



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería
Informática

título del TFG



Presentado por Nombre del alumno
en Universidad de Burgos — 18 de junio
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. Nombre del alumno, con DNI dni, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

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, 18 de junio de 2024

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. José Luis Garrido Labrador

D. José Miguel Ramírez Sanz

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
1. Introducción	1
2. Objetivos del proyecto	3
2.1. Objetivos generales	3
2.2. Objetivos técnicos	3
2.3. Objetivos personales	3
3. Conceptos teóricos	5
3.1. Normalización	5
3.2. Similitud coseno	5
3.3. Dynamic time warping	6
4. Técnicas y herramientas	9
4.1. L ^A T _E X	9
4.2. TeXstudio	9
4.3. Git	10
4.4. Github	10
4.5. Gittyup	10
4.6. Zube	10
4.7. Visual Studio Code	10
4.8. Detectron2	10

4.9. Librerías de Python	10
4.10. Tecnologías para desarrollar la aplicación	11
5. Aspectos relevantes del desarrollo del proyecto	13
6. Trabajos relacionados	15
7. Conclusiones y Líneas de trabajo futuras	17
Bibliografía	19

Índice de figuras

3.1. Alineamiento de dos secuencias mediante DTW	6
--	---

Índice de tablas

4.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	9
---	---

1. Introducción

La enfermedad de Parkinson es una enfermedad neurodegenerativa.

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.

Uno de los tratamientos para esta enfermedad, es la realización de ejercicios físicos 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.

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 video con el de un profesional.

Por ultimo, se ha desarrollado una aplicación web para que el usuario pueda ver la puntuación de sus ejercicios.

2. Objetivos del proyecto

2.1. Objetivos generales

- Investigar sobre la comparación de secuencias temporales.
- Investigar distintos métodos de normalización para poder prepara los datos
- 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 y 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.

2.3. Objetivos personales

- Realizar un aporte que pueda ayudar a personas con Parkinson.

- Aprender sobre Docker y Nginx.
- Aprender tecnologías relacionadas con el desarrollo de aplicaciones web como son Angular y FastAPI.

3. Conceptos teóricos

3.1. 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 mini-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

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

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 será igual a 1.

3.2. 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.

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|} \quad (3.3)$$

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.

3.3. Dynamic time warping

La deformación dinámica permite comparar dos secuencias temporales en la que se realizan los movimientos a distintas velocidades.

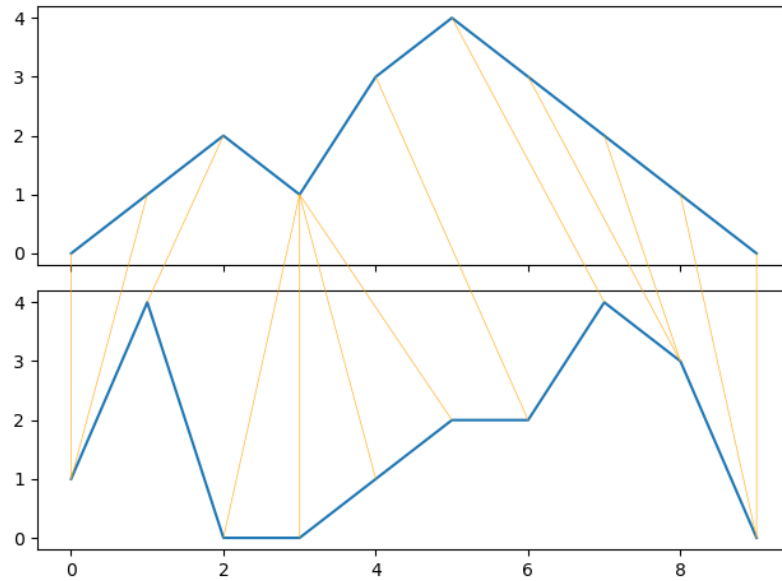


Figura 3.1: Alineamiento de dos secuencias mediante DTW

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. [2]

Matriz de costes acumulados

Una vez emparejados los puntos se usa una matriz de costes acumulados. [7]

En la matriz de costes acumulados se inicializan los valores de la siguiente manera:

1. La primera fila:

$$D(1, j) = C(1, j) \quad (3.4)$$

2. La primera columna:

$$D(i, 1) = \sum_{k=1}^i C(k, 1) \quad (3.5)$$

3. El resto:

$$D(i, j) = \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\} + C(i, j) \quad (3.6)$$

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. [4]

4.2. TeXstudio

TeXstudio es un editor de \LaTeX de código abierto y multiplataforma. [5] 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.

Herramientas	Pruebas	App	Memoria
Jupyter	X		
Git + Github + Gittyup	X	X	X
\LaTeX + TexStudio			X

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

4.3. Git

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.

4.4. Github

Github es una plataforma online que utiliza Git para guardar repositorios y ajustar la visibilidad de los mismos.

4.5. Gittyup

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.

4.6. Zube

Zube es una página web dirigida a la planificación de proyectos utilizando metodologías ágiles como *Scrum* y *Kanban*.

4.7. 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. [6]

4.8. Detectron2

Detectron2 es una librería de python destinada a la detección y segmentación de imágenes.

4.9. Librerías de Python

Pandas

Pandas es una librería de código abierto para análisis y manipulación de datos. [1]

Numpy

Numpy es una librería que facilita el trabajar con matrices en Python.

Dtaidistance

Una librería que implementa distintas funciones relacionadas con DTW.

FastDTW

FastDTW es una librería de Python que implementa una aproximación de DTW que tiene una complejidad lineal en tiempo y espacio. [3] Finalmente se descarto su uso ya que dicha librería no se ha actualizado desde hace tiempo y cambios en las dependencias provocaban errores.

4.10. 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.

FastAPI

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

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

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

PostgreSQL

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

SQLAlchemy

5. Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros³, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

6. Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

7. Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

- [1] pandas. <https://pandas.pydata.org/>, 2024.
- [2] 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.
- [3] Stan Salvador and Philip K. Chan. Fastdtw: Toward accurate dynamic time warping in linear time and space. 2004.
- [4] Wikipedia. Latex — wikipedia, la enciclopedia libre. <https://es.wikipedia.org/w/index.php?title=LaTeX&oldid=84209252>, 2015. [Internet; descargado 30-septiembre-2015].
- [5] Wikipedia. Textstudio — wikipedia, la enciclopedia libre. <https://es.wikipedia.org/w/index.php?title=TeXstudio&oldid=158396992>, 2024. [Internet; descargado 23-febrero-2024].
- [6] 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].
- [7] 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.