

UNIVERSIDAD POLITÉCNICA DE MADRID

E.T.S. DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

PROYECTO FIN DE GRADO

GRADO EN INGENIERÍA DE COMPUTADORES

# Desarrollo de una aplicación móvil para la detección automática del tipo de pisada

Desarrollado por: Andrés Felpeto Calviño

Dirigido por: Jorge Dueñas Lerín

Madrid, 1 de diciembre de 2025



*Desarrollo de una aplicación móvil para la detección automática del tipo de pisada*

**Desarrollado por:** Andrés Felpeto Calviño

**Dirigido por:** Jorge Dueñas Lerín

Proyecto Fin de Grado, 1 de diciembre de 2025

**E.T.S. de Ingeniería de Sistemas Informáticos**

Campus Sur UPM, Carretera de Valencia (A-3), km. 7

28031, Madrid, España

---

Si deseas citar este trabajo, la entrada completa en B<sub>B</sub>T<sub>E</sub>X es la siguiente:

```
@mastersthesis{citekey,  
  title = {Desarrollo de una aplicación móvil para la detección automática del  
tipo de pisada},  
  type = {Bachelor's Thesis},  
  author = {},  
  school = {E.T.S. de Ingeniería de Sistemas Informáticos},  
  year = {2025},  
  month = {12},  
}
```

---

Esta obra está bajo una licencia [Creative Commons «Atribución-NoComercial-CompartirIgual 4.0 Internacional»](https://creativecommons.org/licenses/by-nc-sa/4.0/). Obra derivada de <https://github.com/blazaid/UPM-Report-Template>.



Todo cambio respecto a la obra original es responsabilidad exclusiva del presente autor.

# Resumen

---

El proyecto consiste en el desarrollo de una aplicación móvil para Android que se enfoca en la detección del tipo de pisada (pronadora, neutra o supinadora) a partir de un video de una persona corriendo en una cinta de correr.

Desde la aplicación se permite seleccionar el video a analizar, que es enviado a un servidor remoto. Dicho servidor será el responsable de procesar el video, obtener los resultados y generar un video nuevo que permitirá al usuario visualizar el movimiento de la persona con marcadores corporales. Para ello se usa principalmente MediaPipe Pose, que es una librería de [aprendizaje automático \(AA\)](#) de Python.

En el proyecto se combinan tecnologías como Kotlin, para el desarrollo aplicación basada en Android, Flask, como backend del servidor y herramientas de [inteligencia artificial \(IA\)](#).

El objetivo alcanzado ha sido proporcionar una manera accesible para el análisis del tipo de pisada, con aplicaciones potenciales en el ámbito deportivo y de la salud.

**Palabras clave:** Ángulo del retropié; Servidor VPS; Aplicación Móvil; RunAlyze

# Abstract

---

The project presents the development of an Android app focused on the classification of the footstep type (pronated, neutral or supinated), based on a video of a persons running on a treadmill.

The application allows users to select a video, which is then sent to a remote server. The server is responsible for processing it, calculating the rearfoot angle, classifying the footstrike type and for generate a new video with body markers to visually support the analysis. This solution relies primarily on `MediaPipe Pose`, an open-source [machine learning \(ML\)](#) library for Python.

This project integrates various technologies, including Kotlin for Android app development, Flask as the server-side backend, and [artificial intelligence \(AI\)](#) tools for biomechanical analysis.

The resulting tool provides an accessible way to estimate footstrike type, with potential applications in both sports and health domains.

**Keywords:** Rearfoot angle; VPS server; Mobile application; RunAlyze

# Índice general

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Contexto . . . . .	1
1.2	Problema . . . . .	1
1.3	Objetivo . . . . .	2
<b>2</b>	<b>Estado de la cuestión</b>	<b>3</b>
2.1	Criterios científicos para la clasificación de la pisada . . . . .	3
2.2	Importancia clínica y deportiva del tipo de pisada . . . . .	5
2.3	Tecnologías actuales para el análisis de la pisada . . . . .	7
<b>3</b>	<b>Material y métodos</b>	<b>9</b>
3.1	Entorno de Desarrollo . . . . .	9
3.2	Herramientas de procesamiento de video . . . . .	10
3.3	Estructura del Proyecto . . . . .	11
3.4	Análisis de Tipo de Pisada . . . . .	19
3.5	Presupuesto . . . . .	22
<b>4</b>	<b>Resultados</b>	<b>24</b>
4.1	Aplicación . . . . .	24
4.2	Desarrollo del Servidor . . . . .	31

Índice general	II
4.3 Pruebas . . . . .	36
5 Discusión	55
5.1 Automatización del cálculo del ángulo del retropié . . . . .	55
5.2 Aplicación . . . . .	57
5.3 Servidor . . . . .	58
6 Conclusiones	60

# Índice de figuras

---

2.1	Medición del ángulo del retropié (ADR) a partir de referencias anatómicas marcadas sobre el talón. . . . .	4
2.2	Arco medial del pie señalado en rojo. Adaptado de: Carranza (2021) . . . .	6
2.3	Sensores alrededor de la pierna para el sistema CODA motion. Fuente: Rose, Birch y Kuisma (2011) . . . . .	8
3.1	Puntos corporales de detección de Mediapipe. Fuente: Google (2024) . . .	11
3.2	Estructura general de la aplicación móvil y el servidor . . . . .	11
3.3	Diagrama De flujo del proceso de análisis de video . . . . .	13
3.4	Diagrama de Clases de la aplicación . . . . .	15
3.5	Estructura general de la aplicación móvil y el servidor . . . . .	17
3.6	Recta entre rodilla y talón. . . . .	21
4.1	Título de la aplicación. . . . .	24
4.2	Icono de la aplicación. . . . .	25
4.3	HomeActivity con señalización de sus botones . . . . .	26
4.4	Captura de ExplainingActivity de pisada pronadora. . . . .	27
4.5	Captura de la barra de carga . . . . .	28
4.6	Resultados en ResultActivity . . . . .	29
4.7	Botones de ResultActivity . . . . .	30
4.8	Vista del video utilizado para el análisis del ADR durante la carrera. . . .	36

---

4.9 Barra de direcciones del navegador . . . . .	53
4.10 Captura de paquetes con el servidor en Wireshark . . . . .	53

# Índice de tablas

---

3.1	Estimación del presupuesto del proyecto . . . . .	23
4.1	Promedio de porcentajes de error en la detección de pisada en distintos videos . . . . .	38
4.2	Clases de equivalencia de la función selectVideoFromGallery() . . . . .	39
4.3	Casos de prueba de la función selectVideoFromGallery() . . . . .	39
4.4	Clases de equivalencia de la función sendVideoToServer() . . . . .	40
4.5	Casos de uso de la función sendVideoToServer() . . . . .	40
4.6	Clases de equivalencia de la función de startActivity(ResultActivity) . . . . .	41
4.7	Casos de uso de la función de startActivity(ResultActivity) . . . . .	42
4.8	Clases de equivalencia del botón de retroceso de Android . . . . .	42
4.9	Casos de uso del botón de retroceso de Android . . . . .	43
4.10	Clases de equivalencia de la petición /upload_video . . . . .	44
4.11	Casos de uso de la petición /upload_video . . . . .	44
4.12	Clases de equivalencia de la petición /request_video . . . . .	45
4.13	Casos de uso de la petición /request_video . . . . .	46
4.14	Clases de equivalencia de la petición /get_results . . . . .	47
4.15	Casos de uso de la petición /get_results . . . . .	48
4.16	Clases de equivalencia de la petición /get_frames_zip . . . . .	49
4.17	Casos de uso de la petición /get_frames_zip . . . . .	50

---

4.18	Tiempo de procesamientos de videos medido en segundos . . . . .	51
4.19	Tiempo de procesamientos de videos de manera concurrente medido en segundos . . . . .	52
4.20	Resultados de la prueba de experiencia de uso . . . . .	54

# 1.

# Introducción

---

## 1.1. Contexto

El estudio del tipo de pisada tiene una alta relevancia en la medicina deportiva. La forma en que una persona pisa puede tener un impacto significativo en el rendimiento y en la prevención de lesiones. En corredores especialmente, cuya disciplina genera una repetición elevada de pisadas, conocer el tipo de pisada (pronador, supinador y neutro) es altamente recomendable.

A lo largo de los años se han desarrollado diversas técnicas para su evaluación. Estos métodos en su mayoría requieren la asistencia de un profesional y de equipos costosos, limitando su accesibilidad a un público más amplio.

En este contexto, se ha implementado el uso de visión por computador para elaborar herramientas de análisis de pisada más accesibles. Herramientas como *Mediapipe*, que permite la detección de puntos corporales del cuerpo humano, ofrece una alternativa para estudiar el comportamiento biomecánico de los corredores mediante la captura de un video. De esta manera, se evita la necesidad de usar equipos especializados y se obtiene resultados más rápidos.

## 1.2. Problema

Como se mencionó anteriormente, las herramientas actuales requieren equipos costosos y acceso a un profesional especializado. Aunque estas herramientas son esenciales en contextos clínicos, la mayoría de las personas no requieren un diagnóstico tan exhaustivo. Aun así, el conocimiento del tipo de pisada es valioso para la salud de cualquier individuo, pues puede prevenir lesiones.

El problema es que la mayoría de las personas no tienen acceso a este tipo de análisis o desconocen la importancia de conocer el tipo de pisada. Esto limita la capacidad pa-

ra tomar decisiones informadas sobre salud, principalmente en aquellos que practican deportes que implican correr, donde el tipo de pisada influye directamente en su salud.

Es por ello que una alternativa es la elaboración de un análisis usando herramientas de [aprendizaje automático \(AA\)](#). Aunque algunas tiendas especializadas en corredores ya ofrecen este tipo de pruebas, el acceso sigue siendo limitado a un público reducido. Por lo tanto, se busca ampliar el acceso a un público más amplio, elaborando una herramienta más accesible.

## 1.3. Objetivo

El objetivo principal del proyecto es el desarrollo de una aplicación móvil capaz de realizar un análisis automático del tipo de pisada. Para ello, se implementan tecnologías de visión por computadora y de [inteligencia artificial \(IA\)](#).

El propósito es ampliar el acceso a este tipo de estudios, otorgando a los usuarios una solución accesible, eficaz y sin restricciones geográficas. De esta manera, tanto deportistas como personas interesadas en conocer su tipo de pisada pueden obtener un análisis sin usar equipos costosos ni acudir a ningún profesional.

## 2. Estado de la cuestión

---

### 2.1. Criterios científicos para la clasificación de la pisada

#### 2.1.1. Definiciones biomecánicas

La pisada humana es un proceso complejo en el que intervienen de forma coordinada huesos, articulaciones, músculos y ligamentos, permitiendo el apoyo de los pies sobre una superficie. En este contexto, se reconocen tres tipos de pisadas: neutra, pronadora y supinadora. Esta clasificación depende del ángulo y la alineación que se genera entre la rodilla, el tobillo y el talón durante la fase en que el pie entra en contacto con el suelo.

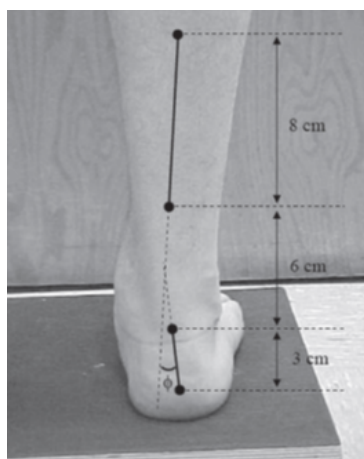
- Se dice que la pisada es neutra cuando la pierna permanece alineada con el talón en el momento de la pisada. Esto genera que la distribución de presiones de la planta del pie se reparta de forma equilibrada.
- La pronación ocurre cuando en el momento de la pisada, el pie rota hacia el interior. Visualmente se aprecia cómo el tobillo se inclina hacia dentro, provocando que la presión plantar sea mayor en la parte interna.
- Por último, está la supinación, que, opuesto a la pronación, ocurre cuando el pie genera una rotación hacia el exterior en el momento de la pisada. Esto genera una inclinación del tobillo hacia fuera, provocando una mayor presión en la parte externa plantar (Redmond, Crane y Menz 2008).

Una correcta identificación de estos patrones tiene una gran relevancia en factores clínicos y deportivos, ya que, aunque no representan unas patologías por sí mismas, pueden influir significativamente en la aparición de lesiones.

## 2.1.2. Métodos clínicos de evaluación

Existen distintos métodos clínicos y caseros que se han desarrollado para evaluar el tipo de pisada y su posible riesgo de lesión. Estos métodos suelen requerir una consulta médica de manera presencial, que pueden aplicarse en una posición estática del paciente, aunque también existen algunas que se aplican mediante observación del paciente durante la marcha o carrera.

El método que se tendrá en cuenta en el proyecto es el del cálculo del **ángulo del retropié (ADR)**, que es obtenido desde detrás de la pierna, mientras el pie esté apoyado en el suelo. El ángulo está formado por dos rectas, una que va desde la pantorrilla hasta la altura del tobillo y la otra va desde el tobillo hasta el talón, como se puede observar en la Figura 2.1.



**Figura 2.1.** Medición del **ADR** a partir de referencias anatómicas marcadas sobre el talón.

A partir de este parámetro, se puede clasificar la pisada como pronadora, neutra o supinadora, de manera automatizada y no invasiva. Esta clasificación replica de manera similar los métodos clínicos usados. De hecho, uno de los instrumentos más conocidos usa algo similar.

La herramienta es **Foot Posture Index (FPI)**. Es una prueba que consiste en una evaluación mediante observaciones y palpaciones, que permite clasificar la postura del pie como supina, neutra o pronadora (Redmond, Crane y Menz 2008). Si bien esta herramienta no calcula directamente el **ADR**, uno de sus criterios evalúa la inclinación del talón respecto al eje de la pierna, en posición estática.

En este sentido, el proyecto toma como referencia este principio clínico y lo adapta a un

sistema que implementa visión computacional. Esto permite emular el razonamiento clínico con una herramienta accesible y automatizada.

## 2.2. Importancia clínica y deportiva del tipo de pisada

### 2.2.1. Influencia en la prevención de lesiones

Una correcta identificación del tipo de pisada previene una lesión por sobreuso, especialmente en el ámbito deportivo. Se ha demostrado que una pisada desalineada puede provocar sobrecargas en algunas partes del cuerpo como las rodillas, tobillos y cadera, aumentando el riesgo de patologías. (Ferber, Hreljac y Kendall 2009)

En una pisada pronadora, se tiende a tener un exceso de rotaciones internas de la tibia, que está asociado a lesiones como la fascitis plantar, una de las patologías más frecuentes derivadas del mal alineamiento del pie. (Ugalde Álvarez y Cuadra Blanco 2009)

En cuanto a las pisadas supinadoras, debido a que el impacto tiende a ser mayor en la parte externa del pie, puede generar una mayor tensión en las rodillas, caderas y tobillos, favoreciendo a lesiones tales como esguinces de tobillo, callosidades laterales, fascitis plantar o fracturas de estrés (Runner's World 2023).

Aunque una pisada neutra no garantiza la ausencia de lesiones, se considera que es más eficiente debido a la distribución equitativa de cargas y absorción del impacto (Princeton Medicine 2023).

### 2.2.2. Personalización del calzado

Tener un conocimiento acerca del tipo de pisada permite seleccionar un calzado más adecuado para la práctica deportiva. Esto es muy importante para correr, sobre todo cuando se trata de largas distancias, debido a la alta cantidad de pisadas que se generan cada vez que se practica este deporte.

En los últimos años, las marcas han podido evolucionar los diseños de sus zapatillas

para generar distintos modelos que se puedan adaptar a los diferentes tipos de pisada y así mejorar la efectividad al correr y reducir las lesiones.

Si bien la mayoría de los calzados comerciales son para tipo de pisada neutra, existen distintas opciones que se pueden adaptar mejor a diferentes tipos de pisada. Principalmente, lo que más cambia es la manera en que el pie absorbe el impacto (Princeton Medicine 2023), explicado en mayor detalle en el siguiente apartado.

### 2.2.3. Plantillas ortopédicas

Aparte de un calzado específico, muchas personas optan por tener unas plantillas ortopédicas, que representan una herramienta clave en las alteraciones biomecánicas asociadas al tipo de pisada. Están diseñadas para modificar el reparto de cargas sobre la planta del pie, mejorar la alineación postural y reducir la tensión sobre estructuras como las rodillas o tobillos.

A diferencia del calzado comercial, que ofrece correcciones genéricas, las plantillas permiten una intervención individualizada. Por ejemplo, en el caso de las pisadas pronadoras, puede incluir un soporte elevado en el arco medial (ver Figura 2.2), para así controlar la rotación excesiva hacia dentro. En las supinadoras, aportan una mayor amortiguación y soporte lateral, mejorando la absorción de impacto y estabilidad del pie. Hay quienes incluso optan por una plantilla neutra, que mejora la comodidad y compensa algunas asimetrías leves (Ferber, Hreljac y Kendall 2009).



**Figura 2.2.** Arco medial del pie señalado en rojo. Adaptado de: Carranza (2021)

Algunos autores han señalado que el uso de plantillas ortopédicas puede contribuir a

la prevención de lesiones por sobreuso en las extremidades inferiores, especialmente en los contextos deportivos (Richter, Austin y Reinking 2011).

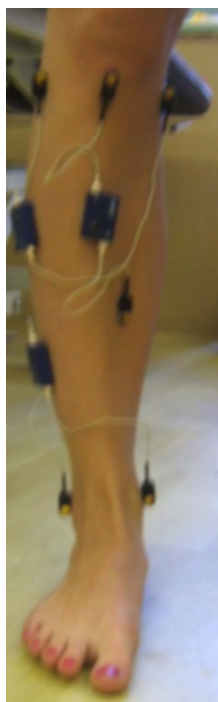
## 2.3. Tecnologías actuales para el análisis de la pisada

### 2.3.1. Herramientas profesionales de laboratorio

Para la evaluación del tipo de pisada, acudir a un método clínico suele ser lo más fiable si se desea un resultado con exactitud. Entre las herramientas más usadas para estas evaluaciones se encuentran capturas de movimiento usando marcadores y el uso de marcadores activos.

Con respecto a la captura de movimiento, en el proyecto de Fong et al. (2008) llevaron a cabo un estudio con 26 niños, clasificado como pronadores, para evaluar la efectividad del uso de plantillas ortopédicas. Para ello, usaron un sistema de captura, llamado *Vi con*, que consta 8 cámaras infrarrojas y 15 marcadores reflexivos posicionados en distintas zonas de las piernas y la cadera, para analizar su movimiento mientras se mantenían de pie, caminando o corriendo. Estas cámaras son capaces de capturar el movimiento de los marcadores con una frecuencia de 120 Hz, que permite calcular a gran velocidad las posiciones de estos marcadores durante las grabaciones. En otro estudio, realizado por Silva et al. (2020), se empleó una solución bastante similar, usando un sistema *Vi con*, con 6 cámaras en vez de 8, que grababan a la misma frecuencia. En este caso, las marcas reflexivas fueron colocadas en la parte interior de la pierna, el talón y en el tendón de Aquiles.

En cuanto al uso de sensores activos, una de las herramientas que se ha empleado para clasificar el tipo de pisada es el sistema *CODA motion*. En el estudio de Rose, Birch y Kuisma (2011), se emplearon marcadores activos alrededor de la pierna y en ambos lados del tobillo (ver Figura 2.3). Estos sensores son captados por los escáneres *CODA*, que se encuentran al frente y detrás del participante, que captan las señales emitidas por los marcadores activos, que permiten la reconstrucción del movimiento en 3D. Esta metodología ofrece una precisión bastante precisa en la estimación de la rotación tibial.



**Figura 2.3.** Sensores alrededor de la pierna para el sistema CODA motion. Fuente: Rose, Birch y Kuisma (2011)

### 2.3.2. Estudio mediante el uso de visión por computador

Como se ha mencionado anteriormente, existen varios métodos de prueba para determinar patologías o problemas en la pisada. Uno de los enfoques más utilizados, y el que se dará mayor énfasis en este proyecto, es el de la observación de la parte trasera de la pierna y analizar los ángulos que se generan durante la fase de apoyo del pie contra el suelo. Tradicionalmente, este procedimiento ha requerido supervisión directa por parte de especialistas, por lo que, a parte de requerir algún especialista para la prueba, también depende de un criterio subjetivo.

Es por ello que la automatización de este proyecto puede eliminar estos inconvenientes, así como lo hicieron Pečar et al. (2018), que propusieron un sistema automatizado capaz de detectar el tipo de pronación. En su estudio, grabaron a un sujeto desde la vista trasera mientras corría en la cinta, procesando el video con un algoritmo que analizaba la pose corporal y calculaba el ángulo de eversión del tobillo. En su trabajo, demostraban que este sistema tiene un error relativo del 1.3 %, sugiriendo que, si bien algunos métodos clínicos pueden obtener un resultado más exacto, este método puede lograr un resultado bastante cercano y de manera más accesible.

## 3.

# Material y métodos

---

## 3.1. Entorno de Desarrollo

### 3.1.1. Android Studio

Android Studio es el entorno recomendado por Android para la creación de aplicaciones en Android y la que fue usada en este proyecto. La versión es la más reciente hasta la fecha en que se realiza el proyecto, es decir, la 2024.3.1 Patch 2.

### 3.1.2. Lenguajes de Programación

Los lenguajes de programación usados fueron Kotlin y Python.

En cuanto a Kotlin, es el lenguaje recomendado por Android para el desarrollo de sus aplicaciones. Kotlin está diseñado para aumentar la productividad de los desarrolladores debido a las facilidades a la hora de programar y también disminuye las probabilidades de que ocurran errores cuando se ejecutan los programas. Es por ello por lo que más del 60.0 de los desarrolladores de Android profesionales usan Kotlin y más del 95 % de las 1000 aplicaciones más populares fueron desarrolladas en Kotlin (Android Developers [s.f.](#)).

Con respecto a Python, usado en el proyecto para el procesamiento y análisis de los videos, es probablemente el lenguaje preferido para el desarrollo de proyectos enfocados con la IA. La alta accesibilidad de librerías de código abierto enfocados en esta área y a la facilidad del manejo de este lenguaje (Raschka, Patterson y Nolet [2020](#)), son unas de las razones de que esto suceda.

## 3.2. Herramientas de procesamiento de video

### 3.2.1. OpenCV

Además de ser Open Source (de código abierto), es la librería más grande para proyectos de visión por computadora. Permite la visualización, análisis o modificación de imágenes o videos (OpenCV [s.f.](#)).

En el proyecto se implementaron funcionalidades que permiten la selección de un video y procesar sus fotogramas individualmente, de manera que se le puedan añadir cambios visuales, y finalmente generar un video nuevo.

Si bien cuenta con funciones ya implementadas de [AA](#), como puede ser el reconocimiento facial o reconocimiento del cuerpo humano, en este caso se usó otra librería para la detección de las partes del cuerpo, que es la mencionada a continuación.

### 3.2.2. MediaPipe

Mediapipe es una librería que contiene una amplia variedad de tareas de [IA](#) y [AA](#). Estas herramientas son usadas para clasificar audios, textos y reconocimiento de visión (Google [2024](#)).

Dentro de la categoría de visión, se incluyen varias herramientas para trabajar con imágenes. En el proyecto se hace uso de las funcionalidades de reconocimiento de patrones o de figuras en la imagen, ya sean personas, caras, manos, objetos, etc. Para la aplicación desarrollada se usó la solución de *detección de pose*, que permite detectar un total de 32 puntos a lo largo del cuerpo de una persona (Ver Figura [3.1](#)).

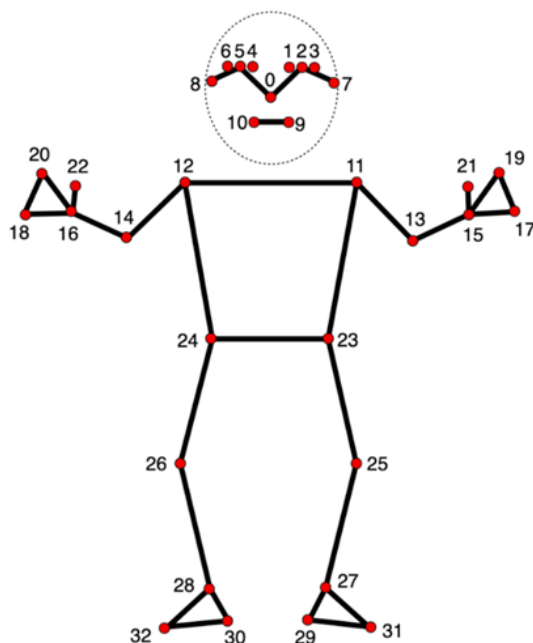


Figura 3.1. Puntos corporales de detección de Mediapipe. Fuente: Google (2024)

## 3.3. Estructura del Proyecto

### 3.3.1. Aplicación

El proyecto se divide en dos grandes componentes: una aplicación móvil y un servidor. (Ver Figura 3.2).

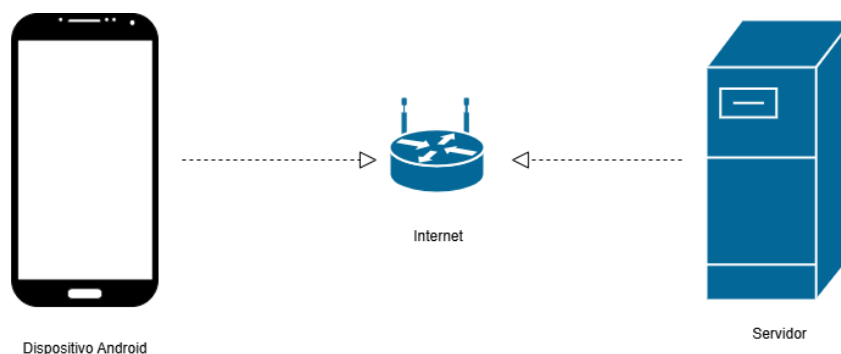


Figura 3.2. Estructura general de la aplicación móvil y el servidor

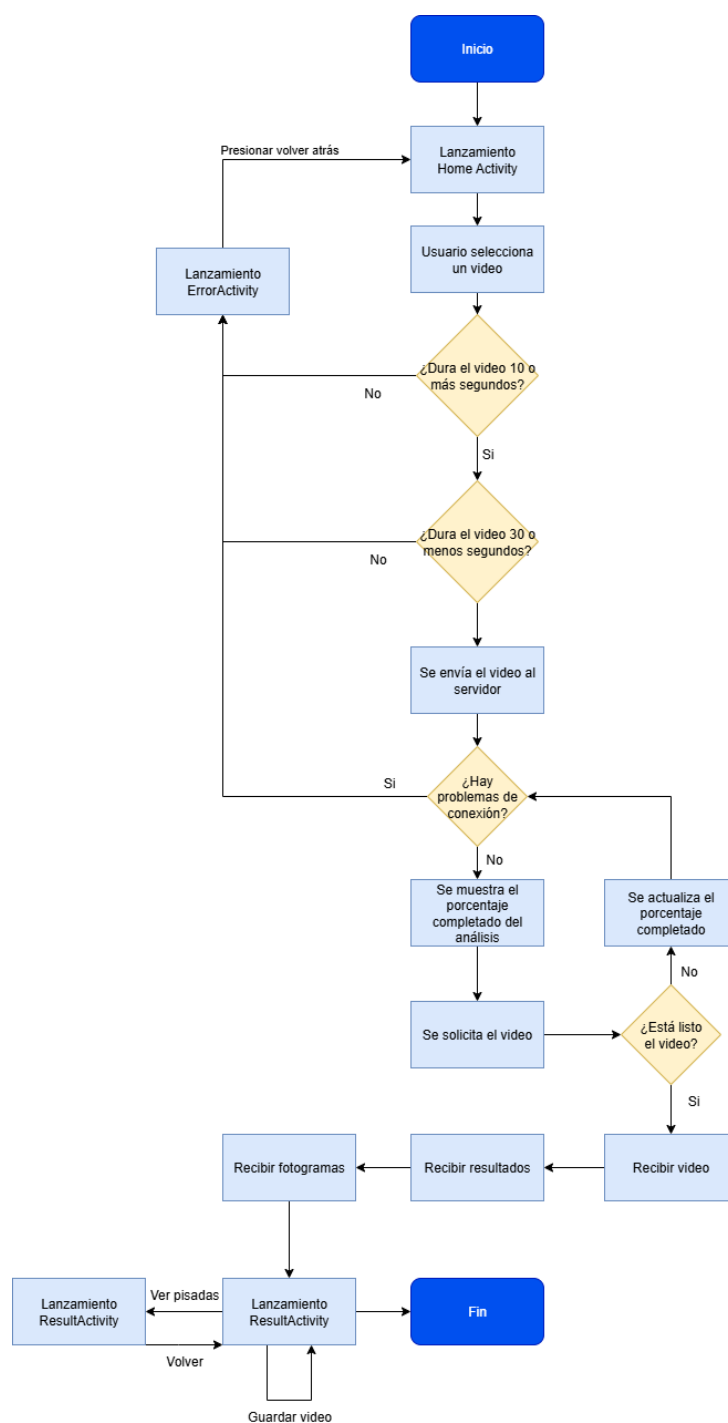
La aplicación móvil, cuyo nombre es RunAlayze, está desarrollada para Android. Su principal funcionalidad es la de interactuar con el usuario, permitiéndole enviar un video seleccionado de la galería para que el servidor lo procese. Luego, recibirá el video analizado y modificado, con los respectivos resultados del tipo de pisada, que serán mostrados en pantalla.

En cuanto al servidor, se encargará de toda la parte del procesamiento. Cuando reciba un video, se inicia el análisis del video. Dicho análisis generará varias respuestas, que incluye un video nuevo, clasificación del tipo de pisada y fotogramas claves. Estos serán enviados de vuelta a la misma dirección IP de origen del mensaje inicial.

De esta manera, se crea una división de responsabilidades de tareas. La aplicación se centra en la interacción con el usuario, y el servidor realiza las tareas con más carga computacional. Esta estructura aporta comodidad y eficacia para la experiencia del usuario, además de ofrecer ventajas de escalabilidad, porque el servidor puede actualizarse o ampliarse sin necesidad de que la aplicación sufra cambios, siempre y cuando el protocolo y la forma de comunicación no cambie.

### 3.3.2. Flujo de la aplicación

A continuación, se muestra el diagrama de flujo que describe el comportamiento de la aplicación durante el procesamiento de un video, desde que se inicia la aplicación hasta que se complete con éxito el análisis.



**Figura 3.3.** Diagrama De flujo del proceso de análisis de video

Inicialmente la aplicación lanza HomeActivity, siendo esta el menú de inicio. El usuario seleccionará un video de la galería que deberá tener una duración entre 10 y 30 segundos para poder seguir el proceso. En el caso opuesto, se lanza ErrorActivity, notificando dicho error y dando la oportunidad de volver a intentarlo.

Al ser validado, el video será enviado mediante peticiones [protocolo de transferencia de hipertexto seguro \(HTTPS\)](#) a un servidor, que será el encargado de analizar el video. El servidor es el responsable de procesar el video y clasificar el tipo de pisada que hay en cada pie.

Mientras se realiza el análisis, la aplicación desplegará una pantalla de carga, mostrando el porcentaje completado a través de una barra visual. El progreso se irá actualizando periódicamente mediante peticiones al servidor, con el fin de crearle al usuario una experiencia más cómoda. En caso de que ocurra un error, ya sea de conexión o con respecto al envío y recibimiento de datos, se lanzará `ErrorActivity`, para informar al usuario.

Al finalizar el procesamiento, el servidor devuelve tres elementos a la aplicación:

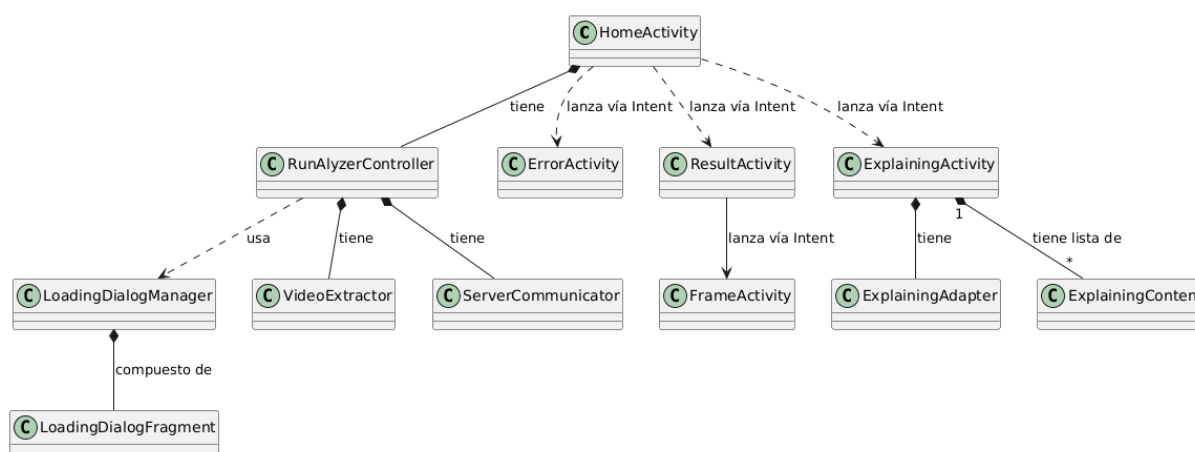
1. El video modificado, donde se marcan los puntos corporales detectados mediante marcadores gráficos.
2. Los resultados del tipo de pisada, que se mostrarán en pantalla de dos maneras:
  - De manera explícita indicando el tipo de pisada.
  - Mediante una barra indicadora, que representará nuevamente el resultado de cada pie, con la diferencia de que se podrá observar la proximidad o desviación del tipo de pisada obtenido como resultado.
3. Ocho fotogramas clave (cuatro por pierna), seleccionados automáticamente como los más representativos del patrón de pisada detectado.

Finalmente, se lanzará la actividad `ResultActivity`, donde se podrá:

- Observar los resultados del análisis.
- Guardar el video generado en la galería.
- Visualizar los fotogramas mediante `FrameActivity`.

### 3.3.3. Diagrama de Clases de la Aplicación

Como se puede observar en la figura [3.4](#), se muestra el diagrama de clases de la aplicación, donde se representan las principales relaciones de los componentes.



**Figura 3.4.** Diagrama de Clases de la aplicación

La primera actividad que es inicializada es HomeActivity, que actúa como menú de inicio de la aplicación.

Su componente central es RunAlyzerController (siendo RunAlyze el título de la aplicación), que tiene como responsabilidad de coordinar el flujo en la ejecución. Los componentes que gestiona son:

- LoadingDialogManager, que controla el ciclo de vida de LoadingDialogFragment, que es una barra de carga que se activa cuando el video está en procesamiento.
- VideoExtractor, que se encarga de la selección y validación del video, asegurando que tenga una duración entre 10 y 30 segundos.
- ServerCommunicator, que establece la comunicación con el servidor, gestionando las peticiones de envío de video, de consulta de su estado y de solicitud de los resultados.

Una vez finalizado el análisis, se lanza la actividad ResultActivity, que se encarga de mostrar al usuario los resultados obtenidos. De esta misma actividad, se puede mostrar FrameActivity, que se usa para visualizar fotogramas representativos del video generado.

Adicionalmente, de HomeActivity se pueden lanzar dos actividades más:

- ExplainingActivity: su función es la de explicar al usuario cómo usar la aplicación, así como también su propósito. Esta actividad utiliza dos componentes, Explainin-

gAdapter, que muestra cada página de la actividad y ExplainingContent, que es un objeto que guarda la información del texto e imagen de cada página.

- ErrorActivity: informa al usuario de un error que caso de que este suceda.

Este diseño permite que cada clase de la aplicación esté delimitado en cuanto a responsabilidad, que favorece su mantenimiento y escalabilidad.

### 3.3.4. Flujo del Servidor

El tipo de servidor que se usará en el backend será una aplicación Flask, que es un framework de Python especializado en la creación de servicios web (Project s.f.).

El Flask será desplegado en un servidor [protocolo de transferencia de hipertexto seguro \(VPS\)](#) con IP pública, usando Docker, que garantizará la portabilidad. La comunicación entre cliente y servidor será mediante peticiones [HTTPS](#), para que se realice de manera segura.

Debido a que se pueden recibir múltiples peticiones simultáneamente desde distintos dispositivos, el servidor está adaptado para manejar cada uno de los procesos de manera independiente. Esto se logra mediante la librería threading de Python, que permitirá atender nuevas peticiones mientras se estén ejecutando otras recibidas anteriormente.

El cliente lanzará tres tipos de peticiones consecutivas cada vez que se desee analizar un video (Ver figura 3.5), siempre que no se produzca algún error en la comunicación:

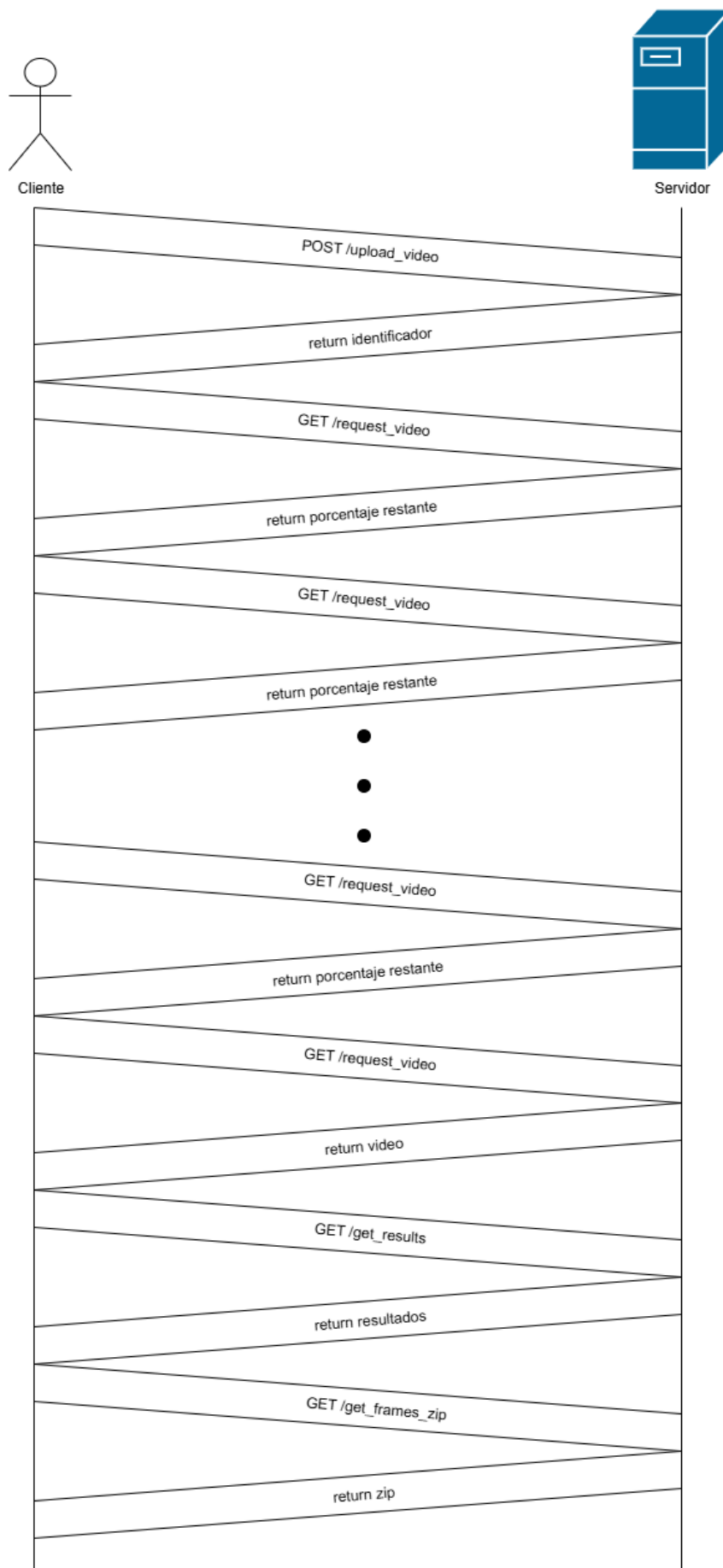


Figura 3.5. Estructura general de la aplicación móvil y el servidor

1. POST /upload\_video: Esta petición envía el video a analizar al servidor. Al recibirlo, el servidor generará un **token** único mediante la librería `Secrets` (Python Software Foundation 2024), que crea un número de 128 bits de forma completamente aleatoria y criptográficamente segura. Esto hace que su obtención por fuerza bruta tenga un coste computacional muy elevado.

El archivo será guardado en el sistema y se empezará a ejecutar el análisis en un hilo en segundo plano. El servidor responderá al cliente la confirmación de recepción del archivo con su **token**. En caso de que no reciba un archivo válido, responderá con un error.

2. GET /request\_video: Esta solicitud se enviará cada 3 segundos luego de enviar la petición POST /upload\_video, con el propósito de solicitar el **video modificado**. Si el análisis no ha terminado, responderá con un mensaje informando el **porcentaje estimado restante** para que finalice el proceso. Una vez se genere el video, el servidor responderá con el nuevo archivo y no se enviarán más peticiones de este tipo.
3. GET /get\_results: la petición la enviará la aplicación una vez reciba el video de la petición /request\_video, con el objetivo de solicitar los **resultados**. El servidor enviará los **resultados numéricos** del análisis de ambos pies con sus respectivas etiquetas (izquierdo o derecho) en formato JSON.
4. GET /get\_frames\_zip: finalmente el servidor recibirá esta petición con el objetivo de proporcionarle unos **fotogramas**, donde se pueda observar el momento de la pisada. En total se envían cuatro fotogramas por pierna (ocho en total), siendo seleccionados aquellos que más se acercaban al resultado final del análisis.

### 3.3.5. Justificación del uso de un servidor remoto

El procesamiento de los videos se realiza en un servidor en lugar del propio dispositivo móvil debido a la demanda computacional que implica. Se debe garantizar un entorno de procesamiento capaz de ejecutar el análisis en un tiempo razonable, que no es posible en la mayoría de los dispositivos móviles.

El móvil usado durante el desarrollo (Redmi Note 8 Pro) cuenta con un procesador Helio G90T y 6 GB de RAM (Corporation 2019) . Si bien ofrece un rendimiento razonable para tareas comunes, no está capacitado para cargas sostenidas de análisis de video.

Se realizó la prueba para evaluar la viabilidad de usar el móvil como herramienta de análisis. En total, el tiempo de ejecución fue de aproximadamente 4 minutos para generar un video de 7 segundos, en escala de grises y con reduciendo el número de frecuencia de fotogramas (1 de cada 4). El resultado obtenido pone en evidencia al dispositivo para este tipo de tareas.

No se excluye la posibilidad de que los dispositivos de gama alta de última generación puedan ofrecer un rendimiento suficiente. Por ejemplo, el procesador del Samsung Galaxy S25 es Qualcomm Snapdragon 8 Elite Mobile for Galaxy (Electronics 2025). Este procesador es caracterizado por su capacidad de procesamiento de tareas complejas, de ejecutar video juegos y de edición de videos (Xataka 2025). Sin embargo, depender exclusivamente de estos dispositivos restringiría el acceso al análisis a un grupo muy limitado de usuarios.

Por ello, se optó por usar un servidor. En este proyecto, el servidor usado cuenta con un procesador AMD EPYC-Genoa, una [Central Processing Unit \(CPU\)](#) de alto rendimiento orientado a entornos de servidor (AMD 2024). Aunque solo son asignados 4 núcleos de los 96 disponibles, el rendimiento y la estabilidad del entorno permite ejecutar el análisis de manera eficiente y estable para múltiples usuarios.

## 3.4. Análisis de Tipo de Pisada

### 3.4.1. Lógica de Clasificación

La clasificación del tipo de pisada se define mediante la medición del [ADR](#), es decir, el ángulo que se genera entre las rectas generadas por el talón y el tobillo y el tobillo con la pantorrilla en el momento de apoyo del pie en el suelo.

Según el valor de dicho ángulo, se establecen tres tipos de pisada:

- Pronado, ángulo mayor a 7°.
- Supinado, ángulo menor a 3.5°.
- Neutro, ángulo entre 3.5° y 7°.

Estos valores se fundamentan en criterios clínicos utilizados en una lectura científica para la evaluación de la alineación del retropié en adolescentes (Fujishita et al. 2023)

### 3.4.2. Detección de Pisada

Para poder realizar el cálculo del [ADR](#), es necesario detectar con precisión el instante en que el pie entre en contacto con el suelo. Esta identificación se logra usando los puntos detectados del cuerpo usando la librería `Mediapipe`.

El procedimiento es:

1. Detección de la **altura mínima del talón**. Este valor es calculado para tomar como referencia un aproximado de la altura del suelo en la imagen.
2. Detección del **talón ascendiendo**. Este valor se obtendrá comparando la altura del talón del anterior fotograma con la actual, detectando que está ascendiendo si ahora está más arriba que antes.
3. Detección de pisada. Se considerará pisada cuando se cumplan los siguientes requisitos:
  - El talón se encuentra cerca de su altura mínima (Calculado en punto 1).
  - Se produce el movimiento ascendente del pie (Calculado en punto 2).
4. Cada vez que se detecte una pisada, se dejarán algunos fotogramas de espera para aceptar un nuevo evento y evitar algún falso positivo.
5. En cada fotograma asignado como pisada, se comparará con los dos anteriores fotogramas (f-2, f-1 y f). Se elegirá aquel que tenga el talón a menor altura.

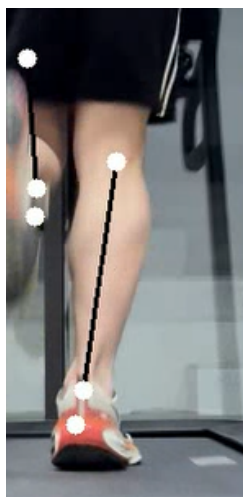
Estos pasos permitirán obtener los instantes de cada video en el que se esté produciendo una pisada de manera automática y precisa, para que dichos fotogramas sean usados para calcular el [ADR](#).

### 3.4.3. Estimación del ángulo del retropié

Luego de obtener los fotogramas donde se producen las pisadas, estos serán usados para calcular el [ADR](#) y poder clasificar el tipo de pisada.

Se recorrerán todos los fotogramas guardados y en cada uno se harán los siguientes pasos:

1. Reconocimiento de puntos del cuerpo usando Mediapipe. En este caso solo son necesarios la rodilla y el tobillo.
2. Generación de las rectas entre rodilla y tobillo. (Ver Figura 3.6).
3. Calcular el ángulo generado por la intersección de la recta calculada en el punto anterior, con el eje "y". Este será el [ADR](#).
4. Una vez obtenido todos los resultados, se obtendrá la media de los valores, sin tener en cuenta aquellos valores que se desvíen más de  $k = 2\sigma$ , siendo  $\sigma$  la desviación estándar.



**Figura 3.6.** Recta entre rodilla y talón.

Este proceso se hará de manera independiente en cada pierna, para finalmente poder clasificarlas según su tipo de pisada y dar por finalizado el análisis.

#### 3.4.4. Generación de Video Analizado

A medida que se vaya calculando el [ADR](#), también se irá generando el video nuevo, donde se podrá seguir el movimiento que produce el corredor con líneas conectadas entre los puntos corporales marcados (rodilla, talón y tobillo).

Para generar el video, se recorre cada uno de los fotogramas del video original y se siguen los siguientes pasos:

1. Al igual que en la estimación de [ADR](#), se detectan los puntos del cuerpo usando `MediaPipe` (igualmente son rodillas, tobillos y talones).
2. Se trazan rectas entre los puntos marcados de rodilla-tobillo y tobillo-talón, en ambas piernas.
3. Cuando se detecte pisada (como se explicó en el apartado [3.4.2](#)) se escribirá en el video un texto en el video que indique que hay pisada, junto con el resultado del cálculo del [ADR](#).
4. Cada fotograma que se modifique pasará a ser parte de un nuevo vídeo.

Una vez se recorran todos los fotogramas, se tendrá como resultado el video modificado, listo para ser enviado como respuesta a la petición del cliente.

## 3.5. Presupuesto

Finalmente, como cierre de la sección 3 de Materiales y métodos, se realiza una estimación del presupuesto asociado al desarrollo del proyecto. Esto incluye tanto los materiales utilizados, como una valoración aproximada del trabajo técnico realizado. Si bien algunos aspectos son estimaciones (como el del salario de un ingeniero de computadores), o algunos recursos ya estaban bajo disposición, se incluyen para reflejar un costo que pueda replicar a un entorno no académico.

La tabla [3.1](#) resume los elementos considerados:

Material	Descripción	Precio (€)
Portátil Dell XPS 13	Utilizado como servidor de prueba y estación de desarrollo	1900
Sueldo ingeniero de computadores	Estimado de un sueldo de ingeniero de computadores recién graduado por trabajo de 5 meses	8000
Smartphone Xiaomi Redmi Note 8 pro	Dispositivo donde se probó la aplicación y usado para captura de videos	200
Iphone 15 pro	Dispositivo para captura de videos	1220
Servidor VPS	Contrato de servicio del servidor virtual durante 2 meses	18.94
Dominio del servidor	Referenciar el servidor con la dirección andrestfg.es	7.25
Programas de Adobe	Se usó Adobe Premiere y Adobe Illustrator con licencia de estudiante	gratis
	<b>Total</b>	11346.19

**Tabla 3.1.** Estimación del presupuesto del proyecto

## 4.1. Aplicación

### 4.1.1. Título e Icono

El título con el que se nombró a la aplicación es RunAlyze (ver figura 4.1), haciendo un juego de palabras entre los términos en inglés run (correr) y analyze (analizar), que reflejan la funcionalidad principal de la aplicación.



***RUN*** alyze

**Figura 4.1.** Título de la aplicación.

En cuanto al icono, se seleccionó un fondo blanco, característico de la estética de la interfaz, con la palabra RUN, con la misma tipografía usada dentro de la aplicación (figura 4.2).



**Figura 4.2.** *Icono de la aplicación.*

### 4.1.2. Menú de Inicio

Al iniciar la aplicación, se despliega la actividad HomeActivity, que funciona como el menú de inicio. Desde esta interfaz se puede acceder a dos apartados:

- Selección de video en la galería, cuyo acceso es mediante el botón con el símbolo “+”, como se señala en la figura 4.3.
- A la actividad ExplainingActivity, a través del botón con el texto “¿En qué consiste un análisis de tipo de pisada?” (ver figura 4.3), cuya función es la de explicar el propósito y funcionamiento de la aplicación. Esta sección es explicada en mayor detalle en el siguiente apartado.

# ***RUN*** alyze



**Figura 4.3.** HomeActivity con señalización de sus botones

## 4.1.3. Actividad de Explicación del Funcionamiento

El propósito de esta interfaz (`ExplainingActivity`) es la de guiar al usuario los pasos necesarios para realizar un análisis correcto del tipo de pisada. Está compuesto por una serie de pantallas, acompañadas de textos, imágenes o videos explicativos, abordando los siguientes aspectos:

- Los tipos de pisada: se explican las tres clasificaciones (pronadora, supinadora y neutra), acompañado de una breve explicación y de una figura.
- Requisitos del video: se describe cómo debe grabarse el video. Los cuales son:
  - Ángulo de grabación desde atrás.
  - El cuerpo completamente visible.
  - Duración entre **10 y 30 segundos**.

- Velocidad de la cinta entre **8 y 10 km/h**, que es una velocidad promedio de una persona corriendo.
- El apartado de resultados: se muestran los resultados generados en el análisis, que incluyen:
  - El tipo de pisada en cada pierna.
  - Video modificado con marcadores corporales y posibilidad de guardarlo en la galería.
  - Visualizar los fotogramas clave del video, para ver en detalle la pisada.

La actividad ofrece al usuario dos botones de desplazamiento en la parte de debajo de la pantalla. El de la derecha es para avanzar a la siguiente página y el de la izquierda es para devolverse a la anterior, como se puede ver en la figura 4.4.

**Pisada Pronadora**  
El pie rota hacia dentro en el momento de la pisada.  
Es común, pero un exceso de pronación puede causar lesiones si no se corrige con calzado adecuado.



**Figura 4.4.** Captura de ExplainingActivity de pisada pronadora.

#### 4.1.4. Barra de Carga

Una vez se selecciona un video válido, se despliega un componente cuyo nombre es `LoadingDialogFragment`, en una interfaz de espera. Su responsabilidad es la de mostrar al usuario mediante una barra de progreso el estado del análisis en el momento. Inicialmente en esta interfaz se indica el primer de dos pasos, el cual es el envío del

video al servidor. Una vez se complete este paso, la aplicación empezará a consultar el estado del análisis mediante peticiones periódicas cada 3 segundos, con el que irá actualizando el progreso de la barra en cada respuesta recibida (ver figura 4.5).



**Figura 4.5.** Captura de la barra de carga

### 4.1.5. Visualización de Resultados

Cuando haya finalizado el análisis del video, y se hayan enviado los resultados al móvil, la aplicación mostrará los resultados a través de la actividad `ResultActivity`. Se podrán ver los resultados de dos formas (Ver figura 4.6):

- Texto explícito: a cada lado de la pantalla, se indicará el tipo de pisada para cada pie.
- Barra gráfica: indica la proximidad o desviación del resultado obtenido.

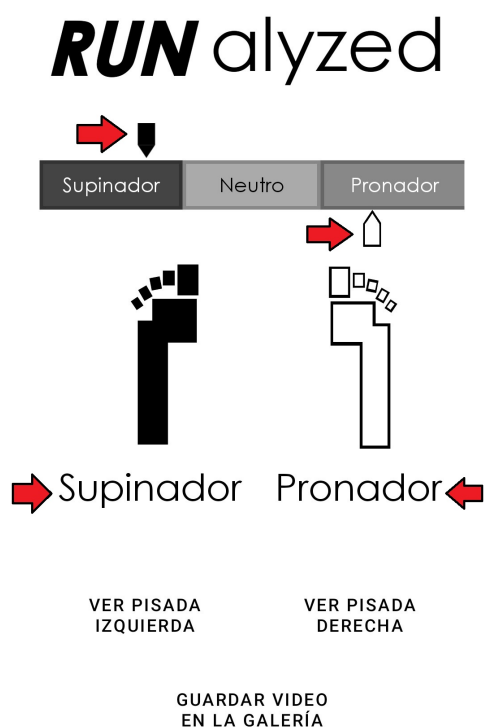


Figura 4.6. Resultados en ResultActivity

Además de los resultados, se podrá realizar las siguientes acciones (ver figura 4.7):

- Guardar el video generado en el análisis: Incluirá marcadores corporales en las rodillas, tobillos y talones, que permitirán al usuario **visualizar** el movimiento al correr de manera más clara. Esto se consigue apretando el botón GUARDAR VIDEO EN LA GALERÍA.
- Ver fotogramas claves: a través de FrameActivity, que muestra algunas de las pisadas con los ángulos **más cercanos** al resultado final del análisis. Se pueden observar al presionar los botones VER PISADA IZQUIERDA o VER PISADA DERECHA.



Figura 4.7. Botones de ResultActivity

### 4.1.6. Posibles Errores

Durante el proceso de análisis, es decir, cuando se activa la barra de carga, pueden ocurrir diferentes errores, que interrumpirían el procesamiento. En caso de error, para que la aplicación pueda seguir su flujo, se redirigirá al usuario a la actividad `ErrorActivity`, con un mensaje claro del problema. Los problemas pueden ser:

- Video inválido: el video no cumpla con el requisito de duración (entre 10 y 30 segundos).
- Problemas de conexión: la aplicación no puede establecer una conexión con el servidor u ocurre un error al enviar el video. Se visualiza en pantalla.
- Error en la consulta del progreso de análisis: puede ser causado por un problema en la comunicación o en el servidor.

- Error al recibir los resultados: hubo un problema en el análisis del video, ya sea en la generación del video, de obtención de los resultados del tipo de pisada o del archivo zip con los fotogramas generados.

## 4.2. Desarrollo del Servidor

### 4.2.1. Características

El servidor de RunAlyze está diseñado para gestionar y procesar los videos móviles enviado desde la aplicación móvil y enviar de vuelta sus resultados. Está implementado usando el lenguaje de programación Python, utilizando la librería Flask, la cual facilita la creación de servicios web ligeros y eficientes.

El servidor está desplegado en un VPS de OVH Cloud, que es un proveedor servidores dedicados (Cloud 2025). El uso de un servidor privado virtual permite un control completo sobre los recursos y garantiza la disponibilidad continua del servicio, manteniendo la disponibilidad en todo momento para atender las solicitudes de los usuarios. Este servidor está ubicado en Madrid, ciudad donde se realiza el proyecto, con el fin de reducir la latencia en la comunicación, mejorando así la eficiencia en la conexión.

Además, el servidor está configurado en un contenedor Docker, una tecnología que permite ejecutar aplicaciones en entornos aislados, incluyendo todo lo necesario para su funcionamiento. Esto garantiza que el servidor se ejecute de manera idéntica en cualquier entorno, facilitando su portabilidad, mantenimiento y despliegue.

Las peticiones entre aplicación y servidor son HTTPS, garantizando la seguridad de los datos transmitidos. Para ello, se utiliza Certbot (E. F. Foundation n.d.), una herramienta de gestión automática de certificados digitales emitidos por Let's Encrypt (Group n.d.), una autoridad de certificación gratuita que permite cifrar la comunicación entre cliente y servidor.

Adicionalmente, se le asignó un dominio personalizado ([andrestfg.es](https://andrestfg.es)), que permite la generación de certificados Secure Sockets Layer (SSL) (Capas de Conexión Segura), necesarios para la habilitación de una comunicación HTTPS.

## 4.2.2. Proceso de configuración

En primer lugar, se instaló Python en el entorno del servidor [VPS](#) adquirido. Adicionalmente, se instalaron las librerías necesarias, especificadas en un archivo denominado `requirements.txt`, para facilitar la instalación.

Luego, se generaron los certificados mediante la herramienta Certbot, ejecutando el siguiente comando:

---

```
sudo certbot certonly --standalone -d andrestfg.es
```

---

Este comando solicita y descarga un certificado [SSL](#) emitido por Let's Encrypt, además de validar el dominio `andrestfg.es`, que pertenece al servidor. A su vez, genera dos archivos: `fullchain.pem` y `privkey.pem`.

El archivo `fullchain.pem` contiene la cadena de certificados públicos requeridos por los clientes para verificar su autenticidad con el servidor. En cuanto a `privkey.pem`, almacena únicamente la clave privada asociada al certificado, que es confidencial y lo usa exclusivamente el servidor para descifrar las claves recibidas.

Una vez generados los certificados, se configuró el contenedor Docker, que aloja el servidor de la aplicación.

Primero, se crea el archivo `requirements.txt`, donde se declaran todas las dependencias, es decir, las librerías de Python usadas en la creación de la aplicación Flask.

Luego, se crea `Dockerfile`, un archivo donde se define la imagen base del sistema, dependencias necesarias y puntos de entrada de la aplicación. La configuración de `Dockerfile` se puede ver a continuación:

---

```
# Version de Python
FROM python:3.10-slim
# Establecer directorio de trabajo
WORKDIR /app
# Copiar los archivos del proyecto
COPY . /app
# Instalar dependencias del sistema necesarias para OpenCV
RUN apt-get update && apt-get install -y --no-install-
```

```
recommends \  
  libgl1 \  
  libgl1 2.0-0 \  
  libsm6 \  
  libxext6 \  
  libxrender1 \  
  certbot && \  
  rm -rf /var/lib/apt/lists/*  
# Instalar dependencias de Python  
RUN pip install --no-cache-dir -r requirements.txt  
# Exponer el puerto HTTPS  
EXPOSE 443  
# Comando para ejecutar Flask  
CMD ["gunicorn", "--certfile=/etc/letsencrypt/live/andrestfg.es/  
  /fullchain.pem", \  
      "--keyfile=/etc/letsencrypt/live/andrestfg.es/  
  privkey.pem", \  
      "--bind=0.0.0.0:443", "server_multiuser:app"]
```

---

En el archivo, se define la imagen base del entorno (`python:3.10-slim`), se instalan las dependencias del sistema necesarias para OpenCV y las de `requirements.txt` y se expone el puerto 443 para el tráfico [HTTPS](#). Finalmente, se lanza la aplicación con Gunicorn.

Gunicorn es un servidor [interfaz de pasarela del servidor web \(WSGI\)](#), actúa como intermediario en la comunicación entre los clientes y el servidor Flask vía [protocolo de transferencia de hipertexto \(HTTP\)](#). A diferencia del servidor por defecto que incluye Flask, Gunicorn está diseñado para manejar múltiples peticiones, permitiendo una mayor eficiencia, escalabilidad y mayor seguridad (Gunicorn [s.f.](#)).

En la configuración de `Dockerfile`, Gunicorn se ejecuta dentro del contenedor Docker, usando los certificados generados anteriormente. Específicamente, se indica que escuche en el puerto 443 (puerto estándar de [HTTPS](#)), y se asigna el punto de entrada al módulo `server_multiuser`, que contiene la instancia de la aplicación Flask.

Por otro lado, para aumentar las medidas de seguridad, se configuró un cortafuegos usando [cortafuegos sin complicaciones \(UFW\)](#). Dicha herramienta permite restringir el acceso a puertos específicos, para limitar el acceso al sistema. Como sólo es necesario

el acceso mediante el puerto 443, únicamente se habilitó esta entrada, de la siguiente forma:

---

```
sudo ufw allow 443/tcp
sudo ufw enable
```

---

Posteriormente, se construye el contenedor, con el siguiente comando:

---

```
docker build -t runalyze_app .
```

---

Finalmente, se ejecuta el contenedor, usando el puerto 443.

---

```
sudo docker run -d \
  --name runalyze_app \
  -p 443:443 \
  -v /etc/letsencrypt:/etc/letsencrypt \
  --restart always \
  runalyze_app
```

---

De este modo, el servidor queda completamente operativo dentro de un entorno aislado y reproducible, garantizando la seguridad, estabilidad y portabilidad del sistema.

### 4.2.3. Manejo de peticiones

El servidor de RunAlyze gestiona diferentes tipos de peticiones provenientes de la aplicación móvil. Cada una de estas peticiones realiza validaciones y acciones específicas, respondiendo con código estándar del protocolo [HTTP](#), informando al usuario del resultado de la petición:

- 200: En caso de que se haya ejecutado correctamente
- 400: En caso de la solicitud del cliente sea incorrecta.
- 500: En caso de que el fallo sea del servidor

A continuación, se detallan las respuestas manejadas por cada petición:

1. /upload\_video:

Esta petición permite al cliente enviar un video a analizar, en el que se comprueba que:

- Se reciba un video. En caso contrario devuelve error 400.

Si se cumple lo anterior, se asigna un token único, se empieza el análisis en segundo plano y se envía una respuesta con código 200 al cliente.

A su vez, en esta petición, donde se inicializa el análisis del video, se pondrá el estado del proceso en error, en caso de que se produzca algún inconveniente. Esto será tenido en cuenta en las peticiones posteriores.

2. /request\_video:

Permite al cliente consultar el video generado, así como conocer su estado de procesamiento en caso de no haber finalizado el análisis. En esta petición se comprueba que:

- El token sea válido. En caso contrario devuelve error 400.
- No haya error en el estado de análisis. En caso contrario devuelve error 500.
- Haya terminado el análisis. En caso contrario devuelve el código 200 con el porcentaje completado hasta el momento.
- Se haya guardado correctamente el video. En caso contrario devuelve error 500.

Si se cumple todo lo anterior, se envía el video generado, con el código 200.

3. /get\_results:

Permite al cliente obtener los ángulos de pisada calculados del video. El servidor comprueba que:

- El token sea válido. En caso contrario devuelve error 400.
- Se haya completado el análisis. En caso contrario devuelve error 400.

Si se cumple todo lo anterior, se envían los resultados, con el código 200.

4. /get\_frames\_zip:

Permite al cliente recibir un archivo ZIP con los fotogramas clave generados durante el análisis del video. Se comprueba que:

- El token sea válido. En caso contrario devuelve error 400.
- Se haya guardado correctamente el zip de los fotogramas. En caso contrario devuelve error 400.

Si se cumple todo lo anterior, se envía el zip con los fotogramas, con el código 200.

## 4.3. Pruebas

### 4.3.1. Pruebas precisión en cálculos de la pisada

A continuación, se presentan los resultados obtenidos en cuanto a la precisión de la detección del momento de la pisada y de los cálculos del [ADR](#) en distintos videos de prueba.

Cada video corresponde a una persona corriendo sobre una cinta a una velocidad de entre 8 y 10 km/h, grabada desde atrás, donde se observa el cuerpo completo, como se muestra en la figura [4.8](#).



**Figura 4.8.** Vista del video utilizado para el análisis del [ADR](#) durante la carrera.

En esta sección se evaluarán dos aspectos:

1. **Error en la detección de la pisada** Se mide la precisión del sistema al identificar los fotogramas correspondientes al momento de apoyo del pie. Para ello, se calculó el porcentaje de error a partir del número de aciertos y del total de fotogramas detectados como pisada.

Se considerará acierto siempre que el pie esté en contacto con el suelo y se ha detectado correctamente el pie correspondiente (izquierdo o derecho).

La formula empleada fue:

$$\text{Error detección pisada} = \left( 1 - \frac{\text{Aciertos}}{\text{Fotogramas totales}} \right) \times 100$$

2. **Error en el cálculo del ADR** También se evalúa la precisión en el cálculo automático del ADR. Para obtenerlo, se obtiene el porcentaje de error al comparar los valores de ángulos calculados automáticamente por el sistema con aquellos medidos manualmente.

Esta métrica se obtiene mediante la siguiente fórmula:

$$\text{Error cálculo ADR} = \left( \frac{|\text{Valor automático} - \text{Valor manual}|}{|\text{Valor manual}|} \right) \times 100$$

En cuanto al valor manual, este fue obtenido usando Kinovea (Charmant y contributors 2024), un software que permite trazar ángulos y medir distancias sobre imágenes. Se utilizaron los mismos fotogramas detectados automáticamente por RunAnalyzer como momento de pisada. En ellos, se trazaron los ángulos entre el tobillo y rodilla, replicando el criterio del cálculo del ADR, cuyo criterio es el usado en la aplicación.

Asimismo, se describen las características de cada video, nombrando aquellas que pueden tener un impacto en la precisión del análisis.

Número de prueba	Promedio de error en detección de pisada	Promedio de error en cálculo ADR	Resolución del video	Calidad del video	Iluminación	Distancia al sujeto	Lleva pantalón largo o corto
1	24 %	7.25 %	720p	Buena	Adecuada	Media	Corto
2	0 %	8.37 %	4K	Buena	Adecuada	Cerca	Corto
3	5 %	9.16 %	4K	Buena	Adecuada	Cerca	Corto
4	3 %	13.91 %	4K	Buena	Adecuada	Cerca	Corto
5	7 %	15.50 %	4K	Buena	Adecuada	Cerca	Corto
6	36 %	15.80 %	720p	Baja	Mala	Larga	Corto
7	9 %	16.27 %	4K	Buena	Adecuada	Cerca	Corto
8	11 %	17.42 %	720p	Baja	Adecuada	Media	Largo
9	21 %	17.42 %	4K	Buena	Adecuada	Cerca	Largo
10	50 %	26.75 %	720p	Baja	Regular	Media	Largo

**Tabla 4.1.** Promedio de porcentajes de error en la detección de pisada en distintos videos

Como se puede observar, el porcentaje de error varía en función de las condiciones del entorno en la que fue capturado el video. Aunque los resultados en entornos adecuados fueron bastante aceptables, en algunos contextos más exigentes evidenciaron un incremento del error. Estos resultados permiten identificar las características que más afectan a la precisión de los resultados, siendo analizados en mayor profundidad en el apartado 5 de discusión.

### 4.3.2. Pruebas unitarias de la aplicación

En esta sección se presentan las pruebas funcionales realizadas sobre la aplicación móvil RunAllyze. El objetivo es comprobar que las funcionalidades de la aplicación respondan correctamente según los parámetros proporcionados y las condiciones del entorno en las que se ejecutan.

Para ello, se han definido las clases de equivalencia válidas e inválidas, conformada por esos parámetros o contextos que determinan los resultados de cada función.

A continuación, se detallan las pruebas realizadas de estas funciones de la aplicación móvil:

1. **Función selectVideoFromGallery():** Selección de video de la galería por parte del

usuario.

Entrada: Duración del video en segundos.

Entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
Video [duración]	V1. 10 segundos V2. 30 segundos	N1. 9 segundos N2. 31 segundos

**Tabla 4.2.** Clases de equivalencia de la función selectVideoFromGallery()

Casos de Prueba	Combinaciones de clase de equivalencia	Salida
CP1	V1. 10 segundos	El video está listo para ser enviado al servidor.
CP2	V2. 30 segundos	El video está listo para ser enviado al servidor.
CP3	N1. 9 segundos	Se indica al usuario que el video debe durar entre 10 y 30 segundos.
CP4	N2. 31 segundos	Se indica al usuario que el video debe durar entre 10 y 30 segundos.

**Tabla 4.3.** Casos de prueba de la función selectVideoFromGallery()

Como se observa, la función se ejecuta correctamente, debido a que al tener de entrada videos cuya duración está fuera del rango valido (10 y 30 segundos), se indica al usuario que no es posible analizar dicho video. En cambio, cuando el video si cumple con la duración establecida, se permite avanzar al siguiente paso del proceso.

## 2. Función sendVideoToServer(): Envío de video validado al servidor.

Entrada: Dominio del servidor, si el móvil está conectado a wifi, si está activado el servidor

Entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
Dominio, conexión a wifi, activado el servidor	V1. https://andrestfg.es, true, true	N1. https://andrestfg.es, true, false  N2. https://andrestfg.es, false, true  N3. https://andrestfg.es, false, false  N4. http://andrestfg.es, true  N5. https://andrestfg.es, true, false  N6. https://andrestfg.es, false, true  N7. https://andrestfg.es, false, false

**Tabla 4.4.** Clases de equivalencia de la función sendVideoToServer()

Casos de Prueba	Combinaciones de clase de equivalencia	Salida
CP1	V1. https://andrestfg.es, true, true	Video enviado a servidor
CP2	N1. https://andrestfg.es, true, false	Se muestra un mensaje de error "Error al subir el video"
CP3	N2. https://andrestfg.es, false, true	Se muestra un mensaje de error "Error al subir el video"
CP4	N3. https://andrestfg.es, false, false	Se muestra un mensaje de error "Error al subir el video"
CP5	N4. http://andrestfg.es, true, true	Se muestra un mensaje de error "Error al subir el video"
CP6	N6. https://andrestfg.es, false, true	Se muestra un mensaje de error "Error al subir el video"
CP7	N6. https://andrestfg.es, false, true	Se muestra un mensaje de error "Error al subir el video"
CP8	N7. https://andrestfg.es, false, false	Se muestra un mensaje de error "Error al subir el video"

**Tabla 4.5.** Casos de uso de la función sendVideoToServer()

En esta función, solo es posible que sea validada en caso de que se ponga el dominio correcto (<https://andrestfg.es>), y que se tenga tanto conexión a wifi por parte del dispositivo móvil, como también que el servidor este activo. En cualquier caso, donde no se cumpla alguna de las condiciones, mostrará un mensaje de error en pantalla.

### 3. `startActivity(ResultActivity)`: Visualización resultados en `ResultActivity`.

Entrada: Video generado, resultados del análisis y fotogramas de la pisada.

Entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
Video.mp4, resultados, fotogramas	V1. Archivo.mp4, jsonify{results}, Archivo zip	N1. Archivo.mp4, jsonify{null}, Archivo zip N2. Archivo.mp4, jsonify{results}, Null N3. Archivo.mp4, jsonify{null}, Null N4. Null, jsonify{results}, Archivo zip N5. Null, jsonify{null}, Archivo zip N6. Null, jsonify{results}, Null N7. Null, jsonify{null}, Null

**Tabla 4.6.** Clases de equivalencia de la función de `startActivity(ResultActivity)`

Casos de Prueba	Combinaciones de clase de equivalencia	Salida
CP1	V1. Archivo.mp4, jsonify{results}, Archivo zip	Se muestra la actividad ResultActivity correctamente.
CP2	N1. Archivo.mp4, jsonify{null}, Archivo zip	Se muestra un mensaje de error "Fallo al recibir los resultados"
CP3	N2. Archivo.mp4, jsonify{results}, Null	Se muestra un mensaje de error "Fallo al recibir los fotogramas"
CP4	N3. Archivo.mp4, jsonify{null}, Null	Se muestra un mensaje de error "Fallo al recibir los resultados"
CP5	N4. Null, jsonify{results}, Archivo zip	Se muestra un mensaje de error "Error al analizar el video"
CP6	N5. Null, jsonify{null}, Archivo zip	Se muestra un mensaje de error "Error al analizar el video"
CP7	N6. Null, jsonify{results}, Null	Se muestra un mensaje de error "Error al analizar el video"
CP8	N7. Null, jsonify{null}, Null	Se muestra un mensaje de error "Error al analizar el video"

**Tabla 4.7.** Casos de uso de la función de startActivity(ResultActivity)

Para que se realice la transición a la actividad ResultActivity, es necesario haber recibido por parte del servidor: el video generado, un JSON que contenga los ángulos calculados y el archivo zip con los fotogramas.

En caso de no recibir alguno de estos tres elementos, se detectará error, informando por pantalla al usuario.

#### 4. Presionar botón de retroceso de Android.

Entrada: Actividad de la aplicación en la que se encuentra actualmente.

Entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
Actividad Actual	V1. HomeActivity V2. ExplainingActivity V3. ResultActivity V4. Pantalla de Carga	

**Tabla 4.8.** Clases de equivalencia del botón de retroceso de Android

Casos de Prueba	Combinaciones de clase de equivalencia	Salida
CP1	V1. HomeActivity	Se sale de la aplicación al menú del móvil
CP2	V2. ExplainingActivity	Se devuelve a HomeActivity
CP3	V3. ResultActivity	Se devuelve a HomeActivity
CP4	V4. Pantalla de Carga	Se devuelve a HomeActivity

**Tabla 4.9.** Casos de uso del botón de retroceso de Android

Al presionar el botón de retroceso de Android, la aplicación responde de acuerdo con el contexto en el que se encuentra. Siempre que se ejecute, se devolverá a la actividad anterior a la que se encontraba. Esto incluye a HomeActivity, que su actividad anterior será la de estar fuera de la aplicación.

### 4.3.3. Pruebas funcionales del servidor

En esta sección se presentan las pruebas funcionales realizadas sobre el servidor de Runalyze. El objetivo es verificar que las funciones que pueden tener distintos resultados respondan correctamente de acuerdo con sus parámetros o el contexto en el que están. Para ello, se han definido las clases de equivalencia válidas e inválidas, conformada por esos parámetros o contextos que determinan los resultados de cada función. A continuación, se detallan las pruebas realizadas de estas funciones del servidor:

1. **/upload\_video:** Recepción de un video para analizar

Entrada: Dominio del servidor, petición, video a analizar.

Entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
Dominio, petición, Video.mp4	V1. https://andrestfg.es, /upload_video, video.mp4	N1. https://andrestfg.es, /up, video.mp4  N2. Archivo .mp4, jsonify{results}, Null  N3. https://andrestfg.es, /up, None  N4. http://andrestfg.es, /upload_video, video.mp4  N5. http://andrestfg.es, /up, video.mp4  N6. http://andrestfg.es, /upload_video, None  N7. http://andrestfg.es, /up, None

**Tabla 4.10.** Clases de equivalencia de la petición /upload\_video

Casos de Prueba	Combinaciones de clase de equivalencia	Salida
CP1	V1. https://andrestfg.es, /upload_video, video.mp4	Video recibido, se empieza el análisis y se genera un process_id
CP2	N1. https://andrestfg.es, /up, video.mp4	Error 400, la ruta no existe
CP3	N2. Archivo .mp4, jsonify{results}, Null	Error 400, no se envió ningún archivo
CP4	N3. https://andrestfg.es, /up, None	Error 400, la ruta no existe
CP5	N4. http://andrestfg.es, /upload_video, video.mp4	Error de conexión
CP6	N5. http://andrestfg.es, /up, video.mp4	Error de conexión
CP7	N6. http://andrestfg.es, /upload_video, None	Error de conexión
CP8	N7. http://andrestfg.es, /up, None	Error de conexión

**Tabla 4.11.** Casos de uso de la petición /upload\_video

Con respecto a esta petición, el servidor solamente se ejecutará de manera correcta y empezará el análisis siempre que se escriba el dominio de correctamente (<https://andrestfg.es>), seguido de escribir la petición que corresponde (/upload\_video) y se adjunte un video a analizar. En caso de que falte alguno de estos elementos, la petición resultará en un error, ya sea por conexión fallida o por parámetros inválidos.

## 2. /request\_video: Recepción de la solicitud del video generado.

Entrada: Dominio del servidor, petición, token válido, si el video está listo.

Entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
Dominio, petición, token válido, video está listo	<p>V1. <a href="https://andrestfg.es">https://andrestfg.es</a>, /request_video, true, true</p> <p>V2. <a href="https://andrestfg.es">https://andrestfg.es</a>, /request_video, true, false</p>	<p>N1. <a href="https://andrestfg.es">https://andrestfg.es</a>, /request_video, false, true</p> <p>N2. <a href="https://andrestfg.es">https://andrestfg.es</a>, /re, true, true</p> <p>N3. <a href="https://andrestfg.es">https://andrestfg.es</a>, /re, false, false</p> <p>N4. <a href="https://andrestfg.es">https://andrestfg.es</a>, /re, true, false</p> <p>N5. <a href="http://andrestfg.es">http://andrestfg.es</a>, /request_video, true, true</p> <p>N6. <a href="http://andrestfg.es">http://andrestfg.es</a>, /request_video, true, false</p> <p>N7. <a href="http://andrestfg.es">http://andrestfg.es</a>, /request_video, false, false</p> <p>N8. <a href="http://andrestfg.es">http://andrestfg.es</a>, /re, true, true</p> <p>N9. <a href="http://andrestfg.es">http://andrestfg.es</a>, /re, true, false</p> <p>N10. <a href="http://andrestfg.es">http://andrestfg.es</a>, /re, false, false</p>

**Tabla 4.12.** Clases de equivalencia de la petición /request\_video

Casos de Prueba	Combinaciones de clase de equivalencia	Salida
CP1	V1. <a href="https://andrestfg.es/request_video">https://andrestfg.es/request_video</a> , true, true	Se envía el video a la aplicación móvil
CP2	V2. <a href="https://andrestfg.es/request_video">https://andrestfg.es/request_video</a> , true, false	Se envía el porcentaje completado del análisis
CP3	N1. <a href="https://andrestfg.es/request_video">https://andrestfg.es/request_video</a> , false, true	Error 400, process_id inválido o inexistente
CP4	N2. <a href="https://andrestfg.es/re">https://andrestfg.es/re</a> , true, true	Error 400, la ruta no existe
CP5	N3. <a href="https://andrestfg.es/re">https://andrestfg.es/re</a> , false, false	Error 400, la ruta no existe
CP6	N4. <a href="https://andrestfg.es/re">https://andrestfg.es/re</a> , true, false	Error 400, la ruta no existe
CP7	N5. <a href="http://andrestfg.es/request_video">http://andrestfg.es/request_video</a> , true, true	Error de conexión
CP8	N6. <a href="http://andrestfg.es/request_video">http://andrestfg.es/request_video</a> , true, false	Error de conexión
CP9	N7. <a href="http://andrestfg.es/request_video">http://andrestfg.es/request_video</a> , false, false	Error de conexión
CP10	N8. <a href="http://andrestfg.es/re">http://andrestfg.es/re</a> , true, true	Error de conexión
CP11	N9. <a href="http://andrestfg.es/re">http://andrestfg.es/re</a> , true, false	Error de conexión
CP12	N10. <a href="http://andrestfg.es/re">http://andrestfg.es/re</a> , false, false	Error de conexión

**Tabla 4.13.** Casos de uso de la petición /request\_video

Como se puede observar, la petición /request\_video se ejecutará correctamente y siempre que se escriba el dominio de manera correcta (<https://andrestfg.es>), seguido de escribir la ruta válida (/request\_video) y que se haya generado un token identificador anteriormente.

En cuanto al video con el resultado, si este ya ha sido procesado, será enviado al dispositivo móvil. En caso contrario, se enviará el porcentaje de progreso alcanzado. Ambas situaciones son válidas y esperadas, siempre que los parámetros mencionados sean correctos.

### 3. /get\_results: Recepción de solicitud de resultados del análisis.

Entrada: Dominio del servidor, petición, token válido, si el análisis está listo.

Entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
Dominio, petición, token válido, análisis listo	V1. https://andrestfg.es, /get_results, true, true	N1. https://andrestfg.es, /get_results, false, true  N2. https://andrestfg.es, /get_results, false, false  N3. https://andrestfg.es, /get, true, true  N4. https://andrestfg.es, /get, false, false  N5. https://andrestfg.es, /get, true, false  N6. https://andrestfg.es, /get_results, true, true  N7. https://andrestfg.es, /get_results, true, false  N8. https://andrestfg.es, /get_results, false, false  N9. https://andrestfg.es, /get, true, true  N10. https://andrestfg.es, /get, true, false  N11. https://andrestfg.es, /get, false, false

**Tabla 4.14.** Clases de equivalencia de la petición /get\_results

Casos de Prueba	Combinaciones de clase de equivalencia	Salida
CP1	V1. <a href="https://andrestfg.es/get_results">https://andrestfg.es, /get_results</a> , true, true	Se envían los resultados a la aplicación móvil
CP2	N1. <a href="https://andrestfg.es/get_results">https://andrestfg.es, /get_results</a> , false, true	Error 400, no se pudieron obtener los resultados
CP3	N2. <a href="https://andrestfg.es/get_results">https://andrestfg.es, /get_results</a> , false, false	Error 400, al consultar el proceso
CP4	N3. <a href="https://andrestfg.es/get">https://andrestfg.es, /get</a> , true, true	No se envía la petición
CP5	N4. <a href="https://andrestfg.es/get">https://andrestfg.es, /get</a> , false, false	No se envía la petición
CP6	N5. <a href="https://andrestfg.es/get">https://andrestfg.es, /get</a> , true, false	No se envía la petición
CP7	N6. <a href="https://andrestfg.es/get_results">https://andrestfg.es, /get_results</a> , true, true	Error de conexión
CP8	N7. <a href="https://andrestfg.es/get_results">https://andrestfg.es, /get_results</a> , true, false	Error de conexión
CP9	N8. <a href="https://andrestfg.es/get_results">https://andrestfg.es, /get_results</a> , false, false	Error de conexión
CP10	N9. <a href="https://andrestfg.es/get">https://andrestfg.es, /get</a> , true, true	No se envía la petición
CP11	N10. <a href="https://andrestfg.es/get">https://andrestfg.es, /get</a> , true, false	No se envía la petición
CP12	N11. <a href="https://andrestfg.es/get">https://andrestfg.es, /get</a> , false, false	No se envía la petición

**Tabla 4.15.** Casos de uso de la petición `/get_results`

Como se aprecia en los casos de prueba, la petición `/get_video` solo se ejecutará correctamente cuando se utilice el dominio seguro (<https://andrestfg.es>), se especifique correctamente la ruta (`/get_video`), y exista un token identificador válido con el análisis ya completado.

En dicho caso, el servidor enviará al cliente los resultados obtenidos. En cualquier otra situación, ya sea por errores de dominio, de ruta, porque el análisis no esté listo o el token no sea válido, la petición fallará devolviendo un mensaje de error o no llegará a procesarse.

#### 4. `/get_frames_zip`: Recepción de solicitud de fotogramas del análisis.

Entrada: Dominio del servidor, petición, token válido, si el análisis está listo.

Entrada	Clases de equivalencia válidas	Clases de equivalencia inválidas
Actividad Actual	V1. https://andrestfg.es, /get_frames_zip, true, true	N1. https://andrestfg.es, /get_frames_zip, false, true  N2. https://andrestfg.es, /get_frames_zip, false, false  N3. https://andrestfg.es, /get, true, true  N4. https://andrestfg.es, /get, false, false  N5. https://andrestfg.es, /get, true, false  N6. http://andrestfg.es, /get_frames_zip, true, true  N7. http://andrestfg.es, /get_frames_zip, true, false  N8. http://andrestfg.es, /get_frames_zip, false, false  N9. http://andrestfg.es, /get, true, true  N10. http://andrestfg.es, /get, true, false  N11. http://andrestfg.es, /get, false, false

**Tabla 4.16.** Clases de equivalencia de la petición /get\_frames\_zip

Casos de Prueba	Combinaciones de clase de equivalencia	Salida
CP1	V1. <a href="https://andrestfg.es/get_frames_zip">https://andrestfg.es, /get_frames_zip</a> , true, true	Se envía el archivo ZIP con los fotogramas al dispositivo móvil
CP2	N1. <a href="https://andrestfg.es/get_frames_zip">https://andrestfg.es, /get_frames_zip</a> , false, true	Error 400, no se pudo obtener el archivo ZIP
CP3	N2. <a href="https://andrestfg.es/get_frames_zip">https://andrestfg.es, /get_frames_zip</a> , false, false	Error 400, al consultar el proceso
CP4	N3. <a href="https://andrestfg.es/get">https://andrestfg.es, /get</a> , true, true	No se envía la petición
CP5	N4. <a href="https://andrestfg.es/get">https://andrestfg.es, /get</a> , false, false	No se envía la petición
CP6	N5. <a href="https://andrestfg.es/get">https://andrestfg.es, /get</a> , true, false	No se envía la petición
CP7	N6. <a href="http://andrestfg.es/get_frames_zip">http://andrestfg.es, /get_frames_zip</a> , true, true	Error de conexión
CP8	N7. <a href="http://andrestfg.es/get_frames_zip">http://andrestfg.es, /get_frames_zip</a> , true, false	Error de conexión
CP9	N8. <a href="http://andrestfg.es/get_frames_zip">http://andrestfg.es, /get_frames_zip</a> , false, false	Error de conexión
CP10	N9. <a href="http://andrestfg.es/get">http://andrestfg.es, /get</a> , true, true	No se envía la petición
CP11	N10. <a href="http://andrestfg.es/get">http://andrestfg.es, /get</a> , true, false	No se envía la petición
CP12	N11. <a href="http://andrestfg.es/get">http://andrestfg.es, /get</a> , false, false	No se envía la petición

**Tabla 4.17.** Casos de uso de la petición `/get_frames_zip`

Finalmente, con respecto a la petición `/get_frames_zip`, solo se ejecutará correctamente cuando se utilice el dominio correcto (<https://andrestfg.es>), se escriba correctamente la ruta (`/get_frames_zip`), y exista un token válido con el análisis previamente completado.

Cualquier desviación en la petición, ya sea por un token inválido, una ruta incorrecta, el uso de **HTTP** en lugar de **HTTPS** o por intentar acceder antes de que finalice el análisis, resultará en un error dentro del servidor o un fallo de conexión, según corresponda.

### 4.3.4. Pruebas de rendimiento

Con el objetivo de evaluar el rendimiento del servidor en el proceso de carga, se realizaron dos tipos de pruebas:

- Una que mida los tiempos de procesamiento de videos individuales de distintas duraciones.
- Una que analice la efectividad en el manejo de la concurrencia por parte del servidor.

#### 1. Tiempo de procesamiento según duración del video:

A continuación, se muestra una tabla con las mediciones de tiempo obtenidas en distintas etapas del procesamiento, desde que el servidor recibe el video hasta que se envían todos los resultados al cliente:

Duración video	Tiempo reconocimiento pisadas	Tiempo de cálculo de ángulos de pisada	Tiempo de comunicación y manejo de datos	Tiempo Total
10	5.05	9.26	6.3	20.61
20	10.45	17.21	9.9	38.06
30	15.92	26.73	14.32	56.97

**Tabla 4.18.** Tiempo de procesamientos de videos medido en segundos

Se ve como el tiempo total de procesamiento crece de manera proporcional a la duración del video. Tanto el reconocimiento de pisadas como el cálculo de ángulos presentan un incremento lineal en tiempo, lo cual es coherente al tratarse de operaciones que dependen del análisis cuadro a cuadro. Por su parte, el tiempo de comunicación y manejo de datos también es bastante lineal, aunque en menor medida. Cabe destacar que este tiempo solamente incluye tareas como la lectura de archivos, envío de datos y respuestas intermedias.

#### 2. Prueba de concurrencia:

Para comprobar el comportamiento del servidor bajo múltiples solicitudes concurrentes, se realizaron pruebas enviando varios videos en paralelo. Es importante destacar que en esta prueba solo se tiene en cuenta la capacidad del servidor, obviando el tiempo de transmisión desde o hacia el dispositivo móvil.

Número de videos enviados	Tiempo medio de análisis por video	Tiempo menor	Tiempo mayor
2	19.84	18.26	21.42
5	28.864	23.66	37.38
10	48.41	7.88	57.22
20	91.93	21.29	106.19

**Tabla 4.19.** Tiempo de procesamientos de videos de manera concurrente medido en segundos

Se observa que, si bien el servidor es capaz de gestionar múltiples procesos simultáneamente, el tiempo medio de procesamiento por video aumenta conforme se incrementa la cantidad de videos enviados. Aunque en algunos casos el tiempo mínimo por video se reduce, tanto el tiempo máximo como el tiempo promedio muestran un crecimiento proporcional al número de peticiones concurrentes. Adicionalmente, se realizó una prueba con 30 solicitudes simultáneas, en la cual el servidor no fue capaz de completar el procesamiento, fallando en su ejecución.

### 4.3.5. Prueba de seguridad

La prueba de seguridad tiene como objetivo verificar que la comunicación entre la aplicación móvil y el servidor sea cifrada, garantizando la seguridad de los datos transmitidos. Para ello, se comprueba de dos maneras:

1. **Mediante el navegador:** Al introducir el dominio del servidor ([andrestfg.es](http://andrestfg.es)) a través de un navegador, se puede verificar en los detalles en la barra de direcciones, que el certificado SSL/TLS está correctamente instalado y que el protocolo [HTTPS](https://) está establecido. El navegador indica que la conexión es segura, con un certificado válido, como se ve en la figura 4.9.

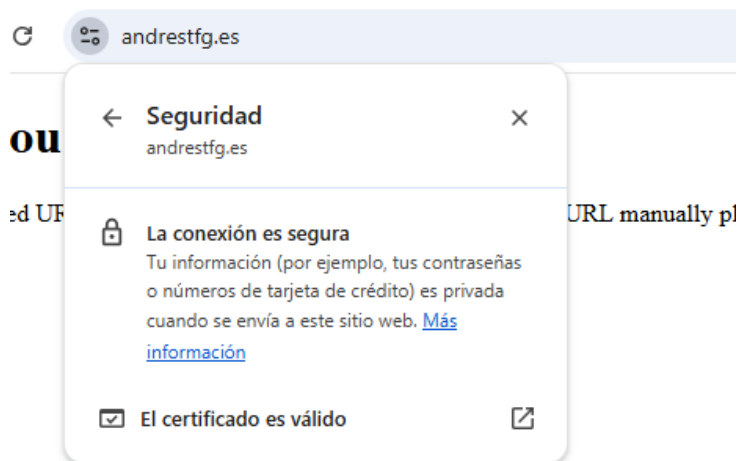


Figura 4.9. Barra de direcciones del navegador

2. **Mediante análisis de red en Wireshark:** Adicionalmente, se usó Wireshark, una aplicación de análisis de paquetes de red (W. Foundation s.f.), para capturar el tráfico hacia el servidor. Para ello, se aplicó un filtro en la captura de paquetes del tráfico, de aquellos que involucren la dirección IP del servidor y el puerto 443 (puerto estándar de HTTPS), Como se ve en la figura 4.10.

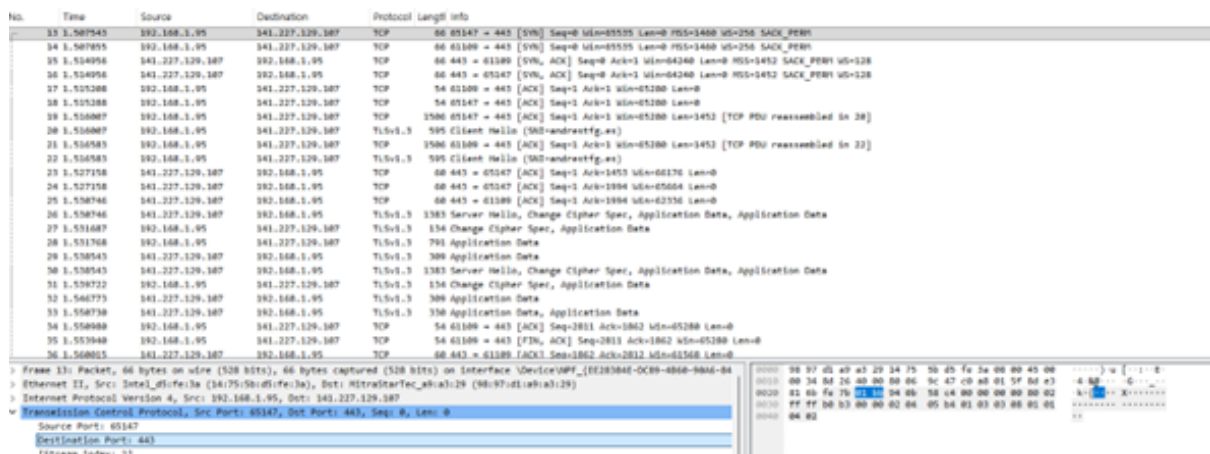


Figura 4.10. Captura de paquetes con el servidor en Wireshark

En la figura se puede ver el primer paquete localizado, con destino a la dirección del servidor (141.227.129.107) y en la parte de abajo que el puerto es el 443. Esto confirma que los paquetes están cifrados, validando el uso del protocolo HTTPS.

### 4.3.6. Prueba de experiencia de uso

Esta prueba tiene como finalidad conocer la experiencia de uso del usuario en el uso de la aplicación. La claridad, su eficacia y el aspecto fueron los factores tenidos en cuenta para la evaluación en la satisfacción en el uso de la aplicación.

Para ello, se seleccionaron cinco participantes, a los que se pidió puntuar cada característica con una nota de 1 a 5.

Participante	Interfaz	Claridad del contenido	Eficacia
1	5	4	5
2	5	4	5
3	5	5	5
4	5	4	5
5	4	4	5

**Tabla 4.20.** Resultados de la prueba de experiencia de uso

En general, los resultados muestran que la experiencia es satisfactoria y que la aplicación se percibe como intuitiva y fácil de manejar. Sin embargo, una confusión en común en 4 de las 5 pruebas fue la existencia de dos resultados distintos (uno para el pie izquierdo y otro para el derecho). Este detalle no estaba claramente explicado en la interfaz inicial, por lo que se decidió incorporarlo posteriormente para mejorar la comprensión por parte del usuario.

## 5.1. Automatización del cálculo del ángulo del retropié

### 5.1.1. Evaluación de la precisión

En el capítulo de resultados, dentro del apartado 4.3.1, se entiende el análisis de la precisión de dos niveles directamente relacionados:

- a) Detección del momento de la pisada.
- b) Cálculo del ADR, que depende del primer punto.

Si la detección de pisada falla, el cálculo del ADR también se ve afectado, por lo que ambos procesos deben evaluarse conjuntamente.

En los resultados de la tabla 4.1, se evidencia que las condiciones del entorno tienen un impacto significativo en la precisión del análisis. La detección de pisada mostró un rendimiento elevado en videos con buena resolución, iluminación adecuada y con el sujeto vistiendo pantalón corto, logrando errores entre el 0% y el 11%. Estas condiciones favorecen que el algoritmo identifique correctamente el contacto del pie con el suelo y seleccione los fotogramas adecuados para el análisis.

Por otro lado, cuando alguna de estas características no era óptima, el porcentaje de error aumentó notablemente, alcanzando valores de del 36% e incluso 50% en el caso más desfavorable. Por ende, el resultado en el cálculo del ADR se ve afectado negativamente, porque un fotograma incorrecto incrementa la desviación respecto a la medición manual.

Esta relación se refleja claramente en los resultados del cálculo del ADR. En videos donde la detección de pisada fue fiable, los errores en el ADR se situaron entre el 7% y el 15%, que son aceptables en el contexto de la estimación automática de puntos corporales. Por el contrario, en videos donde la detección de pisada

presentó fallos, el error del **ADR** aumentó hasta un **26.75 %**, evidenciando la dependencia entre ambos procesos. Aún así, no es una regla que aplique siempre, pues en el caso del video 6, el error en la detección de pisada fue alto, mientras que el del cálculo del **ADR** fue aceptable.

Cabe señalar que cierto margen de error es esperable en este tipo de automatizaciones, en especial cuando se detecta puntos corporales mediante el uso de visión por computador. En otros trabajos, como en Zong, Yang y Baytar 2023, donde busca mediante *Mediapipe* capturar poses del cuerpo, muestra el porcentaje de error en el cálculo de distintos puntos del cuerpo. En su trabajo concluye que, si bien *Mediapipe* muestra un rendimiento razonablemente preciso, también muestra limitaciones, especialmente bajo condiciones no ideales.

### 5.1.2. Posibles mejoras

Para mejorar la eficacia de los resultados, una de las principales líneas de mejora sería el desarrollo de un sistema propio de detección de puntos corporales basados en **AA** para la detección de los puntos corporales del cuerpo. Al ser un enfoque específico, se podría trabajar con redes neuronales enfocadas para el contexto de carrera en una cinta de correr, optimizando así la identificación de los puntos clave. No obstante, representaría un proyecto más complejo y ambicioso, pues se necesitarían recopilar un volumen considerable de datos y una mayor inversión de tiempo y recursos técnicos.

Otra línea de mejora posible sería la incorporación de modelos probabilísticos que, aunque no intervienen directamente en el cálculo geométrico del ángulo, podrían mejorar la clasificación final del tipo de pisada, sirviendo como mecanismo de validación adicional del resultado.

Además, como validación de forma más rigurosa del sistema, se podrían realizar pruebas del tipo de pisada con podólogo o especialistas en el área. Esto serviría como validación para determinar si el resultado automatizado coincide con el resultado del análisis clínico. Sin embargo, llevar a cabo este tipo de validación supondría una inversión monetaria considerable, puesto que el precio medio de consulta puede variar entre 40 y 100 euros en Madrid, lugar donde se realiza este proyecto. Además, teniendo en cuenta que son varios sujetos para realizar pruebas, representaría un factor limitante en una muestra amplia.

## 5.2. Aplicación

### 5.2.1. Evaluación de la experiencia de uso

La aplicación móvil desarrollada cumple con su propósito funcional de permitir al usuario seleccionar un video, enviarlo al servidor y recibir un análisis automático del tipo de pisada. Al ser una interfaz simple, intuitiva y que incluye una guía explicativa sobre su uso, facilita la interacción para el usuario. Las pruebas del apartado 4.3.6, reflejaron una comprensión satisfactoria del funcionamiento general de la aplicación.

No obstante, se pueden encontrar algunas limitaciones. Entre ellas está la dependencia del correcto funcionamiento de un servidor y de una conexión WIFI estable, debido a que la lógica es analizada fuera del dispositivo. Tampoco involucra un sistema que verifique si las condiciones del entorno del video, como la presencia de una persona corriendo en una cinta, la iluminación, etc, son las ideales para realizar el análisis. Esto puede ocasionar resultados erróneos sin que el usuario conozca la causa.

### 5.2.2. Público objetivo

La aplicación está principalmente enfocada a personas aficionadas al mundo de correr, corredores ocasionales o simplemente usuarios que quieran tener conocimiento de su tipo de pisada. Al tratarse de una aplicación de diseño intuitivo, acceso gratis y ausencia de requerimiento técnicos, permite generar un análisis básico sin necesidad de conocimientos especializados.

Como se mencionó previamente en la sección 5.2.1, si bien es aproximado el resultado y puede orientar al usuario, no deben interpretarse como un análisis clínico. Las evaluaciones profesionales realizadas por podólogos o centros especializados incluyen pruebas más precisas, utilizando equipamiento avanzado que garantiza una mayor fiabilidad.

Sin embargo, la mayoría de las personas no requieren una valoración médica formal, por lo que pagar por un análisis clínico podría no ser estrictamente necesario. En este sentido, RunAlyze representa una alternativa práctica y sin coste para una estimación útil del tipo de pisada, suficiente para orientar cambios en el calzado deportivo

### 5.2.3. Líneas futuras de desarrollo

Se identifican tres aspectos clave que pueden ser abordados en futuras versiones de la aplicación:

- a) Modo de grabación interno: actualmente, la grabación del video debe hacerse de forma externa. Se podría integrar un sistema de grabación interno que involucre también el siguiente aspecto.
- b) Guía interactiva durante la grabación: Incluir una guía en tiempo real que indique al usuario si la grabación incluye las condiciones necesarias para un análisis efectivo. Por ejemplo, avisos de una perspectiva correcta, de iluminación es adecuada, de aparición de un sujeto en una cinta, que podrían potenciar la fiabilidad del análisis.
- c) Desarrollo en IOS: la aplicación solo está disponible para dispositivos con el sistema operativo Android. Expandir esta herramienta a IOS permitiría llegar a más usuarios y mejorar la escalabilidad de la herramienta.

## 5.3. Servidor

### 5.3.1. Evaluación del funcionamiento

El servidor desarrollado para RunAlyze ha demostrado ser una pieza clave en la automatización de detección del tipo de pisada. Su diseño mediante Flask ha permitido gestionar las peticiones de manera eficiente, garantizando una comunicación cifrada mediante [HTTPS](#) con la integración de Let's Encrypt y Certbot.

Durante las pruebas, se ha demostrado que el servidor es estable, eficaz y capaz de atender peticiones concurrentemente. Su operatividad es posible de mantener gracias a su despliegue en un contenedor Docker dentro de un [VPS](#), facilitando su portabilidad, mantenimiento y disponibilidad continua.

Dentro de sus limitaciones, la principal es la cantidad de procesos concurrentes que puede manejar en el mismo momento. En particular, al recibir un número elevado de peticiones provoca un colapso en este. En la prueba de rendimiento (apartado [4.3.4](#)), se observa como el tiempo medio de análisis aumenta conforme se aumenta el número de peticiones. A partir de 30 solicitudes, el servidor dejó de responder, sin ser capaz de responder correctamente a las peticiones.

Además, dado que todo el análisis se realiza en el servidor, el correcto funcionamiento del sistema depende de dos factores externos: una conexión a Internet estable y la disponibilidad continua del propio servidor. Esto puede afectar la experiencia del usuario en entornos con conectividad limitada.

### 5.3.2. Líneas futuras de desarrollo

Aunque el servidor demuestra un funcionamiento apto dentro del proyecto, existen oportunidades para la mejora y escalabilidad futura.

El principal campo a mejorar es la optimización del procesamiento concurrente. Para ello, se puede incorporar un sistema de cola de tareas, como Celery (Ask Solem 2025), que permitiría gestionar las tareas de una manera más eficaz. Esta solución reduciría los cuellos de botella mencionados en las pruebas de rendimiento.

Otra línea posible para mejorar es la de segmentar el análisis en varios micro-servicios. Actualmente, todas las funcionalidades de comunicación están integradas en un solo código, que, si bien es funcional, dificulta la escalabilidad y mantenimiento a largo plazo.

También, en cuanto a gestión de almacenamiento, se puede implementar la funcionalidad de eliminar periódicamente los archivos procesados, con el fin de liberar espacio y evitar acumulación de datos innecesarios que puedan afectar negativamente al rendimiento del servidor.

Estas mejoras supondrían abrir el camino a un sistema más robusto, escalable y mejor preparado para dar soporte a un mayor volumen de usuarios.

## 6.

# Conclusiones

---

El presente trabajo ha demostrado la viabilidad de un sistema para la automatización de la clasificación del tipo de pisada. La aplicación móvil RunaAlyze, junto con el servidor, permite procesar videos de corredores sobre una cinta, entregando los resultados del análisis usando herramientas de visión por computadora.

Los resultados obtenidos presentan que en condiciones de grabación favorables, la precisión es aceptable, con errores mínimos de entre **7 y 9 %**. Sin embargo, factores como la calidad del video, la iluminación y la posición de la cámara tienen una alta influencia en el rendimiento. En situaciones muy desfavorables, pueden aumentar el error hasta un **26 %**.

En cuanto a la aplicación, cumple las funciones de facilitar al usuario la selección de un video a analizar y de la visualización de los resultados. Valorada por los usuarios de manera positiva, debido a su simplicidad, guía de ayuda y buen funcionamiento.

Sobre el servidor, se logró un sistema funcional y estable, capaz de procesar múltiples solicitudes. No obstante, las pruebas presentaron que hay un cuello de botella a partir de cierta cantidad de peticiones, que abre puertas a futuras mejoras.

Finalmente, entre los trabajos futuros, se propone trabajar en las líneas de mejora en la precisión del análisis, en la escalabilidad de los servidores y de ampliar las funcionalidades en la aplicación. Estas mejoras permitirían un desarrollo de RunAlyze más robusta y útil para una comunidad más amplia de usuarios.

# Referencias

---

- AMD (2024). *AMD EPYC 9654 – 4th Generation Server Processor*. URL: <https://www.amd.com/es/products/processors/server/epyc/4th-generation-9004-and-8004-series/amd-epyc-9654.html>.
- Android Developers (s.f.). *Cómo comenzar a desarrollar para Android*. URL: <https://developer.android.com/get-started/overview?hl=es-419>.
- Ask Solem, contributors (2025). *Getting Started — Celery 5.5.3 documentation*. URL: <https://docs.celeryq.dev/en/stable/getting-started/introduction.html>.
- Carranza, Rim (2021). *Dolor en el arco interno del pie*. Efisio. URL: <https://www.efisio.es/sintomas/dolor-arco-interno-pie>.
- Charmant, Joan y contributors (2024). *Kinovea (2023.1.2)*. URL: <https://www.kinovea.org>.
- Cloud, OVH (2025). *Virtual Private Servers (VPS)*. OVH Cloud. URL: <https://www.ovhcloud.com/>.
- Corporation, Xiaomi (2019). *Redmi Note 8 Pro – Specifications*. URL: <https://www.mi.com/es/redmi-note-8-pro/>.
- Electronics, Samsung (2025). *Galaxy S25 – Specifications*. URL: <https://www.samsung.com/es/smartphones/galaxy-s25/>.
- Ferber, R., A. Hreljac y K. D. Kendall (2009). «Suspected mechanisms in the cause of overuse running injuries: A clinical review». En: *Sports Health* 1.3, págs. 242-246.
- Fong, D. T. et al. (2008). «Effect of medial arch-heel support in inserts on reducing ankle eversion: a biomechanics study». En: *Journal of Orthopaedic Surgery and Research* 3.1, págs. 1-7.
- Foundation, Electronic Frontier (n.d.). *Certbot: Free SSL/TLS Certificates*. Certbot. URL: <https://certbot.eff.org/>.
- Foundation, Wireshark (s.f.). *Wireshark – The world’s leading network protocol analyzer*. URL: <https://www.wireshark.org/>.
- Fujishita, Hiroaki et al. (2023). «Effects of Rearfoot Eversion on Foot Plantar Pressure and Spatiotemporal Gait Parameters in Adolescent Athletes». En: *Healthcare* 11.13, pág. 1842. DOI: [10.3390/healthcare11131842](https://doi.org/10.3390/healthcare11131842).
- Google, AI (2024). *Pose landmark detection guide — MediaPipe Pose Landmarker*. URL: [https://ai.google.dev/edge/mediapipe/solutions/vision/pose\\_landmarker](https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker).

Group, Internet Security Research (n.d.). *Let's Encrypt - Free SSL/TLS Certificates*. Let's Encrypt. URL: <https://letsencrypt.org/>.

Gunicorn (s.f.). *Green Unicorn - Python WSGI HTTP Server for UNIX*. URL: <https://gunicorn.org/>.

OpenCV (s.f.). *Open Computer Vision Library*. <https://opencv.org/>.

Pečar, A. et al. (2018). «Automatic detection of pronation type of runners using computer vision». En: *Computer Science*. URL: <https://ipsittransactions.org/journals/papers/tir/2019jan/p2.pdf>.

Princeton Medicine (2023). *Understanding pronation in running: A comprehensive guide*. URL: <https://www.princetonmedicine.com/blog/understanding-pronation-in-running-a-comprehensive-guide>.

Project, Pallets (s.f.). *Flask documentation (versión estable)*. <https://flask.palletsprojects.com/en/stable/>

Python Software Foundation (2024). *secrets — Generate secure random numbers for managing secrets*. URL: <https://docs.python.org/es/3.8/library/secrets.html>.

Raschka, S., J. Patterson y C. Nolet (2020). *Machine Learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence*. URL: <https://arxiv.org/abs/2002.04803>.

Redmond, A. C., Y. Z. Crane y H. B. Menz (2008). «Normative values for the Foot Posture Index». En: *Journal of Foot and Ankle Research* 1.1, págs. 1-9.

Richter, R. R., T. M. Austin y M. F. Reinking (2011). «Foot orthoses in lower limb overuse conditions: A systematic review and meta-analysis—Critical appraisal and commentary». En: *Journal of Athletic Training* 46.1, págs. 103-106. URL: <https://www.researchgate.net/publication/49736578>.

Rose, A., I. Birch y R. Kuisma (2011). «Effect of motion control running shoes compared with neutral shoes on tibial rotation during running». En: *Physiotherapy* 97.3, págs. 250-255. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0031940610001112>.

Runner's World (2023). *Runners with too much supination have a 77% greater odds of injury compared to runners with normal foot biomechanics*. [https://ketone.com/blogs/blog/training-supination-for-athletes?srsId=AfmB0oqq-4qyfq3aBFn0oST\\_-rrWv9NtJhHS0T-0N8aLTn5wi94ef0ff](https://ketone.com/blogs/blog/training-supination-for-athletes?srsId=AfmB0oqq-4qyfq3aBFn0oST_-rrWv9NtJhHS0T-0N8aLTn5wi94ef0ff).

Silva, É. Q. et al. (2020). «The association between rearfoot motion while barefoot and shod in different types of running shoes in recreational runners». En: *Journal of Sports Science & Medicine* 19.2, pág. 383. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC7196748/>.

- Ugalde Álvarez, R. y C. de la Cuadra Blanco (2009). *Fascitis plantar: Diagnóstico y tratamiento*. Federación Española de Medicina del Deporte. URL: [https://femedede.es/documentos/Fascitis\\_plantar\\_XXJJTrauma.pdf](https://femedede.es/documentos/Fascitis_plantar_XXJJTrauma.pdf).
- Xataka (2025). *Qualcomm Snapdragon 8 Elite Gen 5 – Características / ficha técnica*. URL: <https://www.xataka.com/componentes/qualcomm-snapdragon-8-elite-gen-5-caracteristicas-ficha-tecnica>.
- Zong, W., J. Yang y F. Baytar (2023). «Predicting Human Body Measurements Using MediaPipe Pose Auto Capture». En: *Proceedings of the 14th Conference and Exhibition on 3D Body Scanning and Processing Technologies (3DBODY.TECH 2023)*. <https://doi.org/10.15221/23.33>. DOI: [10.15221/23.33](https://doi.org/10.15221/23.33).

# Índice de términos

---

## Siglas

**AA** aprendizaje automático. [2](#), [3](#), [10](#), [56](#)

**ADR** ángulo del retropié. [4](#), [19–22](#), [36–38](#), [55](#), [56](#), [II](#), [III](#)

**AI** artificial intelligence. [4](#)

**CPU** Central Processing Unit. [19](#)

**FPI** Foot Posture Index. [4](#)

**HTTP** protocolo de transferencia de hipertexto. [33](#), [34](#), [50](#)

**HTTPS** protocolo de transferencia de hipertexto seguro. [14](#), [16](#), [31](#), [33](#), [50](#), [52](#), [53](#), [58](#)

**IA** inteligencia artificial. [2](#), [3](#), [9](#), [10](#)

**ML** machine learning. [4](#)

**SSL** Secure Sockets Layer. [31](#), [32](#)

**UFW** cortafuegos sin complicaciones. [33](#)

**VPS** protocolo de transferencia de hipertexto seguro. [16](#), [31](#), [32](#), [58](#)

**WGSII** interfaz de pasarela del servidor web. [33](#)

