



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Detección de ejercicios en
vídeos de rehabilitación**



Presentado por Luis Ángel Espinosa Lafuente
en Universidad de Burgos — 9 de julio de 2024

Tutor: José Luis Garrido Labrador y José
Miguel Ramírez Sanz



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. José Luis Garrido Labrador y D. José Miguel Ramírez Sanz, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Luis Ángel Espinosa Lafuente, con DNI ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Detección de ejercicios en vídeos de rehabilitación.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 9 de julio de 2024

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. José Luis Garrido Labrador

D. José Miguel Ramírez Sanz

Resumen

La enfermedad de Parkinson es una enfermedad neurodegenerativa que afecta principalmente a las personas mayores, por lo que es importante que puedan realizar ejercicios de rehabilitación que les permitan retrasar los efectos de la enfermedad lo máximo posible, por ello numerosos estudios buscan ofrecer a los pacientes la posibilidad de realizar los ejercicios de forma telemática.

Este trabajo pretende encontrar una forma de automatizar dicho proceso dando una valoración al usuario de cuan bien se ha realizado el ejercicio comparando ese vídeo con el de un profesional. Para ello se parte de los trabajos de fin de máster de José Miguel Ramírez Sanz y José Luis Garrido Labrador, tutores de este trabajo, en los que se obtenía un esqueleto por cada frame de un vídeo de ejercicios y el trabajo de fin de grado de Lucía Nuñez Calvo en el que se detectaba el inicio y el fin de un ejercicio para poder recortar los vídeos.

Para poder ver los resultados, se desarrollará una aplicación web para que el usuario pueda ver la puntuación de sus ejercicios.

Descriptores

Parkinson, telerehabilitación, secuencias temporales, comparación de vídeos, Dynamic Time Warping, Angular, FastAPI, Docker, investigación.

Abstract

Parkinson's disease is a neurodegenerative disease that mainly affects the elderly, so it is important that they can perform rehabilitation exercises that allow them to delay the effects of the disease as much as possible, so numerous studies seek to offer patients the ability to perform the exercises telematically.

This work aims to find a way to automate this process by giving an assessment to the user of how well the exercise has been performed by comparing this video with that of a professional. For this purpose, we start from the Master's Final Projects of José Miguel Ramírez Sanz and José Luis Garrido Labrador, tutors of this work, in which a skeleton was obtained for each frame of an exercise video and the Degree's Final Project of Lucía Nuñez Calvo in which the beginning and the end of an exercise were detected in order to be able to cut the videos.

To be able to see the results, a web application will be developed so that the user can see the score of his exercises.

Keywords

Parkinson, telerehabilitation, time sequences, video comparison, Dynamic Time Warping, Angular, FastAPI, Docker, research.

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
1. Introducción	1
2. Objetivos del proyecto	5
2.1. Objetivos generales	5
2.2. Objetivos técnicos	5
2.3. Objetivos personales	6
3. Conceptos teóricos	7
3.1. Minería de datos	7
3.2. Ciencia de datos	7
3.3. Visión artificial	8
3.4. Series temporales	9
3.5. Programación dinámica	9
3.6. Distancias	10
3.7. Dynamic Time Warping	10
3.8. Modelos ocultos de Markov	13
4. Técnicas y herramientas	15
4.1. L ^A T _E X	15
4.2. TeXstudio	16
4.3. Control de versiones	16

4.4. Visual Studio Code	16
4.5. Detectron2	16
4.6. Python	17
4.7. Jupyter Notebook	17
4.8. Librerías de Python	17
4.9. Tecnologías para desarrollar la aplicación	17
5. Aspectos relevantes del desarrollo del proyecto	21
5.1. Inicio del proyecto	21
5.2. Investigación de métodos para poder puntuar el ejercicio . .	22
5.3. Investigación sobre las series temporales	23
5.4. Preparación de los datos	24
5.5. Obtención de una distancia	26
5.6. Puntuar un ejercicio	28
5.7. Otras pruebas	28
5.8. Desarrollo de la aplicación	29
6. Trabajos relacionados	33
6.1. Dance Performance Evaluation Using Hidden Markov Models [10]	33
6.2. A Dynamic Time Warping Based Algorithm to Evaluate Kinect-Enabled Home-Based Physical Rehabilitation Exerci- ses for Older People [25]	33
6.3. A Wireless Evaluation System for In-home Physical Rehabili- tation [26]	34
7. Conclusiones y Líneas de trabajo futuras	35
7.1. Conclusiones	35
7.2. Lineas futuras	36
Bibliografía	37

Índice de figuras

3.1. Alineamiento de dos secuencias mediante DTW	11
5.1. Secuencias de un ángulo normalizado del paciente y del profesional.	23
5.2. Representación del esqueleto obtenido a partir de los puntos grabados.	24
5.3. Comparación de dos secuencias sin normalizar.	25
5.4. Comando para poder ver un vídeo no compatible con navegadores en ellos	31
5.5. Comparación de ejercicio.	32
5.6. Vista de nuevo ejercicio.	32

Índice de tablas

4.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	15
5.1. Distancias obtenidas usando todas las partes y todos los ángulos de los pacientes y profesionales.	26
5.2. Distancias obtenidas con solo las partes importantes para ese ejercicio de pacientes y profesionales.	27
5.3. Puntuaciones obtenidas de las posiciones de profesionales y pacientes.	28
5.4. Comparación de puntuación obtenida redimensionando cada una de las partes y de la puntuación calculando cada dimensión por separado.	29
5.5. Puntuaciones obtenida de series multivariantes redimensionando las dimensiones X e Y y sin redimensionarlas.	30

1. Introducción

La enfermedad de Parkinson es una enfermedad neurodegenerativa crónica que no tiene cura actualmente. Esta enfermedad causa síntomas motores como pueden ser la realización de movimientos involuntarios, temblores o la bradicinesia¹, pero también provoca síntomas no motores como son el deterioro cognitivo, los trastornos mentales y la demencia.

La enfermedad tiene un riesgo de incidencia distinto en función del sexo y la edad, afectando en el caso del primero más a los hombres que a las mujeres. En el caso de la edad, la mayoría de personas a las que afecta son de más de 60 años, mientras que entre un 5 y un 10 % de los casos se inicia antes de los 50 años [5].

Esta enfermedad, según estimaciones de 2019 de la Organización Mundial de la Salud afecto a 8,5 millones de personas y causo 329000 muertes desde 2000 [3].

Uno de los tratamientos para esta enfermedad es la fisioterapia, que ayuda a retrasar el avance de la enfermedad, para ello, se realizan ejercicios de rehabilitación moviendo distintas partes del cuerpo como pueden ser articulaciones. Estos ejercicios son evaluados por un profesional que busca fallos en la ejecución de dichos ejercicios.

Estas sesiones típicamente son presenciales, lo que puede ser un problema para los pacientes, que suelen ser personas mayores y con problemas de movilidad, por lo que es necesario que sus familias se adapten a trasladar al paciente al centro de salud en el que vaya a realizar los ejercicios, que en el caso de las zonas rurales puede estar a varios kilómetros, haciendo imposible el traslado andando. Además, dependen de la disponibilidad de

¹La bradicinesia consiste en la ralentización del movimiento o velocidad [4].

un terapeuta. Esto puede ser un problema debido a la falta de personal sanitario, aun más en el caso de las zonas rurales. Además, con el aumento de la esperanza de vida, cada vez habrá más personas con Parkinson, ya que afecta más a las personas mayores, por lo que puede agravarse más la situación [6].

Una solución a esto es la telerehabilitación, que consiste en que el paciente se graba haciendo los ejercicios y se realiza un análisis sobre ellos. Esto hace que ya no sea necesario que los pacientes se tengan que desplazar ni que dependan de nadie para que les lleve, simplemente tendrían que conectarse a la app.

Este trabajo pretende encontrar una forma de automatizar dicho proceso dando una valoración al usuario de cuan bien se ha realizado el ejercicio comparando ese vídeo con el de un profesional.

Por ultimo, para poder ver los resultados, se desarrollará una aplicación web para que el usuario pueda ver la puntuación de sus ejercicios.

Este trabajo hubiera sido imposible de realizar sin los resultados que se obtuvieron de los trabajos de fin de máster de José Miguel Ramírez Sanz y José Luis Garrido Labrador, tutores de este trabajo y del trabajo de fin de grado de Lucía Nuñez Calvo, gracias a ellos se pudo contar con ficheros de datos con las posiciones extraídas de los vídeos recortados por ejercicio y de los ángulos calculados a partir de dichas posiciones.

Estructura de la memoria

La presente memoria se compone de las siguiente secciones:

1. **Introducción:** Descripción del Parkinson y breve introducción al trabajo.
2. **Objetivos del proyecto:** Se muestran los objetivos funcionales, técnicos y personales que motivan el desarrollo del proyecto.
3. **Conceptos teóricos:** Se muestran definiciones y explicaciones de los distintos conocimientos necesarios para el entendimiento de este trabajo.
4. **Técnicas y herramientas:** Conjunto de herramientas y metodologías elegidas para el desarrollo del proyecto.

5. **Aspectos relevantes del desarrollo:** Explicación de las etapas más importantes del proyecto y problemas que surgieron durante el avance del mismo.
6. **Trabajos relacionados:** Muestra trabajos y proyectos similares a este.
7. **Conclusiones y Líneas de trabajo futuras:** Conclusiones extraídas durante el proyecto y posibles áreas de mejora.

Estructura de los anexos

1. **Plan de proyecto:** Descripción de los distintos sprints y estudio de viabilidades económicas y legales del proyecto.
2. **Especificación de los requisitos:** Requisitos y casos de uso del proyecto.
3. **Especificación de diseño:** Conjunto de diseños realizados para la correcta implementación del código.
4. **Documentación técnica de programación:** Descripción del repositorio, así como las fases de instalación y ejecución del proyecto.
5. **Documentación de usuario:** Los manuales de usuario de la aplicación desarrollada.

2. Objetivos del proyecto

2.1. Objetivos generales

- Investigar sobre la comparación de secuencias temporales.
- Investigar distintos métodos de normalización para poder preparar los datos.
- Investigar sobre las distintas formas de evaluar la ejecución de ejercicios.
- Desarrollar una aplicación con la que el usuario pueda comparar ejercicios de forma sencilla.

2.2. Objetivos técnicos

- Desarrollar un algoritmo en *Python* que permita comparar dos ejercicios.
- Utilizar \LaTeX *TexStudio* para escribir la memoria.
- Utilizar un sistema de control de versiones como es *Git* utilizando *Github* para guardar el proyecto en remoto.
- Utilizar *Jupyter Notebook* para mostrar los resultados de la investigación.
- Utilizar metodologías ágiles y marcos de trabajo durante el desarrollo como *Scrum*.

2.3. Objetivos personales

- Realizar un aporte que pueda ayudar a personas con *Parkinson*.
- Poner en práctica varios de los conocimientos adquiridos durante la carrera.
- Introducirme en el desarrollo web aprendiendo a crear interfaces gráficas agradables para el usuario.
- Aprender sobre *Docker* y *Nginx*.
- Aprender tecnologías relacionadas con el desarrollo del *frontend* aplicaciones web como son los lenguajes *Typescript*, *HTML* y *CSS* y el *framework* Angular.
- Aprender tecnologías relacionadas con el desarrollo del *backend* aplicaciones web como son *FastAPI* y *SQLAlchemy*.

3. Conceptos teóricos

En esta sección se expondrán diversos conceptos imprescindibles para la comprensión del proyecto.

3.1. Minería de datos

La minería de datos es un proceso que se utiliza para descubrir patrones, relaciones y tendencias significativas en grandes volúmenes de datos. Para ello se usan técnicas de aprendizaje automático como puede ser la inteligencia artificial.

3.2. Ciencia de datos

La ciencia de datos es un campo académico que trata de recolectar, procesar, analizar e interpretar estos datos para extraer información valiosa y presentar resultados que puedan ser utilizados para tomar decisiones.

Para ello, previamente se deben tratar los datos mediante técnicas como las que se expondrán a continuación.

Normalización

La normalización en el contexto de la ciencia de datos es el proceso para escalar los datos y transformarlos de forma que estén en un rango común.

Normalización Min-Max

La normalización min-max es un método que permite estandarizar valores de forma que estén entre el 0 y el 1.

$$N_i = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad (3.1)$$

Normalización Z-score

Es una técnica para rescalar los valores de forma que la media de los datos sea 0 y la desviación estándar sea 1.

$$N_i = \frac{X_i - \mu}{\sigma} \quad (3.2)$$

Normalización L2

La normalización L2 es un método que se utiliza para transformar los puntos en un vector unitario. La suma de los cuadrados de las posiciones normalizadas sera igual a 1.

$$||x||_2 = \sqrt{\sum_{i=1}^n x^2} \quad (3.3)$$

Traslación

La traslación consiste en aplicar movimientos directos sin cambios de orientación manteniendo la forma y el tamaño de las figuras u objetos [22]. En el caso de la traslación en figuras, se cumplen las siguientes características:

1. La figura trasladada es idéntica a la figura inicial.
2. La figura trasladada conserva la orientación de la figura original.

En el caso de este trabajo, se trasladara el esqueleto formado por el conjunto de partes del cuerpo, trasladando los puntos de las posiciones que lo forman.

3.3. Visión artificial

La visión artificial intenta emular la capacidad de algunos seres vivos para ver una escena, entenderla y actuar en consecuencia [8]. Gracias a la visión artificial es posible extraer las poses de los vídeos de ejercicios.

Segmentación

La segmentación es el proceso por el cual una imagen se divide en distintas partes que comparten alguna característica, lo que permite obtener diferentes objetos de la escena. Esto es vital para poder extraer las poses de los vídeos correctamente.

Estimación de poses

La estimación de poses es la obtención y clasificación de la configuración espacial de las partes del cuerpo en imágenes y vídeos. En el caso de este proyecto se usaron esqueletos en los que se obtuvieron previamente aplicando esta técnica.

Estos esqueletos están formados por distintos puntos, los cuales eran bidimensionales. A partir de estos además se habían calculado los ángulos, que eran unidimensionales.

3.4. Series temporales

Una serie temporal esta compuesta por datos ordenados y equidistantes cronológicamente sobre una (serie univariante) o más (serie multivariante) características [12].

En el caso de este proyecto, las series temporales con las que se va a trabajar son las secuencias de cada punto y de cada ángulo obtenidos de los fotogramas de los vídeos. Cuando se tratan las series por tipo de ángulo, son series univariantes, sin embargo, si se comparan todas a la vez seria multivariante. En el caso de las posiciones, al ser de dos dimensiones, son multivariantes, pero se pueden tratar las dimensiones de forma separa o incluso redimensionarlos, de forma que se pueda trabajar con ellas como univariantes.

3.5. Programación dinámica

La programación dinámica consiste en la división de problemas en subproblemas resolviendo cada problema una sola vez y guardando los resultados para evitar la redundancia [7]. Este método sigue el principio de optimalidad de Bellmean, que se define a continuación:

Principio de optimalidad: Una política óptima tiene la propiedad de que cualquiera que sea el estado inicial y la decisión inicial, las decisiones restantes deben constituir una política óptima en relación con el estado resultante de la primera decisión [2].

Un ejemplo de esta técnica sería el método de alineamiento de series temporales que usa el Dynamic Time Warping, que se explicará en la sección 3.7.

3.6. Distancias

De forma general la distancia entre dos puntos se define como el camino más corto entre dos puntos. Hay diversos métodos de calcular esta distancia, entre ellos se expondrán los que se han usado en este proyecto.

Distancia euclidiana

La distancia euclidiana es la menor distancia entre dos puntos en un espacio de N dimensiones (espacio euclidiano) [9].

$$d_{x,y} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.4)$$

Similitud coseno

La similitud coseno permite diferenciar dos vectores, se utiliza para búsqueda y recuperación de información y comparación de documentos. Es uno de los métodos para diferenciar dos posturas de dos imágenes concretas por medio de la similitud de los vectores de las distintas partes del cuerpo.

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|} \quad (3.5)$$

Si el valor de este fuera 1 es que la imagen sería igual, mientras que si fuese 0 serían ortogonales, es decir que no comparten ninguna similitud.

3.7. Dynamic Time Warping

La deformación dinámica o Dynamic Time Warping en inglés permite acoplar dos series temporales que no tienen porque tener ni el mismo tamaño

ni la misma forma, en el caso de este proyecto secuencias de movimientos a distintas velocidades. Este algoritmo es muy utilizado en reconocimiento de audio.

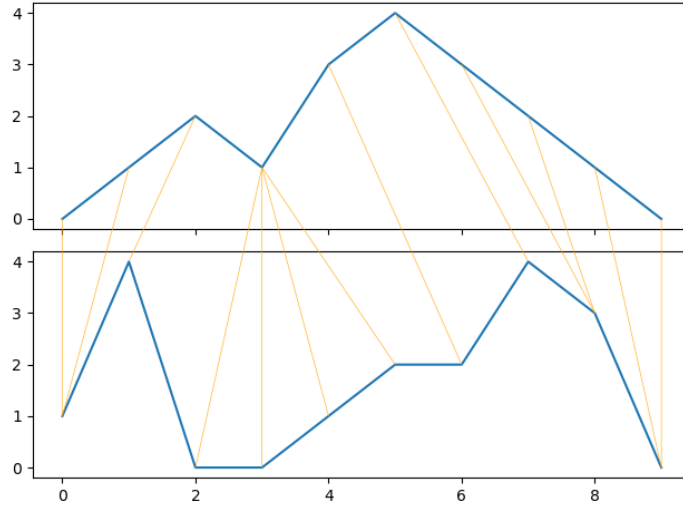


Figura 3.1: Alineamiento de dos secuencias mediante DTW

Dadas dos señales $X := (x_1, x_2, \dots, x_N)$ e $Y := (y_1, y_2, \dots, y_M)$, el objetivo de DTW será encontrar la alineación óptima devolviendo la distancia entre ellas. Para ello, como se explicara de forma detallada en las siguientes subsecciones, se calculan dos matrices, una de costes locales y una de costes acumulados y después se obtiene la ruta óptima.

Matriz de costes locales

Esta técnica se realiza mediante la comparación de las distancias de todos los pares de puntos en dos secuencias. Una distancia menor implica que estos puntos pueden ser candidatos a ser emparejados [16]. La formula que define esta matriz es la siguiente:

$$C(i, j) := c(x_i, y_j) \quad (3.6)$$

$$i \in [1 : N]$$

$$j \in [1 : M]$$

Para esta matriz se debe elegir una medida de distancia entre dos puntos, en el caso de este trabajo se escogió la distancia euclidiana.

Matriz de costes acumulados

Una vez emparejados los puntos se usa una matriz de costes acumulados $D_{N \times M}$ [19, 25]. En esta parte se usa la programación dinámica para calcular la distancia DTW. En la matriz de costes acumulados se inicializan los valores mediante las siguientes formulas:

1. La primera fila:

$$D(1, j) = \sum_{k=1}^j C(1, k) \quad (3.7)$$

$$j \in [1 : M]$$

2. La primera columna:

$$D(i, 1) = \sum_{k=1}^i C(k, 1) \quad (3.8)$$

$$i \in [1 : N]$$

3. El resto aplicando la siguiente formula de forma recursiva:

$$D(i, j) = \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\} + C(i, j) \quad (3.9)$$

$$i \in [2 : N]$$

$$j \in [2 : M]$$

Donde C es la matriz de costes locales. Una vez aplicadas las formulas, la distancia acumulada será $D(N, M)$.

Camino de deformación

El camino de deformación consiste en la alineación entre dos secuencias, este camino debe cumplir las siguientes condiciones [13]:

1. Monotonicidad: Los puntos deben estar ordenados con respecto al tiempo.

$$x_{k-1} \leq x_k \quad \& \quad y_{k-1} \leq y_k \quad (3.10)$$

2. Continuidad: El siguiente punto debe ser vecino del anterior.

$$x_k - x_{k-1} \leq 1 \quad \& \quad y_k - y_{k-1} \leq 1 \quad (3.11)$$

3. Condiciones de frontera: Los puntos de inicio y fin deben ambas señales son emparejados entre ellos.

$$x_1 = 1, y_1 = 1 \quad \& \quad x_k = n, y_k = m \quad (3.12)$$

Ruta óptima

DTW permite la extracción del camino óptimo a través de la matriz de costes acumulados mediante el uso de un algoritmo voraz. Realizando tres tipos de movimientos desde $D(N, M)$ hasta $D(0,0)$ utilizando backtracking:

1. **Movimientos horizontales:** $D(i, j) \implies D(i, j - 1)$
2. **Movimientos verticales:** $D(i, j) \implies D(i - 1, j)$
3. **Movimientos diagonales:** $D(i, j) \implies D(i - 1, j - 1)$

Se elegirá cada movimiento en función del que lleve a una posición con el menor coste.

3.8. Modelos ocultos de Markov

Un modelo oculto de Markov, Hidden Markov Model en inglés con el acrónimo *HMM* es un modelo estadístico que consta de una cadena de Markov no observada y una cadena observada la cual donde la cadena observada depende probabilísticamente de la cadena no observada. Estas cadenas están formadas por estados, el objetivo es conocer el valor del estado no observado mediante el valor observado. Los modelos ocultos de Markov tienen cinco elementos [11]:

1. Un conjunto de estados ocultos $\{S_1, S_2, \dots, S_N\}$.
2. Un conjunto de observaciones posibles $\{O_1, O_2, \dots, O_T\}$.
3. Una matriz de transición $A = [a_{ij}]$, donde a_{ij} es la probabilidad de transitar del estado S_i al estado S_j .
4. Una matriz de emisión $B = [b_j(k)]$, donde $b_j(k)$ es la probabilidad de observar O_k dado el estado S_j .
5. Una distribución inicial $\pi = [\pi_i]$, donde π_i es la probabilidad de que el estado inicial sea S_i .

Algoritmo de Viterbi

Dado un modelo oculto de Markov con un conjunto de estados ocultos $S = \{S_1, S_2, \dots, S_N\}$ y una secuencia de T observaciones $O = \{O_1, O_2, \dots, O_T\}$, el algoritmo de Viterbi encontrará la secuencia de estados que tiene la mayor probabilidad de haber generado esas observaciones. Este algoritmo consiste en [14, 15]:

Inicialización

Para $1 \leq i \leq N$:

$$\delta_1(i) = \pi_i \cdot b_i(O_1) \quad (3.13)$$

$$\psi_1(i) = 0 \quad (3.14)$$

Donde $\delta_t(i)$ es la probabilidad máxima de la secuencia de estados que termina en S_i en el tiempo t y $\psi_t(i)$ es el estado que maximiza $\delta_t(i)$.

Recursión

Para $2 \leq t \leq T$ y $1 \leq j \leq N$:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) \cdot a_{ij}] \cdot b_j(O_t) \quad (3.15)$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) \cdot a_{ij}] \quad (3.16)$$

Donde $\delta_t(j)$ se actualiza con la máxima probabilidad de llegar al estado S_j en el tiempo t considerando todas las posibles transiciones desde el tiempo $t - 1$.

Terminación

$$P^* = \max_{1 \leq i \leq N} \delta_T(i) \quad (3.17)$$

$$q_T^* = \arg \max_{1 \leq i \leq N} \delta_T(i) \quad (3.18)$$

Donde P^* es la probabilidad de la secuencia de estados más probable y q_T^* es el último estado de esa secuencia.

Rastreo hacia atrás

Para $t = T - 1, T - 2, \dots, 1$:

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \quad (3.19)$$

Esta etapa se utiliza para reconstruir la secuencia de estados óptima a partir de los valores almacenados en ψ .

4. Técnicas y herramientas

En este apartado se explicarán las herramientas utilizadas para el trabajo.

4.1. \LaTeX

\LaTeX es un sistema de composición de textos orientado a la creación de documentos escritos que presenten una alta calidad tipográfica [21].

Herramientas	Pruebas	App	Memoria
Git + GitHub + Gittyup	X	X	X
\LaTeX + TexStudio			X
Zube	X	X	X
Visual Studio Code	X	X	X
Python	X	X	
Jupyter	X		
FastAPI + Pydantic		X	
Dtaidistance	X	X	
Pandas + Numpy	X	X	
Angular + Angular Material		X	
Nginx		X	
Docker		X	
PostgreSQL		X	
SQLAlchemy		X	

Tabla 4.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

4.2. TeXstudio

TeXstudio es un editor de L^AT_EX de código abierto y multiplataforma [23]. Ofrece la posibilidad de escribir la memoria de forma local fácilmente. También se contempló la alternativa de usar Overleaf, ya que no requiere instalación, pero finalmente se eligió por TeXstudio ya que al simplificaba el no tener que descargar el documento cada vez que se quisiera actualizar la versión en el repositorio de Github.

4.3. Control de versiones

- *Git* es un programa de control de versiones. Muy útil para poder ver los cambios que se van realizando a medida que avanza el proyecto así como volver a versiones anteriores para deshacer cambios.
- *GitHub* es una plataforma online que utiliza Git para guardar repositorios y ajustar la visibilidad de los mismos. También permite reportar errores en la sección *Issues* y crear bifurcaciones del código mediante la creación de *forks*.
- *Gittyup* permite utilizar Git a través de una interfaz gráfica, es una alternativa a programas como Fork que no están disponibles en Linux. Además, es de código abierto.
- *Zube* es una página web dirigida a la planificación de proyectos utilizando metodologías ágiles como el marco de trabajo *Scrum* y *Kanban*.

4.4. Visual Studio Code

Visual Studio Code (también llamado VS Code) es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web [24]. Además, permite multitud de lenguajes gracias al uso de extensiones.

4.5. Detectron2

Detectron2 es una librería de Python destinada a la detección y segmentación de imágenes. Algo que se observó es que la última versión que viene compilada es la versión 0.6, que salió por última vez en 2021.

4.6. Python

Python es un lenguaje de programación de alto nivel, multiplataforma y de código abierto que usa tipado dinámico.

4.7. Jupyter Notebook

Los cuadernos de *Jupyter* pueden contener en un mismo archivo texto, código, los resultados de la ejecución de ese código, material multimedia y ecuaciones, lo que les brinda de una gran versatilidad y razón por la cual se suelen usar en ámbitos de investigación.

4.8. Librerías de Python

- *Pandas* es una librería de código abierto para análisis y manipulación de datos [1]. Permite trabajar los datos mediante unas clases llamadas *DataFrames*, que además se pueden guardar de forma sencilla.
- *Numpy* es una librería que facilita el trabajar con matrices en Python, facilitando realizar operaciones sobre ellas.
- *Dtaidistance* es una librería que implementa distintas funciones relacionadas con DTW. Ofrece tanto implementaciones en Python como en C, siendo esta segunda mucho más rápida. Para usar las versiones de C, basta con añadir *fast* a los nombres de los metodos y usar *arrays* de la librería Numpy
- *FastDTW* es una librería de Python que implementa una aproximación de DTW que tiene una complejidad lineal en tiempo y espacio [18]. La librería no se ha actualizado desde hace tiempo y cambios en las dependencias provocaban errores por lo que se descarto su uso.

4.9. Tecnologías para desarrollar la aplicación

Se decidió realizar una aplicación web, en la que se divide el *backend* del *frontend* y se comunican mediante el uso de una API. Esto ofrece ventajas como pueden ser:

- **Accesibilidad:** el usuario no tendría que instalar nada, simplemente buscar la url en un navegador, por lo que no necesita tener conocimientos avanzados de informática.
- **Compatibilidad:** El usuario podrá acceder a la aplicación independientemente de la aplicación, siempre que tenga un navegador compatible.

Angular

Angular es un framework de código abierto desarrollado por Google que permite el desarrollo de *frontend* de forma sencilla, usa una arquitectura basada en componentes, lo que permite crear elementos encapsulados reutilizables. Usa Typescript, HTML y SCSS.

Angular Material

Angular Material es una librería de Angular que permite gestionar aspectos globales de la interfaz como son las paletas de colores y las fuentes. También ofrece una gran variedad de componentes para poder utilizar.

FastAPI

FastAPI es un framework que permite crear APIs en Python para el backend de aplicaciones web.

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacionales de código abierto.

SQLAlchemy

SQLAlchemy es una librería de Python para poder realizar consultas mediante el Mapeo Objeto-Relacional (ORM), a través de la creación de clases en Python que representan las tablas y permite utilizar métodos para obtener la información y administrarla.

Pydantic

Pydantic es una librería de Python que permite la gestión de modelos mediante el uso del patrón Data Transfer Object (DTO).

Docker

Docker permite la creación de contenedores aislados, y despliegue de los mismos no haciendo falta instalar en el sistema anfitrión las dependencias del software que se va a ejecutar en los contenedores, ya que se instala en los mismos de forma automatizada.

Nginx

Nginx es un servidor web de código abierto, se usara para ejecutar el *frontend* de Angular de la aplicación.

5. Aspectos relevantes del desarrollo del proyecto

En este apartado se va a comentar el desarrollo que ha tenido el proyecto. Se expondrán todas las líneas de investigación seguidas así como las opciones y las decisiones que se tomaron para resolver estos problemas.

5.1. Inicio del proyecto

Este proyecto parte del trabajos de fin de máster de José Miguel Ramírez Sanz y José Luis Garrido Labrador, tutores de este trabajo, en los que se obtenía un esqueleto por cada *fotograma* de un vídeo de ejercicios y el trabajo de fin de grado de Lucía Nuñez Calvo en el que se detectaba el inicio y el fin de un ejercicio para poder recortar los vídeos. El conjunto de proyectos, realizados en colaboración con el Hospital Universitario, tienen como objetivo poder mejorar la calidad de vida de los pacientes que padecen enfermedad de *Parkinson* mediante la implementación de un sistema que les permita realizar ejercicios cómodamente desde su hogar y si es posible sin la necesidad de la presencia continua de un terapeuta, haciendo así estas sesiones más frecuentes.

Inicialmente se contemplaron dos temas para el trabajo, la mejora en el sistema de detección de poses y la obtención de una puntuación de los ejercicios realizado por los pacientes comparándolos con el vídeo de un terapeuta, se optó por esta ultima, ya que sin una puntuación que permita saber a los pacientes como de bien han realizado el ejercicio es necesario para que el sistema de telerehabilitación sea viable.

En las primeras semanas se investigó sobre *Detectron 2*, la forma en la que este extraía las poses de los vídeos. Más adelante descubrimos que este proceso no era perfecto, ya que en muchos casos o faltaban posiciones o había demasiado errores en la extracción de estos como que los esqueletos hicieran movimientos imposibles. También se aprendió a utilizar *Docker* ya que es una herramienta con la que no tenía experiencia.

5.2. Investigación de métodos para poder puntuar el ejercicio

En todo proyecto de investigación, el primer paso es realizar un estado del arte para saber que técnicas aplicar y cómo hacerlo. En el caso de este trabajo era vital el poder establecer que era importante a la hora de obtener una puntuación. Inicialmente se investigó cómo conseguían este objetivo juegos de baile como pueden ser *Just Dance* o *Dance Central*.

En la primera investigación no se encontró información oficial por parte de los creadores de dichos juegos excepto un artículo de *Ubisoft* que mencionaba que en juegos anteriores a *Just Dance 2023* bastaba con agitar el mando para conseguir la máxima puntuación [20], por lo que una posible solución que se contempló al principio fue detectar cambios en la posición para puntuarlo, pero se descartó rápidamente ya que esto no aseguraba que el usuario hubiera realizado correctamente el ejercicio.

Continuando con la investigación se encontró otra posible forma de obtener una puntuación que era mediante el uso de la distancia coseno para comparar los vectores formados por los puntos del esqueleto. Pero esto presentaba otro problema, que se vio al principio del proyecto, la enfermedad de *Parkinson* provoca problemas de movilidad, por lo que, es más probable que la ejecución de los ejercicios sea más rápida en el caso de un fisioterapeuta que en el caso de un paciente. En la figura 5.1 se puede observar esta diferencia, la secuencia del profesional es mucho más corta que la del paciente, lo que complica bastante la comparación.

Los juegos de baile suelen permitir cierto margen de error al ejecutar un movimiento en un momento distinto. Esto se podría conseguir comparando cada *frame* del usuario con todos los *frames* de una ventana de tiempo del vídeo modelo, pero al ser un problema de velocidad y no de haber ejecutado un movimiento de baile a un ritmo incorrecto, haría que el retraso de los movimientos se fuera acumulando, por lo que no resuelve el problema.

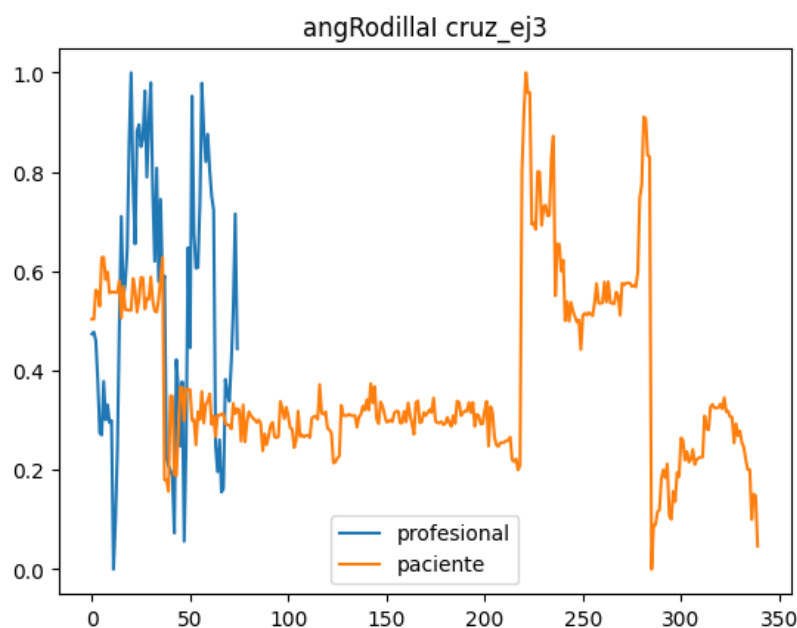


Figura 5.1: Secuencias de un ángulo normalizado del paciente y del profesional.

Esto nos llevó a replantearnos las condiciones que debía cumplir un ejercicio correcto. Para que un ejercicio fuera correcto, el paciente tendría que realizar las poses correctamente, pero no es necesario que realice el ejercicio a la misma velocidad, lo que complica bastante la obtención de esa puntuación.

5.3. Investigación sobre las series temporales

Con los nuevos criterios obtenidos se estudió que métodos se podían usar para comparar secuencias de movimientos a distintas velocidades, esto llevó a investigar sobre las series temporales.

Se encontraron dos posibles soluciones, los modelos ocultos de Markov (HMM) y el Dynamic Time Warping (DTW) [17]. En el caso de los HMM tienen la ventaja de que podrían permitir que, en el caso de que se encontrara una forma de obtener las poses en tiempo real, se podrían usar para dar *feedback* al usuario en tiempo real, además de que con grandes sets de datos ofrecen un mejor rendimiento que DTW. El problema que conllevaría

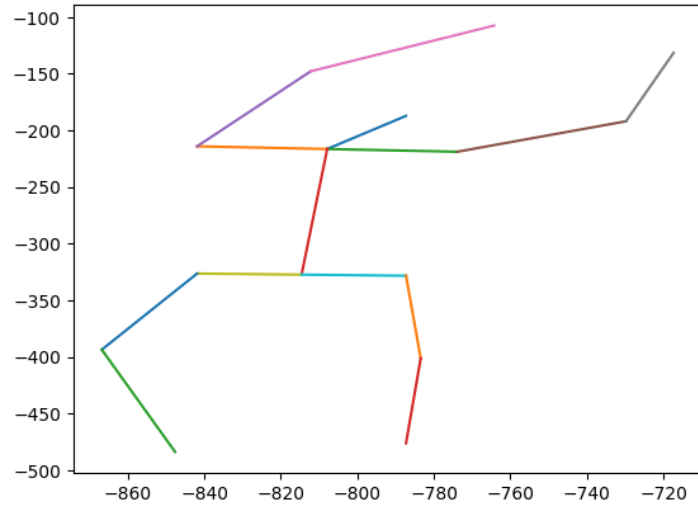


Figura 5.2: Representación del esqueleto obtenido a partir de los puntos grabados.

haberlas usado en lugar de DTW es precisamente la cantidad de datos, en el caso de este proyecto se contó con un set de datos muy pequeño, por lo que elegimos DTW, que se podría utilizar incluso aunque solo se tuviera un vídeo modelo y un vídeo de un paciente.

5.4. Preparación de los datos

Una vez elegido el método que se usaría para comparar los vídeos, había que preparar los datos que se iban a usar para las pruebas. Estos datos se encontraran en dos tipos de ficheros, archivos CSV y *DataFrames*. Dos motivos para ello eran la protección de datos y el gran coste de computo que es obtener las posiciones del esqueleto en un vídeo. Estos ficheros contenían tanto un número que indicaba el numero de fotograma como los datos de las posiciones.

En este caso conviene distinguir de dos tipos de datos, ángulos y coordenadas, los primeros son de una dimensión y las segundas de dos. Dependiendo del tipo de ejercicio se movían unas partes de determinada manera, este proyecto trabajo con 6 tipos de ejercicios distintos, 4 en los que se ejercitaban las piernas y 2 en los que se ejercitaban los brazos. El primer paso fue eliminar los fotogramas que tuvieran alguna posición o ángulo nulo. Para los ejemplos que se mostrarán, se han escogido vídeos de referencia de profesionales y

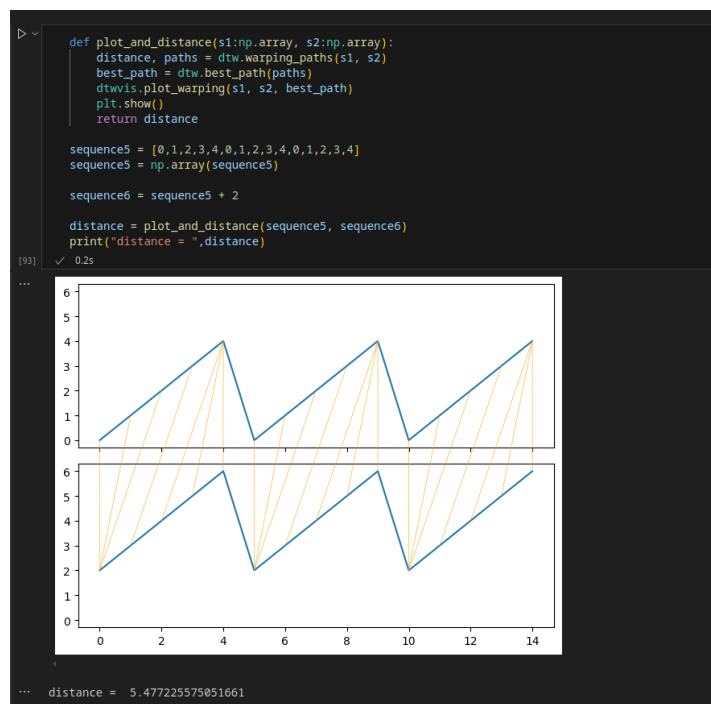


Figura 5.3: Comparación de dos secuencias sin normalizar.

se han comparado con los de otros profesionales y con los de los pacientes, siendo el resultado esperado el que los ejercicios de los profesionales deberían dar distancias menores y las de los pacientes distancias mayores.

No se puede utilizar DTW sin normalizarlos, ya que incluso aunque el ejercicio fuera exactamente igual, la distancia sería distinta de 0, como puede observarse en la figura 5.3. Algunas de estas posiciones contenían datos nulos, por lo que lo primero que se hizo fue descartar los nulos, ya que causaban problemas tanto en las normalizaciones que se explicarán más adelante como en el cálculo de la distancia DTW.

Normalización de ángulos previa

En el caso de los ángulos inicialmente se intentó dividir cada ángulo entre 180° , pero finalmente se utilizó la normalización *min-max* para que estuvieran en un rango entre 0 y 1, ya que con los grados no se obtenían los resultados deseados.

Tipo	Distancia de posiciones	Distancia de ángulos
Profesional	0,1763	1,7774
Profesional	0,2968	2,3090
Profesional	0,2353	1,6997
Paciente	0,5595	2,3495
Paciente	0,5366	2,1405
Paciente	0,5356	2,9488
Paciente	0,5122	2,6827
Paciente	0,4850	3,1227
Paciente	0,4722	3,3067

Tabla 5.1: Distancias obtenidas usando todas las partes y todos los ángulos de los pacientes y profesionales.

Normalización de coordenadas previa

En el caso de los puntos del cuerpo era necesario normalizarlos ya que no todos los fotogramas se encontraban centrados, por lo que era necesario aplicar traslación.

Además, había que garantizar que la traslación no generara que los números fueran negativos para que DTW diera unos resultados aceptables, por ello se aplicó traslación a cada punto mediante el uso de una caja delimitadora por cada esqueleto de cada fotogramas de forma que todos los puntos estén en las cajas y traslada haciendo que el vértice con menores coordenadas sea el $(0, 0)$.

También había otro problema, no todas las personas miden lo mismo, por lo que había que rescalar los datos. Para ello se usó la normalización L2 en cada secuencia de puntos de una parte del cuerpo, normalización explicada en [3.2](#).

5.5. Obtención de una distancia

Una vez que se tenían los datos normalizados era necesario obtener la distancia DTW. En el caso de los ángulos, al ser de una sola dimensión se aplicó DTW directamente, para ello se calculó las distancias de cada ángulo y se hizo la media de estas. En el caso de las coordenadas, que eran de dos dimensiones, se redimensionaron a vectores de una dimensión.

Como inicialmente no se disponían de los datos del terapeuta, se probó a comparar las distancias obtenidas mediante la aplicación de DTW sobre

Tipo	Distancia de posiciones	Distancia de ángulos
Profesional	0,1816	1,2235
Profesional	0,3139	1,8826
Profesional	0,2665	1,4767
Paciente	0,5085	1,1993
Paciente	0,4874	2,1663
Paciente	0,4929	1,3368
Paciente	0,4746	2,2297
Paciente	0,4425	1,9711
Paciente	0,4283	2,9705

Tabla 5.2: Distancias obtenidas con solo las partes importantes para ese ejercicio de pacientes y profesionales.

ejercicios de un mismo tipo con los de distintos ejercicios. Los resultados no fueron los esperados, lo cual puede ser porque en varios de los ejercicios los movimientos son similares y como inicialmente no se filtraban que partes del cuerpo eran importantes para cada tipo de ejercicio, los resultados podían no ser fiables, por ejemplo, en dos ejercicios distintos en los que el paciente esta sentado mientras mueve una pelota con los brazo de distinta manera, comparar las piernas no aportaría información relevante.

Una vez obtenidos los vídeos del terapeuta, se calcularon las distancias, los resultados son los de la tabla 5.1. Como se puede observar, las distancias de los ángulos son más altas que las calculadas a través de las posiciones. Se probó con otras normalizaciones como la *Z-score* y dividirlo entre 180° , pero los resultados seguían sin ser aceptables por la misma razón.

Algo que hay que destacar es que en el caso de las distancias de las posiciones de los profesionales, todas ellas son inferiores a las de los pacientes, lo que tiene sentido, ya que una persona sana normalmente va a ejecutar el ejercicio mejor que una con *Parkinson*.

Después, se tuvo en cuenta la posibilidad de que algunas zonas del cuerpo no se ejercitaran en algún ejercicio, por lo que se filtro que partes eran las importantes, para ello se observó cada vídeo y se guardaron los nombres de las partes ejercitadas. Un ejemplo de los resultados de seleccionar solo las partes importantes se pueden ver en la tabla 5.2.

Tipo	Puntuación de posiciones
Profesional	81,8400
Profesional	68,6029
Profesional	73,3486
Paciente	49,1500
Paciente	51,2527
Paciente	50,7087
Paciente	52,5366
Paciente	55,7411
Paciente	57,1680

Tabla 5.3: Puntuaciones obtenidas de las posiciones de profesionales y pacientes.

5.6. Puntuar un ejercicio

Una vez obtenida la distancia, es necesario obtener la puntuación, la cuál será opuesta a la distancia, cuanto mejor sea la ejecución del ejercicio menor será la distancia y más alta la puntuación. El problema es que para ello, hay que encontrar la mayor distancia de cada caso.

En el caso de los ángulos no fue posible, ya que no se conseguía que la distancia estuviera en un rango, por lo que los resultados de ejercicios mal ejecutados eran negativos debido a que la distancia era muy alta.

En el caso de las posiciones, fue sencillo, se aplicó a cada distancia la formula 5.20, esto permite que la distancia 0 tenga una puntuación de 100.

$$puntuación = 100 - 100 * distancia \quad (5.20)$$

Y por último se calculaba la media de ellas. Los resultados se pueden ver en la tabla 5.3. Hay que tener en cuenta de que en el caso de que la media de las distancias fuera mayor que 1 la puntuación sería negativa, pero en ninguno de los ejercicios ha ocurrido, por lo que no se ha considerado un problema.

5.7. Otras pruebas

Durante la obtención de una distancia, se plantearon otros métodos, pero aunque los resultados que daban son muy similares, se escogió el método explicado en la sección 5.6.

Tipo	Puntuación usada	Dimensiones divididas
Profesional	89,2996	95,9421
Paciente	23,3773	29,4185
Paciente	18,9154	25,5778
Paciente	19,1309	25,3962
Paciente	24,9134	31,1506
Paciente	45,5613	45,1792
Paciente	26,09757	35,1118
Paciente	22,19180	31,4718
Paciente	23,2319	31,4716
Paciente	26,2338	36,6816
Paciente	44,6043	49,1532

Tabla 5.4: Comparación de puntuación obtenida redimensionando cada una de las partes y de la puntuación calculando cada dimensión por separado.

Primero se estudió si era mejor redimensionar las series de posiciones de dos dimensiones X e Y a una sola o no hacerlo y aplicar DTW a cada dimensión por separado. Los resultados se pueden ver en la tabla 5.4. Las puntuaciones son muy similares.

Otra opción era tratar el conjunto de series de posiciones de dos dimensiones redimensionadas a una dimensión como series multidimensionales. Esto se consigue mediante una implementación que ofrece la librería *Dtaidistance* importando el modulo *ndim*. También se probó nuevamente a tratar las dimensiones X e Y por separado no redimensionandolas para ver si se conseguían mejores resultados. Estos resultados se pueden observar en la tabla 5.5.

5.8. Desarrollo de la aplicación

Inicialmente se considero desarrollar una aplicación de escritorio con la que el usuario pudiera instalar la aplicación, para ello se consideraron las siguientes librerías:

1. **Tkinter**: Es una librería gráfica que viene instalada por defecto en *Python*, pero su interfaz esta desactualizada, especialmente en *Linux*, donde tampoco ofrece soporte para *Wayland*, por lo que sería un problema de cara a futuro.

Tipo	Puntuación usada	Ndim redimensionadas	Ndim sin redimensionar
Profesional	89,2996	95,3474	96,1810
Paciente	23,3773	66,1234	42,26262
Paciente	18,9154	64,2140	39,1279
Paciente	19,1309	63,9295	38,5238
Paciente	24,9134	68,1069	43,9163
Paciente	45,5613	75,2601	54,6122
Paciente	26,09757	62,0344	42,3221
Paciente	22,19180	59,7707	38,8841
Paciente	23,2319	59,5271	37,9471
Paciente	26,2338	64,5063	44,8041
Paciente	44,6043	73,8300	56,0422

Tabla 5.5: Puntuaciones obtenida de series multivariantes redimensionando las dimensiones X e Y y sin redimensionarlas.

2. **GTK**: Es una *framework* gráfico muy utilizado en *Linux* y es de código abierto, pero no se ha encontrado información oficial sobre como dar soporte a otros sistemas como por ejemplo *Windows*.
3. **QT**: Este *framework* ofrece varias licencias para poderlo utilizar, pero puede ser complejo saber cual utilizar, otra desventaja era que no se aplican correctamente los estilos en entornos que usan *GTK* como por ejemplo en *Gnome*.

Por ello se cambió de idea y se decidió desarrollar una aplicación web, lo cual permitiría que los pacientes pudieran utilizarla sin que tuvieran que instalar nada y soluciona problemas de compatibilidad entre distintos sistemas operativos. Además, gracias al uso de contenedores *Docker* se puede ejecutar en un entorno local de forma sencilla.

Para ello se eligió el *framework* Angular, esto ofrece la ventaja de que al ser del lado del cliente actuaría como interfaz gráfica para utilizar la API, permitiría separar la lógica del *backend* del *frontend*. Además, existe una librería para este *framework* llamada Angular Material que permite la creación de varios componentes utilizados como por ejemplo selectores. El uso de este *framework* trajo retos como fueron aprender a utilizarlo y aprender *Typescript*, que es un lenguaje que no se imparte en la carrera.

Para el desarrollo de la API se optó por *FastAPI*, un *framework* de *Python* que ofrece ventajas como el poder utilizar el patrón DTO por media

```
1  ffmpeg -i input_vídeo.mp4 -brand mp42  
    output_vídeo.mp4
```

Figura 5.4: Comando para poder ver un vídeo no compatible con navegadores en ellos

de los esquemas de *Pydantic*. Para la base de datos se escogió *PostgreSQL* y la librería de *Python SQLAlchemy*.

La integración con *Docker* trajo retos, el primero de ellos fue que al ejecutarlo por primera vez el contenedor del *backend* antes de que se hubiera creado la base de datos, lo que producía un error. Finalmente se solucionó comprobando que esta se hubiera creado por medio de un comando que permite ver el estado de la base de datos.

Otro contratiempo fue el almacenamiento de los archivos en la base de datos, en el caso de los vídeos eran demasiado grandes, por lo que se eligió guardarlos en una carpeta con un nombre aleatorio y guardar ese nombre en la base de datos, por lo que se investigó como hacer persistente una carpeta en *Docker*.

Por ultimo, los vídeos de los ejercicios estaban codificados en un *codec* que no era compatible con navegadores, por lo que se utilizó *ffmpeg* para codificar los vídeos a un formato compatible, concretamente el siguiente comando 5.4.

El resultado de este desarrollo ha sido una aplicación web que permite a los usuarios crear una cuenta e iniciar sesión para poder ver vídeos de los ejercicios ya creados y subir sus ficheros para que se comparen con los de los profesionales y les devuelva una puntuación como se puede ver en la imagen 5.5.

Para crear un ejercicio es necesario se administrador, para ello la aplicación crea una cuenta de usuario por defecto, que puede ser usada para crear ejercicios, introduciendo el nombre, añadiendo un fichero de datos, el vídeo y seleccionando ángulos y coordenadas que serán usando para comparar el ejercicio como se puede ver en la imagen 5.6. También puede borrar ejercicios.

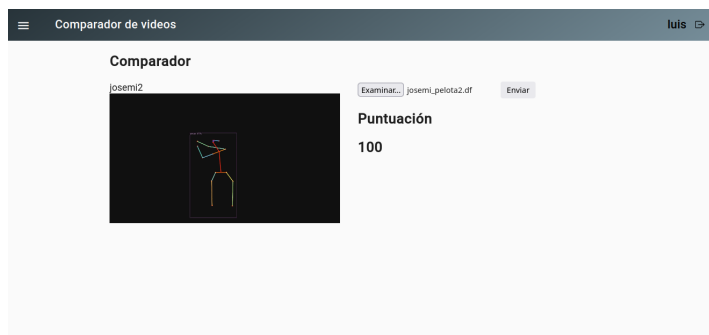


Figura 5.5: Comparación de ejercicio.

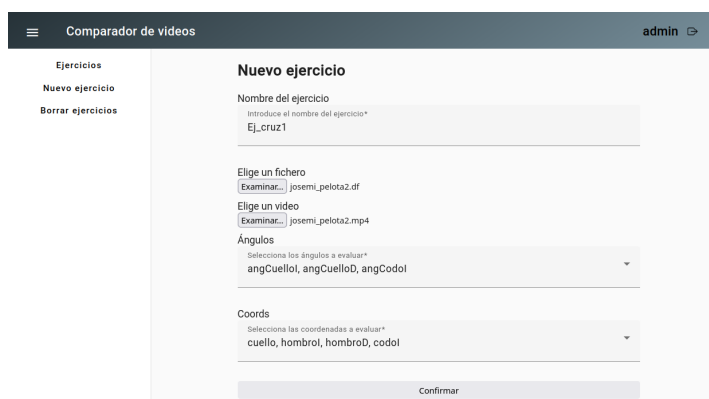


Figura 5.6: Vista de nuevo ejercicio.

6. Trabajos relacionados

En este apartado se mencionan artículos que aportan nuevos conocimientos en la puntuación de ejercicios.

6.1. Dance Performance Evaluation Using Hidden Markov Models [10]

Este artículo, realizado por la Universidad de Mons, usa los modelos ocultos de Markov para comparar las series temporales descomponiendo los movimientos en estados y aplicando el algoritmo de Viterbi para obtener la puntuación de cuan bien se ha realizado un baile.

6.2. A Dynamic Time Warping Based Algorithm to Evaluate Kinect-Enabled Home-Based Physical Rehabilitation Exercises for Older People [25]

En este trabajo se obtiene una puntuación de ejercicios de Tai Chi, para ello, se tomaron las posiciones de las extremidades superiores e inferiores, se calcularon las diferencias entre el esqueleto del paciente y del profesional mediante la aplicación de la similitud coseno en los esqueletos del usuario y del profesional.

Después se aplicó DTW para obtener una puntuación, para ello tiene en cuenta que la máxima diferencia de ángulos entre los correspondientes a dos de los vectores formados por los huesos son 90° en esos ejercicios.

Gracias a esto es capaz de tener una puntuación de rendimiento, en forma de porcentaje.

Se aplicó esta metodología sobre ejercicios de Tai Chi sobre 21 participantes distintos. El resultado de este estudio fue un algoritmo basado en DTW que permitiría el evaluar de forma autónoma los ejercicios de rehabilitación.

6.3. A Wireless Evaluation System for In-home Physical Rehabilitation [26]

Esta investigación trata de aportar una retroalimentación al usuario de como de bien o mal a realizado un ejercicio físico de rehabilitación, para ello obtiene una esqueleto 3D en el que usa la distancia DTW por parte del cuerpo para clasificar los ejercicios en "bueno" y "malo", en este caso, al igual que el anterior.

7. Conclusiones y Líneas de trabajo futuras

En este apartado se van a mostrar las conclusiones que han resultado de la realización del proyecto así como posibles futuras investigaciones y proyectos.

7.1. Conclusiones

Como primera conclusión cabe destacar las dificultades que pueden causar en un problema como es la comparación y puntuación de ejercicios el no tener un set de datos lo suficientemente grande, en el caso de los ejercicios grabados por pacientes sanos solo se disponían de un máximo de tres vídeos y muchos de los ficheros de datos de los ejercicios de los pacientes no tenían el vídeo correspondiente, lo que limita enormemente la formas de solucionarlo y el poder verificar que las puntuaciones sean correctas.

Aun así se han conseguido resultados aceptables en el caso de usar solo las distancias obtenidas de las posiciones y no de los ángulos. Me alegra haber podido participar en este proyecto, ha sido una gran oportunidad para poder ayudar a mejorar la calidad de vida de personas necesitadas.

Otra cosa que he podido obtener conocimientos sobre el desarrollo web de aplicaciones simulando entornos de desarrollo y producción. También he aprendido sobre los contenedores de *Docker*, que ha sido algo que me ha interesado desde el inicio del proyecto.

7.2. Líneas futuras

Algunas de las mejoras y líneas siguientes que se podrían estudiar son las siguientes:

1. Investigar la obtención de la puntuación mediante los modelos ocultos de *Markov*, para ello será necesario tener un set de datos más grande.
2. Mejorar el sistema de detección de movimientos, ya que esto ayudaría a obtener una puntuación mejor. Se observó que a veces, se detectaban mal los esqueletos en algunos fotogramas o valores nulos en alguna de sus partes del cuerpo.
3. La aplicación es una muestra de los resultados de la investigación, pero se pueden añadir más funciones e incluso unificar con el resto de la investigación. El *frontend* se ha desarrollado pensando en ser fácilmente extensible por este motivo.
4. Mejoras de accesibilidad en la aplicación, inicialmente se pensó en añadir la función de cambiar idioma, pero por motivos de tiempo no se pudo hacer. Actualmente los textos se obtienen de un archivo JSON mediante *ngx-translate*. Sería interesante el que pacientes que no hablen español puedan usarla.
5. Investigar librerías alternativas para poder detectar poses, ya que como se mencionó anteriormente no se lanzan nuevas versiones de Detectron precompiladas desde 2021.

Bibliografía

- [1] pandas. <https://pandas.pydata.org/>, 2024.
- [2] R. E. Bellman and S. E. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, 1st edition, 1962.
- [3] Organización Mundial de la Salud. Enfermedad de parkinson. <https://www.who.int/es/news-room/fact-sheets/detail/parkinson-disease/>.
- [4] Parkinson’s Foundation. Bradicinesia (ralentización del movimiento). <https://www.parkinson.org/espanol/entendiendo-parkinson/sintomas-motores/bradicinesia>.
- [5] Parkinson’s Foundation. La enfermedad de parkinson: causas, síntomas y tratamientos. <https://www.nia.nih.gov/espanol/parkinson/enfermedad-parkinson-causas-sintomas-tratamientos>.
- [6] R. García-Ramos, E. López Valdés, L. Ballesteros, S. Jesús, and P. Mir. Informe de la fundación del cerebro sobre el impacto social de la enfermedad de parkinson en españa. *Neurología*, 31(6):401–413, 2016.
- [7] Geeksforgeeks. Dynamic programming or dp. <https://www.geeksforgeeks.org/dynamic-programming>.
- [8] Víctor Caranqui S. Iván García S. La visión artificial y los campos de aplicación. *Tierra Infinita*, 1(1):98–108, Dec 2015.
- [9] Kaggle. Euclidean distance and normalizing a vector. <https://www.kaggle.com/code/paulrohan2020/euclidean-distance-and-normalizing-a-vector>.

- [10] Sohaib Laraba and Joëlle Tilmanne. Dance performance evaluation using hidden markov models. *Computer Animation and Virtual Worlds*, 27:321–329, 05 2016.
- [11] Luciano Maldonado. Los modelos ocultos de markov, mom. *Telos*, 2012.
- [12] JoséAlberto Mauricio. Análisis de series temporales. *Universidad Complutense de Madrid*, 2007.
- [13] Damian Michel-Vera, Francisco Hernandez-Lopez, and Anabel Martin-Gonzalez. Reconocimiento de gestos dinámicos para la manipulación de imágenes. *Research in Computing Science*, 147:229–239, 12 2018.
- [14] T. Quach and M. Farooq. Maximum likelihood track formation with the viterbi algorithm. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 1, pages 271–276 vol.1, 1994.
- [15] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.
- [16] Yves Rybarczyk, Jan Kleine Deters, Arián Aladro Gonzalo, Danilo Esparza, Mario Gonzalez, Santiago Villarreal, and Isabel L. Nunes. Recognition of physiotherapeutic exercises through dtw and low-cost vision-based motion capture. pages 348–360, 2018.
- [17] Yves Rybarczyk, Jorge Luis Pérez Medina, Louis Leconte, Karina Jimenes, Mario González, and Danilo Esparza. Implementation and assessment of an intelligent motor tele-rehabilitation platform. *Electronics*, 8(1), 2019.
- [18] Stan Salvador and Philip K. Chan. Fastdtw: Toward accurate dynamic time warping in linear time and space. 2004.
- [19] Johannes Stübinger and Dominik Walter. Using multi-dimensional dynamic time warping to identify time-varying lead-lag relationships. *Sensors*, 22(18), 2022.
- [20] Ubisoft. Nueva era, nuevas puntuaciones. <https://www.ubisoft.com/es-mx/game/just-dance/2023/news-updates/6JSYL277Jds6As65yqtAvr/nueva-era-nuevas-puntuaciones>, 2023.
- [21] Wikipedia. Latex — wikipedia, la enciclopedia libre. <https://es.wikipedia.org/w/index.php?title=LaTeX&oldid=84209252>, 2015. [Internet; descargado 30-septiembre-2015].

- [22] Wikipedia. Traslación (geometría) — wikipedia, la enciclopedia libre. [https://es.wikipedia.org/w/index.php?title=Traslaci%C3%B3n_\(geometr%C3%ADa\)&oldid=154994082](https://es.wikipedia.org/w/index.php?title=Traslaci%C3%B3n_(geometr%C3%ADa)&oldid=154994082), 2023. [Internet; descargado 31-octubre-2023].
- [23] Wikipedia. Texstudio — wikipedia, la enciclopedia libre. <https://es.wikipedia.org/w/index.php?title=TeXstudio&oldid=158396992>, 2024. [Internet; descargado 23-febrero-2024].
- [24] Wikipedia. Visual studio code — wikipedia, la enciclopedia libre. https://es.wikipedia.org/w/index.php?title=Visual_Studio_Code&oldid=158794261, 2024. [Internet; descargado 14-marzo-2024].
- [25] Xiaoqun Yu and Shuping Xiong. A dynamic time warping based algorithm to evaluate kinect-enabled home-based physical rehabilitation exercises for older people. *Sensors*, 19(13), 2019.
- [26] Xiaoming Zhang, Lin Xu, and Yang Zhang. A wireless evaluation system for in-home physical rehabilitation. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 3375–3380, 2017.