

## ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



### TRABAJO FIN DE GRADO

## DETECCIÓN AUTOMÁTICA DEL LENGUAJE DE SIGNOS MEDIANTE SERIES TEMPORALES

**Titulación:** Grado en Ingeniería en Tecnologías de la  
Telecomunicación

**Mención:** Telemática

**Autor:** Eugenio J. Fleitas Méndez

**Tutor:** Carlos M. Travieso González

**Fecha:** Julio 2024



## Resumen

En este mundo existen un elevado número de barreras que separan a la humanidad, y una de las más perjudiciales es la barrera lingüística que dificultan la comunicación. Las diferencias en el idioma pueden dificultar la comunicación y el entendimiento entre personas de diferentes regiones y culturas. Esta barrera, no solo afecta en las interacciones cotidianas, sino que también limita el posible acceso a oportunidades educativas y laborales, y restringe el intercambio de conocimientos y experiencias. Además, pueden crear malentendidos y prejuicios, pudiendo originar estereotipos, divisiones e incluso guerras.

Una parte importante de esta barrera es la que enfrentan las personas sordas, quienes dependen del lenguaje de signos para comunicarse. La falta de entendimiento y comprensión del lenguaje de signos, por parte de la mayoría de la sociedad, provoca un aislamiento hacia estas personas y dificultan su participación y unión a la sociedad. Aún con los esfuerzos en la traducción y la enseñanza del lenguaje de señas, la barrera lingüística sigue siendo un desafío considerable.

El objetivo de la tesis doctoral fue el dar solución a este problema, creando un sistema capaz de “romper” este muro. Por una parte, han conseguido crear ese sistema, permitiendo detectar y traducir un limitado número de palabras del lenguaje de signos español; Sin embargo, este sistema, aun teniendo grandes avances, presenta varios puntos que pueden mejorarse, para que este sistema sea más preciso, más rápido, más intuitivo, más ameno al uso, etc. Por este motivo, se ha creado este proyecto. Este proyecto tiene como tarea el analizar los diferentes bloques que conforman el proyecto base, en buscar de posibles puntos de mejora. Esto, con el objetivo de mejorar y optimizar dichos bloques, obteniendo así un sistema más preciso y rápido, y por consecuencia, más ameno al uso.



## *Abstract*

In this world, numerous barriers separate humanity, and one of the most damaging is the language barrier that hinders communication. Differences in language can obstruct understanding and interaction between people from diverse regions and cultures. This barrier not only affects daily interactions but also limits access to educational and employment opportunities and restricts the exchange of knowledge and experience. It can also lead to misunderstandings and prejudices, fostering stereotypes, divisions, and even conflicts.

A significant aspect of this barrier is faced by deaf individuals who rely on sign language to communicate. The lack of understanding and knowledge of sign language by the majority of society leads to isolation and makes it difficult for deaf people to participate fully in society. Even with efforts in sign language translation and education, the language barrier remains a substantial challenge.

The aim of the doctoral thesis was to address this issue by creating a system capable of "breaking" this wall. This system was developed to detect and translate a limited number of words in Spanish sign language. Despite significant advances, the system still has several areas for improvement to become more accurate, faster, intuitive, and user-friendly. Therefore, this project was initiated. The goal of this project is to analyze the different components of the basic system to identify potential areas for enhancement. By optimizing these components, we aim to develop a more precise, faster, and ultimately more user-friendly system.



# ÍNDICE GENERAL

## PARTE I – MEMORIA

|  |           |
|--|-----------|
| <b>1. Introducción.....</b>  | <b>1</b>  |
| <b>1.1 Introducción.....</b>   | <b>1</b>  |
| <b>1.2 Motivaciones.....</b>   | <b>1</b>  |
| <b>1.3 Avances tecnológicos en el reconocimiento de la lengua de signos.....</b> | <b>3</b>  |
| <b>1.4 Proyecto base del TFG.....</b>  | <b>6</b>  |
| <b>1.5 Hipótesis y Objetivos.....</b>  | <b>7</b>  |
| <b>1.6 Peticionario.....</b>   | <b>8</b>  |
| <b>1.7 Estructura de la memoria.....</b>   | <b>8</b>  |
| <b>2. Estado del arte.....</b>   | <b>9</b>  |
| <b>2.1. Introducción.....</b>  | <b>9</b>  |
| <b>2.2. Evolución de los Dispositivos de Reconocimiento de Gestos.....</b>       | <b>10</b> |
| <b>2.2.1 Guantes de Datos.....</b>   | <b>10</b> |
| <b>2.2.2. Electrodo EMG (Electromiografía).....</b>                              | <b>12</b> |
| <b>2.2.3. Ultrasonidos.....</b>  | <b>13</b> |
| <b>2.2.4. WiFi.....</b>  | <b>14</b> |
| <b>2.2.5. Cámara RGB y RGB-d.....</b>  | <b>14</b> |
| <b>2.2.5.1. Visión Estereoscópica.....</b>                                       | <b>14</b> |
| <b>2.2.5.2. Luz Estructurada.....</b>  | <b>15</b> |
| <b>2.2.5.3. Cámara ToF o de tiempo de vuelo.....</b>                             | <b>15</b> |
| <b>2.2.6. Leap Motion.....</b>   | <b>15</b> |
| <b>2.2.7. VR o Realidad Virtual.....</b>   | <b>17</b> |
| <b>2.2.8. UltraLeap.....</b>   | <b>18</b> |
| <b>2.2.9. Kinect y Azure Kinect.....</b>   | <b>18</b> |
| <b>2.2.9.1. Kinect.....</b>  | <b>18</b> |
| <b>2.2.9.2. Azure Kinect.....</b>  | <b>19</b> |
| <b>2.3. Conclusiones.....</b>  | <b>19</b> |
| <b>3. Materiales y métodos.....</b>  | <b>35</b> |
| <b>3.1. Introducción.....</b>  | <b>35</b> |
| <b>3.2. Adquisición.....</b>   | <b>35</b> |
| <b>3.2.1. Sensor Leap Motion.....</b>  | <b>35</b> |
| <b>3.2.2. Proceso de Adquisición.....</b>  | <b>36</b> |
| <b>3.3. Base de Datos.....</b>   | <b>37</b> |
| <b>3.3.1. Conjunto de Palabras Usadas en el proyecto base.....</b>               | <b>37</b> |

|  |    |
|--|----|
| 3.3.2. Conjunto de Palabras Usadas en este proyecto. ....      | 38 |
| 3.4. Generación de patrones .....                              | 39 |
| 3.4.1. Parámetros .....  | 39 |
| 3.4.2. Generación .....  | 39 |
| 3.5. Comparador. ....  | 40 |
| 3.5.1. Desarrollo del <i>Dynamic Time Warping</i> , DTW.....   | 40 |
| 3.5.2. Esquema del comparador. ....                            | 40 |
| 4. Metodología experimental .....                              | 43 |
| 4.1. Introducción.....   | 43 |
| 4.1.1. Creación de código. ....                                | 43 |
| 4.1.2. Métricas de calidad. ....                               | 43 |
| 4.2. Optimización de la frecuencia de muestreo.....            | 44 |
| 4.3. Cambio de parámetros base seleccionados. ....             | 47 |
| 4.4. Implementación de mejora en el algoritmo DTW. ....        | 48 |
| 5. Resultados .....  | 49 |
| 5.1. Introducción.....   | 49 |
| 5.2. Optimización de la frecuencia de muestreo.....            | 50 |
| 5.3. Cambio de parámetros base. ....                           | 53 |
| 5.4. Cambio del algoritmo DTW.....                             | 55 |
| 5.5. Resultado global de las mejoras del proyecto. ....        | 56 |
| 5.6. Comparativa .....   | 57 |
| 6. Conclusiones .....  | 59 |
| 6.1. Valoración final.....                                     | 59 |
| 6.2. Posibles puntos de mejora. ....                           | 60 |
| 6.2.1. Puntos ya mejorados con posibilidad de mejora. ....     | 60 |
| 6.2.2. Algoritmo de creación de frases. ....                   | 61 |
| 6.2.3. Proceso traducción inverso. ....                        | 61 |
| 6.2.4. <i>Deep Learning</i> y/o <i>Matching Learning</i> ..... | 62 |
| 6.2.5. Otros puntos de mejora.....                             | 62 |
| Bibliografía .....   | 63 |
| Anexo .....  | 75 |



## PARTE II – PRESUPUESTO

|  |           |
|--|-----------|
| <b>1. Desglose del presupuesto .....</b>                               | <b>79</b> |
| <b>1.1. Recursos materiales.....</b>                                   | <b>79</b> |
| <b>1.1.1. Recursos hardware.....</b>                                   | <b>80</b> |
| <b>1.1.2. Recursos software.....</b>                                   | <b>80</b> |
| <b>1.2. Tarifas basadas en el tiempo empleado .....</b>                | <b>81</b> |
| <b>1.3. Costes relacionados con la elaboración del documento .....</b> | <b>82</b> |
| <b>1.4. Derechos por el visado el COITT.....</b>                       | <b>83</b> |
| <b>1.5. Costo de tramitación y envío .....</b>                         | <b>83</b> |
| <b>1.6. Impuestos aplicables .....</b>                                 | <b>84</b> |

## PARTE III – PLIEGO DE CONDICIONES

|                                       |           |
|---------------------------------------|-----------|
| <b>1.Pliego de condiciones.....</b>   | <b>87</b> |
| <b>1.1. Introducción.....</b>         | <b>87</b> |
| <b>1.2. Requisitos hardware .....</b> | <b>87</b> |
| <b>1.3. Requisitos software .....</b> | <b>87</b> |
| <b>1.4. Recursos humanos.....</b>     | <b>88</b> |



## ÍNDICE DE ILUSTRACIONES

|   |    |
|---|----|
| Ilustración 1 Persona signando una palabra .....  | 2  |
| Ilustración 2 Dispositivos diseñados hasta el momento para el reconocimiento gestual..... | 4  |
| Ilustración 3 Hardware de Leap Motion.....  | 35 |
| Ilustración 4 Esquema del comparador.....   | 41 |
| Ilustración 5 Matriz de confusión.....  | 44 |
| Ilustración 6 Proceso de diezmado. ....   | 45 |
| Ilustración 7 Ventanas y Solapamientos.....   | 46 |
| Ilustración 8 Representación de los porcentajes de acierto de la tabla.....               | 51 |
| Ilustración 9 Proceso de traducción inverso. ....   | 62 |

## ÍNDICE DE TABLAS

|   |    |
|---|----|
| Tabla 1 Experimentos de reconocimiento de signos usando Leap Motion.....                                  | 17 |
| Tabla 2 Dispositivos vistos en este capítulo.....   | 20 |
| Tabla 3 Resumen de todas las investigaciones vistas en este capítulo .....                                | 23 |
| Tabla 4 176 palabras usadas en la investigación del proyecto base.....                                    | 38 |
| Tabla 5 131 palabras usadas en el proyecto.....   | 39 |
| Tabla 6 Resultados de precisión del proyecto base.....  | 49 |
| Tabla 7 Precisiones individuales sin la aplicación de ninguna técnica, Sesión 1 vs Resto .....            | 49 |
| Tabla 8 Precisión media con su desviación.....  | 50 |
| Tabla 9 Precisión y tiempo medio por búsqueda, estableciendo tamaño de ventana .....                      | 50 |
| Tabla 10 Porcentajes de presiones con sus tiempos con 0.25 de solapamiento.....                           | 51 |
| Tabla 11 Precisión de comparativas de sesión con el tamaño de ventana y solapamiento<br>establecido. .... | 52 |
| Tabla 12 Porcentajes de presión y tiempo respecto a n.....  | 52 |
| Tabla 13 Nuevos % de aciertos y tiempo con N = 25.....  | 53 |
| Tabla 14 % de acierto y tiempo con el cambio 1.....   | 53 |
| Tabla 15 % de aciertos y tiempo con el cambio 2. ....   | 53 |
| Tabla 16 % de aciertos y tiempo con el cambio 3. ....   | 54 |
| Tabla 17 % de aciertos y tiempo con el cambio 4. ....   | 54 |
| Tabla 18 % de aciertos y tiempo con el cambio 5. ....   | 54 |
| Tabla 19 % de aciertos y tiempo con el cambio 6. ....   | 54 |
| Tabla 20 % de aciertos y tiempo final con el cambio 2, con todas las sesiones.....                        | 54 |
| Tabla 21 Precisión media y desviación después de realizar cambios.....                                    | 55 |
| Tabla 22 Porcentajes de éxitos iniciales con las diferentes técnicas. ....                                | 55 |
| Tabla 23 Porcentaje de éxito de las diferentes técnicas al realizar los cambios.....                      | 55 |
| Tabla 24 Resultados de precisión de cada sesión con sus tiempos.....                                      | 56 |
| Tabla 25 Media de resultados totales. ....  | 56 |
| Tabla 26 Comparación de resultados .....  | 58 |
| Tabla 27 Análisis de los objetivos del TFG. ....  | 59 |
| Tabla 28 Coste de amortización de los recursos hardware .....   | 80 |
| Tabla 29 Factor de corrección por horas trabajadas. ....  | 81 |
| Tabla 30 Presupuesto por materiales y tiempo empleado. ....   | 82 |
| Tabla 31 Coste total del proyecto.....  | 84 |



# LISTADO DE ACRÓNIMOS

| ACRÓNIMO       | SIGNIFICADO   |
|----------------|---|
| <b>ARSL</b>    | <i>Arabic Sign Language</i> . Lenguaje de signos árabe  |
| <b>ASL</b>     | <i>American Sign Language</i> . Lenguaje de signos americano  |
| <b>BI-GRU</b>  | <i>Bidirectional – Gated Recurrent Unit</i> . Unidad recurrente cerrada bidireccional.                            |
| <b>BI-LSTM</b> | <i>Bidirectional – Long Short-Term Memory</i> . Redes neuronales de memoria de corto-largo plazo bidireccionales. |
| <b>CAD</b>     | Diseño asistido por computadora   |
| <b>CMA</b>     | Cálculo de media móvil centrada.  |
| <b>CNN</b>     | <i>Convolutional Neural Networks</i> . Redes neuronales convolucionales.  |
| <b>COITT</b>   | Colegio Oficial de Ingenieros Técnicos de Telecomunicación  |
| <b>DNN</b>     | <i>Deep Neural Networks</i> . Redes neuronales profundas.   |
| <b>DTW</b>     | <i>Discrete Time Warping</i> . Deformación dinámica del tiempo.   |
| <b>ELM</b>     | <i>Extreme Learning Machine</i> . <i>Machine learning</i> extremo.  |
| <b>EMG</b>     | Electromiografía  |
| <b>FPS</b>     | Fotogramas por segundo.   |
| <b>FVV</b>     | Fusor capturador de vectores.   |
| <b>GSL</b>     | Lenguaje de signo griego.   |
| <b>GUI</b>     | Interfaz gráfica de usuario.  |
| <b>HCNF</b>    | <i>Hidden conditional neural fields</i> . Campos neuronales condicionales ocultos.                                |
| <b>HMM</b>     | <i>Hidden Markov Model</i> . Modelo oculto de Markov.   |
| <b>IGIC</b>    | Impuesto General Indirecto Canario  |
| <b>INE</b>     | Instituto Nacional de Estadística   |
| <b>K-NN</b>    | <i>K-Nearest Neighbors</i> . Vecino más cercano.  |
| <b>LSTM</b>    | <i>Long Short-Term Memory</i> . Redes neuronales de memoria de corto-largo plazo                                  |
| <b>MLP</b>     | Perceptrón multicapa.   |
| <b>NLP</b>     | <i>Natural Language Processing</i> . Proceso del lenguaje natural   |
| <b>SSL</b>     | <i>Spanish Sign Language</i> . Lenguaje de signos español   |
| <b>SVM</b>     | Máquina de soporte de vectores.   |
| <b>TFG</b>     | Trabajo de Fin de Grado.  |
| <b>TOF</b>     | Tiempo de vuelo.  |

|                     |  |
|---------------------|--|
| <b><i>ULPGC</i></b> | Universidad de las Palmas de Gran Canarias.        |
| <b><i>USB</i></b>   | <i>Universal Series BUS</i> . BUS serie universal. |
| <b><i>VR</i></b>    | <i>Virtual Reality</i> . Realidad virtual.         |

# **PARTE I – MEMORIA**





# 1. Introducción

## 1.1 Introducción

En este capítulo se establecerán las motivaciones que conllevaron a la realización de este TFG. Posteriormente, se mostrará los desarrollos e inventos anteriores, relacionados con este proyecto. A continuación, se abordará el proyecto que da base a este TFG. Seguidamente, se establecerán la hipótesis, así como, los objetivos y posibles mejoras a abordar sobre el proyecto base. Por último, se explicará las partes que contendrá esta memoria.

## 1.2 Motivaciones

Según los datos aportados por la Organización de las Naciones Unidas (ONU) [1], el 15 de noviembre de 2022, la población mundial superó los 8000 millones de personas y sugieren que, para 2030, esta supere los 8500 millones. Todas estas personas tienen varias cosas en común. De entre todas estas, la comunicación conforma unos de los elementos esenciales de la experiencia humana y la que más varía respecto a las otras, debido a su diversidad, manifestándose en distintos medios, con distintos idiomas, significados y tipos, incluyendo el lenguaje no verbal.

Por otro lado, a pesar de las aspiraciones de la humanidad por establecer esa comunicación exitosa entre ellos, no todos lo logran. Esto puede ser debido a varios obstáculos de su vida, como deficiencias e impedimentos. Estas grandes dificultades afectan a una considerable parte de la población mundial. Como ejemplo, la Federación Mundial de Sordo [2] indicó que en el mundo existen aproximadamente 70 millones de personas sordas. Como con las lenguas habladas en el mundo, existe más de 300 lenguas de signos diferentes, con sus variaciones y similitudes. Como es un mundo tan amplio, el estudio de ellas sería tardado, pero muy interesante. No obstante, en este proyecto, se centran en el *Spanish Sign Language* (SSL), el utilizado generalmente en España, evitando las pequeñas variaciones de las diferentes zonas territoriales. Esto origina un gran punto de partida para aprender sobre este mundo. Este proyecto, se intenta facilitar la comunicación entre estos dos mundos, estableciendo un puente entre ellos.

Conforme a los datos aportados por el Instituto Nacional de Estadística (INE) [3], en el 2020, en España hay un total de 1.230.000 personas con una discapacidad auditiva de distinto tipo y grado. De toda ella, 27.300 hacen uso del lenguaje de signo para comunicarse. Sin embargo, aun teniendo este grave problema, no mucha gente está dispuesta a aprender dicho lenguaje para comunicarse con ellos.



*Ilustración 1 Persona signando una palabra*

Otro punto del tema es que la digitalización de las cosas está al orden del día. El gobierno de España [4] tiene la intención de aprovechar, en todos sus aspectos, las nuevas tecnologías e introducirlas en todos los sectores posibles, con el objetivo en mente de lograr un crecimiento económico. Además, para obtener una transformación digital mayor, ha establecido a las personas en el centro de esta transformación. Esto puede ser beneficioso para casi toda la población. No obstante, para este grupo de persona, puede significar una barrera mayor a la ya existente.

Debido a todos estos puntos, se ha creado la necesidad de “romper” esos muros e intentar facilitar la comunicación entre estos dos mundos, estableciendo un puente entre ellos. Esto se ha reflejado en diferentes proyectos e investigaciones, a lo largo del mundo, para dar solución a este problema.

En cuanto a lo académico, y este proyecto utiliza Python como lenguaje principal de programación y, durante los cursos de la titulación, no se ha tenido la oportunidad de aprender este lenguaje de programación. Esto otorga una oportunidad de aprendizaje. Además, este proyecto ofrece un grado de dificultad mayor al dado por otras asignaturas de la carrera, lo que originará un desafío personal a superar.

Por último, las bases de este proyecto ya se han afrontado en la asignatura “Práctica de empresa” y dado el gran interés que me ha originado, se ha decidido el continuar con el proyecto, ampliando esas bases.

Por todos estos puntos, se ha decidido la realización de este proyecto como TFG.

### 1.3 Avances tecnológicos en el reconocimiento de la lengua de signos

A lo largo de las décadas, la humanidad ha inventado un sinnúmero de máquinas para realizar diferentes tareas, de manera más sencilla y/o más rápidas, sin requerir al humano en el proceso. Sin embargo, algunos humanos buscaban una posible interacción entre estas dos partes, una interacción humano-máquina y, no es hasta la década de los 60, cuando se desarrolla el sistema “*SketchPad*”, un programa de diseño asistido por computadora (CAD). Posteriormente, a finales de 60, se establece el concepto denominado “espacio sensitivo”, un entorno en donde se proyectaba la silueta de los usuarios y estos podían interactuar con un sistema informático. Paralelamente, se comenzó a investigar la posibilidad de crear unos guantes para capturar y analizar los movimientos de las manos mediante sensores. Esta idea conlleva que, en 1980, se creasen los “*DataGlove*”, guantes con sensores que permitían capturar movimientos de las manos y los dedos. En 1989, Nintendo diseñó y creó su propio guante “*PowerGlove*”, el cual, capturaba movimientos de la mano y los dedos para interactuar con los videojuegos. A partir de los 90, se dieron las primeras aplicaciones comerciales y de investigación, comenzando en 1994 con el “*CyberGlove*”, un guante de datos avanzados utilizado para fines de realidad virtual y simulación. En 1997, se desarrolló el “*FingerWorks iGesture Pad*”, un dispositivo multitáctil que permitía detectar gestos con múltiples dedos. En la década de los 2000, las empresas de videojuegos comercializaron productos con tecnologías de este ámbito. Nintendo, en 2006, comenzó a vender el “*Nintendo Wii Remote*”, un mando para la consola Nintendo Wii, el cual, utilizaba acelerómetros y sensores infrarrojos para detectar el movimiento en el espacio 3D. Sony no se quedó atrás y, en 2007 desarrolló el “*PlayStation Eye*”, una cámara USB que permitía el reconocimiento básico de gestos y movimientos. Por último, Microsoft en 2010, para competir con el resto, creó el “*Kinect*”, una cámara RGB, con sensor de profundidad y micrófono, que permitía el control de gestos y comandos de voz sin ningún mando. A partir de este punto, se han desarrollado diferentes sensores de Alta presión y realizada aumentada, entre estos nos encontramos en “*Leap Motion*”, desarrollado en 2012 y, el cual, es un dispositivo de sensor de movimiento que captura el movimiento de las manos y dedos con alta precisión. Por otra parte, en 2013, Microsoft, mejoró la tecnología “*Kinect*”, creando el “*Kinect 2.0*”, teniendo una mayor precisión y capacidad de detectar movimientos más sutiles. Entre el 2015 y el 2020 se han creado diferentes tecnologías vestibles y de realidad virtual con fines similares a los anteriores. En 2015, se creó el “*Myo Armband*”, un brazalete que utilizaba sensores EMG para detectar señales eléctricas de los músculos para traducirlos en comandos gestuales para controlar ciertos dispositivos electrónicos. Más tarde, en 2016, Microsoft creó unas gafas de realidad mixta, las “*Microsoft HoloLens*”, las cuales, integraban cámaras y sensores avanzados, con el objetivo de permitir la interacción gestual. En 2018, se creó “*Magic Leap One*”, un dispositivo de realidad aumentada con altas capacidades en el seguimiento de movimiento y gestos. En 2020, Apple anuncia el “*Apple LiDAR*”, una tecnología que mejora el reconocimiento

gestual y la realidad aumentada. En 2021, Meta comienza con el proyecto “*Reality Labs*”, con el que se realizan investigaciones en el campo de las interfaces neuronales y el control gestual sin ningún contacto. Por otra parte, Microsoft creó el “*Microsoft Mesh*”, una plataforma de realidad mixta, que utiliza “*HoloLens*” y otros dispositivos compatibles para capturar y renderizar gestos y movimientos de los usuarios con el objetivo de permitir una interacción colaborativa en un entorno virtual. En el año 2022, Google desarrolló el “*Project Starline*”, un avance en las videoconferencias en 3D, que permite la interacción natural mediante el reconocimiento de gestos. Amazon por su parte, creó un dispositivo inteligente con una pantalla, que sigue al usuario y permite la interacción mediante gestos. En el 2023, se desarrollaría las “*Meta Quest Pro*”, unas gafas de realidad virtual con seguimiento ocular y gestual y el “*Ultraleap Hand Tracking*”, un software avanzado de seguimiento de manos integrado en algunas VR. Por último, en el 2024, se comenzaron a comercializar la “*Apple Vision Pro*”, unas gafas de realidad mixta con control gestual avanzado.

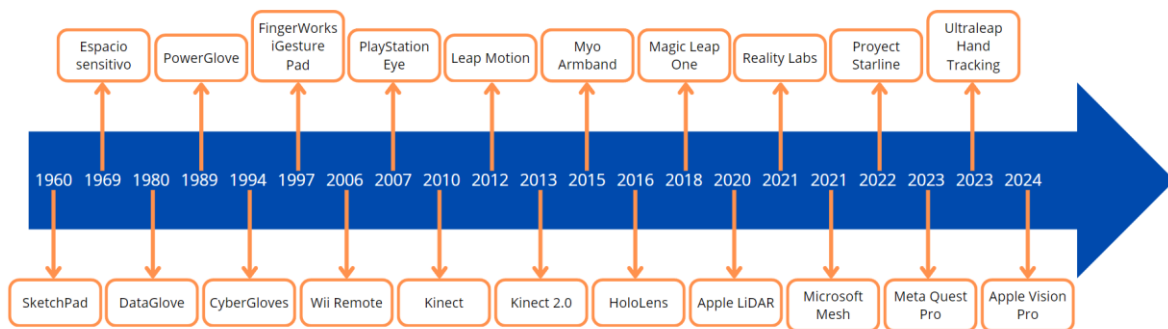


Ilustración 2 Dispositivos diseñados hasta el momento para el reconocimiento gestual

De todo esto dispositivos, el usado por el proyecto base y, por tanto, por este proyecto, es el “*Leap Motion*” [5]. Este dispositivo no ha sido usado únicamente por estos dos proyectos, si no que, se ha usado en varias investigaciones, utilizándolo como sensor volumétrico tridimensional. Estas investigaciones se pueden calificar en tres vertientes. La primera vertiente es aquella que utiliza el dispositivo para trabajar con signos estáticos, compuestos en su mayoría por letras y números. La segunda es aquella que trabaja con signos estáticos, compuestos por palabras del idioma que se esté trabajando. Por último, se encuentran los proyectos que trabajan con signos estáticos y dinámicos.

En la vertiente de las investigaciones con signos estáticos, se encuentra, por ejemplo, el desarrollado por Marín en 2014 [6], usa una máquina de soporte de vectores (SVM), obteniendo una eficiencia, para 10 signos de ASL, del 80.86%. Esta investigación tiene como característica especial que combina el Kinect con el LMC. Por otra parte, en un proyecto de la Universidad Aristóteles de Salónica, en Grecia [7], se usó 24 signos estáticos del lenguaje de signos griego (GSL), trabajando

con Perceptrón Multicapa (MLP), para los dedos y las manos independientemente. Este, obtuvo como resultado una precisión de 98.96% y 99.08%, respectivamente.

En la vertiente de las investigaciones con signos dinámicos y estáticos, se encuentra, la desarrollada por *Hisham y Hamouda* en 2018 [8], estos usaban 16 palabras estáticas con el lenguaje de signos árabe y usando *Leap Motion* y *Kinect*, obteniendo una precisión del 90.35%. Además, obtuvieron una eficiencia del 90.78% hasta el 89.12% con SVM, y del 95.22% usando K-NN. También, utilizando DTW y 20 palabras dinámicas del lenguaje de signos árabe, obtuvieron una precisión del 96.41%. Por último, en la misma investigación trabajaron con 26 letras del lenguaje de signos americano usando el MLP, obteniendo una precisión del 96.15%. Otra investigación de esta vertiente es la desarrollada por *Chong y Lee* en 2018 [9], estos trabajaron con 26 letras y 10 dígitos, obteniendo en el proceso 93.81%, utilizando solo las letras y redes neuronales profundas o DNN, 80.30% con máquinas de soporte de vectores, 88.79% al añadir al proceso los 10 dígitos para DNN y 72.79% para SVM.

Por último, en la vertiente de las investigaciones que usan únicamente signos dinámicos, se encuentra la investigación de *Elons* en 2014 [10], que obtuvo una eficiencia del 88%, usando 50 palabras del *Arabian Signal Language* (ArSL) y *Multilayer Perceptron* (MPL). Por otra parte, en un proyecto presentado en el “2021 IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference” (UEMCON) [11], se desarrolló una comparativa de diferentes técnicas de reconocimiento, usando como base 13 palabras dinámicas del lenguaje de signos americano (ASL). Con una red neuronal se obtuvo una precisión del 99.9%, con un algoritmo de aprendizaje supervisado (SVM), un 99.9%, utilizando un bosque aleatorio (un algoritmo de “*machine learning*”), se obtuvo un 99.7% de precisión, con un clasificador bayesiano ingenuo, 96.4% y, por último, con el algoritmo de K vecinos más cercanos (K-NN), se obtuvo un 98.7% de precisión.

En este bloque se puede encontrar el proyecto base de este TFG. Es el doctorado desarrollado en la Universidad de Las Palmas De G.C (ULPGC) [12], el cual usa como base 176 palabras dinámicas del lenguaje de signos en español (SSL). Este proyecto parte de la obtención de datos de los movimientos en 3D de las manos, utilizando para ello un sensor volumétrico desarrollado por *Leap Motion*, para posteriormente, ser comparadas las muestras y predecir la entrada utilizando *Dynamic Time Warping* (DTW). Con esta metodología, se ha obtenido como resultado, una precisión del 95.17%.

## 1.4 Proyecto base del TFG

Como se ha mencionado anteriormente, este proyecto parte de una tesis doctoral desarrollado por un alumno de la ULPGC, que tuvo como objetivo general el “crear un sistema de reconocimiento de palabras del lenguaje de signos español”, para lograr esto, se desarrolló el siguiente sistema:

Se comienza con un bloque de adquisición de datos, en este caso, estos serían datos de grabaciones de palabras dinámicas de lenguaje de signos español y datos obtenidos, en el momento, para su traducción de lenguaje de signos a voz. Para capturar los movimientos de las palabras dinámicas, se seleccionó el sensor tridimensional “*Leap Motion*”, el cual, puede capturar el movimiento de las manos, los dedos y los brazos y convertirlos en series de tiempo ordenados por momentos de captura y parámetros de la captura, con esto, se es capaz de diferenciar las partes de las manos, así como su posicionamiento en un *frame* exacto. Para cada una de las palabras dinámicas grabadas, se genera un archivo de texto con cada muestra, con sus parámetros. Este dispositivo está diseñado para proporcionar 276 parámetros, en donde se tiene los movimientos de las manos y los dedos, y su posición en un momento dado. Cabe destacar que el “*Leap Motion*” es capaz de alcanzar hasta los 200 *fps* a la hora de realizar las grabaciones, hecho que puede conllevar a la sobreinformación de la muestra y generación de ruido innecesario, y que requiere de alguna técnica para darle solución. En el proyecto base, se da solución a este problema aplicando un diezmado, cosa que será revisada y mejorada en este proyecto.

Con el dispositivo de captura de datos seleccionados, se llegó a un proceso de adquisición de muestras para el sistema. Estas se utilizarían para entrenar al sistema y para verificar su buen funcionamiento. Para obtener dichas muestras, seleccionaron varios participantes con la capacidad de comunicarse en el lenguaje de signos español, y en días y momentos distintos se realizó las grabaciones de las distintas palabras. Una vez realizado las grabaciones, se obtuvieron 5780 muestras, de las cuales, 3520 se utilizarían para entrenar el sistema y 2260 para verificarlo.

Otro de los puntos importante del sistema se centra en que palabras se deberán usar para este sistema, debido a la imposibilidad de tener todo el vocablo del lenguaje de signos español. Se tuvo que seleccionar un ámbito o sector al que estaría orientado el sistema, para este caso se escogió palabras de uso cotidiano, verbos y relacionadas con la salud. Con todo esto en mente, se seleccionó 176 palabras, en total, para el proyecto base.

Otro de los aspectos importantes era la selección de parámetros de la muestra, con el objetivo de mejorar el rendimiento del sistema, punto que también se abarcará en este proyecto. De todos los parámetros existentes en el proyecto base se seleccionó 74 parámetros, los cuales, incluían parámetros tales como: rotación de cada una de las partes de los dedos, posición de cada dedo, etc.

Por otra parte, para maximizar la precisión, se creó un patrón que contuviera todas las características del signo sin que perdiera información importante. Creándose así un patrón para cada signo, es decir, 176 patrones en total.

Con todos estos patrones creados, en un bloque apartado del sistema, se llegaría al bloque comparador que, con la técnica de clasificación DTW, se realizaría una comparación entre la muestra entrante, en ese momento, con cada patrón de cada palabra, obteniendo en el proceso una distancia con cada patrón y se buscaría la distancia mínima, en este caso, cual de estos patrones/signos se parece más a la muestra entrante. En este proyecto, este bloque también se revisará en busca de técnicas de clasificación mejores.

Una vez obtenida la distancia mínima y, por tanto, el patrón correspondiente a la palabra entrante, se tendría la palabra en texto, y con un conversor texto-voz, se obtendría la palabra hablada.

Todo este sistema está sujeto a fallo, debido a que, en algunos casos, puede que no se obtenga el signo correcto. El sistema, por tanto, tiene un margen de precisión, que podría mejorarse, y se mejorará en este proyecto. Por otra parte, el proceso entre la realización del signo, delante del sensor, y su conversión a voz, es tardado, llegando a ser de hasta 5 o 6 segundos por cada signo. Este punto, también tiene margen de mejora, que se mejorará en este proyecto.

## 1.5 Hipótesis y Objetivos

En este TFG, se propone validad la siguiente hipótesis:

“Es posible, dado un proyecto base con unas características y cualidades especificadas, con una velocidad y una precisión establecidas, analizar y mejorar las partes de dicho proyecto”

El objetivo principal que persigue la realización de este Trabajo de Fin de Grado es mejorar y optimizar el proceso de adquisición y procesado de datos del proyecto existente.

Objetivos específicos:

- O1:** Analizar las soluciones de detección automática del lenguaje de signos existentes en la actualidad.
- O2:** Mejorar la velocidad en la detección del lenguaje de signos del proyecto base.
- O3:** Mejorar la precisión en la detección del lenguaje de signos del proyecto base.
- O4:** Programar y mejorar el código del proyecto base.

## 1.6 Peticionario

El peticionario del presente Trabajo Fin de Grado es la Escuela de Ingeniería de Telecomunicación y Electrónica (EITE) de la Universidad de Las Palmas de Gran Canaria (ULPGC), en calidad de institución pública que solicita la realización de dicho trabajo con el fin de superar los requisitos impuestos en obtención del título del Grado de Ingeniería de Telecomunicaciones en la especialidad de Telemática.

## 1.7 Estructura de la memoria

Este TFG está compuesto por 8 capítulos y la bibliografía. Posteriormente de este capítulo, la estructura de la memoria es la siguiente:

- Capítulo 2: contiene el estado del arte, en el cual se establece un estudio de los diferentes proyectos centrados en los sensores existentes en la actualidad y en las técnicas usadas para dicho fin.
- Capítulo 3: se realiza un estudio de los materiales y métodos usados en el proyecto base y este proyecto, así como de la base de datos del proyecto origen, con los diferentes datos de las grabaciones, aportados por el sensor de movimiento “*Leap Motion*”, de las diferentes palabras del lenguaje de signos y se realizará un análisis de estas para su posterior tratamiento. Se comprobará posibles puntos de mejora del proyecto.
- Capítulo 4: se muestra el proceso seleccionado en las técnicas de optimización de la frecuencia de muestreo del sensor de movimiento en *Python*. Se expone un análisis realizado de los parámetros proporcionados por el sensor y un estudio de mejora en la selección de dichos parámetros necesarios en el proyecto. Se especifica un análisis de diferentes algoritmos DTW existentes para su posible implementación en el proyecto.
- Capítulo 5: se exhiben los resultados obtenidos en los diferentes cambios y experimentos del proyecto base para validar la hipótesis establecida y un análisis de estos.
- Capítulo 6: muestra una valoración final y posibles puntos de mejora que puedan ser implementados, a futuro, en el proyecto.
- Bibliografía: en esta parte se incluye toda la bibliografía necesaria para el desarrollo de este proyecto.



## 2. Estado del arte

### 2.1. Introducción.

Un apretón de mano, un saludo educado, el hecho de encoger los hombros para expresar duda sobre algo, el negar o afirmar con la cabeza en respuesta a una pregunta. Todas estas acciones tienen algo en común, que es el hecho de ser formas de comunicación no verbal, ya que no ha requerido ninguna palabra para transmitir información esencial. La comunicación no verbal, especialmente la que requieren gestos, juegan un papel importante en las interacciones entre humano, complementando, reforzando y, en algunos casos sustituyendo, el lenguaje verbal. Permitiendo aportar nuevas capas de emociones e intenciones a las palabras habladas.

Este tipo de comunicación data de antes del nacimiento de la propia humanidad en sí, dando en los comportamientos primigenios de las primeras formas de vida inteligente, muchísimo antes de que los seres humanos crearan y desarrollaran el lenguaje hablado o escrito. Este fue clave para la supervivencia y el desarrollo de las especies, permitiendo a estas coordinarse y avisar en caso de peligro y compartir importante información de su entorno natural.

La comunicación no verbal ha ido evolucionando, pasando de los gestos más habituales como el saludar con una mano, dar una negación moviendo un dedo de un lado a otro, hasta llegar a conformar lenguajes no verbales, lenguajes creados a partir de gestos. Estos lenguajes se han diseñado para diferentes ámbitos: militar, buceo, aviación, etc. Uno de estos ámbitos es facilitar la comunicación con personas sordas.

Por otra parte, las tecnologías han sufrido una gran evolución a lo largo de las décadas, obteniendo, en el proceso, herramientas más pequeñas, más rápidas y precisas, que son capaces de realizar un sinnúmero de tareas. Además, con la creación de estas tecnologías ha surgido una idea, entre la humanidad, de combinar la eficacia de las tecnologías, con la versatilidad, innovación y pensamientos de los humanos, obteniendo una relación hombre-máquina, en donde el humano interactuaría con la máquina, para que realice una acción o proceso y esta, la ejecute, siguiendo las indicaciones del humano. Para que se pudiera llevar esto a cabo, se requería un sistema intermedio, el cual, diera entrada a las indicaciones del humano y se las enviara a la máquina.

En este punto, se han diseñado diferentes métodos para suplir esta idea: pantallas táctiles, botones, mandos de control, teclado-ratón, etc. Sin embargo, todos estos métodos requieren un *input* físico (el pulsar algo), hecho que en ciertas circunstancias no pueden llegar a darse. Por ejemplo, al conducir un vehículo, el conductor no debe distraerse interactuando con algo que no sea estrictamente necesario. También, existen personas que no pueden utilizar estos tipos de *inputs* debido a algún problema y requieran de otro tipo de método.

Para estos casos, existe el reconocimiento de gestos. Con la creación de la fotografía y el vídeo digital, ha permitido capturar imágenes y videos de esos gestos para procesarlos y poder identificarlos y reconocerlos. Por otro lado, la creación de inventos como las cámaras de alta calidad y el sensor de profundidad ha permitido llevar un paso más allá este método, consiguiendo un mejor reconocimiento en tiempo real.

No obstante, este tipo de reconocimiento no es fácil, ya que existen posibles inconvenientes y malentendidos de los gestos. Una posibilidad es que puedan existir varios significados del mismo signo, que el mismo signo pueda ser identificado de diferente manera, según la localización, el idioma, la cultura, etc. Una situación de este problema son las lenguas de signos, ya que, según el idioma o la región o zona, estos puedan significar distinto.

## **2.2. Evolución de los Dispositivos de Reconocimiento de Gestos.**

Uno de los objetivos principales de la comunicación gestual es el conseguir un reconocimiento directo de dichos gestos, para esto, los sensores han significado una parte crucial para cumplir este objetivo. Esta evolución comienza en 1963, con la creación, por parte de *Sutherland* [13], de una máquina capaz de obtener, a través de una pantalla, dibujos realizados por el humano. A finales de los 60, Krueger establece la idea de “espacio sensitivo” [14], una idea inicial de realidad virtual, que supondría el comienzo de un mundo donde el humano podría interactuar con entorno digital de una manera más realista.

Otro tipo de dispositivo de reconocimiento de gestos llegó en la década de los 80. Estos consistían en capturar el movimiento de las manos y/o brazos gracias a unos guantes con distintos sensores. Por otro lado, se estudió el obtener la posición de los elementos a través de mediciones de campos electromagnético, en dónde se encontraría dicho elemento a localizar. Paralelamente, en los 90, surgieron investigaciones que tenían como objetivo el analizar imágenes y vídeos en busca de los gestos, estableciendo la posibilidad de reconocer gestos deseados, localizados en dichos elementos.

### **2.2.1 Guantes de Datos.**

Esta tecnología consiste, como su nombre indica, en guantes que permiten obtener datos referentes al movimiento de las manos, así como su posición respecto al espacio. Para poder lograr esto, la mayoría de los guantes poseen en su interior o exterior un conjunto de sensores, del mismo o distinto tipo. Según el tipo de sensor y la calidad de este, permitirá la obtención de datos más precisos y/o de mayores puntos de referencia, pudiendo obtener no solo valores de la mano en general, sino también de cada parte de la mano, la muñeca, etc. Este dispositivo ofrece la ventaja de obtener los datos, sin requerir procesarlos para obtenerlos, a diferencia de las cámaras. No obstante, esta ventaja tiene el

costo de que esta tecnología es incómoda para el usuario, ya que requiere un constante uso de esta para la obtención de los datos.

Los guantes de datos pueden clasificarse dependiendo de si tienen algún elemento electrónico, siendo en su mayoría sensores. Si no poseen ninguno de estos, son guantes pasivos. Estos suplen esta carencia con alguna característica especial (distintos colores en las diferentes partes de la mano, diferentes texturas, etc.) Si, por el contrario, poseen algún elemento electrónico, se les denomina guantes activos.

Inicialmente, en 1977, se creó los primeros tipos de guantes, estos fueron el *Sayre Glove*. Su funcionamiento se basaba principalmente en tubos flexibles conectados a un emisor de luz en un extremo y a un sensor fotoeléctrico, al otro extremo. Estos sensores median la intensidad de cada haz de luz, la cual, variaría dependiendo de cuanto se doble los tubos y la posición de cada dedo.

Posteriormente, en 1983, el ingeniero Zimmerman, desarrollo lo que se conoció como el primer guante de realidad virtual, el *DataGlove* [15]. Este consistía en unos guantes con fibras ópticas ligeras, pegadas en el dorso de los dedos. Estas fibras irían conectadas a unos sensores, los cuales, recogerían información de cuándo la luz de las fibras se redujera al doblar los dedos, proporcionando el ángulo de movimiento de estos dedos. Debido a la varianza en el tamaño de las manos de los humanos, los guantes requerían una pre-calibración, antes de su uso.

Después de la creación de estos guantes, nuevos proyectos e investigaciones fueron surgiendo, mejorando los puntos negativos de los anteriores y adaptándose a las nuevas tecnologías que iban apareciendo. El *Power Glove* y el *Super Glove* fueron dos de estos proyectos. Estos tenían la característica de tener resistencias eléctricas para medir la flexión de los dedos, obteniendo unos guantes menos precisos, pero más baratos [16]. Para dar solución a este problema de precisión, se mejoraron los guantes, añadiendo sensores magnéticos y acelerómetros.

Hoy en día, el auge de este tipo de guante aún se mantiene, apareciendo y desarrollándose nuevos guantes con nuevas tecnologías y mejorándose alguno de los existente. Estos se utilizan con objetivos comerciales y académicos. Una de estas nuevas creaciones es el *MIT AcceleGlove* [17], utilizado mayoritariamente en el campo de la realidad virtual. Otra invención novedosa fue el *CyberGlove* [18], creado y usado en la industria del cine y la animación gráfica.

Como se puede comprobar, esta tecnología ha tenido una gran evolución a lo largo del tiempo. Además, por su facilidad de aplicación, esta se ha extendido a un sinfín de campos, desde la medicina hasta el entretenimiento, pasando por el campo militar.

### **2.2.2. Electrodo EMG (Electromiografía).**

Todos los proyectos e investigaciones no se han centrado únicamente en los guantes, también existen proyectos que tienen como uno de sus objetivos el reconocimiento gestual, pero utilizan otros tipos de tecnología para lograrlo. Este es el caso de las pulseras compuestas con sensores de electromiografía. Estas se han diseñado para medir potenciales producidos por las células musculares del cuerpo.

Algunos de los estudios que usan esta tecnología son, por ejemplo, las señales de electrocardiograma (propias de la actividad eléctrica del corazón), las señales de electrooculografía (propias de la actividad eléctrica proveniente de los movimientos del ojo), las señales de electroencefalografía (propias de la actividad eléctrica cerebral), entre otros.

Entre todos estos estudios, se encuentra la electromiografía, la cual, se encarga del estudio de actividad eléctrica proveniente de los nervios y músculos. Esta actividad es registrada mediante electrodos pegados a la zona que se desea analizar. Las señales EMG son obtenidas, más específicamente, de los músculos esqueléticos, los encargados de mover el cuerpo humano [19].

Algunos de los usos que se ha dado a esta tecnología han sido para rehabilitación médica [20], diagnósticos médicos [21], algunos estudios centrados en ayudar a personas con discapacidad física a poder interactuar con máquinas. En estos últimos estudios, se encuentran los desarrollados por Barreto, en 1999 [22], Coleman, en 2001 [23], Guerreiro y Jorge, en 2006 [24] y Ahsan, en 2009 [25], los cuales, intentaban sustituir el ratón o aplicar reconocimiento de gestos de forma estática, para poder interactuar con la máquina, los desarrollados por Crawford, en 2005 [26], Kiguchi y Hayashi, en 2012 [27] y Joshi, en 2013 [28], los cuales, incluyen el control robótico para cumplir este objetivo. Otros estudios han sido, el reconocimiento de gestos genéricos, con el objetivo de controlar un reproductor de música, diseñado por Fiste y Tanaka, en 2002 [29], de usar esos gestos como un teclado numérico, diseñado por Wheeler y Jorgensen, en 2003 [30], y por Dhillon, en 2009 [31], o simplemente como un puntero para una interfaz gráfica de usuario (GUI), diseñado por Rosenberg, en 1998 [32].

En el 2024, se ha desarrollado una investigación, la cual, usa un sistema de reconocimiento de gestos basado en electromiografía de superficie que, junto a un algoritmo de Regulación de Tiempo dinámico, obtuvieron una precisión del 93,75 %, enfocándose en seis tipos de movimiento de la mano [33].

En cuanto al uso comercial, se pueden encontrar algunos productos destacados, tales como el brazalete Myo, creado por Thalmic Lab, el cual, ha sido usado en varios experimentos y aplicaciones.

Por ejemplo, en 2015 podemos encontrar el estudio realizado por Boyali [34], en 2016, el realizado por Abreu [35], y el realizado por Sathiyarayanan y Rajan, y en 2017 [36], el realizado por Motoche [37].

En cuanto al reconocimiento del lenguaje de signos, podemos encontrar la investigación de Singh y Chaturvedi, en 2023 [38], los cuales, consiguieron reconocer 24 letras del lenguaje de signos americano, con una precisión del 99,91 %.

### **2.2.3. Ultrasonidos.**

Otra de las tecnologías que pueden ser usadas para capturar gestos es la de ultrasonidos. Esto puede conseguirse de dos maneras distintas. Una es a través del uso del efecto *Doppler*, usando la radiación ultrasónica en cualquier estancia. La otra, utilizando imágenes ultrasónicas.

La primera, consiste en usar transmisores, que emiten unos tonos ultrasónicos, los cuales, rebotan en un objeto, en este caso, manos y brazos. Posteriormente, un sensor es colocado para capturar las señales que se reflejan, con el objetivo de analizarlas y procesarlas. Si el objeto a medir está estático, la frecuencia a la que se emite los tonos no se verá variada, pero si su fase y amplitud. Por el contrario, si el objeto se encuentra en movimiento, se producirá un cambio de frecuencia por el desplazamiento *Doppler* que, al ser recibido por el sensor, es capaz de obtener la velocidad del objeto móvil.

De esta técnica, se pueden encontrar diversas investigaciones, por ejemplo, en el 2000, se desarrolló, un sistema de reconocimiento de gestos simple, que utilizaba ultrasonidos para el control de la música [39]. Por otra parte, en el 2011, se estudió la posibilidad de crear un sistema de reconocimiento de gestos, utilizando, en un robot móvil, un telémetro ultrasónico convencional [40]. Otro desarrollo fue realizado en la Universidad de Berkeley, donde se diseñó un chip con transductores ultrasónicos para el reconocimiento de gestos [41] que, posteriormente, se siguió investigando, con el objetivo de obtener un hardware más comprimido, con el mismo número de sensores [42].

La segunda técnica usada, las imágenes ultrasónicas, consiste en utilizar ondas sonoras de alta frecuencia para crear imágenes del interior del cuerpo. Esta técnica se basa en el principio del eco: ondas sonoras son emitidas la zona deseada y estas rebotan en los tejidos y partes del cuerpo para, posteriormente, ser captadas por sensores para ser analizadas y procesadas.

Alguna de las investigaciones, sobre este tema, surgen a partir del 2003, con la intención de medir la contracción muscular de algunos músculos [43]. Posteriormente, en 2013, surge un proyecto para la creación de una banda, que se colocaría en el antebrazo y los dedos, la cual, tendría un conjunto de transductores ultrasónicos [44]. En 2017, se creó una sonda, que se colocaba en el antebrazo, con el objetivo de detectar y monitorizar los posibles gestos de una mano [45].

#### **2.2.4. WiFi.**

Las redes *WiFi* no han sido utilizadas solamente para dar soporte a Internet. Estas infraestructuras también han sido utilizadas como medio para el reconocimiento de gestos, debido a la posibilidad de reconocer gestos, sin la necesidad de tener una visión directa de estos. Una de las investigaciones, se ha centrado en el indicador de fuerza de la señal recibida, creando en el proceso, *WiGest* [46], un sistema de reconocimiento de gestos, que utiliza esta tecnología. Este sistema posee una precisión de entre el 96 % y el 87,5 %. Otro proyecto, es el *WiSee* [47], un sistema que reconoce nueve gestos corporales, con el fin de interactuar con algunos electrodomésticos, que posean dicha tecnología.

Otros proyectos son el *Witrack* [48] y el *Witrack0* [49]. Estos poseen la cualidad de reconocer los gestos humanos, aun no teniendo visión directa. Por otro lado, se creó un sistema con una precisión del 89 % en lugares de nula visión y del 91% en lugares de visión directa, siendo capaz de reconocer hasta cuatro gestos [50]. Por último, se diseñó y creó el *WiFinger* [51], un sistema con la capacidad de obtener los movimientos de los dedos, con una precisión del 93%.

Actualmente, en el 2024, se ha desarrollado el *WiCGesture* [52], un método que usa las señales *WiFi* para reconocer los gestos humanos, de manera ininterrumpida, a partir de dos conjuntos de gestos.

#### **2.2.5. Cámara RGB y RGB-d.**

Este tipo de cámaras son las más usadas para captura gestos. Caracterizado por hacer uso de los tres canales RGB para capturar la luz y obtener un algoritmo de visión artificial, organizando la imagen, a partir de una matriz de píxeles. Esta visión cobra un significado al interpretar la intensidad y el orden del color.

Dado su gran uso, existen un sinnúmero de investigaciones que usan esta tecnología para el reconocimiento de gestos. Entre todas estas nos encontramos con: una investigación que obtuvo una precisión del 99 %, al utilizar CNN para reconocer 32 signos del lenguaje de signos árabe [53]. Otra que utiliza el mismo lenguaje, donde utiliza *MobileNet* como sistema basado en CNN, utilizado en dispositivos móviles como un modelo de visión artificial, y obteniendo en el proceso, una precisión del 94,96 %, para 32 signos [54]. Un método de clasificación bidireccional con redes neuronales con memoria a corto plazo, el cual, con 100 signos, es capaz de obtener una precisión del 98,75 % [55].

##### **2.2.5.1. Visión Estereoscópica.**

En esta tecnología, se hace uso obligatorio de dos cámaras para obtener las vistas necesarias de la escena, ya que esta, comparte similitud con los ojos humanos, y su capacidad de visualización de forma binocular. Para obtener la profundidad del plano es requerido de cálculos provenientes de las

dos imágenes, de las dos cámaras. Otros aspectos para tener en cuenta, en este caso, es la distancia focal y la separación entre las dos cámaras [56].

### **2.2.5.2. Luz Estructurada.**

A diferencia de la anterior tecnología, hay una fuente de luz, que ilumina la escena completa, variando en las coordenadas X e Y. Por lo tanto, ya no se requiere el uso de dos cámaras [57].

### **2.2.5.3. Cámara ToF o de tiempo de vuelo.**

El funcionamiento principal se centra en la medición de la distancia entre las cámaras, establecidas y el objeto en su campo de visión, para ello, se comprueba el tiempo que tarda una señal, generalmente, una luz infrarroja, en viajar desde la cámara hasta el objeto y viceversa.

Si se quiere crear un sistema para el reconocimiento de gestos, con esta tecnología, se debe establecer unos pasos, los cuales son: recopilar y procesar los datos, modelar el objeto capturado, definir el gesto, detectar los gestos iguales y controlar las transiciones.

Algunos trabajos que se han desarrollado, con esta tecnología, son los siguientes: Un proyecto con el objetivo de ser capaz de reconocer ciertas posturas al caminar, agacharte o sentarte [58]. Un sistema para reconocer movimientos de baile, haciendo uso de las *Kinect* [59]. Esta investigación obtuvo una eficacia del 96,9 %.

### **2.2.6. Leap Motion.**

Proporciona la capacidad de obtener la captura del movimiento de los dedos, los brazos y las manos en tiempo real [60]. Siendo unas de las tecnologías más usadas por su bajo precio y su compacto tamaño. Hecho que ofreció la posibilidad de usar esta herramienta en varios sectores como la música [61], la educación [62], la identificación y autenticación [63], el control de robots militares, a través del reconocimiento de signos [64] o la domótica [65].

Comprobado su calidad y precisión en diferentes estudios [66], se llegó a la conclusión de que con esta tecnología se obtienen los mejores resultados, frente a otros.

Esta tecnología es usada en una gran cantidad de investigaciones de reconocimiento de gestos, tanto dinámicos, como estáticos.

| Referencia | Fecha | Autor           | Procedimiento                         | Resultado            |
|------------|-------|-----------------|---------------------------------------|----------------------|
| [67]       | 2014  | Mohandes, Aliyu | Lenguaje de signos árabe<br>28 letras | 99 % de<br>precisión |

|      |      |                          |  |   |
|------|------|--------------------------|--|---|
|      |      |                          | Redes neuronales multicapa<br>Clasificador Naves Bayes   |   |
| [68] | 2015 | Y. Chen                  | Lenguaje de signos árabe<br>36 gestos entre números y alfabeto<br>Máquina de soporte de vectores               | 80 % de precisión   |
| [69] | 2015 | McCartney                | Lenguaje de signos árabe<br>100 participantes<br>Redes neuronales convolucionales<br>Modelos ocultos de Márkov | 92,4 % de precisión   |
| [70] | 2016 | Lu                       | Varios gestos dinámicos<br>Clasificador HCNF<br>Redes neuronales   | 89,5 % de precisión   |
| [71] | 2022 | Saad                     | Lenguaje de signos americano<br>30 signos<br>Clasificadores Bi-GRU<br>Clasificadores Bi-LSTM                   | 96,97 % de precisión con Bi-GRU<br>97,98 % de precisión con Bi-LSTM |
| [72] | 2022 | B. Wuet                  | Lenguaje de signos chino<br>20 palabras dinámicas<br>LSTM modificado   | 99,55 % de precisión  |
| [73] | 2022 | Abdullahi y Chamnongthai | Lenguaje de signos americano<br>57 palabras<br>Fusor capturador de vectores (FFV) con Bi-LSTM                  | 91 % de precisión   |



|      |      |                         |  |   |
|------|------|-------------------------|--|---|
| [74] | 2022 | Nasir                   | Lenguaje de signos americano<br>26 letras<br>K-NN<br>SVM   | 72,78 % de precisión con K-NN<br>79,83 % de precisión con SVM |
| [75] | 2022 | Rishan                  | Lenguaje de signos de Sri Lanka<br>3 palabras y 3 letras<br>Coincidencia de plantillas geométricas y procesamiento de lenguaje natural (NLP) | 77 % de precisión   |
| [76] | 2023 | Katilmis y Karakuzu     | Lenguaje de signos turco<br>26 palabras dinámicas<br>Extreme learning machine (ELM)<br>Uso del modelo de meta-aprendizaje de ELM (Meta-ELM)  | 93 % de precisión   |
| [77] | 2024 | Bekiri, Sarra y Mohamed | Lenguaje de signos americano<br>Técnica de aprendizaje por transferencia con datos aumentados<br>20000 imágenes                              | 99,12 % de precisión  |

Tabla 1 Experimentos de reconocimiento de signos usando Leap Motion.

### 2.2.7. VR o Realidad Virtual.

Esta tecnología ha tenido un gran avance a lo largo de los años, ofreciéndonos, actualmente, bellos y extensos espacios, los cuales, la humanidad ha aprovechado en varios campos como, por ejemplo, la medicina, usos militares, ocio, educación, la ingeniería, la arqueología, entre varios otros.

Uno de los grandes productos, que han utilizado esta tecnología, son las gafas de realidad virtual con empresas como Valve, Meta, HP, Sony, Apple y HTC desarrollando sus productos: *Oculus Quest*, *HTC Vive*, las gafas *Valve Index* o *Apple Vision Pro*.

Centrado en las investigaciones con el reconocimiento de gestos, podemos encontrar el trabajo de Leth, en 2023 [78], el cual, usando 25 signos del lenguaje danés y el NLP, obtiene una precisión del 44,4 % con signos dinámicos y 41,3 % con signos estáticos.

### **2.2.8. *UltraLeap.***

Este es uno de los inventos más modernos, creado y comercializado en el 2019. Este producto posee la capacidad de manipular el dispositivo, dándote la sensación de tocar los objetos en el aire. Es posible, gracias a su novedosa tecnología de ondas, que genera dicha sensación. El mayor problema de este producto es su precio, llegando a alcanzar los 4000 euros, imposibilitando a algunos investigadores su compra.

No obstante, su precio no ha echado a atrás a algunos de los investigadores, comprando dicho producto y usándolo en algunas de sus investigaciones en robótica, combinándolo con gafas de realidad virtual [79], en campos relacionados con las señales hápticas o táctiles [80], en la creación de interfaces para la automoción [81], en un proyecto que relaciona unas gafas de realidad virtual, para observar objetos, y dicho producto, para ofrecer una sensación táctil [82].

Otro proyecto relacionado a este TFG es el desarrollado, en 2020, por Paneva, el cual, usa la tecnología para el reconocimiento del lenguaje braille para personas ciegas [83].

### **2.2.9. *Kinect y Azure Kinect.***

#### **2.2.9.1. *Kinect.***

Desarrollado por Microsoft, con objetivos de comercialización y entretenimiento. Era una de las mejores elecciones, en su momento, para el reconocimiento corporales y gestuales. Esto era debido a su bajo precio y sus características.

Dado a las características que poseía esta tecnología, existían innumerables proyectos relacionados con el reconocimiento de gestos y signos de diferentes lenguas.

Para el reconocimiento del lenguaje de signos árabes, se puede encontrar el desarrollado por Abdel-Samie, en 2018 [84], el cual, consistía en 30 palabras, con una precisión de entre 95,25 % y 97,58 %. También se puede encontrar un proyecto que trabaja con 16 palabras y obtiene una precisión del 80,47 %, usando HMM [85].

En cuanto al lenguaje de signos americano, está el proyecto que utiliza una máquina de soporte de vectores latente, para reconocer 2000 frases, con una precisión del 84 % [86]. La investigación que trabaja con 1000 signos y con HMM para ofrecer una precisión del 76,12 % [87].

Por otra parte, existen investigaciones que hacen uso de otras lenguas de signos, diferentes a las ya nombrada. Entre estas encontramos: una investigación con lenguaje de signos alemán, con una precisión del 97 % [88]. Una que hace uso de 20 signos del lenguaje de signos italiano, para obtener 91,7 % de precisión, usando redes neuronales convolucionales (CNN) [89]. Otra que utiliza 37 signos del lenguaje de signos indio, que ofrece una precisión del 86,16 % [90].

### **2.2.9.2. Azure Kinect.**

Esta tecnología, a diferencia de su antecesora, fue creada y comercializada, en 2019, para usos en aplicaciones profesionales [91]. Caracterizada por tener una gestión de entorno. Sin embargo, sufre deficiencias en establecer, de forma clara, la posición y el movimiento de las manos y los gestos.

Entre las investigaciones existentes, se encuentra la desarrollada por Brock y Gomez, en 2022 [92], la cual, se caracterizan en trabajar con 18 gestos, obteniendo una precisión del 41 %, usando CNN.

## **2.3. Conclusiones.**

Analizando todas las tecnologías e investigaciones surgidas alrededor del reconocimiento de gestos, se puede llegar a comprobar que estas no han parado de salir y evolucionar, a lo largo de los tiempos, iniciando en la década de los 60 con el *Sketchpad*, pasando por los diferentes guantes, *AcceleGlove*, *PowerGlove*, *CyberGlove*, *SuperGlove* y *DataGlove*. Por otro lado, a partir del 2010, se comenzaron a desarrollar y comercializar diferentes cámaras con tecnologías similares que cumplían la misión de los guantes, de una manera menos incomoda y más precisa. En esta parte podemos encontrar el *Kinect*, las cámaras ToF, el *Leap Motion*, entre otros.

El *Leap Motion* tiene la característica de poseer varios componentes ToF, de ser capaz de reconocer movimientos de las manos, obteniendo el objetivo de interpretar los gestos de estas. Algunos de los puntos positivos de esta tecnología, es su precio bajo, su fácil transporte, debido a su tamaño, su facilidad para reconocer las manos rápidamente, su posibilidad de trabajar con él en múltiples plataformas, su facilidad de integración en diferentes aparatos electrónicos y su elevada precisión, llegando incluso al 99 %, obtenida en varias investigaciones.

En la otra parte, se encuentra el ultrasonido y el Wifi, que presentan problemas de detección de gestos de manos.

Por este motivo, se ha seleccionado el *Leap Motion* para la tarea de capturar y procesar los movimientos gestuales, de manera precisa y eficiente.

Tabla 2 Dispositivos vistos en este capítulo.

| <b>Dispositivo</b> | <b>Autores</b>        | <b>Año</b> | <b>Aplicación</b>                             | <b>Objetivo</b>                             | <b>Resultado</b>   |
|--------------------|-----------------------|------------|---|---|--|
| <i>Sketchpad</i>   | <i>Sutherland</i>     | 1963       | Reconocimiento táctil                         | Primer invento de interacción táctil        | Nuevo campo en la relación hombre-máquina  |
| <i>VideoPlace</i>  | <i>Krueger et al.</i> | 1969       | Realidad virtual                              | Alternativa a la interacción hombre-máquina | Combina imágenes de video con un mundo gráfico.  |
| <i>DataGlove</i>   | <i>Zimmerman</i>      | 1983       | Reconocimientos gestuales                     | Primer guante con sensor de flexión óptica  | Varios usos de las señales de los sensores.  |
| <i>PowerGlove</i>  | <i>Matter</i>         | 1989       | Realidad virtual<br>Reconocimientos gestuales | Para juegos de Nintendo                     | Alternativa de bajo costo para investigaciones en realidad virtual y reconocimientos de gestos |

|                    |                           |           |  |  |   |
|--------------------|---------------------------|-----------|--|--|---|
| <i>SuperGlove</i>  | <i>Nissho Electronics</i> | 1997      | Reconocimientos gestuales  | Sensores resistivos especiales   | Mide la flexión de las articulaciones de los cuatro dedos y el pulgar |
| <i>CyberGlover</i> | <i>CyberGlove System</i>  | 1999      | Realidad virtual<br>Control robótico<br>Rehabilitación médica<br>Reconocimientos gestuales | Guantes<br>Acelerómetro<br>Sensores magnéticos   | El mayor impacto comercial<br>Guantes programables                    |
| <i>AcceleGlove</i> | <i>AnthroThonix</i>       | 1999      | Realidad virtual<br>Control robótico<br>Rehabilitación médica<br>Reconocimientos gestuales | Guantes<br>Acelerómetro<br>Sensores magnéticos   | El mayor impacto comercial<br>Guantes programables                    |
| <i>Myo</i>         | <i>Thalmic Lab</i>        | 2014      | Realidad virtual<br>Reconocimientos<br>Gestuales<br>Automatización del hogar               | Permite leer y recoger la actividad eléctrica de los músculos<br>Permite controlar dispositivos con gestos | Pulsera mioeléctrica<br>Gran impacto comercial                        |
| <i>Kinect</i>      | <i>Microsoft</i>          | 2010-2017 | Juegos<br>Reconocimientos gestuales  | Permite interactuar con la consola, si requerir contacto físico  | 10 millones de ventas.  |

|                     |                       |      |   |  |   |
|---------------------|-----------------------|------|---|--|---|
|                     |                       |      | Control robótico<br>Rehabilitación médica<br>Educación<br>Identificación<br>Autenticación   |  |   |
| <i>Leap Motion</i>  | <i>UltraLeap Inc.</i> | 2012 | Realidad virtual<br>Reconocimientos<br>gestuales<br>Control robótico<br>Rehabilitación médica<br>Educación<br>Identificación<br>Autenticación | Bajo costo   | Startup bien valorado                                 |
| <i>Azure Kinect</i> | <i>Microsoft</i>      | 2019 | Sistemas profesionales<br>(gestión de personas,<br>lugares, almacenes, ...)   | Alto coste<br>Gran desarrollo de IA<br>para detección de<br>personas y objetos | Buenos resultados en<br>IA y tratamiento de<br>imagen |
| <i>UltraLeap</i>    | <i>UltraLeap Inc.</i> | 2020 | Sistemas de detección<br>de posición de la mano<br>e n forma háptica.   | Caro<br>Poco software<br>desarrollado  | Cada vez se vende<br>menos                            |

Tabla 3 Resumen de todas las investigaciones vistas en este capítulo

| Referencia | Año  | Autor/es          | Tipo de dispositivo | Características  | Conclusiones                    |
|------------|------|-------------------|---------------------|--|---------------------------------|
| 20         | 2016 | Merletti y Farina | Electrodos EMG      | ---  | Uso en la rehabilitación médica |
| 21         | 1997 | Luca              | Electrodos EMG      | ---  | Uso en diagnósticos médicos     |
| 22         | 1999 | Barreto           | Electrodos EMG      | Intento de sustitución del ratón<br>Aplicar reconocimiento de gestos de forma estática | Interacción con la máquina      |
| 23         | 2001 | Coleman           | Electrodos EMG      | Intento de sustitución del ratón<br>Aplicar reconocimiento de gestos de forma estática | Interacción con la máquina      |
| 24         | 2006 | Guerreiro y Jorge | Electrodos EMG      | Intento de sustitución del ratón<br>Aplicar reconocimiento de gestos de forma estática | Interacción con la máquina      |
| 25         | 2009 | Ahsan             | Electrodos EMG      | Intento de sustitución del ratón<br>Aplicar reconocimiento de gestos de forma estática | Interacción con la máquina      |

|           |      |                           |                |                                    |  |
|-----------|------|---------------------------|----------------|------------------------------------|--|
| <b>26</b> | 2005 | Crawford                  | Electrodos EMG | Control robótico                   | Reconocimiento de gestos                     |
| <b>27</b> | 2012 | Kiguchi y Hayashi         | Electrodos EMG | Control robótico                   | Reconocimiento de gestos                     |
| <b>28</b> | 2013 | Joshi                     | Electrodos EMG | Control robótico                   | Reconocimiento de gestos                     |
| <b>29</b> | 2002 | Fistre y Tanaka           | Electrodos EMG | Reconocimiento de gestos genéricos | Controlar un reproductor de música           |
| <b>30</b> | 2003 | Wheeler y Jorgensen       | Electrodos EMG | Reconocimiento de gestos genéricos | Teclado numérico                             |
| <b>31</b> | 2009 | Dhillon                   | Electrodos EMG | Reconocimiento de gestos genéricos | Teclado numérico                             |
| <b>32</b> | 1998 | Rosenberg                 | Electrodos EMG | Reconocimiento de gestos genéricos | Puntero para una interfaz gráfica de usuario |
| <b>33</b> | 2024 | Zhang, Han, Huang y Shuai | Electrodos EMG | 6 tipos de movimientos de la mano  | 93,75 % de precisión                         |
| <b>34</b> | 2015 | Boyalı                    | Electrodos EMG | Uso del brazalete Myo              | Reconocimiento de gesto                      |



|    |      |                         |                |   |  |
|----|------|-------------------------|----------------|---|--|
| 35 | 2016 | Abreu                   | Electrodos EMG | Uso del brazalete Myo                                   | Reconocimiento de gesto  |
| 36 | 2016 | Sathiyarayanan y Rajan  | Electrodos EMG | Uso del brazalete Myo                                   | Reconocimiento de gesto  |
| 37 | 2017 | Motoche                 | Electrodos EMG | Uso del brazalete Myo                                   | Reconocimiento de gesto  |
| 38 | 2023 | Singh y Chaturvedi      | Electrodos EMG | 24 letras<br>Lenguaje de signos americano               | 99,91 % de precisión   |
| 39 | 2000 | Fléty                   | Ultrasonido    | Reconocimiento de gestos simple                         | Control de la música   |
| 40 | 2011 | Kreczmer                | Ultrasonido    | Uso de un telémetro ultrasónico convencional            | Sistema de reconocimiento de gestos, utilizando, en un robot móvil |
| 41 | 2011 | Universidad de Berkeley | Ultrasonido    | Chip con transductores ultrasónicos                     | Reconocimiento de gestos   |
| 42 | 2011 | Universidad de Berkeley | Ultrasonido    | Hardware más comprimido<br>Mismo número de sensores     | Reconocimiento de gestos   |
| 43 | 2003 | Hodges                  | Ultrasonido    | Medición de la contracción muscular de algunos músculos | ---  |

|    |      |             |             |  |  |
|----|------|-------------|-------------|--|--|
| 44 | 2013 | Mujibiya    | Ultrasonido | Conjunto de transductores ultrasónicos                 | Creación de una banda  |
| 45 | 2017 | McIntosh    | Ultrasonido | Detectar y monitorizar los posibles gestos de una mano | Creación de una sonda  |
| 46 | 2015 | Abdelnasser | WiFi        | WiGest<br>Sistema de reconocimiento de gestos          | Precisión de entre el 96 % y el 87,5 %                           |
| 47 | 2013 | Pu          | WiFi        | WiSee<br>Nueve gestos corporales                       | Interactuar con algunos electrodomésticos                        |
| 48 | 2014 | Adib        | WiFi        | Witrack  | Reconocimiento de gestos humanos, aun no teniendo visión directa |
| 49 | 2015 | Adib        | WiFi        | Witrack0   | Reconocimiento de gestos humanos, aun no teniendo visión directa |

|           |      |                           |                    |  |   |
|-----------|------|---------------------------|--------------------|--|---|
| <b>50</b> | 2014 | Nandakumar                | WiFi               | Cuatro gestos  | Precisión del 89 % en lugares de nula visión y del 91% en lugares de visión directa |
| <b>51</b> | 2016 | H. Li                     | WiFi               | <i>WiFinger</i><br>Obtención del movimiento de los dedos                                     | 93 % de Precisión   |
| <b>52</b> | 2024 | Gao, Li, Dai, Liu y Zhang | <i>WiFi</i>        | <i>WiCGesture</i>  | Reconocimiento de gestos de manera continua   |
| <b>53</b> | 2023 | Dabwan                    | Cámara RGB y RGB-d | 32 signos<br>Lenguaje de signos árabe<br>CNN   | 99 % de precisión   |
| <b>54</b> | 2023 | Aldhahri                  | Cámara RGB y RGB-d | Utilización de MobileNet como sistema basado en CNN<br>32 signos<br>Lenguaje de signos árabe | 94,96 % de precisión  |
| <b>55</b> | 2022 | Chung                     | Cámara RGB y RGB-d | 100 signos   | 98,75 %   |

|           |      |                   |                       |  |   |
|-----------|------|-------------------|-----------------------|--|---|
|           |      |                   |                       | Clasificación bidireccional con redes neuronales con memoria a corto plazo |   |
| <b>56</b> | 1998 | Trucco y Verri    | Visión Estereoscópica | ---  | Estudio de la distancia focal y la separación entre las dos cámaras   |
| <b>57</b> | 2018 | Manteca Fernández | Luz Estructurada      | ---  | Explicación de la no necesidad de tener dos cámaras, en este sistema. |
| <b>58</b> | 2009 | Mo                | Cámara ToF            | Usos de la tecnología  | Reconocer ciertas posturas al caminar, agacharte o sentarte           |
| <b>59</b> | 2011 | Raptis            | Cámara ToF            | Reconocer movimientos de baile   | 96,9 % de precisión   |
| <b>60</b> | ---  | UltraLeap         | Leap Motion           | Usos de la tecnología  | Obtener movimiento de   |

|           |      |                       |             |   |                                     |
|-----------|------|-----------------------|-------------|---|-------------------------------------|
|           |      |                       |             |   | dedos, manos y brazo en tiempo real |
| <b>61</b> | 2014 | Hantrakul y Kaczmarek | Leap Motion | Usos de la tecnología   | Música                              |
| <b>62</b> | 2016 | Ling y Rui            | Leap Motion | Usos de la tecnología   | Educación                           |
| <b>63</b> | 2015 | Chan                  | Leap Motion | Usos de la tecnología   | Identificación y autenticación      |
| <b>64</b> | 2023 | Sesli                 | Leap Motion | Usos de la tecnología   | Robots militares                    |
| <b>65</b> | 2015 | Gándara y Bauza       | Leap Motion | Usos de la tecnología   | Domótica                            |
| <b>66</b> | 2016 | Smeragliuolo          | Leap Motion | Comprobación de calidad y precisión   | Obtención de mejores resultados     |
| <b>67</b> | 2014 | Mohandes, Aliyu       | Leap Motion | Lenguaje de signos árabe<br>28 letras<br>Redes neuronales multicapa<br>Clasificador Naves Bayes | 99 % de precisión                   |
| <b>68</b> | 2015 | Y. Chen               | Leap Motion | Lenguaje de signos árabe<br>36 gestos entre números y alfabeto                                  | 80 % de precisión                   |

|           |      |           |             |  |   |
|-----------|------|-----------|-------------|--|---|
|           |      |           |             | Máquina de soporte de vectores   |   |
| <b>69</b> | 2015 | McCartney | Leap Motion | Lenguaje de signos árabe<br>100 participantes<br>Redes neuronales convolucionales<br>Modelos ocultos de Márkov | 92,4 % de precisión   |
| <b>70</b> | 2016 | Lu        | Leap Motion | Varios gestos dinámicos<br>Clasificador HCNF<br>Redes neuronales   | 89,5 % de precisión   |
| <b>71</b> | 2022 | Saad      | Leap Motion | Lenguaje de signos americano<br>30 signos<br>Clasificadores Bi-GRU<br>Clasificadores Bi-LSTM                   | 96,97 % de precisión con Bi-GRU<br>97,98 % de precisión con Bi-LSTM |
| <b>72</b> | 2022 | B. Wu     | Leap Motion | Lenguaje de signos chino<br>20 palabras dinámicas<br>LSTM modificado   | 99,55 % de precisión  |

|    |      |                          |             |  |   |
|----|------|--------------------------|-------------|--|---|
| 73 | 2022 | Abdullahi y Chamnongthai | Leap Motion | Lenguaje de signos americano<br>57 palabras<br>Fusor capturador de vectores (FFV) con Bi-LSTM  | 91 % de precisión   |
| 74 | 2022 | Nasir                    | Leap Motion | Lenguaje de signos americano<br>26 letras<br>K-NN<br>SVM   | 72,78 % de precisión con K-NN<br>79,83 % de precisión con SVM |
| 75 | 2022 | Rishan                   | Leap Motion | Lenguaje de signos de Sri Lanka<br>3 palabras y 3 letras<br>Coincidencia de plantillas geométricas y procesamiento de lenguaje natural (NLP) | 77 % de precisión   |
| 76 | 2023 | Katilmis y Karakuzu      | Leap Motion | Lenguaje de signos turco<br>26 palabras dinámicas<br>Extreme learning machine (ELM)  | 93 % de precisión   |

|           |      |                         |             |   |   |
|-----------|------|-------------------------|-------------|---|---|
|           |      |                         |             | Uso del modelo de meta-aprendizaje de ELM (Meta-ELM)  |   |
| <b>77</b> | 2024 | Bekiri, Sarra y Mohamed | Leap Motion | Lenguaje de signos americano<br>Técnica de aprendizaje por transferencia con datos aumentados<br>20000 imágenes | 99,12 % de precisión  |
| <b>78</b> | 2023 | Leth                    | VR          | 25 signos<br>Lenguaje de signos danés<br>NLP  | Precisión del 44,4 % con signos dinámicos y 41,3 % con signos estáticos |
| <b>79</b> | 2022 | Fuchs                   | UltraLeap   | Uso de gafas de realidad virtual junto a este dispositivo   | ---   |
| <b>80</b> | 2021 | Mulot                   | UltraLeap   | Uso de la tecnología  | Obtención de señales hápticas o táctiles                                |
| <b>81</b> | 2020 | Young                   | UltraLeap   | Uso de la tecnología  | Creación de interfaces para la automoción                               |



|           |      |                       |           |   |                                      |
|-----------|------|-----------------------|-----------|---|--------------------------------------|
| <b>82</b> | 2022 | Georgiou,<br>Martinez | UltraLeap | Uso de gafas de realidad virtual para observar objetos.                               | Ofrecer una sensación táctil         |
| <b>83</b> | 2020 | Paneva                | UltraLeap | Uso de la tecnología  | Reconocimiento del lenguaje braille  |
| <b>84</b> | 2018 | Abdel-Samie           | Kinect    | 30 palabras<br>Lenguaje de signos árabe   | Precisión de entre 95,25 % y 97,58 % |
| <b>85</b> | 2015 | Sarhan                | Kinect    | Lenguaje de signos árabe<br>16 palabras<br>HMM  | Precisión del 80,47 %                |
| <b>86</b> | 2015 | Sun                   | Kinect    | Lenguaje de signos americano<br>2000 frases<br>Máquina de soporte de vectores latente | 84 % de precisión                    |
| <b>87</b> | 2011 | Zafrulla              | Kinect    | 1000 signos<br>HMM  | Precisión del 76,12 %                |
| <b>88</b> | 2012 | Lang                  | Kinect    | Lenguaje de signos alemán   | 97 % de precisión                    |
| <b>89</b> | 2015 | Pigou                 | Kinect    | Lenguaje de signos italiano<br>20 signos<br>CNN                                       | 91,7 % de precisión                  |

|           |      |               |              |                                       |                              |
|-----------|------|---------------|--------------|---------------------------------------|------------------------------|
| <b>90</b> | 2015 | Mehrotra      | Kinect       | 37 signos<br>Lenguaje de signos indio | 86,16 % de<br>precisión      |
| <b>91</b> | 2019 | Microsoft     | Azure Kinect | Uso en aplicaciones<br>profesionales  | Creación del<br>Azure Kinect |
| <b>92</b> | 2022 | Brock y Gomez | Azure Kinect | 18 gestos CNN                         | 41 % de precisión            |

## 3. Materiales y métodos

### 3.1. Introducción.

En este capítulo se desarrollará los diferentes bloques que conforman el proyecto base, así como los materiales, que son usados y el procedimiento que se realiza para obtener la traducción del lenguaje de signo de español a voz, de manera exitosa. Primeramente, se explicará el funcionamiento del *Leap Motion*. A continuación, se explicará el proceso de adquisición de datos, así como posibles puntos de mejora de ese bloque. Luego, se mostrará la base de datos, con las palabras usadas en el proyecto base y las palabras usadas para este proyecto. Por último, se mostrará el proceso de generación de patrones, explicando los parámetros existentes y el comparador del sistema, además, se establecerá la técnica DTW y se mostrará un esquema de dicho comparador.

### 3.2. Adquisición.

#### 3.2.1. Sensor *Leap Motion*.

En el capítulo anterior, se ha realizado un análisis de las diferentes tecnologías, que se han utilizado para el reconocimiento de gesto y, se ha llegado a la conclusión de que el Leap Motion es una de las mejores opciones, debido a su bajo precio, a su versatilidad y el tamaño del dispositivo, permitiendo capturar, con gran precisión, los movimientos de los brazos, manos y dedos del cuerpo en 3D. Para conseguir dicha precisión, este dispositivo cuenta con tres LED y una cámara, ambos infrarrojos, con un ángulo de captura de aproximadamente 150 °, trabajando, como se puede intuir, en el rango del espectro no visible, más exactamente, en la longitud de onda de 850 nm. Otras de las ventajas que posee el Leap Motion, es su adaptabilidad a la luz del medio, obteniendo por ello, una alta resolución de imagen. Además, este tiene la posibilidad de alcanzar una tasa de captura de hasta 200 fotogramas por segundo, lo que permite un control muy detallado y preciso, punto que será de estudio en este trabajo.



*Ilustración 3 Hardware de Leap Motion*

Como se ha mencionado anteriormente, el rango de visión de este dispositivo alcanza los 150 °, permitiendo obtener una cobertura amplia del área de trabajo, funcionando efectivamente dentro de un rango aproximado de entre 25 a 600 mm, por encima del dispositivo.

Dada su alta compatibilidad con macOS y Windows y, a la posibilidad de obtener un *Software Development Kits*, con la compra del dispositivo, permitiendo crear programas en varios lenguajes de programación, tales como, C++, C#, *JavaScript*, Python, entre otros, a lo largo de los años, se han ido desarrollando e investigando cientos de proyectos, utilizando este dispositivo, en campos como la Realidad Virtual y Aumentada, los juegos y ocio y las interacciones con usuario (arte, diseño, etc.)

Por otro lado, para usar este dispositivo, únicamente se requiere de un ordenador y un cable USB, facilitando su instalación y uso.

En cuanto al rendimiento, el *Leap Motion* se caracteriza por su baja latencia, siendo capaz de ofrecer respuestas rápidas, y un procesamiento de movimiento capaz de rastrearlos y procesarlos en tiempo real, proporcionando una interacción fluida.

### **3.2.2. Proceso de Adquisición.**

Para la obtención de las muestras de las palabras, inicialmente, los tres Leds, del dispositivo, emiten luz infrarroja hacia el área de interacción (donde se encuentran las manos signando), esta luz emitida, chocará y se reflejará contra las manos y los dedos del usuario y, las dos cámaras infrarrojas del Leap Motion capturarán las imágenes de dicha luz reflejada. Esto se realiza a una gran velocidad, lo que permite un rastreo en tiempo real, con una precisión muy alta. Una vez se obtenga las imágenes, el software del Leap Motion las procesará, en búsqueda de las posiciones y movimientos de los dedos y las manos. El dispositivo, utilizará un análisis de patrones y un algoritmo de visión, para convertir las imágenes en datos tridimensionales. Estos datos tienen la característica de estar divididos por columnas, más precisamente por los diferentes parámetros, establecidos por el dispositivo. Algunos de los parámetros son la posición de los dedos, las manos y brazo, la orientación de la palma, el *frame* exacto de la muestra, entre otros. Finalmente, se tiene que guardar todos estos datos, de la muestra, en una base de datos, creando un diccionario gestual.

Con cada palabra grabada, se ha obtenido un archivo de texto (signos dinámicos), establecidos y distribuidos en los 276 parámetros de sistema.

En el proyecto base, se obtuvo una base de datos total de 5780 muestras, cada una, haciendo referencia a un archivo, a una palabra signada.

### 3.3. Base de Datos.

Analizando la base de datos del proyecto base, se observa que esta, de las 5780 muestras, 3520 las han usado para entrenamiento, y 2260 para verificar. Por otra parte, también se observa que no hay ninguna preposición y los verbos se encuentran en infinitivo. Esto es debido, a que se está utilizando el lenguaje de signos español en este proyecto.

#### 3.3.1. Conjunto de Palabras Usadas en el proyecto base.

Como se mencionó en el capítulo uno, para que el proyecto tuviera mayor practicidad, las palabras grabadas se relacionarían con un caso real. En ese caso, se eligió un contexto médico, en donde, por ejemplo, una persona con discapacidad auditiva acudiría al médico y esta tendría que comunicarse con el doctor. Las palabras escogidas fueron aquellas relacionadas con la salud, además, se añadieron palabras y verbos de uso cotidiano, para una conversación más natural. La selección total fue de 176 palabras. Para evitar problemas de dificultad, en el manejo de la base de datos, se decidió el comenzar con 50 palabras, para posteriormente, aumentar a 100 y 176, respectivamente.

| Verbos                    |           |           |             |            |               |          |
|---------------------------|-----------|-----------|-------------|------------|---------------|----------|
| Caer                      | Caminar   | Cocinar   | Comer       | Consultar  | Correr        | Curar    |
| Desaparecer               | Descansar | Duchar    | Engordar    | Es         | Hacer         | Intentar |
| Ir                        | Llevar    | Medir     | Operar      | Pasear     | Preguntar     | Quemar   |
| Reducir                   | Romper    | Sentir    | Sufrir      | Tener      |               |          |
| Palabras médicas          |           |           |             |            |               |          |
| Accidente                 | Alergia   | Alzheimer | Ambulancia  | Análisis   | Ansiedad      | Asma     |
| Audífono                  | Bacterias | Bata      | Cabeza      | sangre     | Circulación   | Corazón  |
| Cuello                    | Cuidado   | Depresión | Desmayo     | Cáncer     | Eructo        | Espalda  |
| Estómago                  | Estrés    | Fiebre    | Flema       | Diabetes   | Gordo         | Guantes  |
| Hemorragia                | Herida    | Hígado    | Hinchazón   | Glúteo     | Hormona       | Hospital |
| Ictus                     | Implante  | Infarto   | Inflamación | Hombro     | Inyección     | Lesión   |
| Lumbago                   | Mandíbula | Manos     | Mareo       | Intestino  | Medicamentos  | Médico   |
| Molesto                   | Ojos      | Oreja     | Órganos     | Mascarilla | Oxígeno       | Pánico   |
| Párpado                   | Pecho     | Pies      | Próstata    | Ovarios    | Puntos sutura | Salud    |
| Sangre                    | Seguro    | Senos     | Sensación   | Pulmones   | Suero         | Tensión  |
| Urgencia                  | Vacuna    | Vagina    | Virus       | Sordo      | Vómito        |          |
|                           |           | Vejiga    |             |            |               |          |
| Palabras de uso cotidiano |           |           |             |            |               |          |

|               |             |          |             |             |            |          |
|---------------|-------------|----------|-------------|-------------|------------|----------|
| Abril         | Además      | Adiós    | Agosto      | Ahora       |            | Algo     |
| Alta          | Anoche      | Antes    | Ayer        | Bien        | Aire       | Buenas   |
| Buenas tardes | Buenos días | Centro   | Coche       | Codo        | Boca       | noches   |
| De nada       | Después     | Día      | Diciembre   | Ejemplo     | De acuerdo | Delgado  |
| Familia       | Febrero     | Fin      | Forma       | Frio        | Ellos      | Esfuerzo |
| Hambre        | Hasta       | Hombre   | Hora        | Información | Funciona   | Gracias  |
| Mamá          | Marzo No    | Mayor    | Mediodía    | Miércoles   | Junio      | Lluvia   |
| Nada          | Privado     | Noche    | No lo sabía | No sé       | Moto       | Mucho    |
| Poco          | Sábado      | Problema | Qué tal     | Raro        | Octubre    | Peligro  |
| Ruido         | Suyo        | Seguro   | Septiembre  | Sexo        | Regular    | Riesgo   |
| Solo          |             | También  | Viernes     | Ya          | Siempre    | Sin      |

Tabla 4 176 palabras usadas en la investigación del proyecto base

### 3.3.2. Conjunto de Palabras Usadas en este proyecto.

A diferencia del proyecto base, que usa 176 palabras, en el punto final del proyecto, en este se usarán solo 131 palabras, ya que lo que se busca es la mejora de la precisión del sistema y el tiempo de ejecución de este, con una cantidad media de palabras. Además, se realizará la misma acción que en el proyecto base, en donde, del total de palabras se cogerá una parte, se creará el código y se probará, para, luego, aumentar el número de palabras y seguir mejorando el código y realizar nuevas experimentaciones. Por tanto, se creará un código escalable con las 131 palabras que podrá ser probado fácilmente con la totalidad de las palabras del proyecto.

En el final del proyecto, la precisión obtenida, en este, se comparará con la obtenida con 100 y 176 palabras, en el proyecto base.

| <b>Verbos</b>   |
|---|
| Beber, Comer, Curar, Descansar, Dormir, Hacer, Intentar, Ir, Llevar, Medir, Pasear, Quemar,<br>Reducir, Tener, Tomar  |
| <b>Palabras médicas</b>   |
| Alergia, Alzheimer, Ambulancia, Ansiedad, Análisis sangre, Asma, Bacterias, Bata, Cabeza, Cerebro, Corazón, Cuello, Cáncer, Cáncer próstata, Depresión, Desmayo, Diabetes, Dientes, Dolor, Enfermera, Embarazo, Eructo, Espalda, Estrés, Estómago, Esguince, Fiebre, Flema, Gases, Garganta, Gripe, Guantes, Hemofilia, Hemorragia, Herida, Hombro, Hospital, Hígado, Hipo, Ictus, Infarto, Inflamación, Intestino, Inyección, Lumbago, Mareos, Mascarilla, Medicamentos, Médico, Materno, Migraña, Moco, Nauseas, Nariz, Ojos, Oreja, Órganos, |

|  |
|--|
| Ovarios, Oxígeno, Páncreas, Pánico, Parkinson, Pecho, Pelo, Pene, Pies, Próstata, Pulmones, Riñón, Regla Salud, Senos, SIDA, Suero, Tos, Urgencia, Vacuna, Vejiga, Virus, Vómito   |
| <b>Palabras de uso cotidiano</b>   |
| Adiós, Agosto, Aire, Anoche, Año, Ayer, Bien, Boca, Buenas noches, Buenas tardes, Buenos días, Centro, Codo, De nada, Diciembre, Día, Gracias, Hola, Hora, Hoy, Junio, Julio, Mañana, Mañana (Parte día), Mal, Muñeca, Minuto, Mucho, Noche, Octubre, Privado, Qué tal, Raro, Regular, Riesgo, Ruido, Septiembre, Tarde, Una |

Tabla 5 131 palabras usadas en el proyecto

### 3.4. Generación de patrones

#### 3.4.1. Parámetros

El filtrado de parámetros ya se realizaba desde el proyecto base. Este, reducía los 276 parámetros, generados por el *Leap Motion*, a solo 74 parámetros, con el objetivo de obtener los parámetros fundamentales para trabajar con las muestras y, eliminar aquellos que no sean fundamentales o que contienen información adyacente.

Por otra parte, para poder procesar y comparar correctamente las muestras, se debe separar los datos provenientes de la mano derecha, de los datos de la mano izquierda, con el objetivo de reducir la complejidad de sistema.

En este proyecto, se abarcará la posibilidad de cambiar algunos parámetros, para mejorar la precisión y el tiempo de ejecución.

#### 3.4.2. Generación

En este proyecto, se mantendrá el procedimiento seguido para la generación de los patrones. Lo que variará en este caso, será los parámetros que conformarán al patrón y la frecuencia de muestreo de este. Ambas partes, se abordarán en el capítulo 4.

Para generar los patrones, antes se debe ecualizar el tamaño de las muestras usadas, para obtener unas muestras del mismo tamaño, con el mismo número de fotogramas. Esto es debido a que, a la hora de realizar las grabaciones, se puede dar el caso que, para una palabra signada, se realicen varias grabaciones de la misma palabra, con personas distintas, en situaciones distintas y, se originen muestras, de la misma palabra, pero de distinta duración.

Una vez arreglado este problema, se estableció que, para la creación de los patrones, se seguiría la siguiente fórmula:

$$\frac{\sum_{i=1}^{i=20} Q_i(x_n)}{20}$$

Con esta fórmula, se asegura de que el máximo de característica del signo se vería reflejadas en el patrón.

### **3.5. Comparador.**

#### **3.5.1. Desarrollo del *Dynamic Time Warping*, DTW.**

El proyecto base, hace uso del Dynamic Time Warping (DTW), un algoritmo usado para obtener la similitud entre dos series temporales, pudiendo variar estas, de velocidad y duración. No obstante, como se ha mencionado antes, se ha intentado ajustar las muestras, para que, todas ellas tengan una duración similar y, sea más fácil su comparación con los patrones.

Este algoritmo, al usarlo nos devuelve una distancia, esta es una medida del parentesco entre dos series temporales, en este caso, la muestra de entrada y el patrón. DTW obtiene la mínima distancia posible.

En el proyecto base, se hace uso de la librería *DTW* de *Python*, para realizar esta función. En este proyecto, se buscarán posibles alternativas a este algoritmo, buscando algoritmos *DTW* más preciso y/o más rápido.

#### **3.5.2. Esquema del comparador.**

En el comparador, 10 muestras de cada palabra se compararían con los patrones, con el objetivo de obtener la distancia más corta, dependiendo de la entrada.



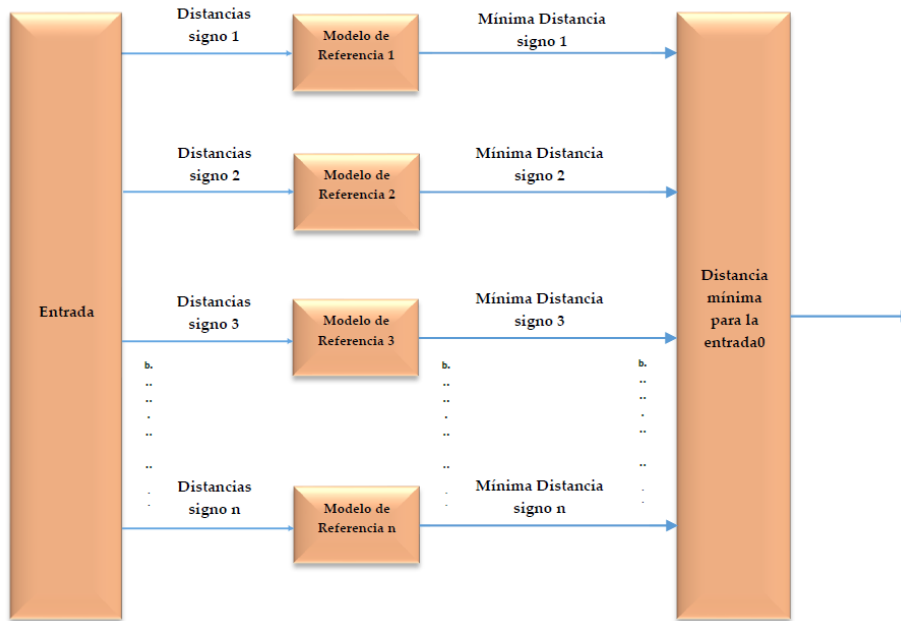


Ilustración 4 Esquema del comparador

Los patrones son la parte del "Modelo de referencia". En este esquema, en la entrada, aparece la muestra de una palabra, la cual, pasará por todos los "Modelo de referencia" y, se obtendrá tantas distancias, como "Modelo de referencia" existentes. Todas estas distancias, se agruparán y se compararán para encontrar la distancia mínima, cuál de los patrones se parece más a la muestra de entrada.



## 4. Metodología experimental

### 4.1. Introducción.

En este capítulo, se mostrará los procedimientos de análisis y mejora de las diferentes partes del proyecto base. Estas mejoras son centradas únicamente en mejora la velocidad, la precisión y el código base. Se comenzará con el análisis del estado actual de la frecuencia de muestreo y la posible técnica a usar para optimizar dicha frecuencia. Posteriormente, se analizará los parámetros, del *Leap Motion*, seleccionados, en el proyecto base, y se plantearán posibles mejoras de esta selección, eliminando o añadiendo algún parámetro. Todo ello será explicado el motivo de cambio. Por último, se analizará el algoritmo de DTW y se comprobará posibles técnicas parecidas, en busca de alguna que mejore el bloque de comparación.

#### 4.1.1. Creación de código.

Para la creación de la primera parte de código, se ha asignado una pequeña base de datos de entrenamiento, conformada por las tres primeras palabras de la base de datos de la sesión 1, esto, con el objetivo de obtener un código escalable e independiente, con la capacidad de ser ejecutado, no solo con las 131 palabras de este proyecto, sino con un número de palabras indefinido. Con esta base de datos, se ha diseñado y creado el código de extracción de datos de las muestras, el código para el separado de las muestras de cada mano, parte del código de la optimización de la frecuencia de muestreo y el código de obtención y filtrado de parámetros. Posteriormente, se ha testado dichos códigos con la base de datos de testeo, conformada por el resto de las palabras de la base de datos de la sesión 1.

Para la segunda parte, la creación del resto del código de la optimización de la frecuencia de muestreo y el código del sistema de comparación DTW, se ha utilizado como base de datos de entrenamiento, la base de datos de la sesión 1 y la sesión 2 y, para el testeo del código, se ha usado la base de datos de testeo, conformada por las sesiones del 3 al 10.

#### 4.1.2. Métricas de calidad.

Todo este proyecto, se sustenta con el objetivo de mejora de la precisión y el tiempo de ejecución, por este motivo, para obtener la precisión del sistema, en cada parte del proyecto, se hará uso de las matrices de confusión.

|              |          | Predicted Class                            |  |  |
|--------------|----------|--|--|--|
|              |          | Positive                                   | Negative   |  |
| Actual Class | Positive | True Positive (TP)                         | False Negative (FN)<br>Type II Error                       | <b>Sensitivity</b><br>$\frac{TP}{(TP + FN)}$             |
|              | Negative | False Positive (FP)<br>Type I Error        | True Negative (TN)   | <b>Specificity</b><br>$\frac{TN}{(TN + FP)}$             |
|              |          | <b>Precision</b><br>$\frac{TP}{(TP + FP)}$ | <b>Negative Predictive Value</b><br>$\frac{TN}{(TN + FN)}$ | <b>Accuracy</b><br>$\frac{TP + TN}{(TP + TN + FP + FN)}$ |

Ilustración 5 Matriz de confusión

En este sistema se establecen los aciertos y fallos. Los aciertos son aquellos, los cuales, al entrar una palabra cualquiera al sistema, este lo detecte, lo analice, con el comparador, y llegue a la conclusión de que es esa palabra y no otra. Los fallos, es el caso contrario, en donde, entre la palabra al sistema y este llegue a la conclusión de que es otra palabra. Un posible ejemplo es que entre, al sistema, la palabra signada “hola”, y este llegue a la errónea conclusión de que se trata de la palabra signada “adiós”.

Para obtener los aciertos y los fallos, se hará pasa al sistema las 131 palabras de una sesión, una a una, y se utilizará como elemento comparador, otra sesión distinta, inicialmente, para posteriormente, utilizar los patrones creados. Esto generará acierto y fallos, los cuales, se irán registrando, hasta obtenerlos todos. Cuando se termine el proceso, se calculará la precisión con la siguiente fórmula:

$$\text{Precisión} = \frac{\text{Aciertos}}{\text{Aciertos} + \text{Fallos}}$$

## 4.2. Optimización de la frecuencia de muestreo.

Como ya se mencionó en el capítulo anterior, la frecuencia de muestreo ofrecida por el *Leap Motion* rondaba los 110 fotogramas por segundo (*fps*), algo que resulta innecesario debido a los movimientos de las manos, a la hora de realizar los gestos. Estos movimientos pueden variar dependiendo de la persona que lo realice, sin embargo, en una media, el movimiento es relativamente calmado, con posibles incrementos de velocidades en los giros, pero en general tranquilo. Por este motivo, al grabar la muestra, esta contendrá sobreinformación (momentos en el que las manos estén paradas), sumado a que se originan dos muestras con la misma frecuencia, una muestra para cada mano, esto aumenta significativamente los datos innecesarios.

El arreglo que propone y establece el proyecto base es la instauración de la técnica del diezmado, la cual, se caracteriza por la eliminación de una cantidad valores, cada cierto rango.

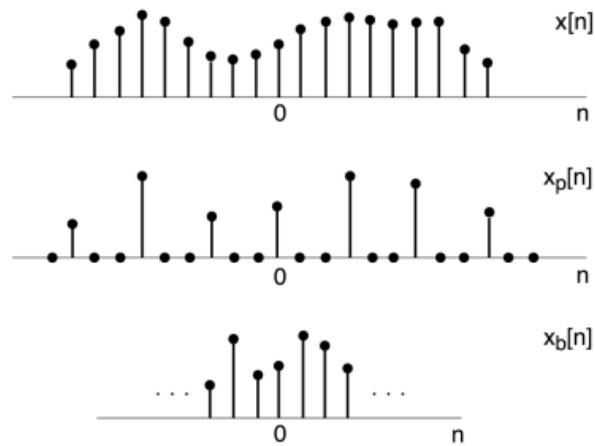


Ilustración 6 Proceso de diezmado.

No obstante, esta técnica puede resultar peligrosa debido a la posibilidad de eliminar datos muy importantes, tales como, un giro de la mano, alguna posición importante, pudiendo originar una muestra corrupta, que no represente el signo correcto.

Para ello, la solución dada en este proyecto es la implementación de la técnica de la media móvil centrado (CMA), la cual consiste en promediar un número igual de datos a ambos lados de un punto central. Significa que, para calcular el valor de la media móvil centrado en un punto establecido, se tomara en cuenta los datos de los periodos anteriores y posteriores al punto, más específicamente, promediar los datos dentro de una ventana de tamaño  $n$ , siendo este a especificar por la investigación.

Esta técnica se usará de la siguiente manera:

Para una serie de tiempo  $y_1, y_2, \dots, y_n$ , el valor de la media móvil centrado se calcula como:

$$CMA_t = \frac{1}{2k + 1} \sum_{i=-k}^k y_{t+i}$$

- $t$  punto de la serie de tiempo
- $k$  número de datos antes y después del punto central.

Con esto en mente, se podrá tomar la muestra, la cual posee una frecuencia de muestreo de 110 *fps* y aplicar dicha técnica en toda la muestra, reduciendo los datos contenidos en esa muestra para que, posteriormente, se reduzca considerablemente el tiempo de procesado de la muestra. Cabe destacar que antes de realizar dicho proceso, si la muestra contiene las series de tiempo de ambas manos, se deberán separar en muestra por separado y aplicar esta técnica.

Sin embargo, aun siendo esta técnica muy útil, no está exenta de peligro ya que, si se seleccionase un tamaño de la ventana  $n$  muy grande, esta podría provocar el mismo efecto que el diezmado o peor y, si se seleccionase una  $n$  muy pequeña, esta no ofrecería ninguna mejora y originaría una pérdida de tiempo innecesaria. Por este motivo, se deberá ir probando diferentes valores de  $n$  e ir comprobando la precisión que ofrece y el tiempo de ejecución de comparación. Para realizar las comparaciones, se cogerán las sesiones 1 y 2, ya mostradas en el capítulo anterior y se le realizará dicho cambio, se comprobará el tiempo de ejecución y el porcentaje de precisión, buscando un punto óptimo donde se reduzca el tiempo de ejecución sin reducir enormemente la precisión. Estos resultados se mostrarán en el siguiente capítulo.

Otro de los posibles problemas surge de la posibilidad de que, ciertos datos de la muestra estén conectados entre sí, por ejemplo, el giro de la mano, ciertos movimientos continuos de los dedos, etc. Esta situación provoca que, a la hora de realizar el CMA, puede darse el caso que algunas ventanas se establezcan cortando esos datos conectados, estableciendo cada parte en su respectiva ventana y al calcular el promedio, no ofrezca un resultado óptimo. Es por eso por lo que, para dar solución a este problema, se ha llegado a la conclusión de que, se deberá establecer un solapamiento entre ventana. Por lo que el tamaño de la ventana será el original más el solapamiento anterior y posterior. Estos solapamientos serán un porcentaje del tamaño de la ventana, comenzando con un 25 % del tamaño de la ventana y se irá variando.

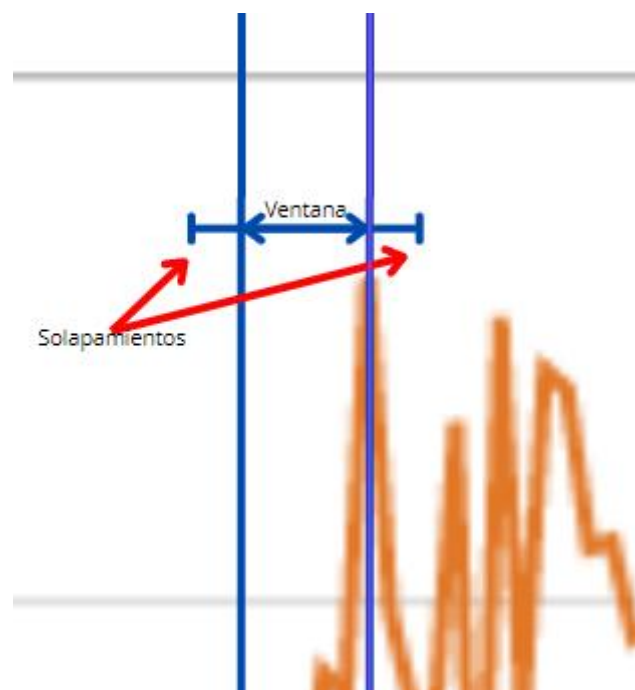


Ilustración 7 Ventanas y Solapamientos

Al realizar medidas, añadiendo el solapamiento, se puede observar que la precisión variará de su estado inicial (en el momento de establecer  $n$ ), debido a esto, se deberá volver a realizar las experimentaciones variando el tamaño  $n$  de la ventana y el solapamiento, hasta obtener el punto óptimo deseado. Para este momento, se cogerán el resto de las sesiones y se obtendrá los tiempos de ejecución y la precisión, para comprobar cómo va variando respectivamente, se obtendrá el enfrentamiento que obtenga la menor precisión y se irá trabajando desde ese punto, una vez obtenido el punto óptimo, se volverá a comprobar con el resto de las sesiones, este proceso se repetirá hasta obtener el resultado deseado. El resultado es mostrado en el siguiente capítulo.

### **4.3. Cambio de parámetros base seleccionados.**

Para este punto, se establecerá como base los parámetros elegidos en el proyecto base y se irán realizando cambios y se comprobará la precisión y el tiempo de ejecución obtenidos en la comparación entre sesiones. Las comparaciones se realizarán entre sesiones al azar para obtener unos resultados imparciales en el proceso. Todos los resultados de los cambios de los parámetros realizados se mostrarán y analizarán en el siguiente capítulo.

El primer cambio que se realizará es la eliminación del parámetro *Palm.position*, debido a la posible variación en la posición de la cámara y en la posición inicial y final de las palmas, se podría originar datos erróneos que afecten negativamente al análisis. Por este motivo, se ha decidido su omisión en los parámetros a usar.

Otro de los cambios que se han pensado es la eliminación del, no solo del parámetro *Palm.posición*, sino también del *Palm.orientation.w*, debido al mismo razonamiento que en el caso anterior.

Manteniendo los cambios anteriores, en el tercer cambio, se ha pensado en añadir el parámetro *palm.normal*, esta idea es debida a que, un posible punto de vista es que no se tenga en cuenta la posición de las palmas, si no su normal. Aunque, con esta idea puede que no se obtengan resultados satisfactorios por lo explicado anteriormente sobre la posición inicial de la mano.

Como no se ha llegado a un punto positivo, se ha decidido el excluir el último cambio y añadir todos los parámetros de *bone.orientation.w*, como método de comprobación ya que, estos no son incluidos como parámetros en el proyecto base y se quería comprobar el impacto de estos en la precisión. Comprobados estos, se ha comprobado una caída de precisión y por tanto se ha excluido dicha idea.

Por otra parte, otra idea es añadir *palm.direction* y comprobar los resultados. Sin embargo, al no haber cambios aparentes y, manteniendo la idea de que la dirección de la palma puede variar en todos los inicios y finales de grabación, se ha descartado la idea.

Por último, para mantener la misma idea de *Palm.orientation* y *bone.orientation* se ha decidido probar una vez más, pero en este caso, con *arm.orientation*. No obstante, como ha pasado con los anteriores, no se ha producido mejora, por tanto, se descarta esta idea.

Una vez, realizado los posibles cambios más prometedores, se realiza una comparación con todas las sesiones para obtener las precisiones de esta.

Cabe destacar que estos cambios de parámetros se deberán realizar también en los patrones, junto a lo establecido en el punto anterior, teniendo que volver a crear dichos patrones, con las indicaciones dadas del proyecto base.

#### **4.4. Implementación de mejora en el algoritmo DTW.**

Se evitará la situación de crear desde cero un algoritmo DTW, se utilizarán los ofrecidos por la librería de Python y el ofrecido por terceros.

En cuanto al ofrecido por terceros, el algoritmo UCRDTW, este se encuentra programado en C++, se ha intentado implementar en el programa de *Python*, con resultados nefastos, Sin embargo, en las librerías de *Python*, también se ha podido encontrar este algoritmo, pero con ciertos cambios que, al probarlo, realizando una comparación entre dos sesiones al azar, ha ofrecido una precisión demasiado baja de lo deseado, por este motivo, se ha optado por el retiro de este. No obstante, en el futuro, se volverá a examinar este algoritmo en búsqueda de algún arreglo de este, con el objetivo de volver a probarlo.

Por otra parte, *Python* también ofrece tres librerías con este algoritmo: *DTW*, *FastDTW* y la librería *Tslearn*. La primera es usada por el proyecto base, así que se ha decidido probar los tres para comprobar cuál de los tres ofrece una mayor precisión con un menor tiempo de ejecución. Los resultados son mostrados en el siguiente capítulo.

Comprobado los resultados, la mejor opción es el *FastDTW* ya que ofrece una precisión un poco mayor al *DTW*, con un menor tiempo de ejecución.

Seleccionado el algoritmo de comparación, se realizan los cambios ya mencionados en la frecuencia de muestreo y los parámetros y se crean los nuevos patrones, con las indicaciones dadas, se comparará, en este caso, no las sesiones consigo mismo, sino que, se comparará las sesiones con los nuevos patrones creados, obteniendo la precisión media y tiempo de ejecución medio del sistema. Esto se lleva a cabo, realizando las comparaciones de cada sesión con cada patrón, obteniendo en el proceso, varios resultados y, se calculará la media y la desviación de los resultados obtenidos. Estos serán mostrados en el siguiente capítulo.



## 5. Resultados

### 5.1. Introducción.

En este capítulo, se mostrarán los diferentes resultados de los experimentos realizados en el capítulo anterior. Cabe destacar que estos serán divididos en partes, según el bloque en el que se han realizado los experimentos.

Primeramente, se deberá establecer el punto de partida, ya mostrado anteriormente, que será los valores de la precisión obtenidos en el proyecto base, para posteriormente, compararlos con los resultados obtenidos en este proyecto.

Analizando el proyecto base, este obtiene una precisión que varía respecto al número de palabras usadas en la base de datos. Como es lógico, la precisión va disminuyendo cuando el número de palabras van aumentando.

| Número de palabras | Precisión      |
|--------------------|----------------|
| 50                 | <b>96,42 %</b> |
| 100                | <b>91,99 %</b> |
| 176                | <b>91,61 %</b> |

Tabla 6 Resultados de precisión del proyecto base

Para este proyecto, como ya se ha indicado en el capítulo 3, se deberá comparar los resultados con el resultado obtenido, del proyecto base, ente 100 palabras y 176 palabras.

Inicialmente, sin haber realizado ninguna modificación al proyecto base, los resultados obtenidos en las comparaciones de la sesión 1 con el resto de las sesiones son los siguientes:

| Sesión | Precisión      |
|--------|----------------|
| 2      | 90,84 %        |
| 3      | <b>88,55 %</b> |
| 4      | 92,37 %        |
| 5      | <b>82,37 %</b> |
| 6      | 83,97 %        |
| 7      | <b>87,68 %</b> |
| 8      | 83,21 %        |
| 9      | <b>87,02 %</b> |
| 10     | <b>82,44 %</b> |

Tabla 7 Precisiones individuales sin la aplicación de ninguna técnica, Sesión 1 vs Resto

Con estos resultados se obtiene que:

|       | Precisión | Desviación |
|-------|-----------|------------|
| Media | 86,50 %   | ± 3,49     |

Tabla 8 Precisión media con su desviación

Como se puede comprobar, el resultado inicial, obtenido en este proyecto, es más bajo al resultado final del proyecto base. Esto es debido a que aún no se ha aplicado ninguna técnica.

## 5.2. Optimización de la frecuencia de muestreo.

Se comenzará con los resultados obtenidos en la selección del n tamaño de la ventana para la media.

- Sesión utilizada para comparar: S1 vs S2.
- Tiempo inicial requerido (sin la aplicación de la media): 17 seg.
- Porcentaje inicial de acierto de esta comparación: 90,84 %.
- Sin solapamiento.

| N  | % de acierto | Tiempo medio por búsqueda (seg.) |
|----|--------------|----------------------------------|
| 1  | 90,89        | 17.00                            |
| 2  | 92,37        | 16.30                            |
| 3  | 92,37        | 15.40                            |
| 4  | 92,37        | 13.80                            |
| 5  | 93,89        | 13.45                            |
| 7  | 93,13        | 11.36                            |
| 10 | 46,95        | 10.13                            |
| 15 | 31,30        | 9.33                             |
| 20 | 23,47        | 8.76                             |
| 25 | 18,78        | 8.33                             |
| 30 | 15,65        | 8.00                             |
| 35 | 13,41        | 7.74                             |
| 40 | 11,74        | 7.50                             |
| 45 | 10,43        | 7.43                             |
| 50 | 9,39         | 7.32                             |
| 55 | 8,54         | 7.13                             |
| 60 | 7,82         | 6.97                             |

Tabla 9 Precisión y tiempo medio por búsqueda, estableciendo tamaño de ventana

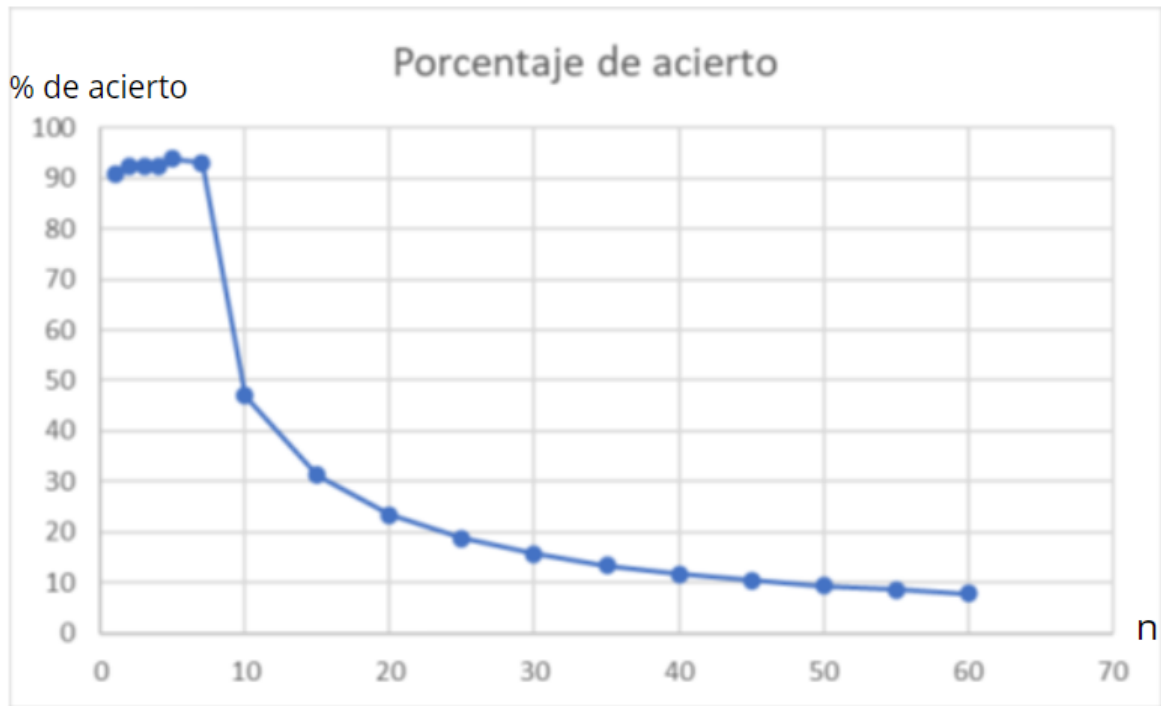


Ilustración 8 Representación de los porcentajes de acierto de la tabla

Observando esta gráfica, se llega a la conclusión de que se sigue una distribución gaussiana, centrada en el  $N = 5$ . Sin embargo, el tiempo que se aprecia es muy grande al buscado, por tanto, se intentará aplicar un 0.25 de solapamiento inicial para comprobar las modificaciones que ocurran.

| N   | Precisión (%) | Tiempo (seg.) |
|-----|---------------|---------------|
| 25  | 91,60         | 6,50          |
| 50  | 92,37         | 5,70          |
| 75  | 90,84         | 5,20          |
| 80  | 92,37         | 5,10          |
| 85  | 91,60         | 5,12          |
| 90  | 90,84         | 5,08          |
| 100 | 91,60         | 5,10          |

Tabla 10 Porcentajes de presiones con sus tiempos con 0.25 de solapamiento.

Teniendo en cuenta estos resultados, se ha centrado en un estudio más especificado en  $N = 80$ , realizando el mismo intento, pero en este caso, realizando una comparativa con las demás sesiones.

| Sesión         | % de acierto | Tiempo medio por búsqueda (seg.) |
|----------------|--------------|----------------------------------|
| <b>1 vs 2</b>  | 92,37        | 5,10                             |
| <b>1 vs 3</b>  | 83,97        | 5,20                             |
| <b>1 vs 4</b>  | 87,02        | 5,16                             |
| <b>1 vs 5</b>  | 76,34        | 5,18                             |
| <b>1 vs 6</b>  | 76,34        | 5,20                             |
| <b>1 vs 7</b>  | 75,58        | 5,15                             |
| <b>1 vs 8</b>  | 74,04        | 5,10                             |
| <b>1 vs 9</b>  | 74,04        | 5,23                             |
| <b>1 vs 10</b> | 71,76        | 5,16                             |

Tabla 11 Precisión de comparativas de sesión con el tamaño de ventana y solapamiento establecido.

Cabe destacar que los tiempos obtenidos pueden variar mínimamente, debido a los subprocesos del ordenador.

Analizando estos porcentajes con los iniciales, se comprueba un aumento en la sesión 1 vs 2, pero una caída en el resto. Por este motivo, se ha seleccionado la sesión con menor porcentaje en este caso (1 vs 10), y se ha repetido el mismo proceso de selección de N que en el caso anterior.

| N          | Precisión (%) | Tiempo (seg.) |
|------------|---------------|---------------|
| <b>5</b>   | 64,12         | 14,97         |
| <b>10</b>  | 77,10         | 9,68          |
| <b>15</b>  | 70,49         | 7,77          |
| <b>25</b>  | 76,34         | 6,63          |
| <b>30</b>  | 75,58         | 6,27          |
| <b>50</b>  | 72,52         | 5,53          |
| <b>100</b> | 69,47         | 5,10          |

Tabla 12 Porcentajes de precisión y tiempo respecto a n.

Con estos resultados y los anteriores, se puede concluir con un posible candidato, teniendo en cuenta, la relación porcentaje-tiempo, puede ser N = 25. Por tanto, se aplicará el mismo proceso de análisis que con N = 80.

| Sesión  | % de acierto | Tiempo medio por búsqueda (seg.) |
|---------|--------------|----------------------------------|
| 1 vs 2  | 91,60        | 6,68                             |
| 1 vs 3  | 85,50        | 6,72                             |
| 1 vs 4  | 86,26        | 6,73                             |
| 1 vs 5  | 78,63        | 6,71                             |
| 1 vs 6  | 77,86        | 6,67                             |
| 1 vs 7  | 78,58        | 6,60                             |
| 1 vs 8  | 77,10        | 6,60                             |
| 1 vs 9  | 78,63        | 6,75                             |
| 1 vs 10 | 76,34        | 6,67                             |

Tabla 13 Nuevos % de aciertos y tiempo con N = 25.

Se observa un aumento de los valores, en comparación a N = 80, a excepción de sesión 1 vs sesión 4, que se produce una caída mínima. Sin embargo, aún sigue con porcentaje menores a la inicial, pero con un tiempo de procesamiento mucho menor (17 segundo inicialmente vs 6,70 segundos finalmente). El porcentaje se aumentará al realizarle los cambios de parámetros iniciales.

### 5.3. Cambio de parámetros base.

En este punto, se pensó posibles cambios de parámetros, originando los siguientes resultados:

| Sesión  | % de acierto | Tiempo medio por búsqueda (seg.) |
|---------|--------------|----------------------------------|
| 1 vs 2  | 94,66        | 7,12                             |
| 1 vs 5  | 86,26        | 7,12                             |
| 1 vs 6  | 90,84        | 6,48                             |
| 1 vs 8  | 84,73        | 7,09                             |
| 1 vs 10 | 87,02        | 6,50                             |

Tabla 14 % de acierto y tiempo con el cambio 1.

Se observa una gran mejoría, respecto al punto inicial.

| Sesión | % de acierto | Tiempo medio por búsqueda (seg.) |
|--------|--------------|----------------------------------|
| 1 vs 2 | 94,66        | 7,12                             |
| 1 vs 5 | 86,26        | 6,66                             |
| 1 vs 6 | 90,84        | 6,94                             |
| 1 vs 8 | 84,73        | 6,53                             |

Tabla 15 % de aciertos y tiempo con el cambio 2.

Observando los resultados, no se obtiene mejora, pero quitar parámetros mejorará la velocidad.

| Sesión         | % de acierto | Tiempo medio por búsqueda (seg.) |
|----------------|--------------|----------------------------------|
| <b>1 vs 2</b>  | 94,66        | 6,67                             |
| <b>1 vs 10</b> | 87,02        | 6,70                             |

Tabla 16 % de aciertos y tiempo con el cambio 3.

| Sesión        | % de acierto | Tiempo medio por búsqueda (seg.) |
|---------------|--------------|----------------------------------|
| <b>1 vs 2</b> | 94,66        | 7,71                             |
| <b>1 vs 3</b> | 86,26        | 5,98                             |
| <b>1 vs 4</b> | 87,79        | 5,59                             |

Tabla 17 % de aciertos y tiempo con el cambio 4.

| Sesión         | % de acierto | Tiempo medio por búsqueda (seg.) |
|----------------|--------------|----------------------------------|
| <b>1 vs 8</b>  | 86,26        | 6,67                             |
| <b>1 vs 10</b> | 87,02        | 6,66                             |

Tabla 18 % de aciertos y tiempo con el cambio 5.

| Sesión         | % de acierto | Tiempo medio por búsqueda (seg.) |
|----------------|--------------|----------------------------------|
| <b>1 vs 2</b>  | 94,66        | 6,67                             |
| <b>1 vs 10</b> | 87,02        | 6,70                             |

Tabla 19 % de aciertos y tiempo con el cambio 6.

Se puede observar que estos 4 cambios no han ofrecido ningún punto de mejora, ya que la precisión y el tiempo han sido los mismos en todos los casos, por tanto, estos cambios no se deberán añadir al proyecto.

Con esto, se ha establecido que el cambio más óptimo, es el cambio 2, ofreciéndonos el siguiente resultado:

| Sesión         | % de acierto | Tiempo medio por búsqueda (seg.) |
|----------------|--------------|----------------------------------|
| <b>1 vs 2</b>  | 94,66        | 6,58                             |
| <b>1 vs 3</b>  | 91,60        | 6,93                             |
| <b>1 vs 4</b>  | 89,31        | 6,55                             |
| <b>1 vs 5</b>  | 86,26        | 6,66                             |
| <b>1 vs 6</b>  | 90,84        | 6,94                             |
| <b>1 vs 7</b>  | 91,12        | 6,56                             |
| <b>1 vs 8</b>  | 86,26        | 6,65                             |
| <b>1 vs 9</b>  | 85,50        | 6,61                             |
| <b>1 vs 10</b> | 87,02        | 6,90                             |

Tabla 20 % de aciertos y tiempo final con el cambio 2, con todas las sesiones.

|       | Precisión | Desviación |
|-------|-----------|------------|
| Media | 89,17 %   | ± 2.94     |

Tabla 21 Precisión media y desviación después de realizar cambios.

Se observa que, aun realizando los cambios, la precisión sigue siendo menor a la del proyecto base. Sin embargo, esto es debido a que las comparativas se han realizado con sesiones en “crudo”, faltaría realizar las comparaciones con los patrones, ya mencionados anteriormente. Con estos se obtendrá un aumento significativo de la precisión.

#### 5.4. Cambio del algoritmo DTW.

Los resultados iniciales, sin aplicar el cálculo de media móvil centrada (CMA), ni el cambio de parámetros, son:

| Técnica               | Porcentaje de éxito |
|-----------------------|---------------------|
| <b>DTW</b>            | 86,37 % ± 5,04      |
| <b><i>FastDTW</i></b> | 86,50 % ± 3,49      |
| <b><i>Tslearn</i></b> | 79,01 % ± 2,93      |
| <b>UCRDTW</b>         | 7,87 %              |

Tabla 22 Porcentajes de éxitos iniciales con las diferentes técnicas.

Los resultados, después de aplicar los cambios, son los siguiente:

| Técnica               | Porcentaje de éxito |
|-----------------------|---------------------|
| <b>DTW</b>            | 88,62 % ± 5,12      |
| <b><i>FastDTW</i></b> | 89,17 % ± 2.94      |
| <b><i>Tslearn</i></b> | 83,78 % ± 3,64      |

Tabla 23 Porcentaje de éxito de las diferentes técnicas al realizar los cambios.

No se ha incluido el resultado de UCRDTW por lo dicho en el capítulo 4.

Observando los resultados, se llega a la conclusión de que la mejor técnica a usar es el *FastDTW*, y es el que se usará para las experimentaciones con los patrones y sesiones unidos.

## 5.5. Resultado global de las mejoras del proyecto.

Una vez obtenido los resultados, con los cambios realizados en los diferentes bloques. Los resultados adquiridos con las experimentaciones de la comparativa de los diferentes patrones con las sesiones son los siguientes:

| Sesión | Porcentaje (%) | Tiempo de ejecución medio (seg.) |
|--------|----------------|----------------------------------|
| 1      | 95,41          | 4,36                             |
| 2      | 94,66          | 4,58                             |
| 3      | 92,37          | 4,46                             |
| 4      | 90,84          | 4,21                             |
| 5      | 90,08          | 4,30                             |
| 6      | 91,6           | 4,85                             |
| 7      | 90,84          | 4,75                             |
| 8      | 93,13          | 4,61                             |
| 9      | 92,37          | 4,69                             |
| 10     | 91,60          | 4,72                             |

Tabla 24 Resultados de precisión de cada sesión con sus tiempos.

|       | Precisión | Desviación | Tiempo (seg.) |
|-------|-----------|------------|---------------|
| Media | 92,29 %   | $\pm 1,62$ | 4,55          |

Tabla 25 Media de resultados totales.

Como se puede observar, se ha producido una mejora considerable de, no solo la precisión, mejorando esta del 91,99 %, del proyecto base, al 92,29 %, de este proyecto, sino que también se ha mejorado el tiempo, de 5 segundos del proyecto base, a 4,55 segundos del proyecto de este TFG. La precisión de este proyecto es superior a la del proyecto base, aun aumentando el número de palabras de 100 a 131. Sin embargo, como es obvio, no supera la precisión con 50 palabra, del proyecto base.

Una vez finalizado las experimentaciones del proyecto, se ha llegado a los siguientes cambios de los elementos de los bloques del proyecto base:

- Creación e implementación de la técnica CMA, con una ventana de 25 elementos y un solapamiento del 25% en los extremos, para optimizar la frecuencia de muestro del sistema.
- Eliminación de los parámetros *Palm.position* y *Palm.orientation*. *w* de los parámetros seleccionados del *Leap Motion*.
- Selección de la técnica *FastDTW* como técnica usada en el comparador.



## 5.6. Comparativa

Una vez obtenido el resultado final, es buen punto el realizar una comparativa entre el resultado obtenido, el resultado del proyecto base y algunos resultados de otras investigaciones.

Primeramente, como se ha mostrado anteriormente, la precisión obtenida, con las mejoras de este proyecto, usando una base de datos de 131 palabras y la técnica de comparación FastDTW, asciende al 92,29 %. Este resultado es superior al proyecto base que, usando una base de datos de 100 palabras y la técnica de comparación DTW, obtuvo en su momento una precisión del 91,99 %.

Por otra parte, si se compara con investigaciones que hagan uso del sensor *Leap Motion*, se puede encontrar, por ejemplo, un proyecto que, con una base de datos de 26 letras del lenguaje de signos americano y que usa K-NN y SVM, obtiene una precisión del 72,78 % y 79,83 %, respectivamente, muy inferior a este proyecto, aún, teniendo una base de datos menor. Por otro lado, se puede encontrar otro proyecto, con una base de datos de solo 26 palabras dinámicas del lenguaje de signos turco, que usa el *Extreme Learning Machine* (ELM), una de las grandes apuestas del futuro y, que ha obtenido una precisión del 93 %. Esta precisión es muy baja, considerando el limitado número de palabras usadas y, al compararlo con este proyecto.

Si se observas otros tipos de tecnología, se puede encontrar el proyecto desarrollado con el *Kinect*, el cual, posee una base de datos de 20 signos del lenguaje de signos italiano y hace uso de las redes neuronales convolucionales, para obtener una precisión del 91,7 %. Otros de los posibles proyectos a comparar, es el desarrollado con el *Azure Kinect*, la evolución del *Kinect*, el cual, hace uso de una base de datos de 18 gestos y CNN, para obtener una precisión del 41 %. Ambos resultados, demuestran que los resultados obtenidos por este proyecto son superiores a estos.

En definitiva, se ha demostrado que se ha realizado una correcta selección del sensor, técnica de comparación y cambios del proyecto.

| Referencia | Tecnologías  | Resultados  |
|------------|--|---|
| [74]       | <i>LEAP MOTION</i><br>Lenguaje de signos americano<br>26 letras<br>K-NN<br>SVM | K-NN -> 72,78 % precisión<br>SVM -> 79,83 % precisión |
| [76]       | <i>LEAP MOTION</i><br>Lenguaje de signos turco<br>ELM                          | 93 % de precisión                                     |

|                           |  |                     |
|---------------------------|--|---------------------|
|                           | 26 letras  |                     |
| <b>[89]</b>               | <i>Kinect</i><br>Lenguaje de signos italiano<br>20 signos<br>Redes neuronales<br>convolucionales | 91,7 % de precisión |
| <b>[92]</b>               | <i>Azure Kinect</i><br>18 gestos<br>CNN  | 41 % de precisión   |
| <b>Proyecto base [12]</b> | <i>LEAP MOTION</i><br>Lenguaje de signos español<br>176 palabras                                 | 91,61 %             |
| <b>Proyecto del TFG</b>   | <i>LEAP MOTION</i><br>Lenguaje de signos español<br>131 palabras<br><i>FastDTW</i>               | 92,29 %             |

Tabla 26 Comparación de resultados

## 6. Conclusiones

### 6.1. Valoración final.

Al principio de este Trabajo de Fin de Grado se establecieron unos objetivos a superar y una hipótesis a demostrar. Estos objetivos se han ido cumplimentando con la realización de diferentes tareas, a lo largo del proyecto. A continuación, se especificarán los objetivos superados, así como un pequeño análisis de estos.

| Objetivos  | Análisis   |
|--|--|
| <b>Objetivo 1:</b> Analizar las soluciones de detección automática del lenguaje de signos existentes en la actualidad. | En el capítulo 2, se hace un análisis exhaustivo de las diferentes tecnologías creadas, así como de los diferentes proyectos existen, referentes al tema principal del TFG.<br>Objetivo cumplido.                                |
| <b>Objetivo 2:</b> Mejorar la velocidad en la detección del lenguaje de signos del proyecto base.                      | En el capítulo 4, se muestra los diferentes cambios realizados en el proyecto base, y en el capítulo 5, se presentan los resultados obtenidos de estos cambios, reflejando la mejora de velocidad buscada.<br>Objetivo cumplido. |
| <b>Objetivo 3:</b> Mejorar la precisión en la detección del lenguaje de signos del proyecto base.                      | En el capítulo 4, se establecen los cambios realizados en el proyecto y se muestran los resultados en el capítulo 5, indicando una mejora de la precisión buscada.<br>Objetivo cumplido.   |
| <b>Objetivo 4:</b> Programar y mejorar el código del proyecto base.  | Para la realización de lo indicado en el capítulo 4 y 5, se ha tenido que modificar y mejora el código existente, así como, crear nuevo código para cumplir los objetivos anteriores.<br>Objetivo cumplido.                      |

Tabla 27 Análisis de los objetivos del TFG.

La hipótesis establecida inicialmente, en el capítulo 1 del TFG, fue la siguiente:

“Es posible, dado un proyecto base con unas características y cualidades especificadas, con una velocidad y una precisión establecidas, analizar y mejorar las partes de dicho proyecto”

Una vez realizado todo el proyecto y comprobando el cumplimiento satisfactorio de los objetivos, se puede llegar a la conclusión de que los resultados confirman la hipótesis.

Por otro lado, la realización del TFG ha aportado nuevas conclusiones. La primera es que, aun obteniendo unos resultados satisfactorios, se ha observado que aún existe un margen enorme de mejora en múltiples vertientes del proyecto, ofreciendo la posibilidad de nuevos proyectos con el objetivo general de seguir mejorando el proyecto, pero en este caso, el proyecto base sería el proyecto de este TFG. La segunda sería que, en el momento de realizar la experimentación y comprobación, se ha notado la necesidad de una base de datos mayor, con más grabaciones, procedentes de distintas personas, en momentos distintos, con el objetivo de obtener datos más precisos y mayor margen de experimentación. La última conclusión es que, aun no teniendo el código completo del proyecto base, se ha podido desarrollar el TFG sin ningún problema, sustituyendo aquellas partes necesarias por código propio.

Este TFG ha aportado al autor una fuente de conocimiento sobre las técnicas de DTW, sobre el lenguaje de signo y el mundo que lo rodea, y sobre los procedimientos, requerimientos y consejos a la hora de realizar estos tipos de proyectos.

## **6.2. Posibles puntos de mejora.**

Como se ha mencionado en el punto anterior, y considerando que este proyecto se trata de un TFG de investigación, este Trabajo de Fin de Grado, tiene como objetivo principal el mejorar el proyecto inicial, pero, aún sigue teniendo margen de mejora, pudiendo, no solo mejorar los puntos ya mejorados en este proyecto, sino que, se podrían aplicar, diseñar y crear nuevos bloques para que el proyecto obtenga nuevas capacidades, y sea un proyecto más completo, más fácil de usar, más intuitivo, entre otras cosas.

### **6.2.1. Puntos ya mejorados con posibilidad de mejora.**

Este TFG, se ha centrado en analizar, modificar y mejorar tres puntos de los diferentes bloques existentes de proyecto.

En primer lugar, se ha optimizado la frecuencia de muestro, aplicando técnicas de cálculo de media móvil centrada (CMA) y ampliando el promediado a un porcentaje de datos externos a la ventana. Este cambio ha tenido como desafío, el obtener el tamaño óptimo de la ventana del promediado y obtener la proporción ideal de datos externos incluidos al promediado. Por tanto, aun habiendo seleccionado uno de esos puntos óptimos, puede que existan otros puntos óptimos, que no se hayan obtenido y requieran de más experimentación. Además, pueden existir otras técnicas que faciliten el proceso u obtengan mejores datos.

En segundo lugar, aún seleccionado los parámetros más idóneos, puede darse el caso de que, se puedan eliminar más parámetros sin afectar a la precisión o, con la implementación a futuro del *Leap*

*Motion Controler 2*, pueda que surjan nuevos parámetros o se eliminen alguno, pudiendo mejorar la precisión en ese caso.

Por último, el cambio de la técnica DTW a *FastDTW* ha proporcionado una mejora en la precisión y en la velocidad. No obstante, aún queda pendiente la técnica fallida, mencionada en su momento. Esta puede llegar a arreglarse y puede que mejore enormemente la precisión y velocidad. Otro punto que mejorar es el intentar arreglar ese programa e implementarlo para que funcione correctamente. También se podría buscar nuevas técnicas más precisas y veloces o crear una desde cero que supla las necesidades del proyecto.

### **6.2.2. Algoritmo de creación de frases.**

Otro aspecto que puede llegar a mejorarse es el hecho de que la traducción se realiza de palabra en palabra, algo que ofrece una sensación antinatural, a la hora de escucharse la traducción. Además, las palabras que son, por ejemplo, verbos, son interpretadas en infinitivo en el lenguaje de signos. Para dar solución a este punto se podría implementar con IA, un sistema que tome varias palabras signadas y las junte en una frase. Por ejemplo, con las palabras signadas: hola, tener, dolor, cabeza, la IA las junte, formando la frase “hola, tengo dolor de cabeza”. Esta idea proporcionaría un aspecto más natural al proceso y un mayor entendimiento entre las dos partes de la conversión. Para crear este bloque, se podría usar programas de IA, tales como, ChatGPT o similar.

### **6.2.3. Proceso traducción inverso.**

Establecido una dirección de la traducción, de la persona que signa las palabras, a otra que las escucha, se podría crear el camino inverso. Este camino consistiría en un micrófono que captaría la frase hablada y la traduciría en imágenes/videos de una persona signando las palabras de esa frase, o una aproximación de estas. Este proceso requeriría de bloques de ejecución nuevos, pero el concepto sería el mismo que en el caso original pero invertido.

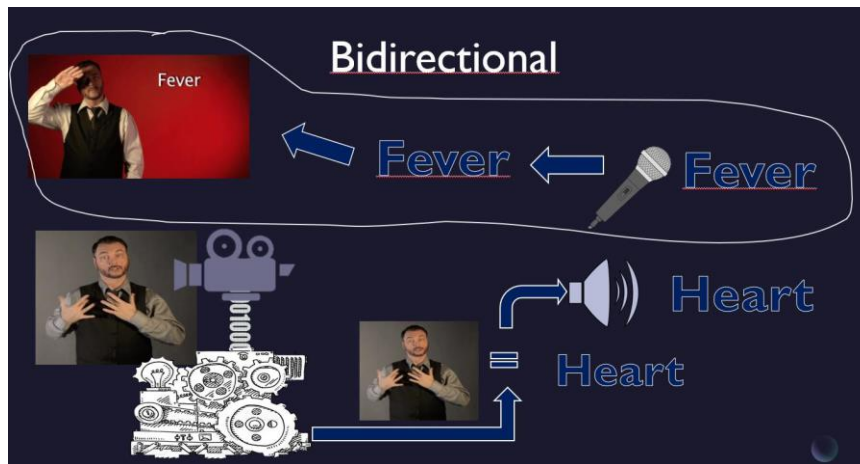


Ilustración 9 Proceso de traducción inverso.

#### 6.2.4. Deep Learning y/o Matching Learning.

Para mejorar el proceso de selección de la palabra signada se podría aplicar técnicas de *Deep Learning* y/o *Matching Learning*. Esta aplicación podría mejorar enormemente la velocidad y precisión de la detección del lenguaje de signos. También ofrecería una capacidad de entrenamiento del sistema enorme, permitiendo entrenar el sistema con todos los datos ofrecidos de las diferentes grabaciones. No obstante, esto requeriría una mayor base de datos con más grabaciones de palabras signadas, en contextos distintos.

#### 6.2.5. Otros puntos de mejora.

Ya mencionado anteriormente, se requeriría de nuevas grabaciones de palabras signadas, en contextos y tiempos distintos. Esto, con el objetivo de tener una mayor base de datos y mejorar así, el rango de entrenamiento, creación de mejores patrones, para así mejorar la precisión y calidad de detección del lenguaje de signos. Además, se podrían ampliar las muestras usadas en la creación del patrón, aumentando este de 50 muestras a 60, incluso llegando a 100. Esta idea, mejoraría enormemente la calidad de los diferentes patrones, y mejoraría la precisión en el bloque de comparación, a la hora de aplicar la técnica de DTW con la muestra de entrada y los respectivos patrones.

Otro posible punto de mejora es el no solo comparar los patrones de las muestras, con los datos entrantes, sino que se podría obtener datos de los gestos de la cara, y usarlos como otro parámetro de comparación. Esto es debido a que ciertas palabras son idénticas, a la hora de signarlas, y solo se diferencian con las expresiones de las caras, a la hora de signar la palabra.

Por último, se podría establecer un sistema que identifique y clasifique las distintas palabras, según si son signadas con una o dos manos, simplificando la búsqueda.

## Bibliografía

- [1] J. Adriko, «La población mundial llega a los 8000 millones de habitantes,» Naciones Unidas, [En línea]. Available: <https://www.un.org/es/desa/la-poblaci%C3%B3n-mundial-llega-los-8000-millones-de-habitantes>. [Último acceso: 23 mayo 2024].
- [2] «Día Internacional de las Lenguas de Señas, 23 de septiembre,» Naciones Unidas, 3 febrero 2024. [En línea]. Available: Disponible en: <https://www.un.org/es/observances/sign-languages-day#:~:text=Seg%C3%BAn%20la%20Federaci%C3%B3n%20Mundial%20de,sordas%20en%20todo%20el%20mundo>.
- [3] INE, «INE - Instituto Nacional de Estadística,» INE, [En línea]. Available: <https://www.ine.es/jaxi/Datos.htm?tpx=51628>. [Último acceso: 3 febrero 2024].
- [4] «España Digital 2026,» Ministerio de Economía, Comercio y Empresa, [En línea]. Available: [https://portal.mineco.gob.es/en-us/ministerio/estrategias/Pages/00\\_Espana\\_Digital.aspx](https://portal.mineco.gob.es/en-us/ministerio/estrategias/Pages/00_Espana_Digital.aspx). [Último acceso: 3 febrero 2024].
- [5] «Getting Started with the Leap Motion SDK,» Ultraleap, [En línea]. Available: <https://blog.leapmotion.com/getting-started-leap-motion-sdk/>. [Último acceso: 10 Julio 2024].
- [6] G. Marin, F. Dominio and P. Zanuttigh, "Hand gesture recognition with leap motion and kinect devices," 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 2014, pp. 1565-1569, doi: 10.1109/ICIP.2014.7025313.
- [7] M. Simos y N. Nikolaidis, "Greek sign language alphabet recognition using the leap motion device", en SETN '16: 9th Hellenic Conf. Artif. Intell., Thessaloniki Greece. New York, NY, USA: ACM, 2016. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.1145/2903220.2903249>

- [8] B. Hisham y A. Hamouda, “Arabic Sign Language Recognition using Microsoft Kinect and Leap Motion Controller”, *SSRN Electron. J.*, 2018. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.2139/ssrn.3389799>
- [9] T.-W. Chong y B.-G. Lee, “American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach”, *Sensors*, vol. 18, n.º 10, p. 3554, octubre de 2018. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.3390/s18103554>
- [10] A. S. Elons, M. Ahmed, H. Shedid y M. F. Tolba, “Arabic sign language recognition using leap motion sensor”, en *2014 9th Int. Conf. Comput. Eng. Syst. (ICCES)*, Cairo, Egypt, 22–23 de diciembre de 2014. IEEE, 2014. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.1109/icces.2014.7030987>
- [11] J. Jenkins y S. Rashad, “An Innovative Method for Automatic American Sign Language Interpretation using Machine Learning and Leap Motion Controller”, en *2021 IEEE 12th Annu. Ubiquitous Comput., Electron. & Mobile Communication Conf. (UEMCON)*, New York, NY, USA, 1–4 de diciembre de 2021. IEEE, 2021, pp. 0633–0638. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.1109/uemcon53757.2021.9666640>
- [12] J. Galván-Ruiz, C. M. Travieso-González, A. Pinan-Roescher y J. B. Alonso-Hernández, “Robust Identification System for Spanish Sign Language Based on Three-Dimensional Frame Information”, *Sensors*, vol. 23, n.º 1, p. 481, enero de 2023. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.3390/s23010481>
- [13] I. E. Sutherland, “Sketchpad”, en *May 21-23, 1963, spring joint comput. conf.*, Detroit, Michigan, 21–23 de mayo de 1963. New York, New York, USA: ACM Press, 1963. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.1145/1461551.1461591>
- [14] M. W. Krueger, T. Gionfriddo y K. Hinrichsen, “VIDEOPLACE---an



- artificial reality”, en SIGCHI conf., San Francisco, California, United States. New York, New York, USA: ACM Press, 1985. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.1145/317456.317463>
- [15] Zimmerman. T.G, “Optical flex sensor”, Patente EE. UU. US4542291A, 17 de septiembre de 1985.
- [16] J. J. LaViola, “A Multimodal Interface Framework for Using Hand Gestures and Speech in Virtual Environment Applications”, en *Gesture-Based Communication in Human-Computer Interaction*. Berlin, Heidelberg: Springer Berl. Heidelb., 1999, pp. 303–314. Accedido el 16 de julio de 2024. [En línea]. Disponible: [https://doi.org/10.1007/3-540-46616-9\\_27](https://doi.org/10.1007/3-540-46616-9_27)
- [17] J. L. Hernandez-Rebollar, N. Kyriakopoulos y R. W. Lindeman, “The AcceleGlove”, en *ACM SIGGRAPH 2002 conf. abstr. appl.*, San Antonio, Texas, 21–26 de julio de 2002. New York, New York, USA: ACM Press, 2002. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.1145/1242073.1242272>
- [18] “CyberGlove Systems LLC”. CyberGlove Systems LLC. Accedido el 16 de julio de 2024. [En línea]. Disponible: <http://www.cyberglovesystems.com/>
- [19] M. E. Valentinuzzi, “Bioelectrical signal processing in cardiac and neurological applications and electromyography: physiology, engineering, and noninvasive applications”, *Biomed. Eng. OnLine*, vol. 6, n.º 1, p. 27, 2007. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.1186/1475-925x-6-27>
- [20] D. Farina, D. F. Stegeman y R. Merletti, “Biophysics of the Generation of EMG Signals”, en *Surface Electromyography: Physiology, Engineering, and Applications*. Hoboken, New Jersey: Wiley, 2016, pp. 1–24. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.1002/9781119082934.ch02>
- [21] C. J. De Luca, “The Use of Surface Electromyography in Biomechanics”, *J. Appl. Biomechanics*, vol. 13, n.º 2, pp. 135–163, mayo de 1997. Accedido el

16 de julio de 2024. [En línea].  
Disponible: <https://doi.org/10.1123/jab.13.2.135>

- [22] BARRETO, Armando; SCARGLE, Scott; ADJOUADI, Malek. “Hands-off human-computer interfaces for individuals with severe motor disabilities”. En Proceedings of the HCI International'99 (the 8th International Conference on Human-Computer Interaction) on Human-Computer Interaction: Communication, Cooperation, and Application Design-Volume 2-Volume 2. 1999. p. 970-974.
- [23] K. Coleman, “Electromyography based human-computer-interface to induce movement in elderly persons with movement impairments”, en 2001 EC/NSF Workshop, Alcácer do Sal, Portugal, 22–25 de mayo de 2001. New York, New York, USA: ACM Press, 2001. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.1145/564526.564547>
- [24] JOA, Tiago, et al. “EMG as a daily wearable interface”. En International Conference on Computer Graphics Theory and Applications. SCITEPRESS, 2006. p. 216-223.
- [25] AHSAN, Md Rezwanul, et al. “EMG signal classification for human computer interaction: a review”. European Journal of Scientific Research, 2009, vol. 33, no 3, p. 480-501.
- [26] CRAWFORD, Beau, et al. “Real-time classification of electromyographic signals for robotic control”. En AAAI. 2005. p. 523-528.
- [27] KIGUCHI, Kazuo; HAYASHI, Yoshiaki. “An EMG-based control for an upper-limb power-assist exoskeleton robot”. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2012, vol. 42, no 4, p. 1064-1071.
- [28] JOSHI, Chetas D.; LAHIRI, Uttama; THAKOR, Nitish V. “Classification of gait phases from lower limb EMG: Application to exoskeleton orthosis”. En 2013 IEEE Point-of-Care Healthcare Technologies (PHT). IEEE, 2013. p. 228-231.

- [29] FISTRE, J.; TANAKA, A. “Real time EMG gesture recognition for consumer electronics device control”. Sony CSL Paris Open House 10/2002, 2002.
- [30] WHEELER, Kevin R.; JORGENSEN, Charles C. “Gestures as input: Neuroelectric joysticks and keyboards. IEEE pervasive computing”, 2003, vol. 2, no 2, p. 56-61.
- [31] DHILLON, Hari Singh, et al. “EOG and EMG based virtual keyboard: A brain-computer interface”. En 2009 2nd IEEE international conference on computer science and information technology. IEEE, 2009. p. 259-262.
- [32] ROSENBERG, Robert. “The biofeedback pointer: EMG control of a two dimensional pointer”. En Digest of Papers. Second International Symposium on Wearable Computers (Cat. No. 98EX215). IEEE, 1998. p. 162-163.
- [33] Z. Zhang, T. Han, C. Huang y C. Shuai, “Hardware and Software Design and Implementation of Surface-EMG-Based Gesture Recognition and Control System”, Electronics, vol. 13, n.º 2, p. 454, enero de 2024. Accedido el 18 de julio de 2024. [En línea]. Disponible: <https://doi.org/10.3390/electronics13020454>
- [34] BOYALI, Ali; HASHIMOTO, Naohisa; MATSUMOTO, Osamu. “Hand posture and gesture recognition using MYO armband and spectral collaborative representation based classification”. En 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE). IEEE, 2015. p. 200-201.
- [35] ABREU, João Gabriel, et al. “Evaluating sign language recognition using the myo armband”. En 2016 XVIII symposium on virtual and augmented reality (SVR). IEEE, 2016. p. 64-70.
- [36] SATHIYANARAYANAN, Mithileysh; RAJAN, Sharanya. “MYO Armband for physiotherapy healthcare: A case study using gesture recognition application”. En 2016 8th International Conference on Communication Systems and Networks (COMSNETS). IEEE, 2016. p. 1-6.
- [37] BENALCÁZAR, Marco E., et al. “Real-time hand gesture recognition using the Myo armband and muscle activity detection”. En 2017 IEEE second

Ecuador technical chapters meeting (ETCM). IEEE, 2017. p. 1-6.

- [38] SINGH, Shashank Kumar; CHATURVEDI, Amrita. “A reliable and efficient machine learning pipeline for american sign language gesture recognition using EMG sensors”. *Multimedia Tools and Applications*, 2023, vol. 82, no 15, p. 23833-23871.
- [39] FLÉTY, Emmanuel. “3D gesture acquisition using ultrasonic sensors”. *Trends in Gestural Control of Music*, 2000, p. 193-207.
- [40] KRECZMER, Bogdan. “Gestures recognition by using ultrasonic range-finders”. En *2011 16th International Conference on Methods & Models in Automation & Robotics*. IEEE, 2011. p. 363-368.
- [41] PRZYBYLA, R. J., et al. “In-air ultrasonic rangefinding and angle estimation using an-array of AIN micromachined transducers”. En *Proc. Hilton Head Solid-State Sensors, Actuators and Microsystems Workshop*. 2012. p. 3-7.
- [42] BOOIJ, Wilfred Edwin; WELLE, Knut Osvald. “Ultrasound detectors”. U.S. Patent No 8,792,305, 29 jul. 2014.
- [43] HODGES, P. W., et al. “Measurement of muscle contraction with ultrasound imaging”. *Muscle & Nerve: Official Journal of the American Association of Electrodiagnostic Medicine*, 2003, vol. 27, no 6, p. 682-692.
- [44] MUJIBIYA, Adiyana, et al. “The sound of touch: on-body touch and gesture sensing based on transdermal ultrasound propagation”. En *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*. 2013. p. 189-198.
- [45] MCINTOSH, Jess, et al. “Echoflex: Hand gesture recognition using ultrasound imaging”. En *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017. p. 1923-1934.
- [46] ABDELNASSER, Heba; YOUSSEF, Moustafa; HARRAS, Khaled A.” Wigest: A ubiquitous wifi-based gesture recognition system”. En *2015 IEEE conference on computer communications (INFOCOM)*. IEEE, 2015. p. 1472-1480.

- [47] PU, Qifan, et al. "Whole-home gesture recognition using wireless signals". En Proceedings of the 19th annual international conference on Mobile computing & networking. 2013. p. 27-38.
- [48] ADIB, Fadel, et al. "3D tracking via body radio reflections". En 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14). 2014. p. 317-329.
- [49] ADIB, Fadel; KABELAC, Zachary; KATABI, Dina. "{Multi-Person} Localization via {RF} Body Reflections". En 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15). 2015. p. 279-292.
- [50] NANDAKUMAR, Rajalakshmi; KELLOGG, Bryce; GOLLAKOTA, Shyamnath. "Wi-fi gesture recognition on existing devices". arXiv preprint arXiv:1411.5394, 2014.
- [51] LI, Hong, et al. "WiFinger: Talk to your smart devices with finger-grained gesture". En Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2016. p. 250-261.
- [52] R. Gao *et al.*, "WiCGesture: Meta-Motion-Based Continuous Gesture Recognition With Wi-Fi," in *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 15087-15099
- [53] DABWAN, Basel A., et al. "Arabic Sign Language Recognition Using EfficientnetB1 and Transfer Learning Technique". En 2023 International Conference on IT Innovation and Knowledge Discovery (ITIKD). IEEE, 2023. p. 1-5.
- [54] ALDHAHRI, Eman, et al. "Arabic sign language recognition using convolutional neural network and mobilenet". Arabian Journal for Science and Engineering, 2023, vol. 48, no 2, p. 2147-2154.
- [55] CHUNG, Wan-Young; XU, Haokai; LEE, Boon Giin. "Chinese sign language recognition with batch sampling ResNet-Bi-LSTM". SN Computer Science, 2022, vol. 3, no 5, p. 414.

- [56] TRUCCO, Emanuele; VERRI, Alessandro. “Introductory techniques for 3-D computer vision”. Englewood Cliffs: Prentice Hall, 1998.
- [57] MANTECA FERNÁNDEZ, Fernando, et al. “Generación de modelos 3D mediante luz estructurada”. 2018.
- [58] MO, Hao-Cheng; LEOU, Jin-Jang; LIN, Cheng-Shian. “Human Behavior Analysis Using Multiple 2D Features and Multicategory Support Vector Machine”. En MVA. 2009. p. 46-49.
- [59] RAPTIS, Michalis; KIROVSKI, Darko; HOPPE, Hugues. “Real-time classification of dance gestures from skeleton animation”. En Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation. 2011. p. 147-156.
- [60] “Digital worlds that feel human | Ultraleap”. Digital worlds that feel human | Ultraleap. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://www.ultraleap.com/>
- [61] HANTRAKUL, Lamtharn; KACZMAREK, Konrad. “Implementations of the Leap Motion device in sound synthesis and interactive live performance”. En Proceedings of the 2014 International Workshop on Movement and Computing. 2014. p. 142-145.
- [62] LING, He; RUI, Li. “VR glasses and leap motion trends in education”. En 2016 11th International Conference on Computer Science & Education (ICCSE). IEEE, 2016. p. 917-920.
- [63] WANG, Chong; LIU, Zhong; CHAN, Shing-Chow. “Superpixel-based hand gesture recognition with kinect depth camera”. IEEE transactions on multimedia, 2014, vol. 17, no 1, p. 29-39.
- [64] SESLI, Erhan. “Human-Robot Interaction (HRI) through hand gestures for possible future war robots: A leap motion controller application. Multimedia Tools and Applications”, 2024, vol. 83, no 12, p. 36547-36570.
- [65] GÁNDARA, Carolina Valdez; BAUZA, Cristian García. “IntelliHome: A framework for the development of ambient assisted living applications based

- in low-cost technology”. En Proceedings of the Latin American Conference on Human Computer Interaction. 2015. p. 1-4.
- [66] SMERAGLIUOLO, Anna H., et al. “Validation of the Leap Motion Controller using markered motion capture technology”. Journal of biomechanics, 2016, vol. 49, no 9, p. 1742-1750.
- [67] MOHANDDES, Mohamed; ALIYU, S.; DERICHE, Mohamed. “Arabic sign language recognition using the leap motion controller”. En 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE). IEEE, 2014. p. 960-965.
- [68] CHEN, Yanmei, et al. “Rapid recognition of dynamic hand gestures using leap motion”. En 2015 IEEE International Conference on Information and Automation. IEEE, 2015. p. 1419-1424.
- [69] MCCARTNEY, Robert; YUAN, Jie; BISCHOF, Hans-Peter. “Gesture recognition with the leap motion controller”. En Proceedings of the international conference on image processing, computer vision, and pattern recognition (IPCV). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015.
- [70] LU, Wei; TONG, Zheng; CHU, Jinghui. “Dynamic hand gesture recognition with leap motion controller”. IEEE Signal Processing Letters, 2016, vol. 23, no 9, p. 1188-1192.
- [71] SAAD, Muhammad; YANG, Tianbao; ZHOU, Hui. “A Comparison of Bidirectional GRU and LSTM for Hand Gesture Recognition Using Leap Motion”. En 2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC). IEEE, 2022. p. 1427-1433.
- [72] WU, Bixiao; LU, Zhenyu; YANG, Chenguang. “A modified LSTM model for Chinese sign language recognition using leap motion”. En 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2022. p. 1612-1617.

- [73] ABDULLAHI, Sunusi Bala; CHAMNONGTHAI, Kosin. "American sign language words recognition using spatio-temporal prosodic and angle features: A sequential learning approach". IEEE Access, 2022, vol. 10, p. 15911-15923.
- [74] NASIR, Muhamad; MUSRI, Tengku; KURNIAWATY, Eva. "Hand Gesture Recognition using Leap Motion Controller for Recognition of Javanese Script. International ABEC, 2023, p. 124-128.
- [75] RISHAN, R. M.; JAYALAL, S.; WIJAYASIRIWARDHANE, T. K. "Translation of sri lankan sign language to sinhala text: A leap motion technology-based approach". En 2022 2nd International Conference on Advanced Research in Computing (ICARC). IEEE, 2022. p. 218-223.
- [76] KATILMIŞ, Zekeriya; KARAKUZU, Cihan. "Double handed dynamic Turkish Sign Language recognition using Leap Motion with meta learning approach". Expert Systems with Applications, 2023, vol. 228, p. 120453.
- [77] R. Bekiri, S. Babahenini and M. C. Babahenini, "Transfer Learning for Improved Hand Gesture Recognition with Neural Networks," 2024 8th International Conference on Image and Signal Processing and their Applications (ISPA), Biskra, Algeria, 2024, pp. 1-6
- [78] LETH, Peter Guld. "Danish Sign Language Recognition in Virtual Reality Using Written Language Ensemble Learning". Universidad de Aalborg, 2023.
- [79] FUCHS, Christian, et al." Investigation on Tactile Perception of Ultrasonic Haptics Devices". En 2022 IEEE 9th International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA). IEEE, 2022. p. 1-6.
- [80] MULOT, Lendy, et al. "Dolphin: A framework for the design and perceptual evaluation of ultrasound mid-air haptic stimuli". En ACM Symposium on Applied Perception 2021. 2021. p. 1-10.
- [81] YOUNG, Gareth, et al. "Designing mid-air haptic gesture controlled user interfaces for cars". Proceedings of the ACM on human-computer interaction,



2020, vol. 4, no EICS, p. 1-23.

- [82] GEORGIU, Orestis, et al. "Mid-air haptic texture exploration in vr". En 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW). IEEE, 2022. p. 964-965.
- [83] PANEVA, Viktorija, et al. "HaptiRead: Reading Braille as mid-air haptic information". En Proceedings of the 2020 ACM designing interactive systems conference. 2020. p. 13-20.
- [84] ABDEL-SAMIE, Abdel-Gawad Abdel-Rabouh, et al. "Arabic sign language recognition using kinect sensor". Research Journal of Applied Sciences, Engineering and Technology, 2018, vol. 15, no 2, p. 57-67.
- [85] SARHAN, Noha A.; EL-SONBATY, Yasser; YOUSSEF, Sherine M. HMM-based "Arabic sign language recognition using Kinect". En 2015 tenth international conference on digital information management (ICDIM). IEEE, 2015. p. 169-174.
- [86] SUN, Chao; ZHANG, Tianzhu; XU, Changsheng. "Latent support vector machine modeling for sign language recognition with Kinect". ACM Transactions on Intelligent Systems and Technology (TIST), 2015, vol. 6, no 2, p. 1-20.
- [87] ZAFRULLA, Zahoor, et al. "American sign language recognition with the kinect. En Proceedings of the 13th international conference on multimodal interfaces". 2011. p. 279-286.
- [88] LANG, Simon; BLOCK, Marco; ROJAS, Raúl. "Sign language recognition using Kinect". En International Conference on Artificial Intelligence and Soft Computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 394-402.
- [89] PIGOU, Lionel, et al. "Sign language recognition using convolutional neural networks". En Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I 13. Springer International Publishing, 2015. p. 572-578.

- [90] MEHROTRA, Kamil; GODBOLE, Atul; BELHE, Swapnil. "Indian sign language recognition using kinect sensor". En Image Analysis and Recognition: 12th International Conference, ICIAR 2015, Niagara Falls, ON, Canada, July 22-24, 2015, Proceedings 12. Springer International Publishing, 2015. p. 528-535.
- [91] "Documentación de Azure Kinect DK". Microsoft Learn: Build skills that open doors in your career. Accedido el 16 de julio de 2024. [En línea]. Disponible: <https://learn.microsoft.com/es-es/azure/kinect-dk/>
- [92] BROCK, Heike; GOMEZ, Randy. "Making The Unknown More Certain: A Stacked Ensemble Classifier for Open Gesture Recognition with a Social Robot". En ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022. p. 3918-3922.

# Anexo

## Ilustraciones

**Ilustración 1.** Persona signando una palabra. Fuente:

<https://www.istockphoto.com/es/search/2/image-film?phrase=black+woman+sign+language>

**Ilustración 2.** Dispositivos diseñados hasta el momento para el reconocimiento gestual. Fuente: Elaboración propia.

**Ilustración 3.** Hardware de Leap Motion. Fuente: <https://www.tecnogeek.com/2013/10/03/leap-motion-controlando-la-pc-con-nuestros-dedos/>

**Ilustración 4.** Esquema del comparador. Fuente: <https://doi.org/10.3390/s23010481>

**Ilustración 5.** Matriz de confusión. Fuente: [https://es.linkedin.com/posts/naren-castellon-1541b8101\\_matriz-de-confusi%C3%B3n-una-herramienta-fundamental-activity-7139356021419896832-TUt9](https://es.linkedin.com/posts/naren-castellon-1541b8101_matriz-de-confusi%C3%B3n-una-herramienta-fundamental-activity-7139356021419896832-TUt9)

**Ilustración 6.** Proceso de diezmado. Fuente:

[http://www.lucamartino.altervista.org/DIEZMADO\\_Interpolacion\\_Secuencia.pdf](http://www.lucamartino.altervista.org/DIEZMADO_Interpolacion_Secuencia.pdf)

**Ilustración 7.** Ventanas y solapamiento. Fuente: Elaboración propia.

**Ilustración 8.** Representación de los porcentajes de acierto de la tabla. Fuente: Elaboración propia.

**Ilustración 9.** Proceso de traducción inverso. Fuente: Elaboración propia.



# **PARTE II – PRESUPUESTO**



# 1. Desglose del presupuesto

Debido al estado del mercado, actualmente liberalizado, todos precios de cada parte son puesto por el ingeniero, el cual, ha realizado un posible estudio de precios para establecerlos y, posteriormente serán revisados por el COITT. No obstante, debido a la imposibilidad de realizar dicho estudio, en este TFG, se seguirán las antiguas recomendaciones que el COITT dieron en su día.

El cálculo del presupuesto para la elaboración del TFG se ha elaborado teniendo en cuenta los precios recomendados, así como las directrices instauradas por el Colegio Oficial de Ingenieros Técnicos de Telecomunicación (COITT). El documento ofrece una estimación de los costos vinculados a la realización del proyecto, fundamentada en los siguientes puntos clave:

- 1.1. Recursos materiales.
- 1.2. Tarifas basadas en el Tiempo Empleado.
- 1.3. Costes relacionados con la Elaboración del Documento.
- 1.4. Derecho por el Visado del COITT.
- 1.5. Costos de Tramitación y Envío.
- 1.6. Impuestos Aplicables

## 1.1. Recursos materiales

Los recursos materiales usados en la creación y desarrollo de este proyecto se pueden subdividir en componentes *hardware* y *software*. Para el cálculo del costo de amortización, se tendrá en cuenta un periodo de 4 años, considerando, en este caso, un sistema de amortización lineal, con un material inmovilizado depreciado de forma constante durante el periodo analizado. Este cálculo se realizará con la siguiente fórmula matemática:

$$\text{Coste de amortización} = \frac{\text{Coste de adquisición} - \text{Valor residual}}{\text{Años de vida útil}}$$

Con esta fórmula se podrá establecer el coste de amortización asociado a los recursos materiales empleados en este proyecto.

### 1.1.1. Recursos hardware

En este caso, la duración del proyecto no llega a 4 años, siendo este de una duración total de 300 horas repartidas en 4 meses, por tanto, el costo de amortización se realizará únicamente de estos 4 meses, la duración total del proyecto.

Los recursos *hardware* usados en este TFG han sido los siguientes:

- Ordenador personal con conexión a Internet (Procesador AMD Ryzen 7 3800X (8 núcleos, 390 GHz), memoria RAM 16 GB 3600 MHz DDR4, Radeon RX 5700 8 GB).
- Pantalla Legion Y25f – 10 24.5” LED FullHD 144Hz

| Recurso                | Valor de Adquisición (€) | Valor Residual (€) | Coste de Amortización (€) |
|------------------------|--------------------------|--------------------|---------------------------|
| Ordenador de Mesa      | 1280,00                  | 614,00             | 166,50                    |
| Pantalla               | 170,00                   | 136,00             | 8,50                      |
| <b>Total, Hardware</b> |                          | <b>175,00 €</b>    |                           |

Tabla 28 Coste de amortización de los recursos hardware

Realizado los cálculos, el coste total de amortización relacionado con los recursos hardware, libre de impuesto, asciende a un total de **ciento setenta y cinco euros (175 €)**.

### 1.1.2. Recursos software

Los recursos *software* utilizados en este TFG son:

- Python (versión 3.10 64-bits)
- Paquete de ofimática *Microsoft Office 2016*

En este caso, el paquete de ofimática es ofrecido por la universidad al pertenecer a esta y, por tanto, no se paga ninguna licencia *Office*. Por otra parte, el lenguaje de código *Python* es un lenguaje de código abierto, por esto no requiere una licencia de pago para su uso. Por todo esto, los costes de amortización del *software* son nulos.



## 1.2. Tarifas basadas en el tiempo empleado

Para calcular los honorarios del ingeniero por su trabajo realizado, se seguirá las recomendaciones, utilizando la siguiente expresión:

$$H = Ct \cdot 74,88 \cdot Hn + Ct \cdot 96,72 \cdot He$$

Donde:

- **H** son los honorarios totales por el tiempo dedicado.
- **Ct** es un factor de corrección función del número de horas trabajadas.
- **Hn** son las horas normales trabajadas (dentro de la jornada laboral).
- **He** son las horas especiales.

Este TFG tiene una duración total de 300 horas, por lo que, para calcular el factor de corrección se ha planteado la siguiente tabla, en función del número de horas empleadas en la elaboración del proyecto.

| Horas empleadas   | Factor de Corrección Ct |
|-------------------|-------------------------|
| Hasta 36          | 1,00                    |
| De 36 hasta 72    | 0,90                    |
| De 72 hasta 108   | 0,80                    |
| De 108 hasta 144  | 0,70                    |
| De 144 hasta 180  | 0,65                    |
| De 180 hasta 360  | 0,60                    |
| De 360 hasta 510  | 0,55                    |
| De 510 hasta 720  | 0,50                    |
| De 720 hasta 1080 | 0,45                    |
| Más de 1080       | 0,40                    |

Tabla 29 Factor de corrección por horas trabajadas.

Como las horas empleadas en el proyecto es de 300 horas, el factor de corrección Ct es de 0,6.

Teniendo en cuenta esto, las horas trabajadas en horario laboral y las horas trabajadas en horario no laboral, los honorarios calculados serán los siguientes:

$$H = 0,6 \cdot 74,88 \cdot 300 + 0,6 \cdot 96,72 \cdot 0 = 13.478,40\text{€}$$

Los honorarios totales, libres de impuesto, derivados del trabajo tarifado por el tiempo empleado, asciende a trece mil cuatrocientos setenta y 8 euros con cuarenta céntimos (13.478,40 €).

### 1.3. Costes relacionados con la elaboración del documento

La fórmula para obtener el importe establecido en la redacción del documento en la siguiente:

$$R = 0,07 \cdot P \cdot Cn$$

Donde:

- **R** son los costes por la redacción del proyecto
- **P** es el presupuesto
- **Cn** es el coeficiente de ponderación en función del presupuesto

Primero, se debe calcular el presupuesto o coste del proyecto, sumando todos los costes calculados hasta el momento:

| Recurso                               | Coste (€)        |
|---------------------------------------|------------------|
| <i>Hardware</i>                       | 175,00           |
| <i>Software</i>                       | 0,00             |
| Tarifas basadas en el Tiempo Empleado | 13.478,40        |
| <b>Total</b>                          | <b>13.653,40</b> |

Tabla 30 Presupuesto por materiales y tiempo empleado.

Calculado el presupuesto y comprobando que es inferior a 30.050,00 €, se puede obtener el coeficiente de ponderación para el presupuesto, para este caso, según lo recomendado, el coeficiente es igual a 1,00. Con esto se podrá obtener R.

$$R = 0.07 \cdot 13.653,40 \cdot 1,00 = 955,74€$$

El coste, libre de impuesto, de la redacción del proyecto, asciende a **novecientos cincuenta y cinco euros con setenta y cuatro céntimos (955,74€)**.

## 1.4. Derechos por el visado el COITT

Para el cálculo de los derechos de visado, para un proyecto o trabajo general, el COITT establece la siguiente fórmula:

$$V = 0,006 \cdot P \cdot Cv$$

Donde:

- V son los gastos de visado.
- P es el presupuesto total del proyecto.
- Cv es el coeficiente reductor según el importe del presupuesto.

Primeramente, como se ha realizado en el punto anterior, se calculará el presupuesto actual del proyecto sumando todos los costes, calculados anteriormente, excluyendo los costes relacionados a material fungible debido a que no se requiere el imprimir este documento:

$$P = 13.653,40 + 955,74 = 14.609,14\text{€}$$

Obtenido el presupuesto total del proyecto, sin impuestos, se comprueba que este no supera los 30.050,00 € y, por tanto, según las recomendaciones, el coeficiente de ponderación será de 1,00. Los gastos de visado serán de:

$$V = 0,006 \cdot 14.609,14 \cdot 1,00 = 87,65\text{€}$$

Los gastos de derivación del derecho del visado, libre de impuestos, serán de **ochenta y siete euros y sesenta y cinco céntimos**.

## 1.5. Costo de tramitación y envío

Los gastos de tramitación y envío están establecidos a **seis euros y un céntimo** (6,01 €), según las recomendaciones.

## 1.6. Impuestos aplicables

Por último, al presupuesto total del proyecto se le debe sumar el 7 % del total, debido al Impuesto General Indirecto Canario (IGIC).

| Coste Total del Proyecto                             |           |
|--|-----------|
| Recurso  | Coste (€) |
| Recursos materiales                                  | 175,00    |
| Tarifas basadas en el Tiempo Empleado                | 13.478,40 |
| Costes relacionados con la Elaboración del Documento | 955,74    |
| Derechos por el Visado del COITT                     | 87,65     |
| Costos de Tramitación y Envío                        | 6,01      |
| <b>Subtotal</b>                                      | 14.702,80 |
| IGIC (7 %)   | 1.029,20  |
| <b>Total</b>   | 15.732,00 |

Tabla 31 Coste total del proyecto

El presupuesto total de este TFG asciende a una cuantía de **quince mil setecientos treinta y dos euros** (15.732 €).

Firma: \_\_\_\_\_

# **PARTE III – PLIEGO DE CONDICIONES**



# 1. Pliego de condiciones

## 1.1. Introducción

En esta parte, se establecerán los requisitos que se han llevado a cabo para la realización del Trabajo de Fin de Grado, especificando los requisitos de *hardware* y los de *software* necesarios.

## 1.2. Requisitos hardware

Primeramente, el ordenador de sobremesa utilizado, para este proyecto, posee las siguientes características técnicas:

- Tipo de dispositivo: Ordenador de sobremesa.
- Procesador: AMD Ryzen 7 3800x (8 núcleos, 390 Ghz).
- Memoria RAM: 16 GB, 3600 MHz DDR4.
- Tarjeta gráfica: Radeon RX 5700 8 GB.
- Almacenamiento: 1 TB.

Por otra parte, se requiere un monitor con las siguientes características:

- Marca: Legion Y25f – 10
- Pulgadas 24,5”
- Frecuencia: 144 Hz

## 1.3. Requisitos software

Para llevar a cabo las simulaciones, programación y redacción del trabajo, se ha establecido las siguientes especificaciones de software:

- Microsoft Windows 10
- Python (versión 3.10 64-bits)
- Paquete de ofimática Microsoft Office 2016

#### **1.4. Recursos humanos**

Para llevar a cabo este proyecto, se ha contado con la tesis doctoral del proyecto base, así como, parte de su código y gran parte de su base de datos. A lo largo del desarrollo del TFG, se ha recibido una excelente tutorización por parte del Dr. Carlos Manuel Travieso Gonzáles, ofreciendo una gran ayuda, consejos y ánimos, durante todo el trayecto de las “Prácticas de empresa” y del TFG.