

# Proyecto Guante traductor de lenguaje de signos



Francisco Javier Del Río Ruiz  
Raúl Morgado Saravia

[https://github.com/FJavierDL/Proyecto\\_Guante.git](https://github.com/FJavierDL/Proyecto_Guante.git)

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Estado del arte</b>	<b>2</b>
<b>3. Objetivos y novedades</b>	<b>2</b>
<b>4. Materiales y métodos empleados</b>	<b>3</b>
<b>5. Requisitos</b>	<b>3</b>
5.1. Requisitos funcionales	3
5.2. Requisitos técnicos	3
<b>6. Desarrollo</b>	<b>4</b>
6.1. Hito 0 - Planteamiento del proyecto	4
6.2. Hito 1 - Entendimiento de los Cuaterniones	4
6.3. Hito 2 - Configuración y lectura de 1 sensor	4
6.4. Hito 3 - Red neuronal con 1 sensor	5
6.5. Hito 4 - Configuración y lectura de 2 sensores	6
6.6. Hito 5 - Red neuronal 2 sensores BNO-055	7
6.7. Hito 6 - Interfaz gráfica	9
<b>7. Planificación temporal</b>	<b>10</b>
<b>8. Presupuesto</b>	<b>11</b>
<b>9. Conclusión</b>	<b>11</b>
<b>10. Bibliografía</b>	<b>12</b>

# 1. Introducción

Con la ayuda del sensor BNO-055 vamos a aprovechar algunos de sus sensores para realizar un sistema el cual sea capaz de representar los movimientos de una mano y traducirlo a lenguaje natural, facilitando la traducción a personas con ciertas discapacidades.

## 2. Estado del arte

En la actualidad es muy necesario para todas las personas poder estar comunicados y poder expresarnos en un entorno en el que todos tengamos las mismas posibilidades, para ello nuestro proyecto se basa en la necesidad que tienen algunas personas de poder expresarse y comunicarse con gente que no entiende el lenguaje de signos.

Antes de empezar nuestro proyecto, hemos investigado y es un negocio que se ha visto muy desarrollado en la actualidad gracias al auge de la inteligencia artificial, sobre todo hemos encontrado estos dos proyectos: el primero es un modelo que se basa en la IA: [BrightSign: el guante inteligente que da voz a quienes no pueden hablar \(wipo.int\)](http://wipo.int) , aunque nuestro proyecto tiene más similitudes con el siguiente proyecto, ya que tiene una solución basada en otro tipo de tecnología, usando sensores y representando el movimiento de la mano mediante estos: [Este guante permite traducir el lenguaje de signos en tiempo real \(xataka.com\)](http://xataka.com)

## 3. Objetivos y novedades

Con este proyecto pretendemos medir los movimientos realizados con la mano para poder representar el lenguaje de signos... Esta medición se realizará haciendo lecturas del sensor BNO-055, el cual posee varios sensores como un giroscopio, un acelerómetro, etc, con el cual clasificaremos los movimientos según sus valores. Estos valores nos los da a través de un vector de 4 elementos, formado por un vector tridimensional ( $v$ ) y un ángulo de rotación  $\theta$  (escalar).

Las últimas novedades que hemos incluido ha sido la incorporación de un segundo sensor, con este segundo sensor para mejorar la predicción de los gestos y poder aumentar la cantidad de movimientos que podemos identificar. Además hemos añadido una aplicación para poder visualizar mejor los resultados.

## 4. Materiales y métodos empleados

Los **materiales** que hemos usado son: Sensor BNO-055, protoboard, cables y NUCLEO-F767ZI.

En cuanto a los **métodos** hemos usado comunicación a través de la UART, con la cual hemos comunicado nuestra NUCLEO-f767zi con el pc, y el bus I2C para comunicar los sensores con la NUCLEO-f767ZI.

Con respecto a la **IA**, hemos usado la librería de tensorflow, con la cual hemos creado una red neuronal y hemos recreado los movimientos para poder entrenarla y, así, sea capaz de reconocer los signos.

## 5. Requisitos

### 5.1. Requisitos funcionales

- El sistema debe ser capaz de capturar los movimientos de la mano utilizando el sensor BNO-055.
- Se requiere la capacidad de clasificar los movimientos de la mano en categorías específicas basadas en las lecturas de los sensores del sensor BNO-055.
- Se debe de implementar una solución de inteligencia artificial que distinga entre los movimientos de la mano de manera eficiente y precisa.
- El sistema debe integrarse en una interfaz gráfica intuitiva con la que visualizar los resultados.

### 5.2. Requisitos técnicos

- El sistema debe integrar el sensor BNO-055 y la NUCLEO STM32F767ZI para la captura y procesamiento de datos de movimiento de la mano.
- El sistema de comunicación entre la placa y los sensores será I2C.
- El sistema de comunicación entre la placa y el ordenador de desarrollo será puerto serie por UART.
- Se deben implementar algoritmos de procesamiento de señales para interpretar y capturar los datos del sensor.
- La inteligencia artificial será programada en Python con la librería de Tensorflow.
- La interfaz gráfica será programada en Python con la librería de Tkinter.

## 6. Desarrollo

### 6.1. Hito 0 - Planteamiento del proyecto

Búsqueda y planteamiento de la idea principal del proyecto, balance en las tecnologías que usaremos en el proyecto y planificación temporal inicial.

### 6.2. Hito 1 - Entendimiento de los Cuaterniones

El sensor BNO-055 tiene dos modos de representar los datos obtenidos, una a través de los ángulos de euler y otra a través de cuaterniones. Nosotros trabajaremos con cuaterniones, de los cuales hemos tenido que investigar y comprender para poder usarlos y representarlos. Este proceso nos tomó las primeras semanas de proyecto, ya que es un término complejo.

### 6.3. Hito 2 - Configuración y lectura de 1 sensor

Para poder trabajar con este sensor, hemos optado por usar una librería [GitHub - ivyknob/bno055\\_stm32: BNO055 library to use with STM32 HAL](https://github.com/ivyknob/bno055_stm32) la cual hemos tenido que entender y aprender a usar ya que no contiene información de uso.

Lo primero que hemos hecho, ha sido configurar un bus I2C de la placa de desarrollo y pasarlo a la librería para que lea y obtenga los datos del sensor.

Después, los hemos transmitido a través una UART de la placa para recogerlos a través del programa **Putty** que leyera el puerto serie COM de la placa y guardarlos en un fichero .txt con cada tipo de gesto.

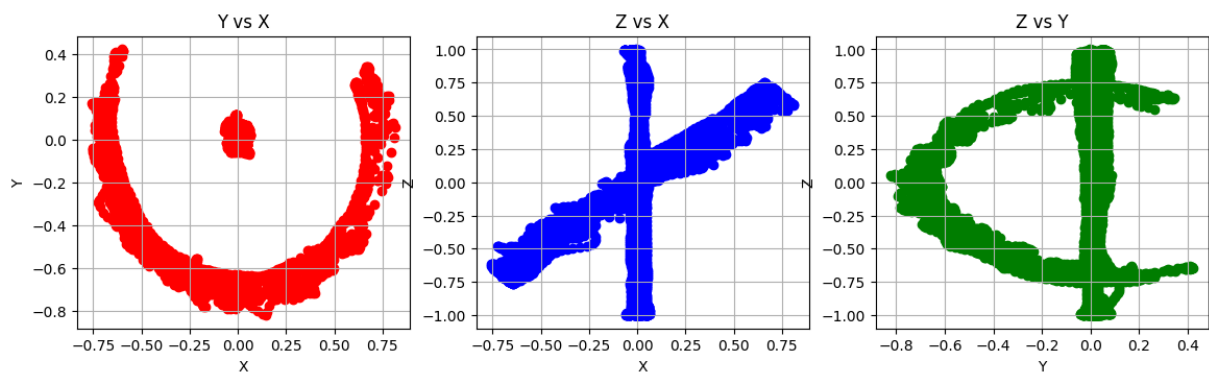
Hemos generado un fichero para cada tipo de gesto que recogía los datos del sensor en un ángulo de 360° porque cada gesto tiene que ser el mismo en todos los ángulos de movimiento de una persona. Con cada fichero hemos recogido 2000 datos de cada gesto. Cada línea del fichero corresponde a [dato, tipo] → [w, x, y, z, tipo].

## 6.4. Hito 3 - Red neuronal con 1 sensor

Para el procesado de datos, hemos seguido estos pasos:

- Paso 1 → Lectura de ficheros de gestos en conjunto de datos.
- Paso 2 → Ordenamos aleatoriamente el conjunto de datos.
- Paso 3 → Dividimos el conjunto en 80% de entrenamiento y 20% de test.
- Paso 4 → Subdividimos en datos y etiquetas de entrenamiento.
- Paso 5 → Subdividimos en datos y etiquetas de test.

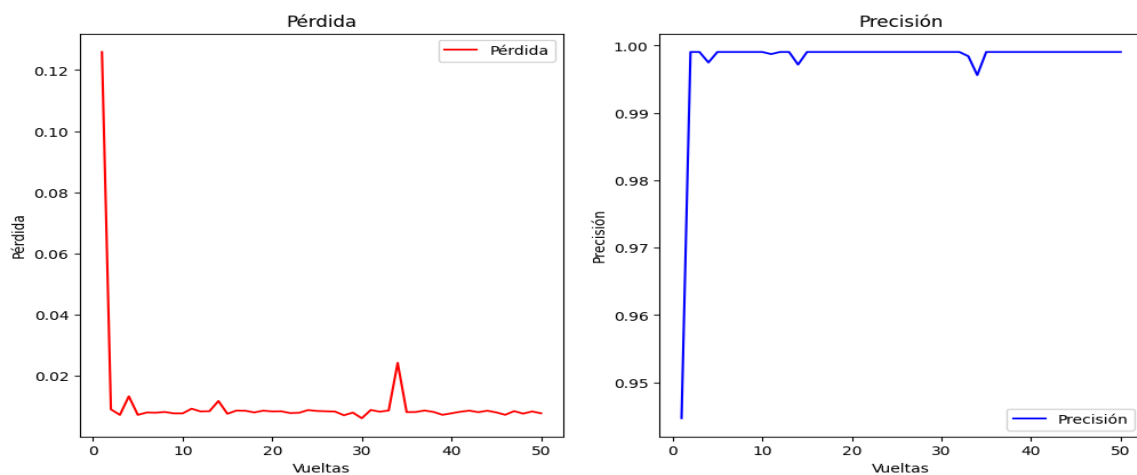
La siguiente imagen es una representación de los 2 gestos en 2 dimensiones con la cual podemos ver si hemos barrido bien todo el rango de movimiento de los gestos. Como vemos, no se termina de cerrar el círculo como debería, ya que hemos recogido datos en 360°.



Para la creación de una red neuronal utilizaremos la librería de **Tensorflow** y como lenguaje de programación **Python**.

Nuestra red neuronal, es una red que tiene 4 datos de entrada y clasifica entre 2 gestos, por tanto, tendrá 2 neuronas en la capa de salida para distinguir con un 1 o un 0 entre los gestos.

Tras varias pruebas, hemos llegado a la conclusión que poner una capa intermedia con 2 neuronas y una función de activación de tipo **sigmoide** es lo suficientemente efectiva para clasificar y distinguir con un 99,906 % de **exactitud** los 2 gestos. El entrenamiento ha sido realizado en 50 **epochs** con la función de pérdida de **Categorical Cross Entropy** y el optimizador de entrenamiento ha sido **Adam** con una tasa de aprendizaje de 0.1.



Este resultado podemos verlo desde antes de empezar a entrenar a la red, ya que en la imagen anterior se puede distinguir claramente a la vista cuales son los gestos, por tanto, la red neuronal con tan pocas neuronas los logra clasificar casi perfectamente. Los resultados son muy buenos, pero no recogen la totalidad del rango de movimiento.

## 6.5. Hito 4 - Configuración y lectura de 2 sensores

En la última semana ha llegado un **segundo sensor** BNO-055. La incorporación de este segundo sensor nos ha hecho modificar el proyecto, ampliando un segundo puerto I2C. También se podría haber realizado con un único I2C, pero nosotros usamos una librería que no nos permitía el uso de dos sensores en un mismo puerto.

El usar varios sensores de este tipo puede llegar a ser un **problema** porque el sensor BNO-055 solo tiene 2 direcciones I2C (0x28 y 0x29), dependiendo de la placa de desarrollo utilizada se podrán usar más o menos por placa. En nuestro caso, por cómo está hecha la librería que hemos usado, la dirección de I2C es fija a 0x28, por tanto, solo podemos usar un sensor por cada canal de I2C. Nuestra placa cuenta con 4 canales I2C, así que, como mucho podríamos usar 4 sensores BNO-055 a la vez.

Tras solventar este problema, hemos vuelto a recoger los datos, de ahora 2 sensores, para entrenar a la nueva red neuronal.

Como antes, hemos generado un fichero para cada tipo de gesto que recogía los datos del sensor en un ángulo de 360° porque cada gesto tiene que ser el mismo en todos los ángulos de movimiento de una persona. Con cada fichero hemos recogido 2000 datos de cada gesto.

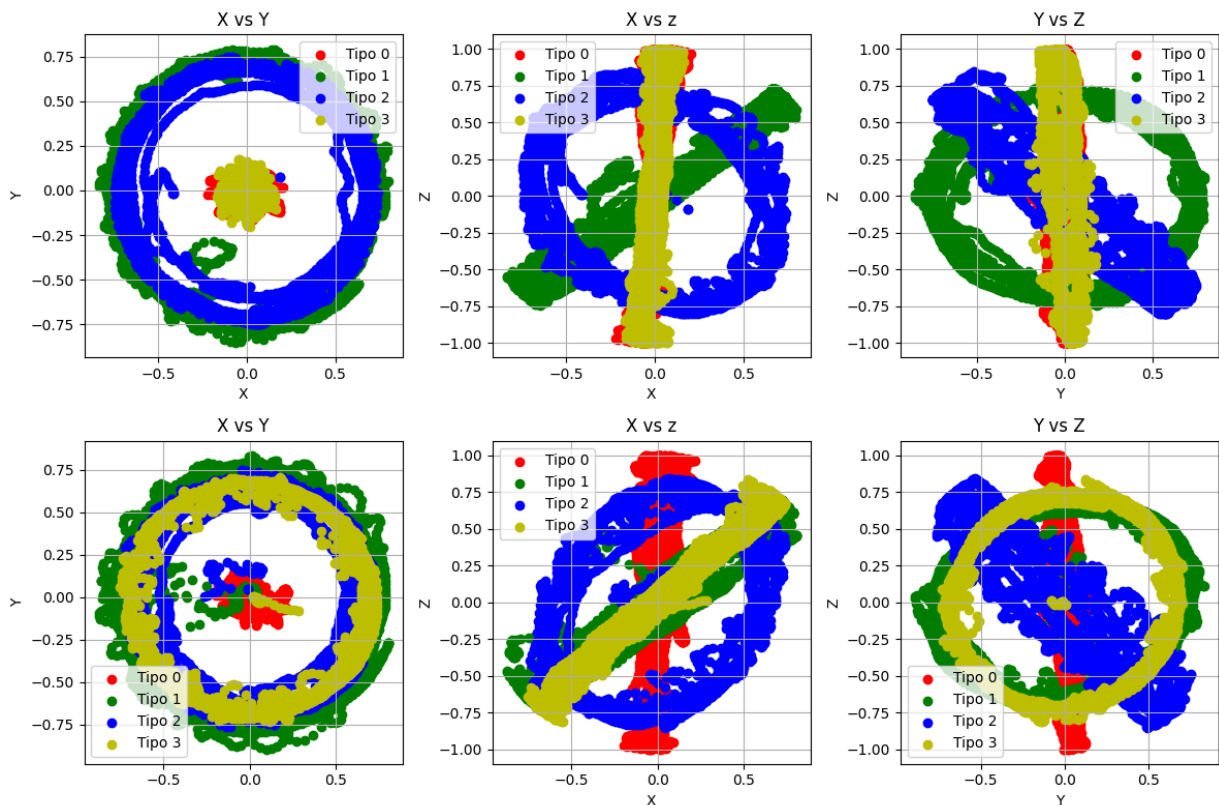
Como vamos a reconocer 4 gestos, habrá 4 ficheros con 2000 datos cada uno haciendo en total 8000 datos. Cada línea del fichero corresponde a [sensor\_1, sensor\_2, tipo] → [w1, x1, y1, z1, w2, x2, y2, z2, tipo].

## 6.6. Hito 5 - Red neuronal 2 sensores BNO-055

El procesado de datos se ha hecho como antes modificando lo necesario para la entrada mayor de variables.

La siguiente imagen es una representación de los **4 gestos en 2 dimensiones** con la cual podemos ver si hemos barrido bien todo el rango de movimiento de los gestos. Como vemos, ahora sí se termina de cerrar el círculo como debería en los 360°.

La primera fila de gráficas corresponden al sensor 1 y la segunda al sensor 2.



Nuestra red neuronal tiene que cambiar ya que ahora tiene que reconocer 4 gestos distintos y el número de entradas ahora es mayor, ahora la red que tiene 8 datos de entrada y 4 neuronas en la capa de salida para distinguir con un 1 o un 0 entre los gestos.

Tras varias pruebas, hemos llegado a la conclusión que poner una capa intermedia con 16 neuronas y una función de activación de tipo **sigmoide** es lo suficientemente efectiva para clasificar y distinguir con un 100 % de **exactitud** los 4 gestos. El entrenamiento ha sido realizado en 100 **epochs** con la función de pérdida de **Categorical Cross Entropy** y el optimizador de entrenamiento ha sido **Adam** con una tasa de aprendizaje de 0.005.



Esta vez hemos representado las métricas de Recall, Precision, AUC y Accuracy:

- **Recall** → Ratio de verdaderos positivos.
- **Precision** → Cuantos positivos son realmente positivos
- **Accuracy** → Porcentaje de valores correctamente clasificados, tanto positivos como negativos.
- **AUC** → Es el área bajo la curva ROC (Representa el Recall contra el ratio de falsos positivos). Es un valor entre 0 y 1. Cuanto más cerca de 1, más óptimo es el resultado e indica que el modelo generaliza muy bien.

Izquierda gráfica de la Pérdida → Mejor cuanto más cerca de 0.

Derecha gráfica Recall, Precision, AUC y Accuracy → Mejor cuanto más cerca de 1.

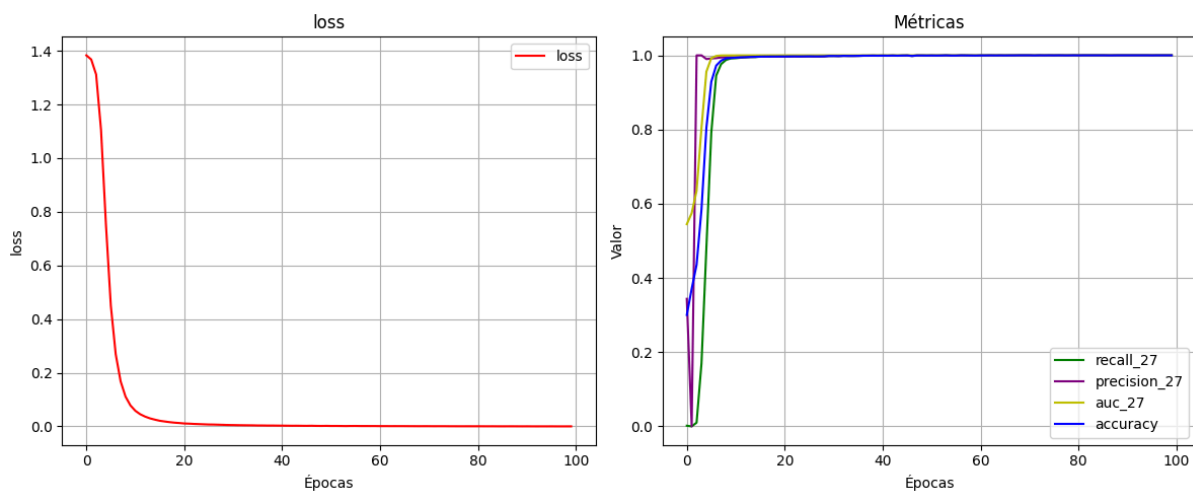
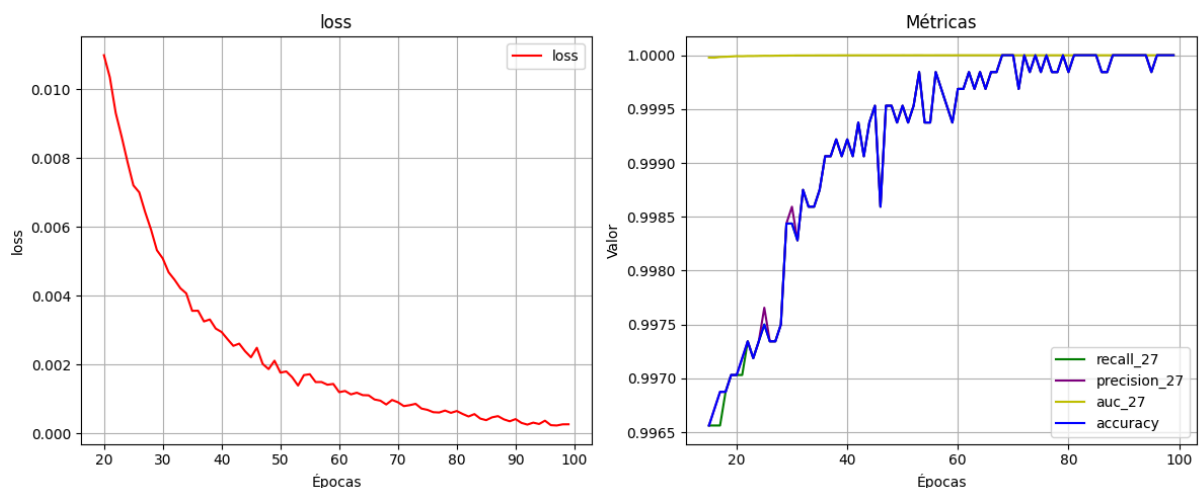


Gráfico a partir de la época 15 para ver ampliado como disminuye la pérdida y aumenta la exactitud del modelo.



Resultado de la última vuelta de entrenamiento.

```
Loss: 0.00017126362945418805
Recall: 1.0
Precision: 1.0
AUC: 1.0
Accuracy: 1.0
```

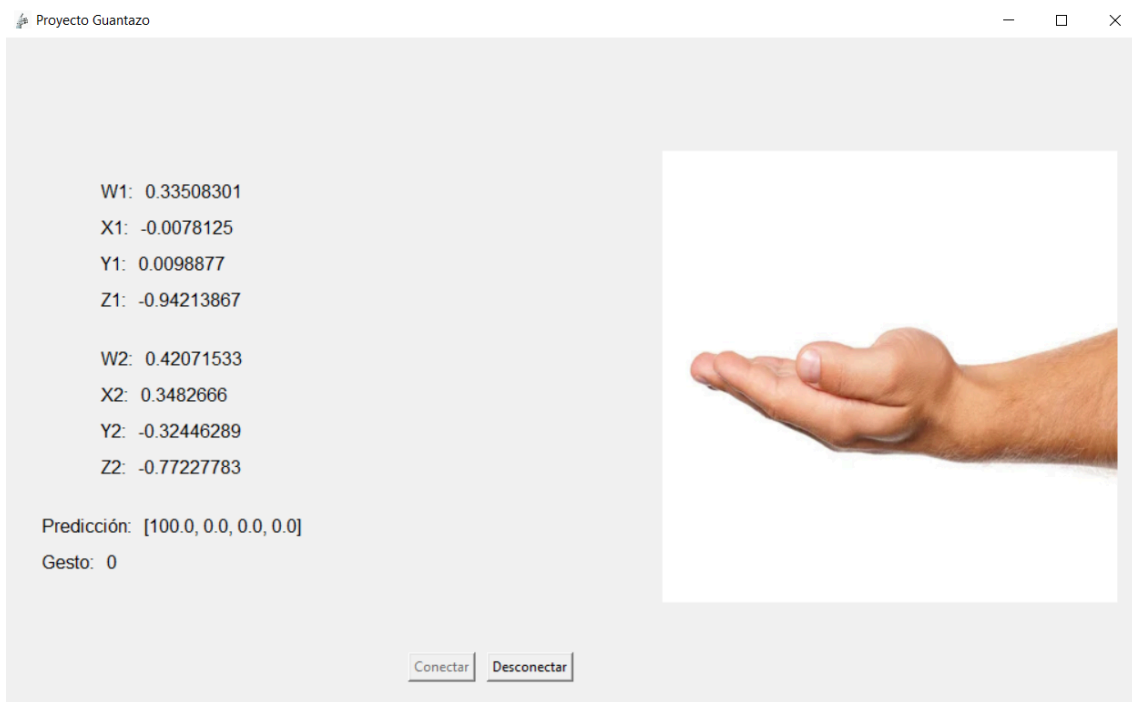
Métricas tras la predicción de todos los datos del conjunto de test.

```
Predicciones Correctas: 1598
Predicciones Incorrectas: 2
Porcentaje de Aciertos: 99.88%
Porcentaje de Errores: 0.12%
```

```
Accuracy: 0.99875
Precision: 1.0
Recall: 0.99915963
AUC: 0.6697479
```

## 6.7. Hito 6 - Interfaz gráfica

Para un uso más cómodo de nuestro proyecto, hemos creado una pequeña **interfaz** que se conecta al **puerto serie** de la placa de desarrollo y en la que se muestra el valor de los **datos obtenidos** (valor de los cuaterniones) junto con los valores de la predicción y el tipo de gesto correspondiente, además contiene una imagen que representa el símbolo que se debería estar realizando.



## 7. Planificación temporal

Semana	Fecha	Hito	Descripción
1	27/02/2024	0	Búsqueda de la idea de proyecto.
2	05/03/2024	1	Presentación de la idea de proyecto.
3	12/03/2024	1	Aprendizaje de los cuaterniones.
4	19/03/2024	2	Búsqueda de librerías para STM32.
5	26/03/2024	2	Configuración STM32 y pruebas de lectura con librerías.
6	02/04/2024	2	Elección de librería definitiva.
7	09/04/2024	2	Preparación de recogida de datos y gestos a reconocer.
8	16/04/2024	3	Recolección de datos de los gestos.
9	23/04/2024	3	Procesado de datos de entrenamiento y testeo.
10	30/04/2024	3	Diseño y entrenamiento de red neuronal.
11	07/05/2024	3	Métricas de entrenamiento y pruebas de rendimiento de la red neuronal.
12	14/05/2024	4, 5, 6	<b>Ampliación</b> → Llegada segundo sensor. Configuración de STM32 y recogida de datos. Ampliación de red neuronal y entrenamiento. Métricas y pruebas de la red neuronal.

## 8. Presupuesto

Material	Cantidad	Coste Unitario (€)	Total (€)
NUCLEO-F767ZI	1	23	23
Sensor BNO-055	2	33,50	67,00
Total proyecto			90,00

## 9. Conclusión

Finalmente no hemos podido terminar el proyecto como nos hubiese gustado por falta de material y tiempo. Sobre todo la parte que nos hubiese gustado mejorar es la presentación del proyecto, nos hubiese encantado poder implementarlo como un guante, pero no teníamos cables preparados para ello. También una posible mejora es la conexión, ya que es un poco incómodo el uso con el cable y haber implementado más gestos.

En general, este proyecto nos ha aportado mucho ya que nos ha enseñado las magnitudes que tienen los proyectos basados en microcontroladores y el tiempo de preparación que conllevan. Nos ha hecho investigar mucho y hemos aprendido mucho sobre microcontroladores y en específico sobre el NUCLEO-F767ZI y el sensor BNO-055 ya que es en lo que hemos basado nuestro proyecto. También hemos llegado a interesarnos e informarnos en la IA ya que también la hemos usado en la parte de procesamiento de datos.

## 10. Bibliografía

[Body Posture Determination for Heart Failure Patients From Ankle Orientation Measurements | IEEE Journals & Magazine | IEEE Xplore](#)

[Quaternions and the spatial rotations in motion enabled wearable devices. Exploiting the potential of smart IMUs attitude estimation. - Pablo Perez Garcia \(embeddedrelated.com\)](#)

[Sensor inercial Absoluto 9 DOF BNO055 Adafruit 4646 | BricoGeek.com](#)

[Este guante permite traducir el lenguaje de signos en tiempo real \(xataka.com\)](#)

[BrightSign: el guante inteligente que da voz a quienes no pueden hablar \(wipo.int\)](#)

[GitHub - ivyknob/bno055\\_stm32: BNO055 library to use with STM32 HAL](#)

[Pinouts | Adafruit BNO055 Absolute Orientation Sensor | Adafruit Learning System](#)

[nucleo\\_f767zi\\_zio\\_right\\_2016\\_10\\_24.png \(960×720\) \(mbed.com\)](#)

[nucleo\\_f767zi\\_zio\\_left\\_2020\\_3\\_30.png \(960×720\) \(mbed.com\)](#)