



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Detección de ejercicios en  
vídeos de rehabilitación  
Documentación Técnica**



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



---

# Índice general

---

<b>Índice general</b>	<b>i</b>
<b>Índice de figuras</b>	<b>iii</b>
<b>Índice de tablas</b>	<b>iv</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Estudio de viabilidad . . . . .	5
<b>Apéndice B Especificación de Requisitos</b>	<b>11</b>
B.1. Introducción . . . . .	11
B.2. Objetivos generales . . . . .	11
B.3. Catálogo de requisitos . . . . .	11
B.4. Especificación de requisitos . . . . .	12
<b>Apéndice C Especificación de diseño</b>	<b>21</b>
C.1. Introducción . . . . .	21
C.2. Diseño de datos . . . . .	21
C.3. Diseño procedimental . . . . .	23
C.4. Diseño arquitectónico . . . . .	23
<b>Apéndice D Documentación técnica de programación</b>	<b>25</b>
D.1. Introducción . . . . .	25
D.2. Estructura de directorios . . . . .	25
D.3. Manual del programador . . . . .	27

D.4. Compilación, instalación y ejecución del proyecto . . . . .	29
D.5. Pruebas del sistema . . . . .	31
<b>Apéndice E Documentación de usuario</b>	<b>33</b>
E.1. Introducción . . . . .	33
E.2. Requisitos de usuarios . . . . .	33
E.3. Instalación . . . . .	33
E.4. Manual del usuario . . . . .	34
<b>Bibliografía</b>	<b>39</b>

---

## Índice de figuras

---

B.1. Diagrama de casos de uso del paciente. . . . .	13
B.2. Diagrama de casos de uso del terapeuta. . . . .	13
C.1. Relación entre la tabla <i>USER</i> y la tabla <i>CREDENTIALS</i> . . . .	22
C.2. Diagrama relacional de las tablas referentes a los ejercicios. . . .	22
C.3. Diagrama de secuencia del flujo del paciente. . . . .	23
C.4. Diagrama de la arquitectura. . . . .	24
D.1. Estructura de directorios del <i>backend</i> . . . . .	26
D.2. Estructura de directorios del <i>frontend</i> . . . . .	28
D.3. Comandos para crear y activar entorno virtual de Python. . . .	29
D.4. Comando para activar el entorno virtual de Python. . . . .	29
D.5. Comando para ejecutar el <i>backend</i> en modo desarrollo. . . . .	30
D.6. Comando para ejecutar el <i>frontend</i> en modo desarrollo. . . . .	30
D.7. Comando para compilar el <i>frontend</i> en modo producción. . . . .	30
D.8. Comando ejecutar los contenedores de docker. . . . .	30
E.1. Pantalla de creación de cuenta. . . . .	34
E.2. Pantalla de inicio de sesión. . . . .	35
E.3. Selector de ejercicios . . . . .	35
E.4. Vista del comparador de vídeos. . . . .	36
E.5. Puntuación del ejercicio. . . . .	36
E.6. Vista de nuevo ejercicio. . . . .	37
E.7. Pantalla para borrar ejercicios. . . . .	38
E.8. Confirmación de que se desea borrar el ejercicio. . . . .	38

---

# Índice de tablas

---

A.1. Sprint 0. . . . .	2
A.2. Sprint 1. . . . .	2
A.3. Sprint 2. . . . .	2
A.4. Sprint 3. . . . .	3
A.5. Sprint 4. . . . .	3
A.6. Sprint 5. . . . .	3
A.7. Sprint 6. . . . .	4
A.8. Sprint 7. . . . .	4
A.9. Sprint 8. . . . .	4
A.10.Sprint 9. . . . .	5
A.11.Costes hardware. . . . .	6
A.12.Tabla con las licencias de las librerías y herramientas utilizadas en el <i>backend</i> . . . . .	7
A.13.Table de dependencias del <i>frontend</i> con sus licencias. . . . .	8
B.1. CU-1 Creación de un nuevo ejercicio. . . . .	14
B.2. CU-2 Eliminación de un ejercicio. . . . .	15
B.3. CU-3 Selección de un nuevo ejercicio. . . . .	16
B.4. CU-4 Comparación de ejercicio. . . . .	17
B.5. CU-5 Cargar fichero de ejercicio . . . . .	18
B.6. CU-6 Creación de cuenta. . . . .	19
B.7. CU-7 Inicio de sesión. . . . .	19

## *Apéndice A*

---

# Plan de Proyecto Software

---

## A.1. Introducción

La planificación de un proyecto es vital para su correcto desarrollo. En este apartado se estudian las diferentes fases del proyecto, los plazos y los recursos necesarios para llevarlo a cabo y otros requisitos de viabilidad.

En esta fase se distinguen dos secciones:

1. **Planificación temporal:** Se analizarán los distintos sprints exponiendo su fecha de inicio y fin y las tareas, especificando tanto el coste estimado como el final que tuvieron.
2. **Estudio de viabilidad:** En esta etapa se estimarán los costes del proyecto. También se tratarán la viabilidad legal del trabajo, en la que se expondrán las licencias de las dependencias usadas, así como la licencia final del proyecto.

## A.2. Planificación temporal

La planificación temporal se ha dividido en distintos sprints de una a dos semanas de duración cada uno, en los cuales se realizaron distintas tareas y reuniones. Las reuniones se realizaban de forma telemática semanalmente, en ellas se exponían las tareas realizadas y las dudas y los problemas que surgían, a lo que se aportaban distintas soluciones y tareas nuevas.

### Sprint 0: 06/02/2024 - 12/03/2024

En el *sprint 0* se buscaron y leyeron artículos mientras se preparaban los vídeos necesarios. También se creó el repositorio y se clonó la plantilla de L<sup>A</sup>T<sub>E</sub>X proporcionada por la Universidad De Burgos.

Tareas	Pto. Estimados	Pto. Reales
Investigar el sistema de puntuaciones de Just Dance	8	9
Investigar métodos para diferenciar las poses.	5	8
Descargar plantilla de L <sup>A</sup> T <sub>E</sub> X	0.5	0.5

Tabla A.1: Sprint 0.

### Sprint 1: 15/03/2024 - 26/03/2024

En este *sprint 1* se usó para hacer pruebas con los datos.

Tareas	Pto. Estimados	Pto. Reales
Añadir archivos de datos	0.5	0.5
Revisar vídeos de los ejercicios	1	2
Dibujar esqueleto con los puntos de un frame	3	4
Rotar esqueleto	1	1
Normalización L2	2	1
Aplicar similitud coseno	2	2

Tabla A.2: Sprint 1.

### Sprint 2: 26/03/2024 - 11/04/2024

Este *sprint* se utilizó para investigar sobre Dynamic Time Warping (DTW).

Tareas	Pto. Estimados	Pto. Reales
Explicar similitud coseno en la memoria	3	3
Añadir Sprint 0 a la memoria	1	1
Añadir Sprint 1 a la memoria	1	1
Añadir herramientas a la memoria	2	2
Investigar sobre DTW	5	5
Limpiar los datos de los CSV y DataFrame	1	0.5

Tabla A.3: Sprint 2.



**Sprint 3: 11/04/2024 - 24/04/2024**

En este *sprint* se continuó con las pruebas de similitud coseno y se comenzó con las pruebas de DTW.

Tareas	Pto. Estimados	Pto. Reales
Comparación de vídeos mediante similitud coseno	5	5
Pruebas con DTW	5	5
Añadir Sprint 2 a la memoria	1	1
Añadir explicación de DTW a la memoria	5	5

Tabla A.4: Sprint 3.

**Sprint 4: 26/04/2024 - 7/05/2024**

Durante el *sprint* anterior se detectó que había posiciones que no se habían detectado correctamente dando valores nulos, por lo que se buscó cuales eran. También se empezó a

Tareas	Pto. Estimados	Pto. Reales
Comprobar en que ficheros hay celdas vacías	3	3
Comprobar 3 vídeos de un mismo ejercicio	5	4
Añadir Sprint 3 a la memoria	1	0.5
DTW con ángulos	5	6

Tabla A.5: Sprint 4.

**Sprint 5: 8/05/2024 - 21/05/2024**

Al haber recibido los datos de los ejercicios ejecutados correctamente, se comenzó a hacer pruebas con ellos.

Tareas	Pto. Estimados	Pto. Reales
Hacer pruebas con ejercicios correctos	2	2
Añadir pruebas con métodos de normalización en dtw	3	5
Añadir Sprint 4 a la memoria	1	1
Reorganizando y añadiendo explicaciones en las pruebas	3	3
Pruebas fastdtw	3	3

Tabla A.6: Sprint 5.

### Sprint 6: 22/05/2024 - 5/06/2024

En este *sprint* se investigó sobre las posibles tecnologías que se iban a usar en la aplicación así como continuar con las pruebas de ángulos.

Tareas	Pto. Estimados	Pto. Reales
Investigar tecnologías para la aplicación	8	8
Investigar métodos de normalización	5	5
Añadir Sprint 5 a la memoria	0.5	0.5
Continuar con las pruebas de ángulos	3	3

Tabla A.7: Sprint 6.

### Sprint 7: 5/06/2024 - 12/06/2024

Se añadieron los métodos de normalización investigados durante el *sprint* anterior.

Tareas	Pto. Estimados	Pto. Reales
Añadir métodos de normalización	8	8
Añadir casos de uso de la aplicación	5	5
Añadir Sprint 6 a la memoria	0.5	0.5

Tabla A.8: Sprint 7.

### Sprint 8: 12/06/2024 - 19/06/2024

Este *sprint* se utilizó para desarrollar la aplicación.

Tareas	Pto. Estimados	Pto. Reales
Empezar a desarrollar la aplicación	5	8
Avanzar con los anexos	5	5
Avanzar con la memoria	5	6

Tabla A.9: Sprint 8.

### Sprint 9: 19/06/2024 - 9/07/2024

Este *sprint* se cerraron tanto las partes de la documentación como las últimas funcionalidad de la aplicación.

Tareas	Pto. Estimados	Pto. Reales
Terminar la memoria	8	8
Terminar los anexos	8	8
Terminar la aplicación	8	8

Tabla A.10: Sprint 9.

## A.3. Estudio de viabilidad

### Viabilidad económica

#### Coste de personal

Suponiendo que este trabajo se hubiera realizado en un entorno profesional hubiera necesario contratar a un desarrollador. Se estima que este proyecto se realizó en 490 horas repartidas en 5 meses, lo que supone 24,5 horas semanales de trabajo. El sueldo medio de un desarrollador junior en España es de 11,61 euros/hora [1].

$$24,5h/semana * 4semanas/mes * 11,61euro/hora = 1137,78euro/mes \quad (A.1)$$

A esta cantidad se le añadirían los impuestos que ha de pagar como empresa por contratar al empleado:

- Contingencias Comunes: Correspondiente el 23,6 %.
- Accidentes de Trabajo y Enfermedades Profesionales: Correspondiente el 1,5 %.
- Desempleo: Correspondiente el 6,70 %.
- Fogasa: Correspondiente el 0,20 %.
- Formación profesional: Correspondiente el 0,60 %.

Teniendo en cuenta los costes de seguridad social para la empresa se podría calcular el coste total empleando la siguiente fórmula.

$$\frac{1137,78euro/mes}{1 - (0,236 + 0,067 + 0,002 + 0,006)} = 1651,35euro/mes \quad (A.2)$$

$$1651,35euro/mes * 5meses = 8256,75euros \quad (A.3)$$

Elemento	Coste	Coste amortizado
Ordenador portátil	1000	83,33
WebCam	100	8,33

Tabla A.11: Costes hardware.

### Costes hardware

Para este trabajo se utilizó un ordenador y una cámara para las reuniones telemáticas. Se han utilizado durante 5 meses, dato que se tendrá en cuenta al calcular el coste amortizado. Como se puede ver en la tabla [A.11](#).

### Costes totales

El coste total es la suma de todos los costes, que será de 8.348,41 euros en 5 meses.

### Viabilidad legal

En esta subsección se van a exponer las distintas licencias que tienen las herramientas y librerías utilizadas, así como la licencia final con la que cuenta este proyecto.

Las tablas [A.12](#) y [A.13](#) reflejan las licencias de las dependencias usadas en este proyecto.

Paquete	Version	Licencia
Jinja2	3.1.4	BSD License
MarkupSafe	2.1.5	BSD License
PyJWT	2.8.0	MIT License
PyYAML	6.0.1	MIT License
Pygments	2.18.0	BSD License
SQLAlchemy	2.0.31	MIT License
annotated-types	0.7.0	MIT License
anyio	4.4.0	MIT License
certifi	2024.7.4	Mozilla Public License 2.0 (MPL 2.0)
click	8.1.7	BSD License
dnspython	2.6.1	ISC License (ISCL)
dtaiDistance	2.3.12	Apache Software License
email_validator	2.2.0	The Unlicense (Unlicense)
fastapi	0.111.0	MIT License

Paquete	Version	Licencia
fastapi-cli	0.0.4	MIT License
greenlet	3.0.3	MIT License
h11	0.14.0	MIT License
httpcore	1.0.5	BSD License
httptools	0.6.1	MIT License
httpx	0.27.0	BSD License
idna	3,7	BSD License
markdown-it-py	3.0.0	MIT License
mdurl	0.1.2	MIT License
numpy	2.0.0	BSD License
orjson	3.10.6	Apache Software License; MIT License
pandas	2.2.2	BSD License
passlib	1.7.4	BSD
psycpg2-binary	2.9.9	GNU Library or LGPL
pydantic	2.8.2	MIT License
pydantic-settings	2.3.4	MIT License
pydantic_core	2.20.1	MIT License
python-dateutil	2.9.0.post0	Apache Software License; BSD License
python-dotenv	1.0.1	BSD License
python-multipart	0.0.9	Apache Software License
pytz	2024,1	MIT License
rich	13.7.1	MIT License
shellingham	1.5.4	ISC License (ISCL)
six	1.16.0	MIT License
sniffio	1.3.1	Apache Software License; MIT License
starlette	0.37.2	BSD License
typer	0.12.3	MIT License
typing_extensions	4.12.2	Python Software Foundation License
tzdata	2024,1	Apache Software License
ujson	5.10.0	BSD License
uvicorn	0.30.1	BSD License
uvloop	0.19.0	Apache Software License; MIT License
watchfiles	0.22.0	MIT License
websockets	12	BSD License

Tabla A.12: Tabla con las licencias de las librerías y herramientas utilizadas en el *backend*.

Paquete	Versión instalada	Versión mínima	Licencia
@angular/animations	18.0.6	^18.0.6	MIT
@angular/cdk	18.0.6	^18.0.6	MIT
@angular/common	18.0.6	^18.0.6	MIT
@angular/compiler	18.0.6	^18.0.6	MIT
@angular/core	18.0.6	^18.0.6	MIT
@angular/forms	18.0.6	^18.0.6	MIT
@angular/material	18.0.6	^18.0.6	MIT
@angular/platform-browser	18.0.6	^18.0.6	MIT
@angular/platform-browser-dynamic	18.0.6	^18.0.6	MIT
@angular/router	18.0.6	^18.0.6	MIT
@auth0/angular-jwt	5.2.0	^5.2.0	MIT
@ngx-translate/http-loader	8.0.0	^8.0.0	MIT
marked	12.0.2	^12.0.2	MIT
ngx-markdown	18.0.0	^18.0.0	MIT
ngx-translate	0.0.1-security	^0.0.1-security	MIT
rxjs	7.8.1	~7.8.0	Apache-2.0
sweetalert2	11.12.2	^11.12.2	MIT
tslib	2.6.3	^2.6.3	0BSD
zone.js	0.14.7	^0.14.7	MIT
@angular-devkit/build-angular	18.0.7	^18.0.7	MIT
@angular/cli	18.0.7	^18.0.7	MIT
@angular/compiler-cli	18.0.6	^18.0.6	MIT
@types/jasmine	5.1.4	~5.1.0	MIT
typescript	5.4.5	~5.4.5	Apache-2.0

Tabla A.13: Table de dependencias del *frontend* con sus licencias.

La licencia final escogida para este proyecto ha sido GPL v3.0 ya que esta licencia permite utilizar el software desarrollado para su uso comercial y modificar las implementaciones realizadas, distribuirlas, realizar patentes sobre ellas y usarlas de forma privada que no entra en conflicto con ninguna licencia de las herramientas y librerías utilizadas.





## *Apéndice B*

---

# Especificación de Requisitos

---

### B.1. Introducción

En esta sección se tratarán los objetivos y requisitos del proyecto. También se definirán los distintos actores que utilizarán la aplicación y se establecerán los casos de uso de la misma.

### B.2. Objetivos generales

Los objetivos generales que se buscan alcanzar son los siguientes:

- 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.

### B.3. Catálogo de requisitos

#### Actores

Es importante distinguir los roles de los usuarios de la aplicación, a efectos de uso de la misma se pueden distinguir dos roles:

- **Terapeuta:** Persona que se encarga de crear los ejercicios que realizarán los pacientes.
- **Paciente:** Persona que quiere comparar su ejercicio con el del terapeuta.

## Requisitos funcionales

En este apartado se tratarán los requisitos funcionales de la

- **RF-1** Gestión de ejercicios
  - **RF-1.1** El terapeuta podrá crear un ejercicio nuevo.
  - **RF-1.2** El terapeuta podrá eliminar un ejercicio existente.
- **RF-2** Selección de ejercicios
  - **RF-2.1** El terapeuta podrá ver los ejercicios que hay disponibles.
  - **RF-2.2** El usuario podrá ver los ejercicios que hay disponibles.
  - **RF-2.3** El usuario podrá seleccionar el ejercicio que se va a comparar.
- **RF-3** Gestión de ficheros
  - **RF-3.1** El terapeuta podrá guardar los ficheros con los datos en la aplicación.
  - **RF-3.2** El usuario podrá seleccionar el fichero de datos que se va a comparar.
- **RF-4** Comparación de ejercicios
  - **RF-4.1** El usuario podrá ver una puntuación numérica del ejercicio.
- **RF-5** Gestión de usuarios
  - **RF-5.1** El usuario podrá crear una cuenta en la aplicación.
  - **RF-5.2** El usuario podrá iniciar sesión en la aplicación.
  - **RF-5.3** El terapeuta podrá iniciar sesión en la aplicación.

## B.4. Especificación de requisitos

En esta sección se especificarán tanto los diagramas de casos de uso como los requisitos necesarios para que la aplicación se pueda desarrollar.

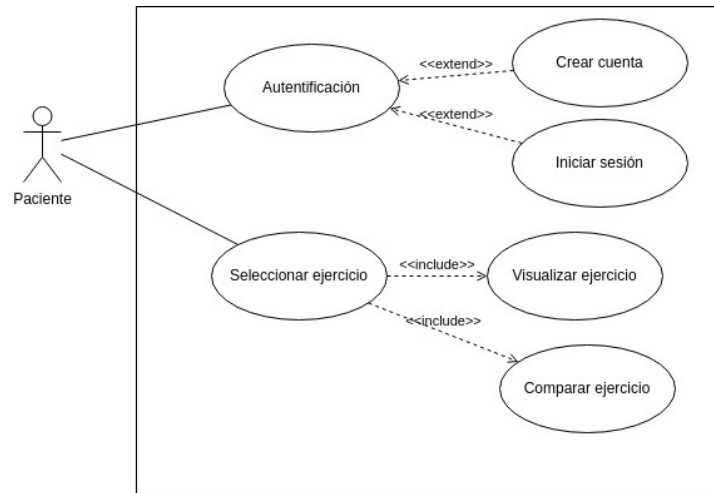


Figura B.1: Diagrama de casos de uso del paciente.



Figura B.2: Diagrama de casos de uso del terapeuta.

CU-1	Creación de un nuevo ejercicio
<b>Versión</b>	1.2
<b>Autor</b>	Luis Ángel Espinosa Lafuente
<b>Requisitos asociados</b>	RF-1.1 RF-2.1
<b>Descripción</b>	Permite al terapeuta crear un ejercicio.
<b>Precondición</b>	El terapeuta ha iniciado sesión con una cuenta con el rol correspondiente
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El terapeuta introduce el nombre del ejercicio.</li> <li>2. El terapeuta añade un fichero de datos.</li> <li>3. El terapeuta añade un vídeo.</li> <li>4. El terapeuta selecciona los ángulos y coordenadas que serán usados para la puntuación.</li> <li>5. El terapeuta presiona el botón de confirmar.</li> </ol>
<b>Postcondición</b>	Ninguna
<b>Excepciones</b>	Acceso no autorizado en el caso de que el usuario no haya iniciado sesión con el rol de terapeuta
<b>Importancia</b>	Alta

Tabla B.1: CU-1 Creación de un nuevo ejercicio.

<b>CU-2</b>	<b>Eliminación de un ejercicio</b>
<b>Versión</b>	1.1
<b>Autor</b>	Luis Ángel Espinosa Lafuente
<b>Requisitos asociados</b>	RF-1.2 RF-2.1
<b>Descripción</b>	Permite al terapeuta eliminar un ejercicio.
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El terapeuta ha iniciado sesión con una cuenta con el rol correspondiente.</li> <li>2. El terapeuta ha creado ejercicios previamente.</li> </ol>
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El terapeuta selecciona el ejercicio.</li> <li>2. El terapeuta presiona el botón de confirmar.</li> </ol>
<b>Postcondición</b>	Ninguna
<b>Excepciones</b>	Acceso no autorizado en el caso de que el usuario no haya iniciado sesión con el rol de terapeuta.
<b>Importancia</b>	Alta

Tabla B.2: CU-2 Eliminación de un ejercicio.

CU-3	Selección de ejercicio
<b>Versión</b>	1.1
<b>Autor</b>	Luis Ángel Espinosa Lafuente
<b>Requisitos asociados</b>	RF-2.2 RF-2.3
<b>Descripción</b>	Permite al usuario ver los ejercicios disponibles y seleccionar el ejercicio que se va a comparar
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El usuario ha iniciado sesión en la aplicación correctamente.</li> <li>2. El terapeuta ha creado ejercicios previamente.</li> </ol>
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona un ejercicio de un listado.</li> </ol>
<b>Postcondición</b>	Ninguna
<b>Excepciones</b>	Acceso no autorizado en el caso de que el usuario no haya iniciado sesión.
<b>Importancia</b>	Alta

Tabla B.3: CU-3 Selección de un nuevo ejercicio.

CU-4	Comparación de ejercicio
<b>Versión</b>	1.1
<b>Autor</b>	Luis Ángel Espinosa Lafuente
<b>Requisitos asociados</b>	RF-4.1
<b>Descripción</b>	Permite al usuario comparar el ejercicio.
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El usuario ha iniciado sesión en la aplicación.</li> <li>2. El usuario ha seleccionado un ejercicio.</li> <li>3. El usuario ha seleccionado un fichero.</li> <li>4. El fichero se ha cargado correctamente.</li> </ol>
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario visualiza la puntuación de un ejercicio.</li> <li>2. El usuario visualiza el vídeo del ejercicio.</li> </ol>
<b>Postcondición</b>	Ninguna
<b>Excepciones</b>	Acceso no autorizado en el caso de que el usuario no haya iniciado sesión
<b>Importancia</b>	Alta

Tabla B.4: CU-4 Comparación de ejercicio.

CU-5	Cargar fichero de ejercicio
<b>Versión</b>	1.1
<b>Autor</b>	Luis Ángel Espinosa Lafuente
<b>Requisitos asociados</b>	RF-3.1
<b>Descripción</b>	Permite al usuario ver los ejercicios disponibles y seleccionar el ejercicio que se va a comparar.
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El usuario ha iniciado sesión correctamente.</li> <li>2. El usuario a seleccionado un ejercicio.</li> </ol>
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona un fichero.</li> <li>2. El usuario presiona el botón de confirmar.</li> </ol>
<b>Postcondición</b>	El usuario ha seleccionado un fichero correctamente.
<b>Excepciones</b>	Acceso no autorizado en el caso de que el usuario no haya iniciado sesión
<b>Importancia</b>	Alta

Tabla B.5: CU-5 Cargar fichero de ejercicio



CU-6	Creación de cuenta
<b>Versión</b>	1.0
<b>Autor</b>	Luis Ángel Espinosa Lafuente
<b>Requisitos asociados</b>	RF-5.1
<b>Descripción</b>	Permite al usuario crear una cuenta para poder usar la aplicación.
<b>Precondición</b>	Ninguna
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario introduce el usuario.</li> <li>2. El usuario introduce una contraseña.</li> <li>3. El usuario presiona el botón de confirmar.</li> </ol>
<b>Postcondición</b>	Ninguna
<b>Excepciones</b>	El usuario ya existe.
<b>Importancia</b>	Baja

Tabla B.6: CU-6 Creación de cuenta.

CU-7	Inicio de sesión
<b>Versión</b>	1.0
<b>Autor</b>	Luis Ángel Espinosa Lafuente
<b>Requisitos asociados</b>	RF-5.2 RF-5.3
<b>Descripción</b>	El usuario podrá iniciar sesión para poder acceder a la aplicación.
<b>Precondición</b>	El usuario tiene una cuenta.
<b>Acciones</b>	<ol style="list-style-type: none"> <li>1. El usuario introduce el usuario.</li> <li>2. El usuario introduce una contraseña.</li> <li>3. El usuario presiona el botón de confirmar.</li> </ol>
<b>Postcondición</b>	Ninguna
<b>Excepciones</b>	Usuario o contraseña incorrectos.
<b>Importancia</b>	Baja

Tabla B.7: CU-7 Inicio de sesión.



## *Apéndice C*

---

# Especificación de diseño

---

### C.1. Introducción

En esta sección se mostrara el conjunto de diseños realizados para la correcta implementación del código. Para ello se distinguen de tres diseños:

- **Diseño de datos:** En esta parte del diseño se mostrarán las estructuras de datos utilizadas.
- **Diseño procedimental:** En esta sección se mostraran diagramas relacionados con el flujo del programa.
- **Diseño arquitectónico:** En esta parte del diseño se especificará la arquitectura del proyecto.

### C.2. Diseño de datos

Durante el desarrollo de la aplicación se trabajó con ficheros de datos que contenían una fila por cada fotograma del vídeo el cual a su vez contenía el número de fotograma los datos de las partes del cuerpo y los ángulos.

Para guardar los datos se ha implementado la persistencia utilizando dos bases de datos, dependiendo del entorno en el que se ejecute se utilizara una u otra. Si se ejecuta en el entorno de desarrollo se utiliza SQLite, si se ejecuta en el entorno de producción se utiliza PostgreSQL.

Para poder cumplir los requisitos de usuarios relacionados con la autenticación era necesario almacenar los datos de los usuarios y sus credenciales

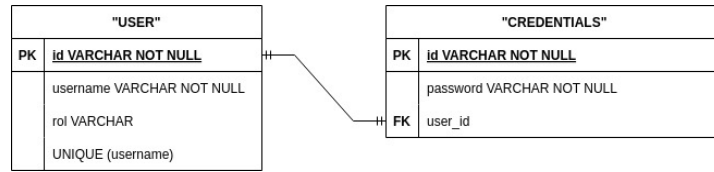
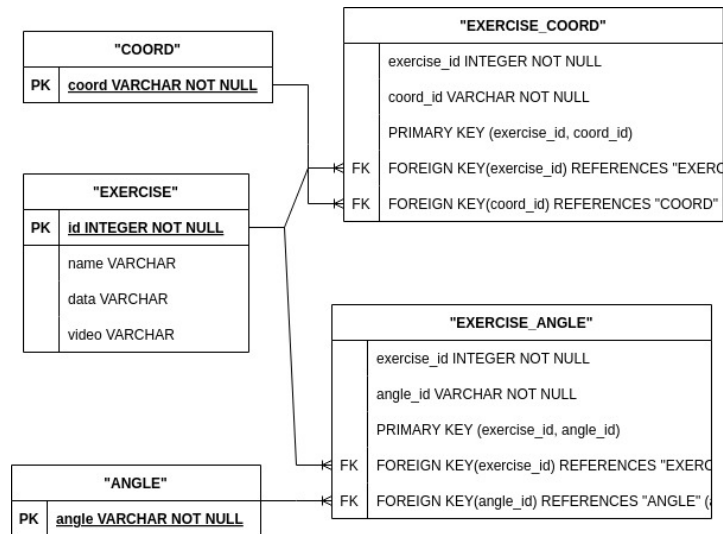
Figura C.1: Relación entre la tabla *USER* y la tabla *CREDENTIALS*.

Figura C.2: Diagrama relacional de las tablas referentes a los ejercicios.

encriptadas, para ello se creó dos tablas con una relación uno a uno, la tabla *USER* y la tabla *CREDENTIALS* estas se pueden ver en la figura C.1.

Por otro lado, era necesario guardar los ejercicios con los que el usuario podrá comparar sus datos. Para ello se creó la tabla *EXERCISE* que guardaba tanto el nombre del ejercicio como las rutas relativas de los ficheros de datos y de los vídeos. Como en la comparación de un ejercicio se pueden usar tanto las coordenadas de las posiciones del cuerpo como los ángulos, por ello se crearon dos tablas para guardar los nombres de las posiciones y de los ángulos, que fueron las tablas *COORD* y *ANGLE* respectivamente. Estas tablas tenían una relación muchos a muchos con la tabla *EXERCISE*, por lo que se crearon dos tablas intermedias, como se puede observar en la figura C.2.

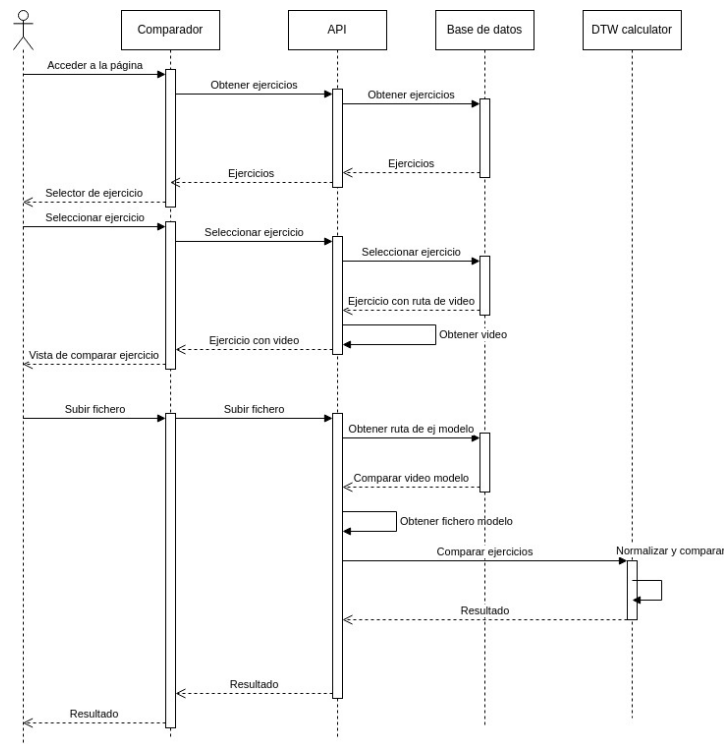


Figura C.3: Diagrama de secuencia del flujo del paciente.

## C.3. Diseño procedimental

En este apartado se encuentra la explicación del flujo del programa durante su ejecución. En la figura C.3 se mostrará el diagrama de secuencias de un usuario que va a usar la aplicación para obtener un puntuación de su vídeo. Como se puede observar, el usuario (habiendo iniciado sesión previamente) selecciona un ejercicio, lo que hace que se cargue la pagina con el vídeo del ejercicio, sube el fichero de datos de ese ejercicio y se compara con el fichero de datos

## C.4. Diseño arquitectónico

La arquitectura de la aplicación consiste en dos paquetes, el *backend* y el *frontend*, además de la base de datos.

La arquitectura usada ha sido el modelo multicapa, dividido en 4 capas [3]:

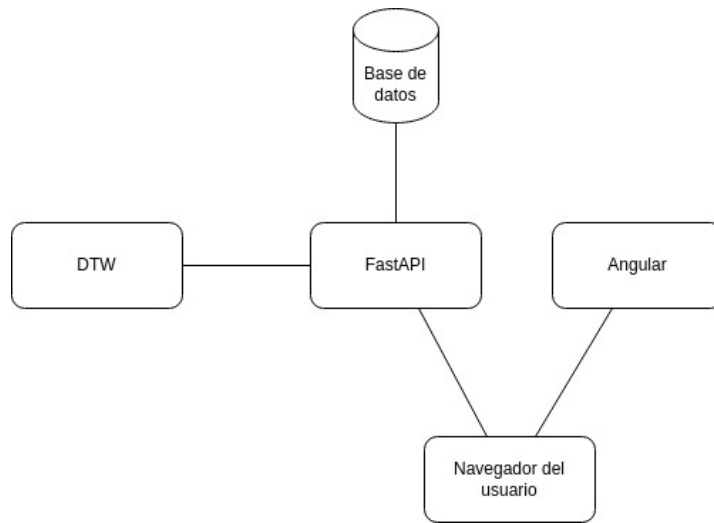


Figura C.4: Diagrama de la arquitectura.

1. **Capa de presentación:** Es la interfaz de la aplicación, en este caso el *frontend*.
2. **Capa de aplicación o capa de servicios:** Es la fachada entre la capa de presentación y la capa de lógica de negocios, en este caso sería la API.
3. **Capa de logica de negocio:** Se encarga tanto de la lógica de la aplicación como de gestionar los modelos.
4. **Capa de origen de datos:** Se encarga de la persistencia de la aplicación, es decir las bases de datos.

En el caso de que se despliegue la aplicación en mediante los contenedores de docker la aplicación constará de tres contenedores: el contenedor de la base de datos, el contenedor del *backend*, que depende del de la base de datos, y el del *frontend*.

## *Apéndice D*

---

# **Documentación técnica de programación**

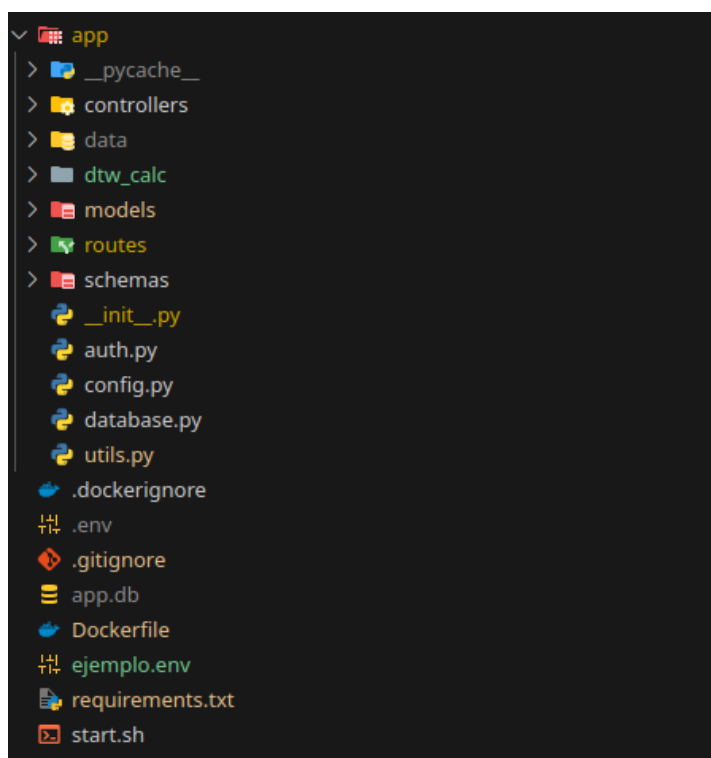
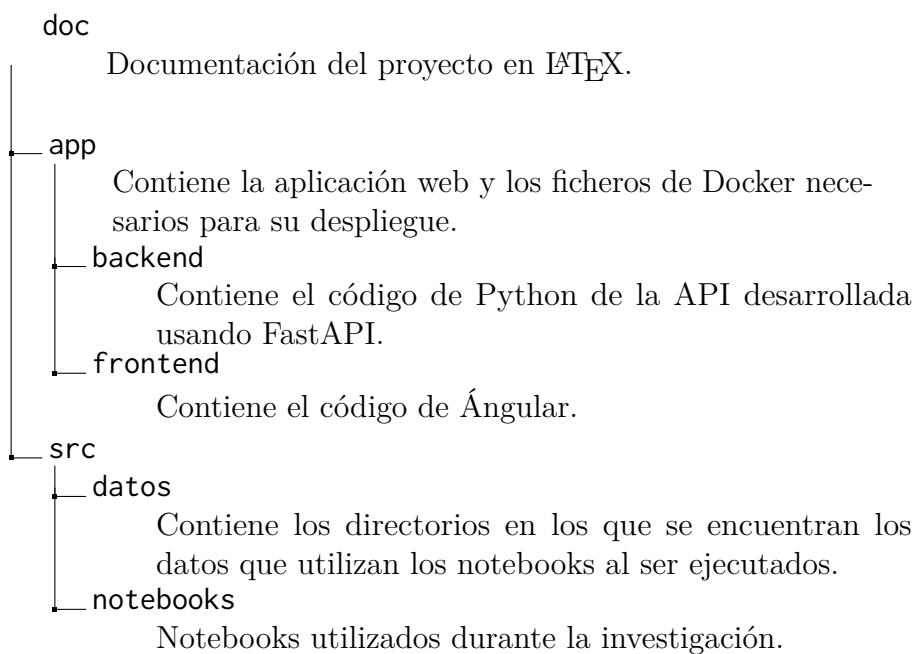
---

### **D.1. Introducción**

En este apéndice se proporciona la documentación fundamental para la adecuada programación del proyecto. Aquí se detalla la estructura y organización necesarias para su funcionamiento, así como un manual para el programador que facilita la comprensión del código. También se describe el proceso de instalación, ejecución y pruebas pertinentes.

### **D.2. Estructura de directorios**

Los directorios más relevantes son los siguientes:

Figura D.1: Estructura de directorios del *backend*.

En el caso del *backend* se puede observar su estructura en la figura [D.1](#)



En cuanto al *frontend* esta sería la estructura que se puede observar en la figura D.2.

## D.3. Manual del programador

- **Sistema operativo:** Para este proyecto se ha utilizado Arch Linux para desarrollar tanto la investigación como el desarrollo de la aplicación.
- **Versión de Python:** Durante el desarrollo se ha utilizado la versión 3.12 de Python.

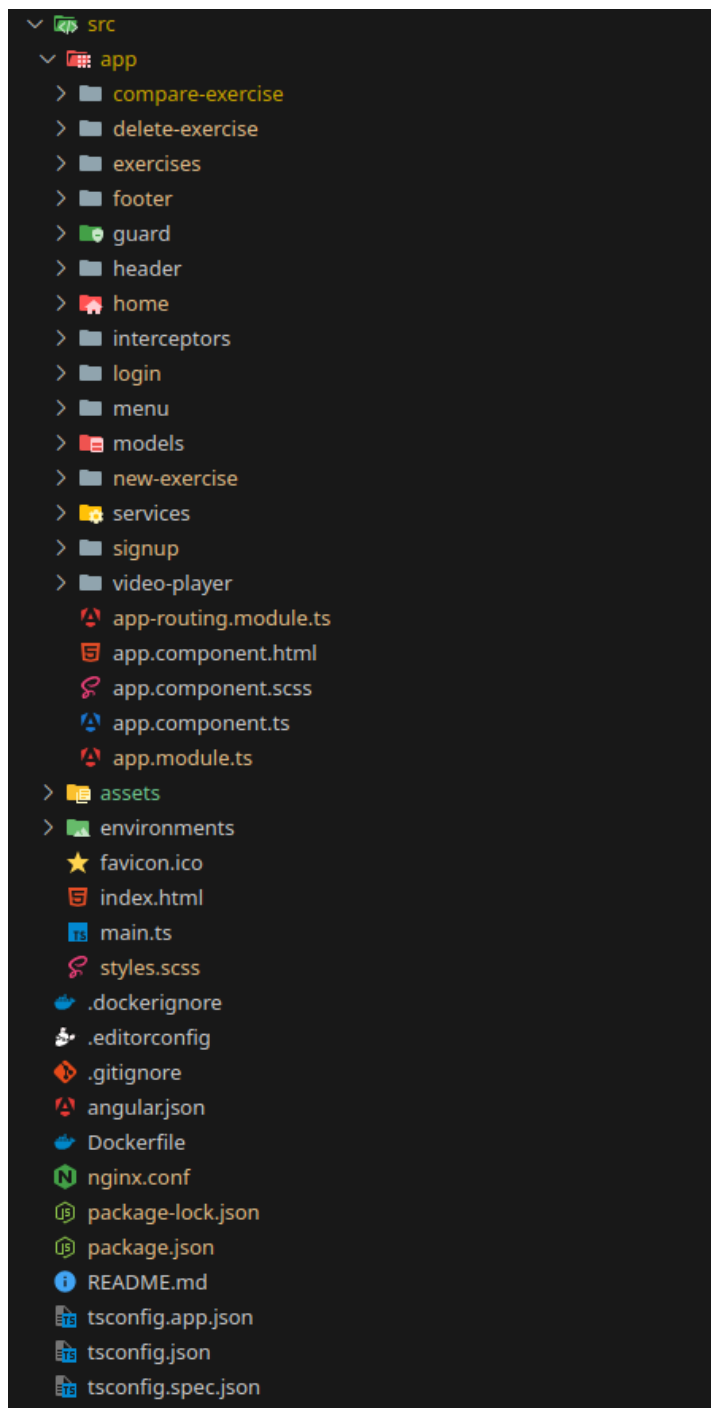
### Notebooks

Para ejecutar los *notebooks* de Jupyter, se ha utilizado Visual Studio Code. Una vez se habrá por primera vez un *notebook* en dicho programa, Visual Studio Code preguntará si se desea instalar las extensiones relacionadas con el tipo de archivo, es necesario responder de forma afirmativa.

### Aplicación Web

La aplicación web se divide en varias partes, por lo que se tratará a continuación la instalación de cada parte por separado.

- **Base de Datos:** Para ejecutar el proyecto es necesario tener instalado SQLite, que es la base de datos que se ha utilizado para el entorno de desarrollo ya que no requiere configuración para poderla usar.
- **API:** Para poder ejecutar la API es necesario tener instalada una versión de Python reciente. Concretamente se ha usado la 3.12.
- **Frontend:** Para la aplicación web es necesario tener Angular 18 instalado y Node 22.4.

Figura D.2: Estructura de directorios del *frontend*.

```
1 python -m venv .venv
2 source .venv/bin/activate
```

Figura D.3: Comandos para crear y activar entorno virtual de Python.

```
1 pip install -r requirements.txt
```

Figura D.4: Comando para activar el entorno virtual de Python.

## D.4. Compilación, instalación y ejecución del proyecto

### Descargar el repositorio

Clonar el repositorio de GitHub mediante *git clone* o descargar y descomprimir el ZIP que se puede obtener pulsando el botón *Code* que se encuentra en la página de GitHub [2].

### Entorno de Python

Tanto para ejecutar los notebooks de Jupyter como la API es necesario crear y activar un entorno virtual de Python, para ello hay que ejecutar los comandos D.3. Una vez que se ha activado el entorno virtual es necesario instalarlas las dependencias mediante el comando D.4.

### API

Para ejecutar la API es necesario ir al directorio *app/backend* y renombrar el archivo *ejemplo.env* de dicha carpeta a *.env*. Después se puede o ejecutar el script *start.sh* o ejecutar el comando D.5 en el directorio *app/backend*.

### Frontend

En el caso del *frontend*, al estar desarrollado en Angular, se puede compilar, para ello se puede ejecutar el comando D.7.

```
1 fastapi dev app
```

Figura D.5: Comando para ejecutar el *backend* en modo desarrollo.

```
1 ng serve
```

Figura D.6: Comando para ejecutar el *frontend* en modo desarrollo.

```
1 ng build --prod
```

Figura D.7: Comando para compilar el *frontend* en modo producción.

A la hora de programar en modo desarrollo, resulta más sencillo ejecutar el comando **D.6** que recompila el código cada vez que detecta cambios en un archivo y lo ejecuta.

## Docker

Para crear la red de contenedores de Docker basta con utilizar el siguiente comando **D.8**. Esta red consta de tres contenedores, uno para la base de datos, otro para el *backend* y otro para el *frontend*. Añadir la opción *-build* hace que los contenedores se recompilen, lo que será necesario si se hacen modificaciones en el código.

```
1 docker compose up
```

Figura D.8: Comando ejecutar los contenedores de docker.

## **D.5. Pruebas del sistema**

Las pruebas son fundamentales en el desarrollo de software porque permiten comprobar y validar que un sistema funciona correctamente. A través de las pruebas, se pueden identificar errores, fallos y problemas de rendimiento. Además, contribuyen a garantizar la calidad y fiabilidad del software, mejorando la experiencia del usuario y minimizando la necesidad de corregir errores en etapas avanzadas del desarrollo.

En este proyecto, se han llevado a cabo pruebas manuales para validar el software. Aunque las pruebas manuales pueden ser más laboriosas y consumir más tiempo, proporcionan flexibilidad y permiten una evaluación más detallada del software en términos de usabilidad, compatibilidad y funcionalidad. Esto es especialmente útil para interfaces de usuario complejas o interacciones específicas con el software que son difíciles de automatizar, como en el caso de esta aplicación.



## Apéndice *E*

---

# Documentación de usuario

---

### E.1. Introducción

En esta sección se cubren los aspectos fundamentales relacionados con los requisitos y procedimientos necesarios para la correcta ejecución y uso del programa desarrollado. Se especifican tanto los requisitos que la aplicación necesita, como las instrucciones para su instalación y uso por parte del usuario final.

### E.2. Requisitos de usuarios

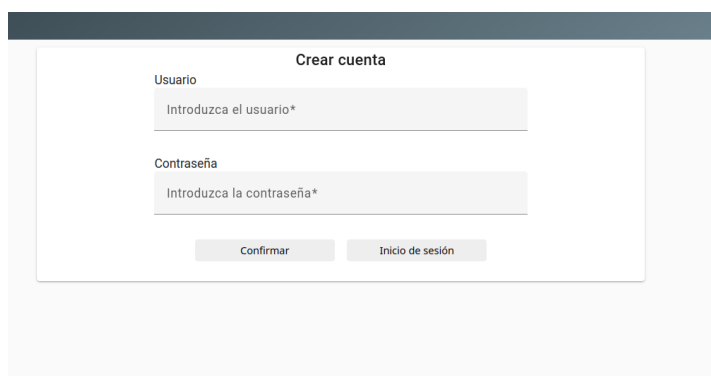
Para desplegar el servidor mediante Docker el usuario solo necesita tener instalado Docker.

Para poder utilizar la aplicación web una vez desplegada bastará con tener instalado un navegador con una versión reciente (preferiblemente de menos de un año de antigüedad).

### E.3. Instalación

Clonar el repositorio de GitHub mediante *git clone* o descargar y descomprimir el ZIP que se puede obtener pulsando el botón *Code* que se encuentra en la página de GitHub [2].

Ir al directorio de la aplicación y renombrar el fichero *ejemplo.env* del directorio *app/backend* a *.env* y moverlo a la carpeta *app*. En este fichero



The image shows a web form titled "Crear cuenta". It contains two text input fields. The first is labeled "Usuario" and has the placeholder text "Introduzca el usuario\*". The second is labeled "Contraseña" and has the placeholder text "Introduzca la contraseña\*". Below these fields are two buttons: "Confirmar" and "Inicio de sesión".

Figura E.1: Pantalla de creación de cuenta.

cambiar `False` por `True` en la variable `PRODUCTION`, que se encuentra en la primera línea.

Ejecutar el comando [D.8](#), la primera vez tardará un poco.

## E.4. Manual del usuario

### Paciente

El paciente podrá seleccionar un ejercicio, lo que le mostrará el vídeo de dicho ejercicio y le permitirá subir un fichero para que sea comparado con los datos que subió el terapeuta.

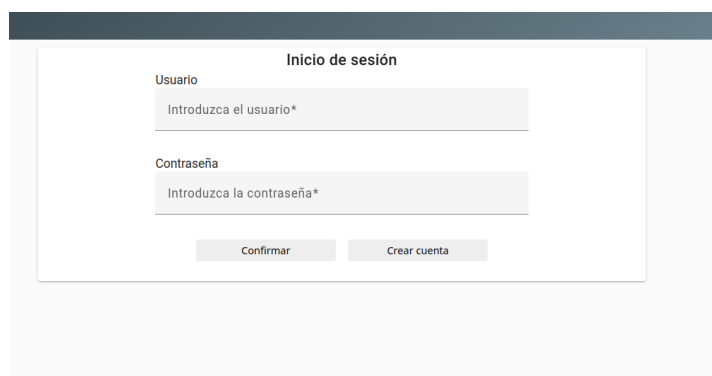
### Creación de cuenta

Introducir el usuario y la contraseña en sus respectivas entradas de texto y pulsar el botón de confirmar, lo que llevara al usuario a la pantalla de inicio de sesión. La pantalla que ve el usuario al iniciar sesión se puede ver en la figura [E.1](#).

### Inicio de sesión

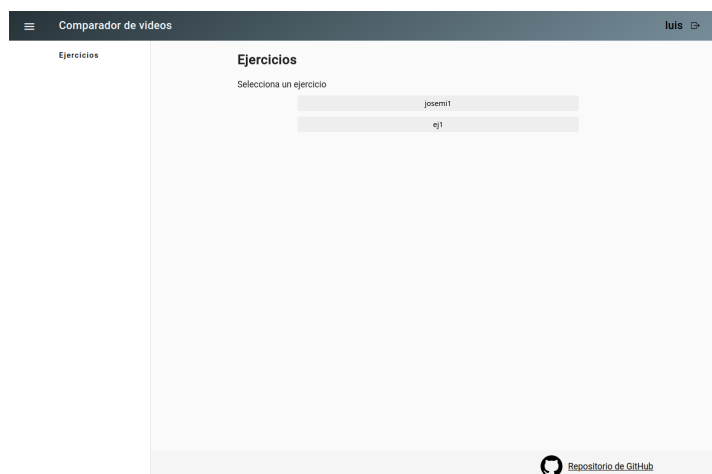
Introducir el usuario y la contraseña en sus respectivas entradas de texto y pulsar el botón de confirmar, lo que llevara al usuario a la pantalla de selección de ejercicios. La pantalla de inicio de sesión se puede ver en la figura [E.2](#).





The screenshot shows a login form titled "Inicio de sesión". It contains two input fields: "Usuario" with the placeholder text "Introduzca el usuario\*" and "Contraseña" with the placeholder text "Introduzca la contraseña\*". Below the fields are two buttons: "Confirmar" and "Crear cuenta".

Figura E.2: Pantalla de inicio de sesión.



The screenshot shows a web page titled "Comparador de videos" with a user profile "luis". The main content area is titled "Ejercicios" and contains the text "Selecciona un ejercicio". Below this text are two buttons labeled "josemi1" and "ej1". At the bottom of the page, there is a footer that says "Repositorio de GitHub".

Figura E.3: Selector de ejercicios

## Comparación de ejercicio

Ir a la pagina *Ejercicios* mediante el menú y seleccionar un ejercicio de la lista pulsando su nombre. Esta pagina se puede ver en la figura E.3.

Añadir un fichero y pulsar el botón de enviar. Esperar a que termine la comparación de vídeos y lleguen los resultados.

Esperar a que se comparé el vídeo y llegue la puntuación, como se puede ver en la figura E.5.

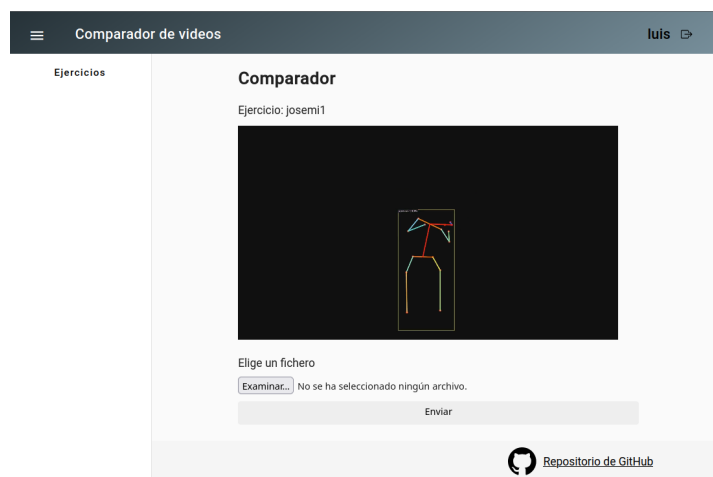


Figura E.4: Vista del comparador de vídeos.

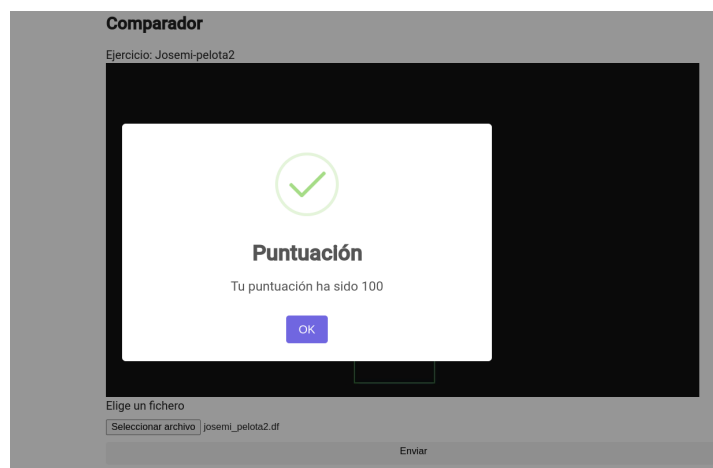


Figura E.5: Puntuación del ejercicio.

The screenshot shows a web application interface for creating a new exercise. The header bar is dark blue with a hamburger menu icon, the text 'Comparador de videos', and a user profile icon labeled 'admin'. On the left, a sidebar contains the links 'Ejercicios', 'Nuevo ejercicio', and 'Borrar ejercicios'. The main content area is titled 'Nuevo ejercicio' and contains the following fields:

- Nombre del ejercicio:** A text input field with the placeholder 'Introduce el nombre del ejercicio\*'. Below it is a small 'Examinar...' button.
- Elige un fichero:** A section with a label and a text input field. Below it is a small 'Examinar...' button.
- Elige un video:** A section with a label and a text input field. Below it is a small 'Examinar...' button.
- Ángulos:** A dropdown menu with the placeholder 'Selecciona los ángulos a evaluar\*'. Below it is a small 'Examinar...' button.
- Coords:** A dropdown menu with the placeholder 'Selecciona las coordenadas a evaluar\*'. Below it is a small 'Examinar...' button.

At the bottom of the form is a 'Confirmar' button. The footer of the page features the GitHub logo and the text 'Repositorio de GitHub'.

Figura E.6: Vista de nuevo ejercicio.

## Terapeuta

El terapeuta podrá crear y borrar nuevos ejercicios para que los pacientes puedan comparar sus ejercicios con los que suba el terapeuta. La cuenta del terapeuta se crea por defecto, en el `.env` se encuentran el usuario y la contraseña del mismo en las variables `FIRST_SUPERUSER` y `FIRST_SUPERUSER_PASSWORD`.

### Creación de ejercicio

1. Escribir el nombre del ejercicio.
2. Añadir el fichero de datos del ejercicio.
3. Añadir un vídeo.
4. Seleccionar los ángulos y coordenadas.
5. Pulsar el botón de confirmar.

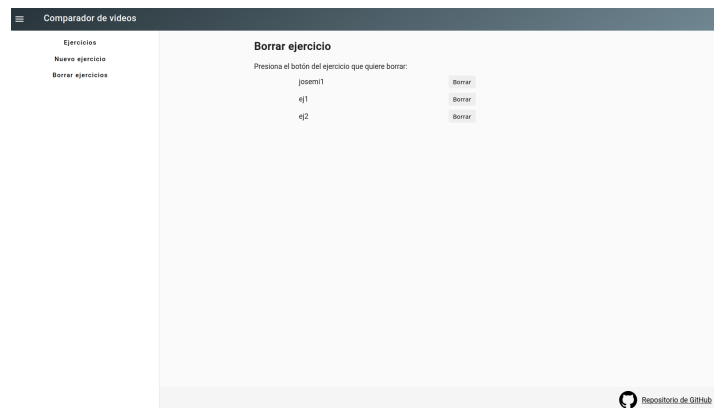


Figura E.7: Pantalla para borrar ejercicios.

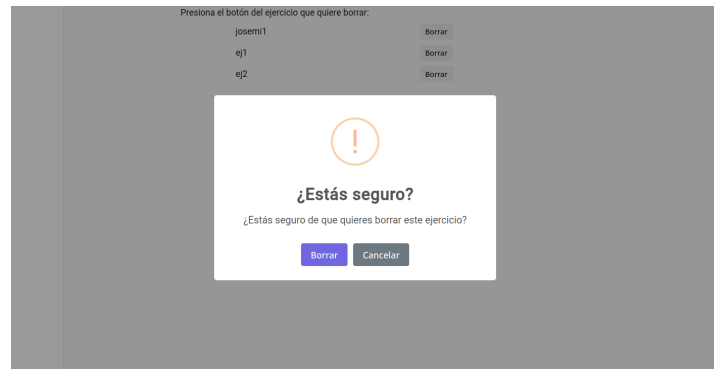


Figura E.8: Confirmación de que se desea borrar el ejercicio.

## Borrar ejercicio

Para eliminar el ejercicio bastará con pulsar el botón de borrar que se encuentra al lado del nombre del ejercicio como se puede ver en la pantalla E.7.

Una vez presionado el botón saldrá una modal en la que habrá que presionar el botón *Borrar* como se puede observar en la figura E.8.

---

## Bibliografía

---

- [1] Indeed. <https://es.indeed.com/career/programador-junior/salaries>.
- [2] Luis Ángel Espinosa Lafuente. Detección de ejercicios en vídeos de rehabilitación. <https://github.com/fravian99/Deteccion-de-ejercicios-en-videos-de-rehabilitacion>.
- [3] Wikipedia contributors. Multitier architecture — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Multitier\\_architecture&oldid=1226040587](https://en.wikipedia.org/w/index.php?title=Multitier_architecture&oldid=1226040587), 2024. [Online; accessed 7-July-2024].