

UniversidadeVigo

ESCUELA DE INGENIERÍA DE TELECOMUNICACIONES



Memoria resumen ejecutivo LPRO – GYMVision

Autores

Aaron Granja López
Paula González Abalde
Belén Gómez Caracoche
Javier Berjoyo Madera
Néstor Lomba Lomba
Álvaro Soto Barja

Tutores

Martín Llamas Nistal
Lucía Costas Pérez

Mayo 2022

1. Introducción

Que el deporte es algo bueno para nuestra salud es algo que a estas alturas sabemos todos, ¿quién no ha tenido alguna vez la inquietud de intentarlo? Sin embargo, existe mucha gente que no consigue conciliar el deporte con su día a día, desde problemas por falta de tiempo, hasta el propio desencanto con esta actividad, asociándola al sufrimiento. Por eso, desde GYMVision queremos proponer una solución remota a este problema, de forma que se pueda realizar ejercicio desde cualquier lugar de manera cómoda, privada y segura.

Con esto en mente, nos planteamos una serie de preguntas: ¿cómo podemos ayudar a los usuarios a hacer deporte habitualmente, disfrutando y manteniendo la motivación? ¿Y cómo asegurar que esta actividad no resulte perjudicial por llevar a cabo malas prácticas debido a la falta de supervisión? Buscando la respuesta a estas y más preguntas, y analizando los resultados de las encuestas realizadas a potenciales usuarios, han surgido las funcionalidades que caracterizan a nuestra aplicación, las cuales iremos desgranando a lo largo de este documento.

2. Estado del arte

En la actualidad existen numerosas aplicaciones orientadas al ámbito deportivo: desde aquellas que generan rutinas y calendarios alimenticios[1], hasta las que ofrecen clases en directo online[2]; incluso las hay más novedosas que incluyen funcionalidades como la corrección de la postura en tiempo real[3]. Sin embargo, todas ellas tienen un punto en común: no están pensadas para personas que no tienen el hábito de hacer ejercicio.

De ahí el enfoque que mencionamos anteriormente, pues las soluciones que existen son de carácter generalista, es decir, buscan llegar a un máximo número de personas[4]. Un diseño que ponga el foco en generar hábitos saludables y motivación en los usuarios resultaría una novedad y una vía interesante de explorar.

3. Objetivos y funcionalidades

Todas las funcionalidades desarrolladas se han implementado siguiendo un objetivo común: que la interfaz sea intuitiva y fácil de utilizar. Nuestro público objetivo es gente principiante o *amateur*, ya que queremos dar facilidades para que el deporte llegue a más personas; por tanto, no hay distinciones de edad, lo que hace que tengamos que tener muy en cuenta que no todos estos potenciales usuarios tienen un manejo excelente de las tecnologías.

A continuación se exponen dichas funcionalidades, así como los objetivos que perseguimos con cada una de ellas.

3.1. Gamificación y retos

Con el objetivo de hacer del deporte una actividad entretenida y motivar a los usuarios a superarse día a día, incorporamos a nuestra aplicación procesos de gamificación basados en el abordaje de retos y obtención de puntos que el usuario podría utilizar posteriormente.

En nuestro perfil de usuario existe un apartado de logros en el que podemos observar qué retos podemos abordar y en qué nivel de cada logro nos encontramos actualmente (existen 3 niveles por logro)[figura 1]. De esta lista podemos elegir intentar alguno de ellos, de forma que se establecería el límite de tiempo que tenemos para lograrlo y se empezaría a tener en cuenta el progreso. Los retos que actualmente estemos abordando podremos consultarlos en la ventana de retos de nuestro perfil, junto con dicho progreso y fecha límite[figura 1].

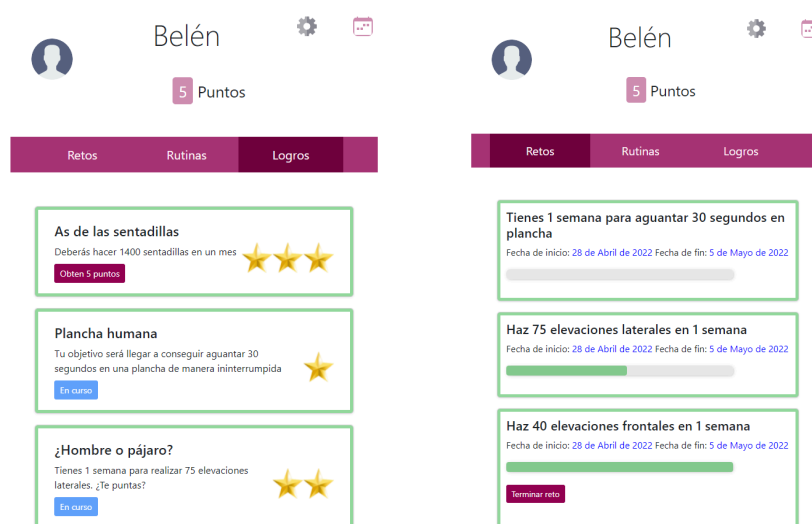


Figura 1: Pantallas de logros (izquierda) y retos (derecha)

3.2. Selección de rutinas y corrección de la postura

“¿Cómo sé si estoy haciendo bien este ejercicio?” Esta funcionalidad responde a esa pregunta. Hemos implementado un algoritmo que, obteniendo los puntos de tus articulaciones de la imagen de tu cámara, observa y corrige en tiempo real los errores que puedas estar cometiendo, disminuyendo así el riesgo de sufrir una lesión.

El usuario tendrá la posibilidad de hacer una de las rutinas estándar que definimos o, si lo prefiriese, personalizar su propia rutina con la biblioteca de ejercicios disponible en nuestra aplicación, eligiendo tanto estos como el número de repeticiones y series que desea hacer para cada uno. Una vez el usuario tiene hecha su elección, se le redirigirá a nuestro asistente de corrección donde se llevará cuenta de las repeticiones (que se sumarán al progreso de los retos, si los hubiera) y se le avisará e informará, como se adelantaba anteriormente, en caso de que cometa algún error [figura 2].

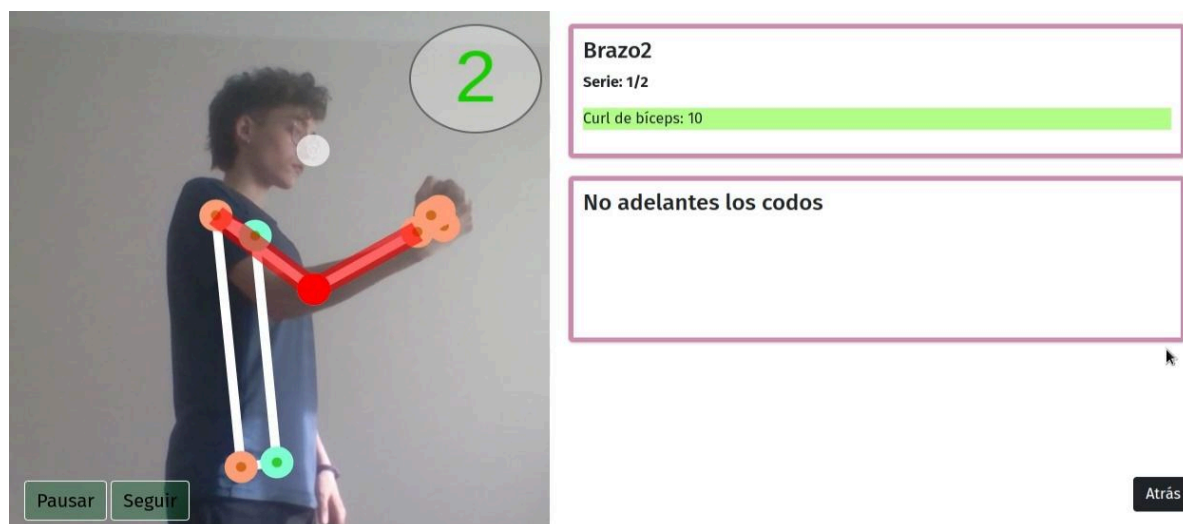


Figura 2: Pantalla del asistente de corrección

3.3. Clases en directo

Ofrecemos la posibilidad de registrar a un usuario como entrenador, quien tendría privilegios para crear salas en las que hacer una retransmisión en directo. Los usuarios pueden visualizar una lista de directos activos en el momento en el que se conectan y unirse al que prefieran.

Las salas son de un único uso, esto es, se eliminarán cuando la retransmisión finalice y todos los usuarios se desconecten. Dentro de ellas [figura 3] se ofrece, además, una herramienta de chat para que los usuarios puedan plantear ahí sus dudas de manera puntual, si así lo desean. Pretendemos así ofrecer una alternativa menos solitaria a los usuarios, de forma que podamos llevar las clases presenciales de un gimnasio hasta los hogares.

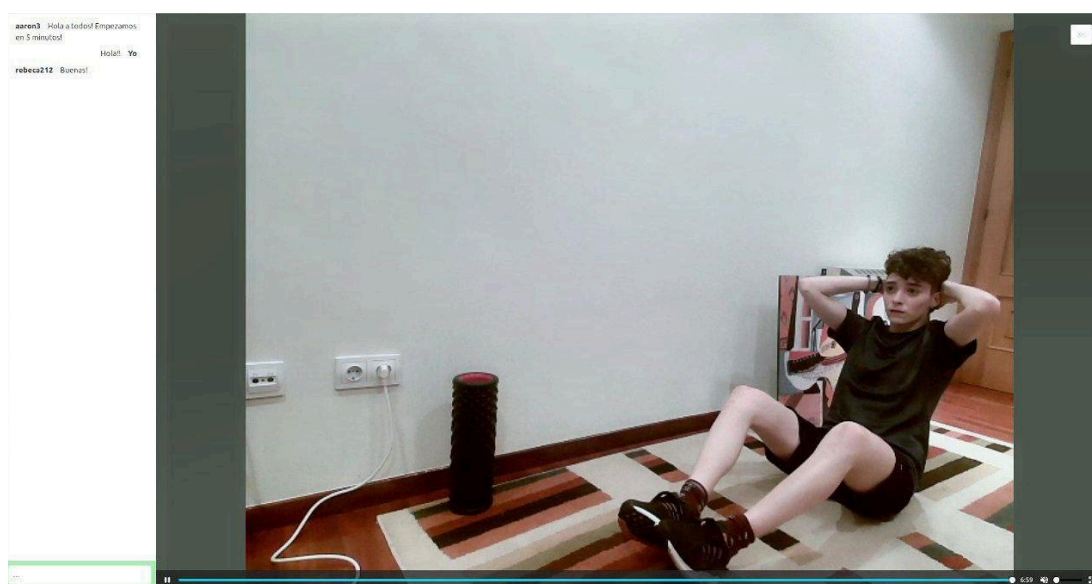


Figura 3: Pantalla de una clase en directo

3.4. Calendario de entrenamiento y alarmas

Cada usuario dispondrá de un calendario de entrenamientos en el que podrá establecer sus propios horarios para hacer ejercicio. Con esto pretendemos ayudar al usuario a marcarse unos objetivos, además de impulsarle a no perder el hábito enviando notificaciones a su correo media hora antes de las horas que haya establecido.

3.4. Login y registro de usuarios

Por la naturaleza de todas las funcionalidades ya descritas, ha sido necesario incluir un registro de los usuarios. Entre los datos asociados a cada uno de los usuarios se incluyen rutinas, progreso en retos y logros, y programación de los eventos del calendario.

4. Arquitectura y tecnologías empleadas

Una vez definidas las funcionalidades y objetivos de nuestra aplicación, damos paso a cómo se ha llevado a cabo su implementación. En la figura 4 podemos ver un esquema de la arquitectura del proyecto con todas las tecnologías implicadas.

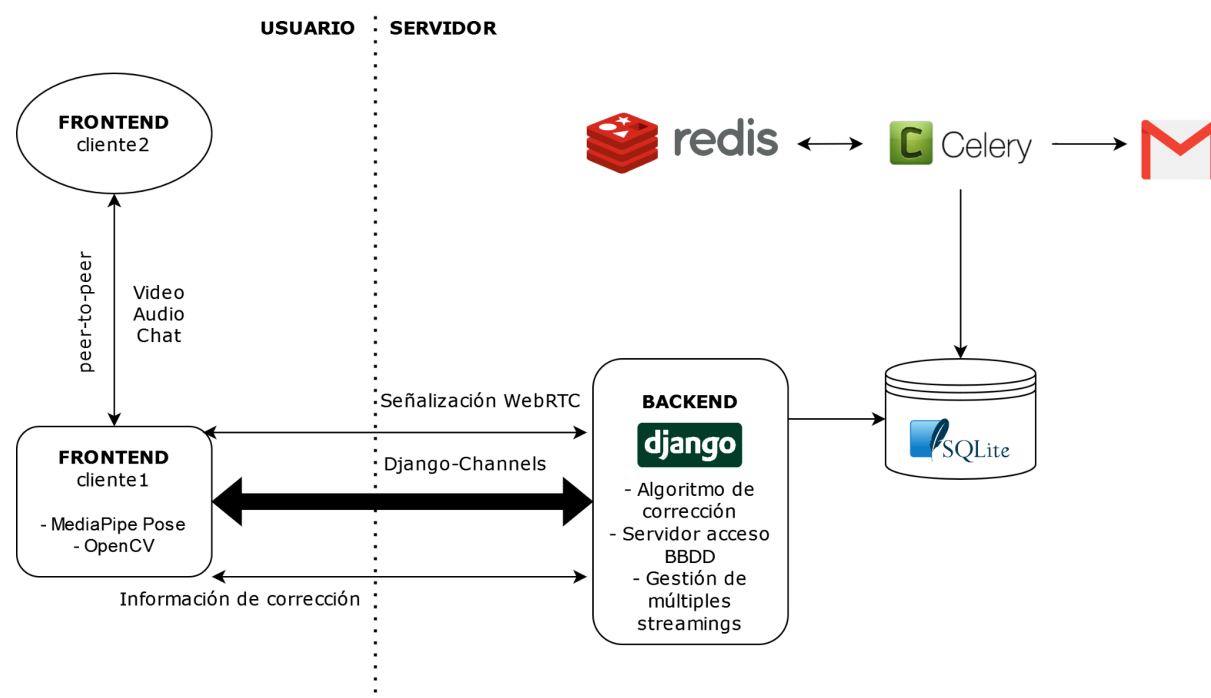


Figura 4: Arquitectura del sistema

Esta infraestructura es la resultante de satisfacer los requisitos de nuestra aplicación y del tiempo disponible para el desarrollo. A continuación detallamos el por qué de utilizar cada una de estas tecnologías y su función dentro del conjunto del proyecto.

4.1. MediaPipe Pose y OpenCV

MediaPipe[5] es un *framework* que contiene modelos de *machine learning* para la detección de distintas partes del cuerpo; en concreto, a nosotros nos interesa la librería Pose de entre todas las que ofrece, pues detecta 33 puntos de todo el cuerpo en tiempo real, puntos que de aquí en adelante denominaremos *landmarks*.

El flujo de datos es el siguiente: en el *frontend* se detectan los *landmarks* utilizando la librería MediaPipe Pose sobre el flujo de vídeo obtenido de la cámara del cliente con OpenCV; estos datos viajan al servidor mediante un socket implementado con Django-channels hasta el backend; una vez allí se procesan y se envía el resultado de vuelta al usuario.

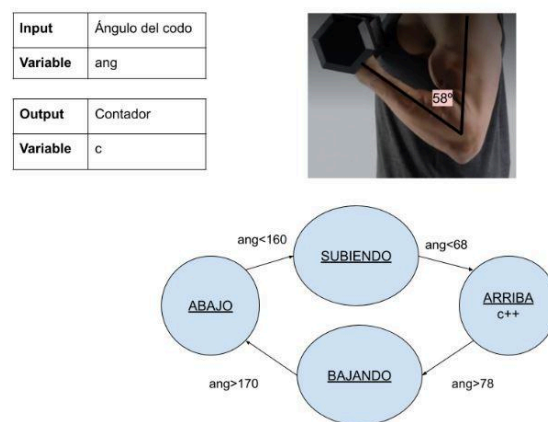


Figura 5: Esquema máquina de estados para el curl de bíceps

Ahora bien, ¿en qué consiste este procesado? El primer paso es obtener ángulos entre ternas de puntos, ternas que se elegirán dependiendo del ejercicio que el usuario esté realizando (información que se proporciona previamente desde el cliente). Una vez tenemos esta información, la podremos utilizar para dos cosas: la primera es hacer un conteo de repeticiones, llevado a cabo mediante una máquina de estados de tipo Mealy [figura 5]; la segunda, comprobar si el usuario está cometiendo errores comparando dichos ángulos con valores de referencia (permitiendo cierto margen). Cuando tenemos todo el procesamiento completo, estamos en situación de mandar dicha información al cliente mediante el mismo socket y mostrarla, como ya podíamos observar en la figura 2.

4.2. Django

Con la decisión tomada de qué librería utilizar para el algoritmo de corrección, nos encontramos que con Python ya había mucho camino recorrido al respecto, con lo que, teniendo en cuenta el tiempo de desarrollo del que disponíamos, su utilización para nuestro proyecto resultó ser lo más adecuado. Esto nos llevó a escoger Django[6], un *framework* de desarrollo web escrito en este lenguaje.

Además, ofrece muchas facilidades para el manejo y comunicación con la base de datos, lo cual, teniendo en cuenta los requisitos que teníamos (apartado 3.4), también era de nuestro interés.

4.3. Redis y Celery

La funcionalidad del calendario (apartado 3.4) se ha implementado utilizando Redis[9] y Celery[10], dos tecnologías fuera del *framework* Django. Por una parte, Celery, que es un paquete de gestión de colas de tareas, se encarga de programar una tarea cada minuto que comprueba si existe algún evento en el registro del calendario de cada usuario en los próximos 30 minutos. Por otra parte tenemos Redis, un almacén de datos de clave/valor en memoria, el cual utilizamos como *broker* para encolar las tareas pendientes. Por último, Celery también es el encargado de llevar a cabo las tareas encoladas en Redis, enviando un correo al usuario en caso de que se cumplan las condiciones previas.

5. Pruebas y resultados

Para validar el correcto funcionamiento de nuestro prototipo se han llevado a cabo una serie de pruebas para cada una de las funcionalidades. Por ejemplo, podemos comprobar que se pueden afrontar logros, que se establecen correctamente como retos (con fechas tanto de inicio como de fin acordes al instante en el que se inició) y que podemos tanto fallar como tener éxito en ellos, comprobando que los puntos se suman y se sube de nivel en caso de éxito, o que esto no sucede en caso de fracaso. En este proceso también podemos observar que el conteo de repeticiones funciona acorde a la actividad del usuario, así como la corrección de los errores que pueda estar cometiendo. Además, podemos probar a hacer dos salas de directos y verificar que los datos compartidos (tanto chat como vídeo y audio) en cada una de ellas no interfiere con la otra. Cabe destacar que estas pruebas se realizaron accediendo con varios usuarios al mismo tiempo, comprobando que la información de unos no entra en conflicto con la de otros y que efectivamente cada usuario accede únicamente a sus datos.

6. Posibilidades comerciales del proyecto

Desde GYMVision planteamos una posible comercialización del proyecto ofreciendo alguna de las funcionalidades exclusivamente en lo que sería una versión de pago. Consideramos incluir en dicha versión las que implican tener a profesionales al otro lado de la pantalla, de forma que estos se llevasen un porcentaje de ingresos proporcional al número de usuarios que utilizasen sus servicios. En el estado actual del proyecto, la única funcionalidad que cumple esta definición es la de asistir a clases en directo, pero queda en vías de estudio el incluir la posibilidad de asistir a citas con ellos o que estos pudiesen publicar sus propias rutinas y recomendaciones.

Desde hace 3 años, con el comienzo de la pandemia, el número de suscripciones en los gimnasios ha descendido preocupantemente[11]. Aunque actualmente el sector ya se está recuperando, esto tuvo como consecuencia evidente un aumento en el número de servicios remotos que se ofrecían[12]. De esta manera deducimos que existe un interés en seguir ofreciendo estos servicios y por lo tanto nuestra aplicación podría encajar en este ámbito.

7. Posibles líneas futuras

Se han explorado numerosas posibilidades con respecto a qué funcionalidades ofrecer desde nuestra aplicación, como, por ejemplo, incluir paneles informativos acerca de los beneficios de aquellos ejercicios que realice el usuario o consejos sobre hábitos saludables. Sin embargo, al tratarse de algo que se sale de nuestro campo de conocimiento, son ideas que han quedado fuera del planteamiento desde un comienzo (dentro del marco de esta asignatura).

Existen, no obstante, un par de caminos que comenzamos a explorar y que explicaremos a continuación.

7.1. IA predictiva

Planteamos el desarrollo de un algoritmo que, para cada usuario, observe los errores que comete, de manera que pueda encontrar patrones en ellos a lo largo de las repeticiones de cada ejercicio y le avise antes de cometer sus errores más comunes. Una primera aproximación sería desarrollarlo con una red neuronal, en concreto un SOM, donde la longitud de las neuronas representaría el número de repeticiones que se tendrían en cuenta para estudiar los patrones.

En el estado actual en el que se encuentra el proyecto, el algoritmo de corrección (que es del que dependería) está preparado para su integración y se ha diseñado y desarrollado teniendo en cuenta esta posible dirección, por lo que el trabajo restante sería la propia implementación.

7.2. Algoritmo k-NN

Además, hemos explorado la vía de cambiar el algoritmo de seguimiento de ejercicios, actualmente implementado mediante una máquina de estados [figura 5], por un algoritmo que no solo llevase dicho seguimiento, sino que también detectase automáticamente el ejercicio que el usuario estuviese realizando.

Planteamos implementar esto mediante el algoritmo de *machine learning* k-NN. Este permite clasificar en clases las muestras de un conjunto de datos por proximidad a las k muestras más cercanas a él. Este planteamiento encaja perfectamente con lo que queremos conseguir, ya que tenemos un conjunto de posturas que queremos clasificar, las cuales se pueden agrupar por su parecido con las de entrenamiento, que son conocidas.

El estado de implementación de esta vía es casi completo, ya que se han conseguido obtener buenos resultados en cuanto a la clasificación de distintos ejercicios utilizando la app demo de MediaPipe como soporte. El camino por recorrer es, por tanto, la adaptación e integración de este desarrollo con el resto de tecnologías y funcionalidades de nuestro proyecto.

8. Referencias

- [1] [Ictiva](#)
- [2] [8fit](#)
- [3] [Fittonic](#)
- [4] [Las “apps” de ejercicio pueden ser una herramienta educativa para adolescentes - Faro de Vigo](#)
- [5] [MediaPipe](#)
- [6] [Django](#)
- [7] [Django Channels](#)
- [8] [django-pwa · PyPI](#)
- [9] [Redis](#)
- [10] [Celery](#)
- [11] [Fitness español: tasa de abandono del 10% y pérdida del 58,9% del negocio en el año de la Covid](#)
- [12] [Las tendencias del mercado del fitness español en 2022 - CMD Sport](#)