UNIVERSIDAD DE ANTIOQUIA

VERIFICACIÓN DE IDENTIDAD MEDIANTE ANÁLISIS BIOMÉTRICO BASADO EN LA DINÁMICA DEL TECLEO

Autor

Daniel Escobar Grisales

Universidad de Antioquia

Facultad de Ingeniería, Departamento de Ingeniería

Electrónica y Telecomunicaciones

Medellín, Colombia

2019

Verificación de identidad mediante análisis biométrico basado en la dinámica del tecleo

Daniel Escobar Grisales

Trabajo de grado como requisito para optar al título de:
Ingeniero Electrónico

Asesor:
PhD. Juan Rafael Orozco Arroyave

Co-Asesor:
MSc. Juan Camilo Vásquez Correa

Línea de investigación:
Procesamiento digital de señales y análisis de patrones
Grupo de Investigación en Telecomunicaciones Aplicadas
GITA

Universidad de Antioquia
Facultad de Ingeniería, Departamento de Ingeniería
Electrónica y Telecomunicaciones
Medellín, Colombia
2019.

Agradecimientos

Agradezco principalmente a mi madre Amparo Grisales, a mi padre Owsvaldo Escobar y a mi hermana Diana Escobar, en quienes siempre he encontrado el concejo y el apoyo necesario para culminar cada etapa de mi vida, a quienes debo cada uno de mis logros. A mi tía Cecilia Grisales, quien me recibió en su hogar, sin conocerme, depositando su confianza en mi y motivándome en los momentos complicados. A mi familia, que siempre me ha acompañado en este proceso y de quienes recibí enseñanzas, motivaciones y el afecto necesario, al estar lejos de casa.

Agradezco a mis compañeros Cristian Rios, Felipe Orlando López y Luis Felipe Parra, quienes, además de ayudarme con algunos aspectos técnicos de este trabajo, también estuvieron en los momentos mas complicados y alegres de toda mi vida universitaria. A mis compañeros de la línea de investigación, Paula Pérez, Tomas Arias, Surley Berrio, Sebastian Guerreo, Nicanor Garcia y Felipe Gómez, a quienes mas que colegas, considero amigos. Al grupo de investigación GITA, en donde encontré un espacio ideal, para potenciar mis capacidades.

Por último quiero agradecer a mi asesor Rafael Orozco y a mi co-asesor Juan Camilo Vásquez, de los cuales cada día aprendo cosas nuevas, que me permiten crecer como profesional y como persona. Quienes con su paciencia y concejos, permitieron el desarrollo de esta tesis.

Índice

Índice de figuras

Índice de tablas

Resumen

Los cursos virtuales, son una estrategia que busca ampliar la cobertura del sistema educativo y aunque este tipo de estrategias favorecen a la comunidad en general, también generan diversas problemáticas, que involucran el fraude en actividades evaluativas. El problema principal de este tipo de fraude es que el usuario desea ser suplantado, con el fin de obtener mejores resultados en las calificaciones del curso, por ende se hace necesario diseñar un sistema en el cual se logre detectar la identidad del usuario a través de un identificador intransferible. En este trabajo se propone un sistema de verificación de identidad, basado en la dinámica con la que teclea el usuario, lo cual permite que aunque el usuario desee ser suplantado, se logre detectar el fraude. La dinámica de tecleo, es un patrón característico de cada persona, el cual no puede ser transferido de una persona a otra, pues depende de factores neurofisiológicos propios de cada uno. La metodología general de este trabajo consiste en construir y almacenar un modelo de tecleo del usuario, cuando este se registra. Posteriormente cuando el usuario ingresa a la plataforma, el sistema extrae el modelo de tecleo y lo compara con el modelo generado en el registro. Basado en la similitud de dichos modelos, el sistema decide si se trata de un intruso, o si por el contrario se trata de un usuario válido. El modelo se probó para dos casos específicos, cuando el modelo de tecleo, en el ingreso, se genera a partir de textos iguales o similares a los textos del registro y cuando el modelo, del ingreso, se genera a partir de un texto diferente al del registro.

Los resultados mostraron un buen desempeño para verificar usuarios según su forma de teclear, lo cual motivó al desarrollo de la aplicación Web VIUT (Verificación de identidad usando tecleo), la cual es una plataforma Web que permite el registro y el ingreso de usuarios, con el fin de verificar la identidad del usuario de manera no intrusiva. Aunque este sistema esta basado en un identificador biométrico comportamental, lo que lo hace sensible a los cambios transitorios del usuario, se logran tasas de falsos positivos de 12 % y tasas de falsos negativos de hasta un 10 %, lo que lo hace un sistema con un buen desempeño, que no requiere hardware adicional y que permite que el usuario no se percate si está siendo verificado. Otro resultado de este trabajo, es una base de datos de tecleo, la cual consta de hasta 170 usuarios, con por lo menos 141780 tecleos.

1. Introducción.

La educación superior en modalidad virtual cada día toma más fuerza en Colombia. Cursar un pregrado o posgrado en línea se ha convertido en una gran oportunidad para muchas personas. En 2015, de la oferta total de programas de pregrado y posgrado de las instituciones de educación superior en Colombia, el 4.22 % corresponde a programas de educación virtual y se estima que esta cifra aumente año tras año en las principales ciudades [1]. Aunque este tipo de educación trae enormes beneficios que permiten ampliar la cobertura del sistema educativo, también trae consigo diversas problemáticas, entre las cuales se destaca el fraude en actividades evaluativas. Esta problemática es evidente en los cursos virtuales, como los que ofrece la Universidad de Antioquia. Para acceder a estos cursos cada estudiante tiene un nombre de usuario y una contraseña, lo que funciona bien para proteger la cuenta del estudiante frente al ingreso de intrusos. Pero si el estudiante es quien desea ser suplantado, esta estrategia es defectuosa, ya que el estudiante simplemente debe dar información de su usuario y contraseña, para que el impostor pueda ingresar. Por lo cual el docente no sabe con certeza si quien está presentando un examen o una actividad, realmente es el estudiante matriculado o se trata de un impostor. Se han presentado casos, por ejemplo, cursos de Inglés, en donde el estudiante no es quien presenta los exámenes. Estos son realizados por una persona con más experiencia en el tema y el estudiante brinda acceso al impostor con el fin de lograr mejores calificaciones. Para resolver este problema los proveedores de este tipo de servicios digitales están implementando estrategias para regular el fraude, buscando que ello no adicione una complejidad extra para el usuario y permita verificar si realmente el usuario que esta ingresando, es el usuario registrado. De este tipo de estrategias, las más exitosas son las basadas en biometría [2].

La mayoría de sistemas que usan un análisis biométrico para verificar la identidad de usuarios, requieren un hardware adicional, lo cual provoca que este tipo de análisis sea poco usado por los proveedores de contenido digital. Pues no pueden garantizar que el consumidor cuente con una cámara, un lector de huella u otro dispositivo que permita la verificación [3]. Sin embargo la biometría basada en la dinámica del tecleo no tiene esta dificultad, puesto que el teclado es una herramienta indispensable para los computadores, que generalmente, usa la mayoría de usuarios.

En este trabajo se busca verificar la identidad de un usuario que fue previamente registrado de acuerdo con sus patrones de tecleo. Para ello, se desarrolló una plataforma que permite al usuario registrarse con un nombre de usuario y un número de identificación. En el proceso de registro el usuario deberá realizar algunas tareas, de forma que el sistema logre definir un modelo

de su dinámica al teclear. Posteriormente, cuando un usuario intente ingresar al sistema con el número de identificación de un usuario registrado, el sistema pedirá que escriba uno o varios textos con el fin de construir su modelo de usuario basado en la dinámica al teclear. Luego compara la dinámica del usuario que esta ingresando, con la dinámica del usuario registrado, para definir si el usuario es válido. De esta forma aunque el usuario desee ser suplantado, el sistema sabrá que se trata de un intruso. Porque, aunque el usuario pueda transferir los datos de su cuenta, al impostor, el usuario no podrá transferir una característica propia de su comportamiento, como lo es la dinámica al teclear.

1.1. Estado del arte.

La idea de analizar la dinámica del tecleo surge en el siglo XX, cuando los operadores de telégrafo podían transmitir decenas de palabras por minuto y desarrollaban un ritmo distintivo [4]. Este ritmo distintivo era captado por los operarios del otro lado de la línea y de esa forma tenían una idea de que usuario estaba transmitiendo. En 1990 Joyce y Gupta desarrollaron un sistema de autenticación de usuario [5], en el que logran obtener una firma digital del usuario, basada en la latencia con la que escribe caracteres consecutivos. Para generar esta firma digital, se extrae la curva de latencias con las que escribió el nombre de usuario y la contraseña. En el caso del registro, el usuario escribe su nombre 8 veces y la contraseña 8 veces, con el fin de construir una curva promedio que caracterice al usuario. Luego, cuando un usuario intenta ingresar con el nombre y la contraseña de un usuario registrado, el sistema compara la curva promedio del usuario registrado con la curva del usuario que desea ingresar. Si al comparar estas curvas se supera cierto umbral, el sistema clasifica al usuario como intruso, en caso contrario como valido. Los autores evaluaron el sistema con datos de 30 usuarios de registro y cada usuario se evaluó con 27 impostores diferentes. Para un total de 30 usuarios registrados y 810 intentos de entrada de usuarios no validos, obteniendo una FPR (del inglés, *False Positives Rate*) del 0.25 % y una FNR (del inglés, *False Negeatives Rate*) del 16 %. Sin embargo, este sistema tiene la limitante de que el usuario solo será reconocido si escribe su nombre y contraseña de forma correcta (sin errores). Si el usuario escribe algo diferente, el sistema no podrá comparar las curvas de forma adecuada, lo cual afectara el funcionamiento del sistema.

En trabajos más recientes como en [6], los autores logran identificar programadores según el ritmo con el que teclean. En dicho trabajo tomaron estudiantes de un curso de programación en java de 7 semanas en la universidad de Helsinki. Cada perfil del estudiante consta de 4 promedios, el

promedio de tiempo con el que se oprime cualquier tecla, con el que se oprime una tecla particular, con el que se oprime dos teclas particulares y por último, la combinación de los 3 promedios anteriores. Para identificar al programador calculaban la distancia euclidiana entre la muestra de evaluación y las diferentes muestras contenidas en la base de datos. Con este enfoque obtuvieron tasas de acierto de hasta un 97.7 %, logrando identificar 169 programadores de 173 [6]. A pesar de que en [6] se busca identificar mas no verificar, la forma en que caracterizan a cada usuario es de utilidad para nuestro trabajo. Puesto que a diferencia de trabajos anteriores, en [6] logran caracterizar al programador no solo en textos fijos, sino también en textos de longitud variable.

Por último en [7], buscan verificar la identidad de los usuarios, mediante los patrones de tecleo generados al escribir una contraseña en un dispositivo móvil. Para este trabajo los autores pidieron a 94 usuarios diferentes, escribir la contraseña ".tie5Roanl" y dado que se trabajaba con un dispositivo móvil, lograron extraer datos como, presión al tocar la pantalla, coordenadas donde se tocó la pantalla, tiempo en el cual se tocó la pantalla y tiempo en el cual se levanto el dedo, luego de tocar la pantalla. Con estos datos logran obtener 155 métricas, que caracterizan la dinámica del usuario al teclear la contraseña. Posteriormente reducen la dimensión del conjunto de características, aplicando el selector de características mRMR (del inglés, minimum Redundancy Maximum Relevance), logrando identificar que las características relacionadas con las coordenadas donde se tocó la pantalla y la presión al tocarla, son las principales responsables de autenticar al usuario como legitimo. En [7], prueban diversos clasificadores, como la SVM (del inglés Support Vector Machine) y el RF (del inglés, Random Forest), para decidir si un usuario es valido o no. Obteniendo, una precisión de hasta un 97.40 %. Aunque en nuestro trabajo no se usaron dispositivos móviles, las características derivadas del tiempo en que se presiono la pantalla y el tiempo cuando se dejo de presionar, son útiles, pues en nuestro caso, los datos serian, el tiempo en que se oprimió una tecla y los tiempos donde se liberó la tecla.

1.2. Hipótesis

Los factores neurofisiológicos que hacen que las firmas sean únicas para cada persona, son los mismos que ocasionan que cada persona tenga una dinámica diferente al teclear [8], [9]. Por lo tanto es posible desarrollar un sistema capaz de verificar la identidad de un usuario mediante su dinámica de tecleo.

1.3. Objetivos

1.3.1. Objetivo general

Implementar una plataforma que permita el registro de usuarios y para cada uno de ellos esté en la capacidad de verificar su identidad a partir de un análisis biométrico basado en la dinámica del tecleo.

1.3.2. Objetivos específicos

1. Implementar una plataforma que permita el registro y la verificación de usuarios a partir de su dinámica de tecleo.

2. Implementar algoritmos de caracterización y verificación de identidad con base en las características de dinámica de tecleo.

3. Evaluar el desempeño del sistema implementado, mediante métricas de precisión y usabilidad tales como FNR, FPR, ERR (Del inglés, *Equal Error Rate*), CUA (Del inglés, *Cost to a User to Authenticate*), y CUE (Del inglés, *Cost to a User to Enroll*).

1.4. Contribución de este trabajo

En este trabajo se propone un sistema de verificación de identidad basado en la dinámica de tecleo. Dentro de las ventajas se tiene un sistema que no requiere hardware adicional, un sistema no invasivo para el usuario y en algunos casos el usuario no se percata que es verificado constantemente, lo cual aumenta la seguridad en este tipo de sistemas. Dentro de las desventajas, dado a que es un sistema basado en biometría comportamental, es un sistema sensible a los cambios en el estado de animo del usuario.

2. Marco teórico.

2.1. Biometría

La biometría se refiere al establecimiento de una identidad, basado en las características físicas y de comportamiento (también conocidas como rasgos o identificadores) de un individuo. Identificadores como cara, huella digital, geometría de la mano, iris, pulsación de tecla, firma, voz, entre otros. Generalmente los sistemas biométricos basados en características físicas son consideradas mas exitosas, comparados con los que se basan en características de comportamiento, tales como voz, tecleo y escritura (ver Figura 1). Esto debido a que las características físicas son mas estables en condiciones normales y no varían mucho con el tiempo. Mientras que el comportamiento, cambia rápidamente, pues puede verse ampliamente influenciado por situaciones transitorias, tales como el estrés o una enfermedad [10].

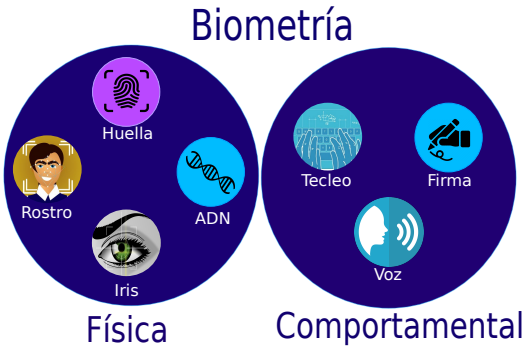


Figura 1. Biometría basada en identificadores físicos y comportamentales.

Dependiendo del contexto de la aplicación un sistema biométrico puede funcionar en modo verificación o en modo identificación [11]. En el modo verificación el sistema valida la identidad de una persona, comparando los datos biométricos del usuario que intenta acceder, con los datos del usuario que se registro. Retornando verdadero o falso dependiendo si el usuario es válido o no. En el modo identificación, el sistema compara los datos biométricos del usuario que intenta acceder con los datos de múltiples usuarios que están en la base de datos. Retornando la identificación que corresponde al modelo más similar a esos datos capturados. En este trabajo se busca verificar si el usuario es valido o no.

2.2. Dinámica de tecleo

Al escribir en el computador cada persona lo hace de diferentes formas, algunas personas teclean rápidamente y otras lentamente, algunas usan todos los dedos y otras usan solo dos o tres, para algunos las teclas inferiores representan cierta dificultad y para otros las superiores. Son muchas la características que permiten diferenciar a una persona de otra según la forma con la que teclean [12]. Un sistema recibe estas características como métricas que intentan describir el ritmo con el que la persona teclea. Entre las características mas comunes se encuentran:

1. Tiempo de vuelo: Es el tiempo entre el cual una tecla se está dejando de presionar y de manera consecutiva se presiona la siguiente, como se muestra en la Figura 2. Este tiempo generalmente oscila entre 50 a 800 ms [13].

2. Tiempo de retención: Tiempo en el cual se mantiene presionada una tecla (ver Figura 2), esta medida suele oscilar entre 60 y 140 ms [13].

3. Tecla: Es la tecla que se presionó, esta característica da información del lado del teclado que está siendo usado. Esta característica se usa para llevar a contexto los tiempos de vuelo y retención.

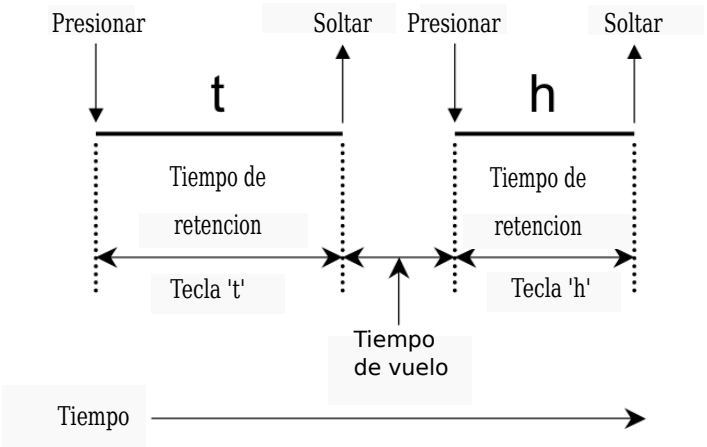


Figura 2. Relación entre el tiempo de retención y la latencia. Figura adaptado de [14]

Las características de tecleo se pueden tomar para una tecla de forma individual, como se muestra en la Figura 2, o para dos o mas teclas consecutivas. Dado el caso de tomar dos teclas consecutivas, se le llama un modelo

de texto Di-grafo, si se toman 3 se conoce como un modelo Tri-grafo y de esa forma para n teclas consecutivas se conoce como un modelo N-grafo.

La principal ventaja de este tipo de características es que no se necesita hardware adicional para obtenerlas. sin embargo existen otro tipo de características tales como, la presión en las teclas, la mano que presiona la tecla y qué dedo presionó la tecla. Estas características exhiben de mejor forma el comportamiento particular de cada usuario, sin embargo estas características implican un hardware adicional, lo cual no es lo deseado, dados los objetivos de este trabajo.

2.3. Caracterización de usuarios

La caracterización de usuarios, es un proceso en el cual se buscan diferentes métricas, que definen la dinámica de tecleo. Este proceso puede ser dividido en 2 pasos:

1. Segmentación
2. Caracterización de dinámica.

2.3.1. Segmentación.

En este trabajo para la segmentación se crearon modelos tri-grafos, es decir, se crearon pequeños paquetes con la información de tres caracteres consecutivos. Se define este modelo, dado a que, modelos similares han dado buenos resultados en los sistemas de verificación por voz, de trabajos anteriores [15]. Luego, se definió un tamaño de ventana de 5 Tri-grafos, con un solapamiento de 3 Tri-grafos (ver Figura 3). Si la ultima muestra no puede completarse con los 5 Trigrafos, esta se incluye de forma incompleta.

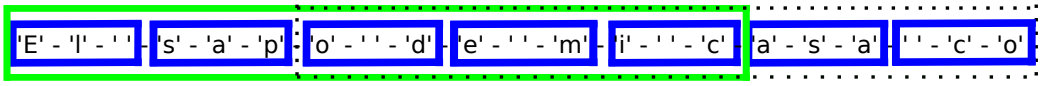


Figura 3. Ejemplo segmentación de la frase: "*El sapo de mi casa co*", tri-grafos (azul), ventana 1 (verde) y ventana 2 (linea punteada).

2.3.2. Caracterización de dinámica.

De cada caracter se extraen 6 características (ver Tabla 2), como se vera mas adelante. Tres del instante en que se presionó el carácter y tres de cuando el carácter se liberó. Por lo cual es posible extraer tres métricas por carácter, el **tiempo de retención**, el **tiempo de vuelo** y la **tecla usada**.

$$T_R = T_L - T_P. \tag{1}$$

$$T_V = T_{L_i} - T_{Pi+1}. \tag{2}$$

En la Ecuación 1, T_R es el tiempo de retención, T_L es el tiempo en el que se liberó la tecla y T_P es el tiempo en el que se presiono la tecla. Para la Ecuación 2, T_V es el tiempo de vuelo, T_{Li} es el tiempo en el que se liberó la tecla i y T_{Pi+1} es el tiempo en que se presiona la siguiente tecla.

Partiendo de las tres métricas en cada tecla, es posible generar las métricas para cada ventana de 5 trigrafos, estas metricas son:

- ✓ **Tiempo de retención total** (T_{RT}): Esta métrica se define como la suma de los tiempos de retención de los caracteres que están en la ventana. En la Ecuación 3, T_{Rk} es el tiempo de retención total de la $k - ésima$ ventana y I_k es el numero de teclas de la ventana k.

$$T_{Rk} = \sum_{i}^{I_k} T_{R_i} \tag{3}$$

- ✓ **Tiempo de retención promedio** (T_{RP}): Es el promedio de los tiempos de retención de la ventana.
- ✓ **Desviación estándar del tiempo de retención** (σ_{RP}): Es la desviación estándar de los tiempos de retención de la ventana.
- ✓ **Tecla Fuerte** (K_F): Es el código de la tecla, con menor tiempo de retención de la ventana.
- ✓ **Tiempo de tecla Fuerte** (T_{K_F}): Es el tiempo de retención mínimo, de la ventana.
- ✓ **Tecla Débil** (K_D): Es el código de la tecla, con mayor tiempo de retención de la ventana.
- ✓ **Tiempo de tecla Débil** (T_{K_D}): Es el mayor tiempo de retención de la ventana.
- ✓ **Tiempo de vuelo promedio** (T_{VP}): Es el promedio de los tiempos de vuelo de la ventana.
- ✓ **Desviación estándar del tiempo de vuelo** (σ_{VP}): Es la desviación estándar de los tiempos de vuelo de la ventana.

- ✓ **Tecla Fuerte en vuelo (K_{FV})**: Es el código de la tecla, con menor tiempo de vuelo de la ventana.
- ✓ **Tiempo de tecla Fuerte en vuelo ($T_{K_{FV}}$)**: Es el tiempo de vuelo mínimo de la ventana.
- ✓ **Tecla Débil en vuelo (K_{DV})**: Es el código de la tecla, con mayor tiempo de vuelo de la ventana.
- ✓ **Tiempo de tecla Débil en vuelo ($T_{K_{DV}}$)**: Es el mayor tiempo de vuelo de la ventana.

Cada usuario genera K ventanas, cada una con 13 métricas. Las cuales permiten formar una matriz de la forma.

$$\begin{bmatrix} T_{RT_1} & T_{RP1} & \sigma_{RP1} & K_{F1} & T_{K_{F1}} & K_{D1} & T_{K_{D1}} & T_{V_{P1}} & \sigma_{VP1} & K_{FV1} & T_{K_{FV1}} & K_{DV1} & T_{K_{DV1}} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \\ T_{RT_K} & T_{RPK} & \sigma_{RPK} & K_{FK} & T_{K_{FK}} & K_{DK} & T_{K_{DK}} & T_{V_{PK}} & \sigma_{VPK} & K_{FVK} & T_{K_{FVK}} & K_{DVK} & T_{K_{DVK}} \end{bmatrix} \quad (4)$$

A esta matriz, la llamaremos la **matriz de características**.

2.4. Modelo de Mezclas Gaussianas

Una Modelo de mezclas Gaussianas (del inglés, Gaussian Mixture Model, *GMM*) es un modelo probabilístico que busca representar una población a partir de una combinación de distribuciones de probabilidad Gaussianas [16]. En este trabajo se usa una GMM para modelar los datos de tecleo del usuario, tanto en el registro, como en el ingreso, ya que este tipo de enfoque ha sido exitoso en sistemas de verificación de identidad a partir de la voz [17]. Antes de definir formalmente el GMM, es necesario revisar algunos conceptos iniciales, sobre el comportamiento de una distribución de probabilidad Gaussiana. En el anexo A se incluyen algunos conceptos básicos de variable aleatoria continua, los cuales pueden ser útiles antes de continuar.

2.4.1. Distribución Gaussiana

Una variable aleatoria continua x está distribuida de forma normal o Gaussiana si su función de densidad de probabilidad (PDF del inglés, Density Probability Funtion) es de la forma.

$$f(x) = \frac{1}{\sigma\sqrt[2]{2\pi}} \exp\left[-\frac{(x-\mu^2)}{2\sigma^2}\right] \doteq \mathcal{N}(x;\mu,\sigma^2) \quad (5)$$

Donde μ indica la media y σ la desviación estándar de la distribución. Esta distribución es comúnmente usada en muchas disciplinas de ingeniería y ciencia, por su habilidad para aproximar muchos datos del mundo real, gracias a la ley de los grandes números [15]. En este trabajo usaremos estas distribuciones para modelar sub-conjuntos de los datos de tecleo de los usuarios, como se vera mas adelante.

En caso que la variable aleatoria sea multivariada, $\mathbf{x} = (x_1, x_2, ..., x_D)^\intercal$, se dice que tiene una distribución Gaussiana, si su PDF es de la forma:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}\,|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{6}$$

$$f(\boldsymbol{x}) \doteq N(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{7}$$

Donde $\boldsymbol{\mu} \in \mathbb{R}^D$ es el vector de medias de la distribución y se compone de las medias de cada dimensión de la variable aleatoria multivariada. Por otra parte $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$ es la matriz de covarianzas. Esta matriz de covarianzas $\boldsymbol{\Sigma}$ es el equivalente al concepto de varianza de una variable aleatoria en dimensiones superiores o multivariada y se define como:

$$\boldsymbol{\Sigma}= \begin{vmatrix} E\left[(x_1-\mu_1)(x_1-\mu_1)\right] & E\left[(x_1-\mu_1)(x_2-\mu_2)\right] & ... & E\left[(x_1-\mu_1)(x_D-\mu_D)\right] \\ E\left[(x_2-\mu_2)(x_1-\mu_1)\right] & E\left[(x_2-\mu_2)(x_2-\mu_2)\right] & ... & E\left[(x_2-\mu_2)(x_D-\mu_D)\right] \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ E\left[(x_D-\mu_D)(x_1-\mu_1)\right] & E\left[(x_D-\mu_D)(x_2-\mu_2)\right] & ... & E\left[(x_D-\mu_D)(x_D-\mu_D)\right] \end{vmatrix} \tag{8}$$

$$\boldsymbol{\Sigma} = E\left[(\boldsymbol{x}-\bar{\boldsymbol{x}})(\boldsymbol{x}-\bar{\boldsymbol{x}})^T\right] \tag{9}$$

2.4.2. GMM

Los GMMs juegan un papel importante en el modelamiento acústico de los sistemas de reconocimiento de voz convencionales [15]. En este trabajo se hace uso de el GMM para lograr modelar la dinámica de los usuarios y posteriormente medir la distancia o la similitud de el GMM del usuario al momento de registrarse, con el GMM del usuario que desea ingresar. Esto con el fin de determinar si el usuario es válido o no. El GMM permite modelar un conjunto de datos dado, a partir de la combinación lineal de un número finito de PDFs Gaussianas. Cada distribución busca modelar una sub-población, para que en conjunto la mezcla de Gaussianas modele toda la población en general. El GMM para una variable aleatoria en una dimensión (univariada) se define como:

$$f(x) = \sum_{m=1}^{M} \frac{c_m}{(2\pi)^{\frac{1}{2}}\sigma^2} \exp(-\frac{(x-\mu_m)^2}{\sigma_m}), \tag{10}$$

$$\sum_{m=1}^{M} c_m = 1. \tag{11}$$

Como puede notar, la Ecuación 10 es una suma de M distribuciones Gaussianas univariadas. El parámetro c_m se refiere al peso o a la ponderación de cada PDF. Este c_m define la influencia de la m-ésima PDF Gaussiana en la mezcla y se busca que el peso c_m sea alto, cuando la distribución m modela una sub-población que contiene mucha información de la población en general. Al final el vector de pesos debe cumplir que la suma de los m pesos debe ser igual a la unidad, como se observa en la Ecuación 11. Esto debido a que el GMM es una suma de distribuciones y a su vez es una distribución y por lo tanto debe cumplir las propiedades de estas (ver anexo A).

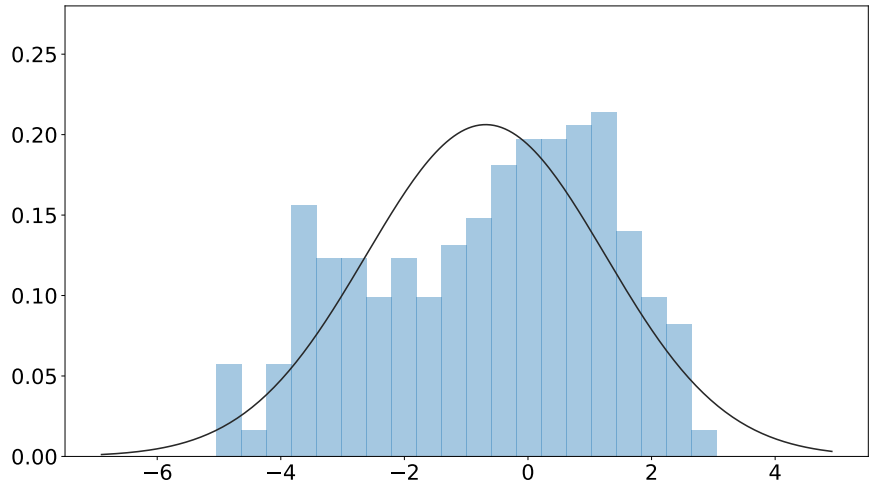


Figura 4. Modelamiento de datos con 1 Gaussiana

En la Figura 4 se muestra un conjunto de datos de una variable aleatoria y se busca modelar estos datos con una distribución Gaussiana. Como se puede ver este modelo no logra adaptarse de forma adecuada a los datos, lo que ocasiona principalmente perdida de información. Sin embargo es posible generar una mezcla con tres Gaussianas y obtener un modelo mas adecuado para los datos (ver Figura 5). Como puede ver una distribución de mezcla de Gaussianas tiene una propiedad multimodal, ($M > 1$ en la Ecuación 10) la cual le permite modelar de mejor manera diversos datos del mundo real.

Como se vio anteriormente la variable aleatoria $\boldsymbol{x}$ puede ser multivariada y de la misma forma en que fue posible generalizar la Ecuación 5 de una PDF univariada, a la Ecuación 29 de una PDF multivariada, también es

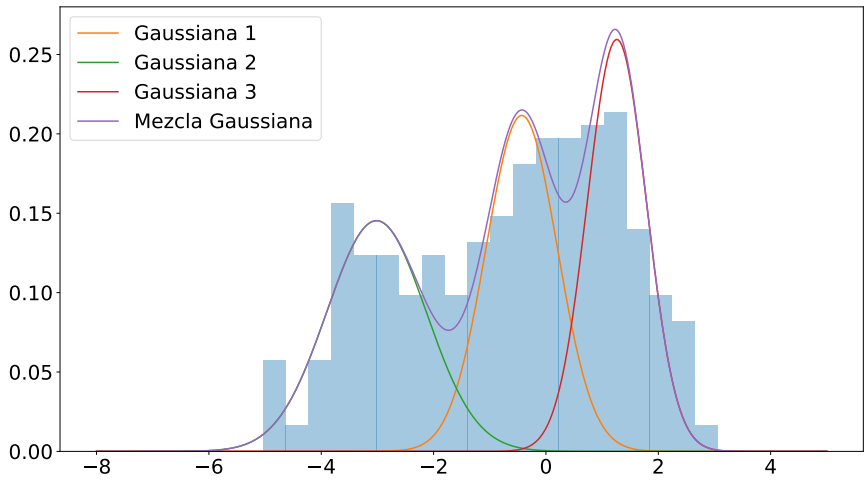


Figura 5. Modelamiento de datos con una GMM de 3 componentes.

posible generalizar la Ecuación 10 para una variable aleatoria $\boldsymbol{x}$ multivariada. Como se ve en la Ecuación 12, donde M es el número de Gaussianas y c_m corresponde a la ponderación, dentro de la mezcla, de la $m-$*ésima* distribución Gaussiana.

$$f(\boldsymbol{x}) = \sum_{m=1}^{M} \frac{c_m}{(2\pi)^{D/2} \left|\boldsymbol{\Sigma}_m\right|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_m)\right] \tag{12}$$

.

$$f(\boldsymbol{x}) = \sum_{m=1}^{M} c_m N(\boldsymbol{x}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \tag{13}$$

Existen criterios como el de información Bayesiano BIC (Del inglés, *Bayesian Information Criteria*) que ayudan a estimar el número de componentes, evaluando la cantidad de información que se pierde al usar el modelo. Pero en muchas aplicaciones el número de componentes M se escoge dependiendo de un antecedente previo o de una experimentación previa [18]. En este trabajo dado que no se conoce un antecedente, se determinará este parámetro de forma experimental.

En nuestro caso el número de componentes a usar es muy importante puesto que un número de componentes pequeño ocasionará que el modelo no se adapte correctamente a los datos de dinámica del usuario. Pero un número

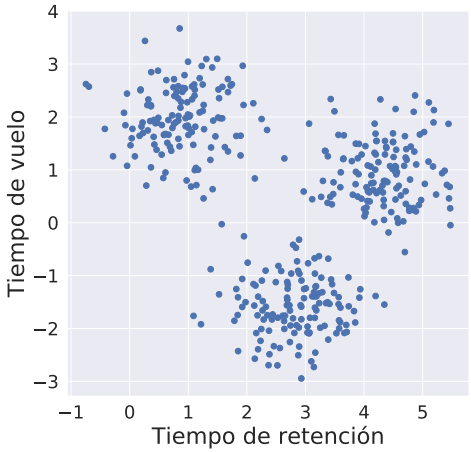
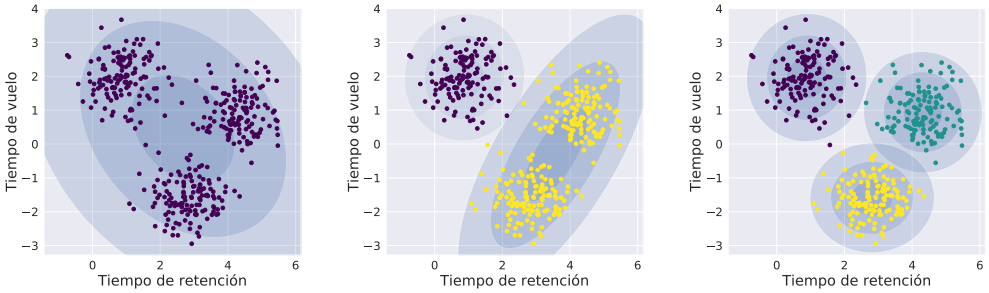


Figura 6. Representación de un conjunto de datos en dos dimensiones.

excesivo de componentes ocasionará que el modelo este sobre-ajustado. Por ejemplo, asumamos que los datos representados en la Figura 6 corresponden a los datos de un usuario en el momento de registrarse y que cada punto describe el tiempo de retención promedio (eje x) y el tiempo de vuelo promedio (eje y), con el que el usuario escribe una palabra.



(a) **(b)** **(c)**

Figura 7. (a) Modelamiento de datos con 1 Gaussiana, (b) Modelamiento de datos con 2 Gaussianas, (c) Modelamiento de datos con 3 Gaussianas.

Ahora intentemos modelar este usuario con una GMM de una sola componente Gaussiana (ver Figura 7a), como puede ver el modelo no se adapta correctamente a los datos del usuario, lo cual ocasionará problemas cuando este se compare con los modelos del ingreso, como se verá más adelante. Lo

mismo ocurre cuando se intenta modelar con dos componentes (ver Figura 7b), a pesar de que la distribución de la parte izquierda modela bien una de las subpoblaciones, la distribución de la derecha no logra adaptarse correctamente a los datos, lo cual ocasionará los mismos problemas del caso anterior. Para el caso donde se usan tres Gaussianas (ver Figura 7c) el modelo se adapta correctamente al modelo de usuario, lo cual ayudará a mejorar el desempeño del sistema. Por último en la figura Figura 8 se muestra el resultado de mezclar las distribuciones, obteniendo un GMM que modela al usuario en el registro.

En este ejemplo se usó una variable aleatoria multivariada de dos dimensiones, para lograr una representación gráfica del modelo. Sin embargo, en este trabajo el modelo se genera a partir de una variable aleatoria multivariada de 13 dimensiones, como se verá más adelante.

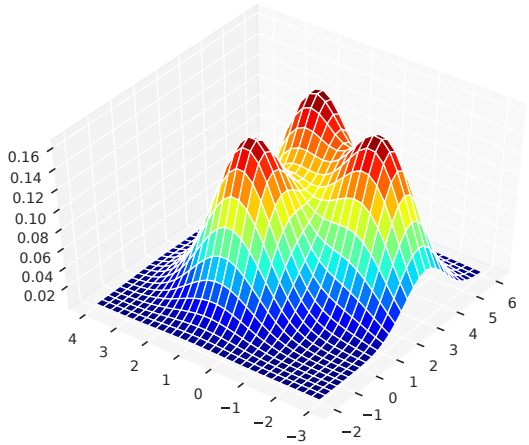


Figura 8. GMM con 3 Gaussianas

2.4.3. Estimación de parámetros

Como se vio anteriormente a una distribución Gaussiana de una variable aleatoria multivariada, se le asocia una media $\boldsymbol{\mu}$ y una matriz de covarianzas $\boldsymbol{\Sigma}$. Las cuales construyen el conjunto de parámetros de la distribución. En una distribución de mezclas Gaussianas, adicionalmente, se tiene el parámetro C_m, por ende el conjunto de parámetros es $\boldsymbol{\Theta} = \{c_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}$, para cada componente Gaussiana m, perteneciente a la mezcla.

Los parámetros $\boldsymbol{\mu}_m$ y $\boldsymbol{\Sigma}_m$ definen que sub-población es modelada por la distribución, el parámetro c_m define la contribución de la distribución en el modelo general. En principio estos parámetros son desconocidos y generalmente se estiman mediante el uso del algoritmo de maximización de esperanza

EM (del ingles, *Estimation Maximization*)[19]. Al problema de estimar estos parámetros se le conoce como aprendizaje.

Algoritmo de maximización de esperanza

El algoritmo EM es una de las técnicas más usada para estimar los parámetros de una mezcla cuando se tiene un numero fijo de Gaussianas. Este itera, alternando entre una etapa de esperanza o etapa E y una etapa de maximización o etapa M [20].

Para la primera iteración el conjunto de parámetros $\boldsymbol{\Theta}$ puede ser inicializado de forma aleatoria, o mediante un método de agrupamiento como el K-medias. Esto debido a que en la etapa E, es necesario tener un conjunto de parámetros iniciales [20].

En la etapa E se busca encontrar $h_m^{(j)}(t)$, la cual es la probabilidad de que un dato t pertenezca a una Gaussiana m en la iteracion j. Esta probabilidad se define como

$$h_m^{(j)}(t) = \frac{c_m^{(j)} \mathcal{N}(\boldsymbol{x}^{(t)}; \boldsymbol{\mu}_m^{(j)}, \boldsymbol{\Sigma}_m^{(j)})}{\sum_{i=1}^{M} c_i^{(j)} \mathcal{N}(\boldsymbol{x}^{(t)}; \boldsymbol{\mu}_i^{(j)}, \boldsymbol{\Sigma}_i^{(j)})}, \tag{14}$$

en donde, $\mathcal{N}(\boldsymbol{x}^{(t)}; \boldsymbol{\mu}_m^{(j)}, \boldsymbol{\Sigma}_m^{(j)})$ es la distribución de probabilidad de la Gaussiana m, evaluada en el dato t, en la iteracion j y $c_m^{(j)}$ es el peso de la distribución m, en la iteración j. Luego, de encontrar $h_m^{(j)}(t)$ se continua con la etapa de maximización o etapa M.

En la etapa M se busca encontrar un nuevo conjunto Θ, tal que maximice la probabilidad anterior. Para ello se usan las Ecuaciones:

$$c_m^{(j+1)} = \frac{1}{N} \sum_{t=1}^{N} h_m^{(j)}(t), \tag{15}$$

$$\boldsymbol{\mu}_m^{(j+1)} = \frac{\sum_{t=1}^{N} h_m^{(j)}(t) \boldsymbol{x}^{(t)}}{\sum_{t=1}^{N} h_m^{(j)}(t)}, \tag{16}$$

$$\boldsymbol{\Sigma}_m^{(j+1)} = \frac{\sum_{t=1}^{N} h_m^{(j)}(t) [\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_m^{(t)}][\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_m^{(t)}]^\intercal}{\sum_{t=1}^{N} h_m^{(j)}(t)}. \tag{17}$$

La Ecuación 15 determinará el peso de la distribución m para la siguiente iteración. Como puede ver el peso dependerá de la cantidad de datos que tengan una alta probabilidad de pertenecer a la $m-$ésima distribución. Aunque cada dato aporta para aumentar el peso, dicho aumento será proporcional a la probabilidad de que el dato sea modelado por la distribución m. Las Ecuaciones 16 y 17 determinan el $\boldsymbol{\mu}_m$ y $\boldsymbol{\Sigma}_m$ de la próxima iteración. En cada

iteración se busca que cada Gaussiana modele de mejor forma los datos en donde $h_m^{(j)}(t)$ presenta un valor más alto.

Finalmente el algoritmo itera entre la etapa E y la etapa M hasta que la diferencia entre $h_m^{(j+1)}(t) - h_m^{(j)}(t)$ de cada distribución, no supere cierto umbral de convergencia.

En la Figura 9 se muestra un ejemplo de cómo funciona el algoritmo EM, con una variable aleatoria multivariada de dos dimensiones ($D = 2$), para una GMM de 2 componentes ($M = 2$).

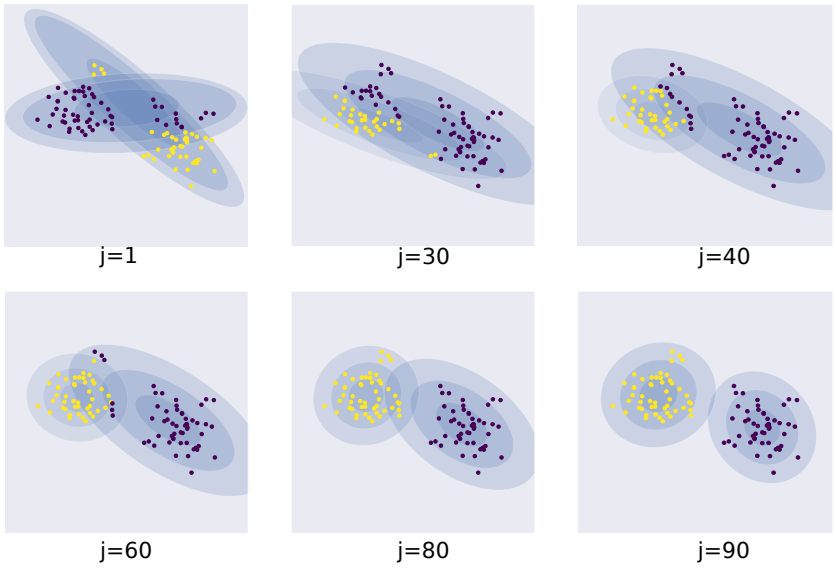


Figura 9. Algoritmo EM con $M = 2$, (j corresponde a la iteración del algoritmo)

2.5. Distancia de Bhattacharyya

La distancia de Bhattacharyya (D_{Bha}) es una métrica propuesta por el estadístico indio Anil Kumar Bhattacharyya, en 1930 [21]. esta distancia mide la similitud de dos distribuciones de probabilidad y se considera más confiable que la distancia de Mahalanobis, ya que este es un caso particular de la distancia de Bhattacharyya cuando las varianzas de las dos distribuciones son iguales [22]. Asumiendo dos distribuciones de probabilidad $f(x)$ y $g(x)$ la D_{Bha} es de la forma

$$D_{Bha}(f(x), g(x)) = -\ln(Bha_C(f(x), g(x))), \tag{18}$$

donde Bha_C se conoce como el coeficiente de Bhattacharyya y se define como

$$Bha_C(f(x), g(x)) = \int \sqrt{f(x)g(x)}. \tag{19}$$

Para este trabajo se usará la D_{Bha} para calcular la similitud de un GMM generado en el registro y un GMM generado en el ingreso. En este caso la D_{Bha} se calcula como.

$$D_{Bha} = -\ln\left[\int_{R^D} \sqrt{\sum_{m=1}^{M} f_i(\boldsymbol{x})}\sqrt{\sum_{m=1}^{M} g_i(\boldsymbol{x})}dx\right], \tag{20}$$

donde $f(\boldsymbol{x})$, es el GMM generado en el registro y $g(\boldsymbol{x})$, es el GMM generado en el ingreso, esto se explicará con más detalle más adelante.

La Ecuación 20, también puede ser escrita como la suma de una media μ_{Bha} y una matriz de covarianzas Σ_{Bha} (ver Ecuación 21), con el fin de encontrar D_{Bha} a partir del conjunto de parámetros $\boldsymbol{\Theta}$ de cada una de las GMMs [23].

$$D_{Bha} = \mu_{Bha} + \Sigma_{Bha}. \tag{21}$$

El primer termino de la Ecuación 21, es la media estadística y se puede calcular a partir de la Ecuación 22. En donde μ_{f_i} y Σ_{f_i}, son la media y la matriz de covarianza de la i-ésima distribución Gaussiana del GMM $f_i(\boldsymbol{x})$. De igual forma μ_{g_i} y Σ_{g_i}, son la media y la matriz de covarianza de la i-ésima distribución de el GMM $g_i(\boldsymbol{x})$.

$$\mu_{Bha} = \frac{1}{8}\sum_{i=1}^{M}\left\{(\mu_{f_i} - \mu_{g_i})^{\intercal})\left[\frac{\Sigma_{f_i} + \Sigma_{g_i}}{2}\right]^{-1}(\mu_{g_i} - \mu_{f_i})\right\}. \tag{22}$$

El segundo término es la covarianza estadística y está definida como

$$\Sigma_{Bha} = \frac{1}{2}\sum_{i=1}^{M}\left[\ln\frac{\frac{\Sigma_{f_i}+\Sigma_{g_i}}{2}}{\sqrt{|\Sigma_{f_i}|\,|\Sigma_{g_i}|}}\right] - \omega_{Bha}, \tag{23}$$

en donde, ω_{Bha} es

$$C_{Bha} = \frac{1}{2}\sum_{i=1}^{M}\ln(C_{f_i}C_{g_i}). \tag{24}$$

En la Ecuación 24, C_{f_i} es el peso de la i-ésima distribución del GMM $f_i(\boldsymbol{x})$ y C_{g_i} es el peso de la i-ésima distribución del GMM $g_i(\boldsymbol{x})$. La distancia D_{Bha} permite comparar los GMMs generados en el registro y en el ingreso.

2.6. Metricas de desempeño.

En el campo de la biometría, existen diferentes medidas que permiten evaluar el desempeño del sistema, estas medidas se pueden dividir en dos grupos, métricas de precisión y métricas de usabilidad [14].

2.6.1. Métricas de precisión.

Estas métricas se encargan de describir el desempeño del verificador. Típicamente se usan tres [24].

- ✓ **Tasa de falsos negativos** (en inglés false negative rate, FNR), es el porcentaje de usuarios válidos clasificados como impostores. También se le conoce como tasa de falsos rechazos (en inglés false rejection rate, FRR.
- ✓ **Tasa de falsos positivos** (en inglés false positive rate, FPR), es el porcentaje de impostores identificados como usuarios válidos. También se le conoce como tasa de aceptación falsa (en inglés false positive rate, FAR.
- ✓ **Índice de error** (en inglés equal error rate, ERR) Es el punto de cruce entre el FNR y el FPR.

2.6.2. Métricas usabilidad.

Estas métricas deciden cuánto trabajo debe hacer una persona para usar el sistema de forma exitosa. Para el caso particular de la biometría basada en la dinámica del tecleo, en [25] se mencionan dos métricas:

- ✓ **Costo para inscribir un usuario al sistema** (en inglés cost to a user to enroll, CUE), esta métrica mide el número de tecleos que debe hacer el usuario antes de registrarse como válido.
- ✓ **Costo para autenticar un usuario** (en inglés cost to a user to authenticate, CUA), esta métrica mide el número de tecleos que debe hacer el usuario para que el sistema verifique la identidad del usuario que está tecleando.

3. Base de datos

Para este trabajo fue necesario crear una aplicación Web que supliera dos funciones principales. Servir como plataforma de captura y como plataforma de verificación. La plataforma de captura se desarrolló con la finalidad principal de construir la base de datos. La plataforma de verificación se diseñó con el fin de implementar, el sistema desarrollado en este trabajo.

3.1. Aplicación Web

La aplicación Web está basada en el Framework Django 2.1.5. Para el diseño gráfico de la página se usa el lenguaje CSS, para la extracción de datos de tecleo se uso, Javascript y el sistema de gestión de la base de datos esta montado en MariaDB.

Plataforma de captura

Esta plataforma se diseñó con el fin de recolectar los datos de dinámica de tecleo de múltiples miembros de la comunidad universitaria. Por ende se diseñó de forma que fuera de fácil manejo y brindara la posibilidad que los diferentes usuarios pudieran realizar varias pruebas de forma independiente y en cualquier lugar donde tuvieran acceso a Internet. En la Figura 10 se muestra el esquema de la plataforma de captura.

Figura 10. Esquema de la plataforma de captura

La plataforma de captura se compone de dos fases principales, la fase de registro y la fase de captura. En la fase de registro la aplicación pide al usuario sus datos personales, datos como, nombre, número de identificación, edad, género, programa al que pertenece y nivel de escolaridad. Antes de permitir el registro, el usuario debe aceptar los términos y condiciones que permiten el uso de sus datos, para actividades académicas. Finalmente, en la fase de captura, el usuario debe realizar una serie de tareas en las cuales debe hacer uso del teclado. Estas tareas se verán con más detalle más adelante.

Plataforma de verificación

Esta plataforma se diseñó con la finalidad de que el usuario pudiera ingresar y escribir un texto. Basado en dicho texto, la plataforma verifica si quien estaba tecleando era un usuario válido o si por el contrario se trataba de un impostor. En la Figura 11 se muestra el esquema de la plataforma de verificación.

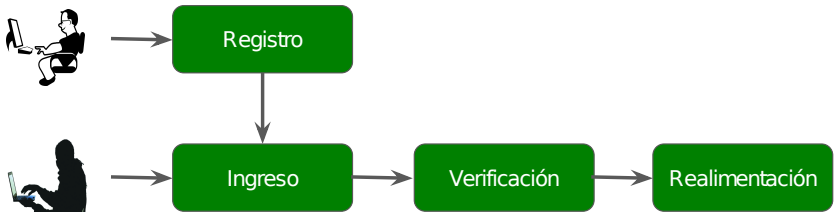


Figura 11. Esquema de la plataforma de verificación

Esta plataforma de verificación se compone de 4 fases, fase de registro, de ingreso, de verificación y de realimentación. La fase de registro, de esta plataforma, es idéntica a la de la plataforma de captura. Esta fase permite registrar un usuario, si este no se encuentra previamente registrado. Cuando el usuario se registra la plataforma pide que realice una serie de tareas que permiten generar su modelo de dinámica de tecleo. Esta fase es opcional, dependerá si el usuario ya tiene un modelo de tecleo generado, en un registro anterior. En la fase de ingreso el usuario previamente registrado, intenta acceder a su cuenta, la cual es identificada por su nombre y su número de identificación. Cuando el usuario ingresa, la plataforma pide que escriba un texto con el fin de verificar si el usuario que ingresó realmente es el usuario registrado. Note que hasta este momento la plataforma no tiene ningún tipo de seguridad, cualquier persona que desee ingresar a una cuenta particular solo necesita conocer los datos del usuario de dicha cuenta (caso similar al fraude en los cursos virtuales). En la fase de verificación se genera el modelo del usuario que acaba de ingresar y se compara con el modelo del usuario registrado con ese número de identificación. Finalmente en la fase de realimentación, la plataforma indica, basado en la comparación de los modelos, si el usuario que ingreso es realmente el usuario que se registró, o si por el contrario se trata de un impostor.

3.2. Descripción de la base de datos

Para desarrollar y evaluar el sistema de verificación, fue necesario recolectar datos de diferentes miembros de la comunidad universitaria, mediante la

plataforma de captura previamente descrita. Se solicitó a los usuarios, ingresar a la pagina y diligenciar la información de registro, para posteriormente realizar una serie de pruebas que requieren el uso del teclado. Las pruebas se componen de 5 tareas diferentes. De la tarea 1 a la 4, el usuario transcribe una frase corta, que no sobrepasa los 150 caracteres. En la tarea 5, el usuario escribe un pequeño párrafo de un texto. La Tabla 1 describe las diferentes tareas.

Tabla 1. Descripción de las tareas del proceso de captura

Tareas	**Descripción de la tarea**
1	El sapo de mi casa come kiwi, queso, zapallo y xoubas.
2	En un pueblo un niño juega afuera y tu vejez es notable.
3	La leña está partida, la tijera se ha roto, yo quiero jugar y reír, dale a la,gata sus gatitos y las fresas y las patatas del huerto.
4	La vaca flaca, las lañas malvas, las jacas blancas, a la sal acabas la salsa, zancada flaca.
5	Fragmento extraído de Frankestein o el eterno Prometeo (Mary Shelley)

Tabla 2. Ejemplo de datos retornados por la plataforma de captura. p: presión, r: liberación

Tecla	**Operación**	**Tiempo(s)**
72	p	3301
111	p	3524
72	r	3556
111	r	3612
108	p	3644
108	r	3692
97	p	3716
97	r	3820

En las tareas 1, 2, 3 y 4, se buscó que las frases permitieran que el usuario recorriera las diferentes partes del teclado. Por ejemplo, en la tarea 1, se busca definir la dinámica del usuario en desplazamientos horizontales largos. Note que, cada palabra de esta tarea obliga a conectar caracteres de lados opuestos del teclado. En la tarea 2, se busca definir la dinámica de tecleo en desplazamientos horizontales cortos. Por ende, las palabras de esta frase, pretenden conectar caracteres de la mitad del teclado, con caracteres de los extremos, para que así los desplazamientos sean más cortos, comparados con los desplazamientos de la tarea anterior. Las tareas 3 y 4, definen la dinámica

de los desplazamientos verticales. La tarea 3 fomenta los desplazamientos entre la fila del medio y la fila superior y la tarea 4 entre la fila del medio y la fila inferior. La tarea 5 es más extensa (aproximadamente 500 caracteres), busca definir la dinámica del usuario cuando se esta escribiendo un texto de forma continúa.

En cada tarea el sistema toma 3 datos por caracter, el ASCII del caracter, la operación, es decir, si se esta presionando o liberando la tecla y el tiempo del computador, en el cual se presionó o se liberó la tecla. En la Tabla 2, se muestra un ejemplo del archivo generado al escribir la palabra *Hola*. En la columna *operación* aparece una `p` si el usuario presionó la tecla o una `r` si el usuario liberó la tecla.

Tabla 3. Información de los participantes que forman la base de datos

	Hombres	**Mujeres**
Número de personas	116	54
Edad ($\mu \pm \sigma$)	23.8 ± 5.8	24.4 ± 7.1
Estudiantes de Pregrado	102	44
Profesionales	6	7
Magísters	4	-
Doctores	4	3

Al final se obtuvo una base de datos de 170 usuarios que cumplieron con las primeras 5 tareas. La información de los usuarios, se muestra en la Tabla 3. Adicional a estos datos, para uno de los experimentos, fue necesario que algunos usuarios generaran una entrada adicional, repitiendo las tareas 1, 2, 3 y 4. En total solo se logro que 20 usuarios repitieran las cuatro tareas. El motivo de esta segunda muestra, se explicará con más detalle en el experimento 1.

4. Metodología.

La metodología general se muestra en la Figura 12. Primero el usuario debe registrarse para que el sistema genere un modelo de su dinámica al teclear. El modelo se almacena en la base de datos de *Modelos de usuario*. Posteriormente cuando el usuario desea ingresar, el sistema extrae el modelo del ingreso y compara dicho modelo con el modelo del usuario en el registro. Esta comparación se realiza mediante el cálculo de la distancia de Bhattacharrya, entre los dos GMMs (el del registro y el del ingreso). Para decidir si el usuario es válido o no, el sistema compara la distancia, con un nivel de umbral, que definiremos mas adelante. Si la distancia entre los modelos supera el nivel de umbral, quiere decir que los modelos están muy alejados entre sí y por ende se trata de un impostor. Si por el contrario la distancia entre los modelos es menor al nivel de umbral, quiere decir que los modelos son similares entre si y por ende es posible que el usuario del ingreso sea el usuario registrado.

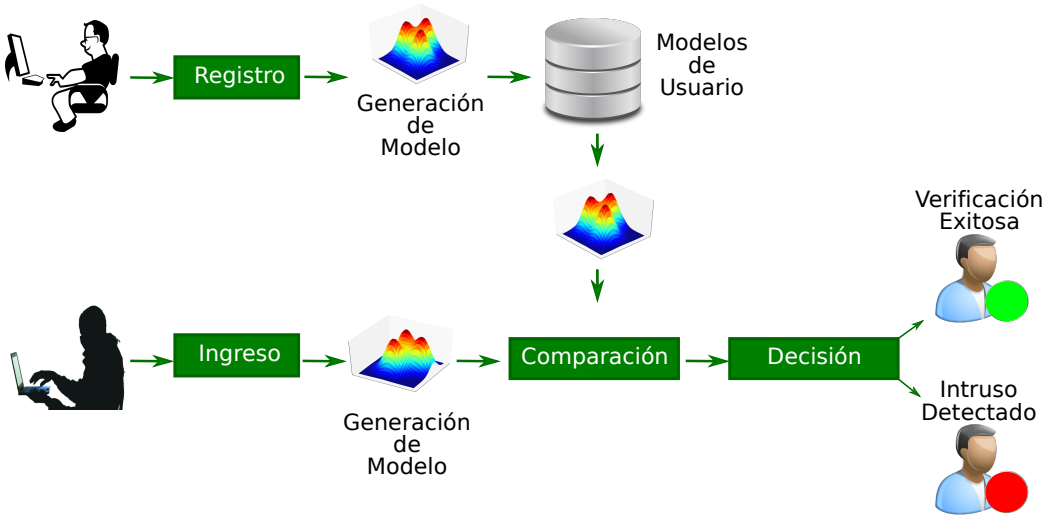


Figura 12. Metodología General

La matriz de características definida en la Sección 2.3, es usada para generar un GMM que modele la dinámica de tecleo del usuario. Ahora es necesario definir la estrategia de inicialización del algoritmo EM y el número de componentes de los GMMs. Para este trabajo el algoritmo EM se inicializó mediante el método de agrupamiento K-medias, con el fin de que el modelo converja rápidamente. Para la elección del número de componentes M es necesario definir una etapa a la que llamaremos etapa de desarrollo, en la cual se define no solo el número de Gaussianas de los GMMs, sino también

el umbral de decisión U.

4.1. Experimentos

4.1.1. Etapa de desarrollo

El sistema de verificación descrito hasta este momento depende de dos parámetros claves, M y U, los cuales deben ser calculados de forma experimental. Por ende para esta etapa fue necesario usar 100 usuarios de la base de datos.

Antes de continuar, es necesario aclarar que usaremos un número de Gaussianas fijo para todos los usuarios. Es decir, cada usuario se generara con un GMM de M componentes. Este M no debe ser muy bajo porque, como vimos en la Figura 7a, si se tiene un numero bajo de Gaussianas el modelo no se ajusta correctamente a los datos. Por el contrario, si se toma un número alto de Gaussianas, se corre el riesgo de que el modelo se sobre-ajuste a esos datos. Por ende al comparar los modelos de registro y de ingreso, las distancias de los intrusos y de los usuarios válidos serán muy próximas, lo que entorpece el funcionamiento del sistema.

Para determinar M se construyo una curva que permite visualizar el comportamiento del EER, al variar M. Esta curva se obtiene mediante el siguiente algoritmo:

1. Definir el intervalo en el que varía M.
2. Iniciar M en el intervalo definido.
3. Generar los modelos de registro y de ingreso de los 100 usuarios, mediante un conjunto de datos (tareas de cada usuario) y un GMM con M componentes.
4. Calcular la distancia de los GMMs del registro y los GMMs del ingreso.
5. Variar U hasta encontrar el valor que minimice el EER.
6. Almacenar el EER generado con los parámetros U y M.
7. Actualizar el valor del parámetro M, según el intervalo establecido.
8. Repetir los pasos 3,4,5,6 y 7 hasta completar el intervalo de variación para M.

El intervalo de variación se definió entre 1 y 50, debido a que el número mínimo de componentes es 1 y la tarea más pequeña genera aproximadamente 50 ventanas. Por lo cual el número máximo de componentes, con el

que se puede modelar una matriz de características, proveniente de una tarea pequeña, es 50. Para el caso del registro, el modelo fue generado a partir de las tareas 1, 2, 3 y 4, de cada usuario. Esto, dado que, un modelo generado a partir de estas tareas, posee datos de desplazamientos cortos, largos, horizontales y verticales, como se explicó en la Sección 3.2. Además en estas tareas se involucran todas las letras del abecedario, lo cual ayuda a definir de forma más precisa, el modelo del usuario. Para el caso del ingreso, se usa la tarea 5, ya que las tareas 1, 2, 3 y 4 se usaron para el registro. Dado a que esta tarea es más extensa comparada con las demás (500 caracteres aproximadamente), esta fue dividida en 5 segmentos de igual longitud. De esta forma para cada usuario se tendrá 495 (5×99) intentos de acceso de intrusos y 5 intentos de acceso de usuarios válidos.

El parámetro U se establece entre 0 y 1, con pasos de 0.001, dado a que las distancias fueron previamente normalizadas, dividiendo cada distancia entre la distancia máxima, de cada iteración.

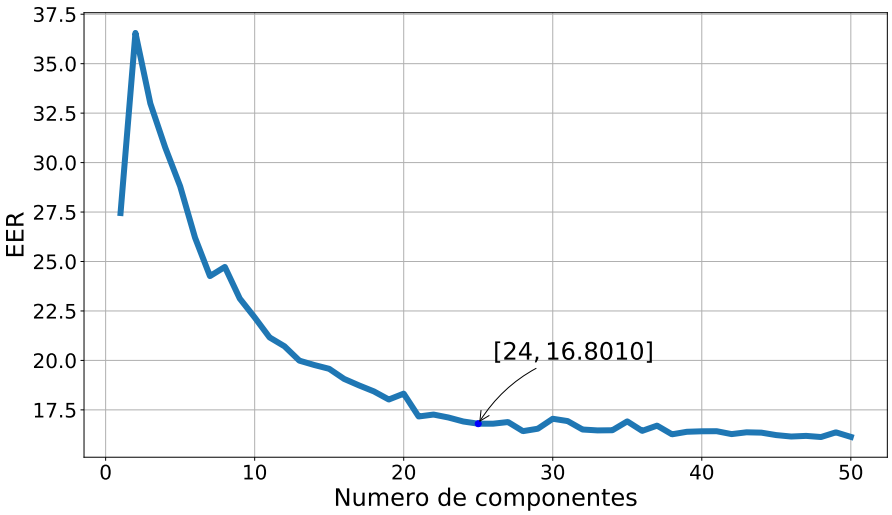


Figura 13. EER vs Numero de componentes

En la Figura 13 se muestra la curva resultante de la etapa de desarrollo descrita. Note que a medida que, aumenta el número de Gaussianas en los GMM, el EER disminuye, pero a su vez, a medida que aumenta el número de componentes, el EER decrece mas lentamente. Por ende se toma como el mejor parámetro M, aquel a partir del cual el EER no decrece considerablemente. Es decir el codo se saturación de la curva. Porque nos brindará un modelo adaptado correctamente y evitará el sobre-ajuste del modelo. En este

caso se puede ver que el codo de saturación, se encuentra cuando el número de componentes es 24 con un EER de 16.8010.

En la Figura 14a, se muestra la variación del EER al variar el parámetro U para un $M = 24$. Para este modelo, el mejor U es el 0.059, porque es aquel que permite el menor FPR con el menor FNR. En la Figura 14b, se muestra la curva ROC, al generar los modelos con un $M = 24$. El área bajo la curva ROC es de 0.8336.

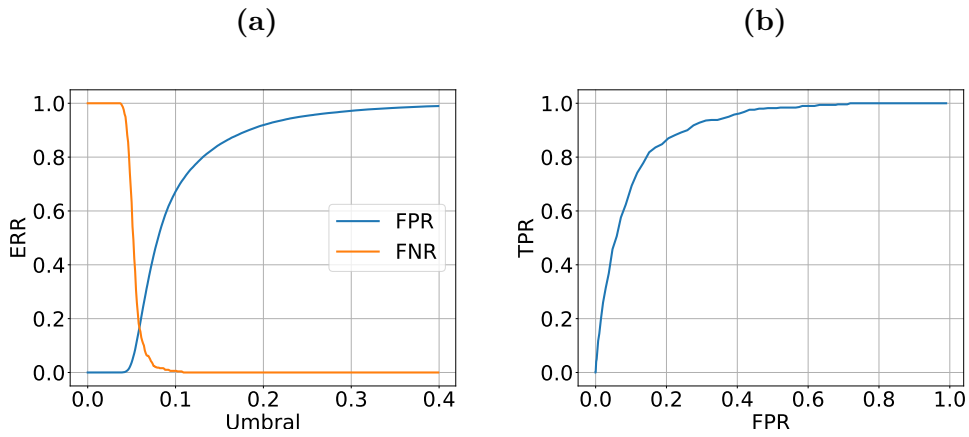


Figura 14. (a) EER Vs Umbral, (b) Curva ROC.

En conclusión, al final de esta etapa logramos obtener un conjunto de parámetros M y U. Los cuales permiten obtener un EER de 16.8010, al generar el modelo de registro con las tareas 1, 2, 3 y 4, y el modelo de ingreso con los segmentos de la tarea 5.

Una característica de esta etapa es que se validaron varios intentos de intruso y varios intentos de usuarios válidos. Lo que le da generalidad al cálculo de los parámetros M y U. Como se vio anteriormente el número de intentos de acceso de intrusos es de 49500 (495×100) y el número de intentos de accesos de usuarios validos es de 500 (5×100).

Con los parámetros M y U, es posible comenzar la etapa de evaluación. En esta etapa se busca generar un conjunto de experimentos que permitan evaluar el desempeño del sistema.

4.1.2. Etapa de evaluación

Para esta etapa se tomaron los 70 usuarios restantes de la base de datos y se generó el modelo de registro basado en las tareas 1, 2, 3 y 4, igual que en la etapa de desarrollo. En cada experimento cambian, los datos con los que se genera el modelo de ingreso o los criterios para decidir si el usuario es un

impostor o un usuario valido, los modelos de registro siempre son generados de la misma forma que en la etapa de desarrollo.

Experimento 1

Como se vio en la Sección 1.1, una de las modalidades de los sistemas de verificación basados en tecleo, es la verificación de usuarios mediante una palabra clave. Lo cual es útil para aplicaciones, donde el usuario sabe que esta siendo verificado. Por lo cual, en este experimento se buscó evaluar el desempeño del sistema de verificación, cuando el usuario ingresa una tarea conocida por el registro. Es decir, en este experimento el modelo del usuario de ingreso, se generó a partir de alguna de las tareas usadas para el registro (1, 2, 3 o 4). Para ello es necesario, tener una entrada diferente del usuario, porque de otro modo el experimento no tendría validez. Se tomaron 20 usuarios de la base de datos y se les pidió que repitieran por segunda vez las tareas 1, 2, 3 y 4, como se explico en la Sección 3.2. Posteriormente se evalúa el desempeño del sistema, cuando el modelo de usuario se genera a partir de cada una de las tareas. Este experimento genera 19 intentos de acceso de intrusos y 1 intento de acceso valido, para cada uno de los usuarios. Por tanto se tienen 380 intentos de acceso de intrusos y 20 intentos de acceso de usuarios válidos.

Experimento 2

Otra de las modalidades de los sistemas de verificación basados en tecleo, es la verificación de usuarios de forma no intrusiva y continua. Es decir, el usuario no sabe que esta siendo verificado constantemente. Pensando en esta modalidad, en este experimento, se buscó evaluar el desempeño del sistema, cuando la tarea con la que ingresa el usuario es una tarea desconocida por el modelo de registro. Para este caso no se hace necesario tener una entrada adicional del usuario para simular el ingreso, porque se genera el modelo del ingreso mediante la tarea 5. De igual forma que en la etapa de desarrollo, esta tarea 5 es dividida en 5 partes. Esto se hace con dos objetivos, el primero buscar que el sistema siempre evalué con pequeñas porciones de texto (dado a que la tarea 5 consta de alrededor de 500 caracteres) y el segundo, al dividir esta tarea, permite por cada usuario tener 5 intentos de acceso validos. Dicho esto, para este experimento se tienen 345 (5×69) intentos de acceso de intrusos y 5 intentos de acceso válidos, para cada usuario. Por tanto se tienen, 24150 intentos de acceso de intrusos y 350 intentos de acceso de usuarios válidos.

Experimento 3

En los experimentos anteriores, se mide la distancia entre el modelo de registro y el modelo de usuario. Dicha distancia es comparada con U y dependiendo de esta comparación, el sistema decide si el usuario es válido o no. El anterior es el algoritmo de decisión que tiene el sistema. En este experimento se evaluó el comportamiento del sistema cuando en lugar de decidir basado en una sola distancia, se decide a partir de un promedio de distancias.

Se dividió la tarea 5 en 5 segmentos (de igual forma que los casos anteriores). Luego, para cada segmento se genera un modelo, pero los 5 modelos pertenecen a un mismo ingreso. Se calcula la distancia entre cada modelo y el modelo de registro, obteniendo 5 distancias entre el modelo de registro y el modelo de ingreso (una por segmento del párrafo). Luego, la distancia final entre el modelo de registro y el modelo de ingreso se define como el promedio de las 5 distancias. Al final, se decide si el usuario es válido o no, comparando la distancia promedio, con el parámetro U. Para este caso solo se tendrían 69 intentos de acceso de intrusos y 1 intento de acceso válido por usuario. Lo que daría en total 4830 intentos de acceso de intrusos y 70 intentos de acceso de usuarios válidos.

5. Resultados

Los siguientes resultados muestran las métricas de desempeño (FNR y FPR) y las métricas de usabilidad (CUE y CUA). Las métricas de desempeño se dan en porcentaje y las métricas de usabilidad se dan en número de caracteres necesarios para generar el modelo de registro, en el caso CUE y el modelo de ingreso en el caso del CUA.

5.1. Experimento 1

Los resultados del experimento 1, se muestran en la Tabla 4. Como se puede ver, con la tarea 3, se obtiene un equilibrio entre el FPR y el FNR. Pero dependiendo la aplicación podría ser mas importante garantizar un FPR bajo, sacrificando el FNR, Para dichos casos, los modelos de ingreso basados en la tarea 4, podrían brindar un mejor desempeño del sistema. También se puede ver que el CUA de estas dos tareas oscila entre 90 y 140 caracteres.

Tabla 4. Métricas de desempeño y usabilidad al generar los modelos de ingreso con tareas conocidas por el modelo de registro (experimento 1)

Tareas de Ingreso	FPR	FNR	CUE	CUA
Tarea 1	18.68	10.00	314	54
Tarea 2	12.63	40.00	314	56
Tarea 3	**15.72**	**15.00**	**314**	**133**
Tarea 4	8.16	40.00	314	91

5.2. Experimento 2

Como se puede ver en la Tabla 5, al evaluar el sistema con tareas desconocidas por el modelo de registro. El FNR aumenta considerablemente, comparado, con los resultados del experimento anterior. Esto se puede dar por diversas causas, pero, principalmente se debe al hecho de que en este experimento, se realiza la verificación con un segmento de un párrafo. Por lo cual no es posible garantizar que todos los segmentos contengan desplazamientos largos, cortos, verticales u horizontales, que permitan definir una dinámica especifica. Aun así se puede resaltar que el FPR se mantiene a pesar de las condiciones.

5.3. Experimento 3

Este experimento es una extensión del experimento anterior, pero en este caso no se decide, si el usuario es un intruso o un usuario válido usando un

Tabla 5. Métricas de desempeño y usabilidad al generar los modelos de ingreso con tareas desconocidas por el modelo de registro (experimento 2)

Tarea de ingreso	FPR	FNR	CUE	CUA
Tarea 5	15.37	19.43	314	104

segmento y una distancia, sino usando varios segmentos y una distancia promedio. En la Tabla 6 se muestran las métricas de desempeño y usabilidad de este experimento. Al usar más segmentos para decidir si el usuario es válido o no, mejora el FPR y el FNR. Sin embargo a medida que se reduce el FPR y el FNR también se aumenta el CUE y el CUA, esto se debe a que a medida que aumentamos el número de caracteres necesarios para definir el modelo del ingreso, es más probable que el modelo tenga datos característicos de la dinámica del usuario. Por lo cual el sistema obtiene un mejor desempeño. Esto no quiere decir que para realizar una correcta verificación el usuario debe escribir muchos caracteres. Lo que sucede es que el texto debe ser tal, que contenga diversos desplazamientos y diversas letras que permitan modelar de forma adecuada la dinámica del usuario. Un texto extenso, que no tenga diversos desplazamientos y que no contenga diversos caracteres, posiblemente brindará un desempeño pobre. Pero para este caso al aumentar el número de segmentos, se aumenta la probabilidad de que el texto contenga diversos componentes de la dinámica natural del usuario y por ende mejora el desempeño del sistema.

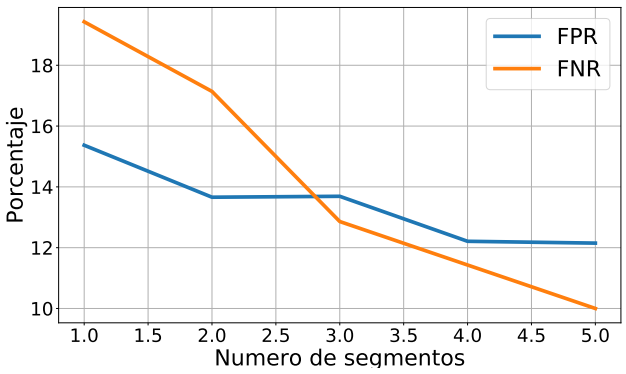


Figura 15. Desempeño Vs Numero de segmentos tomados

En la Figura 15 se muestran los cambios de FPR y FNR, al variar el número de segmentos. Como se puede ver el FPR tiende a mantenerse entre 12 % y 16 %, mientras que el FNR con pocos segmentos tiende a ser del 20 %, pero decrece mas rápidamente. Esto demuestra la capacidad del sistema para

Tabla 6. Métricas de desempeño y usabilidad al generar los modelos de ingreso, con tareas desconocidas por el modelo de registro, usando una distancia promedio (experimento 3)

Segmentos usados	FPR	FNR	CUE	CUA
2	13.66	17.14	314	104
3	13.69	12.86	314	208
4	12.21	11.43	314	416
5	**12.15**	**10.00**	**314**	**520**

mantener un bajo FPR, al variar el CUA. Nuevamente esta característica es útil para aplicaciones donde se requiera mantener un FPR bajo, sin importar que aumente la probabilidad de que el usuario deba autenticarse más de una vez.

5.4. Aplicación web

Como resultado final se entrega la aplicación Web VIUT (Verificación de Identidad Usando Tecleo), la cual permite registrar nuevos usuarios y verificar la identidad de usuarios registrados. La aplicación se encuentra disponible en el enlace: http://pbiometriaconductual.udea.edu.co/.

Figura 16. Plataforma Web inicio

En la Figura 16, se muestra la ventana de inicio del sistema de verificación de identidad VIUT. En esta ventana se brinda una pequeña introducción al sistema. La aplicación puede ser dividida en dos partes registro y ingreso.

5.4.1. Registro

A partir de la ventana de inicio, cuando el usuario se dirige a la opción "Empezar", esta lo llevará a la ventana de ingreso, en la cual el encontrará dos opciones, registrarse e ingresar. Al presionar registrarse, el usuario es redirigido a la ventana de registro. Allí el usuario debe ingresar sus datos personales y aceptar los términos y condiciones (ver Figura 17).

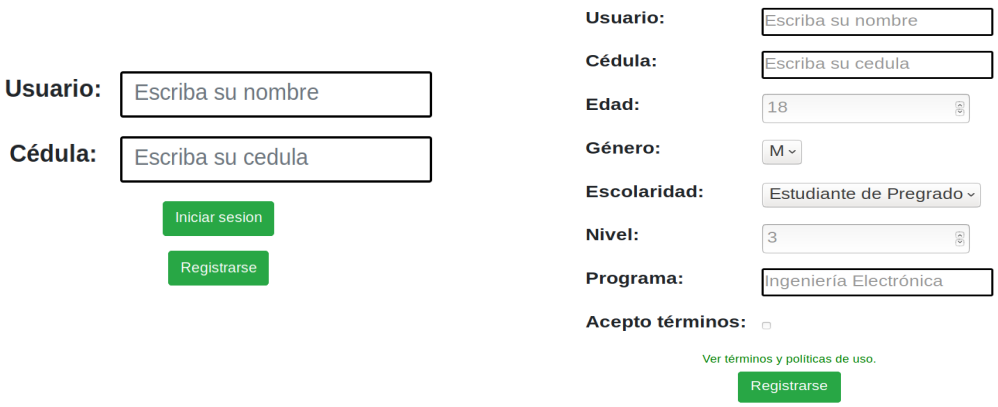


Figura 17. Ventana de registro

Luego de que el usuario brinda sus datos personales. Debe realizar las tareas de registro, las cuales permiten generar el modelo de registro (ver Figura 18). Estas tareas son las tareas 1, 2, 3 y 4 de las cuales se ha hablado anteriormente. Al terminar la tarea 4 el sistema genera el modelo del usuario y lo almacena en la base de datos de los modelos de usuario.

5.4.2. Ingreso

En el caso de que el usuario no haga el registro, la base de datos no tiene un modelo del usuario, por lo cual no permitirá el ingreso. En la Figura 19 se muestra los datos necesarios para ingresar a la plataforma y la tarea que usará el sistema para verificar la identidad del usuario. En este caso queremos verificar la identidad de un usuario cuando este escribe un texto extenso, de forma continua y desconocido por el registro.

Cada cierto número de caracteres, el sistema genera un modelo de ingreso y calcula la distancia de dicho modelo al modelo de registro. Cuando el usuario termina la tarea, el sistema compara la distancia promedio de los modelos, con el nivel de umbral (como en el experimento 3). Si el usuario es válido, el sistema redirige al usuario al mensaje de la Figura 20a. Por

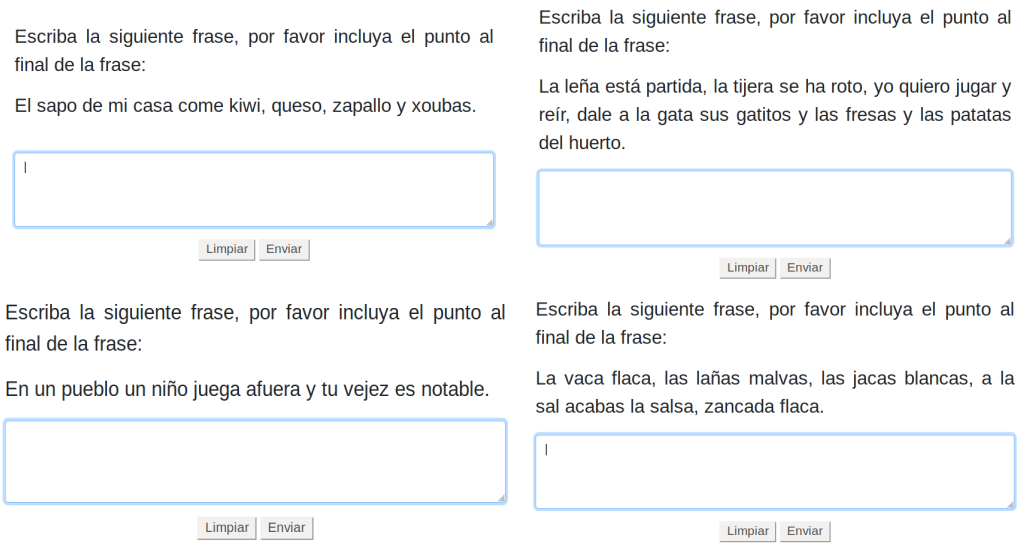


Figura 18. Tareas para generar el modelo en el registro.

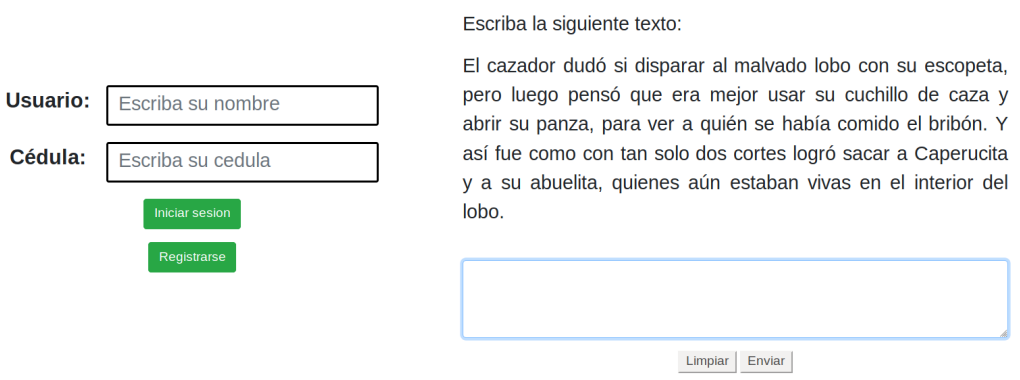


Figura 19. Ventana de ingreso.

el contrario, si el usuario es detectado como intruso, el sistema redirige al usuario al mensaje de la Figura 20b.

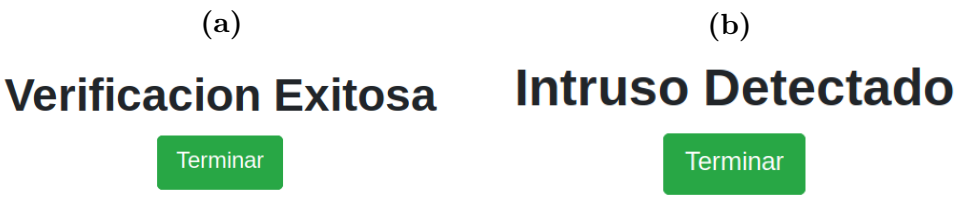


Figura 20. (a) Verificación exitosa, (b) Intruso detectado.

6. Conclusiones

En este trabajo se presenta una metodología para verificar identidad a partir de la dinámica de tecleo de los usuarios. Primero se hace necesario generar una etapa de desarrollo con el fin de obtener los parámetros necesarios para configurar el sistema. Posteriormente se probó el sistema, realizando ingresos con tareas conocidas por el modelo de registro, lo cual arrojo un FPR del 16 % y un FNR del 15 %. Más adelante se evaluó la generalidad del sistema, al probar con tareas desconocidas por el modelo de registro, lo que arrojo un FPR del 15 % y un FNR del 19 %. Este incremento en el FNR, motivó a variar el algoritmo de decisión, con el cual se determina si un usuario es válido o no. Esta variación consiste en no tomar la decisión, con una sola entrada del usuario, sino con múltiples entradas. Provocando que no se compare el nivel de umbral, con una distancia solamente, sino con un promedio de distancias. Esta variación ocasionó, que el FPR se redujera al 12 % y el FNR, al 10 %.

Según las métricas de desempeño presentadas en los experimentos anteriores, el sistema no logra superar los sistemas biométricos actuales, como por ejemplo los basados en voz o rostro. Sin embargo, este sistema posee diversas ventajas respecto a otros sistemas de verificación. Principalmente es un sistema que no requiere hardware adicional, como cámara o diadema, ya que únicamente requiere el uso del teclado. Dado que no requiere un hardware adicional, no necesita permisos para acceder a la cámara o al micrófono, por lo cual el sistema es totalmente transparente para el usuario, lo que incrementa la seguridad y reduce el riesgo de fraude.

Los experimentos implementados en este trabajo, presentan variaciones, que permiten analizar el sistema de verificación desde dos enfoques diferentes. Sistemas de verificación intrusivos, en los cuales el sistema no permite el acceso hasta que el usuario brinda un identificador. Y sistemas de verificación no intrusivos, donde la plataforma brinda acceso al usuario y después, generalmente sin que el usuario lo sepa, la plataforma verifica la identidad.

Para el caso de sistemas intrusivos, la metodología del experimento 1 brinda una solución aceptable. Como se puede ver en la Tabla 4, cuando el modelo del registro es generado con unas tareas específicas y en el modelo del ingreso, se genera con alguna de estas tareas. El FPR y el FNR pueden llegar a ser cercanos a un 15 %, lo cual generaría una mayor seguridad en este tipo de sistemas. Por ejemplo, asumamos un sistema intrusivo típico, una red social. En una red social generalmente cada usuario posee una contraseña. Dado el caso que algún intruso tuviera acceso a la contraseña de un usuario, con una probabilidad alta, el intruso podrá ingresar como si fuera el usuario real. Por el contrario, usando la metodología de este trabajo, aunque el intruso conozca

la contraseña, la probabilidad de que el intruso logre ingresar a la cuenta, puede ser reducida hasta el 15.72 %. Por lo tanto el enfoque del experimento 1, puede servir para mejorar la seguridad en algunos sistemas de verificación intrusivos.

Para sistemas no intrusivos, la metodología implementada en el experimento 3, muestra una solución. Para este experimento, el modelo de registro se genera a partir de unas tareas especificas, pero el modelo de ingreso puede ser un texto de cualquier tema. Esta flexibilidad, permite que esta metodología pueda ser empleada para verificar la identidad de forma transparente al usuario. Mientras el usuario teclea, el sistema va evaluando si el usuario es valido o no. Este enfoque es ideal para las actividades evaluativas en los cursos virtuales.

Para cualquiera de los enfoques el sistema tiende a rechazar muy bien los impostores, pero para verificar el usuario como valido, presenta diversas dificultades que pueden ser solucionadas aumentando el CUA. Como se puede ver en la Figura 15 el FNR se reduce rápidamente al aumentar el número de segmentos necesarios para determinar si el usuario es valido o no. A diferencia del FPR, que tiende a mantenerse estable al aumentar el numero de segmentos. Este comportamiento permite estimar, que con un CUA alto, el FNR tiende a disminuir sin aumentar el FPR.

Como trabajo futuro es necesario realizar más experimentos con un mayor número de textos diferentes y con diferentes metodologías, con el fin de evaluar y construir un sistema más robusto. Además el sistema deberá ser evaluado en otros idiomas. Pues dado que no es un sistema que se enfoque en lo que se escribe, sino en como se escribe, es de esperarse que pueda funcionar en diferentes idiomas. Por último, este tipo de señales son sensibles a los cambios de estado de ánimo de los usuarios, por lo cual se pretende clasificar las emociones de los usuarios.

Anexos

A. Variable Aleatoria Continua

Una variable aleatoria continua es una función que asigna un valor numérico en el dominio de los reales al resultado de un experimento aleatorio [26]. Debido a que el experimento es aleatorio, el valor que toma la variable, también lo es. Por lo cual, dada una variable aleatoria no es posible conocer con certeza el valor que tomará esta al ser medida o determinada, pero si es posible conocer su distribución de probabilidad. La distribución o función de densidad de probabilidad PDF (Del inglés, *Probability Density Function*) es una de las características mas importantes de una variable aleatoria x.

Dado a que una variable aleatoria continua puede tomar valores entre $-\infty$ y ∞, la probabilidad de que la variable aleatoria tome un valor particular siempre dará como resultado 0 puesto que el tamaño del conjunto es infinito (ver Ecuación 27). Por ende cuando hablamos de una variable aleatoria, siempre se relaciona con ella una PDF la cual es una descripción de la ley de probabilidad que rige a esa variable aleatoria [26]. La Ecuación 25 muestra como obtener una representación de la probabilidad de que la variable aleatoria tome un valor dentro del intervalo a y b a partir de su PDF.

$$P(a \leq x \leq b) = \int_a^b f(x)dx. \tag{25}$$

Aunque la PDF no es una probabilidad como tal, sino más bien una probabilidad por unidad de longitud (en el caso de una variable aleatoria de una dimensión), cumple con las propiedades de no negatividad (ver Ecuación 26) y con la propiedad de normalización (ver Ecuación 28):

$$P(\cdot) = \int f(x)dx \geq 0, \tag{26}$$

$$P(x = a) = \int_a^a f(x)dx = 0, \tag{27}$$

$$P(-\infty \leq x \geq \infty) = \int_{-\infty}^{\infty} f(x)dx = 1. \tag{28}$$

Para un vector aleatorio continuo $\boldsymbol{x} = (x_1, x_2...., x_D)^\intercal \in \mathbb{R}^D$, es posible definir su PDF conjunta $f(x_1, x_2, ..., x_D)$. Como se muestra en la Ecuación 29. A este tipo de variable aleatoria se le conoce como variable aleatoria multivariada.

$$\int_{x_1}\int_{x_2} ..... \int_{x_D} f(x_1, x_2, ..., x_D)dx_1, dx_2, ...dx_D \tag{29}$$

En este trabajo cada usuario genera una variable aleatoria multivariada, donde una dimensión podría ser el tiempo de retención, otra podría ser el tiempo de vuelo y otra la tecla que pulsa el usuario.

7. Referencias

[1] N. Arias Velandia, D. M. Cardona Román, J. M. Sánchez Torres, A. Márquez, C. Andrés, J. Guarnizo Mosquera, D. Ortiz Romero, E. Gómez Villarreal, L. Rojas Benavides, E. Huitrón y col., *Aportes a la investigación sobre educación superior virtual desde América Latina. Comunicación, redes, aprendizaje y desarrollo institucional y social*. 2018.

[2] V. Matyáš y Z. Řıha, "Biometric authentication—security and usability", en *Advanced Communications and Multimedia Security*, Springer, 2002, págs. 227-239.

[3] D. Bhattacharyya, R. Ranjan, F. Alisherov, M. Choi y col., "Biometric authentication: A review", *International Journal of u-and e-Service, Science and Technology*, vol. 2, n.º 3, págs. 13-28, 2009.

[4] W. L. Bryan y N. Harter, "Studies in the physiology and psychology of the telegraphic language.", *Psychological Review*, vol. 4, n.º 1, pág. 27, 1897.

[5] R. Joyce y G. Gupta, "Identity authentication based on keystroke latencies", *Communications of the ACM*, vol. 33, n.º 2, págs. 168-176, 1990.

[6] K. Longi, J. Leinonen, H. Nygren, J. Salmi, A. Klami y A. Vihavainen, "Identification of programmers from typing patterns", en *Proceedings of the 15th Koli Calling Conference on Computing Education Research*, ACM, 2015, págs. 60-67.

[7] S. Krishnamoorthy, L. Rueda, S. Saad y H. Elmiligi, "Identification of user behavioral biometrics for authentication using keystroke dynamics and machine learning", en *Proceedings of the 2018 2nd International Conference on Biometric Engineering and Applications*, ACM, 2018, págs. 50-57.

[8] S. Bajaj y S. Kaur, "Typing speed analysis of human for password protection (based on keystrokes dynamics)", *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 3, n.º 2, págs. 88-91, 2013.

[9] F. Monrose y A. D. Rubin, "Keystroke dynamics as a biometric for authentication", *Future Generation computer systems*, vol. 16, n.º 4, págs. 351-359, 2000.

[10] D. Gunetti y C. Picardi, "Keystroke analysis of free text", *ACM Transactions on Information and System Security (TISSEC)*, vol. 8, n.º 3, págs. 312-347, 2005.

[11] A. K. Jain, A. Ross, S. Prabhakar y col., "An introduction to biometric recognition", *IEEE Transactions on circuits and systems for video technology*, vol. 14, n.º 1, 2004.

[12] M. S. Obaidat y B. Sadoun, "Verification of computer users using keystroke dynamics", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 27, n.º 2, págs. 261-269, 1997.

[13] W. R. Adams, "High-accuracy detection of early Parkinson's Disease using multiple characteristics of finger movement while typing", *PloS one*, vol. 12, n.º 11, e0188226, 2017.

[14] H. Crawford, "Keystroke dynamics: Characteristics and opportunities", en *2010 Eighth International Conference on Privacy, Security and Trust*, IEEE, 2010, págs. 205-212.

[15] D. Yu y L. Deng, *Automatic speech recognition.* Springer, 2016.

[16] D. A. Reynolds y R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models", *IEEE transactions on speech and audio processing*, vol. 3, n.º 1, págs. 72-83, 1995.

[17] D. A. Reynolds, T. F. Quatieri y R. B. Dunn, "Speaker verification using adapted Gaussian mixture models", *Digital signal processing*, vol. 10, n.º 1-3, págs. 19-41, 2000.

[18] C. E. Rasmussen, "The infinite Gaussian mixture model", en *Advances in neural information processing systems*, 2000, págs. 554-560.

[19] A. P. Dempster, N. M. Laird y D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, n.º 1, págs. 1-22, 1977.

[20] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[21] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions", *Bull. Calcutta Math. Soc.*, vol. 35, págs. 99-109, 1943.

[22] P. C. Mahalanobis, "On the generalized distance in statistics", National Institute of Science of India, 1936.

[23] T. Arias-Vergara, J. C. Vásquez-Correa, J. R. Orozco-Arroyave, J. F. V. Bonilla y E. Nöth, "Parkinson's Disease Progression Assessment from Speech Using GMM-UBM.", en *INTERSPEECH*, 2016, págs. 1933-1937.

[24] D. Reynolds, "Gaussian mixture models", *Encyclopedia of biometrics*, págs. 827-832, 2015.

[25] A. Peacock, X. Ke y M. Wilkerson, "Typing patterns: A key to user identification", *IEEE Security & Privacy*, vol. 2, n.º 5, págs. 40-47, 2004.

[26] D. P. Bertsekas y J. N. Tsitsiklis, *Introduction to probability*. Athena Scientific Belmont, MA, 2002, vol. 1.