si el sistema productor implementa la funcionalidad definida del cliente correctamente, proporcionar la retroalimentación al equipo de desarrollo de los defectos y errores encontrados en la funcionalidad del software para ser corregidos estos defectos encontrados (Salo & Hulkko, 2008). Véase fig. 2.6.

Los objetivos de la fase de pruebas son:

- Probar el sistema basado en la documentación producida en el proyecto.
- Proporcionar información de defectos encontrados.
- Planificar la solución a los defectos encontrados.
- Fijar los errores hallados.
- Producir un sistema libre de errores como sea posible.
- Las entradas de esta fase son las siguientes:
- La funcionalidad implementada.
- Documentación de aceptación de pruebas.
- Funcionalidad del usuario definida completamente.
- Descripción de la interfaz de usuario que se utiliza para crear casos de pruebas.

Las salidas de la fase de pruebas son:

- Un sistema testeado y corregido (versión final)
- Documentación de errores encontrados.
- Informe de pruebas del sistema descripción del proceso de pruebas y los errores y defectos encontrados en el software.
- Registro de pruebas realizados en el sistema y los resultados obtenidos al momento de ejecutar el testeo.

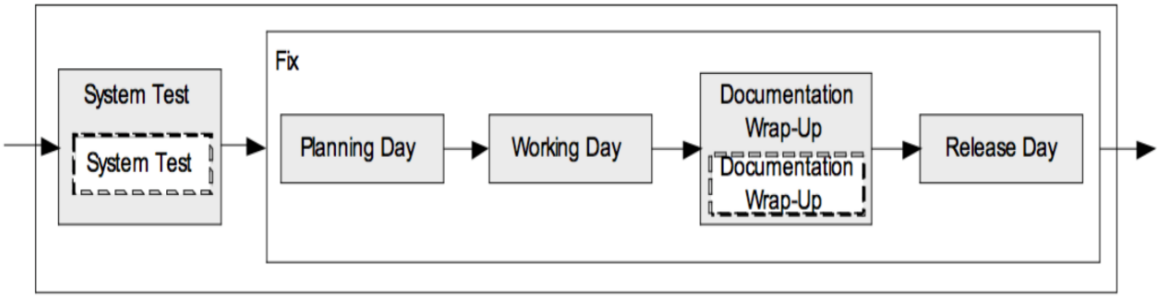

Figura 2.6 Procesa fase de pruebas

Fuente: Ihme, 2004

La prueba del sistema es una etapa en la que el sistema se prueba como se describe en el patrón de tareas de prueba del sistema. Los defectos encontrados se documentan con el propósito de la iteración Fix.

Fix es una variación de la iteración normal; Sin embargo no se implementa ninguna nueva funcionalidad y la escala de tiempo puede ser notablemente más corta. La entrada para esta iteración son los defectos encontrados en la etapa de prueba del sistema. Día de la planificación. El objetivo de la etapa del día de planificación de la fase Prueba y arreglo del sistema es definir los contenidos (por ejemplo, historias y tareas) para la iteración Fix. Los defectos encontrados en la etapa de prueba del sistema son entradas para las descripciones de las tareas.

Día de trabajo. El objetivo de la etapa de día de trabajo de la fase de prueba y arreglo del sistema es solucionar los defectos encontrados en la etapa de prueba del sistema y finalizar la implementación del producto.

Resumen de la documentación. El objetivo de la fase de Documentación de la fase final es finalizar la arquitectura del software, el diseño y los documentos de la interfaz de usuario. La documentación se actualiza para corresponder con los cambios realizados durante la iteración Fix. Como se muestra en la figura, la etapa de Documentación de Wrap-Up incluye una sola tarea, la tarea de Documentación que se puede llevar a cabo utilizando el patrón de tarea de Documentación de Wrap-Up.

Día de lanzamiento. El propósito de la etapa Día de lanzamiento es verificar y validar la funcionalidad implementada y la calidad de todo el software y su documentación. El día de lanzamiento culmina en la versión final de todo el software.

Registro de pruebas realizados en el sistema y los resultados obtenidos al momento de ejecutar el testeo.

En la última etapa, en la fase de prueba se tiene los siguientes roles:

- Equipo del proyecto.
- Grupo de soporte.
- Cliente.
- Grupo directivo.
- Grupo de pruebas del sistema

2.8 Sistemas operativos para móviles

Sistema operativo iOS es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad. No permite la instalación de iOS en hardware de terceros. Tenía el 26 % de cuota de mercado de sistemas operativos móviles vendidos en el último cuatrimestre de 2010, detrás de Android y Windows Phone. Actualmente su sistema operativo se encuentra en la décima versión, mejor conocida como iOS 10.

Android

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tablets o tabléfonos; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró. Android fue presentado en 2007 junto la fundación del Open Handset Alliance que es un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. Los dispositivos de Android venden más que las ventas combinadas de Windows Phone e IOS.

El éxito del sistema operativo se ha convertido en objeto de litigios sobre patentes en el marco de las llamadas «Guerras por patentes de teléfonos inteligentes» en inglés, *Smartphone patent wars*

entre las empresas de tecnología. Según documentos secretos filtrados en 2013 y 2014, el sistema operativo es uno de los objetivos de las agencias de inteligencia internacionales.

La versión básica de Android es conocida como Android Open Source Project (AOSP).

El 25 de junio de 2014 en la Conferencia de Desarrolladores Google I/O, Google mostró una evolución de la marca Android, con el fin de unificar tanto el hardware como el software y ampliar mercados (Topolsky ,2007).

2.9 Phonegap

Phonegap es un paquete de librerías que permite empaquetar aplicaciones HTML5 de manera que puedan ser usadas como apps para móviles o Web Apps. Te explicamos los detalles más relevantes sobre esta versátil solución que nos permitirá dar el salto fácilmente al desarrollo de aplicaciones multidispositivo, aunque no podemos olvidar que también tiene sus limitaciones.

PhoneGap es una solución de Adobe que nos permite llevar el desarrollo para la web al mundo de los dispositivos. Se basa en una "envoltura" que nos permite ejecutar aplicaciones desarrolladas con HTML, CSS y Javascript como si fueran aplicaciones nativas para los teléfonos móviles o tablets.

- Las aplicaciones que podemos desarrollar con PhoneGap se pueden publicar en las conocidas tiendas de aplicaciones (Google Play, Windows Store o App Store de Apple) y, al igual que las aplicaciones nativas, también son capaces de acceder a los periféricos de los dispositivos como la cámara, acelerómetro, etc.
- Lo mejor de PhoneGap es que permite escribir una única vez el código de la aplicación, con tecnologías HTML5, y publicarlo en cualquier plataforma móvil dentro de las más conocidas. Así que, o bien desarrollas tu aplicación en nativo para cada uno de los sistemas operativos, con el consiguiente trabajo que ello conlleva, o la programas una única vez usando alguna plataforma como PhoneGap.

2.9.1 PhoneGap y Apache Cordova

PhoneGap no es más que un producto derivado de Apache Cordova, lo que conocemos técnicamente como una distribución. Sin embargo, hay que aclarar que el proyecto originalmente nació con el nombre de PhoneGap, y luego se cedió a la fundación Apache como software libre.

La comunidad del proyecto, Apache, decidió en 2012 cambiarle el nombre, entre otros motivos para diferenciarlo de la marca PhoneGap, que continúa en poder de Adobe.

Si te preocupa cuál deberías usar, cabe decir que, a día de hoy no hay grandes diferencias. Apache Cordova es susceptible de actualizarse más frecuentemente o más rápido, ya que se trata del producto principal. Por su parte, PhoneGap incluye algunas librerías adicionales que sirven para integrar el sistema con diversos productos de Adobe, pero salvo eso no encontrarás mucho más que los distinga (Alberto Blanco, 2015).

2.9.2 Quién debería usar PhoneGap

PhoneGap es un excelente camino para resolver necesidades de creación de aplicaciones de una manera única y compatible con todos los dispositivos. Las ventajas saltan a la vista en este sentido, pero como todo en la vida también tiene su lado malo, ya que en rendimiento y posibilidades nunca va a poder llegar a la altura del desarrollo nativo. Por tanto, escoger o no PhoneGap para el desarrollo de un nuevo producto es una decisión que hay que tomar con cuidado.

Usa PhoneGap si:

- Tienes intención de usar tus amplios conocimientos de Javascript, HTML5.
- Tu aplicación no requiere exprimir el rendimiento del dispositivo. Son ideales programas de gestión o donde el contenido es parecido al que encontrarías en una web.
- Tienes prisa en lanzar el desarrollo y necesitas un alcance global en todos los dispositivos.
- No requieres usar intensivamente mucha variedad de sensores y periféricos.

No uses PhoneGap si:

- No tienes conocimientos de HTML5 y/o no te importa aprender varios lenguajes para realizar el desarrollo nativo en cada sistema operativo.
- Tu aplicación va a requerir mucha cantidad de procesamiento y quieres exprimir el rendimiento del teléfono o tablet. Los juegos son buenos ejemplos de aplicaciones que sería mejor desarrollar en nativo.
- Solo quieres desarrollar para un único sistema operativo.
- Quieres alcanzar todos los sistemas operativos, pero no tienes prisa para conseguirlo.

- Quieres hacer uso de muchos sensores o periféricos específicos de cada dispositivo (Alberto Blanco, 2015).

2.9.3 Cómo funciona PhoneGap

PhoneGap contiene una serie de librerías que te facilitan todas las utilidades que puedes llegar a necesitar sobre un teléfono. En vez de aprender las librerías propias de cada sistema con sus lenguajes, haces uso de aquellas que te proporciona el framework, usando un único lenguaje de programación, Javascript

Por ejemplo, en vez de comunicar directamente con la cámara, comunicas con las librerías de PhoneGap y éstas son las que, por medio de una especie de puente o interfaz, te permiten interactuar con la cámara, hablando en el idioma que el sistema operativo del dispositivo requiere.

Escribiendo tu proyecto para PhoneGap, en HTML5, podrás luego compilar en los siguientes sistemas operativos:

- Android
- iOS
- WebOS
- Windows Phone

Pero ojo, que las cosas no son tan fáciles. Aunque programes únicamente en Javascript, necesitas instalar todo el conjunto de librerías del sistema donde quieras publicar tu proyecto. Por ejemplo, si estás programando para iOS, necesitas tener tu entorno de desarrollo en un Mac, comúnmente el IDE Xcode, el SDK para desarrollo iOS, una serie de plugins adicionales y la licencia de desarrollador de Apple (Alberto Blanco, 2015)

2.10 Para el desarrollo del algoritmo

2.10.1 Modelo Estático

Este modelo consta de:

- Estimación paramétrica.

- Distancia de dos puntos.
- Máximos y mínimos.
- La media μ.

2.10.2 Estimación Paramétrica

Estimación paramétrica es un tipo de estimación que se usa mucho cuando se implementa sistemas que son similares unos a otros, sistemas con la misma tecnología, o proyectos en áreas funcionales similares. Consiste en detectar variables clave del sistema, indicadores, parámetros, que son los principales determinantes del tamaño del sistema. Se procede listando los parámetros más importantes y preguntarte: ¿Cómo eran esos parámetros en los sistemas anteriores que implementamos? (Esterkin, 2009).

2.10.3 Distancia de dos puntos

El Plano cartesiano se usa como un sistema de referencia para localizar puntos en un plano.

Otra de las utilidades de dominar los conceptos sobre el Plano cartesiano radica en que, a partir de la ubicación de las coordenadas de dos puntos es posible calcular la distancia entre ellos.

Cuando los puntos se encuentran ubicados sobre el eje x (de las abscisas) o en una recta paralela a este eje, la distancia entre los puntos corresponde al valor absoluto de la diferencia de sus abscisas (x_2, x_1).

Cuando los puntos se encuentran ubicados sobre el eje y (de las ordenadas) o en una recta paralela a este eje, la distancia entre los puntos corresponde al valor absoluto de la diferencia de sus ordenadas. (y_2, y_1).

Ahora, si los puntos se encuentran en cualquier lugar del sistema de coordenadas, la distancia queda determinada por la relación (Profesor en linea, 2015):

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Fórmula 2.1 Distancia de dos puntos.

Para demostrar esta relación se deben ubicar los puntos $P_1(x_1, y_1)\ y\ P_2(x_2, y_2)$ en el sistema de coordenadas, luego formar un triángulo rectángulo de hipotenusa P_1, P_2 y emplear el Teorema de Pitágoras.

Demostrando:

Sean $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$ dos puntos en el plano.

La distancia entre los puntos P_1 y P_2 denotada por $d = |P_1, P_2|$ esta dada por:

$$d = |P_1, P_2| = \sqrt{(\boldsymbol{x_2 - x_1})^2 + (\boldsymbol{y_2 - y_1})^2}$$

En la Fig. 2.7 hemos localizado los puntos $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$así como también el segmento de recta $\overline{P_1P_2}$

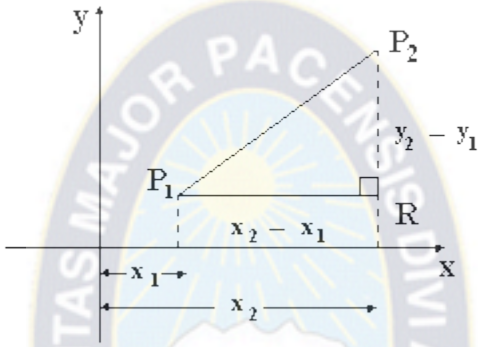

Figura 2.7 Grafia de distancia de dos puntos

Fuente: Ecured

Al trazar por el punto P_1 una paralela al eje x (abscisas) y por P_2 una paralela al eje y (ordenadas), éstas se interceptan en el punto R, determinado el triángulo rectángulo P_1RP_2 y en el cual podemos aplicar el Teorema de Pitágoras:

$$(\overline{P_1P_2})^2 = (\overline{P_1R})^2 + (\overline{RP_2})^2$$

$$Pero\colon (\overline{P_1P_2})^2 = |P_1P_2|^2\ ;$$

$$\overline{P_1R} = x_2 - x_1 \ \ y \ \ , y$$

$$\overline{RP_2} = y_2 - y_1$$

$$Luego, |P_1P_2| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Fórmula 2.2 Demostracion de la distancia de dos puntos

2.10.4 Máximos y mínimos

Ya que max() es un método estático de Math, siempre debe usarse como Math.max(), en lugar de un método de un objeto Math instanciado (Math no es un constructor).

Si no se proporcionan argumentos, el resultado es -Infinity.

Si al menos uno de los argumentos no puede ser convertido a número, el resultado es NaN.

2.10.5 La Media Aritmética.

La media aritmética es el valor obtenido al sumar todos los datos y dividir el resultado entre el número total de datos (Vitutor.net, 2015).

$\bar{x}$ Símbolo de la media aritmética es el símbolo de la media aritmética.

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \cdots + x_n}{N}$$

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{N}$$

Fórmula 2.3Media aritmetica.

Fuente: Vitutor (2018).

2.11 Métodos de la Investigación

2.11.1 Método cuantitativo

El método de investigación cualitativa es la recogida de información basada en la observación de comportamientos naturales, discursos, respuestas abiertas para la posterior interpretación de significados.

Mientras que los métodos cuantitativos aportan valores numéricos de encuestas, experimentos, entrevistas con respuestas concretas para realizar estudios estadísticos y ver cómo se comportan sus variables. Muy aplicado en el muestreo.

Sin embargo, el concepto de método cualitativo analiza el conjunto del discurso entre los sujetos y la relación de significado para ellos, según contextos culturales, ideológicos y sociológicos. Si hay una selección hecha en base a algún parámetro, ya no se considerará cualitativo.

Digamos que es el método de investigación cualitativa no descubre, sino que construye el conocimiento, gracias al comportamiento entre las personas implicadas y toda su conducta observable (Anonimo, S.f.).

2.11.2 Aplicación

Entre las técnicas y los tipos de metodología de investigación cualitativa más populares nos encontramos con la comunicación entre los individuos, como la base de toda ellas.

Los tipos de metodología de investigación cualitativa son, principalmente tres (Anonimo, S.f.):

- **Observación participativa**: el investigador participa del problema o situación a analizar. Vive en primera persona las experiencias y eso es una ventaja a la hora de entender a los sujetos de la investigación.
- **Observación no participativa**: el investigador no participa del problema o situación. Dos ejemplos de este tipo de observación son: simulaciones y estudios de caso. En los primeros se crea una situación y los participantes actúan. Se les observa. Y la segunda práctica, lleva a cabo un estudio exhaustivo de una persona o empresa, institución, etc.
- **Investigación etnográfica**: combina los dos tipos de observación anteriores. Se utiliza para extraer el máximo de datos, al aplicarse tanto técnicas participativas como tipos de observación en los que el investigador no se involucra.

Mientras que las técnicas de análisis de la información cualitativa, pueden ser varias. Destacamos las más comunes (Anonimo, S.f.):

- **Técnicas grupales** En ella destacan los grupos de discusión. La información con diferentes puntos de vista será la más valorada. Pero también se dan técnicas para fomentar la creatividad, como la tormenta de ideas o el Brainstorming.
- **Técnica del Grupo Nominal**. De las más democráticas. Hace posible alcanzar un consenso rápido con relación a cuestiones, problemas, soluciones o proyectos, evitando los términos de 'perdedores' y 'ganadores' entre los miembros del grupo.
- **Técnica del Grupo de Enfoque**. Forma de entrevista grupal que utiliza la comunicación entre investigador y participante.
- **Técnica Delphi**. Se extrae información sobre predicciones y se basa en un panel de expertos.

CAPITULO III

MARCO APLICATIVO

En este capítulo desarrollaremos el método de autenticación basado en la dinámica de tecleo con todo lo descrito en el capítulo 2 así poder alcanzar nuestro objetivo detallado en el capítulo 1.
En la primera etapa de este capítulo procederemos a explicar a detalle todo lo referente al método de autenticación basado en la dinámica de tecleo, en la segunda parte se pasara al desarrollo de la aplicación móvil para así demostrar el método que propone este trabajo para ello surge la necesidad de aplicar un método ágil de desarrollo de iteraciones cortas para el desarrollo de la aplicación por lo cual se eligió la metodología Mobile-D, cuyas fases se desarrolla en el presente capitulo.

3.1 Fases del Método de la Dinámica de Tecleo

El método propuesto para la presente investigación consta de las siguientes fases:

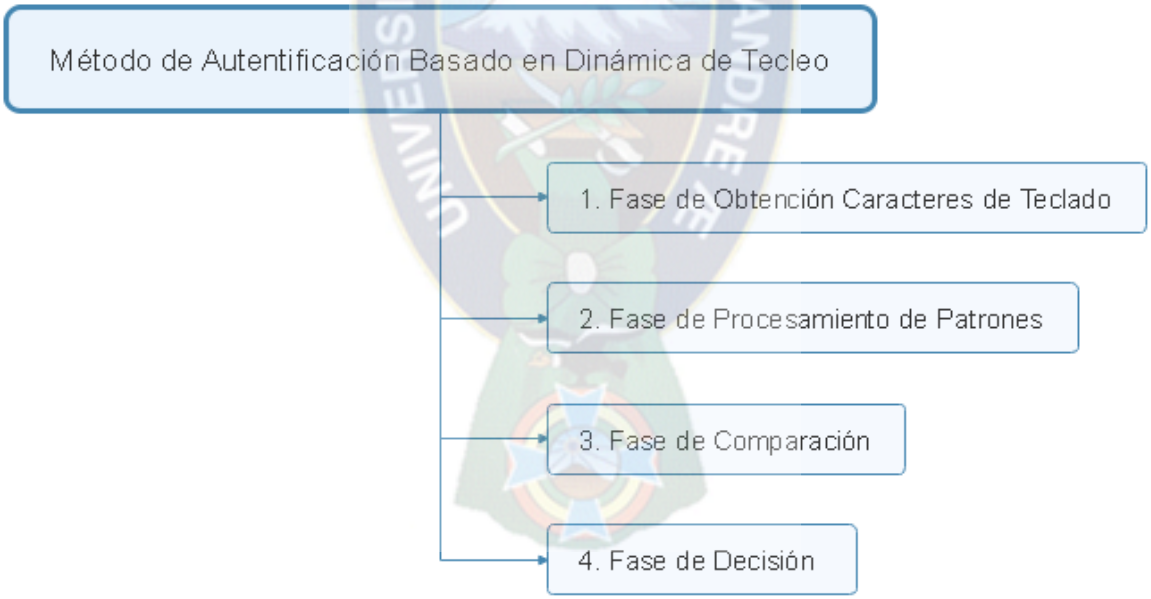

Figura 3.1 Fases del Método de Autentificación Basado en la Dinámica de Tecleo

Fuente: Elaboración propia

3.1.1 Fase 1 Obtención Caracteres de Teclado

Los objetivos propuestos para esta fase son:

- Algoritmo de captura con precisión de milisegundos

- Funcionamiento transparente al usuario

A continuación se detalla las consideraciones a seguir para esta fase:

Consideración 1: Una tecla podría estar presionada durante un largo período, disparando sucesivos keypress (presión de tecla) durante la misma presión de la tecla. En este caso, sólo consideraremos el tiempo de realce desde el último keypress ver fig. 3.2.

```
KeyDown   ← Se comienza a presionar
KeyPress
KeyDown
KeyPress
KeyDown
KeyPress  ← tercera vez que el sistema toma a la tecla como ingresada
KeyDown
KeyPress
KeyDown
KeyPress  ← Consideramos fin de presión y realce desde este momento
KeyUp     ← Se termina de presionar
```

Figura 3.2 Secuencia Iterativa de Presión de Teclas

Fuente: Propia

Consideración 2: Se debe guardar el momento en que se comenzó y se terminó de presionar cada tecla independientemente, ya que el usuario podría presionar más de una simultáneamente ver fig. 3.3.

```
KeyDown  "A"   ← Se comienza a presionar la tecla "A"
KeyPress "A"
KeyDown  "B"   ← Se comienza a presionar la tecla "B"
KeyPress "B"
KeyDown  "B"
KeyPress "B"
KeyDown  "B"
KeyPress "B"
KeyUp    "A"   ← Se termina de presionar la tecla "A"
KeyUp    "B"   ← Se termina de presionar la tecla "B"
```

Figura 3.3 Secuencia Iterativa de intervalos entre teclas

Fuente: Propia

Consideración 3: Cuando se presiona una tecla normalmente, el evento keypress suele dispararse casi inmediatamente después de keydown, por lo que la presión pasa a tomar valores casi insignificantes, y el realce en realidad estará tomando valores muy cercanos al tiempo total de presión. Además, cuando se mantiene presionada una tecla durante sucesivos ingresos de caracteres, el evento keypress estará muy cercano a keyup, por lo que se invierte la relación (presión cercana al total y realce insignificante).

Consideración 4: La demora con la que pueden dispararse cada uno de los eventos, también depende de la carga del sistema, o del cálculo que se esté realizando luego de cada evento. Lo recomendable para disminuir el grado de error, es intentar realizar la menor cantidad de procesamiento durante el tecleo, postergando el análisis o cualquier otra tarea que consuma recursos para una etapa posterior a la adquisición de datos.

Para el procesamiento que no se pueda postergar a una etapa posterior, se recomienda tratar de homogeneizar, buscando realizar exactamente el mismo cálculo independientemente de la tecla presionada.

Consideraciones Técnicas:

Consideración Técnica 1: Por otro lado el evento key press ni siquiera es disparado o no se reconoce en algunos dispositivos móviles.

Esto es debido a que los teclados virtuales de los dispositivos móviles no son capaces de generar un identificador de tecla para la tecla que ha sido pulsada, caso contrario a los teclados físicos. El teclado estándar que proporciona Google no tiene forma de generar estos códigos. Además, si esto fuese posible, sería muy difícil controlar que todos los IME de los dispositivos móviles se adaptasen a dicho comportamiento (StackOverflow, 2017).

Consideración Técnica 2: El lenguaje de programación JavaScript se encuentra muy limitado para obtener los tiempos reales en los que una tecla se presiona o se deja de presionar. No obstante, podemos considerar otra variable con un menor grado de incerteza, en el que el error en la medición tiende a no ser significativo: el tiempo entre teclas. Esta variable puede ser utilizada con mayor confiabilidad para determinar patrones de tecleo.

Es decir, el tiempo que el usuario tarda entre que deja de presionar una tecla A hasta que comienza a presionar una tecla B suele ser un mejor indicador en estos casos. Además, nos permite aumentar la casuística, teniendo en cuenta que el tiempo A⟶B no será el mismo que el tiempo A⟶C. Entonces, también vamos a medir los tiempos inter tecla: E ⟶ j ⟶ e ⟶ m ⟶ p ⟶ l ⟶ o donde se esperaría tener tiempos con diferencias significativas para cualquier par de teclas. Después de las consideraciones se puede proceder a la ejecución de la fase.

Considérese los siguientes tiempos para la fase de toma de muestras:

Variables de Tiempos:

$T_p = Tiempo\ presion\ (Key\ Press)$

$T_l = Tiempo\ levaten\ (Key\ Up)$

$T_e = Tiempo\ entre\ teclas\ (Key\ Down)$

Tomamos como guía una secuencia de tecleo el cual se denomina secuencia normal

$$k_d \rightarrow k_p \rightarrow k_u$$

Fórmula 3.1 Secuencia normal

Fuente: Elaboración propia

Paso 1 Tiempo entre teclas (T_e)

Teniendo en cuenta la secuencia normal ver Fór. 3.1 de tecleo podemos determinar los siguientes casos para la toma de muestra de tiempo entre teclas.

$$k_{d1}\ , k_{p1}\ , k_{u1}, k_{d2}, k_{p2}, k_{u2}$$

Fórmula 3.2 Ejemplo secuencia normal

Fuente: Elaboración propia

Caso 1.

Cuando se tiene una secuencia normal ver For. 3.3 se puede tomar como tiempo entre teclas.

$$T_e = (k_{d2} - k_{u1}) \quad , \quad [k_d \cong k_p]$$

Fórmula 3.3 Tiempo entre teclas secuencia normal

Fuente: Elaboración Propia

Caso 2.

Cuando se tiene una secuencia anormal $k_{d1}, k_{p1}, k_{d2}, k_{p2}, k_{u1}, k_{u2}$ el tiempo entre teclas será.

Para calcular el tiempo entre teclas se debe aplicar la fórmula ver For. 3.5:

$$k_{d1}, k_{p1}, k_{d2}, k_{p2}, k_{u1}, k_{u2}$$

Fórmula 3.4 Secuencia anormal

Fuente: Elaboración Propia

$$T_e = (k_{u1} - k_{d2})$$

Fórmula 3.5 Tiempo entre teclas secuancia anormal

Fuente: Elaboración Propia

Caso 3.

Generalizando y sabiendo que el tiempo es positivo se tiene.

$$T_e = |k_{d2} - k_{u1}| \quad entonces\ siempre\ sera\ positivo$$

Fórmula 3.6 Generalizando el tiempo entre teclas

Fuente: Elaboración Propia

Paso 2. Tiempo de Presión de una Tecla (T_p)

Si bien el evento Key Press (k_p) de (Javascrip) mide el tiempo en milisegundos, este no es el tiempo total de presión sino el tiempo en que se presiona, lo mismo para key down (k_d) que es el tiempo de inicio, por tanto:

Para toda secuencia normal como secuencias anormales tenemos la for 3.7.

$$T_p = k_{u1} - k_{d1}$$

Fórmula 3.7 Tiempo de presion de una tecla

Fuente: Elaboración Propia

Paso 3 Tiempo de Levante (T_l)

Así como en los pasos anteriores obtendremos el tiempo de levante de una tecla con la siguiente For. 3.8 para todo tipo de secuencia normal ver for. 3.1 como secuencia anormal ver For. 3.4

$$T_l = (k_{u1} - k_{p1}) \qquad T_p \cong T_l$$

Fórmula 3.8 Tiempo de lebante de una tecla

Fuente: Elaboración Propia

3.1.2 Fase 2 Procesamiento de Patrones

Para realizar la extracción de características relevantes (patrón) se debe seguir los siguientes pasos:

Paso 1: El usuario ingresa su contraseña tres veces. A medida que se va ingresando los caracteres se debe ir analizando los tiempos de cada letra que el usuario teclea, esto quiere decir que los tiempos se van almacenando en tres vectores en el vector m_1 se almacenaran los tiempos key downs en el vector m_2 se almacenaran los tiempos de key press por ultimo en el vector m_3 se almacenaran los tiempos de key up ver fig. 3.4.

$$m_1 = [vk_{d1}, \quad vk_{u1}]$$

$$(a, a_1), \quad (b, b_1)$$

$$m_2 = [vk_{d2}, \quad vk_{u2}]$$

$$(c, c_1), (d, d_1)$$

$$m_3 = [vk_{d3}, \quad vk_{u3}]$$

$$(e, e_1), (f, f_1)$$

Figura 3.4 Vector de muestras

Fuente: Elaboración Propia

Paso 2 teniendo los vectores de cada muestra pasaremos a realizar las combinaciones de todas las muestras.

$$\#muestras(m_i) * \#elementos = \#\ distancias$$

Fórmula 3.9 Numero de distancias

Fuente: Elaboración Propia

Ya teniendo la For. 3.9 podemos pasar a realizar las combinaciones.

$$d(m_1, m_2) = \begin{bmatrix} a \\ a_1 \end{bmatrix} \begin{bmatrix} g \\ g_1 \end{bmatrix} = \begin{matrix} d_1(a, g) \\ d_2(a_1, g_1) \end{matrix} \qquad v_1[d_i]$$

$$d(m_1, m_3) = \begin{bmatrix} a \\ a_1 \end{bmatrix} \begin{bmatrix} e \\ e_1 \end{bmatrix} = \begin{matrix} d_3(a, e) \\ d_4(a_1, e_1) \end{matrix} \qquad v_2[d_i]$$

$$d(m_2, m_3) = \begin{bmatrix} c \\ c_1 \end{bmatrix} \begin{bmatrix} e \\ e_1 \end{bmatrix} = \begin{matrix} d_5(c, e) \\ d_6(c_1, e_1) \end{matrix} \qquad v_3[d_i]$$

Paso 3 hallamos los máximos y mínimos de los vectores distancia que obtuvimos en el paso 2.

$$I_{max} = max(v_i) \quad , \quad I_{min} = min(v_i)$$

Los intervalos nos ayudaran en la parte de comparación de usuarios entre un usuario verdadero y un usuario falso.

3.1.3 Fase de Comparación

Se observa que el tomar muestras con key down, key up se obtuvo una dinámica resultante.

La contraseña a teclear será: Seguridad

Figura 3.5 Contraseña a teclear

Fuente: Elaboración Propia

$$\frac{k_{d1} \rightarrow k_{p1}}{S}, \frac{k_{d2} \rightarrow k_{p2}}{E}, \frac{k_{u1} \rightarrow}{S}, \frac{k_{u2}}{E}, \frac{k_{d3} \rightarrow k_{p3} \rightarrow k_{u3}}{G}$$

Figura 3.6 Dinámica de tecleo del usuario verdadero

Fuente: Elaboración Propia

Se evidencia en la fig. 3.5 que el usuario teclea tan rápido que el sistema infiere que S, E son presionadas casi al mismo tiempo, por tal motivo el key up se dispara después.

Entonces el usuario original presenta su dinámica de tecleo para las teclas S, E.
Sin embargo el usuario falso no tiene la misma dinámica de tecleo en las letras S, E

$$\frac{k_{d1} \rightarrow k_{p1} \rightarrow k_{u1}}{S}, \frac{k_{d2} \rightarrow k_{p2} \rightarrow k_{u2}}{E}$$

Figura 3.7 Dinamica de tecleo del usuario falso

Fuente: Elaboración Propia

Haciendo una secuencia de comparación determinamos los tres vectores For. 3.10 esto nos ayudara a realizar un análisis de comparación por niveles.

$$V[k_d] = [k_{d1}, k_{d2}, k_{d3} \cdots\cdots]$$

$$V[k_p] = [k_{p1}, k_{p2}, k_{p3} \cdots\cdots]$$

$$V[k_u] = [k_{u1}, k_{u2}, k_{u3} \cdots\cdots]$$

Fórmula 3.10 Vectores de key down, key press, key up

Fuente Elaboración propia

Teniendo el vector de cada uno de los eventos pasaremos a la búsqueda de los vectores distancias con la ayuda de las combinaciones For. 3.9 estas combinaciones realizaremos para obtener los máximos y mínimos de todas las distancias y al final poder comparar las distancias obtenidas del usuario verdadero con las del usuario falso.

3.1.4 ALGORITMO DE DECISIÓN

En esta etapa pasaremos a analizar los datos que encontremos tanto para la autenticación como para el usuario falso.

- Elección de mejor opción
- Minimizar los falsos rechazos
- Minimizar las falsas aceptaciones

Para realizar los puntos antes mencionados daremos las siguientes consideraciones:

Coincidencias exactas: Cuando la distancia entre puntos sea 0.

Coincidencias permitidas: Cuando la distancia entre puntos este en el rango y/o intervalo permitido, por el método de autenticación de la dinámica de tecleo del usuario

Coincidencias fallidas: Cuando la distancia entre puntos sea menor o mayor al intervalo permitido.

Para la toma de muestras se realizara una toma de 3 muestras cada una comprende:

$m_1 = [vk_{d1}, vk_{u1}]$

$m_2 = [vk_{d2}, vk_{u2}]$

$m_3 = [vk_{d3}, vk_{u3}]$

Dónde:

$vk_{d1}[\ \ ] = (valores\ del\ key\ down\ por\ tecla\ presionada)$

$vk_{u1}[\ \ ] = (valores\ del\ key\ up\ por\ tecla\ precionada)$

Una vez tomadas las muestras se procede a realizar la combinación (entre muestras m1,m2,m3) de distancias entre puntos para hallar el intervalo valido.

Ejemplo. Si la contraseña es: SEGURIDAD

Los tiempos keydown y keyup son:

Tiempos KeyDown	Tiempos KeyUps
@103@69@105@71@74	@51.5@34.5@52.5@35.5@37
@69@70@141@69@107	@34.5@35@70.5@34.5@53.5
@72@69@67@76@68	@36@34.5@33.5@38@34
@105@140@105@104	@52.5@70@52.5@52
@70@104@68@107	@35@52@34@53.5@35.5
@71@65@176@69	@32.5@88@34.5

Tabla 3.1 Tiempos de key down y key up

Fuente Elaboración propia

A continuación calculamos las distancias de punto a punto.

D1= 103-107=4=2	D2=69-72=3=1,73	D3=105-69=36=6
D4=71-67=4=2	D5=74-76=2=1.41	D6=69-68=1=1
D7=70-105=35=5,91	D8=141-140=1=1	D9=69-105=36=6
D10=103-104=1=1	D11=69-70=1=1	D12=105-104=1=1
D13=71-68=3=1.7	D14=69-71=2=1.41	D15=70-65=5=2.23
D16=141-176=35=5.91	D17=69-69=0=0	D18=107-104=3=1.73
D19=72-70=2=1.41	D20=69-104=35=5.91	D21=67-68=1=1
D22=76-107=31=5.5	D23=68-71=3=1.73	D24=105-65=6.3
D25=140-176=36=6	D26=74-107=33=5.7	D27=105-69=36=6

Tabla 3.2 Distancias de las muestras

Fuente Elaboración propia

Intervalo de confianza:

Ahora en base a lo obtenido se procede a generar los intervalos de confianza mínimo y máximo

KD (S)	Datos de : 4,1,3	: min= 1 max= 4
KUP (S)	Datos de : 51.5,53.5,52	: min= 51.5 max=53.5
KD (E)	Datos de : 3,1,2	: min= 1 max= 3
KUP (E)	Datos de : 34.5,36,35	: min= 34.5 max= 36
KD (G)	Datos de : 6,1,5.9	: min= 1 max= 6
KUP (G)	Datos de : 52.5,34.5,52	: min=34.5 max= 52.5
KD (U)	Datos de : 4,3,1	: min= 1 max= 4
KUP (U)	Datos de : 35.5,33.5,34	: min= 33.5 max= 35.5

Tabla 3.3 Máximos y mínimos de las muestras

Fuente Elaboración propia

Una vez encontrado los intervalos de confianza tomamos la decisión de quien es el usuario verdadero y el usuario falso en base a su dinámica de tecleo:

Primero el usuario debe ingresar su contraseña: Seguridad

Tiempos KD	103@69@105@71@74@69@70@141@69
Tiempos KU	51.5@34.5@52.5@35.5@37@34.5@35@70.5@34.5

Luego se procede a hallar las distancias combinadas con cada muestra, y posteriormente se compara si la distancia obtenida es válida, es decir si está incluida en el intervalo de confianza hallado previamente.

Si evaluamos la letra presionada “S” se obtiene el siguiente resultado:

Distancias Combinadas [1]: 1, 3, 0

Intervalos POSITIVO: 1-4 PUNTO: 1

Intervalos POSITIVO: 1-4 PUNTO: 3

Intervalos NEGATIVO: 1-4 PUNTO: 0

Si evaluamos la letra presionada “E” se obtiene el siguiente resultado:

Distancias Combinadas [1]: 0.5, 1.5, 0

Intervalos POSITIVO: 0.5-2 PUNTO: 05

Intervalos POSITIVO: 0.5-2 PUNTO: 1.5

Intervalos NEGATIVO: 0.5-2 PUNTO: 0

Por tal motivo se observa que el usuario concuerda en la presión de las teclas S y E, debido a que sus distancias se encuentran en el intervalo de confianza.

3.2 Fase de desarrollo de la aplicación

En esta fase de desarrollo de la aplicación usamos la metodología Movile-D se caracteriza por sus cinco fases.

- Fase de exploración.
- Fase de inicialización.
- Fase de producción.
- Fase de estabilización.
- Fase de prueba.

3.2.1 Fase de Exploración

En esta fase se realizara la planificación y alcance para el desarrollo de la aplicación, así como la investigación, diferentes librerías de desarrollo para la programación en phonegap para el desarrollo de la aplicación en Android e IOS.

- Revisión y análisis del SDK de phonegap.
- Pruebas y revisión de la conexión entre phonegap y el dispositivo móvil.
- Pruebas de las librerías necesarias en android.

3.2.1.1 Ambiente Físico

El ambiente físico o hardware que se eligió para el desarrollo de la aplicaciones una laptop HP y para el testeo de la aplicación un móvil (Huawei Gr5) ya que cuenta con un teclado dagtil.

- Huawei Gr5, 2Gb RAM, 16Gb ROM, procesador octa-core a 1.5Ghz.
- Samsung Tablet 2 Vercion 4.2.2, 2Gb RAM, 16 Gb Rom.
- Una laptop Hp de 17 pulgadas, RAM 4 Gb, Procesador Intel(r) core(TM)2 Duo.

Para realizar pruebas del prototipo se utilizó los emuladores:

- Sndroid: BlueStacks
- IOS: iPadian

3.2.1.2 Ambiente Técnico

En el ambiente técnico se eligió para el desarrollo de la aplicación y el testeo el paquete de librerías de phonegap que nos brinda desarrollar aplicación para diferentes sistemas operativos como android, ISO, Windows.

- Phonegap.
- Cordova.
- Sublime txt.
- Android.

Instalación de librerías necesarias para hacer la conexión entre phonegap y el movil.

3.2.2 Fase de inicialización

Para esta fase se realiza casos de uso Fig. 3.8, el diagrama de actividad Fig.3.11 y la tabla de planificación de la aplicación.

3.2.2.1 Casos de Uso

El método propone que para acceder a los recursos del sistema informático el usuario tendrá que teclear primero su usuario y a continuación su contraseña.

El usuario es un actor muy importante en este punto de la investigación ya que debemos tomar en cuenta que el usuario tiene 3 cosas muy importantes que son: Algo que el usuario tiene (dispositivo), algo que usuario sabe (contraseña),algo que el usuario es (su dinámica de teclo)

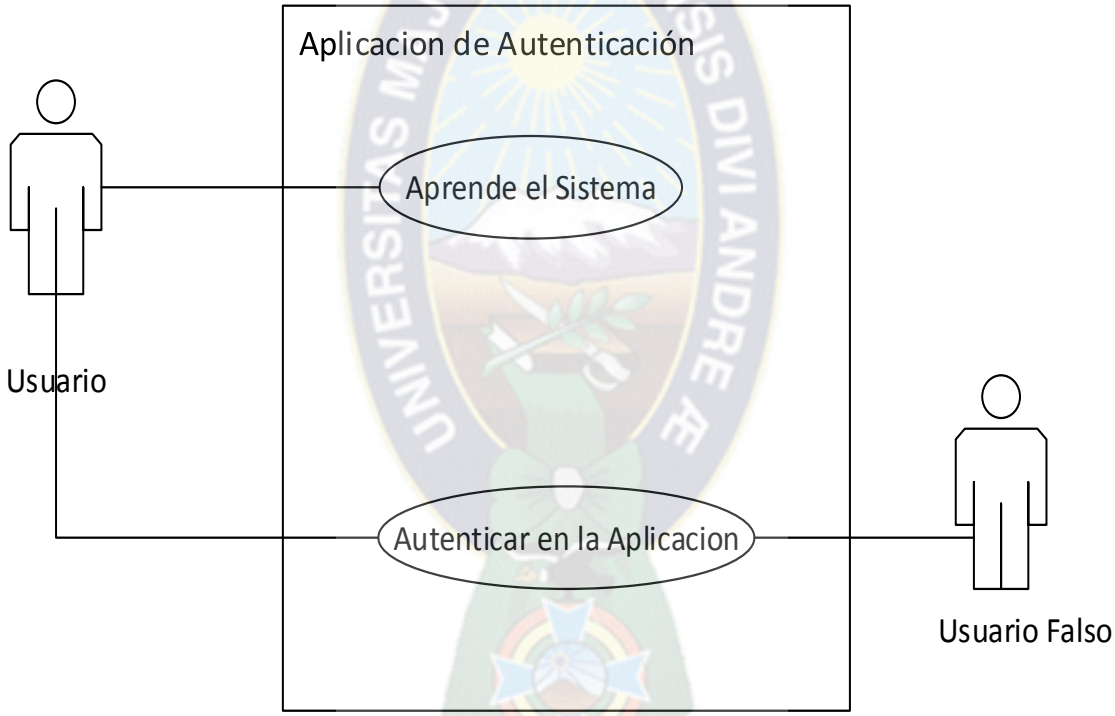

Figura 3.8 Flujo copleto de la aplicación

Fuente: Elaboración propia

Aprende el Sistema: En este caso de uso ver la figura 3.9, y tabla 3.1 en esta etapa de aprendizaje del sistema el usuario deberá introducción la palabra clave “SEGURIDAD” un determinado tiempo (tres veces) para así poder tomar las muestras de su dinámica de tecleo del usuario a autenticarse como se explica en la fase de toma de muestra del método de dinámica de

tecleo para así poder almacenarlo en nuestra base de datos y clasificarlo como un usuario verdadero.

Figura 3.9 Caso de uso de aprendizaje del sistema

Fuente: Elaboración Propia

Descripción de los casos de uso:

Caso de uso : Aprende el sistema
Objetivo: la obtención de los tiempos de key down, key press,key up. **Actores:** Usuario, Móvil, Aplicación de Autenticación.

Tabla 3.4 Descripcion aprendizaje del sistema

Fuente: Elaboración propio

Autenticar en la Aplicación: En este caso de uso ver la fig. 3.10 y la tabla 3.2 el usuario podrá autenticarse en la aplicación introduciendo usuario y contraseña para así poder analizar con el algoritmo y poder verificar si es el usuario autenticado o es un usuario verdadero o al contrario un usuario impostor vale decir usuario falso.

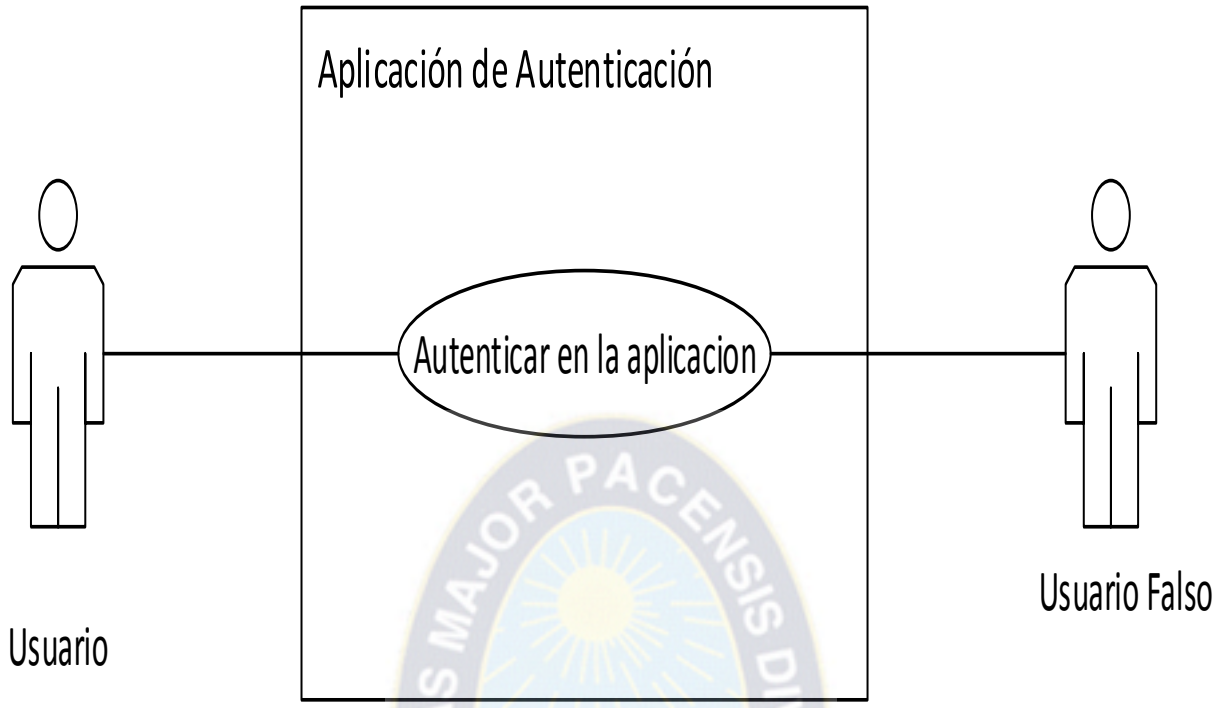

Figura 3.10 Caso de Uso de Autenticar en la aplicación

Fuente: Elaboración propio

Descripción del caso de uso

Caso de Uso: Autenticar en la aplicación.
Objetivo: Obtener datos del usuario, analizar mediante el algoritmo al usuario verdadero como al usuario falso pero el usuario falso no será autenticado.
Actores: Usuario verdadero, móvil, usuario falso**.**

Tabla 3.5 Descripcion del caso de uso de autenticacion en la aplicación

Fuente: Elaboración propio

3.2.2.2 Diagrama de actividades

Este es el enfoque general de la funcionalidad de la aplicación y la iteración de los casos de uso.

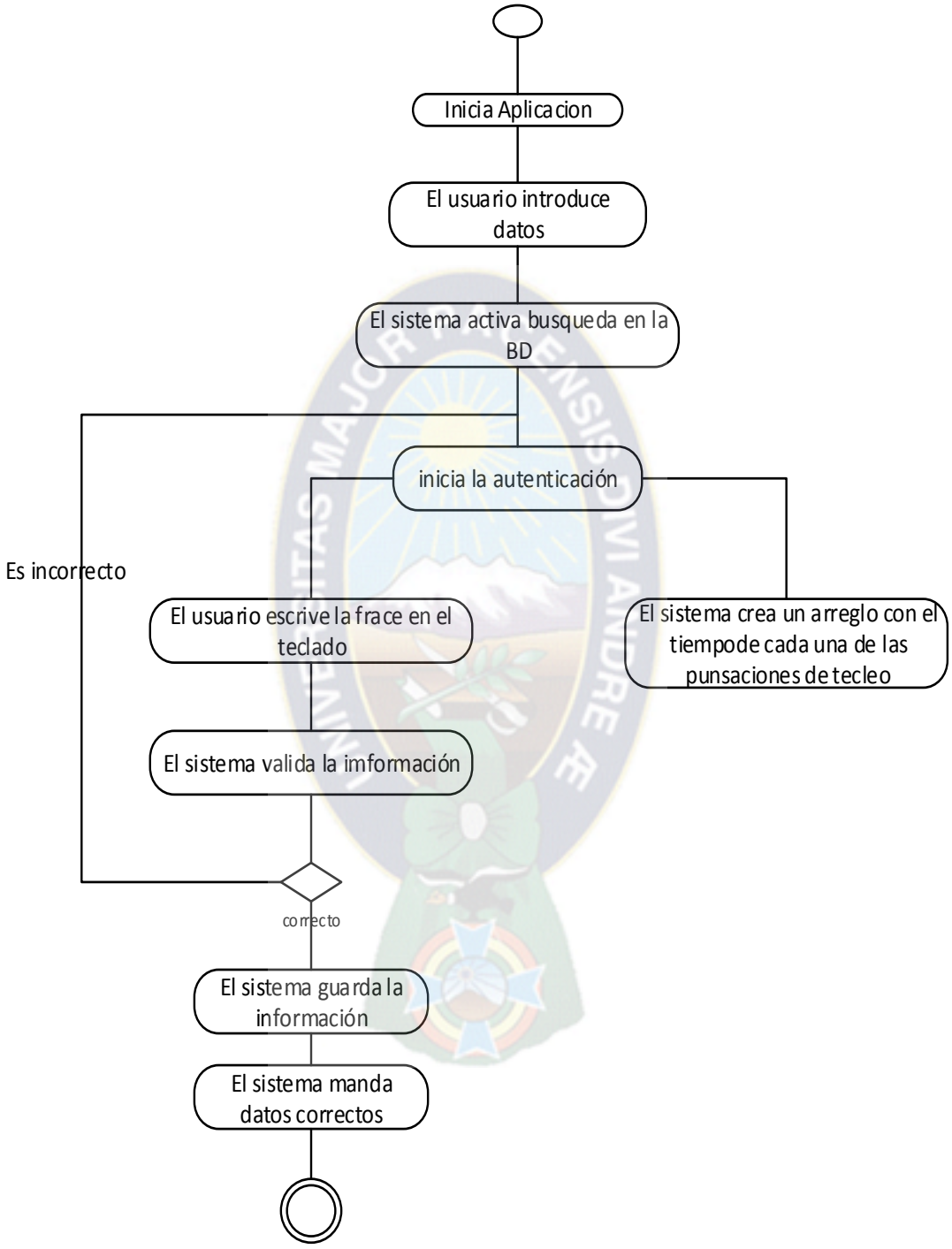

Figura 3.11 Diagrama de actividad de la aplicación

Fuente Elaboración propia

3.2.2.3 Configuración Inicial del Proyecto

Los objetivos principales que debe alcanzar la aplicación en el desarrollo son los siguientes:

- Conectar córdoba con Phonegap.
- Creación inicial del proyecto en android con Jquerymovile.
- Implementación de PHP y Mysql para la conexión a la base de datos.
- Implementar una fase de aprendizaje para el sistema (captura de la dinámica de tecleo).
- Implementar fase de autenticación del usuario (comparación del usuario real y falso).

3.2.2.4 Planificación del proyecto

En la tabla 3.3 se realiza la planificación del método de la aplicación para una mejor organización y llegar a nuestra meta sin ningún problema.

Tareas	Febrero	Marzo	Abril	Mayo	Junio
Fase exploración	Análisis de las variables				
Fase inicialización		Descripción de los casos de uso			
Fase producción		Implementación de la aplicación			
Fase estabilización		Revisar errores del método			
Fase pruebas		Pruebas			
Capítulo IV			Demostración de la hipótesis		
Capítulo V			Conclusiones		
Prototipo				Presentación del prototipo	
Defensa de tesis					Defensa de tesis

Tabla 3.6 Planificación del proyecto

Fuente Elaboración propia

3.2.2.5 Diseño de la Arquitectura del Proyecto

La arquitectura del proyecto está definida por dos fases: la primera fase es la de aprendizaje para el sistema donde recolecta las muestras de todos los tiempos para así amar su dinámica de tecleo. La segunda fase es la de autenticación del usuario en esta fase el usuario envía su dinámica de tecleo para ser comparado en la dinámica que esta guardada en la base de datos y asi determinar si es un usuario falso o es el usuario que dice ser Ver la Fig . 3.10.

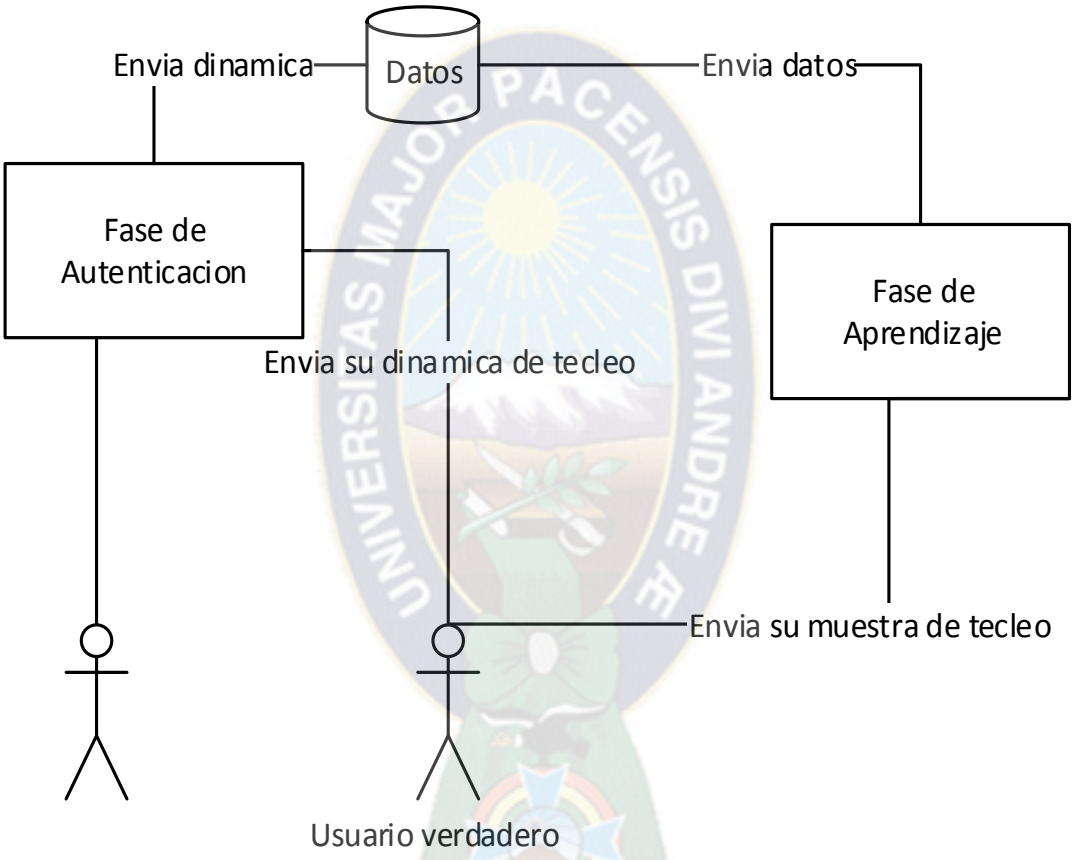

Figura 3.10 Estructura de la aplicacion

Fuente Elaboración propia

Para la implementación de la base de datos usaremos el gestor mysql, phpMyadmin y el uso será para el almacenamiento de muestras y recolección de eventos del usuario, por tanto se define las siguientes tablas con las siguientes columnas, ver la fig. 3.11.

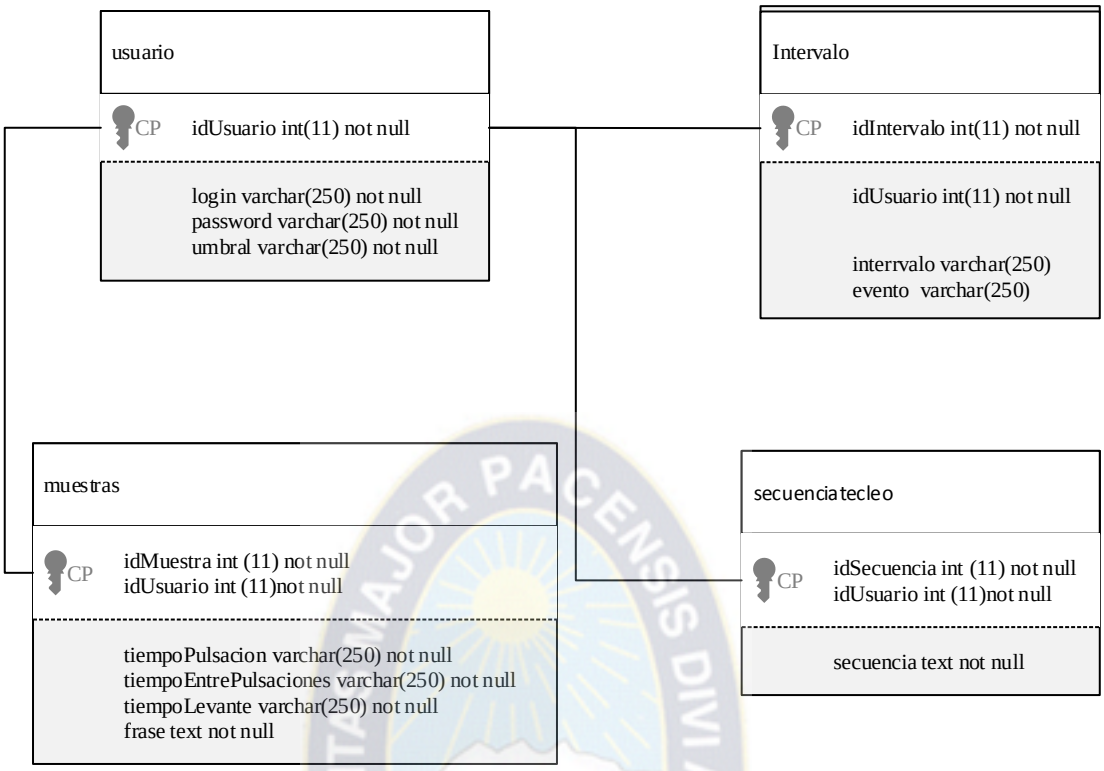

Figura 3.11 Base de datos

Fuente Elaboración propia

La pantalla de configuración inicial para la aplicación de córdoba en phonegap ver Fig. 3.12, en donde se puede observar un menú de las dos fases de la aplicación:

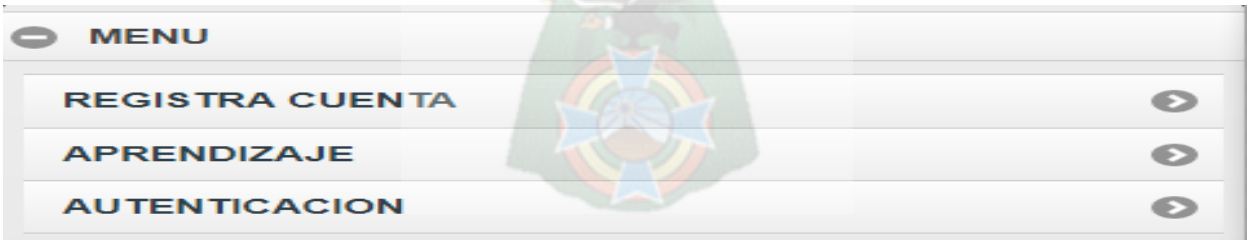

Figura 3.12 Menu de la aplicación

Fuente Elaboración propia

3.2.3 Fase de Producción

En esta fase se completa todos los objetivos planteados en la fase de inicialización, para tal propósito el desarrollo de la aplicación se dividió en 2 iteraciones cada una de 2 semanas de desarrollo.

3.2.3.1 Iteración 1.

3.2.3.2 Día de Planeación

En la iteración 1, se trabajó en la primera tarea, fase de aprendizaje de la aplicación y así poder empezar con la toma de muestras por usuario almacenando en la base de datos y así puedan ser evaluadas sin problemas ni perdida de datos.

Los objetivos de la iteración 1 se muestran a continuación:

- Creación y diseño de la vista de la fase de aprendizaje de la aplicación ver Fig.3.13.
- Implementación del método para la toma de muestras.

```
<script type="text/javascript">
$(document).bind("mobileinit", function () {
    $.mobile.ajaxEnabled = false;
});
</script>
<script src="jquery.mobile-1.0.min.js" type="text/javascript"></script>
</head>
<body>
<div data-role="page" id="pagina3">
<div data-role="header" data-theme="d">
<h1>Autencicacion</h1>
</div>
<div data-role="fieldcontain">
<form action="" method="">
<label for="usuario">Usuario</label>
<input type="text" name="usuario" id="usuario">
<label for="password">Password</label>
<input type="password" name="password" id="password"></form>
</div>
<div data-role="fieldcontain">
    <label for="search">Buscar</label>
    <input type="search" name="search" id="search" placeholder="texto a buscar" value="" />
</div>
<div data-role="footer" data-theme="b">
<h1>salir</h1>
```

Figura 3.13 Vista de la fase de aprendizaje de la aplicación.

Fuente Elaboración propia

3.2.3.3 Fase de Aprendizaje de la Aplicación

El prototipo con el algoritmo de dinámica de tecleo tiene es la siguiente vista:

En la parte principal donde el usuario se registra en el sistema

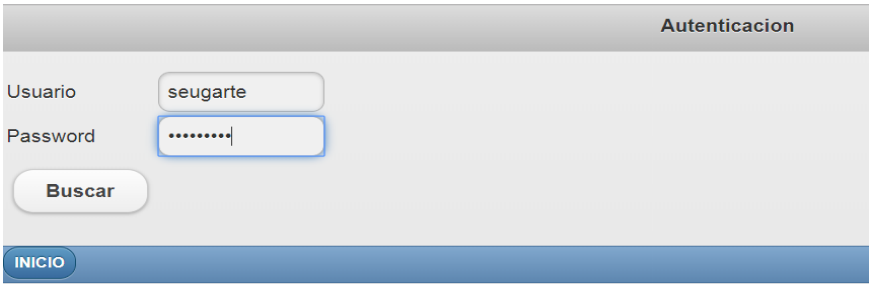

Figura 3.14 Ingreso a la fase de aprendizaje

Fuente: Elaboración propia

3.2.3.4 Fase de Aprendizaje del Sistema

En esta fase el usuario deberá entrenar al sistema introduciendo la frase unas tres veces para así poder capturar el patrón del usuario ver Fig.3.15.

Se toma como ejemplo la frase Fig. 3.15:

Figura 3.15 Frase para generar patrones de comportamiento

Fuente: Elaboración propia

Dónde:

$T_p = Tiempo\ de\ presion\ de\ la\ tecla$

$T_e = Tiempo\ entre\ teclas$

$T_l = Tiempo\ levante\ de\ la\ tecla$

Se introdujo 3 veces la misma frase para generar los patrones:

Figura 3.16 Fase de aprendizaje del sistema

Fuente: Elaboración propia

3.2.3.5 Día de Trabajo (Ciclo)

Durante las 2 semanas del primer ciclo se completa el desarrollo de todos los objetivos planteados en el día de planificación.

3.2.3.6 Día de Entrega

A continuación se muestra los resultados de la fase de aprendizaje como se menciona en la primera iteración, primero entramos al menú ver fig.3.12 y entrar a la fase de aprendizaje ver fig.3.14 introducir usuario y contraseña y buscar en la aplicación donde el usuario ver fig 3.16 introduce la frase "SEGURIDAD" para tomar las muestras del usuario y se guarda su dinámica de tecleo hallando sus máximos y mínimos de su dinámica de tecleo ver Fig.3.17.

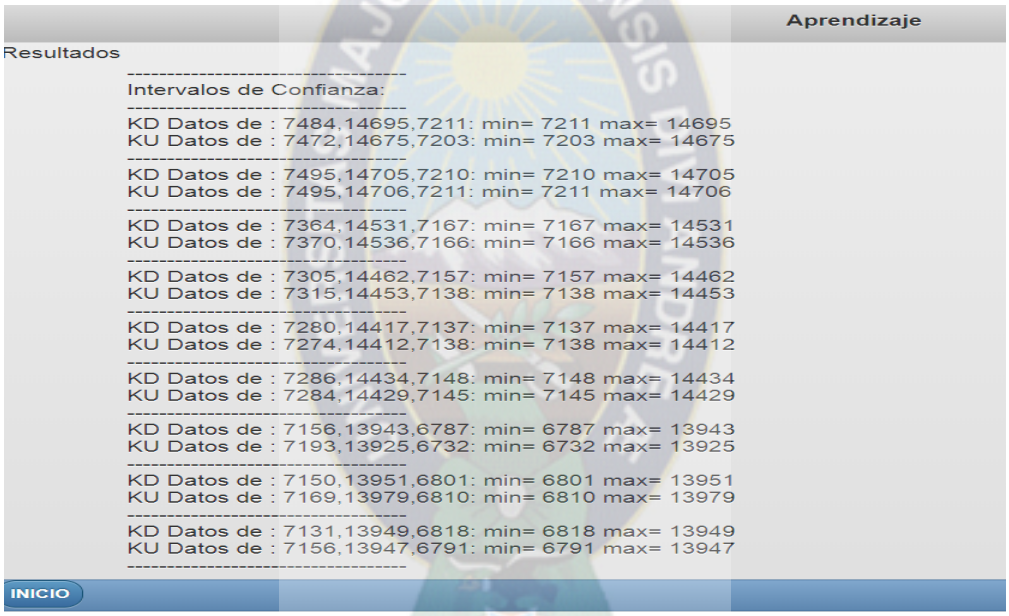

Figura 3.17 Resultados de la fase de aprendizaje

Fuente: Elaboración propia

3.2.4.1 Iteración 2

3.2.4.2 Día de Planeación

En la iteración 2, se trabajó para poder tener la fase de autenticación de la aplicación y realizar la comparación entre un usuario verdadero y un usuario falso.

Los objetivos de la iteración 1 se muestran a continuación:

- Creación y diseño de la vista de la fase de autenticación de la aplicación ver Fig.3.18.
- Implementación del método de la dinámica de tecleo.

```php
public function distanciaPuntos($cadenaPuntosOrignal,$cadenaPuntosRevisar){
    //---------------- vector original
    $vectorPuntos=explode("@",$cadenaPuntosOrignal);//la cadena esta separada por @
    $n=count($vectorPuntos)-1;//suponiendo que no contamos el primer dato nulo
    //---------------- vector a evaluar
    $vectorEvaluar=explode("@",$cadenaPuntosRevisar);
    //---------------- vector resultante
    $vectorResultante[]="";
    //echo "Distancia []----------------------------------------<br>";
    for($i=1;$i<=$n;$i++){
        //echo"->$i -----------------<br>";
        $re1=$vectorEvaluar[$i] - $vectorPuntos[$i];
        //echo"$vectorEvaluar[$i] - $vectorPuntos[$i] <br>";
        $re1=pow($re1, 2);
        //echo"$re1 <br>";
        $d=sqrt($re1);
        //echo"$d <br>";
        $vectorResultante[$i]=$d;
    //  echo"$d <br>";
        //echo"->$i -----------------<br>";
    }
    //echo "----------------------------------------<br>";
    return $vectorResultante;
}
```

Figura 3.18 Código de la distancia de punto a punto

Fuente Elaboración propia

3.2.4.3 Fase de Autenticación del Usuario con el Método de Autenticación.

Después de realizar la toma de muestras el usuario debe autentificarse.

El usuario debe ingresar los datos de usuario y pasword y confirmar si es quien dice ser ver Fig.3.19.

Figura 3.19 Ingreso a la fase de autenticacion

Fuente: Elaboración propia

3.2.3.5 Día de Trabajo (Ciclo)

Durante las 3 semanas del segundo ciclo se completa el desarrollo de todos los objetivos planteados en el día de planificación de la segunda iteración.

3.2.3.6 Día de Entrega

Se puede observar que el usuario introdujo su usuario y contraseña e inmediatamente la aplicación lo que realiza es una búsqueda en nuestra base de datos y realiza un análisis de su dinámica de tecleo para confirmar si es un usuario verdadero o un usuario falso ver Fig. 3.19.

3.2.4 Fase de Estabilización

En esta fase se llevan a cabo las últimas acciones de integración para asegurar que la aplicación completa funcione correctamente.

Para la integración de la aplicación móvil, se tuvo que analizar el algoritmo de aprendizaje para una mejor captura de patrones del usuario.

También se hizo una verificación del módulo de autenticación de la aplicación haciendo énfasis en algunos cambios necesarios para que la aplicación móvil funcione correctamente ver el código a continuación.

```
//codigo para combinar 3 muestras: m1, m2, m3.Combinaciones:  (m1,m2) (m1,m3) (m2,m3)
// distancias de los key downs
$vector1Kd=$this->distanciaPuntos($vk1,$tiempoKeyDown);
$vector2Kd=$this->distanciaPuntos($vk3,$tiempoKeyDown);//
$vector3Kd=$this->distanciaPuntos($vk5,$tiempoKeyDown);

//distancia de los keyUps
$vector1Ku=$this->distanciaPuntos($vk2,$tiempoKeyUps);
$vector2Ku=$this->distanciaPuntos($vk4,$tiempoKeyUps);
$vector3Ku=$this->distanciaPuntos($vk6,$tiempoKeyUps);


 //recupera solo los intervalos kd-ku
 public function obtenerIntervalosKDU($idUsuario){
     $conn=new SERVER();
     $conexion=mysql_connect($conn->servidor(),$conn->usuario(),$conn->password());
     $db=mysql_select_db($conn->baseDatos());
     $queryM=mysql_query("SELECT * FROM intervalos where idUsuario='$idUsuario'
      and evento in ('KD-KU') order by idIntervalo");
     $vecInt[]="";
     $i=0;
     while($lin=mysql_fetch_array($queryM)){
         $vecInt[$i]=$lin[2];
         $i++;
     }
     return $vecInt;
 }
```

```php
//funcion para comparar distancias entre fases de aprendizaje y autentificacion
public function comparaIntervaloConfianza($idUsuario,$tiempoKeyDown,$tiempoKeyUps){
    $conn=new SERVER();
    $conexion=mysql_connect($conn->servidor(),$conn->usuario(),$conn->password());
    $db=mysql_select_db($conn->baseDatos());
    $queryMM=mysql_query("select tiempoPulsacion,tiempoLevante from muestras where idUsuario='$idUsuario'");


    //echo"select tiempoPulsacion,tiempoLevante from muestras where idUsuario='$idUsuario'";

    //sacamos las distanzias minimas y maximas por cada kp y ku de cada letra presionada por el usuario
    $vecMuestrasKd[]="";
    $vecMuestrasKu[]="";
    $i=0;
    echo"<hr>";
    echo"RESULTADOS:";

    /*while($lin=mysql_fetch_array($queryM)){
        echo"<hr>Vector: $lin[0] <br>";
        echo"Vector: $lin[1]<hr>";
        $vecMuestrasKd[$i]=$lin[0];
        $vecMuestrasKu[$i]=$lin[1];


echo"<br>";
echo"Datos Tecleados<br> KeyDowns: $tiempoKeyDown<br>";
echo"KeyUps: $tiempoKeyUps";
//adicionamos en la posicion [4] los datos de autenticacion
$vecMuestrasKd[$i]=$tiempoKeyDown;
$vecMuestrasKu[$i]=$tiempoKeyUps;


$lin=mysql_fetch_array($queryMM);
echo"<br>Valores de Muestras:<br>";
$vk1=$lin[0];
$vk2=$lin[1];
echo"<br> Valores 1: $vk1 .... $vk2 <br>";
$lin=mysql_fetch_array($queryMM);
$vk3=$lin[0];
$vk4=$lin[1];
echo"Valores 2: $vk3 .... $vk4 <br>";
$lin=mysql_fetch_array($queryMM);
$vk5=$lin[0];
$vk6=$lin[1];
echo"Valores 3: $vk5 .... $vk6<br>";
```

3.2.5 Fase de Pruebas y Entrega

Esta fase de pruebas se realizó pruebas en cuanto a la aplicación ya terminada y se encontraron algunos errores que se solucionaron en una iteración.

La aplicación una vez concluida cuenta con las siguientes características:

- Inicio de la aplicación Fig.3.20.

- Menú de la aplicación Fig.3.12.
- Fase de entrenamiento de la aplicación Fig.3.14.
- Toma de muestras de la dinámica del usuario Fig. 3.16.
- Fase de autenticación del usuario Fig. 3.19.

Figura 3.20 Inicio de la aplicaion

Fuente: Elaboración propia

CAPITULO IV

PRUEBA DE HIPOTESIS

Para la prueba de hipótesis se usó el método estadístico, para ello se utilizó una tabla de dinámica de tecleo como premisa, que son datos de tecleo de dos usuarios uno real y falso:

Los datos a considerar son los tiempos de presión (Key Press), cambio de tecle (Key Dows) y levanté (Key Up).

Partiendo de un secuencia normal $Kd \rightarrow Kp \rightarrow kU$.

Numero de Prueba	**Datos de tecleo de usuario verdadero X1**	**Datos de tecleo de usuario falso X2**
1	(Key Dows) 0,078	(Key Dows) 0,199
2	(Key Press) 0,110	(Key Press) 0,499
3	(Key Up) 0,063	(Key Up) 0,499
4	(Key Dows) 0,094	(Key Dows) 0,299
5	(Key Press) 0,062	(Key Press) 0,400
6	(Key Up) 0,062	(Key Dows) 0,399
7	(Key Dows) 0,063	(Key Press) 0,300
8	(Key Press) 0,062	(Key Up) 0,400
9	(Key Up) 0,109	(Key Dows) 0,199
10	(Key Press) 0,109	(Key Press) 0,393
11	(Key Dows) 0,125	(Key Up) 0,328
12	(Key Up) 0,078	(Key Dows) 0,470
13	(Key Press) 0,156	(Key Press) 0,340
14	(Key Dows) 0,141	(Key Dows) 0,318
15	(Key Up) 0,297	(Key Press) 0,361
16	(Key Press) 0,109	(Key Up) 0,152
17	(Key Dows) 0,141	(Key Dows) 0,427
Promedio $\overline{D_{X_2,\overline{D_{X_1}}}}$	0,10935294	0,35194118

Desviación estándar	0,00310952	0,00973841

Tabla 4.1 Comparación de tiempos

Fuente: Elaboración propia

Con estos datos podemos utilizar una prueba de hipótesis para comparar dos muestras que antes de analizar la diferencia entre los promedios muéstrales debemos responder algunas preguntas importantes que harán la diferencia en el procedimiento a seguir.

¿Se trata de dos muestras independientes o pareadas?

Respondiendo a la pregunta

Diremos que dos muestras son independientes cuando no se establece ninguna relación previa al análisis entre las unidades de una y otra muestra. Por ejemplo, sujetos de uno y otro curso, enfermos de dos consultorios, hombres comparados con mujeres.

En cambio diremos que se trata de muestras pareadas si en forma previa al análisis, se forman parejas entre los individuos de una muestra con los individuos de la otra muestra. Por ejemplo el caso con su control, distintas dietas pueden probarse en dos animales de la misma camada. Sin embargo, cuando queda más clara esta situación es cuando se comparan distintas medidas para los mismos individuos; por ejemplo, al medir antes y después del tratamiento a un mismo grupo de individuos se obtienen resultados pareados o correlacionados (Anonimo, 2018)

Como la muestra responde a la pregunta podemos comparar los promedios de dos muestras pareadas:

Por tanto para la comprobación de la tesis se usa t-student.

La prueba t-Student se fundamenta en dos premisas; la primera: en la distribución de normalidad, y la segunda: en que las muestras sean independientes. Permite comparar muestras, $N \leq 30$ y/o establece la diferencia entre las medias de las muestras. El análisis matemático y estadístico de la prueba con frecuencia se minimiza para $N > 30$, utilizando pruebas no paramétricas, cuando la prueba tiene suficiente poder estadístico.

El objetivo de esta comunicación es plantear correctamente la prueba y distribución t. La distribución t es un conjunto de curvas estructurada por un grupo de datos de unas muestras en particular. La contribución de esta prueba, específicamente, es para comparar dos muestras de tamaño ≤ 30. La primera presunción es formular la hipótesis nula y la hipótesis alterna, que establece que no hay diferencias en la media de las dos muestras independientes y que de existir esta diferencia, sólo se debe al azar. Si la t calculada que se origina de las dos muestras es desmesurada (valor de p que se encuentra en las tablas respectivas), entonces se rechazaría la hipótesis nula (error tipo I). Es importante mencionar que este valor depende del valor de significancia establecido con anterioridad de lo que se quiere probar, para la diferencia entre las medias de las dos muestras. Este valor de significancia es la probabilidad de rechazar erróneamente la hipótesis nula. (McGRAW-HILL, 1997).

Este método t-Student es un procedimiento estadístico que permite aceptar o rechazar una afirmación hecha con respecto a un fenómeno o suceso (WIKIVERSIDAD, 2017).

El procedimiento consta de seis (6) pasos:

El primer paso es determinar nuestra hipótesis nula, hipótesis alternativa.

Hipótesis nula, es el reverso de las hipótesis de investigación. También constituyen proposiciones acerca de la relación entre variables solamente que sirven para refutar o negar lo que afirma la hipótesis de investigación. (McGRAW-HILL, 1997).

Hipotesis alternativa, son posibilidades alternativas - ante las hipótesis de investigación y nula. (McGRAW-HILL, 1997).

Dado el concepto de hipótesis nula y alternativa, se define una hipótesis nula y si esta se rechaza demostraríamos que nuestra hipótesis principal es verdadera.

Hipótesis

H_0 =La dinámica de tecleo no genera patrones únicos para verificar la autentificación de usuarios.

H_1 = La dinámica de tecleo genera patrones únicos para verificar la autentificación de usuarios

Segundo paso nivel de significancia.

Se le conoce así al error máximo adoptado al momento de rechazar la hipótesis nula (H_o) cuando es verdadera (WIKIVERSIDAD, 2017).

Dependiendo del tipo de significación que se da al estudio, hay tres grados:

- **α = 0.01** → Demasiado significativo
- **α = 0.05** → Significativo
- **α = 0.10** → Poco significativo

Dado el concepto de t-student y teniendo n=17 se puede sacar las siguientes afirmaciones de la Distribución T de student ver figura 4.1, para rechazar la hipótesis nula necesitamos un t obtenido con un nivel de aceptación de 90% entonces de la figura 4.1 tenemos en la intersección (Metodos y formulas pareadas, 2017):

Por tanto aplicaremos la fórmula para determinar el nivel de confianza para una aceptación del 90%.

$\propto \rightarrow 1 - nivel\ de\ confianza/100$

$$\propto = 1 - \frac{90}{100} = 0{,}10$$

Tercer paso región de aceptación (WIKIVERSIDAD, 2017).

$$t_{1-\propto/_2} = 1 - \frac{10}{2} = 1 - 0{,}05 = 0{,}95$$

Con el nivel de confianza encontrado revisamos la tabla de t-student con un n=16 como se apresia en la figura 4.1.

$$t_{1-\propto/_2} = 1{,}746$$

Cuarto paso prueba de hipótesis para variables pareadas (Metodos y formulas pareadas, 2017):

Utilizaremos las formulas estadísticas: $t_0 = D\ \frac{1}{S_{D\,\bar{n}}}$

Dónde:

t_0 Tiene la probabilidad de ocurrencia en la tabla de distribución d t student con n-1 grados de libertad.

D $x_1 - x_2$ y x_1, x_2 son observaciones pareadas de la población 1y 2 respectivamente

$D = \frac{\sum D}{n}$, la diferencia entre los promedios de ambas muestras $\overline{D_{X_2}}$- $\overline{D_{X_1}}$ ver tabla 4.1.

$D = 0{,}00973841 - 0{,}00310952 = 0{,}00662889$

S_D es la desviación estándar de las diferencias entre las parejas de datos.

La desviación estándar es un índice numérico de la dispersión de un conjunto de datos o población. Mientras mayor es la desviación estándar, mayor es la dispersión de la población. La desviación estándar es un promedio de las desviaciones individuales de cada observación con respecto a la media de una distribución. Así, la desviación estándar mide el grado de dispersión o variabilidad. En primer lugar, midiendo la diferencia entre cada valor del conjunto de datos y la media del conjunto de datos. Luego, sumando todas estas diferencias individuales para dar el total de todas las diferencias. Por último, dividiendo el resultado por el número total de observaciones para llegar a un promedio de las distancias entre cada observación individual y la media. Este promedio de las distancias es la desviación estándar y de esta manera representa dispersión (McGRAW-HILL, 1997).

n es el tamaño de la muestra.

- $n = 17$
- $S_d = \sqrt{\frac{\sum(D_i - \bar{D})^2}{(n-1)}}$ = 0,027412931 3

Aplicando las formulas y los datos de la muestra se tiene el siguiente resultado para :

$t_0 = D\ \frac{1}{S_D\ \bar{n}} = \frac{0{,}00662889}{(0{,}0274129313 * 17)} = 0{,}0142244804$

$t_{1-\propto/2} = 1{,}734$=t obtenido

Y comparando el $t_0 < t_{1-\propto/2}$, $0{,}0142244804 < 1{,}734$ obtenido podemos rechazar la hipótesis nula, por lo tanto se concluye que la hipótesis planteada:" La dinámica de tecleo genera patrones únicos para verificar la autentificación de usuarios".

Distribución *T* de Student

k \ p	0,55	0,60	0,65	0,70	0,75	0,80	0,85	0,90	0,95	0,975	0,99	0,995	0,9995
1	0,158	0,325	0,510	0,727	1,000	1,38	1,96	3,078	6,314	12,71	31,8	63,7	637
2	0,142	0,289	0,445	0,617	0,816	1,06	1,39	1,886	2,920	4,30	6,96	9,92	31,6
3	0,137	0,277	0,424	0,584	0,765	0,978	1,25	1,638	2,353	3,18	4,54	5,84	12,9
4	0,134	0,271	0,414	0,569	0,741	0,941	1,19	1,533	2,132	2,78	3,75	4,60	8,61
5	0,132	0,267	0,408	0,559	0,727	0,920	1,16	1,476	2,015	2,57	3,36	4,03	6,86
6	0,131	0,265	0,404	0,553	0,718	0,906	1,13	1,440	1,943	2,45	3,14	3,71	5,96
7	0,130	0,263	0,402	0,549	0,711	0,896	1,12	1,415	1,895	2,36	3,00	3,50	5,40
8	0,130	0,262	0,399	0,546	0,706	0,889	1,11	1,397	1,860	2,31	2,90	3,36	5,04
9	0,129	0,261	0,398	0,543	0,703	0,883	1,10	1,383	1,833	2,26	2,82	3,25	4,78
10	0,129	0,260	0,397	0,542	0,700	0,879	1,09	1,372	1,812	2,23	2,76	3,17	4,59
11	0,129	0,260	0,396	0,540	0,697	0,876	1,09	1,363	1,796	2,20	2,72	3,11	4,44
12	0,128	0,259	0,395	0,539	0,695	0,873	1,08	1,356	1,782	2,18	2,68	3,06	4,32
13	0,128	0,259	0,394	0,538	0,694	0,870	1,08	1,350	1,771	2,16	2,65	3,01	4,22
14	0,128	0,258	0,393	0,537	0,692	0,868	1,08	1,341	1,761	2,14	2,62	2,98	4,14
15	0,128	0,258	0,393	0,536	0,691	0,866	1,07	1,337	1,753	2,13	2,60	2,95	4,07
16	0,128	0,258	0,392	0,535	0,690	0,865	1,07	1,333	1,746	2,12	2,58	2,92	4,02
17	0,128	0,257	0,392	0,534	0,689	0,863	1,07	1,330	1,740	2,11	2,57	2,90	3,96
18	0,127	0,257	0,392	0,534	0,688	0,862	1,07	1,328	1,734	2,10	2,55	2,88	3,92
19	0,127	0,257	0,391	0,533	0,688	0,861	1,07	1,325	1,729	2,09	2,54	2,86	3,88
20	0,127	0,257	0,391	0,533	0,687	0,860	1,06	1,323	1,725	2,09	2,53	2,84	3,85
21	0,127	0,257	0,391	0,532	0,686	0,859	1,06	1,321	1,721	2,08	2,52	2,83	3,82
22	0,127	0,256	0,390	0,532	0,686	0,858	1,06	1,319	1,717	2,07	2,51	2,82	3,79

Figura 4.2 Distribucion t-student

Fuente: Dawson , 1993

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Se verifico que los patrones almacenados por cada usuario autenticado, contribuyo al proceso de autenticación de un usuario original.
- Se verifico que aplicar la técnica de distancia punto a punto a los vectores key down y key up se obtuvo un resultado sesgado a diferencia de una combinación por separado.
- La toma de muestras de letra por letra ayudo para un análisis detallado de la dinámica de tecleo por cada usuario.
- Se definió un conjunto de caracteres a teclear por el usuario, para reducir la variedad de caracteres especiales que puede contener una contraseña.
- Se delimito el número de palabras por cada dato introducido por el usuario, para reducir el tiempo de análisis.
- La metodología propuesta en este trabajo es de bajo costo, pues usa un teclado convencional para la adquisición de las muestras biométricas y no es invasiva pues el usuario utiliza el acceso clásico a sistemas informáticos del tipo usuario/clave.
- Se realizó el algoritmo multiplataforma como se menciona el los objetivos específicos por tal motivo el algoritmo aplicado al método puede funcionar en ios y android.

5.2 RECOMENDACIONES

Se recomienda aplicar un módulo de aprendizaje continuo, es decir no solo aplicar el método a la autenticación, sino también a todos los procesos iniciados por el tecleo en el sistema, para que de esta manera se tenga mayor confiabilidad de la respuesta del método.

Así mismo, para aumentar el rendimiento del sistema es vital cuidar las condiciones de adquisición. Durante la fase de aprendizaje se observó cómo el hecho de que la toma de muestras debe ser regulada de acuerdo al estado de ánimo del usuario.

5.3 SUGERENCIAS PARA FUTURAS INVESTIGACIONES

El método propuesto, es aplicable a la vida diaria, y por tal motivo se puede extender al Internet de las cosas, redes sociales e incluso combinar esta técnica con la dinámica de comportamiento del mouse, verificando cuantos clics se realiza para ciertas operaciones.

Respecto al trabajo a futuro hay varios puntos importantes sobre los cuales enfocarse, el primero de ellos consiste en agregar al sistema un método de adaptación, es decir, que le permita al software reconocer y aprender las variaciones y evolución que va sufriendo la dinámica de tecleo del usuario a lo largo del tiempo, lo que permitiría mejorar el desempeño. En un segundo punto se trataría de modificar el software para que sea tolerante a los errores de escritura tanto en el proceso de registro como en el de autenticación.

Un último aspecto sobre el cual hay que trabajar es el portar el sistema a teléfonos celulares convencionales, los cuales no cuentan con pantallas táctiles para el ingreso de datos.

Uso de medidas de calidad. Consiste en realizar un procesamiento basado en calidad, por ejemplo ajustando el umbral de decisión en función de los parámetros de calidad seleccionados.

Descartar de forma automática muestras inválidas según la medida de calidad obtenida sería otra posibilidad. Esto último permitiría avisar al usuario de la necesidad de adquirir una nueva muestra durante el proceso de registro/test, de forma que el proceso de aprendizaje no necesitara estar supervisado. Para la selección de las medidas de calidad aplicadas a la textura del tecleo por cada dedo, un posible punto de partida podría consistir en aplicar parámetros de calidad extensamente utilizados en huella dactilar.

BIBLIOGRAFIA

Perez .J y Merino.M. (2009).Definición de autorización [Web log Post].Recuperado de https://definicion.de/autorizacion/.

blogMabz. (2010,16 de Abril).Ex`posición Paradigma Positivista [Web log Post].recuperado de https://es.slideshare.net/BlogMabz/exposicion-paradigma-positivista diapositiva 2.

Zayas.P.M.(s.f.).Biblioteca Virtual de Derecho ,economía y Ciencias. EL ROMBO DE LAS INVESTIGACIONES DE LAS CIENCIAS SOCIALES. Servicios Académicos Internacionales S.C .recuperado de http://www.eumed.net/libros-gratis/2010e/822/Paradigma%20positivista.htm

Doupovec. M. et al. (2010). Métodos de la investigación [Web Blog Post].recuperado de: http://metodologia02.blogspot.com/p/metodos-de-la-inventigacion.html.

Vincenzi M. (2014) Nuevas Tecnologías Biométricas Informáticas disponible en: http://slideplayer.es/slide/1622277/.

EcuRed,(2017,31 octubre).Ataque Informático, EcuRed recuperado de https://www.ecured.cu/Ataque_inform%C3%A1tico.

Marroquín, B.C. (2005, Abril). Tecnología Dinámica recuperado de. http://biblioteca.usac.edu.gt/tesis/08/08_0244_CS.pdf.

Anonimo.(S.f.). Sinnaps Metodoligia cuantitativa Recuperado de https://www.sinnaps.com/blog-gestion-proyectos/metodologia-cualitativa.

Alberto Blanco, M. G. (25 de 05 de 2015). *Arsys*. Obtenido de Arsys: https://www.arsys.es/blog/programacion/diseno-web/que-es-phonegap/

Anonimo. (S.f.). *Sinnaps*. Obtenido de https://www.sinnaps.com/blog-gestion-proyectos/metodologia-cualitativa

Banerjee. (2014).

EcuRed. (31 de octubre de 2017). *EcuRed*. Obtenido de https://www.ecured.cu/Ataque_inform%C3%A1tico

Edgardo. (21 de Mayo de 2010). *N-Ideas. Nuevas Ideas*. Obtenido de N-Ideas. Nuevas Ideas: http://nidea-soluciones.blogspot.com/2010/05/control-de-acceso-web-de-tecleo_21.html

Esterkin. (23 de Junio de 2009). Biometria y su Avanse. En Esterkin.

Gonsalez, M. (25 de febreo de 2013). *Distribucion Normal*. Obtenido de http://distribuc.blogspot.com/2013/02/distribucion-normal-de-probabilidad.html

Hulkko, S. (2008).

Prieto, M. D. (s.f.). *Capas de seguridad en dispocitivos moviles.* Catalunya.

sales@Aware.com. (s.f.). *Aware.* Obtenido de https://www.aware.com/es/que-es-la-biometria/aplicaciones-biometricas/