

Enrique P. Calot

Robustez de las métricas de clasificación de cadencia de tecleo frente a variaciones emocionales

**TESIS DOCTORAL EN CIENCIAS INFORMÁTICAS
PREMIO DR. RAÚL GALLARD | Año 2020**

**Robustez de las métricas de clasificación de
cadencia de tecleo frente a variaciones
emocionales**

Enrique P. Calot

UNIVERSIDAD NACIONAL DE LA PLATA FACULTAD DE INFORMÁTICA

TESIS DOCTORAL EN CIENCIAS INFORMÁTICAS

Robustez de las métricas de clasificación de cadencia de tecleo frente a variaciones emocionales

Enrique P. Calot

Director: Dr. Jorge S. Ierache
Codirector: Dr. Waldo Hasperué

La Plata, Argentina
Junio de 2020



Calot, Enrique P.

Robustez de las métricas de clasificación de cadencia de tecleo frente a variaciones emocionales / Enrique P. Calot. - 1a ed. -

La Plata: EDULP, 2021.

Libro digital, PDF

Archivo Digital: descarga y online

ISBN 978-987-8348-97-1

1. Tecnología Informática. I. Título.

CDD 004.02

Robustez de las métricas de clasificación de cadencia de tecleo frente a variaciones emocionales

ENRIQUE P. CALOT



EDITORIAL DE LA UNIVERSIDAD NACIONAL DE LA PLATA (EDULP)

48 N° 551-599 4º Piso/ La Plata B1900AMX / Buenos Aires, Argentina

+54 221 44-7150

edulp.editorial@gmail.com

www.editorial.unlp.edu.ar

Edulp integra la Red de Editoriales de las Universidades Nacionales (REUN)

ISBN 978-987-8348-97-1

Queda hecho el depósito que marca la Ley 11.723

© 2021 - Edulp

Universidad Nacional de La Plata

Abstract

Facultad de Informática

Doctorado en Ciencias Informáticas

Robustness of keystroke dynamics classification metrics
facing emotional variations by

Enrique P. Calot

Keystroke dynamics facilitates the identification of a person by the way they type. This thesis focuses on analysing the robustness of keystroke dynamics algorithms against variations in biometric records through electroencephalography and a self-report questionnaire, using the dimensional approach to model affective states. An experiment was conducted to capture typing patterns in different affective states. The results suggested specific classification distances, such as the A and R metrics, Canberra, Manhattan and a Minkowski-based distance have their accuracy slightly and negatively influenced by the changes in arousal and valence responses. Euclidean distance was the least affected among the six evaluated distances.

Universidad Nacional de La Plata

Resumen

Facultad de Informática

Doctorado en Ciencias Informáticas

Robustez de las métricas de clasificación de cadencia de tecleo frente
a variaciones emocionales por Enrique P. Calot

La dinámica de tecleo permite la identificación de una persona por la forma en que escribe. Esta tesis se enfoca en analizar la robustez de los algoritmos de análisis de cadencia de tecleo frente a variaciones en los registros biométricos mediante electroencefalografía y cuestionario de autoinforme, utilizando el enfoque dimensional para modelizar estados afectivos. Se realizó un experimento para capturar patrones de tecleo en diferentes estados afectivos. Los resultados sugirieron que la tasa de aciertos para ciertas distancias de clasificación, como las métricas A y R, la distancia de Camberra, Manhattan y una distancia basada en Minkowski se ven influenciadas negativamente por los cambios en las respuestas de excitación y valencia. La distancia euclídea fue la menos afectada de las seis evaluadas.

A mis padres...

Agradecimientos

Durante estos años recibí el soporte de infinidad de personas, mencionarlos a todos me sería imposible y pido disculpas a quienes dejo afuera. He aquí un intento de mencionar a los más importantes. Quiero agradecer primero al Dr. Jorge Ierache, quien me dirigió, por incentivar me, motivarme, ayudarme, soportarme —en todo sentido—, atraerme al análisis de emociones y brindarme sus valiosos consejos. Al Dr. Waldo Hasperué, mi codirector, por todas las correcciones. A la Facultad de Informática de la Universidad Nacional de La Plata por brindarme, a través de los cursos de doctorado, un mayor conocimiento. A los docentes y personal de la Secretaría de Posgrado, por el apoyo recibido. También quisiera agradecer al Laboratorio de Sistemas de Información Avanzados (LSIA) de la Facultad de Ingeniería de la Universidad de Buenos Aires, donde radiqué la tesis. A Nahuel González, Federico Rossi y Germán Concilio del LSIA, por sus contribuciones y el apoyo brindado. A mis compañeros de trabajo de Cloudie y Amazon, por comprender mi dedicación a este doctorado. A Leslie y Liia Miller, por el humor y todas las correcciones a los trabajos que presentamos en el exterior.

Además, quiero agradecer a mis padres, Enrique y Mónica, y el resto de mi familia por haberme apoyado ciegamente cualquiera sea mi proyecto. Oma y Bompá, que ya no están, pero me enseñaron todo; sin ellos no sería yo. Y muy especialmente a Paula, mi compañera, por el sustento incondicional, la revisión del texto y el aguante durante tantas noches sin dormir; habernos privado de tantos fines de semanas de paseos finalmente rindió sus frutos.

Por último, mi agradecimiento al lector, por la curiosidad y el tiempo dedicado para leer este trabajo.

ÍNDICE

CAPÍTULO 1

Introducción.....	9
1.1 Fundamentos y contexto.....	9
1.2 Motivación.....	10
1.3 Trabajos previos en congresos.....	11
1.4 Producción científica derivada de resultados parciales de la tesis..	12
1.5 Estructura de la tesis	14
1.6 Reconocimientos.....	15

CAPÍTULO 2

Antecedentes, definiciones y estado del arte	17
2.1 Esquemas de autenticación	18
2.2 Conceptos utilizados en biometría	21
2.3 Modelado y captura de emociones.....	28
2.4 Modelado de la cadencia de tecleo.....	43
2.5 Detección de impostores con cadencia de tecleo	51
2.6 Detección de estados emocionales con cadencia de tecleo	69
2.7 Consideraciones técnicas.....	73

CAPÍTULO 3

Descripción del problema y objetivos.....	79
3.1 Descripción del problema.....	79
3.2 Objetivos.....	80
3.3 Hipótesis.....	80
3.4 Alcance.....	81

CAPÍTULO 4

Diseño del experimento.....	82
-----------------------------	----

4.1 Adquisición de datos	82
4.2 Validación con simulacros de prueba	87
4.3 Implementación	88
CAPÍTULO 5	
Validación del experimento	90
5.1 Datasets de control	90
5.2 Verificación de cambio de estado afectivo.....	95
5.3 Elección de métricas y parámetros.....	100
5.4 Elección de la cantidad de keystrokes	107
(tamaño de la muestra)	107
5.5 Configuración del experimento y consideraciones	121
CAPÍTULO 6	
Resultados experimentales	124
6.1 Análisis gráfico de los casos	127
6.2 Análisis de los casos.....	129
CAPÍTULO 7	
Conclusiones	133
7.1 Discusión de los resultados.....	133
7.2 Aportes.....	135
7.3 Futuras líneas de investigación.....	136
7.4 Futuras líneas de trabajo	137
APÉNDICE A	
Definiciones, métodos y conceptos algebraicos	138
A.1 Métricas y distancias.....	138
A.2 Redes neuronales y algoritmo de <i>backpropagation</i>	145
A.3 Bosques aleatorios.....	156
A.4 Árboles de decisión.....	157
A.5 Tests de hipótesis.....	160

APÉNDICE B

Herramientas y <i>frameworks</i> desarrollados	165
B.1 Captura de la cadencia de tecleo	165
B.2 <i>Framework</i> en Python	167
CÓDIGO B.2: CÓDIGO PARA POSTPROCESAMIENTO DE DATOS EN PYTHON	175
B.3 <i>Framework</i> en C.....	175
descarte y filtrado para cálculo posterior de EER.....	177
CÓDIGO B.5: MÓDULO DE SOPORTE DE VARIAS MÉTRICAS	178

APÉNDICE C

Formatos recolectados	179
C.1 Lecturas crudas en formato de valores separados por coma	179
C.2 Archivos JSON.....	180
C.3 Cadencia de tecleo.....	181
C.4 Video	183
C.5 Frecuencia cardíaca	184
C.6 Archivos de EEG.....	184

APÉNDICE D

Cámara de inmersión emocional	185
D.1 Estimulación.....	185
D.2 Adquisición de datos y modalidades utilizadas por esta tesis .	187
D.3 Adquisición de datos y modalidades no utilizadas por esta tesis .	189
D.4 Descripción del entorno.....	194

APÉNDICE E.....

Revisión sistematizada de la literatura	196
E.1 Objetivos de la revisión.....	196
E.2 Expresiones de búsqueda utilizadas	197
E.3 Bases de datos consultadas.....	198

E.4 Criterios de selección	199
E.5 Resultados obtenidos.....	200
E.6 Conjuntos de datos públicos	201
APÉNDICE F	
Valores obtenidos por EEG_y SAM	203
APÉNDICE G	
Consumo de recursos y capacidad computacional	232
Referencias bibliográficas	234
Lista de Acrónimos	259

Introducción

Identificar y autenticar a un usuario en un sistema se ha tornado una necesidad fundamental en los sistemas informáticos.

Existen tres tipos de seguridad en sistemas computacionales: la basada en el conocimiento —como la utilización de palabras o frases secretas—, la basada en la posesión de algo físico —como una tarjeta de coordenadas, huella dactilar, o algún dispositivo, como por ejemplo, un teléfono móvil o un *token* generador de códigos— y la basada en el comportamiento. La primera ha sido el principal medio de autenticación de usuarios desde que existen los sistemas computacionales. El avance de la tecnología y la disminución de los costos de hardware han generado durante los últimos diez años un crecimiento en el uso de métodos alternativos enmarcados en lo que es seguridad biométrica y/o basada en la posesión (Jain *et al.*, 2006; Wong y Anderson, 1999).

1.1 Fundamentos y contexto

Según O’Gorman (2003), las limitaciones inherentes a los esquemas basados en el conocimiento y sus problemas de seguridad asociados permitieron, junto con el avance tecnológico, masificar las tecnologías necesarias para la seguridad biométrica durante los últimos años. Estas limitaciones incluyen la utilización de contraseñas cortas o conocidas (Zviran y Haga, 1999), la reutilización de las mismas en distintos sistemas (Ives *et al.*, 2004) y su administración irresponsable (Bryant y Campbell, 2006). Muchos sistemas manejan políticas de creación de contraseñas para garantizar un mayor espacio posible de combinaciones y así prevenir un ataque por fuerza bruta. Estas políticas pueden derivar en contraseñas con un largo mínimo, con ciertos caracteres, símbolos o cantidad mínima de caracteres en mayúsculas, minúsculas o cualquier combinación posible. Estas políticas generan que los usuarios deban

utilizar contraseñas distintas según el sitio y deban recordarlas cada vez; lo que disminuye el manejo irresponsable de las mismas pero, a su vez, incrementa la posibilidad de que el usuario olvide o pierda su contraseña (Campbell *et al.*, 2007; González *et al.*, 2016).

Conociendo las limitaciones de la seguridad por conocimiento, surgen estos otros dos tipos alternativos de seguridad, basadas en la posesión y el comportamiento. Dentro de los basados en el comportamiento, se encuentra el modelado de la *cadencia de tecleo*, que es fácilmente integrable en un sistema, no es intrusivo ni requiere de pasos adicionales para el usuario, asumiendo que es él mismo quien se encuentra frente al teclado. Esto permite combinar ambos métodos en lo que se llama una *clave endurecida* (Monrose *et al.*, 1999), representada por la contraseña y el ritmo de tecleo del usuario. Además, la cadencia de tecleo permite autenticar individuos en tiempo real mientras los mismo se encuentran tipeando textos libres, técnica llamada autenticación continua (Flori y Kowalski, 2010). Hoy en día la seguridad es uno de los factores más importantes en lo que respecta la utilización de sistemas online, como por ejemplo los sistemas bancarios o sitios de comercio electrónico (Lai *et al.*, 2014) donde se almacena información financiera o la medicina, que administra datos sumamente confidenciales.

Por otra parte, el estado de ánimo de una persona afecta los patrones de comportamiento y, con esto, es de esperar que afecte tanto su firma manuscrita como su patrón de ritmo de tecleo (Epp *et al.*, 2011). Lo mismo podría ocurrir con el transcurso del tiempo, de la misma forma en que se modifica una huella dactilar. Varios métodos han sido desarrollados y explorados con el objetivo de identificar estados emocionales. Estos métodos están basados en diferentes recursos fisiológicos (Solanki y Shukla, 2014).

1.2 Motivación

Desde 2013, dentro del Laboratorio de Sistemas de Información Avanzados de la Universidad de Buenos Aires (LSIA), hemos realizado investigaciones que exploran la cadencia de tecleo y el estado emocional.

En lo personal, tras haber trabajado en seguridad de la información en el área de la salud y el comercio electrónico, he sido testigo de la

necesidad de utilizar algoritmos que incrementen la seguridad de los usuarios. A diferencia de los sistemas bancarios, en este tipo de sistemas existen mayores oportunidades de ingreso de textos libres y validar el tipeo continuo de los usuarios. La cadencia de tecleo se combina de manera natural con los sistemas web para proveer una mayor seguridad. Tanto al ingresar una opinión sobre un producto o una historia clínica, los usuarios se pueden ver afectados por distintas emociones y esto podría afectar la capacidad de clasificación de los algoritmos.

1.3 Trabajos previos en congresos

En el contexto del LSIA trabajamos en las siguientes publicaciones vinculadas a la presente tesis doctoral:

- Enrique P. Calot, Juan Manuel Rodríguez y Jorge S. Ierache. Improving versatility in keystroke dynamic systems. En *Proceedings del XIX Congreso Argentino de Ciencias de la Computación*, number 5606, 2013. ISBN 978-987-23963-1-2. URL <http://hdl.handle.net/10915/32428>
- Enrique P. Calot, Juan Manuel Rodríguez y Jorge S. Ierache. Improving versatility in keystroke dynamic systems. En Jorge Raúl Finochietto y Patricia Mabel Pesado, editors, *Computer Science & Technology Series. XIX Argentine Congress of Computer Science, Selected papers*, páginas 289–298. Editorial de la Universidad Nacional de La Plata (EDULP), 2014b. ISBN 978-987-1985-49-4. URL <http://lsia.fi.uba.ar/papers/calot14b.pdf>
- Enrique P. Calot, Francisco Pirra, Juan Manuel Rodríguez, Gustavo Pereira, Juan Iribarren y Jorge S. Ierache. Métodos adaptativos de educación de dinámica de tecleo centrado en el contexto emocional de un individuo aplicando interfaz cerebro computadora. En *XVI Workshop de Investigadores en Ciencias de la Computación*, mar. 2014a. ISBN 978-950-34-1084-4. URL <http://lsia.fi.uba.ar/papers/calot14.pdf>

1.4 Producción científica derivada de resultados parciales de la tesis

1.4.1 Series internacionales y capítulos de libro

- Enrique P. Calot, Jorge S. Ierache y Waldo Hasperué. Robustness of keystroke dynamics identification algorithms against brain-wave variations associated with emotional variations. En *Intelligent Systems and Applications*, volume 1037 de *Advances in Intelligent Systems and Computing*. Springer, 2019c. ISBN 978-3-030-29515-8. doi: 10.1007/978-3-030-29516-5_15
- Nahuel González, Enrique P. Calot y Jorge S. Ierache. A replication of two free text keystroke dynamics experiments under harsher conditions. En *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, páginas 1–6, sep. 2016. doi: 10.1109/BIOSIG.2016.7736905
- Nahuel González y Enrique P. Calot. Finite context modeling of keystroke dynamics in free text. En *Biometrics Special Interest Group (BIOSIG), 2015 International Conference of the*, páginas 1–5, sep. 2015. ISBN 978-3-88579-639-8. doi: 10.1109/BIOSIG.2015.7314606

1.4.2 Congresos internacionales

- Enrique P. Calot, Jorge S. Ierache y Waldo Hasperué. Document typist identification by classification metrics applying keystroke dynamics under unidealised conditions. En *Second International Workshop on Computational Document Forensics, IWCDF at ICDAR 2019, Sydney, Australia, September 22–25, 2019*, páginas 19–24. IEEE, 2019b. ISBN 978-1-7281-5054-3. doi: 10.1109/ICDARW.2019.70136
- Enrique P. Calot, Nahuel González, Jorge S. Ierache y Waldo Hasperué. A convergence model for distance-based keystroke dynamics classification methods, 2019a. Comunicado el 7 de febrero de 2019

1.4.3 Congresos nacionales y workshops

- Enrique P. Calot y Jorge S. Ierache. Multimodal biometric recording architecture for the exploitation of applications in the context of affective computing. En *Proceedings del XXIII Congreso Argentino de Ciencias de la Computación (La Plata, 2017)*, number 10529, páginas 1030–1039, 2017. ISBN 978-950-34-1539-9. URL <http://hdl.handle.net/10915/63866>
- Germán M. Concilio, Enrique P. Calot, Jorge S. Ierache y Hernán D. Merlino. Application of Keystroke Dynamics Modelling Techniques to Strengthen the User Identification in the Context of E-commerce. En Armando Eduardo De Giusti, editor, *VII Workshop de Seguridad Informática (CACIC 2018, Tandil, Buenos Aires, Argentina)*, páginas 1051–1059, 2018. ISBN 978-950-658-472-6
- Gustavo Pereira, Facundo Nervo, Juan Iribarren, Enrique P. Calot, Jorge S. Ierache y Norberto Mazza. Líneas de investigación aplicada al empleo de interfaces cerebromaquina. En *Proceedings del XVII Workshop de Investigadores en Ciencias de la Computación (Salta, 2015)*, página 5, jun. 2015. URL <http://hdl.handle.net/10915/46146>
- Nahuel González, Enrique P. Calot, Ariel Liguori, Francisco Pirra, Juan Manuel Rodríguez, Gustavo Pereira, Facundo Nervo, Juan Iribarren y Jorge S. Ierache. Educación de dinámica de tecleo centrado en el contexto emocional de un individuo. En *Proceedings del XVII Workshop de Investigadores en Ciencias de la Computación (Salta, 2015)*, página 5, jun. 2015. URL <http://hdl.handle.net/10915/46260>
- Enrique P. Calot, Ezequiel L. Aceto, Juan Manuel Rodríguez, Ariel Liguori, María Alejandra Ochoa, Hernán D. Merlino, Enrique Fernández, Nahuel González, Francisco Pirra y Jorge S. Ierache. Líneas de investigación del laboratorio de sistemas de información avanzados. En *Proceedings del XVII Workshop de Investigadores en Ciencias de la Computación (Salta, 2015)*, página 4, jun. 2015. ISBN 978987-633-134-0. URL <http://hdl.handle.net/10915/46107>
- Enrique P. Calot, Federico M. Rossi, Nahuel Gonzalez, Waldo Hasperué y Jorge S. Ierache. Avances en educación de dinámica de

tecleo y el contexto emocional de un individuo aplicando interfaz cerebro computadora.

En *XVIII Workshop de Investigadores en Ciencias de la Computación (WICC 2016, Entre Ríos, Argentina)*, páginas 872–876, jun. 2016. ISBN 978-950-698-377-2. URL <http://hdl.handle.net/10915/53247>

- Federico M. Rossi, Enrique P. Calot y Jorge S. Ierache. Educación emocional de un individuo en contextos multimodales en computación afectiva. En *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, Buenos Aires, Argentina)*, jul. 2017. ISBN 978-987-42-5143-5

1.5 Estructura de la tesis

Luego de la presente introducción —que familiariza al lector en el contexto de la tesis—, se realiza un relevamiento del estado actual del arte en el capítulo 2. En ella se repasa brevemente la literatura. La sección 2.1 define los distintos esquemas de autenticación, su forma de combinarlos y la diferencia entre identificación y validación. La sección 2.2 define conceptos básicos de la biometría y diferentes métricas que son comúnmente utilizados por la literatura. La sección 2.3 explica las formas existentes de modelar emociones y los antecedentes en captura de las mismas; a continuación, la sección 2.4 explica las distintas formas de modelizar y atacar el problema de cadencia de tecleo. Las siguientes secciones relevan los antecedentes de trabajos en cadencia de tecleo relacionados a la detección de impostores (sec. 2.5) y de emociones (sec. 2.6). La sección 2.7 repasa consideraciones, desafíos y problemas existentes en esta área.

El capítulo 3 describe de la manera más sintética posible, el problema (sec. 3.1), los objetivos (sec. 3.2), la hipótesis (sec. 3.3) y el alcance (sec. 3.4) de esta tesis.

El capítulo 4 describe la solución propuesta y el diseño del experimento para evaluar la hipótesis. La sección 4.1 desarrolla la metodología de adquisición, su arquitectura y las fases utilizadas. La sección 4.1.1, en cambio, define la mecánica a utilizar para el procesamiento de los datos obtenidos y validar la hipótesis. La sección 6.1 describe las lecciones aprendidas a la hora de realizar el

experimento. Finalmente, en la sección 4.3 se explica la ejecución de la captura y sus participantes.

Durante el desarrollo de esta tesis se investigan tres preguntas, las dos primeras son secundarias y necesarias para avanzar sobre la tercera, que es la central. El capítulo 5 describe estas dos preguntas, comunicadas en Calot *et al.* (2019b) y Calot *et al.* (2019a) respectivamente en las secciones 5.3 —respondiendo qué métricas se deben utilizar en el análisis de los resultados— y 5.4 —respondiendo qué cantidad de keystrokes es necesaria para que las pruebas sean estadísticamente significativas—. Para analizar estos dos puntos fue necesaria la utilización de dos conjuntos de datos definidos previamente en la sección 5.1. La solución propuesta relacionada al componente principal se define en la sección 5.2, donde además se incluyen los gráficos y tablas asociadas a un caso de prueba realizado para el participante 6. Finalmente, en la sección 5.5 expone consideraciones relacionadas.

El capítulo 6 presenta los resultados obtenidos y, en la sección 6.2, analiza caso por caso los estados emocionales de los participantes.

El capítulo 7 discute los resultados, exponiendo conclusiones (sec. 7.1) y aportes de la tesis (sec. 7.2) y definiendo futuras líneas de trabajo (sec. 7.4) e investigación (sec. 7.3).

Este documento, además, cuenta con siete apéndices; a saber, con definiciones, métodos y conceptos algebraicos (apéndice A); las herramientas y *frameworks* desarrollados para esta tesis (apéndice B); los formatos de los datos recolectados durante la etapa de experimentación (apéndice C); los detalles del experimento y la descripción de la cámara de inmersión emocional (apéndice D); una revisión sistematizada de la literatura (apéndice E); todos los valores y gráficas obtenidos por electroencefalografía (EEG) y un autoinforme del tipo *self-assessment manikin* o SAM (apéndice F) y, finalmente, un análisis del consumo de recursos y capacidad computacional para el procesamiento de los resultados (apéndice G).

1.6 Reconocimientos

Esta tesis también reconoce a la Universidad de Buenos Aires, donde bajo los proyectos UBACyT 20020130200140BA y PDE-44-2019

radicados en el Laboratorio de Sistemas de Información Avanzados de la Facultad de Ingeniería, recibimos fondos y el espacio físico para la instalación de la cámara de inmersión emocional, donde realizamos los experimentos; además de fondos para investigación, compra de equipos y asistencia a congresos nacionales e internacionales. Adicionalmente, el ISIER (FICCTE-SECYT de la Universidad de Morón) prestó asistencia en el área de aplicación de BCI, bajo el PID 01-001/12/14.

Quiero reconocer a los trabajos de González (2016); Oude Bos (2006); Rossi (2019), donde la sintética y estética de sus figuras sirvieron de inspiración para algunas de las presentadas en esta tesis. Parte del estado del arte de *keystroke dynamics* es una actualización del presentado en González (2016) y los modelos emocionales de Rossi (2019); ambas tesis donde tuve el placer de colaborar, además de haber realizado en conjunto con dichos autores, investigaciones que se encuentran incluidas en la presente tesis. En la revisión sistematizada de la literatura reutilizo la metodología aplicada en Concilio (2019), a quien quiero agradecer también por el apoyo en este aspecto.

Antecedentes, definiciones y estado del arte

Hacia fines del siglo XIX, Bryan y Harter, estudiando las características psicológicas del aprendizaje de la telegrafía, detectan que los operadores telegráficos tienen estilos de tipeo el código morse lo suficientemente distintivos como para ser atribuidos a su originante (Bryan y Harter, 1897; González *et al.*, 2016). Durante la Segunda Guerra Mundial, los operadores de radio británicos pudieron interceptar e identificar a los radiotelegrafistas alemanes por su estilo de transmisión (Gladwell y Ruitter, 2008) y, de este modo, seguir sus movimientos. Incluso definieron su estilo como *el puño del telegrafista*. El ritmo de escritura es único para cada persona, y las variaciones entre las personas son tan profundas que permiten la identificación correcta del mecanógrafo. Análogamente, en el teclado, tanto estudio y como el estilo de tipeo en sí se denominan *keystroke dynamics*. En las computadoras, el inicio de esta área particular de investigación se remonta a Gaines *et al.* (1980), quien en 1980 realizó el primer análisis estadístico con textos libres y palabras sueltas. Cada estilo de escritura es personal, único y no puede ser robado, perdido u olvidado (Polemi, 1997).

El análisis del ritmo de tipeo, como característica biométrica, se puede aplicar sobre un texto estático o un texto libre continuo. Los textos estáticos, como las contraseñas, se escriben frecuentemente con la misma secuencia; mientras que los textos libres continuos comparan los estilos de escritura utilizando secuencias de teclas comunes entre dos textos para determinar si pertenecen a la misma persona. La longitud de estos textos puede diferir considerablemente. Los textos libres requieren cadenas de teclas más largas, para que coincidan con la precisión de los textos estáticos. Los textos libres, tienen la complicación de que los patrones deben adquirirse durante un período prolongado. La fatiga diaria, el estrés o los cambios

emocionales son efectos a corto plazo que no se pueden evitar. A largo plazo, la salud y el cambio en las habilidades de escritura pueden afectar la cadencia de tipeo (Gunetti y Picardi, 2005).

En el presente capítulo se realizará un breve repaso de la literatura. La sección 2.1 define los distintos esquemas de autenticación, su forma de combinarlos y la diferencia entre identificación y validación. La sección 2.2 define conceptos básicos de la biometría y diferentes métricas que son comúnmente utilizados por la literatura. La sección 2.3 explica las formas existentes de modelizar emociones y los antecedentes en captura de las mismas; a continuación, la sección 2.4 explica las distintas formas de modelizar y atacar el problema de cadencia de tecleo.

Las siguientes secciones relevan los antecedentes de trabajos en cadencia de tecleo relacionados a la detección de impostores (sec. 2.5) y de emociones (sec. 2.6). La sección 2.7 repasa consideraciones, desafíos y problemas existentes en esta área.

2.1 Esquemas de autenticación

La autenticación es el acto de confirmar la verdad de un atributo de una sola pieza de datos declarada verdadera por una entidad. Aplicado a una persona, la autenticación es el mecanismo que corrobora que esta sea quien dice ser. Existen tres divisiones fundamentales de los esquemas de autenticación, estos dependen del medio utilizado (Kim *et al.*, 2013).

Por algo que se sabe: basado en el conocimiento, el factor de autenticación es una pieza de conocimiento que el usuario legítimo posee y se supone secreta, salvo tal vez para el sistema que autentica. La verificación de identidad más característica y extendida es por medio de códigos PIN, palabras o frases clave. También se encuentran los patrones de puntos para desbloqueo de teléfonos celulares.

Por algo que se posee: basado en la posesión, el factor de autenticación es un objeto físico único o con una cantidad de copias limitada, que se supone difícil o costoso de replicar. Las llaves son el ejemplo más común; además, tenemos documentos

de identificación, dispositivos RFID, tarjetas inteligentes, tarjetas de coordenadas, *tokens* criptográficos entre otros.

Por algo que se es: biométrico; el factor de autenticación es una característica o una combinación de características biológicas medibles que identifican con suficiente precisión al usuario legítimo, como ser las huellas dactilares, el patrón de vasos sanguíneos de la retina o su contorno facial. También se consideran factores de autenticación biométrica los patrones de comportamiento cuantificables, como la modulación de la voz, la forma de caminar o la cadencia de tecleo.

2.1.1 Esquemas de autenticación mixta o de múltiples factores

La búsqueda de niveles de seguridad más elevados y, a la vez, más sencillos en su empleo, ha llevado a la experimentación con esquemas de autenticación que mezclen seguridad basada en el conocimiento, en la posesión y por caracteres biométricos. Todas las combinaciones de esquemas simples han resultado en implementaciones prácticas que superan a sus combinados.

La utilización de teléfonos celulares como *tokens* de seguridad complementarios a las claves para la realización de transferencias monetarias en *home banking*, es un ejemplo de combinación eficaz de seguridad basada en el conocimiento y en la posesión. Asimismo, un *token* criptográfico cuya activación dependa de factores biométricos combina estos últimos con seguridad basada en la posesión. La tercera selección posible de los factores de autenticación pueden ser las claves endurecidas con una cadencia de tecleo del usuario.

Es fácil imaginar que, unificando todas las opciones anteriores, sea factible lograr un nivel de seguridad muy elevado. De un dispositivo llave cuya activación requiera uno o más caracteres biométricos y una clave secreta, se dice que utiliza autenticación de tres factores.

2.1.2 Identificación y verificación

El proceso de autenticación de usuarios puede realizarse en uno o varios pasos y, según dónde intervenga el sistema en consideración, se

puede hablar de un proceso de *identificación* o de *verificación* (Kim *et al.*, 2013). Se denomina *identificación*, al proceso de determinación de la identidad de un usuario, dentro de un conjunto previamente registrado en base a la información dada al momento de la autenticación. Para la *verificación*, se considera que el usuario ya ha sido identificado y el objetivo del proceso es determinar si él mismo es efectivamente quien se corresponde con las credenciales dadas, o si se trata de un impostor.

En forma equivalente, la identificación puede ser considerada un problema de clasificación de uno entre muchos, posiblemente con detección de valores anómalos, mientras que la verificación es un problema de clasificación con solo dos clases: usuario legítimo e impostores.

Los esquemas de autenticación más sencillos, como por ejemplo el control de acceso a computadoras portátiles por medio de huellas dactilares, combinan ambos procesos en un único paso. El tamaño reducido del conjunto de usuarios, la elevada precisión del método de verificación y los bajos requerimientos de seguridad hacen aceptable y práctico este último esquema. Sin embargo, conjuntos de usuarios más grandes, métodos menos precisos o mayores requerimientos de seguridad, exigen uno o más pasos tanto de identificación como de verificación, como es fácilmente observable en los sistemas bancarios y otros que manejen información sensible o valiosa.

A pesar de que es posible aplicarlo a la identificación de usuarios, el análisis de la cadencia de tecleo se aplica fundamentalmente a la verificación, teniendo como función comprobar que la identidad del usuario legítimo no está siendo usurpada o que él mismo no ha cedido su sesión a un usuario no autorizado o de menores privilegios. El motivo es simple: la eficiencia en la clasificación, a pesar de ser aceptable para complementar otros esquemas mejorando el nivel de seguridad total, no es suficiente por sí mismo para cumplir ambas funciones.

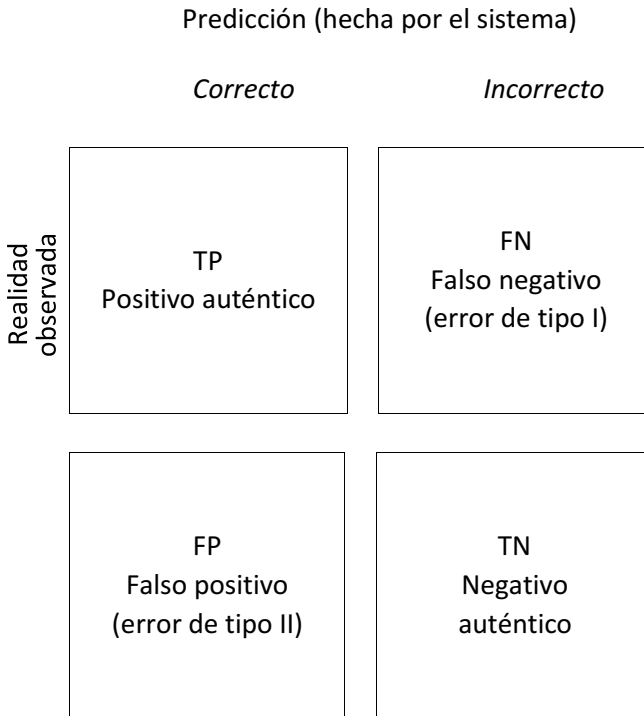


Figura 2.1: Matriz de confusión para las clasificaciones correcto e incorrecto.

2.2 Conceptos utilizados en biometría

2.2.1 La matriz de confusión

Los algoritmos de clasificación pueden obtener resultados de manera correcta o incorrecta. En particular, los algoritmos binarios donde hay solo dos clases generan —dependiendo de si erraron o acertaron prediciendo la realidad— cuatro combinaciones posibles; a saber: falsos positivos, falsos negativos, positivos auténticos y negativos auténticos. A estos resultados se los suele representar en la denominada *matriz de confusión* (ver Figura 2.1). Cuando la clase observada por un experimento y la predicha coinciden, estamos en presencia de un valor auténtico (la diagonal). Las clasificaciones incorrectas de usuarios legítimos e impostores se denominan respectivamente falsos negativos y falsos positivos o, en la terminología usual de la estadística, errores de tipo I y II.

Estos cuatro valores son utilizados luego para definir varias métricas y así calificar a los clasificadores. Es importante mencionar que los clasificadores pueden requerir de un cierto nivel de tolerancia para decidir si una muestra es positiva o negativa, en ese caso la matriz de confusión puede variar dependiendo de este valor de tolerancia (o, a veces llamado, sensibilidad) del clasificador.

2.2.2 Medidas derivadas

Las siguientes medidas se aplican de forma general a todos los sistemas de clasificación, salvo las que se especifiquen como propias de la biometría. En particular, los términos *False Acceptance Rate* FAR y *False rejection rate* (FRR) son aplicados en biometría, ya que explicitan las tasas de aceptados y rechazados, respectivamente; en otros sistemas de clasificación, se los denomina *false positive rate* y *false negative rate*. Todos los valores posteriores dependen de la tolerancia del algoritmo, ya que la misma matriz de confusión lo hace.

La tasa de fallos, *miss rate*, *false negative rate* (FNR) o, en biometría, tasa de rechazos falsos o *false rejection rate* (FRR) o menos común, *false non-match rate* (FNMR) es

$$FRR = \frac{FN}{FN + TP} = 1 - TPR \quad (2.1)$$

Su conjugado: la caída, *fall-out*, la tasa de falsos positivos, *false positive rate* (FPR) o, en biometría, tasa de aciertos falsos o *false acceptance rate* (FAR) o menos común, *false match rate* (FMR) es

$$FAR = \frac{FP}{FP + TN} = 1 - TNR \quad (2.2)$$

La sensibilidad (no confundir con la sensibilidad o tolerancia del clasificador), tasa de aciertos, *recall*, o *true positive rate* (TPR) se define como

$$TPR = \frac{TP}{TP + FN} = 1 - FRR \quad (2.3)$$

La especificidad, selectividad o *true negative rate* (TNR) es

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} = 1 - \text{FAR} \quad (2.4)$$

La precisión, valor predictivo positivo o *positive predictive value* (PPV) es

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.5)$$

Su conjudado, el valor predictivo negativo o *negative predictive value* (NPV) es

$$\text{NPV} = \frac{\text{TN}}{\text{TN} + \text{FN}} \quad (2.6)$$

La tasa de descubrimiento falso o *false discovery rate* (FDR) se define como

$$\text{FDR} = \frac{\text{FP}}{\text{FP} + \text{TP}} = 1 - \text{PPV} \quad (2.7)$$

La tasa de omisiones falsas o *false omission rate* (FOR) es

$$\text{FOR} = \frac{\text{FN}}{\text{FN} + \text{TN}} = 1 - \text{NPV} \quad (2.8)$$

El F1 score es el promedio armónico entre precisión y sensibilidad, o sea

$$F_1 = \frac{2\text{PPV} \times \text{TPR}}{\text{PPV} + \text{TPR}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \quad (2.9)$$

La exactitud o *accuracy* (ACC), se define como los valores auténticos (TP y TN) dividido el total, es decir

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.10)$$

Además, particularmente en biometría, Maxion y Killourhy (2010) proponen el costo ponderado de error o *weighted error rate* (WER) como

$$WER_w = w_{FAR}FAR + w_{FRR}FRR = w_{FAR} \frac{FP}{FP + TN} + w_{FRR} \frac{FN}{FN + TP} \quad (2.11)$$

donde w_{FAR} y w_{FRR} son pesos arbitrarios asignados a FAR y FRR. El caso particular donde $w_{FAR} = w_{FRR} = 1$ se define al costo de error no ponderado como

$$C_u = FAR + FRR = \frac{FP}{FP + TN} + \frac{FN}{FN + TP} \quad (2.12)$$

Por lo general, se exigen valores más bajos para la FAR que para la FRR, por lo que las ponderaciones resultan una buena solución.

2.2.3 False acceptance rate (FAR) y false rejection rate (FRR)

En la sección anterior, el FAR y el FRR corresponden a las tasas de errores de tipo I y de tipo II, respectivamente. Considerando como hipótesis nula la ausencia de correlación entre el modelo de la cadencia de tecleo del usuario y las características biométricas del intento realizado, el rechazo de la hipótesis es equivalente a la aceptación de un impostor e, inversamente, su aceptación equivale al rechazo de un usuario legítimo.

Para los esquemas de verificación de claves estáticas, es común que la magnitud de ambas métricas sea muy parecida; valores del 5% o menos son usualmente reportados en la literatura (Killourhy y Maxion, 2009). En los esquemas de autenticación continua, se suele exigir un FRR más bajo —generalmente logrado a costa de elevar significativamente el FAR—, ya que la mayor complejidad de los

modelos, la mayor desviación en los parámetros de tecleo durante una sesión de usuario o entre sesiones, pero sobre todo la cantidad de evaluaciones realizadas durante una sesión, hacen que valores de FRR tolerables en la verificación de claves estáticas vuelvan imposible prolongar una sesión con autenticación continua.

Como ejemplo de esta última afirmación, consideremos la experiencia de usuario de un sistema de verificación estática de claves con FAR y FRR del orden de 5%. En promedio, uno de cada veinte intentos de un usuario legítimo es rechazado por el sistema, mientras que un impostor que haya logrado conocimiento de una clave válida y no sepa imitar la cadencia de tecleo del usuario legítimo será rechazado veinte veces —cantidad suficiente para activar otros mecanismos de protección— antes de ser aceptado como falso positivo. La situación descrita es deseable y muestra, una clara mejora sobre la autenticación basada exclusivamente en el conocimiento. Por el contrario, el mismo FRR en un sistema de autenticación continua puntuada por oraciones generaría un cierre de sesión o un alerta de seguridad cada veinte oraciones, tornándolo inutilizable (González, 2016).

2.2.4 Equal error rate (EER)

Los valores representados previamente siempre dependen de una tolerancia o sensibilidad, y muchos pueden representarse en dos dimensiones (FAR y FRR o combinaciones lineales de las mismas). Ahora, es posible que se desee medir un valor único, un balance entre ambas y, en ese contexto, se define al *equal error rate* (EER) como un balance entre ambas. Dado que la matriz de confusión depende de la sensibilidad del clasificador, es de esperar que exista un punto medio donde se logre igualar FAR y FRR y ese punto se denomina EER y tendrá una tolerancia o sensibilidad asociada.

Es razonable esperar que cuanto más baja sea la sensibilidad del sistema, más probabilidades haya de confundir al impostor con el usuario legítimo y, a la inversa, cuanto más sensible sea el usuario legítimo, más frecuentemente se rechazará (González, 2016). La Figura 2.2 refleja un ejemplo posible del cruce de FAR y FRR.

El nivel en el cual ambas curvas se cortan se denomina ERR (*Equal*

Error Rate, tasa de errores idénticos o de cruce de errores) y es un indicador de la calidad del sistema biométrico. Un valor más bajo de ERR se debe necesariamente a curvas de FAR y FRR más aplastadas o, lo que es lo mismo, a tasas más bajas de falsos positivos y negativos para una sensibilidad dada.

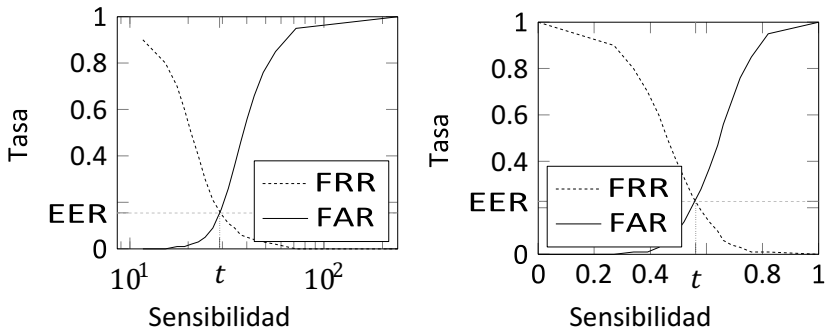


Figura 2.2: Ejemplos FAR y EER de diagrama FAR y FRR vs. sensibilidad.

2.2.5 Curvas *receiver operating characteristic* (ROC) y *detection error tradeoff* (DET)

Para evitar cualquier referencia al parámetro de sensibilidad, que no siempre puede hacerse explícito, pero sin perder la información proporcionada por las curvas FAR y FRR, podemos trazar una curva con una en función a la otra. Esta curva se denomina DET (del inglés *Detection Error Tradeoff*), que suele graficarse con escalas logarítmicas, mientras que la curva ROC (del inglés *Receiver Operating Characteristic*) surge de graficar la tasa de positivos auténticos (TPR) — que según 2.3 es la inversa de la FRR — también en función del FAR. Las curvas ROC y DET correspondientes al sistema de la figura 2.2a, pueden ser observadas en la figura 2.3.

Dado que una asignación completamente aleatoria de clases produce una tasa de falsos positivos igual a la tasa de positivos auténticos, la distancia a la diagonal de la gráfica de la curva ROC es una medida de la calidad del clasificador para una sensibilidad o tasa

de error dada. Hasta la diagonal se puede considerar que el algoritmo utilizado es mejor que el azar, con un límite en la esquina superior

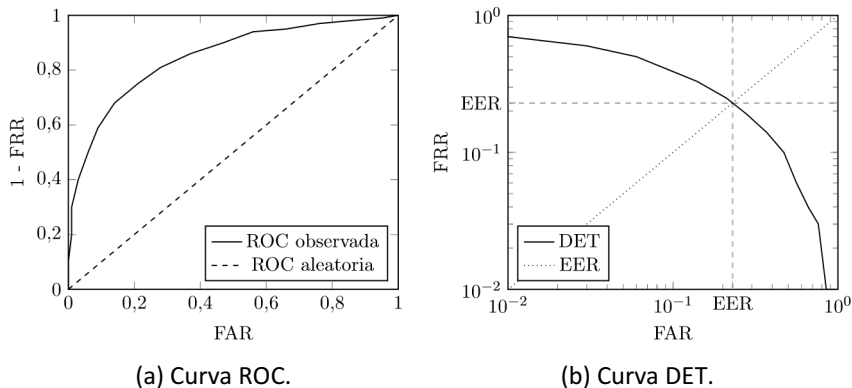


Figura 2.3: Ejemplos de curvas ROC y DET.

izquierda del gráfico, donde el algoritmo perfecto es capaz de clasificar correctamente a todos los usuarios legítimos, sin autorizar ningún impostor. Esta es la razón por la que el área bajo la curva ROC, donde los valores mayores implican una mejor calidad, permite evaluar un clasificador en todo su rango de sensibilidad.

Un punto debajo de la diagonal implica que el algoritmo de clasificación es peor que una elección aleatoria. En la práctica, tales valores no se observan ya que, de existir, solo invirtiendo la clase asignada por ellos obtendríamos un mejor clasificador que el clasificador aleatorio.

El concepto de la curva DET es similar pero tiene la ventaja de que la sección importante de la curva ocupa un área más grande y, por lo tanto, permite una visualización más precisa. En este caso, la diagonal corresponde a las mismas tasas de falsos positivos y negativos, que corresponde al concepto de ERR descrito anteriormente (González, 2014b).

2.3 Modelado y captura de emociones

Durante el último medio siglo, los investigadores han propuesto diferentes formas de modelar las emociones. Representaban emociones desde diferentes perspectivas. Dos de ellas son ampliamente utilizados por la literatura, a saber, el enfoque categórico y el enfoque dimensional (Gunes *et al.*, 2011). El enfoque categórico fue desarrollado inicialmente por el psicólogo Paul Ekman (Ekman, 1994; Ekman y Friesen, 2003; Ekman y O’Sullivan, 1988), quien afirmó que había un conjunto de seis emociones básicas y universales que no están determinadas por las culturas. A este conjunto lo conforman la alegría, el miedo, la tristeza, la ira, el disgusto y la sorpresa. Después de estudiar culturas de acceso limitado a la civilización y algunas especies de primates, postuló que estas seis emociones están arraigadas en nuestra biología y que no necesitan ser aprendidas; ya las poseemos desde el nacimiento. También analiza las emociones secundarias, que se consideran una combinación de las primarias y estas sí pueden ser el producto de un factor cultural (Plutchik, 2000).

Por otro lado, el enfoque dimensional establece que los estados afectivos se distribuyen en un espacio continuo cuyos ejes dimensionales indican una cuantificación de una característica. Varios modelos dimensionales están relacionados entre sí de manera sistemática (Grandjean *et al.*, 2008; Mehrabian, 1996; Russell, 1980; Scherer *et al.*, 2001). Uno de los modelos más aceptados es el *Circumplex of Affectivity* o *Circumplejo of Afectividad* de James Russell (Posner *et al.*, 2005; Russell, 1980), también conocido como *circumplejo* o modelo Arousal–Valence. Este modelo es bidimensional siendo los ejes Excitación o *Arousal* (relajado vs. excitado) y Valencia o *Valence* (placer vs. disgusto). La figura 2.4a ejemplifica la ubicación de ocho emociones usando este modelo (Gunes *et al.*, 2011); la figura 2.4b muestra las fases que emplearemos para nuestra investigación y sus respectivos nombres: fase 1, activación alta con valencia positiva (HAPV); fase 2, activación baja con valencia positiva (LAPV); fase 3, activación baja con valencia negativa (LANV); y fase 4, activación alta con valencia negativa (HANV).

Otra descripción dimensional aceptada y utilizada comúnmente es el espacio emocional tridimensional de placer–disgusto, excitación–no excitación y dominio–sumisión (Mehrabian, 1996), en ocasiones

haciéndose referencia al mismo como el espacio emocional PAD¹ (Jia *et al.*, 2011) o como primitivas emocionales (Pérez Espinosa *et al.*, 2010). Para garantizar una descripción más completa de la coloración afectiva, algunos investigadores incluyen la expectativa (el grado de anticipación o de consciencia adquirida) como la cuarta Arousal

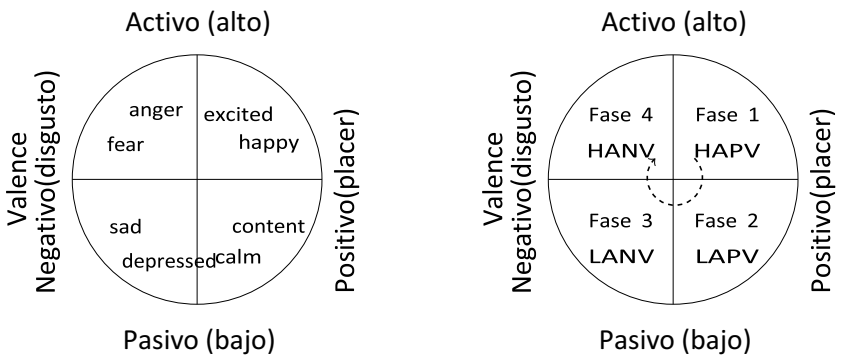


Figura 2.4: Modelo Arousal-Valence.

Dimensión (Fontaine *et al.*, 2007), y la intensidad (hasta qué punto una persona está alejada de un estado de racionalidad puro) como la quinta dimensión (por ejemplo, McKeown *et al.* [2010]). Sherer y sus colegas introdujeron otro conjunto de modelos psicológicos, haciendo referencia a estos como modelos componenciales de emoción, los cuales están basados en la teoría de evaluación (*appraisal theory* en inglés) (Fontaine *et al.*, 2007; Grandjean *et al.*, 2008; Scherer *et al.*, 2001). En este enfoque valorativo, las emociones son generadas a través de una continua evaluación recursiva subjetiva de nuestro estado interno y del estado del mundo externo (preocupaciones/necesidades relevantes) (Fontaine *et al.*, 2007; Frijda, 1986; Grandjean *et al.*, 2008; Scherer *et al.*, 2001). A pesar de los esfuerzos precursores de Scherer y sus colegas (por ejemplo, Sander *et al.*, 2005), cómo utilizar el enfoque basado en la evaluación para la medición automática de la afectividad es una pregunta de investigación aún abierta ya que, este enfoque requiere mediciones de

¹ Por sus siglas en inglés de *Pleasure* (placer), *Arousal* (excitación) y *Dominance* (dominancia).

cambio complejas, multicomponenciales y sofisticadas. Una posibilidad es reducir los modelos de evaluación a modelos dimensionales (por ejemplo, el espacio bidimensional de excitación–valencia). Ortony y sus colegas han propuesto un modelo computacionalmente abordable para las bases cognitivas de la obtención de emociones, conocido como OCC (Ortony *et al.*, 1990). OCC ha sido establecido como un modelo estándar (evaluación cognitiva) para las emociones. A pesar de la existencia de diversos modelos afectivos, la búsqueda de representaciones óptimas, de dimensiones mínimas, de la afectividad para análisis y síntesis, permanece abierta (Fontaine *et al.*, 2007).

2.3.1 Modelo de Excitación–Valencia

A partir de los primeros estudios sobre estados de ánimo de Wundt (1896), se ha reconocido que las diferencias emocionales entre estímulos —palabras, objetos, acontecimientos— pueden ser descriptos sucintamente por unas pocas dimensiones afectivas (Bradley y Lang, 2006).

En concordancia con lo anterior, investigaciones y trabajos sobre la categorización del lenguaje natural (Ortony *et al.*, 1990; Shaver *et al.*, 1987) sugieren que el conocimiento de las personas sobre las emociones se encuentra organizado jerárquicamente y que la división de orden superior se da entre la positividad (estados placenteros: amor y alegría) y la negatividad (estados no placenteros: ira, tristeza y miedo) (Lang, 1995).

Las investigaciones sobre los estados anímicos han producido un consenso razonable en la estructura más general de la experiencia afectiva: un orden circunflejo, o circular, de estímulos a lo largo de dos dimensiones (Larsen y Cutler, 1992, 1996; Russell, 1980; Schlosberg, 1941; Watson y Tellegen, 1985). Esta estructura abstracta especifica las relaciones entre las clasificaciones de los estímulos afectivos (es decir, clasificaciones de expresiones faciales, semánticas de palabras relacionadas con la afectividad, autoinformadas² del estado de ánimo;

² Estudios autoinformados hacen referencia a *self-report studies*, un tipo de encuesta donde quien responde lee la pregunta y elige la respuesta por sí mismo sin la intervención del investigador.

para más detalles véase Russell [1980]). Sus dimensiones representan atributos que capturan la varianza compartida entre los estímulos, es decir, describen los estados afectivos. Cualesquiera de estos últimos pueden ser definidos por su posicionamiento relativo a las dimensiones circumplejas (Feldman, 1995).

Aunque persisten, entre distintos investigadores, diferencias en la rotación y en el etiquetado de las dimensiones afectivas (por ejemplo,

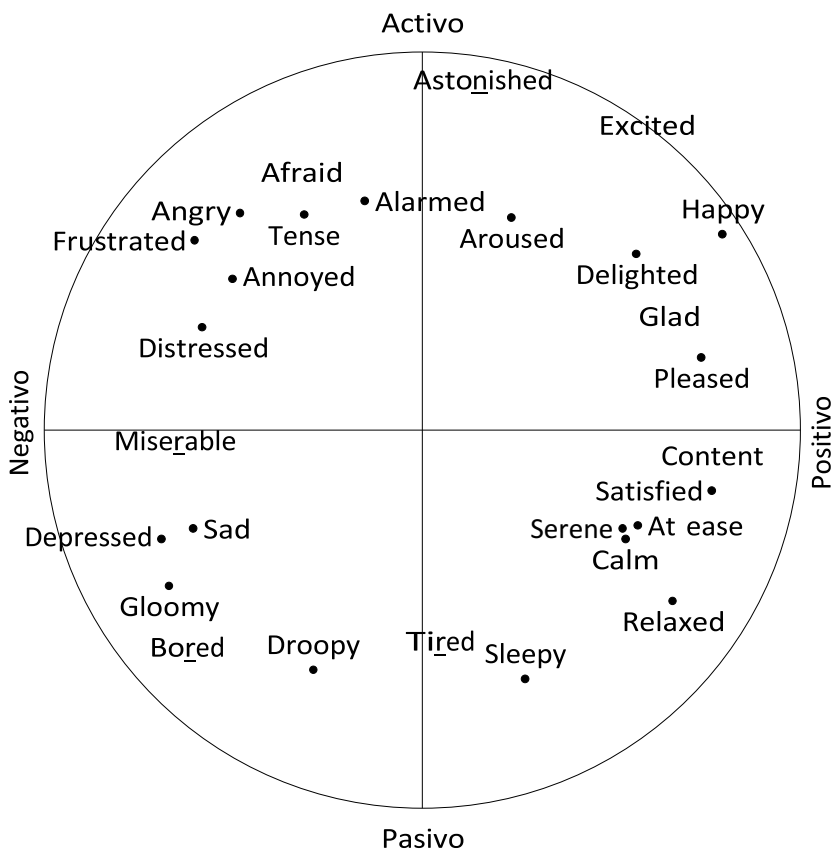


Figura 2.5: Modelo de excitación-valencia con algunas emociones en inglés según Russell.

Diener *et al.*, 1985; Larsen y Cutler, 1992; Thayer, 1990; Watson y Tellegen, 1985), todas las dimensiones pueden ser definidas como combinaciones de las dimensiones originales del modelo de

excitación–valencia (por ejemplo, Reizenzein, 1994; Russell, 1980; Schlosberg, 1954). La dimensión de valencia se refiere a la calidad hedónica o placentera de una experiencia afectiva. La dimensión de excitación hace referencia a la percepción de excitación asociada a tal experiencia (Russell, 1989). En la figura 2.5 se muestra una de las tantas representaciones de este modelo (Feldman, 1995).

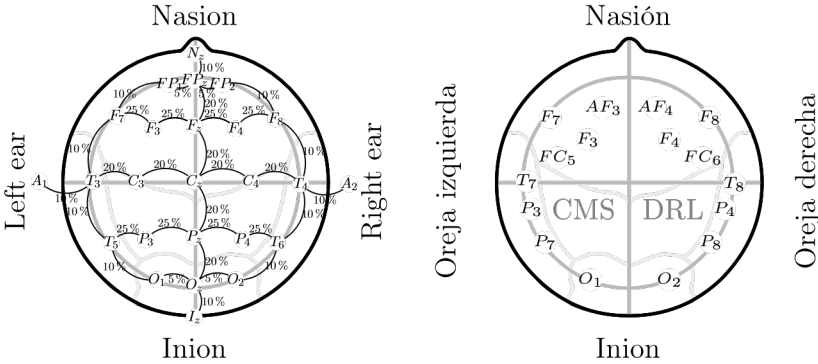


Figura 2.6: Ubicación de los electrodos de EEG.

2.3.2 Adquisición por interfaz cerebro–computador (BCI)

Una interfaz cerebro–computadora —BCI por sus siglas en inglés (*brain–computer interface*)—, permite a una persona comunicarse o controlar el mundo externo sin necesidad de utilizar las vías de salida normales del cerebro, de los nervios periféricos o los músculos. Los mensajes y los comandos se expresan no por contracciones musculares, sino más bien por fenómenos electrofisiológicos como características EEG evocados o espontáneos (por ejemplo, SCP, P300, ritmos mu/beta) o la actividad neuronal cortical. La operación BCI depende de la interacción de dos controladores adaptativos, el usuario (quien debe mantener estrecha correlación entre su intención y estos fenómenos) y el BCI (el que debe traducir los fenómenos en comandos de dispositivos que interpreten la intención del usuario).

Hoy en día, las BCI tienen tasas máximas de transferencia de información de 25 bits/min. Con esta capacidad, pueden proporcionar las funciones de comunicación y de control básicas (por ejemplo,

controles ambientales, un procesamiento de textos sencillo, etc.) a personas con discapacidades neuromusculares más graves, como la Esclerosis Lateral Amiotrófica (ELA) en etapa tardía o un derrame cerebral. También podrían controlar una neuroprótesis, que proporcione agarre de la mano a las personas con lesiones de la médula espinal cervical de nivel medio.

Aplicaciones más complejas útiles para población más grande de usuarios dependen del logro de una mayor velocidad y precisión, es decir, el aumento de las tasas de transferencia de información. El progreso a futuro depende de la atención a una serie de factores cruciales. Estos incluyen: el reconocimiento de que el desarrollo de tecnologías BCI es un problema interdisciplinario, involucrando a la neurobiología, la psicología, la ingeniería, la matemática, la informática y la rehabilitación clínica; identificación de las características de la señal, si los potenciales evocados, ritmos espontáneos, o las tasas de disparo neuronal, que los usuarios son más capaces de controlar; la medida en que este control puede ser independiente de la actividad en la salida del motor convencional y los canales de entrada sensoriales; la medida en que este control depende de la función normal del cerebro; la identificación de los mejores métodos de extracción de características y los mejores algoritmos para la traducción de estas características en los comandos de control del dispositivo; desarrollo de métodos para maximizar el control de cada usuario de estas características de la señal; atención a la identificación y eliminación de artefactos tales como EMG y la actividad EOG; adopción de procedimientos precisos y objetivos para evaluar el rendimiento de BCI; el reconocimiento de la necesidad de que a largo plazo, así como la evaluación a corto plazo del rendimiento; identificación de aplicaciones apropiadas; una correcta adecuación de las aplicaciones BCI y usuarios; cerrar atención a los factores que determinan la aceptación de los usuarios de la tecnología aumentativa; y el énfasis en las publicaciones con referato y la respuesta adecuada a la atención de los medios conservadores.

Con el reconocimiento adecuado y la participación efectiva de estas cuestiones, los sistemas BCI podrían proporcionar una nueva opción de comunicación y control importante para las personas con discapacidad que deterioran los canales de comunicación y de control normales. También, podrían proporcionar a las personas sin discapacidad un canal de control suplementario o un canal de control útil en circunstancias especiales (Wolpaw y McFarland, 2004; Wolpaw *et al.*, 1991, 2002).

2.3.2.1 Emotiv EPOC

Una de las posibles formas de lectura de la actividad bioeléctrica cerebral es mediante el uso del neuro-auricular³ *Emotiv EPOC*, un dispositivo de bajo costo compuesto por 14 electrodos más 2 electrodos de referencia (utilizados además para reducir el ruido), preparado para ser colocado sobre la cabeza. Los electrodos se encuentran dispuestos sobre el cuero cabelludo en posiciones basadas en el *Sistema Internacional 10–20* (ver Figura 2.6a) (Yurci, 2014). En este sistema, las distancias entre electrodos suelen estar a un 10% o un 20% del tamaño de la cabeza. La configuración de electrodos que utiliza EPOC, incluye electrodos intermedios (ver Figura 2.6b).

El Emotiv EPOC fue desarrollado originalmente para su uso en videojuegos y en ambientes virtuales, utilizándose para controlar varios aspectos de los entornos de juego. En los últimos años, el EPOC se ha utilizado cada vez más en diversas áreas de investigación. El dispositivo es capaz de detectar pensamientos, sentimientos y expresiones de los usuarios mediante la captura de las ondas cerebrales producidas por estos. En particular, para la detección de emociones, se suelen utilizar las señales provenientes de los electrodos ubicados en la zona del lóbulo prefrontal (figura 2.6b), área en donde se centra el control de las emociones. Las señales recibidas por los electrodos son transferidas a una computadora a través de un receptor USB inalámbrico. Por otra parte, se requiere que los electrodos del Emotiv se encuentren ligeramente humedecidos con una solución conductora para establecer mayor contacto con el cuero cabelludo (Harrison, 2013).

Los datos son constantemente recolectados por el EPOC. Tras ser recogidos, son convertidos a datos digitales, que luego se transmiten de forma inalámbrica al receptor USB (Harrison, 2013). Al arribar a la computadora, es posible procesar la información utilizando las herramientas de software provistas por Emotiv EPOC o haciendo uso de otros softwares de código abierto compatibles con el dispositivo (p. ej. OpenVibe).

³ El término utilizado en inglés es *headset*, que no puede traducirse al español y denota a cualquier equipo electrónico utilizable en la cabeza. Por lo general se traduce este término como auricular o audífono, pero en realidad es mucho más abarcativo y puede incluir un equipo de EEG. Para este último caso, lo referiremos en español como neuroauricular

Estudios previos, han encontrado que el Emotiv EPOC presenta algunas limitaciones. Un ejemplo de ello es que, el tiempo de preparación puede ser muy largo particularmente en personas cuyo cabello es largo o grueso. Algunos participantes, presentaron incomodidad al tenerlo colocado (Mampusti *et al.*, 2011). También se ha encontrado por Mampusti *et al.* (2011) que el EPOC puede ser muy sensible al ruido, pudiendo introducirse este por una señal débil o a través de la implantación de datos recolectados por el EPOC. El ruido también puede ser introducido a través de amplificadores electrónicos, interferencias de líneas eléctricas o interferencias externas (Lievesley *et al.*, 2011) citado en Mampusti *et al.* (2011).

2.3.2.2 Trabajos Académicos con BCI

En la mayoría de los trabajos, se utilizan diferentes videos para estimular. Por ejemplo Samara *et al.* (2016), presentando videos a los participantes, probó varias configuraciones y logró, en la mejor configuración, una tasa de éxito para la valencia de 85,99% y 73,67% para la excitación mediante la clasificación por SVM. También, estimulando con videos y utilizando el circunplejo de Russell, Li *et al.* (2016) obtiene 72,06% para la valencia y 74,12 para excitación con un clasificador de *Long short-term memory* (LSTM); Liu *et al.* (2016) obtiene 69,9% para valencia y 71.2% para excitación con k vecinos más cercanos (k-NN) y bosques aleatorios. Garcia *et al.* (2016) logra un 88,33% para valencia y 90,56% para excitación clasificando (en solo tres clases por dimensión) con SVM una adquisición multimodal que además incluye electromiogramas (EMG), electrooculogramas (EOG), entre otros canales. Zhang *et al.* (2016), también utilizando videos y el modelo de Russell logra una tasa del 81,76% con redes neuronales probabilísticas y 82,00% con SVM. Los videos de la estimulación fueron tomados de la base de datos SEED (Duan *et al.*, 2013; Zheng y Lu, 2015). En esta misma línea, Zheng *et al.* (2017) utiliza machine learning para clasificar en los 4 cuadrantes de Russell obteniendo un 69,67% en la base de datos DEAP (Koelstra *et al.*, 2012) y 91,07% con SEED. Fuera del modelo de Russell, estimulando también con videos, Verma y Tiwary (2014) utilizan 8 canales de EEG y otras señales periféricas y logra una tasa de aciertos del 81% solamente con EEG y un 78% con el resto de las señales. Como clasificadores, utilizaron SVM, perceptrón multicapa, k-NN y *Maximum Margin Classifier* (MMC) con las siguientes emociones: terrible, amor, odio, sentimental, encantador,

feliz, divertido, conmoción, alegre, deprimente, emocionante, melancólico. Shin *et al.* (2016) reporta 98,06% con redes bayesianas en las emociones: diversión, miedo, tristeza, alegría, ira y disgusto.

Estimulando con imágenes, específicamente con la base de datos IAPS (Lang *et al.*, 2008), Petrantonakis y Hadjileontiadis (2010) publica la separación de las emociones básicas de Ekman; felicidad, sorpresa, ira, miedo, disgusto y tristeza con una tasa de 85,17% mediante k-NN, análisis discriminante cuadrático (*Quadratic discriminant analysis* en inglés, QDA), distancia de Mahalanobis y SVM. Lan *et al.* (2016) estimula con los sonidos de IADS (Bradley y Lang, 2007b) y clasifica con SVM en emociones positivas y negativas con una tasa de éxito del 73,10%.

El experimento de Ierache *et al.* (2015), realizado en el marco del ISIER de la Universidad de Morón, evalúa el estado emocional utilizando Emotiv EPOC estimulándose con un llamado a la policía. Se encuentran grandes correlaciones de excitación en los momentos más tensos del llamado.

Respecto a otras aplicaciones fuera del contexto afectivo realizadas por el ISIER, se encuentran Ierache *et al.* (2013^a), donde se evalúa la posibilidad de realizar control domótico mediante BCI, en Ierache *et al.* (2013c) se investiga sobre el control de robots y, en Ierache *et al.* (2014) y Pereira *et al.* (2011), se logra controlar y dirigir el movimiento de un robot únicamente mediante una interfaz BCI. Pereira *et al.* (2015) detalla estas y futuras líneas de trabajo.

2.3.3 Evaluación SAM (*Self-Assessment Manikin*)

Una herramienta de encuesta y/o evaluación ampliamente utilizado para capturar emociones es SAM (por sus siglas en inglés *SelfAssessment Manikin*) propuesto por Lang (1980).

Tal como lo describen sus autores en Bradley y Lang (1994), SAM es un método no verbal fácil de administrar para evaluar rápidamente el placer, la excitación y la dominancia asociadas con la reacción emocional de una persona ante un evento. El discernimiento con respecto a la cantidad de placer y excitación experimentadas durante la visualización de una imagen usando SAM se correlaciona altamente con las puntuaciones obtenidas utilizando la más larga escala diferencial semántica verbal. Las diferencias obtenidas en los juicios de dominancia sugieren que SAM podría ser más preciso en el seguimiento de los sentimientos de control del sujeto —en lugar los estímulos—. Debido a

estas propiedades, SAM es un instrumento útil para determinar la experiencia subjetiva de la emoción asociada con el procesamiento de la mayoría de los estímulos, y puede emplearse a una gran variedad de poblaciones, incluyendo personas que no hablan inglés, niños, personas con trastornos del lenguaje y, por supuesto, todos los síndromes clínicos. Finalmente, SAM permite una evaluación rápida de lo que parecen ser las dimensiones fundamentales en la organización de la experiencia emocional humana, así como una evaluación de la forma del espacio afectivo resultante. La covariación entre los informes de placer y excitación utilizando SAM y las respuestas en los sistemas fisiológicos y de comportamiento han demostrado ya una fuerte concordancia. Tomados en conjunto, estos datos indican que SAM es un método eficaz para medir los estados sentimentales existentes, relacionándolos con otros índices de respuesta emocional, y evaluando los cambios debidos al paso del tiempo, por intervención terapéutica, y otros procesos que afectan a las reacciones afectivas ante estímulos contextuales.

SAM se implementó originalmente como un programa informático interactivo, y más tarde fue ampliado para incluir una versión en papel y lápiz para su uso en grupos y evaluaciones en masa. La figura D.3 muestra la versión en papel y lápiz de SAM, ilustrando su representación no verbal, con gráficos de diversos puntos a lo largo de cada una de las tres grandes dimensiones afectivas. SAM se extiende desde una figura sonriente y feliz hacia una figura con ceño fruncido e infeliz cuando se representa la dimensión de placer, y se extiende desde una figura excitada con los ojos abiertos hacia una figura relajada con sueño para la dimensión de excitación. La dimensión de dominancia representa cambios en el control, con variaciones en el tamaño de SAM: una gran figura indica el máximo control de la situación (Bradley y Lang, 1994).

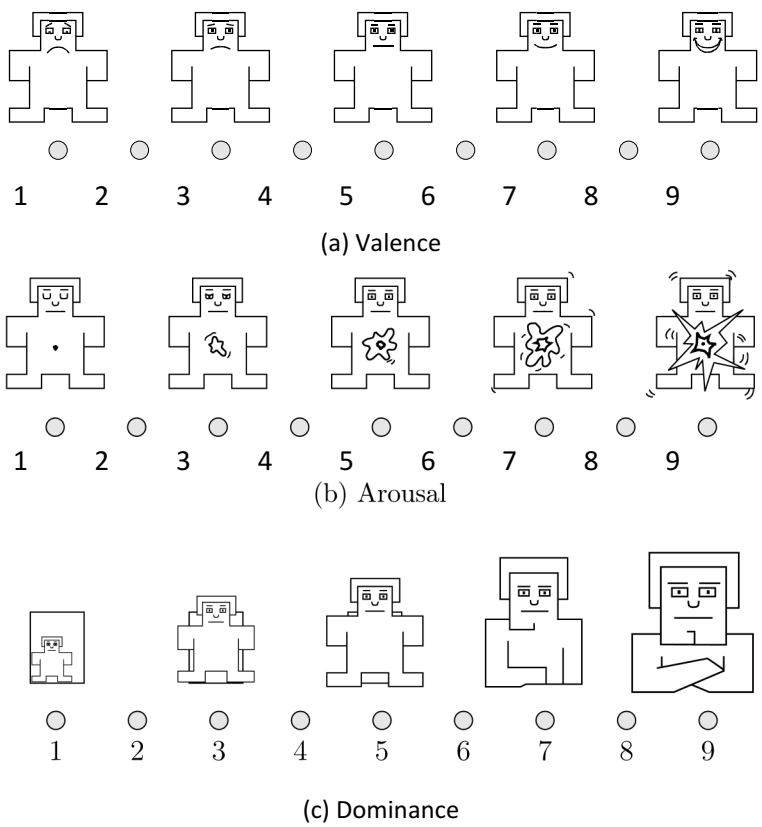


Figura 2.7: SAM y sus distintos estados según cada dimensión.

Los valores dimensionales provistos por SAM no necesitan ser transformados. Existen dos escalas, una en el rango 1–5, que coincide con las cinco representaciones del personaje de izquierda (1) a derecha (5). La segunda escala permite agregar cuatro valores intermedios, de esta forma se obtiene un rango posible de 1–9; en este caso, no se suele dibujar al personaje nueve veces sino que se dibuja el punto posible en el medio. Sam puede ser utilizado en papel y permitir el llenado de las casillas o bien en una interfaz por computadora, clickeando en un botón radial de selección. Se prefiere el marcado con una cruz en papel o el botón radial en una computadora al ingreso del número correspondiente, de esta forma los niños no necesitan saber escribir para completar los autoinformes. El valor recuperado, si bien el participante no lo puede ver de manera directa, se deduce de observar el círculo marcado.

2.3.3.1 Trabajos Académicos con SAM

SAM ha sido utilizado eficazmente para medir las respuestas emocionales en una gran variedad de situaciones, incluyendo reacciones a fotos (Greenwald *et al.*, 1989; Lang *et al.*, 1993), imágenes (Miller *et al.*, 1987), sonidos (Bradley, 1994), publicidades (Morris *et al.*, 1992), estímulos dolorosos (McNeil y Brunetti, 1992), entre otros. Además, SAM ha sido aplicado a niños (Greenbaum *et al.*, 1990), pacientes con trastornos de ansiedad (Cook *et al.*, 1988), personas con fobias análogas (Hamm *et al.*, 1991), psicópatas (Patrick *et al.*, 1993), y otros estados clínicos (Bradley y Lang, 1994).

2.3.4 Otros canales de adquisición

Además de los ya mencionados, existen diversos canales desde donde adquirir información que derive en el estado afectivo de la persona. Algunos ejemplos de estos son los gestos faciales (vía video), reconocimiento del timbre de la voz, la conductividad o respuesta electrodermida/galvánica de la piel, electrocardiografía (ECG), la frecuencia cardíaca (no confundir con ECG), electromiografía (EMG), frecuencia respiratoria, temperatura de la piel.

2.3.4.1 Trabajos académicos con otros canales

Valenza *et al.* (2012) analizó electrocardiogramas, la respuesta electrodermida y respiratoria para clasificar en el modelo de Russell utilizando análisis discriminante cuadrático (QDA). Estimuló a los usuarios con imágenes de IAPS y obtuvo una tasa de éxito de más del 90%. Siguiendo esta línea, en Valenza *et al.* (2014), adquiere ECG también utilizando IAPS para estimular y obtiene resultados de clasificaciones de 79,15% para valencia y 83,55% para excitación utilizando SVM. En Basu *et al.* (2015), nuevamente estimulando con IAPS pero utilizando la respuesta galvánica de la piel, frecuencia cardíaca, frecuencia respiratoria y temperatura de la piel para las adquisiciones reportaron clasificaciones en los cuatro cuadrantes del 98% para HAPV, 96% para LAPV, 93% para HANV y 97% para LANV mediante la aplicación de análisis de discriminante lineal y cuadrático. Si el lector desea profundizar en estos aspectos, Shu *et al.* (2018) proveen un relevamiento muy preciso de la literatura.

Con respecto a la adquisición de estados afectivos a través de medios audiovisuales (Wang y Guan, 2008), en los últimos años una gran cantidad de estudios han sido realizados en el reconocimiento de emociones humanas por parte de computadoras. La mayoría de estos trabajos se enfocan únicamente en el reconocimiento de voz, o bien de la expresión facial. Como se muestra en De Silva *et al.* (1997), algunas de las emociones se detectan mejor en el audio, mientras que otras en el video. Cuando una modalidad falla o no es lo suficientemente buena como para determinar una cierta emoción, la otra modalidad puede ayudar a mejorar el rendimiento del reconocimiento. La integración de los datos de audio y video proveerá más información sobre el estado emocional humano. La relación de complementariedad de estas dos modalidades ayudará a lograr una mayor precisión. Recientemente, los métodos de reconocimiento de emociones basados en datos audiovisuales comenzaron a llamar la atención de la comunidad científica. Song *et al.* (2004a,b) utilizaron el tono y la energía como características del audio, y utilizaron el movimiento de las cejas, párpados y mejillas como características de la expresión, mientras que la de los labios y la mandíbula como características visibles del habla.

Estas características audiovisuales extraídas de estos tres canales audiovisuales fueron fusionadas en un triple modelo oculto de Márkov (HMM) para su clasificación. El sistema de clasificación propuesto fue probado para un problema de siete clases (sorpresa, ira, alegría, tristeza, disgusto, miedo, neutro) y se afirmó que obtuvo una tasa de reconocimiento en torno al 85%. De Silva y Ng (2000) construyeron un sistema de audio y video por separado. En el sistema de audio, el tono se extrajo como característica y se utilizó un método k-NN para la clasificación. En el sistema de video, los investigadores siguieron el movimiento del contorno de los labios, comisuras de la boca y las cejas mediante el uso de un algoritmo de flujo óptico, mientras que el HMM fue entrenado como el clasificador. Se adoptó un sistema basado en reglas para fusionar los resultados de la clasificación de audio y video.

Su sistema clasifica las seis principales emociones y logró una tasa de reconocimiento del 72%, sin embargo, esto fue en una base de datos que contiene solo dos sujetos. Chen *et al.* (1998) utilizaron estadísticas del contorno del tono, la envolvente de la señal y sus derivados para representar las características de expresión emocional, y la información visual se obtuvo mediante el seguimiento de la posición de la ceja, levantamiento de la mejilla y apertura de la boca.

Al utilizar el clasificador k-NN, su método produjo un 97,2% de precisión en la clasificación de las seis emociones principales. Sin embargo, el conjunto de datos empleado solo contiene también dos voces. Chen *et al.* (2005) realizaron el análisis de la expresión facial mediante el seguimiento del movimiento de los ojos, las cejas, los labios y surcos. El contorno de tono, la curva de intensidad y el espectro de energía fueron analizados para la representación de las características acústicas. La extracción de características acústicas y faciales fueron combinadas para ser introducidas en una máquina de vectores de soporte (SVM) para el reconocimiento de las emociones. Se clasificaron las seis emociones principales (con el estado neutro) en un conjunto de datos que alcanzaron un promedio de 84% de precisión. Hoch *et al.* (2005) introdujo un sistema de reconocimiento de la emoción, dependiente de la persona para aplicarlo en un entorno de conducción en el tráfico. Las señales acústicas fueron reconocidas por un método de redes neuronales en las características de bajo nivel como el tono, potencia, formantes y la duración de segmentos sonoros. Las expresiones faciales fueron evaluadas por un clasificador SVM en regiones de la cara filtradas con Gabor.

La fusión bimodal se llevó a cabo en el nivel de decisión a través de una combinación lineal ponderada. Su sistema clasifica tres emociones (positivas, negativas, neutras) con un promedio de tasa de reconocimiento del 90,7%. Zeng *et al.* (2004) extrajo la energía, la tasa de sílabas y el tono como características prosódicas y trazó un conjunto predefinido de unidades de acción para la representación de la expresión facial. Se aplicó una técnica de alisamiento para reducir la influencia del habla en las expresiones faciales. La fusión bimodal se realizó a nivel de toma de decisión, utilizando un método de votación. Fue realizada, en otro trabajo (Zeng *et al.*, 2005), una extracción de características similar a la de Zeng *et al.* (2004), mientras que en este se utilizó un enfoque basado en HMM para fusionar los flujos de audio y video. Este enfoque produjo un reconocimiento del 85,24%.

2.3.5 Inducción de emociones

Muchas son las técnicas que han sido utilizadas en el pasado para inducir emociones, siendo los videos y las imágenes dos de las más comúnmente utilizadas. El Sistema Internacional de Imágenes Afectivas (*IAPS* por sus siglas en inglés de *Internacional Affective Picture*

System) es un sistema de gran importancia, ya que actúa como estímulos emocionales aplicados a la investigación experimental de la emoción y la atención (Harrison, 2013).

IAPS fue desarrollado para proporcionar valoraciones de afectividad a un gran conjunto de fotografías a color emocionalmente evocadoras e internacionalmente accesibles, incluyendo contenidos con una amplia gama de categorías semánticas. El IAPS, junto con el Sistema Internacional de Sonido Afectivo Digitalizado (*IADS* por sus siglas en inglés de *International Affective Digitized Sound system*) (Bradley y Lang, 2007b), las Normas Afectivas para Palabras en Inglés (*ANEW* por sus siglas en inglés de *Affective Norms for English Words*), así como otras colecciones de estímulos afectivos, están siendo desarrolladas y distribuidas por el Centro de Emoción y Atención NIMH (CSEA) en la Universidad de Florida, con el fin de proporcionar materiales normalizados disponibles para los investigadores en el estudio de la emoción y la atención. La existencia de estas colecciones de estímulos afectivos categorizados normativamente deben: 1) permitir un mejor control experimental en la selección de estímulos emocionales, 2) facilitar la comparación de resultados entre distintos estudios conducidos por un mismo o diferentes laboratorios, y 3) fomentar y permitir la replicación exacta dentro y a través de los laboratorios de investigación que están aplicando a problemas y haciendo evaluaciones básicas en la ciencia psicológica (Lang *et al.*, 2008).

Para evaluar las tres dimensiones: placer, excitación y dominancia; han utilizado la herramienta SAM, un sistema de clasificación afectiva ideado por Lang (1980). En este sistema, una figura gráfica que representa los valores a lo largo de cada una de las tres dimensiones en una escala de variación continua, se utiliza para indicar las reacciones emocionales (véase Sección D.2.3) (Lang *et al.*, 2008).

IAPS (Lang *et al.*, 2008) incluye actualmente más de 1.000 ejemplares de la experiencia humana —personas alegres, tristes, temerosas, enojadas, amenazantes, atractivas, feas, vestidas y desnudas; casas, objetos de arte, objetos de uso doméstico; proyectos de vivienda; parejas eróticas; funerales; contaminación; baños sucios; paisajes urbanos y marinos; eventos deportivos; fotoperiodismo de las guerras y de desastres; tratamientos médicos, pacientes enfermos, cuerpos mutilados; animales bebés, animales e insectos amenazantes; familias amorosas; cascadas; niños jugando—. Cada imagen en el IAPS es clasificada por un gran grupo de personas (hombres y mujeres) por

los sentimientos de placer y excitación que evocan durante su visualización. Las imágenes son luego numeradas (4 dígitos) y catalogadas en función de la media y la desviación estándar de estos criterios de valoración afectivos, y distribuidas gratuitamente a los investigadores académicos. Haciendo uso de estas clasificaciones, los científicos pueden seleccionar y/o combinar imágenes sobre las bases del impacto emocional promedio registrado para esa imagen, siendo capaces de controlar la excitación emocional al momento de investigar efectos de valencia hedónica y viceversa. Lo más importante, el IAPS fomenta la replicación y extensión de informes experimentales publicados utilizando el mismo o similar material de estímulo, promoviendo el incremento acumulado del conocimiento científico (Bradley y Lang, 2007a).

2.4 Modelado de la cadencia de tecleo

2.4.1 Introducción

El modelado de cadencias de tecleo, como rama de la seguridad biométrica no invasiva, utiliza herramientas de captura de datos, generalmente provistas por los sistemas operativos, para obtener información de temporización u otras relevantes de los periféricos de entrada. Por medio de herramientas estadísticas se modela el comportamiento del conjunto de usuarios legítimos utilizando secuencias de entrenamiento de duración variable con la intención de aplicar, posteriormente, herramientas de clasificación para la identificación o verificación de la identidad de un usuario. La clase asignada podrá determinar o influir en el acceso al sistema o solamente utilizarse como información de auditoría o forense. Algunos esquemas de modelado cuentan con la posibilidad de adaptarse gradualmente a los cambios naturales que suceden a lo largo del tiempo en la forma de teclear del usuario.

Los problemas del almacenamiento de los modelos y la transmisión de la información biométrica en forma segura, además de la conservación de la privacidad o la prevención y mitigación de los efectos de la publicación o sustracción accidental o malintencionada de los anteriores, si bien pueden estar relacionados con las técnicas de

modelado de la cadencia de tecleo, no pertenecen estrictamente a este campo sino a la seguridad informática entendida en su forma más general y no serán tratados en detalle.

2.4.2 Características biométricas

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 utilizado una amplia variedad de técnicas de clasificación y aprendizaje automático para el análisis de los parámetros anteriores. El enfoque utilizado ha dependido fundamentalmente de la longitud del texto utilizado, para la autenticación y de la posibilidad de que la herramienta de verificación se haya entrenado con el mismo texto que se utilizará para la verificación. Ambas características están íntimamente relacionadas y dan lugar a dos enfoques: la verificación de claves estáticas, con la posibilidad de una adaptación gradual a los cambios en la cadencia de tecleo (González, 2014a; Monroe *et al.*, 1999) del usuario y la autenticación de texto libre. Esta última modalidad se ha utilizado tanto para la autenticación continua (Flori y Kowalski, 2010; González, 2014^a) como para la autenticación basada en desafío–respuesta, lo que brinda la posibilidad de generar nuevas frases de desafío en el momento de la autenticación o el uso de un conjunto predeterminado de textos (González, 2014^a).

Se ha reportado también la medición de las diferencias en la presión al teclear (Tang *et al.*, 2010), aunque la necesidad de tener un teclado especialmente diseñado para este propósito hace que esta técnica no

sea ampliamente utilizada. Sin embargo, si se contara con mayor información en el modelo de de cadencia de tecleo del usuario, se podría mejorar la eficiencia en la clasificación y la técnica mencionada puede resultar útil en aplicaciones donde el nivel de seguridad buscado es un requisito de mayor peso que el costo y la complejidad de la solución.





Los parámetros más utilizados son los relacionados con el tiempo entre presiones y/o liberaciones de teclas; estos eventos son llamados respectivamente *Key Down* y *Key Up*. Entre ellos, podemos considerar:

WT *Wait time* o latencia entre presión de teclas consecutivas. Es el tiempo transcurrido desde que se liberó la tecla anterior hasta que se presiona la próxima tecla. Puede ser negativo si el usuario invierte el orden de liberación de las teclas o, sin que constituya una excepción, cuando se utilizan las teclas de mayúsculas, control, alteración o *host*, entre otras.

HT *Hold time* o tiempo de retención de la tecla. Mide el tiempo transcurrido desde que se presiona una tecla hasta que la misma es liberada. Siempre es positivo.

FT *Flight time* o tiempo intertecla. Mide el tiempo transcurrido desde que se presionó la tecla anterior hasta la presión de la tecla actual, sin considerar el instante de liberación de ninguna de ellas. Siempre es positivo.

2.4.3 Ejemplo concreto

Una representación gráfica del posible ingreso de la palabra “robustez” en donde se detallan los anteriores se puede observar en la figura 2.8. En el primer renglón se encuentran las teclas en el momento en que son presionadas mientras que en el segundo renglón se encuentran las mismas al ser liberadas. El evento, al ser presionado, se llama *Key Down* o bien *Press*; el evento de liberación de una tecla es *Key Up* o *Release*. Los *Wait Time* WT_{a-b} y *Flight Time* FT_{a-b} se indican entre la tecla a y la tecla b , por ejemplo WT_{O-B} corresponde a la liberación de  y la presión de B. Nótese que el *Wait time* $WT_{Shift-U}$ entre la teclas  y  es negativo, ya que  es presionada antes

de liberar shift ↑. En el caso del *Hold Time* solamente se indica la tecla presionada.

La combinación de todos los eventos WT y HT en una colección crea lo que describimos como un *set of events*. Este conjunto contiene información estadística agregada (como la media, la varianza u otro indicador estadístico) para cada evento para una clave específica (en HT) o un par de claves (en WT). En el ejemplo de figura 2.8, el conjunto de eventos que poseen los tiempos medios sería R:55, RO:13, O:29, OB:13, B:15, BU:14, U:12, US:24, S:20, ST:16, T:15, TE:13, E:17, EZ:20, Z:22. Los eventos de dos teclas se producen a partir de WT, los de una sola se derivan de los HT. Cuando se compara con otro patrón, intersecamos ambos conjuntos para obtener un par de vectores de igual tamaño que representan la información estadística de los eventos en común, es decir, la media.

Siguiendo nuestro ejemplo, asumiendo que necesitamos compararlo con la palabra “sobrepuesto”, formada por el conjunto de eventos S:66.5, SO:50, O:31, OB:30, B:13, BR:17, R:20, RE:58, E:24.5, EP:33, P:13, PU:24, U:13, UE:55, ES:33, ST:32, T:20, TO:40, el resultado serán los vectores siguientes —para los eventos compartidos O, OB, B, S, ST, T, E— respectivamente: $\sim x = (29,13,15,20,16,15,17)$ e $\sim y = (31,30,13,66,5,32,20,24,5)$.

Luego, los vectores se pueden contrastar aplicando varias métricas, como Manhattan (L^1), Euclidean (L^2), entre otros. Además, también es posible asociar un contexto particular a cada evento (González *et al.*, 2016) para producir información adicional.

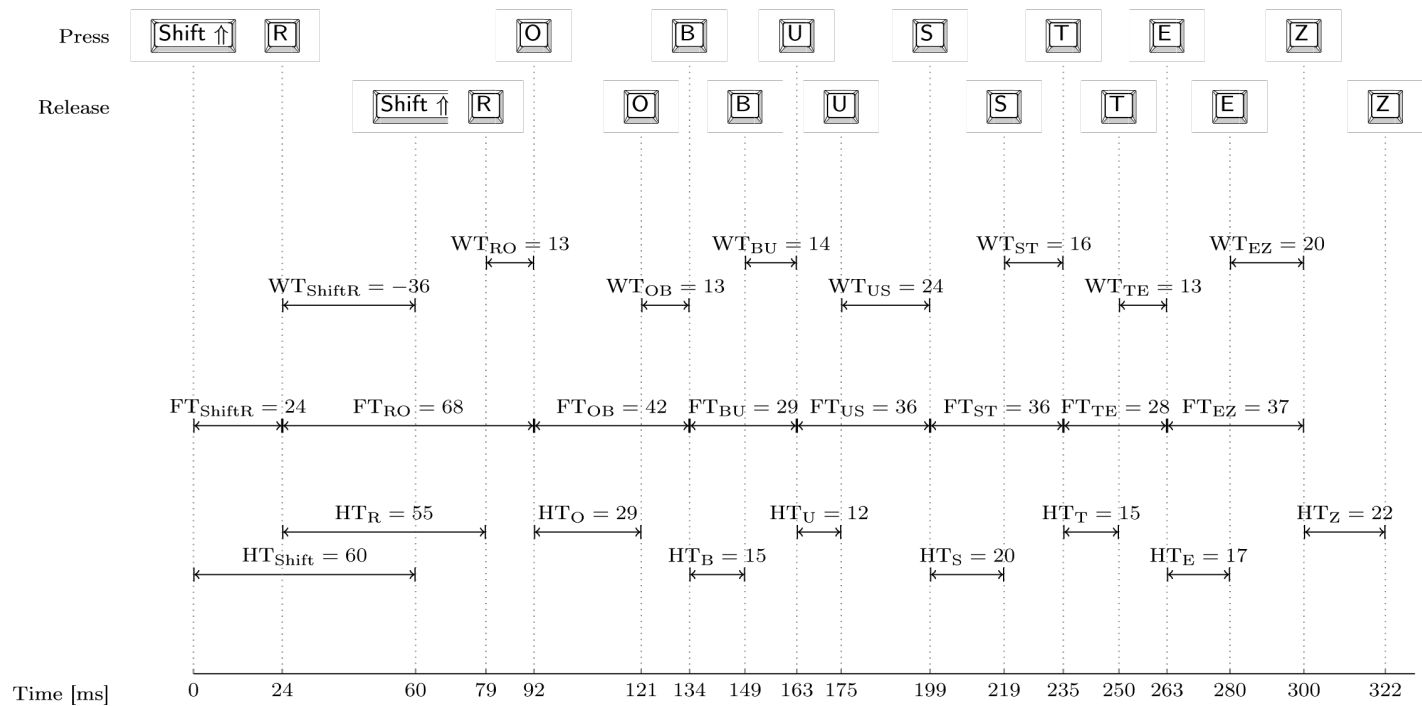


Figura 2.8: Parámetros temporales entre presiones de teclas.

2.4.4 Parámetros relacionados con el comportamiento

Otros parámetros utilizados para el modelado de la cadencia de tecleo del usuario, que no son directamente medibles pero si cuantificables indirectamente, son los referidos a sus patrones generales de comportamiento durante la acción de ingreso de datos. Es en estos donde mayor fruto rinde la utilización de contextos finitos en los modelos.

Velocidad general de tecleo Es un resultado directo del nivel de instrucción y experiencia dactilográfica del usuario y por lo tanto varía lentamente con el paso del tiempo.

Probabilidad de error de tecleo Se mide con la frecuencia de uso de las teclas *backspace* y *delete*. Al igual que en el caso anterior, es un resultado directo del nivel de instrucción y experiencia dactilográfica del usuario, pero puede presentar variaciones significativas entre sesiones o dentro de la misma sesión de usuario. Para un usuario dado, el contexto anterior suele ser un buen predictor de la probabilidad de error.

Hábitos de sectorización El empleo específico de utilitarios que hace el usuario durante su sesión puede expresarse en hábitos de sectorización al teclear, como por ejemplo diferentes frecuencias relativas en el uso de números y de caracteres alfabéticos o especiales. Un programador abundará en estos últimos, mientras que un contador probablemente presente una mayor frecuencia de los primeros.

Orden de liberación de teclas Al utilizar las teclas de mayúsculas, control, alteración o *host*, entre otras, el orden de liberación de la tecla afectada y la tecla afectante no produce cambios visibles en el texto ingresado. Dicho orden suele respetarse para las mismas teclas bajo los mismos contextos en un usuario dado, por lo que agrega información para la clasificación.

Ninguno de los mencionados es suficientemente significativo para ser considerado en forma aislada, pero todos ellos permiten mejorar la eficiencia de la clasificación, ya que agregan información de comportamiento correlacionada con la identidad del usuario.

2.4.5 Textos libres y estáticos

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 (González, 2014^a).

La construcción del modelo de cadencia de tecleo para un usuario dado puede considerarse un problema de clasificación con dos clases. Para tal fin, se han ensayado los clasificadores estadísticos comunes, paramétricos y no paramétricos, como ser k-NN (Saevanee y Bhatarakosol, 2008), k-means (Obaidat y Sadoun, 1997) y bayesianos (Mahalanobis, 1936). Adicionalmente, se han ensayado otros métodos avanzados de clasificación como las redes neuronales (Cho *et al.*, 2000), lógica difusa (de Ru y Eloff, 1997), algoritmos genéticos y máquinas de vectores de soporte (Teh *et al.*, 2012).

Una recopilación de resultados para todos los clasificadores mencionados utilizando diversas métricas sobre un juego de datos estandarizado ha sido realizada por Killourhy y Maxion (2009). En Calot *et al.* (2013) comparamos las distintas distancias midiendo el área bajo la curva característica de operaciones, obteniendo valores prácticamente idénticos para la distancia euclídea y de Mahalanobis.

La autenticación continua o de textos libres presenta desafíos adicionales a la autenticación de claves estáticas. A diferencia de esta última, la multiplicidad de textos posibles hace que la verificación deba realizarse, en virtualmente todos los casos, con combinaciones distintas a las utilizadas para el entrenamiento, por lo que se requieren nuevas estrategias que permitan medir la similitud de cadencia entre textos disímiles. De más está decir que la forma de teclear una clave estática tiende a estabilizarse en el tiempo, cosa que no sucede con el

texto libre. Un problema adicional, que será tratado en la sección 2.7 es que el usuario puede variar el lenguaje utilizado durante una sesión.

No todos los clasificadores idóneos para claves estáticas funcionan bien con textos libres y, en muchos casos, ni siquiera es evidente cómo adaptarlos para esta situación. Entre los clasificadores de propósito general, los basados en espacios métricos han dado buenos resultados (Flior y Kowalski, 2010; Leggett y Williams, 1988). Sin embargo, dos nuevas técnicas de clasificación de propósito específico denominados métrica R y métrica A, inicialmente propuestos por Bergadano, Gunetti y Picardi en Bergadano et al. (2002) y Gunetti y Picardi (2005), se llevan las palmas y han sido utilizadas por otros autores (Messerman *et al.*, 2011) luego de optimizarlas para su ejecución en tiempo real, ya que el costo computacional de estas técnicas no es irrelevante. En la sección A.1.3.1 se expone la métrica R y se presenta un ejemplo.

La evaluación empírica muestra que distintos textos ingresados por el mismo usuario producen valores del grado de desorden relativo consistentemente pequeños en comparación con los de otro usuario, aún descontando factores de diversidad como el largo de las particiones, el texto ingresado o el idioma. Los resultados de la clasificación no solamente son excelentes, con un FAR del orden del 5% a pesar de las dificultades adicionales, sino que además se logran con conjuntos de entrenamiento pequeños en comparación con otros métodos. Como ventaja adicional, se puede mencionar el hecho de que, al considerar solo el orden y no los tiempos absolutos, este esquema se muestra robusto frente a las variaciones espurias en la velocidad general de tecleo.

Para el proceso de asignación a las clases de usuarios legítimos o impostores de los vectores característicos, se han ensayado los clasificadores usuales, paramétricos y no paramétricos, como ser k-NN (Cho *et al.*, 2000; Killourhy y Maxion, 2009; Saevanee y Bhatarakosol, 2008), k-means (Cho *et al.*, 2000; Obaidat y Sadoun, 1997) y bayesianos (Bleha *et al.*, 1990). Adicionalmente, se han ensayado otros métodos avanzados de clasificación como las redes neuronales (Cho *et al.*, 2000; Haider *et al.*, 2000; Obaidat y Sadoun, 1997), lógica difusa (de Ru y Eloff, 1997), algoritmos genéticos (Obaidat y Sadoun, 1997), máquinas de vectores de soporte (Yu y Cho, 2003) y bosques aleatorios (Maxion

y Killourhy, 2010). La descripción de los mismos y la reseña de los resultados obtenidos con ellos, es tema del capítulo siguiente.

Entre las métricas de distancia utilizadas para comparar el vector característico para la autenticación y el o los vectores patrón se encuentran la distancia euclídea (Messerman *et al.*, 2011), L_1 o de Manhattan (Joyce y Gupta, 1990), y la distancia de Mahalanobis (Bleha *et al.*, 1990). Se han propuesto también otras métricas de propósito específico (Zhong *et al.*, 2012), que buscan aprovechar las condiciones particulares del modelado de cadencias de tecleo.

2.5 Detección de impostores con cadencia de tecleo

El significado de una tasa de error de clasificación reportada, ya sea FAR o FRR, depende del modelo de ataque que se está presuponiendo y con el cual se ha evaluado el modelo de autenticación que se propone. Si se supone que los usuarios impostores no tienen conocimiento de la existencia de una capa de autenticación biométrica oculta basada en la cadencia de tecleo, además de las formas explícitas y expuestas a la vista como pueden ser la exigencia de una clave u algún otro identificador biométrico, es esperable que estos no intenten disimular su forma de escribir; hablamos en este caso del modelo de ataque con *impostores no entrenados*. Contrariamente, si el impostor es consciente de la existencia de un esquema tal, es razonable que busque imitar la cadencia del usuario legítimo y nos encontramos ante un modelo de ataque con *impostores entrenados*. En la mayoría de los casos se ha utilizado un modelo de ataque con impostores entrenados, para evaluar el rendimiento de los clasificadores.

Es intuitivamente aprehensible que la detección de impostores entrenados es, por lejos, el problema más difícil y deben, por lo tanto, interpretarse los valores de FAR en forma más benigna que con impostores no entrenados. En otras palabras, es esperable un FAR más elevado en el primer caso, aunque no se ha realizado un estudio sistemático que permita cuantificar la influencia de este factor en términos de rendimiento. En forma indirecta, el FRR también se ve influido, ya que la búsqueda de niveles más bajos de FAR obliga a

ajustar la sensibilidad del sistema de autenticación hacia niveles más estrictos, lo que a su vez produce un aumento de la primera tasa.

Existen diversas técnicas para procesar los datos obtenidos por cadencia de tecleo. Las mismas pueden ser entrenamientos con redes neuronales, SVM, árboles de decisión, bosques aleatorios o bien utilizando distancias para medir la diferencia entre dos patrones. Las distancias pueden ser euclídeas (o euclidianas, L_2), euclídeas Normalizadas, de Mahalanobis, Manhattan L_1 , Normas A y R, entre otras.

El cuadro 2.1, compara distintos estudios publicados mostrando ya sea FAR y FRR o bien EER. Como este trabajo se centrará en texto libre, es posible observar que los mejores resultados se han dado con distancias, principalmente las normas A y R.

2.5.1 Clasificadores de propósito general

La clasificación es el proceso de asignación de una categoría perteneciente a un conjunto predeterminado a una nueva observación, en base a la información dada por un conjunto de observaciones de entrenamiento cuya categoría de pertenencia es conocida. Al utilizar la cadencia de tecleo para verificar la identidad de un usuario, la observación es la matriz característica correspondiente al ingreso de una clave estática o a una partición del texto de entrada y las categorías posibles son las de usuario legítimo o impostor. Para el caso de la identificación, la observación es la misma y existe una categoría posible por cada usuario enrolado en el sistema; adicionalmente, puede existir una categoría para señalar la no correspondencia con ninguno de los anteriores. El conjunto de datos de entrenamiento varía según el enfoque de autenticación. Para claves estáticas consiste en las matrices características de una serie de repeticiones de las mismas, mientras que en la autenticación continua, forense, o por desafío–respuesta está dado por un subconjunto o la totalidad de las particiones del texto ingresado durante las sesiones registradas del usuario.

Referencia	Textolibre	Clasificador/Distancia Dataset público	Clasificadores	Original/Réplica	¿Réplica exitosa?	FAR	FRR	EER
Bleha <i>et al.</i> [1990]	D		Mahalanobis	O	-	2,8	8,1	
Joyce y Gupta [1990]	D		L ₁ (mod.)	O	-	0,2	16,3	
Obaidat y Sadoun [1997]	C		Varios	O	-			0,8
	C		R. neuronal	O	-			0
Monrose <i>et al.</i> [1999]	C		Estadístico	O	-	0	22,9	
Cho <i>et al.</i> [2000]	C		k-NN	O	-			1,6
	C		R. neuronal	O	-			0
Haider <i>et al.</i> [2000]	C		Atípicos	O	-	2	3	
	C		R. neuronal	O	-	20	41	
	C		Fuzzy	O	-	11	26	
Bergadano <i>et al.</i> [2002]	D		R	O	-	0,01	4	
Yu y Cho [2003]	C		SVM	O	-	0	3,5	
Araujo <i>et al.</i> [2005]	D		L ₁ (mod.)	O	-			1,6
Gunetti y Picardi [2005]	D	X	R	O	-	0,005	5	
Sung y Cho [2005]	C		SVM	O/R	×	3,8	13,1	
Sheng <i>et al.</i> [2005]	C		Árboles	O	-	0,9	9,6	
Teh <i>et al.</i> [2007]	C		Ensamble	O	-			6,4
Hu <i>et al.</i> [2008]	C		k-NN	R X		0,05	0	
Killourhy y Maxion [2009]	-	X	Varios	R	×			9,6
Flior y Kowalski [2010]	X	D	Distancias	O	-	-	-	-

Maxion y Killourhy [2010]	C	X	Random Forest	O	-	0,5	12,5	1,5
González y Calot [2015]	X	D	X Distancias	O	-			7,5

Cuadro 2.1: Comparación de técnicas según los distintos autores.

En el modelado de cadencias de tecleo, el entrenamiento es necesariamente supervisado, ya que se presupone que al momento del entrenamiento es el usuario legítimo quien efectivamente se encuentra tecleando, eliminando la necesidad de un proceso inicial de *clustering*. Al algoritmo que realiza el proceso de clasificación se lo denomina *clasificador*. A continuación, se hace una revisión de los clasificadores de propósito general utilizados en la literatura publicada sobre modelado de cadencias de tecleo.

2.5.2 Definiciones

La matriz característica M de una clave estática o de una partición del texto de entrada es un vector de $n \times m$ dimensiones. Se denomina *espacio de parámetros* al espacio vectorial $V = \mathbb{R}^{n \times m}$ sobre \mathbb{R} junto con las operaciones usuales, al cual pertenece la matriz y en el que cada dimensión representa el valor medido de uno de los parámetros característicos. Al adjuntar cualquiera de las normas tratadas anteriormente, el mismo deviene un espacio normado. Nótese que para claves de diferente tamaño o múltiples particiones de un texto continuo varía el tamaño de la matriz y, por lo tanto, el espacio asociado, así que se no se habla de un único espacio de parámetros para todos los intentos de clasificación.

El valor nulo \emptyset utilizado para denotar la ausencia de medición de un parámetro específico en una tecla específica no representa ningún valor de $r \in \mathbb{R}$, sino que se utiliza solamente a los fines de simplificar las expresiones y las representaciones, tanto visuales como algorítmicas. Formalmente, una matriz característica M de $n \times m$ con k valores nulos debe ser considerado un vector $v \in V = \mathbb{R}^{n \times m - k}$.

2.5.3 Espacios normados

A pesar de ser muy sencillas de implementar, las distancias normalizadas (ver sección A.1.1.2) han arrojado excelentes resultados en multitud de experimentos (algunos ejemplos son Bleha *et al.*, 1990; Calot *et al.*, 2013; Killourhy y Maxion, 2009). Las diversas metodologías de reporte de rendimiento hacen complicado compararlos pero, en general, presentan un ERR entre el 10% y el 15%, con valores de FAR y FRR más bajos si se compromete uno de ellos en beneficio del otro. La distancia de Manhattan suele producir un error de clasificación marginalmente más bajo. Su simplicidad de implementación las hace ideales para emplearlas como caso base contra el cual comparar clasificadores más avanzados.

2.5.3.1 Variaciones

En Araujo *et al.* (2005) se describe el uso de una variante de la distancia de Manhattan que brinda excelentes resultados según sus propias pruebas y cuyo rendimiento ha sido confirmado independientemente por Killourhy y Maxion (2009), bajo cuyo conjunto de entrenamiento y evaluación obtiene el menor EER y un punto de FRR cero entre los mejores de todos los clasificadores considerados. La expresión de la distancia de Manhattan modificada entre un vector $\mathbf{v} = (v_1, \dots, v_n)$ y la distribución correspondiente al usuario i de media μ_i es

$$d(\mu_i, \mathbf{v}) = \sum_{j=1}^n \frac{|\mu_{ij} - v_j|}{a_{ij}} \quad (2.13)$$

Esta distancia L_1 modificada cambia el factor de escala, utilizando en lugar del desvío estándar al término a_{ij} , que es el desvío absoluto medio correspondiente a la j -ésima componente y se calcula en base a los datos capturados durante el entrenamiento en la forma

$$a_{ij} = \frac{1}{m} \sum_{k=1}^m |\mu_{ij} - t_{ijk}| \quad (2.14)$$

Aquí t_{ijk} es el valor medido para el j -ésimo parámetro característico durante la iteración k del entrenamiento del usuario i y m es la cantidad de instancias utilizadas para el mismo.

También Bleha *et al.* (1990) proponen una modificación, esta vez tanto a la distancia euclídea como de Mahalanobis, cuyo fin sería permitir que se acomoden en forma transparente a la variación en el largo de las claves. La misma consiste en escalar la distancia original por el producto de las normas de la nueva observación y de la media correspondiente al usuario. Formalmente

$$D(\mu_i, \mathbf{M}_{nueva}) = \frac{d(\mu_i, \mathbf{M}_{nueva})}{\|\mu_i\| \|\mathbf{M}_{nueva}\|} \quad (2.15)$$

Siguiendo, una vez más, a Killourhy y Maxion, el propósito no se habría realizado, con la penalización adicional de que el efecto de la reforma sobre la clasificación es deletéreo; tanto el EER como el punto de FRR cero serían peores que los de su contraparte original.

2.5.3.2 La distancia de Mahalanobis

Es esperable que los múltiples parámetros que conforman una matriz característica se encuentren correlacionados entre sí y que, adicionalmente, las distintas observaciones de los mismos en teclas sucesivas también lo estén. Las oscilaciones en la velocidad de tecleo promedio son medibles en todos los parámetros; no resulta razonable esperar que un usuario cuya latencia entre presión de teclas consecutivas (HT) sea elevada conserve un tiempo de retención (WT) de escasos milisegundos o que oscile en un orden de magnitud al escribir.

Una manera natural de considerar las correlaciones entre mediciones sucesivas de parámetros característicos es utilizar la distancia de Mahalanobis (ver sección A.1.5.2), que mide las desviaciones sobre las direcciones de los ejes principales dadas por los autovectores de la matriz de covarianza. El mérito de proponer esta métrica para la clasificación de usuarios en base a su cadencia de tecleo corresponde a Bleha *et al.* (1990), quienes reportan un FAR de 2,8% y un FRR de 8,1% sobre más de 500 clasificaciones con 39 usuarios, los cuales se dividían en 14 legítimos, 13 impostores entrenados y 12 impostores no entrenados.

Contrariamente a lo esperado, la consideración de las correlaciones no siempre es provechosa. En Calot *et al.* (2013) comparamos el

rendimiento de la clasificación con la distancia euclídea normalizada y con la distancia de Mahalanobis utilizando el área bajo la curva de ROC, concluyendo, como se observa en la figura 2.9, que los resultados con ambas son virtualmente idénticos para una cantidad de hasta veinte eventos —registran una variación del 0,24% para este último valor—, utilizando el mismo juego de datos de entrenamiento y prueba que en Killourhy y Maxion (2009). Notamos que, desde cualquier otro punto de vista, la distancia euclídea normalizada resulta más conveniente, ya que sus requerimientos de almacenamiento y de potencia de cómputo son menores al no requerir persistencia ni cálculo de la matriz de covarianza. Adicionalmente, existen mecanismos de implementación casi trivial que permiten la adaptación gradual del modelo cuya implementación con la distancia de Mahalanobis requiere recursos computacionales considerables. Esto se debe, principalmente, a que el conjunto de datos utilizado por Killourhy y Maxion (2009) utiliza FT, HT y WT, y —notando que hay una correlación lineal entre ellos dada por $WT = HT + FT$ — la distancia de Mahalanobis solo se limita a eliminar esta dependencia lineal.

Otras métricas de rendimiento revelan en la distancia de Mahalanobis un rendimiento mixto, al compararla con las distancias normalizadas de Manhattan y euclídea. Bajo el criterio del EER, es superada por la primera y seguida por la segunda, mientras que supera a ambas en el punto de FRR cero (Killourhy y Maxion, 2009). Este fenómeno puede explicarse en base a la observación de que ciertos fragmentos de texto se encuentran sobrerrepresentados en la escritura y, por lo tanto, presentan un entrenamiento reiterado de mucha mayor frecuencia que cualquier otra pieza de texto (Gaines *et al.*, 1980; Killourhy, 2012), compactando las observaciones del espacio de parámetros en torno a la media e independizándolas parcialmente de variaciones debidas a factores pasajeros o del contexto, de las que podría sacar mayor provecho la consideración de las correlaciones.

Claramente, las claves estáticas pertenecen a la categoría de los fragmentos de texto sobrerrepresentados y sobreentrenados, ya que en el curso usual de las sesiones de usuario es probable que estas aparezcan con frecuencia. Inversamente, es de esperar que, al realizar identificación continua por desafío–respuesta o forense, la consideración de las correlaciones sea mucho más provechosa; en las secciones siguientes será analizada esta sospecha.

2.5.3.3 Vecinos más cercanos

La idea subyacente al clasificador de vecinos más cercanos o kNN (del inglés *k nearest neighbours*) es no utilizar la media de una distribución como patrón de su clase, sino la totalidad de las observaciones de entrenamiento, considerando cuántas de una cierta clase son las más cercanas a la nueva observación a clasificar.

Tanto el vecino más cercano como la forma general con $k \leq 3$ fue utilizado por Cho *et al.* (2000) como caso base para comparar contra un clasificador implementado por medio de una red neuronal autoasociativa de múltiples capas, que será detallado más abajo. Informan que el mejor resultado obtenido fue con $k = 1$, pero sin cuantificar el mismo. Irónicamente, k-NN resulta ser el mejor

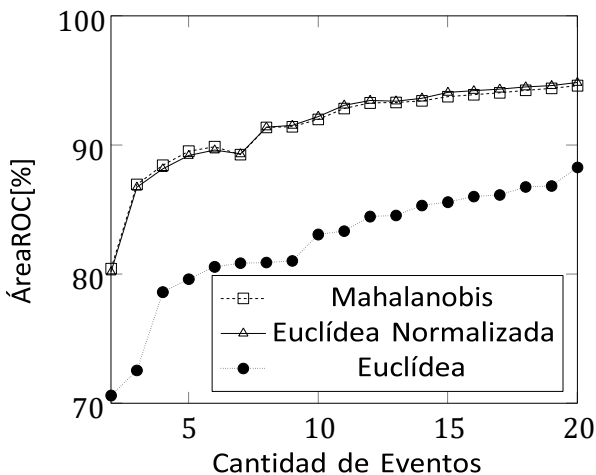


Figura 2.9: Área bajo la curva de ROC para las distancias euclídea, euclídea normalizada y de Mahalanobis.

clasificador en base al punto de FRR cero y el segundo por un escaso margen de 0,4% en base al EER según Killourhy y Maxion (2009), mientras que la red neuronal propuesta es consistentemente peor. En Gunetti y Picardi (2005), este método es revisitado y los autores logran, en concordancia con los anteriores, un FAR significativamente bajo en el orden del 0,045%, aunque a costa de un tiempo de autenticación inaceptable.

A diferencia de los clasificadores vistos anteriormente y la mayoría de los que siguen, en los que el costo computacional no es restrictivo ni requiere consideraciones especiales para su implementación práctica, el algoritmo k-NN debe ser cuidadosamente implementado para que el tiempo de autenticación sea aceptable. La necesidad de calcular la distancia de la nueva observación a todas las instancias de entrenamiento, obliga a restringir la cantidad de usuarios y cardinalidad de sus conjuntos patrón además, de utilizar estrategias de *clustering* para descartar en bloque las instancias lejanas. Hu *et al.* (2008) mediante diversas consideraciones logran reducir los tiempos de Gunetti y Picardi (2005), que excede los 20 segundos con solamente 5 usuarios y se eleva hasta casi 40 con 15 usuarios, a entre 16 y 19 segundos para dicha cantidad de perfiles. Si bien estos números deben entenderse bajo la consideración de que aplican a un *hardware* no especificado del año 2008, no dejan de ser sorprendentemente elevados.

2.5.3.4 k-means

El algoritmo de *k-means* no es estrictamente un clasificador sino un algoritmo de *clustering* que ha sido utilizado en varias ocasiones (Cho *et al.*, 2000; Obaidat y Sadoun, 1997) como caso base para generar un clasificador sencillo y de fácil comparación contra otros de mayor complejidad. El resultado esperado es la partición del espacio métrico en una cantidad parametrizable k (de donde el algoritmo toma su nombre) de celdas donde cada observación se encuentra en la celda cuyo centroide está a menor distancia que todos los demás. Agregando a la representación las líneas (o, en general, hiperplanos) que se encuentran a distancia mínima de los centroides tomados de a pares, se obtiene la representación usualmente denominada *celdas de Voronoi*.

La complejidad del problema de encontrar los centroides o, lo que es lo mismo, la partición del espacio métrico, varía entre *NP-hard* y polinómica con un exponente muy alto ($dk + 1$, donde d es la cantidad de dimensiones y k la cantidad de celdas), por lo que se requiere la utilización de técnicas heurísticas, que en la práctica suelen funcionar en forma casi óptima, para particionar conjuntos de datos que consistan en más que unas pocas observaciones. El principio general

de funcionamiento de las mismas consiste en calcular iterativamente la posición de los centroides en un espacio métrico, partiendo de una configuración inicial que puede ser aleatoria, basada en un subconjunto de los datos o una mezcla de ambas. Al agregar datos de entrenamiento, se busca actualizar la posición de los centroides para minimizar un cierto índice de rendimiento, que puede ser alguna de las distancias usuales tratadas anteriormente. Existen múltiples estrategias para la inicialización y actualización de los centroides que pueden ser consultadas en la bibliografía específica.

Una vez obtenida una partición, la tarea de clasificación se reduce a determinar en qué celda cae una nueva observación. Obaidat y Sadoun (1997) reportan un FAR y FRR de aproximadamente 10% y 15%, muy cercanos a los obtenidos en sus pruebas con distancia euclídea.

2.5.3.5 Conteo de valores atípicos

El conteo de valores atípicos es una puntuación muy simple que indica la cantidad de valores de los parámetros característicos que difieren de la media en más de una cierta cota, medida en desvíos estándar. La observación debe normalizarse inicialmente ecualizandola con su propia media y desvío —en (5.2) se define la ecuación de ecualización— para luego ser comparada contra un umbral T . Formalmente, dado un usuario j el conteo de valores atípicos de una observación $V \sim = (v_1, \dots, v_n)$ respecto de T se calcula como

$$C_j = \sum_{i=1}^n H \left(\frac{|v_i - \mu_{ji}|}{\sigma_{ji}} - T \right) \quad (2.16)$$

con $H(x)$ el escalón unitario, μ_{ji} la media para el i -ésimo parámetro y σ_{ji} su desvío estándar. A pesar de su simplicidad, permite obtener resultados similares a muchos clasificadores más complejos (Haider *et al.*, 2000; Killourhy y Maxion, 2009).

2.5.3.6 Máquinas de vectores de soporte

Las *máquinas de vectores de soporte* o SVM, del inglés *support vector machine*, se encuentran entre los mejores clasificadores modernos por su bajo error de clasificación, además de otras características deseables, como la maximización del margen geométrico entre regiones del espacio de parámetros que corresponden a cada clase posible y la versatilidad que brinda poder particionar este espacio en forma no lineal utilizando distintas funciones de *kernel*. Es un método muy eficaz, generalmente considerado el mejor clasificador o uno de los mejores por escaso margen en una amplia variedad de problemas complejos, entre ellos la clasificación de imágenes o el reconocimiento de la caligrafía cursiva.

Durante el entrenamiento, este clasificador intenta inferir la localización de un hiperplano —o un conjunto de ellos— en el espacio de parámetros que separa las observaciones en dos categorías, buscando que la distancia de este hiperplano a las observaciones más cercanas sea la mayor posible. Para problemas con más de dos clases, se requiere reducirlos a múltiples problemas de dos clases mediante la implementación de varios clasificadores. Adicionalmente, deformando el espacio de parámetros se puede lograr un efecto análogo a la utilización de fronteras curvadas que representan mejor la estructura del problema.

Como contraparte a su eficacia, las máquinas de soporte vectorial no son clasificadores sencillos. La implementación de la maquinaria algebraica es compleja e involucra el ajuste de múltiples parámetros de diseño al problema que se intenta atacar, que pueden influir negativamente en la eficacia si no son correctamente elegidos. Una excelente introducción a las sutilezas de diseño e implementación de estos algoritmos se puede consultar en Cristianini y Shawe-Taylor (2000). Es importante mencionar que el costo computacional del entrenamiento y la clasificación se encuentran entre los más altos de todos los algoritmos mencionados, exceptuando tal vez a las redes neuronales.

Yu y Cho (2003) han probado este clasificador sobre un conjunto de datos de entrenamiento que comprende entre 150 y 400 ingresos de claves con largos de entre seis y diez caracteres por parte de veintiún

participantes; adicionalmente, se incluyeron cinco repeticiones de la clave de cada usuario, ingresadas por quince impostores con entrenamiento. Los resultados se reportan bajo la forma del FRR en el punto de FAR cero, obteniéndose valores que oscilan entre 1,25% y 4,68% con una tendencia a obtener valores inferiores para claves más largas. Aunque el resultado no parezca sorprendente en vista del rendimiento esperado por parte de las máquinas de soporte vectorial para la clasificación, debe tenerse en cuenta que la utilización de impostores con entrenamiento como desafío es mucho menos benigna que la práctica usual de comparar los usuarios entre sí sin presuponer intentos malintencionados y, por lo tanto, debe considerarse excelentes a los resultados obtenidos. Para alcanzar estos valores no se utilizó la totalidad del vector característico, sino que se propone un ingenioso esquema que, por medio de algoritmos genéticos, permite extraer los componentes relevantes para mejorar la eficacia en la clasificación. El empleo ingenuo de SVM con el vector característico completo eleva el FRR en el punto de FAR cero a un promedio de 15% con picos de hasta 20%, lo que ejemplifica lo dicho más arriba sobre la necesidad de ajustar los parámetros de diseño del clasificador.

Posteriormente, Sung y Cho (2005) han propuesto ciertas mejoras al esquema de selección de componentes que permiten disminuir los errores en la clasificación. La comparación con los resultados anteriores no es inmediata, ya que, en esta ocasión, se reporta un FAR promedio de 3,85% y un FRR de 13,10% en lugar del FRR para el punto de FAR cero. Con valores respectivos para el anterior de 13,13% y 7,25%, el nuevo método lograría una mejora en el FAR a costa de una degradación del FRR.

2.5.4 Otros clasificadores

No todos los clasificadores basan su funcionamiento en la representación de los vectores característicos, dentro de un espacio métrico y la utilización más o menos sutil de alguna noción de distancia, para determinar el grado de semejanza con los vectores patrón inferidos durante el entrenamiento. Existen otros métodos que intentan extraer en forma directa las características significativas de los vectores que se comparan para llegar a la asignación de una categoría sin utilizar valores continuos, sino más bien en forma taxativa, a todo

o nada, como son los árboles de decisión, los ensambles o los bosques aleatorios. Los valores continuos pueden aparecer también sin hacer referencia a una métrica; es el caso de la lógica difusa, que reemplaza la noción de distancia por la de grado de pertenencia a un conjunto. O, aunque los procesos internos del algoritmo de clasificación puedan describirse con la maquinaria algebraica de los espacios métricos, puede ocurrir que, como en las redes neuronales, sea complejo y poco esclarecedor intentar establecer la relación entre dicha representación interna y la forma explícita e intuitiva de visualizar el vector característico y los vectores patrón que ha caracterizado a todos los anteriores.

A continuación, se reseña el modo de funcionamiento y el rendimiento reportado de los clasificadores explorados en la literatura, sobre clasificación de cadencias de tecleo que comparten la característica de no ser descriptibles en forma inmediata, en términos de espacios métricos.

2.5.4.1 Redes neuronales

Dos variantes de redes neuronales artificiales que se han ensayado como clasificador único para cadencias de tecleo son las redes autoasociativas y el perceptrón de tres capas. Como la topología de la red neuronal debe conservarse fija para el entrenamiento, ambos tipos —y probablemente cualquier otro también— solo pueden ser utilizados para la autenticación con claves estáticas. El tamaño del vector característico debe ser igual a la cantidad de neuronas en la capa de entrada; el resto de los parámetros estructurales presentan mayor libertad para su elección. En la sección A.2, se encuentra una descripción detallada de las redes neuronales mencionadas a continuación.

En Cho *et al.* (2000) se ha ensayado un clasificador implementado en base a una red neuronal autoasociativa de tres capas, en donde todas ellas tienen tantas neuronas como valores de parámetros tiene el vector característico. Para un usuario dado, se forma una red cuyos pesos se inicializan en forma aleatoria y se ingresan a la misma los vectores de entrenamiento utilizando el algoritmo de *backpropagation* (Rumelhart *et al.*, 1986a,b) (ver sección A.2) para el aprendizaje. El mismo vector que se ingresa a la entrada se utiliza como salida

esperada, forzando a la red a codificar las características salientes del usuario en la capa oculta. De esta manera, cuando el entrenamiento ha finalizado, al presentarse en la entrada un vector de similares características a las del usuario legítimo, se obtendrá a la salida un vector muy parecido; inversamente, un vector generado por el tecleo de un impostor, al no corresponderse en las características fundamentales con el patrón codificado en la capa oculta, debería producir una salida que difiera sustancialmente de la entrada. Para cuantificar la diferencia entre ambos vectores, se puede utilizar cualquiera de las distancias mencionadas; los autores no aclaran la utilizada en sus pruebas.

Una aproximación más convencional al problema de la clasificación de usuarios en base a su cadencia de tecleo mediante el uso de un perceptrón de tres capas fue realizada por Haider *et al.* (2000).

Debido a su simplicidad, el perceptrón fue una de las primeras redes neuronales artificiales en ser estudiadas (Anderson, 1995) e implementadas y se cuenta entre las que mayor escrutinio han recibido, tanto en su forma original con dos capas como en las extensiones que agregan una o varias capas ocultas y otras variantes. Con un modelo más sencillo que el de la red autoasociativa, la última capa del perceptrón contiene una única neurona cuya intensidad de salida refleja la cercanía del vector utilizado como entrada al modelo inferido por la red neuronal durante el entrenamiento. Una vez más, se requiere una red neuronal para cada usuario que se pretende enrollar. Los autores proponen, sin justificación alguna, la utilización de dos nodos en la capa oculta por cada tres nodos en la capa de entrada y aprendizaje por *backtracking*.

Con valores de FAR y FRR de 22% y 20% respectivamente para dos intentos sucesivos en el ingreso de la clave y un FRR del 41% (FAR no reportado) para un único intento, los resultados logrados con el perceptrón de tres capas no resultan prometedores. Inversamente, Cho *et al.* (2000) reportan “autenticación perfecta” utilizando como criterio el punto de FRR cero para aproximadamente la mitad de los usuarios y, para el resto, una tasa de error promedio del 1% con un máximo del 4%. Los resultados de Killourhy y Maxion (2009) replican el desempeño pobre del perceptrón de tres capas, que resulta consistentemente el peor de los clasificadores estudiados, pero

contradican las afirmaciones de Cho *et al.* (2000) respecto del desempeño de la red autoasociativa.

Una característica de la utilización de redes neuronales como clasificadores que debe ser resaltada es el hecho de que la representación del conocimiento en las mismas no se encuentra codificado en forma explícita y significativa, en el sentido de que puede ser convertido en categorías comprensibles para el investigador, sino en forma implícita distribuido en los pesos de las entradas de cada neurona y es muy difícil, cuando no imposible, referir el resultado de la clasificación a uno o varios motivos. Contrariamente, el resto de los clasificadores aquí tratados permiten referir con facilidad el resultado de la clasificación a una o varias anomalías en el vector de entrada y almacenan el modelo de la cadencia de tecleo del usuario bajo una representación explícita muy similar a los vectores de entrenamiento; por ejemplo, en todos los clasificadores que utilizan espacios métricos, el modelo del usuario es un vector promedio con los correspondientes desvíos estándar de los componentes, mientras que dado el rechazo de un intento de autenticación es trivial identificar los componentes que más se alejan de la media. Es claro que esta consideración carece de utilidad en los sistemas cuya finalidad última es la identificación de usuarios, pero para el análisis forense de las cadencias de tecleo resulta necesario contar adicionalmente con una representación explicitable que justifique la decisión del clasificador.

Obaidat y Sadoun (1997) han comparado múltiples tipos adicionales de redes con una cantidad variable de capas ocultas y estrategias de aprendizaje tan diversas como *backpropagation*, *counterpropagation* y LVQ (entre otras) reportando errores de clasificación muy cercanos a cero e incluso valores nulos de FAR para algunas como RBFM y ARTMAP, aunque no todas resultaran tan promisorias e incluso algunas puntuaran por debajo de otros clasificadores utilizados como caso base.

2.5.4.2 Lógica difusa

Junto con el perceptrón de tres capas y el conteo de valores atípicos, Haider *et al.* (2000) han propuesto un clasificador novedoso basado en conceptos de lógica difusa, donde al tiempo medido para cada parámetro característico de cada tecla se le asignan grados de

pertenencia entre cero y uno a cinco clases posibles (muy rápido, rápido, moderado, lento y muy lento) de acuerdo a su cercanía al centro de un intervalo arbitrariamente elegido por los autores. Al conjunto de vectores de entrenamiento, conformados por la sucesión de dichos grados de pertenencia, se le calcula el centro de masa y se asigna una de las categorías al resultado para cada componente. Luego se clasifica como válido al usuario solo si cada tiempo registrado en el vector característico pertenece al intervalo correspondiente a la categoría que se obtuvo al calcular el centro de masa del conjunto de vectores de entrenamiento.

Con un FAR y FRR de 19% y 11%, respectivamente, para dos intentos sucesivos en el ingreso de la clave, este último ascendiendo a 26% para un único intento, el clasificador basado en lógica difusa no se encuentra en condiciones de competir con los más destacados.

Consistentemente, los resultados de Killourhy y Maxion (2009) lo sitúan antepenúltimo entre los clasificadores considerados.

2.5.4.3 Árboles de decisión

El enfoque y fundamento teórico del uso de árboles de decisión (ver sección A.4 para una descripción sobre los mismos) como herramientas clasificadoras difiere notablemente del que corresponde a espacios métricos. Mientras que, en estos últimos, todos los parámetros aportan por igual una magnitud correspondiente a su desviación normalizada respecto de la media y no se puede, por tanto, considerarlos más o menos significativos salvando el factor de escala que introduce el desvío estándar. Para la construcción de árboles de decisión se pretende identificar un subconjunto de los parámetros o de los valores que estos toman, sin presuponer una distribución de probabilidad para los mismos, que sean significativos para la clasificación, jerarquizando implícitamente algunos de ellos. Un árbol de decisión no es único para un conjunto de entrenamiento dado; existen diversos algoritmos para su generación, que producen resultados variables respecto de la robustez en la clasificación, tolerancia al ruido y otros parámetros de importancia. Una referencia exhaustiva del tema, con aplicaciones a la minería de datos puede encontrarse en Rokach (2008).

Es común utilizar más de un árbol de decisión simultáneamente para evitar el sesgo introducido por el algoritmo de generación, por ruido en los datos o simplemente por el sobreentrenamiento. Una forma óptima de agrupar árboles de decisión en lo que respecta a la clasificación de cadencias de tecleo será descrita en la sección siguiente; existen muchas otras, con ventajas y desventajas según el conjunto de datos al que se apliquen, que pueden consultarse en la misma referencia anterior.

La autenticación de claves estáticas con clasificación de la cadencia de tecleo por medio de árboles de decisión ha sido propuesta por Sheng *et al.* (2005), obteniendo resultados promisorios. Para la clasificación se particionan los vectores característicos y se generan hasta ocho árboles de decisión basados en subconjuntos de los parámetros; el proceso valida al usuario como legítimo si al menos tres de los árboles así lo convalidan. Un FAR del 0,88% con una tasa de rechazos del orden del 10% pueden obtenerse con frases de entre treinta y cuarenta caracteres, que debe considerarse una longitud mínima, ya que la eficiencia en la clasificación decae fuertemente al disminuir dicho largo. Los autores notan que no es requisito para una implementación práctica exigir una clave de tal extensión, sino que basta con considerar nombre de usuario, clave, apellido y nombre del sujeto registrado para llegar a la cantidad de caracteres necesaria.

Es destacable que solo nueve vectores de entrenamiento por usuario sirven para alcanzar el rendimiento mencionado. Las técnicas utilizadas por los autores para lograr una exigencia tan baja son dos. En primer lugar, la generación de nuevos vectores característicos en forma aleatoria utilizando la distribución implicada por el conjunto de entrenamiento permite, contrariamente a lo que sucedería en los métodos que utilizan espacios métricos, evitar el sobreentrenamiento del árbol, lo que ilustra lo dicho más arriba respecto de la diferencia fundamental en el enfoque teórico de ambas técnicas. En segundo lugar, la representación del vector característico en una base ortogonal obtenida por medio de *wavelets* o de una transformada discreta de fourier agrega un grado de variedad en la representación de la entrada, que hace posible variar el FAR y el FRR con mayor flexibilidad y así alcanzar un compromiso razonable.

A modo de ejemplo, en la figura A.5 de la sección A.4 se muestra un árbol equivalente al de Sheng *et al.* (2005) para el vector $\mathbf{v} = (v_1, \dots, v_{58})$, este corresponde a uno de las ocho particiones de los parámetros característicos, por lo que se observa que no todos los valores v_i aparecen en él; lo que es más, ni siquiera aparecen los siete u ocho (según el redondeo) parámetros esperables, lo que muestra que el algoritmo de generación del árbol consideró irrelevantes a todos los demás.

2.5.4.4 Bosques aleatorios

Maxion y Killourhy (2010) aplican el algoritmo de bosques aleatorios al problema de clasificación de cadencias de tecleo en un ambiente poco convencional. En lugar de utilizar un teclado estándar, estudian la posibilidad de reconocer usuarios en base a claves cortas tipo *pin* ingresadas en un teclado numérico, como los que pueden encontrarse en cajeros automáticos o terminales de autenticación para tarjetas de crédito y débito. Este ambiente presenta en forma simultánea diversos problemas metodológicos, ya que, además de utilizarse claves cortas (once números), la variedad de las mismas está restringida al mínimo por el tipo de teclado y la disposición geométrica de las teclas hace que la mayoría de las veces los usuarios utilicen un único dedo para el ingreso de información, lo que hace esperar mayor similitud en los patrones de tecleo. A pesar de las complicaciones, los resultados son sorprendentemente robustos; un FAR de 0,46% *versus* un FRR de 12,5% posicionan a la clasificación con bosques aleatorios entre las más destacadas de todos los algoritmos aquí reseñados. Adicionalmente, el EER de 1,45% alcanzado es uno de los mejores reportados en la literatura. Es importante notar que para alcanzar un tal FAR se utilizaron cien repeticiones de entrenamiento, lo cual constituye un número exageradamente alto para una implementación práctica, pero que, a pesar de ser mayor, no dista demasiado del número usualmente empleado para evaluar otros clasificadores. Para una mayor descripción sobre bosques aleatorios, referirse a la sección A.3.

2.5.5 Fusión de esquemas

De la misma forma que en los métodos de ensamble como los bosques aleatorios se reduce el error de clasificación mezclando los resultados de múltiples clasificadores relacionados, el principio se puede aplicar a distintos algoritmos que no tengan relación entre sí. Haider *et al.* (2000) reportan una mejora consistente en el FAR y FRR al utilizar en conjunto métodos tan disímiles como la lógica difusa, las redes neuronales y el conteo de valores atípicos, a pesar de que algunos de ellos den por separado resultados poco alentadores. Son similares las conclusiones en Teh *et al.* (2007), donde la combinación de distancias normalizadas y un clasificador de propósito específico supera a estos métodos por separado. En este último, se aplica una suma ponderada de los resultados de ambos clasificadores variando los pesos para cada usuario.

2.6 Detección de estados emocionales con cadencia de tecleo

Un breve apartado de Monroe y Rubin (1997) puede considerarse el primer intento en la disciplina de inferir información adicional a la identidad del usuario. Los autores notan que entre los participantes del experimento se puede distinguir a los zurdos de los diestros ordenando las latencias de los digrafos, ya que en los primeros todas aquellas que corresponden a presiones sucesivas de teclas del lado izquierdo del teclado tienen en conjunto un promedio de duración menor que las del lado derecho e inversamente en los segundos. No se han llevado adelante investigaciones adicionales para elucidar qué otras características físicas del usuario pueden inferirse a partir de patrones de gran escala que emerjan de un conjunto de datos de entrenamiento.

La influencia de factores emocionales ha sido estudiada con mayor profundidad. Vizer *et al.* (2009) analizando el efecto del estrés físico y cognitivo sobre los patrones de tecleo, concluyen que pueden inferirse las variaciones graduales o repentinas del estado del usuario en esta dimensión emocional con la suficiente confiabilidad como para utilizarlo en aplicaciones médicas en las que este factor pueda ser de

interés aunque no crítico. Más general es el enfoque de Epp *et al.* (2011), donde se intentan clasificar una docena de dimensiones emocionales entre las que se incluyen aburrimiento, frustración, felicidad, ansiedad y otras utilizando una metodología de muestreo en la cual se interrumpe aleatoriamente la tarea cotidiana del usuario para solicitarle que reporte su estado emocional en el momento para luego asociarlo con el registro de tecleo. Los resultados son promisorios, ya que, a pesar de las dificultades metodológicas y de utilizar un modelo muy sencillo, se logra una eficacia en la clasificación cercana al 80%. Una reseña actual de métodos y resultados obtenidos en la educación de emociones utilizando cadencias de tecleo puede consultarse en Kořakowska (2013).

La aplicación de estas técnicas está siendo utilizada para realizar reconocimientos biométricos y como un segundo método de identificación del usuario; sin embargo, en otras áreas relacionadas con factores emocionales —como el marketing, control inteligente de juegos (Lv *et al.*, 2008), entre otras— se encuentra en sus albores, habiendo limitadas publicaciones y siendo las más importantes Epp *et al.* (2011); Kořakowska (2013); Lv *et al.* (2008).

En Kořakowska (2013) se realizó un estudio comparativo (ver el cuadro 2.2) entre todos los métodos existentes hasta 2013, a los cuales se agregaron Nahin *et al.* (2014); Shukla y Solanki (2013). En esta comparación se encuentran —entre otras cosas— las emociones detectadas, los métodos de entrada y los resultados obtenidos. Como puede verse en el cuadro, no hay resultados obtenidos en la educación de emociones mediante métodos que aplican EEG en el contexto de aplicaciones de cadencia de tecleo. La articulación de ambas ramas será un aporte de la presente tesis.

Ref.	Emociones analizadas	Método de Adquisición	Origen de características	Modelo	Método de etiquetado	Método	Resultados	Aplicación
Zimmermann et al. [2006]	PVHA, PVLA, NVHA, NVLA, nVnA	video	keystrokes, vimientos mouse	model	cuestionario (SAM)	análisis estadístico	conclusión: posible discriminación entre categoría neutral y otras cuatro.	compras online
Epp et al. [2011]	anger, boredom, confidence, distraction, excitement, focused, frustration, happiness, hesitance, nervousness, overwhelmed, relaxation, sadness, stress, tired	—	keystrokes		cuestionario (5 puntos en la escala de Likert) llenado de vez en cuando	C4.5 (un clasificador de dos clases para cada emoción)	precisión de clasificación: 77,4–87,8% para algunas emociones (confidence, hesitance, nervousness, relaxation, sadness, tiredness)	actividades diarias habitual en la computadora, p. ej. procesador de textos, aplicación de correo
Althothali [2011]	confusión, aburrimiento y frustración en el caso de la valencia negativa; deleite y neutral en el caso de valencia positiva	—	keystrokes, quality of responses, biographical data		cuestionario (5 puntos en la escala de Likert) llenado por participantes y dos jueces	dos etapas: 1) positive vs. negative valence; 2) reconociendo seis emociones; métodos probados: análisis discriminante, naive Bayes, k-NN, árboles de decisión, redes neuronales	mejor precisión obtenida para redes neuronales: 72,02– 82,82% en la primera etapa; 45,60–53,89% en la segunda etapa	intelligent tutoring system
Lee et al. [2012]	happiness, surprise, anger, disgust, sadness, fear, neutral	—	keystrokes, incomodidad, tiempo, clima	parámetros de la pantalla táctil, índice de ubicación,	red continuamente actualizada con nuevos datos	bayesiana actualizada	67,52% en promedio, la mejor precisión para happiness, surprise y estado neutral	teléfonos móviles
Vizer et al. [2009]	stress	tareas que inducen estrés cognitivo y físico	keystrokes, parameters	language	cuestionario (11point Likert scale)	árboles de decisión, SVM, k-NN, AdaBoost, redes neuronales	75,0% para estrés cognitivo (k-NN), 62,5% para estrés físico (AdaBoost, SVM, redes neuronales) conclusion: number of mistakes made while typing decreases under stress	
Sottolare y Proctor [2012]	busca cualquier influencia en el estado de ánimo y el rendimiento	—	movimientos del mouse		cuestionario (SAM)	análisis estadístico	conclusiones: existe una correlación entre la variable placer y la tasa de selección de control; correlación entre el rendimiento en la aplicación del conocimiento adquirido y la tasa de movimiento del mouse y el sistema de tutoría inteligente	

Cuadro 2.2: Reconocimiento de emoción basado en información obtenida de dispositivos estándares (continuado).

Ref.	Emociones analizadas	Método de Adquisición	Origen de características	Método de etiquetado	Método	Resultados	Aplicación
Khanna y Sasikumar [2010]	positive, negative, neutral	textos fijos para ser retipeados	keystrokes	cuestionario	regresión logística lineal, SVM, redes neuronales, C4.5, árbol aleatorio, árbol binario	efectividad 89,02% (negative vs. neutral usando un árbol binario); 88,88% observación (positiva vs. neutral utilizando C4.5): La velocidad de tipeo disminuye en el caso de las emociones negativas	
Lv et al. [2008]	anger, fear, happiness, sadness, surprise and neutral	historias para ser escuchadas	keystrokes, secuencias de presión del teclado	etiquetas asignadas según el tipo de historia escuchada	k-NN con diferentes medidas de distancia (incluyendo DTW) dependiendo de los subconjuntos de características y una combinación de medida de distancia combinada de clasificadores basados en distancia, DTW	efectividad 93,4% promediado sobre estados emocionales usando una medida de distancia combinada	
Schuller et [2002]	surprise, joy, anger, fear, disgust, sadness, neutral	navegador de internet simulado provocando emociones	movimientos del mouse, touchscreen parameters, speech	combinación de clasificadores basados en la distancia, DTW y HMM	de efectividad: 76–95% dependiendo del tipo de emoción	interfaz del navegador de internet	
Maehr [2005]	sadness, happiness, neutral	videos	movimientos del mouse	cuestionario online	análisis estadístico	conclusiones: una menor excitación significa movimientos más precisos y suaves y menos clics; mayor excitación significa mayor aceleración, menor precisión y menos interrupciones	
Tsouluhas boredom et [2011]	al.	—	movimientos del mouse, tipo de objeto de aprendizaje	preguntando a los alumnos si están aburridos	C4.5	more than 90%	sistema inteligente de tutoría
Nahin [2014]	Joy, Fear, Anger, Sadness, Disgust	al.	cuestionario keystrokes y patrones de texto	modelo de espacio vectorial y método de similitud Jaccard para clasificar la entrada de texto libre		encima de 80% en la identificación de emociones	

Cuadro 2.2: Reconocimiento de emoción basado en información obtenida de dispositivos estándares.

2.7 Consideraciones técnicas

La técnica de reporte de resultados en la literatura histórica sobre cadencias de tecleo consiste en señalar el FAR y el FRR —o alguna de las otras métricas— que se alcanzan al clasificar una cantidad de datos que se ha separado en un conjunto de entrenamiento y un conjunto de pruebas para evaluar el clasificador luego de construidos los modelos de los usuarios. Solo recientemente ha virado el rumbo de la investigación en el sentido de intentar comprender, modelar y superar las dificultades metodológicas inherentes al manejo de una cierta clase de información biométrica comportamental que está sujeta a toda clase de influencias que agregan ruido a los modelos y complican las inferencias que se pretende realizar en base a ella. Las cadencias de tecleo son fundamentalmente ruidosas; no solo nos enfrentamos a las variaciones impredecibles e incontrolables que se produzcan por cambios fisiológicos del usuario como cansancio, estados emocionales y consumo de medicinas u otras sustancias que puedan afectar el desempeño motor en forma sutil, sino que tomamos la información de un ecosistema de teclados fijos, portátiles, táctiles, etc., con distintas configuraciones y además pretendemos integrar el modelado de cadencias de tecleo en sistemas prácticos.

Al momento de formular el experimento de captura de la información, entran en juego consideraciones metodológicas respecto de la psicología. ¿Es idéntica la manera de escribir la clave para un usuario cuando está consciente de estar siendo estudiado, que cuando la ingresa en el curso habitual de su desempeño diario? No solo parece evidente que sí, sino que probablemente el mismo desempeño diario tenga mayor variación que la que hay entre este y el experimento. ¿Qué procesos de mediano o largo plazo modifican la cadencia de tecleo y en qué medida lo hacen? A pesar de ser una consideración fundamental para la construcción de modelos adaptables, ninguno de los experimentos reportados tiene un alcance temporal que llegue al año de seguimiento de algunos usuarios como para considerar estas variaciones.

Desde otro punto de vista más afín, la inserción de métodos de autenticación o auditoría basados en cadencias de tecleo tiene consideraciones prácticas que deben ser resueltas. Poca utilidad tiene

un FAR cercano a cero si requiere un enrolamiento que demande cien repeticiones de la clave o escribir libremente veinte páginas y, sin embargo, la mayoría de los reportes con excelentes métricas suelen caer en la trampa de no señalar la velocidad de convergencia del modelo. Idénticamente, se puede conseguir una mejora artificial evaluando la totalidad de los ingresos en un único teclado para disminuir la varianza de las muestras, pero este enfoque no permite predecir los resultados que se obtendrían en un ingreso remoto o a través de la web.

Las consideraciones técnicas del experimento de captura de datos para analizar un esquema de modelado, pueden responder a dos ejes. En primer lugar tenemos la estructura cuantitativa del experimento, que está dada por la cantidad de usuarios, las repeticiones exigidas a cada uno, la extensión y variedad del texto solicitado; también se incluye en esta categoría la extensión temporal del experimento, la duración de las sesiones de usuario y el tiempo transcurrido entre sesiones. La diferencia en estos y otros parámetros que se puede encontrar en distintos reportes resulta sorprendente cuando uno considera cuán sensibles son los resultados de la clasificación a todas estas variables. En Giot *et al.* (2011) se resumen las diferencias encontradas en la conformación de los experimentos; en Killourhy (2012) se encuentra una revisión extremadamente completa de la literatura agrupando los reportes según todos los anteriores.

El otro eje fundamental en la confección de los experimentos de captura de datos surge de considerar que los mismos no se realizan sobre procesos inertes, sino sobre seres humanos y, por lo tanto, entran en juego aspectos psicológicos y comportamentales. Joyce y Gupta (1990) dividen a los voluntarios en dos grupos, informando solo a uno de ellos la finalidad y el método del experimento aunque solicitándoles que intenten no modificar su cadencia de tecleo. Aunque la intención es válida, no se reporta el efecto diferencial sobre los dos grupos o la ausencia de él; lo que es más, resulta esperable que la solicitud no pueda tener sino el efecto contrario al esperado, como lo puede demostrar la extensa literatura psicológica sobre la parcialidad y la necesidad de experimentos ciegos y doble ciegos. No debe recriminarse a los autores por la omisión sino que, inversamente, se los debe elogiar por considerar, si bien parcialmente, una posible fuente de error que por desgracia no ha recibido la atención que se merece.

Killourhy (2012) hace notar un sesgo común en los grupos de voluntarios, que es su pertenencia a un rango etario, profesional y de habilidad de tecleo extremadamente restringido.

Generalmente limitados por el presupuesto u otras condiciones, los experimentadores toman voluntarios en el ámbito de la institución en la que se desempeñan quitando diversidad de la base de usuarios a evaluar. Afortunadamente, esta fuente de sesgo ha sido ampliamente reconocida y en la literatura reciente se explicitan los esfuerzos tendientes a minimizarla. En el caso particular de la educación de emociones, donde algunos experimentos (Epp *et al.*, 2011; Vizer *et al.*, 2009) utilizan informes provistos por el propio usuario en forma espontánea como fuente de datos, nos encontramos con nuevas fuentes de error propias de esta metodología. Un revisión muy completa de las fuentes de error en experimentos relacionados con el comportamiento humano y posibles soluciones puede verse en Podsakoff *et al.* (2003); el lector técnico queda advertido que dicho artículo pertenece al ámbito de las ciencias psicológicas y puede resultar poco accesible para el que no está versado en ellas.

2.7.1 Fuentes de error

La tesis doctoral de Killourhy (2012) y su artículo previo (Killourhy y Maxion, 2010) analizan detalladamente las fuentes de error en la clasificación utilizando la técnica de modelado lineal de efectos mixtos, presuponiendo una distribución normal para las variables implicadas. Los modelos resultantes son complejos y abundan en parámetros, entre los que podemos mencionar rasgos del usuario como su género, las condiciones físicas de las manos y sus estilos de tecleo además de características del clasificador y de los impostores. Entre las conclusiones alcanzadas el evaluar extensivamente estos modelos destaca una que, siendo relativamente reciente, debería tener un efecto significativo en la manera de encarar el problema del modelado de cadencias de tecleo. El autor concluye que, si bien hay diferencias de calidad entre los clasificadores —lo cual está suficientemente reflejado en la literatura—, los factores experimentales y las variaciones propias del usuario tienen mayor influencia en las métricas de eficiencia y producen fluctuaciones en las tasas de error tan significativas como para poder afirmar que no tiene sentido hablar de

una única tasa de error en un clasificador dado. Propone, en su lugar, factorizar otras condiciones y hablar de la tasa de error de un clasificador para un cierto usuario bajo ciertas condiciones experimentales. Cuantitativamente, deberían considerarse miles de sujetos para poder establecer la tasa de error con una precisión del orden del uno por ciento y solo una vez que se han estabilizado los efectos de los factores experimentales. Por otro lado, es optimista la conclusión de que la mayoría de los rasgos del usuario, si bien pueden ser utilizados para clasificaciones secundarias, no tienen influencia en la tasa de error del clasificador. Los factores más influyentes resultan ser el estilo de tecleo del usuario y sus cambios pero sobre todo el tipo de impostor al que se enfrenta el sistema. Ambos han sido virtualmente ignorados en la literatura y prometen líneas de investigación fructíferas.

Respecto de la variación temporal en la cadencia de tecleo, no existen experimentos de larga duración ni modelos teóricos planteados que permitan extraer conclusiones. Destaca entre las conclusiones de Killourhy (2012) el hecho de que la utilización de esquemas con adaptación gradual de los modelos mejora consistentemente la clasificación de manera transversal entre clasificadores, usuarios, estilos, impostores y tipos de texto. Parece ser evidencia suficiente de la existencia de efectos de largo plazo que modifican duraderamente los tiempos de presión y latencias entre teclas, u otros arcos de comportamiento aún no analizados y que pueden ser explotados para mejorar el modelado y disminuir el error de clasificación.

2.7.2 Influencia del idioma

El hecho de que el usuario utilice múltiples idiomas durante su sesión no parece ser un problema insalvable, al menos si se utiliza el clasificador de Gunetti *et al.* (2005). La conclusión de los autores luego de analizar ingresos de texto en italiano e inglés realizadas por 31 voluntarios es que, si bien los fragmentos de diferentes idiomas de un mismo usuario se diferencian más que los del mismo idioma, ambos lo hacen menos que los de cualesquiera otros usuarios. Los valores de FAR y FRR obtenidos, en el orden de 2% y 5%, si bien son peores que los que se logran al comparar textos del mismo idioma, no dejan de ser

promisorios considerando que este es el primer estudio al respecto de la influencia de los idiomas en la clasificación.

2.7.3 Variaciones en los teclados

Los experimentos de clasificación y verificación de usuarios suelen realizarse bajo la consideración de que el computador físico en el que se pretende iniciar sesión no sufre reemplazos del teclado ni modificaciones en la distribución lógica del mismo. Esta consideración es razonable hasta cierto punto cuando las sesiones de usuario se ejercen en un lugar invariable, pero deja de ser válida al enfrentarse a sesiones remotas, por ejemplo en sitios web, que pueden iniciarse desde teclados con disposiciones físicas o configuraciones lógicas lo suficientemente diferentes como para influir en la cadencia de tecleo. Como ejemplo de variaciones en la primera disposición, podemos mencionar los teclados ergonómicos o la distribuciones AZERTY —muy común en países francoparlantes— y QWERTZ; sin ir tan lejos, ya los teclados de escritorio y los teclados de portátil difieren notoriamente entre ellos, y más aún estos últimos entre sí. Del segundo caso, todo programador puede atestiguar sobre la merma en productividad que conlleva el uso temporario de una terminal cuya configuración regional no es la acostumbrada. Más en general, el ingreso de una clave que incluya símbolos debe necesariamente verse afectado ya que la posición de estos últimos, a diferencia de las letras y números, suele ser la que varía con mayor frecuencia.

Al figurarse la variabilidad indicada surgen dos interrogantes. ¿Es posible conservar la tasa de error de clasificación ante variaciones en la configuración física o lógica del teclado? ¿Es posible, considerando las variaciones producidas en la cadencia de tecleo, inferir características del teclado utilizado? Esta última pregunta no carece de interés, ya que tiene implicancias para el análisis forense. Los resultados de Giot *et al.* (2012) señalan un EER notoriamente peor al plantear un experimento más realista en donde los usuarios pueden ingresar a un sistema web desde múltiples localizaciones, habiendo libremente elegido en forma previa una clave propia y en donde la separación temporal entre intentos de ingreso es del orden de una semana.

Exceptuando unos pocos contraejemplos (Epp *et al.*, 2011; Gunetti y Picardi, 2005), los experimentos para el análisis de clasificadores y otras técnicas en este campo se han realizado sobre teclados estándar de escritorio o de notebook, generalmente capturando los conjuntos de entrenamiento de todos los usuarios con el mismo teclado. En Maxion y Killourhy (2010) se ha estudiado la posibilidad de verificar usuarios utilizando exclusivamente un teclado numérico y una clave corta. Incluso para los usuarios que ingresaban su PIN con un único dedo la precisión en la clasificación lograda, con un FAR del 0,26%, es sorprendentemente alta aunque debido al FRR de 7,14% no logra el objetivo de los autores, que es alcanzar un costo de error suficientemente bajo para que esta técnica pueda ser utilizada como único mecanismo de autenticación cumpliendo el estándar europeo EN-50133-1.

Algunos ejemplos donde se utiliza un parámetro no temporal son Tang *et al.* (2010) y Kotani y Horii (2005). En ambos se evalúan teclados especialmente diseñados para el experimento, que miden la curva de fuerza aplicada sobre las teclas a través de una película resistiva sensible a la presión y son indistinguibles de un teclado convencional, táctil y visualmente. Un aspecto interesante de este enfoque es que la presión utilizada es un parámetro más difícil de imitar por parte de un impostor entrenado, que solo escuchando al usuario legítimo ingresar su clave podría inferir las relaciones temporales entre teclas. Saevanee y Bhatarakosol (2009) concluyen que los patrones de la presión dactilar son más efectivos que los temporales para discriminar usuarios válidos de impostores en pantallas táctiles.

Descripción del problema y objetivos

Durante el desarrollo de la presente tesis, se justificará una hipótesis general. Pero para desarrollar el experimento que la valida es necesario previamente verificar cada uno de los pasos que nos permitieron diseñar el experimento. Para ello será necesario el desarrollo de un *framework* que permita no solo la estimulación emocional de la persona, sino también la adquisición de los patrones de tecleo y las ondas cerebrales de la persona. Además, es necesario procesar toda esa información y analizar previamente otros conjuntos de datos externos al experimento para validar métricas y definir el tamaño correcto de la muestra a adquirir durante la ejecución de un experimento. A esto no se puede llegar si no se define debidamente cuál es el problema a atacar y cuáles serán los objetivos y alcances de esta tesis.

3.1 Descripción del problema

Los sistemas de identificación y reconocimiento de personas mediante la cadencia de tecleo son ampliamente utilizados; existen varios estudios que muestran diferentes tasas de errores para distintas situaciones y métricas aplicadas; sin embargo, no existen estudios que muestren el efecto de los estados emocionales sobre la capacidad de detectar a las personas en términos de patrones de tecleo en el contexto de marcadores biométricos de EEG. Es de gran significancia entender cómo una persona exaltada, deprimida, relajada, etc. podría tipear más rápido, más lento o simplemente modificar su patrón para así poder ajustar los márgenes de error de esta técnica.

3.2 Objetivos

Crear un marco de trabajo que permita capturar el patrón de tecleo del usuario frente a estímulos generados, adquirir sus ondas cerebrales y procesar los datos adquiridos. Con el empleo de BCI, el EEG se transformará en valores dimensionales según el modelo de Russell permitiendo su comparación con valores encuestados de SAM.

Con la información obtenida se desea contrastar el patrón de tecleo y las ondas cerebrales para observar la variabilidad en la tasa de error al identificar personas, frente a diferentes estímulos emocionales.

3.3 Hipótesis

La contribución de esta tesis se centra en la cadencia de tecleo, analizando la robustez de los algoritmos frente a variaciones en registros biométricos a través de EEG, recurriendo a la obtención de esta información con el empleo de una interfaz cerebro-computadora (BCI) y modelos emocionales no categóricos —más específicamente, el modelo de Russell—, en la explotación en el campo de las neurociencias aplicadas. La hipótesis central de esta tesis se puede sintetizar como:

Los algoritmos de identificación de personas que utilizan cadencia de tecleo continua son robustos frente a variaciones cerebrales asociadas a cambios emocionales de relajación y excitación.

Esta tesis solamente se centrará en analizar la robustez de los algoritmos frente a variaciones en registros biométricos a través de EEG en particular ondas alfa, beta o aquellas asociadas, según la literatura (Garcia *et al.*, 2016; Li *et al.*, 2016; Liu *et al.*, 2016; Oude Bos, 2006; Petrantonakis y Hadjileontiadis, 2010; Samara *et al.*, 2016; Shin *et al.*, 2016; Shu *et al.*, 2018; Verma y Tiwary, 2014; Zhang *et al.*, 2016; Zheng *et al.*, 2017), con estados de relajación y excitación. También se utilizará un autoinforme (mediante SAM) para contrastar los resultados. Adicionalmente, se liberan futuras líneas de investigación en la correlación entre estas ondas y diversos estados emocionales.

Se explorarán, inicialmente, resultados a los fines de conformar el grado de robustez, frente a la variabilidad de situaciones de excitación o relajación del individuo que se encuentra tecleando.

3.4 Alcance

Se desarrollará un prototipo capaz de leer, comprimir, almacenar y clasificar patrones de tecleo. También será capaz de adquirir, almacenar y analizar señales de EEG y respuestas del usuario (SAM) estimulando al usuario con imágenes, audios y videos de alto contenido emocional. Se analizará el efecto que los estados afectivos pueden ejercer sobre la tasa de error de los algoritmos de identificación de personas por cadencia de tecleo.

Diseño del experimento

Definiremos en el presente capítulo la solución que se propone y el diseño del experimento. Para realizar el experimento fue necesario diagramar la adquisición de los datos de forma detallada, realizar pruebas iniciales para minimizar la posibilidad de errores durante la captura real y desarrollar una arquitectura de adquisición que permita la sincronización temporal de todos los dispositivos involucrados. El producto final fue un experimento cuya adquisición se dividió en una fase preliminar adaptativa que es solamente para adaptación del usuario, y los datos recogidos durante esta fase no fueron considerados para el desarrollo del experimento de esta tesis. De todas formas, los mismos podían utilizarse para futuras exploraciones. El objetivo de la fase preliminar adaptativa es neutralizar la emoción que el individuo traía consigo, por este motivo también la denominaremos fase neutral.

Luego siguen de cuatro fases iterativas que realizaban las capturas necesarias. La sección 4.1 desarrolla la metodología de adquisición, su arquitectura y las fases utilizadas. La sección 4.1.1, en cambio, define la solución propuesta en forma de la mecánica a utilizar para el procesamiento de los datos obtenidos y validar la hipótesis. La sección 6.1 describe un caso de prueba realizado previamente y las lecciones aprendidas a la hora de realizar el experimento. Finalmente, en la sección 4.3 se explica la ejecución de la captura y sus participantes.

4.1 Adquisición de datos

Para realizar la adquisición de datos, las sesiones individuales fueron organizadas para durar aproximadamente una hora. Para cada fase, los participantes son estimulados por imágenes obtenidas de IAPS (Lang *et al.*, 2008) con un fuerte contenido emocional (ver sección 2.3.5 para más detalles). De este conjunto de imágenes, se conoce la ubicación en el circunplejo de Russell debido al trabajo de Lang *et al.* (2008).

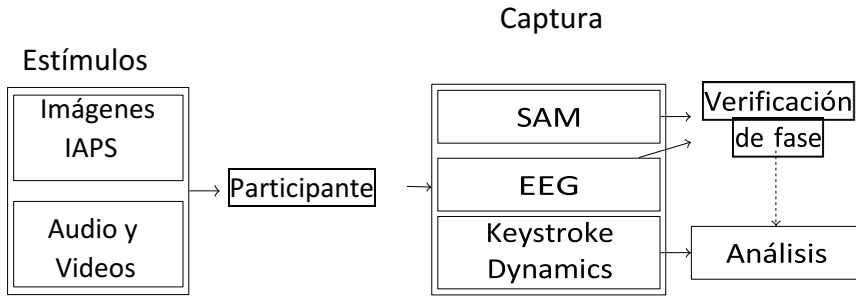


Figura 4.1: Para cada fase, se envían estímulos específicos al individuo, quien proporcionará las capturas multimodales.

También son utilizados un audio o video cuyo contenido se encaja en cada uno de los cuadrantes. En los primeros tres cuadrantes se utilizaron videos cuyo contenido era altamente emocional —tristeza: un vagabundo que cuida un local y termina siendo asesinado; alegría: un compilado de *bloopers* con caídas; calma: un video muy relajante de un *drone* produciendo luces sobre un lago—, para el cuarto cuadrante se utilizó un audio con un llamado muy perturbador al 911 que ya fue probado en Ierache *et al.* (2015), este cuadrante se dejó para el final por ser el más difícil de olvidar.

Para la captura de datos, se utilizan varios métodos. El apéndice D detalla los mismos. La figura 4.1 detalla la arquitectura del experimento. Los tres métodos de adquisición utilizados en el análisis de esta tesis son: keystroke dynamics, SAM (ver sección D.2.3) y EEG (ver sección 2.3.2). La captura de EEG es realizada permanentemente mientras que SAM (ver sección D.2.3) y KD son adquiridos al finalizar la estimulación: primero se realiza la encuesta SAM y rápidamente se procede a solicitar al usuario a que tipee una lista de frases.

El objetivo de EEG y SAM es validar el cambio de estado emocional de los individuos, es decir, nos permitirían rechazar la muestra si el usuario no registra variaciones.

La iteración a lo largo de las fases y su respectivo orden se describe en figura 4.2, y representa la gráfica ondas α obtenidas del electrodo AF3 para el participante número 6 mostrando, en el eje temporal, todas las estimulaciones pertenecientes a la fase neutral y las cuatro fases estimuladas; esta figura da una buena idea de los tiempos que transcurren entre estímulos.

El paso 0, fase neutral o de inicial adaptativo, consiste en una inducción previa donde el sujeto es transportado a un estado neutral. Para ello, se prepara un momento de silencio y relajación. Luego, se ejecuta un videojuego sencillo con el objetivo de contracturar al participante y hacerlo olvidarse del entorno en el que se encuentra. Después de eso, el usuario rellena un cuestionario con información personal: nombre, fecha de nacimiento e idiomas que conoce. Luego, comienza con la fase HAPV (alta activación, valencia positiva) y se progresa a través de las cuatro fases hasta HANV (alta activación, valencia negativa). Para cada fase, que representa un cuadrante específico, se presenta las imágenes de IAPS asociadas, se solicita una encuesta SAM y se solicita la introducción de las frases para leer la dinámica de tecleo. El usuario no es consciente de que se está leyendo su patrón de tecleo. En el apéndice D y en Calot y Ierache (2017) se detalla la configuración del experimento, además de otras adquisiciones realizadas durante el mismo.

4.1.1 Mecánica de procesamiento experimental

El patrón de tecleo para cada persona se debe dividir en dos partes elegidas al azar (parte A y B, ver figura 4.3). Este proceso consiste en separar los valores del *conjunto de eventos* (ver sección 2.4.2) de una manera equilibrada y aleatoria. A continuación, se calcula el EER resultante de clasificar los individuos de la parte A en la parte B. La clasificación se realizó únicamente mediante distancias. Para ello fueron utilizadas seis métricas a saber: A con parámetro 1,25 y R, Camberra, Manhattan, Euclídea y Minkowski con parámetro 0,4. Esta clasificación se ejecutó dos veces comparando cada fase de A con tres de B. La figura 4.4 muestra las fases y cómo se comparan entre sí. Las fases en gris de la parte A se comparan con las fases en gris de la parte B para todos los usuarios y con todas las combinaciones entre sí. De esta manera, el EER producido combinando la misma fase se aísla contra el EER producido al no hacerlo. Para garantizar un tamaño igual de las muestras, se realiza un cambio entre las fases. Por ejemplo, en la prueba “Emociones no combinadas”, la fase i_A de la parte A se compara con las otras tres fases j_B de la parte B (con $i_A = 6 j_B \forall i_A, j_B \in \{1, 2, 3, 4\}$).

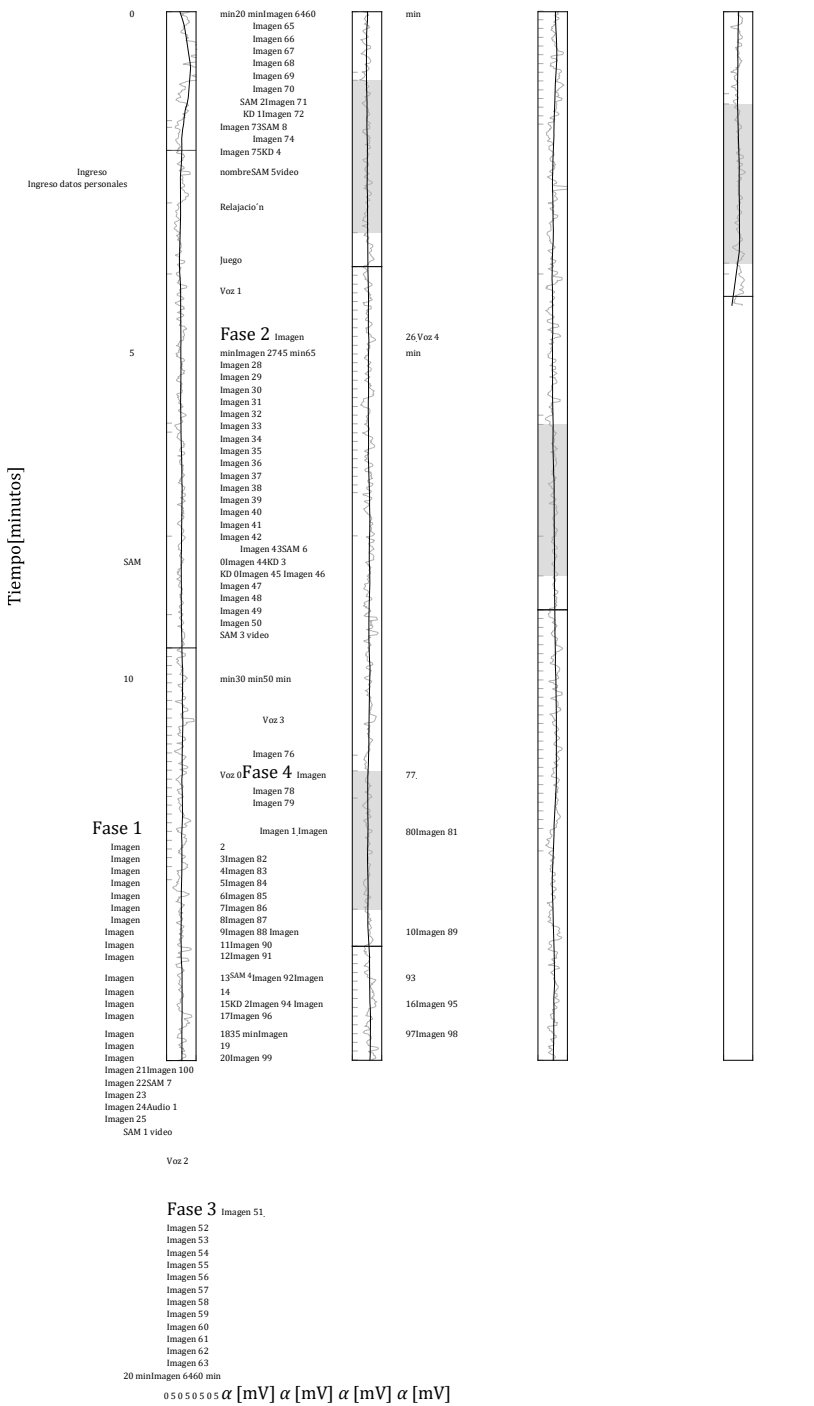


figura 4.2: Línea temporal de captura para el caso del participante 6. Se observa la onda alfa para el electrodo AF3. Se incluye el comienzo de los distintos eventos ocurridos durante la captura.

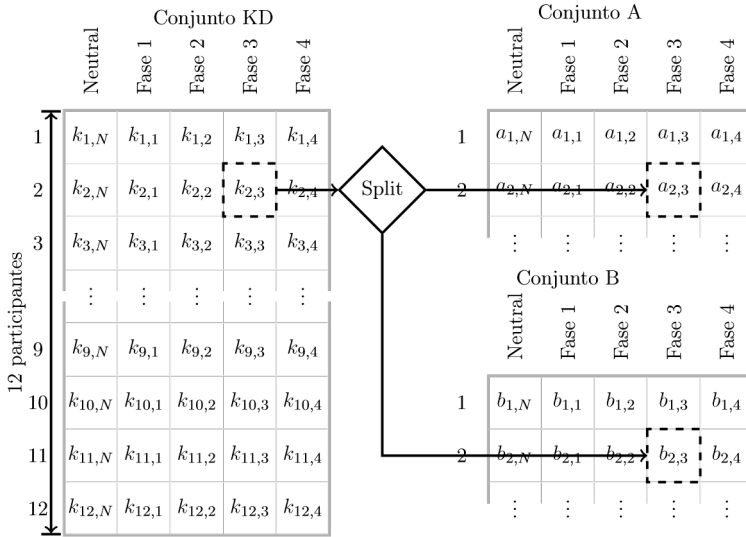


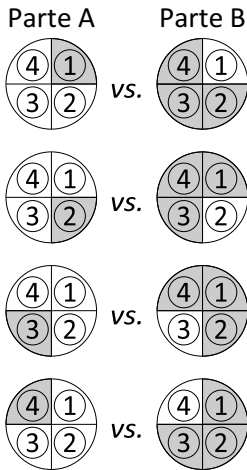
Figura 4.3: Particionamiento de la entrada en dos conjuntos de prueba A y B.

En la prueba “emociones combinadas”, la fase i_A de la parte A se comparó con tres fases j_B de la parte B (con $i_A + 2 \cdot 6 = j_B \forall i_A, j_B \in \{1, 2, 3, 4\}$; módulo 4, para convertirlo en un círculo), que causa $i_A = j_B$ para cada fase para algún i y j .

El procedimiento anterior se realizó diez veces utilizando diferentes semillas aleatorias para obtener valores medios del EER. El azar permitió tomar distintos eventos a la hora de realizar el *split* generando, así, distintos experimentos que se comportan de manera similar. Al reportar los resultados se presentarán los valores medios y desvíos obtenidos para todas las muestras. La cantidad de ejecuciones, fijada en 10 fue un balance entre la necesidad de obtener una cantidad sustancial de ejecuciones pero minimizando el tiempo de procesamiento, que crece de manera proporcional a ellas.

Este diseño nos ayuda a verificar la hipótesis de que existe una variación en la EER media de ambas pruebas; de ser así, sería evidente que los estados emocionales pueden afectar la efectividad.

Emociones no combinadas



Emociones combinadas

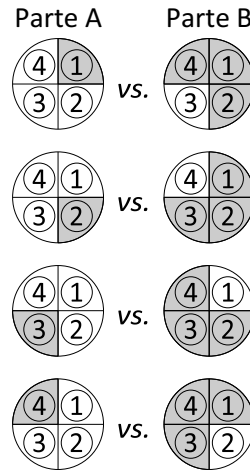


Figura 4.4: Conjuntos de experimentación y fases para el cálculo del EER.

de la clasificación⁴.

4.2 Validación con simulacros de prueba

Previamente a la experimentación, dos casos de prueba fueron realizados para así evitar errores durante la adquisición. Resulta una tarea no trivial conseguir participantes dispuestos a invertir una hora de su tiempo; por esto mismo, se decidió inicialmente realizar simulacros del experimento utilizando investigadores del laboratorio e iterativamente mejorar la configuración del mismo. Finalmente, con solo dos simulacros se obtuvo una adquisición exitosa y se consideró finalizada esta etapa.

⁴ El código fuente del experimento se puede descargar desde <https://github.com/lisia/pocs>.

4.2.1 Lecciones aprendidas

Durante estos simulacros aprendimos que la sincronización de los dispositivos no era una tarea trivial, necesitábamos tener una precisión de milisegundos entre los datos adquiridos por EEG y los datos adquiridos por cadencia de tecleo, video, audio, etc. Para esto, instalamos un servidor que proveía la hora exacta a ser sincronizada mediante el protocolo *Network Time Protocol* (NTP). También decidimos hacer un aplauso al comienzo para sincronizar audio y video, los cuales eran capturados por la computadora utilizada por el usuario. El EEG fue capturado por una segunda computadora, cuyo timer fue sincronizado mediante NTP a partir del segundo simulacro. Al procesar los datos de cadencia de tecleo, se encontró que los tiempos adquiridos no eran relativos al *UNIX epoch* (fecha estándar en UNIX cuyo cero se sitúa al comenzar el 1 de enero de 1970) sino al momento de ejecución del navegador (Mozilla Firefox). Para corregir esto se incluyó el dato de la ejecución en la muestra.

También se detectó un mayor nivel de ruido externo que podría generar distracciones, por lo que se decidió realizar las pruebas en horario vespertino, entre las 19:00 y 21:00 horas.

4.3 Implementación

Se evaluó con esta metodología a 12 participantes voluntarios. Los resultados para cada uno de ellos se detalla en la sección 6.2. Los mismos fueron 10 hombres y 2 mujeres con edades entre 20 y 30 años.

El Ccadro 4.1 muestra un ejemplo del cruce de todos los participantes.

Participante	A		R		Camberra		L ₁		L ₂		L ₂ norm.	
	IDPredicho ¿Clasificó?	Valor	IDPredicho ¿Clasificó?	Valor	IDPredicho ¿Clasificó?	Valor	IDPredicho ¿Clasificó?	Valor	IDPredicho ¿Clasificó?	Valor	IDPredicho ¿Clasificó?	Valor
1	1 X	0.48	1 X	0.31	12 ×	0.07	12 ×	12.21	12 ×	3.80	12 ×	0.38
2	2 X	0.55	2 X	0.36	2 X	0.04	2 X	4.91	2 X	1.69	2 X	0.16
3	6 ×	0.53	4 ×	0.41	3 X	0.06	5 ×	11.62	9 ×	5.59	3 X	0.41
4	4 X	0.52	9 ×	0.32	4 X	0.03	4 X	4.53	4 X	1.93	4 X	0.24
5	5 X	0.58	4 ×	0.35	12 ×	0.07	4 ×	9.24	4 ×	3.98	12 ×	0.31
6	6 X	0.45	6 X	0.25	8 ×	0.06	6 X	10.17	6 X	3.90	6 X	0.25
7	7 X	0.55	2 ×	0.36	4 ×	0.06	4 ×	9.15	4 ×	3.44	4 ×	0.29
8	8 X	0.50	5 ×	0.33	8 X	0.04	8 X	5.29	8 X	2.48	8 X	0.23
9	6 ×	0.55	9 X	0.31	3 ×	0.04	3 ×	8.17	9 X	4.25	12 ×	0.29
10	10 X	0.51	10 X	0.27	3 ×	0.03	3 ×	5.18	10 X	2.16	3 ×	0.20
11	11 X	0.52	4 ×	0.35	11 X	0.05	11 X	7.55	11 X	2.60	11 X	0.18
12	12 X	0.57	12 X	0.38	12 X	0.06	9 ×	9.84	12 X	2.97	12 X	0.19
	10/12		6/12		6/12		5/12		8/12		7/12	

Cuadro 4.1: Resultados de identificación de personas mediante las distancias entre personas de los grupos A y B mediante distintas métricas. Las cruces indican clasificaciones incorrectas.

Validación del experimento

Para aceptar el experimento es necesario primero realizar otras pruebas previas. Estas mismas son necesarias para la elección de las métricas a utilizar y la verificación de que el usuario haya cambiado de estado emocional. Además se requiere configurar ciertos parámetros para distintas métricas de cadencia de tecleo y definir qué valores van a ser analizados. El presente capítulo valida todos estos prerequisites para garantizar una buena ejecución del experimento.

Si se considera que esta tesis tiene tres componentes, este capítulo describe los primeros dos: tomados de Calot *et al.* (2019b) y Calot *et al.* (2019a), respectivamente, en las secciones 5.3 —respondiendo qué métricas se deben utilizar en el análisis de los resultados— y 5.4 —respondiendo qué cantidad de keystrokes es necesaria para que las pruebas sean estadísticamente significativas—. Para analizar estos dos puntos fue necesaria la utilización de dos conjuntos de datos definidos previamente en la sección 5.1. La solución propuesta relacionada al componente principal se define en la sección 5.2. Finalmente, en la sección 5.5 explora consideraciones relacionadas.

5.1 Datasets de control

Para calcular el tamaño de la muestra es necesario analizar otros conjuntos de datos. El primer conjunto de datos utilizado es una ampliación al utilizado en González *et al.* (2016) con tres años adicionales de datos agregados, lo que nos dio un total de cuatro años de trabajo que forman el conjunto de datos para este documento, nos referiremos al mismo como LSIA. El segundo conjunto de datos utilizado contiene una gran cantidad de información y esta disponible públicamente en la web. Nos referiremos a este como Prosody.

No se puede dejar de mencionar un tercer gran conjunto de datos con 136 millones de keystrokes, al igual que Prosody también es público, y fue recopilado por Dhakal *et al.* (2018). Como era de esperar, el dataset se llama *136M Keystrokes*. Este contiene 168,000 usuarios con alrededor de 800 keystrokes por persona. Si bien el mismo tiene un tamaño inmenso por la cantidad de participantes, 800 keystrokes es un número muy bajo para realizar clasificaciones y crear subconjuntos de contraste.

5.1.1 Dataset LSIA

El conjunto de datos contiene aproximadamente 239.000 muestras, de las cuales 68.808 tuvieron más de 300 keystrokes y fueron útiles para este trabajo. Comprende un promedio de 820 keystrokes cada una (con un rango de 301 a 6383) con 234 eventos comunes (con un rango de 151 a 736). El conjunto de datos completo contenía 56.440.126 keystrokes⁵, y aún continúa creciendo.

La cadencia de tecleo proviene de una base de datos real, rigurosa y no idealizada, generada por un sistema que se ejecuta en un entorno de producción en una institución de salud. Los participantes no son conscientes de las capturas y todos sus metadatos se enmascaran debido a problemas de privacidad. Los usuarios operan diferentes teclados disponibles en toda la organización para escribir textos en este sistema de información que está basado en la web.

La edad promedio de los 409 usuarios para cada muestra es de 33 años, con un máximo de 65 y un mínimo de 23. Las mujeres representaron un 37,5% del conjunto de datos.

De este dataset solo se tomó una parte, filtrando las sesiones chicas o permitiendo hasta 10 sesiones por usuario. Como preprocesamiento separamos dos subconjuntos independientes, el primario con las primeras 10 sesiones por usuario y el de contraste con las siguientes 10. Debido a la naturaleza del conjunto de datos, no todos los usuarios contaron con 20 sesiones, por lo que el segundo subconjunto resultó de menor tamaño. El tamaño de ambos subconjuntos supera con creces a los mencionados en la literatura (Monaco, 2018) y estos

⁵ Número tomado en diciembre de 2018. La base de datos fue utilizada por primera vez en nuestra publicación de González *et al.* (2016).

contienen 3071 y 2500 muestras respectivamente. Luego pareamos cada sesión (no por usuario sino por sesiones; es decir, todas contra todas) mediante un producto cartesiano generando una matriz de combinaciones posibles.

n	Usuarios	Pares	n	Usuarios	Pares
		±			±
12	280	75.3±22	264	269	188±7
14	279	86.6±22	316	266	188±7
17	279	94.9±22	379	262	189±6
21	279	108±20	455	253	189±6
25	279	131±20	546	241	189±6
30	279	143±18	655	230	189±6
36	278	156±15	785	216	189±6
43	278	165±13	942	192	189±6
51	276	178±10	1129	162	189±4
62	275	184±8	1354	132	189±5
74	275	187±8	1624	98	188±6
74	275	187±8	1624	98	188±6
89	273	187±8	1948	41	189±4
106	273	188±7	2337	22	189±3
127	273	188±7	2803	14	188±6
153	272	188±7	3361	5	190±0
183	272	188±7			

Cuadro 5.1: Este cuadro representa el número de usuarios disponibles que tienen al menos diez muestras con al menos n keystrokes. Los pares son todas las combinaciones para ese n particular con al menos dos eventos comunes. A medida que aumenta n , la dispersión del número de pares se desvanece así como el número de usuarios que poseen al menos diez muestras de n keystrokes.

Excluyendo pares sin eventos comunes, para el subconjunto primario, el total de pares resultó de 4.660.123, mientras que para el contraste fue de 3.067.375. Para evitar cualquier variación temporal, recogimos valores consecutivos desde el principio del conjunto de datos.

Una diferencia sustancial en el EER producido por los conjuntos de datos primarios y de contraste será un indicador confiable de que las métricas no funcionan correctamente. Otra forma de averiguar si las métricas no funcionan correctamente sería una inversión en cualquier tendencia entre ambos conjuntos de datos.

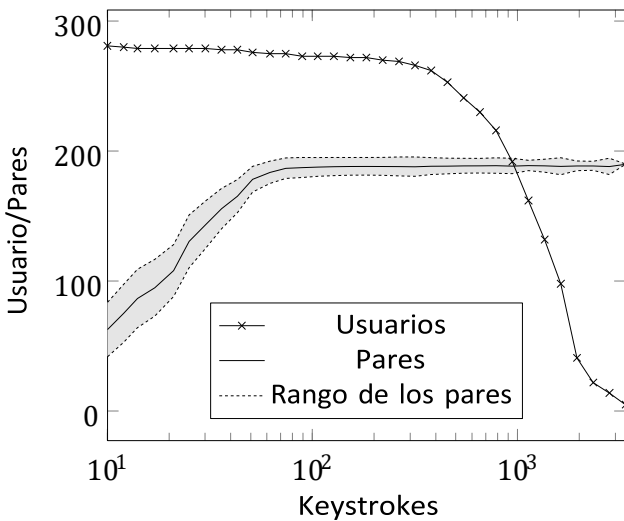


Figura 5.1: Número de usuarios disponibles en el conjunto de datos LSIA y pares de muestras resultantes para evaluar, ambos por cantidad de keystrokes. El rango de pares es el intervalo que resulta de un desvío estándar.

Debido a que no todos los usuarios proporcionaron suficiente información para ser estudiados, el cuadro 5.1 muestra el número de usuarios con al menos diez muestras de n keystrokes. Dado que el proceso de análisis exigirá una comparación entre un usuario y otros, la columna *pares* representa el total de pares posibles entre cada una de las diez muestras de usuario y diez muestras aleatorias adicionales que comparten al menos dos pulsaciones de tecla. La figura 5.1 ilustra el cuadro con una escala logarítmica, mostrando cómo caen las bandas de pares y la cantidad de los usuarios.

Dada la extensión del conjunto de datos, los tiempos de computación fueron altamente significativos. Para acelerar el

procesamiento, ha sido esencial construir una plataforma que emplee un lenguaje compilado para optimizar el tiempo de CPU y ejecutar operaciones paralelas. Este software redujo el tiempo de una semana estimada, en una implementación usando lenguaje interpretado, a 5 minutos mientras se ejecuta en una instancia de 72 núcleos. El marco resultante aún tenía los problemas de memoria presentes en el software inicial. Cargar toda la base de datos en la memoria para realizar todas las comparaciones no resultó viable. En consecuencia, la solución fue elegir el número 10 como las muestras máximas por usuario.

5.1.2 Prosody database

Banerjee *et al.* (2014) reunieron tres cuerpos a través de Amazon Mechanical Turk⁶. Según ellos, se condujo a los participantes, o *turkers*, a un sitio web diferente que captura la cadencia de tecleo. El sitio web inhabilitó las operaciones de copiar y pegar. Se pidió a los participantes que escribieran dos textos, con más de 100 palabras, sobre uno de los tres temas, a saber, reseña de restaurantes (Review), matrimonio gay (Gay) y control de armas (Gun). Un texto estaba a favor del tema, y el otro estaba en contra. Después de completar los textos, los participantes tuvieron que volver a escribirlo. Además, los autores revisaron manualmente los textos, filtrando los que no cumplían con sus requisitos.

Para el cuerpo de reseña de restaurantes, recolectaron datos de 500 participantes. Además, recopilaron datos para 400 participantes para cada uno de los otros dos cuerpos. Cada persona contribuyó con cuatro textos derivados de la combinación de los textos a favor y en contra, y el original y la copia. Aunque Banerjee *et al.* (2014) capturaron eventos del mouse y metadatos adicionales, esta información será pasada por alto ya que no es necesaria para el propósito de esta tesis.

⁶ Amazon Mechanical Turk (o MTurk) es un marketplace de crowdsourcing que subcontrata una fuerza de trabajo virtual distribuida para realizar tareas determinadas, la herramienta está disponible en línea en <https://www.mturk.com/> (verificada en enero de 2019).

De acuerdo con el relevamiento de conjuntos de datos presentado por Monaco (2018), esta es una de las bases de datos de *keystroke dynamics* de texto libre más extensas, con 4,6M pulsaciones.

5.2 Verificación de cambio de estado afectivo

Con el objetivo de validar que los usuarios fueran trasladados a distintos estados afectivos, se realizó una prueba U de Mann–Witney. La elección del test se debe a la continuidad de las señales de EEG y, por consiguiente, de los valores de excitación–valencia; estos no se distribuyen normalmente, por lo que una prueba de tipo t-test de Student no pareada no es factible. La prueba U de Mann–Witney verifica la igualdad de las medias y no pide normalidad en las muestras (Hodges y Lehmann, 1963; Wilcoxon, 1945). El cuadro 5.2 muestra las correlaciones entre fases necesarias para que la estimulación haya logrado correr al usuario a un estado afectivo diferente. La hipótesis nula plantea que no hay variación significativa en las medias de las muestras obtenidas.

El valor A significa “Hipótesis Aceptada”, NA es “Hipótesis No Aceptada”, mientras que los signos de interrogación (?) representan valores donde no es importante el resultado. Esto último se debe a que la excitación (Arousal) de las fases 1 (HAPV) y 4 (HANV) y de las fases 2 (LAPV) y 3 (LANV) son, en la teoría, iguales entre sí. Análogamente, la valencia (Valence) entre las fases 1 (HAPV) y 2 (LAPV) y las fases 3 (LANV) y 4 (HANV) son iguales también en la teoría. La hipótesis de igualdad no es necesaria para esta tesis.

En todos los casos, las muestras variaron la media de al menos una dimensión (excitación valencia). En ciertos casos, como por ejemplo, el participante 6 que mantuvo la valencia entre las fases 1 (HAPV) y 2 (LAPV); la media no varió en una dimensión, pero en todos esos casos se mantuvo la dimensión correcta (en el caso del participante 6, es de esperar que las fases 1 (HAPV) y 2 (LAPV) compartan varianzas). El cuadro 5.3 muestra el resultado para el participante 6 y puede ser comparada con el cuadro 5.2.

Existen varias fórmulas utilizadas para transformar valores de electrodos en puntos del circunplejo Arousal–Valence, para esta

validación decidimos utilizar las fórmulas propuestas por Ramirez y Vamvakousis (2012, ver pág. 180) y por Yurci (2014, ver cuadro 6, pág. 25), las cuales se detalla en 5.1. El framework producido también soporta varias fórmulas, las cuales son intercambiables. En los trabajos citados se encuentran las justificaciones teóricas y citas a otras posibles fórmulas. Los electrodos que se utilizan suelen ser AF_3 , AF_4 , F_3 y F_4 (ver Figura 5.2 para sus ubicaciones) por encontrarse sobre el lóbulo frontal.

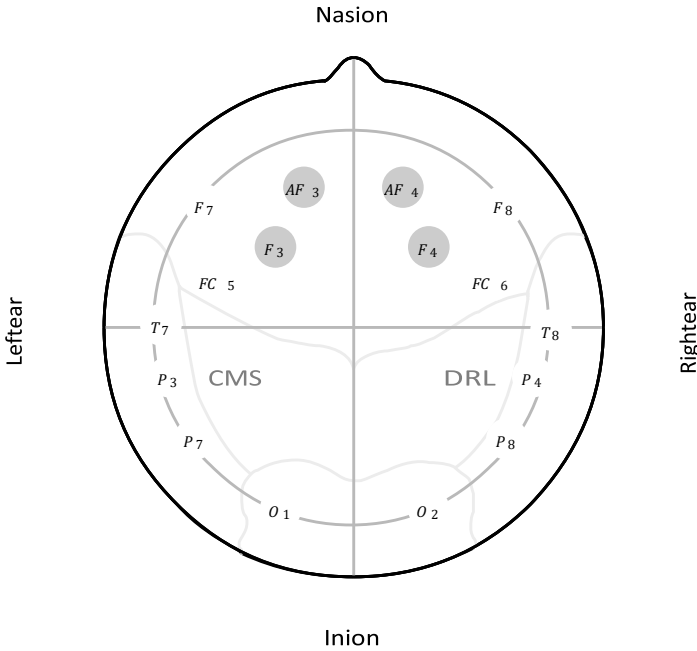


Figura 5.2: Ubicación de los electrodos sensados. Las emociones se encuentran en los grisados: AF_3 , AF_4 , F_3 y F_4 .

$$= \frac{\beta AF_4 + \beta AF_3 + \beta F_4 + \beta F_3}{\alpha AF_4 + \alpha AF_3 + \alpha F_4 + \alpha F_3}$$

Arousal

(5.1)

$$= \frac{\beta AF_4 + \beta F_4}{\alpha AF_4 + \alpha F_4} - \frac{\beta AF_3 + \beta F_3}{\alpha AF_3 + \alpha F_3}$$

Valence

Una vez obtenidos los valores, los mismos son ecualizados, para ello se asume la fase neutra como la base tomando un valor medio y el desvío estándar de la misma. Luego se procede a ecualizar las lecturas de Arousal–Valence aplicando la transformación

$$v_e(t) = \frac{v_i(t) - \mu}{\sigma} \tag{5.2}$$

donde $v_e(t)$ es el valor ecualizado y $v_i(t)$ es el valor original.

En el caso de SAM, la información ingresa directamente al ser el usuario quien responde los valores de excitación y valencia de manera directa al observar los dibujos. La figura 5.3 muestra los valores de Arousal–Valence para cada fase, superponiendo los valores subjetivos obtenidos de la señal EEG y transformados con el método de Ramirez y Vamvakousis (2012) con los valores subjetivos obtenidos por SAM. Los participantes nunca repitieron las mismas elecciones para ambas encuestas (imágenes/audio o video) entre dos fases consecutivas. Sí hubo un caso donde luego de IAPS se repitió el punto en el circuplejo, aunque al finalizar la estimulación por video esto se modificó antes de ser evaluada la cadencia.

H_0 necesaria	Fase	Arousal				Valence			
		1	2	3	4	1	2	3	4
	1	A	NA	NA	?	A	?	NA	NA
	2	NA	A	?	NA	?	A	NA	NA
	3	NA	?	A	NA	NA	NA	A	?
	4	?	NA	NA	A	NA	NA	?	A

Cuadro 5.2: Hipótesis necesarias para todos los participantes.

Estadístico	Fase	Arousal				Valence			
		1	2	3	4	1	2	3	4
	1	15138	8016	0	0	15138	13651	13986	1534
	2	8016	12482	0	601	13651	12482	11914	605
	3	0	0	14964	12520	13986	11914	14964	900
	4	0	601	12520	16562	1534	605	900	16562

<i>p</i> -value	Fase	1	2	3	4	1	2	3	4
	1	1.000	0.000	0.000	0.000	1.000	0.914	0.255	0.000
2	0.000	1.000	0.000	0.000	0.914	1.000	0.044	0.000	
3	0.000	0.000	1.000	0.001	0.255	0.044	1.000	0.000	
4	0.000	0.000	0.001	1.000	0.000	0.000	0.000	1.000	

<i>H</i> ₀ aceptada	Fase	Arousal				Valence			
		1	2	3	4	1	2	3	4
1	A	NA	NA	NA	A	A	NA	NA	
2	NA	A	NA	NA	A	A	NA	NA	
3	NA	NA	A	NA	NA	NA	A	NA	
4	NA	NA	NA	A	NA	NA	NA	A	

Cuadro 5.3: Resultados de cambios de fase para el participante número 6.

Fase	SAM Img.		SAM Video		EEG				
	A	V	A	V	A	V	σ_A^2	σ_V^2	σ_{AV}
1	7	5	5	7	5.65	4.84	0.04	0.78	-0.08
Δ	0	-2	-1	-2	-0.22	0.00	$\ \Delta\ = 0,22$		
2	5	5	4	5	5.43	4.84	0.11	0.26	0.07
Δ	0	-2	2	-3	-1.37	0.08	$\ \Delta\ = 1,37$		
3	4	3	6	2	4.06	4.92	0.07	0.40	-0.14
Δ	0	1	-1	0	-0.01	2.24	$\ \Delta\ = 2,24$		
4	6	3	5	2	4.05	7.16	0.23	1.17	-0.26

Cuadro 5.4: Diferencias numéricas entre los valores Arousal–Valence adquiridos mediante SAM luego de imágenes, SAM luego de videos (o audio) y EEG para el participante número 6. En el caso de EEG es un valor medio y se muestra la varianza en cada eje y su covarianza. Los valores intermedios marcados como Δ muestran las diferencias; es importante señalar que la diferencia en SAM luego imágenes está calculada contra SAM luego del video de la fase previa.

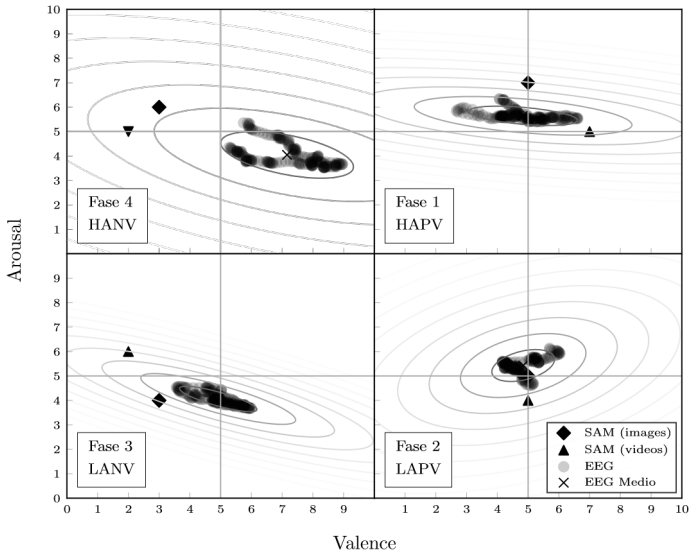
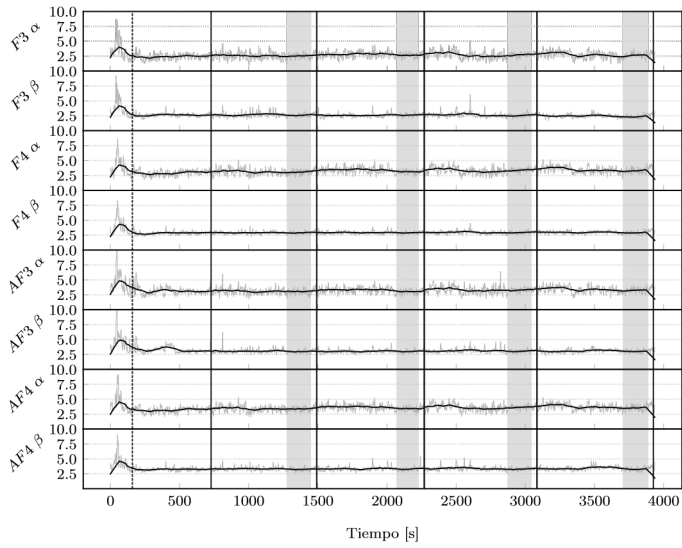
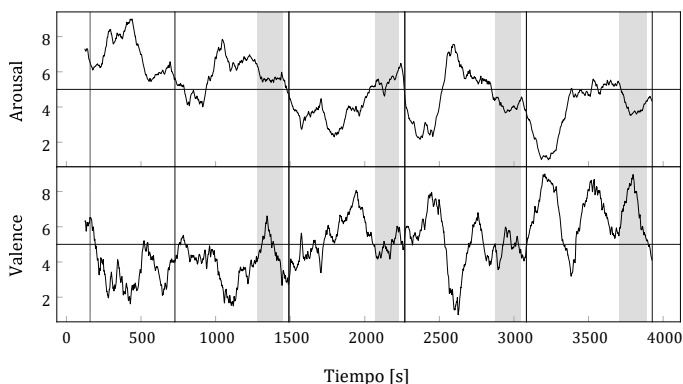


Figura 5.3: Movimientos de Arousal y Valence del participante número 6 durante la captura de teclado.



(a) Registros de EEG del participante número 6.



- (b) Arousal y Valence del participante número 6 según algoritmo de Ramirez. La franja grisada representa la captura de teclado.

Figura 5.4: Registros obtenidos del participante número 6.

5.3 Elección de métricas y parámetros

Definir qué métricas y sus respectivos parámetros serán analizados es una decisión significativa para esta tesis. Para ello será necesario analizar ambos minuciosamente; clasificarlos y definir métricas o combinaciones de las mismas que resulten eficaces para la tarea que pretendemos realizar. Esta sección replicará las métricas utilizando los sets de datos previamente definidos en la sección 5.1 y analizará qué parámetros resultan más útiles.

5.3.1 Métricas comunes

Inicialmente, medimos las distancias Manhattan, euclídea, Camberra, Minkowski (para varios valores de p), Chebyshev y las métricas de Gunetti y Picardi (2005), A y R, además de una combinación de estas dos. Las correlaciones y varianzas estadísticas que conectan los elementos se excluyeron de esta investigación; por lo tanto, métricas como Mahalanobis y euclídea normalizadas no fueron incluidas. Aun así, se nota que los vectores de ponderación por la cantidad de información (número de keystrokes por tipo de evento) fueron más prometedores que ponderarlos por la varianza o la desviación estándar. Las métricas más exitosas fueron, principalmente, las métricas de

Manhattan, euclídea y A y R. Las presentamos en la figura 5.5 con muestras para diferentes tamaños. Chebyshev resultó ser pobre para analizar los ritmos de escritura. El cuadro 5.5 describe los valores detallados por cada métrica e indican cada EER en porcentaje por número de muestras y métricas para ambas muestras: la primaria y la

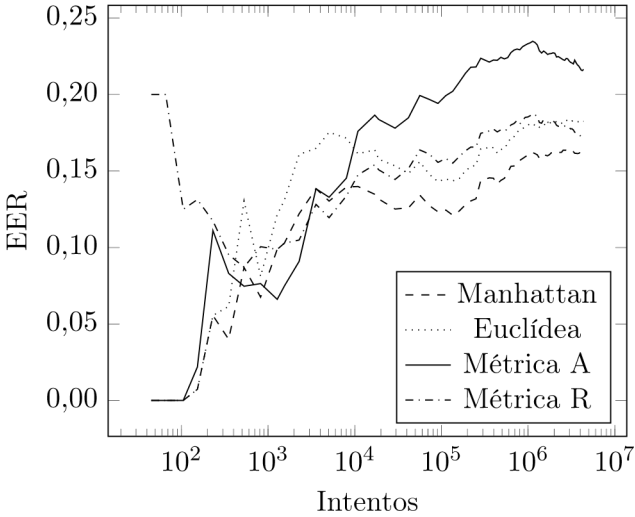


Figura 5.5: Comparación de métricas Manhattan, euclídea y métricas A y R para un número creciente de muestras.

de contraste. Las métricas analizadas son la métrica A (A), una combinación de métricas A y R (A+R), la métrica R (R), Camberra (C), Camberra ponderada por el número de eventos (C/n) y tres distancias Minkowski: Minkowski con el parámetro 0.4 (L^{0.4}), Manhattan (L¹) y euclídea (L²).

5.3.2 Réplica de la métrica A

Gunetti y Picardi (2005) experimentaron con la métrica A utilizando valores de t de 1,20, 1,25, 1,33 y 1,40. Propusieron el valor de 1,25 que inferieron empíricamente y produjo buenos resultados. En la réplica realizada para esta tesis, figura 5.6 muestra los efectos para diferentes valores de t , reforzando la propuesta original de Gunetti y

Picardi (2005). El rango óptimo observado por esta réplica comprende 1,25–1,32.

Pairs	105/105	19k/19k	487k/491k	3M/3M	4.6M
Samples	15/15	200/200	1000/1000	2500/2500	3070
A	0/2	18/15	22/21	22.1/21.5	21.9
A+R	1/0	15/11	19/17	18.5/17.6	18.1
R	13/2	15/11	18/16	17.8/16.6	17.4
C	2/0	24/20	45/37	34.5/32.1	38.5
C/n	1/2	23/19	29/32	26.8/27.3	28.6
L _{0,4}	0/4	13/11	15/14	15.9/15.4	15.6
L ¹	0/4	13/12	14/14	16.2/15.6	16.1
L ²	0/9	16/15	16/16	18.2/18.0	18.2

Cuadro 5.5: Comparación entre métricas. Los valores de EER son porcentuales. La barra inclinada separa los subconjuntos primarios y de contraste (independiente). C significa Camberra, C/n es la distancia de Camberra ponderada por el número de eventos.

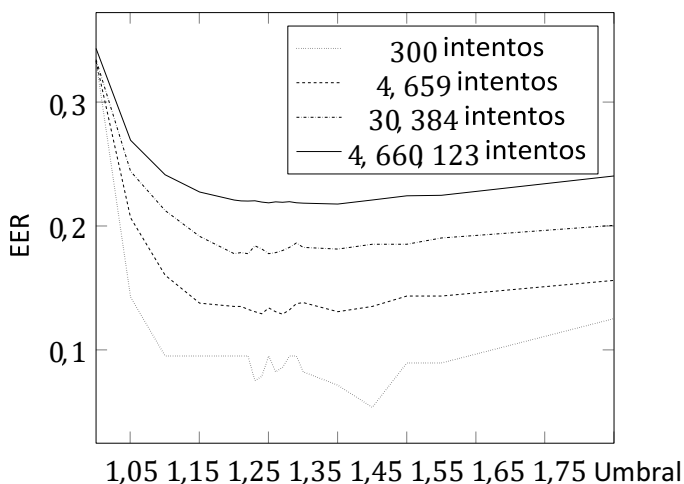


Figura 5.6: Valores de umbral para la métrica A vs. intentos de identificación. Un intento es una comparación de dos muestras diferentes para determinar si pertenecen a la misma persona. El mismo incrementa cuadráticamente con la cantidad de personas.

5.3.3 Réplica de las distancias de Minkowski

En textos libres, un problema habitual es el tamaño cambiante que adquieren los vectores en la intersección entre los patrones de entrenamiento y evaluación. Estos difieren dependiendo de cada caso, lo que hace que una métrica no pueda compararse contra casos diferentes. Por ejemplo, si se cuenta con 20 eventos entre la persona A y la persona B, y con 10 eventos entre la persona B y la C, sería injusto comparar las distancias entre ambos sin realizar una normalización previa. La solución principal para esto es dividir los cálculos resultantes por el número de eventos compartidos. Por lo tanto, la métrica se volverá independiente de la extensión de los vectores comparados. De esa manera, la métrica de Manhattan se representará como el promedio del valor absoluto de las diferencias y no como la norma de la diferencia. Dejando aparte el caso de Manhattan, otras métricas de Minkowski plantean la cuestión de si dividir antes o después de la última operación de potencia. Por lo tanto, es posible definir tres valores diferentes para cada métrica de Minkowski, a saber, la norma como en (5.3), el promedio general como en (5.4) y el promedio de los términos como en (5.5). En todos los casos, $\vec{x}, \vec{y} \subset \mathbb{R}^n$ y n es el tamaño del vector.

$$\mathcal{L}^p(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\|_p = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (5.3)$$

$$\check{\mathcal{L}}^p(\vec{x}, \vec{y}) = \left\| \frac{\vec{x} - \vec{y}}{n} \right\|_p = \left(\sum_{i=1}^n \left| \frac{x_i - y_i}{n} \right|^p \right)^{\frac{1}{p}} = \frac{\mathcal{L}^p(\vec{x}, \vec{y})}{n} \quad (5.4)$$

$$\hat{\mathcal{L}}^p(\vec{x}, \vec{y}) = \frac{\|\vec{x} - \vec{y}\|_p}{\sqrt[p]{n}} = \left(\sum_{i=1}^n \frac{|x_i - y_i|^p}{n} \right)^{\frac{1}{p}} = \frac{\mathcal{L}^p(\vec{x}, \vec{y})}{\sqrt[p]{n}} \quad (5.5)$$

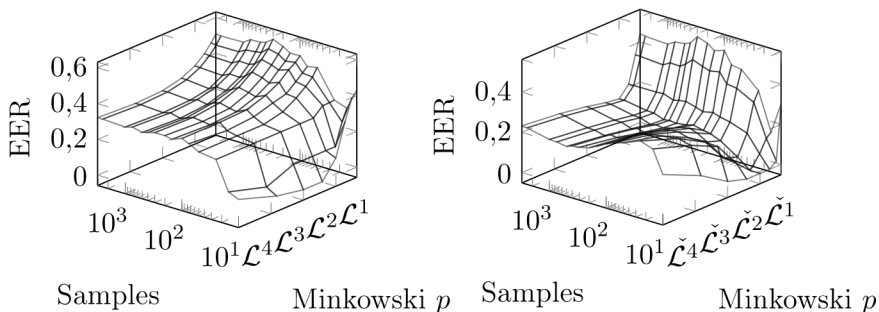
El uso de normas produce métricas que no se pueden comparar como se muestra en figura 5.7a, mientras que las normas promediadas $\check{\mathcal{L}}^p$ (general) y $\hat{\mathcal{L}}^p$ (por términos) se encuentran en figura 5.7b y 5.7c

respectivamente. Mis experimentos encontraron que los promedios de los términos dan mejores resultados, obteniendo un valle alrededor de $p = 0,4$ Ver Cuadro 5.6 para referencia. Hay que tener en cuenta que para los valores $0 < p < 1$, Minkowski no es una norma por definición, por no ser subaditiva, pero aún así arroja los mejores resultados.

Aumentar el número de muestras produce mayores tasas de error, ya que crece el número de usuarios; de ahí es de esperar que disminuya la posibilidad de detectar al usuario correcto. Por esta razón, cada cuadro y figura proporcionadas producirán resultados inferiores para mayores cantidades de sesiones.

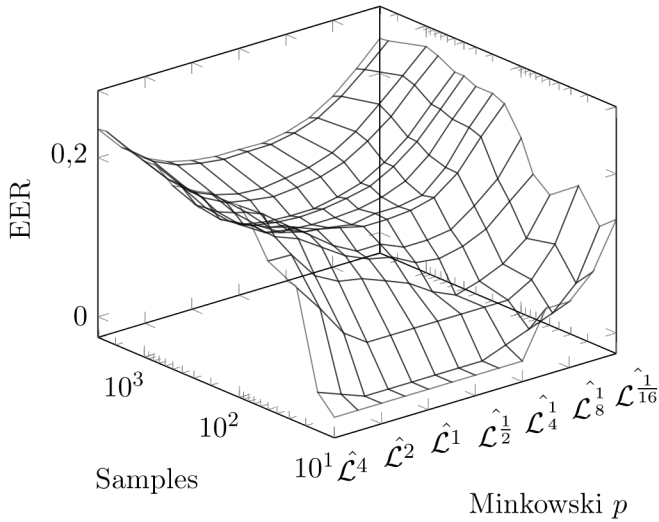
5.3.4 Análisis de indicadores biometricos

Comprender el EER y el umbral correspondiente de una métrica particular tiende a ser beneficioso porque, si este umbral se usa para clasificar biometría, entonces se puede emplear el EER como una medida del error. En ese sentido, es razonable afirmar



(a) Comparación utilizando normas (b) Promedio general de las diferencias sin promediar o normalizar para distancias de Minkowski

sultan todas en valores superiores a 0.2.



(c) Promedio de términos desde $\hat{\mathcal{L}}^{\frac{1}{16}}$ a $\hat{\mathcal{L}}^4$.
El valle se sitúa alrededor de $p = 0,4$. Este gráfico utiliza la escala logarítmica en el eje de p .

Figura 5.7: Pruebas con la norma de Minkowski.

1, 2 y ∞ corresponden a Manhattan, euclídea y Chebyshev respectivamente.

que, en circunstancias comparables y dimensiones de conjuntos de datos similares, si un atacante P está intentando acceder a un sistema como usuario Q , solo analizando la dinámica de tecleo, las posibilidades de éxito al usar Minkowski con $p = 0,4$ son 15,5% (evaluando si $\hat{L}^{0,4}(P,Q) < 29,044$ es cierto). Por definición de EER, las posibilidades de que el usuario real Q sea rechazado también son 15,5%.

Para finalizar los experimentos, realizamos una revisión exhaustiva de la métrica A con $t = 1,25$ ($A^{1,25}$) y Minkowski con $p = 0,4$ ($L^{0,4}$). Para probar nuestros resultados, evaluamos 40.000 muestras del 17% inicial del conjunto de datos y descartamos las muestras con menos de 500 keystrokes. El resultado fue de 1.191 muestras. A continuación, les aplicamos un producto cartesiano entre sí y, después de eliminar los duplicados, obtuvimos 694.301 pares de muestras. Luego calculamos los indicadores biométricos FAR, FRR y EER. Las figuras 2.2a y 2.2b de la sección 2.2.4 exhiben FAR y FRR para diferentes valores de umbral.

La suavidad de las curvas es causada por el alto número de muestras analizadas. Los resultados para $\hat{L}^{0,4}$ fueron de un EER de 15,5% y para $A^{1,25}$ un EER de 22,8% que, considerando el enorme volumen de comparaciones, terminan siendo métricas excepcionales.

5.3.5 Filtrado de latencia

El siguiente experimento no se ha destacado en la literatura y tiene un impacto considerable en la selección de la métrica. Ha habido varios enfoques acerca de cuándo cortar la cadena analizada debido a una alta latencia, como una pausa o una parada. En un momento crítico, la latencia no tiene nada que ver con el rasgo neurofisiológico sino con otro proceso. Las implementaciones disponibles generalmente esperan alrededor de un segundo para separar el texto en diferentes instancias de escritura. Sin embargo, es necesario realizar una prueba adicional: ¿Cuál es la latencia que minimiza el EER?

Para llevar a cabo esta prueba, es necesario tener un gran conjunto de datos. La idea de filtrar las pulsaciones con valores bajos de latencia —descartando FT o WT mayores a un valor pero sin partir la cadena— puede, en un punto particular, comenzar a reducir el ruido originado por otros rasgos que pueden no haber sido incluidos en el patrón original. En modelizaciones con contextos finitos este tipo de filtros no es recomendable para valores muy bajos porque se perdería información contextual o sería necesario definir si la cadena debe ser partida en dos o es la misma. En la figura 5.8 mostramos los resultados de dicha prueba utilizando tres métricas, a saber, métricas A y R, y Minkowski.

Los resultados son particularmente inesperados pero tienen sentido completo. Minkowski logra el mejor EER a 250 ms, sin embargo, al pasar ese punto, el EER comienza a aumentar. Este resultado se condice con el hallado por Concilio (2019) (ver página 79 de dicha tesis). Las métricas R y A se ocupan de este problema y realizan una disminución estable durante todo el proceso. Las propiedades inherentes de las métricas explican este comportamiento. Si bien Minkowski es una métrica topológica normalizada, un incremento de los valores elevará los errores relativos. Mientras tanto, la métrica R usa rangos para calcular la distancia y, en consecuencia, es seguro procesar números altos. La métrica A cuenta los valores atípicos en las

relaciones entre los valores. Por lo tanto, como los ratios también son independientes de la unidad, una métrica no recibirá ningún impacto negativo procesando número altos y, de hecho, mejora con ellos.

Este experimento concluye que Minkowski es particularmente útil para valores menores a 250 ms, mientras que la métrica R es para valores más largos. Además, podemos señalar que Minkowski puede lograr mejores resultados con menos información, ya que se pueden filtrar muchos valores.

5.3.6 Discusión

La presente tesis utilizará principalmente las métricas A (con $t = 1,25$) y R, la distancia de Minkowski con $p = 0,4$ y, para proporcionar puntos de comparación con otros trabajos, las distancias Manhattan y euclídea donde resulten útiles.

5.4 Elección de la cantidad de keystrokes (tamaño de la muestra)

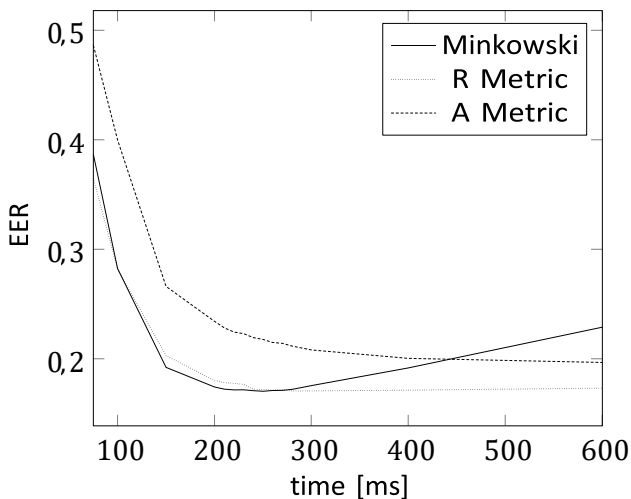


Figura 5.8: Tres métricas que analizan diferentes umbrales de filtrado de tiempos.

Para seleccionar la cantidad de keystrokes a obtener en nuestro experimento, será necesaria la realización de un experimento previo sobre otros conjuntos de datos públicos y de gran cantidad de keystrokes.

El conjunto básico de propiedades que debe poseer cualquier experimento general para ser valioso incluye al menos *validez externa*, *replicabilidad* y *generalizabilidad* (Schofield, 2002). Más específicamente, en el campo de los experimentos informáticos sobre biometría y dinámica de tecleo, se ha enfatizado la necesidad de experimentos comparativos e inferencias estadísticas para abordar los tres anteriores (Killourhy y Maxion, 2011); las evaluaciones ingenuas en las que se usa un conjunto de datos novedosos para evaluar un método o hipótesis igualmente novedosos no pueden servir solamente como fundamentos sólidos para la disciplina. Siguiendo las directrices de Jain *et al.* (2015) para las mejores prácticas en la investigación biométrica, los requisitos se abordan en el presente estudio tal como se describen.

Validez externa como lo exige la directriz 9 de Jain *et al.* (2015), nuestro conjunto de datos es amplio, desafiante y corresponde a los datos obtenidos en la operación en el mundo real de un sistema de información ampliamente implementado. Los usuarios cuyas secuencias de escritura fueron capturadas realizan sus tareas diarias en el sistema sin darse cuenta de que están grabados.

Generalizabilidad Como lo requieren las pautas 10 y 11 de Jain *et al.* (2015), los datos de prueba se adquirieron en varias sesiones que duraron más de dos años y los conjuntos de capacitación, pruebas y validación no se superponen.

Replicabilidad Como lo requiere la directriz 12 de Jain *et al.* (2015), los detalles pertinentes sobre el protocolo experimental y las características del conjunto de datos se incluyen en este trabajo o en el material adicional en línea cuando superan el alcance del documento, como el código fuente y resultados para cada usuario.

Un error común en los experimentos de dinámica de tecleo ha sido la tendencia a considerar diferentes métodos de EER, y también otras métricas de rendimiento, como cantidades puntuales que tienen un valor definido en lugar de variables aleatorias donde las observaciones

siguen una cierta distribución (Killourhy, 2012). Cuando decimos que un método de clasificación específico basado en características específicas puede alcanzar un EER de 10%, ¿qué queremos decir? De hecho, no es que el EER sea del 10% en cualquier momento, para todos y cada uno de los usuarios, bajo cualquier condición experimental imaginable. A lo sumo, queremos decir que esperamos seriamente que dicha métrica orbite alrededor del valor medio que hemos alcanzado cuidadosamente, los investigadores, promediando un conjunto realista de condiciones experimentales y bases de usuarios que varían. El número subyacente se trata como una variable aleatoria, ya sea que se informe explícitamente si su distribución, sus parámetros, o al menos sus intervalos de confianza. Por obvia que parezca esta observación, solo 6 de los 80 estudios considerados en Killourhy y Maxion (2011) extraen una inferencia estadística de los resultados obtenidos. Para evitar este riesgo metodológico en el presente estudio, se informan las curvas de convergencia, incluidos los valores extremos, los intervalos de uno y dos sigmas, y se hace explícita la cantidad de muestras promediadas para cada punto, de modo que se pueden crear intervalos de confianza.

Los intentos pasados de replicación de resultados en los experimentos de dinámica de tecleo a veces han dado lugar a sorpresas desagradables (González *et al.*, 2016; Killourhy y Maxion, 2010), con tasas de error que saltan de insignificantes según lo informado por los proponentes originales de un método a significativo o inaceptable cuando se replican. Intentando evitar una invalidación temprana de la hipótesis principal de este documento y asegurar la generalización, el conjunto de datos Prosody de Banerjee *et al.* (2014), que fue puesto a disposición del público por sus curadores, se evaluó para respaldar la conclusión y los valores obtenidos con los nuestros, que fueron confirmados. Es una dura realidad que los conjuntos de datos públicos de alta calidad de las sesiones de mecanografía son escasos (Giot *et al.*, 2015). A nuestro mejor propósito, no se pudo encontrar ningún otro que incluyera suficientes sesiones con una cantidad de keystrokes necesaria para la realización de este experimento. Lamentablemente, la base de 136 millones de keystrokes contiene una cantidad muy chica de keystrokes por persona.

5.4.1 Hipótesis del modelo de convergencia

La hipótesis central de este trabajo se puede exponer de la siguiente manera:

Para la mayoría de los métodos basados en la distancia de verificación de keystroke dynamics de texto libre, el EER asintótico más bajo se alcanza, en promedio, de manera exponencialmente decreciente desde el EER máximo en función del número de pulsaciones de la sesión.

La razón detrás de la hipótesis es la simple observación de que para la mayoría de las distancias, cada observación de tiempo ingresa simétricamente en la fórmula y, como tal, tiene una probabilidad independiente y muy pequeña de cambiar el valor total por encima o por debajo del umbral de verificación. Un clasificador aleatorio imparcial tendrá un EER de 50%, donde la mitad de los impostores se clasifican como legítimos y la mitad de los usuarios legítimos como impostores, con un error de clasificación del 50% en ambos casos. Se espera que cada pulsación sucesiva presionada reduzca ligeramente el EER en promedio. Teniendo en cuenta una cantidad limitada de datos, los errores de clasificación pueden atribuirse a tres fuentes: una tasa de error mínima asintóticamente alcanzable para el usuario y el método, E_∞ ; una tasa de error debido a la falta de información suficiente que depende del número de pulsaciones, $E(n)$; y una (a falta de una mejor hipótesis) normalmente distribuida residual, ε . Así

$$EER(n) = E_\infty + E(n) + \varepsilon \quad (5.6)$$

Ahora, si el último término tiene un valor máximo de E_M y cada dato de pulsación tiene una probabilidad independiente de p de contribuir a una clasificación correcta,

$$E(n) = (E_M - E_\infty)(1 - p)^n \quad (5.7)$$

y reemplazando con λ tal que

$$1 - p = e^{-\lambda_1} \quad (5.8)$$

tenemos

$$EER(n) = E_{\infty} + (E_M - E_{\infty})e^{-n\lambda} + \varepsilon \quad (5.9)$$

A continuación estableceremos la validez de la primera para la mayoría de los usuarios y las distancias sobre dos conjuntos de datos radicalmente diferentes de manera empírica y recurrente a la inferencia estadística, así como la normalidad de los residuos ε , y procederemos a encontrar valores promedio para las constantes E_{∞} , E_M y λ que cubren una amplia gama de casos. Un vistazo a la figura 5.9 debería convencer al lector de que nuestro modelo se ve prometedor (Calot *et al.*, 2019b).

5.4.2 Validación del modelo de convergencia

Los datos sin procesar que constituyen la mayor parte de los resultados que se analizarán se obtuvieron mediante la iteración del

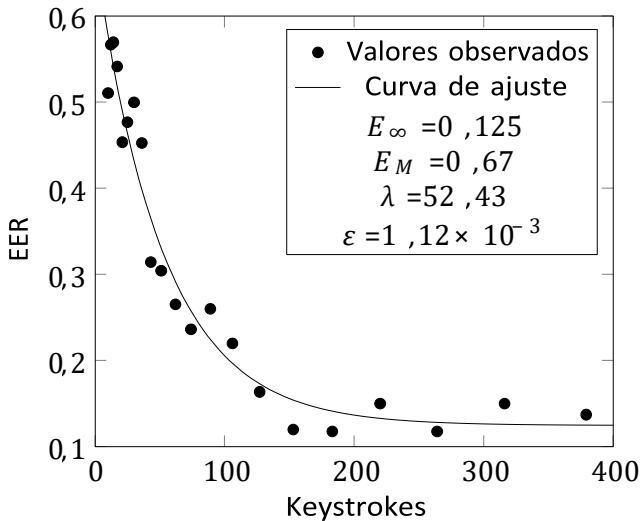


Figura 5.9: Ejemplo de EER obtenidos de la métrica A entre nuestro usuario 193 contra sí mismo y diez muestras aleatorias auténticas negativas.

Dataset	Usuarios rechazados	Usuarios aceptados	Hipótesis rechazada	Hipótesis aceptada
LSIA	2929	171	2683	246
Gay	1333	839	1016	317
Gun	918	1326	807	111
Review	1344	943	1139	205

Cuadro 5.7: Resultados del test de hipótesis.

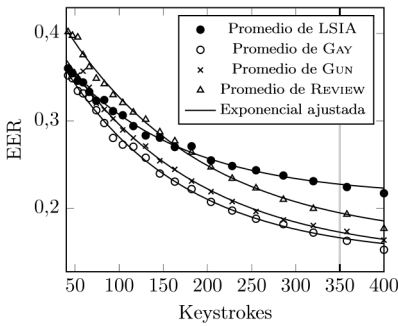
siguiente procedimiento para cada conjunto de datos, métrica de distancia y usuario. Después de filtrar para asegurar sesiones de al menos n keystrokes, se calculó un EER por usuario y métrica comparando hasta 10 muestras por clase —siendo una clase, usuario vs no usuario—. Este límite superior de 10 se definió para tener la clasificación equilibrada ya que el número de muestras útiles se deteriora rápidamente con valores más altos de n . Usando una secuencia de comandos en el lenguaje R, los puntos de los resultados de EER, tabulados contra el rango de conteos de keystrokes del conjunto correspondiente de sesiones, se enviaron a un estimador de cuadrados mínimos ponderados no lineales para aproximar los parámetros E_∞ , E_M y λ de (5.9). Fueron elegidos como parámetros para el procedimiento de estimación, un tamaño pequeño de paso de 10^{-5} y valores iniciales de 0,5 para E_M , 0,25 para E_∞ y 50 para λ , todos consistentes con observaciones empíricas.

Solo se consideraron las sesiones hasta 400 keystrokes debido a que, las más largas son demasiado escasas para que la mayoría de los usuarios proporcionen datos útiles al modelo en lugar de ruido aleatorio. Los usuarios con valores EER considerados anormales fueron rechazados para su posterior consideración. El criterio de aceptación fue que un modelo lineal de valores EER debe tener una pendiente descendente con un incremento en el conteo de teclas para las sesiones; el tipo de artefacto inverso aparece cuando el número de sesiones muy cortas o muy largas no es suficiente para proporcionar un promedio significativo, ya que, por ejemplo, un conjunto de solo dos sesiones puede hacer que el EER salte de un 0% demasiado optimista a un extremadamente alto y pesimista 100%. Por lo tanto, se impuso un valor positivo de λ . Un problema similar puede surgir cuando el número de sesiones cortas es pequeño, pero afecta a E_M o las

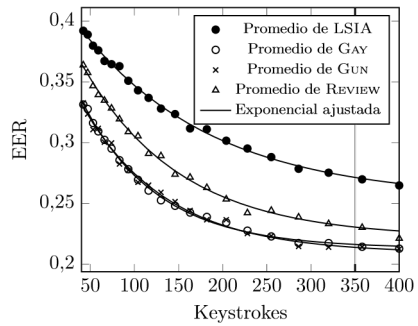
magnitudes relativas de $E_M > E_\infty$. Para ellos, se requirió que $E_M > E_\infty$ y el máximo aceptable de E_M se limitara a 3. De esta manera, como los parámetros de función exponencial son muy sensibles a los valores ajustados más a la izquierda, filtrando a los usuarios con los promedios representativos de EER para sesiones pequeñas cercanas al 100% evitan la introducción de sesgos hacia arriba en los parámetros que no son detectados por la regla de pendiente descendente.

La calidad del ajuste se evaluó utilizando el error de la media al cuadrado y los residuos ε se verificaron con respecto a la normalidad utilizando la prueba de Shapiro–Wilk Shapiro y Wilk (1965). La normalidad fue rechazada por $p < 0,1$.

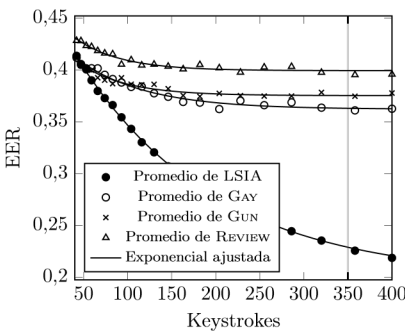
El cuadro 5.7 resume el número de usuarios en cada conjunto de datos aceptado y rechazado *a priori* para los promedios globales y la consideración de la hipótesis central. El gran número de usuarios rechazados en los cuerpos de Prosody (Gay, Gun y Review) en comparación con nuestro propio conjunto de datos LSIA se explica por los criterios de exclusión detallados en la sección anterior, y el motivo se puede resumir como no tener suficientes sesiones muy cortas y muy largas para evitar la introducción de ruido en los parámetros estimados. En el mismo cuadro, las dos columnas de



(a) Usando la Métrica R.



(b) Usando la Métrica A.



(c) Usando la Distancia Manhattan. (d) Usando la Distancia de Minkowski con $p=0,4$.

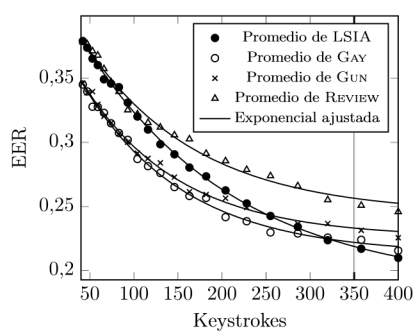
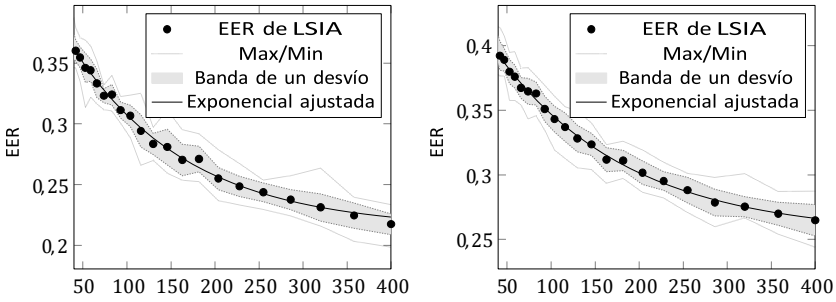
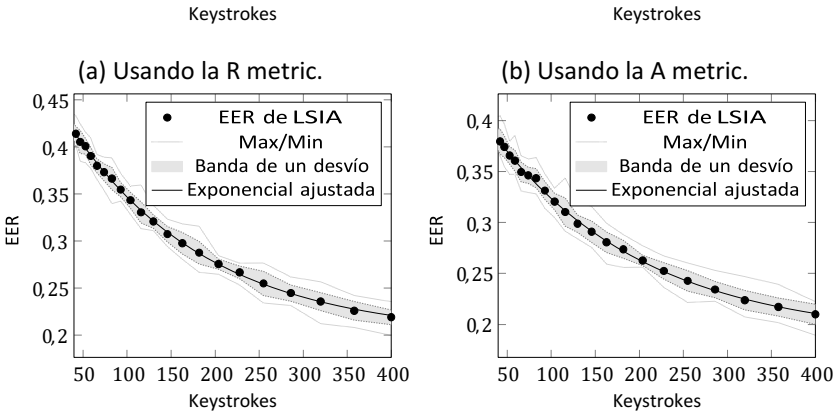


Figura 5.10: Curvas exponenciales producidas por el EER para pulsaciones de teclado que varían en el rango 40–400 para diferentes conjuntos de datos y métricas.

la derecha separan a los usuarios donde se aceptó la hipótesis de normalidad de los residuos, lo que demuestra que (5.9) encaja bien, de aquellos en los que se rechazó. A pesar de que la tasa de rechazo de hipótesis de nuestro conjunto de datos, alrededor del 10%, es consistente con $\alpha = 0,1$, las tres partes de Prosody muestran tasas de rechazo más altas de lo esperado. El cuerpo Gun, el de mejor comportamiento, está cerca del valor esperado con 13,75% pero Gay llega a más del 30%. Por lo tanto, nuestro modelo exponencial es adecuado para la mayoría de los usuarios de ambos conjuntos de datos, aunque no podemos descartar que una minoría considerable, de hasta el 20%, pueda estar experimentando un comportamiento de



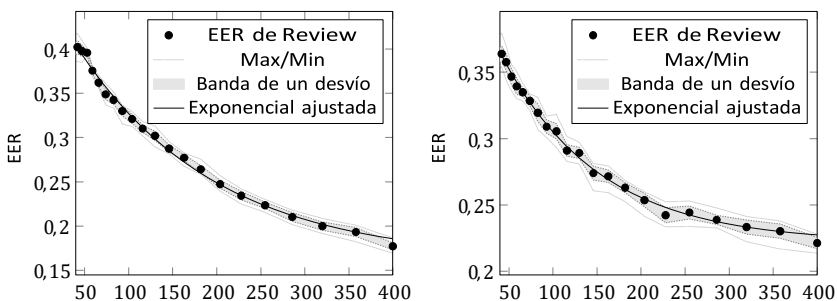
convergencia diferente. Una explicación más benévola, que se insinúa en las desviaciones estándar mucho más grandes en el cuadro 5.8 para los conjuntos de datos de control en varios parámetros,



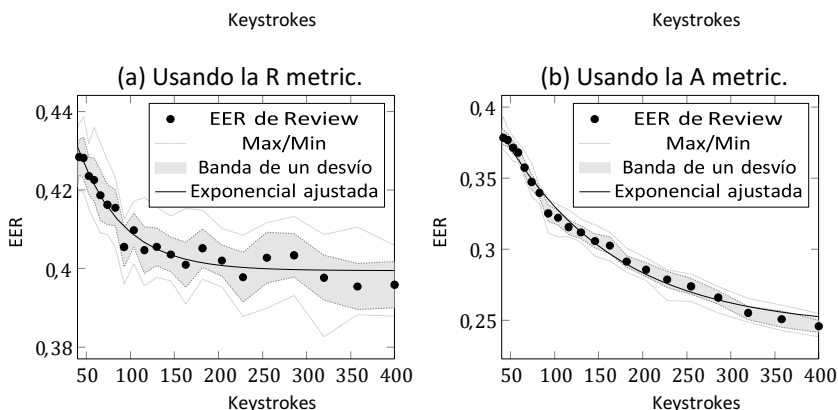
(c) Usando la Manhattan distance. (d) Usando la Minkowski distance with $p = 0,4$.

Figura 5.11: Curvas exponenciales producidas por el EER para una cantidad de keystrokes en el rango 40–400 en el conjunto de datos de LSIA. La banda gris muestra una desviación estándar.

es que el pequeño número de sesiones por usuario en el rango muy corto y muy largo introduce ruido que paraliza el procedimiento de ajuste. Establecer una hipótesis u otra para usuarios individuales, y posiblemente encontrar una función más adecuada para los casos que fallan, requerirá de una investigación más exhaustiva. Sin embargo, como puede verse en la figura 5.10, los promedios sobre todos los usuarios de cada conjunto de datos para las métricas R, A, Manhattan y Minkowski (se omite la euclídea debido a la falta de espacio, aunque



se la muestra en el cuadro 5.8), sigan de cerca el modelo propuesto. Por lo tanto, el resultado central de este documento, que confirma la hipótesis central, se puede expresar como



(c) Usando la Manhattan distance. (d) Usando la Minkowski distance with $p = 0,4$.

Figura 5.12: Curvas exponenciales producidas por el EER para una cantidad de keystrokes en el rango 40–400 en el cuerpo Review del conjunto de datos Prosody. La banda gris muestra una desviación estándar.

Para todos los métodos considerados basados en la distancia de la verificación de keystroke dynamics de texto libre (A, R, Manhattan, Minkowski y euclídea), tanto en nuestro conjunto de datos LSIA como en el conjunto de datos externo Prosody, se alcanza el EER asintótico más bajo, en promedio, de manera exponencialmente decreciente desde el EER máximo en función del número de pulsaciones de la sesión. La mayoría de los usuarios muestran el mismo comportamiento

individualmente, aunque el modelo propuesto podría no describir adecuadamente el comportamiento de convergencia de una minoría no despreciable de hasta el 20%.

Una gráfica detallada del comportamiento de convergencia de los EER, que incluye una banda de un desvío estándar, así como el valor máximo y mínimo, para cada distancia específica, promediada sobre cada usuario de LSIA dataset se muestra en figura 5.11. La figura 5.12 hace lo mismo con el cuerpo Review del conjunto de datos Prosody. Excepto la distancia de Manhattan, que muestra fluctuaciones más fuertes que las otras, el comportamiento de convergencia empírica es muy similar en ambos conjuntos de datos y sigue de cerca el modelo teórico. Los valores de los parámetros para cada exponencial de ajuste se dan en el cuadro 5.8 junto con el error cuadrático medio, que se mantiene por debajo del 3,5% en el subconjunto Review de Prosody y por debajo del 1,5% en LSIA implicando que la calidad del ajuste por usuario para los perfiles válidos en los que se aceptó la hipótesis de normalidad de los residuos es excelente.

		E_M [%]		E_∞ [%]		λ [ms]		ε [‰]	
		Value	max/min	Value	max/min	Value	max/min	Value	max/min
LSIA	\mathcal{L}^1	54.6±13.0	28.4/150	27.1±10.3	3.1/46	105.2±101.7	9/784	1.31±1.03	0.24/8.52
	\mathcal{L}^2	56.0±10.7	27.1/123	30.0±11.0	6.0/51	117.3±94.0	13/625	1.43±1.05	0.22/8.48
	$\mathcal{L}^{0,4}$	52.6±15.9	19.9/177	26.0±10.7	0.3/49	106.0±144.4	6/1212	1.25±1.15	0.15/9.14
	A	52.0±10.0	30.0/104	24.4±11.6	0.6/48	124.2±110.8	10/756	1.23±1.18	0.25/11.33
	R	67.5±25.3	34.7/230	33.0±6.2	2.6/49	85.2±182.2	6/2189	1.11±0.92	0.24/5.81
GAY	\mathcal{L}^1	66.3±23.9	25.9/219	33.7±10.8	5.1/51	97.2±104.6	7/499	2.85±1.81	0.61/8.85
	\mathcal{L}^2	70.4±27.6	33.2/270	37.5±10.1	6.7/57	96.6±138.0	5/845	3.33±1.88	0.65/10.48
	$\mathcal{L}^{0,4}$	62.4±19.8	17.4/146	31.9±14.0	0.3/52	125.5±354.0	6/3820	2.89±1.61	0.36/7.74
	A	56.4±29.7	31.0/294	29.1±12.7	0.3/48	110.6±118.1	8/646	3.38±2.32	1.04/13.74
	R	70.5±39.2	42.8/252	43.4±5.5	22/52	93.6±171.0	7/739	2.99±1.75	0.70/7.77
GUN	\mathcal{L}^1	73.9±39.0	40.3/207	45.5±10.4	21/61	103.1±157.1	6/594	2.14±1.03	0.65/3.99
	\mathcal{L}^2	78.2±29.5	52.2/158	50.6±5.6	37/61	68.5±84.2	8/288	2.27±1.22	0.54/5.08
	$\mathcal{L}^{0,4}$	58.7±18.0	32.2/137	36.3±11.3	5.8/51	187.8±231.3	6/1205	2.08±0.81	0.79/4.44
	A	65.4±17.0	39.6/123	37.3±9.8	0.7/52	68.0±89.8	7/533	2.03±0.78	0.84/5.23
	R	63.7±28.3	35.2/264	35.9±10.8	0.8/50	112.1±160.6	3/791	1.77±0.88	0.56/5.33
REVIEW	\mathcal{L}^1	66.2±30.3	43.5/264	37.2±10.4	0.4/56	252.3±265.3	5/1868	2.83±1.64	0.84/8.67
	\mathcal{L}^2	74.5±32.3	40.1/233	42.0±9.2	3.6/58	193.4±305.9	4/1852	2.81±1.57	0.64/11.97
	$\mathcal{L}^{0,4}$	54.8±7.8	42.4/79	33.1±11.5	9.5/51	285.6±275.7	24/1095	3.47±2.08	1.28/10.89
	A	68.4±39.7	40.6/230	31.9±11.0	4.1/50	144.8±200.1	5/1360	3.41±1.94	0.79/9.19
	R	59.7±26.7	37.1/231	27.6±10.2	0.5/50	175.3±183.1	5/1044	3.36±1.98	0.50/10.26

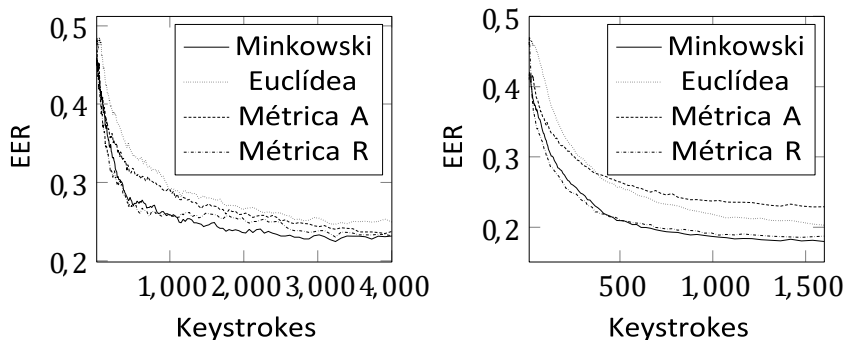
Cuadro 5.8: Valores ajustados por cuerpo o conjunto de datos y distancia.

Los subconjuntos Gay y Gun mantienen un comportamiento de convergencia idéntico solo con un error cuadrático medio más pequeño. El valor de λ , la constante de tiempo, agrupa alrededor de 100 keystrokes aunque existen varias excepciones que lo exceden en un factor de casi tres. Se eligió empíricamente un valor inicial de cuarenta keystrokes para los gráficos, porque en valores más bajos los EER se comportan casi al azar.

Para comprender mejor el efecto del número variable de sesiones, se incluye figura 5.13, donde en lugar de usar todas las sesiones disponibles para cada intervalo de conteo de teclas, se eligió un n fijo. El comportamiento promedio puede ser similar a los casos anteriores, aunque, como se esperaba, un poco más peculiar con un tamaño de muestra más pequeño.

5.4.3 Discusión

A diferencia de la mayoría de los estudios, adquirir el estado emocional requiere que el mecanógrafo no sea consciente del estudio o, de lo contrario, se olvide de que está siendo analizado. En el caso de EEG, la primera opción no es posible, por lo que es necesario invertir tiempo y considerar ejercicios de distracción para aclimatarse y acostumbrar al usuario a lo que está haciendo para que pueda olvidarse de los electrodos en su cabeza. Por otro lado, como pretendemos estimular emocionalmente al usuario, es necesario mostrar imágenes y videos para transportar al mismo al estado deseado. Un gran problema radica en el hecho de que si un participante está leyendo la pantalla, su estilo de escritura puede verse afectado; pocos usuarios tienen la capacidad de escribir mientras miran la pantalla. Por esta razón, es necesaria una estimulación previa para llevar al usuario al estado deseado y luego realizar la adquisición de patrones de pulsación de tecla. También esperamos obtener la cadencia de tecleo del usuario impidiendo que el mismo repita la misma frase literalmente, es por eso que diferentes oraciones desordenadas formando un texto sin sentido pueden ser óptimas para esta tarea. Como nuestro experimento se centra en la tipificación de textos libres, la adquisición de datos con esta metodología se ve favorecida en este contexto porque se puede evitar el tipeo repetitiva, como sucede cuando se escriben las contraseñas.



(a) EER en función de keystrokes tamaño fijo de 75 muestras. Es posible ilustrar sesiones de hasta 4000 keystrokes.

(b) EER en función de keystrokes con un tamaño fijo de 2450 muestras. Aquí solamente es posible graficar sesiones de hasta 1600 keystrokes.

Figura 5.13: Comparación de la suavidad de la curva en función del número de muestras. Como los tamaños de las sesiones son variables, el aumento de los keystrokes necesarios disminuye con el número de sesiones disponibles. Usar menos sesiones da como resultado malas estimaciones de los parámetros estadísticos.

Además, el estado de estimulación actual varía con el tiempo y es necesario medir el estado promedio del usuario. Si se le pide al usuario que ingrese textos con más de 700 keystrokes, es posible que comience a perder el estado emocional alcanzado durante la estimulación. El número 700 es un valor empírico obtenido como el valor más bajo posible para el cual podemos adquirir dos subconjuntos viables (de 350 keystrokes) para realizar el análisis correcto. Además, no podemos dejar de considerar la larga duración del experimento cuando tenemos que pasar por cada cuadrante del estado emocional.

La figura 5.10 representa los valores promedio y la curva exponencial de ajuste que denota las convergencias de los conjuntos de datos por número de pulsaciones para cuatro métricas: Métrica R (Gunetti y Picardi, 2005, Métrica A (Gunetti y Picardi, 2005), Manhattan (L^1) y Minkowski con $p = 0,4$ ($L^{0,4}$) (Calot *et al.*, 2019b). El valor 350 está resaltado en todos los casos; donde una diferencia no significativa con la asíntota es observable.

En Enrique P. Calot En González *et al.* (2016) se realiza una réplica del método de (Gunetti y Picardi, 2005), utilizando un conjunto de datos

adquirido en condiciones más severas. Sus sesiones, 14 en total, se limitaron a 700–900 keystrokes. Una duración de sesión superior a 700 keystrokes favorece los métodos basados en sus métricas definidas A y R, ya que las muestras más cortas aumentan las tasas de error rápidamente antes de este tamaño, como puede verse en el Cuadro XIII de Gunetti y Picardi (2005) y en el cuadro VI de González *et al.* (2016). En el artículo, informamos que se alcanza un rendimiento de autenticación casi óptimo con 150 keystrokes con un modelado de contexto finito. Teniendo esto en cuenta, hemos elegido 10 oraciones de aproximadamente 50 caracteres cada una, sumando hasta 533 caracteres. Esta decisión nos permitió lograr que durante la captura del experimento, los usuarios generaran un promedio de 739 keystrokes (que oscilan entre 666 y 823 con una desviación estándar de 37.7) y conjuntos de eventos de dimensión promedio 282 (que oscilan entre 220 y 365 con un desvío estándar de 46,3) gracias a la alta repetición de los keystrokes. Aunque las oraciones se introducirán de manera no clasificada, los keystrokes serán las mismas en todos los participantes y fases.

Solicitar más keystrokes pone en peligro todo el experimento a medida que el usuario comienza a aburrirse y ponerse ansioso por llegar a la siguiente fase. En conclusión, la utilización de sesiones de 350 keystrokes es factible para realizar el experimento principal.

5.5 Configuración del experimento y consideraciones

Un diseño efectivo de un experimento es tan importante para obtener resultados exitosos como la ejecución del mismo. Recopilar estados emocionales es una tarea difícil que implica un problema central inherente al dominio: la información obtenida puede resultar subjetiva (porque se adquirió a través de una encuesta) y, si es objetiva (adquirida a través de EEG u otros dispositivos biométricos) los datos tuvieron que ser modelados en base a descripciones iniciales que, de hecho, fueron subjetivas en algún momento. En otras palabras, en algún momento, alguien tuvo que describir subjetivamente los estados emocionales para crear una escala cuantitativa.

Además, la descripción del estado emocional varía de un lenguaje a otro, lo que hace que los términos afectivos no sean traducibles con absoluta precisión.

Por otro lado, la literatura aún no ha acordado una metodología universal para correlacionar estados emocionales con adquisiciones objetivas; lo que es menos, todavía no existe un modelo universal para las emociones debido a las discrepancias consagradas en las diversas corrientes de pensamiento. Para superar esta complicación, es necesario abstraernos de la metodología y parte del modelado de los estados emocionales sin perder de vista la necesidad de garantizar que una persona haya alterado su estado emocional.

Este trabajo tiene como objetivo investigar si esos estados afectivos alteran o no la identificación mediante la dinámica de pulsaciones de teclas. En este orden, se pretende, con métodos subjetivos (Self-Assessment Manikin, o SAM [Lang, 1980]) y objetivo (EEG utilizando interfaces cerebro-máquina) para estimular a los participantes con el uso de un conjunto de imágenes IAPS (Lang *et al.*, 2008) con alto contenido emocional. De esta manera, atravesarán diferentes estados que corresponden a los cuatro cuadrantes del modelo *circumplejo* (Arousal-Valence); con el objetivo de poder obtener los patrones de tecleo de un individuo durante cada uno de los cuadrantes que representan estados emocionales.

Se optó por utilizar BMI como una estrategia que permite una evaluación objetiva del estado de ánimo de manera inconsciente que se encuentra aplicada en diversos trabajos que requieren determinar el estado de excitación-valencia del modelo propuesto por Russell detallado en el estado del arte. En particular, se seleccionó el headset Emotiv EPOC por la cantidad de sensores y la disponibilidad para obtener el equipo. Esto permite aplicar el modelo de Ramírez en la determinación de variación de excitación-valencia. Se intentó utilizar otros headsets con BMI (Neuroskype, Emotiv insight) pero por la escasa cantidad de electrodos y su ubicación la aplicabilidad del algoritmo se vio comprometida.

Se decidió contrastar con SAM porque esta técnica fue validada por Lang y Bradley contra valores del circumplejo de Russell obteniendo una base de datos de imágenes muy grande y ampliamente utilizada por la literatura (Lang *et al.*, 2008); en el experimento se utilizaron estas imágenes para generar estímulos, por lo que SAM resultaba ser la opción

más simple, considerando que es subjetiva y, a su vez, contrastable con los experimentos de Lang (mismos estímulos, mismo modelo emocional, y mismo sistema de adquisición; pero incorporando EEG y cadencia de tecleo).

Resultados experimentales

Para analizar los datos, fueron necesarias varias pruebas de hipótesis, en todas ellas el intervalo de confianza definido fue del 5%. Los cuadros 6.1 y 6.2 muestran los resultados obtenidos para seis métricas diferentes, donde el p -value muestra la probabilidad de que se rechace la hipótesis. Todos los valores de EER se expresan en porcentaje (%), a menos que se especifique lo contrario.

El proceso de análisis comenzó con una prueba de normalidad, para la cual se eligió la prueba Shapiro–Wilk (Shapiro y Wilk, 1965) (SWT). Esto se aplicó a cada muestra individualmente, tanto las que incluían la misma emoción como las que la excluían. Las muestras para las métricas A, R y la distancia de Camberra no rechazaron la hipótesis de normalidad, por lo que fue posible realizar una prueba t pareada (Goulden, 1956) (conocida en inglés como *paired t-Test* o PtT) —ya que este tipo de prueba exige las muestras para ser distribuidos normalmente—, comparando datos que incluyen las mismas fases emocionales con datos que no lo hacen. La hipótesis nula de la prueba t pareada fue que los valores medios del EER por usuario son los mismos para ambos conjuntos, mientras que la hipótesis alternativa es que no lo son. Para las tres métricas, se rechazó la hipótesis nula (ver Cuadro 6.1), lo que indica un cambio en el EER medio, es decir, la exactitud de la clasificación se modificó comparando con un conjunto de datos que contienen diferentes emociones. La diferencia se representa en el cuadro 6.1 como $\Delta\mu$ y sus unidades son por mil.

Por otro lado, las distancias de Minkowski (por $p = 0,4$, Manhattan y euclídea) rechazaron la hipótesis de normalidad propuesta por la prueba de Shapiro–Wilk y necesaria para realizar una prueba t pareada. Este rechazo se debe principalmente al usuario número 2, que en forma de valor atípico, presentó valores muy bajos (ver cuadro 6.2). Por esta razón, se decidió realizar una prueba no paramétrica; la prueba de los rangos con signo de Wilcoxon (1945) (conocida en inglés como *Wilcoxon signed rank test* o SRT). Con un intervalo de confianza de $\alpha = 0,05$, la prueba SRT

rechazó las métricas A, R, Camberra, Minkowski $p = 0,4$ y Manhattan. Sin embargo, no rechazó la norma euclídea.

En las primeras cinco métricas, se observa un cambio cerca de $\Delta\mu = 1\%$. Para la norma euclídea, en cambio, se ve un valor de $\Delta\mu = 2h$.

		Métrica A		Métrica R		Camberra	
Emocionescombinadas	Usuario	EER	máx/mín	EER	máx/mín	EER	áx/mín
	1	29.1±5	37/23	22.3±2	23/19	31.0±3	34/27
	2	29.2±4	35/23	30.5±4	35/23	31.2±4	35/25
	3	30.8±0	31/31	23.7±2	27/20	28.5±2	31/27
	4	26.5±3	31/22	18.9±3	22/15	22.8±3	27/19
	5	34.7±2	38/31	25.8±5	35/20	35.8±2	38/34
	6	31.2±3	37/29	27.8±4	35/23	33.8±3	38/31
	7	30.9±0	31/31	24.1±3	27/19	28.8±2	31/27
	8	29.7±4	35/23	20.9±3	26/17	29.5±3	31/23
	9	31.5±1	34/31	25.1±2	27/23	33.8±3	37/29
	10	30.8±0	31/31	26.4±3	31/23	32.6±3	38/29
	11	34.1±1	35/33	32.5±8	38/19	35.7±2	38/35
	12	30.3±4	38/27	21.9±4	29/19	26.5±3	31/23
Total	30.7	38/22	25.0	38/15	30.8	38/19	
SWT	<i>p</i> -value Estad.	4.302E-01		9.722E-01		7.190E-01	
		0.9345		0.9776		0.9556	
	<i>H</i> ₀	No Rechazada		No Rechazada		No Rechazada	

		Métrica A		Métrica R		Camberra	
Emocionescombinadas	Usuario	EER	máx/mín	EER	máx/mín	EER	áx/mín
	1	28.5±5	37/22	18.6±1	20/17	27.3±2	31/25
	2	28.0±3	31/26	30.3±3	34/26	32.0±2	34/29
	3	29.4±1	31/29	22.3±1	23/20	26.3±2	29/23
	4	26.5±4	31/20	16.9±2	20/14	21.2±2	23/19
	5	33.7±1	34/31	23.7±5	31/20	33.0±1	34/31
	6	31.4±4	37/25	26.7±3	31/23	32.8±3	37/29
	7	29.1±1	31/29	23.3±2	26/20	27.7±2	31/26
	8	27.6±3	31/23	17.8±3	23/14	27.4±3	31/23
	9	29.1±1	31/29	22.9±2	26/20	31.6±3	35/26
	10	28.6±0	29/29	26.3±3	31/23	30.5±2	34/29
	11	34.4±2	37/32	32.8±7	40/23	35.4±1	37/34
	12	29.2±5	37/23	18.2±2	23/17	24.8±5	33/20

	Total	29.6	37/20	23.3	40/14	29.2	37/19
SWT	<i>p</i> -value	6.20E-02		4.89E-01		8.32E-01	
	Estad.	0.8682		0.9393		0.9635	
	<i>H</i> ₀	No Rechazada		No Rechazada		No Rechazada	
PtT	<i>p</i> -value	1.658E-03		1.384E-03		5.779E-04	
	Estad.	4.13		4.24		4.77	
	<i>H</i> ₀	Rechazada		Rechazada		Rechazada	
SRT	<i>p</i> -value	8.590E-03		5.355E-03		4.193E-03	
	Estad.	73		75		76	
	<i>H</i> ₀	Rechazada		Rechazada		Rechazada	
	$\Delta\mu$ []	11		17		16	
	h						

Cuadro 6.1: Resultados del experimento para las métricas A, R y Camberra.

Emociones combinadas	Usuario	Minkowski ($L^{0.4}$)		Manhattan (L^1)		Euclídea (L^2)	
		EER	máx/mín	EER	máx/mín	EER	máx/mín
	1	33.4±3	38/29	35.3±2	38/34	39.3±5	46/31
	2	9.1±6	19/4	11.3±6	19/4	13.0±3	19/11
	3	32.4±2	35/31	34.7±2	38/31	36.1±2	38/34
	4	27.7±2	31/27	30.6±3	35/27	32.9±3	39/31
	5	33.5±3	37/27	33.5±4	37/27	39.2±3	43/35
	6	37.0±2	40/35	39.2±3	42/34	41.4±5	50/37
	7	24.6±2	27/23	23.4±4	27/17	32.4±6	42/23
	8	32.3±2	35/31	33.8±4	38/27	39.0±3	42/35
	9	28.5±2	31/27	31.5±4	35/23	38.9±5	46/31
	10	27.3±3	31/23	28.8±2	31/27	31.9±4	37/27
	11	27.7±4	35/23	31.8±4	39/27	36.2±2	38/33
	12	30.0±6	35/23	31.8±3	35/27	36.7±4	42/31
	Total	28.6	40/4	30.5	42/4	34.7	50/11
SWT	<i>p</i> -value	8.938E-03		1.077E-02		8.870E-04	
	Estad.	0.7982		0.8053		0.7023	
	<i>H</i> ₀	Rechazada		Rechazada		Rechazada	

Emociones combinadas	Minkowski ($L^{0.4}$)		Manhattan (L^1)		Euclídea (L^2)	
	Usuario	EER	máx/mín	EER	máx/mín	EER
1	32.6±2	35/29	34.8±3	40/31	38.7±3	43/35
2	8.1±3	15/6	9.7±4	14/3	11.9±3	17/9
3	30.9±2	34/29	34.2±3	40/31	38.2±2	40/34
4	26.3±1	29/26	29.0±3	33/26	32.6±2	35/29
5	30.7±3	33/26	33.9±3	38/29	39.3±2	43/37
6	34.3±2	37/31	38.4±2	40/34	41.7±4	48/37
7	23.6±2	26/21	22.5±3	26/17	31.8±5	40/25
8	30.9±3	37/29	33.5±3	37/29	38.2±3	44/34
9	28.0±2	31/26	30.3±4	34/23	36.1±4	43/29
10	26.4±2	29/23	28.3±1	29/27	33.2±3	38/29
11	27.0±3	31/23	30.9±4	37/26	36.5±2	38/33
12	28.8±4	33/23	32.0±3	37/29	36.6±3	42/31
Total	27.3	37/6	29.8	40/3	34.6	48/9
SWT	<i>p</i> -value	3.14E-03	1.16E-02	6.62E-04		
	Estad.	0.7567	0.8081	0.6890		
	<i>H</i> ₀	Rechazada	Rechazada	Rechazada		
SRT	<i>p</i> -value	2.526E-03	8.590E-03	5.563E-01		
	Estad.	78	73	47		
	<i>H</i> ₀	Rechazada	Rechazada	No Rechazada		
$\Delta\mu$ []	13	7	2			
h						

Cuadro 6.2: Resultados del experimento para las distancias de Minkowski: con $p = 0,4$, Manhattan y euclídea.

6.1 Análisis gráfico de los casos

La presente sección explicará cada uno de los los tres gráficos y dos cuadros necesarios para la interpretación de los resultados de los participantes, siendo el capítulo 5 el que contiene al participante 6 como ejemplo y el apéndice F el que contiene el resto de los participantes. Estos son

- Cuadros con los resultados de la prueba de hipótesis U de Mann–Witney. Muestran cuán parecidas son las mediciones de cada fase. Las matrices son simétricas. Para Arousal y Valence se presenta primero el producto cartesiano de los estadísticos del test de hipótesis, el p -value y finalmente si la hipótesis nula fue aceptada o no lo fue. En el capítulo anterior, el cuadro 5.2 muestra un caso ideal mientras que el cuadro 5.3 muestra un ejemplo con el participante 6.
- Cuadros con los valores de Arousal y Valence obtenidos con SAM y EEG y las diferencias de los mismos para cada cambio de fase. Inicialmente se presenta los valores obtenidos mediante SAM luego de una estimulación por imágenes, SAM luego de una estimulación por videos (o en el último caso un audio) y cinco valores de EEG correspondientes a las medias de Arousal y Valence, sus dos respectivas varianzas y la covarianza entre ambos. Los valores intermedios figuran como Δ y ambos están calculados luego de la estimulación con imágenes de la fase anterior por ser esta la previa a la adquisición de cadencia de tecleo y la que más interesa para este trabajo. En el capítulo anterior, el cuadro 5.4 muestra un ejemplo con el participante 6.
- Graficando Valence en función de Arousal es posible llegar a este gráfico paramétrico en función del tiempo que muestra el movimiento realizado por los valores de Arousal y Valence calculados a partir de las señales de EEG durante el transcurso del tipeo. Las elipses marcan la varianza sobre los componentes principales de la captura graficando una elipse cada cuatro desvíos estándar. Esta figura se parte en cuatro siendo un cuadrante para cada fase. Aquí se superponen los valores obtenidos por las dos encuestas SAM realizadas en cada fase.

En el capítulo anterior, la figura 5.3 muestra un ejemplo con el participante 6.

- Este tipo de figuras se compone de dos, la primera representa los registros crudos obtenidos del participante. Los mismos muestran los valores de las ondas α y β medidas en milivoltios para los electrodos que se encuentran en la corteza prefrontal: F3, F4, AF3, AF4. El área grisada corresponde a la captura de cadencia de tecleo.

En el capítulo anterior, la figura 5.4a muestra un ejemplo con el participante 6.

- La segunda parte de la figura representa las transformaciones a valores de Arousal–Valence utilizando el algoritmo de Ramirez y Vamvakousis (2012). Aquí también el área grisada corresponde a la captura de cadencia de tecleo. En el capítulo anterior, la figura 5.4b muestra un ejemplo con el participante 6.

6.2 Análisis de los casos

A continuación, se presenta una descripción de los resultados para cada participante y un resumen general para sumarizar las mismas.

Participante 1. Al tipear, según EEG, este participante aumentó su valencia y disminuyó la excitación en los cuadrantes 1 (HAPV) y 4 (HANV), mientras que en los 2 (LAPV) y 3 (LANV) se mantuvo constante. SAM presentó variaciones en todos sus cuadrantes, incluso entre las fases 2 (LAPV) y 3 (LANV) presentó cambios de 6 puntos tanto en excitación como en valencia. Los gráficos y tablas comienzan en la página 202.

Participante 2. Al tipear, el EEG de este participante no mostró grandes desvíos. Sus variaciones en SAM fueron chicas pero existieron, probablemente la persona haya decidido ingresar resultados más conservadores a la hora de autoinformar. Los gráficos y tablas comienzan en la página 205.

Participante 3. Este participante fue menos conservador al responder SAM, mostrando grandes diferencias entre las fases. Al tipear, su EEG mostró varianzas muy bajas, nunca superando el valor de 0,11. Luego de la estimulación, el participante varió muy rápidamente su estado acercándose al neutro, por lo que a la hora de tipear las diferencias registradas no fueron tan acentuadas. Los gráficos y tablas comienzan en la página 208.

Participante 4. Los registros de EEG muestran que este participante comenzó a frustrarse a la hora de tipear durante las 4 fases, mantuvo su valencia siempre, excepto en el último tipeo, donde la

valencia creció abruptamente mostrando una respuesta muy positiva a la finalización del experimento. Según SAM, la estimulación logró modificar sus valores en todos los casos. Los gráficos y tablas comienzan en la página 211.

Participante 5. En su EEG, este participante presentó variaciones muy altas mostrándose muy susceptible a la estimulación; durante el tipeo, aumentó su nivel excitación, mientras que la valencia no varió demasiado, aunque sí presentó algunos picos locales positivos. Al igual que los participantes previos, su SAM varió entre fases; en este caso entre 2 y 3 puntos. Los gráficos y tablas comienzan en la página 214.

Participante 6. Este es el participante del ejemplo. Este participante, en EEG, aceptó la hipótesis para igualdad de medias en la valencia entre las fases 1 (HAPV) y 2; este resultado es esperable, ya que ambas fases, según el modelo, mantendrían la valencia de la persona. En SAM, las diferencias entre todas las fases rondan entre 1 y 3 unidades. Los gráficos y tablas comienzan en la página 90.

Participante 7. Este participante autoinformó mediante SAM variaciones emocionales entre 1 y 6 puntos de norma entre sus fases. Su EEG fue más disperso que los otros participantes y su nivel de excitación presentó valles durante la adquisición de cadencia. La estimulación se ve muy marcadamente en las gráficas de señales. Los gráficos y tablas comienzan en la página 217.

Participante 8. Presenta, según valores de EEG, picos de excitación en el medio del tipeo, comenzándose a frustrar hacia el final del mismo. La valencia permaneció constante excepto en la última fase, al igual que el participante número 4. Según

SAM entre fases su estado afectivo varió entre 2 y 4 puntos. Los gráficos y tablas comienzan en la página 220.

Participante 9. Este participante resalta por haber completado 7 puntos en valencia y excitación en las primeras 5 de las 8 encuestas de SAM. Esto representa un problema porque las etapas 1 (HAPV) y 2 (LAPV) presentaron exactamente los mismos valores: 7. Los valores de EEG, en cambio, marcan cambios muy marcados entre fases por lo que se entiende que el participante podría ser extremadamente conservador a la hora de completar el autoinforme.

Contradictoriamente, la norma de la distancia entre los promedios de excitación y valencia ($k\Delta k$) para las fases 1 (HAPV) y 2 (LAPV) marcó un valor de 2,89; el segundo más alto de todo el estudio. Frente a estos resultados podríamos inferir que el usuario no completó correctamente su autoinforme. Durante la escritura tanto su excitación como valencia subieron para las primeras tres fases y bajaron para la última. Los gráficos y tablas comienzan en la página 223.

Participante 10. Este es el participante cuyas señales de EEG mejor mostrar los ciclos de las fases, llegando a hacer todos sus picos de excitación y valles de valencia durante la captura de tecleo. La única excepción es un gran valle de valencia en la etapa de relajación, donde el valor negativo es uno de los más bajos registrados. Según su SAM no se registraron cambios entre sus autoinformes posteriores a la excitación del video y audio de las etapas 3 (LANV) y 4 (HANV), sin embargo el autoinforme intermedio, posterior a la excitación con imágenes, mostró cambios; como la cadencia se tomó luego del video pero antes de las imágenes podemos concluir que según su sam su estado fue modificado. Los gráficos y tablas comienzan en la página 226.

Participante 1. Al igual que el participante 10, este participante presenta los mismos valores de SAM para etapas consecutivas, pero al igual que este, la etapa intermedia varió, registrándose, según el autoinforme, cambios en el estado afectivo a la hora de registrar el tipeo. Sus señales de EEG fueron muy dispares y, también al igual que el participante 10, su excitación presenta picos y su valencia valles muy marcados en a la hora de tipear. Los gráficos y tablas comienzan en la página 229.

Participante 12. Al igual que el participante 9, nuestro último participante aceptó la hipótesis de conservación de la media en la valencia entre las fases 3 (LANV) y 4 (HANV), lo cual es esperado. De manera completamente inversa a los dos participantes previos —10 y 11—, observó valles en la excitación y picos en la valencia a la hora de tipear en todas las fases. Los gráficos y tablas comienzan en la página 232.

En la mayoría de los casos se observan valores cíclicos teniendo un pico al finalizar la estimulación y estabilizándose a la hora de tipear. En algunos casos, este patrón fue distinto y el estado de excitación aumentó al acercarse al final de la etapa de tipeo, probablemente denotando ansiedad por terminar. Lo mismo ocurrió con la valencia, en ciertos casos aumentó y en otros disminuyó, pero en todos los casos los valores de valencia fueron coherentes entre sí para las cuatro fases de cada participante. Esto quiere decir que, si bien las personas reaccionan de manera distinta frente a los estímulos, su reacción es coherente entre sí y sigue el mismo patrón.

Conclusiones

Durante el desarrollo de este trabajo fue creada una plataforma de experimentación que permitió articular la entrada de tres herramientas distintas de captura de información: la cadencia de tecleo es la principal, siendo BCI y la interpretación automática del video capturado las herramientas de contrastación. Gracias a este marco de trabajo fue posible inferir distintas afirmaciones y fueron planteadas más incógnitas que continúan esta rama de investigación.

7.1 Discusión de los resultados

La presente sección discutirá las dos principales hipótesis secundarias, para luego discutir la hipótesis central: la robustez que poseen las métricas de cadencia de tecleo frente a variaciones emocionales.

7.1.1 Modelo de convergencia

Una hipótesis secundaria de esta tesis, sobre el modelo de convergencia, se confirmó empíricamente en nuestro conjunto de datos LSIA, así como también en el conjunto de datos externo Prosody: se puede afirmar que para los métodos considerados basados en distancias —A, R, Manhattan, Minkowski y euclídea—, el EER asintótico más bajo se alcanza, en promedio, de manera exponencialmente decreciente desde el EER máximo en función del número de keystrokes de la sesión. No todos los usuarios, aunque la mayoría lo hacen, muestran este comportamiento de manera individual, y se requiere de estudios adicionales para comprender la convergencia en el nivel de granularidad del usuario. Un valor de λ alrededor de 100 keystrokes es común; aún así, algunas excepciones pueden aumentar este valor hasta tres veces. El ajuste demostró ser preciso,

con un error cuadrático medio inferior al 3,5% en el subconjunto de ajuste más desfavorable.

7.1.2 Robustez de las métricas en general

Esta tesis examina y revisa las métricas de clasificación que prevalecen actualmente. Además, se contribuye a la verificación de los algoritmos basados en distancias: la métrica A y su valor empírico del umbral $t = 1,25$ encontrado por Gunetti y Picardi (2005). Adicionalmente, se propone el uso de la métrica de Minkowski con $p = 0,4$ y se muestra que Minkowski se comporta de manera adecuada frente a un amplio conjunto de datos tomados de un entorno de producción. Tras haber recopilado información en un entorno hostil y no idealizado, esta tesis pone en evidencia la solidez de las métricas. La importancia de esto se basa en el hecho de que ahora podemos emplear este tipo de métricas contra una gama más amplia de conjuntos de datos y datos de producción real, sabiendo que se comportarán como se espera.

En tales circunstancias, nuestra distancia específica de Minkowski puede disminuir el EER medio de 21,9% en la métrica A o 17,4% en la métrica R a una tasa media de 15,6%, lo cual es una mejora sustancial. Esto es importante por su aplicabilidad en el área de la seguridad, exponiendo a posibles intrusos. Además, se hace una diferenciación de los tres tipos de agregación en las métricas de Minkowski especificando cuándo dividir por el número de eventos; en la literatura revisada, la mayoría de los artículos y tesis suelen pasar por alto esta aclaración. Nuestras réplicas, utilizando un conjunto de datos de 56.440.126 pulsaciones de tecla, son altamente compatibles con los parámetros 1,25 para una métrica y 0,4 para Minkowski, ya que otras pruebas con este volumen de datos masivos aún no se han introducido en la literatura.

7.1.3 Robustez de las métricas frente a variaciones emocionales

Este experimento es una prueba preliminar que predice resultados futuros prometedores en conjuntos de datos más grandes. El EER entre una muestra que incluye un estado emocional varió alrededor del 1% en cinco métricas. De esto, se puede concluir que hay indicios de que los estados emocionales influyen en la calidad de la clasificación de los

individuos y que, si un patrón de la persona capturado durante ese estado emocional no está disponible, la precisión del experimento puede empeorar. La métrica euclídea, por otro lado, fue notablemente resistente al cambio emocional y es digna de más investigación, especialmente con muestras más grandes. Es importante enfatizar que la variación obtenida por las otras métricas, aunque estadísticamente visible, no es lo suficientemente significativa a la hora de clasificar como para descartar la cadencia de tipeo como una metodología para identificar personas. Por el contrario, las alteraciones son lo suficientemente pequeñas para que el método siga siendo funcional incluso si no hay un patrón completo disponible en toda la gama de emociones posibles.

7.2 Aportes

Esta tesis obtiene valores que muestran la caída de la efectividad de los algoritmos de distancias de cadencia de tecléo para identificar personas. Además, considerando el cuadro 2.2, se añade como aporte el empleo de EEG y su contrastación con SAM para la validación de la variabilidad del estado emocional.

7.2.1 Frameworks producidos

Se creó un *framework* de postprocesamiento de datos biométricos en Python compatible con hojas de notas de Jupyter. Para cuando se requiera procesar datos de manera intensiva, en el contexto de esta tesis se portó el *framework* al lenguaje ANSI C y se agregó soporte de paralelización.

Para la captura de *keystrokes* fueron diseñados cuatro *keystroke loggers* (*keylogger* que además registra los tiempos de las pulsaciones), los primero tres, desarrollados en ANSI C: uno para Mac OS X, uno para Microsoft Windows y uno para GNU/Linux. El cuarto, desarrollado en JavaScript, es capaz de adquirir cadencia de tecléo en un entorno web.

Los cuatro capturadores son compatibles entre sí produciendo información en un formato estandarizado y creado en el contexto de esta tesis. Este formato resultó de gran aporte por permitir compatibilidad con los *frameworks* de postprocesamiento de los datos. El mismo forma parte de la transferencia tecnológica producida por esta tesis y, actualmente,

sobre estas bases se trabaja en el proyecto de desarrollo estratégico PDE-44-2019 con una importante empresa del mercado y la UBA.

El capturador de *keystrokes* para Microsoft Windows también es compatible con Emotiv EPOC (cuyas librerías y SDK estaban disponibles en su totalidad para dicho sistema operativo) y permite la adquisición sincronizada de datos utilizando la misma máquina.

También se desarrolló toda la mecánica de adquisición de datos multimedial utilizando dichos *frameworks* y el posterior procesamiento de los datos de cadencia de tecleo entre otras modalidades. En colaboración con otros investigadores, aportando únicamente la parte de cadencia de tecleo, se desarrolló el *framework* web para de captura multimedial realizada en la etapa de experimentación de la presente tesis.

Todo el software producido se encuentra disponible para su descarga desde la página web del Laboratorio de Sistemas de Información Avanzados (LSIA) en <http://lsia.fi.uba.ar/> vinculando a nuestro repositorio oficial de Github en <https://github.com/lsia>.

7.2.2 Conjuntos de datos producidos

Un aporte fundamental de esta tesis son los datos crudos capturados por el experimento, el mismo fue realizado en el contexto del LSIA colaborando con otras tesis. Al igual que los *frameworks*, este conjunto de datos se encuentra disponible para su descarga desde el repositorio de Github en <https://github.com/lsia>, sin embargo, algunos archivos EEG crudos y videos cuyo tamaño no hacía práctica su descarga desde la web, pueden ser solicitados al LSIA. En el caso particular de los videos, estos requerirán de la firma de un acuerdo de confidencialidad para proteger la identidad de los participantes.

7.3 Futuras líneas de investigación

Sobre el comportamiento de convergencia de las curvas EER promedio para distancias dadas, se abre un nuevo campo de estudio en relación a la modelización de las mismas. Esta tesis aporta las bases de este campo al ajustar un modelo exponencial negativo, para mostrar que 350 *keystrokes* son suficientes para la realización de la misma; sin embargo, no se ahonda

más en el tema, ya que el modelo utilizado en esta tesis ajustó correctamente para el 80% de los usuarios y, si bien el 20% restante no se ajustaba al mismo, observaba valores convergentes cercanos a los 350 *keystrokes*.

Además, queremos establecer como futuras líneas de investigación la solidez de estas métricas frente a las fluctuaciones temporales (el conjunto de datos contiene cuatro años de pulsaciones de los usuarios).

7.4 Futuras líneas de trabajo

Entre las futuras líneas de trabajo que planeamos abordar, se encuentran la normalización de las métricas a lo largo de la varianza estadística o la cantidad de información disponible, como en los eventos de pulsación de teclas, en consecuencia, la norma de Mahalanobis; las normas clasificadas; y el análisis de las métricas modeladas a través de contextos finitos.

La adquisición de información multimodal resulta de mayor importancia para validar los resultados obtenidos en esta tesis. Actualmente, además de EEG, SAM e información audiovisual, se cuenta con otros datos biométricos, de rostro —que pueden ser procesados con los métodos referidos en la sección D.3.2—, voz y fisiológicos (ritmo cardíaco) en el conjunto de datos adquiridos. Los mismos pueden ser utilizados para la obtención de otros valores de excitación/valencia. Es posible trabajar sobre esta línea validando las correlaciones existentes entre los estados emocionales educidos mediante diferentes modalidades.

Definiciones, métodos y conceptos algebraicos

Ampliar la información disponible sobre algunos conceptos algebraicos resulta sumamente necesario para la correcta justificación de la tesis. Estos conceptos, si bien pueden ser centrales a la tesis, aclararlos en el cuerpo de la misma podría generar confusiones o distraer al lector. Es por ello que el presente apéndice explayará y justificará algunos métodos y conceptos algebraicos y presentará definiciones importantes.

La sección A.1 definirá normas y métricas de modo matemático y su aplicación a *keystroke dynamics*. Las secciones A.2 hasta A.4.1 abordarán métodos alternativos no utilizados en la tesis pero que son necesarios para comprender el estado del arte. Finalmente, los distintos tests de hipótesis utilizados se detallan en la sección A.5.

A.1 Métricas y distancias

Esta tesis trata varios conceptos que pueden resultar confusos al lector no técnico. Dos de ellos que están muy relacionados entre sí son las *normas* y las *métricas* o *distancias*.

A.1.1 Definiciones matemáticas

A.1.1.1 Normas

Se denomina *norma* a una función que asigna a cada vector de un espacio vectorial un valor real escalar. Las normas dependen del tipo de espacio de origen y deben cumplir con ciertos requisitos.

Formalmente, dado un espacio vectorial V sobre un subcuerpo K

de los números complejos, llamamos norma en V a una función $f: V \rightarrow \mathbb{R}$ que cumple las siguientes propiedades $\forall k \in \mathbb{K}$ y $\sim u, \sim v \in V$,

1. *Escalabilidad*

$$f(k\sim v) = |k|f(\sim v) \quad (\text{A.1})$$

2. *Subaditividad*

$$f(\sim u + \sim v) \leq f(\sim u) + f(\sim v) \quad (\text{A.2})$$

3. *Norma cero del vector nulo*

$$f(\sim v) = 0 \leftrightarrow \sim v = \sim 0 \quad (\text{A.3})$$

La propiedad de subaditividad requiere que se cumpla la desigualdad de Minkowski o triangular. Cuando un espacio vectorial está asociado a al menos una norma, este se denomina *espacio normado*. Cuando la norma asociada al espacio vectorial resulta trivial, se utiliza la nomenclatura $\|\sim v\| \stackrel{\text{def}}{=} f(\sim v)$.

A.1.1.2 Distancias

Se denomina *distancia* o *métrica* a una función que permite cuantificar la lejanía entre dos puntos de un espacio vectorial específico. También debe cumplir ciertas características. Formalmente, una distancia d sobre un conjunto V es una función $d: V \times V \rightarrow \mathbb{R}$ que cumple las siguientes propiedades $\forall \sim x, \sim y, \sim z \in V$

1. *Positividad*

$$d(\sim x, \sim y) \geq 0, \sim x = \sim y \leftrightarrow d(\sim x, \sim y) = 0 \quad (\text{A.4})$$

2. *Simetría*

$$d(\sim x, \sim y) = d(\sim y, \sim x) \quad (\text{A.5})$$

3. *Subaditividad*

$$d(\sim x, \sim z) \leq d(\sim x, \sim y) + d(\sim y, \sim z) \quad (\text{A.6})$$

Un espacio asociado a una distancia se denomina espacio métrico. En todo espacio normado V , la norma $\| \cdot \|$ induce una función de distancia

$$d(\vec{x}, \vec{y}) = \| \vec{x} - \vec{y} \| \tag{A.7}$$

A.1.1.3 Distancia de Minkowski

La distancia más intuitiva es la euclídea. Puede ser entendida como la aplicación del teorema de Pitágoras y se define formalmente de la siguiente forma

$$\mathcal{L}^2(\vec{x}, \vec{y}) = \| \vec{x} - \vec{y} \|_2 = \sqrt{\sum_{i=1}^n |x_i - y_i|^2} \tag{A.8}$$

En cambio, la distancia de Manhattan se define como

$$\mathcal{L}^1(\vec{x}, \vec{y}) = \| \vec{x} - \vec{y} \|_1 = \sum_{i=1}^n |x_i - y_i| \tag{A.9}$$

La distancia de Minkowski es una generalización de varias distancias que incluye estas dos junto con la de Chebyshev. La distancia usa un parámetro p , que en el caso de Manhattan es $p = 1$ (denotada L^1), euclídea es $p = 2$ (denotada L^2) y Chebyshev (denotada L^∞) es el límite de $p = \infty$. Su expresión es

$$\mathcal{L}^p(\vec{x}, \vec{y}) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{A.10}$$

para valores de $p \geq 1$ conforma una métrica, pero para valores menores no lo es por no garantizar la subaditividad. Pese a esto, esta tesis utiliza valores menores a 1 ya que los mismos responden mejor que los mayores a 1. La figura A.1 grafica circunferencias unitarias en espacios normados relacionados a distancias de Minkowski para distintos valores de p .

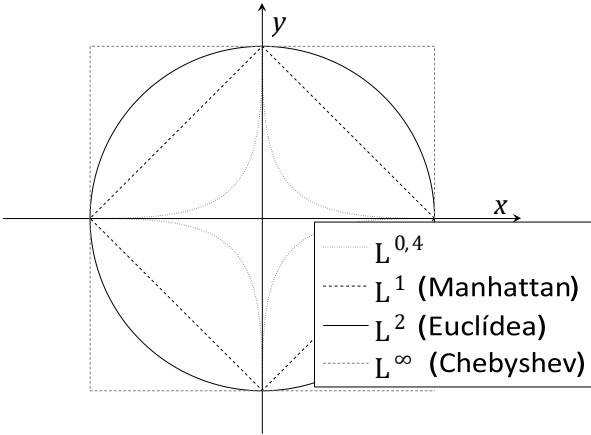


Figura A.1: Circunferencias unitarias según distintas normas.

En el ejemplo de la figura 2.8 de la sección 2.4.3 contamos con los vectores $\sim x = (29,13,15,20,16,15,17)$ e $\sim y = (31,30,13,66,5,32,20,24,5)$. Para el caso particular de $p = 0,4$, la distancia de Minkowski se calcularía Como

$$\begin{aligned}
 \mathcal{L}^{0,4}(\vec{x}, \vec{y}) &= \left(\sum_{i=1}^n |x_i - y_i|^{0,4} \right)^{\frac{1}{0,4}} \\
 &= (|29 - 31|^{0,4} + |13 - 30|^{0,4} + |15 - 13|^{0,4} + |20 - 66,5|^{0,4} \\
 &\quad + |16 - 32|^{0,4} + |15 - 20|^{0,4} + |17 - 24,5|^{0,4})^{\frac{1}{0,4}} \\
 &= 17,56^{2,5} = 1292,83
 \end{aligned}
 \tag{A.11}$$

En este caso, la métrica R transformaría ambos vectores a $\text{rank}(\sim x) = p \sim = (7,1,2,6,4,3,5)$ y $\text{rank}(\sim y) = \sim q = (5,4,1,7,6,2,3)$.

A.1.2 Métricas utilizadas en keystroke dynamics

Además de las variantes de Minkowski, la mayoría de las métricas existentes han sido utilizadas. A continuación se detallan las más exitosas en nuestros experimentos.

A.1.2.1 Camberra

Otra métrica examinada es la distancia de Camberra. Es una versión ponderada de la distancia de Manhattan propuesta por Lance y Williams (1966, 1967) y ya ha sido empleada en detección de intrusos por Emran y Ye (2001, 2002). Se define como

$$C(\vec{x}, \vec{y}) = \sum_{i=1}^n \frac{|x_i - y_i|}{|x_i| + |y_i|} \quad (\text{A.12})$$

A.1.3 Distancias ranqueadas

En estadística, *ranking* se refiere a la transformación de datos en la que los valores numéricos u ordinales se reemplazan por su clasificación cuando se ordenan los mismos. Reemplazar los valores del vector con rangos secuenciales puede definir una alternativa adicional de una métrica que no usa sus valores en sí sino el orden entre ellos. Este tipo de métricas las definimos como *métricas ranqueadas*.

A.1.3.1 Métrica R

La métrica R de Gunetti y Picardi (2005) no es nada menos que la forma rankeada de la distancia Manhattan, más conocida como *Spearman's footrule* y mide el desorden del vector.

En el ejemplo de la figura 2.8 de la sección 2.4.3 contamos con los vectores $\sim x = (29, 13, 15, 20, 16, 15, 17)$ e $\sim y = (31, 30, 13, 66, 5, 32, 20, 24, 5)$. En este caso, la métrica R transformaría ambos vectores a $\text{rank}(\sim x) = p \sim = (7, 1, 2, 6, 4, 3, 5)$ y $\text{rank}(\sim y) = \sim q = (5, 4, 1, 7, 6, 2, 3)$. Luego, calculamos la distancia de Manhattan dividida por el mayor denominador posible que es $\lfloor \frac{7^2}{2} \rfloor = 24$. El resultado es

$$\begin{aligned} \mathcal{L}^1(\vec{p}, \vec{q}) &= \frac{|7 - 5| + |1 - 4| + \dots + |3 - 2| + |5 - 3|}{24} \\ &= \frac{2 + 3 + 1 + 1 + 2 + 1 + 2}{24} = \frac{1}{2} \end{aligned} \quad (\text{A.13})$$

A.1.4 Otras distancias

Existen muchos otros tipos de métricas, una de ellas es el conteo de valores atípicos mencionada en la sección 2.5.3.5. Una métrica muy similar es la métrica A de Gunetti y Picardi (2005).

A.1.4.1 Métrica A

A diferencia de las otras métricas, la métrica A no analiza las distancias entre los vectores, sino que la recorre y verifica que cada par de valores no difiera más que un umbral relativo establecido.

La definición se muestra en (A.14), donde Ω_t se define en (A.15). La constante t es un valor de umbral inherente a la métrica.

$$A^t(\vec{x}, \vec{y}) = \sum_{i=1}^n \frac{\Omega_t(x_i, y_i)}{n} \quad (\text{A.14})$$

$$\frac{x}{t} \leq y \leq xt \quad \Omega_t(x, y) = \begin{cases} 0 \\ 1 \end{cases} \quad (\text{A.15})$$

En el ejemplo de la figura 2.8 de la sección 2.4.3 contamos con los vectores $\sim x = (29,13,15,20,16,15,17)$ y $\sim y = (31,30,13,66,5,32,20,24,5)$. En este caso la métrica A, utilizando un umbral $t = 1,25$ calcularía el valor de Ω para cada par x_i, y_i obteniendo $(0,1,0,1,1,1,1)$, luego

$$A^{1,25}(\vec{x}, \vec{y}) = \frac{0 + 1 + 0 + 1 + 1 + 1 + 1}{7} = \frac{5}{7} \quad (\text{A.16})$$

A.1.5 Distancias normalizadas por desvío, varianza o covarianza

Al contarse con información estadística sobre los vectores, una mejora propuesta por varios autores (ver Killourhy y Maxion, 2009) sobre las métricas es la ponderación de los datos utilizando una medida de dispersión, ya sea la varianza o el desvío. Esta idea se justifica en que las teclas con alta variabilidad no deberían aportar la misma información que las teclas con variabilidad muy baja. Por ejemplo, si una persona siempre tarda 20 ms en presionar la tecla E y es comparado ese patrón con uno donde se tardó 25 ms, es de esperar que esos 5 ms de diferencia pesen más que la diferencia entre otra tecla, supongamos la tecla O con un patrón entre 10 y 50 ms, con una media de 30ms. Una muestra de 35ms en la tecla O debería atenuar su distancia porque en definitiva, se encuentra dentro del rango observado.

Es por esto que las distancias pueden dividir los valores de cada dimensión por su desvío, para así normalizar las muestras, que ahora se medirán en cantidad de desvíos y no en milisegundos.

A.1.5.1 Distancia euclídea normalizada

Utilizando este método se puede derivar la fórmula para la distancia euclídea como

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^N \frac{(\vec{x} - \vec{y})^2}{s_i^2}} \quad (\text{A.17})$$

donde s_i^2 es la varianza de cada elemento del patrón (*vecx*). Algunas implementaciones utilizan la suma de ambas varianzas. También suele ser común sumar una constante pequeña para evitar la división por cero cuando solo se cuenta con una muestra para ese evento y la varianza, por consiguiente es nula.

Si se utiliza álgebra vectorial al cuadrado puede escribirse como

$$d(\vec{x}, \vec{y})^2 = \frac{\|\vec{x} - \vec{y}\|^2}{s_i^2} = (\vec{x} - \vec{y})^T \cdot S^{-1} \cdot (\vec{x} - \vec{y}) \quad (\text{A.18})$$

donde S^{-1} es una matriz diagonal conteniendo los valores inversos de las varianzas. Si S^{-1} es la matriz identidad, se está en presencia de una distancia euclídea común.

A.1.5.2 Distancia de Mahalanobis

La distancia euclídea normalizada utiliza la matriz S^{-1} con ceros fuera de la diagonal, pero estos valores podrían modificarse teniendo en cuenta las correlaciones entre las teclas. Una nueva ampliación a esta distancia es reemplazar S^{-1} por C^{-1} donde C es la matriz de correlación. De esta forma, las correlaciones entre los valores se normalizarán.

Finalmente, se deriva la siguiente ecuación Mahalanobis (1936)

$$d(\sim x, \sim y) = (\sim x - \sim y)^T \cdot C^{-1} \cdot (\sim x - \sim y) \quad (\text{A.19})$$

A.2 Redes neuronales y algoritmo de *backpropagation*

Las redes neuronales artificiales son modelos computacionales cuya estructura intenta imitar, de manera simplificada, la configuración del sistema nervioso central de los animales y especialmente al cerebro. A través de este paralelo biológico, la investigación de las redes neuronales artificiales no solo pretende arrojar luz sobre el funcionamiento de este último sino que también intenta superar las limitaciones inherentes a los sistemas deductivos y basados en reglas, buscando alcanzar la posibilidad de realizar inferencias complejas sobre conjuntos de datos con un elevado nivel de ruido o ambigüedad que puedan aproximarse a lo que en términos coloquiales se considera intuitivamente sencillo. Algunas tareas que parecen elementales a un ser humano, incluso no entrenado, como la clasificación, el reconocimiento de patrones o la motilidad son particularmente difíciles de lograr con sistemas inteligentes artificiales (Pinker, 1994); es precisamente en este tipo de tareas donde las redes neuronales obtienen buenos resultados.

Una red neuronal artificial consiste en un conjunto de elementos denominados *neuronas artificiales* o simplemente *neuronas* (ver figura A.2), que poseen un conjunto de entradas x_1, \dots, x_n con pesos w_1, \dots, w_n asignados a cada una de ellas, una función de propagación σ y una salida o , que es el resultado de aplicar la función de propagación a la suma ponderada de las entradas.

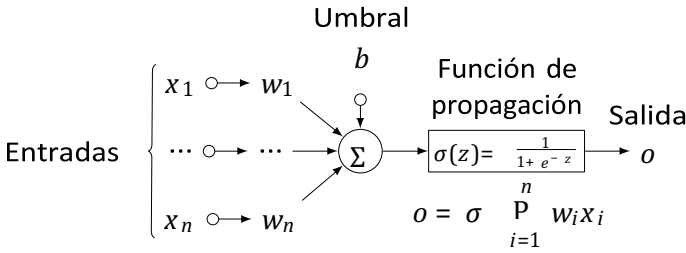


Figura A.2: Modelo de neurona utilizado para la clasificación de usuarios.

Entre las entradas puede existir una destacada que se denomina *umbral*, cuyo efecto es agregar a la neurona artificial una excitación continua y es equivalente a fijar el valor de dicha entrada a una constante. Las neuronas se organizan en *capas*, cuyos elementos no suelen estar conectados entre sí sino solamente con los de la capa anterior y siguiente. La interconexión, también denominada *sinápsis* se realiza alimentando la entrada de una neurona con la salida de otra única neurona; una entrada puede conectarse a una única salida, pero una salida puede estar conectada a cualquier cantidad de entradas. Las capas exteriores de la red neuronal tienen respectivamente el nombre de *capa de entrada* y *capa de salida*; todas las restantes se denominan *capas ocultas*.

Una red neuronal con entrenamiento *backpropagation* permite aprender mediante un conjunto de ejemplo (entrada/salida) comúnmente denominado *training set*. Al haber aprendido mediante este conjunto, se puede obtener una salida coherente para una entrada dada. Este algoritmo fue propuesto en 1986 por Rumelhart *et al.* (1986^a). Gran parte del contenido de esta sección fue tomado de Rumelhart *et al.* (1986b).

El conocimiento de una red neuronal se encuentra codificado en los pesos de las entradas y umbrales. Como un peso $w_i = 0$ asignado a la entrada x_i es equivalente a la desconexión de dicha entrada, se puede presuponer sin pérdida de generalidad que todas las neuronas de una capa se conectan con todas las de la siguiente. Un algoritmo de aprendizaje intenta actualizar dichos pesos para corregir los errores encontrados en los valores de salida cuando se presentan vectores de entrenamiento a la entrada de la red. Los detalles de funcionamiento de los mismos exceden el alcance de esta tesis;

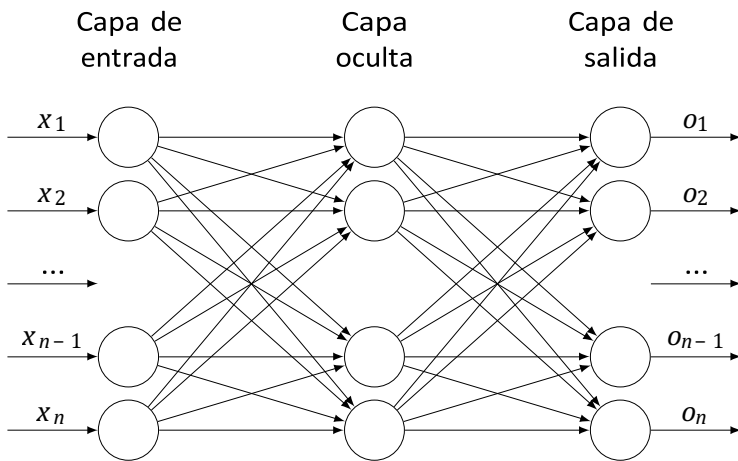


Figura A.3: Red neuronal autoasociativa.

una excelente referencia sobre redes neuronales en donde se trata la relación con los sistemas biológicos, las diversas estructuras básicas y sus correspondientes algoritmos de aprendizaje se puede encontrar en Anderson (1995) y, particularmente sobre la temática de clasificación de patrones y aprendizaje supervisado, en Bishop (1996). La estructura de las redes autoasociativas y el perceptrón de tres capas está ilustrada en las figuras A.4 y A.3, respectivamente. Como se puede apreciar en las figuras A.4 y A.3, cada neurona de entrada —que posee un valor en el rango $[0;1]$ — pasa ese valor a todas las neuronas de salida. Ese valor es multiplicado por el peso $w_{i,j}$ representado en las aristas. El valor de o_j es igual al de una función (denominada de transferencia) aplicada a la sumatoria de todos los productos definidos como $z_j = w_{i,j}x_i$ para el j de esa neurona de salida.

La información que almacena el aprendizaje de una red se encuentra en los pesos $w_{i,j}$ y en ningún lado más. Es muy importante comprender esto, ya que son los pesos los que hay que ajustar en el proceso de entrenamiento.

Existen dos salidas posibles para una red neuronal en modo de entrenamiento la obtenida mediante la red y la deseada (provista en el conjunto de datos, aunque no fue utilizada en la entrada). Al comparar ambas podemos observar cuán buena fue la predicción.

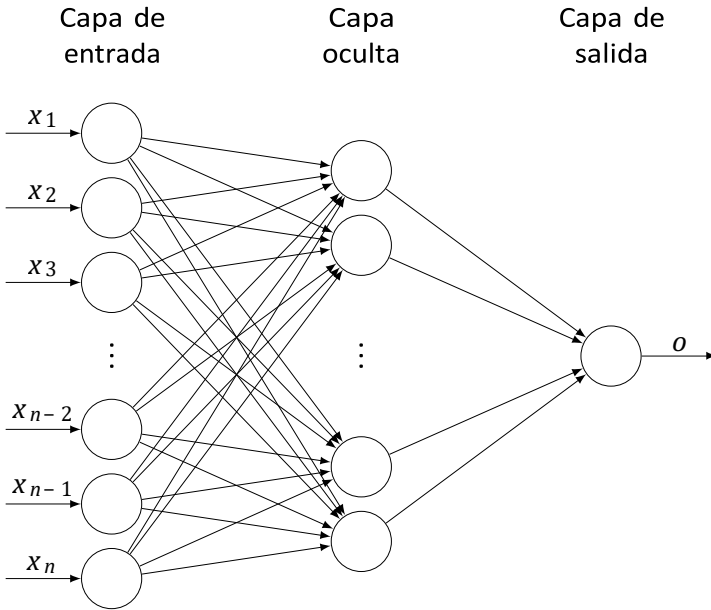


Figura A.4: Red neuronal tipo perceptrón de tres capas.

El objetivo del proceso de entrenamiento es minimizar el error de la predicción y para ello, como se mencionó anteriormente, solo es posible modificar los pesos de la red.

El proceso de entrenamiento es iterativo, se inicializa la red con pesos cargados de manera arbitraria como configuración inicial y luego se tiende a modificarlos de la mejor forma posible. Para ellos se utiliza la propagación del error hacia atrás mediante sus derivadas y es por ello que la red toma el nombre de *back propagation*. Para el error obtenido se encuentra un vector Δw que sumado al vector de pesos w se obtiene una red que arroja un error más pequeño para esa entrada.

Como es de esperar, si se corre para la misma entrada este proceso varias veces, el resultado final sería, siempre y cuando la configuración de la red lo permita, una red con error nulo para ese valor. Ese no es el objetivo, sino lo que se desea es entrenar a la red con varias entradas y luego ver que sucede cuando ingresamos alguna que no estaba en el conjunto de datos de entrenamiento. Es por esta razón que no se itera sobre un mismo elemento del conjunto de datos hasta eliminar el error sino que se realiza un acercamiento con un elemento, luego con otro y así hasta recorrer todo el conjunto de datos. A esta recorrida sobre el conjunto de datos se la suele

denominar *epoch*. El error no será bajo, pero la red se habrá acercado hacia una zona donde convergen todos los elementos. Al repetir el proceso varias veces, es decir iterar varios *epoch*, la red comenzará a entrenarse.

A.2.1 Demostración formal

Ahora procederemos a definir estos conceptos detalladamente. Llamaremos

- \tilde{x} es nuestro vector de entrada, cuyos n elementos denominaremos x_i .
- \tilde{o} es nuestro vector de salida obtenida (por *output*, entrada en inglés), cuyos m elementos denominaremos o_i .
- \tilde{t} es nuestro vector de salida deseada (por *target*, objetivo o blanco en inglés), cuyos m elementos denominaremos t_i .
- w_{\sim} es nuestro vector pesos (por *weight*, peso en inglés), cuyos $n \times m$ elementos denominaremos $w_{i,j}$. Notemos que es un vector unidimensional que pertenece a R^{nm} y no es una matriz perteneciente a $R^{n \times m}$.
- $w_{\sim j}$ es el vector de pesos que llegan a un determinado o_j .
- W es la matriz de pesos, los cuales son los mismos que $w_{i,j}$ pero ahora sí estarán ordenados de manera matricial en $R^{n \times m}$.
- \tilde{z} es un vector intermedio previo a \tilde{o} , cuyos m elementos denominaremos $z_j = \prod_{\forall i} x_i w_{i,j} = \tilde{x} W \tilde{w}_j$.
- $\Gamma(j)$ conjunto de unidades cuyas entradas inmediatas son las salidas de j .

Además, mencionamos la función de transferencia, la cual definimos como

$$o_j = f(z_j) = f\left(\sum_{\forall i} x_i w_{i,j}\right) = f(\vec{x} \vec{w}_j) \quad (\text{A.20})$$

El objetivo de la iteración es, dado \tilde{x} y \tilde{t} , obtener un valor, Δw_{\sim} , que sumado a w_{\sim} nos permita disponer de un mejor conjunto de pesos que acerque más los valores de \tilde{o} a los de \tilde{t} . Para ellos definiremos una función de error, que será un número escalar no negativo cuyo objetivo será minimizarlo mediante el ajuste de los pesos.

Una buena medida del error viene dada por la distancia euclídea de ambos vectores, es decir $\|\vec{o} - \vec{t}\|$. Debido a que no nos interesa la magnitud del error y que a futuro nos simplificará las cuentas, agregaremos una constante multiplicativa de $1/2$ al principio. Definimos nuestra función del error como

$$E = \frac{1}{2} \|\vec{o} - \vec{t}\|^2 = \frac{1}{2} \sum_{j=1}^m (o_j - t_j)^2 \tag{A.21}$$

que para garantizar la continuidad utilizará la norma L_2 al cuadrado como se encuentra en (A.8).

Con el error definido, aplicaremos el operador gradiente para obtener su dirección de máximo crecimiento (siempre derivando con respecto a los w_{ij} que son nuestras variables independientes. La multiplicaremos por -1 para obtener la dirección de máximo decrecimiento y luego la multiplicaremos por un coeficiente que indicará la velocidad en que el algoritmo avanzará. Un coeficiente alto puede hacer que nos pasemos y que el algoritmo diverja, pero un valor muy bajo puede hacernos tardar mucho en llegar el objetivo. Por lo general se utilizan valores entre $0,6$ y $0,1$. Denominaremos al coeficiente como η .

Finalmente, obtenemos la formula que nos permitirá calcular nuestro algoritmo

$$\Delta w \sim = -\eta \nabla E \tag{A.22}$$

donde

$$w \sim_{n+1} = w \sim_n + \Delta w \sim_n \tag{A.23}$$

Siendo $w \sim_0$ un vector aleatorio. Notemos que estos coeficientes se refieren al número de iteración y no a la definición dada anteriormente de $w \sim_j$.

A.2.1.1 Propiedades previas

Antes de comenzar, debemos repasar propiedades matemáticas que son aisladas de nuestro problema ya que su descripción detallada en medio de la explicación del proceso de entrenamiento puede generar confusión.

1. Propiedades de la distribución sigmoidal

La distribución sigmoidal viene dada por la fórmula

$$\sigma_k(x) = \frac{1}{e^{-xk} + 1} \quad (\text{A.24})$$

Podemos observar varias propiedades: La primera es que tiene valores entre 0 y 1 y es biyectiva. Esto se prueba mostrando que su máximo valor es el 1 cuando $\lim_{x \rightarrow \infty} \sigma_k(x) = 1$ y su mínimo valor es 0 cuando x tiende a $-\infty$, según el límite $\lim_{x \rightarrow -\infty} \sigma_k(x) = 0$. Al mostrar que la función es monótonamente creciente, veremos que es biyectiva con un dominio en todos los números reales y una imagen en el intervalo (0;1). Otra propiedad es que

$$1 - \sigma_k(x) = 1 - \frac{1}{e^{-xk} + 1} \quad (\text{A.25})$$

$$= \frac{e^{-xk} + 1}{e^{-xk} + 1} - \frac{1}{e^{-xk} + 1} \quad (\text{A.26})$$

$$= \frac{e^{-xk} + 1 - 1}{e^{-xk} + 1} \quad (\text{A.27})$$

$$= \frac{e^{-xk}}{e^{-xk} + 1} \quad (\text{A.28})$$

$$= \frac{1}{1 + e^{xk}} \quad (\text{A.29})$$

$$= \sigma_k(-x) \quad (\text{A.30})$$

Esto nos muestra que $\sigma_k(x) + \sigma_k(-x) = 1$, o sea que la función es impar sobre un eje de simetría ubicado en 1/2. La derivada de la función sigmoidal se calcula mediante la regla de la cadena como

$$\sigma'_k(x) = -1\left(\frac{1}{e^{-xk} + 1}\right)^{-2} e^{-xk} (-k) \quad (\text{A.31})$$

$$= k \frac{1}{e^{-xk} + 1} \frac{e^{-xk}}{e^{-xk} + 1} \quad (\text{A.32})$$

$$= k \frac{1}{e^{-xk} + 1} \frac{1}{1 + e^{xk}} \quad (\text{A.33})$$

$$= k \sigma_k(x) \sigma_k(-x) \quad (\text{A.34})$$

Como podemos apreciar, es siempre positiva en el intervalo $(0;1)$, nuestra función es monótonamente creciente y, por lo tanto, como se menciona arriba, es biyectiva en todo el intervalo.

Además, por la propiedad de simetría, podemos decir que $\sigma_k^0(x) =$

$$k \sigma_k(x) \sigma_k(-x) = k \sigma_k(x) (1 - \sigma_k(x)) \quad (\text{A.35})$$

por lo que, si se conoce el resultado, es posible calcular su derivada sin la necesidad de hallar el x que la genera, acelerando los cálculos de derivadas.

Por simplicidad de cuentas utilizaremos $\sigma(x) = \sigma_1(x) = \frac{1}{e^{-x} + 1}$ a la función sigmoideal de $k = 1$.

2. Regla de la cadena

La regla de la cadena nos permite derivar funciones compuestas de manera independiente una de la otra, en nuestro caso la utilizaremos para derivar funciones con varias variables.

La regla viene dada por la formula

$$\frac{\partial E}{\partial w} = \sum_{\forall i} \frac{\partial E}{\partial p_i} \frac{\partial p_i}{\partial w} \quad (\text{A.36})$$

pudiéndose agregar arbitrariamente cualquier cantidad de variables intermedias siempre y cuando se sumen todas las combinaciones.

3. Derivada de una sumatoria con constantes.

Es conveniente aclarar esta propiedad matemática antes de empezar el desarrollo de redes neuronales.

Al derivar en una sumatoria con varias constantes, solo sobreviven las derivadas que dependen de la variable sobre la cual derivamos, es decir

$$\frac{\partial(x_1 + x_2 + x_3 + x_4)}{\partial x_3} = \frac{\partial x_3}{\partial x_3} = 1 \quad (\text{A.37})$$

Ahora, si generalizamos este planteo es fácil de ver que

$$\frac{\partial \sum_{\forall i} x_i}{\partial x_j} = \frac{\partial x_j}{\partial x_j} = 1 \quad (\text{A.38})$$

Es decir, la única variable que sobrevive es en el caso $i = j$, el resto son constantes y su derivada es nula.

Agregando una constante multiplicativa (w_i) a cada término, vemos que solo sobrevive la constante del término perteneciente a la variable independiente.

$$\frac{\partial \sum_{\forall i} w_i x_i}{\partial x_j} = \frac{\partial (w_j x_j)}{\partial x_j} = w_j \quad (\text{A.39})$$

Generalmente, se utiliza la definición del delta de Kronecker para eliminar sumatorias de este tipo.

A.2.1.2 Demostración del algoritmo sin capas intermedias

En la presente sección veremos cómo hallar Δw a partir de un vector de entradas \tilde{x} , uno de salidas deseadas \tilde{t} , nuestros pesos w y el valor del paso (o velocidad de convergencia) η .

Como mencionamos anteriormente, debemos hallar el gradiente de E con respecto a los pesos, es decir

$$\nabla E = \frac{\partial E}{\partial w_{i,j}} \quad (\text{A.40})$$

Por la regla de la cadena (propiedad 2, ecuación A.36), sabemos que

$$\nabla E_u = \frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial z_i} \frac{\partial z_i}{\partial w_{i,j}} \quad (\text{A.41})$$

siendo u la posición en E de cada par i,j .

Calcularemos ambos términos por partes comenzando por la derivada del error con respecto a z_i y aplicando la propiedad 3 a la ecuación A.21 obtenemos que

$$\frac{\partial E}{\partial z_j} = \frac{1}{2} \frac{\partial \sum_{n=1}^m (o_n - t_n)^2}{\partial z_j} = \frac{1}{2} \frac{\partial (o_j - t_j)^2}{\partial z_j} \quad (\text{A.42})$$

Luego, por las propiedades de la función sigmoideal (utilizaremos la de $k = 1$) y que $o_j = \sigma(z_j)$, podemos llegar a la siguiente expresión

$$\frac{\partial E}{\partial z_j} = \frac{1}{2} \frac{\partial (o_j - t_j)^2}{\partial z_j} \quad (\text{A.43})$$

$$= \frac{1}{2} \frac{\partial (\sigma(z_j) - t_j)^2}{\partial z_j} \quad (\text{A.44})$$

$$= \frac{1}{2} 2 \frac{\partial (\sigma(z_j) - t_j)}{\partial z_j} (\sigma(z_j) - t_j) \quad (\text{A.45})$$

$$= \sigma(z_j)(1 - \sigma(z_j))(\sigma(z_j) - t_j) \quad (\text{A.46})$$

$$= o_j(1 - o_j)(o_j - t_j) \quad (\text{A.47})$$

Definiremos

$$\delta_j = \frac{\partial E}{\partial z_j} = o_j(1 - o_j)(o_j - t_j) \quad (\text{A.48})$$

ya que solo depende de j , con lo cual, al iterar este cálculo puede realizarse una sola vez.

Ahora veremos la segunda parte que es $\frac{\partial z_i}{\partial w_{i,j}}$, la cual, por la propiedad 3, se puede resolver en

$$\frac{\partial z_i}{\partial w_{i,j}} = \frac{\partial \sum_{\forall k} x_k w_{k,j}}{\partial w_{i,j}} = x_i \quad (\text{A.49})$$

Finalmente, obtuvimos que

$$\frac{\partial E}{\partial w_{i,j}} = \delta_j x_i = o_j(1 - o_j)(o_j - t_j)x_i \quad (\text{A.50})$$

Por lo tanto, para ajustar un peso $w_{i,j}$ debemos hacer

$$w_{n+1,i,j} = w_{n,i,j} - \eta o_j(1 - o_j)(o_j - t_j)x_i \quad (\text{A.51})$$

A.2.1.3 Ampliación a n capas intermedias Utilizando la definición previa de $\Gamma(j)$ y aplicando las derivadas parciales se obtiene, mediante la regla de la cadena la siguiente sumatoria

$$\frac{\partial E}{\partial w_{ji}} = \sum_{\forall k \in \Gamma(j)} \frac{\partial E}{\partial z_k} \frac{\partial z_k}{\partial o_j} \frac{\partial o_j}{\partial z_j} \frac{\partial z_j}{\partial w_{ji}} = \sum_{\forall k \in \Gamma(j)} \frac{\partial E}{\partial z_k} \frac{\partial z_k}{\partial o_j} \frac{\partial o_j}{\partial z_j} x_{ji} = \delta_j x_{ji} \quad (\text{A.52})$$

Podemos observar que δ_j ahora se calcula mediante una sumatoria, pero todo el proceso se realiza análogamente.

Sustituyendo $\frac{\partial E}{\partial z_k} = \delta_k$, $\frac{\partial z_k}{\partial o_j} = w_{k,j}$ y $\frac{\partial o_j}{\partial z_j} = o_j(1 - o_j)$ nueva δ_j obtenemos que

$$\delta_j = \sum_{\forall k \in \Gamma(j)} \frac{\partial E}{\partial z_k} \frac{\partial z_k}{\partial o_j} \frac{\partial o_j}{\partial z_j} = \sum_{\forall k \in \Gamma(j)} \delta_k w_{kj} o_j(1 - o_j) \quad (\text{A.53})$$

Como los o_j no dependen de k se pueden quitar de la sumatoria obteniendo

$$\delta_j = o_j(1 - o_j) \sum_{\forall k \in \Gamma(j)} \delta_k w_{kj} \quad (\text{A.54})$$

Como se puede observar, el entrenamiento es muy similar al de la capa final, salvo que esta vez tenemos una sumatoria.

Para generalizar decimos que en la última capa calculamos δ_{k_0} como

$$\delta_{k_0} = o_{k_0}(1 - o_{k_0})(t_{k_0} - o_{k_0}) \quad (\text{A.55})$$

mientras que en el resto lo hacemos con δ_{k_n} como

$$\delta_{k_n} = o_{k_n}(1 - o_{k_n}) \sum_{\forall k_{n-1} \in \Gamma(k_n)} w_{k_{n-1}k_n} \delta_{k_{n-1}} \quad (\text{A.56})$$

Finalmente, los pesos se ajustan mediante

$$w_{n+1,ij} = w_{n,ij} - \eta o_j \delta_j x_i \quad (\text{A.57})$$

A.3 Bosques aleatorios

Los *bosques aleatorios* fueron introducidos por Breiman (2001) como un método novedoso para la clasificación y se encuentran entre los mejores algoritmos existentes, superando en algunos casos a las máquinas de soporte vectorial. Algunas de sus ventajas son el comportamiento robusto frente a conjuntos de datos de entrenamiento ruidosos y su capacidad de extraer características significativas en forma automática, permitiendo de esta manera una representación explícita del conocimiento de la que carecen las redes neuronales. Es demostrable que alcanzan asintóticamente la mínima tasa de error posible para el conjunto de parámetros característicos de entrenamiento.

Como método de aprendizaje conjunto, los bosques aleatorios consisten en una agrupación de clasificadores del tipo árbol de decisión; cuando se les presenta un vector de entrada, cada árbol lo clasifica y el algoritmo elige como categoría final aquella que ha recibido mayor cantidad de votos de árboles individuales. A los fines de elaborar cada clasificador, durante la etapa de entrenamiento se particiona el conjunto de vectores característicos por medio de una técnica denominada *bagging*, del inglés *bootstrap aggregating*. Con un conjunto de n vectores D se construyen m nuevos conjuntos D_1, \dots, D_m

de tamaño n^0 extrayendo muestras uniformemente y con reposición. Es posible que un cierto $\mathbf{v} \in D$ aparezca repetido en alguno de los D_i pero esto no afecta el desempeño del algoritmo. El efecto de este muestreo es estabilizar el algoritmo conjunto respecto del ruido que puedan introducir los clasificadores individuales.

A su vez, cada vector de l componentes $\mathbf{v} = (v_1, \dots, v_l) \in D_i$ es proyectado sobre un subespacio de menor dimensión, obteniéndose un nuevo vector de k componentes $\mathbf{v}^0 = (v_{j_1}, \dots, v_{j_k})$ en donde

$1 \leq j_1 < \dots < j_k$ y los valores de k oscilan típicamente en torno a l . Los valores de j_1, \dots, j_k se eligen en forma aleatoria y uniforme para cada D_i . El efecto de tal selección es producir una multitud de árboles que se entrenan en base a subconjuntos pequeños de los parámetros característicos; de esta forma, los árboles a los que se les hayan asignado parámetros poco significativos tenderán a clasificar en forma aleatoria pero todos aquellos que respondan a los más importantes clasificarán en forma correcta y correlacionada, obteniéndose la mencionada ventaja de extraer características significativas automáticamente.

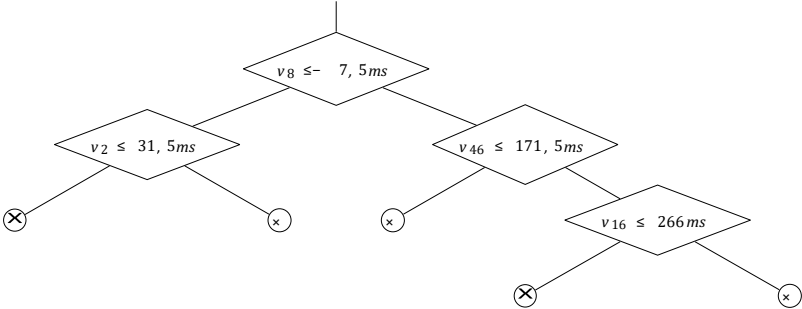


Figura A.5: Árbol de decisión

A.4 Árboles de decisión

En el contexto que nos compete, un *árbol de decisión* es una estructura en forma de árbol cuyos nodos representan condiciones que hacen referencia a valores de los parámetros del vector característico y sus hojas representan la clasificación asignada. De esta forma, dado un vector característico, el algoritmo de clasificación recorre el árbol

desde su raíz tomando la bifurcación que corresponda según qué valor tomen los parámetros indicados en el condicional y descendiendo de nodo en nodo hasta llegar a una hoja, momento en el cual el proceso finaliza y la categoría asignada es la que está indicada en esta última.

A.4.1 Generación con C4.5

Dentro de los algoritmos de generación de árboles de decisión, uno muy popular es C4.5, desarrollado por Quinlan (1993). Este es una mejora del algoritmo ID3 del mismo autor.

C4.5 utiliza un enfoque *divide-and-conquer* para hacer crecer árboles de decisión que fue utilizado inicialmente por Hunt *et al.* (1966).

Este algoritmo genera un árbol de decisión de un conjunto de casos denotado D :

- Si D cumple con un *criterio de corte*, el árbol para D es una hoja asociada con la clase más frecuente de D . Una razón para detenerse es que D contiene solamente casos de su clase, sin embargo otros criterios podrían ser formulados.
- Alguna condición T con productos mutuamente excluyentes T_1, T_2, \dots, T_k es utilizada para particionar D en subconjuntos D_1, D_2, \dots, D_k , donde D_i contiene aquellos casos que tienen el resultado T_i . El árbol para D contiene la condición T como su raíz y un subárbol por cada resultado de T_i que se construye aplicando el mismo procedimiento de manera recursiva a los casos en D_i .

Dado que no hay casos con valores de atributos idénticos que pertenecen a distintas clases, cualquier condición T que produce una partición no trivial de D llegará eventualmente a producir subconjuntos de una sola clase. Sin embargo, sabiendo que los árboles más chicos son preferibles (por su facilidad de entendimiento y en algunos casos predictores más precisos), una familia de condiciones posibles es examinada y uno de ellos es elegido para maximizar el valor

de algunos *criterios de corte*. Las condiciones consideradas por defecto en C4.5 son:

- $A = ?$ para un atributo discreto A , con un solo resultado para cada valor de A .
- $A \leq t$ para un atributo continuo A , con dos resultados, *verdadero* y *falso*. Para encontrar el umbral t que maximiza el criterio de partición, los casos en D son ordenados por sus valores del atributo A para generar distintos valores v_1, v_2, \dots, v_N . Cada par de valores adyacentes sugiere un umbral potencial $t = (v_i + v_{i+1})/2$ y una partición correspondiente de D . El umbral con el mejor valor del criterio de partición es seleccionado.

El criterio de partición utilizado por defecto en C4.5 es la ganancia de información, una medida basada en la información que toma en cuenta diferentes números (y diferentes probabilidades) de los resultados producidos por la condición. Se denota C el número de clases y $p(D,j)$ la proporción de casos en D que pertenece a la clase j -ésima. La incertidumbre residual sobre la clase a la cual pertenece un caso en D puede ser expresada como

$$\text{Info}(D) = -\sum_{j=1}^C p(D,j) \times \log_2(p(D,j)) \quad (\text{A.58})$$

y la información ganada correspondiente por la condición T con k resultados como

$$\text{Gain}(D, T) = \text{Info}(D) - \sum_{i=1}^k \frac{|D_i|}{|D|} \times \text{Info}(D_i) \quad (\text{A.59})$$

La información ganada por una condición está fuertemente afectada por el número de resultados y es máximo cuando hay un caso en cada subconjunto D_i . Por otra parte, la información potencial obtenido al particionar un ser de casos está basada en saber en qué caso cae cada subconjunto D_i ; esta partición de información

$$\text{Split } (D, T) = - \sum_{i=1}^k \frac{|D_i|}{|D|} \times \log_2 \left(\frac{|D_i|}{|D|} \right) \quad (\text{A.60})$$

tiende a incrementar con el número de resultados de una condición. El criterio de ganancia de información evalúa la conveniencia de una condición como la proporción de su ganancia de información a su información de partición. La ganancia de cada condición posible es determinada y, entre aquellos con al menos una ganancia promedio, la partición con ganancia máxima es seleccionada.

En algunas situaciones, cada condición posible parte a D en subconjuntos que tienen la misma distribución de clases. Todas las condiciones tienen entonces ganancia cero, y C4.5 utiliza esto como un criterio adicional de corte.

La estrategia de partición recursiva arriba mencionada resulta en árboles que son consistentes con los datos de entrenamiento, de ser esto posible. En aplicaciones prácticas, los datos suelen ser ruidosos — los valores de los atributos fueron recogidos incorrectamente y los casos son clasificados erróneamente—. El ruido conlleva a árboles demasiado complejos que intentan tener en cuenta estas anomalías. La mayoría de los sistemas *podan* el árbol inicial, identificando subárboles que contribuyen poco a la precisión predictiva reemplazándolos por una hoja (Quinlan, 1996).

A.5 Tests de hipótesis

Los tests de hipótesis son necesarios para una validación rigurosa de los resultados. No basta con que los mismos se vean sólidos, hay que realizar un test estadístico para garantizar que esta solidez no fue producto del azar. Aumentar el tamaño de la muestra influye positivamente a la hora de mejorar la calidad de los resultados, sin embargo no siempre es posible incrementar la muestra. Los tests ayudan a verificar que ese tamaño es lo suficientemente grande como para garantizar, bajo cierto margen de error, o (CI por sus siglas en inglés, denotado α), que los resultados realmente validan o no una hipótesis.

Algunos tests, como el t-Test pareado, requieren normalidad, es decir, que sus muestras hayan sido tomadas de una población distribuida normalmente. Por este motivo, es necesario aplicar otros tests previamente para garantizar normalidad en las muestras; de lo contrario, se estaría fallando en cumplir con los requisitos del test inicial y su resultado sería incierto.

La normalidad puede ser observada gráficamente mediante la realización de un histograma. Las pruebas de Shapiro–Wilk (Shapiro y Wilk, 1965) o la de Kolmogorov–Smirnov responden, numéricamente, si las muestras son normales. Estos tests son muy sensibles al efecto del tamaño de la muestra.

A diferencia de Shapiro–Wilk, las pruebas de Kolmogorov–Smirnov requieren que no existan valores iguales en las muestras, por lo que a los propósitos de este trabajo, fue descartada.

El p -value o nivel de significancia es la probabilidad correspondiente a al estadístico muestral (valor arrojado por los tests) de que la hipótesis sea cierta. Es decir que si $p < \alpha$ se acepta la hipótesis y de lo contrario, se rechaza. En esta tesis se presentarán ambos valores, tanto el estadístico muestral como el valor de p .

A.5.1 Shapiro–Wilk

La prueba de Shapiro–Wilk es una manera de determinar si una muestra aleatoria proviene de una distribución normal. La prueba arroja un valor de W ; valores pequeños indican que la muestra no está distribuida normalmente (puede rechazar la hipótesis nula de que su población está distribuida normalmente si sus valores están por debajo de un cierto umbral).

Conceptualmente, la prueba de Shapiro–Wilk examina la cercanía entre las muestras de datos después de que se ordenaron y estandarizaron (es decir, se transformaron a una media nula y varianza unitaria) y qué muestras habrían sido si se hubieran extraído de una distribución normal estándar ordenado. Las muestras ordenadas extraídas de la distribución normal estándar serían cuantiles monótonicamente crecientes —por la función de densidad acumulativa inversa (CDF por sus siglas en inglés)— de la distribución

normal estándar. Esta métrica de cercanía (que es el estadístico de prueba W de Shapiro–Wilk) es básicamente el cuadrado de la correlación entre el conjunto de datos ordenados y estandarizados y los valores de CDF inversos. El estadístico de prueba es, por lo tanto, un valor positivo con un límite superior unitario: cuanto más alto es este valor, más Normales o Gaussianos son los datos.

La fórmula para el valor W es:

$$W = \frac{(\sum_{i=1}^n a_i x_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (\text{A.61})$$

donde: x_i son los valores muestrales aleatorios ordenados, a_i son los coeficientes de Shapiro–Wilk, que dependen de n y se derivan de una muestra distribuida normalmente.

La prueba tiene limitaciones, lo más importante es que la prueba tiene un sesgo por el tamaño de la muestra. Cuanto mayor sea la muestra, más probable es que obtenga un resultado estadísticamente significativo.

A.5.2 Prueba de los rangos con signo

La prueba de los rangos con signo de Wilcoxon es una alternativa no paramétrica a la prueba t pareada que es similar a la prueba del signo de Fisher. Esta prueba asume que hay información en las magnitudes de las diferencias entre las observaciones pareadas, así como los signos.

La hipótesis nula establece que la diferencia entre los pares sigue una distribución simétrica alrededor de cero.

Para calcular el estadístico se toman las observaciones pareadas y se calculan las diferencias

$$\omega_i = (X_i - Y_i) \forall i < n \quad (\text{A.62})$$

Se eliminan los casos donde $\omega_i = 0$ y se ranquea el resto de menor a mayor según el calor de $|\omega_i|$. El vector resultante R , contendrá n^0 observaciones con $n^0 \leq n$. Finalmente, el estadístico se calcula como

$$W = \sum_{i=1}^{n^0} \text{sgn}(\omega_i) R_i \quad (\text{A.63})$$

Finalmente, el p -value asociado con este estadístico se encuentra en el cuadro correspondiente.

A.5.3 t-Test pareado

Dados dos conjuntos emparejados X_i y Y_i de n valores medidos, la prueba t pareada determina si difieren entre sí de una manera significativa bajo los supuestos de que las diferencias emparejadas son independientes y se distribuyen de la misma manera.

Para aplicar la prueba, primero es necesario correr las muestras hasta centrarlas en 0.

$$\begin{aligned} \hat{X}_i &= (X_i - \bar{X}), \\ \hat{Y}_i &= (Y_i - \bar{Y}) \forall i < n \end{aligned} \quad (\text{A.64})$$

luego el estadístico t se define como

$$t = (\bar{X} - \bar{Y}) \sqrt{\frac{n(n-1)}{\sum_{i=1}^n (\hat{X}_i - \hat{Y}_i)^2}} \quad (\text{A.65})$$

Se puede usar una tabla de los intervalos de confianza de la distribución t de Student para determinar el p -value.

A.5.4 Prueba U de Mann–Whitney

En las estadísticas, la prueba U de Mann–Whitney —también llamada prueba de Mann–Whitney–Wilcoxon (MWW), Wilcoxon de suma de rangos o prueba de Wilcoxon–Mann–Whitney— es una prueba no

paramétrica de la hipótesis nula, verificando si es igualmente probable que un valor seleccionado al azar de una muestra será menor o mayor que un valor seleccionado al azar de una segunda muestra.

A diferencia de la prueba t , no requiere el supuesto de distribuciones normales. Es casi tan eficiente como la prueba t en distribuciones normales.

Esta prueba se puede usar para determinar si se seleccionaron dos muestras independientes de poblaciones que tienen la misma distribución; una prueba no paramétrica similar utilizada en muestras dependientes es la de los rangos con signo de Wilcoxon (Hodges y Lehmann, 1963).

Herramientas y *frameworks* desarrollados

Resulta necesario garantizar la replicabilidad del experimento y para ello los datos y algoritmos fueron publicados en nuestro repositorio de *github*⁷. No obstante, los fragmentos de código de mayor importancia serán incluidos en este apéndice para evitarle al lector la navegación y búsqueda de los mismos en el repositorio cuando el mismo desee despejar dudas sobre los algoritmos utilizados. Para entender los códigos es necesario poseer conocimientos en programación, además de conocer como funciona la arquitectura web y entender varios aspectos técnicos.

B.1 Captura de la cadencia de tecleo

El presente código utiliza los estándares de la web y funciona en Google Chrome, Safari, Mozilla Firefox y navegadores compatibles; sin embargo, los códigos capturados pueden diferir según el navegador. Esto se debe a que el estándar no especifica qué tecla corresponde a qué código y se dejó a libre interpretación de los desarrolladores de navegadores, quienes han utilizando códigos distintos para ciertas teclas de control en algunos casos. Las teclas comunes comparten el mismo código en todos los navegadores, que coincide con el código ASCII de sus respectivos caracteres. Para capturar la cadencia de tecleo se implementó el código B.1, que, además, estará presente en la demostración incluida en la defensa del presente trabajo.

```
1 window['Keylogger']=function($selector,callbackEnterFunction, on_end_callback) {
```

⁷ Disponible en <https://github.com/lisia>, cuenta con varios repositorios *Git* disponibles destinados a distintos set de datos y los algoritmos.

```

2      var
3      n=navigator['userAgent'],last,
4      keystroke,
5      bufferstroke,textstroke,delstroke,
6      add=function(v) {keystroke+=v;},
7      radd=function(v){add(radix2(v))},
8      init=function() {
9      keystroke=bufferstroke=delstroke=textstroke="";
10     last=0;
11     add(radix2(n.length)+n);
12     }
13     init();
14
15
16     $selector['on']('keydown keyup',function(e){
17     radd(e['originalEvent']['location']<<9|e['keyCode']<<1|(e
18     ['type']!= 'keyup'?0:1))
19     last=e['timeStamp'];
20     });
21     $selector['on']('keypress',function(e){
22     var eventChar = String.fromCharCode(e.keyCode || e.
23     charCode);
24     if (e['keyCode']==8) {
25     if (textstroke) {
26     delstroke=textstroke.slice(-1)+delstroke;
27     textstroke=textstroke.slice(0,-1);
28     }
29     } else if (e['keyCode']==13){
30     callbackEnterFunction(bufferstroke+textstroke,keystroke );
31
32     } else if ( eventChar.length==1){
33     if (delstroke) {
34     bufferstroke+=textstroke+'<s>'+delstroke+'</s>';
35     delstroke=textstroke="";
36     }
37     textstroke+=eventChar;
38     } else {
39     textstroke+=' ';
40     }
41     });
42     function radix2(number) {
43     var
44     out="";
45     /** @const */var
46     encode1="0124689qwertyuiopsdfgASDFGHJKLZX", 45     encode2="35
47     ahjklzxcvbnmQWERTYUIOPCVBNM7#";
48     while (number>31) {
49     out+=encode1.charAt(number & 31);

```

```

48     number=number>>5;
49     }
50     return out+encode2.charAt(number);
51     }
52
53     return function() {
54     on_end_callback(bufferstroke+textstroke,keystroke);
55     init();
56     }
57     }

```

Código B.1: Código del Keystroke Logger

B.2 Framework en Python

Un módulo en Python fue desarrollado y puede ser utilizado en combinación con Jupyter para el análisis posterior de los datos adquiridos. Permite clasificar de forma genérica los datos utilizando múltiples métricas, calcular EER y generar los respectivos gráficos. Las métricas son muy simples de definir utilizando una función lambda de Python. Su gran ventaja es la simpleza y facilidad de uso aunque, desafortunadamente, procesar grandes lotes de datos con esta plataforma se hace una tarea tediosa y en muchos casos inviable. El tiempo procesamiento necesario para los experimentos de esta tesis utilizando una máquina de 72 núcleos fue estimado en 11 meses; además, la cantidad de memoria RAM necesaria resultó inviable. Por ello se decidió codificar el mismo en el lenguaje C para optimizar al máximo las ejecuciones. El código B.2 muestra la base del *framework*. El contenido es accesible desde la cuenta del Laboratorio de Sistemas de Información Avanzados en github.

```

1     """
2     Usage:
3     Example of classifier by gender with 100 records per gender and filtered to
4     300 key events using Manhattan distance.
5
6
7     import db
8     import kd
9
10
11                                     it = db.read('dump.2020-12-16.csv') # iterator
12                                     agg = Aggregator(it) # Aggregator
13                                     dataset = agg.collect(lambda x: x['gender'], bin_size
                                     =100,
                                     total_bins={'m', 'f'}) # dict

```

```

14
15         def prepare(cls, x):
16             ed = kd.Keystroke(x['ks']).event_dict(**kd.
DEFAULT_CONFIG)
17             if len(ed) < 300:
18                 return None
19             return {'id': x['id'], 'ed': ed}
20
21         c = Classifier(classify(
22             dataset,
23             distance=lambda x, y: Distance(x['ed'], y['ed']),
24             prepare=prepare,
25             filter=lambda x, y: x['id'] <= y['id']
26             )) # Classifier
27
28         results = c.evaluate_metric('cityblock') #
ClassificationResults
29
30     Other examples:
31
32     * 50 sessions from 100 users:
33     dataset = agg.collect(lambda x: x['user'], bin_size=50,
total_bins=100)
34
35         * Don't filter small events
36         prepare=lambda cls, x: {'id': x['id'],
37             'ed': kd.Keystroke(x['ks']).event_dict(**kd.
DEFAULT_CONFIG)}
38
39     * All users, max 1000 sessions per person (full walk)
40     dataset = agg.collect(lambda x: x['user'], bin_size=1000) # Classifier
41
42     * 100 users, all sessions per person (full walk)
43     dataset = agg.collect(lambda x: x['user'], total_bins
=100) # Classifier
44
45     * classify users, limit to the first 1000 sessions
46     dataset = agg.collect(lambda x: x['user'], slice=(1000,)) # Classifier
47
48     """
49
50 import numpy as np
51
52     from .biometrics import attempts_to_info, info_to_auc, info_to_eer, plot_eer, \
53         sorted_attempts
54     from .latex import latex_det, latex_eer, latex_roc
55     from .utils import cached_property
56
57     from bisect import bisect
58     from collections import defaultdict
59     from itertools import islice
60     from operator import itemgetter
61     from scipy.spatial import distance as scd
62     from scipy.stats import rankdata
63     from sklearn import metrics as skm
64

```



```

65
66     class ClassificationResults(object):
67     def __init__(self, attempts):
68     info = attempts_to_info(attempts)
69
70     self.cumulative = info['cumulative']
71     self.impostor = info['impostor']
72     self.legit = info['legit']
73     self.max_sensitivity = info['max_sensitivity']
74
75     self._auc = None
76     self._eer = None
77
78     @cached_property
79     def
total(self):
80         return self.impostor + self.legit
81
82     def confusion_matrix(self, threshold=None):
83     if threshold is None:
84     threshold = self.eer['sensitivity']
85
86         i = bisect(self.cumulative[:, 0], threshold)
87
88         (_, frr, far) = self.cumulative[i]
89
90     false_rejection = frr * self.impostor
91     false_acceptance = far * self.legit
92     true_impostor = self.impostor - false_rejection
93     true_legit = self.legit - false_acceptance
94
95     return np.array(
96     [[true_legit, false_acceptance], [false_rejection
, true_impostor]],
97     dtype=np.int32)
98
99     @property
100    def auc(self):
101    if self._auc is None:
102    self._auc = info_to_auc(self.cumulative)
103    return self._auc
104
105    @property
106    def eer(self):
107    if self._eer is None:
108    self._eer = info_to_eer(self.cumulative)
109    return self._eer
110
111    def plot(self):
112    plot_eer(self.cumulative, eer=self.eer['eer'],
113    sensitivity=self.eer['sensitivity'])
114
115    def latex_eer(self, eer=True, sensitivity=True, nsteps
=40):
116    return latex_eer(self.cumulative,
117    eer=eer and self.eer['eer'],
118    sensitivity=sensitivity and self.eer

```

```

119         ['sensitivity'],
120                                     nsteps=nsteps)
121     def latex_roc(self, auc=True, nsteps=40):
122         return latex_roc(self.cumulative, auc=auc and self.
123         auc, nsteps=nsteps)
124
125     def latex_det(self, eer=True, nsteps=60):
126         return latex_det(self.cumulative, eer=eer and self.
127         eer['eer'],
128                                     nsteps=nsteps)
129
130 class Aggregator(object):
131     def __init__(self, iterator): 131 self.iterator = iterator
132
133     def collect(self, classification_function, slice=None, bin_size=None,
134     total_bins=None):
135         # type: (Callable, Union[tuple, NoneType], Union[int,
136     NoneType],
137                 # Union[int, NoneType]) -> dict
138         iterator = self.iterator
139         if slice:
140             iterator = islice(iterator, *slice)
141             bins = defaultdict(list)
142
143         total_bins_count = None 143     if total_bins:
144             if type(total_bins) is set: 145                 total_bins_count = len(total_bins)
145             else:
146                 total_bins_count = total_bins
147             total_bins = None
148
149         bin_sizes = defaultdict(int)
150         bins_full_count = 0
151         bins_full = set()
152
153         for line in iterator:
154             cls = classification_function(line)
155
156             # ignore classes that were not requested by
157     total_bins
158
159             if total_bins:
160                 if cls not in total_bins:
161                     continue
162
163             if bin_size:
164                 bin_sizes[cls] += 1
165                 if bin_sizes[cls] == bin_size:
166                     bins_full_count += 1
167                     bins_full.add(cls)
168
169     # ignore if the class bin is full 169 if bin_sizes[cls] <= bin_size:
170                 bins[cls].append(line)
171
172             # break if the total number of full bins is
173     total_bins_count

```

```

173             if total_bins_count and bins_full_count >= total_bins_count:
174                 return {k: v for k, v in bins.items()}
175
176                 else:
177                     bin_sizes[cls] += 1
178                     bins[cls].append(line)
179
180     # return the total_bins_count biggest bins 180             if total_bins_count:
181                 bins_to_be_returned = {k for k, v in
182                 sorted(bin_sizes.items()),
183                 key=itemgetter(1)}[
184                 -total_bins_count:]}
185                 return {k: v for k, v in bins.items()} if k in
186                 bins_to_be_returned}
187
188             return bins
189
190     class Distance:
191     def __init__(self, a, b, ids=None):
192         aux = list(Distance.merge(a, b)) 192             self.len = len(aux) 193             if self.len < 2:
193             raise ValueError("Too few keystrokes to evaluate")
194
195         )
196         (
197         self.n1,
198         self.m1,
199         self.v1,
200         self.n2,
201         self.m2,
202         self.v2,
203         ) = np.array(aux).T 203             self.kw = None 204             self.ids = ids
205
206         METRICS = dict(
207         a_metric=lambda x: np.heaviside(np.matmul(
208         np.sort(np.array([x.m1, x.m2])).T, axis=1),
209         [-x.kw.get('threshold', 1.25), 1] 210             ), 1).sum() / x.len,
210         braycurtis=lambda x: scd.braycurtis(x.m1, x.m2) / x.len,
211         canberra=lambda x: scd.canberra(x.m1, x.m2) / x.len,
212         chebyshev=lambda x: scd.chebyshev(x.m1, x.m2) / x.len
213
214         ,
215         cityblock=lambda x: scd.cityblock(x.m1, x.m2) / x.len
216
217         ,
218         correlation=lambda x: scd.correlation(x.m1, x.m2) / x
219         .len,
220         cosine=lambda x: scd.cosine(x.m1, x.m2) / x.len,
221         euclidean=lambda x: scd.euclidean(x.m1, x.m2) / x.len
222
223         ,
224         ncanberra=lambda x: scd.canberra(x.m1, x.m2, w=x.n) /
225         x.len,
226         r2=lambda x: 1 - skm.r2_score(x.m1, x.m2),
227         r_metric=lambda x: float(scd.cityblock(x.rank1, x
228         .rank2)) / (
229         x.len ** 2 >> 1),
230         ranked_canberra=lambda x: float(
231         scd.canberra(x.rank1, x.rank2, w=x.n) / (x.len
232         ** 2 >> 1),

```

```

224         seucclidean=lambda x: scd.seucclidean(x.m1, x.m2, V=x.v) / x.len,
225         squeueclidean=lambda x: scd.squeueclidean(x.m1, x.m2, w=x.w) / x.len,
226         wcityblock=lambda x: scd.cityblock(x.m1, x.m2, w=x.w) / x.len,
227         wsqueueclidean=lambda x: ((x.m1 - x.m2) ** 2 / (x.v +
1)).sum() / x.len
228     )
229
230     @staticmethod
231     def merge(a, b):
232         for k in set(a) & set(b):
233             yield a[k].stats + b[k].stats
234
235     @cached_property
236     def pickup_factor(self):
237         return self.n1.sum() * self.n2.sum()
238
239     @cached_property
240     def cross_pickup_factor(self):
241         return np.dot(self.n1, self.n2)
242
243     @cached_property 244
244     def w(self):
245         return np.sqrt(self.v)
246
247     @cached_property 248
248     def v(self):
249         return self.v1 + self.v2 + 1
250
251     @cached_property 252
252     def n(self):
253         return np.sqrt(self.n1 + self.n2)
254
255     @cached_property 256
256     def rank1(self):
257         return rankdata(self.m1, method='ordinal')
258
259     @cached_property 260
260     def rank2(self):
261         return rankdata(self.m2, method='ordinal')
262
263     def eval(self, metric, **kwargs):
264         self.kw = kwargs
265         if not callable(metric):
266             if metric not in Distance.METRICS:
267                 raise ValueError('distance not recognized and
not callable')
268             metric = Distance.METRICS[metric]
269             return metric(self)
270
271     def is_good(self, metric, threshold, **kwargs):
272         return self.eval(metric, **kwargs) < threshold
273
274
275     class Classifier(object):
276         """Contains a classification and evaluates metrics over it.
277

```

```

278     Examples
279
280     * Get EER for the Canberra distance
281     Classifier(classification).evaluate_metric('canberra ').eer
282
283     * Get Area Under ROC for the GP R metric
284     Classifier(classification).evaluate_metric('r_metric ').auc
285
286     * List noisy records id frequencies using the manhattan
287     distance
288     Classifier(classification).noise('cityblock', conjugate=True).id_count
289
290     """
291
292     def __init__(self, iterator):
293         self.values = list(iterator)
294
295     def __len__(self):
296         return len(self.values)
297
298     def evaluate_metric(self, metric, **kwargs):
299         return ClassificationResults(sorted_attempts(
300             (d.eval(metric, **kwargs), truth) for d, truth in
301             self.values))
302
303     def noise(self, metric, threshold=None, conjugate=False, **kwargs):
304         # if threshold is None we have to calculate it
305         if threshold is None:
306             threshold = self.evaluate_metric(metric, **kwargs)
307
308         ).eer[
309             'sensitivity']
310
311         return self.filter(
312             lambda d, t: d.is_good(metric, threshold, **kwargs) ^ conjugate ^ t)
313
314     def statistics(self, metric, threshold=None, **kwargs):
315         # if threshold is None
316         # we have to calculate it
317         if threshold is None:
318             threshold = self.evaluate_metric(metric, **kwargs)
319
320         ).eer[
321             'sensitivity']
322
323         aux = {
324             True: defaultdict(int),
325             False: defaultdict(int)
326         }
327
328         try:
329             for d, truth in self.values:
330                 index = d.is_good(metric, threshold, **kwargs) ^ truth
331                 (x, y) = d.ids
332                 aux[index][x] += 1
333                 aux[index][y] += 1
334             except TypeError:
335                 pass
336
337         for k in set(aux[True]) | set(aux[False]):
338             yield (k, aux[True][k],
339                   aux[False][k])
340
341 """

```

```

333         def filter(self, filter):
334             return Classifier(
335                 (d, truth) for d, truth in self.values if filter(
d, truth))
336
337     @cached_property
338     def id_count(self): 339         aux = defaultdict(int)
340             try:
341                 for d, truth in self.values:
342                     (x, y) = d.ids
343                     aux[x] += 1
344                     aux[y] += 1
345             except TypeError:
346                 return None
347             return aux.items()
348
349         def count(self, group_by):
350             aux = defaultdict(int)
351             for ids, count in self.id_count:
352                 cls = group_by(ids)
353                 aux[cls] += count
354
355         return aux.items()
356
357         def count_statistics(self, group_by, metric, threshold= None, **kwargs):
358             """Given a lambda group_by it returns a list of
classifications with a
359             tuple of grouped and a numpy array with two values:
the number of
360             failures, and the number of successes.
361
362             Example:
363             * For a composed tuple key, group by the second
value
364             c.count_statistics(itemgetter(1), 'r_metric')
365             """
366             aux = defaultdict(lambda: np.zeros(2))
367             for ids, f, t in self.statistics(metric, threshold,
**kwargs):
368                 cls = group_by(ids)
369                 aux[cls] += [f, t]
370
371             return aux.items()
372
373
374         def classify(bins, distance, prepare=None, filter=None):
375             prepare = prepare or (lambda cls, x: x)
376             filter = filter or (lambda x, y: False)
377
378             # prepare and filter the bins 379 aux = []
380                 for cls in bins.keys():
381                     aux.extend(
382                         (cls, prepared) for prepared in (prepare(cls, x)
for x in bins[cls])
383                         if prepared is not None)
384

```

```

385 # apply product aux x aux 386 for cls1, x1 in aux:
387     for cls2, x2 in aux:
388         if not filter(x1, x2):
389             yield distance(x1, x2), cls1 == cls2

```

Código B.2: Código para postprocesamiento de datos en Python

B.3 Framework en C

El *framework* optimizado permite mayor velocidad y soporta multiprocesamiento para maximizar la utilización de todas las CPU disponibles. Al igual que el módulo de Python, el contenido en C del *framework* optimizado es accesible desde la cuenta del Laboratorio de Sistemas de Información Avanzados en github.

Los códigos B.3, B.4 y B.5 muestran el núcleo de la plataforma.

```

1  #ifndef RESEARCH_CLASSIFIER_H
2  #define RESEARCH_CLASSIFIER_H
3
4  #include <stdint.h>
5  #include "include/distance.h"
6  #include "include/collector.h"
7
8      typedef struct {
9          distance_t *distances;
10         int size;
11         int original;
12     } classifier_t;
13
14         int classifier_init(classifier_t *self, const collector_t *
15         collector,
16         int (*distance_callback)(distance_t *,
17         const collector_node_t *, const
18         collector_node_t *, int),
19         int (*pair_filter)(const collector_node_t
20         *,
21         const collector_node_t *));
22
23 void classifier_destroy(classifier_t *self);
24
25         int classifier_filter_by_results(classifier_t
26         *self, classifier_t *aux,
27         real_t threshold,
28         ptr_metric_t metric,
29         int desired_truth);

```

```

26                                     int classifier_balance_results(classifier_t *self,
27                                     classifier_t *aux,
28                                     int diff_tolerance);
29 void classifier_export_statistics(const classifier_t *self);
30
31 #endif /* RESEARCH_CLASSIFIER_H */

```

Código B.3: Clasificador genérico de muestras para cálculo de
EER

```

1  #ifndef RESEARCH_COLLECTOR_H
2  #define RESEARCH_COLLECTOR_H
3
4  #include "db.h" /* sample_t */ #include
   "include/metrics.h"
5
6
7  typedef uint64_t classification_t;
8
9      typedef struct {
10         const sample_t *sample;
11         classification_t classification;
12         void *extra;
13     } collector_node_t;
14
15     typedef struct {
16         int size;
17         collector_node_t *nodes;
18     } collector_t;
19
20     void collector_init(
21         collector_t *collector, const sample_t *samples, int
   sample_size,
22         int (*pre_filter)(const sample_t *),
23         classification_t (*classify)(const sample_t *),
24         int (*post_filter)(const sample_t *, classification_t, void **)
   );
25
26
27 void collector_destroy(collector_t *collector, void free_extra(void *));
28
29                                     int collector_evaluate(collector_t collector,
30                                     int (*distance_callback)(distance_t *,
31                                     const
32                                     collector_node_t *, const
33                                     collector_node_t *, int),
34                                     int metric_count, ptr_metric_t *
   metrics, real_t *eers,
35                                     int *total_crosses, int (*pair_filter)
   (const
36                                     collector_node_t *, const
   collector_node_t *));

```



```
36 #endif /* RESEARCH_COLLECTOR_H */
```

Código B.4: Recolector genérico de muestras con soporte para
descarte y filtrado para cálculo posterior de EER

```
1  #ifndef RESEARCH_METRICS_H
2  #define RESEARCH_METRICS_H
3
4  #include "include/distance.h"
5
6  real_t dst_manhattan(const distance_t *distance);
7
8  real_t dst_ranked_nmanhattan(const distance_t *distance);
9
10 real_t dst_euclidean(const distance_t *distance);
11
12 real_t dst_veuclidean(const distance_t *distance);
13
14 real_t dst_neuclidean(const distance_t *distance);
15
16 real_t dst_canberra(const distance_t *distance);
17
18 real_t dst_ncanberra(const distance_t *distance);
19
20 real_t dst_ranked_ncanberra(const distance_t *distance);
21
22 real_t dst_a_metric(const distance_t *distance);
23
24 real_t dst_va_metric(const distance_t *distance);
25
26 real_t dst_na_metric(const distance_t *distance);
27
28 real_t dst_r_metric(const distance_t *distance);
29
30 real_t dst_a_r_metric(const distance_t *distance);
31
32 real_t dst_minkowski_norm_param(const distance_t *distance, real_t power);
33
34 real_t dst_minkowski_param(const distance_t *distance, real_t power);
35
36 real_t dst_minkowski_normalised_param(const distance_t *distance, real_t power);
37
38 real_t dst_minkowski_e(const distance_t *distance);
39
40 real_t dst_minkowski_40(const distance_t *distance);
41
42 real_t dst_chebyshev_norm(const distance_t *distance);
43
44 real_t dst_chebyshev(const distance_t *distance);
45
46 real_t dst_chebyshev_normalised(const distance_t *distance);
47
48     #define METRIC_NAMES "manhattan", "ranked_nmanhattan", " euclidean",
        "veuclidean", \
```

```

49     "neucleidean", "canberra", "ncanberra", "minkowski_e",
    "a_metric",\
50     "minkowski_40", "na_metric", "r_metric", "a_r_metric"
51
52     #define METRICS dst_manhattan, dst_ranked_nmanhattan, dst_euclidean,\
53     dst_veuclidean, dst_neucleidean, dst_canberra,
    dst_ncanberra,\
54     dst_minkowski_e, dst_a_metric, dst_minkowski_40,
    dst_na_metric,\
55     dst_r_metric, dst_a_r_metric
56
57 #define TOTAL_DISTANCES 13
58
59 real_t dst_a_metric_param(const distance_t *distance, real_t threshold);
60
61 #endif /* RESEARCH_METRICS_H */

```

Código B.5: Módulo de soporte de varias métricas

Formatos recolectados

Los datos capturados por el experimento realizado para esta tesis se encuentran disponibles desde el sitio de Github del laboratorio, para acceder a algunos datos confidenciales como el video con la cara del participante es necesario firmar un contrato de confidencialidad, estos datos no están publicados. Este apéndice detalla los formatos y resulta indispensable para descodificar los archivos pertenecientes al dataset. El cuadro C.1 muestra la estructura de archivos del mismo y describe los formatos.

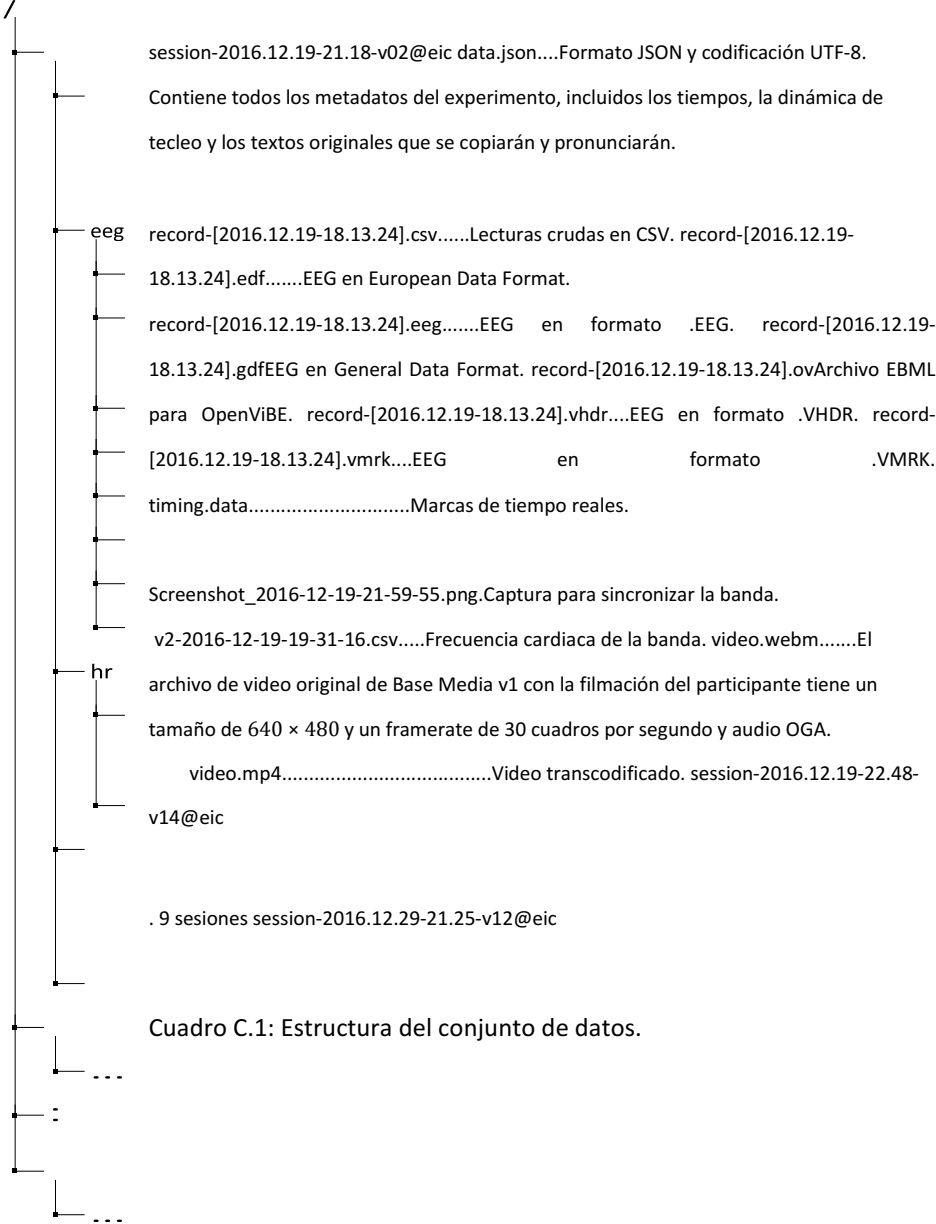
C.1 Lecturas crudas en formato de valores separados por coma

Un archivo de valores separados por comas (CSV) es un archivo de texto delimitado que usa comas para separar los valores. Los CSV almacenan datos tabulares (números y texto) en texto plano. Cada línea del archivo es un registro. Cada registro consta de uno o más campos, separados por comas. El uso de la coma como separador de campo da origen al nombre del formato. Por su simplicidad, el mismo es soportado por una enorme variedad de programas volviéndose un estándar de *de facto* para el intercambio de datos tabulares.

Es por la simplicidad de este formado que se decidió almacenar las lecturas crudas de los valores de EEG utilizando marcas de tiempo y electrodos como campos (columnas) y una lectura por cada fila.

C.2 Archivos JSON

La notación de objetos de JavaScript (JSON) es un formato de archivo abierto que utiliza texto legible por humanos para transmitir objetos que consisten en pares clave/valor y tipos de datos



Cuadro C.1: Estructura del conjunto de datos.

como arreglos, booleanos, números y texto; o cualquier otro tipo serializable en uno de estos. Es un formato de datos ubicuo que ganó popularidad como formato de intercambio asíncrono entre los clientes y servidores web y ahora, debido a su simplicidad, está siendo utilizado para muchas otras funciones.

En este formato hemos almacenado toda la información adquirida para una sesión que no se encuentra en otro lado. Esto es en el archivo `data.json` e incluye información personal: Nombre —en un objeto separado para realizar más fácilmente el enmascaramiento del archivo—, edad, género, idiomas que habla, enfermedades que padece; también incluye la cadencia de tecleo de lo tipeado en el formulario de información personal y todas las fases. Se incluyen marcas de tiempo del momento en que ocurrió cada evento, los códigos de imágenes mostradas y las frases que el participante debió tipear y pronunciar.

El archivo `timing.data` guarda un ajuste de sincronización para EEG y también se encuentra en este formato.

C.3 Cadencia de tecleo

Dentro del archivo JSON en formato de cadena de texto se encuentra la cadencia de tecleo. La misma se encuentra en un formato definido para esta tesis y que es utilizado por varios trabajos para adquirir y postprocesar cadencia de tecleo. Para esta tesis desarrollamos varios SDK para leer y escribir cadencia en JavaScript, Python, ANSI C y PHP; además de capturadores en ANSI C para Windows, GNU/Linux y Mac OS X.

El formato consta de la siguiente estructura:

$$\text{número} + \text{navegador} + n \times (\text{número}, \text{número})$$

Donde el primer número indica el tamaño en bytes del string navegador. Luego se lee un par de números por cada keystroke: código de evento y delta de tiempo. Los números tienen su propia forma de indicar la marca de finalización.

C.3.1 Formato de los números

Se parte de la siguiente cadena de 64 caracteres, la cual fue cuidadosamente armada para minimizar el tamaño de la cadencia:

```
0124689qwertyuiopsdfgASDFGHJKLZX35 ahjklzxcvbnmQWERTYUIOPCVBNM7#
```

Cada carácter representa un valor de 0 a 63 según su orden en la cadena. Los 32 primeros caracteres pertenecen al grupo «continuar» y los siguientes 32 al grupo «frenar». Para codificar un número simplemente hay que convertirlo a la base 32 y utilizar los caracteres del grupo «continuar» a excepción del último carácter que pertenece al grupo «frenar». De esta forma, el descodificador frenará al encontrar un carácter perteneciente a este grupo y emitirá el número. La implementación del mismo se encuentra en la línea 40 del código B.1.

Por ejemplo para convertir el número 1234. Primero se lo transforma a base 32. Sabemos que $1234 = 18 + 38 \times 32 = 18 + 6 \times 32 + 1 \times 32^2$ siendo (18,6,1) los valores a convertir. Buscando en la cadena de texto obtendremos que los primeros dos corresponden a d9 (tomados de «continuar») y el último a 5 (tomado de «frenar»), es decir el valor $32 + 1$ en vez de 1. Finalmente, el número 1234 se convierte en d95.

Por lo general, los números son bajos y la proporción de caracteres «frenar» y «continuar» no difiere en órdenes de magnitud y por la simplicidad del algoritmo resulta útil para almacenar cadencia.

C.3.2 Códigos de eventos

Los eventos son un valor entero obtenido a partir de los siguientes datos de último a primero:

1 bit press (0) o release (1), 8 bits *keycode* en formato JavaScript (algunos códigos especiales dependen del navegador y deben ser traducidos) y 3 bits (pero se reservan 3) ubicación en el teclado; generalmente es 0, pero en teclas como Shift ↑, o Alt se utiliza Ctrl 1 para izquierda y 2 para derecha. Los valores 3, 4 y 5 se reservan para el *numpad*, celulares y *joystick* respectivamente (los últimos dos obsoletos en Mozilla desde la versión Gecko 38).

Esta tesis utiliza solo las teclas pertenecientes a caracteres por lo que ignora los bits correspondientes a la ubicación (los más significativos) para reutilizar el espacio y almacenar eventos múltiples en un solo registro de la CPU.

C.3.3 Código completo

Cada código de evento viene acompañado por una diferencia (delta) entre ese evento y el anterior, dada en milisegundos. El primer evento contiene la cantidad de milisegundos desde el momento inicial de la muestra. Este momento no está definido y puede utilizarse el UNIX epoch, el momento del inicio del navegador (que es el timestamp por defecto en la implementación del código B.1) o simplemente utilizar cero e ignorarse. Utilizar deltas produce valores bajos que luego serán codificados con a lo sumo dos caracteres para latencias, salvo que las mismas superen los 512 ms (valores en el rango 0,512–32,768 segundos utilizan tres caracteres).

El siguiente ejemplo muestra la cadencia producida por el tipeo de la palabra Robustez generando

```
waMozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87  
Safari/537.3668055gkm1055uLH7904a52GceaG 5 KiaJ5s1 j50qF5oIA 5Asjt7JXR5pe8  
54kG055tojr710j5K2mFa1N5esjt7e 655yJlG7L#5gpaK58 #5XBML055Hojt  
7X#5Zkm40556XH7r0c5fj7
```

C.4 Video

El video del participante (tomado por la cámara de la PC que utilizaba para el experimento) se almacenó en dos formatos: MP4 y webm. Debido a que el *framework* de adquisición utilizaba web para capturar el formato soportado era webm (formato de Base Media v1) con una resolución de 640 × 480 y un *framerate* de 30 cuadros por segundo y pistas de audio en formato OGA de Vorbis. La versión en mp4 es una transcodificación posterior.

C.5 Frecuencia cardíaca

Para la frecuencia cardíaca se almacenan dos archivos, uno en CSV con las muestras y una captura de pantalla en formato PNG que contiene la información necesaria para sincronizar la banda lectora.

C.6 Archivos de EEG

Los registros de EEG se encuentran en varios formatos para evitar problemas de compatibilidad, los mismos son CSV para las lecturas crudas (ver sección C.1), European Data Format, formato.EEG (definido en Bourke, 1997), General Data Format (definido en Schlögl, 2006), EBML para OpenViBE y los formatos de BrainVision Analyzer: VHDR y VMRK para encabezados y marcas respectivamente.

Cámara de inmersión emocional

Para estudiar las correlaciones entre las diferentes modalidades de adquisición, creamos un conjunto de datos con 12 participantes voluntarios y grabamos audio, video, ritmo cardíaco, dinámica de trcleo y actividad cerebral. Para registrar diferentes emociones, las generamos utilizando estímulos visuales y sonoros. Saber cómo cada modalidad corresponde a otra ayudará a predecir las emociones contando con otras modalidades o permitirá analizar las correlaciones entre estas. La configuración del experimento, así como el desarrollo de la cámara de inmersión emocional, se encuentran detallados en Calot y Ierache (2017); Rossi *et al.* (2017).

D.1 Estimulación

Cada sesión duró aproximadamente una hora, aunque el tiempo exacto depende de cada participante. El primer paso fue llevar a los sujetos a un estado inicial similar considerado neutral y libre de cualquier estímulo. Este paso consistió en un medio minuto de ingreso de datos, un minuto de relajación con silencio, cuatro minutos y medio con un juego y tres minutos para obtener respuestas de su estado (considerado neutral o inicial). El juego consistió en usar el mouse para atrapar ciertos puntos de colores en la pantalla mientras se reproduce una relajante música de fondo. Luego, las cuatro fases comenzaron en el orden: felicidad, calma, tristeza y enojo que duraron aproximadamente trece minutos cada una. El experimento barre los cuatro cuadrantes del modelo de excitación– valencia (Bradley y Lang, 1994) (ver figura 2.4b) en una trayectoria en sentido horario que se mueve sobre cada fase en la que se aplica la misma secuencia estimuladora. Combinaron cinco minutos de estimulación visual (imágenes), cinco minutos de audiovisuales (videos y audio) y alrededor

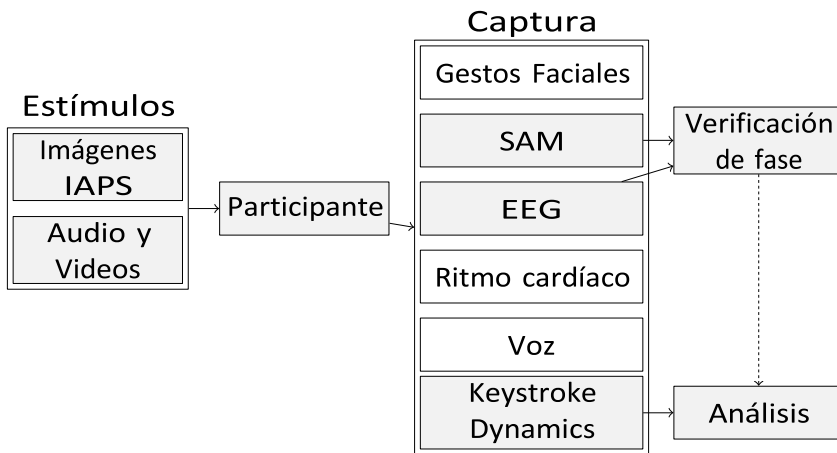


Figura D.1: Para cada fase, se envían estímulos específicos al sujeto de prueba quién proporcionará capturas multimodales. Este trabajo solo utilizará SAM y EEG para validar que el sujeto de la prueba alcanzó la fase correcta y luego utilizará los datos de la cadencia de tecleo para un análisis más detallado; en color grisado se indican los componentes utilizados durante esta tesis.

de otros tres para las respuestas al estado actual (autoinforme, voz y escritura).

Para realizar la estimulación visual, solicitamos un conjunto de datos compilados por Lang *et al.* (2008) llamado IAPS, una base de datos de imágenes utilizada en diversos estudios científicos relacionados con las emociones. Se considera un estándar de facto para la estimulación afectiva con imágenes. También hemos descargado tres videos y un audio con un alto contenido emocional. En primer lugar, se muestra un subconjunto de imágenes tomadas de IAPS (100 imágenes, 25 por fase) y luego un video o audio, cuyo tema está relacionado con la fase afectiva transitoria. Para cada fase, una vez completada la imagen, video y estimulación de audio, se le pide al sujeto que complete un autoinforme usando la encuesta de autoevaluación SAM propuesta por Lang (1980) (ver la sección D.2.3). Luego, procedemos a capturar dinámicas de voz y pulsaciones de teclas. Ver la figura D.1 para un diagrama de la arquitectura del experimento.

D.2 Adquisición de datos y modalidades utilizadas por esta tesis

Se han utilizado varias interfaces de entrada para crear este conjunto de datos. Las siguientes secciones resumen las características principales de cada modalidad.

D.2.1 Encefalografía (EEG)

Una interfaz cerebro-máquina (BMI) facilita la comunicación entre las funciones mentales o cognitivas creadas a partir del cerebro de una persona, al capturar señales eléctricas, para ser procesadas, clasificadas y comunicadas con aplicaciones o dispositivos específicos (Ierache *et al.*, 2013a,b, 2015). Este conjunto de datos utilizó el neuroauricular Emotiv EPOC para la adquisición de señales encefalográficas por medio de 14 electrodos, que inicialmente se consideraron los que se encuentran en la corteza prefrontal (F3, F4, AF3, AF4, según el sistema internacional 10-20 [Jasper, 1958], ver figura 2.6). Se empleó OpenViBE para recibir y almacenar las muestras sin procesar y luego aplicar filtros a los electrodos y realizar el procesamiento posterior. Esto se puede usar para calcular la posición del estado emocional en el modelo de valencia de activación bidimensional (Feldman, 1995; Oude Bos, 2006; Ramirez y Vamvakousis, 2012).

La figura D.2 muestra un ejemplo de un electrodo obteniendo ondas α y marca las distintas fases. El area grisada corresponde a la captura de patrones de tecleo.

D.2.2 Keystroke dynamics

La dinámica de tecleo se adquiere después de la estimulación, solicitando al participante que ingrese a una lista de oraciones de una carga emocional neutral. Esos pasajes se mezclan aleatoriamente para cada una de las cuatro fases. Tanto el FT como el WT se almacenan para reconstruir todos los eventos del teclado.

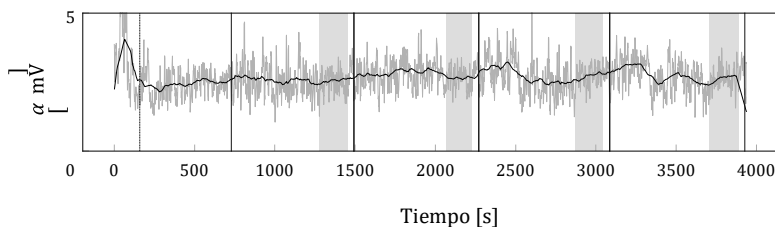


Figura D.2: Línea temporal de captura para el caso de un participante. Se observa la onda alfa para el electrodo AF3 del participante 6. El area grisada corresponde a la captura de patrones de teclado.

D.2.3 Autoinforme (SAM)

Se realiza una encuesta de último minuto preguntando al participante sobre su estado emocional actual. Hace uso de la metodología ampliamente adoptada para capturar emociones, llamada SAM. Como lo describen Bradley y Lang (1994), SAM es un método no verbal fácil de administrar para evaluar rápidamente la valencia, la excitación y el dominio asociados con la reacción emocional de una persona ante un evento. Hemos omitido una dimensión de dominio para simplificar el experimento: no se tomó información con respecto al dominio; solamente se capturó valencia y excitación. Consta de 5 dibujos que caracterizan si la persona está feliz o triste (valencia) y qué tan excitada (excitación). Para cada dominio, hay 9 puntos, uno por figura y un punto intermedio entre ellos.

La figura D.3 presenta la versión en papel y lápiz de SAM, que ilustra su representación no verbal, con figuras de varios puntos a lo largo de cada una de las dos dimensiones. SAM se extiende desde una figura sonriente y feliz a una figura arrugada e infeliz cuando se representa la dimensión de valencia y se extiende desde una figura emocionada con los ojos abiertos hasta una figura relajada y adormecida para la dimensión de excitación (Bradley y Lang, 1994).

Para nuestro experimento se utilizó SAM con puntos intermedios, es decir, nueve puntos en total arrojando valores de Excitación– Valencia en el rango 1–9. Los mismos no necesitan ser transformados.

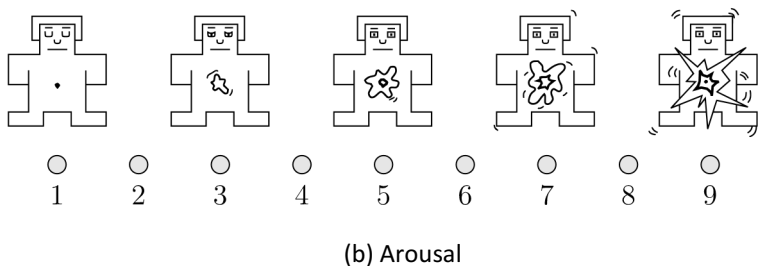
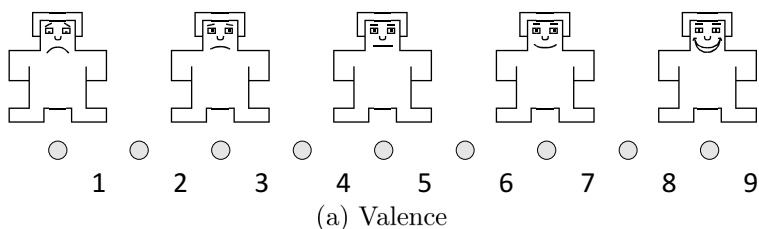


Figura D.3: SAM y sus diferentes estados a lo largo del eje de dimensión.

D.3 Adquisición de datos y modalidades no utilizadas por esta tesis

Si bien estas modalidades fueron adquiridas, no fueron utilizadas durante el diseño o análisis de datos por la presente tesis.

D.3.1 Ritmo cardíaco

La frecuencia cardíaca se adquirió a través de una pulsera colocada en el participante (*Xiaomi Mi Band 1s*), que registra permanentemente la frecuencia cardíaca. Es una banda de policarbonato con un sensor óptico de frecuencia cardíaca que pesa 14,5g.

D.3.2 Video

La captura de video también se realiza en todo momento. Fueron comparadas las cinco herramientas más conocidas. La comparación, se centró en las características necesarias para el diseño de un eventual experimento y no en el costo —los modelos de *pricing* son muy complejos

y escapan al objetivo de esta comparación—, la tasa de éxito —en todos los casos resultó ser funcional— u otros factores.

Las herramientas encontradas son:

Clmtrackr software libre muy documentado. Fácil de extender a nuevas emociones entrenando con cualquier bases de datos: Contiene tutoriales para hacerlo⁸. Utiliza una implementación de modelos locales restringidos ajustados por puntos de referencia regularizados (*constrained local models fitted by regularized landmark mean-shift*) desarrollado por Audun Mathias Øygaard y basado en el artículo de Saragih *et al.* (2011). Para entrenar los puntos de referencia, Øygaard utilizó la base de datos MUCTMilborrow *et al.* (2010) agregando algunas imágenes propias. El modelo emocional, lo provee con 18 coeficientes. Además, no incorporó en su código dos emociones que sí tiene desarrolladas: miedo (*fear*) y disgusto (*disgust*).

EmoVu el tercer software comercial analizado. Es necesario obtener el producto para obtener la documentación, aunque la misma es extensa y contiene ejemplos. El soporte JavaScript del mismo requiere desarrollar una interfaz⁹.

FaceReader de Noldus es otro software comercial muy avanzado, pero que requiere una instalación del mismo en la máquina anfitrión y del sistema operativo Windows para correr, es utilizado por 300 sitios alrededor del mundo¹⁰ y afirma lograr detecciones del 95,9% (Langner *et al.*, 2010).

⁸ Es posible descargarlo del repositorio de Github en <https://github.com/auduno/clmtrackr>, contiene una versión demo disponible en línea.

⁹ Información sobre la API se encuentra disponible en <http://emovu.com/e/developers/api/>.

¹⁰ Información tomada de <http://www.noldus.com/facereader/facereader-api> el día 13 de junio de 2019.

Kairos un software comercial cuyo soporte de detección de emociones se encuentra en modo beta¹¹.

OpenCV es una colección de software libre utilizada para *computer vision*, que permite detectar rostros y puntos. Es la base para desarrollar la herramienta, la cual con toda la información que hay disponible no sería muy difícil de hacer¹² (una grosera estimación, basada en la documentación disponible, sería de 50 horas hombre). Aún no está completamente portado a *JavaScript*. Fácil de entrenar con cualquier base de datos de emociones.

El resultado de la comparación se compaginó en el cuadro D.1. Cabe destacar que provee un *Software Development Kit* (SDK) instalable localmente para poder desarrollar interfaces locales entre esa aplicación (generalmente compilada e inmodificable) y otro programa a desarrollar, mientras que API Cloud significa que permite desarrollar una interfaz, pero que la misma es con un sistema que corre remotamente en la nube. Ambas opciones son producto de modelos de negocio comerciales que no quieren proveer el código fuente (en el primer caso se provee de un ejecutable binario y en el segundo de un servicio en la nube) al usuario pero sí desean proveer entornos de desarrollo y conexión con la herramienta. El caso ideal, es un software libre que contenga el código directamente en *JavaScript*.

	Kairos	NoldusFaceReader	EmoVu	OpenCV	clmtrackr
--	--------	------------------	-------	--------	-----------

¹¹ La documentación se encuentra disponible en <https://www.kairos.com/docs/face-recognition/>.

¹² Existe un tutorial para reconocimiento facial en http://docs.opencv.org/modules/contrib/doc/face_recog/doc/face_recog_tutorial.html.

SDK Local		X	X	X	X
API Cloud	X		X		
Código Abierto				X	X
Nivel de Documentación (0 a 10)	4	5	8	10	9
2D to 3D Modeling	X	X	X		
Soporta JavaScript	X				X
Formatos de Captura					

WebCam (captura de video)	†	X	X	X	X
Video	X	X	X	X	X
avi	X	X	X	X	X
flv			X	X	
mp4		X	X	X	X
mov		X	X	X	
Imagen	X	X	X	X	†
jpg	X	X	X	X	X
bmp		X	X	X	X
png		X	X	X	X
Emociones ‡					

anger	X	X	X		X
contempt		X			
disgust		X	X		

† Posible con programación extra (pocas horas de trabajo)

‡ Se conserva el vocabulario en inglés para evitar diferencias de traducción

Cuadro D.1: Comparación de características de distintas herramientas (continuado).

	Kairos	NoldusFaceReader	EmoVu	OpenCV	clmtrackr
dislike	X				
drowsiness	X				
fear			X		
frowns	X				
happy		X			X
joy			X		
neutral		X	X		
sadness		X	X		X
scared		X			
smiles	X				
surprise	X	X	X		X
Extensible a más emociones				X	X
Total de emociones	6	8	7	0	4
Posibilidad de embeber en browser localmente					X
Posibilidad de embeber en browser remotamente	X		X	X	X

† Posible con programación extra (pocas horas de trabajo)

‡ Se conserva el vocabulario en inglés para evitar diferencias de traducción

Cuadro D.1: Comparación de características de distintas herramientas.

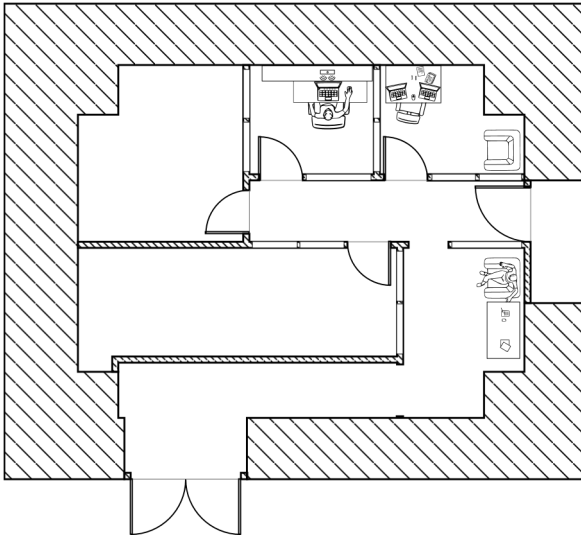
D.3.3 Audio

Todas las pistas de audio se conservan en estéreo para su posterior análisis. Se requiere que el participante lea un texto que es una mezcla diferente de las oraciones correspondientes empleadas para la dinámica de tecleo.

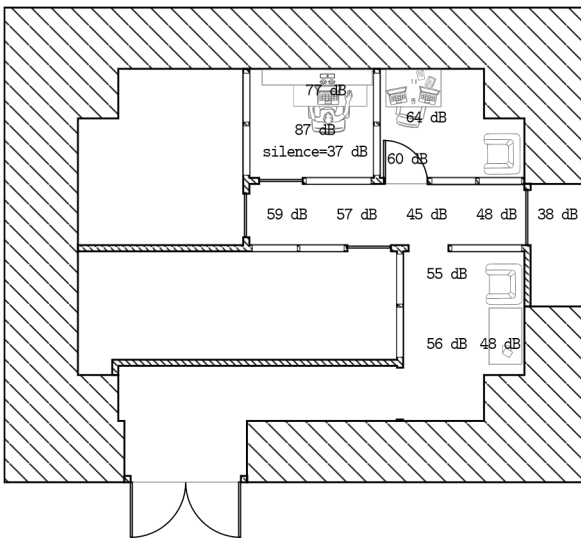
D.4 Descripción del entorno

En total, 12 personas realizaron las cuatro fases más dos investigadores que realizaron una prueba de captura inicial. Las capturas se realizaron en diciembre de 2016 durante el período de dos semanas. La construcción del espacio, la programación de la interfaz de prueba y la configuración de los dispositivos tomaron aproximadamente seis meses de preparación.

Para evitar distracciones ambientales, la adquisición se realizó en una sala acondicionada dentro de las instalaciones del Laboratorio de Sistemas de Información Avanzados de la Facultad de Ingeniería de la UBA. Este espacio se denominó Cámara de inmersión emocional (EIC) y consta de aproximadamente 4m² (ver Figura D.4a), se comunica con una sala de control adyacente donde los investigadores supervisan permanentemente conectividad y captura. También tiene una sala de espera donde los participantes se quedan mientras los investigadores preparan el equipo y comienzan el experimento. La sala está aislada tanto en sonido como en visibilidad, las ventanas están cubiertas con paneles verdes para facilitar la extracción de imágenes del video. Se realizaron pruebas de sonido para medir el nivel de ruido y distracción en el espacio como se muestra en la figura D.4b. El nivel de ruido base en la cámara fue de 37 decibelios. Se emitió un sonido de 87 decibeles en el altavoz y se midió cuanto decayó para cada punto, para así garantizar el aislamiento sonoro. Debido a la restricción de los tiempos no se llegó a instalar paneles acústicos, sin embargo, dada la hora en que fueron realizadas las capturas, no se oyeron ruidos durante las mismas.



(a) Plano de la EIC.



(b) Insonorización de la EIC.

Figura D.4: Detalles físicos de la EIC.

Revisión sistematizada de la literatura

La revisión sistemática de la literatura es una metodología para realizar revisiones bibliográficas de manera formal, siguiendo pasos bien definidos que permiten que otros investigadores puedan reproducir los resultados. Este método de revisión se usa comúnmente en otras áreas, principalmente en medicina y tiene varios beneficios reportados, como una menor susceptibilidad al sesgo. En el área de las ciencias informáticas, este método de revisión se difunde más en ingeniería del software (Concilio, 2019).

La aplicación de la revisión sistemática implica tres etapas principales: *planificación, conducción y presentación* de los resultados. En la primera etapa se define un protocolo de revisión, en el que se especifican las consultas de investigación junto con las estrategias de búsqueda. En la segunda etapa, se aplica el protocolo de revisión definido en la etapa anterior y se extrae la información de las referencias obtenidas. Finalmente, en la tercera etapa se define la forma de presentar los resultados y se realiza el informe final (Concilio, 2019).

E.1 Objetivos de la revisión

En esta tesis actualizaremos lo ya realizado en Concilio (2019) y aplicaremos la revisión sistemática con el objetivo de estudiar el estado del arte en cadencia de teclado para identificar:

- Autores, revistas y publicaciones clave en el campo
- *Features* más comúnmente utilizados
- Clasificadores estudiados
- Métricas de evaluación comúnmente adoptadas

- Conjuntos de datos de evaluación públicos —*Benchmarking datasets*—, que son útiles para realizar experimentos comparativos en el área.
- Metodología utilizada
- Áreas de oportunidad

E.2 Expresiones de búsqueda utilizadas

Después de definir los objetivos de la investigación, enumeramos una lista de términos relacionados con los trabajos científicos publicados que podrían ayudar a responder estos puntos: *keystroke dynamics*, *typing dynamics*, *keystroke biometric(s)*, *keystroke authentication*, *keystroke identification*, *keystroke pattern(s)*, *typing pattern(s)*, *behaviour intrusion detection*, *biometric intrusion detection*, *user profiling*, *behavioural biometrics*, *continuous authentication*, *typing biometric(s)*, *keypress biometric(s)*, *keystroke analysis*, *free-text analysis*, *fixed text analysis*.

El uso de varios términos para el mismo tema, a veces incluso sinónimos, contribuye a la integridad de la búsqueda. A partir de esta lista de términos, creamos expresiones de búsqueda para cada base de datos de consultada. La expresión de búsqueda básica es la conjunción de cada término en la lista mediante el OR lógico conectivo.

Sin embargo, después de algunas pruebas con esta expresión de búsqueda, observamos que muchas de las referencias devueltas trataban temas no relacionados con la pregunta de investigación, como sistemas de recomendación o trabajos basados en autenticación por comportamiento en teléfonos móviles —o smartphones—. Por este motivo, se identificaron algunos términos que podrían excluir estos temas no relacionados: *sistema de recomendaciones*, *sistema de recomendaciones*, *nanogenerators*. La expresión de búsqueda básica se modificó para considerar los términos de exclusión con el uso de la lógica conectiva AND NOT, de la siguiente manera:

```
(
"keystroke dynamics" OR
"typing dynamics" OR
"keystroke biometric(s)" OR
"keystroke authentication" OR
"keystroke identification" OR
"keystroke pattern(s)" OR
```

"typing pattern(s)" OR
"behaviour intrusion detection" OR
"biometric intrusion detection" OR
"user profiling" OR
"behavioural biometrics" OR
"continuous authentication" OR
"typing biometric(s)" OR
"keypress biometric(s)" OR
"keystroke analysis" OR
"free-text analysis" OR
"fixed text analysis"
) AND NOT (
"nanogenerators" OR
"recommendation system" OR
"android"
)

E.3 Bases de datos consultadas

Para consultar las distintas bases de datos más importantes en el área de computación se ha utilizado el software *Publish or Perish 6*¹³ que permite automatizar el proceso de obtención de listas de referencias a partir de expresiones de búsquedas definidas. Además, permite de forma simple e intuitiva la exportación de las listas para su posterior análisis y generación de métricas (Harzing, 2007).

Las siguientes bases de datos se consideraron en ambos trabajos (la presente tesis y Concilio (2019):

Google Scholar Google Scholar es un motor de búsqueda gratuito que indexa el texto completo o los metadatos de la literatura académica en varios formatos y disciplinas de publicaciones. Lanzado en versión beta en noviembre de 2004, el índice de Google Scholar incluye revistas y revistas académicas revisadas por expertos, ponencias de conferencias, tesis y disertaciones,

¹³ Al 25 de febrero de 2019 se encontraba disponible para su descarga gratuita en <https://harzing.com/resources/publish-or-perish/os-x>.

preimpresos, resúmenes, informes técnicos y otra literatura académica, incluidas opiniones judiciales y patentes¹⁴.

Microsoft Academic Microsoft Academic es un motor de búsqueda gratuito para publicaciones científicas y literatura desarrollada por Microsoft Research. Relanzada en 2016, la herramienta presenta una estructura de datos y un motor de búsqueda completamente nuevos que utilizan tecnología de búsqueda semántica. Actualmente, más de 375 millones de entradas están indexadas, de las cuales, 170 millones son trabajos académicos¹⁵.

Web of Science Web of Science es un servicio de información científica en línea proporcionado por Thomson Reuters, integrado en ISI Web of Knowledge, WoK⁴.

Scopus Scopus es una base de datos bibliográfica de resúmenes y enlaces a artículos de revistas científicas. Cubre alrededor de 18.000 títulos de más de 5000 editores internacionales, incluida la cobertura de 16.500 revistas revisadas por pares en los campos de la ciencia, tecnología, medicina y ciencias sociales, incluidas las artes y las humanidades. Es editado por Elsevier y está disponible en línea para los suscriptores. Las búsquedas en Scopus incluyen búsquedas en sitios científicos a través de Scirus, incluso de Elsevier, y bases de datos de patentes¹⁶.

E.4 Criterios de selección

En esta revisión sistemática, todas las referencias devueltas se incluyen para su análisis en los siguientes pasos, excepto los que cumplen con los siguientes criterios de exclusión:

- Publicaciones que no tratan temas relacionados con modelado de cadencia de tecleo:
- Publicaciones con una página, pósters, presentaciones, extractos y editoriales, textos en revistas / periódicos y publicaciones duplicadas en términos de resultados, excepto la versión más completa: referencias sin

¹⁴ Disponible en <https://scholar.google.com.ar/>.

¹⁵ Disponible en <https://academic.microsoft.com/>. ⁴Disponible en <https://webofknowledge.com/>.

¹⁶ Disponible en <https://www.scopus.com/home.uri>.

información suficiente para responder a la pregunta de investigación. Este criterio también evita el trabajo innecesario para los casos en que el mismo estudio se publica en diferentes versiones.

E.5 Resultados obtenidos

El primer paso fue aplicar las expresiones de búsqueda en cada base de datos de referencias y guardar los resultados devueltos. Además de las referencias devueltas, también incluimos las referencias previamente conocidas por los autores, pero no indexadas por las bases de datos utilizadas en esta revisión.

Fueron analizadas 657 referencias relacionadas a *keystroke dynamics* en el período entre enero de 2005 a febrero de 2019. Los resultados encontrados por Concilio *et al.* (2018) se mantuvieron vigentes: la cantidad de nuevas publicaciones se mantuvo entre 20 y 25 durante los últimos 7 años. Antes de esto era mucho menor. Es interesante ver que la mayoría de los estudios trabajan con la metodología de texto fijo. Los estudios de texto libre tuvieron un auge entre 2005 y 2010, pero siempre mantuvieron la minoría. La Figura E.1 muestra las publicaciones obtenidas por año de publicación.

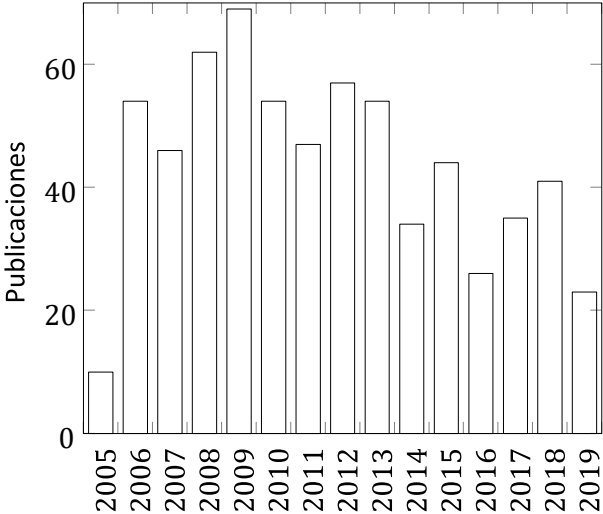


Figura E.1: Publicaciones desde 2005.

E.6 Conjuntos de datos públicos

El rendimiento en *keystroke dynamics* depende en gran medida del conjunto de datos utilizado. Es por esto que la identificación de los conjuntos de datos de referencia *Benchmarking datasets* resulta fundamental. Además, el uso de conjuntos de datos fácilmente disponibles nos permite comparar los algoritmos rápidamente. A continuación se listan los distintos conjuntos de datos públicos de acceso público más relevantes que se conocen al día de la fecha:

GREYC 133 usuarios escribieron el texto “greyc labora tory” en dos teclados diferentes, en los que 100 de los usuarios proporcionaron muestras en al menos 5 sesiones. Las muestras fueron recolectadas en un período de 2 meses. De Giot *et al.* (2009)¹⁷.

Web-GREYC 118 usuarios teclearon las contraseñas y contraseñas durante un año. De Giot *et al.* (2009)¹⁸.

Stonybrook Keystroke Patterns as Prosody in Digital Writings Ya descrita en la Sección 5.1.2 como Prosody. De Banerjee *et al.* (2014)¹⁹.

CMU 51 usuarios escribieron el texto .tie5Ronald en ocho sesiones. De Killourhy y Maxion (2010)²⁰.

CMU-2 20 usuarios proporcionaron datos de pulsaciones para texto libre y texto transcripto. De Killourhy y Maxion (2012)²¹.

¹⁷ Estaba disponible en <http://www.epaymentbiometrics.ensicaen.fr/greyc-keystroke-dataset> a diciembre de 2018.

¹⁸ Estaba disponible en <http://www.epaymentbiometrics.ensicaen.fr/greyc-web-based-keystroke-dynamics-dataset> a diciembre de 2018. Una copia espejo se encuentra en <http://www.labri.fr/perso/rgiot/ressources/GREYC-WebDataset.html>.

¹⁹ Disponible en <http://www3.cs.stonybrook.edu/~rbanerjee/project-pages/keystrokes/keystrokes.html>.

²⁰ Disponible en <http://www.cs.cmu.edu/~keystroke/>.

²¹ Disponible en <http://www.cs.cmu.edu/~keystroke/laser-2012/>.

Person Identification Challenge 12 usuarios. En cada una de las sesiones de escritura, se pidió a los usuarios que escribieran el mismo texto breve de unas pocas frases. De Buza (2016)²².

136 Million Keystrokes 168.000 usuarios. Se pidió a los usuarios que escribieran una serie de frases iguales para cada uno. Contiene aproximadamente 800 keystrokes por usuario. Mencionada en la Sección 5.1. De Dhakal *et al.* (2018)²³.

²² Disponible en <http://biointelligence.hu/typing-challenge/>.

²³ Disponible en <https://userinterfaces.aalto.fi/136Mkeystrokes/>.

Valores obtenidos por EEG y SAM

El presente apéndice muestra los resultados obtenidos del electroencefalograma de cada uno de los 12 participantes a excepción del número 6, que se encuentra graficado como ejemplo en el Capítulo 6. Presenta los valores de los electrodos utilizados y el resultado al aplicarle el algoritmo de Ramirez y Vamvakousis (2012) (Figuras 5.4, F.2, F.4, F.6, F.8, F.10, F.12, F.14, F.16, F.18, F.20 y F.22). Además muestra un cuadro por cada participante con el resultado de la prueba U de Mann–Witney (cuadros 5.3, F.1, F.3, F.5, F.7, F.9, F.11, F.13, F.15, F.17, F.19 y F.21). Las figuras 5.3, F.1, F.3, F.5, F.7, F.9, F.11, F.13, F.15, F.17, F.19 y F.21 representan el camino que el participante recorrió durante la captura de cadencia, incluyendo media y dispersión. Nótese el rápido corrimiento existente, este es el motivo principal por el cual se optó por reducir el tamaño de la captura. Las Cuadros 5.4, F.2, F.4, F.6, F.8, F.10, F.12, F.14, F.16, F.18, F.20 y F.22 suman todos los valores obtenidos por participante y las diferencias entre fases consecutivas.

		Arousal				Valence			
Estadístico	Fase	1	2	3	4	1	2	3	4
	1	12482	9197	11160	10083	12482	4956	11070	11924
	2	9197	12640	7327	11699	4956	12640	19	2696
	3	11160	7327	11250	9534	11070	19	11250	11936
	4	10083	11699	9534	13944	11924	2696	11936	13944

		Arousal				Valence			
p-value	Fase	1	2	3	4	1	2	3	4
	1	1.000	0.000	0.377	0.000	1.000	0.000	0.318	0.134
	2	0.000	1.000	0.000	0.064	0.000	1.000	0.000	0.000
	3	0.377	0.000	0.999	0.000	0.318	0.000	0.999	0.470
	4	0.000	0.064	0.000	1.000	0.134	0.000	0.470	1.000

		Arousal				Valence			
H ₀ aceptada	Fase	1	2	3	4	1	2	3	4
	1	A	NA	NA	NA	A	NA	NA	NA
	2	NA	A	NA	NA	NA	A	NA	NA
	3	NA	NA	A	NA	NA	NA	A	NA
	4	NA	NA	NA	A	NA	NA	NA	A

Cuadro F.1: Resultados de cambios de fase para el participante número 1.

Fase	SAM Img.		SAM Video		EEG				
	A	V	A	V	A	V	σ_A^2	σ_V^2	σ_{AV}
1	6	9	7	9	5.33	6.60	0.45	0.40	-0.35
Δ	-4	0	-5	-2	0.04	0.61	$\ \Delta\ = 0,61$		
2	3	9	2	7	5.37	7.21	0.09	0.03	-0.02
Δ	5	-6	6	-6	-0.16	-0.57	$\ \Delta\ = 0,59$		
3	7	1	8	1	5.22	6.64	0.13	0.01	-0.01
Δ	1	1	1	0	0.57	-0.31	$\ \Delta\ = 0,65$		
4	9	2	9	1	5.79	6.32	1.03	0.67	-0.66

Cuadro F.2: Diferencias numéricas entre los valores Arousal– Valence adquiridos mediante SAM luego de imágenes, SAM luego de videos (o audio) y EEG para el participante número 1. En el caso de EEG es un valor medio y se muestra la varianza en cada eje y su covarianza. Los valores intermedios marcados como Δ muestran las diferencias; es importante señalar que la diferencia en SAM luego imágenes está calculada contra SAM luego del video de la fase previa.

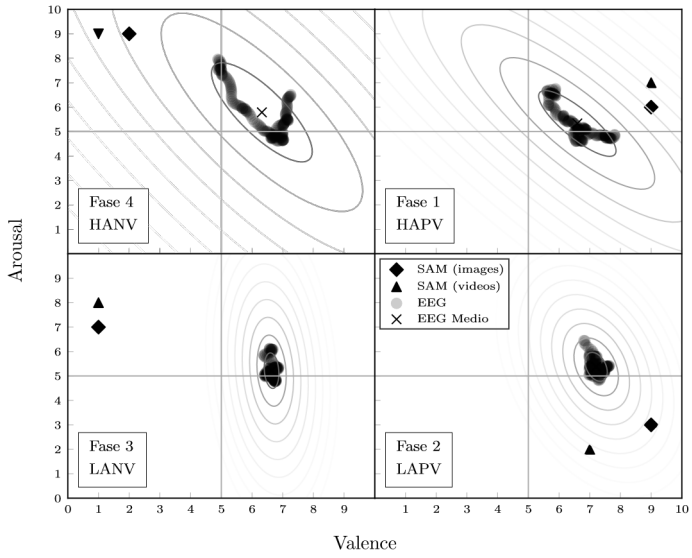
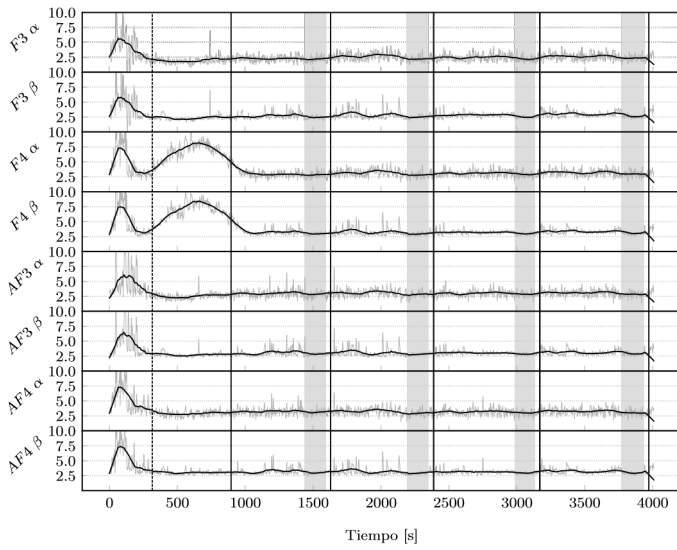
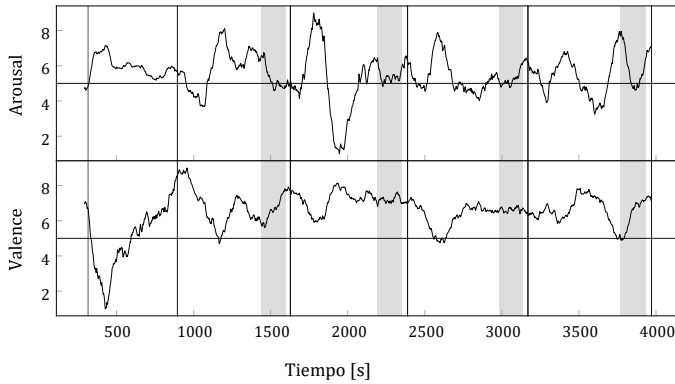


Figura F.1: Movimientos de Arousal y Valence del participante número 1 durante la captura de teclado.



(a) Registros de EEG del participante número 1.



(b) Arousal y Valence del participante número 1 según algoritmo de Ramirez. La franja grisada representa la captura de teclado.

Figura F.2: Registros obtenidos del participante número 1.

		Arousal				Valence			
Estadístico	Fase	1	2	3	4	1	2	3	4
	1	8192	491	4012	4318	8192	78	0	0
	2	491	10082	4552	752	78	10082	4023	2066
	3	4012	4552	7938	4789	0	4023	7938	3240
	4	4318	752	4789	6498	0	2066	3240	6498

		Arousal				Valence			
p value	Fase	1	2	3	4	1	2	3	4
	1	0.999	0.000	0.000	0.000	0.999	0.000	0.000	0.000
	2	0.000	0.999	0.000	0.000	0.000	0.999	0.000	0.000
	3	0.000	0.000	0.999	0.000	0.000	0.000	0.999	0.000
	4	0.000	0.000	0.000	0.999	0.000	0.000	0.000	0.999

		Arousal				Valence			
H ₀ aceptada	Fase	1	2	3	4	1	2	3	4
	1	A	NA	NA	NA	A	NA	NA	NA
	2	NA	A	NA	NA	NA	A	NA	NA
	3	NA	NA	A	NA	NA	NA	A	NA
	4	NA	NA	NA	A	NA	NA	NA	A

Cuadro F.3: Resultados de cambios de fase para el participante número 2.

Fase	SAM Img.		SAM Video		EEG				
	A	V	A	V	A	V	σ_A^2	σ_V^2	σ_{AV}
1	6	7	6	6	6.66	5.45	0.19	0.23	-0.16
Δ	-1	0	-2	0	-1.40	1.52	$\ \Delta\ = 2,07$		
2	5	6	4	6	5.26	6.97	0.46	0.24	0.23
Δ	1	-1	2	-2	0.78	0.45	$\ \Delta\ = 0,90$		
3	5	5	6	4	6.04	7.42	0.41	0.07	-0.07
Δ	-1	2	1	2	0.32	0.24	$\ \Delta\ = 0,40$		
4	5	6	7	6	6.36	7.66	0.05	0.07	0.02

Cuadro F.4: Diferencias numéricas entre los valores Arousal– Valence adquiridos mediante SAM luego de imágenes, SAM luego de videos (o audio) y EEG para el participante número 2. En el caso de EEG es un valor medio y se muestra la varianza en cada eje y su covarianza. Los valores intermedios marcados como Δ muestran las diferencias; es importante señalar que la diferencia en SAM luego imágenes está calculada contra SAM luego del video de la fase previa.

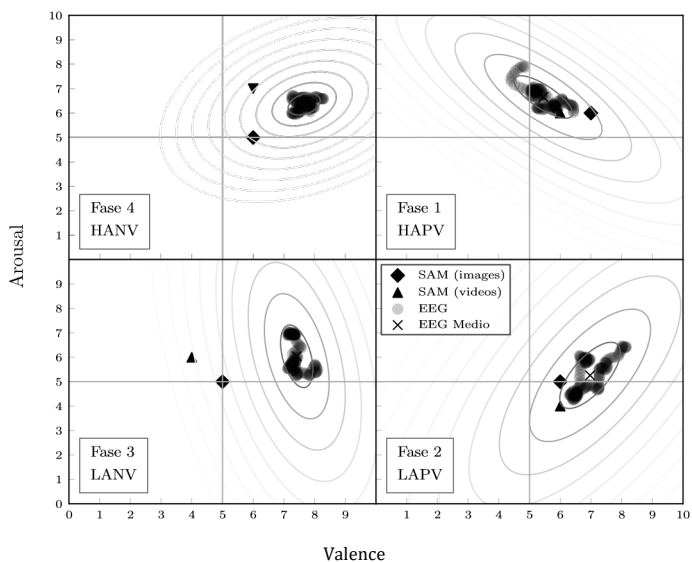
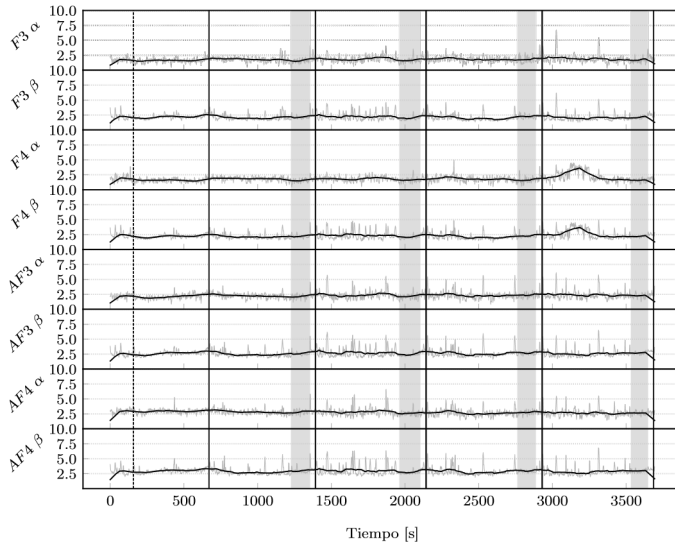
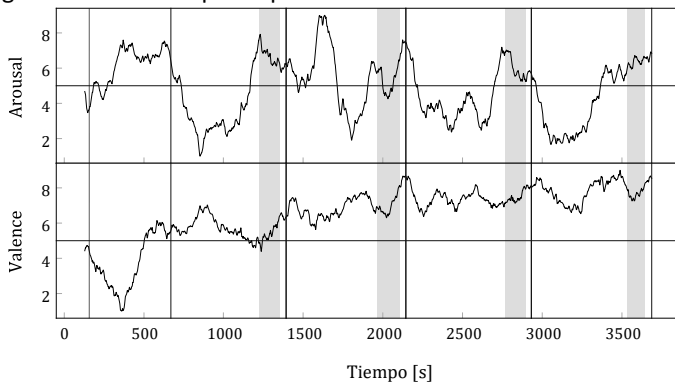


Figura F.3: Movimientos de Arousal y Valence del participante número 2 durante la captura de teclado.



(a) Registros de EEG del participante número 2.



(b) Arousal y Valence del participante número 2 según algoritmo de Ramirez. La franja grisada representa la captura de teclado.

Figura F.4: Registros obtenidos del participante número 2.

		Arousal				Valence			
Estadístico	Fase	1	2	3	4	1	2	3	4
	1	2592	0	294	0	2592	44	830	0
	2	0	3784	1184	21	44	3784	2672	11
	3	294	1184	2888	26	830	2672	2888	44
	4	0	21	26	2520	0	11	44	2520

		Arousal				Valence			
Fase	1	2	3	4	1	2	3	4	

<i>p</i> -value								
	1	0.998	0.000	0.000	0.000	0.998	0.000	0.000
2	0.000	0.999	0.000	0.000	0.000	0.999	0.035	0.000
3	0.000	0.000	0.999	0.000	0.000	0.035	0.999	0.000
4	0.000	0.000	0.000	0.998	0.000	0.000	0.000	0.998

<i>H</i> ₀ aceptada	Fase	Arousal				Valence			
		1	2	3	4	1	2	3	4
1	1	A	NA	NA	NA	A	NA	NA	NA
2	2	NA	A	NA	NA	NA	A	NA	NA
3	3	NA	NA	A	NA	NA	NA	A	NA
4	4	NA	NA	NA	A	NA	NA	NA	A

Cuadro F.5: Resultados de cambios de fase para el participante número 3.

Fase	SAM Img.		SAM Video		EEG				
	A	V	A	V	A	V	σ_A^2	σ_V^2	σ_{AV}
1	5	6	7	7	1.57	6.55	0.01	0.01	0.01
Δ	-5	-2	-6	-2	0.67	-0.49	$\ \Delta\ = 0,83$		
2	2	5	1	5	2.25	6.06	0.02	0.02	0.02
Δ	5	-2	2	-2	-0.30	0.03	$\ \Delta\ = 0,30$		
3	6	3	3	3	1.94	6.09	0.08	0.11	0.09
Δ	4	2	5	-2	1.18	-0.42	$\ \Delta\ = 1,25$		
4	7	5	8	1	3.12	5.67	0.06	0.01	0.02

Cuadro F.6: Diferencias numéricas entre los valores Arousal– Valence adquiridos mediante SAM luego de imágenes, SAM luego de videos (o audio) y EEG para el participante número 3. En el caso de EEG es un valor medio y se muestra la varianza en cada eje y su covarianza. Los valores intermedios marcados como Δ muestran las diferencias; es importante señalar que la diferencia en SAM luego imágenes está calculada contra SAM luego del video de la fase previa.

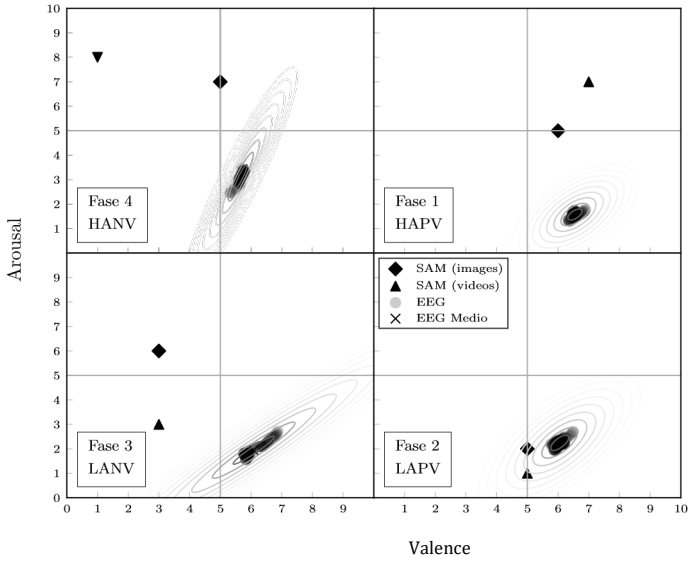
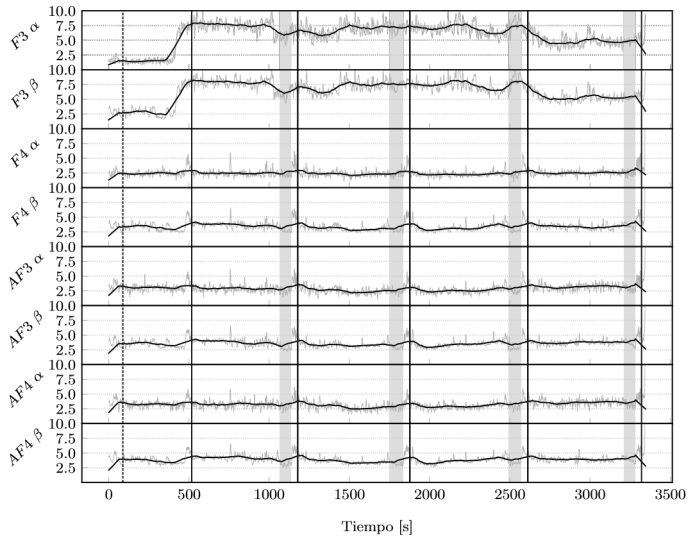
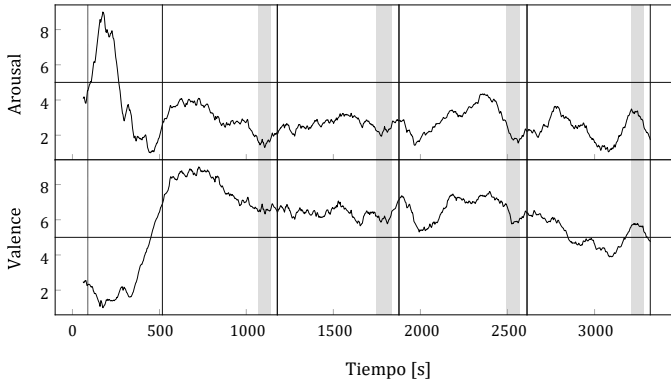


Figura F.5: Movimientos de Arousal y Valence del participante número 3 durante la captura de teclado.



(a) Registros de EEG del participante número 3.



(b) Arousal y Valence del participante número 3 según algoritmo de Ramirez. La franja grisada representa la captura de teclado.

Figura F.6: Registros obtenidos del participante número 3.

		Arousal				Valence			
Estadístico	Fase	1	2	3	4	1	2	3	4
	1		3872	2287	1356	1088	3872	3924	1218
2		2287	4140	1301	1068	3924	4140	1131	1610
3		1356	1301	4232	3385	1218	1131	4232	3055
4		1088	1068	3385	3200	1606	1610	3055	3200

		Arousal				Valence			
p -value	Fase	1	2	3	4	1	2	3	4
	1		0.999	0.000	0.000	0.000	0.999	0.819	0.000
2		0.000	0.999	0.000	0.000	0.819	0.999	0.000	0.000
3		0.000	0.000	0.999	0.366	0.000	0.000	0.999	0.055
4		0.000	0.000	0.366	0.999	0.000	0.000	0.055	0.999

		Arousal				Valence			
H_0 aceptada	Fase	1	2	3	4	1	2	3	4
	1		A	NA	NA	NA	A	NA	NA
2		NA	A	NA	NA	NA	A	NA	NA
3		NA	NA	A	NA	NA	NA	A	NA
4		NA	NA	NA	A	NA	NA	NA	A

Cuadro F.7: Resultados de cambios de fase para el participante número 4.

Fase	SAM Img.		SAM Video		EEG				
	A	V	A	V	A	V	σ_A^2	σ_V^2	σ_{AV}

1	6	5	6	7	3.88	5.51	0.11	0.20	-0.11
Δ	-3	-2	-3	-2	-0.55	-0.01	$\ \Delta\ = 0,55$		
2	3	5	3	5	3.32	5.50	0.60	0.15	-0.07
Δ	2	0	3	-2	1.12	-0.86	$\ \Delta\ = 1,41$		
3	5	5	6	3	4.44	4.64	0.21	0.45	-0.29
Δ	-1	1	1	0	0.13	0.29	$\ \Delta\ = 0,32$		
4	5	4	7	3	4.57	4.93	0.40	0.72	-0.47

Cuadro F.8: Diferencias numéricas entre los valores Arousal– Valence adquiridos mediante SAM luego de imágenes, SAM luego de videos (o audio) y EEG para el participante número 4. En el caso de EEG es un valor medio y se muestra la varianza en cada eje y su covarianza. Los valores intermedios marcados como Δ muestran las diferencias; es importante señalar que la diferencia en SAM luego imágenes está calculada contra SAM luego del video de la fase previa.

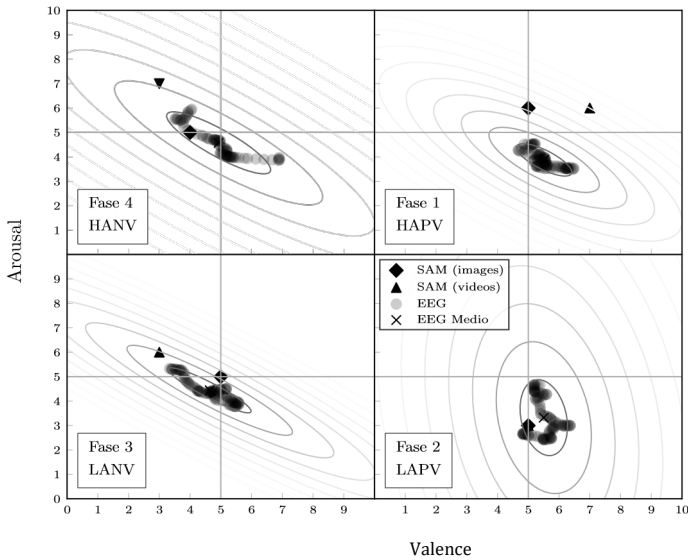
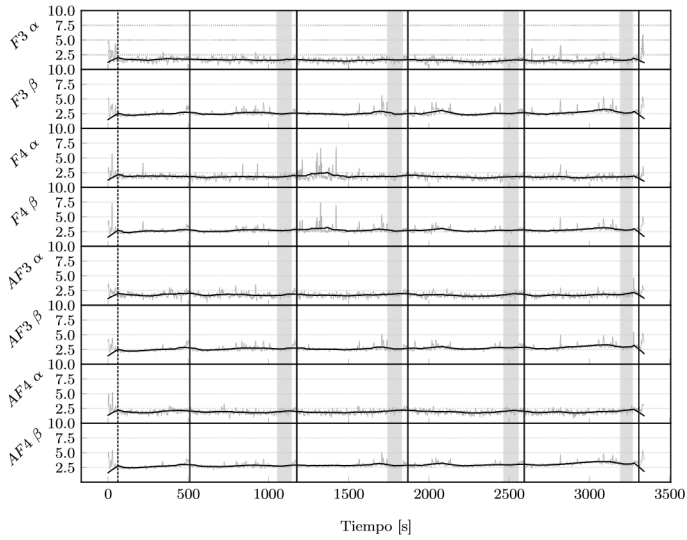
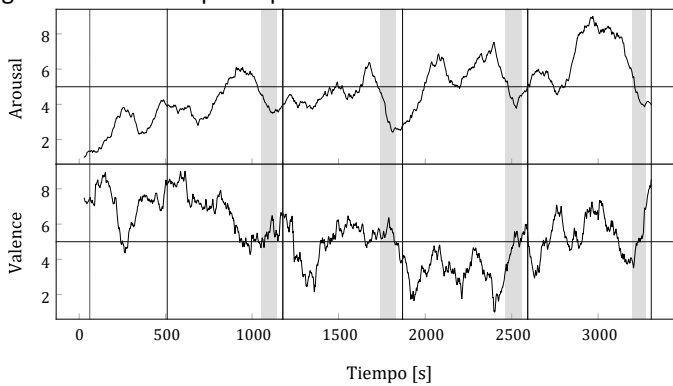


Figura F.7: Movimientos de Arousal y Valence del participante número 4 durante la captura de teclado.



(a) Registros de EEG del participante número 4.



(b) Arousal y Valence del participante número 4 según algoritmo de Ramirez. La franja grisada representa la captura de tecleo.

Figura F.8: Registros obtenidos del participante número 4.

		Arousal				Valence			
Fase		1	2	3	4	1	2	3	4
Estadístico	1	5512	0	4598	3813	5512	0	0	0
	2	0	6160	318	264	0	6160	0	618
	3	4598	318	5304	2923	0	0	5304	4478
	4	3813	264	2923	4802	0	618	4478	4802

		Arousal				Valence			
Fase		1	2	3	4	1	2	3	4

<i>p</i> -value								
	1	0.999	0.000	0.062	0.001	0.999	0.000	0.000
2	0.000	0.999	0.000	0.000	0.000	0.999	0.000	0.000
3	0.062	0.000	0.999	0.000	0.000	0.000	0.999	0.168
4	0.001	0.000	0.000	0.999	0.000	0.000	0.168	0.999

<i>H</i> ₀ aceptada	Fase	Arousal				Valence			
		1	2	3	4	1	2	3	4
1	1	A	NA	NA	NA	A	NA	NA	NA
2	2	NA	A	NA	NA	NA	A	NA	NA
3	3	NA	NA	A	NA	NA	NA	A	NA
4	4	NA	NA	NA	A	NA	NA	NA	A

Cuadro F.9: Resultados de cambios de fase para el participante número 5.

Fase	SAM Img.		SAM Video		EEG				
	A	V	A	V	A	V	σ_A^2	σ_V^2	σ_{AV}
1	6	5	6	5	7.74	4.54	0.02	0.42	0.01
Δ	-1	3	-3	0	-2.26	3.31	$\ \Delta\ = 4,01$		
2	5	8	3	5	5.47	7.85	1.31	0.05	-0.12
Δ	2	-1	2	-2	2.04	-1.05	$\ \Delta\ = 2,29$		
3	5	4	5	3	7.51	6.80	0.27	0.04	0.06
Δ	1	2	1	-1	0.39	0.11	$\ \Delta\ = 0,40$		
4	6	5	6	2	7.90	6.91	0.52	0.31	0.32

Cuadro F.10: Diferencias numéricas entre los valores Arousal– Valence adquiridos mediante SAM luego de imágenes, SAM luego de videos (o audio) y EEG para el participante número 5. En el caso de EEG es un valor medio y se muestra la varianza en cada eje y su covarianza. Los valores intermedios marcados como Δ muestran las diferencias; es importante señalar que la diferencia en SAM luego imágenes está calculada contra SAM luego del video de la fase previa.

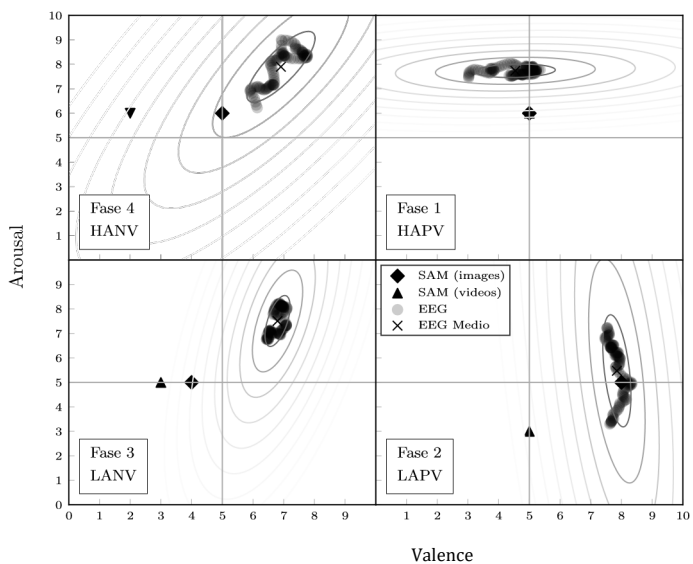
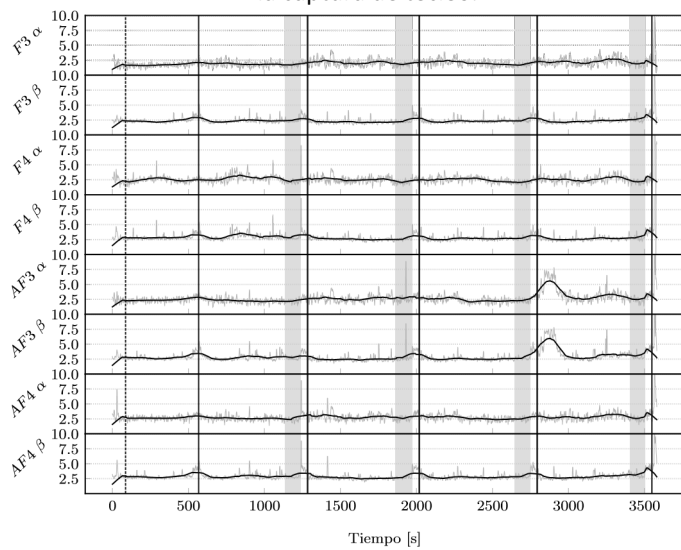
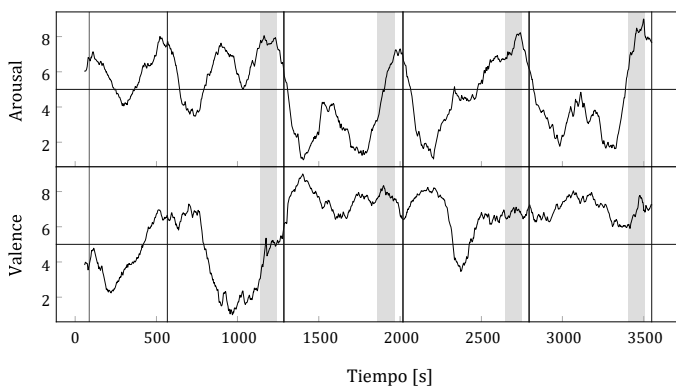


Figura F.9: Movimientos de Arousal y Valence del participante número 5 durante la captura de teclado.



(a) Registros de EEG del participante número 5.



(b) Arousal y Valence del participante número 5 según algoritmo de Ramirez. La franja grisada representa la captura de teclado.

Figura F.10: Registros obtenidos del participante número 5.

		Arousal				Valence			
Estadístico	Fase	1	2	3	4	1	2	3	4
	1	1	5512	730	392	1064	5512	5345	63
2	2	730	6050	3077	5238	5345	6050	120	3483
3	3	392	3077	5202	3687	63	120	5202	3860
4	4	1064	5238	3687	5724	2730	3483	3860	5724

		Arousal				Valence			
p -value	Fase	1	2	3	4	1	2	3	4
	1	1	0.999	0.000	0.000	0.000	0.999	0.346	0.000
2	2	0.000	0.999	0.000	0.162	0.346	0.999	0.000	0.000
3	3	0.000	0.000	0.999	0.000	0.000	0.000	0.999	0.000
4	4	0.000	0.162	0.000	0.999	0.000	0.000	0.000	0.999

		Arousal				Valence			
H_0 aceptada	Fase	1	2	3	4	1	2	3	4
	1	1	A	NA	NA	NA	A	NA	NA
2	2	NA	A	NA	NA	NA	A	NA	NA
3	3	NA	NA	A	NA	NA	NA	A	NA
4	4	NA	NA	NA	A	NA	NA	NA	A

Cuadro F.11: Resultados de cambios de fase para el participante número 7.

Fase	SAM Img.		SAM Video		EEG		
	A	V	A	V	A	V	σ_{AV}^2

1	6	8	6	9	2.71	5.88	0.34	0.24	-0.17
Δ	-3	-4	-4	-5	1.20	-0.06	$\ \Delta\ = 1,20$		
2	3	5	2	4	3.91	5.82	0.25	0.32	0.03
Δ	1	0	3	-2	0.34	-1.16	$\ \Delta\ = 1,21$		
3	3	4	5	2	4.26	4.65	0.28	0.06	0.02
Δ	1	2	2	0	-0.15	0.40	$\ \Delta\ = 0,42$		
4	6	4	7	2	4.11	5.05	1.15	0.77	-0.91

Cuadro F.12: Diferencias numéricas entre los valores Arousal– Valence adquiridos mediante SAM luego de imágenes, SAM luego de videos (o audio) y EEG para el participante número 7. En el caso de EEG es un valor medio y se muestra la varianza en cada eje y su covarianza. Los valores intermedios marcados como Δ muestran las diferencias; es importante señalar que la diferencia en SAM luego imágenes está calculada contra SAM luego del video de la fase previa.

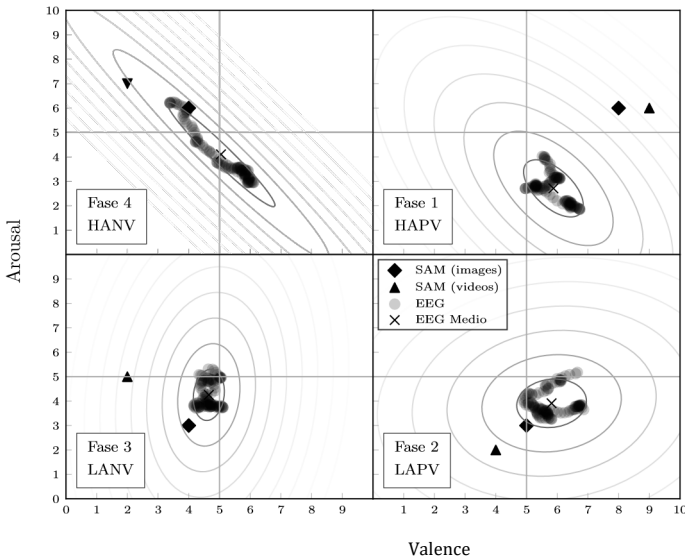
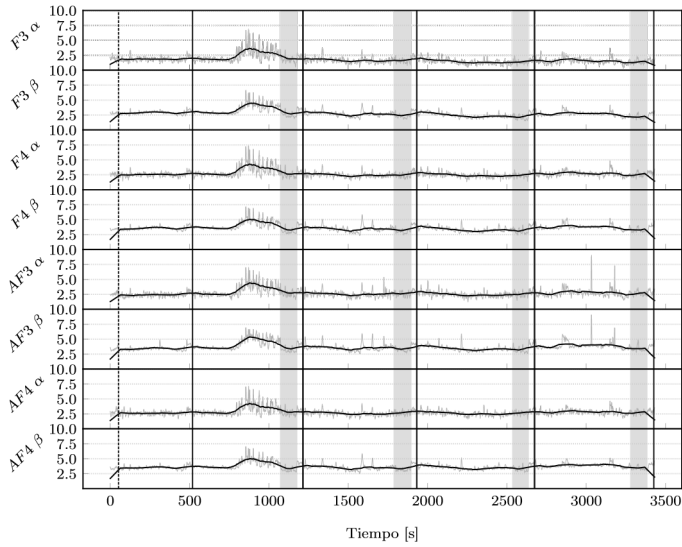
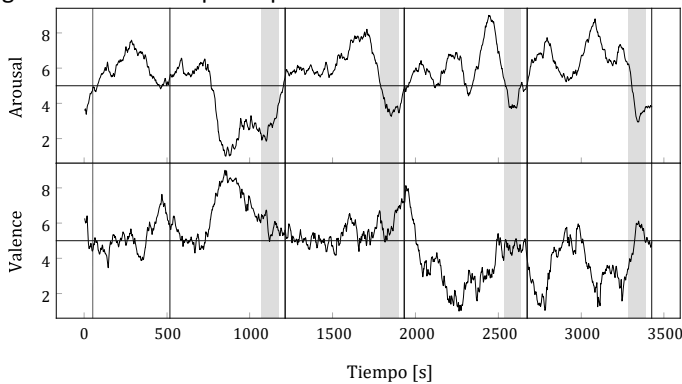


Figura F.11: Movimientos de Arousal y Valence del participante número 7 durante la captura de teclado.



(a) Registros de EEG del participante número 7.



(b) Arousal y Valence del participante número 7 según algoritmo de Ramirez. La franja grisada representa la captura de teclado.

Figura F.12: Registros obtenidos del participante número 7.

		Arousal				Valence			
Estadístico	Fase	1	2	3	4	1	2	3	4
	1	6498	2751	422	2094	6498	0	4299	5243
	2	2751	6612	3182	4136	0	6612	0	645
	3	422	3182	5304	4457	4299	0	5304	4551
	4	2094	4136	4457	4512	5243	645	4551	4512

		Arousal				Valence			
Fase	1	2	3	4	1	2	3	4	
1	6498	2751	422	2094	6498	0	4299	5243	
2	2751	6612	3182	4136	0	6612	0	645	
3	422	3182	5304	4457	4299	0	5304	4551	
4	2094	4136	4457	4512	5243	645	4551	4512	

<i>p</i> -value								
	1	0.999	0.000	0.000	0.000	0.999	0.000	0.001
2	0.000	0.999	0.000	0.002	0.000	0.999	0.000	0.000
3	0.000	0.000	0.999	0.280	0.001	0.000	0.999	0.397
4	0.000	0.002	0.280	0.999	0.694	0.000	0.397	0.999

<i>H</i> ₀ aceptada	Fase	Arousal				Valence			
		1	2	3	4	1	2	3	4
	1	A	NA	NA	NA	A	NA	NA	NA
	2	NA	A	NA	NA	NA	A	NA	NA
	3	NA	NA	A	NA	NA	NA	A	NA
	4	NA	NA	NA	A	NA	NA	NA	A

Cuadro F.13: Resultados de cambios de fase para el participante número 8.

Fase	SAM Img.		SAM Video		EEG				
	A	V	A	V	A	V	σ_A^2	σ_V^2	σ_{AV}
1	5	5	7	8	7.74	4.62	0.27	0.08	0.09
Δ	-4	-3	-1	-2	-0.56	-1.01	$\ \Delta\ = 1,16$		
2	3	5	6	6	7.18	3.61	0.23	0.07	-0.01
Δ	-4	-2	-2	-3	-0.42	1.14	$\ \Delta\ = 1,21$		
3	2	4	4	3	6.76	4.75	0.12	0.03	0.00
Δ	-2	2	2	2	0.11	-0.18	$\ \Delta\ = 0,21$		
4	2	5	6	5	6.88	4.57	0.52	0.27	-0.34

Cuadro F.14: Diferencias numéricas entre los valores Arousal– Valence adquiridos mediante SAM luego de imágenes, SAM luego de videos (o audio) y EEG para el participante número 8. En el caso de EEG es un valor medio y se muestra la varianza en cada eje y su covarianza. Los valores intermedios marcados como Δ muestran las diferencias; es importante señalar que la diferencia en SAM luego imágenes está calculada contra SAM luego del video de la fase previa.

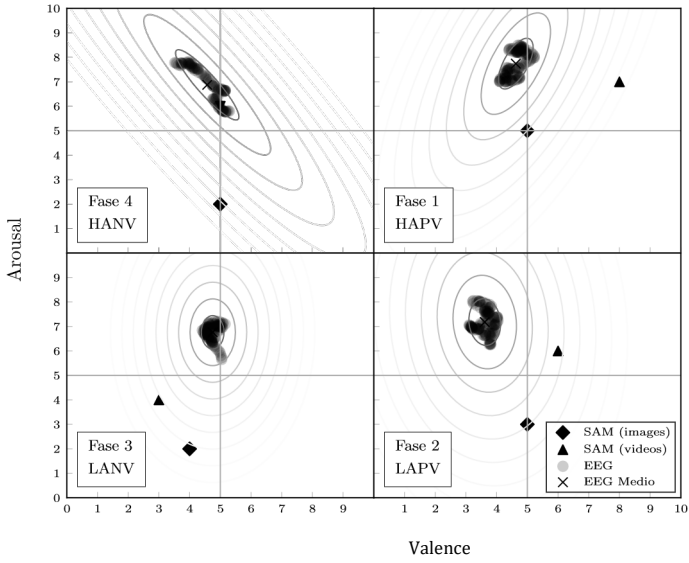
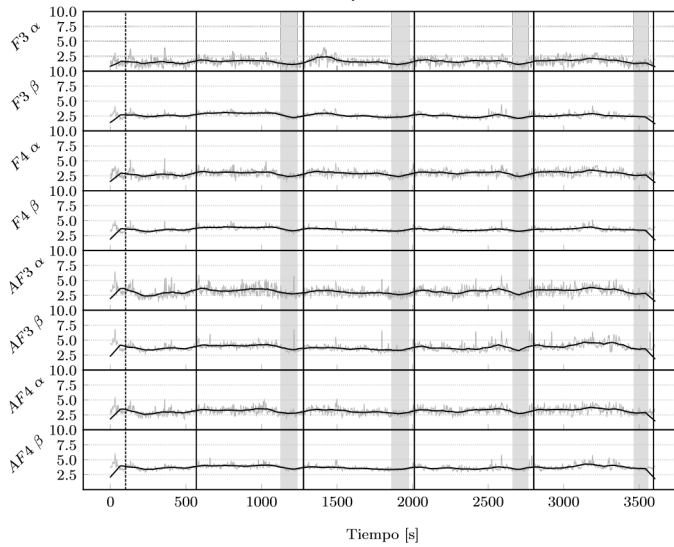
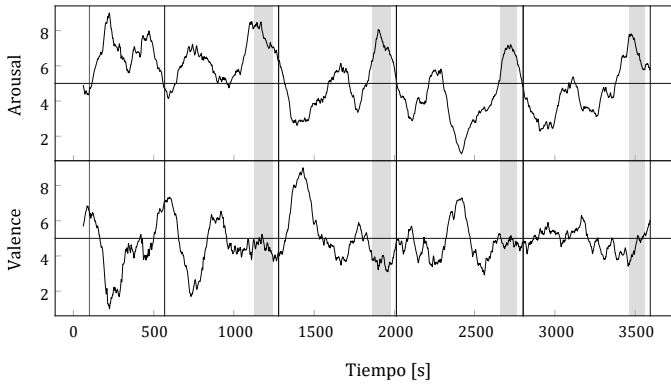


Figura F.13: Movimientos de Arousal y Valence del participante número 8 durante la captura de teclado.



(a) Registros de EEG del participante número 8.



(b) Arousal y Valence del participante número 8 según algoritmo de Ramirez. La franja grisada representa la captura de teclado.

Figura F.14: Registros obtenidos del participante número 8.

		Arousal				Valence			
Estadístico	Fase	1	2	3	4	1	2	3	4
	1	6050	2061	4093	4	6050	0	2	34
2	2061	5304	1532	0	0	5304	618	3975	
3	4093	1532	4608	487	2	618	4608	3673	
4	4	0	487	4802	34	3975	3673	4802	

		Arousal				Valence			
<i>p</i> -value	Fase	1	2	3	4	1	2	3	4
	1	0.999	0.000	0.005	0.000	0.999	0.000	0.000	0.000
2	0.000	0.999	0.000	0.000	0.000	0.999	0.000	0.009	
3	0.005	0.000	0.999	0.000	0.000	0.000	0.999	0.008	
4	0.000	0.000	0.000	0.999	0.000	0.009	0.008	0.999	

		Arousal				Valence			
<i>H</i> ₀ aceptada	Fase	1	2	3	4	1	2	3	4
	1	A	NA	NA	NA	A	NA	NA	NA
2	NA	A	NA	NA	NA	A	NA	NA	
3	NA	NA	A	NA	NA	NA	A	NA	
4	NA	NA	NA	A	NA	NA	NA	A	

Cuadro F.15: Resultados de cambios de fase para el participante número 9.

Fase	SAM Img.		SAM Video		EEG				
	A	V	A	V	A	V	σ_A^2	σ_V^2	σ_{AV}
1	7	7	7	7	6.36	3.30	0.21	0.16	-0.02
Δ	0	0	0	0	-0.88	2.75	$\ \Delta\ = 2,89$		
2	7	7	7	7	5.48	6.05	0.54	0.10	0.16
Δ	0	0	-2	-6	1.01	-0.85	$\ \Delta\ = 1,32$		
3	7	7	5	1	6.49	5.20	0.48	0.16	-0.22
Δ	-2	4	0	2	1.07	0.49	$\ \Delta\ = 1,18$		
4	3	5	5	3	7.56	5.69	0.16	1.16	0.36

Cuadro F.16: Diferencias numéricas entre los valores Arousal– Valence adquiridos mediante SAM luego de imágenes, SAM luego de videos (o audio) y EEG para el participante número 9. En el caso de EEG es un valor medio y se muestra la varianza en cada eje y su covarianza. Los valores intermedios marcados como Δ muestran las diferencias; es importante señalar que la diferencia en SAM luego imágenes está calculada contra SAM luego del video de la fase previa.

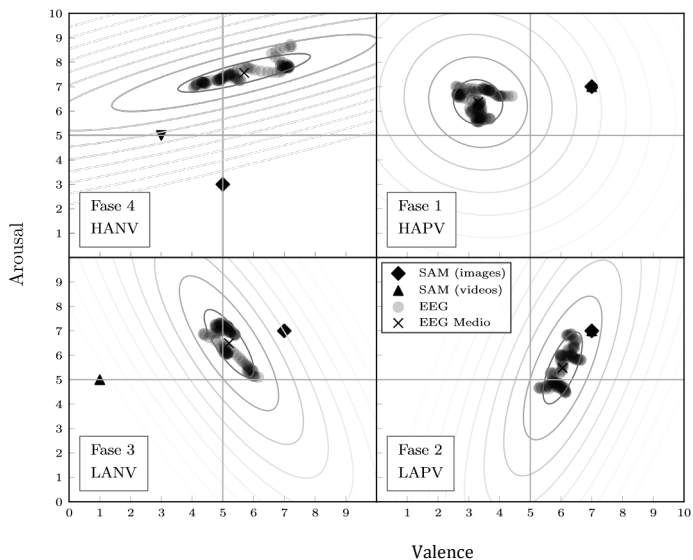
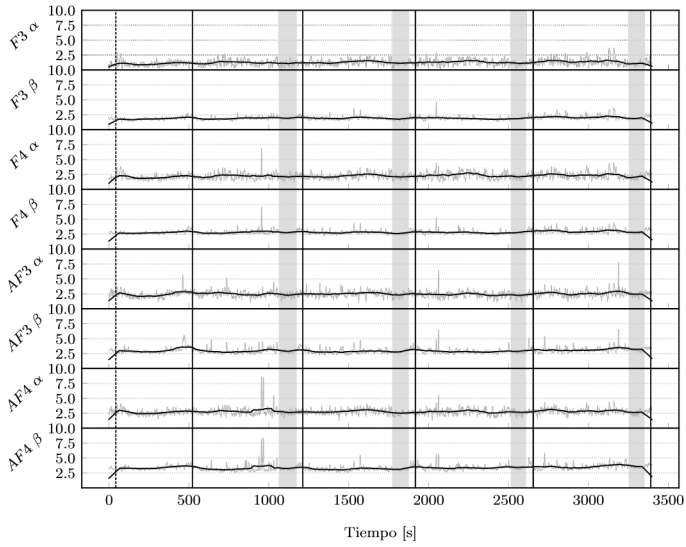
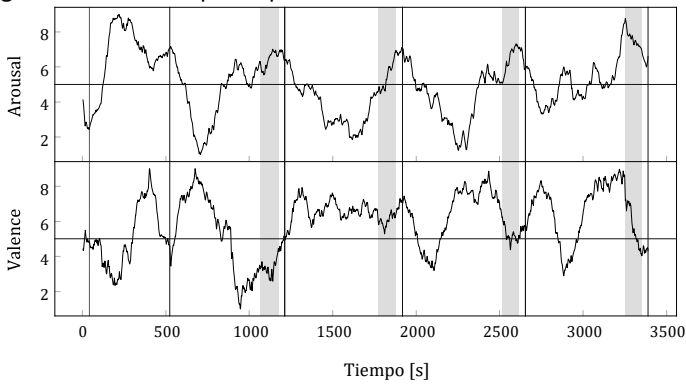


Figura F.15: Movimientos de Arousal y Valence del participante número 9 durante la captura de teclado.



(a) Registros de EEG del participante número 9.



(b) Arousal y Valence del participante número 9 según algoritmo de Ramirez. La franja grisada representa la captura de teclado.

Figura F.16: Registros obtenidos del participante número 9.

		Arousal				Valence			
Estadístico	Fase	1	2	3	4	1	2	3	4
	1		7938	4878	5432	3462	7938	6143	0
2		4878	8192	3060	1863	6143	8192	115	0
3		5432	3060	8978	6114	0	115	8978	2104
4		3462	1863	6114	6384	0	0	2104	6384

		Arousal				Valence			
p-value	Fase	1	2	3	4	1	2	3	4
	1		0.999	0.000	0.000	0.000	0.999	0.001	0.000
2		0.000	0.999	0.000	0.000	0.001	0.999	0.000	0.000
3		0.000	0.000	0.999	0.009	0.000	0.000	0.999	0.000
4		0.000	0.000	0.009	0.999	0.000	0.000	0.000	0.999

		Arousal				Valence			
H ₀ aceptada	Fase	1	2	3	4	1	2	3	4
	1		A	NA	NA	NA	A	NA	NA
2		NA	A	NA	NA	NA	A	NA	NA
3		NA	NA	A	NA	NA	NA	A	NA
4		NA	NA	NA	A	NA	NA	NA	A

Cuadro F.17: Resultados de cambios de fase para el participante número 10.

Fase	SAM Img.		SAM Video		EEG				
	A	V	A	V	A	V	σ_A^2	σ_V^2	σ_{AV}
1	6	5	7	4	7.10	6.10	1.58	0.06	0.02
Δ	-3	4	-5	1	-0.99	0.26	$\ \Delta\ = 1,02$		
2	4	8	2	5	6.11	6.36	1.94	0.46	-0.92
Δ	4	-2	7	-4	1.70	-1.56	$\ \Delta\ = 2,30$		
3	6	3	9	1	7.80	4.80	0.73	0.10	-0.15
Δ	-3	3	0	0	0.26	-0.62	$\ \Delta\ = 0,67$		
4	6	4	9	1	8.06	4.18	0.95	0.19	-0.40

Cuadro F.18: Diferencias numéricas entre los valores Arousal-Valence adquiridos mediante SAM luego de imágenes, SAM luego de videos (o audio) y EEG para el participante número 10. En el caso de EEG es un valor medio y se muestra la varianza en cada eje y su covarianza. Los valores intermedios marcados como Δ muestran las diferencias; es importante señalar que la diferencia en SAM luego imágenes está calculada contra SAM luego del video de la fase previa.

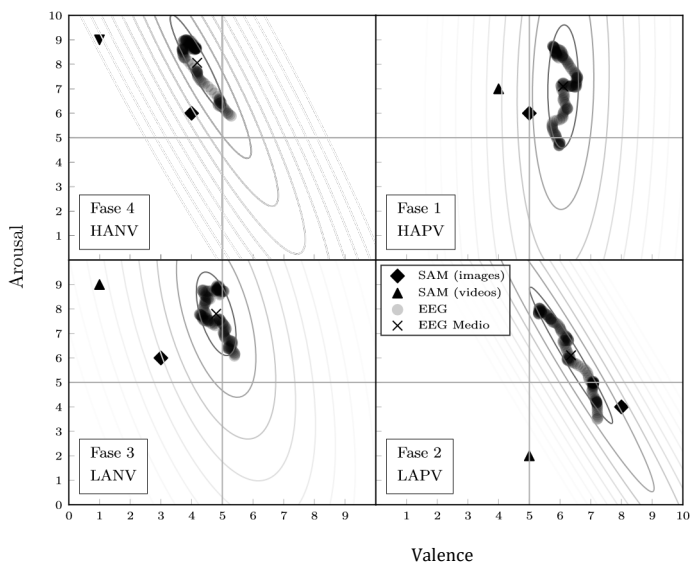
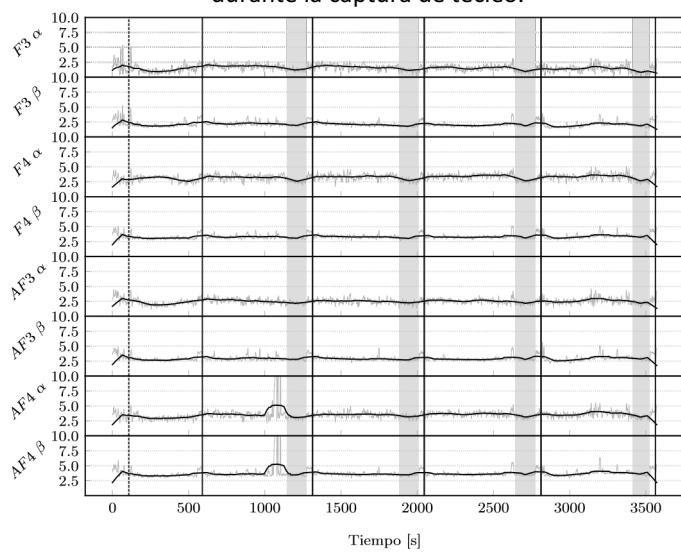
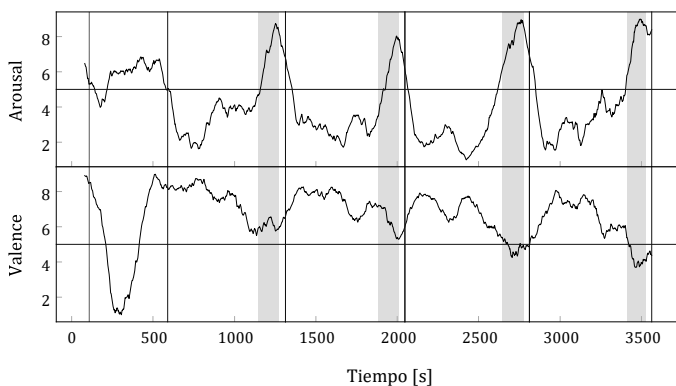


Figura F.17: Movimientos de Arousal y Valence del participante número 10 durante la captura de teclado.



(a) Registros de EEG del participante número 10.



(b) Arousal y Valence del participante número 10 según algoritmo de Ramirez. La franja grisada representa la captura de tecleo.

Figura F.18: Registros obtenidos del participante número 10.

		Arousal				Valence			
Fase		1	2	3	4	1	2	3	4
Estadístico	1	5724	0	5006	2931	5724	0	0	0
	2	0	4608	248	0	0	4608	1614	1872
	3	5006	248	4512	2948	0	1614	4512	4334
	4	2931	0	2948	4232	0	1872	4334	4232

		Arousal				Valence			
Fase		1	2	3	4	1	2	3	4
p -value	1	0.999	0.000	0.855	0.000	0.999	0.000	0.000	0.000
	2	0.000	0.999	0.000	0.000	0.000	0.999	0.000	0.000
	3	0.855	0.000	0.999	0.000	0.000	0.000	0.999	0.924
	4	0.000	0.000	0.000	0.999	0.000	0.000	0.924	0.999

		Arousal				Valence			
Fase		1	2	3	4	1	2	3	4
H_0 aceptada	1	A	NA	NA	NA	A	NA	NA	NA
	2	NA	A	NA	NA	NA	A	NA	NA
	3	NA	NA	A	NA	NA	NA	A	A
	4	NA	NA	NA	A	NA	NA	A	A

Cuadro F.19: Resultados de cambios de fase para el participante número 11.

Fase	SAM Img.		SAM Video		EEG				
	A	V	A	V	A	V	σ_A^2	σ_V^2	σ_{AV}
1	5	7	7	8	7.48	5.16	0.23	0.25	-0.18
Δ	-2	-1	-2	-1	1.08	-3.55	$\ \Delta\ = 3,71$		
2	5	7	5	7	8.57	1.61	0.08	0.21	-0.11
Δ	0	0	0	-1	-1.13	0.60	$\ \Delta\ = 1,28$		
3	5	7	5	6	7.43	2.21	0.40	0.13	0.09
Δ	1	0	0	-3	-0.35	0.12	$\ \Delta\ = 0,37$		
4	6	6	5	3	7.08	2.32	0.45	0.48	0.38

Cuadro F.20: Diferencias numéricas entre los valores Arousal–Valence adquiridos mediante SAM luego de imágenes, SAM luego de videos (o audio) y EEG para el participante número 11. En el caso de EEG es un valor medio y se muestra la varianza en cada eje y su covarianza. Los valores intermedios marcados como Δ muestran las diferencias; es importante señalar que la diferencia en SAM luego imágenes está calculada contra SAM luego del video de la fase previa.

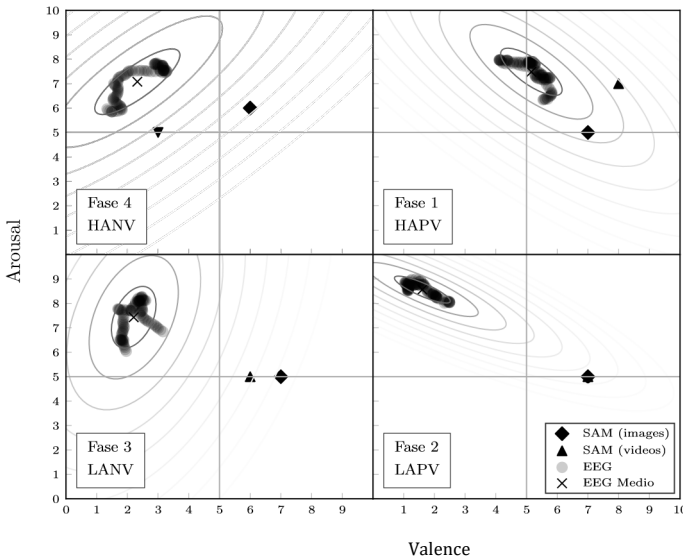
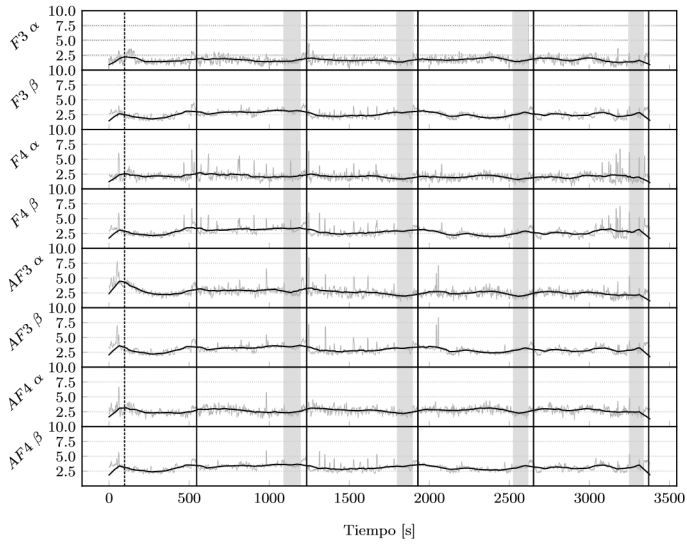
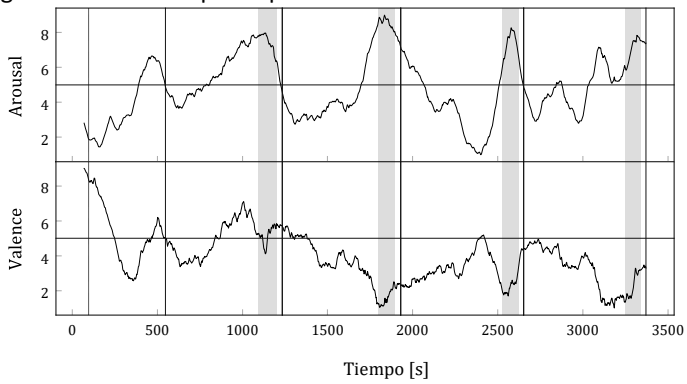


Figura F.19: Movimientos de Arousal y Valence del participante número 11 durante la captura de teclado.



(a) Registros de EEG del participante número 11.



(b) Arousal y Valence del participante número 11 según algoritmo de Ramirez. La franja grisada representa la captura de teclado.

Figura F.20: Registros obtenidos del participante número 11.

		Arousal				Valence			
Estadístico	Fase	1	2	3	4	1	2	3	4
	1	9248	4119	4715	126	9248	1811	7151	171
	2	4119	8844	7339	6523	1811	8844	5045	8408
	3	4715	7339	9522	1796	7151	5045	9522	4169
	4	126	6523	1796	9660	171	8408	4169	9660

		Arousal				Valence			
p-value	Fase	1	2	3	4	1	2	3	4
	1	0.999	0.000	0.000	0.000	0.999	0.000	0.001	0.000
	2	0.000	0.999	0.004	0.000	0.000	0.999	0.000	0.198
	3	0.000	0.004	0.999	0.000	0.001	0.000	0.999	0.000
	4	0.000	0.000	0.000	0.999	0.000	0.198	0.000	0.999

		Arousal				Valence			
H ₀ aceptada	Fase	1	2	3	4	1	2	3	4
	1	A	NA	NA	NA	A	NA	NA	NA
	2	NA	A	NA	NA	NA	A	NA	NA
	3	NA	NA	A	NA	NA	NA	A	NA
	4	NA	NA	NA	A	NA	NA	NA	A

Cuadro F.21: Resultados de cambios de fase para el participante número 12.

Fase	SAM Img.		SAM Video		EEG				
	A	V	A	V	A	V	σ_A^2	σ_V^2	σ_{AV}
1	7	8	8	9	3.25	7.36	1.98	0.39	-0.83
Δ	-5	-5	-6	-4	-1.17	1.07	$\ \Delta\ = 1,58$		
2	3	4	2	5	2.08	8.42	0.86	0.31	-0.45
Δ	0	-2	1	-4	0.16	-0.85	$\ \Delta\ = 0,87$		
3	2	3	3	1	2.24	7.57	0.45	1.42	-0.79
Δ	2	4	-2	0	-0.73	1.05	$\ \Delta\ = 1,28$		
4	5	5	1	1	1.51	8.62	0.03	0.05	-0.04

Cuadro F.22: Diferencias numéricas entre los valores Arousal-Valence adquiridos mediante SAM luego de imágenes, SAM luego de videos (o audio) y EEG para el participante número 12. En el caso de EEG es un valor medio y se muestra la varianza en cada eje y su covarianza. Los valores intermedios marcados como Δ muestran las diferencias; es importante señalar que la diferencia en SAM luego imágenes está calculada contra SAM luego del video de la fase previa.

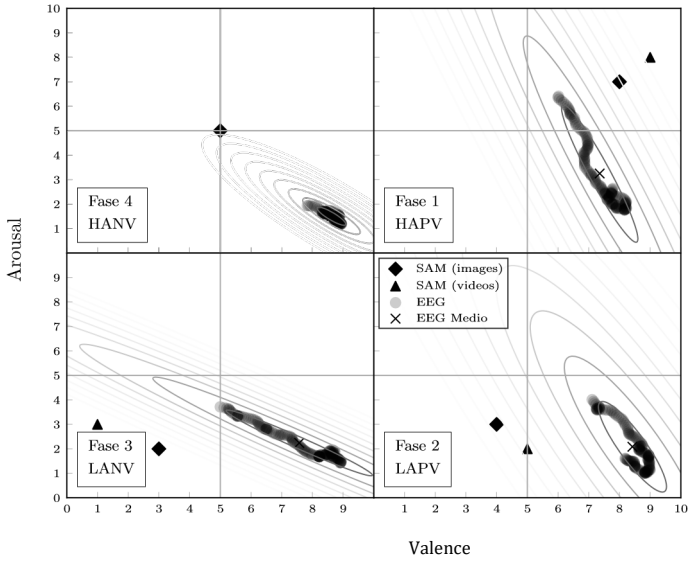
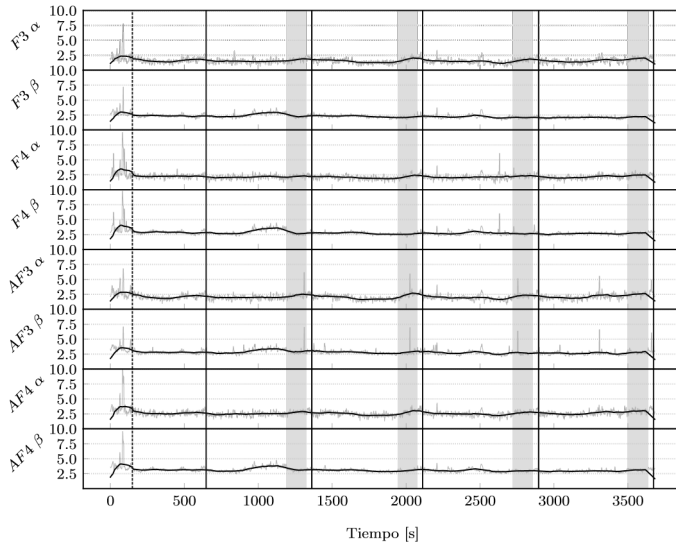
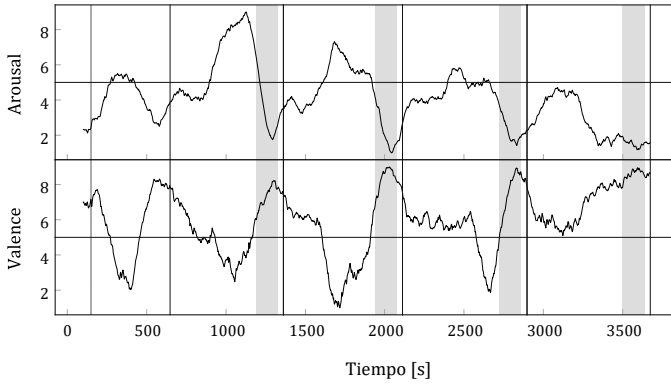


Figura F.21: Movimientos de Arousal y Valence del participante número 12 durante la captura de teclado.



(a) Registros de EEG del participante número 12.



(b) Arousal y Valence del participante número 12 según algoritmo de Ramirez. La franja grisada representa la captura de tecleo.

Figura F.22: Registros obtenidos del participante número 12.

Consumo de recursos y capacidad computacional

El tiempo de ejecución de las pruebas no fue trivial, en principio, utilizando un primer *framework* en Python fue posible realizar algunas aproximaciones y divisar resultados, pero esta solución no era escalable a grandes cantidades de datos, por lo que fue necesario reimplementar el *framework* en una plataforma compilada. Para ello se utilizó el lenguaje C y se optimizó el tiempo de procesamiento lo máximo posible. Una de estas optimizaciones fue la carga total del dataset en memoria RAM. En python esto resultó imposible, la ejecución era terminada por el sistema operativo con un exit code 137 (*Out of memory*), es decir, el programa moría al acabarse la memoria. Utilizar memoria swap no era una opción, ya que alentaría toda la ejecución y terminaría siendo equivalente en tiempo a leer los registros en lotes desde el disco. Se tomó la decisión de reimplementar en C luego de probar la implementación en Python en una máquina de 72 cores y 744 Mb de RAM con resultados fallidos y el programa terminado luego de 40 minutos de ejecución. Esa misma ejecución en la reimplemmtación en C, adicionadas las comparaciones finales y el cálculo de EER, tomó solamente 5 minutos y 22 segundos.

top - 06:48:30 up 8 min, 4 users, load average: 32.27, 12.80, 4.83 Tasks: 697 total, 33 running, 341 sleeping, 0 stopped, 0 zombie
 %Cpu(s): 43.5 us, 0.6 sy, 0.0 ni, 55.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
 KiB Mem : 19369113+total, 18710457+free, 5286604 used, 1299972 buff/cache

KiB Swap:		0 total,		0 free,		0 used. 18719830+avail Mem	
PID	USER	PR	NI	VIRT	RES	SHR	S %CPU %MEM TIME+ COMMAND
3173	ubuntu	20	0	868636	826060	824	R 100.0 0.4 1:40.55 biometrics
3146	ubuntu	20	0	1154008	806524	852	R 99.7 0.4 1:39.99 biometrics
3148	ubuntu	20	0	993916	808240	860	R 99.7 0.4 1:39.71 biometrics
3149	ubuntu	20	0	1184072	805360	848	R 99.7 0.4 1:40.27 biometrics
3153	ubuntu	20	0	1069604	812384	844	R 99.7 0.4 1:40.05 biometrics
3155	ubuntu	20	0	1124912	810852	836	R 99.7 0.4 1:40.11 biometrics
3157	ubuntu	20	0	1043408	816292	848	R 99.7 0.4 1:40.15 biometrics
3160	ubuntu	20	0	1096776	802844	832	R 99.7 0.4 1:39.94 biometrics
3172	ubuntu	20	0	887100	827328	860	R 99.7 0.4 1:40.64 biometrics
3174	ubuntu	20	0	948308	815612	860	R 99.7 0.4 1:40.46 biometrics
3175	ubuntu	20	0	970636	816064	860	R 99.7 0.4 1:40.71 biometrics
3120	ubuntu	20	0	1030672	806484	812	R 99.3 0.4 1:38.28 biometrics
3122	ubuntu	20	0	859764	812136	772	R 99.3 0.4 1:38.35 biometrics
3123	ubuntu	20	0	1139340	809004	860	R 99.3 0.4 1:39.08 biometrics
3131	ubuntu	20	0	1168920	806184	852	R 99.3 0.4 1:39.47 biometrics
3133	ubuntu	20	0	916616	815892	860	R 99.3 0.4 1:39.50 biometrics
3142	ubuntu	20	0	959352	810488	860	R 99.3 0.4 1:38.67 biometrics
3147	ubuntu	20	0	834596	822160	860	R 99.3 0.4 1:39.80 biometrics
3154	ubuntu	20	0	851136	819964	832	R 99.3 0.4 1:40.12 biometrics
3159	ubuntu	20	0	906536	815992	860	R 99.3 0.4 1:39.66 biometrics
3161	ubuntu	20	0	926932	817504	820	R 99.3 0.4 1:40.17 biometrics
3116	ubuntu	20	0	896696	817484	848	R 99.0 0.4 1:38.57 biometrics
3117	ubuntu	20	0	1083072	811760	836	R 99.0 0.4 1:39.03 biometrics
3121	ubuntu	20	0	937504	803400	760	R 99.0 0.4 1:38.00 biometrics
3132	ubuntu	20	0	842748	817312	860	R 99.0 0.4 1:39.27 biometrics
3134	ubuntu	20	0	1056384	817428	860	R 99.0 0.4 1:39.23 biometrics
3115	ubuntu	20	0	982164	810300	844	R 98.7 0.4 1:38.17 biometrics
3156	ubuntu	20	0	1018180	817928	852	R 98.7 0.4 1:40.21 biometrics
3113	ubuntu	20	0	826692	811300	860	R 97.4 0.4 1:35.11 biometrics

Ejemplo de ejecución del software configurado para 64 núcleos en una computadora con 36 núcleos.

top - 05:11:34 up 14 min, 2 users, load average: 63.97, 56.04, 31.31
 Tasks: 444 total, 65 running, 199 sleeping, 0 stopped, 0 zombie
 %Cpu(s): 99.2 us, 0.4 sy, 0.0 ni, 0.3 id, 0.1 wa, 0.0 hi, 0.0 si, 0.0 st KiB Mem : 61818888 total, 41729584 free, 18529828 used, 1559476 buff/cache

KiB Swap:		0 total,		0 free,		0 used. 42721204 avail Mem	
PID	USER	PR	NI	VIRT	RES	SHR	S %CPU %MEM TIME+ COMMAND
8639	ubuntu	20	0	1068584	990924	2040	R 83.6 1.6 7:11.29 biometrics
8645	ubuntu	20	0	1068584	987016	2036	R 79.9 1.6 7:02.88 biometrics
8642	ubuntu	20	0	1068584	991136	2032	R 79.3 1.6 7:11.16 biometrics
8654	ubuntu	20	0	1068584	977272	2040	R 78.0 1.6 6:49.21 biometrics
8652	ubuntu	20	0	1068584	978060	2040	R 77.3 1.6 6:51.06 biometrics
8637	ubuntu	20	0	1068584	985464	2040	R 77.0 1.6 6:59.71 biometrics
8641	ubuntu	20	0	1068584	989496	2036	R 76.0 1.6 7:07.88 biometrics
8634	ubuntu	20	0	1068584	996556	2040	R 75.0 1.6 7:22.18 biometrics

Apéndice G. Consumo de recursos y capacidad computacional

Referencias bibliográficas

- Areej Alhothali. Modeling user affect using interaction events. Master's thesis, University of Waterloo, jul. 2011. URL <http://hdl.handle.net/10012/6027>. (En página 63).
- James A. Anderson. *An Introduction to Neural Networks*. A Bradford Book, 1995. ISBN 978-0-262-51081-3. (En páginas 56 y 142).
- Livia C. F. Araujo, Luiz H. R. Jr. Sucupira, Miguel Gustavo Lizarraga, Lee Luan Ling y João Baptista T. Yabu-uti. User authentication through typing biometrics features. *Signal Processing, IEEE Transactions on*, 53(2):851–855, 2005. (En páginas 45 y 47).
- Ritwik Banerjee, Song Feng, Jun Seok Kang y Yejin Choi. Keystroke patterns as prosody in digital writings: A case study with deceptive reviews and essays. En *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, páginas 1469–1473, Doha, Qatar, oct. 2014. Association for Computational Linguistics. (En páginas 87, 103 y 199).
- Saikat Basu, Nabakumar Jana, Arnab Bag, M Mahadevappa, Jayanta Mukherjee, Somesh Kumar y Rajlakshmi Guha. Emotion recognition based on physiological signals using valence–arousal model. En *2015 Third International Conference on Image Information Processing (ICIIP)*, páginas 50–55. Institute of Electrical and Electronics Engineers (IEEE), dic. 2015. ISBN 978-1-5090-0148-4. doi: 10.1109/ICIIP.2015.7414739. (En página 31).
- Francesco Bergadano, Daniele Gunetti y Claudia Picardi. User authentication through keystroke dynamics. *ACM Transactions on Information and System Security (TISSEC)*, 5(4):367–397, nov. 2002. ISSN 1094-9224. doi: 10.1145/581271.581272. (En páginas 42 y 45).
- Christopher M. Bishop. *Neural networks for pattern recognition*. Clarendon press Oxford, 1996. ISBN 978-0-19-853864-6. (En página 142).
- Saleh Bleha, Charles Slivinsky y Bassam Hussien. Computeraccess security systems using keystroke dynamics. *Pattern Analysis and*

- Machine Intelligence, IEEE Transactions on*, 12(12): 1217–1222, 1990. doi: 10.1109/34.62613. (En páginas 42, 43, 45, 46, 47 y 48).
- Paul Bourke. Various eeg file formats and conventions, abr. 1997. Disponible en <http://paulbourke.net/dataformats/eeg/>. (En página 180).
- Margaret M. Bradley. Emotional memory: A dimensional analysis. En *Emotions: Essays on Emotion Theory*, páginas 97–134. Taylor & Francis, Hillsdale, NJ: Erlbaum, 1994. ISBN 978-1-317-78197-4. URL <https://books.google.com.ar/books?id=H0TrAgAAQBAJ>. (en página 31).
- Margaret M. Bradley y Peter J. Lang. Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*, 25(1):49–59, 1994. (En páginas 29, 30, 31, 181 y 184).
- Margaret M. Bradley y Peter J. Lang. Emotion and motivation. *Handbook of Psychophysiology*, páginas 581–607, 2006. doi: 10.1017/cbo9780511546396.025. (En página 22).
- Margaret M. Bradley y Peter J. Lang. *Handbook of Emotion Elicitation and Assessment*, volume 29 de *Series in Affective Science*. Oxford University Press, New-York, NY, USA, 2007a. ISBN 978-0-19-972766-7. (En página 35).
- Margaret M Bradley y Peter J. Lang. The international affective digitized sounds (; IADS-2): Affective ratings of sounds and instruction manual. *University of Florida, Gainesville, FL, Tech. Rep. B-3*, 2007b. (En páginas 28 y 34).
- Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. (En página 151).
- William Lowe Bryan y Noble Harter. Studies in the physiology and psychology of the telegraphic language. *Psychological Review*, 4(1):27, 1897. (En página 9).
- Kay Bryant y John Campbell. User behaviours associated with password security and management. *Australasian Journal of Information Systems*, 14(1), 2006. (En página 1).
- Krisztian Buza. Person identification based on keystroke dynamics: Demo and open challenge. En *CAiSE Forum*, páginas 161–168, 2016. (En página 199).

- Enrique P. Calot y Jorge S. Ierache. Multimodal biometric recording architecture for the exploitation of applications in the context of affective computing. En *Proceedings del XXIII Congreso Argentino de Ciencias de la Computación (La Plata, 2017)*, number 10529, páginas 1030–1039, 2017. ISBN 978-950-34-1539-9. URL <http://hdl.handle.net/10915/63866>. (En páginas 77 y 181).
- Enrique P. Calot, Juan Manuel Rodríguez y Jorge S. Ierache. Improving versatility in keystroke dynamic systems. En *Proceedings del XIX Congreso Argentino de Ciencias de la Computación*, number 5606, 2013. ISBN 978-987-23963-1-2. URL <http://hdl.handle.net/10915/32428>. (En páginas 41, 46 y 48).
- Enrique P. Calot, Francisco Pirra, Juan Manuel Rodríguez, Gustavo Pereira, Juan Iribarren y Jorge S. Ierache. Métodos adaptativos de educación de dinámica de tecleo centrado en el contexto emocional de un individuo aplicando interfaz cerebro computadora. En *XVI Workshop de Investigadores en Ciencias de la Computación*, mar. 2014a. ISBN 978-950-34-1084-4. URL <http://isia.fi.uba.ar/papers/calot14.pdf>.
- Enrique P. Calot, Juan Manuel Rodríguez y Jorge S. Ierache. Improving versatility in keystroke dynamic systems. En Jorge Raúl Finochietto y Patricia Mabel Pesado, editors, *Computer Science & Technology Series. XIX Argentine Congress of Computer Science, Selected papers*, páginas 289–298. Editorial de la Universidad Nacional de La Plata (EDULP), 2014b. ISBN 978-987-1985-49-4. URL <http://isia.fi.uba.ar/papers/calot14b.pdf>.
- Enrique P. Calot, Ezequiel L. Aceto, Juan Manuel Rodríguez, Ariel Liguori, María Alejandra Ochoa, Hernán D. Merlino, Enrique Fernández, Nahuel González, Francisco Pirra y Jorge S. Ierache. Líneas de investigación del laboratorio de sistemas de información avanzados. En *Proceedings del XVII Workshop de Investigadores en Ciencias de la Computación (Salta, 2015)*, página 4, jun. 2015. ISBN 978-987-633-134-0. URL <http://hdl.handle.net/10915/46107>.
- Enrique P. Calot, Federico M. Rossi, Nahuel Gonzalez, Waldo Hasperué y Jorge S. Ierache. Avances en educación de dinámica de tecleo y el contexto emocional de un individuo aplicando interfaz cerebro computadora. En *XVIII Workshop de Investigadores en Ciencias de la Computación (WICC 2016, Entre Ríos, Argentina)*, páginas 872–

876, jun. 2016. ISBN 978-950-698-377-2. URL <http://hdl.handle.net/10915/53247>.

- Enrique P. Calot, Nahuel González, Jorge S. Ierache y Waldo Hasperué. A convergence model for distance-based keystroke dynamics classification methods, 2019a. Comunicado el 7 de febrero de 2019. (En páginas 7 y 83).
- Enrique P. Calot, Jorge S. Ierache y Waldo Hasperué. Document typist identification by classification metrics applying keystroke dynamics under unidealised conditions. En *Second International Workshop on Computational Document Forensics, IWCDF at ICDAR 2019, Sydney, Australia, September 22–25, 2019*, páginas 19–24. IEEE, 2019b. ISBN 978-1-7281-5054-3. doi: 10.1109/ICDARW.2019.70136. (En páginas 7, 83, 104 y 113).
- Enrique P. Calot, Jorge S. Ierache y Waldo Hasperué. Robustness of keystroke dynamics identification algorithms against brain-wave variations associated with emotional variations. En *Intelligent Systems and Applications*, volume 1037 de *Advances in Intelligent Systems and Computing*. Springer, 2019c. ISBN 978-3-030-29515-8. doi: 10.1007/978-3-030-29516-5_15.
- John Campbell, Dale Kleeman y Wanli Ma. The good and not so good of enforcing password composition rules. *Information Systems Security*, 16(1):2–8, 2007. (En página 2).
- Cheng-Yao Chen, Yue-Kai Huang y Perry Cook. Visual/acoustic emotion recognition. En *2005 IEEE International Conference on Multimedia and Expo*, páginas 1468–1471, jul. 2005. doi: 10.1109/ICME.2005.1521709. (En página 33).
- Lawrence S. Chen, Hai Tao, Thomas S. Huang, Tsutomu Miyasato y Ryohei Nakatsu. Emotion recognition from audiovisual information. En *Multimedia Signal Processing, 1998 IEEE Second Workshop on*, páginas 83–88, dic. 1998. doi: 10.1109/MMSP.1998.738917. (En página 33).
- Sungzoon Cho, Chigeun Han, Dae Hee Han y Hyung-Il Kim. Web based keystroke dynamics identity verification using neural network. *Journal of Organizational Computing and Electronic Commerce*, 10(4):295–307, 2000. (En páginas 41, 42, 45, 49, 51, 55 y 56).
- Germán M. Concilio. *Aplicación de técnicas de modelado de la cadencia de tecleo para fortalecer la identificación del usuario en el comercio electrónico*. Tesis de Grado, Facultad de Ingeniería, UBA,

Paseo Colón 850 (CABA) Argentina, feb. 2019. (En páginas 8, 100, 193 y 195).

Germán M. Concilio, Enrique P. Calot, Jorge S. Ierache y Hernán D. Merlino. Application of Keystroke Dynamics Modelling Techniques to Strengthen the User Identification in the Context of E-commerce. En Armando Eduardo De Giusti, editor, *VII Workshop de Seguridad Informática (CACIC 2018, Tandil, Buenos Aires, Argentina)*, páginas 1051–1059, 2018. ISBN 978-950-658-472-6. (En página 197).

Edwin W. Cook (III), Barbara G. Melamed, Bruce N. Cuthbert, Daniel W. McNeil y Peter J. Lang. Emotional imagery and the differential diagnosis of anxiety. *Journal of Consulting and Clinical Psychology*, 56(5):734–740, 1988. ISSN 0022-006X. doi: 10.1037/0022-006x.56.5.734. (En página 31).

Nello Cristianini y John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000. ISBN 978-0-521-78019-3. (En página 53).

Willem G. de Ru y Jan H. P. Eloff. Enhanced password authentication through fuzzy logic. *IEEE Expert*, 12(6):38–45, nov. 1997. ISSN 0885-9000. doi: 10.1109/64.642960. (En páginas 41 y 42).

Liyanage C. De Silva y Pei Chi Ng. Bimodal emotion recognition. En *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, páginas 332–335, 2000. doi: 10.1109/AFGR.2000.840655. (En página 32).

Liyanage C. De Silva, Tsutomu Miyasato y Ryohei Nakatsu. Facial emotion recognition using multi-modal information. En *Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on*, volume 1, páginas 397–401, sep. 1997. doi: 10.1109/ICICS.1997.647126. (En página 32).

Vivek Dhakal, Anna Feit, Per Ola Kristensson y Antti Oulasvirta. Observations on Typing from 136 Million Keystrokes. En *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, 2018. doi: 10.1145/3173574.3174220. (En páginas 84 y 199).

Ed Diener, Randy J. Larsen, Steven Levine y Robert A. Emmons. Intensity and frequency: Dimensions underlying positive and negative affect. *Journal of Personality and Social Psychology*, 48(5):1253–

- 1265, mayo 1985. ISSN 0022-3514. doi: 10.1037/0022-3514.48.5.1253. (En página 23).
- Ruo-Nan Duan, Jia-Yi Zhu y Bao-Liang Lu. Differential entropy feature for EEG-based emotion classification. En *6th International IEEE/EMBS Conference on Neural Engineering (NER)*, páginas 81–84. IEEE, Institute of Electrical and Electronics Engineers (IEEE), 2013. (En página 27).
- Paul Ekman. Strong evidence for universals in facial expressions: A reply to Russell's mistaken critique. *Psychological Bulletin*, 115 (2):268–287, mar. 1994. ISSN 0033-2909. doi: 10.1037/0033-2909.115.2.268. (En página 20).
- Paul Ekman y Wallace V. Friesen. *Unmasking the Face: A Guide to Recognizing Emotions from Facial Clues*. Malor Books, 2003. ISBN 978-1-883536-36-7. URL <https://books.google.com.ar/books?id=TukNoJDgMTUC>. (En página 20).
- Paul Ekman y Maureen O'Sullivan. The role of context in interpreting facial expression: Comment on Russell and Fehr (1987). *Journal of Experimental Psychology: General*, 117(1):86–88, mar. 1988. ISSN 0096-3445. doi: 10.1037/0096-3445.117.1.86. (En página 20).
- Syed Masum Emran y Nong Ye. Robustness of canberra metric in computer intrusion detection. En *Proc. IEEE Workshop on Information Assurance and Security, West Point, NY, USA*, páginas 80–84. IEEE Security & Privacy 5, jun. 2001. ISBN 978-0-7803-9814-6. (En página 137).
- Syed Masum Emran y Nong Ye. Robustness of chi-square and canberra distance metrics for computer intrusion detection. *Quality and Reliability Engineering International*, 18(1):19–28, mar. 2002. ISSN 0748-8017. doi: 10.1002/qre.441. (En página 137).
- Clayton Epp, Michael Lippold y Regan L. Mandryk. Identifying emotional states using keystroke dynamics. En *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11*, páginas 715–724, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979046. (En páginas 2, 61, 62, 63, 67 y 69).
- Lisa A. Feldman. Valence focus and arousal focus: Individual differences in the structure of affective experience. *Journal of personality and social psychology*, 69(1):153, 1995. (En páginas 22, 23 y 183).

- Eric Flior y Kazimierz Kowalski. Continuous biometric user authentication in online examinations. *2014 11th International Conference on Information Technology: New Generations*, 0:488– 492, abr. 2010. doi: 10.1109/ITNG.2010.250. (En páginas 2, 37, 42 y 45).
- Johnny R. J. Fontaine, Klaus R. Scherer, Etienne B. Roesch y Phoebe C. Ellsworth. The world of emotions is not twodimensional. *Psychological science*, 18(12):1050–1057, 2007. (En páginas 21 y 22).
- Nico H. Frijda. *The Emotions*. Studies in Emotion and Social Interaction. Cambridge University Press, 1986. ISBN 978-0-521-31600-2. URL <https://books.google.com.ar/books?id= QkNuuVf-pBMC>. (En página 21).
- R. Stockton Gaines, William Lisowski, S. James Press y Norman Shapiro. Authentication by keystroke timing: Some preliminary results. Reporte técnico, DTIC Document, 1980. (En páginas 9 y 49).
- Hernan F. Garcia, Mauricio A. Alvarez y Alvaro A. Orozco. Gaussian process dynamical models for multimodal affect recognition. En *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Institute of Electrical and Electronics Engineers (IEEE), ago. 2016. ISBN 978-1-4577-0220-4. doi: 10.1109/embc.2016.7590834. (En páginas 27 y 72).
- Romain Giot, Mohamad El-Abed y Christophe Rosenberger. Greyc keystroke: a benchmark for keystroke dynamics biometric systems. En *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*, páginas 1–6. IEEE, 2009. (En páginas 198 y 199).
- Romain Giot, Mohamad El-Abed, Baptiste Hemery y Christophe Rosenberger. Unconstrained keystroke dynamics authentication with shared secret. *Computers & Security*, 30(6):427–445, 2011. (En página 66).
- Romain Giot, Mohamad El-Abed y Christophe Rosenberger. Web-based benchmark for keystroke dynamics biometric systems: A statistical analysis. En *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on*, páginas 11–15. IEEE, 2012. (En página 69).
- Romain Giot, Bernadette Dorizzi y Christophe Rosenberger. A review on the public benchmark databases for static keystroke dynamics. *Computers & Security*, 55:46–61, 2015. (En página 103).

- Malcolm Gladwell y Madelon E. Ruiter. *Blink: The power of thinking without thinking*. Little, Brown, 2008. ISBN 978-0-31600504-3. (En página 9).
- Nahuel González. Utilización de contextos finitos para el modelado de la dinámica de tecleo en esquemas de autenticación mixta. Internal paper from Laboratorio de Sistemas de Información Avanzada, Facultad de Ingeniería, Universidad de Buenos Aires, Ciudad Autónoma de Buenos Aires, Argentina, abr. 2014a. URL <http://lsia.fi.uba.ar/papers/gonzalez14.pdf>. (En páginas 37 y 41).
- Nahuel González. Finite context modelling of keystroke dynamics in mixed authentication schemes. Internal paper from Laboratorio de Sistemas de Información Avanzada, Facultad de Ingeniería, Universidad de Buenos Aires, Ciudad Autónoma de Buenos Aires, Argentina, jun. 2014b. URL <http://lsia.fi.uba.ar/papers/gonzalez14b.pdf>. (En página 19).
- Nahuel González. *Utilización de contextos finitos para el modelado de cadencias de tecleo en esquemas de autenticación mixta*. Tesis de Grado, Facultad de Ingeniería, UBA, Paseo Colón 850 (CABA) Argentina, feb. 2016. (en páginas 8 y 17).
- Nahuel González y Enrique P. Calot. Finite context modeling of keystroke dynamics in free text. En *Biometrics Special Interest Group (BIOSIG), 2015 International Conference of the*, páginas 1–5, sep. 2015. ISBN 978-3-88579-639-8. doi: 10.1109/BIOSIG.2015.7314606. (En página 45).
- Nahuel González, Enrique P. Calot, Ariel Liguori, Francisco Pirra, Juan Manuel Rodríguez, Gustavo Pereira, Facundo Nervo, Juan Iribarren y Jorge S. Ierache. Educación de dinámica de tecleo centrado en el contexto emocional de un individuo. En *Proceedings del XVII Workshop de Investigadores en Ciencias de la Computación (Salta, 2015)*, página 5, jun. 2015. URL <http://hdl.handle.net/10915/46260>.
- Nahuel González, Enrique P. Calot y Jorge S. Ierache. A replication of two free text keystroke dynamics experiments under harsher conditions. En *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, páginas 1–6, sep. 2016. doi: 10.1109/BIOSIG.2016.7736905. (En páginas 2, 9, 38, 83, 84, 103 y 114).

- Cyril H. Goulden. *Methods of statistical analysis*. chapter 4, páginas 50–55. John Wiley & Sons, Inc., New York, 2 edition, mayo 1956. (En página 117).
- Didier Grandjean, David Sander y Klaus R. Scherer. Conscious emotional experience emerges as a function of multilevel, appraisal-driven response synchronization. *Consciousness and cognition*, 17(2):484–495, 2008. (En páginas 20 y 21)
- Paul E. Greenbaum, Clara Turner, Edwin W. Cook (III) y Barbara G. Melamed. Dentists' voice control: Effects on children's disruptive and affective behavior. *Health Psychology*, 9 (5):546–558, 1990. ISSN 0278-6133. doi: 10.1037/0278-6133.9.5. 546. (En página 31).
- Mark K. Greenwald, Edwin W. Cook (III) y Peter J. Lang. Affective judgment and psychophysiological response: Dimensional covariation in the evaluation of pictorial stimuli. *Journal of psychophysiology*, 3(1):51–64, 1989. (En página 31).
- Hatice Gunes, Björn Schuller, Maja Pantic y Roddy Cowie. Emotion representation, analysis and synthesis in continuous space: A survey. En *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, páginas 827–834. IEEE, mar. 2011. doi: 10.1109/FG.2011.5771357. (En páginas 19 y 20).
- Daniele Gunetti y Claudia Picardi. Keystroke analysis of free text. *ACM Transactions on Information and System Security (TISSEC)*, 8(3):312–347, ago. 2005. ISSN 1094-9224. doi: 10.1145/1085126.1085129. (En páginas 10, 42, 45, 50, 69, 93, 94, 113, 114, 128, 137 y 138).
- Daniele Gunetti, Claudia Picardi y Giancarlo Ruffo. Keystroke analysis of different languages: A case study. En *Advances in Intelligent Data Analysis VI*, páginas 133–144. Springer, 2005. (En página 68).
- Sajjad Haider, Ahmed Abbas y Abbas K. Zaidi. A multitechnique approach for user identification through keystroke dynamics. En *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 2, páginas 1336–1341 vol.2. IEEE, 2000. doi: 10.1109/ICSMC.2000.886039. (En páginas 42, 45, 52, 55, 57 y 60).
- Alfons O. Hamm, Jutta Globisch, Bruce N. Cuthbert y Dieter Vaitl. Startle reflex modulation in simple phobics and normals. *Psychophysiology*, 28:228, 1991. (En página 31).

- Tegan Harrison. *The Emotiv mind: Investigating the accuracy of the Emotiv EPOC in identifying emotions and its use in an Intelligent Tutoring System*. Tesis de Grado, Department of Computer Science and Software Engineering, University of Canterbury, 2013. (en páginas 26 y 34).
- Anne-Wil Harzing. Publish or perish, 2007. Disponible desde <https://harzing.com/resources/publish-or-perish>, version 6.45.6354 (19 de enero 2019). (En página 195).
- Stefan Hoch, Frank Althoff, Gregor McGlaun y Gerhard Rigoll. Bimodal fusion of emotional data in an automotive environment. En *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 2, páginas 1085–1088, mar. 2005. doi: 10.1109/ICASSP.2005.1415597. (En página 33).
- Joseph L. Hodges (Jr) y Erich L. Lehmann. Estimates of location based on rank tests. *The Annals of Mathematical Statistics*, 34(2): 598–611, 1963. ISSN 00034851. doi: 10.1214/aoms/1177704172. (En páginas 88 y 159).
- Jiankun Hu, Don Gingrich y Andy Sentosa. A k-nearest neighbor approach for user authentication through biometric keystroke dynamics. En *Communications, 2008. ICC'08. IEEE International Conference on*, páginas 1556–1560. IEEE, 2008. (En páginas 45 y 50).
- Earl B. Hunt, Janet Marin y Philip J. Stone. *Experiments in induction*. Academic Press, Oxford, England, 1966. ISBN 978-1-55860-238-0. (En página 153).
- Jorge S. Ierache, Gustavo Pereira, Enrique P. Calot y Juan Iribarren. Framework for brain computer interface implemented to control devices in the context of home automation. En *XIX Congreso Argentino de Ciencias de la Computación*, number 5736, 2013a. ISBN 978-987-23963-1-2. URL <http://Isia.fi.uba.ar/papers/ierache13b.pdf>. (En páginas 28 y 183).
- Jorge S. Ierache, Gustavo Pereira, Iris Sattolo y Juan Iribarren. Aplicación de interfases lectoras de bioseñales en el contexto de la domótica. En *XV Workshop de Investigadores en Ciencias de la Computación*, 2013b. ISBN 978-987-23963-1-2. URL <http://Isia.fi.uba.ar/papers/ierache13.pdf>. (En página 183).

- Jorge S. Ierache, Gustavo Pereira, Iris Sattolo, Juan Iribarren y Nicolás Suiffet. Robot control on the basis of bio-electrical signals. En Jong-Hwan Kim, Eric T. Matson, Hyun Myung y Peter Xu, editors, *Robot Intelligence Technology and Applications 2012*, volume 208 de *Advances in Intelligent Systems and Computing*, páginas 337–346. Springer Berlin Heidelberg, 2013c. ISBN 978-3-642-37373-2. doi: 10.1007/978-3-642-37374-9_33. (En página 28).
- Jorge S. Ierache, Gustavo Pereira y Juan Iribarren. Navigation control of a robot from a remote location via the internet using brain-machine interface. En Jong-Hwan Kim, Eric T. Matson, Hyun Myung, Peter Xu y Fakhri Karray, editors, *Robot Intelligence Technology and Applications 2*, volume 274 de *Advances in Intelligent Systems and Computing*, páginas 297–310. Springer International Publishing, 2014. ISBN 978-3-319-05581-7. doi: 10.1007/978-3-319-05582-4_26. (En página 28).
- Jorge S. Ierache, Gustavo Pereira, Facundo Nervo y Juan Iribarren. Emotional status focused on stimuli by applying brain-machine interface. En Guillermo Eugenio Feierherd, Patricia Pesado y Osvaldo Mario Sposito, editors, *Computer Science & Technology Series. XX Argentine Congress of Computer Science, Selected papers*, páginas 245–256. Editorial de la Universidad Nacional de La Plata (EDULP), 2015. ISBN 978-987-1985-71-5. (En páginas 28, 76 y 183).
- Blake Ives, Kenneth R Walsh y Helmut Schneider. The domino effect of password reuse. *Communications of the ACM*, 47(4): 75–78, abr. 2004. ISSN 0001-0782. doi: 10.1145/975817.975820. (En página 1).
- Anil K. Jain, Arun Ross y Sharath Pankanti. Biometrics: A tool for information security. *IEEE Transactions on Information Forensics and Security*, 1(2):125–143, 2006. (En página 1).
- Anil K. Jain, Brendan Klare y Arun Ross. Guidelines for best practices in biometrics research. En *Proceedings of 8th IAPR International Conference on Biometrics*, páginas 541–545. IEEE, mayo 2015. ISBN 978-1-4799-7824-3. doi: 10.1109/ICB.2015.7139116. (En páginas 101 y 102).
- H. H. Jasper. Report of the committee on methods of clinical examination in electroencephalography. *Electroencephalography and Clinical Neurophysiology*, 10(2):370–375, 2017/07/18 1958.

ISSN 0013-4694. doi: 10.1016/0013-4694(58)90053-1. (En página 183).

- Jia Jia, Shen Zhang, Fanbo Meng, Yongxin Wang y Lianhong Cai. Emotional audio-visual speech synthesis based on pad. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(3):570–582, 2011. (En página 20).
- Rick Joyce y Gopal Gupta. Identity authentication based on keystroke latencies. *Commun. ACM*, 33(2):168–176, feb. 1990. ISSN 0001-0782. doi: 10.1145/75577.75582. (En páginas 43, 45 y 66).
- Preeti Khanna y Mukundan Sasikumar. Recognising emotions from keyboard stroke pattern. *International journal of computer applications*, 11(9):1–5, dic. 2010. (En página 64).
- Kevin S. Killourhy. A scientific understanding of keystroke dynamics. Reporte técnico, Carnegie Institute of Technology, Department of Computer Science, 2012. URL <https://apps.dtic.mil/dtic/tr/fulltext/u2/a557327.pdf>. (En páginas 49, 66, 67, 68 y 102).
- Kevin S. Killourhy y Roy A. Maxion. Comparing anomalydetection algorithms for keystroke dynamics. En *International Conference on Dependable Systems & Networks (DSN-09)*, páginas 125–134. IEEE Computer Society Press, Los Alamitos, California, 2009. (En páginas 16, 41, 42, 45, 46, 47, 48, 49, 50, 52, 56, 58 y 138).
- Kevin S. Killourhy y Roy A. Maxion. Why did my detector do that?! En *Recent Advances in Intrusion Detection*, páginas 256–276. Springer, 2010. ISBN 978-3-642-15511-6. doi: 10.1007/978-3-642-15512-3_14. (En páginas 67, 103 y 199).
- Kevin S. Killourhy y Roy A. Maxion. Should security researchers experiment more and draw more inferences? En *Proceedings of the 4th Conference on Cyber Security Experimentation and Test, CSET'11*, Berkeley, CA, USA, 2011. USENIX Association. URL <http://dl.acm.org/citation.cfm?id=2027999.2028000>. (En páginas 101 y 102).
- Kevin S. Killourhy y Roy A. Maxion. Free vs. transcribed text for keystroke-dynamics evaluations. En *Proceedings of the 2012 Workshop on Learning from Authoritative Security Experiment Results*, páginas 1–8, 2012. (En página 199).
- David Kim, Michael G. Solomon *et al.* *Fundamentals of information systems security*. Jones & Bartlett Publishers, 2013. ISBN 978-1-284-03162-1. (En páginas 10 y 12).

- Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt y Ioannis Patras. Deap: A database for emotion analysis ;using physiological signals. *IEEE Transactions on Affective Computing*, 3(1):18–31, ene. 2012. ISSN 1949-3045. doi: 10.1109/T-AFFC.2011.15. (En página 27).
- Agata Kołakowska. A review of emotion recognition methods based on keystroke dynamics and mouse movements. En *Human System Interaction (HSI), 2013 The 6th International Conference on*, páginas 548–555, jun. 2013. doi: 10.1109/HSI.2013.6577879. (En páginas 61 y 62).
- Kentaro Kotani y Ken Horii. Evaluation on a keystroke authentication system by keying force incorporated with temporal characteristics of keystroke dynamics. *Behaviour & Information Technology*, 24(4):289–302, 2005. (En página 70).
- Sen Tarng Lai, Fang Yie Leu y William Cheng Chung Chu. A multi-layer secure prevention scheme for improving e-commerce security. En *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2014 Eighth International Conference on*, páginas 505–510, jul. 2014. doi: 10.1109/IMIS.2014.73. (En página 2).
- Zirui Lan, Olga Sourina, Lipo Wang y Yisi Liu. Real-time EEG-based emotion monitoring using stable features. *The Visual Computer*, 32(3):347–358, mar. 2016. ISSN 1432-2315. doi: 10.1007/s00371-015-1183-y. (En página 28).
- Godfrey N. Lance y William T. Williams. Computer programs for hierarchical polythetic classification (“similarity analyses”). *The Computer Journal*, 9(1):60–64, mayo 1966. doi: 10.1093/comjnl/9.1.60. (En página 137).
- Godfrey N. Lance y William T. Williams. Mixed-Data Classificatory Programs I – Agglomerative Systems. *Australian Computer Journal*, 1(1):15–20, ene. 1967. (En página 137).
- Peter J. Lang. Behavioral treatment and bio-behavioral assessment: computer applications. En Joseph B. Sidowski, James H. Johnson y Thomas A. Williams, editors, *Technology in mental health care delivery systems*, páginas 119–137. Ablex, Norwood, NJ, 1980. ISBN 978-0-89391-023-5. (En páginas 28, 34, 115 y 182).
- Peter J. Lang. The emotion probe: studies of motivation and attention. *American psychologist*, 50(5):372, 1995. (En página 22).

- Peter J. Lang, Mark K. Greenwald, Margaret M. Bradley y Alfons O. Hamm. Looking at pictures: Affective, facial, visceral, and behavioral reactions. *Psychophysiology*, 30(3):261–273, 1993. (En página 31).
- Peter J. Lang, Margaret M. Bradley y Bruce N. Cuthbert. International affective picture system (IAPS): Affective ratings of pictures and instruction manual. *Technical report A-8*, 2008. (En páginas 28, 34, 35, 75, 76, 115 y 182).
- Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel H. J. Wigboldus, Skyler T. Hawk y Ad van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and Emotion*, 24(8):1377–1388, 2010. doi: 10.1080/02699930903485076. (En página 186).
- Randy J. Larsen y Susan E. Cutler. The complexity of individual emotional lives: A process analysis of affect structure. 1992. (En páginas 22 y 23).
- Randy J. Larsen y Susan E. Cutler. The complexity of individual emotional lives: A within-subject analysis of affect structure. *Journal of Social and Clinical Psychology*, 15(2):206–230, jun. 1996. ISSN 0736-7236. doi: 10.1521/jscp.1996.15.2.206. (En página 22).
- Hosub Lee, Young Sang Choi, Sunjae Lee y I.P. Park. Towards unobtrusive emotion recognition for affective social communication. En *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, páginas 260–264. IEEE, ene. 2012. doi: 10.1109/CCNC.2012.6181098. (En página 63).
- John Leggett y Glen Williams. Verifying identity via keystroke characteristics. *International Journal of Man-Machine Studies*, 28(1):67–76, 1988. (En página 42).
- Xiang Li, Dawei Song, Peng Zhang, Guangliang Yu, Yuexian Hou y Bin Hu. Emotion recognition from multi-channel EEG data through convolutional recurrent neural network. En *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. Institute of Electrical and Electronics Engineers (IEEE), dic. 2016. ISBN 978-1-5090-1611-2. doi: 10.1109/bibm.2016.7822545. (En páginas 27 y 72).
- Robert Lievesley, Martin Wozencroft y David Ewins. The Emotiv EPOC neuroheadset: an inexpensive method of controlling assistive technologies using facial expressions and thoughts? *Journal of*

Assistive Technologies, 5(2):67–82, 2011. doi: 10.1108/17549451111149278. (En página 27).

Jingxin Liu, Hongying Meng, Asoke Nandi y Maozhen Li. Emotion detection from eeg recordings. En *12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, páginas 2173–2178, 2016. URL <http://bura.brunel.ac.uk/handle/2438/13138>. (En páginas 27 y 72).

Hai-Rong Lv, Zhong-Lin Lin, Wen-Jun Yin y Jin Dong. Emotion recognition based on pressure sensor keyboards. En *Multimedia and Expo, 2008 IEEE International Conference on*, páginas 1089–1092, jun. 2008. doi: 10.1109/ICME.2008.4607628. (En páginas 61, 62 y 64).

Wolfgang Maehr. *eMotion: Estimation of the User's Emotional State by Mouse Motions*. Tesis Doctoral, Diploma thesis for Fachhochschule Vorarlberg, Dornbirn, Austria. iTec–Information and Communication Engineering, 2005. URL <http://www.cse.chalmers.se/research/group/idc/theses/05/pdf/Maehr.pdf>. (En página 64).

Prasanta Chandra Mahalanobis. On the generalised distance in statistics. En *Proceedings National Institute of Science, India*, volume 2, páginas 49–55, abr. 1936. URL http://www.new.dli.ernet.in/rawdataupload/upload/insa/INSA_1/20006193_49.pdf. (En páginas 41 y 140).

Ella T. Mampusti, Jose S. Ng, Jarren James I. Quinto, Grizelda L. Teng, Merlin Teodosia C. Suarez y Rhia S. Trogo. Measuring academic affective states of students via brainwave signals. *Knowledge and Systems Engineering, International Conference on*, 0:226–231, 2011. doi: 10.1109/KSE.2011.43. (En página 27).

Roy A. Maxion y Kevin S Killourhy. Keystroke biometrics with number-pad input. En *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, páginas 201–210. IEEE, 2010. (En páginas 15, 43, 45, 59 y 70).

Gary McKeown, Michel F. Valstar, Roderick Cowie y Maja Pantic. The semaine corpus of emotionally coloured character interactions. En *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, páginas 1079–1084. IEEE, 2010. (En página 21).

Daniel W. McNeil y David G. Brunetti. Pain and fear: A bioinformational perspective on responsivity to imagery. *Behaviour research and*

- therapy*, 30(5):513–520, sep. 1992. doi: 10.1016/0005-7967(92)90035-F. (En página 31).
- Albert Mehrabian. Pleasure–arousal–dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4):261–292, 1996. (En página 20).
- Arik Messerman, Tarik Mustafic, Seyit Ahmet Camtepe y Sahin Albayrak. Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics. En *Biometrics (IJCB), 2011 International Joint Conference on*, páginas 1–8. IEEE, 2011. (En páginas 42 y 43).
- Stephen Milborrow, John Morkel y Fred Nicolls. The muct landmarked face database. *Pattern Recognition Association of South Africa*, (0), 2010. URL <http://www.dip.ee.uct.ac.za/~nicolls/publish/sm10-prasa.pdf>. <http://www.milbo.org/muct>. (En página 186).
- Gregory A. Miller, Daniel N. Levin, Michael J. Kozak, Edwin W. Cook (III), Alvin McLean (Jr) y Peter J. Lang. Individual differences in imagery and the psychophysiology of emotion. *Cognition and Emotion*, 1(4):367–390, 1987. (En página 31).
- John Vincent Monaco. Sok: Keylogging side channels. En *2018 IEEE Symposium on Security and Privacy (SP)*, páginas 211–228. IEEE, mayo 2018. ISBN 978-1-5386-4353-2. doi: 10.1109/SP.2018.00026. (En páginas 84 y 87).
- Fabian Monroe y Aviel D. Rubin. Authentication via keystroke dynamics. En *Proceedings of the 4th ACM conference on Computer and communications security*, páginas 48–56. ACM, 1997. (En página 61).
- Fabian Monroe, Michael K. Reiter y Susanne Wetzel. Password hardening based on keystroke dynamics. En *Proceedings of the 6th ACM Conference on Computer and Communications Security, CCS '99*, páginas 73–82, New York, NY, USA, 1999. ACM. ISBN 978-1-58113-148-2. doi: 10.1145/319709.319720. (En páginas 2, 37 y 45).
- Jon D. Morris, Margaret M. Bradley, C. A. Wayne y J. B. Lang. Assessing affective reactions to advertisements with the selfassessment manikin (sam). En *Southern Marketing Association Conference*, 1992. (En página 31).

- A.F.M. Nazmul Haque Nahin, Jawad Mohammad Alam, Hasan ahmud y Kamrul Hasan. Identifying emotion by keystroke dynamics and text pattern analysis. *Behaviour & Information Technology*, 33(9):987–996, 2014. doi: 10.1080/0144929X.2014. 907343. (En páginas 62 y 64).
- Mohammad S. Obaidat y Balqies Sadoun. Verification of computer users using keystroke dynamics. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 27(2): 261–269, abr. 1997. ISSN 1083-4419. doi: 10.1109/3477.558812. (En páginas 41, 42, 45, 51 y 57).
- Lawrence O’Gorman. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12): 2021–2040, dic. 2003. ISSN 0018-9219. doi: 10.1109/JPROC. 2003.819611. (En página 1).
- Andrew Ortony, Gerald L. Clore y Allan Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, mayo 1990. ISBN 978-0-521-38664-7. URL <http://www.cambridge.org/us/academic/subjects/Referenciasbibliograficaspsychology/cognition/cognitiv e-structure-emotions>. (En página 22).
- Danny Oude Bos. EEG-based emotion recognition-The Influence of Visual and Auditory Stimuli. *Capita Selecta (MSc course)*, 2006. (En páginas 8, 72 y 183).
- Christopher J. Patrick, Margaret M. Bradley y Peter J. Lang. Emotion in the criminal psychopath: Startle reflex modulation. *Journal of Abnormal Psychology*, 102(1):82–92, 1993. ISSN 0021-843X. doi: 10.1037/0021-843x.102.1.82. (En página 31).
- Gustavo Pereira, Juan D’Alotto, Iris Sattolo, Alejandro Guerrero, Adolfo Marcelo Iribarren y Jorge S. Ierache. Control vía internet de un robot ubicado en un sitio remoto aplicando una interfase cerebro-máquina. En *XVII Congreso Argentino de Ciencias de la Computación*, 2011. URL <http://isia.fi.uba.ar/papers/pereira11.pdf>. (En página 28).
- Gustavo Pereira, Facundo Nervo, Juan Iribarren, Enrique P. Calot, Jorge S. Ierache y Norberto Mazza. Líneas de investigación aplicada al empleo de interfases cerebro-maquina. En *Proceedings del XVII Workshop de Investigadores en Ciencias de la Computación (Salta, 2015)*, página 5, jun. 2015. URL <http://hdl.handle.net/10915/46146>. (En página 28).

- Humberto Pérez Espinosa, Carlos A. Reyes García y Luis Villaseñor Pineda. Features selection for primitives estimation on emotional speech. En *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, páginas 5138–5141. IEEE, 2010. (En página 20).
- Panagiotis C. Petrantonakis y Leontios J. Hadjileontiadis. Emotion recognition from brain signals using hybrid adaptive filtering and higher order crossings analysis. *IEEE Transactions on Affective Computing*, 1(2):81–97, jul. 2010. ISSN 1949-3045. doi: 10.1109/t-affc.2010.7. (En páginas 28 y 72).
- Steven Pinker. *The language instinct: The new science of language and mind*, volume 7529. Penguin UK, 1994. ISBN 978-0-14198077-5. (En página 140).
- Robert Plutchik. *Emotions in the practice of psychotherapy: Clinical implications of affect theories*. American Psychological Association, 2000. ISBN 978-1-55798-694-8. doi: 10.1037/10366-000. (En página 20).
- Philip M. Podsakoff, Scott B. MacKenzie, Jeong-Yeon Lee y Nathan P. Podsakoff. Common method biases in behavioral research: a critical review of the literature and recommended remedies. *Journal of applied psychology*, 88(5):879, 2003. (En página 67).
- Despina Polemi. Biometric techniques: review and evaluation of biometric techniques for identification and authentication, including an appraisal of the areas where they are most applicable. *Reported prepared for the European Commission DG XIII*, 4, 1997. (En página 9).
- Jonathan Posner, James A. Russell y Bradley S. Peterson. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology*, 17(03), sep. 2005. doi: 10.1017/s0954579405050340. (En página 20).
- John Ross Quinlan. *C4.5: Programs for Machine Learning*, volume 1. Morgan Kaufmann, 1993. ISBN 978-1-55860-238-0. (En página 153).
- John Ross Quinlan. Improved use of continuous attributes in c4.5. *Journal of artificial intelligence research*, páginas 77–90, mar. 1996. (En página 156).

- Rafael Ramirez y Zacharias Vamvakousis. Detecting emotion from eeg signals using the emotive epoc device. En *Brain Informatics*, páginas 175–184. Springer, 2012. (En páginas 88, 90, 122, 183 y 201).
- Ranier Reisenzein. Pleasure-arousal theory and the intensity of emotions. *Journal of Personality and Social Psychology*, 67(3): 525–539, 1994. ISSN 0022-3514. doi: 10.1037/0022-3514.67.3.525 (En página 23).
- Lior Rokach. *Data mining with decision trees: theory and applications*, volume 69. World scientific, 2008. ISBN 978-981-4590-07-5. (En página 58).
- Federico M. Rossi. Educación de Estados Emocionales basada en Captura Multimodal de Bioseñales. Tesis de grado. Ingeniería en Informática. Universidad de Buenos Aires, 2019. (En página 8).
- Federico M. Rossi, Enrique P. Calot y Jorge S. Ierache. Educación emocional de un individuo en contextos multimodales en computación afectiva. En *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, Buenos Aires, Argentina)*, jul. 2017. ISBN 978-987-42-5143-5. (En página 181).
- David E. Rumelhart, Geoffrey E. Hinton y Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, oct. 1986a. (En páginas 55 y 141).
- David E. Rumelhart, Geoffrey E. Hinton y Ronald J. Williams. Learning internal representations by error propagation. 1:319–362, 1986b. (En páginas 55 y 141).
- James A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980. ISSN 0022-3514. doi: 10.1037/h0077714. (En páginas 20, 22 y 23).
- James A. Russell. *Measures of emotion.*, páginas 83–111. Academic Press, 1989. ISBN 978-0-12-558704-4. (En página 23).
- Hataichanok Saevanee y Pattarasinee Bhatarakosol. User authentication using combination of behavioral biometrics over the touchpad acting like touch screen of mobile device. En *Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on*, páginas 82–86, dic. 2008. doi: 10.1109/ICCEE.2008.157. (En páginas 41 y 42).

- Hataichanok Saevanee y Pattarasinee Bhatarakosol. Authenticating user using keystroke dynamics and finger pressure. En *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, páginas 1–2. IEEE, 2009. (En página 70).
- Anas Samara, Maria Luiza Recena Menezes y Leo Galway. Feature extraction for emotion recognition and modelling using europsychological data. En *2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*. Institute of Electrical and Electronics Engineers (IEEE), dic. 2016. ISBN 978-1-5090-5566-1. doi: 10.1109/iucc-css.2016.027. (En páginas 27 y 72).
- David Sander, Didier Grandjean y Klaus R. Scherer. A systems approach to appraisal mechanisms in emotion. *Neural networks*, 18(4):317–352, 2005. (En página 21).
- Jason M. Saragih, Simon Lucey y Jeffrey F. Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, ene. 2011. ISSN 0920-5691. doi: 10.1007/s11263-010-0380-4. (En página 186).
- Klaus R. Scherer, Angela Schorr y Tom Johnstone. *Appraisal Processes in Emotion: Theory, Methods, Research*. Series in Affective Science. Oxford University Press, 2001. ISBN 978-0-19-028470-1. (En páginas 20 y 21).
- Alois Schlögl. GDF – A general dataformat for BIOSIGNALS. *CoRR*, abs/cs/0608052, ago. 2006. URL <http://arxiv.org/abs/cs/0608052>. (en página 180)
- Harold H. Schlosberg. A scale for the judgment of facial expressions. *Journal of Experimental Psychology*, 29(6):497–510, 1941. ISSN 0022-1015. doi: 10.1037/h0061489. (En página 22).
- Harold H. Schlosberg. Three dimensions of emotion. *Psychological Review*, 61(2):81–88, 1954. ISSN 0033-295X. doi: 10.1037/h0054570. (en página 23)
- Janet Ward Schofield. Increasing the generalizability of qualitative research. *The qualitative researcher's companion*, páginas 171–203, 2002. (en página 101)
- Björn Schuller, Manfred Lang y Gerhard Rigoll. Multimodal emotion recognition in audiovisual communication. En *Multimedia and Expo*,

2002. ICME '02. Proceedings. 2002 IEEE International Conference on, volume 1, páginas 745–748 vol.1. IEEE, 2002. ISBN 978-0-7803-7304-4. doi: 10.1109/ICME.2002.1035889. (En página 64)
- Samuel Sanford Shapiro y Martin B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591– 611, 1965. doi: 10.1093/biomet/52.3-4.591. (En páginas 106, 117 y 156).
- Phillip Shaver, Judith Schwartz, Donald Kirson y Cary O'connor. Emotion knowledge: further exploration of a prototype approach. *Journal of personality and social psychology*, 52(6):1061–1086, 1987. (En página 22).
- Yong Sheng, Vir Phoha y Steven M. Rovnyak. A parallel decision tree-based method for user authentication based on keystroke patterns. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 35(4):826–833, 2005. (En páginas 45, 58 y 59).
- Dongmin Shin, Dongil Shin y Dongkyoo Shin. Development of emotion recognition interface using complex EEG/ECG biosignal for interactive contents. *Multimedia Tools and Applications*, 76(9):11449–11470, dic. 2016. ISSN 1380-7501. doi: 10.1007/s11042-016-4203-7. (En páginas 28 y 72).
- Lin Shu, Jinyan Xie, Mingyue Yang, Ziyi Li, Zhenqi Li, Dan Liao, Xiangmin Xu y Xinyi Yang. A review of emotion recognition using physiological signals. *Sensors*, 18(7), 2018. ISSN 1424-8220. doi: 10.3390/s18072074. (En páginas 32 y 72).
- Pragya Shukla y Rinky Solanki. Web based keystroke dynamics application for identifying emotional state. *International Journal of Advanced Research in Computer and Communication Engineering*, 2, nov. 2013. ISSN 2319-5940. (En página 62).
- Rinky Solanki y Pragya Shukla. Article: Estimation of the user's emotional state by keystroke dynamics. *International Journal of Computer Applications*, 94(13):21–23, mayo 2014. Full text available. (En página 2).
- Mingli Song, Jiajun Bu, Chun Chen y Nan Li. Audio-visual based emotion recognition - a new approach. En *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, páginas II– 1020–II– 1025 Vol.2, jun. 2004a. doi: 10.1109/CVPR.2004.1315276. (En página 32).

- Mingli Song, Chun Chen y Mingyu You. Audio-visual based emotion recognition using tripled hidden markov model. En *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 5, páginas V–877– 80 vol.5, mayo 2004b. doi: 10.1109/ICASSP.2004.1327251. (En página 32).
- Robert A. Sottolare y Michael D. Proctor. Passively classifying student mood and performance within intelligent tutors. *Educational Technology & Society*, 15(2):101–114, 2012. (En página 63).
- Ki-seok Sung y Sungzoon Cho. Ga svm wrapper ensemble for keystroke dynamics authentication. En *Advances in Biometrics*, páginas 654–660. Springer, 2005. (En páginas 45 y 54).
- Liang Tang, Changqi Yang y Zhiwen Wu. User authentication based on force-sensing keypad by fuzzy c-means clustering method. *Journal of Computational Information Systems*, 6(11):3659–3667, 2010. (En páginas 37 y 70).
- Pin Shen Teh, Andrew Beng Jin Teoh, Thian Song Ong y Han Foon Neo. Statistical fusion approach on keystroke dynamics. En *Signal-Image Technologies and Internet-Based System, 2007. SITIS'07. Third International IEEE Conference on*, páginas 918–923. IEEE, dic. 2007. doi: 10.1109/SITIS.2007.46. (En páginas 45 y 60).
- Pin Shen Teh, Shigang Yue y Andrew Beng Jin Teoh. Improving keystroke dynamics authentication system via multiple feature fusion scheme. En *Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on*, páginas 277–282, jun. 2012. doi: 10.1109/CyberSec.2012.6246096. (En página 41).
- Robert E. Thayer. *The Biopsychology of Mood and Arousal*. Oxford University Press, sep. 1990. ISBN 978-0-19-506827-6. (En página 23).
- Georgios Tsoulouhas, Dimitrios Georgiou y Alexandros Karakos. Detection of learner's affective state based on mouse movements. *Journal of Computing*, 3(11):9–18, nov. 2011. ISSN 2151-9617. URL <http://journalofcomputing.org/volume-3-issue-11november2011/>. (En página 64).
- Gaetano Valenza, Antonio Lanata y Enzo Pasquale Scilingo. The role of nonlinear dynamics in affective valence and arousal recognition. *IEEE Transactions on Affective Computing*, 3(2):237– 249, abr. 2012. ISSN 1949-3045. doi: 10.1109/T-AFFC.2011.30. (En página 31).

- Gaetano Valenza, Luca Citi, Antonio Lanatá, Enzo Pasquale Scilingo y Riccardo Barbieri. Revealing real-time emotional responses: a personalized assessment based on heartbeat dynamics. *Scientific Reports*, 4, 05 2014. (En página 31).
- Gyanendra K. Verma y Uma Shanker Tiwary. Multimodal fusion framework: A multiresolution approach for emotion classification and recognition from physiological signals. *NeuroImage*, 102: 162–172, nov. 2014. doi: 10.1016/j.neuroimage.2013.11.007. (En páginas 28 y 72).
- Lisa M. Vizer, Lina Zhou y Andrew Sears. Automated stress detection using keystroke and linguistic features: An exploratory study. *International Journal of Human-Computer Studies*, 67(10): 870–886, 2009. ISSN 1071-5819. doi: 10.1016/j.ijhcs.2009.07.005. (En páginas 61, 63 y 67).
- Yongjin Wang y Ling Guan. Recognizing human emotional state from audiovisual signals. *IEEE Transactions on Multimedia*, 10 (5):936–946, ago. 2008. ISSN 1520-9210. doi: 10.1109/TMM.2008.927665. (En página 32).
- David Watson y Auke Tellegen. Toward a consensual structure of mood. *Psychological bulletin*, 98(2):219–235, 1985. ISSN 0033-2909. doi: 10.1037/0033-2909.98.2.219. (En páginas 22 y 23).
- Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945. doi: 10.2307/3001968. (En páginas 88 y 117).
- Jonathan R. Wolpaw y Dennis J. McFarland. Control of a twodimensional movement signal by a noninvasive brain-computer interface in humans. En *Proceedings of the National Academy of Sciences of the United States of America*, volume 101, páginas 17849–17854. National Academy of Sciences, dic. 2004. doi: 10.1073/pnas.0403504101. (En página 25).
- Jonathan R. Wolpaw, Dennis J. McFarland, Gregory W. Neat y Catherine A. Forneris. An EEG-based brain-computer interface for cursor control. *Electroencephalography and clinical neurophysiology*, 78(3):252–259, mar. 1991. ISSN 0013-4694. URL <http://view.ncbi.nlm.nih.gov/pubmed/1707798>. (En página 25).
- Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller y Theresa M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*,

- 113(6):767–791, 2002. doi: 10.1016/S1388-2457(02)00057-3. (En página 25).
- Jacob Y. Wong y Roy L. Anderson. System for secured credit card transactions on the internet, sep. 1999. URL <https://www.google.com.ar/patents/US5956699>. US Patent 5,956,699. (En página 1).
- Wilhelm Wundt. *Grundriss der Psychologie*. Wilhelm Engelmann, Leipzig, 1896. (En página 22).
- Enzhe Yu y Sungzoon Cho. GA-SVM wrapper approach for feature subset selection in keystroke dynamics identity verification. En *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 3, páginas 2253–2257 vol.3, jul. 2003. doi: 10.1109/IJCNN.2003.1223761. (En páginas 43, 45 y 53).
- Erim Yurci. Emotion detection from eeg signals: Correlating cerebral cortex activity with music evoked emotion. *Universitat Pompeu Fabra, Barcelona, 2014*. (En páginas 26 y 88).
- Zhihong Zeng, Jilin Tu, Ming Liu, Tong Zhang, Nicholas Rizzolo, Zhenqiu Zhang, Thomas S. Huang, Dan Roth y Stephen Levinson. Bimodal hci-related affect recognition. En *Proceedings of the 6th International Conference on Multimodal Interfaces, ICMI '04*, páginas 137–143, New York, NY, USA, 2004. ACM. ISBN 1-58113-995-0. doi: 10.1145/1027933.1027958. (En página 33).
- Zhihong Zeng, Jilin Tu, Brian Pianfetti, Ming Liu, Tong Zhang, Zhenqiu Zhang, Thomas S. Huang y Stephen Levinson. Audiovisual affect recognition through multi-stream fused hmm for hci. En *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, páginas 967–972, jun. 2005. doi: 10.1109/CVPR.2005.77. (En página 33).
- Jianhai Zhang, Ming Chen, Sanqing Hu, Yu Cao y Robert Kozma. PNN for EEG-based emotion recognition. En *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Institute of Electrical and Electronics Engineers (IEEE), oct. 2016. ISBN 978-1-5090-1897-0. doi: 10.1109/smc.2016.7844584. (En páginas 27 y 72)
- Wei-Long Zheng y Bao-Liang Lu. Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. *IEEE Transactions on Autonomous Mental Development*, 7(3):162–175, 2015. doi: 10.1109/TAMD.2015.2431497. (En página 27).

- Wei-Long Zheng, Jia-Yi Zhu y Bao-Liang Lu. Identifying stable patterns over time for emotion recognition from EEG. *IEEE Transactions on Affective Computing*, 2017. ISSN 1949-3045. doi: 10.1109/taffc.2017.2712143. (En páginas 27 y 72).
- Yu Zhong, Yunbin Deng y Anil K. Jain. Keystroke dynamics for user authentication. En *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, páginas 117–123. IEEE, jun. 2012. ISBN 978-1-4673-1611-8. doi: 10.1109/CVPRW.2012.6239225. (En página 43).
- Philippe G. Zimmermann, Patrick Gomez, Brigitta Danuser y Sissel Guttormsen Schär. Extending usability: putting affect into the user-experience. *Proceedings of NordiCHI'06*, páginas 27–32, 2006. URL <https://www.irit.fr/recherches/ICS/projects/cost294/upload/408.pdf>. (En página 63).
- Moshe Zviran y William J. Haga. Password security: an empirical study. *Journal of Management Information Systems*, 15(4):161–185, 1999. (En página 1).

Lista de Acrónimos

A	Absolute (por la métrica A).
A	Aceptada (por la hipótesis).
A/V	Arousal/Valence • Modelo de Excitación/Valencia.
ACC	Accuracy • Exactitud.
API	Application Programming Interface • Interfaz de programación de Aplicaciones.
ARTMAP	Adaptive Resonance Theory MAP.
ASCII	American Standard Code for Information Interchange.
BCI	Brain-Computer Interface • Interfaz Cerebro– Computador.
BMI	Brain-Machine Interface • Interfaz Cerebro– Máquina (ver BCI).
C	Camberra (por la distancia de Camberra)
CDF	Cumulative Density Function • Función Acumulativa de Densidad.
CPU	Central Processing Unit • Unidad Central de Procesamiento.
CV	Computer Vision • Visión por Computadora.

DET	Detection Error Tradeoff.
DTW	Dynamic Time Warping • Alineamiento Temporal Dinámico.
ECG	Electrocardiografía.
EEG	Electroencefalografía.
EER	Equal Error Rate.
EIC	Emotion Immersion Chamber • Cámara de Inmersión Emocional.
EMG	Electromiografía.
EN	European Norm.
EOG	Electrooculografía.
EPOC	No es un acrónimo. Dispositivo BCI de Emotiv.
FAR	False Acceptance Rate • Tasa de Falsos Positivos.
FDR	False Discovery Rate/tasa de descubrimiento falso.
FMR	False Match Rate.
FN	Falso Negativo.
FNMR	False Non-Match Rate.
FOR	False Omission Rate.
FP	Falso Positivo.
FRR	False Rejection Rate • Tasa de Falsos Negativos.
FT	Flight Time.
GNU	GNU is Not Unix.
HANV	High Arousal Negative Valence.
HMM	Hidden Markov Model • Modelo Oculto de Márkov.
HT	Hold Time.

HTML5	HyperText Markup Language version 5.
IADS	International Affective Digitized Sounds.
IAPS	International Affective Picture System.
ID3	Interactive Dichotomizer 3.
k-NN	k-Nearest Neighbour.
KD	Keystroke Dynamics.
LSIA	Laboratorio de Sistemas de Información Avanzados (por los dataset producidos allí).
LVQ	Learning Vector Quantization • Aprendizaje de Cuantificación Vectorial.
MUCT	Milborrow/University of Cape Town (por MUCT Landmarked Face Database).
NA	No Aceptada (hipótesis).
NPV	Negative Predictive Value.
OCC	Ortony, Clore & Collins (modelo de estructura cognitiva de emociones).
OGA	Ogg Vorbis Audio File.
PC	Personal Computer.
PPV	Positive Predictive Value.
PtT	Paired t-Test • t-Test pareado.
R	Relative (por la métrica R).
RAM	Random Access Memory • Memoria de Acceso Aleatorio.
RBFM	Randomized Best-First Minimax.
ROC	Receiver Operating Characteristic.
SAM	Self-Assessment Manikin.
SDK	Software Development Kit.

SRT	Wilcoxon Signed Rank Test.
SVM	Support-vector Machines • Máquinas de Vectores de Soporte.
SWT	Shapiro–Wilk Test.
TN	True Negative.
TNR	True Negative Rate • Tasa de Negativos Auténticos.
TP	True Positive.
TPR	True Positive Rate • Tasa de Positivos Auténticos.
UBA	Universidad de Buenos Aires.
UNLP	Universidad Nacional de La Plata.
USB	Universal Serial Bus.
VHDR	BrainVision Analyzer Header Data.
VMRK	BrainVision Analyzer Marker Data.
WER	Weighted Error Rate.
WT	Wait Time.



Esta red se constituyó formalmente en noviembre de 1996 y actualmente 51 universidades argentinas son miembros activos. Sus objetivos son:

“Coordinar actividades académicas relacionadas con el perfeccionamiento docente, la actualización curricular y la utilización de recursos compartidos en el apoyo al desarrollo de las carreras de Ciencia de la Computación y/o Informática en Argentina”.

“Establecer un marco de colaboración para el desarrollo de las actividades de posgrado en Ciencia de la Computación y/o Informática de modo de optimizar la asignación y el aprovechamiento de recursos”.

La dinámica de tecleo permite la identificación de una persona por la forma en que escribe. Esta tesis se enfoca en analizar la robustez de los algoritmos de análisis de cadencia de tecleo frente a variaciones en los registros biométricos mediante electroencefalografía y cuestionario de autoinforme, utilizando el enfoque dimensional para modelizar estados afectivos. Se realizó un experimento para capturar patrones de tecleo en diferentes estados afectivos. Los resultados sugirieron que la tasa de aciertos para ciertas distancias de clasificación, como las métricas A y R, la distancia de Camberra, Manhattan y una distancia basada en Minkowski se ven influenciadas negativamente por los cambios en las respuestas de excitación y valencia. La distancia euclídea fue la menos afectada de las seis evaluadas.

