



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

TELECOM ESCUELA  
TÉCNICA VLC SUPERIOR  
DE UPV INGENIEROS  
DE TELECOMUNICACIÓN

# Realidad aumentada aplicada a la escalada

**Autor: Pablo Aguilar García**

**Tutor: Antonio Albiol Colomer**

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2016-2017

Valencia, 8 de Septiembre de 2017

## **Resumen**

El objetivo de este proyecto es la creación de una aplicación de realidad aumentada aplicada a la escalada para permitir a cualquier escalador sin conocimiento técnicos al respecto, instalar y configurar de un modo sencillo la aplicación en su lugar de escalada preferido una aplicación que le permita interactuar de un modo diferente con la superficie de escalada, desarrollando habilidades diferentes de un modo didáctico y entretenido.

Gracias al tratamiento digital de imágenes el usuario no necesitará mas que su propio cuerpo para navegar entre los menús de la aplicación y realizar los entrenamientos seleccionados. Tan solo necesitará disponer de un sistema operativo Linux, un dispositivo de captura de imágenes en tiempo real y un sistema de proyección de imágenes.

## **Resum**

L'objectiu d'este projecte és la creació d'una aplicació de realitat augmentada aplicada a l'escalada per a permetre a qualsevol escalador sense coneixements tècnics al respecte instal·lar i configurar d'una manera facil l'aplicació en el seu lloc d'escalada preferit una aplicació que li permeta interactuar d'una manera diferent amb la superfície d'escalada, desenvolupant habilitats diferents d'una manera didàctica i entretinguda.

Gràcies al tractament digital d'imatges l'usuari no necessitarà mes que el seu propi cos per a navegar entre els menús de l'aplicació i realitzar els entrenaments seleccionats. Tan sols necessitarà disposar d'un sistema operatiu Linux, un dispositiu de captura d'imatges en temps real i un sistema de projecció d'imatges.

## **Abstract**

The objective of this project is to create an application of augmented reality applied to climbing, which allows anybody without technical knowledge on IT to install and configure in a simple way the application in his favorite climbing wall. An application that allows one to interact with the climbing surface in a different way, developing different skills in a didactic and entertaining way.

Thanks to the digital images processing the user will only need his own body to navigate among menus of the application and complete different trainings. He will only need to have a device with a Linux based operating system, a real-time image capture device and an image projector.

# Índice de contenido

1	Introducción.....	1
2	Estado del arte.....	2
3	Metodología.....	3
3.1	Gestión del proyecto.....	3
3.2	Diagrama temporal.....	4
4	Desarrollo.....	6
4.1	Análisis de requisitos.....	6
4.2	Solución propuesta.....	7
4.2.1	Herramientas.....	10
4.2.2	Arquitectura de software.....	11
4.3	Implementación funcional.....	13
4.4	Implementación técnica.....	21
5	Resultados.....	30
6	Conclusiones y propuestas de futuro.....	38
7	Bibliografía.....	39
8	Anexos.....	40

# 1 Introducción

La motivación de este proyecto reside en la creación de una aplicación que ayude al aprendizaje y fomente la practica de la escalada mediante un interface interactivo de realidad aumentada, integrando tecnologías de tratamiento digital de imagen para la detección y análisis del movimiento del escalador.

Hasta el momento las aplicaciones de realidad aumentada aplicada a la escalada son prácticamente inexistentes y de difícil acceso para el usuario. Mediante esta aplicación se permitirá con un equipamiento de bajo coste la integración de un sistema de tratamiento digital de la imagen en un entorno deportivo de una manera sencilla y mediante la utilización de software libre.

Gracias a la sencillez de instalación y mantenimiento cualquier usuario podrá desarrollar sus propios entrenamientos personalizados para una actividad especifica a cada usuario, optimizando así el entrenamiento de manera personal y rentabilizando el tiempo invertido en cada sesión de entrenamiento.

Por otro lado la aplicación cuenta con juegos interactivos que permitirán tanto a niños como a adultos disfrutar del deporte de la escalada de un modo fuera de lo común con lo que se desarrollaran aptitudes no entrenadas mediante un entrenamiento estándar.

## 2 Estado del arte

En la actualidad existen pocas aplicaciones diseñadas con fines similares, como es la aplicación diseñada por la compañía Augmented climbing wall<sup>1</sup>, ofreciendo una solución cerrada sin posibilidad de adaptación a las necesidades de cada rocodromo. Este tipo de soluciones cerradas necesitan su propio equipo de proyección, grabación y una superficie para la escalada con unas características determinadas.

La solución propuesta en este proyecto permite la utilización del hardware que usuario prefiera, tanto dispositivo de grabación como de proyección. Esta solución esta diseñada con el principio de trabajo sobre cualquier superficie, por lo que su aplicación se puede realizar casi sobre cualquier superficie mediante una sencilla y rápida calibración realizada de manera automática por el software de la aplicación.

Con esta propuesta se consigue la posibilidad de adaptar la aplicación a cada uno de los usuarios permitiendo de manera sencilla y económica la integración de la tecnología de realidad aumentada en el ámbito de la escalada, sin necesidad de un conocimiento técnico para su instalación y configuración.

---

1 [Augmented Climbing Wall](#)

## 3 Metodología

El desarrollo de este proyecto se ha gestionado mediante la metodología del Diagrama de Gantt, desde hace décadas el método mas utilizado en la gestión de proyectos, debido a su sencillez de configuración y visión global de la situación real del proyecto con respecto a la definitiva.

### 3.1 Gestión del proyecto

El primer paso para la correcta gestión de un proyecto de esta índole es elaborar una lista de todas las actividades que requiere el proyecto para su correcta implantación. Tras analizar los componentes mínimos necesarios de la aplicación deseada, se obtuvieron las siguientes funcionalidades principales de las cuales se ramificarán otras más específicas:

- ◆ **Propuesta**
  - Consensuar objetivos y necesidades del proyecto.
  - Investigación de viabilidad.
  - Memoria.
- ◆ **Formación**
  - Lenguaje de programación.
  - Librerías de tratamiento digital de imagen<sup>2</sup>.
- ◆ **Desarrollo de la Aplicación**
  - Creación del entorno.
  - Diseño gráfico de la aplicación.
  - Integración de herramientas de análisis y tratamiento digital de imagen.
  - Integración de sistema de proyección.
  - Integración de sistema de interactividad con el usuario.
  - Pruebas de campo.
- ◆ **Seguimiento.**
  - Reuniones periódicas con Tutores.
  - Presentaciones periódicas.

---

2 [Opencv](#)

## 3.2 Diagrama temporal

Si nos ceñimos a un tiempo realista para la realización del proyecto, teniendo en cuenta el tiempo utilizado para la investigación y estudio de viabilidad del mismo, cada tarea debe estar perfectamente delimitada por un periodo razonable para su realización e intentar en la medida de lo posible no excederse en los plazos de finalización de cada fase.

Para asegurar el correcto cumplimiento de los tiempos establecidos es necesario diseñar un diagrama temporal<sup>3</sup> para así determinar en cada momento si el desarrollo del proyecto se encuentra dentro o fuera de plazo. A continuación se explicarán las diferentes fases del proyecto así como su finalización fuera de plazo afectaría al resto de ellas:

- ◆ **Fase 0 - Investigación y viabilidad (mes 1)**
  - Investigar viabilidad.
  - Consensuar objetivos y necesidades.
- ◆ **Fase 1 – Formación (mes 2 )**
  - Lenguajes de programación necesarios.
  - Librerías de tratamiento digital de imagen.
- ◆ **Fase 2 – Desarrollo Entorno (mes 3)**
  - Entorno aplicaciones.
  - Integración de librerías esenciales.
- ◆ **Fase 3 – Tratamiento y análisis digital de la imagen (meses 4-6 )**
  - Extracción de información relevante.
  - Proyección sobre el muro.
  - Exportar información a ficheros.
- ◆ **Fase 4 – Creación de un entorno interactivo con el usuario (meses 7 y 8)**
  - Controles de la aplicación mediante realidad aumentada.
  - Creación de entorno configurable por el usuario.
- ◆ **Fase 5 – Finalización (meses 9 y 10)**
  - Perfeccionamiento de la aplicación.
  - Desarrollo de memoria.

---

3 Diagrama Temporal Inicial incluido en el Anexo

ID	Proceso	Tiempo estimado ( días )	Inicio	Fin	Dependencias
<b>0</b>	<b>Fase 0</b>	<b>30</b>	<b>0 / 300</b>	<b>30 / 300</b>	-
0.1	Consensuar objetivos y necesidades	25	0 / 300	25 / 300	-
0.2	Investigar viabilidad	5	25 / 300	30 / 300	0.1
<b>1</b>	<b>Fase 1</b>	<b>30</b>	<b>30 / 300</b>	<b>60 / 300</b>	<b>0</b>
1.1	Librerías de tratamiento de imagen	15	30 / 300	45 / 300	-
1.2	Lenguajes de programación necesarios	15	45 / 300	60 / 300	0
<b>2</b>	<b>Fase 2</b>	<b>30</b>	<b>60 / 300</b>	<b>90 / 300</b>	<b>1</b>
2.1	Entorno aplicaciones	20	60 / 300	80 / 300	1
2.2	Integración de librerías esenciales	10	80 / 300	90 / 300	1, 2.1
<b>3</b>	<b>Fase 3</b>	<b>90</b>	<b>90 / 300</b>	<b>180 / 300</b>	<b>2</b>
3.1	Extracción de información relevante	30	90 / 300	120 / 300	2
3.2	Proyección sobre el muro	30	120 / 300	150 / 300	2 , 3.1
3.3	Exportar información a ficheros	30	150 / 300	180 / 300	2, 3.1
<b>4</b>	<b>Fase 4</b>	<b>60</b>	<b>180 / 300</b>	<b>240 / 300</b>	<b>2</b>
4.1	Controles de la aplicación mediante realidad aumentada	30	180 / 300	210 / 300	2, 3
4.2	Creación de entorno configurable por el usuario	30	210 / 300	240 / 300	2, 3
<b>5</b>	<b>Fase 5</b>	<b>60</b>	<b>240 / 300</b>	<b>300 / 300</b>	<b>3, 4</b>
5.1	Perfeccionamiento de la aplicaciones	40	240 / 300	280 / 300	3, 4
5.2	Desarrollo de memoria	20	280 / 300	300 / 300	3, 4, 5.1



## 4 Desarrollo

En este apartado se definirán los requisitos que debe cumplir la aplicación y las soluciones propuestas para solventar los problemas que estos plantean. Se definirá detalladamente el hardware utilizado, la arquitectura de estos y los resultados obtenidos tras la implementación de las soluciones.

### 4.1 Análisis de requisitos

- ◆ ***Fácil integración.***
  - Fácil instalación.
  - Fácil mantenimiento.
  - Fácil configuración.
- ◆ ***Hardware externo.***
  - Posibilidad de utilización de diferentes cámaras.
  - Posibilidad de utilización de diferentes proyectores .
- ◆ ***Adaptabilidad a diferentes superficies.***
  - Auto-calibración.
  - Detección automática de presas.
- ◆ ***Entrenamiento configurables.***
  - Configuración de rutas por el usuario.
  - Configuración de juegos por el usuario.
- ◆ ***Tratamiento digital de imagen.***
  - Funcionamiento en diferentes condiciones lumínicas.
  - Detección de movimiento.
- ◆ ***Interactividad con el usuario.***
  - Sistema de control de la aplicación mediante realidad aumentada
  - Sin necesidad de controladores electrónicos.
- ◆ ***Interface Gráfico.***
  - Implementación del análisis y ataque en un mismo interface.
  - Sencilla visualización de la información.

## 4.2 Solución propuesta

Para que la aplicación cuente con una fácil instalación, mantenimiento y configuración se propuso integrar toda la lógica en un único dispositivo fácil de conseguir en el mercado y con gran versatilidad para la integración del hardware necesario para desarrollar el proyecto.

Investigando las soluciones que nos propone el mercado se decidió utilizar como sistema central de la aplicación un dispositivo Raspberry Pi 3, dispositivo que cuenta con las siguientes características:

- Salida de video HDMI: para la comunicación con el sistema de proyección.
- Puertos usb: para la conexión del sistema de grabación.
- Comunicación Ethernet/WiFi: para el mantenimiento y configuración
- Sistema plug&play: tan solo es necesario conectar una fuente de alimentación para su funcionamiento.



*Figura 1. Dispositivo Raspberry Pi 3*

Este tipo de dispositivos SoC ( System on Chip) permiten la integración de múltiples elementos externos en un sistema operativo embebido con el objetivo de crear pequeñas centrales de procesamiento de información con un mantenimiento escaso, portabilidad, bajo consumo y precio competitivo.

Por contra este tipo de dispositivos de reducido tamaño y consumo tienen un poder computacional reducido por lo que la optimización del software es un punto a tener en cuenta en el desarrollo de la aplicación, puesto que a menudo los procesos de tratamiento digital de imagen suelen ser computacionalmente pesados y podría suponer un problema en aplicaciones de respuesta en tiempo real como es el caso de este proyecto.

En cuanto al hardware externo tan solo será necesario que el sistema de grabación cuente con conexión usb para la transmisión en directo de imágenes, y que el sistema de proyección cuente con salida de video compatible con HDMI. Teniendo estas dos condiciones el usuario final podrá seleccionar cualquier tipo de hardware a su libre elección, pero se recomienda que los dispositivos cuenten con las siguientes características para que los resultados sean óptimos:

-Sistema de grabación:

-Resolución mínima: 800x600

-Balance de blancos configurable ( posibilidad de desactivar auto-calibrado)

-Sistema de proyección:

-Resolución mínima: 800x600

-Potencia mínima: 3000 lúmenes (optimo  $\geq 5000$  lúmenes)

Para el desarrollo de la aplicación se opto por la utilización de hardware específicamente diseñado para el dispositivo Raspberry Pi como sistema de grabación, puesto que este permite una comunicación entre el sistema de procesado y el de grabación más rápida, lo cual repercutirá positivamente en la sensación de interactividad entre la aplicación y el escalador.



Figura 2. Camera Pi v2.1

Como se puede apreciar en las características técnicas de la cámara<sup>4</sup> cumple con las recomendaciones anteriores sobre resolución y transmisión de información en tiempo real, por lo que los resultados para el software diseñado serán los óptimos. Gracias al diseño específico para la utilización de la cámara junto con el dispositivo Raspberry Pi, este cuenta con una conexión integrada en la base aumentado la velocidad de transmisión considerablemente.

---

4 [Especificaciones Camera Pi v2.1](#)

En cuanto al proyector realizado se utilizó el modelo BenQ Ms506 (800x600) con una potencia de 3000 lúmenes. Cabe mencionar que a mayor potencia de proyección mayor será la sensación de interactividad, puesto que el escalador podrá percibir los resultados proyectados sobre el muro con mayor facilidad que si, debido a la baja potencia, el interface proyectado es tenue y difícilmente distinguible del fondo.

Debido a su reducido tamaño y peso el dispositivo se puede colocar en cualquier parte del rocodromo facilitando así la utilización sobre diferentes superficies y su posible recolocación. Para que su adaptabilidad sea alta se ha integrado en la aplicación un sistema de auto-calibrado para la detección de la posición con respecto al sistema de proyección para así modificar los patrones, permitiendo que la desalineación entre sistema de proyección y sistema de grabación no suponga un problema a la hora de recolocar los dispositivos.

Para facilitar la reconfiguración de la aplicación una vez reposicionados los dispositivos la aplicación incluye un algoritmo de detección automática de presas para que el usuario tan solo tenga que conectar el dispositivo a una fuente de alimentación y comenzar a utilizarlo sin necesidad de ningún conocimiento técnico sobre el software o algoritmos de tratamiento digital de imagen.

Una vez detectadas las rutas la aplicación permite al usuario la creación de rutas personalizadas a partir de las presas detectadas de manera automática. Las rutas creadas son almacenadas en una base de datos que permite al usuario reutilizarlas en un futuro si este así lo desea.

Hasta el momento tan solo se han descrito los procesos de calibración y configuración de la aplicación, unicamente concernientes al propietario del rocodromo encargado de colocar y configurar los dispositivos. Los procesos en los que se verán involucrados los usuarios finales no tienen nada que ver con calibración o tratamiento digital de imagen, estos unicamente se comunicarán con la aplicación mediante un interface interactivo proyectado sobre el muro de escalada, separando así completamente la parte técnica del usuario final, consiguiendo que la utilización de la aplicación no se limite unicamente a usuarios con un perfil técnico.

Para que la interacción con la aplicación sea lo mas sencilla e intuitiva posible la aplicación funciona de modo similar a un dispositivo táctil, en el cual mediante la pulsación de botones virtuales proyectados sobre la pared, el usuario podrá navegar entre menús y seleccionar el tipo de ruta o juego que desea realizar. Una vez dentro del juego diferentes botones aparecerán para que el usuario comience, detenga o resetee el juego a su antojo. De este modo una vez configurado el juego y almacenada dicha configuración en la base de datos, tan solo será necesario utilizar tu propio cuerpo para el manejo de la aplicación, sin necesidad de dispositivos electrónicos como teclados o mandos.

## 4.2.1 Herramientas

Una vez seleccionados los dispositivos es necesario determinar el entorno sobre el cual se comunicaran todos estos dispositivos. Sabiendo que el sistema operativo más utilizado para los dispositivos Raspberry Pi es Raspbian<sup>5</sup> se decidió utilizar el lenguaje de programación Python v2.7<sup>6</sup> compatible con la librería de tratamiento digital de imágenes OpenCV<sup>7</sup>.

En primer lugar cabe destacar que el Sistema Operativo utilizada (Raspbian) esta basado en debian (código abierto) y esta especialmente optimizado para Raspberry Pi. Este incluye los programas y módulos básicos para el correcto funcionamiento del dispositivo Raspberry Pi, así como todos sus periféricos. A pesar de que el sistema operativo Raspbian no esta afiliado a Raspberry Pi, ya que fue creado por un pequeño grupo de desarrolladores entusiastas del dispositivo Raspberry Pi, estos ofrecen un gran servicio de actualización y resolución de consultas técnicas para cualquier usuario.

Por otro lado el lenguaje de programación utilizado es Python 2.7, ampliamente utilizado junto con proyectos relacionados con dispositivos Raspberry Pi. Las características principales de Python es su fácil integración con módulos externos y su sencillez de sintaxis. Uno de los principales motivos por los que se decidió utilizar este lenguaje fue la posibilidad de importar fácilmente las librerías de OpenCV, utilizadas en multitud de proyectos de tratamiento digital de imagen.

En ultimo lugar OpenCV es el modulo que vertebraba este proyecto. A pesar de que en una primera instancia fue desarrollado para el lenguaje de programación C++, debido a su amplia utilización por programadores de todo el mundo, el equipo de desarrollo de OpenCV migro todas sus librerías a otros lenguajes como Java y Python. Gracias a su migración a lenguajes como Python, junto a su compatibilidad con las librerías Numpy<sup>8</sup>, SciPy<sup>9</sup> y Matplotlib<sup>10</sup>, su popularidad creció sustancialmente en poco tiempo.

---

5 [Raspbian S.O.](#)

6 [Lenguaje Python 2.7](#)

7 [Librería OpenCV \(tratamiento digital de imagenes\)](#)

8 [Librería Numpy](#)

9 [Librería SciPy](#)

10 [Librería Matplotlib](#)

## 4.2.2 Arquitectura de software

Una vez definido el hardware a utilizar el siguiente paso sería definir las conexiones entre ellos, así como los protocolos de comunicación que se utilizaran, definiendo una arquitectura que muestre una visión global de la aplicación y sus periféricos.

Como ya se ha mencionado en apartados anteriores el elemento vertebrador de la aplicación es el dispositivo Raspberry Pi, con el cual gracias a la captura en tiempo real de imágenes se aplicaran diferentes procesos de tratamiento digital de imagen para determinar la posición y movimiento del escalador. Con esta información se determinan las imágenes que serán proyectadas sobre la superficie de escalada para crear en el usuario una sensación de realidad aumentada e interactividad con la aplicación únicamente usando su propio cuerpo, sin necesidad de hardware externo para el control de la aplicación.

Para que el dispositivo Raspberry Pi pueda recibir imágenes en tiempo real una de las soluciones es utilizar el puerto CSI-2 (Camera Serial Interface Type 2) mediante el cual se conectará el capturador de imágenes Camera Pi v2.1 antes mencionado. Este puerto fue especialmente diseñado para una rápida transmisión de video entre dispositivos externos de grabación y dispositivos Raspberry Pi, permitiendo un ancho de banda de 4 Gbps, mediante una conexión directa entre la cámara y procesador a través de la conexión CSI-2 en la PCB del dispositivo.

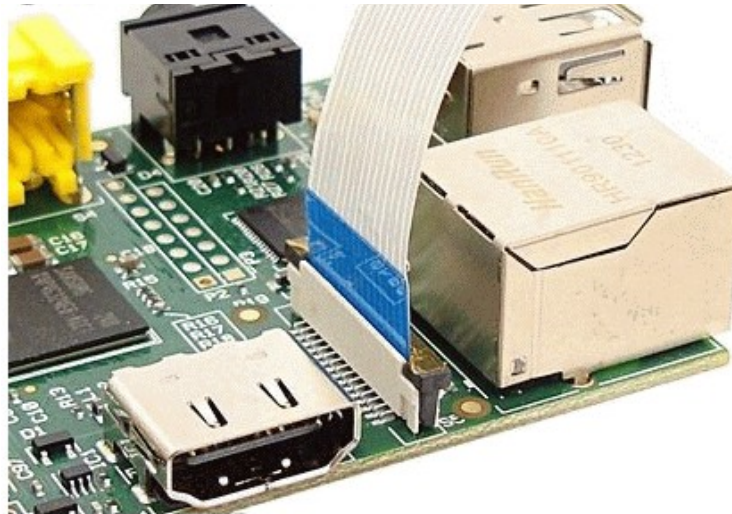
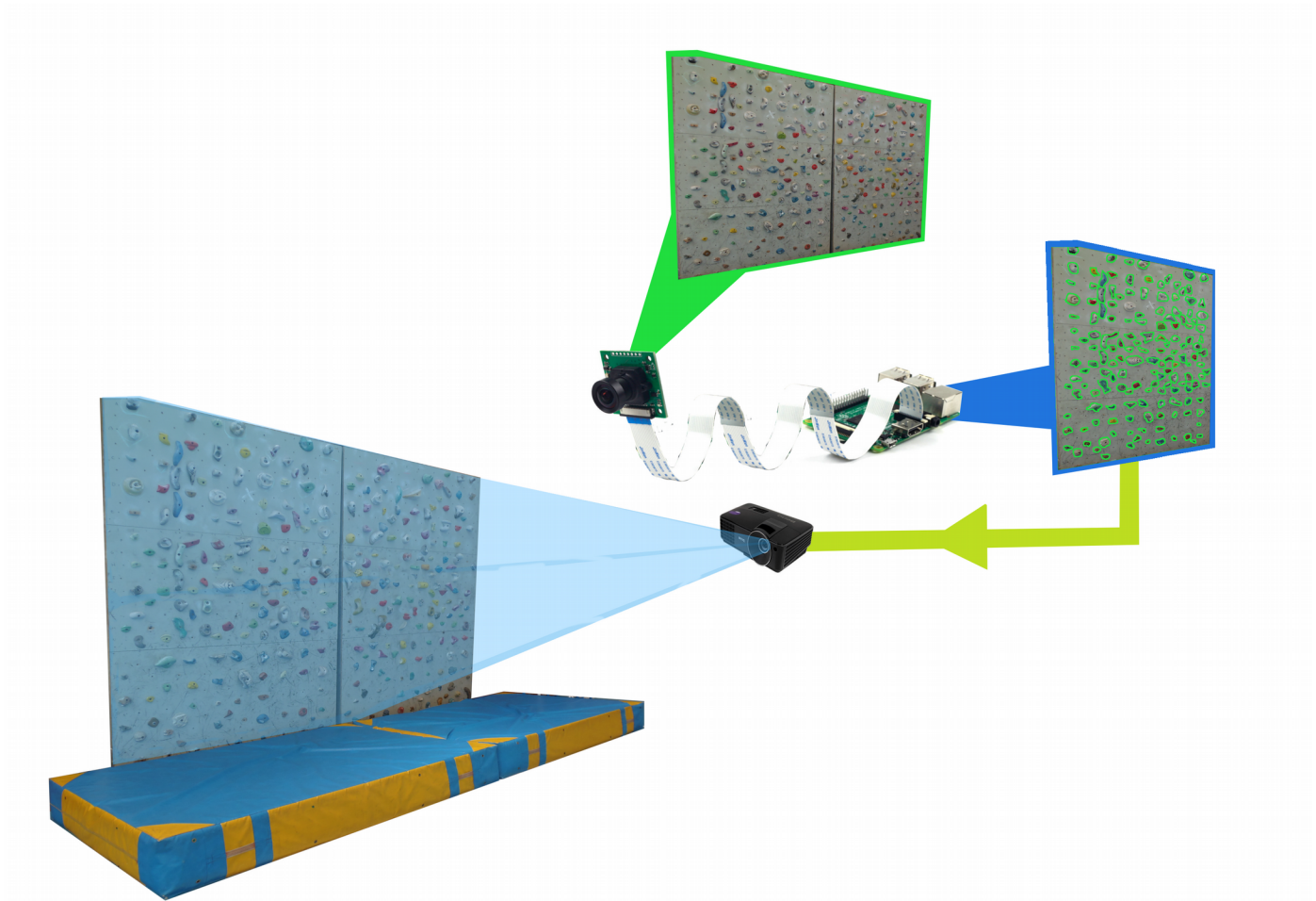


Figura 3. Puerto CSI-2 del dispositivo Raspberry Pi

Una vez la imagen es transmitida al dispositivo Raspberry Pi, este se encarga de aplicarle diversos procesos de tratamiento digital para determinar la posición y movimiento del escalador y determinar el estado en el que se encuentra dentro de la aplicación para responder interactivamente a demanda del usuario. Dicha respuesta es mostrada por al usuario mediante la proyección de imágenes sobre la superficie de escalada, permitiendo así al escalador estar informado en todo momento de la situación actual sin necesidad de detener su entrenamiento.

Para la transmisión de la información entre el dispositivo Raspberry Pi y el proyector de imágenes se utilizará el puerto de salida de video HDMI incorporado en el dispositivo Raspberry Pi, el cual transmitirá las imágenes necesarias para informar al escalador de su situación actual en el entrenamiento.



*Figura 4. Arquitectura de la aplicación.*

Por último para el almacenamiento de los diversos entrenamientos y la configuración preestablecida para la calibración de la aplicación se utilizarán ficheros de almacenamiento de información permitiendo almacenar resultados de entrenamientos anteriores, parámetros de sensibilidad de los dispositivos y configuración personalizada para cada usuario.

## 4.3 Implementación funcional

Para crear una sensación de interactividad entre el escalador y la aplicación se ha diseñado una serie de menús de configuración y selección de entrenamiento basados en el principio de funcionamiento de los dispositivos táctiles. Para ello, mediante el proyector, se muestra el menú actual sobre la superficie de escalada en el cual se encuentran los botones de control de la aplicación.

Estos botones son totalmente virtuales, pero mediante el procesado de imagen que se aplicará a las capturas obtenidas por la cámara se podrá determinar si el usuario a pulsado cada uno de los botones, creando la sensación de realidad aumentada al interactuar con los elementos virtuales proyectados sobre el muro.

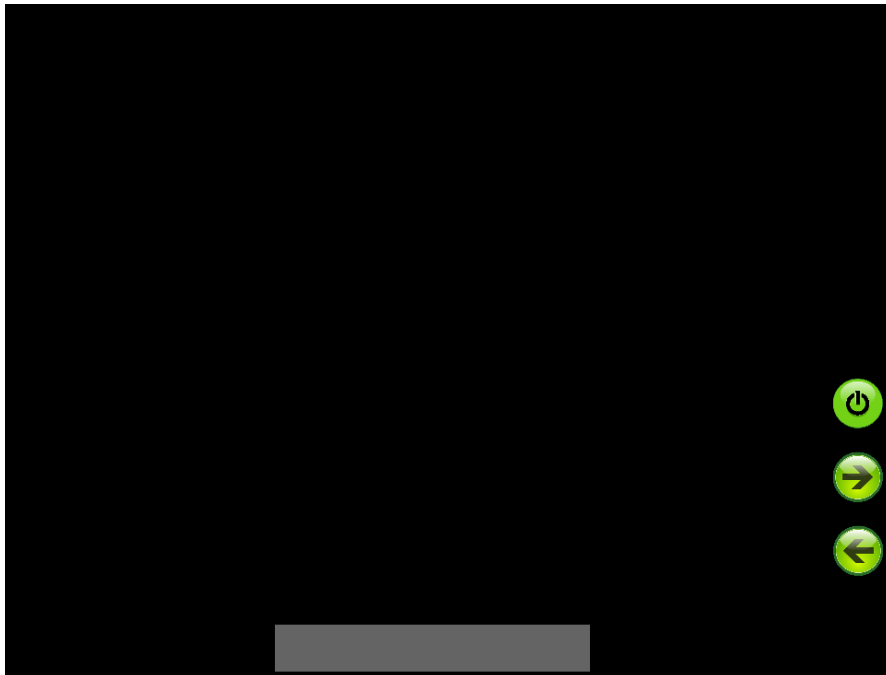


Figura 5. Menú principal de la aplicación

Dentro de la aplicación existen diferentes menús de configuración para cada uno de los tipos de entrenamiento creados, entre los cuales se encuentran los siguientes:

- Menú principal.
- Menú configuración.
- Menú específico para cada tipo de entrenamiento.
- Menú informativo al finalizar un entrenamiento.

Dentro del Menú principal se podrá navegar entre los diferentes tipos de entrenamiento, para que una vez encontrado el entrenamiento que se desea realizar se pueda acceder al menú específico del tipo seleccionado.



Este menú cuenta con un cuadro de texto inferior que mostrará un breve resumen del estado de navegación y tres botones para el manejo de la aplicación.

Dos de los tres botones que se encuentran en el menú principal se utilizan para la navegación (izquierda y derecha) entre menús, alternando entre tipos de entrenamientos. Una vez en el menú del tipo de juego deseado el tercer botón nos permitirá acceder al menú específico para ese tipo de entrenamiento concreto.

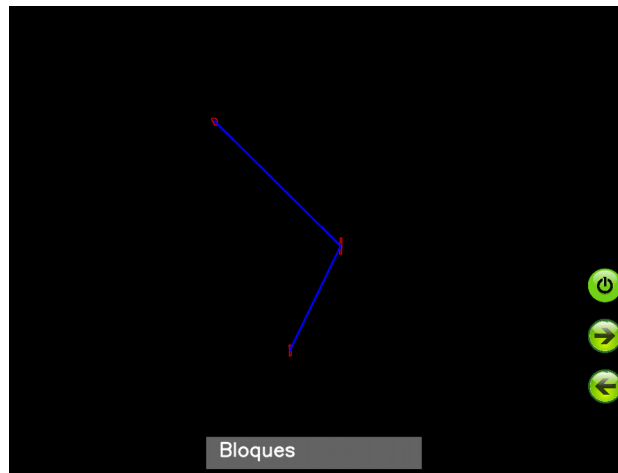


Figura 6. Menú selección de tipo de entrenamiento.

Para aumentar la sensación de interactividad con el usuario cada vez que un botón sea detectado como pulsado por el dispositivo de procesado, se reproducirá un sonido indicando que el botón ha sido pulsado, así como un momentáneo cambio de color del botón del color.



Figura 7. Cambio de color en botones.

Los tipos de entrenamientos implementados en la aplicación son:

- Entrenamiento basado en rutas/bloques.
- Entrenamiento basado en plataformas.
- Entrenamiento basado en esquiva.

El entrenamiento basado en rutas esta basado en el método mas común de entrenamiento entre los escaladores. Este método consiste en la determinación previa de una serie de presas por las que el escalador deberá pasar, habiendo una presa inicial y una final.

Una vez en el menú específico de entrenamiento basado en rutas, el usuario tendrá la posibilidad de navegar entre las diferentes rutas guardadas previamente en la base de datos. Para facilitar la visualización del entrenamiento que se va a seleccionar, al mismo tiempo que se navega entre las diferentes rutas guardas se proyectará la ruta paso a paso sobre la superficie de escalada, permitiendo así al escalador ver en tiempo real la ruta en la que se encuentra y conociendo los detalles de dicha ruta, como son numero de pasos, punto inicial y punto final.

En este tipo de entrenamiento un parámetro que determina la evolución o mejora es el tiempo invertido para llegar desde el punto inicial al punto final de la ruta, por lo que al iniciar el entrenamiento un cronometro comenzará a contar el tiempo al mismo tiempo que sera mostrado al usuario en la propia superficie de entrenamiento para que pueda comprobar el tiempo transcurrido en cualquier momento.

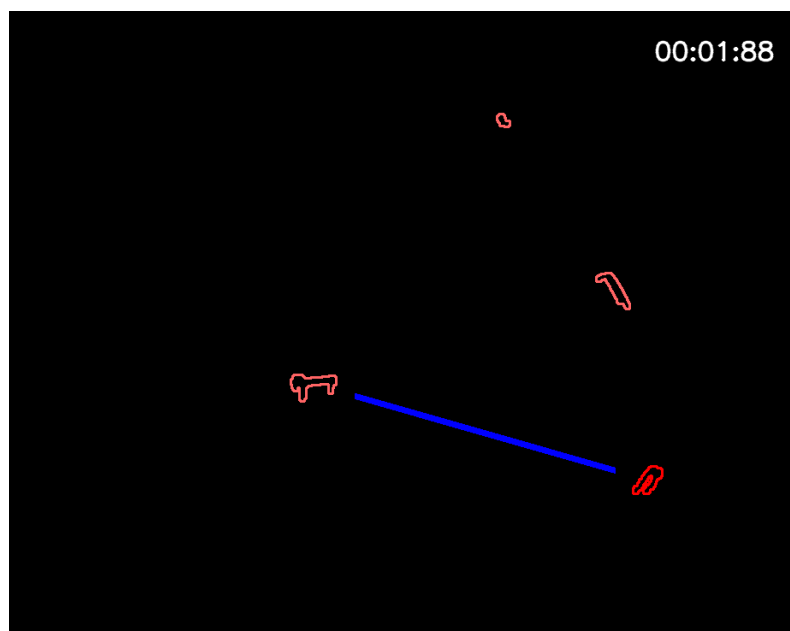


Figura 8. Entrenamiento basado en ruta iniciado.

Para facilitar la realización del entrenamiento al usuario se proyectará un contorno coloreado al rededor de cada una de las presas por las que el escalador debe pasar. A modo de ayuda también se proyectará una línea desde la presa actual detectada como agarrada hasta el siguiente punto de la ruta para indicar al escalador el siguiente paso a realizar.

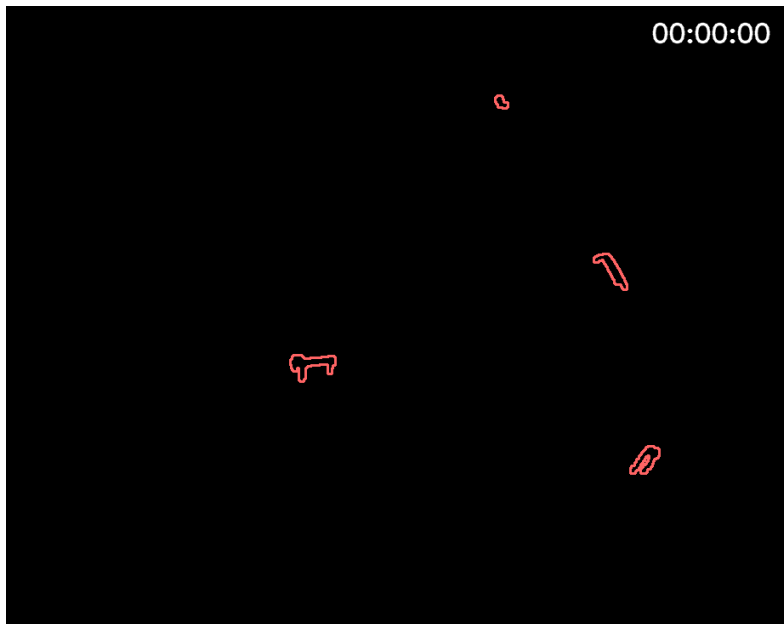


Figura 9. Contorno alrededor de las presas de la ruta.

Al tratarse de un entrenamiento interactivo de realidad aumentada cada vez que el dispositivo central de procesado detecte una modificación en la posición del escalador se re calcularán los parámetros a proyectar, para mantener al escalador informado de la situación del entrenamiento en tiempo real.

Para comenzar el entrenamiento el escalador deberá colocar una o ambas manos sobre la presa de inicio, momento en el cual se mostrará un círculo al rededor de esta presa mostrando la cuenta atrás de inicio del entrenamiento. Si este quitase la mano de la presa antes de terminar la cuenta atrás esta se reiniciaría y esperaría a que el escalador volviese a colocar la mano sobre la presa.

El objetivo de esta cuenta atrás es permitir al usuario colocarse en una postura cómoda para el inicio del entrenamiento y poder así comenzar el entrenamiento cuando el este realmente preparado.

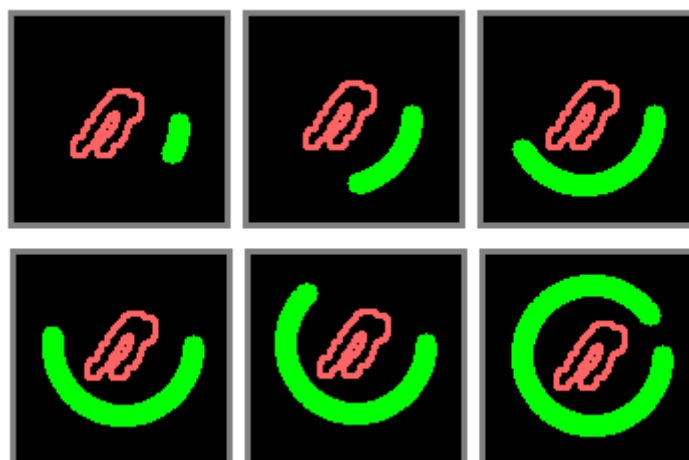


Figura 10. Circulo cuenta atrás inicio entrenamiento.

Si el escalador consigue llegar al punto final de la ruta el cronometro se detendrá y sobre el muro se mostrará un menú en informándole del tiempo invertido en el entrenamiento con una serie de botones para reiniciar el entrenamiento o volver al menú principal.



Figura 11. Menú informativo final.

El segundo tipo de entrenamiento ofrecido esta basado en plataformas, teniendo como principal objetivo llegar desde el punto inicial al punto final sin entrar en contacto con ninguna de las plataformas virtuales proyectadas sobre la superficie de escalada. Este tipo de entrenamiento permitirá al escalador desarrollar nuevas formas de afrontar los desafíos encontrados en la escalada debido al objetivo radicalmente diferente al ofrecido por la escalada deportiva.

Al igual que en el entrenamiento basado en rutas, en el menú específico de selección de plataforma también se podrá ver en tiempo real y sobre el muro de escalada la plataforma de entrenamiento que se va a seleccionar.

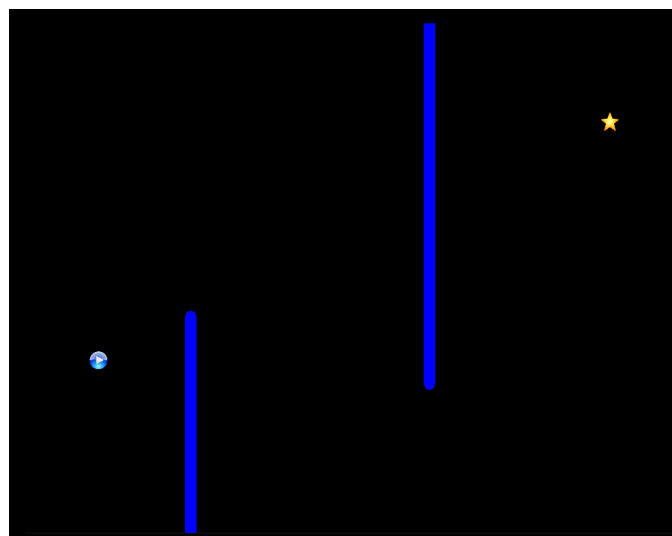
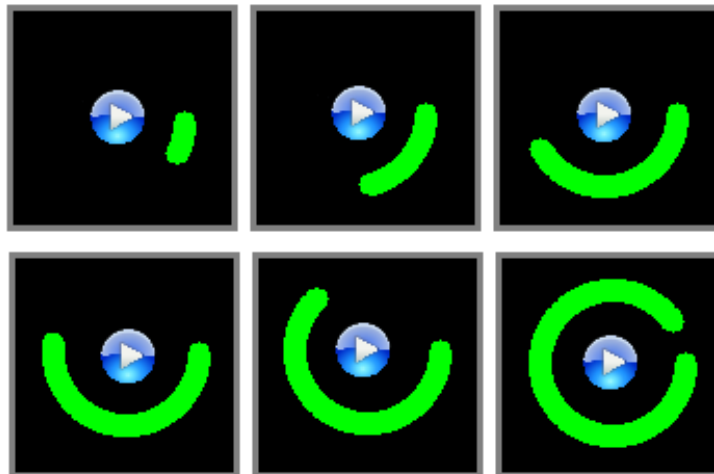


Figura 12. Entrenamiento basado en plataformas.

En la escalada deportiva convencional el objetivo es llegar desde un punto inicial hasta uno final sin importar el camino realizado. Con este tipo de entrenamiento la importancia de una buena realización radica en el proceso intermedio, siendo trivial el inicio y final el entrenamiento. Gracias al planteamiento de un entrenamiento basado en la adaptabilidad del escalador a las plataformas proyectadas, este podrá desarrollar nuevas habilidades de aplicabilidad transversal al resto de estilos de escalada.

El punto inicial del entrenamiento estará marcado por un botón de color azul con el símbolo de “play”, y el punto final con una estrella de color amarillo. Cuando el escalador coloque su mano sobre el botón de inicio se mostrará un círculo al rededor del botón mostrando la cuenta atrás de inicio del entrenamiento. Si este quitase la mano del botón antes de terminar la cuenta atrás esta se reiniciaría y esperaría a que el escalador volviese a colocar la mano sobre el botón.

El objetivo de esta cuenta atrás es permitir al usuario colocarse en una postura cómoda para el inicio del entrenamiento y poder así comenzar el entrenamiento cuando el este realmente preparado.



*Figura 13. Circulo cuenta atrás inicio de entrenamiento.*

Durante el desarrollo de este entrenamiento el sistema central de procesado comprobará continuamente la posición del escalador con respecto a las plataformas para determinar si se ha producido una colisión entre ambos. En caso de que se produjese una colisión, el juego se detendría y se mostraría sobre la superficie de escalada la silueta del escalador en contacto con la plataforma para indicar al usuario en que parte de la plataforma se ha producido la colisión y poder prevenir el mismo error en futuros entrenamientos.

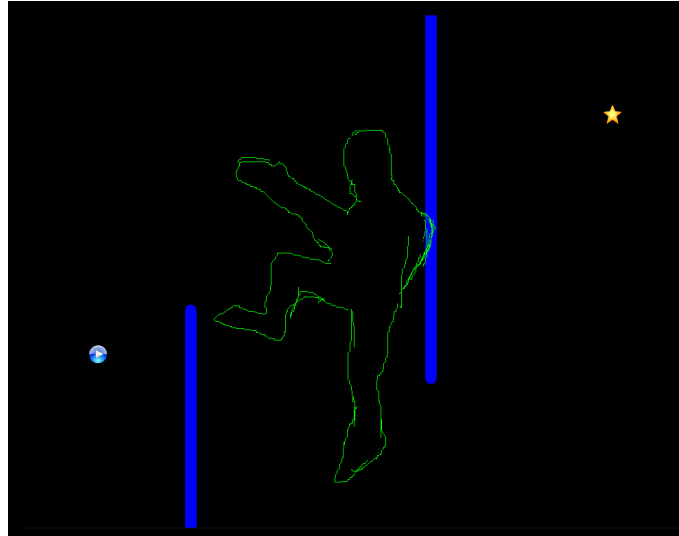


Figura 14. Silueta de colisión en entrenamiento basado en plataformas.

Tanto si el usuario termina con éxito el entrenamiento, como si se produce una colisión virtual entre las plataformas y el mismo, en una última instancia siempre se mostrará un menú de información de resultados del entrenamiento similar al mostrado en el entrenamiento basado en rutas. Permite a este reiniciar el entrenamiento o volver al menú principal para la selección de un juego diferente.

El tercer tipo de entrenamiento diseñado para la aplicación está basado en un principio similar al entrenamiento basado en plataformas, pero con la diferencia de que la posición de la plataforma es dinámica con respecto al muro de escalada. Este entrenamiento recibe el nombre de evitar la pelota.

En este entrenamiento el objetivo es evitar que la pelota, que se encuentra en constante movimiento, colisione con el escalador durante el mayor tiempo posible. El parámetro medido para determinar la evolución o mejora es el tiempo total transcurrido desde el inicio del entrenamiento hasta la detección de una colisión.

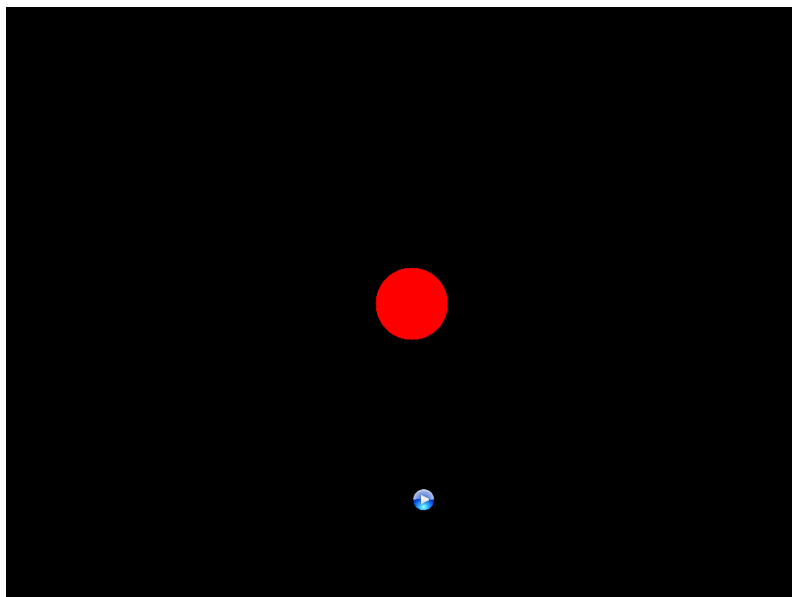


Figura 15. Entrenamiento basado en esquivar la pelota.

El punto inicial del entrenamiento estará marcado por un botón de color azul con el símbolo de “play”. Cuando el escalador coloque su mano sobre el botón de inicio se mostrará un círculo al rededor del botón mostrando la cuenta atrás de inicio del entrenamiento. Si este quitase la mano del botón antes de terminar la cuenta atrás esta se reiniciaría y esperaría a que el escalador volviese a colocar la mano sobre el botón.

El objetivo de esta cuenta atrás es permitir al usuario colocarse en una postura cómoda para el inicio del entrenamiento y poder así comenzar el entrenamiento cuando el este realmente preparado.

Durante el desarrollo de este entrenamiento el sistema central de procesado comprobará continuamente la posición del escalador con respecto a la pelota para determinar si se ha producido una colisión entre ambos. En caso de que se produjese una colisión, el juego se detendría y se mostraría sobre la superficie de escalada la silueta del escalador en contacto con la pelota para indicar al usuario con mas detalle el lugar en el cual se ha producido la colisión y poder prevenir el mismo error en futuros entrenamientos.

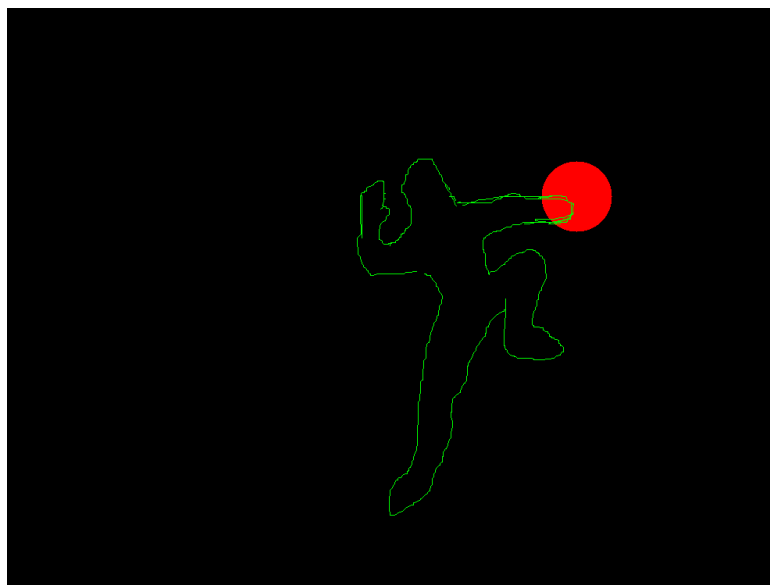


Figura 16. Silueta de colisión en entrenamiento basado en esquivar de pelota.

Al igual que en el resto de entrenamientos, al finalizar el entrenamiento siempre se mostrara un menú de información de resultados del entrenamiento si permitiendo reiniciar el entrenamiento o volver al menú principal para la selección de un entrenamiento diferente.

Existe un menú no mencionado hasta el momento utilizado para la configuración y calibración de la aplicación, al cual el usuario estándar no tendrá acceso durante el desarrollo de su entrenamiento. Este menú esta reservado al encargado de colocar y configurar la aplicación, evitando así la modificación de parámetros de calibración por accidente y desestabilizando el correcto funcionamiento de la aplicación.

En este menú la navegación y acceso a menús se realizara mediante un teclado inalámbrico conectado al dispositivo Raspberry Pi, mediante el cual se realizarán las tareas de calibración, configuración de sensibilidad y creación de rutas de entrenamiento para su posterior almacenamiento como ficheros de configuración a los que tendrá acceso el escalador durante sus entrenamientos.

El acceso a estos menús solo será necesario en caso de que los dispositivos de captura o proyección de imágenes sean recolocados. En este caso será necesario el acceso a los antes mencionados menús de configuración para la auto-calibración de la aplicación.

## 4.4 Implementación técnica

Uno de los principios básicos sobre el cual funciona la aplicación es la posición relativa entre captador de imágenes y proyector con respecto a la superficie de calibración. Mediante esta determinación de distancias la aplicación es capaz de realizar una homografía<sup>11</sup> entre la imagen capturada y la imagen que será proyectada para poder crear la sensación de realidad aumentada en el usuario.

Debido a esta necesidad de determinar la posición relativa entre, captador, proyector y superficie de proyección se creó el menú de auto-calibración, permitiendo mediante el tratamiento de la imagen capturada y la proyección de diferentes imágenes de referencia calcular dichas distancias.

Una vez ejecutado el proceso de auto-calibración este seguirá los siguientes pasos:

- Ajuste de posición de la capturadora de video.
- Proyección de imágenes de referencia.
- Tratamiento de las imágenes de referencia capturadas.
- Determinación de los puntos Homográficos.

En primer lugar la aplicación mostrará la imagen obtenida por la capturadora de video en tiempo real para que el encargado del posicionamiento de los dispositivos de la aplicación coloque la cámara en una posición en la cual la superficie de proyección se encuentre dentro de su rango de captura de imágenes. En este paso no es necesario alinear perfectamente capturadora y proyector, únicamente es necesario que la superficie de la pared de escalada sobre la cual se vaya a proyectar sea capturable por la cámara, puesto que si esto no fuese posible la posición del escalador nunca podría ser detectada por la cámara.

---

11 Homografía: En geometría, se denomina homografía a toda transformación proyectiva que determina una correspondencia entre dos figuras geométricas planas, de forma que a cada uno de los puntos y las rectas de una de ellas le corresponden, respectivamente, un punto y una recta de la otra.



Una vez cerciorados de que el captador se encuentra correctamente posicionado el sistema de proyección mostrara diferentes imágenes de referencia sobre la superficie de escalada. Estas imágenes serán capturadas por la cámara y procesadas por el dispositivo central Raspberry Pi para determinar la posición relativa entre los dispositivos de captura y proyección.

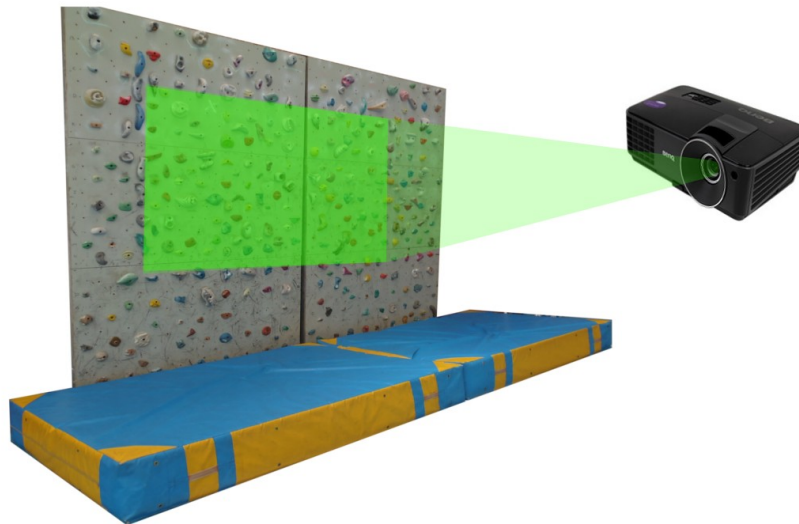


Figura 17. Imagen de referencia

Como se observa en la imagen anterior, la referencia utilizada para determinar la posición entre cámara y proyector es un simple fondo de color verde. Al procesar la imagen original de la superficie de proyección y compararla con la captura de la imagen de referencia se puede determinar mediante un proceso de sustracción de fondo la zona de la superficie sobre la cual es proyector sera capaz de mostrar información al usuario.

Para la sustracción de fondo se utilizaron las herramientas del modulo OpenCV antes mencionado, entre las cuales se encuentra la función **absdiff**<sup>2</sup>, la cual se encarga de obtener la diferencia pixel a pixel entre dos imágenes, devolviendo una imagen del mismo tamaño que las originales en la cual se almacena la diferencia de colores entre ambas en la posición correspondiente de cada pixel.

```
# Ejemplo de substraccion de fondo:  
Import cv2  
Diferencia = cv2.absdiff(img_1, img_2)  
diferencia_gris = cv2.cvtColor(diferencia, cv2.COLOR_BGR2GRAY)  
  
Cv2.imshow("diferencia", diferencia_gris)
```

Al ejecutar el código anterior utilizando como *img\_1* e *img\_2* las siguientes capturas, se obtendría el siguiente resultado:

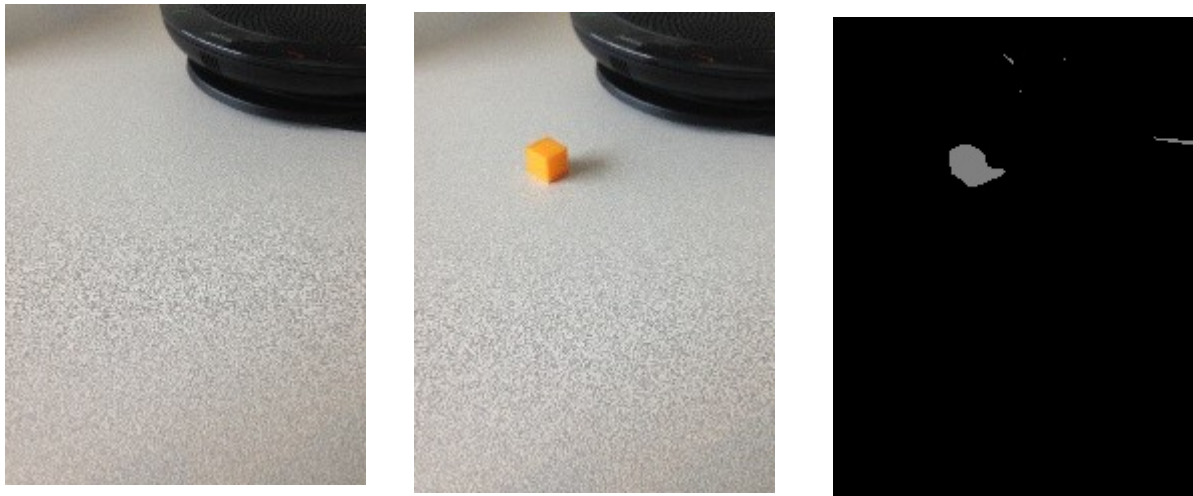


Figura 18. Resultado de la sustracción de fondo

El ejemplo anterior de sustracción es una muestra rápida del potencial del modulo OpenCV. Si se observan detenidamente los resultados se puede apreciar que el resultado de la resta entre ambas imágenes también incluye la sombra arrojada por el dado sobre la superficie, este es un problema sobre el cual se hablara mas detalladamente en el proceso de captura de movimiento aplicado al análisis de imágenes durante los entrenamientos antes mencionados.

Siguiendo este principio se puede obtener la diferencia entre las capturas de la superficie de escalada antes y después de la proyección de imágenes de referencia, y así sabiendo la posición original de la referencia en la imagen de proyección con respecto a la imagen capturada se puede realizar un sencillo calculo trigonométrico que permita determinar la posición relativa entre ambos dispositivos.

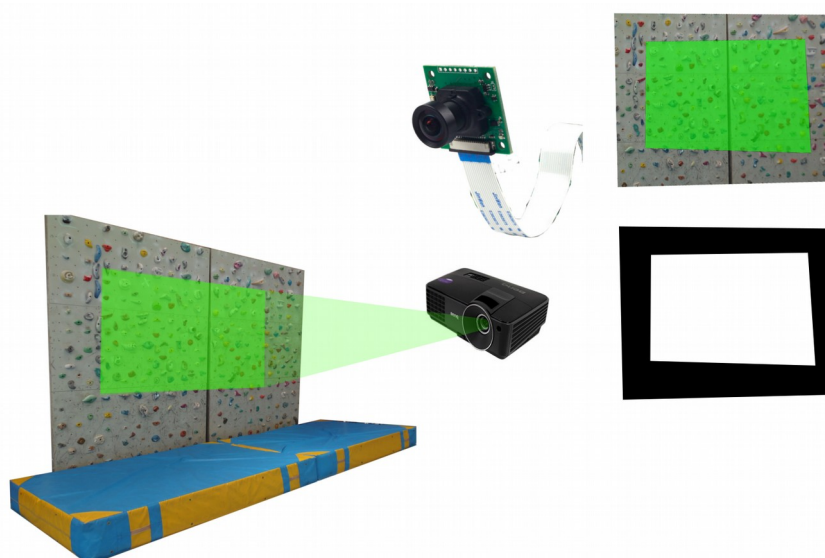


Figura 19. Proceso de calibración.

La imagen en blanco y negro muestra el resultado de sustracción de fondo de la imagen de referencia a partir de la cual se obtendrán los punto homográficos que se utilizaran para calcular la posición relativa entre proyector y capturadora mediante sencillo aunque tedioso calculo<sup>13</sup> matemático, que puede ser sustituido por la función findHomography<sup>14</sup>:

```
# Calculo de la matriz de homografia:
import cv2
import numpy as np

pts_src = np.array([corner_1, corner_2, corner_3, corner_4])
pts_dst = np.array([[0, 0], [width, 0], [width, height],[0, height]])

h, status = cv2.findHomography(pts_src, pts_dst)

img = cv2.warpPerspective(captura , h, (width, height))

Cv2.imshow("diferencia", img)
```

-pts\_src: esquinas del rectángulo blanco obtenido al restar la imagen de referencia

-pts\_dst: tamaño de la imagen destino de procesado.

Al aplicar la matriz  $H$  obtenida en el paso anterior a una imagen capturada por la cámara siempre se obtendrá la homografía de la original ajustada al tamaño deseado de procesado, de este modo la imagen homografía puede ser mostrada por el proyector y encajara perfectamente con la posición original, siempre y cuando los puntos de referencia se hayan calculado precisamente.

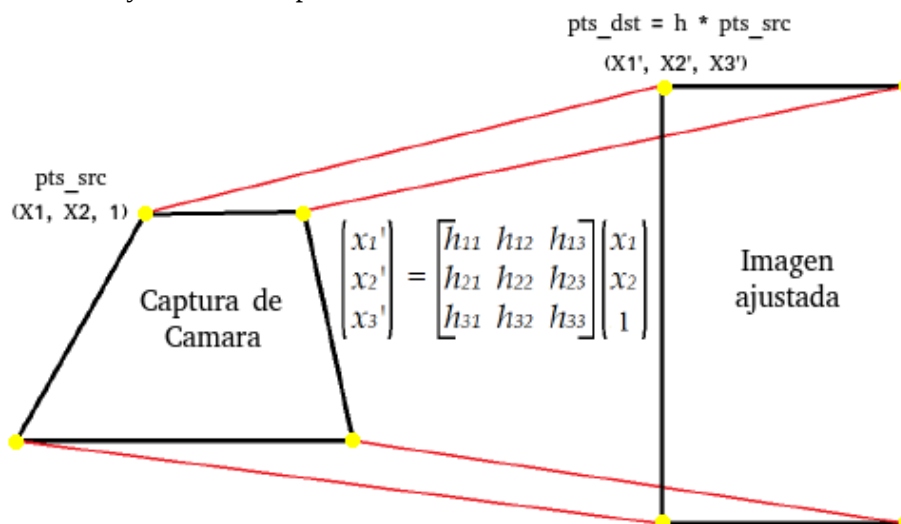


Figura 20. Proceso de homografía.

13 [Caclulo de la matriz de homografia matematicamente.](#)

14 [Metodo findHomography \(OpenCV\)](#)

Determinada la posición relativa entre proyector y capturadora se creará un archivo de calibración que contendrá los puntos homográficos utilizados para ajustar posición de los objetos detectados por la cámara con respecto al proyector. Estos puntos de calibración se utilizarán en todos los procesos en los que se necesite una interacción entre el usuario y la aplicación para determinar su posición o al interactuar con alguno de los menús.

El siguiente paso en el proceso de configuración es detectar las presas que se encuentran dentro de la superficie de escalada y a su vez dentro de la zona de proyección mediante funciones incluidas en el módulo OpenCV como por ejemplo la detección de bordes<sup>15</sup>. Para ello se utilizarán las coordenadas de homografía calculadas anteriormente para determinar la zona sobre la cual el proyector será capaz de proyectar imágenes dentro del rango de captura de la cámara.

Una vez recortada la zona de la imagen capturada por la cámara considerada relevante se procederá a aplicar métodos de detección de bordes como es el caso del método **Canny Edge Detection**, ampliamente conocido entre los procesos de tratamiento digital de imágenes. Antes de realizar la detección de bordes es recomendable realizar un filtrado para eliminar el ruido y evitar falsos positivos en la detección de bordes.

A continuación se muestra un sencillo ejemplo de la utilización del módulo OpenCV para la detección de bordes en una imagen, así como el resultado obtenido al procesar una imagen cualquiera:

```
# Calculo de bordes:  
import cv2  
import numpy as np  
  
kernel = np.ones((5,5),np.float32)/25  
img_filter = cv2.filter2D(img,-1,kernel)  
  
edges = cv2.Canny(img_filter,100,200)  
  
Cv2.imshow("diferencia", edges)
```

---

15 [Deteccion de bordes \(OpenCV\)](#)

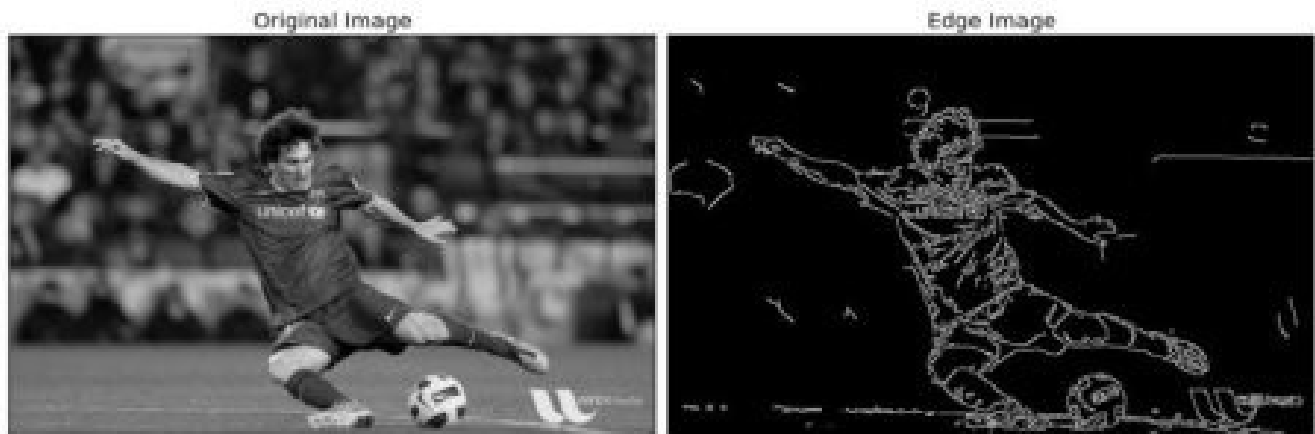


Figura 21. Resultado de detección de bordes mediante Canny edge.

El método Canny edge detector es un algoritmo para detectar amplios rangos de bordes en imágenes. Este algoritmo extrae información estructural de las imágenes tratadas para reducir considerablemente la cantidad de información a procesar, reduciendo la carga computacional y ahorrando tiempo de procesado.

Entre otros muchos métodos de detección de bordes el método Canny es uno de los algoritmos con mejores resultados y por ello se ha convertido en uno de los métodos mas populares de detección de bordes. El algoritmo de Canny se puede resumir en 5 pasos:

- 1: Aplicación de filtro Gaussiano para suavizar la imagen.
- 2: Detectar la intensidad de los gradientes en la imagen.
- 3: Aplicación de supresión de valores máximos para reducir la información relevante.
- 4: Aplicación de un doble umbral delimitando bordes potenciales.
- 5: Detección de bordes mediante Histéresis.

Tras múltiples ensayos con imágenes reales capturadas de la superficie de escalada mediante la cámara utilizada para la aplicación se determino un proceso de pre-tratado de la imagen antes de aplicar el algoritmo Canny para obtener unos mejores resultados a la hora de detectar los bordes de las presas. Este proceso se puede resumir en los siguientes pasos:

- 1: Filtro Bilateral (eliminación de ruido)
- 2: Filtro 2D de afilado (aumentando el contraste entre esquinas)

```
# Deteccion de bordes con pre-procesado:
Import cv2
Import numpy as np

Sharp = np.array([[ 0, -1,  0],
                  [-1,  5, -1],
                  [ 0, -1,  0]]),

pre_fil = cv2.bilateralFilter(img,9,75,75)
cv2.filter2D(pre_fil, -1, sharp, pre_fil)

edges = cv2.Canny(img_filter,100,200)

contours = cv2.findContours(post_filtrado, cv2.RETR_EXT,cv2.CHAIN_APPROX_SIMP)[-2]
cv2.drawContours(img_dst, contours, -1, (0,255,0) , 2)

Cv2.imshow("diferencia", img_dst)
```

El objetivo de este pre-tratamiento de la imagen es suavizar las imperfecciones o ruido introducido por la capturadora de video, y tras eliminar las impurezas de la imagen se intentará acentuar los contornos propios de la imagen para realzar el contraste entre los objetos y el fondo, aumentando considerablemente la capacidad de detección de objetos.

Cabe destacar que durante el proceso de detección de presas el usuario tendrá la opción de modificar la sensibilidad de algunos de los parámetros de detección como son los mostrados anteriormente. A parte de los antes mencionados también se pueden modificar parámetros como el tamaño máximo y mínimo de detección para evitar la detección de objetos no relevantes para la aplicación, como es el caso de tornillos o pequeños agujeros en la superficie de escalada.

Una vez detectadas las presas y almacenadas en la base de datos es el momento de que el usuario determine cuales serán las rutas de entrenamiento a las que el escalador tendrá acceso. Para este proceso es necesario utilizar periféricos para el control del puntero de selección.

Para la selección de las rutas se proyectarán las presas detectadas sobre las mismas para que el usuario mediante el puntero de selección escoja las presas que desee para la ruta simplemente clicando sobre cada una de ellas en el orden en el cual desee realizar la ruta. Si desea eliminar una de las presas de la ruta simplemente tendrá que volver a pulsar sobre la presa indicada y se eliminara automáticamente de la ruta.

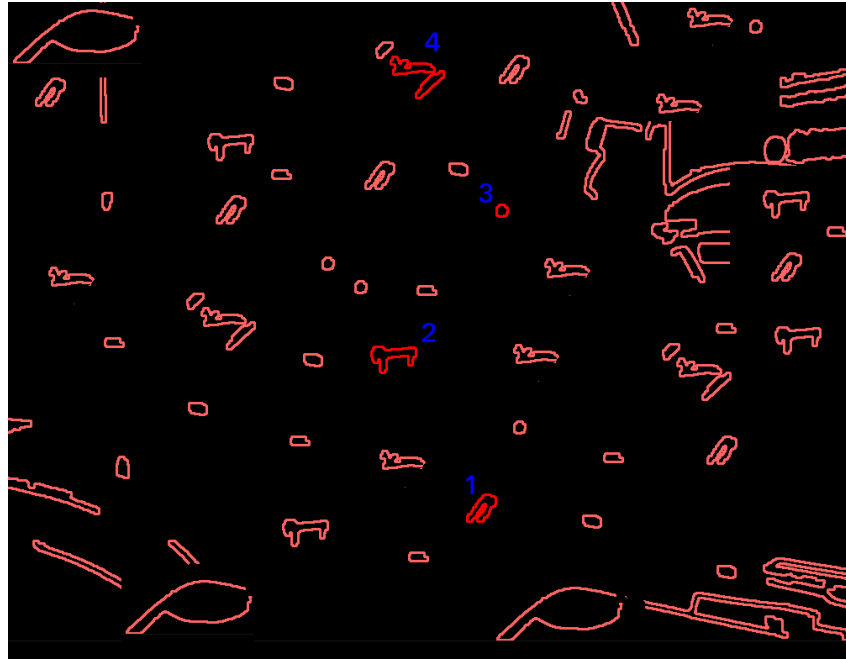


Figura 22. Menú de creación de rutas.

Una vez establecidas las presas que compondrán una ruta mediante el teclado se pulsará la tecla de guardado, la ruta se almacenará automáticamente y el proceso de creación de rutas se reiniciará permitiendo al usuario seleccionar una nueva ruta o terminar con el proceso de configuración y comenzar el entrenamiento.

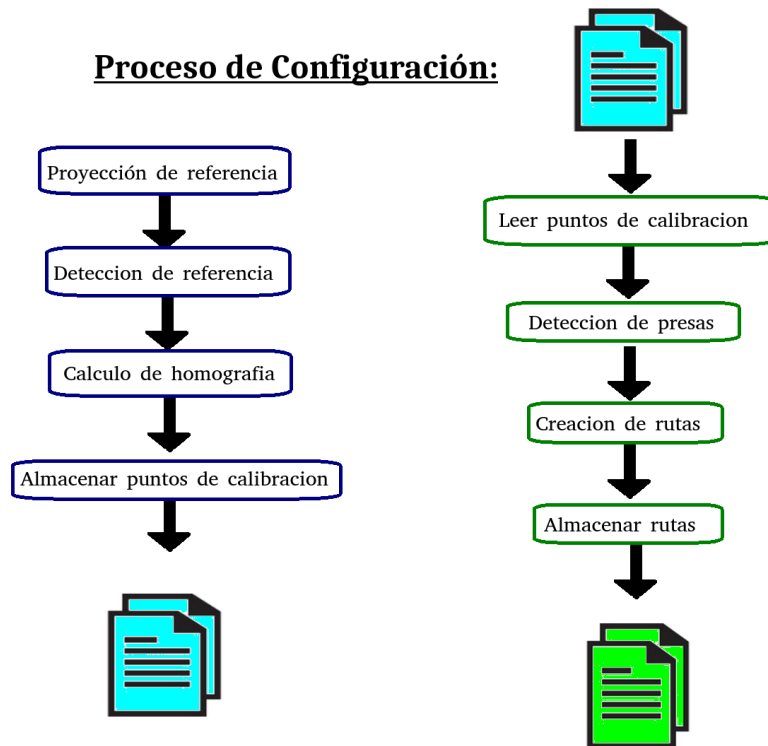


Figura 23. Proceso de configuración de la aplicación.

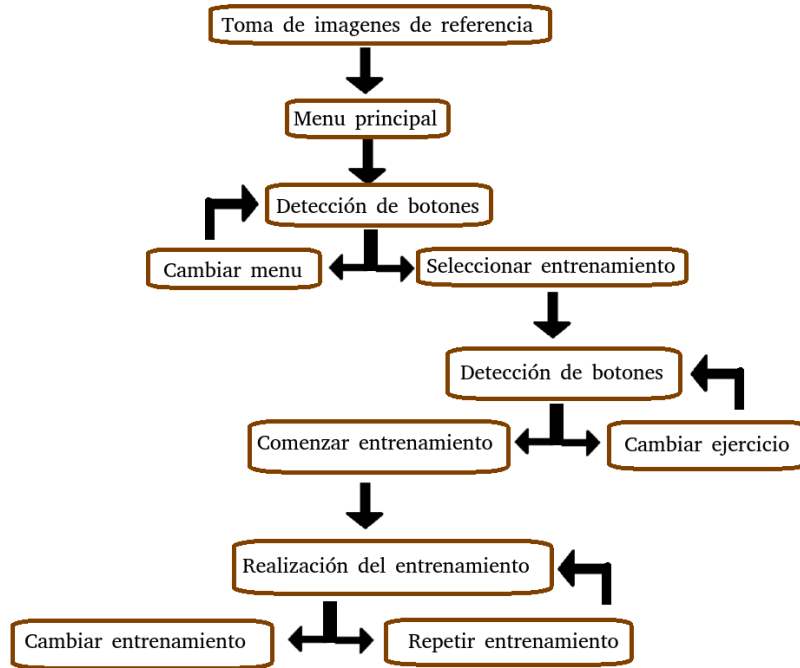


Figura 24. Proceso de selección de entrenamiento.



## 5 Resultados

Tras implementar el código y realizar pruebas aisladas es necesario realizar pruebas de campo para comprobar el comportamiento de la aplicación en un entorno real de trabajo. Para ello se colocó un proyector a una distancia razonable junto con el dispositivo Raspberry Pi al cual lleva conectada la anteriormente mencionada Camera Pi v2.1.



*Figura 25. Colocación de los dispositivos de la aplicación.*

Para aumentar la superficie de proyección es necesario colocar el proyector a una distancia ligeramente alejada de la superficie de entrenamiento, pero por contra al alejar más los dispositivos de proyección se alejan los sistemas de captura de imágenes, con lo cual se reduce la calidad de captura de las imágenes y puede perjudicar al correcto procesamiento de imágenes.

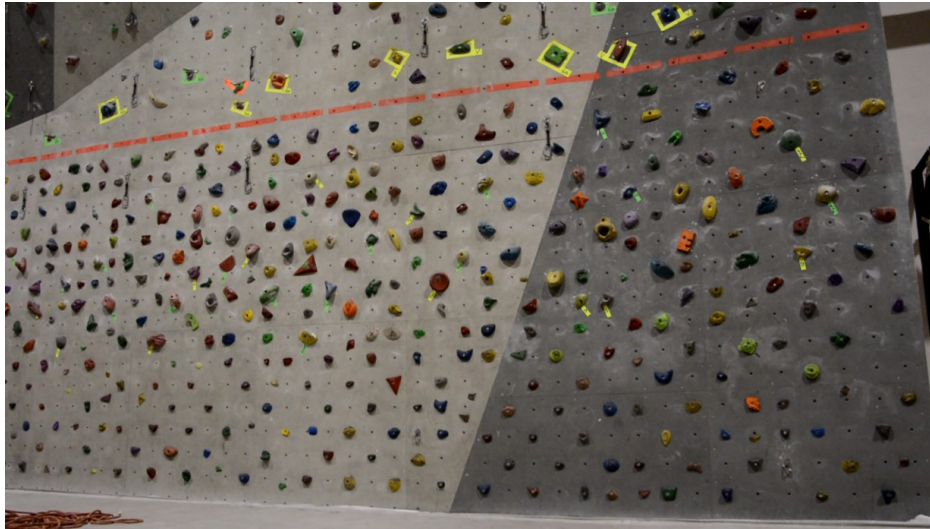


Figura 26. Superficie de entrenamiento.

Una vez colocados los dispositivos el siguiente paso es ejecutar la función de auto-calibrado antes mencionada. Por lo tanto se tomará una imagen de referencia de la superficie de entrenamiento y posteriormente se proyectará la imagen de referencia sobre esta para determinar la posición relativa entre captador y proyector para determinar los puntos de homografía que permitirán la calibración del resto de módulos de la aplicación.

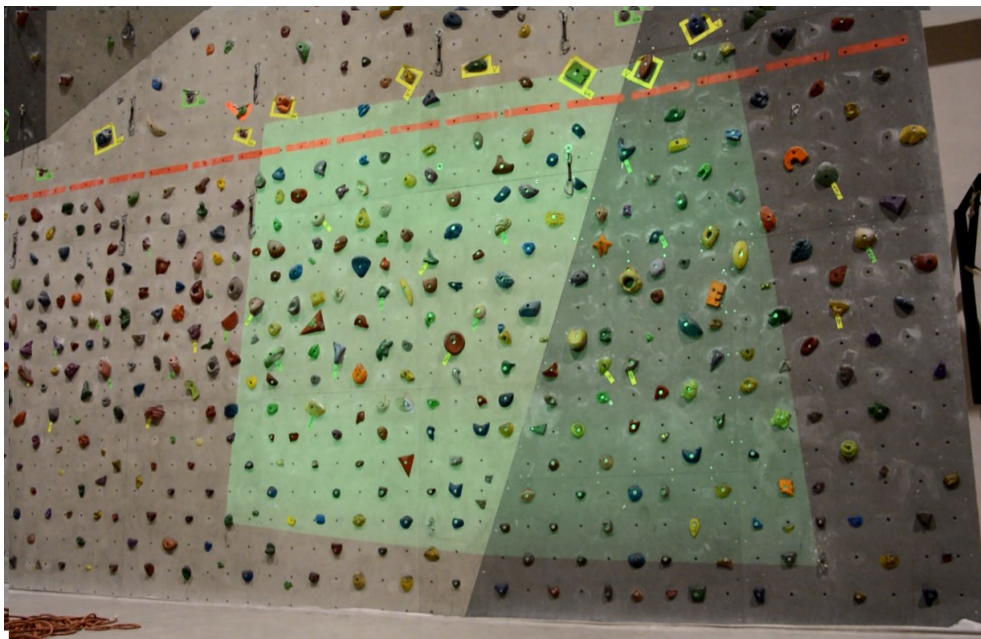


Figura 27. Superficie de entrenamiento con imagen de referencia proyectada.

Tras la determinación de los puntos de homografía la aplicación detectará de forma automática cualquier presa presente en la superficie de entrenamiento y mostrará los resultados sobre el muro de entrenamiento. De modo que el resultado final será mostrado al usuario proyectando los contornos de las presas detectadas sobre el propio muro de entrenamiento.



*Figura 28. Presas detectadas automáticamente.*

Llegados a este punto el usuario podrá modificar los parámetros de sensibilidad de detección de presas o podrá continuar con la creación de las rutas de entrenamiento. Una finalizada la creación de rutas la aplicación almacenará automáticamente toda la información en la base de datos para no tener que repetir ninguno de estos procesos de calibración, a no ser que se modifique la posición del sistema de proyección o captura de imágenes.

Tras la calibración el paso natural es comenzar el entrenamiento, para lo cual es necesario ejecutar la función principal de la aplicación que mostrará en una primera instancia el menú principal<sup>16</sup> para poder seleccionar el tipo de entrenamiento y su configuración. Durante todo el proceso de selección de entrenamiento tan solo se podrá interactuar con la aplicación mediante los botones proyectados sobre la pared los cuales reaccionaran a su pulsado reproduciendo un sonido de confirmación y cambiando su color momentáneamente de verde a rojo.



Figura 29. Menú principal de la aplicación.

Una vez mostrado el menú principal el usuario navegará entre los menú pulsando los botones proyectados sobre la pared para crear la sensación de interactividad con la aplicación y crear un entorno de realidad aumentada en el cual el entorno reacciona al modificarse la situación del escalador.

El primero de los entrenamientos mostrados es el basado en rutas<sup>17</sup>. Tras la selección de este tipo de entrenamiento se mostrará un temporizador en la parte superior derecha, el cual permanecerá a cero hasta que el escalador active el temporizador.

Para activar el temporizador el escalador deberá colocar una o ambas manos sobre la presa inicial de la ruta resaltada mediante la proyección de su contorno en color rojo al rededor de la misma. Una vez el usuario coloque su mano sobre la presa comenzara mostrarse un circulo de color verde rodeando la misma presa. Una vez el circulo este completamente cerrado se iniciara el temporizador de tiempo y se proyectarán la indicaciones necesarias para que el escalador complete el entrenamiento.

16 Anexo: Video mostrando el funcionamiento del menú principal ( video\_01.mp4)

17 Anexo: Video mostrando el funcionamiento del entrenamiento basado en ruta ( video\_02.mp4)



Figura 30. Temporizador de inicio de entrenamiento basado en ruta.

Al detectar que el escalador ha alcanzado una de las presas de la ruta se mostrará una animación y se reproducirá un sonido para indicar que se ha alcanzado un nuevo punto en el entrenamiento. La animación consiste en un círculo rojo que aumentará progresivamente su radio para simular una onda expansiva con el mismo centro que la presa alcanzada.



Figura 31. Animación proyectada al detectar una presa alcanzada.

Para ayudar al usuario a completar el entrenamiento se proyectará una línea azul que indicará la posición de la siguiente presa de la ruta. Esta línea será dibujada tras cada detección y eliminada si el escalador suelta alguna de las presas.



Figura 32. Entrenamiento basado en ruta iniciado.

Por ultimo al alcanzar la presa final del entrenamiento se mostrará una animación que indicará que el entrenamiento ha finalizado correctamente. También se reproducirá un sonido determinado indicando el fin del entrenamiento.



Figura 33. Animación al finalizar entrenamiento basado en ruta.

Una vez finalizado el entrenamiento se mostrara un panel informando del tiempo invertido en el entrenamiento así como dos botones en el centro de la superficie de escalada con los que el usuario podrá repetir el mismo entrenamiento, o volver al menú principal para seleccionar un entrenamiento diferente.



Figura 34. Menú informativo al finalizar el entrenamiento basado en ruta.

En el caso del entrenamiento basado en plataformas<sup>18</sup> el funcionamiento es similar al anterior. Tras la selección de este tipo de entrenamiento en el menú principal, se proyectara la imagen inicial del entrenamiento sobre la superficie de escalada y se esperará a que el usuario active el temporizador inicial que dará paso al comienzo del entrenamiento.



Figura 35. Inicio de entrenamiento basado en ruta.

A diferencia del entrenamiento anterior, en este se utiliza un botón de color azul proyectado sobre la superficie de escalada para indicar la posición en la cual el escalador debe colocar la mano para comenzar el temporizador inicial. El entrenamiento comenzará cuando el círculo este completamente cerrado también en este caso.

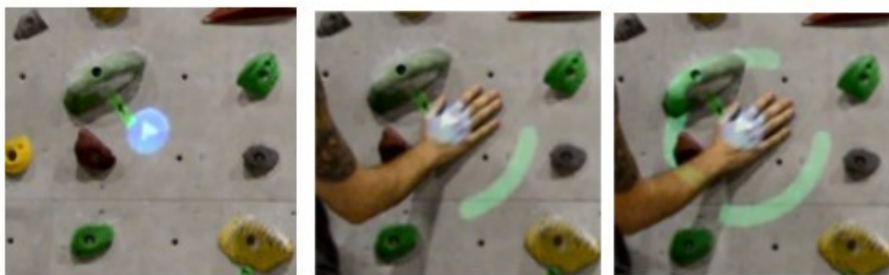


Figura 36. Temporizador de inicio de entrenamiento basado en plataformas

18 Anexo: Video mostrando el funcionamiento del entrenamiento basado en plataformas ( video\_03.mp4)

Para terminar con éxito el entrenamiento basado en plataformas el escalador deberá moverse desde el punto inicial marcado por el botón azul hasta el punto final, marcado por una estrella, sin entrar en contacto con ninguna las plataformas proyectadas sobre la superficie de escalada.



Figura 37. Procesos del entrenamiento basado en plataformas.

Una vez alcanzado el punto final del entrenamiento, si este no ha entrado en contacto con ninguna de las plataformas, se proyectará una animación indicando la correcta finalización del entrenamiento y se procederá a mostrar un menú informativo al igual que en el caso anterior.



Figura 38. Animación final entrenamiento basado en plataformas.

El proceso se repite para el ultimo entrenamiento basado en esquivar la pelota<sup>19</sup>, el cual cuenta con las mismas características que los entrenamientos anteriores.

19 Anexo: Video mostrando el funcionamiento del entrenamiento basado en esquivar la pelota ( video\_04.mp4)



## 6 Conclusiones y propuestas de futuro

Tras múltiples pruebas de campo y analizando los resultados mostrados en el apartado anterior se puede llegar a la conclusión de que se ha cumplido con los objetivos propuesto en una primera instancia. El primer objetivo cumplido la simplicidad de implantación y configuración de la aplicación, cumplido mediante la implantación de procesos automáticos integrados en un único dispositivo que no necesita mas configuración que ser colocado frente a la superficie de entrenamiento y conectado a una fuente de alimentación.

Gracias a la sencillez del proceso de calibración la aplicación puede ser colocada en cualquier superficie estable sin necesidad de conocimientos técnicos sobre la aplicación. Unicamente es necesario colocar los dispositivos a una distancia razonable y orientados hacia la superficie de proyección con lo que se conseguirá utilizar la aplicación en cualquier superficie que el usuario desee de manera fácil y rápida.

Como se muestra en apartados anteriores el usuario es capaz de elegir entre diferentes tipos de entrenamientos, así como la posibilidad de configurar las rutas utilizadas para los entrenamientos cuando este lo desee, añadiendo un grado mas de personalización a la aplicación.

Se puede afirmar que mediante las múltiples animaciones proyectadas y la reproducción de sonidos al avanzar el estado del entrenamiento o al pulsar alguno de los botones la aplicación consigue crear la sensación de interactividad entre el escalador y la aplicación, cumpliendo así el el objetivo principal de la aplicación que es sumergir al escalador en una experiencia interactiva que le permita mejorar su técnica al mismo tiempo que se entretiene con métodos diferentes de escalada.

Por otro lado, gracias a la implementación de una aplicación modular la incorporación de nuevos tipos de entrenamiento o la posibilidad de modificar los entrenamientos actuales es muy sencilla puesto que el entorno sobre el cual el usuario navegará para seleccionar y configurar cada uno de ellos esta diseñado desde un punto de vista escalable para poder aumentar sin problemas las funcionalidades de la aplicación.

En cuanto a los puntos débiles de la aplicación se podría hablar de la capacidad de procesado que soportan los dispositivos Raspberry Pi. Durante el desarrollo de la aplicación se tuvieron que crear procesos de tratamiento digital de la imagen propios debido a que algunos de los propuestos por las librerías de OpenCV eran computacionalmente demasiado pesados para que el dispositivo Raspberry Pi pudiese tratar la imagen y mostrar los resultados en tiempo real, generándose un ligero retraso entre el estado real del escalador y la proyección de resultados sobre la superficie de escalada.

Esta falta de capacidad de procesado se ve reflejada en la creación de métodos menos precisos para el reconocimiento del a posición del escalador y su seguimiento en tiempo real, que al utilizar menor recursos y estar menos optimizados que los ofrecidos por las librerías de OpenCV producen unos resultados menos precisos en cuanto a la detección de la silueta del escalador.

Como conclusión la aplicación ha alcanzado los requisitos propuestos inicialmente obteniendo unos resultados aceptables teniendo en cuenta los equipos utilizados para ello y las dificultades que supone ofrecer resultados competitivos sin una gran capacidad de procesado. De cualquier modo, la utilización de conocimientos de electrónica, programación y tratamiento digital de imágenes ha permitido que esta aplicación cumpla con su objetivo de crear un entorno de realidad aumentada para la ayuda y disfrute de cualquier escalador sin conocimientos técnicos al respecto.

## 7 Bibliografía

- [1] Aplicación de Realidad aumentada aplicada a la escalada.  
<https://augmentedclimbing.com/>
- [2] Especificaciones técnicas de Camera Pi v2.1.  
<https://www.raspberrypi.org/documentation/hardware/camera/>
- [3] Sistema Operativo Raspbian, basado en Debian Linux.  
<https://www.raspberrypi.org/downloads/raspbian/>
- [4] Lenguaje de programación Python 2.7.  
<https://www.python.org/download/releases/2.7/>
- [5] Tutoriales sobre tratamiento digital de la imagen OpenCv.  
<https://opencv-python-tutroals.readthedocs.io/en/latest/>
- [6] Especificaciones de la librería NumPy  
<http://www.numpy.org/>
- [7] Cálculos matemáticos para la obtención de la matriz de homografía.  
<http://www.csc.kth.se/~perrose/files/pose-init-model/node17.html>

## 8 Anexos

**Anexo A:** Calendario plazos previstos para completar las fases del proyecto.

- Imagen adjunta: calendario.png

**Anexo B:** Video mostrando el funcionamiento del menú principal de la aplicación.

-Video adjunto: video\_01.mp4

**Anexo C:** Video mostrando el funcionamiento del entrenamiento basado en rutas.

-Video adjunto: video\_02.mp4

**Anexo D:** Video mostrando el funcionamiento del entrenamiento basado en plataformas.

-Video adjunto: video\_03.mp4

**Anexo E:** Video mostrando el funcionamiento del entrenamiento basado en esquivas.

-Video adjunto: video\_04.mp4