

Tipo de artículo: Artículo original

# Reconocimiento de Emociones mediante el Mouse y Teclado

## *Emotion Recognition through Mouse and Keyboard*

Ramón Toala Dueñas <sup>1\*</sup> , <https://orcid.org/0000-0001-5397-9054>

Cristhian Maldonado Toala <sup>2</sup> , <https://orcid.org/0000-0003-0970-2386>

Diego Menéndez Navia <sup>3</sup> , <https://orcid.org/0000-0001-7540-557X>

<sup>1</sup> Departamento Sistemas Computacionales Facultad de Ciencias Informáticas, Universidad Técnica de Manabí. Ecuador.  
[ramon.toala@utm.edu.ec](mailto:ramon.toala@utm.edu.ec)

<sup>2</sup> Departamento Sistemas Computacionales Facultad de Ciencias Informáticas, Universidad Técnica de Manabí. Ecuador.  
[cmaldonado6769@utm.edu.ec](mailto:cmaldonado6769@utm.edu.ec)

<sup>3</sup> Departamento Sistemas Computacionales Facultad de Ciencias Informáticas, Universidad Técnica de Manabí. Ecuador.  
[dmenendez1179@utm.edu.ec](mailto:dmenendez1179@utm.edu.ec)

\* Autor para correspondencia: [ramon.toala@utm.edu.ec](mailto:ramon.toala@utm.edu.ec)

### Resumen

El objetivo principal del presente trabajo fue desarrollar un micro-servicio para el reconocimiento de emociones a través del uso del mouse y teclado, esta investigación se realizó de forma deductiva e inductiva las cuales permitieron conocer el funcionamiento que puede llegar a tener un micro-servicio como también el requerimiento necesario para cumplir las funcionalidades del mismo. Para el desarrollo de este trabajo se requirió el uso de una aplicación web para la obtención de información para así crear un DataSet que no es más que datos almacenados utilizados para el entrenamiento de una red neuronal, la cual fue necesaria para el desarrollo del micro-servicio. La metodología que se utilizó para el desarrollo del proyecto fue la metodología Cascada, la cual consta de cinco etapas a realizar: Análisis de Requerimientos, Diseño, Implementación, Verificación y Mantenimiento. Como entorno de desarrollo para la aplicación web fue JavaScript lo cual es un lenguaje de programación, como gestor de base de datos se utilizó MySQL y para su conexión se utilizó como parte principal PHP, también se utilizó los framework TensorFlow y Keras, para el desarrollo del micro-servicio se utilizó los datos almacenados del DataSet lo cual se implementó el lenguaje de programación Python para el entrenamiento de la red neuronal como también el uso del framework Flask. Una vez concluido la aplicación web como micro-servicio se realizó una verificación y mantenimientos necesarios, por lo que se concluye que el micro-servicio puede llegar a tener un mayor índice de acierto teniendo más datos en el dataset, lo que facilitaría en la detección de la emoción.

**Palabras clave:** metodología cascada; reconocimiento de emoción; aplicación web; framework; microservicio

### Abstract

*The main objective of this work was to develop a micro-service for the recognition of emotions through the use of the mouse and keyboard, this research was carried out in a deductive and inductive way, which allowed us to know the operation that a micro-service can have as well as the necessary requirement to fulfill its functionalities. For the development of this work, the use of a web application was required to obtain information in order to create a DataSet that is nothing more than stored data used for the training of a neural network, which was necessary for the development of the micro-service. The methodology used for the development of the project was the Cascade methodology, which consists of five stages to be carried out: Requirements Analysis, Design, Implementation, Verification and Maintenance. As a development environment for the web application, JavaScript was used, which is a programming language, MySQL was used as a database manager and PHP was used as the main part for its connection, the TensorFlow and Keras frameworks were also used for development. From the micro-service, the data stored in the DataSet was used, which implemented the Python programming language for the training of the neural network, as well as the use of the Flask framework. Once the web application as a micro-service was completed, a necessary verification and*



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional**  
(CC BY 4.0)

*maintenance was carried out, so it is concluded that the micro-service can have a higher success rate by having more data in the dataset, which would facilitate detection of emotion.*

**Keywords:** *waterfall methodology; emotion recognition; web application; framework; microservice*

**Recibido:** 21/02/2023  
**Aceptado:** 27/05/2023  
**En línea:** 01/06/2023

## Introducción

Las técnicas de *Machine Learning* han logrado importantes avances dentro de los últimos tiempos y han sido utilizados en muchos de los diferentes campos como clasificación de imágenes, procesamiento del lenguaje natural, reconocimiento de voz, entre otros. En algunas aplicaciones tales como la clasificación de imágenes, la precisión de *Machine Learning* incluso esta puede llegar a superar la capacidad que tiene el ser humano (Xue M., Yuan C., Wu H., Zhang Y., Liu W., 2020).

El reconocimiento de emociones mediante sensores específicos ha surgido como una tecnología que revoluciona diversas aplicaciones (Nakisa B., 2020). Sin embargo, la detección de emociones mediante dichos dispositivos que no están especializados con esta tarea se ha vuelto una evolución de dichas técnicas. Centrado en estos dos conceptos, la tarea de usar *Machine Learning* para detectar emociones usando como medio los dispositivos de entrada de uso cotidiano tales como el teclado y el mouse llevará a conocer las emociones que presenta el usuario al usar por ejemplo una plataforma virtual.

El presente trabajo busca solventar el problema que presentan muchas plataformas existentes en donde no consideran el perfil del usuario ni su estado emocional causando que este no se adapte del todo a las necesidades del alumno. Para esto lo ideal es tener un sistema que monitoree el perfil del alumno y sus emociones, con entornos inteligentes y un aprendizaje informático afectivo mejorado. El desarrollo de este micro-servicio permitirá adaptar esta funcionalidad a las plataformas que deseen ser más dinámica e interactivas con el usuario, sin incomodar al usuario, ya que no es invasivo.

La aplicación a desarrollar servirá para el monitoreo de emociones, lo que permitirá identificar el estado emocional de los estudiantes y mejorando su aprendizaje en la plataforma de la Facultad de Ciencias Informáticas (FCI), implementando la inteligencia artificial, de esta forma solventar la rigidez que actualmente se tiene en dicha plataforma.



Estudios realizados por (Navarro Cantos, 2018) en la Universidad de Sevilla demostró que mediante técnicas de machine Learning se pudo obtener un modelo de predicción sobre los estados de estrés y ansiedad en pilotos y trabajadores del sector aeronáutico. Utilizando sensores se estudió la respuesta a la sudoración de cada persona para diferentes pruebas realizadas. Se utilizaron 3 pruebas diferentes, utilizando audios, canciones y por último de manera visual.

En la Universidad de Alicante ubicada en España se realizó desarrollo un sistema para la detección de emociones el cual dispone de un módulo para la detección de rostros con el algoritmo de detección CNN+MMOD en Dlib, además de un clasificador de emociones todos basados en redes neuronales convolucionales (Gallud Baños, 2019). Así como (Gallud Baños, 2019), en la Universidad Politécnica de Valencia se realizó un proyecto en el cual se buscaba realizar el reconocimiento de emociones por medio de redes convolucionales. Al igual que (Gallud Baños, 2019), (Cebrián Chuliá, 2016) y (Saez De La Pascua, 2019) plantean una idea de reconocimiento de emociones a través del reconocimiento facial y el uso de librerías como Dlib. En este proyecto se utilizó una base de datos y un algoritmo establecido para ir midiendo la evolución de las características del rostro y asociándolas con sus emociones a través de puntos encontrados alrededor de las facciones faciales.

Las técnicas para el reconocimiento de emociones más utilizadas son las del reconocimiento facial. En la Universidad Técnica de Machala [8] se realizó el entrenamiento de una red neuronal en el cual se alcanzó una precisión del 67.13%. Además, se llegó a la conclusión de que el detectar emociones es una tarea difícil incluso para el ser humano. (Romero, 2020). Mediante la clasificación de emociones en la Universidad Católica de Santiago de Guayaquil se concluyó que el monitoreo de las expresiones faciales se puede utilizar para enriquecer la recopilación de datos en la atención de una forma inconsciente hacia un producto o servicio (Navarro Briones, 2020). Dentro de la Universidad Técnica de Manabí no se han realizado mucho proyecto relacionado con la detección y clasificación de emociones, aun menos con dispositivos que no sean diferentes a una cámara o con la detección de rostros.

En la Universidad de Pensilvania se desarrolló un trabajo en la cual su principal objetivo es predecir el afectivo estado de los alumnos, y para ello se centran en obtener un modelo fácilmente disponible para la detección rápida de estados afectivos, en el que los estudiantes etiquetan su propio estado afectivo usando una escala validada psicológicamente conocida y ampliamente utilizada que representa la intensidad de sus reacciones afectivas en dos diferentes dimensiones, a saber, valencia y excitación (Salmeron-Majadas, 2018).

Basado en esto se plantea como problema de la investigación: ¿El diseño de un micro-servicio, podrá detectar las emociones a través del mouse y del teclado, sin ser invasivos?



Los objetivos que persigue esta investigación:

- Analizar estudios relacionados a la detección y clasificación del estado emocional, así como identificar las conductas o los patrones que tiene una persona según sus emociones.
- Diseñar una aplicación web que obtenga los datos de los usuarios sobre la interacción del ratón y el teclado para crear el DataSet.
- Entrenar la red neuronal hasta tener un grado de precisión satisfactorio
- Desarrollar un micro-servicio para la detección de emociones a través del teclado y el ratón utilizando un DataSet

Con estos objetivos y sabiendo que la Universidad Técnica de Manabí no cuenta con una plataforma dinámica, nos motivó el poder adaptar a sus plataformas de estudio y poder monitorear la interacción del teclado y del mouse para obtener el tipo de emoción (positiva, neutra o negativa) que genere, mediante este proyecto se buscó tener un manejo más dinámico de las plataformas en las cuales no se considera el perfil ni el estado emocional de los alumnos.

## Materiales y métodos

La investigación se realizó en la Universidad Técnica de Manabí y justamente para proveerla de un micro-servicio para la detección de las emociones de los estudiantes, aportando con esto a la realización y construcción de futuros proyectos que requieran la implementación del mismo.

La aplicación web utiliza una arquitectura Modelo Vista Controlador, el modelo está encargado de almacenar los datos obtenidos durante el test que realiza el estudiante, la vista es la que representa la interfaz que llega a observar el estudiante y por último tenemos el controlador que se encargada de las funcionalidades de nuestra aplicación web.

Para la realización de la investigación implicó la utilización de la metodología en cascada, casos de uso y sus plantillas, en el análisis de requerimientos se detallan los procedimientos y técnicas que permitieron conocer el funcionamiento o elementos que definen dicha investigación. En la siguiente Tabla se puede observar las especificaciones de los requerimientos del sistema:

**Tabla 1.** Requerimientos funcionales y no funcionales.

Requerimientos Funcionales	Requerimientos No Funcionales
El microservicio enviara el estado emocional en tiempo real cada 10 segundos a través de consola	El microservicio se ejecutará en segundo plano.



---

El microservicio realizará una comparación de los datos obtenidos con los resultados de los datos entrenados.	La aplicación web deberá ser utilizada con dispositivo de entrada mouse y teclado
La aplicación web realizara la recopilación de datos del uso del teclado y mouse.	La aplicación web será desarrollada para las plataformas PC.

---

Fuente: Los Autores.

Los actores del sistema son aquellos que tienen una relación externa con el sistema y al mismo tiempo demanda una funcionalidad determinada, para la realización del caso de uso del proyecto se tomaron en cuenta el siguiente actor: Estudiante, a continuación, se presentan los diferentes casos de usos.

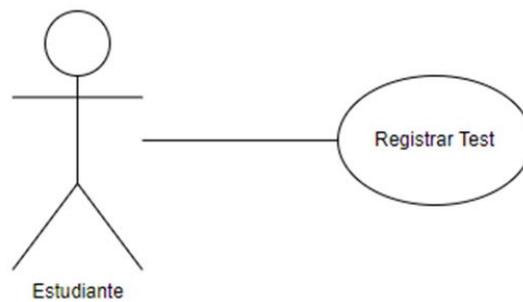


Figura 1. Caso de uso de registro de datos.

En la Figura 1 se puede observar que el Estudiante por medio de la aplicación web realizará un test la cual registrará la interacción del teclado y mouse, datos que se almacenará en la base de datos, que servirá para la creación del DataSet.

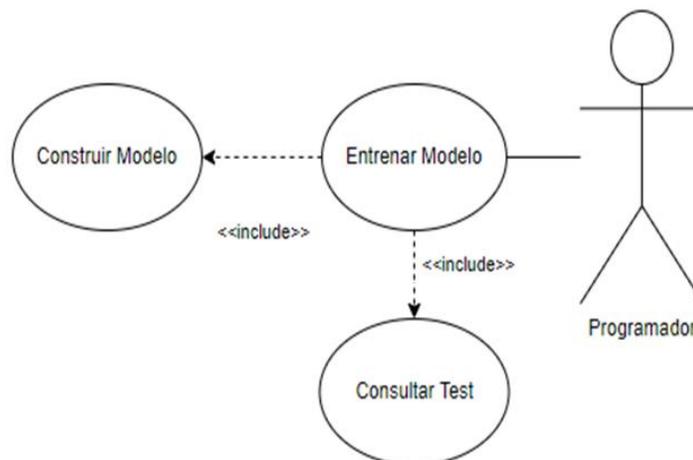


Figura 2. Caso de uso entrenamiento del Modelo.



El Programador utiliza los datos obtenidos en la base de datos del test para procesarlo y crear el DataSet teniendo como resultado un archivo .csv el cual servirá para su posterior entrenamiento.

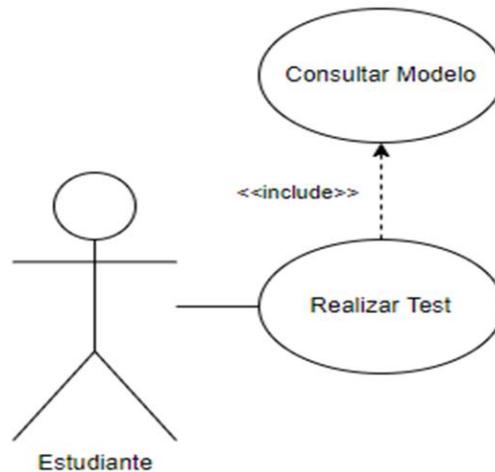


Figura 3. Caso de uso manejo del micro servicio.

El Estudiante utiliza el micro-servicio el cual consulta con el modelo entrenado para mostrar la emoción más cercana a la interacción dada por el estudiante.

El funcionamiento interno del proceso de entrenamiento para el modelo y el funcionamiento del microservicio, se detalla a continuación, para esto primero mostraremos la arquitectura de nivel 0 el cual muestra el funcionamiento de todo el sistema.

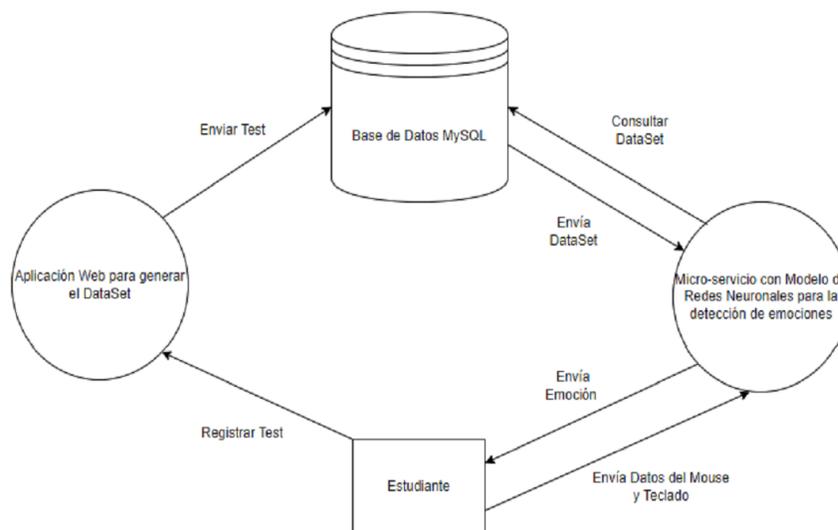
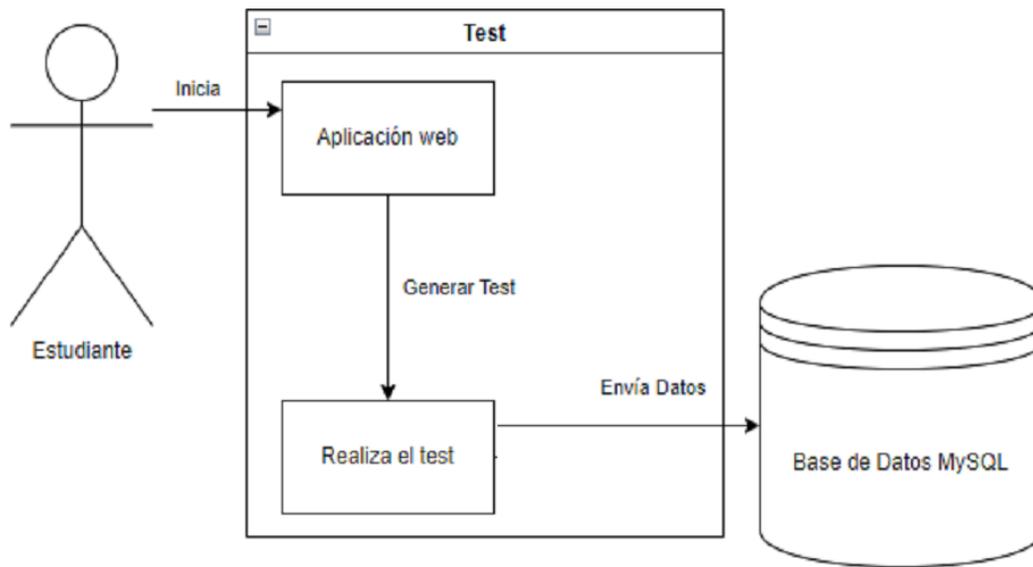


Figura 4. Modelo de Arquitectura del Software Nivel 0.



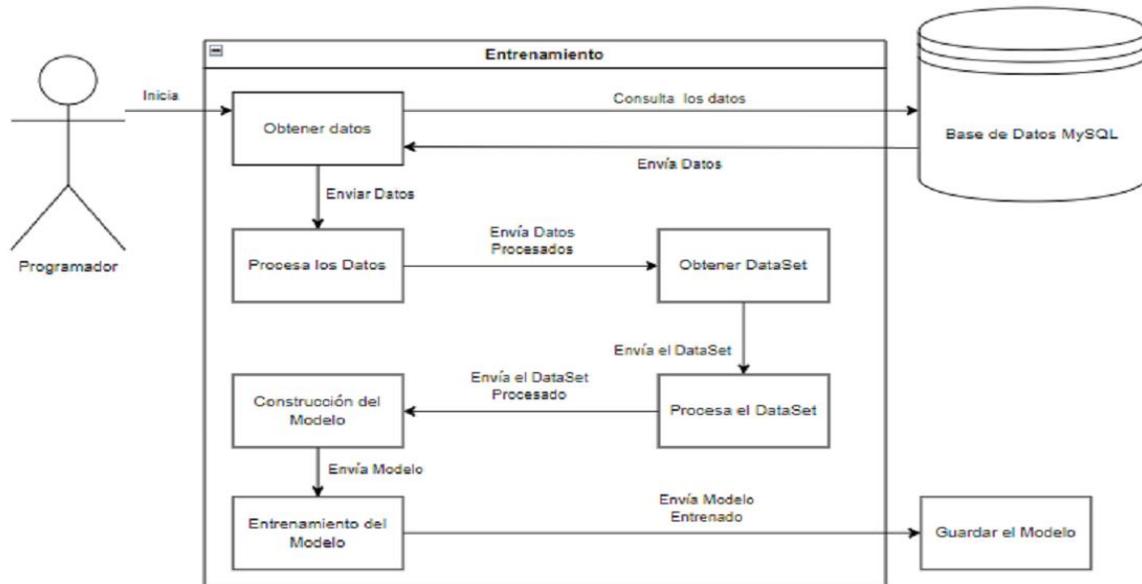
En esta arquitectura se muestra el funcionamiento, en donde el estudiante al enviar el test a través de nuestra aplicación web, los datos que se almacenara en la base de datos para la generación del DataSet, el Estudiante al utilizar el micro-servicio este enviará los datos del mouse y teclado para que este consulte con el modelo entrenado y envíe la emoción que está generando el estudiante en tiempo real.

A continuación, mostraremos la arquitectura de sistema tanto del test, entrenamiento como también del micro-servicio, esta arquitectura nos permite mostrar cómo funciona internamente el proceso para el desarrollo de estos.



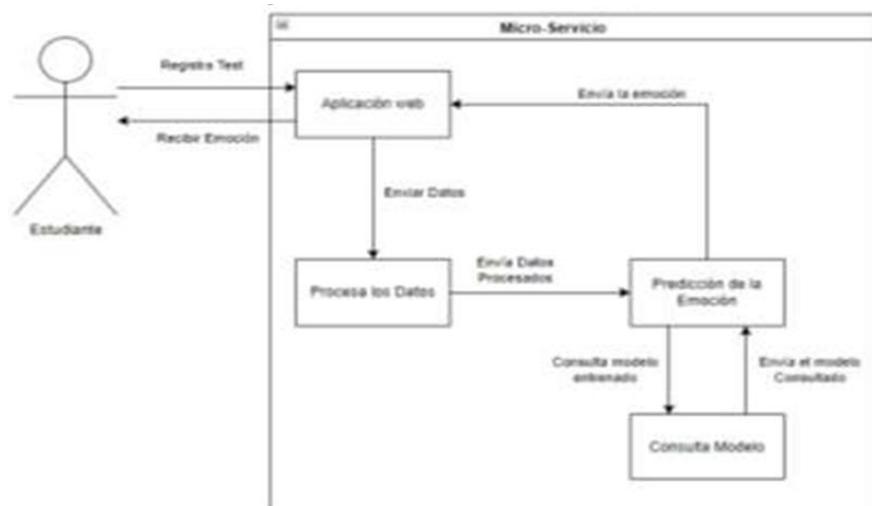
**Figura 5.** Modelo de Arquitectura de Sistema de Test

En esta arquitectura de sistema se explica lo que sucede cuando el estudiante realiza el test en la aplicación web la cual consiste en una serie de preguntas sobre informática (disciplina que se seleccionó para realizar el test y aplicado en la Facultad de Informática), además de preguntas sobre el estado de ánimo actual del estudiante para que él conteste de forma que los datos de interacción con el mouse y ratón se almacenen dentro de una Base de Datos en MySQL.



**Figura 6.** Modelo de Arquitectura de Sistema de entrenamiento.

La arquitectura de sistema del entrenamiento se explica como el programador, comenzamos a realizar el modelo entrenado, partiendo de la base de datos del test obtendremos los datos para ser procesados por un algoritmo que se encargará de construir el DataSet, luego se procesa para separar las variables característica de las etiquetas para ser incluidas en la construcción del modelo y una vez construido se realiza el entrenamiento del modelo para posteriormente ser guardado.



**Figura 7.** Modelo de Arquitectura del micro servicio



En la presente arquitectura de sistema se explica cómo trabaja el micro-servicio internamente, cuando el estudiante realiza el test en la aplicación él envía datos los cuales son procesados para realizar la predicción, el cual esta consulta con el modelo entrenado y este devuelve la emoción seleccionada a la aplicación para que este le muestre al estudiante la emoción.

## Resultados y discusión

A continuación, se muestra el desarrollo que se realizó para obtener los resultados una vez aplicado el test en el micro servicio para la recolección de información utilizando herramientas como JavaScript como lenguaje de programación y PHP para la conexión con el servidor, así mismo por parte del servidor para el almacenamiento de los datos se utilizó MySQL, todo esto para la creación del DataSet que nos permitirá el entrenamiento del modelo para la creación del micro-servicio.

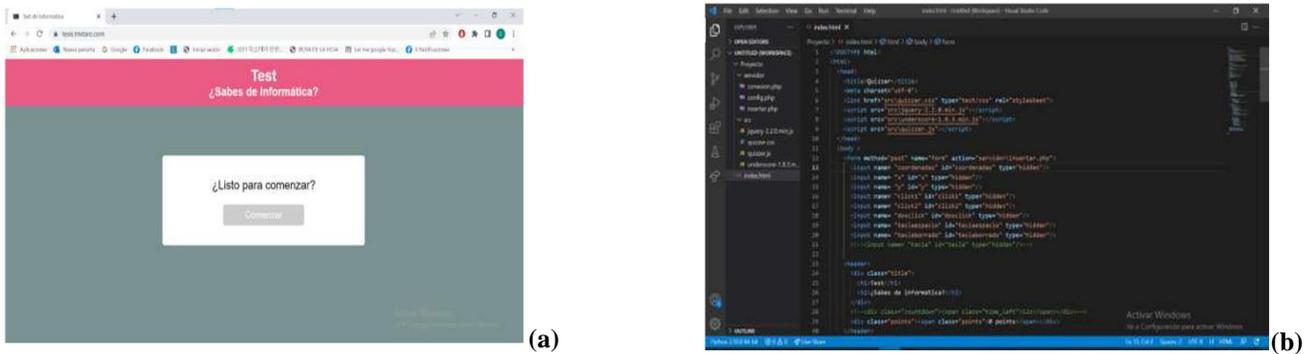


Figura 8. (a) Interfaz de la aplicación web (b) Código fuente de la aplicación.

Para la obtención de los datos se envió a todos los estudiantes de la Facultad de Ciencias Informáticas de la Universidad Técnica de Manabí, estos datos almacenados son tanto del teclado, mouse y la emoción que proporciona a la realización del test, estos datos fueron consiguiéndose en un periodo de 6 días los cuales representaremos cada uno de los datos obtenidos durante ese tiempo.

La cantidad de pruebas que se obtuvieron durante estos 6 días fue un total 129, aunque parezca una cantidad muy baja para el entrenamiento de una red neuronal cabe mencionar que en cada una de las pruebas realizadas se recolecta 5 veces los datos por prueba, por lo que la cantidad total se verá multiplicada por 5 dando un total de 645, Los tres datos obtenidos durante el test los cuales son el mouse, el teclado y la emoción se guarda de forma independiente dentro de un array.



<pre> 2020-10-25 09:52:47,283: 2 2020-10-25 09:53:32,844: 2 2020-10-25 09:54:17,389: 2 2020-10-25 09:55:01,827: 2 2020-10-25 09:55:47,243: 2 2020-10-25 09:56:33,762: 2 2020-10-25 09:57:17,490: 2 2020-10-25 09:58:03,107: 2 2020-10-25 09:58:47,650: 2 2020-10-25 09:59:32,770: 2 2020-10-25 10:00:18,171: 2 2020-10-25 10:01:05,354: 1 2020-10-25 10:01:47,537: 1 2020-10-25 10:02:32,881: 1 2020-10-25 10:03:19,041: 2 2020-10-25 10:04:06,201: 2                 </pre> <p>(a) Nivel de emoción</p>	<pre> 2020-10-25 09:58:07,978: Key.print_screen 2020-10-25 09:58:24,007: Key.ctrl_l 2020-10-25 09:58:24,220: '\x16' 2020-10-25 09:58:32,565: Key.print_screen 2020-10-25 09:58:32,910: Key.alt_l 2020-10-25 09:58:33,031: Key.tab 2020-10-25 09:58:33,463: Key.ctrl_l 2020-10-25 09:58:33,586: '\x16' 2020-10-25 09:58:33,964: Key.alt_l 2020-10-25 09:58:34,083: Key.tab 2020-10-25 09:59:19,932: Key.print_screen 2020-10-25 09:59:22,469: Key.ctrl_l 2020-10-25 09:59:22,664: '\x16' 2020-10-25 10:00:36,070: Key.backspace 2020-10-25 10:00:36,328: Key.enter 2020-10-25 10:14:41,678: Key.shift                 </pre> <p>(b) Interacción del Teclado</p>	<pre> 2020-10-25 09:52:46,499: Mouse moved to (651, 298) 2020-10-25 09:52:46,516: Mouse moved to (651, 298) 2020-10-25 09:52:46,667: Mouse moved to (650, 298) 2020-10-25 09:52:46,675: Mouse moved to (650, 298) 2020-10-25 09:52:46,683: Mouse moved to (650, 302) 2020-10-25 09:52:46,691: Mouse moved to (650, 311) 2020-10-25 09:52:46,699: Mouse moved to (652, 319) 2020-10-25 09:52:46,707: Mouse moved to (654, 328) 2020-10-25 09:52:46,715: Mouse moved to (657, 338) 2020-10-25 09:52:46,723: Mouse moved to (659, 348) 2020-10-25 09:52:46,731: Mouse moved to (661, 356) 2020-10-25 09:52:46,739: Mouse moved to (663, 363) 2020-10-25 09:52:46,747: Mouse moved to (666, 370) 2020-10-25 09:52:46,755: Mouse moved to (669, 377) 2020-10-25 09:52:46,764: Mouse moved to (672, 384)                 </pre> <p>(c) Interacción del mouse</p>
--	--	--

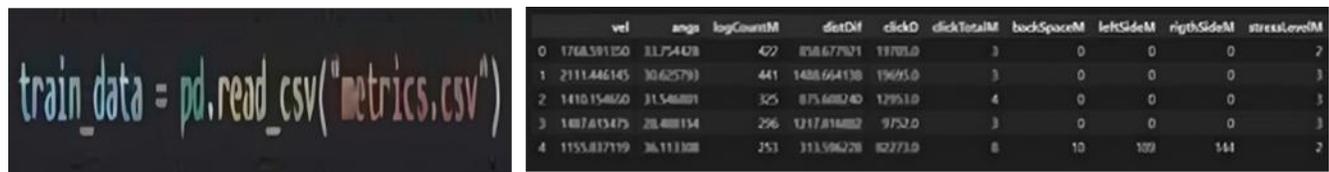
**Figura 9.** Datos almacenados para la construcción del Dataset.

Una vez que se obtuvieron los datos se comenzó a la creación del DataSet por lo que se importó todos los datos de nuestro MySQL a un archivo .csv por lo que facilitara el entrenamiento de la red neuronal.

	A	B	C	D	E	F
1	velocity,angles	logCount,distStraighLine,clickDuration,clickTotal,distBetweenClick,				
2	2493.08353572,141797.341031,413,0,0,20,2,0,0,0,2,0,1					
3	1030.14383729,36570.4735539,93,240.457977272,12056,2,1926.6748894,0,27,29,1					
4	2023.07678603,1174967.02753,431,1420.15884277,25578,8,2026.01055435,0,30,26,1					
5	2930.80436313,72113.9531076,506,1392.55745729,47920,8,2931.06608188,0,30,27,1					
6	2869.38553068,60930.7744953,360,2244.57379734,46721,20,2926.12231125,0,0,1,1					
7	1668.22182032,298317.511988,258,561.917831504,4358,6,780.77587248,1,31,28,1					
8	1317.10841226,166092.000415,396,608.138641836,32116,6,1277.95787915,0,32,31,1					
9	362.19239836,7076.48685191,99,379.77256389,3697,6,413.984143244,0,14,14,1					
10	1664.17245914,1091636.74869,417,1023.70091455,24479,8,1668.48223162,4,23,36,1					
11	1286.57891667,72298.0886654,306,778.973761811,52035,4,875.599070772,1,27,35,1					
12	1272.866876,89643.2257941,437,706.824597603,18055,8,1730.21165575,0,33,32,1					
13	891.146645948,9823.39462715,141,694.369336268,21294,4,897.975008035,0,4,4,1					
14	3811.19914458,208671.501126,804,2013.22636416,43356,12,4064.60539355,0,33,32,1					
15	1821.39755155,363526.08871,296,1080.81075859,26490,10,1839.45405062,0,14,11,1					
16	1644.67434625,37692.5246653,237,1364.96197262,31407,10,1751.14148023,0,29,18,1					
17	1110.59465803,951972.922164,238,752.750602428,696057,8,1430.7718928,0,11,10,1					

**Figura 10.** Contenido del Dataset.

Posterior a esto, se realizó la importación del Dataset Figura 11 (a) (todo el proceso se lo realizó utilizando las librerías de Python), una vez importado observamos algunos datos en la Figura 11 (b)



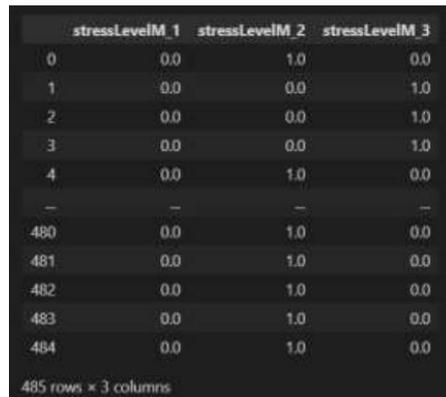
(a) Importación del Dataset

(b) Datos Importados

**Figura 11.** Proceso de importación.

Ya que el problema es multiclase, se necesita que la etiqueta sea dividida en 3 categorías, ya que esta corresponderá a la salida Figura 12.





	stressLevelM 1	stressLevelM 2	stressLevelM 3
0	0.0	1.0	0.0
1	0.0	0.0	1.0
2	0.0	0.0	1.0
3	0.0	0.0	1.0
4	0.0	1.0	0.0
—	—	—	—
480	0.0	1.0	0.0
481	0.0	1.0	0.0
482	0.0	1.0	0.0
483	0.0	1.0	0.0
484	0.0	1.0	0.0

485 rows x 3 columns

Figura 12. Variables etiquetadas.

Para la creación del modelo se utilizó un modelo secuencial en el cual tenemos una capa de entrada con 9 neuronas, se agregó una capa oculta con activación *relu* de 4 neuronas basándonos en la cantidad de entradas y la cantidad de salidas que tenemos para evitar el sobreajuste y para la capa de salida, dado que este es un problema multiclase empleamos el método de activación *softmax* que dada las 3 neuronas de salida establecidas selecciona la de mayor valor, como se observa en la siguiente figura.

```
model = tf.keras.Sequential([
    tf.keras.layers.Input((9,)),
    tf.keras.layers.Dense(4, activation='tanh'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(3, activation='sigmoid'),
    tf.keras.layers.Dense(3, activation='softmax'),
])
```

Figura 13. Creación del modelo de red neuronal.

El entrenamiento del modelo empieza asignándole las características y las etiquetas además de seleccionar un porcentaje del DataSet para validación de 20% y realizaremos el entrenamiento 100 épocas.

Una vez obtenida la información se realizó el entrenamiento utilizando una red neuronal, cuya arquitectura se presenta en la Figura 14.



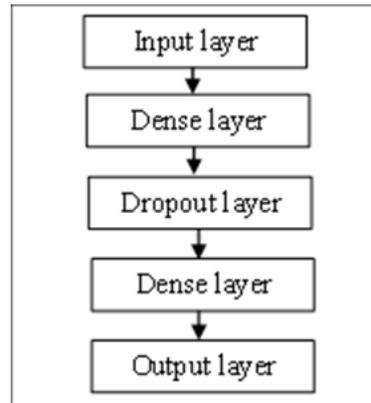


Figura 14. Modelo de la red neuronal.

Por otra parte, tenemos el micro-servicio, el cual se utiliza el modelo entrenado para conseguir un grado de acierto satisfactorio el cual nos permitirá conocer de forma eficiente la emoción resultante. Cuando la emoción sea neutra la interfaz de toda la aplicación se tornará de un color gris, positivo color verde y negativa color rojo.

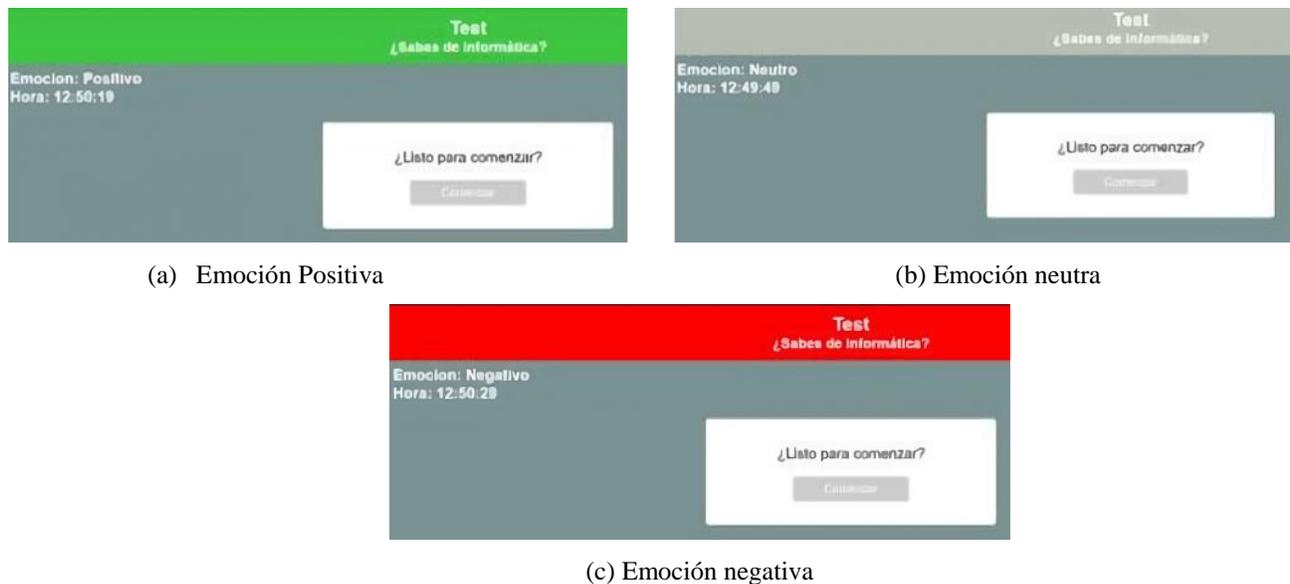


Figura 15. Emociones resultantes.

Para el desarrollo del proyecto aparte de la metodología escogida se utilizó herramientas tecnológicas como JavaScript para la creación de la aplicación web, MySQL como nuestro gestor de almacenamiento de datos, también se utilizó PHP para el establecimiento con la conexión entre nuestro servidor y el navegador como nuestro gestor de almacenamiento de los datos, los framework TensorFlow y Keras para el entrenamiento de las redes neuronales, y para el desarrollo del micro-servicio se utilizó el framework Flask.



Los resultados que se han obtenido durante el desarrollo del presente trabajo, fueron los siguientes:

- Se pudo llegar a conseguir información suficiente para el entendimiento y desarrollo de un micro-servicio para la detección de las emociones a través de teclado y mouse.
- Con la construcción de la aplicación web se pudo llegar a obtener los datos para la creación del DataSet.
- Por medio del modelo de redes neuronales construido y el DataSet obtenido se logró un correcto entrenamiento.
- Se construyó un micro-servicio que detecta la interacción del uso del teclado y ratón para mostrar la emoción seleccionada por el modelo de redes neuronales periódicamente cada 10 segundos

## Conclusiones

La detección de emociones a través del teclado y mouse existen varias investigaciones en los que se detectaban emociones, pero de forma intrusiva ejecutando alguna aplicación adicional o con la utilización de hardware adicional como una webcam o micrófono, por lo que la realización de este proyecto abrirá la posibilidad de aumentar la información acerca de la detección de emociones de forma no intrusiva.

El correcto entrenamiento de una red neuronal se basa en su arquitectura y en la cantidad de datos a entrenar, con este fin se probaron distintos tipos de arquitectura los cuales no cambiaron la precisión establecida por la arquitectura base que utilizamos con la que obtuvimos una precisión del 52%, y si lograba llegar a más de ese porcentaje los valores de perdida eran muy altos e inestables, por lo que establecimos que los datos utilizados para entrenar el modelo no eran suficientes para superar la precisión obtenida.

Para próximos trabajos basados en este tipo de reconocimiento de emociones, se sugiere que el Dataset tenga la mayor cantidad de Datos, así como también se utilice un computador con mayores prestaciones para el entrenamiento de manera que se obtenga la mayor precisión de los mismos.

## Agradecimientos

Se le agradece tanto a los estudiantes, profesores y autoridades de la Facultad de Ciencias Informáticas, por la apertura y apoyo a la investigación realizada.

## Conflictos de intereses

No se declaran conflictos de intereses.

## Contribución de los autores

1. Conceptualización: Cristhian Alberto Maldonado Toala.



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

2. Curación de datos: Cristhian Alberto Maldonado Toala, Diego Menéndez Navia, Ramón Toala Dueñas
3. Análisis formal: Ramón Toala Dueñas
4. Investigación: Cristhian Alberto Maldonado Toala, Diego Menéndez Navia
5. Metodología: Ramón Toala Dueñas
6. Administración: Cristhian Alberto Maldonado Toala, Diego Menéndez Navia, Ramón Toala Dueñas
7. Recursos: Cristhian Alberto Maldonado Toala, Diego Menéndez Navia
8. Software: Cristhian Alberto Maldonado Toala, Diego Menéndez Navia
9. Supervisión: Ramón Toala Dueñas
10. Validación: Ramón Toala Dueñas
11. Visualización: Ramón Toala Dueñas
12. Redacción – borrador original: Cristhian Alberto Maldonado Toala, Diego Menéndez Navia, Ramón Toala Dueñas
13. Redacción – revisión y edición: Cristhian Alberto Maldonado Toala, Diego Menéndez Navia, Ramón Toala Dueñas

## Financiamiento

La investigación fue financiada por los autores.

## Referencias

- Cebrián Chuliá, L. (2016). Reconocimiento de Emociones mediante técnicas de aprendizaje profundo (Doctoral dissertation, Universitat Politècnica de València).
- Gallud Baños, G. (2019). Reconocimiento de emociones humanas y su aplicación a la Robótica Social.
- Nakisa B., R. M. (2020). Automatic emotion recognition using temporal multimodal deep learning. IEEE Access, 225463-225474.
- Navarro Briones, J. D. (2020). Sistema de reconocimiento de expresión facial para la detección de emociones: prototipo para medir el nivel de satisfacción de servicio al cliente en la recepción de la Carrera de Computación de la Universidad Católica Santi. Guayaquil: <http://repositorio.ucsg.edu.ec/bitstream/3317/14344/1/T-UCSG-PRE-INGCIS-252.pdf>.



- Navarro Cantos, C. (2018). Detección de los niveles de estrés y ansiedad en pilotos aplicando técnicas de Machine Learning.
- Romero, S. E. (2020). Detección de emociones y reconocimiento facial. Machala: <http://repositorio.utmachala.edu.ec/bitstream/48000/15453/1/ECFIC-2020-ISDE->.
- Saez De La Pascua, A. (2019). Deep learning para el reconocimiento facial de emociones básicas (Bachelor's thesis, Universitat Politècnica de Catalunya).
- Salmeron-Majadas, S. B. (2018). A machine learning approach to leverage individual keyboard and mouse interaction behavior from multiple users in real-world learning scenarios. *IEEE Access*, 6, , 39154-39179.
- Xue M., Yuan C., Wu H., Zhang Y., Liu W. (2020). Machine Learning Security: Threats, Countermeasures, and Evaluations.

