

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**DISEÑO Y APLICACIÓN DE UN MODELO DE RECONOCIMIENTO
DE ONCE GESTOS DE LA MANO USANDO SEÑALES EMG Y
DEEP LEARNING**

**DISEÑO DE UN MODELO DE RECONOCIMIENTO DE ONCE
GESTOS DE LA MANO QUE FUNCIONE EN TIEMPO REAL**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE
SOFTWARE**

JOSÉ MIGUEL MUÑOZ OÑA

jose.munoz01@epn.edu.ec

DIRECTOR: LORENA ISABEL BARONA LÓPEZ, PHD

lorena.barona@epn.edu.ec

DMQ, septiembre 2022

CERTIFICACIONES

Yo, José Miguel Muñoz Oña declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

JOSÉ MIGUEL MUÑOZ OÑA

Certifico que el presente trabajo de integración curricular fue desarrollado por José Miguel Muñoz Oña, bajo mi supervisión.

LORENA ISABEL BARONA LÓPEZ
DIRECTORA

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

JOSÉ MIGUEL MUÑOZ OÑA

LORENA ISABEL BARONA LÓPEZ

DEDICATORIA

A Dios, por iluminar mi mente y permitirme seguir adelante.

A mis padres, mi hermana y mis abuelitos, por todo el amor, la paciencia y el esfuerzo que me han permitido culminar esta etapa.

A mis amigos (Angie, Carlos, Carolina, César, Jorge, Mahatma, Renato, Santiago, Vanesa), con quienes viví y compartí muchas experiencias e hicieron de este trayecto algo mucho más llevadero.

AGRADECIMIENTO

A todos mis profesores, en particular a los doctores Lorena Barona, Marco Benalcázar y Leonardo Valdivieso, por ese voto de confianza para realizar el trabajo resumido en este documento.

A todos los que forman parte del Laboratorio de Investigación en Inteligencia y Visión Artificial “Alan Turing”, por su ayuda y por compartir sus conocimientos conmigo.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VI
ABSTRACT	VII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general	1
1.2 Objetivos específicos	1
1.3 Alcance	2
1.4 Marco teórico	4
2 METODOLOGÍA.....	7
2.1 Entendimiento del negocio y de los datos	7
2.2 Preparación de datos	10
2.3 Modelamiento.....	13
2.4 Evaluación.....	16
2.5 Despliegue	17
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	19
3.1 Resultados	19
3.2 Conclusiones.....	23
3.3 Recomendaciones.....	24
4 REFERENCIAS BIBLIOGRÁFICAS	25
5 ANEXOS.....	29
ANEXO I. Lienzo de aprendizaje de máquina	29
ANEXO II. Enlaces	30
ANEXO III. Tabla de verificación del conjunto de datos	31
ANEXO IV. Configuraciones de los optimizadores para las redes CNN-LSTM....	45
ANEXO V. Tabla de resultados de evaluación clasificador lineal logístico	46

RESUMEN

En la actualidad, muchas aplicaciones y productos de software utilizan modelos de aprendizaje de máquina para entregar funcionalidades valiosas a los usuarios finales. Sin embargo, muchos de estos proyectos fracasan por problemas relacionados con la salida a entornos de producción, la reproductibilidad de modelos y la deuda técnica. En el presente trabajo, se desarrolló un sistema de reconocimiento para once gestos de la mano utilizando redes neuronales artificiales CNN-LSTM guiado por el modelo de proceso CRISP-ML(Q) con el fin de introducir prácticas de la ingeniería de software a un proyecto de aprendizaje de máquina. El sistema toma como entrada señales electromiográficas (EMG) y de cuaterniones (como unidad de medición inercial) para determinar la etiqueta y el instante de tiempo de ocurrencia del gesto en la señal y se conecta con una aplicación que utiliza los gestos. El documento describe desde el detalle de las señales de entrada, el proceso de recolección, limpieza, selección y construcción de datos, hasta la evaluación y despliegue del sistema, en el cual se determinó una exactitud de clasificación del $55.33\% \pm 18.45\%$ y una exactitud de reconocimiento del $49.15\% \pm 18.41\%$, con un tiempo de respuesta de 29.26 ± 19.82 [ms].

PALABRAS CLAVE: Aprendizaje de máquina, cuaternión, electromiografía, ingeniería de software, reconocimiento de gestos de la mano, redes neuronales artificiales, tiempo real, unidades de medición inercial.

ABSTRACT

Many software products and applications use machine learning models to deliver valuable functions to end users. Despite of the development efforts, many projects fail due to issues on deployment to production, model reproducibility, and technical debt. In this work, a hand gesture recognition system was developed for 11 gestures, using CNN-LSTM artificial neural networks, and guided by CRISP-ML(Q) process model to introduce software engineering practices into machine learning projects. The system uses electromyographic (EMG) and quaternion (inertial measurement unit) signals as input to compute a gesture label and the time of occurrence in the signal. The system is connected to an application that uses hand gestures. This document records the data description, collection, cleaning, and selection processes up until the system evaluation and deployment, where a classification accuracy of $55.33\% \pm 18.45\%$ and a recognition accuracy of $49.15\% \pm 18.41\%$ were computed, with a response time of 29.26 ± 19.82 [ms].

KEYWORDS: Artificial Neural Networks, electromyography, hand gesture recognition, inertial measurement unit, machine learning, quaternion, real time, software engineering.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En este componente se construyó un sistema de reconocimiento para once gestos de la mano y el brazo: mano a la izquierda (*wave in*), mano a la derecha (*wave out*), puño (*fist*), mano abierta (*open*), doble golpe de los dedos (*pinch*), arriba (*up*), abajo (*down*), izquierda (*left*), derecha (*right*), apuntar al frente (*forward*) y apuntar hacia atrás (*backward*). Para esto, se experimentó con redes neuronales artificiales entrenadas utilizando espectrogramas de señales basadas en electromiografía y unidades de medición inercial. Tras la fase de entrenamiento y pruebas en los modelos se construyó el sistema, el cual se conecta al sensor *Myo armband* mediante un SDK. El sistema recolecta mediante el sensor señales electromiográficas y cuaterniones (unidad de medición inercial) durante un período de tiempo definido. Luego, recorta la señal usando ventanas y realiza un tratamiento en la señal (filtrado para señales electromiográficas o normalización en señales de cuaterniones). Después, el sistema genera una representación en el dominio tiempo frecuencia (espectrograma) por cada señal y, usando los modelos previamente entrenados, retorna una secuencia de etiquetas en función del número de ventanas de la señal. Este componente es usado en una aplicación (componente dos) como un reemplazo de la interacción por teclado y ratón, razón por la cual el sistema funciona en tiempo real. El sistema se comunica con la aplicación mediante sockets para enviar información de los gestos realizados por el usuario.

1.1 Objetivo general

Reconocer en tiempo real once gestos de la mano mediante un programa de computadora basado en redes neuronales convolucionales LSTM

1.2 Objetivos específicos

1. Revisar el estado del arte en arquitecturas de redes neuronales que utilicen señales electromiográficas (EMG) y unidades de medición inercial para el reconocimiento de gestos de la mano.
2. Desarrollar un modelo de reconocimiento de 11 gestos de la mano con redes neuronales convolucionales LSTM, basado en un modelo de proceso CRISP-ML(Q) para guiar el ciclo de vida de la aplicación de aprendizaje de máquina.

3. Determinar la eficacia de clasificación de una red neuronal convolucional LSTM para 11 gestos de la mano usando como entrada señales electromiográficas y cuaterniones.
4. Integrar el modelo de reconocimiento y validar el funcionamiento integrado de la aplicación de aprendizaje de máquina con la aplicación del componente 2.

1.3 Alcance

En este componente se diseñará un modelo de reconocimiento de once gestos de la mano con un tiempo de respuesta menor a los 300 ms. El tiempo mencionado toma en consideración el ingreso de las señales EMG y de los cuaterniones hasta la generación de la etiqueta. El modelo utilizará redes neuronales convolucionales LSTM y reconocerá los gestos indicados en la Figura 1.

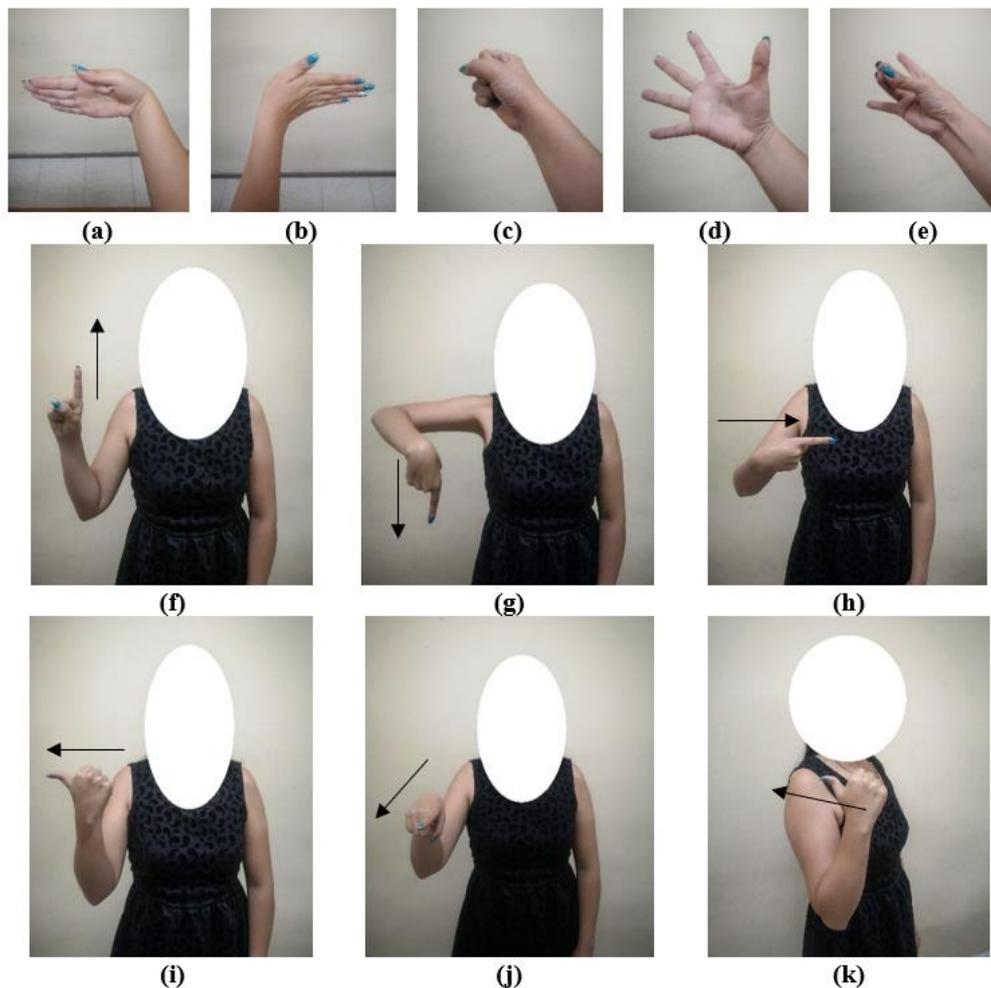


Figura 1 Gestos que se plantea reconocer en este proyecto. (a) wave in, (b) wave out, (c) fist, (d) open, (e) pinch, (f) up, (g) down, (h) left, (i) right, (j) forward, y (k) backward.

El cronograma para el desarrollo del proyecto se encuentra detallado en la Tabla 1.

Tabla 1 Cronograma de actividades y entregables del componente

Semana referencial	Tareas específicas	Resultado esperado (si aplica)
1	Definición de objetivos del negocio y de la aplicación ML	Objetivos del negocio y de la aplicación de machine learning
2	Descripción y verificación de datos	Documentación de las propiedades de los datos y de su proceso de generación
3	Selección y limpieza de conjuntos de datos	Muestras de gestos segmentadas
4	Construcción de datos: caracterización de los datos	Código fuente del módulo de preprocesamiento
5	Construcción de datos: extracción de características y estandarización	Espectrogramas de las muestras segmentadas
6	Implementación del modelo	Código fuente del clasificador
7	Entrenamiento de la red neuronal	Modelo de reconocimiento de gestos
8	Adaptación del módulo de posprocesamiento	Código fuente del módulo de posprocesamiento
9	Validación del rendimiento del modelo	Matriz de confusión
10	Definición de infraestructura para despliegue	Código y/o documentación de la infraestructura de comunicación
11	Evaluación del modelo en producción	Resultados de evaluación del modelo

1.4 Marco teórico

El reconocimiento de gestos es un campo de estudio donde se busca introducir un sistema que permita detectar gestos humanos y usarlos para comunicar información o con el propósito de comandar o controlar dispositivos [1]. En consecuencia, el problema de reconocimiento de gestos de la mano consiste en identificar una clase, a partir de un conjunto predefinido de clases, y el instante de ocurrencia de un movimiento de la mano [2]. Es decir, una interpretación al movimiento realizado, y el seguimiento de los movimientos a lo largo del tiempo. Los sistemas de reconocimiento de gestos ofrecen una forma natural de interacción y eliminan la dependencia de dispositivos como el ratón, teclado, pantallas táctiles y controles. Dentro de las áreas de aplicación del reconocimiento de gestos de la mano más populares se tienen: lenguaje de señas, aplicaciones médicas, robótica, automatización de hogares, entornos virtuales y aplicaciones de computadoras [1].

Existen varios tipos de sensores que permiten registrar datos de un gesto de la mano, tales como guantes [3], cámaras web [4], cámaras infrarrojas [5], [6], unidades de medición inercial (IMU, por sus siglas en inglés) [7] y electromiografía (EMG) [8]. En particular, la electromiografía superficial (sEMG) registra la actividad eléctrica de un músculo desde la superficie de la piel, reflejando la generación y propagación del potencial de acción por unidad motora del músculo [8]. Las interfaces músculo computador que usan este tipo de señal han demostrado tener una muy buena exactitud de reconocimiento sobre gestos estáticos que involucran distintas activaciones musculares, razón por la cual se usan para controlar prótesis, sillas de ruedas y exoesqueletos [8], [9]. Por otro lado, las unidades de medición inercial permiten registrar cambios direccionales como rotación y traslación, logrando mejores resultados al discriminar gestos dinámicos [7], [9]. De ahí que la combinación de las unidades de medida inercial junto con la electromiografía pueda tener la capacidad de aumentar la exactitud de clasificación en un sistema de reconocimiento de gestos estáticos y dinámicos. Existen evidencias en la literatura del funcionamiento de sistemas de reconocimiento que usan esta combinación y obtienen resultados aceptables, usando diferentes modelos y conjuntos de gestos. Un resumen de los resultados obtenidos en la literatura se muestra en Tabla 2.

Tabla 2 Exactitud de clasificación de modelos entrenados usando EMG e IMU en la literatura

Modelo	Exactitud de clasificación	Número de usuarios
Análisis del discriminante lineal (LDA) [9]	95.97 %	4
Modelo oculto de Márkov (HMM) [10]	74.3 %	5
Clasificador lineal Bayesiano [11]	94.32 %	5
Máquina de soporte vectorial (SVM) [12]	99 %	3

Como se ha podido apreciar en la literatura, los sistemas de reconocimiento de gestos son principalmente proyectos relacionados con el aprendizaje de máquina. De acuerdo con un reporte del estado del aprendizaje de máquina en entornos empresariales [13], uno de los grandes retos de este tipo de proyectos es lograr que cumplan los objetivos que generen valor para las partes interesadas. Esto se debe a que muchos de estos proyectos no han podido ser puestos en producción, así como a otras causas como la falta de reproductibilidad, la poca escalabilidad de las aplicaciones de aprendizaje de máquina, el escaso control de versiones sobre los modelos y la deuda técnica asociada [13], [14]. Por lo tanto, es necesaria la introducción de la ingeniería de software para proyectos de aprendizaje de máquina. La ingeniería de software es la aplicación sistemática de principios, métodos, técnicas y herramientas para el desarrollo, operación y mantenimiento del software. E involucrando el uso de un número de personas distintas en papeles distintos, gestiona proyectos de software mediante un enfoque sistemático, disciplinado y cuantificable para producir un producto confiable, mantenible, reutilizable, valioso, dentro del presupuesto y a tiempo, junto con su documentación asociada [15]–[19]. Sin embargo, la aplicación de esta ciencia para este tipo de proyectos no es directa o explícita. Mientras un producto de software tradicional codifica los requerimientos como reglas dentro del código (y, por ende, la funcionalidad se puede trazar a los requisitos y viceversa), en el software de aprendizaje de máquina las reglas se configuran de forma indirecta mediante el aprendizaje o captura de patrones de los datos [20]. En consecuencia, resulta necesario alinear los principios, métodos, técnicas y herramientas de la ingeniería de software de tal modo que sean aplicables en proyectos de aprendizaje de máquina y consigan los mismos resultados de aplicación que existen en el software tradicional.

En concreto, existe una metodología para el desarrollo de aplicaciones de aprendizaje de máquina que busca mitigar problemas para entregar el valor esperado por las partes interesadas, así como reducir la deuda técnica de este tipo de aplicaciones: CRISP-ML(Q) [21]. Esta metodología apunta a escenarios en los cuales un modelo de aprendizaje de máquina se despliega y mantiene como parte de un producto o servicio. Para ello, CRISP-ML(Q) toma los principios planteados por el modelo de proceso industrial estándar para minería de datos (CRISP-DM) [22], pero se modifica para ajustarse a los requerimientos particulares de las aplicaciones de aprendizaje de máquina. Las actividades sugeridas por esta metodología, así como el objetivo que persigue cada una de las mismas se detallan en la sección 2 del presente documento.

2 METODOLOGÍA

Esta sección describe cómo las actividades propuestas en CRISP-ML(Q) guiaron el **desarrollo** de la aplicación de aprendizaje de máquina para el reconocimiento de gestos. Un resumen de todas las fases consideradas en esta metodología se muestra en Figura 2.

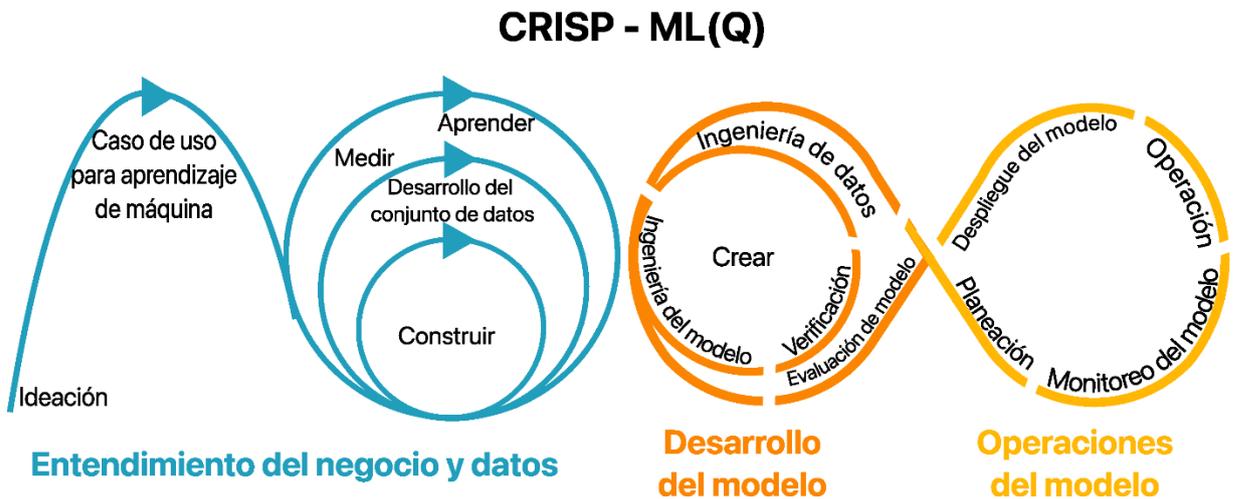


Figura 2 Ciclo de vida del aprendizaje de máquina con CRISP-ML(Q) [23]

A continuación, se describen y aplican las fases de la metodología detalladas en Figura 2.

2.1 Entendimiento del negocio y de los datos

En esta actividad se necesita definir el alcance, los criterios de éxito del proyecto y, sobre todo, determinar la factibilidad del proyecto [21], [23]. El alcance del proyecto se encuentra detallado en la sección 1.3 del documento. Por otro lado, tomando en cuenta los objetivos planteados al inicio de este trabajo, el criterio de éxito es un tiempo de respuesta inferior a los 300 ms. Este tiempo se considera como el tiempo máximo en el que un sistema puede responder para ser considerado como un sistema a tiempo real [24].

Además, se utilizó el lienzo de aprendizaje de máquina (*Machine Learning Canvas*) [25] como un marco de trabajo para confirmar la factibilidad del proyecto mediante la comunicación con las partes interesadas sobre la disponibilidad de datos, restricciones regulatorias y requerimientos no funcionales de la aplicación [23]. El lienzo utiliza la propuesta de valor como eje central para responder al resto de secciones, por lo cual esta sección busca aclarar el objetivo de la aplicación, así como las personas que la van a usar y el impacto que puede tener en ellas. Las secciones a la izquierda de la propuesta de valor

tienen como objetivo determinar el tipo de tarea de predicción, las entradas y salidas, cómo las predicciones realizadas proporcionan el valor que necesita el usuario, cuándo se realizan las predicciones sobre nuevos datos y cómo se van a medir las predicciones. Por otro lado, las secciones a la derecha de la propuesta de valor buscan determinar cómo se va a aprender de los datos, para lo cual se determinan las fuentes de datos crudos, el tiempo de recolección de datos, cuál será la representación extraída desde los datos crudos, y cuándo se deben crear o actualizar modelos. La parte final del lienzo busca medir cuán bien funciona el sistema después del desarrollo, de modo que se pueda cuantificar la creación de valor [25]. El lienzo de aprendizaje de máquina completado para este proyecto se encuentra disponible en el Anexo I.

Gracias a la aplicación del lienzo, se pudo determinar que la propuesta de valor (u objetivo de negocio) es mejorar la facilidad de interacción humano computador mediante una interacción directa y natural. Además, se ha comprobado la disponibilidad de un conjunto de datos de considerable tamaño, con 42 usuarios para entrenamiento y 43 usuarios para pruebas. Por otra parte, para determinar la aplicabilidad de la tecnología de aprendizaje de máquina para este sistema, existen evidencias en la literatura de los buenos resultados al usar redes neuronales convolucionales LSTM [26], [27].

La recolección de datos no fue realizada dado que el conjunto de datos ya se encontraba disponible, el proceso de adquisición se describe en [28]. El enlace al conjunto de datos se puede encontrar en el Anexo II. Los sensores Myo Armband y GForce retornan un vector normalizado y discreto $\mathbf{E}_{n,\omega} \in [-1,1]^8$ para cada $n,\omega \in \mathbb{Z}^+ \times \mathbb{Z}^+$, donde n es un instante de tiempo, ω un número de muestra y cada componente $E_i_{n,\omega} \in [-1,1]$ contiene la medición del canal i registrado por el sensor, con $i = 1,2, \dots, 8$ [29]. Los sensores previamente mencionados y la enumeración de los canales se muestran en Figura 3. Del mismo modo, el sensor retorna un vector de rotación $\theta_n \in \mathbb{R}^4$ que representa los componentes w, x, y, z del cuaternión, tal que $|\theta_n| = 1$ [30]. Tomando en cuenta que el sensor registra estos vectores por intervalos de tiempo determinados, como resultado se obtienen dos señales: $\underline{S} \in [-1,1]^{I \times 8}$ para los EMGs y $\underline{Q} \in \mathbb{R}^{J \times 4}$ para los cuaterniones, con $I, J \in \mathbb{Z}^+$.



Figura 3 Mapeo de canales enumerado en los sensores Myo Armband y GForce Pro.

Tras haber realizado la descripción de los datos, se recopilaron los requerimientos que determinan las condiciones para considerar a un conjunto de muestras de usuario como válidas. Las propiedades requeridas para las muestras se detallan en la Tabla 3.

Tabla 3 Propiedades y valores admitidos de un conjunto de muestras de un usuario

Propiedad	Valor admitido
Número de muestras por usuario	360
Número de muestras por gesto	30
Intervalo de tiempo para el registro de una muestra	5 segundos
Frecuencia de muestreo del EMG	200 Hz (Myo) ó 500 Hz (GForce)
Frecuencia de muestreo de los cuaterniones	50 Hz
Número de canales del EMG	8
Forma de representación de los cuaterniones	Vector de coeficientes del cuaternión (w, x, y, z)
Tipo de dato de la etiqueta del gesto	Categorical, escogido del conjunto de clases compuesto por: <i>backward</i> , <i>forward</i> , <i>up</i> , <i>down</i> , <i>left</i> , <i>right</i> , <i>fist</i> , <i>pinch</i> , <i>waveIn</i> , <i>waveOut</i> , <i>relax</i>
Número máximo de muestras con valores faltantes (<i>missing values</i>)	4

Para finalizar, se realizó una verificación de los datos tomando en cuenta los requerimientos antes mencionados, pero con énfasis especial en muestras con valores faltantes o anómalos. El documento de verificación de las muestras se puede encontrar en el Anexo III.

2.2 Preparación de datos

El objetivo de esta actividad es producir un conjunto de datos que se pueda utilizar como entrada para la actividad de modelado. Para lograr esta salida, se tienen tareas de selección, limpieza, construcción y estandarización de datos [21].

Para la selección de datos se utilizó el software desarrollado en [28], el cual propone dos representaciones para las señales. Una señal EMG definida por $emg' t = \sum_{i=1}^8 \phi |x_i t|$, donde $x_i t$ es la señal EMG medida del i -ésimo canal centrada en cero y ϕ es un filtro Butterworth pasa bajos de quinto orden. Para la señal de los cuaterniones, se propone una señal filtrada $\theta' t = \sum_{i \in \{w,x,y,z\}} |\theta_i t - \bar{\theta}_i t|$, donde $\theta_i t$ representa el vector del componente w, x, y, z de los cuaterniones y $\bar{\theta}_i t$ es el promedio de cada componente de los cuaterniones.

Después de la selección de características se realizó la selección de datos, para lo cual se realizaron dos tareas. La primera tarea es una segmentación manual que consiste en registrar referencias o marcas temporales (*Ground Truth*) en las señales, de modo que delimiten la actividad muscular o de rotación para cada muestra del conjunto de datos. Se utilizó el programa desarrollado en [28] para la segmentación manual de los datos. La segunda tarea involucra la identificación de ventanas de señal cuya información se encuentre en un subconjunto del intervalo de tiempo delimitado en la tarea anterior (*Ground Truth*). Para esto, se recorren las señales usando una ventana deslizante y contando la cantidad de puntos que se encuentran dentro del intervalo. La selección de una ventana se determina mediante dos umbrales: el 75% o más puntos de la ventana son puntos del *ground truth*, o el 50% o más puntos del *ground truth* se encuentran en la ventana. También se computó un vector de características de los datos usando la función energía [31], representada por $E = \sum_{i=2}^L |x_i \cdot |x_i| - x_{i-1} \cdot |x_{i-1}|$, donde x_i es una muestra de una señal

EMG (o de cuaterniones) y L es la longitud total de la señal EMG (o de cuaterniones). Este vector se usará como una de las entradas para la actividad de modelamiento.

Durante la limpieza de datos, se aplicó una reducción del ruido sobre la señal electromiográfica usando rectificación con el valor absoluto y filtrado de la señal con un filtro Butterworth pasa bajos de quinto orden y una frecuencia de corte de 10 Hz [26]. Para la señal de los cuaterniones se aplicó una función de normalización definida por $N t; \theta_i = \frac{\theta_i t - \min \theta_i}{\max \theta_i - \min \theta_i}$, donde θ_i representa el vector del componente w, x, y, z de los cuaterniones en el intervalo de tiempo $T \subset \mathbb{Z}^+$, con $t \in T$.

La construcción de los datos utiliza características derivadas a partir de las existentes, pues transforma los datos desde el dominio temporal al dominio tiempo frecuencia mediante el cómputo de espectrogramas sobre ventanas de señal para cada uno de los canales de la señal EMG \underline{S} [32], y para cada vector de un componente de la señal de cuaterniones \underline{Q} . Si bien la construcción de datos en este trabajo es análoga a la planteada en [26], sufre modificaciones en lo que respecta a los tamaños de ventana deslizante, tamaño del deslizamiento (*stride*), ventana deslizante interna y puntos de superposición (*overlapping*). Esto se debe a que el conjunto de datos cuenta con dos frecuencias de muestreo distintas para las señales EMG, como se mostró en la sección anterior. De ahí que en este trabajo los tamaños de las variables anteriormente mencionadas estén representadas mediante el tiempo, y no mediante número de puntos. Con esta nueva representación, la cantidad de información en la ventana es siempre equivalente entre datos tomados con un sensor u otro. Un resumen de la selección, limpieza y construcción de datos para una señal EMG se muestra en Figura 4. El procedimiento para una señal de cuaterniones es análogo al de la señal EMG, excepto en la limpieza (Figura 4.b) pues para este tipo de señales se utiliza la normalización.

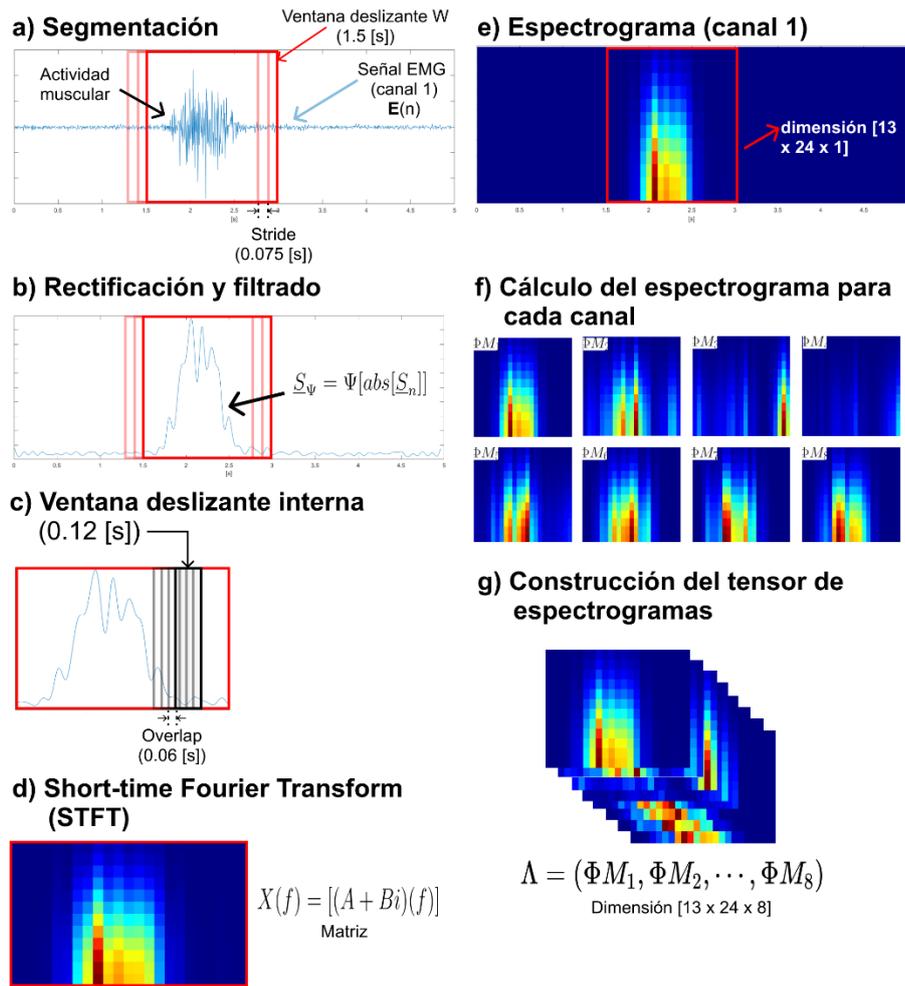


Figura 4 Selección, limpieza y construcción de datos para una señal EMG.

Para finalizar, se estandarizaron los datos generados mediante una estructura de archivo que contiene una secuencia de tensores EMG y/o de cuaterniones junto con sus etiquetas, su marca de tiempo y el vector de *Ground Truth*, como se muestra en Figura 5.

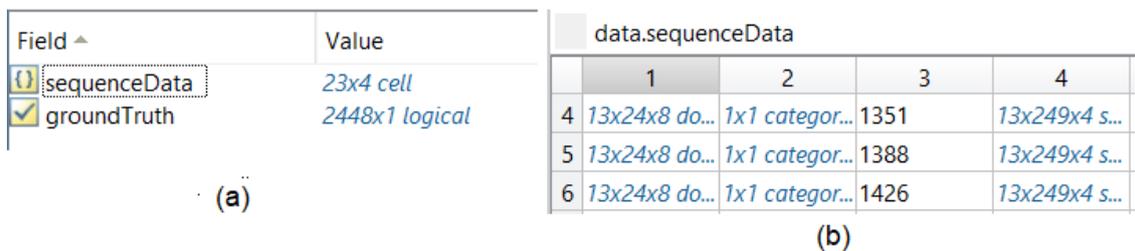


Figura 5 Formato de archivo para almacenamiento de características. a) Formato inicial del archivo, b) formato de almacenamiento para la secuencia de datos

Los enlaces a los códigos utilizados para esta sección se encuentran en el Anexo II.

2.3 Modelamiento

Esta actividad persigue la creación de uno o múltiples modelos que satisfagan las restricciones y entreguen el valor planteado en la actividad de entendimiento del negocio y de los datos. Tomando en cuenta que la selección del modelo ya ha sido definida, la literatura ha demostrado de manera empírica [27] que las redes neuronales CNN LSTM son capaces de aprender a reconocer gestos de diferente duración y complejidad. A continuación, se detalla la arquitectura del modelo seleccionado, así como el detalle de los optimizadores, de modo que se pueda asegurar la reproducibilidad a nivel de método [21].

La arquitectura de este modelo tiene dos partes. La primera consiste en una red neuronal convolucional que permite la extracción de características implícitas y puede aprender a representar imágenes a través de dominios [27]. La estructura propuesta para la red convolucional se basa en la arquitectura usada por GoogLeNet [33], la cual introduce capas dispersas dentro de las capas de convolución con miras a solucionar los problemas de *overfitting* y el alto uso de recursos computacionales en este tipo de redes neuronales artificiales. La capa base de convolución dispersa usada para este trabajo es la propuesta en [26], y su estructura se detalla en Figura 6.

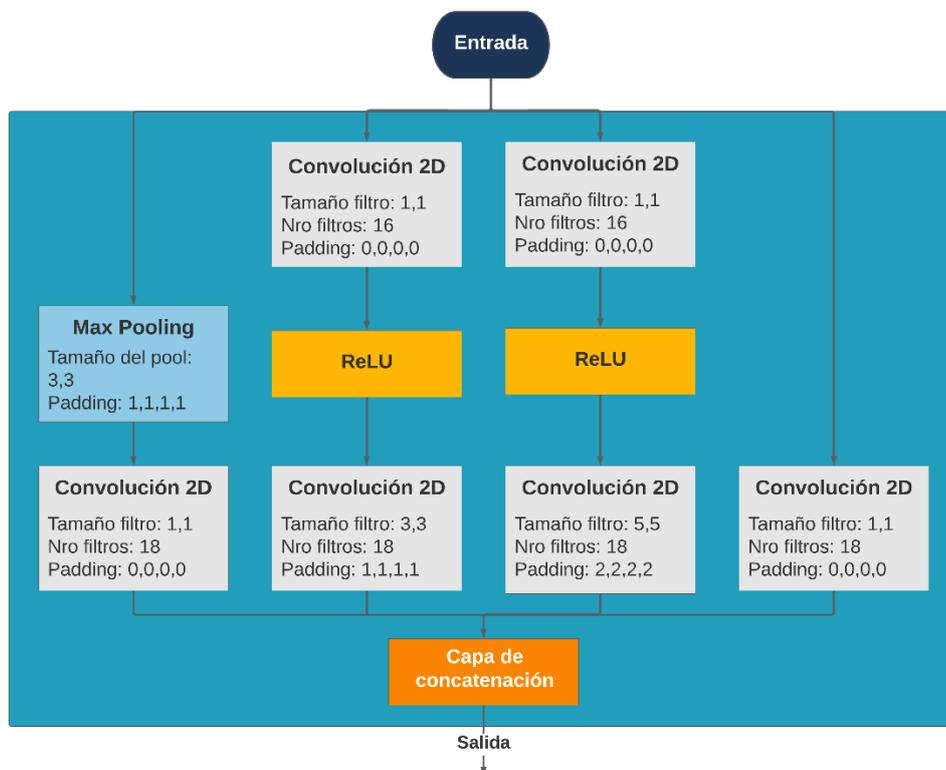


Figura 6 Capa convolucional dispersa

La capa de extracción convolucional de características se construyó usando seis capas dispersas y dos bloques residuales aditivos [26]. La estructura completa de la capa de extracción de características se puede visualizar en la Figura 7.

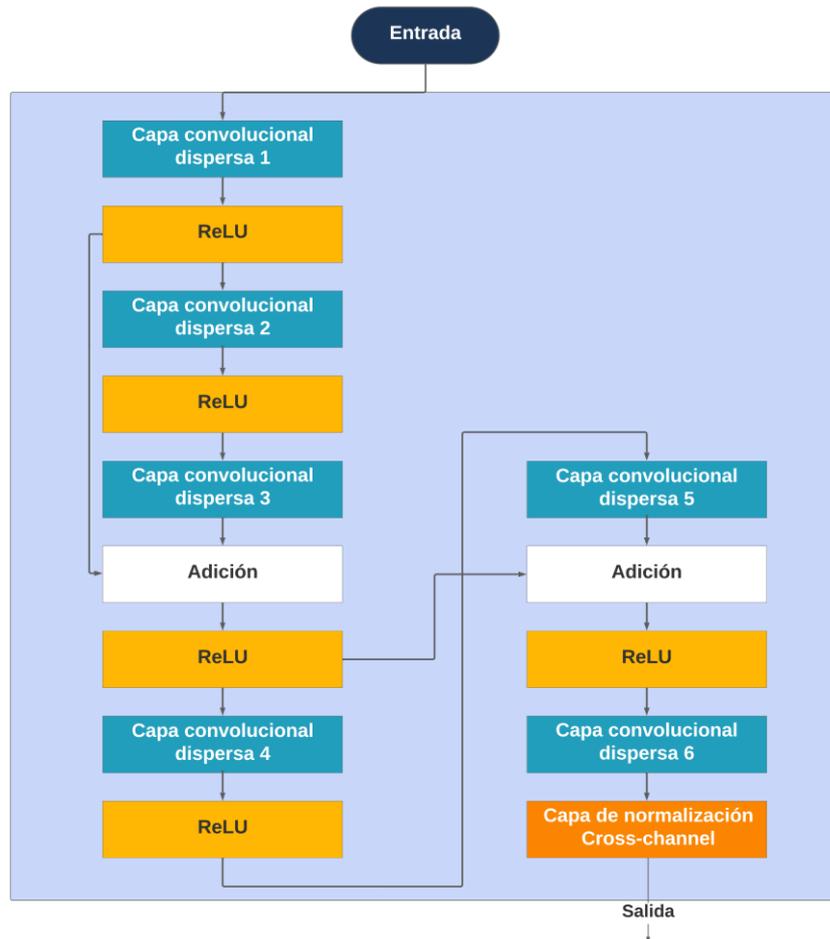


Figura 7 Capa de extracción convolucional de características

La segunda parte de la arquitectura consiste en una red neuronal LSTM, que tiene la habilidad de aprender dependencias sobre datos secuenciales de un largo rango. Además, este tipo de redes tratan las limitaciones de las redes neuronales *feed-forward*, como son el uso de ventanas de tamaño fijo y la falta de flexibilidad en el aprendizaje de secuencias de diferentes tamaños [27]. La siguiente capa de la arquitectura es una capa completamente conectada, seguida de una función de activación *softmax* y una capa de salida. Para este trabajo, se utilizaron arquitecturas similares para los modelos de gestos dinámicos y estáticos a crear, donde la diferencia radica en la adición de una capa de *dropout* para disminuir el *overfitting*. El detalle por capas de la arquitectura completa del modelo se muestra en Figura 8.

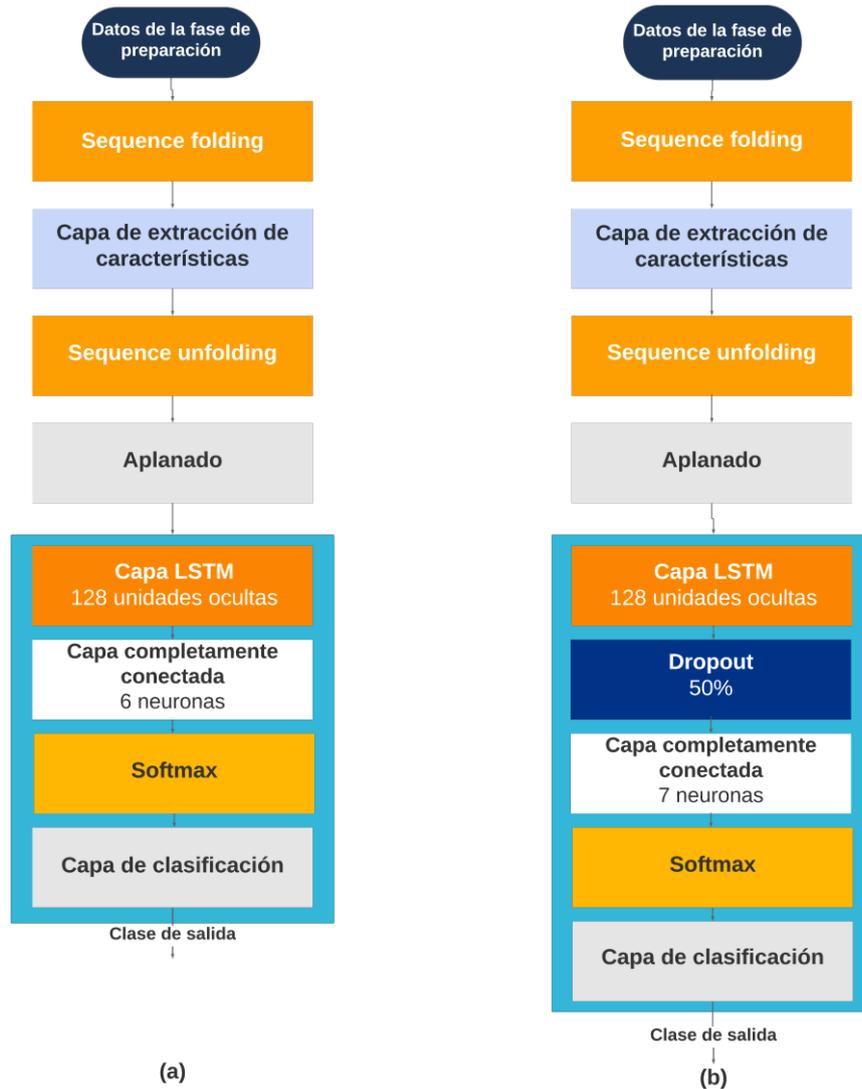


Figura 8 Arquitecturas CNN - LSTM para reconocimiento de a) gestos estáticos y b) gestos dinámicos

Tras haber escogido arquitecturas para modelos de gestos estáticos y dinámicos, se realizó una serie de entrenamientos para encontrar las configuraciones del optimizador adecuadas para el modelo de gestos dinámicos. Cabe mencionar que, en ambos casos, la entrada para el entrenamiento es una secuencia de tensores de EMGs y de cuaterniones, dependiendo del modelo, junto con sus etiquetas. También se realizó el entrenamiento de un clasificador lineal logístico para determinar en base a un vector de características basado en la energía si el gesto es dinámico o estático, y de esta manera conectar los dos modelos previamente mencionados. Los enlaces al código adaptado y desarrollado para

los modelos entrenados se encuentran en el Anexo II. Las configuraciones de los optimizadores se encuentran en el Anexo IV.

2.4 Evaluación

Esta actividad tiene como objetivo evaluar el rendimiento del modelo entrenado sobre un conjunto de datos de prueba [23]. Para los modelos CNN-LSTM entrenados en la sección 2.3, esta actividad es directa dado que la validación ocurrió durante el mismo entrenamiento del modelo. Sin embargo, para tener mayor granularidad en la evaluación, se calcularon las matrices de confusión para cada uno de los modelos. Por otra parte, el clasificador lineal logístico se evaluó utilizando los datos para validación disponibles en la porción de entrenamiento del conjunto de datos.

También es necesario evaluar a los modelos de forma conjunta, para poder determinar cuál es la exactitud de clasificación y de reconocimiento cuando el sistema de reconocimiento tiene como entrada cualquier gesto, ya sea estático o dinámico. Para esto, se propone un procedimiento (*pipeline*) el cual empieza con las señales EMG y de los cuaterniones, se calcula un vector de 12 características utilizando la función energía, se utiliza el clasificador lineal logístico como un *switch* que determina si el gesto es estático o dinámico, se preprocesa y clasifica el gesto usando uno de los dos modelos CNN-LSTM, y se obtiene un vector de instantes de tiempo y etiquetas. Mediante un algoritmo de posprocesamiento [26] se obtiene consistencia entre la secuencia de etiquetas, se recopilan estos vectores para cada muestra, y se consolidan en un solo archivo el cual se envía a una aplicación de pruebas que posee los datos reales de las clases y el *Ground Truth*, para obtener las métricas de exactitud antes mencionadas. Es importante notar que las etiquetas obtenidas en cualquiera de los modelos deben superar el umbral del 50% de puntuación de predicción para no ser etiquetadas con “noGesture”. El esquema del procedimiento se muestra en Figura 9.

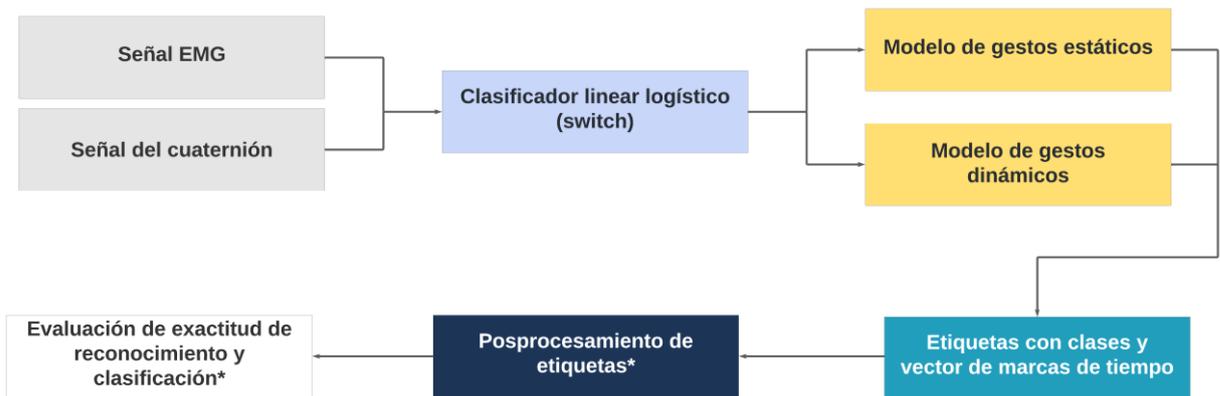


Figura 9 Procedimiento para el funcionamiento y evaluación de la aplicación desarrollada. Las etapas marcadas con * son etapas exclusivas de evaluación.

Los resultados de la evaluación para cada uno de los modelos y de la aplicación de aprendizaje de máquina se describen en la sección 3.

2.5 Despliegue

Durante esta actividad, se integran los modelos de aprendizaje de máquina con el resto del producto de software. Se utilizará el procedimiento mostrado en la Figura 9 para el funcionamiento de la aplicación. Además, se utilizará el kit de desarrollo de software (SDK) del sensor Myo Armband para obtener la entrada de gestos tomada por el sensor como punto de partida de la aplicación, como se muestra en Figura 10 [34].

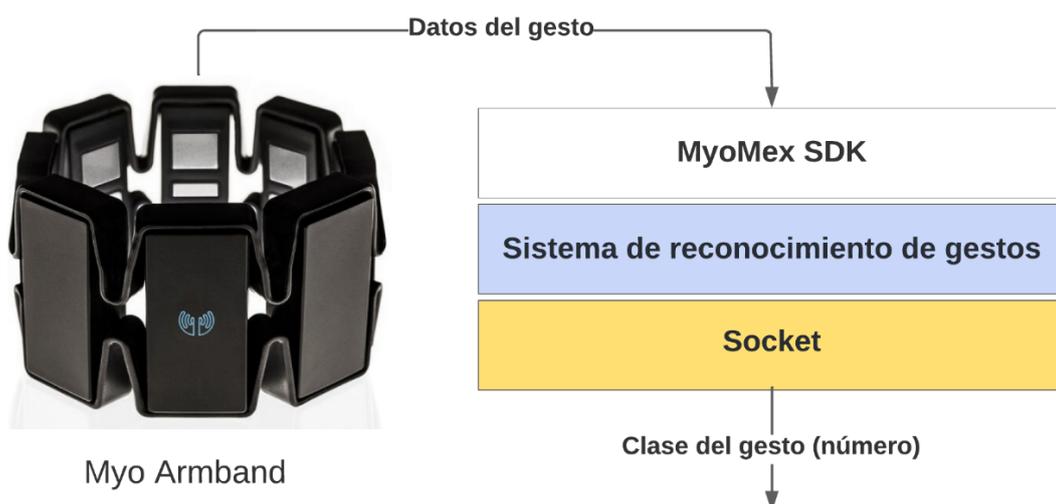


Figura 10 Esquema de comunicación del sistema de reconocimiento de gestos con la aplicación del componente dos

Para la puesta en marcha de ambos componentes, se requiere una interfaz de comunicación que le permita a la aplicación del componente dos recibir información sobre el gesto realizado por el usuario. Se ha escogido una comunicación entre sockets considerando que las aplicaciones funcionarán en el mismo dispositivo. En esta comunicación, el sistema de reconocimiento envía un número que representa a uno de los once gestos a la aplicación del componente dos (Figura 10). El enlace del código de conexión del sistema de reconocimiento de gestos con la aplicación del componente 2 se encuentra disponible en el Anexo II.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

Tras realizar el entrenamiento de dos modelos CNN-LSTM para gestos dinámicos y estáticos, se han calculado las matrices de confusión para cada uno de los modelos. Los datos usados para esta evaluación pertenecen al conjunto de datos del directorio de validación que poseen etiquetas de las clases a las que pertenece cada muestra. La Tabla 4 muestra los resultados de la evaluación sobre cada uno de los gestos estáticos, con una exactitud de clasificación del 84.49 % sobre este conjunto de datos. De igual manera, la Tabla 5 muestra los resultados de evaluación sobre cada uno de los gestos dinámicos, con una exactitud de clasificación del 80.77 %.

Tabla 4 Matriz de confusión del modelo de gestos estáticos sobre el conjunto de datos de validación

		Hand gestures - validation							
True Class	fist	11000	278	58	361	468		90.4%	9.6%
	noGesture	883	23995	1017	2103	1583	1255	77.8%	22.2%
	open	171	438	9402	1350	114	753	76.9%	23.1%
	pinch	45	225	126	11715	117	8	95.7%	4.3%
	waveIn	309	359	135	367	10659	88	89.4%	10.6%
	waveOut	2	471	579	339	157	10378	87.0%	13.0%
		88.6%	93.1%	83.1%	72.2%	81.4%	83.1%		
		11.4%	6.9%	16.9%	27.8%	18.6%	16.9%		
		fist	noGesture	open	pinch	waveIn	waveOut	Predicted Class	

Tabla 5 Matriz de confusión del modelo de reconocimiento de gestos dinámicos sobre el conjunto de datos de validación

		Hand gestures - validation								
True Class	noGesture	27798	1838	2282	1770	1551	2285	1950	70.4%	29.6%
	up	351	11250	70	285	459	205	1567	79.3%	20.7%
	down	557	93	11889	36	48	802	59	88.2%	11.8%
	left	283	281	58	12541	177	91	739	88.5%	11.5%
	right	190	615	56	145	12552	157	671	87.3%	12.7%
	forward	723	96	656	104	48	12408	99	87.8%	12.2%
	backward	218	1201	65	427	400	63	11461	82.8%	17.2%
			92.3%	73.2%	78.9%	81.9%	82.4%	77.5%	69.3%	
		7.7%	26.8%	21.1%	18.1%	17.6%	22.5%	30.7%		
		noGesture	up	down	left	right	forward	backward	Predicted Class	

Para el clasificador lineal logístico se realizó un entrenamiento usando los datos en bruto de usuarios del directorio de entrenamiento del conjunto de datos. Se definió un conjunto de usuarios y se utilizó la mitad de sus muestras para entrenamiento, y la otra mitad para evaluación. Los resultados de la evaluación de este modelo sobre los datos de prueba alcanzan en promedio un 93.02 % de exactitud de clasificación. El porcentaje de clasificación por cada usuario se puede revisar en el Anexo V.

Se ha podido comprobar que, de manera individual, cada modelo tiene una exactitud de clasificación deseable. Sin embargo, se debe evaluar el comportamiento del sistema de reconocimiento de gestos de manera holística, para lo cual la evaluación fue llevada a cabo por otra persona, y sobre la porción del conjunto de datos de validación que no contenía etiquetas ni valores de *Ground Truth*. En Figura 11 y Figura 12 se pueden observar los resultados obtenidos para el sistema de reconocimiento, donde la exactitud de clasificación y de reconocimiento es $55.33\% \pm 18.45\%$ y $49.15\% \pm 18.41\%$, respectivamente.

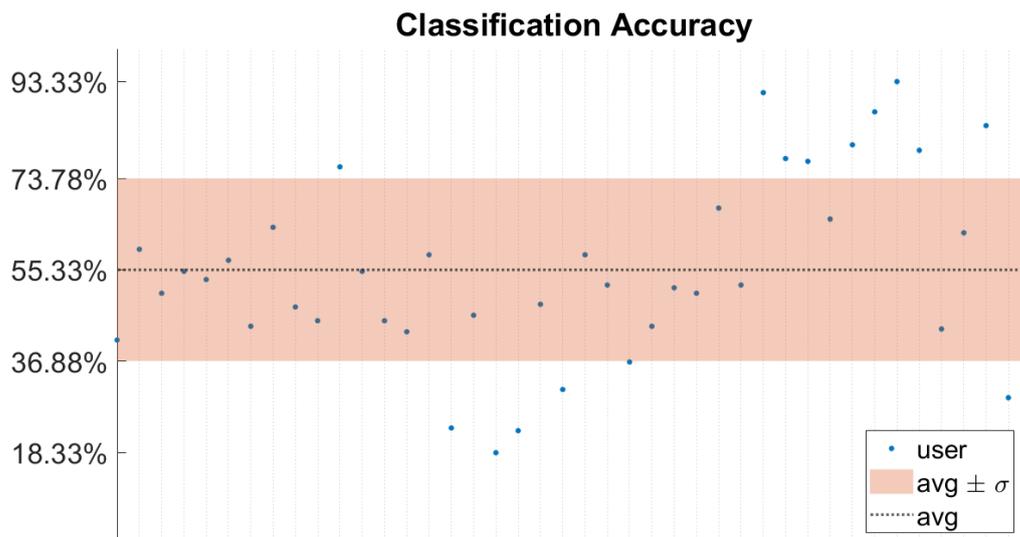


Figura 11 Exactitud de clasificación para el sistema de reconocimiento de gestos desarrollado

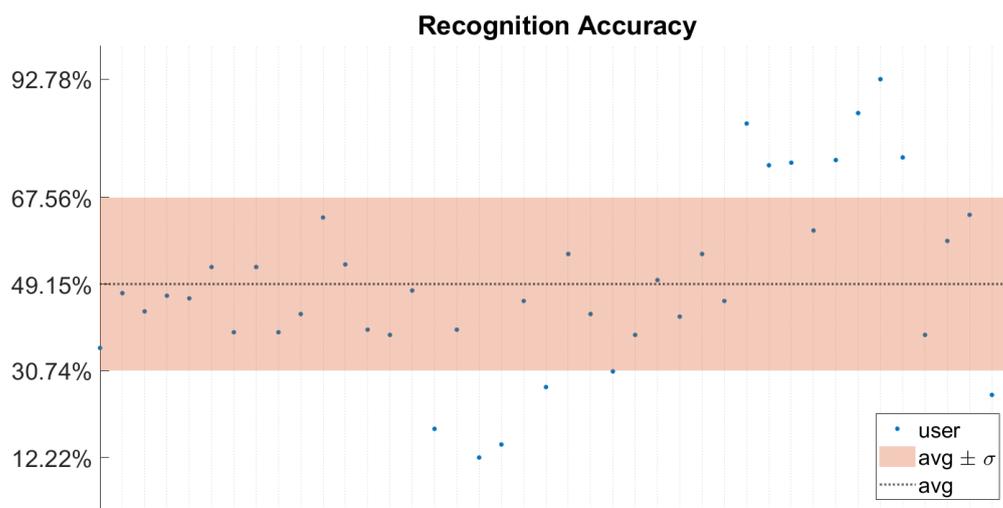


Figura 12 Exactitud de reconocimiento para el sistema de reconocimiento de gestos desarrollado

Del mismo modo, la Tabla 6 muestra los resultados de evaluación sobre el modelo, detallando la cantidad de muestras correcta e incorrectamente clasificadas. También se puede observar que para los gestos dinámicos existe una cantidad considerable de muestras mal clasificadas como “up”.

Tabla 6 Matriz de confusión del sistema de reconocimiento de gestos desarrollado

		Confusion Matrix													
Output Class	forward	273 3.6%	2 0.0%	7 0.1%	34 0.4%	1 0.0%	7 0.1%	0 0.0%	9 0.1%	18 0.2%	21 0.3%	47 0.6%	34 0.4%	60.3%	
	fist	13 0.2%	419 5.5%	62 0.8%	6 0.1%	27 0.4%	6 0.1%	14 0.2%	54 0.7%	3 0.0%	19 0.3%	24 0.3%	1 0.0%	64.7%	
	waveIn	47 0.6%	134 1.8%	402 5.3%	12 0.2%	38 0.5%	36 0.5%	8 0.1%	56 0.7%	10 0.1%	5 0.1%	63 0.8%	21 0.3%	48.3%	
	right	31 0.4%	0 0.0%	0 0.0%	231 3.1%	0 0.0%	7 0.1%	1 0.0%	0 0.0%	67 0.9%	91 1.2%	6 0.1%	42 0.6%	48.5%	
	waveOut	4 0.1%	0 0.0%	14 0.2%	0 0.0%	427 5.6%	2 0.0%	0 0.0%	57 0.8%	11 0.1%	5 0.1%	2 0.0%	2 0.0%	81.5%	
	pinch	58 0.8%	59 0.8%	108 1.4%	15 0.2%	84 1.1%	539 7.1%	21 0.3%	190 2.5%	12 0.2%	13 0.2%	36 0.5%	20 0.3%	46.7%	
	relax	4 0.1%	6 0.1%	21 0.3%	4 0.1%	9 0.1%	13 0.2%	586 7.8%	21 0.3%	6 0.1%	5 0.1%	3 0.0%	2 0.0%	86.2%	
	open	7 0.1%	4 0.1%	5 0.1%	2 0.0%	26 0.3%	1 0.0%	0 0.0%	236 3.1%	0 0.0%	9 0.1%	9 0.1%	0 0.0%	78.9%	
	backward	27 0.4%	1 0.0%	2 0.0%	78 1.0%	3 0.0%	1 0.0%	0 0.0%	1 0.0%	164 2.2%	124 1.6%	8 0.1%	37 0.5%	36.8%	
	up	23 0.3%	2 0.0%	4 0.1%	165 2.2%	13 0.2%	12 0.2%	0 0.0%	1 0.0%	244 3.2%	228 3.0%	22 0.3%	118 1.6%	27.4%	
	down	116 1.5%	2 0.0%	5 0.1%	50 0.7%	2 0.0%	4 0.1%	0 0.0%	5 0.1%	40 0.5%	57 0.8%	391 5.2%	66 0.9%	53.0%	
	left	27 0.4%	1 0.0%	0 0.0%	33 0.4%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	55 0.7%	53 0.7%	19 0.3%	287 3.8%	60.2%	
			43.3%	66.5%	63.8%	36.7%	67.8%	85.6%	93.0%	37.5%	26.0%	36.2%	62.1%	45.6%	55.3%
			56.7%	33.5%	36.2%	63.3%	32.2%	14.4%	7.0%	62.5%	74.0%	63.8%	37.9%	54.4%	44.7%
		forward	fist	waveIn	right	waveOut	pinch	relax	open	backward	up	down	left		
		Target Class													

Además, se debe verificar si el sistema cumple con el criterio de éxito referente al tiempo de respuesta. Para ello, durante la preparación de datos para la fase de evaluación del sistema, se cronometró el tiempo desde el procesamiento del vector de características para el clasificador lineal logístico, hasta la obtención de las etiquetas y marcas de tiempo. La Figura 13 muestra la distribución de los tiempos de procesamiento, de donde se determinó que el tiempo promedio de procesamiento del sistema es 29.26 ± 19.82 [ms].

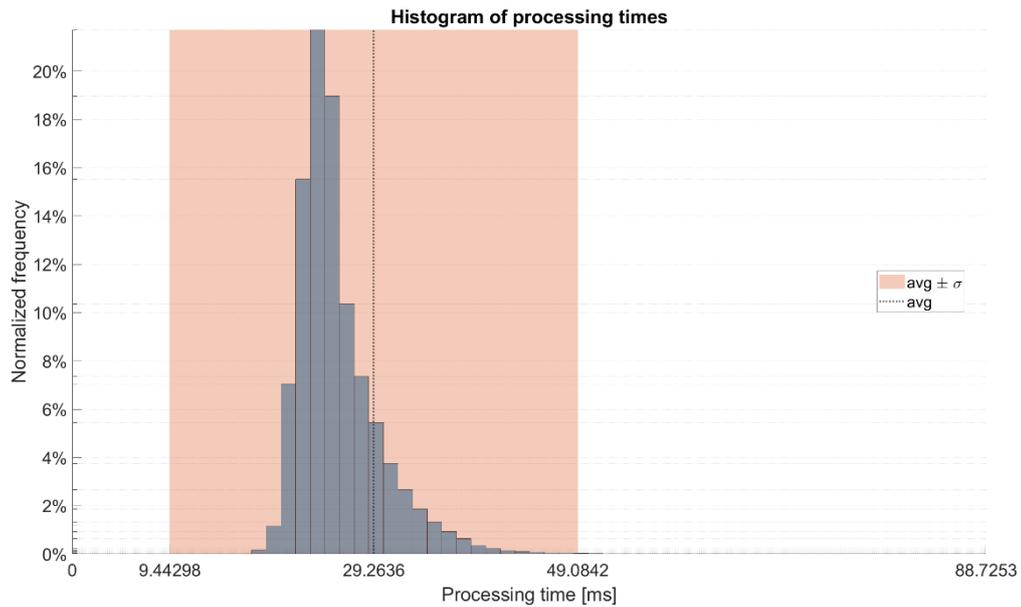


Figura 13 Histograma de tiempos de procesamiento de las señales del sistema desarrollado

3.2 Conclusiones

En el presente trabajo de integración curricular, se han introducido algunos métodos, herramientas y prácticas de la ingeniería de software para el desarrollo de aplicaciones de aprendizaje de máquina, como la comunicación con las partes interesadas, la reducción de la deuda técnica y el enfoque en el valor que se entrega a los usuarios mediante el uso de la metodología CRISP-ML(Q).

Si bien se ha podido constatar que hasta el momento no se pueden aplicar todos los principios de la ingeniería de software a los proyectos de aprendizaje de máquina, esta rama de la ingeniería va adaptándose para ayudar a solucionar el problema que afectaba a las aplicaciones tradicionales hace más de cinco décadas: la gran tasa de fallo de los proyectos de software (en este caso, proyectos de aprendizaje de máquina).

Se ha desarrollado un sistema de reconocimiento para once gestos de la mano utilizando electromiografía y unidades de medición inercial con redes neuronales convolucionales LSTM, y se ha determinado una exactitud de clasificación del $55.33\% \pm 18.45\%$, junto con

una exactitud de reconocimiento del $49.15\% \pm 18.41\%$ para el sistema de reconocimiento propuesto.

Se ha verificado de forma empírica que la arquitectura usada para los modelos de reconocimiento permite que el sistema trabaje en tiempo real, con un tiempo de respuesta de 29.26 ± 19.82 [ms]. Esto permite que, tras integrar el sistema de reconocimiento con la aplicación desarrollada en el componente dos del proyecto, parte de la experiencia del usuario en la aplicación no sea vea afectada por el sistema de reconocimiento.

3.3 Recomendaciones

Después de haber realizado una revisión exhaustiva de la literatura relacionada con la ingeniería de software y el desarrollo de aplicaciones de aprendizaje de máquina, resulta muy necesario continuar aproximando estas dos áreas mediante la creación, exploración y aplicación de metodologías y modelos de procesos que guíen el ciclo de vida de una aplicación de aprendizaje de máquina. Sobre todo, si se tiene en cuenta que la inteligencia artificial es una de las áreas que guiará la nueva revolución industrial.

Se sugiere implementar prácticas de integración y despliegue continuos para poder incrementar la cantidad de experimentos sobre la arquitectura utilizada en el trabajo y así incrementar la capacidad de reconocimiento del sistema de reconocimiento de gestos.

Se podría simplificar la complejidad del código relacionado con la selección y construcción de datos con la adición de un arreglo *Ground Truth* que tenga la misma longitud que una señal de cuaterniones.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] M. Oudah, A. Al-Naji, and J. Chahl, "Imaging Hand Gesture Recognition Based on Computer Vision: A Review of Techniques", doi: 10.3390/jimaging6080073.
- [2] M. E. Benalcazar *et al.*, "Real-time hand gesture recognition using the Myo armband and muscle activity detection," in *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, Oct. 2017, pp. 1–6. doi: 10.1109/ETCM.2017.8247458.
- [3] L. A. Estrada Jiménez, M. E. Benalcázar, and N. Sotomayor, "Gesture recognition and machine learning applied to sign language translation," *IFMBE Proc*, vol. 60, pp. 233–236, 2017, doi: 10.1007/978-981-10-4086-3_59/COVER.
- [4] L. Lamberti and F. Camastra, "Handy: A real-time three color glove-based gesture recognizer with learning vector quantization," *Expert Syst Appl*, vol. 39, no. 12, pp. 10489–10494, Sep. 2012, doi: 10.1016/J.ESWA.2012.02.081.
- [5] T. Mantecón, C. R. del-Blanco, F. Jaureguizar, and N. García, "Hand gesture recognition using infrared imagery provided by leap motion controller," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10016 LNCS, pp. 47–57, 2016, doi: 10.1007/978-3-319-48680-2_5/COVER.
- [6] W. Lu, Z. Tong, and J. Chu, "Dynamic hand gesture recognition with leap motion controller," *IEEE Signal Process Lett*, vol. 23, no. 9, pp. 1188–1192, Sep. 2016, doi: 10.1109/LSP.2016.2590470.
- [7] A. Moschetti, L. Fiorini, D. Esposito, P. Dario, and F. Cavallo, "Recognition of Daily Gestures with Wearable Inertial Rings and Bracelets," *Sensors (Basel)*, vol. 16, no. 8, Aug. 2016, doi: 10.3390/S16081341.
- [8] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface EMG images OPEN," 2016, doi: 10.1038/srep36571.
- [9] Y. Huang *et al.*, "Preliminary Testing of a Hand Gesture Recognition Wristband Based on EMG and Inertial Sensor Fusion," 2015, pp. 359–367. doi: 10.1007/978-3-319-22879-2_33.
- [10] M. Georgi, C. Amma, and T. Schultz, "Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing", doi: 10.5220/0005276900990108.

- [11] X. Chen, X. Zhang, Z. Y. Zhao, J. H. Yang, V. Lantz, and K. Q. Wang, "Hand gesture recognition research based on surface EMG sensors and 2D-accelerometers," *Proceedings - International Symposium on Wearable Computers, ISWC*, pp. 11–14, 2007, doi: 10.1109/ISWC.2007.4373769.
- [12] M. T. Wolf *et al.*, "Decoding static and dynamic arm and hand gestures from the JPL BioSleeve," *IEEE Aerospace Conference Proceedings*, 2013, doi: 10.1109/AERO.2013.6497171.
- [13] Algorithmia, "2020 state of enterprise machine learning." Accessed: Apr. 13, 2022. [Online]. Available: https://info.algorithmia.com/hubfs/2019/Whitepapers/The-State-of-Enterprise-ML-2020/Algorithmia_2020_State_of_Enterprise_ML.pdf?utm_campaign=The%20Batch&utm_source=hs_email&utm_medium=email&utm_content=80984419&_hsenc=p2ANqtz--sz-e2gfgUeDvVSmjsXfvwOnLHB2ZkSdQsO1IRRAdnBlb0emf-JTh8NnwFxB-FeZberlw7_rI9ERTy8zFW8jvoTzjOfA&_hsmi=80984419
- [14] D. Sculley *et al.*, "Hidden Technical Debt in Machine Learning Systems".
- [15] B. Fritz, "Software Engineering: A report on a conference sponsored by NATO Science Committee," 1968.
- [16] B. W. Boehm, "Software Engineering," *IEEE TRANSACTIONS ON COMPUTERS*, vol. 25, no. 12, 1976.
- [17] R. S. Pressman and B. R. Maxim, "Software engineering : a practitioner's approach," p. 671.
- [18] Ieee, "IEEE Standard Glossary of Software Engineering Terminology," *Office*, vol. 121990, no. 1, p. 1, 1990, doi: 10.1109/IEEESTD.1990.101064.
- [19] I. Sommerville, "Software engineering," p. 810, 2016.
- [20] R. Dawson, "Why is DevOps for Machine Learning so Different? | HackerNoon," Nov. 11, 2019. <https://hackernoon.com/why-is-devops-for-machine-learning-so-different-384z32f1> (accessed Aug. 02, 2022).
- [21] S. Studer *et al.*, "Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology", Accessed: Feb. 18, 2022. [Online]. Available: <https://doi.org/>

- [22] R. Wirth and J. Hipp, "CRISP-DM: Towards a Standard Process Model for Data Mining".
- [23] "CRISP-ML(Q)." <https://ml-ops.org/content/crisp-ml> (accessed Aug. 02, 2022).
- [24] H. Mizuno, N. Tsujiuchi, and T. Koizumi, "Forearm motion discrimination technique using real-time EMG signals," *Annu Int Conf IEEE Eng Med Biol Soc*, vol. 2011, pp. 4435–4438, 2011, doi: 10.1109/IEMBS.2011.6091100.
- [25] L. Dorard, "From Data to AI with the Machine Learning Canvas (Part I) Own Machine Learning | Medium," Nov. 18, 2016. <https://medium.com/louis-dorard/from-data-to-ai-with-the-machine-learning-canvas-part-i-d171b867b047> (accessed Aug. 02, 2022).
- [26] F. M. Ferri Ripalda, "Desarrollo de un modelo de reconocimiento de gestos de la mano utilizando señales EMG y Deep Learning," Escuela Politécnica Nacional, Quito, 2021.
- [27] E. Tsironi, P. Barros, C. Weber, and S. Wermter, "An analysis of Convolutional Long Short-Term Memory Recurrent Neural Networks for gesture recognition," *Neurocomputing*, vol. 268, pp. 76–86, Dec. 2017, doi: 10.1016/J.NEUCOM.2016.12.088.
- [28] J. Zea, M. E. Benalcázar, L. I. Barona López, and Á. L. Valdivieso Caraguay, "An Open-Source Data Acquisition and Manual Segmentation System for Hand Gesture Recognition based on EMG," *ETCM 2021 - 5th Ecuador Technical Chapters Meeting*, Oct. 2021, doi: 10.1109/ETCM53643.2021.9590811.
- [29] M. E. Benalcázar, A. G. Jaramillo, J. A. Zea, A. Paéz, and V. H. Andaluz, "Hand gesture recognition using machine learning and the myo armband," *25th European Signal Processing Conference, EUSIPCO 2017*, vol. 2017-January, pp. 1040–1044, Oct. 2017, doi: 10.23919/EUSIPCO.2017.8081366.
- [30] P. Bernhardt, "How I learned to Stop Worrying and Love Quaternions," May 15, 2015. <https://developerblog.myo.com/quaternions/> (accessed Aug. 02, 2022).
- [31] L. I. B. López *et al.*, "An Energy-Based Method for Orientation Correction of EMG Bracelet Sensors in Hand Gesture Recognition Systems," *Sensors 2020, Vol. 20, Page 6327*, vol. 20, no. 21, p. 6327, Nov. 2020, doi: 10.3390/S20216327.

- [32] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, "Techniques of EMG signal analysis: detection, processing, classification and applications," *Biol Proced Online*, vol. 8, no. 1, p. 11, Mar. 2006, doi: 10.1251/BPO115.
- [33] C. Szegedy *et al.*, "Going Deeper with Convolutions".
- [34] "Myo SDK MATLAB MEX Wrapper - File Exchange - MATLAB Central." <https://la.mathworks.com/matlabcentral/fileexchange/55817-myo-sdk-matlab-mex-wrapper> (accessed Aug. 03, 2022).

5 ANEXOS

ANEXO I. Lienzo de aprendizaje de máquina

THE MACHINE LEARNING CANVAS Designed for: Reconocimiento de 11 gestos de la mano con CNN LSTM Designed by: Miguel Muñoz

Date: 17-03-2022 Iteration: 1

<p>PREDICTION TASK </p> <p><small>Type of task? Entity on which predictions are made? Possible outcomes? Wait time before observation?</small></p> <p>Tipo de tarea: Clasificación</p> <p>Entradas: Señales electromiográficas y de cuaterniones</p> <p>Salidas: Categoría a la que pertenece un gesto dentro de las clases: wave-in, wave-out, fist, open, pinch, up, down, left, right, forward, backward.</p>	<p>DECISIONS </p> <p><small>How are predictions turned into proposed value for the end-user? Mention parameters of the process / application that does that.</small></p> <p>La clasificación permite identificar si el usuario realizó el gesto indicado en el juego durante el intervalo de tiempo adecuado y determinar su puntuación.</p>	<p>VALUE PROPOSITION </p> <p><small>Who is the end-user? What are their objectives? How will they benefit from the ML system? Mention workflow/interfaces.</small></p> <p>Mejorar la facilidad de interacción humano computador en videojuegos mediante el reconocimiento de gestos de la mano y el brazo, permitiendo una interacción más natural, cómoda y con menos carga cognitiva para el usuario final.</p>	<p>DATA COLLECTION </p> <p><small>Strategy for initial train set & continuous update. Mention collection rate, holdout on production entities, cost/constraints to observe outcomes.</small></p> <p>Existe un programa para la toma de muestras que obtiene la información de los brazaletes con sensores de medición inercial y de EMGs.</p> <p>Tiempo de recolección de datos por usuario: De una hora a una hora y media.</p>	<p>DATA SOURCES </p> <p><small>Where can we get (raw) information on entities and observed outcomes? Mention database tables, API methods, websites to scrape, etc.</small></p> <p>Conjunto de datos del Laboratorio de Inteligencia y Visión Artificial con muestras de 11 gestos sobre 85 usuarios.</p>
<p>OFFLINE EVALUATION </p> <p><small>Can models be deployed? Which test data to assess performance? Cost/gain values for (in)correct decisions? Fairness constraint?</small></p> <p>Métricas:</p> <p>Exactitud de clasificación</p> <p>Exactitud de reconocimiento</p> <p>Los errores de predicción pueden hacer que el usuario pierda el juego o gane puntos.</p> <p>Los datos de prueba son la mitad de todas las muestras del dataset.</p>	<p>MAKING PREDICTIONS </p> <p><small>When do we make real-time / batch pred.? Time available for this + featurization + post-processing? Compute target?</small></p> <p>La clasificación se realiza sobre un lote de señales en un intervalo de tiempo determinado. El tiempo máximo disponible para obtener la clase del gesto son 300 ms.</p>	<p>BUILDING MODELS </p> <p><small>How many prod models are needed? When would we update? Time available for this (including featurization and analysis)?</small></p> <p>Se necesita un único modelo en producción. La actualización del modelo sucederá cuando existan conjuntos de datos sobre nuevas clases de gestos. El tiempo máximo para la construcción del modelo es de 6 meses.</p>	<p>FEATURES </p> <p><small>Input representations available at prediction time, extracted from raw data sources.</small></p> <p>Se extraen características de los datos crudos mediante la segmentación de señales de forma manual y la generación de representaciones altamente redundantes de las señales en los dominios de tiempo-frecuencia</p> <p>Las entradas se representan mediante tensores de espectrogramas de las señales electromiográficas y de las señales de los cuaterniones.</p>	
<p>MONITORING </p> <p><small>Metrics to quantify value creation and measure the ML system's impact in production (on end-users and business)?</small></p> <p>Tiempo de respuesta menor a 300 ms, precisión y exactitud</p>				

Version 1.1. Created by Louis Dorard, Ph.D. Licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/). Please keep this mention and the link to ownml.co when sharing. **OWNML.CO**

ANEXO II. Enlaces

Repositorio en GitHub: [Mikkel-14/LSTMCNN: Código adaptado y desarrollado para un sistema de reconocimiento de 11 gestos de la mano con redes CNN-LSTM \(github.com\)](https://github.com/Mikkel-14/LSTMCNN).

Nota: Solicitar acceso enviando un correo electrónico a jose.munoz01@epn.edu.ec

Vídeo de demostración del producto: <https://youtu.be/gCcrnLGinaQ>

Conjunto de datos utilizado: <https://laboratorio-ia.epn.edu.ec/es/recursos/dataset/emg-imu-epn-100>

A continuación, se presenta el detalle de los archivos usados en este proyecto.

Actividad	Rama del repositorio	Archivo
Preparación de datos (generación de espectrogramas señal EMG)	EstaticGestures	LSTM-CNN/ spectrogramDatasetGeneration.m
Preparación de datos (generación de espectrogramas señal de cuaterniones)	DynamicGestures	exp/ quatSpectrogramDatasetGeneration.m
Modelamiento y evaluación individual (modelo de gestos estáticos)	EstaticGestures	LSTM-CNN/modelCreation.m
Modelamiento y evaluación individual (modelo de gestos dinámicos)	DynamicGestures	LSTM-CNN/modelCreation.m
Modelamiento y evaluación individual (Clasificador lineal logístico)	Exp	exp/ModelSwitch.m
Evaluación (procedimiento de evaluación del sistema combinado)	GesturePipeline	GesturePipeline/trainingEval.m
Despliegue (recolección del gesto y envío de mensaje usando sockets)	GesturePipeline	ConnectionInterface/MyoConnection.m

ANEXO III. Tabla de verificación del conjunto de datos

Número	Colaborador (toma muestras)	Sensor	Nombre Usuario (original)	Nombre ID (nuevo)	Observación	Corrección
1	Alex Chico	Myo-EPN- Marco Benalcázar	Alex Chico	user_001	Sin observación	
2	Alex Chico	Myo-EPN- Marco Benalcázar	Bety Godoy	user_002	Sin observación	
3	Alex Chico	Myo-EPN- Marco Benalcázar	Mishell Cupueran	user_003	Sin observación	
4	Alex Chico	Myo-EPN- Marco Benalcázar	Raul Andres Chico	user_004	(g8 s30)=sin quaternion	Reemplazar una muestra con una buena
5	Alex Chico	Myo-EPN- Marco Benalcázar	RaulChico	user_005	(g7 s5)=demasiada activación en canales(posible ruido)	No se requiere corrección

6	Francis Ferri	gForce_3	AdrianFerri	user_006	(g6 s1-30)=Señal Emg muy baja, (g7 s4,28)=Señal Emg muy baja,(g8 s1,3,5,6,8,16,18,19,20,23,25,26,27,30)=Señal Emg muy baja,(g9 s7,20)=Señal Emg muy baja,(g10 s1-30)=Señal Emg muy baja)	No se requiere corrección
7	Francis Ferri	gForce_3	CarlosFerri	user_007	(g5 s1)=Señal Emg baja, (g6 s1)=Señal Emg muy baja,(g8 s6,9,10,11,12,13,14,15,17,20)=Quaternion muy bajo, (g11 s5,12,14)=Señal Emg muy baja	No se requiere corrección, señales con ruidos extraños
8	Francis Ferri	gForce_3	EstefaniaChinga	user_008	(g1 s6,16,18,21-30)=Señal Emg muy baja,(g4 s17)=Señal Emg muy baja,(g5 s1-30)=Señal emg muy baja,(g6 s1,2,8,9,13,14,16,,17,18,20,23,26,27,29)=Señal Emg muy baja,(g7 s1,5,28,30)=Señal Emg muy baja,(g8 s2,3,24,25,26,27,28,29)=Señal Emg muy baja,(g9 s15,25)=Señal Emg muy baja,(g10 s1-30)=Señal emg muy baja,(g11 s9,10,11,23,27,28,29,30)=Señal Emg muy baja	No se requiere corrección
9	Francis Ferri	gForce_3	LorenaRipalda	user_009	(g5 s1-30)=Señal Emg muy baja,(g6 s19)=Señal Emg muy baja,(g7 s23,24,25)=Señal Emg muy baja,(g10 s1)=Señal Emg muy baja	No se requiere corrección

10	Francis Ferri	gForce_3	Mauricio Ripalda	user_010	(g6 s2,3,4,6,7,8,9,10,12,14,18,21,22,23,24)= Señal Emg muy baja,(g7 s25,29)=Señal Emg muy baja,(g8 s8,10,12,13)= Señal Emg muy baja,(g10 s9,27)=Señal Emg muy baja	No se requiere corrección
11	Francis Soria	gForce_1	Alexander Soria	user_011	Señales Emg centradas en 1.5, (g5 s1,2)=Señal Emg muy baja, (g6 s1-26,28,29,30)= Señal Emg muy baja,(g8 s6,9,11,13,14,22)=Señal Emg muy baja,(g10 s1,10,11,26,)= Señal Emg muy baja,(g10 s10)=Quaternion a medias,(g11 s26)=Señal Emg muy baja	No se requiere corrección
12	Francis Soria	gForce_1	Consuelo Soria	user_012	Señales Emg centradas en 1.5	Restarle el valor medio de la señal al canal 7
13	Francis Soria	gForce_1	Evelin Vinuesa	user_013	Señales Emg centradas en 1.5, (g5 s4,5,8,9,13)=Señal Emg muy baja,(g7 s2,13,15,22,27)= Señal Emg muy baja,(g8 s9,10,11,13,14,15,16,17,18,19,20,22,26,27,28,29,30)=Señal emg muy baja,(g9 s7,,13,25,26,29,30)= Señal Emg muy baja,(g10 s1-26,30)=Señal Emg muy baja	Restarle el valor medio de la señal al canal 7

14	Francis Soria	gForce_1	EvelynAgila	user_014	Señales Emg centradas en 1.5,(g5 s1-30)=Señal Emg muy baja,(g6 s3-30)=Señal emg muy baja,(g7 s17)=Señal Emg muy baja,(g8 s14,18,24,25,28)=Señal emg muy baja,(g9 s1,3,14,15,17,18,20,21,22,23,24,25,26,27,28,29,30)=Señal emg muy baja,(g10 s1-28,30)=Señal Emg muy baja,,(g11 s4,5,26,27)=Señal Emg muy baja	Restarle el valor medio de la señal al canal 7
15	Francis Soria	gForce_1	FrancisSoria	user_015	Señales Emg centradas en 1.5,(g6 s2,9,15,16)=Señal Emg muy baja, (g9 s29,30)=Señal emg muy baja,(g10 s23)=Señal Emg muy baja	Restarle el valor medio de la señal al canal 7
16	Francis Soria	gForce_1	GabrielProaño	user_016	Señales Emg centradas en 1.5,(g2 s3,8)= Señal Emg a medias,(g9 s21,22)=Señal Emg muy baja,(g11 s8,10)=Señal Emg muy baja	Restarle el valor medio de la señal al canal 7
17	Francis Soria	gForce_1	JorgeVillafuerte	user_017	Señales Emg centradas en 1.5, (g6,s25) = Señal Emg muy baja, (g10 s3)= Señal de Quaternion extraño,(g11 s2,16,19,20)=Señal de Quaternion extraño	Gesto 10 sample 3 reemplazar por una correcta, gesto 11 todos los samples reemplazar por correctas, restarle el valor medio de la señal al canal 7

18	Francis Soria	gForce_1	MarceloSoria	user_018	Señales Emg centradas en 1.5, (g1,8)=Demasiado ruido,(g3 s1-30)=Demasiado ruido,(g4 s10)=Demasiado ruido,(g5 s30)=Demasiado ruido,(g6 s10,13,16)=Señal Emg muy baja, (g7,s1-30)=Demasiado ruido,(g8, s1-30)=Demasiado ruido,(g9 s1-30)=Demasiado ruido,(g10 s1-30)=Demasiado ruido	Restarle el valor medio de la señal al canal 7, señales con ruidos extraños
19	Francis Soria	gForce_1	PaulProanho	user_019	Señales Emg centradas en 1.5,(g1 s1,2)=Señal Emg muy baja,(g5 s1-17,s19-30)=Señal Emg muy baja,(g6 s1-15,17-30)=Señal Emg muy baja,(g7 s27)=Quaternion a medias,(g8,s3,9,10,11,13,14,15,17,18)=Señal emg muy baja,(g9 s9,11)=Señal emg muy baja,(g10 s1-17,19,20,21,23-30)=Señal Emg muy baja,(g11 s1,12,20)=Señal Emg muy baja	Restarle el valor medio de la señal al canal 7

20	Francis Soria	gForce_1	SalomeSoria	user_020	Señales Emg centradas en 1.5,(g8 s1,2,3,5,6,7,10,13,14,17,18,20,21,22,23,24,25,26,27)= Señal Emg muy baja,(g10 s3,4,5,7,8,11,15,29)=Señal Emg muy baja,(g12 s1-5)=demasiado ruido	Restarle el valor medio de la señal al canal 7, reemplazar una muestra con una buena de otro usuario en el gesto SYNC
21	Francis Soria	gForce_1	AdrianAleman	user_021	Señales Emg centradas en 1.5,(g6 s14,15,16,21,22,23,24,25,26,29)=Señal Emg muy baja,(g9 s2,4,6,7,16,23,29) =Señal Emg muy baja,(g10 s3,19) =Señal Emg muy baja, (g12 s1-5)=Señales canal 7 movidas	Restarle el valor medio de la señal al canal 7
22	Francis Soria	gForce_1	AlexAraujo	user_022	Señales Emg centradas en 1.5, (g6 s1=Señal Emg muy baja,(g9 s25)=Quaternion a medias,(g11 s17)=Quaternion a medias	Restarle el valor medio de la señal al canal 7
23	Francis Soria	gForce_1	ArielZurita	user_023	(g3 s2) Sin gesto, (g3 s12,18) Ruido,	Reemplazar(g3 s2,12,18)
24	Francis Soria	gForce_1	CamilaCepeda	user_024	(g11) Canal 2 con ruido	Sin corrección
25	Francis Soria	gForce_1	CarlosSoria	user_025		
26	Francis Soria	gForce_1	DerekAguilar	user_026		

27	Francis Soria	gForce_1	IanAguilar	user_027	(g4 s23,24,26,27) Sin gesto,	Reemplazar
28	Francis Soria	gForce_1	JennyVillarroel	user_028	(g1 s1,2,3,6,) Sin gesto,(g2 s3)Ruido,(g3 s7,25)Ruido,(g3 s24) Sin gesto, (g5 s10) Gesto incompleto, (g7) Ruido en EMG	Reemplazar sin gesto y Ruido g7 Válido
29	Francis Soria	gForce_1	ManuelSoriana	user_029	(g3 s6,7,17,18) Gesto incompleto, (g4 s15,16,17,20) Gesto incompleto, (g5 s28) Sin gesto, (g9 s16) Gesto incompleto	Reemplazar sin gesto
30	Francis Soria	gForce_1	XavierSoria	user_030	(g6 s1,30) gesto incompleto, (g8 s15,17,22,23) gesto incompleto, (g9 s21) gesto incompleto	Sin corrección
31	Juan Pablo Vazconez	gForce_1	CeciliaMeleendez	user_031	(g2 s6) Sin gesto, (g6 s27,28) Sin gesto, (g10 s19) Sin gesto	reemplazar
32	Juan Pablo Vazconez	gForce_1	ClaudiaHurtado	user_032	(g4 s15,29) gesto incompleto	Sin corrección
33	Juan Pablo Vazconez	gForce_1	DianaParra	user_033	(g7 s28) Sin gesto	reemplazar
34	Juan Pablo Vazconez	gForce_1	GabyQuisiguiña	user_034	(g6 s23) Gesto incompleto	Sin corrección
35	Juan Pablo Vazconez	gForce_1	IsabelChango	user_035	(g3 s15) Gesto incompleto, (g9 s22) Sin gesto	Reemplazar sin gesto
36	Juan Pablo Vazconez	gForce_1	JuanPabloVasconez	user_036		

37	Juan Pablo Vazconez	gForce_1	KarenCarra sco	user_037	(g1) Sin gesto - EMG, (g6 s6) Gesto incompleto	usuario problemático
38	Juan Pablo Vazconez	gForce_1	KerllyCeden o	user_038		
39	Juan Pablo Vazconez	gForce_1	LuisVascon ez	user_039	(g6 s1) Sin gesto	Reemplazar
40	Juan Pablo Vazconez	gForce_1	MichelleVas conez	user_040		
41	Leonardo Valdivieso	gForce_1	GustavoVal divieso	user_041		
42	Leonardo Valdivieso	gForce_1	LeonardoVa ldivieso	user_042		
43	Lorena Barona	gForce_1	GisselaVela stegui	user_043	(g11 s1) Gesto incompleto	Sin corrección
44	Lorena Barona	gForce_1	LorenaBaro na	user_044		
45	Lorena Barona	gForce_1	PatricioLope z	user_045		No se requiere corrección
46	Rashid	Myo-EPN-Marco Benalcázar	AndreaAguir re	user_046		No se requiere corrección

47	Rashid	Myo-EPN- Marco Benalcázar	KATHERIN EGALARZA	user_047		No se requiere corrección
48	Rashid	Myo-EPN- Marco Benalcázar	RASHIDAN DAGAMA	user_048		No se requiere corrección
49	Rashid	Myo-EPN- Marco Benalcázar	SAMUELCE DENhO	user_049		No se requiere corrección
50	Ricardo Romero	Myo-EPN- Marco Benalcázar	CristinaBay as	user_050		No se requiere corrección
51	Ricardo Romero	Myo-EPN- Marco Benalcázar	GabrielSant os	user_051		No se requiere corrección
52	Ricardo Romero	Myo-EPN- Marco Benalcázar	JorgeFlores	user_052		No se requiere corrección
53	Ricardo Romero	Myo-EPN- Marco Benalcázar	LuisMedina	user_053		No se requiere corrección

54	Ricardo Romero	Myo-EPN-Marco Benalcázar	LuisRomero	user_054		No se requiere corrección
55	Ricardo Romero	Myo-EPN-Marco Benalcázar	RicardoRomero	user_055		No se requiere corrección
56	Ricardo Romero	Myo-EPN-Marco Benalcázar	SoniaParedes	user_056	G11 s27 gesto incompleto quaternion	No se requiere corrección
57	Victor Vimos	gForce_1	alexona	user_057		No se requiere corrección
58	Victor Vimos	gForce_1	andresgarces	user_058	G10 lote EMG sin información	No se requiere corrección
59	Victor Vimos	gForce_2	andresmayorga	user_059	G6 lote - s15, G9 s20-30, G10 s1-10 EMGs sin info	No se requiere corrección
60	Victor Vimos	gForce_1	andresplacencia	user_060	G6 lote , G9 s2,10, G10 lote completo EMGs sin info	No se requiere corrección
61	Victor Vimos	gForce_2	danielmoreno	user_061	G6 s1,2,3,10 EMGs sin info	No se requiere corrección

62	Victor Vimos	gForce_2	gabrielamoren	user_062	G6 s1-27, G10 s1-11 23-25, G11 s21, G12 EMGS sin info	No se requiere corrección
63	Victor Vimos	gForce_1	hernanvimos	user_063	G6 s11,15,16,17,19,27, G10 s23,29,30, G12 EMGs sin info	No se requiere corrección
64	Victor Vimos	gForce_1	hugovimos	user_064	G6 s10,12,13,19,21, G9 s3,11,12, G11 s16	No se requiere corrección
65	Victor Vimos	gForce_1	jorgesuarez	user_065	G6 s12,13,14,15,19,20,21,29,30, G9 s2,3,4,5,6,7,11,12,13,14,17,18,20,22,23,24,26,28,29, G10 2,30 EMGs sin info	No se requiere corrección
66	Victor Vimos	gForce_1	leonardomoren	user_066	G6 lote, G9 s4,14,16, G10 lote excepto 21,24,26,27, G12 s3 EMGs sin info	No se requiere corrección
67	Victor Vimos	gForce_2	pabloperez	user_067	Movimientos del brazo con EMGs imperceptibles	No se requiere corrección
68	Victor Vimos	gForce_2	victorvimos	user_068	Movimientos del brazo con EMGs imperceptibles	No se requiere corrección
69	Alex Chico	Myo-EPN-Marco Benalcázar	CristianChico	user_069	Movimientos del brazo con EMGs no claras	No se requiere corrección
70	Alex Chico	Myo-EPN-Marco Benalcázar	MariaMagdalena	user_070	Sin observación	No se requiere corrección

71	Francis Soria	gForce_1	Eleana Ojeda	user_071	EMGS para gestos son débiles. Movimientos del brazo con EMGs imperceptibles	No se requiere corrección
72	Patricio Cruz	Myo-EPN-Marco Benalcázar	Alan Castelo	user_072	- canales activos siempre	No se requiere corrección
73	Patricio Cruz	Myo-EPN-Marco Benalcázar	Amanda Ponce	user_073	Sin observación	No se requiere corrección
74	Patricio Cruz	Myo-EPN-Marco Benalcázar	Ana Pichucho	user_074	11 19; canales 45 ruido	No se requiere corrección
75	Patricio Cruz	Myo-EPN-Marco Benalcázar	Axel Betancourt	user_075	Movimientos del brazo con EMGs imperceptibles	No se requiere corrección
76	Patricio Cruz	Myo-EPN-Marco Benalcázar	Bryan Castelo	user_076	Sin observación	No se requiere corrección
77	Patricio Cruz	Myo-EPN-Marco Benalcázar	Edison Castelo	user_077	6 12; muestras nulas -(3,28,29,30)	

78	Patricio Cruz	Myo-EPN- Marco Benalcázar	Genesis Bermeo	user_078	Sin observación	No se requiere corrección
79	Patricio Cruz	Myo-EPN- Marco Benalcázar	Josue Betancourt	user_079	Sin observación	No se requiere corrección
80	Patricio Cruz	Myo-EPN- Marco Benalcázar	Josue Castelo	user_080	7 12 canales 4578 ruido	No se requiere corrección
81	Patricio Cruz	Myo-EPN- Marco Benalcázar	Leandro Ponce	user_081	1 12 muestra 30 nula	Reemplazo de muestra nula
82	Patricio Cruz	Myo-EPN- Marco Benalcázar	Maria Mosquera	user_082	Sin observación	No se requiere corrección
83	Patricio Cruz	Myo-EPN- Marco Benalcázar	Marizol Lopez	user_083	11 12 ch 7 activo siempre 30 nula	4 12 muestra Reemplazo de muestra nula
84	Patricio Cruz	Myo-EPN- Marco Benalcázar	Patricio Ponce	user_084	Sin observación	No se requiere corrección

85	Patricio Cruz	Myo-EPN- Marco Benalcázar	Yolanda Lopez	user_085	-señales débiles	No se requiere corrección
----	---------------	---------------------------------	------------------	----------	------------------	------------------------------

ANEXO IV. Configuraciones de los optimizadores para las redes CNN-LSTM

Optimizador para el modelo de gestos estáticos

```
options = trainingOptions('adam', ...  
    'InitialLearnRate', 0.001, ...  
    'LearnRateSchedule','piecewise', ...  
    'LearnRateDropFactor',0.2, ...  
    'LearnRateDropPeriod',8, ... %5 8  
    'ExecutionEnvironment','gpu', ... %gpu  
    'GradientThreshold',1, ...  
    'MaxEpochs',maxEpochs, ...  
    'MiniBatchSize',miniBatchSize, ...  
    'Shuffle','never', ...  
    'Verbose',0, ...  
    'ValidationData', validationDatastore, ...  
    'ValidationFrequency',floor(trainingDatastore.NumObservations/ miniBatchSize), ...  
    'ValidationPatience',5, ...  
    'Plots','training-progress');
```

Optimizador para el modelo de gestos dinámicos

```
options = trainingOptions('sgdm', ...  
    'InitialLearnRate', 0.04, ... %0.0002  
    'LearnRateSchedule','piecewise', ...  
    'LearnRateDropFactor',0.2, ...  
    'LearnRateDropPeriod',8, ... %5 8  
    'ExecutionEnvironment','gpu', ... %gpu  
    'GradientThreshold',1, ...  
    'MaxEpochs',maxEpochs, ...  
    'MiniBatchSize',miniBatchSize, ...  
    'Shuffle','never', ...  
    'Verbose',0, ...  
    'ValidationData', validationDatastore, ...  
    'ValidationFrequency',floor(trainingDatastore.NumObservations/ miniBatchSize), ...  
    'ValidationPatience',5, ...  
    'Plots','training-progress');
```

ANEXO V. Tabla de resultados de evaluación clasificador linear logístico

Usuario	Exactitud de clasificación training [%]	Exactitud de clasificación testing [%]
1	98.79	98.79
24	100.00	99.39
25	70.91	70.30
27	96.97	94.55
32	98.79	100.00
33	67.88	67.27
35	98.79	99.39
37	94.55	96.97
39	86.67	84.85
40	98.79	99.39
41	81.21	77.58
42	90.91	89.09
26	96.97	98.18
34	77.58	76.36
3	84.85	81.82
5	93.94	94.55
7	92.73	92.12
10	97.58	98.79
13	100.00	100.00
15	100.00	99.39
17	100.00	100.00
23	92.12	95.15
30	96.36	96.36
28	100.00	100.00
4	94.55	92.12
6	100.00	99.39
8	100.00	100.00
9	96.97	98.79
11	87.88	87.88
12	96.97	96.36

14	90.91	90.30
16	95.76	96.97
18	84.85	84.85
19	83.03	81.82
20	97.58	97.58
21	92.12	92.12
22	100.00	99.39
29	100.00	100.00
31	100.00	100.00
Total	93.26	93.02