

GRADO EN INGENIERÍA INFORMÁTICA



Trabajo Fin de Grado

Aplicación para seguimiento de objetos en experimentos de
comportamiento animal

Autor: Javier Martín Garay

Tutor: Miguel Ángel García Garrido

2019

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

GRADO EN INGENIERÍA INFORMÁTICA

Trabajo de Fin de Grado

Aplicación para seguimiento de objetos en experimentos de
comportamiento animal

Autor: Javier Martín Garay

Tutor: Miguel Ángel García Garrido

TRIBUNAL

Presidente: Cristina Losada Gutiérrez

Vocal 1º: M^a del Carmen Pérez Rubio

Vocal 2º: Miguel Ángel García Garrido

FECHA:

Resumen

El objetivo de los neurocientíficos es entender las bases del aprendizaje, la memoria, el comportamiento, la percepción sensorial y la consciencia.

Aquellos que estudian la memoria se valen de experimentos con organismos en espacios, condiciones y tiempo controlados para obtener métricas de sus movimientos.

Se necesitan software comercial de elevado coste para analizar las imágenes de los movimientos del organismo y obtener estas métricas. En ocasiones el coste es inasumible para el neurocientífico.

El objetivo de este proyecto es desarrollar una herramienta software no comercial y basada en visión artificial, que el investigador use para obtener las métricas necesarias.

Palabras clave Visión artificial · Detección de primer plano · Seguimiento de objetos, Neurociencia, Memoria

Abstract

The goal of neuroscientific is the understanding of the biological basis of learning, memory, behavior, perception, and consciousness.

Those who study the memory use experiments with living body under controlled space, conditions and time.

High-cost commercial software is required to analyze the images of body movements and obtain these metrics. Sometimes the cost is unaffordable for the neuroscientist.

The goal of this project is to develop a non-commercial computer vision based software tool, which the researcher may use to obtain the necessary metrics.

Keywords Computer vision · Foreground subtraction · Object tracking · Neuroscience · Memory

Índice

1 Introducción	9
2 Estado del arte	13
2.1 Investigación y experimentos	13
2.1.1 Flujo de trabajo	15
2.1.2 Registro de imágenes y análisis	15
2.1.3 Técnicas para seguimiento de objetos	17
2.1.4 Técnicas de predicción y corrección de posición	20
2.2 Soluciones disponibles y características	23
2.3 Necesidades	25
3 Análisis y diseño	27
3.1 Medios	27
3.2 Requisitos	28
3.3 Arquitectura	29
3.4 Estructuras de datos	32
3.5 Base de datos	32
3.6 Interfaz de usuario	38
4 Implementación	43
4.1 Modelo	43
4.1.1 Módulos	43
4.1.1.1 Audio Video Software Development Kit	44
4.2 Controlador	48
4.2.1 Reproductor de video	49
4.2.2 Seguimiento de objetos	51
4.2.3 Filtros de localización	53
4.2.4 Módulos	53
4.2.4.1 Computer Vision Software Development Kit	54
4.3 Vista	56
5 Pruebas	57
5.2 Validación de requisitos	57
5.1 Validación de las métricas	59
6 Conclusiones y trabajo futuro	63
6.1 Conclusiones	63
6.2 Trabajo futuro	63

7 Manual de usuario	65
7.1 Instalación del intérprete y dependencias	65
7.2 Instalación de la aplicación.....	69
7.3 Abrir la aplicación	69
7.4 Flujo de trabajo.....	70
7.5 Nuevo experimento y carga de experimentos.....	70
7.6 Configuración del experimento	70
7.7 Añadir y asignar fuentes de video	71
7.8 Modelo digital del aparato	75
7.9 Áreas de interés	77
7.10 Reproductor de video.....	77
7.11 Zonas	78
7.12 Configuración del sistema de seguimiento de objetos.....	81
7.13 Análisis de las pruebas	83
7.14 Grabación de la prueba	84
7.15 Visualizar resultados.....	85
7.16 Exportar resultados	87
8 Análisis económico.....	89
8.1 Análisis económico.....	89
8.2 Plan de negocio.....	90
Apéndice 1	91
Estructura interna de la librería LibAV	91
Apéndice 2	93
Estructura interna de la librería LibCV	93
Apéndice 3	95
Estructura interna de la librería DEVKit	95
Bibliografía.....	97
Glosario de términos	99
Glosario de definiciones	99
Glosario de acrónimos y abreviaturas	101

Índice de figuras

Figura 1. Tipos de aparato	13
Figura 2. Instalación habitual en experimentos de memoria	14
Figura 3. Sistema de registro de imágenes y análisis.	15
Figura 4. Modelado digital del aparato.....	16
Figura 5. Izquierda: Mapa de calor, Derecha: Mapa de recorrido.....	17
Figura 6. Resultado del seguimiento y localización del animal.	17
Figura 7. Pasos para el seguimiento de objetos.	18
Figura 8. Detección de primer plano por diferenciación entre imágenes.....	19
Figura 9. Detección de primer plano por filtro de mediana temporal.	19
Figura 10. Resultado al aplicar un filtro Kalman a una señal.....	20
Figura 11. Algoritmo recursivo del filtro Kalman.....	21
Figura 12. Patrón Modelo-Vista-Controlador o MVC.	30
Figura 13. Estructura del modelo.	30
Figura 14. Estructura de la vista.	31
Figura 15. Estructura de del controlador.	31
Figura 16. Estructura de la base de datos.	32
Figura 17. Interfaz. de la pestaña ‘Archivo’.....	38
Figura 18. Interfaz. de la pestaña ‘Experimento’.....	39
Figura 19. Interfaz. de la pestaña ‘Fuente’.....	39
Figura 20. Interfaz. de la pestaña ‘Aparato’.....	40
Figura 21. Interfaz. de la pestaña ‘Zonas’.....	40
Figura 22. Interfaz. de la pestaña ‘Seguimiento’.....	41
Figura 23. Interfaz. de la pestaña ‘Análisis’.....	41
Figura 24. Interfaz. de la pestaña ‘Resultados’.....	42
Figura 25. Modelo.	43
Figura 26. Visión general del sistema de módulos y librerías externas.	44
Figura 27. Estructura interna del módulo “Audio Video Software Development Kit”..	44
Figura 28. Flujo de funcionamiento del módulo “Audio Video Software Development Kit”.....	47
Figura 29. Estructura interna del controlador.....	48
Figura 30. Vista en detalle de los módulos y clases del controlador.....	49
Figura 31. Datos del fotograma de video.	50
Figura 32. Sincronización de reloj del reproductor de video.	50
Figura 33. Implementación: Pasos para el seguimiento de objetos con visión artificial.	51
Figura 34. Proceso para enmascarar la imagen.	51
Figura 35. Estructura interna del módulo “Computer Vision Software Development Kit”.....	54
Figura 36. Flujo de funcionamiento del módulo “Computer Vision Software Development Kit”.....	55
Figura 37. Estructura interna de la vista.	56
Figura 38. Grafica de distancia recorrida. La distancia se muestra en metros.	60
Figura 39. Grafica de tiempo de análisis. Los tiempos se muestran en minutos.	60
Figura 40. Ventana para ejecutar aplicaciones de Windows 10.....	65
Figura 41. Vista de la consola de Windows 10.	66

Figura 42. Vista del terminal de MacOS.	67
Figura 43. Vista de la pestaña “Experiment”.	70
Figura 44. Detalle de los controles para selección de los grupos/animales.	71
Figura 45. Detalle de los controles para selección de los grupos/intentos.	71
Figura 46. Vista de la pestaña “Sources”.	71
Figura 47. Vista del navegador de pruebas o test.	72
Figura 48. Acceso a un elemento del navegador de pruebas.	73
Figura 49. Selección de un elemento del navegador de pruebas.	73
Figura 50. Prueba seleccionada.	74
Figura 51. Prueba con fuente asignada.	74
Figura 52. Borrado de la asignación de fuente.	74
Figura 53. Borrado de una fuente de video.	75
Figura 54. Vista de la pestaña “Aparato”.	75
Figura 55. Selección del aparato, divisiones del aparato, brazos del aparato, regla y color.	76
Figura 56. Visualización del aparato, puntos de tamaño y giro. Regla configurada	76
Figura 57. Selección de área, tipo y color.	77
Figura 58. Vista del control del reproductor.	78
Figura 59. Selección de fuente en el reproductor.	78
Figura 60. Vista de la pestaña de selección de zonas.	78
Figura 61. Controles de zona y listado de zonas.	79
Figura 62. Cambio de configuración de las métricas de una zona.	79
Figura 63. Vista de las áreas. Área central no seleccionada y seleccionada.	80
Figura 64. Guardado de selección de áreas de una Zona.	80
Figura 65. Selección de vista (Todos/Aparato/Áreas).	81
Figura 66. Vista de la pestaña “Tracking”.	81
Figura 67. Vista de la configuración del sistema de seguimiento.	82
Figura 68. Vista de la configuración del sistema de seguimiento.	83
Figura 69. Vista de la pestaña de “Analysis”.	83
Figura 70. Controles para el análisis de archivos de video.	84
Figura 71. Vista del monitor.	84
Figura 72. Controles para la grabación y análisis.	85
Figura 73. Vista de la pestaña “Results”.	85
Figura 74. Vista de los resultados y controles.	86
Figura 75. Mapa del recorrido y mapa de calor.	86

Índice de cuadros

Tabla 1. Soluciones software disponibles.....	23
Tabla 2. Características de las soluciones.	23
Tabla 3. Coste de las soluciones.....	25
Tabla 4. Requisitos funcionales.....	29
Tabla 5. Requisitos no funcionales.....	29
Tabla 6. Definición SQL de la tabla "Experiment".	33
Tabla 7. Definición SQL de la tabla "Animals Group".	33
Tabla 8. Definición SQL de la tabla "Animal".	33
Tabla 9. Definición SQL de la tabla "Trials Group".	34
Tabla 10. Definición SQL de la tabla "Trial".	34
Tabla 11. Definición SQL de la tabla "Source".	34
Tabla 12. Definición SQL de la tabla "Test".	34
Tabla 13. Definición SQL de la tabla "Result".	35
Tabla 14. Definición SQL de la tabla "Result zones".	35
Tabla 15. Definición SQL de la tabla "Apparatus".	35
Tabla 16. Definición SQL de la tabla "Apparatus area".	36
Tabla 17. Definición SQL de la tabla "Áreat".	36
Tabla 18. Definición SQL de la tabla "Zone".	36
Tabla 19. Definición SQL de la tabla "Zone apparatus area".	36
Tabla 20. Definición SQL de la tabla "Zone area".	36
Tabla 21. Definición SQL de la tabla "Rulert".	37
Tabla 22. Definición SQL de la tabla "Trackert".	37
Tabla 23. Definición SQL de la tabla "Scalert".	37
Tabla 24. Pseudocódigo para segmentar una imagen.....	52
Tabla 25. Requisitos funcionales validados.	58
Tabla 26. Requisitos no funcionales validados.	59
Tabla 27. Horas de trabajo del personal.	89
Tabla 28. Coste del personal.....	90
Tabla 29. Coste total de la ejecución material del proyecto.	90

1 Introducción

Neurociencia es la disciplina científica que estudia el desarrollo del sistema nervioso, su estructura, funciones, bases moleculares y patologías entre otros.

Su estudio se realiza a varios niveles, desde el puramente molecular hasta las funciones cognitivas, pasando por el neuronal, redes neuronales de pequeño tamaño y redes neuronales de gran tamaño.

El objetivo de los investigadores de esta disciplina científica o neurocientíficos, es entender las bases del aprendizaje, la memoria, el comportamiento, la percepción sensorial y la consciencia.

Concretamente, aquellos que estudian la memoria, lo hacen para entender como esta función cerebral permite a un organismo codificar la información, almacenarla y recuperarla.

Para ello, se valen de experimentos diseñados específicamente, que permiten medir en un organismo aspectos concretos de la memoria, como la memoria espacial o la memoria a corto o largo plazo.

Habitualmente, estos experimentos se llevan a cabo en ratas, ratones y peces usando espacios controlados, en condiciones de aislamiento y periodos de tiempo limitado.

De estos experimentos, los investigadores necesitan extraer métricas con las que evaluar los resultados experimentales. Estas métricas, se obtienen de los movimientos del animal durante el experimento y suelen comprender aspectos como distancia recorrida, velocidad media y tiempo en inmovilidad. También, se recoge el orden de entrada, número de entradas y tiempo dentro de aquellas zonas determinadas por el investigador. Finalmente, estas métricas suelen visualizarse como mapas de calor y mapas del camino recorrido.

Extraer estas métricas requiere el uso de herramientas como cámaras de video, cableado, aparatos especialmente diseñados, como laberintos y herramientas software que procesen las imágenes de video mediante técnicas de visión artificial que detecten, sigan y contabilicen los movimientos del animal.

Desafortunadamente, no es viable contabilizarlos manualmente debido al tiempo requerido y la necesidad de métricas fiables, libres de errores e imparciales.

Si bien, existen herramientas y software adecuados, generalmente tienen un coste elevado que en ocasiones es inasumible y/o justificable. Especialmente, en el caso del software, no existen prácticamente alternativas no comerciales o no son lo suficientemente flexibles.

El objetivo de este proyecto es desarrollar una herramienta software no comercial que, el investigador use para obtener las métricas que necesita.

Dotarla con la funcionalidad y posibilidad de configuración necesaria para definir el experimento, realizar el análisis y visualizar los resultados.

- Grupos y número de animales.
- Grupos y número de test.
- Selección de fuentes de video.
- Modelado digital del aparato.
- Selección de zonas de interés.
- Selección de las métricas a contabilizar.
- Visualización y exportación de resultados.

Desarrollar e integrar un reproductor de video, librerías de acceso a video y cámaras basándonos en librerías no comerciales como OpenCV y FFMPEG.

Usar técnicas de visión artificial integradas en OpenCV, como *background subtraction* para la detección en primer plano del animal o de técnicas de predicción y corrección de señales como el filtro Kalman, para la corrección y estimación de la posición del animal.

La herramienta proporcionará algunas de las métricas más habituales.

- Distancia recorrida.
- Velocidad media.
- Tiempo de estancia, número de entradas y orden de entrada por zona de interés.
- Tiempo en inmovilidad.
- Mapa del camino recorrido.
- Mapa de calor del camino recorrido.

Finalmente, se seguirán las siguientes directrices durante el desarrollo, en lo posible.

- Mantener la portabilidad entre sistemas operativos.
- Conseguir la máxima compatibilidad con fuentes de video.
- Facilidad de uso y eficiencia.

El desarrollo se llevará a cabo en Python para garantizar en lo posible su portabilidad entre sistemas operativos.

Este documento detalla en los siguientes apartados el proceso de estudio, análisis, desarrollo, pruebas y conclusión llevado a cabo.

1. **Estado del arte:** documenta el proceso investigador y sus necesidades. Se estudian las técnicas usadas, las herramientas disponibles en el mercado y sus características. Por último, se reflexiona sobre las necesidades de los investigadores y las oportunidades presentan.
2. **Análisis y diseño:** estudia los requisitos, alternativas técnicas para el diseño y solución elegida.
3. **Implementación:** documenta el desarrollo y los elementos usados como estructuras de datos, módulos, herramientas y librerías externas.
4. **Pruebas:** documenta las pruebas realizadas.

5. **Conclusión y trabajo futuro:** comentario sobre el proyecto, cobertura de objetivos y posibles mejoras.
6. **Apéndices:** Esquemas ampliados, condiciones técnicas/económicas de despliegue e información adicional.
7. **Glosario de términos:** Relación de términos usados.
8. **Manual de usuario:** Manual de uso de la aplicación.

2 Estado del arte

Estudiaremos ahora el panorama actual de la investigación, sus características y flujo de trabajo en el ámbito de la neurociencia.

Continuaremos con un repaso a las soluciones disponibles en el mercado, sus características, ventajas y desventajas.

Finalmente expondremos las demandas del personal investigador, sus necesidades particulares y oportunidades para nuevas soluciones.

2.1 Investigación y experimentos

Como comentábamos en el capítulo anterior, la neurociencia es la disciplina científica que estudia el desarrollo del sistema nervioso, su estructura, funciones, bases moleculares y patologías entre otros.

Los investigadores de esta disciplina o neurocientíficos tienen como objetivo entender las bases del aprendizaje, la memoria, el comportamiento, la percepción sensorial y la consciencia.

Es un campo de estudio amplio y complejo, pero en nuestro caso nos centraremos en aquellos que estudian la memoria, concretamente la memoria espacial y la memoria a corto/largo plazo.

Los neurocientíficos se valen de los experimentos (entre otras cosas) para su estudio. Estos experimentos, se diseñan específicamente para medir aspectos concretos de la memoria y se llevan a cabo en ratas, ratones y peces en espacios controlados, aislados y por periodos de tiempo limitado.

Cuando hablamos de espacios controlados y aislados, nos referimos a aparatos con formas especialmente diseñadas para provocar una decisión y aislar al animal del entorno. Existen múltiples variantes y tamaños de estos aparatos y depende del neurocientífico cual usar.

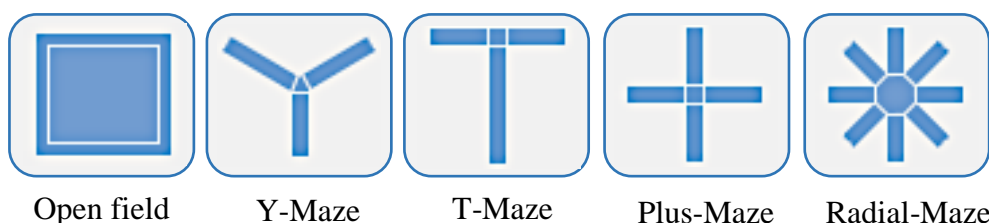


Figura 1. Tipos de aparato

Los aparatos más habituales como se observa en la Figura 1 se clasifican en cinco tipos.

- **Openfield:** Llamado así por constar de una base rectangular y cuatro paredes donde el animal tiene visión completa del espacio siempre.

- **Y-Maze:** Conocido por su forma en I griega o Y, consta de tres brazos con una zona central en forma de triángulo equilátero.
- **T-Maze:** Conocido por su forma en T, consta también de tres brazos, siendo el central de mayor longitud habitualmente y tiene una base central cuadrada.
- **Plus-Maze:** Llamado así por su forma en signo de más, tiene cuatro brazos de igual tamaño y una base central cuadrada. En ocasiones dos de sus brazos pueden carecer de paredes.
- **Radial-Maze:** Consta de una base poligonal de entre 5 y 24 lados habitualmente y brazos de igual tamaño.

Existen más tipos de aparatos como el laberinto de Morris o *Water-Maze* (usado también en el estudio de la memoria espacial) y múltiples variantes de los anteriores.

Estos aparatos suelen estar fabricados en plástico y/o madera, con diferentes tamaños y colores. Suelen colocarse sobre una mesa o elevados de algún modo, pero no es estrictamente necesario.

Para garantizar que el experimento se lleva a cabo sin elementos ajenos al mismo, el aparato desinfectado se coloca en el interior de una habitación con elementos controlados, como tipo, número de luces y ruido ambiental mínimo.

Dado que estos experimentos suelen constar de varios grupos de animales y de varias pruebas por animal, suelen realizarse en varios días o semanas, por lo que se intenta mantener todo inalterable entre sesiones.

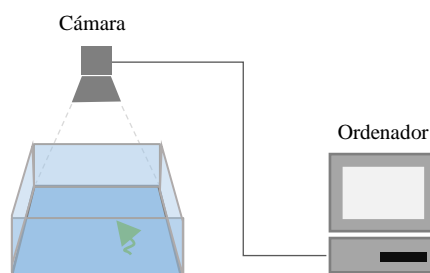


Figura 2. Instalación habitual en experimentos de memoria

Hemos definido el aparato, los animales, y el entorno, pero son necesarias otros elementos como se puede ver en la Figura 2.

Es necesario una cámara de video analógica o digital colocada sobre el aparato a una distancia lo más pequeña posible, pero suficiente para captar el aparato completo en la imagen.

Un ordenador conectado a la cámara (usando el cableado y si fuera necesario usando una tarjeta digitalizadora) que registra en video el experimento.

Estos videos, es necesario analizarlos con una aplicación informática que usando técnicas de visión artificial extrae las métricas necesarias. Según la aplicación usada o a decisión del neurocientífico, puede que los videos se registren y analicen en un solo paso o se registren y analicen en la misma o diferentes aplicaciones.

Con estas métricas, el neurocientífico busca patrones y/o características que validen sus hipótesis.

2.1.1 Flujo de trabajo

El flujo de trabajo en estos experimentos, depende de las características propias del mismo y del criterio del neurocientífico.

Habitualmente el experimento se compone de varios grupos de pruebas o *Trial* que se realizan a varios grupos de animales.

En el caso de ratas o ratones se suele realizar la misma prueba en todos ellos (uno detrás de otro) y después se cambia a la siguiente prueba. Sin embargo, en el caso de los peces es habitual realizar todas las pruebas en un animal y después continuar con el siguiente animal.

En cada prueba se realizan los siguientes pasos:

1. El neurocientífico inicia en la aplicación informática la grabación y/o análisis.
2. Retira al animal de su caja y lo coloca dentro de del aparato.
3. Espera el tiempo que se requiera para que se complete el intento según tenga establecido.
4. Retira al animal del aparato y lo devuelve a su caja.
5. Detiene la grabación y/o análisis en la aplicación informática.

Una vez el neurocientífico ha realizado todas las pruebas que necesitaba procede a analizar las imágenes si fuera necesario.

Recordemos, que no todas las aplicaciones registran y analizan las imágenes a la vez y que el neurocientífico puede decidir no hacerlo simultáneamente (por ejemplo, por no disponer del ordenador adecuado).

Cuando todas las imágenes han sido analizadas por la aplicación, las métricas suelen exportarse a otro formato, (por ejemplo, Microsoft Excel), para estudiarlo.

2.1.2 Registro de imágenes y análisis

En este capítulo hemos aprendido que el registro en video de cada prueba que se realiza a cada animal en los experimentos es la base de donde obtener las métricas que el neurocientífico necesita.

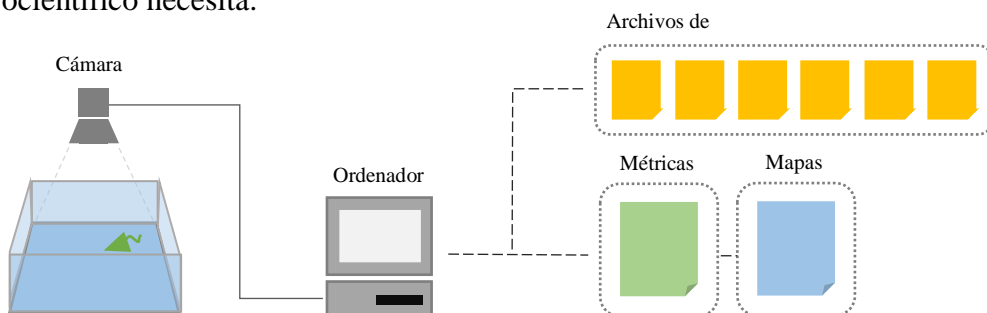


Figura 3. Sistema de registro de imágenes y análisis.

Este proceso de análisis, como muestra la Figura 3, requiere por normal general una serie de pasos previos en la aplicación, para definir una serie de elementos.

1. Definir tipo y número de pruebas.
2. Definir tipo y número de animales.
3. Definir que la(s) fuente(s) de imagen a usar Cámara o archivo(s) de video(s).
4. Modelado digital del aparato (Forma, tamaño, etc.) Ver Figura 4.
5. Definir las zonas de interés del aparato y las métricas asociadas. (Ejemplo: Brazos del aparato, número de entradas y orden de la primera entrada).
6. Ajustes del sistema de seguimiento.
7. Ajustes de la duración de la prueba (Ejemplo: Parada automática después de 10 minutos desde el comienzo).

Una vez se han definido todos los elementos anteriores se procede al registro de la prueba como se ha indicado en el apartado anterior (Flujo de trabajo).

Al finalizar este proceso, si no se ha realizado a la vez que el registro, se procederá al análisis de las imágenes y extracción de métricas.

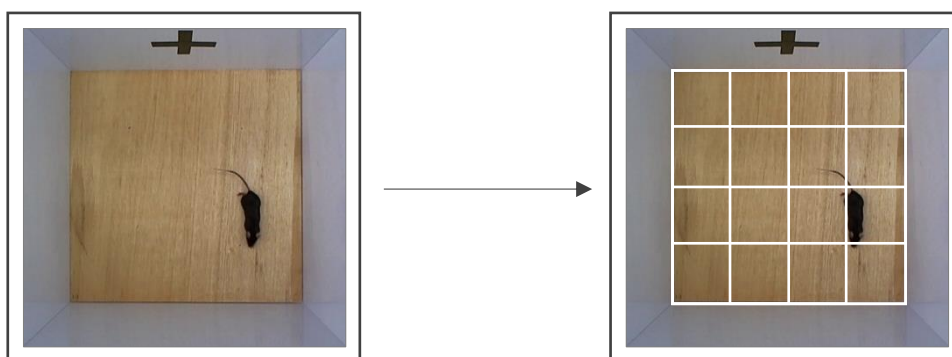


Figura 4. Modelado digital del aparato.

Las métricas más habituales comprenden las siguiente.

- Tiempo en zona central del aparato
- Tiempo en zona cercana a la pared (tigmotaxia)
- Tiempo en las esquinas
- Distancia recorrida total
- Distancia recorrida en zona central
- Distancia recorrida en zona cercana a la pared (tigmotaxia)
- Porcentaje de tiempo en la zona central
- Número de veces que entró en la zona central
- Número de veces que cruzo la zona central
- Tiempo en cada brazo del aparato
- Número de veces que entró en cada brazo del aparato
- Secuencia de entrada en los brazos del aparato
- Número total de veces paralizado
- Tiempo total paralizado

Es habitual proporcionar mapas de recorrido y mapas de calor como los que muestra la Figura 5, que evidencien las zonas donde ha permanecido más tiempo y el camino recorrido.

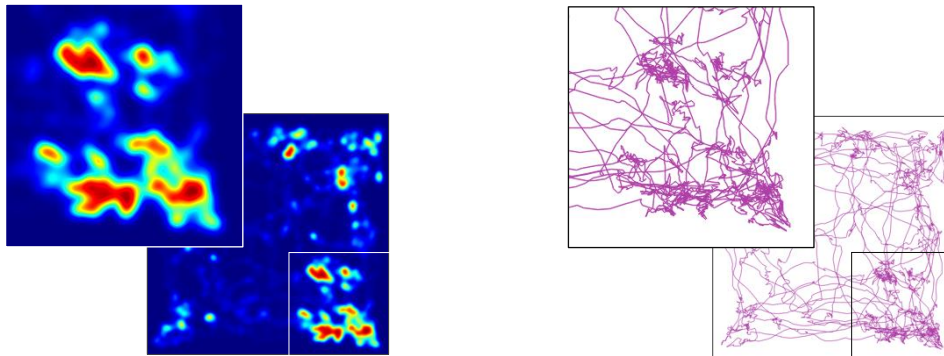


Figura 5. Izquierda: Mapa de calor, Derecha: Mapa de recorrido.

2.1.3 Técnicas para seguimiento de objetos

Hemos hablado durante todo el capítulo del término “Análisis”, sin embargo, no se han dado detalles de cómo se realiza dicho proceso.

En este apartado nos adentramos en qué procesos son necesarios para realizar el análisis de las imágenes de un experimento de este tipo y de que **técnicas de visión artificial** se usan para extraer la información que necesitamos.

Fundamentalmente, el análisis de las imágenes requiere seguir al animal para localizar su posición en cada imagen o fotograma que compone el video como vemos en la Figura 6.



Figura 6. Resultado del seguimiento y localización del animal.

En base a la posición en cada fotograma se calculan todas las métricas necesarias, como la distancia, velocidad media, mapas, etc. Otras métricas necesitan de una

combinación de la posición y del modelo digital del aparato y las zonas de interés que el neurocientífico previamente ha definido en la aplicación.

El seguimiento o *Tracking* requiere de técnicas de visión artificial que permiten distinguir los objetos en primer plano (como el animal) del resto de la imagen.

Generalmente, requiere tres pasos bien diferenciados, Mostrados en la **¡Error! No se encuentra el origen de la referencia.**

1. **Detección de primer plano:** Descartar el fondo de la imagen, quedando solo el/los objeto(s) en primer plano (Ejemplo: el animal).
2. **Segmentación de imagen:** División de la imagen en varias partes que representan los objetos, es decir, su contorno.
3. **Localización del centro de masas:** Calculo del punto central para el objeto. (Ejemplo: punto central del cuerpo del animal).

El conjunto de técnicas para la detección de primer plano del objeto se denomina, *background subtraction*.

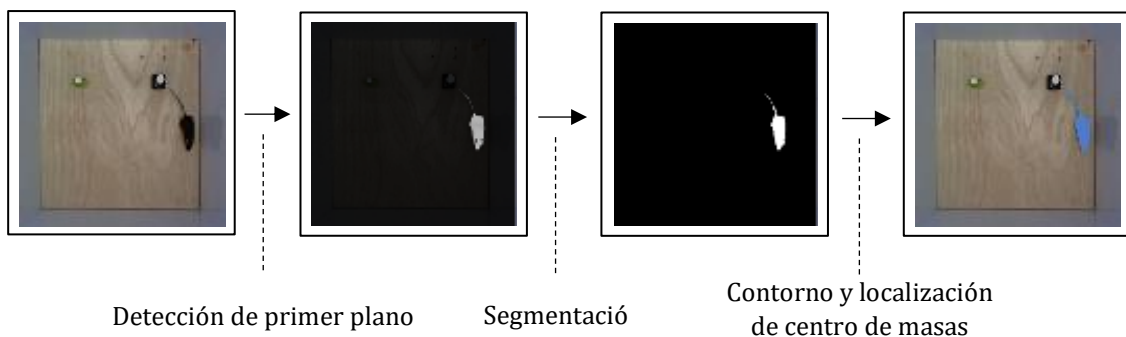


Figura 7. Pasos para el seguimiento de objetos.

Dentro de la detección en primer plano existen varias **técnicas para modelar el fondo y detectar los objetos**. Veamos en detalle las más habituales.

- Diferencia entre imágenes

Es el método más simple, como se ve en la Figura 8. Consiste en obtener la diferencia entre píxeles de una imagen sin el objeto (fondo) y cualquier otra con el objeto. Aquellos píxeles con un valor residual mayor que cierto *Threshold* o valor límite se considera que no pertenecen al fondo y por tanto son parte de alguno(s) objeto(s) en primer plano.

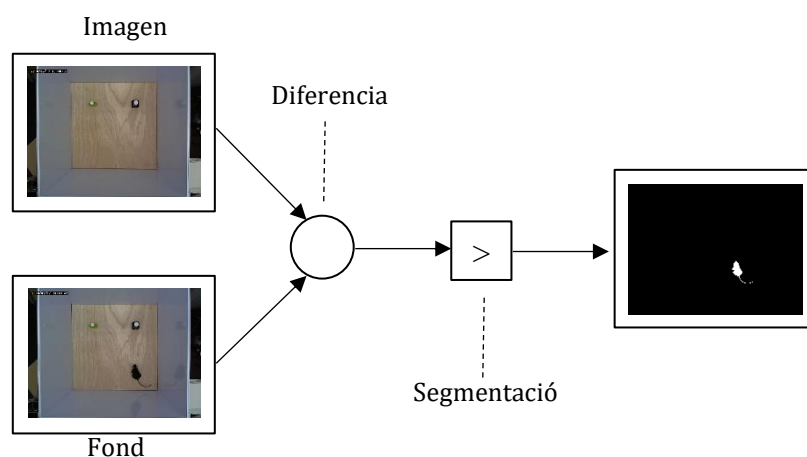


Figura 8. Detección de primer plano por diferenciación entre imágenes.

Esta técnica es aplicable cuando el fondo es estático y se dispone de al menos una imagen del fondo sin ningún objeto en primer plano.

- Filtro de mediana temporal

Método que estima el fondo de la imagen con la mediana de los píxeles de un número determinado de imágenes anteriores al fotograma actual. El proceso se ilustra en la Figura 9. Requiere, por tanto, de un periodo de entrenamiento equivalente a el número de imágenes necesario para el cálculo de la mediana y obtener el modelo para el fondo de la imagen.

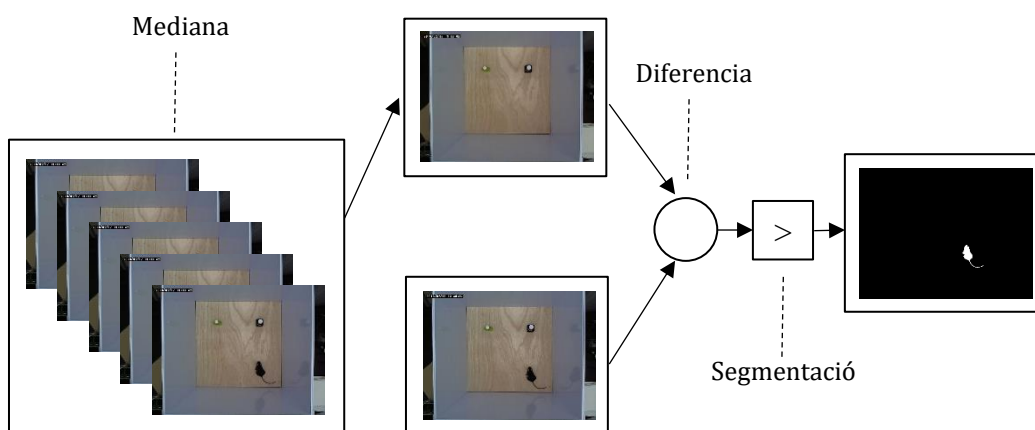


Figura 9. Detección de primer plano por filtro de mediana temporal.

- Running Gaussian average

Este método modela el fondo usando la probabilidad gaussiana por píxel caracterizado por la media y la varianza.

- Gaussian Mixture Model

Aprovecha la idea del método anterior, pero considera que cada píxel puede tener diferentes estados que modela mediante funciones gaussianas.

La **segmentación de imágenes** intenta simplificar las imágenes en grupos de píxeles o *super-pixels* usando la similitud entre píxeles adyacentes.

Existen multitud de métodos, pero las bases de estos algoritmos suelen usar una de estas dos técnicas:

- División de la imagen basándose en los cambios grandes del nivel de gris entre píxeles, lo que marca las discontinuidades, es decir, los bordes de los grupos de píxeles.
- Siguiendo el método contrario, es decir, buscando píxeles con un nivel de gris similar se van agrupando en grupos que forman las distintas partes de la imagen.

Finalmente, para la **localización del contorno y del centro de masas** se usa el concepto matemático de momento, determinado por el promedio ponderado de las intensidades de los píxeles. Esto permite calcular la forma específica del grupo(s) de píxeles o contorno detectados durante la segmentación de la imagen, lo que a su vez nos permite conocer su área y por tanto su centro.

Localizado el centro de masas podemos usar sus coordenadas para realizar los cálculos que necesitemos.

2.1.4 Técnicas de predicción y corrección de posición

Probablemente, la técnica más usada es el filtro Kalman, inventado por Rudolf E. Kalman en el año 1960. Es común su uso en corrección de señales, posicionamiento vía GPS, o la guía y navegación de vehículos entre otros.

Es resistente al ruido y permite predecir la posición en base a las posiciones anteriores en el tiempo. Factores que suelen formar parte de estos cálculos son velocidad, aceleración y tiempo.

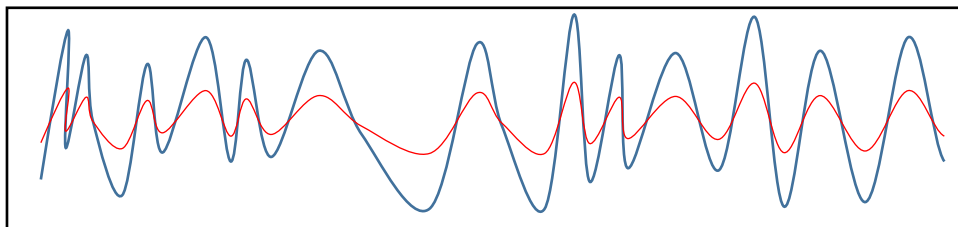


Figura 10. Resultado al aplicar un filtro Kalman a una señal. Azul: Señal, Rojo: Señal corregida.

Combinado con las técnicas de visión artificial para seguimiento de objetos, nos permite.

- Corregir la localización del centro de masas, eliminando los pequeños cambios debidos a las técnicas de seguimiento.
- Suavizar el camino recorrido para obtener métricas más fieles a la realidad. Por ejemplo, la distancia recorrida.

- Posibilidad de predecir la localización del objeto. Por ejemplo, si la detección fallara o no fuera posible durante periodos muy cortos de tiempo, la localización podría ser una estimación.

El filtro kalman suele describirse en dos pasos.

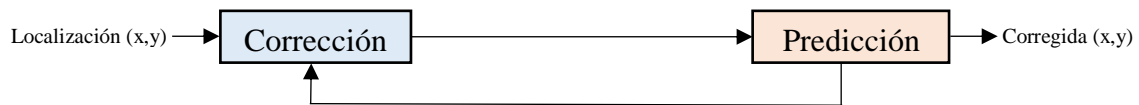
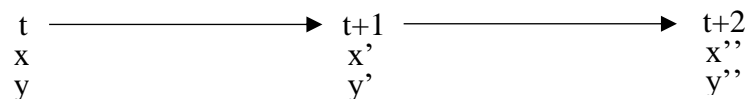


Figura 11. Algoritmo recursivo del filtro Kalman.

Vamos a ver en detalle como usamos el filtro Kalman en el seguimiento de objetos para modelar un sistema dinámico que represente la realidad.

Como resultado del seguimiento del objeto, obtenemos una medida, la posición del mismo, que viene definida por su posición en el eje “x” y su posición en el eje “y” en cada fotograma, es decir, en cada momento en el tiempo “t”.



Estas medidas conforman nuestro vector de estado “x”.

$$x = \begin{bmatrix} x \\ y \\ x' \\ y' \end{bmatrix}$$

Este vector de estado puede ser inicializado si conocemos previamente alguno de sus valores (Ejemplo: Conocemos la posición inicial del objeto), si no se proporciona un valor inicial, su valor será cero.

Lo siguiente a definir es la matriz de transición de estados “A”. Con ella definimos la relación entre las medidas y el vector de estado en el momento “t”.

$$A = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ x' \\ y' \end{bmatrix}_{t+1} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ x' \\ y' \end{bmatrix}_t$$

debemos especificar la matriz de co-varianza del ruido en las mediciones que se produce por efectos externos en el propio sistema de detección y dice al filtro como pasar de un estado al siguiente. (Ejemplo: Alteraciones en la luz produce cambios entre fotogramas que provocan que la detección sea imprecisa temporalmente y contenga errores o ruido.)

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * 0.5$$

En este ejemplo asignamos el valor 0.5 a la matriz identidad, pero dicho valor debe ser especificado intuitivamente ya que no hay un método específico que lo determine.

Finalmente, falta por definir la matriz de mediciones “H” que especifica qué es medido y como se relaciona con el vector de estado “x”.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Dado que el valor de la posición es medido de manera directa lo indicamos colocando un “1” para la coordenada “x” y la coordenada “y”.

El filtro Kalman admite otras dos matrices, “B” y “R”. La matriz de control de entrada y la matriz de ruido del sensor, ambas no son aplicables en el caso que nos ocupa ya que no consideramos que la cámara tenga un error en la captación de imágenes ni este influida por factores externos como movimientos, giros, golpes etc, ya que si existieran invalidarían el experimento al completo.

Definidas las matrices del filtro estudiamos ahora en más detalle los pasos “Predicción” y “Corrección”.

- Predicción: Este paso intenta predecir el estado de todo el sistema en el tiempo “t+1”. En nuestro caso, si la detección fallara durante unas milésimas de segundo podríamos predecir la posición del objeto.

$$X_{t+1} = A * X_t$$

$$P = A * P * A^T + Q$$

Se calcula el nuevo estado con la matriz de transición de estados una vez se ha inicializado el vector de estado con alguna medida. Y se calcula la matriz “P” que identifica la covarianza del error asociada a la estimación a priori.

Y con esto ya se ha realizado una predicción de la localización.

- Corrección: En este paso el filtro en base a las nuevas mediciones que se le proporcionan comprueba cómo se compara la predicción con la nueva medición. Para ello se calcula la matriz de la ganancia de Kalman “K”, cuyos valores decrecen o aumentan según la predicción se acerque o se aleje de la medida real.

$$K = P * H^T * (H * P * H^T + R)$$

Esta matriz “K” se usa para actualizar el estado del sistema, y determinar una nueva matriz de covarianza del error “P” para un nuevo paso de predicción.

$$x = x + K * y)$$

$$P = P - (K * H * P)$$

Al igual que nos sirve para predecir una localización incierta, también realiza un suavizado del camino al procesar el ruido que existe en la detección mediante el valor que indiquemos en la matriz “Q”.

2.2 Soluciones disponibles y características

Están disponibles soluciones tanto comerciales como de libre uso con distinto grado de funcionalidad y características. Las principales soluciones disponibles pueden consultarse en el Tabla 1.

Nombre	Licencia	Versión	Sistema Operativo	Año
Any-Maze	Comercial	5.2	Windows	2017
Biobserve Viewer	Comercial	2017	Windows	2017
EthoVision XT	Comercial	14	Windows	2018
IDTracker	Comercial / Open source	2.1	Windows	2014
OpenControl	Open source	1.2	Windows	2008
SwissTrack	Adaptative Public License	4.0.1	Windows / MacOS / Linux	2008
ToxTrack	Creative Commons	2.84	Windows	2018
VideoMot 2	Comercial	5.76	Windows	2015

* Algunos datos pueden no estar actualizados o ser incorrectos al no estar disponibles al público.

Tabla 1. Soluciones software disponibles.

Todas estas soluciones son capaces de realizar un seguimiento adecuado de uno o varios animales y generan como mínimo unas estadísticas básicas, por lo que dichas características no vamos a incluirlas en este estudio.

	Any-Maze	Biobserve Viewer	Ethovision XT	IDTracker	OpenControl	SwissTrack	ToxTrack	VideoMot2
Requiere compilación o compilación de dependencias						√		
Comunicación con sensores externos	√	√	√			√		√
Fuente: Archivos de video	√	√	√	√		√	√	√
Fuente: Dispositivos de video (Cámaras)	√	√	√		√	√	√	√
Formatos de video: Soporte completo								
Formatos de video: Soporte limitado	√	√	√	√		√	√	√
Carga masiva de archivos de video							√	
Grabación de video	√	√	√			√		√
Soporta cámaras simultáneas	√	√	√			√		√
Soporte para múltiples animales por cámara o archivo	√	√	√	√		√	√	√
Soporte para identificación de animales	√	√	√	√		√	√	√
Gestión del experimento con archivos de video							√	
Gestión del experimento con dispositivos de video	√	√	√			√		√
Detección de comportamiento básico	√	√	√	√	√	√	√	√
Detección de comportamiento avanzado	√	√	√					√
Visualización de mapas de calor y/o caminos	√	√	√	√		√	√	√

* Algunos datos pueden no estar actualizados o ser incorrectos al no estar disponibles al público.

Tabla 2. Características de las soluciones.

Sin embargo, para poder compararlas de forma adecuada, es necesario examinar características más específicas y que ayuden a diferenciarlas. Estas características se muestran en la Tabla 2.

Como norma general, las soluciones comerciales cubren más características que las de libre uso. En concreto, podemos destacar una característica que no tiene soporte.

- Gestión del experimento con archivos de video

En general, todas las aplicaciones necesitan de la carga manual (uno a uno) de cada archivo de video y son analizados a la velocidad real del video con la consiguiente pérdida de tiempo.

No supone un problema si el análisis se realiza durante el registro de las imágenes. Sin embargo, supone un impedimento bastante importante en caso contrario.

ToxTrack si soporta múltiples archivos de video, aunque no permite asignar animal y/o prueba por lo que el neurocientífico necesitara reorganizar todos los datos manualmente después del análisis.

Destaca también, una característica que ninguna solución de libre uso ofrece, pero que existen en todas las soluciones comerciales:

- Detección de comportamiento avanzado

Capacidad para detectar movimientos específicos de animales, por ejemplo: retroceso, giros de cabeza, acicalamiento, etc.

Finalmente, aquellas características que lejos de ser positivas implican o pueden implicar algún problema:

- Requiere compilación o compilación de dependencias

En concreto, SwissTrack necesita ser compilada por el neurocientífico para su uso en el sistema operativo MacOS.

Ahora que conocemos las soluciones disponibles y sus características, podemos evaluar un elemento fundamental, el coste económico. Los detalles se muestran en la Tabla 3.

- Coste:

Coste de la solución para obtener la funcionalidad equivalente a las características mencionadas previamente. En este coste, se incluyen módulos adicionales necesarios y/o actualizaciones. No se incluyen el coste del aparato, cámaras, cableado e instalación.

Nombre	Coste
Any-Maze	5.955,00\$ + Módulos + Actualizaciones
Biobserve Viewer	Alto (No disponible al público)
EthoVision XT	5.675,00\$ + Módulos + Actualizaciones
IDTracker	Sin coste para fines no comerciales
OpenControl	Sin coste para fines no comerciales
SwissTrack	Sin coste para fines no comerciales
ToxTrack	Sin coste para fines no comerciales
VideoMot 2	Alto (No disponible al público)

Tabla 3. Coste de las soluciones.

En general, estas soluciones tienen un coste elevado y requieren actualizaciones de pago para obtener nuevas funciones y soporte. Tienen la ventaja de tener el elenco de características más completo y son aplicaciones probadas y reconocidas.

Las opciones sin coste, aunque existen, tienen ciertas limitaciones, siendo ToxTrack posiblemente la más completa.

2.3 Necesidades

Después de repasar el estado del arte a lo largo de este capítulo, se ha recogido toda la información posible de primera mano de neurocientíficos para evaluar sus necesidades.

Principalmente, existen ocasiones en las que por falta de presupuesto en el laboratorio o de financiación del neurocientífico no tienen acceso a las aplicaciones comerciales, lo que les supone una desventaja competitiva muy grande.

Por ello, están abiertos a nuevas aplicaciones no comerciales que ofrezcan algunas de las características y métricas básicas. En concreto, desean tener la carga masiva de archivos de video y poder analizarlos sin tener que reproducirlos a velocidad estándar.

Finalmente, en algún caso les gustaría poder usar la aplicación en sistemas operativos alternativos a Microsoft Windows.

3 Análisis y diseño

En este capítulo expondremos los medios necesarios para llevar a cabo el desarrollo de este proyecto junto a los requisitos identificados.

Como parte del análisis definimos la arquitectura de la aplicación junto a las estructuras de datos necesarias, lenguajes y dependencias externas. Finalmente, se ha diseñado una interfaz de usuario siguiendo las directrices marcadas en los requisitos.

3.1 Medios

Para el desarrollo del proyecto se requieren un conjunto de herramientas software y hardware. El hardware está compuesto por los ordenadores necesarios para el desarrollo y la realización de las pruebas.

1. Ordenador (desarrollo y pruebas):
 - **CPU:** Intel Core i7 6770HQ 2.6 Ghz / 3.5 Ghz
 - **Memoria:** 16 GB RAM DDR4 2133 Mhz
 - **Disco:** 500 GB SSD
 - **Sistema:** Windows 10 64bit

2. Ordenador (pruebas):
 - **CPU:** Intel Core 2 Duo 2.26 Ghz
 - **Memoria:** 4 GB RAM DDR3 1333Mhz
 - **Disco:** 320 GB HDD
 - **Sistema:** MacOS High Sierra
 - **Cámara:** iSight 640x480 30 Fotogramas/segundo

3. Ordenador (pruebas):
 - **CPU:** Intel Core i5 4590 3.30 Ghz / 3.70 Ghz
 - **Memoria:** 8 GB RAM DDR3 1600 Mhz
 - **Disco:** 128 GB SSD
 - **Sistema:** Linux Ubuntu 18.04 LTS 64 bit

El software necesario se divide en sistemas operativos, intérpretes del lenguaje, librerías y entorno de desarrollo. Algunas de estas librerías y/o intérpretes pueden estar incluidas en el sistema operativo por defecto.

1. **Sistemas operativos:**
 - Windows 10 64 bit
 - MacOS High Sierra 10.13.1 64 bit
 - Linux Ubuntu 18.04 LTS 64 bit

2. Intérpretes y lenguajes:

- Python 3.7 64 bit

3. Librerías:

- Tkinter 8.6 incluida en Python 3.7
- Numpy 1.16.3 para Python 3.7
- OpenCV 4.1.0 para Python 3.7
- FFMPEG 4.1.3 (Librerías dinámicas/compartidas)
- XLSXWriter 1.1.8 para Python 3.7

4. Entorno de desarrollo:

- Notepad++ 7.6.6 (64 bit)

Para realizar las pruebas, se ha usado el siguiente material, cedido por los neurocientíficos que han colaborado.

1. Archivos de video:

- Openfield (Ratón): 24 horas de video divididas en 120 archivos. Cada archivo tiene una resolución de 640x480 pixel y 25 Fotogramas/segundo.

3.2 Requisitos

Teniendo en cuenta el estudio realizado en el capítulo anterior, podemos identificar los requisitos para la solución y la prioridad dentro del proyecto que se le ha asignado.

Comenzamos definiendo los requisitos funcionales en la Tabla 4. Estos requisitos representan el flujo de trabajo marcado por los neurocientíficos y se asemeja a las que proporcionan a algunas de las aplicaciones estudiadas en el capítulo anterior.

	Prioridad	Descripción
1	Media	Definir grupos y número de animales
2	Media	Definir grupos y número de pruebas.
3	Media	Añadir archivos de video
4	Media	Detección automática de dispositivos de video (Cámaras)
5	Media	Navegador de grupos/pruebas
6	Alta	Asignar manualmente archivos de video o dispositivos por prueba.
7	Alta	Asignar dispositivos de video a todas las pruebas automáticamente.
8	Baja	Selección de vista de las pruebas (Por animal/Por Intento o <i>Trial</i>)
9	Alta	Selección del tipo de aparato
10	Alta	Selección de las características del aparato (Divisiones y/o Brazos)

11	Media	Añadir áreas adicionales.
12	Alta	Reproducción de archivos de video (Reproductor)
13	Media	Selección del archivo de video a reproducir
14	Alta	Reproducción de video del dispositivo (Cámara) (Monitor)
15	Media	Selección del dispositivo (Cámara) a visualizar
16	Media	Creación automática de zonas
17	Media	Creación manual de zonas
19	Media	Selección de múltiples áreas por zona
20	Media	Selección de métricas por zona
21	Alta	Visualización de áreas (Todas/Solo aparato/Solo áreas)
22	Alta	Selección del algoritmo de seguimiento
23	Alta	Ajuste del valor límite o <i>threshold</i> (Manual: Valor/ Automático)
24	Alta	Selección del área a seguir (Área mínima/ Área máxima)
25	Alta	Ajuste de la condición de parada por tiempo total
26	Alta	Ajuste de condición de parada por número de entradas
27	Alta	Ajuste del tiempo mínimo para considerar inmóvil al objeto
28	Alta	Análisis de todos las pruebas con archivos de video asignados
29	Media	Selección y análisis de una prueba con archivo de video asignado
30	Media	Selección de una prueba con dispositivo (Cámara) asignado
31	Alta	Inicio / Parada de la grabación de video del dispositivo (Cámara)
32	Alta	Análisis de las imágenes del dispositivo y generación de las métricas
33	Media	Navegador de resultados
34	Alta	Selección del resultado a visualizar (Métricas / Mapas)
35	Baja	Borrado del resultado seleccionado
36	Media	Exportar resultados a archivo externo
37	Media	Carga de un experimento anterior

Tabla 4. Requisitos funcionales.

Pasamos a definir los requisitos no funcionales en la Tabla 5. Es decir, aquellos requisitos que indican características del entorno hardware y software en el que debe funcionar, tiempos de respuesta, etc. que afectan directamente al diseño y la implementación.

	Prioridad	Descripción
1	Alta	Ejecución en ordenador personal de capacidad/velocidad media.
2	Media	Multiplataforma (En la medida de lo posible)
3	Alta	Facilidad de uso, configuración y puesta en marcha.
4	Alta	Distribución/Uso libre y sin coste
5	Alta	Modular y de fácil mantenimiento

Tabla 5. Requisitos no funcionales.

3.3 Arquitectura

La arquitectura de la aplicación se basará en el reconocido y clásico patrón Modelo-vista-Controlador o MVC. Su estructura se muestra en la

Figura 12. Nos permite modularidad en el código y se ajusta perfectamente al desarrollo de una aplicación de escritorio.

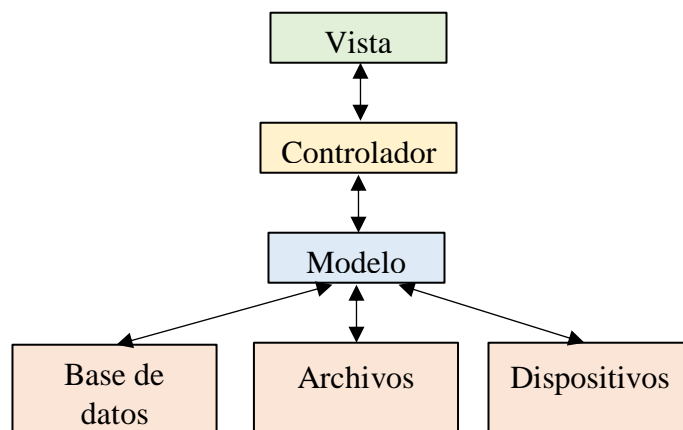


Figura 12. Patrón Modelo-Vista-Controlador o MVC.

En el patrón de diseño MVC tanto la Vista como el Modelo son entidades que desconocen por completo la existencia de las demás. Tan solo exponen una interfaz pública a determinadas funciones. Si seguimos con rigor este patrón ambas serán independientes.

Además, proporciona ventajas como intercambiar estas entidades por otras que proporcionen las misma o mayor funcionalidad manteniendo la interfaz, su funcionamiento interno, etc, sin afectar a las demás. También, permite cambiar la implementación interna sin afectar a las demás entidades.

Pasamos a detallar cada uno de estos elementos de la arquitectura, sus componentes y la funcionalidad que prestara al conjunto de la aplicación.

Modelo

El modelo gestiona los datos bajo las órdenes del controlador. Su estructura se muestra en la Figura 13. En concreto, realiza las siguientes funciones.

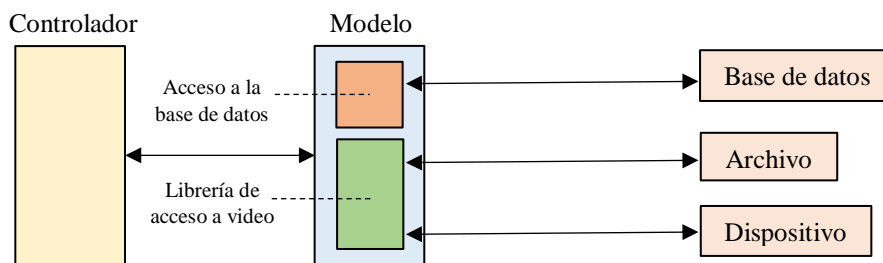


Figura 13. Estructura del modelo.

- Gestión de la base de datos que contiene la configuración, pruebas, resultados, etc, del experimento.
- Proporciona acceso nativo a los datos mediante una capa de traducción de los resultados de la base de datos a estructuras de datos nativas al lenguaje. (Ejemplo: Consulta de pruebas del experimento (SQL) se transforma a una lista de elementos de Python).

- Gestiona el acceso a los archivos de video y los hace visibles al controlador. Incluye una librería propia que interacciona con librerías externas para decodificación de video, proporcionando traducción de cada fotograma al formato nativo del lenguaje. (Ejemplo: Lee un fotograma del video y lo transforma en una matriz en formato Numpy de Python)
- Gestiona el acceso a los dispositivos de video, los encuentra y accede a ellos capturando cada fotograma. La librería propia usada para archivos de video soporta el acceso a dispositivos.

Vista

La vista está compuesta por la interfaz de la aplicación, es decir aquellos elementos que muestran los datos en pantalla, los elementos con los que el usuario interactúa directa o indirectamente, controla el aspecto de la aplicación y redirige todas las solicitudes al controlador.

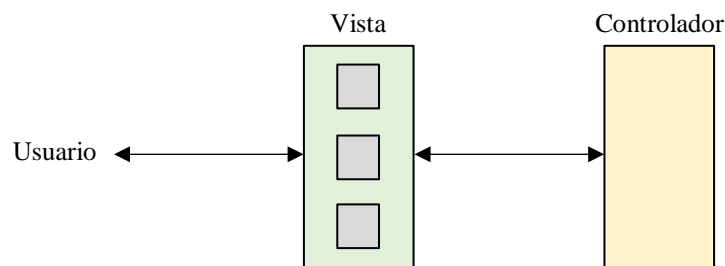


Figura 14. Estructura de la vista.

Controlador

El controlador es la única entidad en este patrón que conoce la existencia de la Vista y del Modelo. Contiene toda la lógica de la aplicación y hace uso del modelo para el acceso a los datos y de la vista para mostrarlos. Se muestra su estructura en la Figura 15.

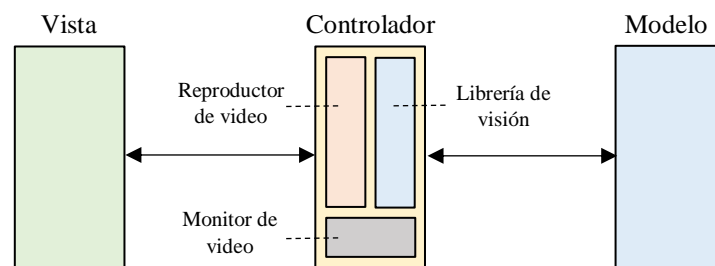


Figura 15. Estructura de del controlador.

Además, recibe las peticiones del usuario a través de la vista y redirige dichas peticiones al modelo cuando es necesario el acceso a los datos.

Incluye un reproductor de video, un monitor de visualización de video para los dispositivos, una librería basada en visión artificial para el seguimiento de objetos y toda la lógica necesaria para gestionar la aplicación y sus transacciones.

3.4 Estructuras de datos

En base a la arquitectura y la funcionalidad prevista, se han identificado las estructuras de datos necesarias.

- Teniendo en cuenta la arquitectura de la aplicación parece lógico el uso de programación orientada a objetos pasando los objetos a formar parte de las estructuras de datos ya que encapsulan datos y funcionalidad.
- Se hará un uso intensivo de listas, vectores y matrices como base para definir estructuras de datos más complejas. (Ejemplo: Definimos un fotograma como una matriz. Representando cada uno de los valores (Rojo, Verde y Azul) de cada pixel)
- Se crean estructuras de datos basadas en objetos de elementos del interfaz basándonos en estructuras más básicas que ofrece la librería estándar. (Ejemplo: Navegador de grupos/pruebas, usando elementos básicos como etiquetas, cuadros o *Frames*, barras de desplazamiento vertical, etc.).
- Se usa una base de datos para almacenar los datos relativos al experimento en disco.
- Las imágenes generadas como mapas de calor y recorrido se guardan en disco en archivos individuales en formato PNG.

3.5 Base de datos

Es necesario usar una base de datos para mantener todos los datos relativos a un experimento (con excepción de archivos de video e imágenes como mapas de calor o recorrido), en disco.

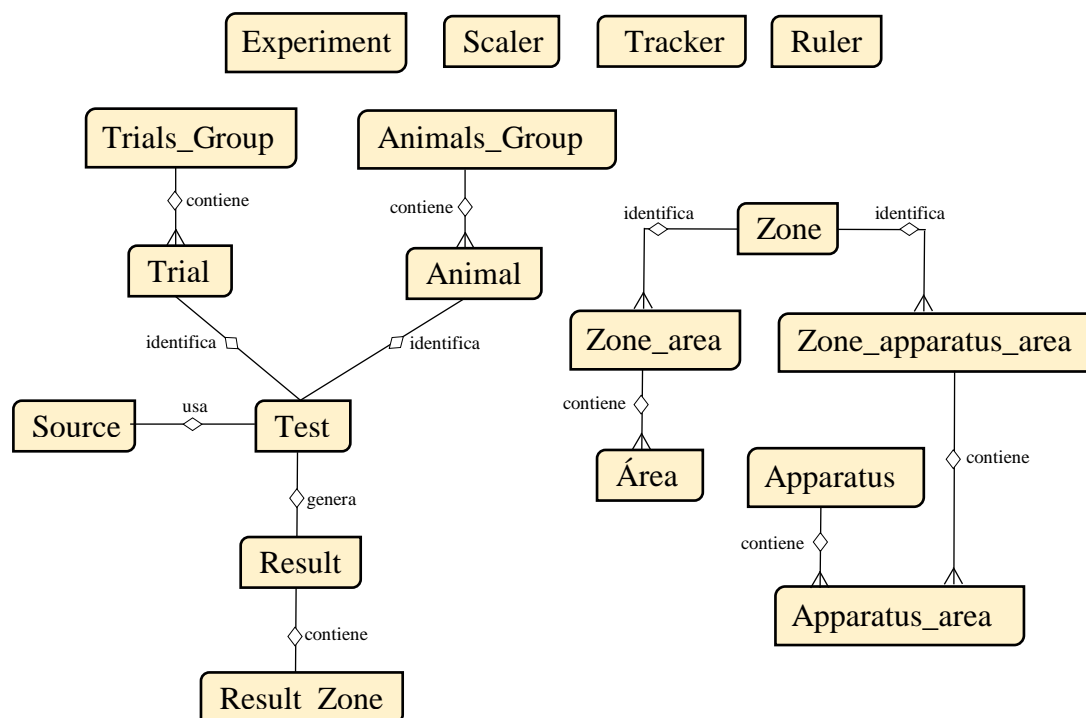


Figura 16. Estructura de la base de datos.

Las tablas y relaciones que definen la base de datos se detallan en la Figura 16.

Esta estructura se replica en cada base de datos que se genera, ya que cada experimento consta de una base de datos propia. De este modo se proporciona portabilidad del experimento de un ordenador a otro y facilita la carga del experimento. Este procedimiento está explicado en el capítulo manual de usuario de este mismo documento.

A continuación, tenemos la definición de cada una de las tablas que componen la base de datos, sus campos, tipos y una breve explicación de que representa.

Experiment

Almacena los datos más básicos, nombre del experimento y descripción.

Experiment		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
name	TEXT	NOT NULL
notes	TEXT	NOT NULL

Tabla 6. Definición SQL de la tabla "Experiment".

Animals_Group y Animal

Representan los grupos de animales usados en el experimento, su tipo, color y número.

Animals_Group		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
group_color	TEXT	NOT NULL
elements	INTEGER	NOT NULL
type	TEXT	NOT NULL
color	TEXT	NOT NULL

Tabla 7. Definición SQL de la tabla "Animals Group".

Animal		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
group_id	INTEGER	FOREING KEY
public_id	INTEGER	NOT NULL

Tabla 8. Definición SQL de la tabla "Animal".

Trials_Group y Trial

Representan los grupos de pruebas usados en el experimento y su número.

Trials_Group		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
group_color	TEXT	NOT NULL
elements	INTEGER	NOT NULL
type	TEXT	NOT NULL
color	TEXT	NOT NULL

Tabla 9. Definición SQL de la tabla "Trials Group".

Trial		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
group_id	INTEGER	FOREING KEY
public_id	INTEGER	NOT NULL

Tabla 10. Definición SQL de la tabla "Trial".

Source

Representa todas las fuentes de video que se han detectado, en el caso de los dispositivos o de las que se han añadido al experimento, en el caso de archivos de video.

Source		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
path	TEXT	NOT NULL
name	TEXT	NOT NULL
extensión	TEXT	NOT NULL
hash	TEXT	NOT NULL
type	TEXT	NOT NULL

Tabla 11. Definición SQL de la tabla "Source".

Test

Un test representa una prueba o *Trial* que se realiza en un animal. Representa un elemento central para una gran parte de la base de datos.

Test		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
path	TEXT	NOT NULL
name	TEXT	NOT NULL
extensión	TEXT	NOT NULL
hash	TEXT	NOT NULL
type	TEXT	NOT NULL

Tabla 12. Definición SQL de la tabla "Test".

Result

Almacena los resultados de cada test. Las métricas básicas de tiempo total, distancia y tiempo en inmovilidad.

Result		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
test_id	INTEGER	FOREING KEY
duration	REAL	NOT NULL
distance	REAL	NOT NULL
freeze	REAL	

Tabla 13. Definición SQL de la tabla "Result".

Result_zones

Almacena las métricas de cada zona. Orden de entrada, número de entradas y tiempo que el animal ha permanecido dentro de ellas.

Result_zones		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
test_id	INTEGER	FOREING KEY
zone_id	INTEGER	FOREING KEY
order	NTEGER	
entries	INTEGER	
time	REAL	

Tabla 14. Definición SQL de la tabla "Result zones".

Apparatus y Apparatus_area

Almacenan el modelo digital del aparato y todas las zonas que lo componen.

Apparatus		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
type	TEXT	NOT NULL
center_x	INTEGER	NOT NULL
center_y	NTEGER	NOT NULL
angle	INTEGER	NOT NULL
width	INTEGER	
height	INTEGER	
sides	INTEGER	
divisions	INTEGER	
inner_radius	INTEGER	
ouer_radius	INTEGER	
color	TEXT	NOT NULL

Tabla 15. Definición SQL de la tabla "Apparatus".

Apparatus_area		
Nombre	Tipo de datos	
area_id	INTEGER	PRIMARY KEY

Tabla 16. Definición SQL de la tabla "Apparatus area".

Área

Representan áreas definidas a parte del aparato y que pueden ser seleccionadas en las zonas.

Área		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
type	TEXT	NOT NULL
center_x	INTEGER	NOT NULL
center_y	INTEGER	NOT NULL
angle	INTEGER	NOT NULL
width	INTEGER	
height	INTEGER	
sides	INTEGER	
radius	INTEGER	
color	TEXT	NOT NULL

Tabla 17. Definición SQL de la tabla "Áreat".

Zone, Zone_apparatus_area y Zone_area

Representan cada zona, su definición, áreas asociadas y métricas seleccionadas en cada zona.

Zone		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
name	TEXT	NOT NULL
order	INTEGER	NOT NULL
entries	INTEGER	NOT NULL
time	INTEGER	NOT NULL

Tabla 18. Definición SQL de la tabla "Zone".

Zone_apparatus_area		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
zone_id	INTEGER	FOREING KEY
área_id	INTEGER	FOREING KEY

Tabla 19. Definición SQL de la tabla "Zone apparatus area".

Zone_area		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
zone_id	INTEGER	FOREING KEY
apparatus_area	INTEGER	NOT NULL

Tabla 20. Definición SQL de la tabla "Zone area".

Ruler

Almacena las coordenadas de posición de la regla, color en el interfaz y su longitud en milímetros. La regla se usa para establecer una equivalencia entre la distancia en pixeles de una parte del aparato y el tamaño real del mismo.

Ruler		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
x1	TEXT	NOT NULL
y1	INTEGER	NOT NULL
x2	NTEGER	NOT NULL
y2	INTEGER	NOT NULL
length	INTEGER	NOT NULL
color	TEXT	NOT NULL
metric	INTEGER	NOT NULL

Tabla 21. Definición SQL de la tabla "Rulert".

Tracker

Almacena los ajustes del algoritmo de visión artificial para el seguimiento de objetos.

Tracker		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
subtractor	TEXT	NOT NULL
threshold	INTEGER	NOT NULL
threshold_min	NTEGER	NOT NULL
area_min	INTEGER	NOT NULL
area_max	INTEGER	NOT NULL
zoom	REAL	NOT NULL
entries	INTEGER	
time	INTEGER	
freeze	INTEGER	

Tabla 22. Definición SQL de la tabla "Trackert".

Scaler

Finalmente, en esta tabla se almacenan datos relativos al tamaño del aparato en pantalla.

Scaler		
Nombre	Tipo de datos	
id	INTEGER	PRIMARY KEY
width	INTEGER	NOT NULL
height	INTEGER	NOT NULL
ratio_x	REAL	NOT NULL
ratio_y	REAL	NOT NULL
offset	REAL	NOT NULL

Tabla 23. Definición SQL de la tabla "Scalert".

Son necesarios ya que el tamaño del fotograma del video puede distar del mostrado en pantalla y es necesario hacer un cambio de escala previo para adaptarlo al tamaño del fotograma.

3.6 Interfaz de usuario

En relación a los requisitos, se ha diseñado una interfaz de usuario. Han sido necesarias sucesivas fases de refinamiento hasta obtener un diseño que combine adecuadamente la sencillez de uso que demandan los usuarios y la funcionalidad necesaria.

Se ha dispuesto una interfaz con pestañas, que siguiendo un movimiento de izquierda a derecha recuerda el flujo de trabajo del investigador.

La primera pestaña ‘Archivo’, permite funciones básicas como creación de un proyecto nuevo, carga de un proyecto anterior y guardado.

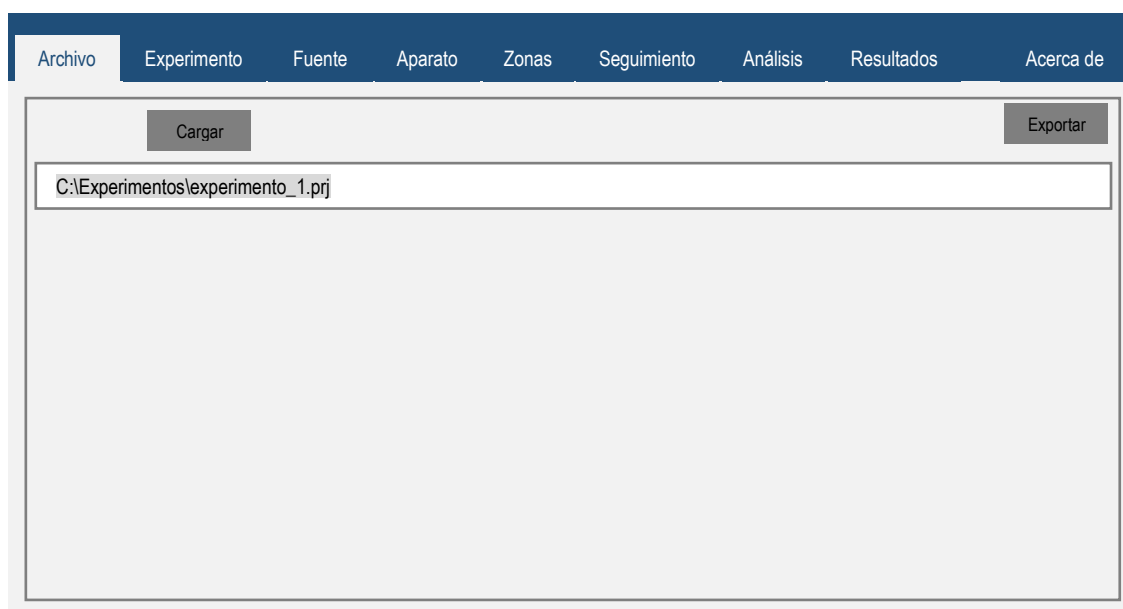


Figura 17. Interfaz. de la pestaña ‘Archivo’.

La pestaña ‘Experimento’ nos permite definir los parámetros básicos del mismo como nombre, grupos, animales, tipos, test y notas.

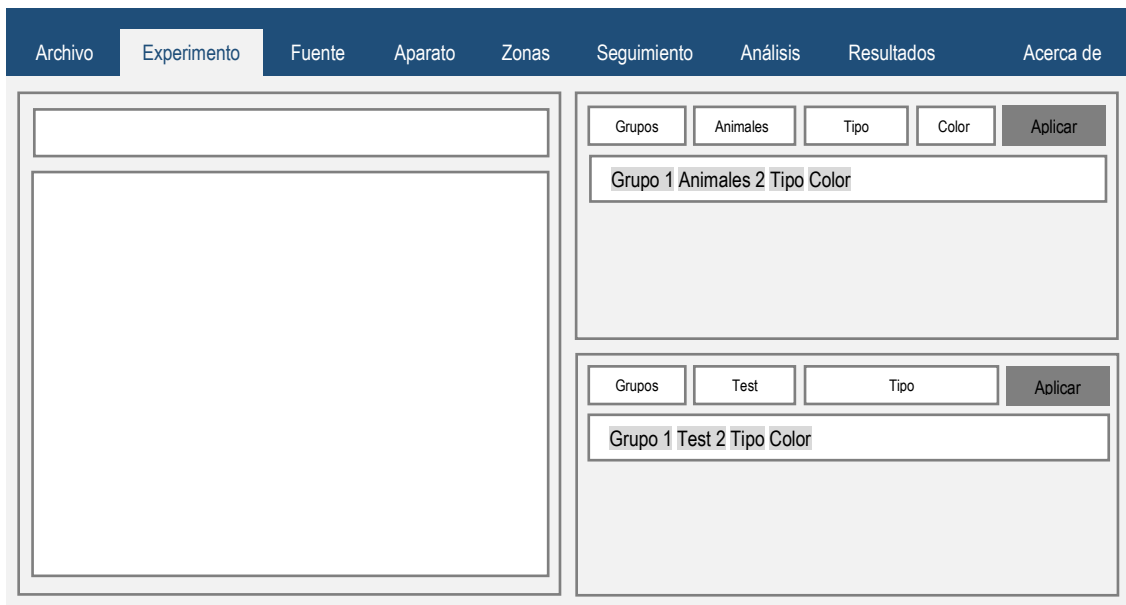


Figura 18. Interfaz. de la pestaña ‘Experimento’.

En la siguiente pestaña ‘Fuentes’ muestra a su izquierda el navegador con todos los Test y a su derecha un listado de fuentes de video (Dispositivos y Archivos de video).

Los dispositivos se cargan en el listado automáticamente.

Permite asignar una fuente a cada Test.

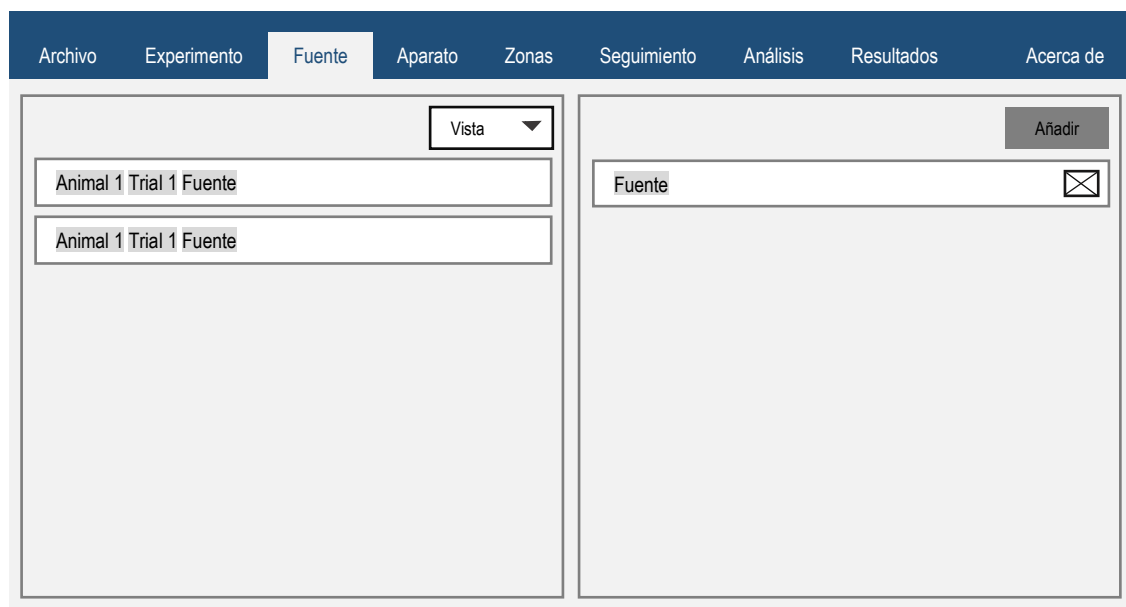


Figura 19. Interfaz. de la pestaña ‘Fuente’.

La pestaña ‘Aparato’, permite en su parte izquierda, seleccionar el tipo de aparato, sus características, como divisiones y/o brazos. Se pueden añadir áreas adicionales y seleccionar la fuente de video a reproducir.

En la parte derecha se reproduce un video o un dispositivo y sobre ella se muestran la regla y el aparato.

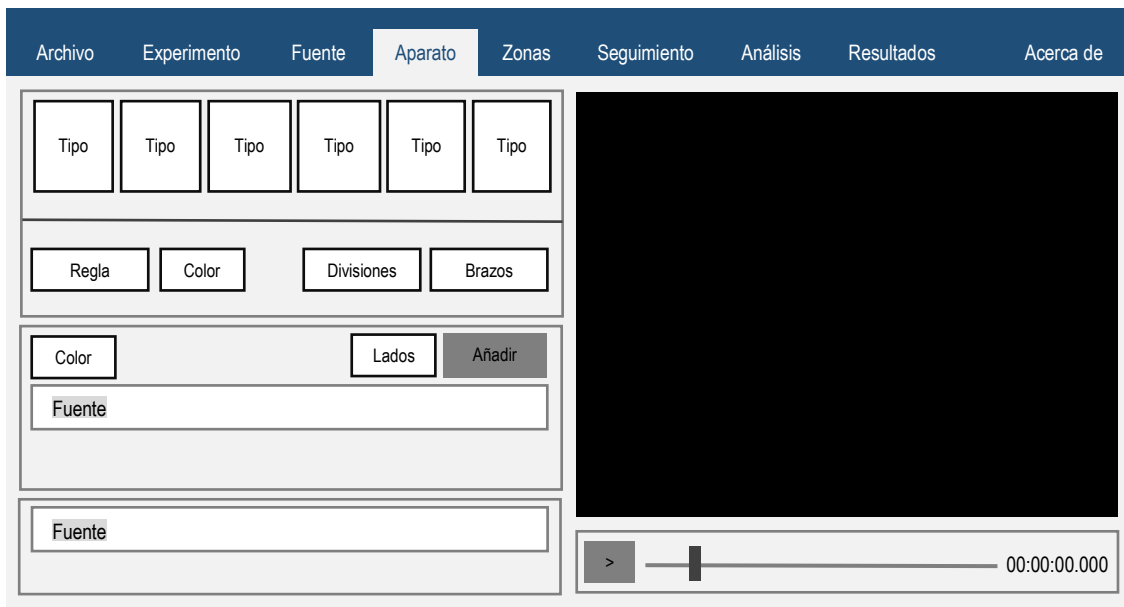


Figura 20. Interfaz. de la pestaña ‘Aparato’.

La pestaña ‘Zonas’, en su parte izquierda, permite añadir nuevas zonas y seleccionar, áreas asociadas y tipos de datos a recolectar en cada zona.

En la parte derecha se muestra una representación del aparato y áreas adicionales.

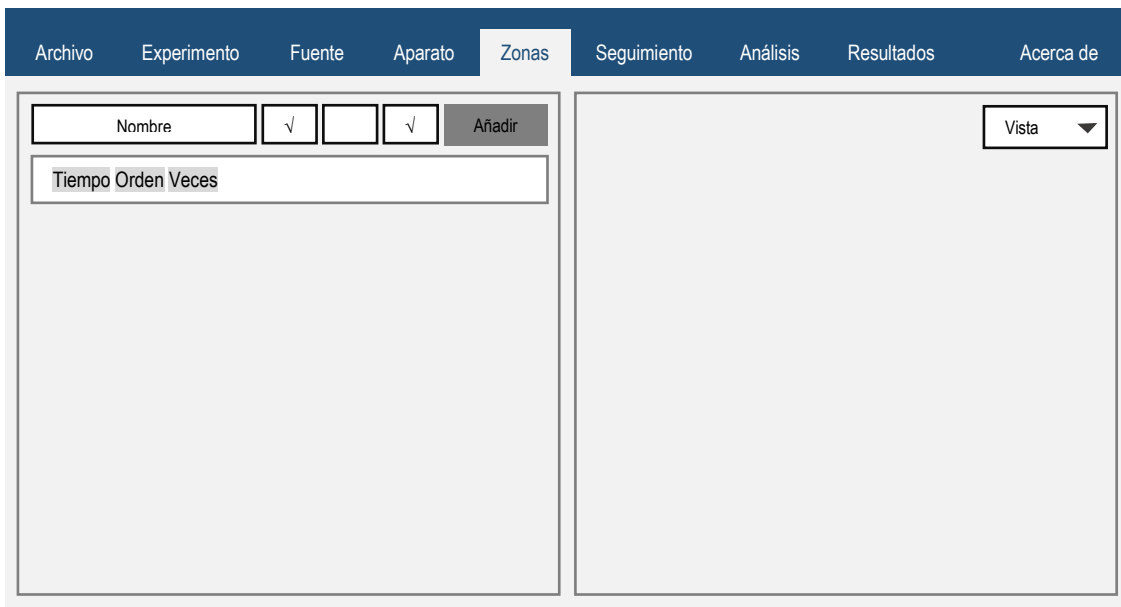


Figura 21. Interfaz. de la pestaña ‘Zonas’.

La pestaña ‘Seguimiento’, nos permite seleccionar el sistema de seguimiento, sus características, comportamiento y finalización con una visualización en tiempo real.

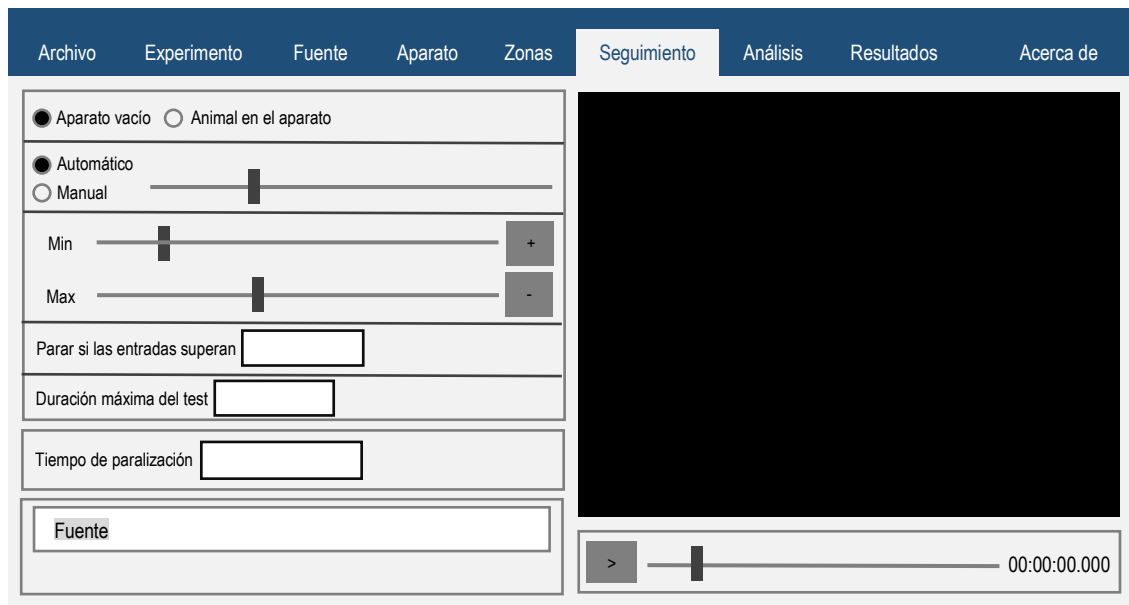


Figura 22. Interfaz. de la pestaña 'Seguimiento'.

La pestaña 'Análisis', nos muestra un navegador con todos los Test listas para su análisis y en la parte derecha los controles para ejecutar uno, todos los análisis o en el caso de una cámara comenzar o detener la grabación.

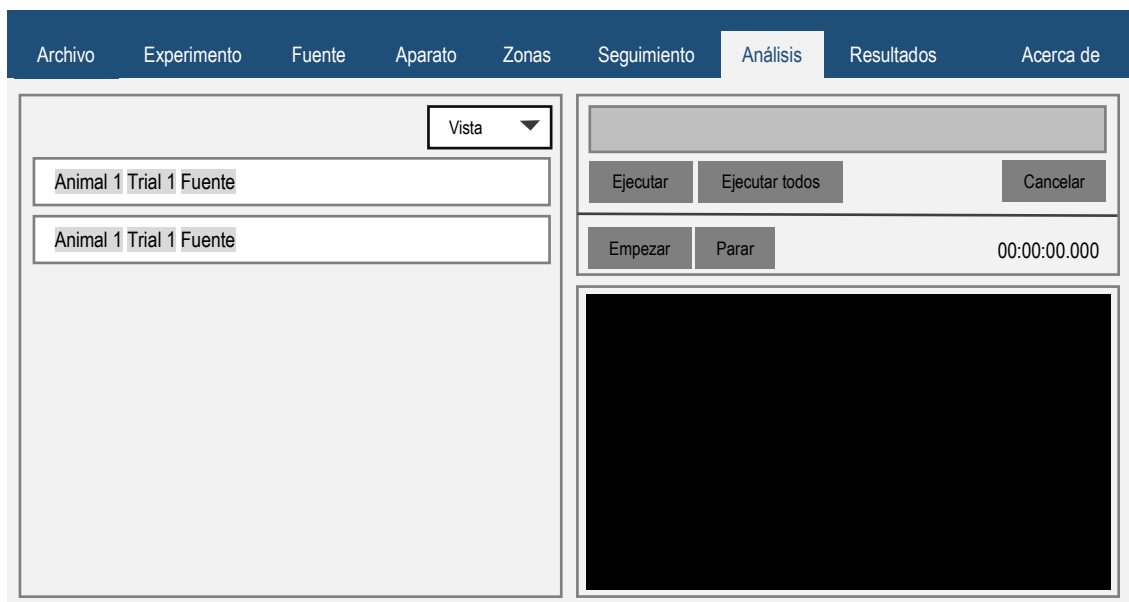


Figura 23. Interfaz. de la pestaña 'Análisis'.

La pestaña 'Resultados', Finalmente, tenemos en la parte izquierda un navegador de resultados y en la parte derecha las métricas correspondientes al resultado seleccionado del navegador.

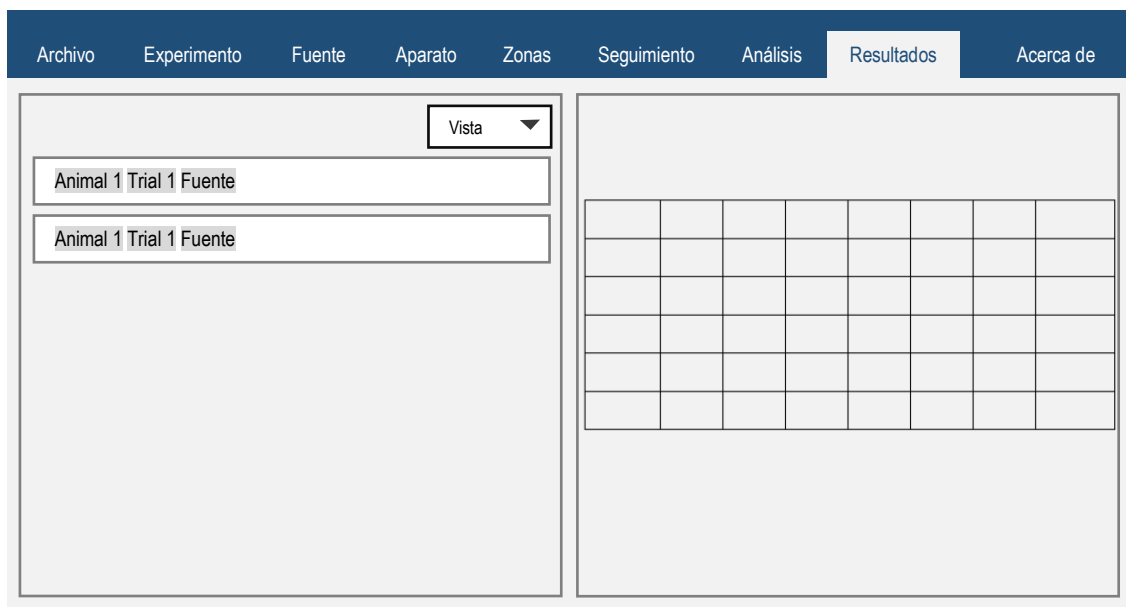


Figura 24. Interfaz. de la pestaña 'Resultados'.

Finalmente, una última pestaña 'Acerca de', nos muestra opcionalmente la información y licencias del software y sus componentes.

4 Implementación

En este capítulo se detalla la implementación en detalle de la arquitectura, estructuras de datos y diseño que se ha analizado en el capítulo anterior.

Siguiendo el patrón Modelo-Vista-Controlador explicaremos como se ha estructurado e implementado cada parte.

4.1 Modelo

El modelo es una parte compleja que facilita el acceso a las fuentes de video y realiza todas las funciones necesarias para el acceso a la base de datos.

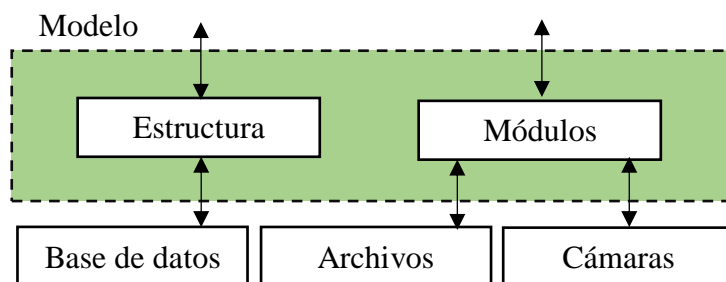


Figura 25. Modelo.

En la Figura 25 se muestran los elementos que componen el modelo y que se explican a continuación.

- **Estructura:** Permite realizar consultas, inserciones, borrado y actualización de la base de datos. Actúa como una capa de traducción devolviendo siempre listas estándar de Python.
- **Módulos:** se compone de los módulos que nos permiten el acceso de lectura a los dispositivos, a los archivos de video y el registro de las imágenes en archivos de video.

4.1.1 Módulos

Como parte del modelo y para facilitar el acceso a las fuentes de video (archivos y cámaras), se ha desarrollado un módulo que engloba ambas funcionalidades bajo el nombre de “*Audio Video Software Development Kit*” (AVSDK).

4.1.1.1 Audio Video Software Development Kit

Este módulo es el responsable de las funciones de captación de imágenes, desde un dispositivo de video (Cámaras) o desde archivos de video. También, posee funciones para la conversión de formato, color, resolución y diversas estructuras de datos internas.

Si bien, forma parte de este proyecto, se tomó la decisión de desarrollarlo como módulo, para dotarlo de la máxima portabilidad y de este modo poder incorporarlo (sin cambios) a cualquier otro proyecto desarrollado en Python.

Se apoya directamente en dos librerías externas, FFMPEG y OpenCV. El acceso a OpenCV es directo al estar disponibles los *bindings* oficiales para Python, sin embargo, para el acceso a FFMPEG desde Python ha sido necesario el desarrollo de *bindings* propios usando CTypes.

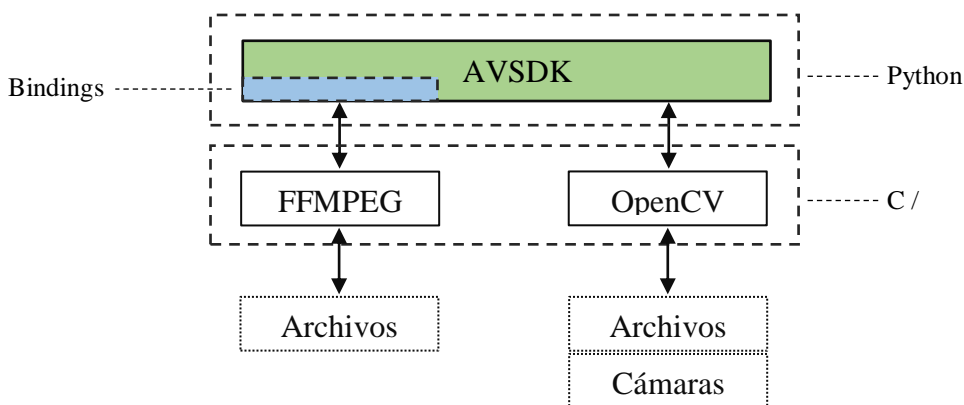


Figura 26. Visión general del sistema de módulos y librerías externas.

Para el acceso se han creado dos librerías AVKit y DEVKit integradas en AVSDK. AVKit permite el acceso a archivos de video y DEVKit a dispositivos de video (Cámaras).

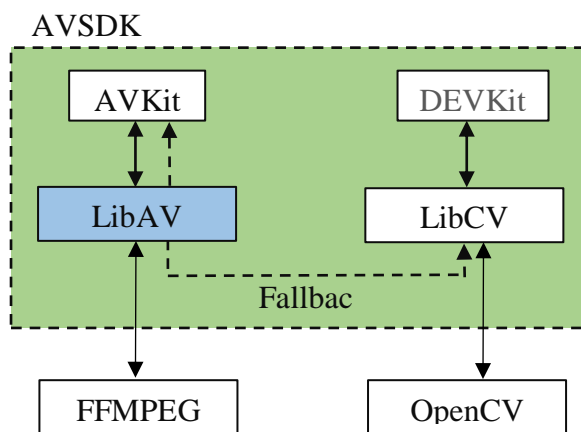


Figura 27. Estructura interna del módulo “Audio Video Software Development Kit”.

Ambas librerías tienen el mismo flujo de funcionamiento con independencia de la fuente y librería externa a la que acceda. Exponen una interfaz externa común que se detalla a continuación.

- Abre una fuente de video.

`open (string filename)`

`filename` Ruta al archivo de video o cámara.

Devuelve -1 en caso de error sino 0.

- Lee el siguiente fotograma.

`read ()`

Devuelve un puntero a la instancia del fotograma.

- Transforma y redimensiona un fotograma a formato imagen BGR o RGB.

`rescale(Frame frame, integer width, integer height, integer pix_fmt)`

`frame` Puntero a la instancia del fotograma.

`width` Número de columnas de la imagen de salida.

`height` Número de filas de la imagen de salida.

`pix_fmt` Formato de imagen de salida (BGR o RGB)

Devuelve un puntero a la instancia de la imagen.

- Realiza una búsqueda del fotograma al que corresponde el valor de tiempo.

`seek(integer time, [integer flag])`

`time` Tiempo a buscar en milisegundos.

`flag` Parámetro opcional.

AVSEEK_FLAG_BACKWARD = 1

AVSEEK_FLAG_ANY = 4

AVSEEK_FLAG_FRAME = 8

Devuelve -1 en caso de error sino 0.

- Cierra una fuente de video.

`close ()`

No devuelve nada.

- Permite obtener la duración de la fuente de video (solo archivos de video), alto del fotograma, ancho del fotograma, relación de aspecto del fotograma y punto temporal exacto correspondiente al último fotograma leído.

`get(integer option)`

option Este valor puede especificarse como:

```
FRAME_WIDTH = 3
FRAME_HEIGHT = 4
FRAME_TIME = 7
STREAM_DURATION = 9
FRAME_ASPECT = 13
```

Devuelve el valor solicitado o NULL si no existe.

- Permite obtener la versión de todos los componentes de FFMPEG/ OpenCV en uso.

`version()`

Devuelve una lista que contiene los valores correspondientes a la versión de cada uno de los componentes de FFMPEG u OpenCV en uso.

- Permite obtener el nivel de registro de eventos de la librería.

`get_log_level ()`

Devuelve el valor numérico correspondiente al nivel actual.

```
AV_LOG_QUIET = -8
AV_LOG_DEBUG = 32
```

- Permite seleccionar el nivel de registro de eventos de la librería.

`set_log_level ()`

No devuelve nada

Finalmente, tenemos los detalles de las dos estructuras de datos respónsables de almacenar cada fotograma y su representación en imagen.

- Objeto del último fotograma leído.

Frame (**pointer** data)

data Puntero que contiene el fotograma en la librería externa.

- Objeto de la imagen generada desde el fotograma.

Picture (**integer** pts
array data
integer width
integer height
integer pixel_format)

pts "Presentation Timestamp", Tiempo asociado al fotograma de la imagen.

data Vector de vectores. Contiene cada pixel de la imagen.

width Número de columnas de la imagen.

height Número de filas de la imagen.

pixel_format Número que identifica el formato de imagen, BGR o RGB.

Observar el flujo de funcionamiento en la Figura 28 nos ayudará a entender en qué lugar inciden cada una de las funciones expuestas.

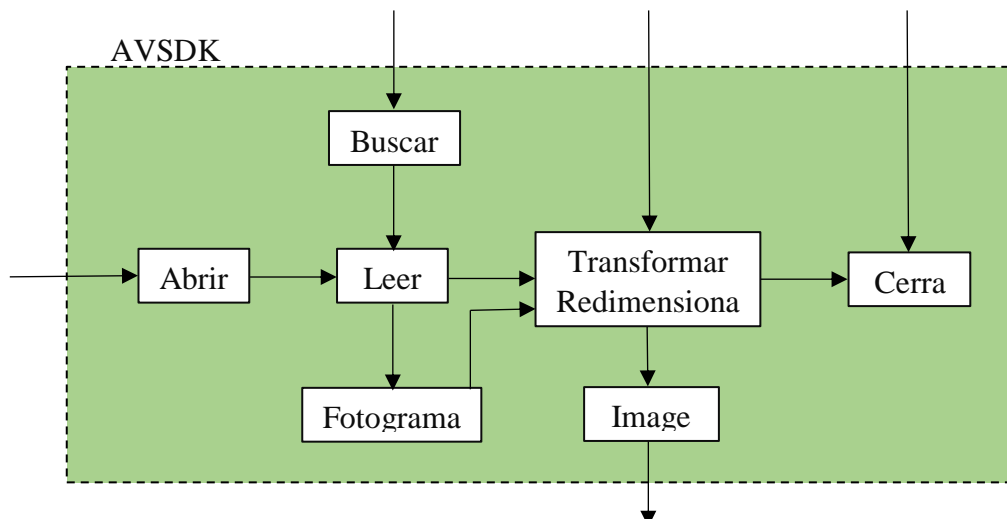


Figura 28. Flujo de funcionamiento del módulo "Audio Video Software Development Kit".

AVKit

AVKit usa FFMPEG u OpenCV indistintamente con una interfaz interna común. Además, se ha implementado un sistema *fallback* o alternativa, que nos permite cambiar FFMPEG por OpenCV automáticamente si el primero no estuviera disponible.

Este sistema de *fallback* dota de mayor robustez a la librería e intenta garantizar su funcionamiento en la medida de lo posible.

En la Figura-24 observamos que hace uso de LibAV y de LibCV. Ambas son dos librerías que se han desarrollado expresamente en este proyecto.

LibAV es la más compleja y amplia, su desarrollo se ha centrado en la portabilidad y eficiencia. La estructura interna de LibAV no se ha incluido en este apartado por su tamaño (Apendice-2). Se compone de los *bindings* programados con CTypes y funciones adicionales para la interfaz para comunicarse con las librerías de FFMPEG t proporcionar los servicios de decodificación de video.

LibCV es la librería análoga a LibAV, replica su funcionalidad e interfaz. Su estructura interna es más simple al no necesitar de bindings propios (Apéndice-3). Permite la misma funcionalidad que LibAV pero en este caso haciendo uso de OpenCV.

DEVKit

DEVKit usa en exclusiva OpenCV a través de LibCV para el acceso a los dispositivos (Cámaras). Exponiendo la misma interfaz, su estructura interna es más sencilla (Apendice-4).

4.2 Controlador

El controlador está estructurado con una clase padre que tiene como hijos una clase por cada pestaña del interfaz. En la Figura 29 se muestra su estructura. Esta estructura nos permite encapsular de forma clara las funcionalidades de la interfaz en el controlador.

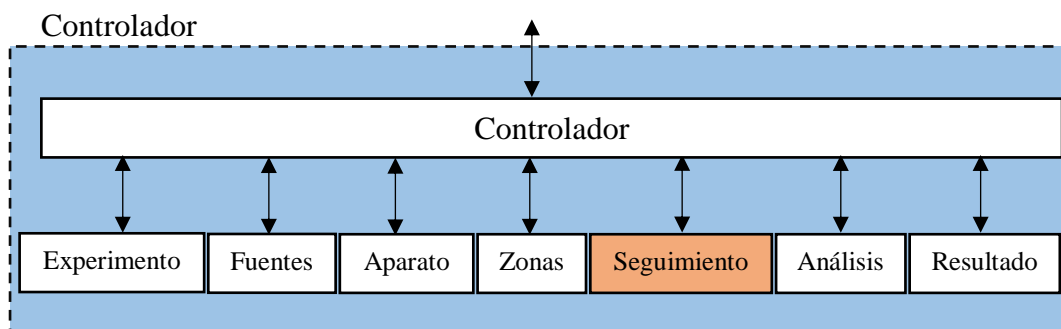


Figura 29. Estructura interna del controlador.

En la Figura 30, se detallan los módulos y clases que internamente necesita el controlador para realizar las funciones que exigen cálculos, procesado de imágenes, segmentación, escalado de formas y un reproductor completamente funcional para visualizar una cámara o archivo de video.

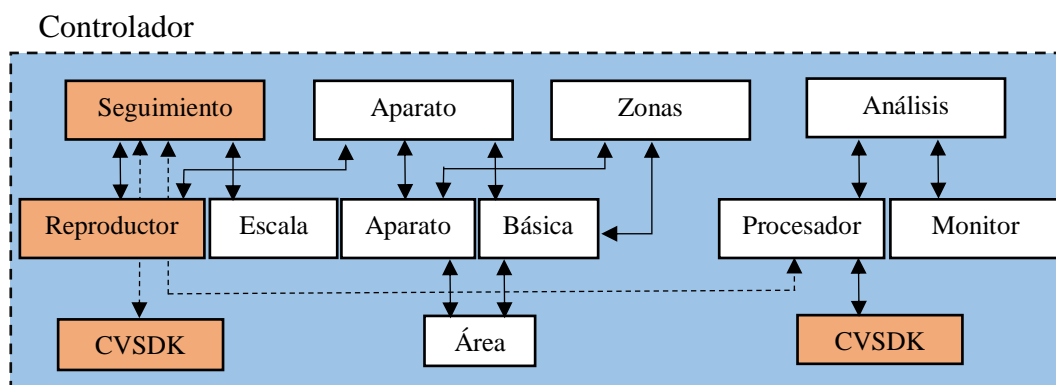


Figura 30. Vista en detalle de los módulos y clases del controlador.

Destacan tres elementos.

- **Reproductor:** Ha sido necesario programar completamente un reproductor de video en Python con una interfaz basada en la librería Tkinter..
- **Seguimiento:** Parte del controlador encargada de gestionar el seguimiento de objetos. Se apoya directamente CVSDK “Computer Vision Software Development Kit”.
- **CVSDK “Computer Vision Software Development Kit”:** Librería para el acceso a las funciones de visión artificial. Capaz de realizar detección en primer plano, segmentación y localización de objetos. Hace uso directo de OpenCV.

4.2.1 Reproductor de video

Debido a los requisitos del proyecto, el lenguaje de programación elegido y en especial la incapacidad de OpenCV para proporcionar datos fiables imprescindibles como el sello temporal de cada fotograma o la duración total de un archivo de video.

Por todo ello, ha sido necesario desarrollar un reproductor de video usando el módulo “Audio Video Software Development Kit”, que ya hemos explicado anteriormente.

Tkinter es la única librería que el lenguaje de programación Python incluye para el desarrollo de interfaces de usuario. Aunque es bastante completa, tiene algunas limitaciones técnicas. Concretamente, no permite modificar la interfaz desde otros procesos o hilos de ejecución, excepto el principal.

Teniendo en cuenta esta limitación y algunas otras, se ha implementado un sistema basado en eventos e hilos que nos permiten actualizar la interfaz a la vez que decodificamos, transformamos y creamos nuevas imágenes con márgenes de tiempo del orden de los milisegundos.

Para reproducir un video es necesario conocer el momento exacto en el que debe presentarse en pantalla cada fotograma, por lo que todos los videos incluyen un sello de tiempo que nos especifica dicho momento llamado PTS o “Presentation Time Stamp”.

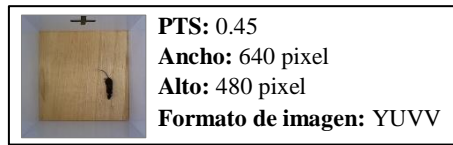


Figura 31. Datos del fotograma de video.

Es habitual sincronizar la presentación de los fotogramas respecto al reloj del sistema. De este modo se realiza una sincronización entre el reloj y el video que nos permite saber si la velocidad de reproducción es fiel al original. Este sistema se puede ver en la Figura 32.

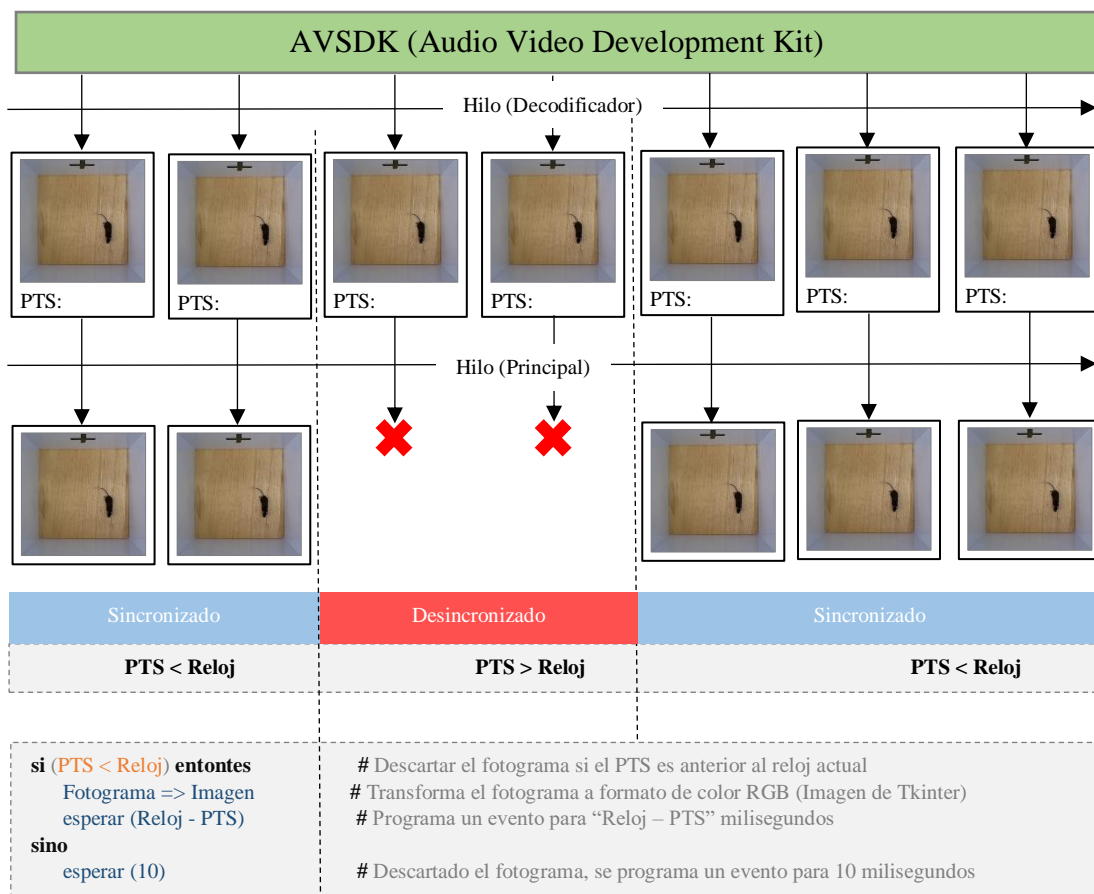


Figura 32. Sincronización de reloj del reproductor de video.

Cada fotograma es decodificado e inmediatamente se compara su PTS con el reloj. Si el PTS anterior al reloj, es decir, tiene un valor menor, significa que la decodificación ha llevado demasiado tiempo y es tarde para presentarlo en pantalla (Al descartarlo, “ahorramos” unos milisegundos para intentar obtener un fotograma que tenga un PTS mayor que nuestro reloj). En caso contrario, el hilo principal actualiza la imagen en la interfaz.

Si el hardware y software es suficientemente veloz, no habrá necesidad de re-sincronizar y la reproducción será fluida.

4.2.2 Seguimiento de objetos

Una parte fundamental del controlador es el seguimiento de los objetos usando la librería de visión artificial. Para ello, se usa CVSDK “*Computer Vision Software Development Kit*” que explicaremos próximamente. De momento, veamos que algoritmos se han usado y como se ha implementado esta funcionalidad.

Si recordamos, en el capítulo 1 hemos hablado de las técnicas usadas para el seguimiento de objetos con visión artificial.

En concreto, se ha implementado siguiendo el siguiente esquema.

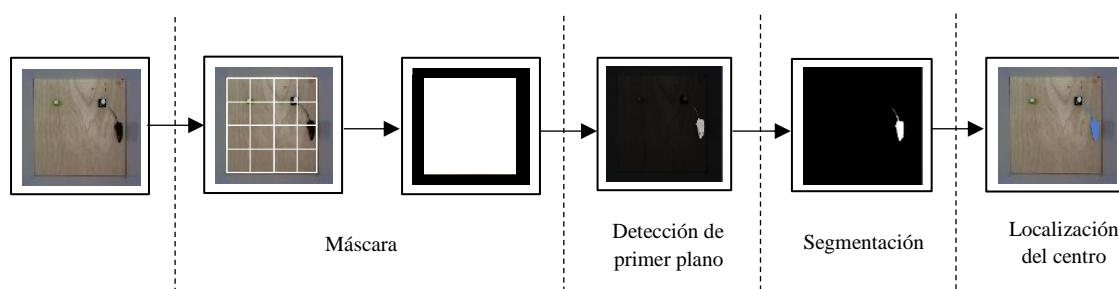


Figura 33. Implementación: Pasos para el seguimiento de objetos con visión artificial.

Los pasos de la Figura 33, son los que ya conocíamos a excepción de la Máscara. Este paso previo es necesario para el comienzo automático del seguimiento sin intervención del usuario.

La técnica detrás de este paso es sencilla, se pintan de negro todos los píxeles que se encuentran fuera del perímetro marcado por el aparato y de blanco todos los que queden dentro. (Recordemos que el aparato fue modelado por el usuario previamente en la aplicación).

Con la máscara creada podemos recortar los mismos píxeles que sean blancos de la imagen original, lo que da como resultado una imagen que contiene únicamente la porción de la original que nos interesa.

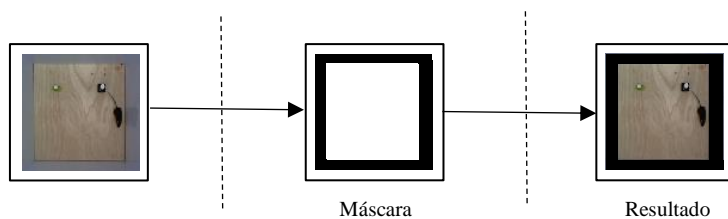


Figura 34. Proceso para enmascarar la imagen.

Al estar enmascarada la imagen, solo se van a detectar objetos en la región que queda visible y por tanto en el momento que un objeto de las características adecuadas se detecte daremos comienzo al experimento.

El flujo para el seguimiento de objetos queda definido con los siguientes pasos.

1. Crear máscara y enmascarar la imagen.
2. Detección de primer plano.
3. Segmentación.
4. Localización del centro de masas.

Detección de primer plano

Para este paso se ha implementado un sistema que permite seleccionar el tipo de algoritmo. En esta aplicación se usan sólo dos algoritmos.

Background subtractor: basado en la diferencia de imágenes e implementado en OpenCV. Probablemente el más rápido y con muy buenos resultados si la iluminación en el video no cambia demasiado. Es el seleccionado por defecto en la aplicación, aunque no es recomendable si el animal ya está dentro del objeto.

Nota: en el caso del *background subtractor*, es necesario realizar un *threshold* de la imagen resultante previo a la segmentación.

Background subtractor CNT: bastante rápido y disponible en OpenCV modela dinámicamente el fondo para extraer el objeto en primer plano. Es el recomendado cuando el animal ya está en el interior del aparato.

Nota: en este caso no se necesita un *threshold* ya que todo el proceso esta implementado en el algoritmo.

Segmentación

La segmentación de la imagen se realiza con OpenCV y después se filtran los contornos encontrados por tamaño para eliminar el ruido que puede aparecer.

```
contornos = encontrar_contornos(imagen)
Si contornos entonces
    contornos = ordenar_por_tamaño(contornos)
    contornos = filtrar_por_tamaño(primer(contornos), minimo, máximo)
Si contornos entonces
    devolver contornos
devolver vacio
```

Tabla 24. Pseudocódigo para segmentar una imagen.

Localización del centro de masas

OpenCV proporciona funciones dedicadas al cálculo del momento de un contorno. Usando estas funciones es sencillo obtener el centro de masas del objeto que representan las coordenadas a usar en esta imagen para el cálculo de las métricas del experimento.

4.2.3 Filtros de localización

Cuando tenemos la localización del objeto usamos un filtro Kalman, que usa técnicas de predicción y corrección de señales para corregir los pequeños errores que se producen con las técnicas de visión artificial.

En concreto se ha usado con dos objetivos.

1. Corregir los pequeños movimientos o “tintineo” del centro de masas entre fotogramas adyacentes. Reduciendo estos micro movimientos obtenemos una detección más fiel y evitamos aumentos artificiales en métricas, como por ejemplo la distancia recorrida.
2. En las ocasiones en las que durante periodos de tiempo muy pequeños perdamos la localización del objeto y no pueda ser detectado, podemos predecir y usar como válidos las coordenadas que predice el filtro Kalman

Se ha utilizado la implementación que ofrece la librería OpenCV con las siguientes matrices.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * 0.03$$

MeasurementMatrix

TransitionMatrix

ProcessNoiseCov

En cada iteración del bucle de seguimiento se realiza un paso de predicción y otro de corrección con las coordenadas detectadas. Y se obtienen las coordenadas filtradas y corregidas para su uso en las métricas.

4.2.4 Módulos

Como parte del controlador, se ha desarrollado otro módulo con la funcionalidad necesaria para la visión por ordenador bajo el nombre “*Computer Vision Software Development Kit*” (CVSDK).

4.2.4.1 Computer Vision Software Development Kit

Este módulo es el encargado de las funciones de visión artificial de la solución, usando exclusivamente OpenCV con los *bindings* disponibles para Python. Como única librería tenemos CVKit.

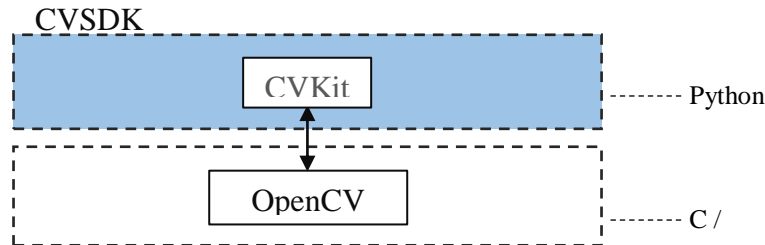


Figura 35. Estructura interna del módulo “Computer Vision Software Development Kit”.

La interfaz externa que expone se compone de unas pocas funciones que detallamos a continuación.

- Inicializa y selecciona el algoritmo a usar.

```
init ( integer subtractor)
```

subtractor Valor que identifica el algoritmo a usar.
 BACKGROUND_SUBTRACTOR = 1
 BACKGROUND_SUBTRACTOR_CNT = 3

No devuelve nada.

- Crea una máscara con las coordenadas seleccionadas, número de canales de la imagen y la invierte si es necesario.

```
set_mask( integer width, integer height, array coords, [[ integer channels ],
           integer invert ] )
```

width Número de columnas de la máscara
height Número de filas de la máscara
coords Coordenadas de los polígonos que definen la máscara.
channels Número de canales de la imagen de salida.
invert Si es necesario invertirla la máscara el valor es 1 sino 0.

No devuelve nada.

- Inicializa el sistema de threshold.

```
set_threshold( integer mode, integer min, integer invert )
```

mode Tipo de threshold a usar.

Automático: Algoritmo CNT de OpenCV.

THRESHOLD_AUTO = 3

Manual: Algoritmo de substracción con fondo fijo.

THRESHOLD_MANUAL = 5

min Valor del threshold.

invert Si es necesario invertir el filtro se pasa 1 sino 0.

No devuelve nada.

- Inicializa el sistema de segmentación de la imagen.

```
set_segmenter( integer filter, integer min, integer max )
```

filter Tipo de filtro a usar.

FILTER_AREA = 1

min Valor mínimo del filtro.

max Valor máximo del filtro.

No devuelve nada.

- Realiza todos los pasos para la detección del objeto.

```
track( Picture picture, [ boolean overlay, [ integer max ] ] )
```

picture Instancia de la imagen.

overlay Selecciona el dibujado del contorno, área y centro de masas. Por defecto True.

max Número de contornos a devolver.

Devuelve la imagen, un vector con los contornos y una tupla con el centro de masas de cada contorno.

Su estructura interna es sencilla dado que la funcionalidad que requiere es limitada.

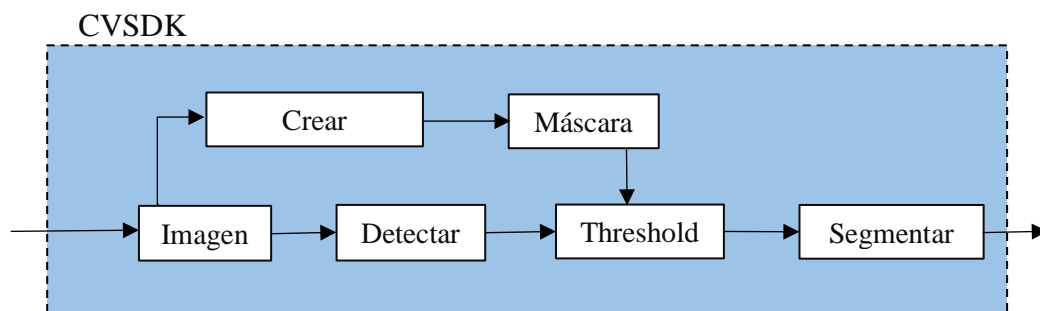


Figura 36. Flujo de funcionamiento del módulo “Computer Vision Software Development Kit”.

4.3 Vista

La vista contiene la interfaz de usuario y llama al controlador para todas las operaciones que no sean exclusivamente dependientes de la interfaz.

La estructura interna no es compleja y refleja la estructura de pestañas que se muestra al usuario. Detallamos la estructura interna en la Figura 37.

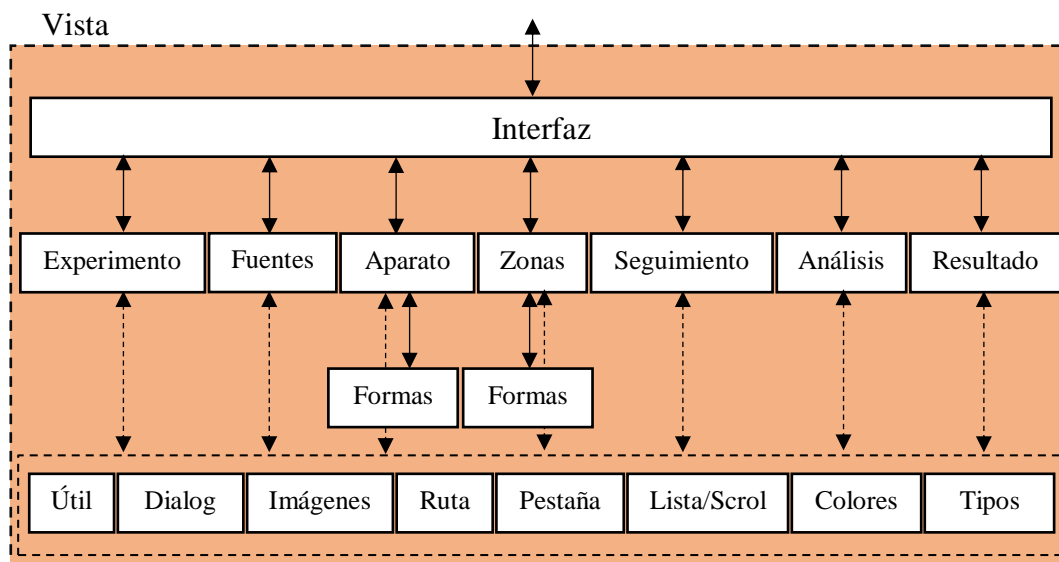


Figura 37. Estructura interna de la vista.

Esta parte del proyecto usa la librería Tkinter para interfaces de Python que viene incluida en el intérprete. En el manual de usuario incluido en la parte final de este documento se detalla toda su funcionalidad.

5 Pruebas

En este capítulo realizaremos el análisis de un conjunto de videos en la aplicación desarrollada, la solución comercial Anymaze* y la aplicación no comercial ToxTrac. Se compararán las métricas obtenidas para verificar su validez y los tiempos de ejecución.

Finalmente, se comprobará que todos los requisitos funcionales y no funcionales han sido implementados acorde a lo estipulado.

*Nota: Dado que Anymaze es un software comercial y no se dispone de una licencia, los resultados incluidos en estas pruebas pertenecen a los proporcionados por los neurocientíficos que han colaborado. Estos resultados son las métricas que se obtuvieron con una versión “4.x” de Anymaze.

5.2 Validación de requisitos

Se ha seguido la lista requisitos comprobando que la funcionalidad esta implementada y tiene el comportamiento esperado. El resultado de esta comprobación de detalla en la Tabla 25.

Requisitos funcionales

		Descripción
1	Si	Definir grupos y número de animales
2	Si	Definir grupos y número de pruebas.
3	Si	Añadir archivos de video
4	Si	Detección automática de dispositivos de video (Cámaras)
5	Si	Navegador de grupos/pruebas
6	Si	Asignar manualmente archivos de video o dispositivos por prueba.
7	Si	Asignar dispositivos de video a todas las pruebas automáticamente.
8	Si	Selección de vista de las pruebas (Por animal/Por Intento o <i>Trial</i>)
9	Si	Selección del tipo de aparato
10	Si	Selección de las características del aparato (Divisiones y/o Brazos)
11	Si	Añadir áreas adicionales.
12	Si	Reproducción de archivos de video (Reproductor)
13	Si	Selección del archivo de video a reproducir
14	Si	Reproducción de video del dispositivo (Cámara) (Monitor)
15	Si	Selección del dispositivo (Cámara) a visualizar
16	Si	Creación automática de zonas
17	Si	Creación manual de zonas

19	Si	Selección de múltiples áreas por zona
20	Si	Selección de métricas por zona
21	Si	Visualización de áreas (Todas/Solo aparato/Solo áreas)
22	Si	Selección del algoritmo de seguimiento
23	Si	Ajuste del valor límite o <i>threshold</i> (Manual: Valor/Automático)
24	Si	Selección del área a seguir (Área mínima/ Área máxima)
25	Si	Ajuste de la condición de parada por tiempo total
26	Si	Ajuste de condición de parada por número de entradas
27	Si	Ajuste del tiempo mínimo para considerar inmóvil al objeto
28	Si	Análisis de todos las pruebas con archivos de video asignados
29	Si	Selección y análisis de una prueba con archivo de video asignado
30	Si	Selección de una prueba con dispositivo (Cámara) asignado
31	Si	Inicio / Parada de la grabación de video del dispositivo (Cámara)
32	Si	Análisis de las imágenes del dispositivo y generación de las métricas
33	Si	Navegador de resultados
34	Si	Selección del resultado a visualizar (Métricas / Mapas)
35	Si	Borrado del resultado seleccionado
36	Si	Exportar resultados a archivo externo
37	Si	Carga de un experimento anterior

Tabla 25. Requisitos funcionales validados.

Requisitos no funcionales

Los requisitos no funcionales que se listan en la Tabla 26, en general se han conseguido.

Para la comprobación de los requisitos 1* y 3* se ha contado con neurocientíficos voluntarios que han instalado, configurado y realizado un flujo completo de trabajo para un archivo de video incluyendo análisis y exportación de los resultados.

Sin embargo, hay que comentar algunas limitaciones que se han encontrado durante las pruebas.

- La ejecución de la aplicación es correcta en un ordenador de capacidad y velocidad de proceso media. Pero determinado hardware, como cámaras o los controladores software asociados pueden causar ralentizaciones en el acceso, lo que puede acarrear problemas al mostrar de velocidad en las imágenes y en los archivos de video generados.
- La aplicación es fácil de usar y configurar en general. Su instalación es inmediata y sencilla sin embargo en ocasiones la instalación del intérprete de Python y sus dependencias pueden resultar confusas o complicadas para algunos usuarios.

	Prioridad	Descripción
1	Si*	Ejecución en ordenador personal de capacidad/velocidad media.
2	Si	Multiplataforma (En la medida de lo posible)
3	Si*	Facilidad de uso, configuración y puesta en marcha.
4	Si	Distribución/Uso libre y sin coste
5	Si	Modular y de fácil mantenimiento

Tabla 26. Requisitos no funcionales validados.

5.1 Validación de las métricas

Se han usado 5 videos escogidos aleatoriamente de los disponibles, realizado la configuración en cada programa y se han analizado con los siguientes resultados.

La máquina usada para la comparativa se detalla a continuación.

- **CPU:** Intel Core i7 6770HQ 2.6 Ghz / 3.5 Ghz
- **Memoria:** 16 GB RAM DDR4 2133 Mhz
- **Disco:** 500 GB SSD
- **Sistema:** Windows 10 64bit

Distancia recorrida

Se ha elegido la distancia recorrida ya que es la más representativa. El gráfico de la representa la distancia recorrida por cada aplicación para un tiempo fijo de 10 minutos.

Tomando como base la distancia estimada por la aplicación “Anymaze”, por ser uno de los referentes en este campo, vemos que tanto Toxtrac como nuestra aplicación muestran ligeras discrepancias.

En general, en los videos 1, 2, 3, y 5 se estiman distancias ligeramente superiores siendo la excepción el video 4 en el que se estima una distancia algo inferior.

Los datos indican que estas discrepancias quedan por debajo de 1 metro, lo que parece razonable debido a los pequeños cambios que se introducen al modelar digitalmente el aparato y posicionar la regla a mano. Lo que en parte explicaría las diferencias en los resultados.

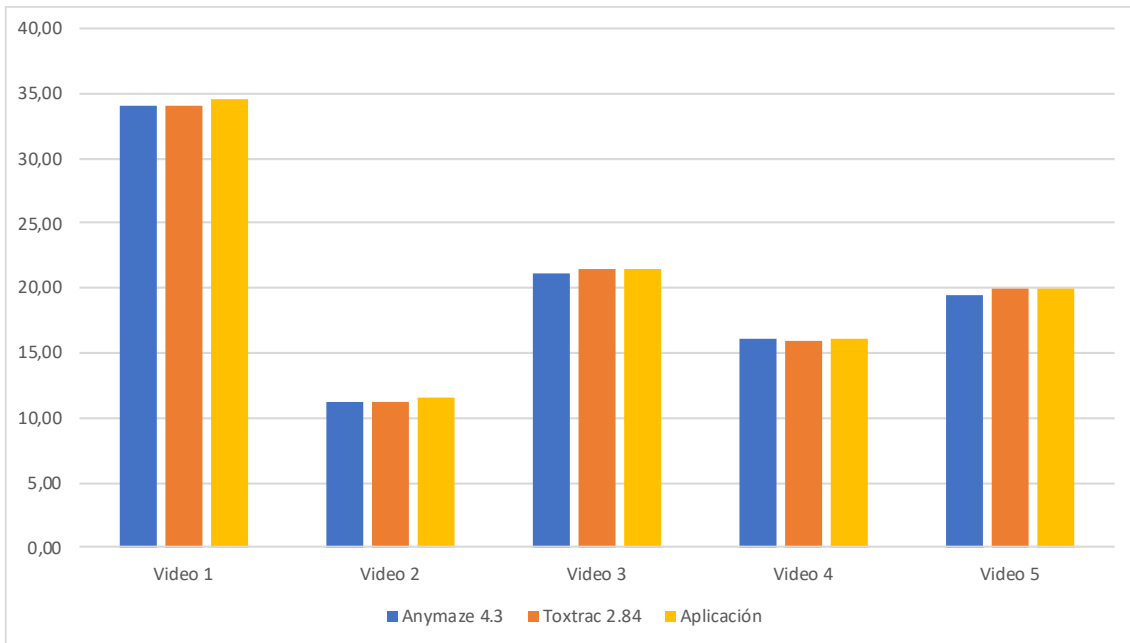


Figura 38. Grafica de distancia recorrida. La distancia se muestra en metros.

Siendo el resto de métricas calculadas en base a la distancia y el tiempo, no considero que sea necesario tenerlas en cuenta en esta comparativa.

Tiempo de análisis

Si bien, no afecta a las métricas si afecta al neurocientífico, por lo que vamos a comparar los tiempos de análisis que cada aplicación requiere en cada video.

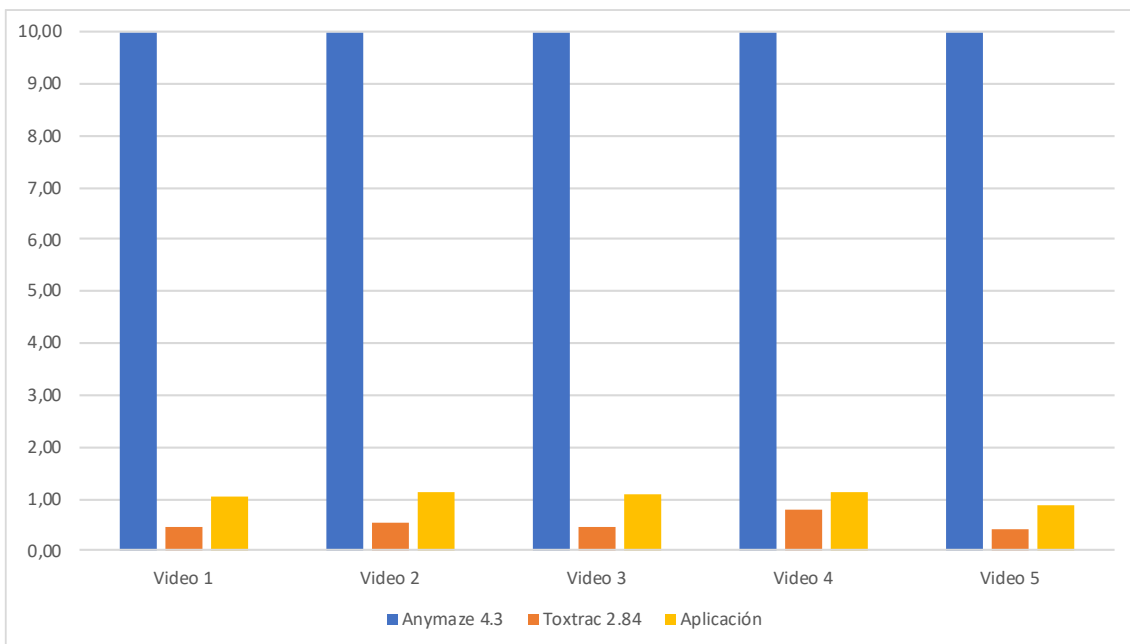


Figura 39. Grafica de tiempo de análisis. Los tiempos se muestran en minutos.

El tiempo se ha fijado a 10 minutos con independencia de la duración del video.

Los grafica arroja una gran diferencia entre Anymaze y los demás, pero tiene una explicación. Anymaze analiza cada video como si se tratara de video en tiempo real, por

lo que se visualiza por completo siendo siempre igual la dirección del análisis a la duración fijada.

En cuanto a Toxtrac, es una aplicación muy rápida escrita en C/C++, un lenguaje que tiene un mejor soporte nativo para la gestión de cargas multi-hilo más eficiente que Python, lo que probablemente explica que sea hasta un 60% más rápida.

En cualquier caso, nuestra aplicación es capaz de analizar 10 minutos de video en alrededor de 1 minuto, lo que es razonablemente eficiente y marca una enorme diferencia con la aplicación de referencia.

6 Conclusiones y trabajo futuro

En este capítulo reflexionamos sobre el proyecto, los objetivos y desafíos para finalmente exponer nuevas ideas para el futuro de esta aplicación.

6.1 Conclusiones

A lo largo de este proyecto hemos aprendido como los neurocientíficos estudian la memoria, que tipos de experimentos realizan y que necesidades tienen.

También, hemos estudiado que aplicaciones usan para medir estos experimentos, las técnicas de visión artificial y corrección de señales que incorporan, sus características, ventajas y desventajas.

Con el desarrollo de esta aplicación intentamos ayudar a los neurocientíficos que carecen de presupuesto para soluciones comerciales a la vez que ofrecemos una alternativa válida, técnicamente comparable, multiplataforma y sin coste.

Para llevar a cabo este proyecto, se ha realizado una labor de estudio y documentación de técnicas de visión artificial, decodificación de video, técnicas para reproducción de video y creación de librerías cruzadas Python-C/C++ mediante bindings de lenguaje.

La labor de análisis y diseño de este proyecto ha facilitado una implementación ordenada, de fácil mantenimiento y correcto cumplimiento de los requisitos funcionales y no funcionales.

Ha sido una experiencia en ocasiones, complicada, abrumadora, extenuante pero también muy gratificante, inspiradora, creativa y recomendable.

6.2 Trabajo futuro

Este proyecto abre las puertas a colaborar con otros ingenieros y neurocientíficos que quieran aportar ideas y trabajo de desarrollo.

Las posibilidades para ampliar, mejorar y perfeccionar son casi infinitas. Si nos comparamos con las aplicaciones comerciales, algunas con 20 años de desarrollo, encontramos los siguientes puntos, entre otros.

- Soporte multicámara. Posibilidad de analizar varias fuentes de video en tiempo real.

-
- Soporte para más de un aparato por fuente de video. En ocasiones para acelerar el proceso se realizan experimentos con varios aparatos y animales simultáneamente y se graban con una única cámara.
 - Soporte para identificación y análisis de varios animales en un mismo aparato.
 - Soporte para sensores externos.
 - Soporte para nuevos aparatos como Morris Water-Maze.
 - Soporte para scripts de usuario que definan dinámicas complejas de experimento, como por ejemplo experimentos de miedo.
 - Generación de estadísticas complejas y gráficas para los grupos y subgrupos definidos por el usuario.

7 Manual de usuario

En este capítulo se incluye un manual, primero con las indicaciones para la instalación y puesta en marcha de la aplicación y después con la dinámica habitual de uso en un experimento.

7.1 Instalación del intérprete y dependencias

La instalación del intérprete de lenguaje Python y las dependencias divergen según el sistema operativo utilizado.

Windows

1. Nos dirigimos a la página web oficial <http://www.python.org> y descargamos la última versión de Python 3 64 bit, hacemos doble clic sobre el archivo descargado y seguimos las instrucciones del instalador.

<https://www.python.org/ftp/python/3.7.3/python-3.7.3-amd64.exe>

2. Instalación de PIP (Instalador de paquetes para Python).
Descargamos el script “get-pip.py” desde el siguiente enlace y lo guardamos en el directorio raíz del disco duro, por ejemplo, “C:\”.

<https://bootstrap.pypa.io/get-pip.py>

Una vez descargado abrimos una consola pulsando las teclas Windows + R, en la ventana que aparece escribimos “powershell” (sin comillas) y pulsamos “OK”.

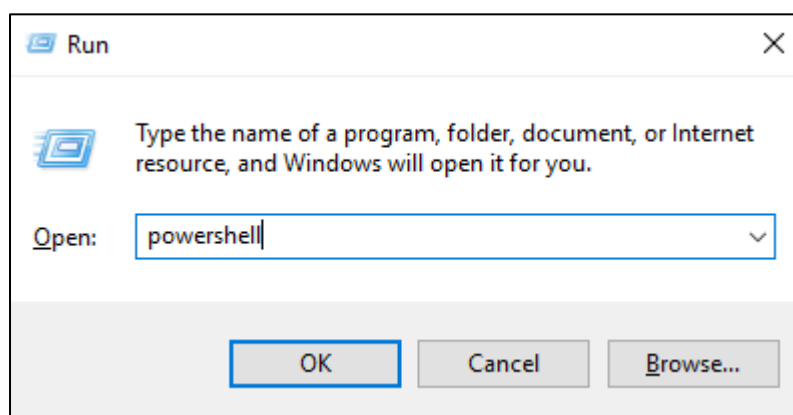


Figura 40. Ventana para ejecutar aplicaciones de Windows 10.

Se abrirá una consola.

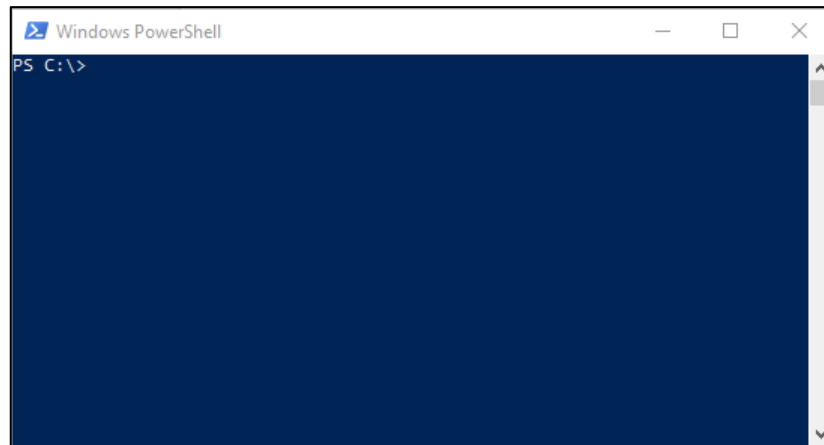


Figura 41. Vista de la consola de Windows 10.

Escribimos el siguiente comando en la consola y se instalará el gestor de paquetes.

```
python get-pip.py
```

3. Para instalar las dependencias en la misma consola escribimos los siguientes comandos.

Para instalar la librería matemática Numpy.

```
pip install numpy
```

Para instalar la librería gráfica.

```
pip install Pillow
```

Para instalar OpenCV.

```
pip install opencv-contrib-python
```

Para instalar XLSXWriter.

```
pip install XlsxWriter
```

MacOS

1. Nos dirigimos a la página web oficial <http://www.python.org> y descargamos la última versión de Python 3 64bit (Actualmente 3.7.3), hacemos doble clic sobre el archivo descargado y seguimos las instrucciones del instalador.

La última versión a fecha 22 de Mayo de 2019 es la siguiente.

<https://www.python.org/ftp/python/3.7.3/python-3.7.3-macosx10.9.pkg>

2. Instalación de PIP (Instalador de paquetes para Python).

Abrimos un “terminal” dirigiéndonos a Aplicaciones > Utilidades > y hacemos doble clic en Terminal. Se abrirá una ventana.

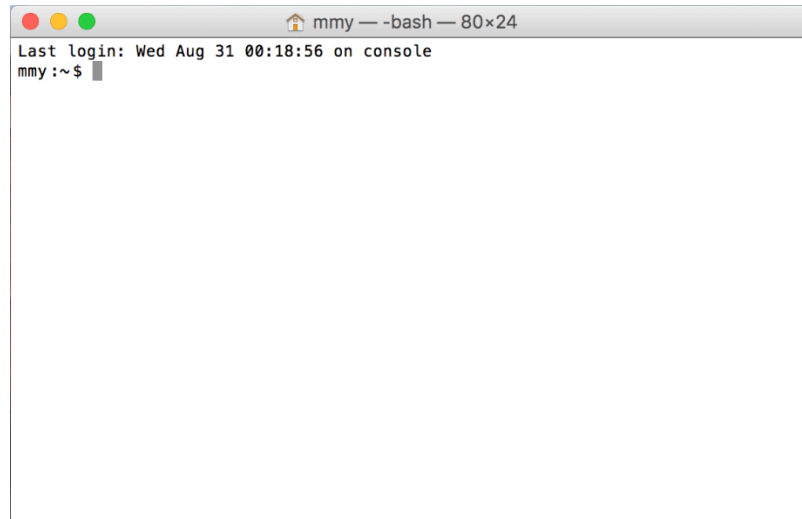


Figura 42. Vista del terminal de MacOS.

Escribimos el siguiente comando en el terminal y se instalará el gestor de paquetes (es probable que soliciten la clave del usuario administrador).

```
sudo easy_install pip
```

3. Para instalar las dependencias en la misma consola escribimos los siguientes comandos.

Para instalar la librería matemática Numpy.

```
pip install numpy
```

Para instalar la librería gráfica.

```
pip install Pillow
```

Para instalar OpenCV.

```
pip install opencv-contrib-python
```

Para instalar XLSXWriter.

```
pip install XlsxWriter
```

Linux

1. Abriremos un terminal (Varia según la distribución, por lo que se ruega consultar la documentación al respecto de la distribución que se esté usando).
2. Escribimos los siguientes comandos para instalar los requisitos para compilar el intérprete de Python (Es probable que debamos introducir nuestra contraseña de usuario).

```
sudo apt-get install build-essential checkinstall
```

```
sudo apt-get install libreadline-gplv2-dev libncursesw5-dev  
libssl-dev libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-  
dev libffi-dev zlib1g-dev
```

3. Descargamos el código fuente de Python 3.7.3, escribiendo los siguientes comandos.

```
cd /usr/src
```

```
sudo wget https://www.python.org/ftp/python/3.7.3/Python-  
3.7.3.tgz
```

4. Extraemos el código fuente de Python 3.7.3.

```
sudo tar xzf Python-3.7.3.tgz
```

5. Compilamos el intérprete y lo instalamos, (Puede llevar unos minutos desentendiendo del ordenador que se use).

```
cd Python-3.7.3
```

```
sudo ./configure --enable-optimizations
```

```
sudo make altinstall
```

6. Instalación de PIP (Instalador de paquetes para Python).
Escribimos el siguiente comando en el terminal y se instalará el gestor de paquetes (es probable que soliciten la clave del usuario administrador).

```
sudo apt install python3-pip
```

7. Para instalar las dependencias en la misma consola escribimos los siguientes comandos.

Para instalar la librería matemática Numpy.

```
pip install numpy
```

Para instalar la librería gráfica.

```
pip install Pillow
```

Para instalar OpenCV.

```
pip install opencv-contrib-python
```

Para instalar XLSXWriter.

```
pip install XlsxWriter
```

7.2 Instalación de la aplicación

La aplicación no necesita instalación por lo que podemos descomprimir el archivo “Zip” en el directorio que prefiramos.

7.3 Abrir la aplicación

Dentro del directorio de la aplicación debemos abrir haciendo doble clic el siguiente archivo.

Windows

```
run.bat
```

MacOS y Linux

```
run.sh
```

Otro método para abrirla es usar este comando en una consola o terminal dentro del directorio de la aplicación.

```
python main.py
```


7.4 Flujo de trabajo

Para facilitar su comprensión, este manual se ha estructurado siguiendo el flujo de trabajo que se debe seguir en una sesión del experimento.

7.5 Nuevo experimento y carga de experimentos

La aplicación se abre por defecto en la pestaña “File”.

Nuevo experimento

Introducimos un nombre para el experimento. Hacemos clic en el botón “New”. Se abrirá una ventana para seleccionar el directorio donde guardar el nuevo experimento. Haciendo clic en “Aceptar” estará creado.

Carga de un experimento

Para cargar un experimento anterior, tan solo debemos hacer clic en el botón “Load” de la pestaña “Experiment”, se abrirá una ventana para seleccionar el archivo “mzl” del experimento. Una vez seleccionado hacemos clic en “Abrir” y la carga del proyecto estará completa.

7.6 Configuración del experimento

En la pestaña “Experimento” de la Figura 43. En la parte derecha, estableceremos animales y los intentos o *trial*.

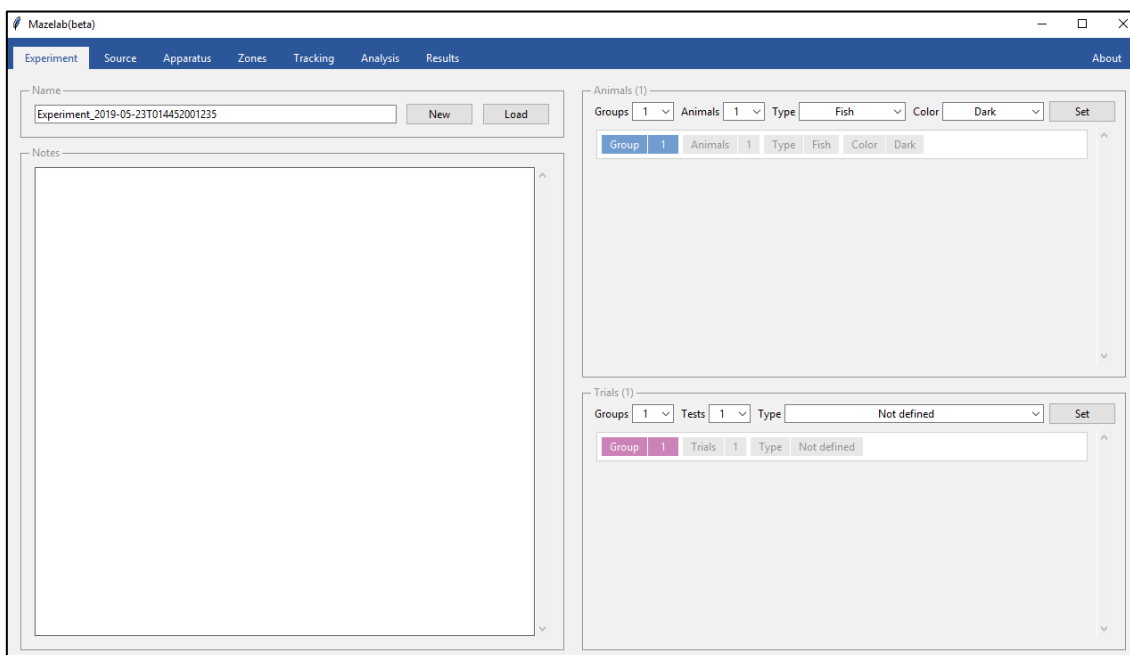


Figura 43. Vista de la pestaña “Experiment”.

En el apartado “Animales” de la Figura 44, establecemos cuantos animales van a formar parte del experimento, sus características y como queremos que estén distribuidos.

Cuando, hayamos seleccionado cada apartado presionamos “Set” y se crearan todos los grupos de animales con sus respectivas características.

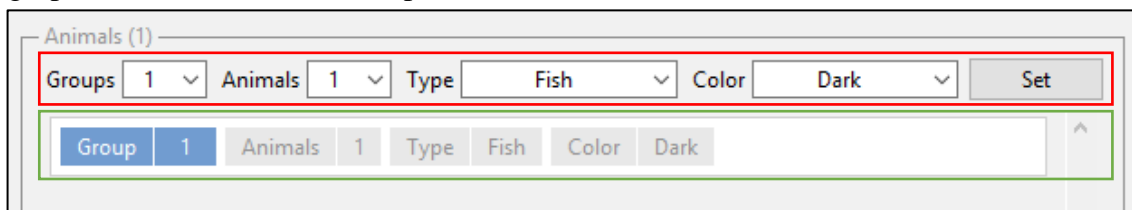


Figura 44. Detalle de los controles para selección de los grupos/animales.

En el apartado “Trials” de Figura 45, establecemos cuantos intentos por animal, grupos y tipo. Cuando, hayamos seleccionado cada apartado presionamos “Set”.

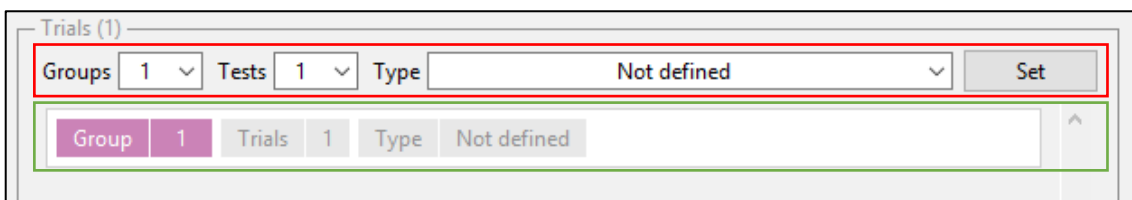


Figura 45. Detalle de los controles para selección de los grupos/intentos.

Si necesitamos cambiar algún parámetro debemos seguir los pasos anteriores.

AVISO: Si se vuelven a seleccionar se borrarán todos los resultados y asignaciones de fuentes de video que existan, ya que estamos cambiando los elementos que definen el experimento.

7.7 Añadir y asignar fuentes de video

En la pestaña “Sources” de la Figura 46, en la parte derecha, añadimos las fuentes de video que usaremos en el experimento y en la parte izquierda seleccionamos una prueba o *test* y seguidamente una fuente de video.

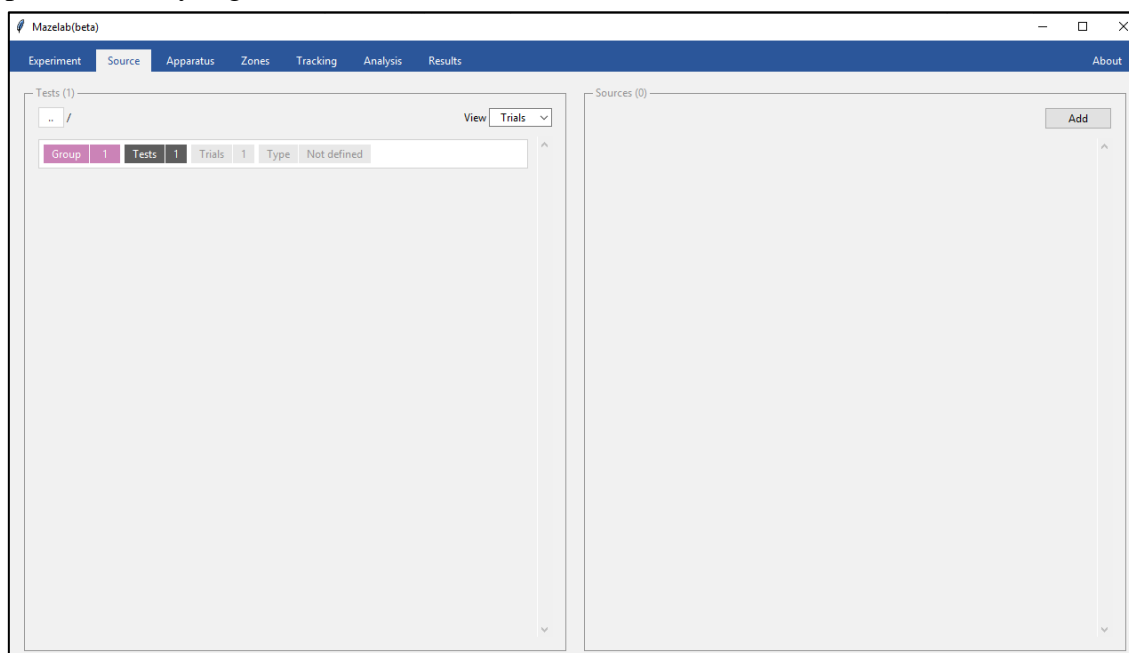


Figura 46. Vista de la pestaña “Sources”.

Veamos paso a paso como realizar todas estas acciones.

Añadir una fuente de video

Las fuentes de video como cámaras no hace falta añadirlas, ya que aparecerán automáticamente al ser detectadas por la aplicación al inicio de las misma.

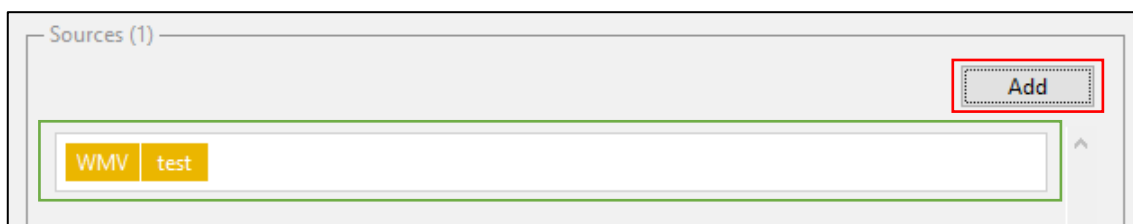


Figura 44.- Vista de la pestaña “Sources”.

Para añadir uno o varios archivos de video debemos hacer clic en el botón “Add” (Figura-44). Se abrirá un diálogo donde podremos seleccionar uno o varios archivos de video y pulsaremos “Aceptar”.

Una vez añadidos, se mostrarán todos los archivos de video y cámaras en el listado inferior con el formato “Extensión - Nombre” si es un archivo o “RAW - Cámara” si es un dispositivo.

Uso del navegador de pruebas o test

En la Figura 47 se observa que en la parte derecha se encuentra el navegador de pruebas, su funcionamiento se asemeja al del explorador de archivos.

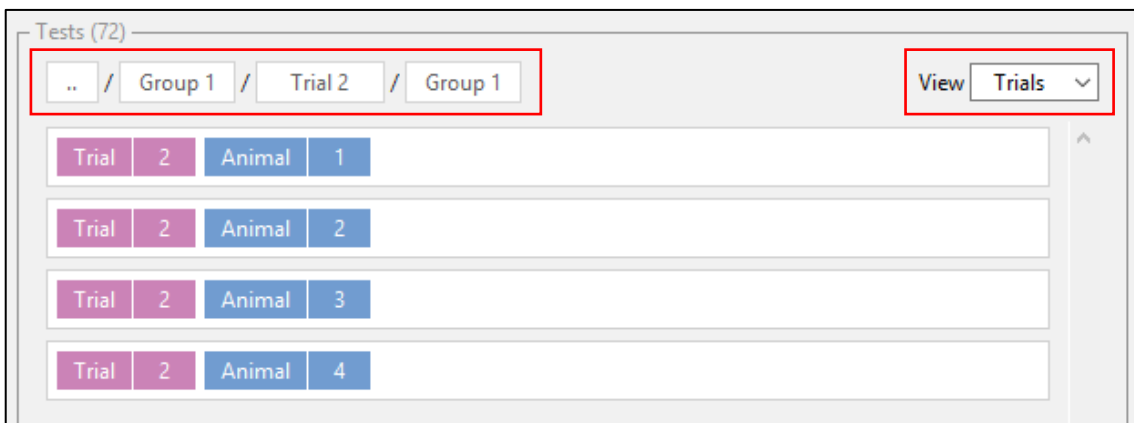


Figura 47. Vista del navegador de pruebas o test.

En la parte superior tenemos dos elementos, en **la izquierda se muestra la ruta** que seguimos al ir seleccionando entre grupos, intentos o animales.

A la derecha tenemos una lista desplegable que nos permite **cambiar la vista**.

- **Intentos o trial:** Se muestra una estructura del tipo:
 - Grupos de intentos o trial*
 - > *Intentos o trial*
 - > *Grupos de Animales*
 - > *Animales*

- **Animales:** Se muestra una estructura del tipo:
Grupos de animales
> *Animales*
> *Grupos de intentos o trial*
|> *Intentos o trial*

Para abrir un nuevo elemento del navegador debemos posicionar el ratón sobre el elemento que cambiara a “View” y hacer clic. Se muestra en la Figura 48.

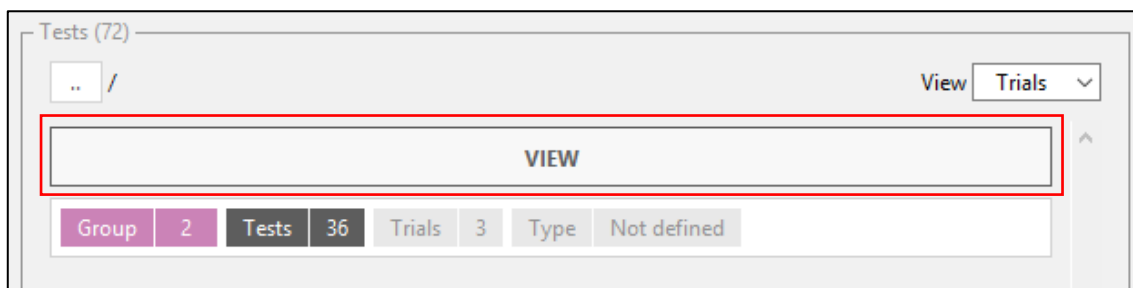


Figura 48. Acceso a un elemento del navegador de pruebas.

Asignar fuentes de video

Continuaremos seleccionando según deseemos hasta llegar al último nivel donde se mostrarán las pruebas o test que correspondan. Ver la Figura 49.

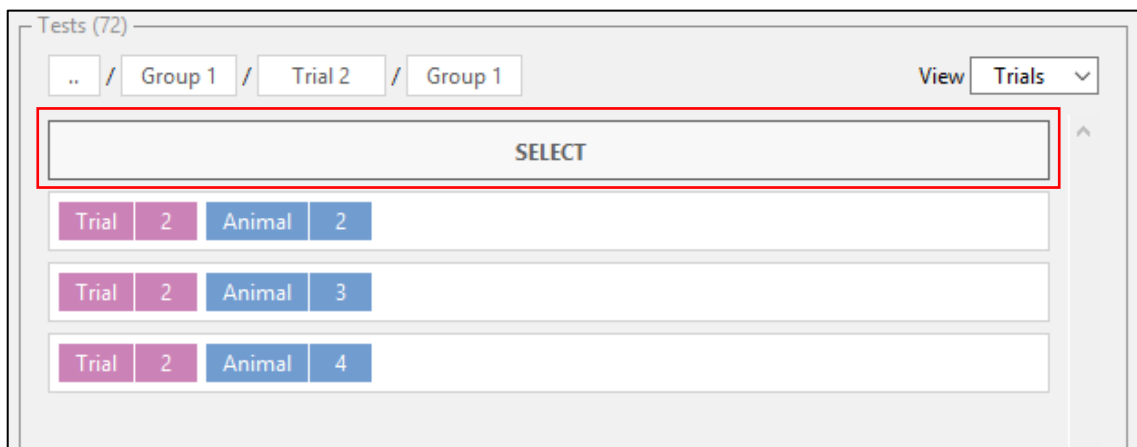


Figura 49. Selección de un elemento del navegador de pruebas.

Para seleccionar una prueba o test posicionamos el ratón sobre el elemento. Cambiará a “SELECT”, y haremos clic, cambiando entonces a “DESELECT” y quedará seleccionada.

Si quisiéramos, quitar la selección tan solo deberemos hacer clic de nuevo sobre el elemento como muestra la Figura 50.

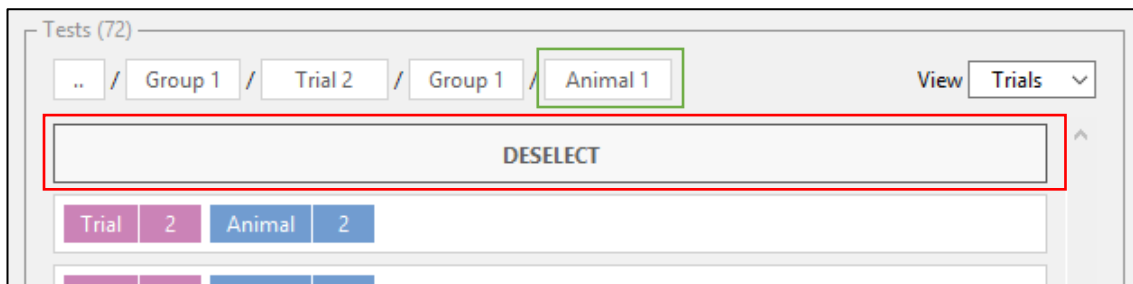


Figura 50. Prueba seleccionada.

Una vez lo tenemos seleccionado, debemos buscar en el listado de la parte derecha la fuente de video que deseamos asignar.

Para asignarla colocamos el ratón sobre el elemento. Éste cambiará a “SELECT” y haciendo clic se asignará y desaparecerá del listado derecho.

En ese momento la prueba que teníamos seleccionada mostrara la fuente que acabamos de asignarle.

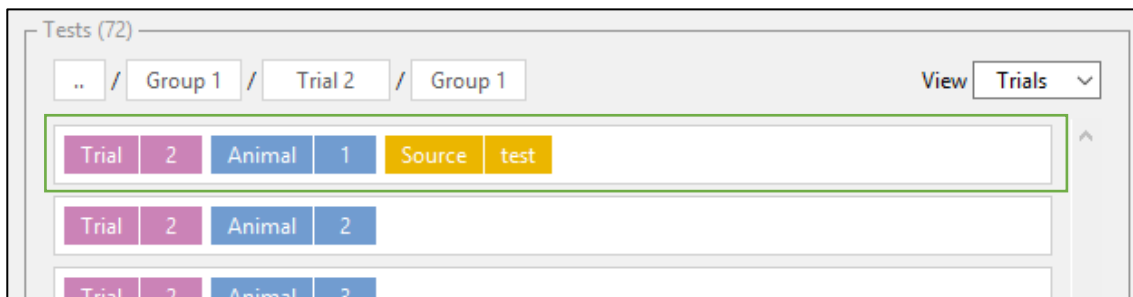


Figura 51. Prueba con fuente asignada.

Repetiremos este proceso, con todas las pruebas que desosemos analizar.

Borrar asignación

Si necesitáramos cambiar la asignación de una fuente o simplemente borrar la asignación, debemos colocar el ratón sobre la prueba que cambiará a “CLEAR”. Ver la Figura 52.



Figura 52. Borrado de la asignación de fuente.

En este momento si hacemos clic sobre ella, la asignación desaparecerá y la prueba queda sin asignar volviendo la fuente a mostrarse en el listado de la parte derecha.

Eliminar fuentes de video

Si deseamos eliminar una fuente de video es suficiente con posicionar el ratón sobre la fuente a eliminar y hacer clic en el botón “X” de la parte derecha. Consultar la Figura 53.



Figura 53. Borrado de una fuente de video.

7.8 Modelo digital del aparato

En la pestaña “Apparatus” de la Figura 54 debemos modelar el aparato sobre alguna de las fuentes de video que hemos el apartado anterior.

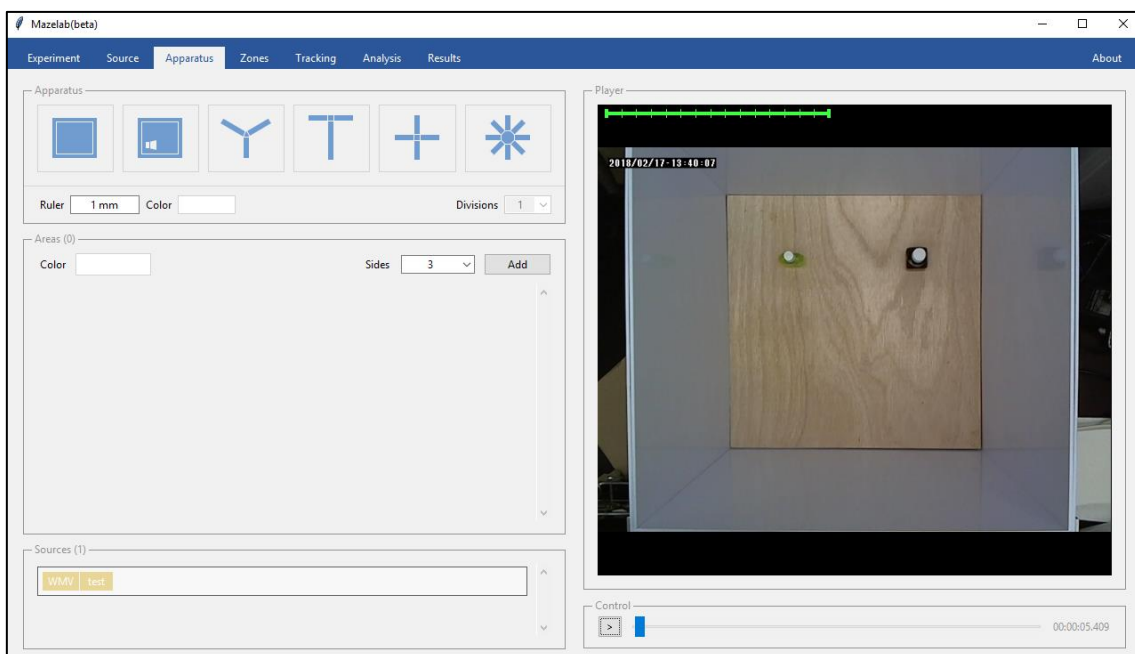


Figura 54. Vista de la pestaña “Aparato”.

Selección del aparato

En la parte superior izquierda tenemos un apartado llamado “Apparatus” de la Figura 55. En este apartado se muestran 6 posibles tipos de aparato.

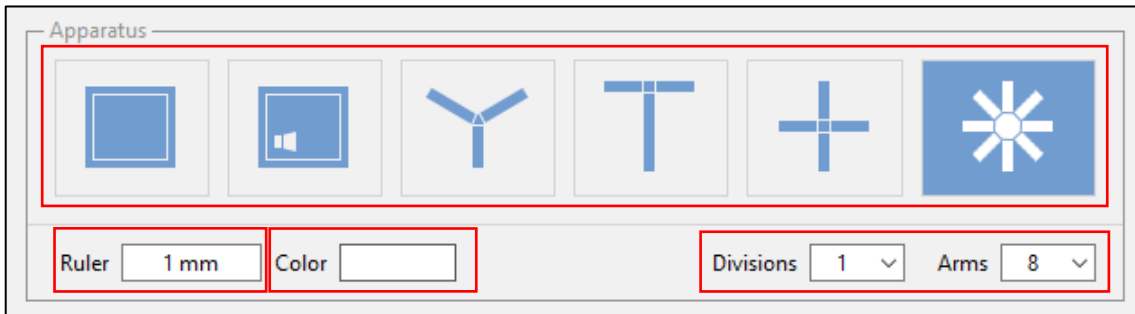


Figura 55. Selección del aparato, divisiones del aparato, brazos del aparato (Radial-Maze), regla y color.

Para seleccionar un aparato y añadirlo solo hay que posicionar el ratón sobre uno de ellos y hacer clic. Para borrar la selección haremos clic de nuevo.

Cuando se seleccione aparecerá en la parte derecha sobre la imagen de video. Ver la Figura 56.

Configuración del aparato

Para configurar el aparato, tan solo debemos establecer en cuantas partes queremos dividirlo y en el caso sólo del “Radial-Maze” (El primero por la derecha) podemos establecer el número de brazos.

El color del aparato es blanco por defecto, pero puede ser cambiado haciendo clic en la casilla blanca a la derecha del texto “Color” y seleccionar el que nos permita una mejor visualización.

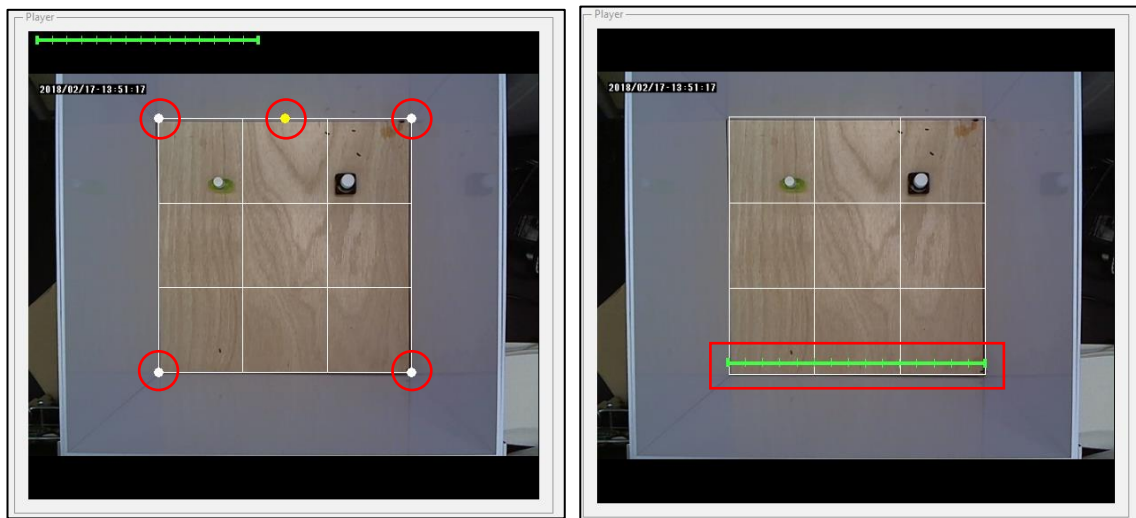


Figura 56. Izquierda: Visualización del aparato, puntos de tamaño y giro, Derecha: Regla configurada.

Tan solo nos queda posicionar y redimensionar el aparato sobre la imagen de video. Para ello usaremos los puntos blancos que permiten redimensionar el aparato libremente.

Para girarlo colocaremos el ratón sobre el punto amarillo, haremos clic y mantendremos pulsado el botón del ratón mientras lo movemos hasta conseguir la posición deseada.

Configuración de la regla

Para configurar la regla tan solo tenemos que posicionarla en la imagen sobre una zona del aparato cuya longitud física conozcamos, como muestra la Figura 56. Para finalizar escribiremos la longitud física de la regla en la casilla correspondiente “Ruler” del apartado “Apparatus” expresada en milímetros como muestra la Figura 55.

7.9 Áreas de interés

En la pestaña “Apparatus” de la Figura 54, en la parte izquierda tenemos el apartado “Áreas”. En este apartado podemos añadir áreas de interés que usaremos en la pestaña siguiente al definir las zonas.

Estas áreas, suelen posicionarse sobre objetos que en ocasiones el investigador introduce en el aparato alternativamente como parte de prueba de memoria.

Añadir un Área

Para añadir área tan solo debemos seleccionar el número de lados del polígono que dará forma al área y hacer clic sobre el botón “Add”. Ver la Figura 57.



Figura 57. Selección de área, tipo y color.

Una vez añadida podemos posicionarla, moverla y cambiar el tamaño en la visualización (sobre la imagen de video).

También es posible cambiar el color de la visualización facilitando el contraste con la imagen.

Eliminar un Área

Para eliminar un área tan solo debemos hacer clic en el botón “X” que se muestra sobre el elemento del listado cuando posicionamos el ratón sobre él.

7.10 Reproductor de video

El reproductor de video se compone del control, la visualización y de las fuentes.

El control es similar a cualquier otro reproductor de video. Permite pausar la reproducción haciendo clic sobre el botón de la parte izquierda y muestra el tiempo de video que se ha reproducido en la parte derecha.

En la zona central muestra una barra de desplazamiento, que indica la posición actual respecto a la duración total del video.

Si hacemos clic sobre la barra avanzaremos hacia adelante o hacia atrás y si mantenemos el botón izquierdo podemos hacer una búsqueda según la movamos.

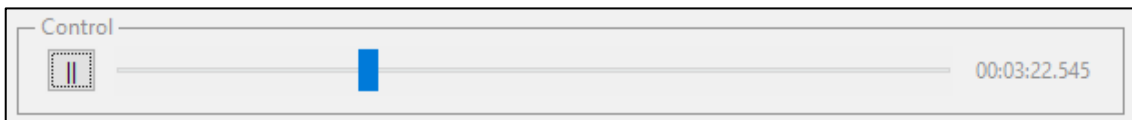


Figura 58. Vista del control del reproductor.

Abrir o cambia una fuente

Si necesitamos cambiar la fuente que se reproduce, tenemos que posicionar el ratón sobre el elemento a seleccionar, éste cambiara a “OPEN” como se ve en la Figura 59. Al hacer clic sobre él la fuente de video comenzara a reproducirse y queda seleccionada. La fuente en reproducción se muestra sombreada en gris.

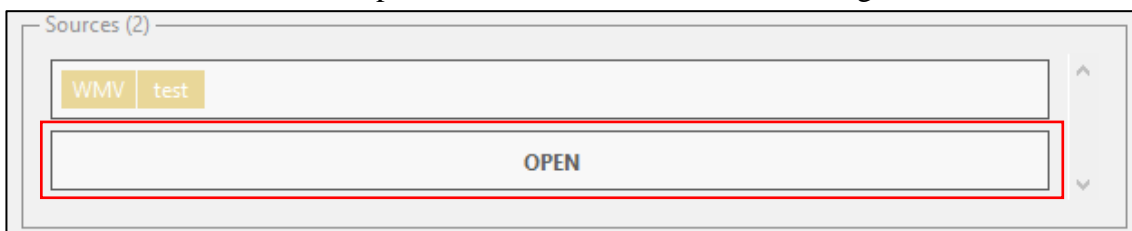


Figura 59. Selección de fuente en el reproductor.

7.11 Zonas

La pestaña “Zones” de la Figura 60, se compone en su parte izquierda el listado de zonas y los campos y botón para añadir una nueva zona. En la parte derecha se muestran las zonas del aparato y las áreas definidas en la pestaña “Apparatus”.

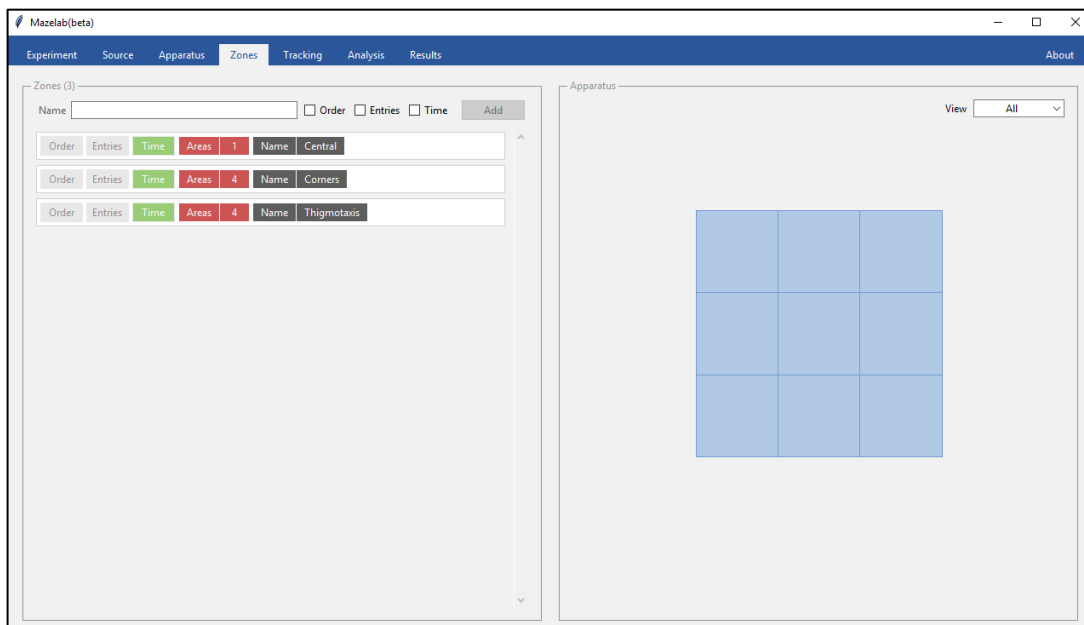


Figura 60. Vista de la pestaña de selección de zonas.

Añadir una nueva zona

Para añadir una nueva zona, debemos fijarnos en la parte superior izquierda de la Figura 61. En ella encontramos el campo nombre y tres cajas de selección “Orden”, “Entradas” y “Tiempo”, que representan las métricas que queremos asociar a la nueva zona.

Una vez seleccionadas según nuestras necesidades y con un nombre en la casilla correspondiente hacemos clic en el botón “Add” y automáticamente la lista de zonas se refresca y muestra la nueva zona.

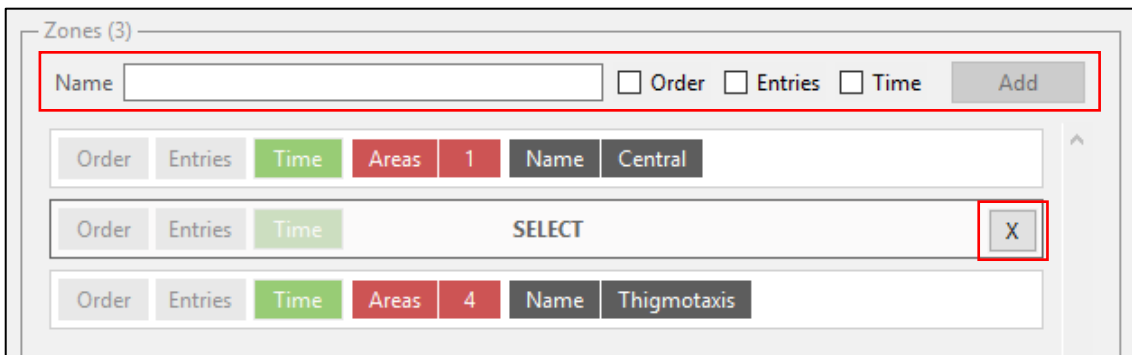


Figura 61. Controles de zona y listado de zonas.

Eliminar una zona

Para eliminar una zona, solo tenemos que situar el ratón sobre la zona a eliminar y hacer clic en el botón “X” en la parte derecha de la misma.

Cambiar la selección de métricas en una zona

Para cambiar las métricas de una zona, debemos situar el ratón sobre la métrica a cambiar dentro de la zona como muestra la Figura 62. La métrica mostrará un borde de color negro.

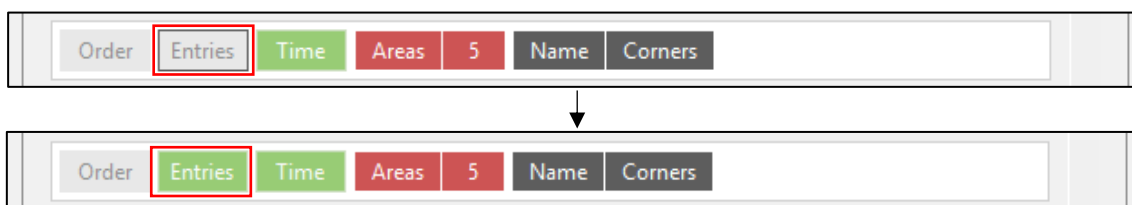


Figura 62. Cambio de configuración de las métricas de una zona.

Haciendo clic sobre la métrica cambiará a color verde indicando que está activa.

Si quisiéramos desactivarla bastaría con hacer clic otra vez y pasaría a tener un color gris que indica que esta desactivada.

Mostrar las áreas de una zona

Las áreas asociadas a una zona se muestran al situar el ratón sobre la zona. En la parte derecha se iluminan en color rojo las zonas asociadas.

Añadir / Eliminar áreas asociadas a una zona

Las áreas asociadas pueden añadirse o eliminarse de una zona situando el ratón sobre la zona a cambiar. La zona mostrará el mensaje “SELECT”. Ver la Figura 61. Si hacemos clic en la zona cambiará a “APPLY” como muestra la Figura 64.

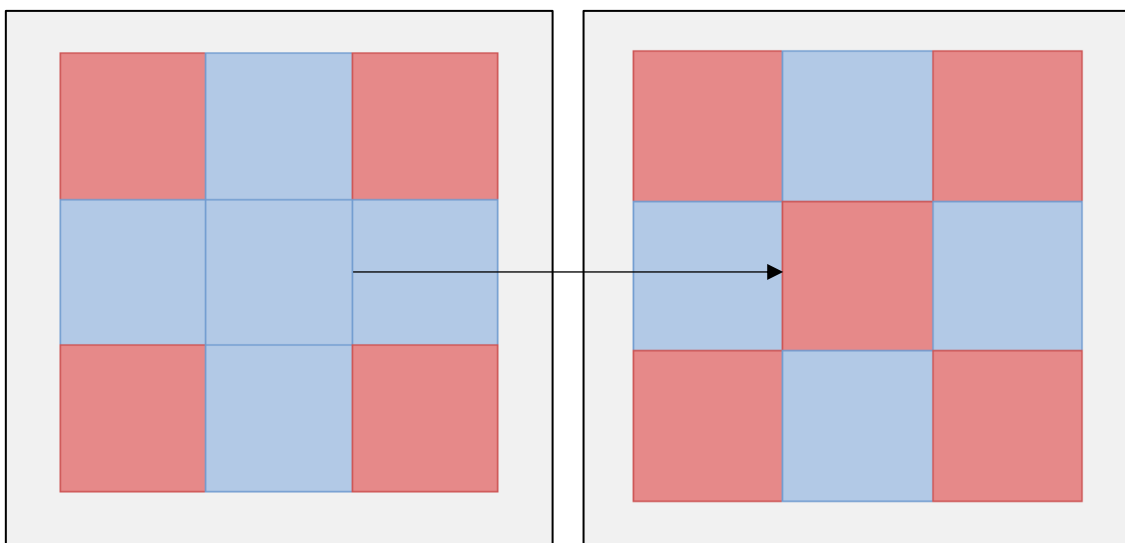


Figura 63. Vista de las áreas. Izquierda: Área central no está seleccionada, Derecha: Área central seleccionada.

En este momento debemos fijarnos en la parte derecha. Las zonas asociadas tendrán color rojo y las no asociadas color azul.

Haciendo clic en un área, asociamos o eliminamos la asociación. Cambiará de color acorde a su estado Figura 63.

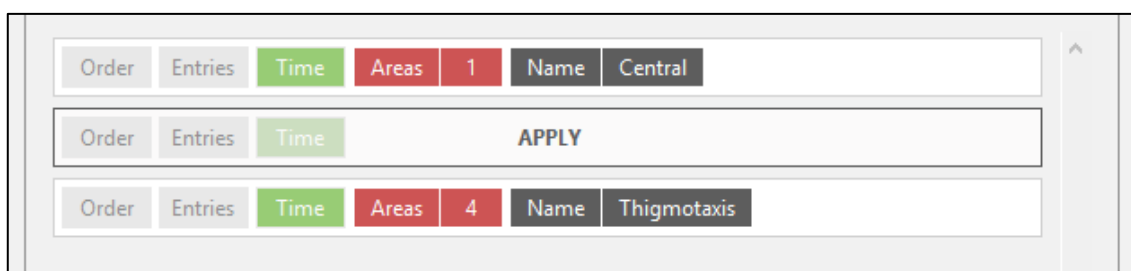


Figura 64. Guardado de selección de áreas de una Zona.

Para guardar los cambios en la asociación de las áreas a una zona, debemos hacer clic en la zona con el mensaje “APPLY”.

Selección de vista

En la parte derecha se muestran por defecto todas las áreas del aparato y todas las áreas creadas en la pestaña “Apparatus”.

Dado que es habitual que se solapen total o parcialmente, podemos seleccionar entre visualizar las áreas del aparato o las demás áreas. Para seleccionar la vista se debe seleccionar en la lista desplegable, como muestra la Figura 65.

De este modo se facilita enormemente la visualización y la selección de las áreas.



Figura 65. Selección de vista (Todos/Aparato/Áreas).

7.12 Configuración del sistema de seguimiento de objetos

La pestaña “Tracking” de la Figura 66, contiene en la parte izquierda, todos los controles del sistema de seguimiento por visión artificial de los objetos.

Contiene en la parte derecha un reproductor de video que además muestra en tiempo real los objetos detectados por el sistema de seguimiento.

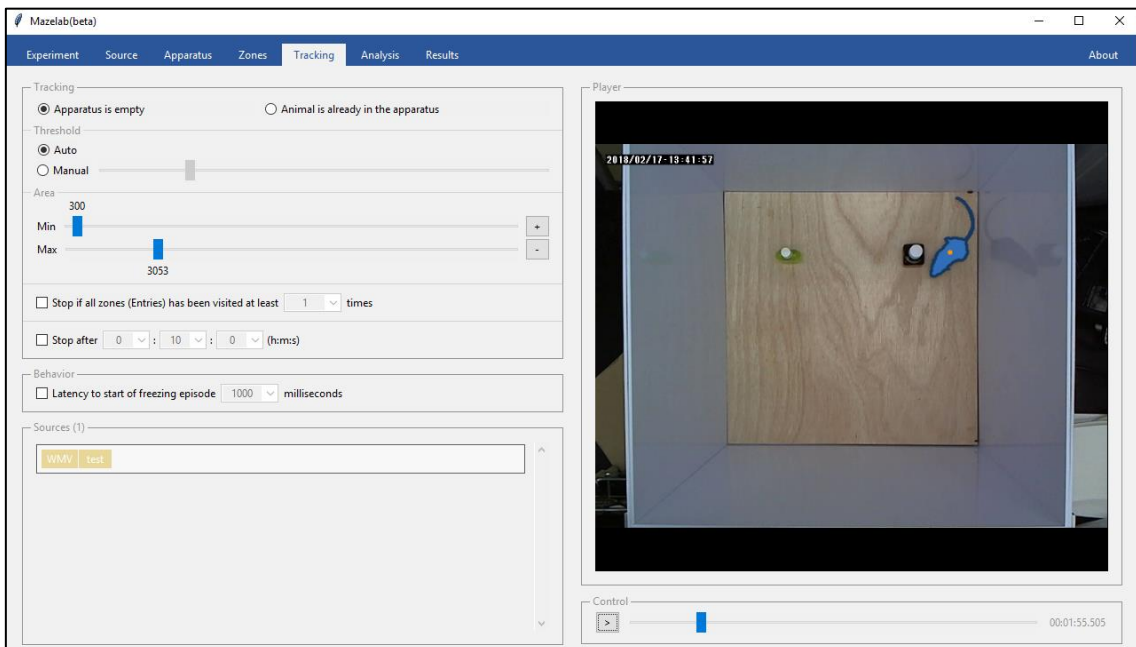


Figura 66. Vista de la pestaña “Tracking”.

Selección del algoritmo de seguimiento

En la parte superior derecha “Tracking”, tenemos un control que nos permite cambiar en cualquier momento entre los dos algoritmos que están implementados.

- **“Apparatus is empty”**: Se refiere al algoritmo de diferencia entre imágenes, por lo que se necesita que al comienzo del video el animal no esté dentro del aparato, convirtiéndose esa imagen inicial en el fondo de referencia.
- **“Animal is already in the apparatus”**: se refiere al algoritmo dinámico como el de mediana temporal que genera con las últimas imágenes el fondo para

detectar el animal. En general es la opción recomendable cuando el animal ya está dentro del aparato al comienzo del video.

Selección del threshold

Esta opción solo es necesaria si el algoritmo seleccionado es “Apparatus is empty” y por defecto es automática. Se recomienda usar el modo manual sólo si la detección no es correcta.

En el modo manual, la barra situada a la derecha se activa y se puede seleccionar un valor entre 0 y 255.

Se recomienda moverlo poco a poco en función de los resultados mostrados en el reproductor.

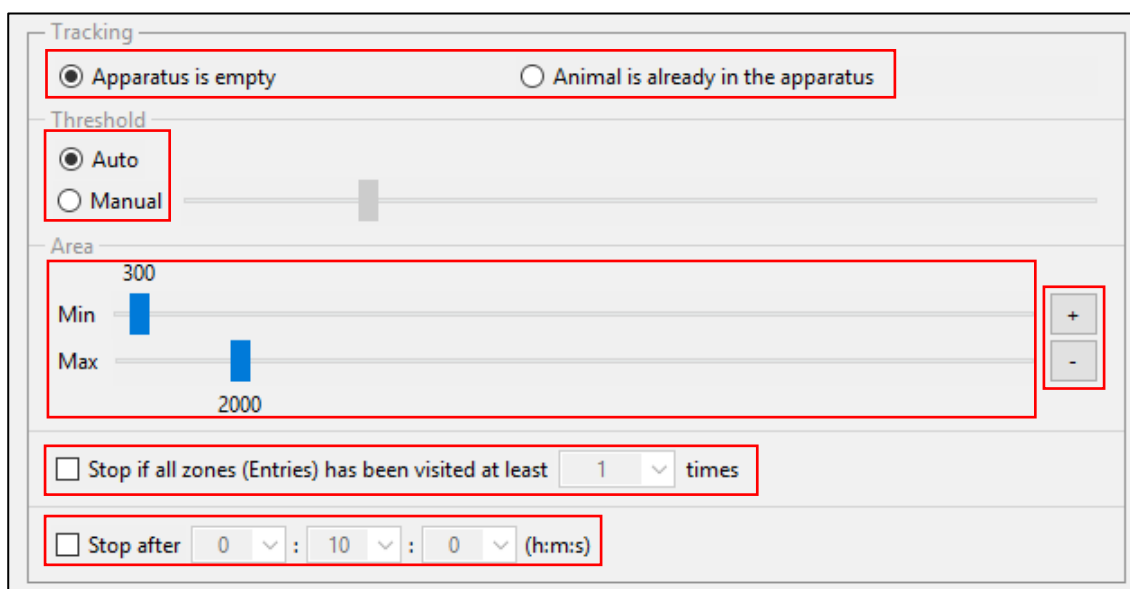


Figura 67. Vista de la configuración del sistema de seguimiento.

Configuración del tamaño del área

Esta opción consta de dos barras de desplazamiento enlazadas. Una selecciona el tamaño mínimo para el área del objeto a detectar y la otra el tamaño máximo.

Estos controles son necesarios para descartar el ruido de la imagen que pueden detectarse como objetos de manera errónea.

Los botones “+” y “-“, cambian la escala o “zoom” para poder seleccionar el tamaño con más precisión o para seleccionar rápidamente tamaños mayores.

Configuración de las condiciones de parada

La aplicación permite establecer condiciones de parada en base al tiempo o al número de entradas registradas en las zonas con dicha métrica activa. Ver la Figura 67.

Esto es muy útil cuando sólo se requiere que el animal visite determinadas zonas un determinado número de veces sin esperar a que se complete el tiempo de experimento.

Otra posibilidad es definir un tiempo límite para terminar la prueba. Esta es la opción más habitual.

Configuración del tiempo mínimo de inmovilidad

En ocasiones es necesario medir el tiempo que un animal está paralizado o detenido por completo durante la prueba.

Este control permite establecer un tiempo mínimo para considerar que el animal ha entrado en un periodo de inmovilidad.

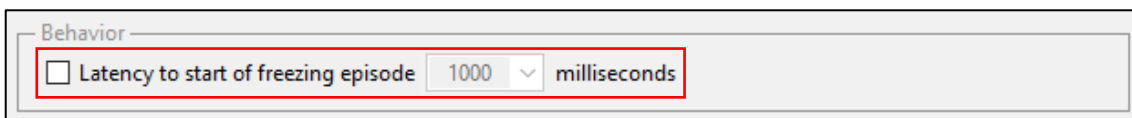


Figura 68. Vista de la configuración del sistema de seguimiento.

7.13 Análisis de las pruebas

En la pestaña “Analysis” de la Figura 69, encontramos en la parte izquierda el navegador de pruebas o test que ya conocemos de pestañas anteriores.

En la parte derecha, tenemos los controles para comenzar el análisis de una prueba o de todas las pruebas. También los controles para grabación y comienzo de una prueba con el dispositivo (Cámara) y un monitor donde se muestra la imagen del dispositivo en uso.

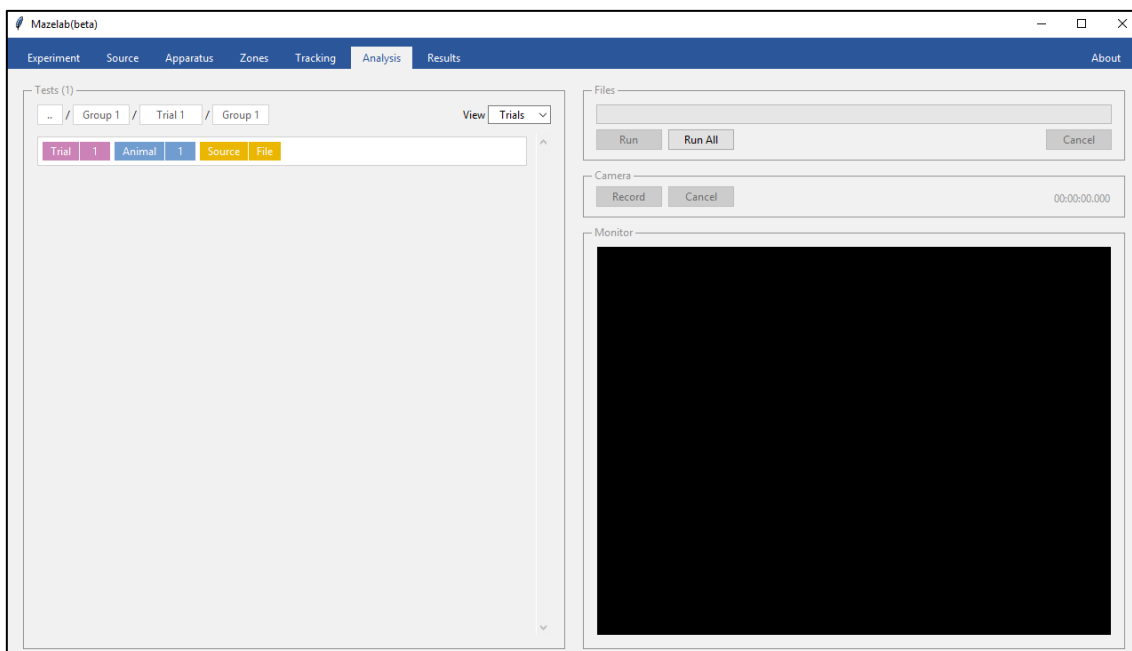


Figura 69. Vista de la pestaña de “Analysis”.

Analizar pruebas con archivos de video

En la parte superior derecha tenemos los controles para el análisis de las pruebas asociadas a archivos de video.

- **Análisis de una prueba:** Seleccionamos en el navegador de pruebas la prueba y hacemos clic en el botón “Run”. El análisis comenzará y la barra mostrara el progreso del análisis que terminará cuando la barra se complete.
- **Análisis de todas las pruebas:** si queremos analizar todas las pruebas que tengan archivos de video asociados tan sólo hay que hacer clic en “Run All” y el proceso comenzara terminando cuando la barra de progreso se complete.

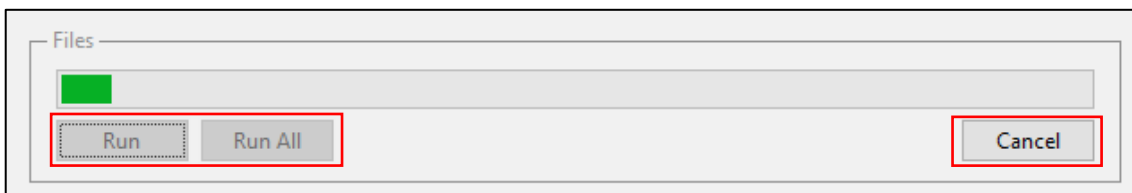


Figura 70. Controles para el análisis de archivos de video.

El análisis puede cancelarse en cualquier momento pulsando el botón “Cancel”.

7.14 Grabación de la prueba

Dentro de la pestaña “Analysis” de la Figura 71, en la parte central derecha, se encuentran los controles para comenzar y finalizar la grabación y análisis de una prueba.

El proceso comienza seleccionando en el navegador de pruebas de la parte izquierda la prueba que queremos comenzar.

Nota: Sólo las pruebas con un dispositivo (Cámara) asociado pueden ser grabadas.



Figura 71. Vista del monitor.

En el momento que se selecciona se muestra la imagen de video de la cámara en tiempo real en el Monitor. Ver la Figura 71.

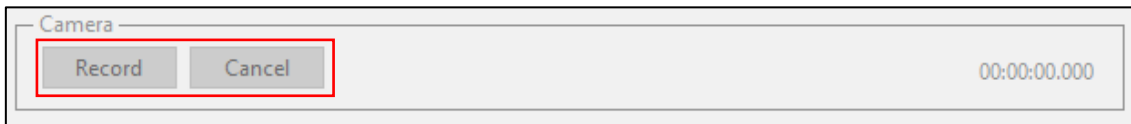


Figura 72. Controles para la grabación y análisis.

Cuando se esté preparado para comenzar la prueba, se hace clic en el botón “Record” que cambiará a “Stop”. En este momento si el animal no se encuentra dentro del aparato debemos meterlo dentro.

Una vez finalizado, retiramos al animal del aparato y presionamos el botón “Stop”. Si se quiere cancelar la prueba debe presionarse el botón “Cancel”.

7.15 Visualizar resultados

La pestaña “Results” de la Figura 73, permite consultar las métricas de cada prueba analizada y visualizar los mapas de calor y recorrido.

En la parte izquierda tiene un navegador de pruebas y en la parte derecha muestra las métricas de la prueba seleccionada.

Dispone de un botón para eliminar los resultados de la prueba seleccionada en ese momento y un botón para exportar a Excel todos los resultados.

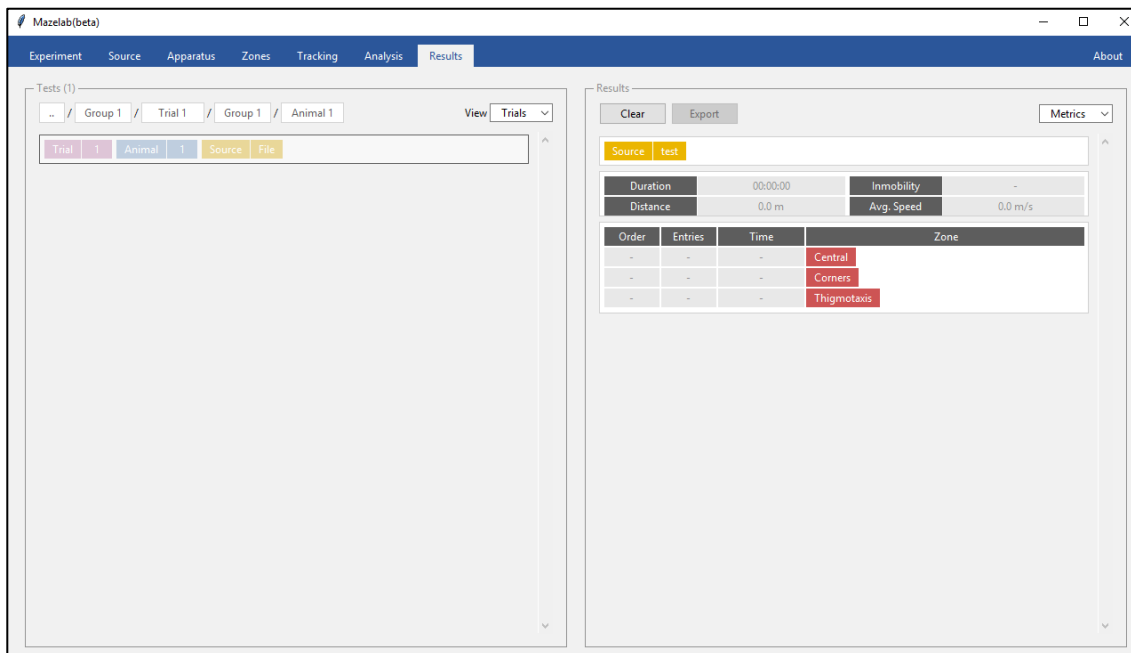


Figura 73. Vista de la pestaña “Results”.

Mostrar las métricas de una prueba

Para visualizar los resultados es necesario seleccionar la prueba desde el navegador de pruebas haciendo clic en la prueba, ésta queda seleccionada y sus resultados se muestran en la parte derecha. Ver la Figura 74.

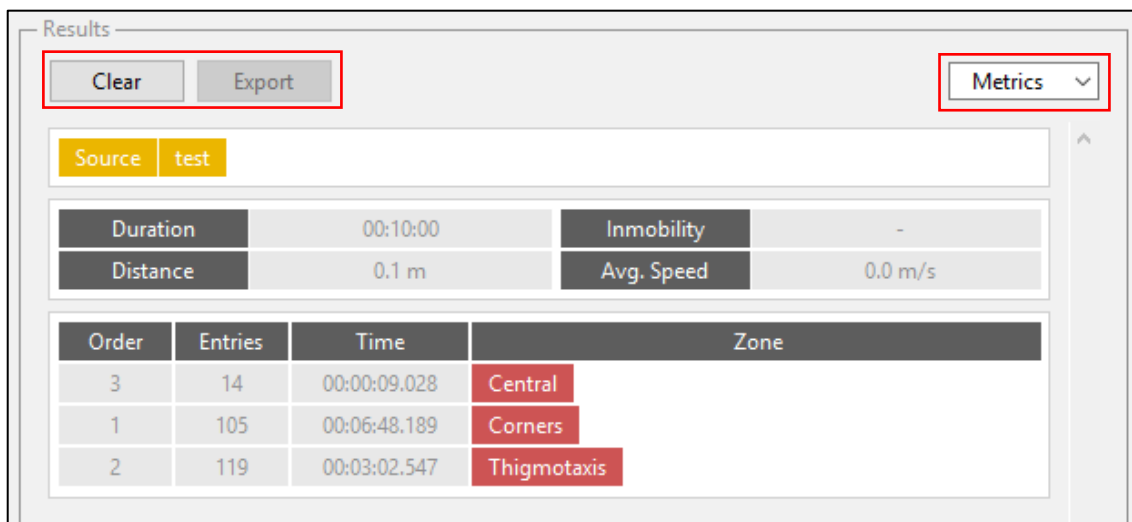


Figura 74. Vista de los resultados y controles.

Mostrar los mapas de una prueba

Para mostrar los mapas de calor y recorrido alternativamente, se debe hacer clic en el desplegable de la parte superior derecha que permite elegir entre tres opciones:

- **Metrics:** Muestra las métricas
- **Heatmap:** Muestra el mapa de calor
- **Track:** Muestra el mapa del recorrido

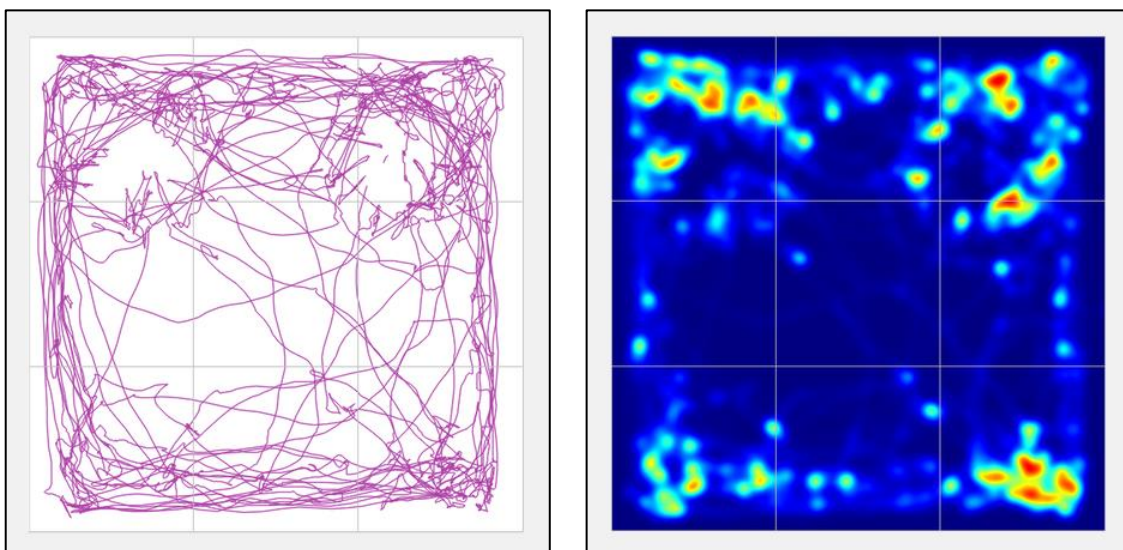


Figura 75. Izquierda: Mapa del recorrido, Derecha: Mapa de calor.

7.16 Exportar resultados

Para exportar los resultados del experimento, basta con hacer clic en el botón “Export” de la pestaña resultados como muestra la Figura 74.

Se abrirá una ventana en la que seleccionar el directorio donde guardar el archivo Excel.

8 Análisis económico

8.1 Análisis económico

Para la puesta en marcha de este proyecto se han definido unos requisitos económicos y técnicos.

Los requisitos económicos se dividen en los siguientes apartados.

- Coste del material
- Coste del personal

Coste del material

Considerando los ordenadores que se han usado en el desarrollo del proyecto y pruebas. El coste mínimo actual del material es el siguiente si se adquiriera hoy.

- **Modelo:** Dell 3020 SFF
- **CPU:** Intel Core i5 4590 3.30 Ghz / 3.70 Ghz
- **Memoria:** 8 GB RAM DDR3 1600 Mhz
- **Disco:** 128 GB SSD
- **Sistema:** Windows 10 64 bit / Linux Ubuntu 18.04 LTS 64 bit
Coste: 345€ (Licencia Windows 10 Home incluida)
- **Modelo:** Apple Macbook
- **CPU:** Intel Core 2 Duo 2.26 Ghz
- **Memoria:** 4 GB RAM DDR3 1333Mhz
- **Disco:** 320 GB HDD
- **Sistema:** MacOS High Sierra
- **Cámara:** iSight 640x480 30 Fotogramas/segundo
Coste: 245€

Coste del personal

Primero debemos especificar la duración del proyecto en toda su extensión.

Concepto	Horas
Documentación y búsqueda de información	80
Reuniones con el usuario	80
Análisis y pruebas de concepto	115
Desarrollo	442
Pruebas	38
Documentación	83
	838

Tabla 27. Horas de trabajo del personal.

Pasamos ahora a especificar el personal necesario y sus cualificaciones.

- 1 Ingeniero de desarrollo de software

Según el último convenio vigente del año 2018 “[XVII Convenio colectivo estatal de empresas de consultoría y estudios de mercado y de la opinión pública](#)” y siguiendo las tablas salariales del ”Anexo I” (página 57.).

Concepto	Coste
Sueldo bruto (5,24 meses)	12.911,55 €
Retenciones por I.R.P.F.	1.839,86 €
Cuotas a la Seguridad Social	819,85 €
Sueldo neto (5,24 meses)	10.251,64 €
Tipo de retención sobre la nómina	14,25 %
Sueldo neto (5,24 meses)	1.373,20 €
Pagas extra (x1)	1.509,20 €

Tabla 28. Coste del personal.

Coste total de la ejecución material del proyecto

Con esta información obtenemos el coste total del desarrollo que incluye el coste material y el coste del personal.

Concepto	Coste
Coste material	590 €
Coste del personal	12.911,55 €
	13.501,55 €

Tabla 29. Coste total de la ejecución material del proyecto.

8.2 Plan de negocio

Finalmente, se plantea el plan de negocio del proyecto.

Nos encontramos ante una aplicación basada en librerías de código abierto y por lo tanto estamos obligados a proporcionar todo el trabajo realizado sin coste.

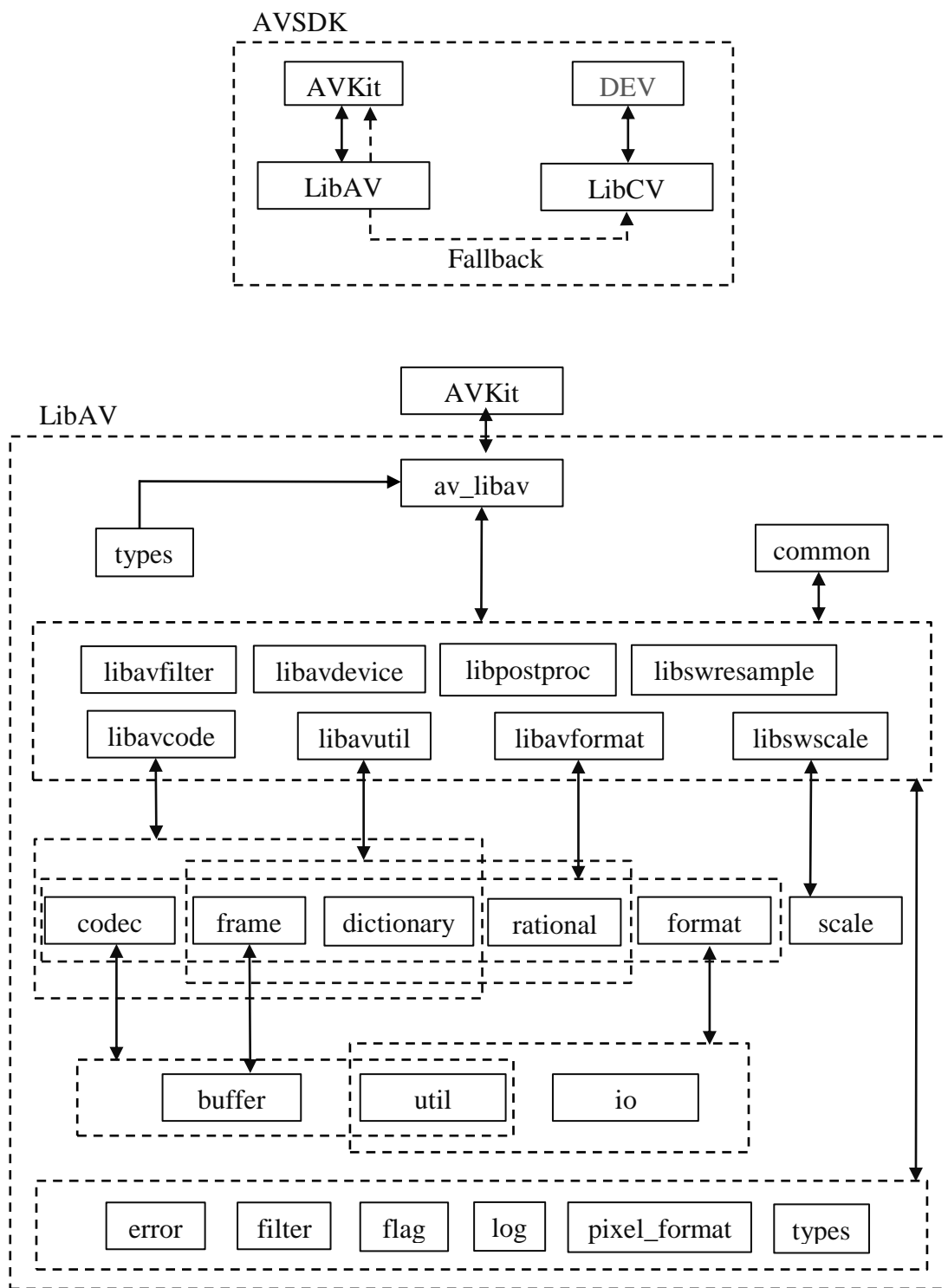
Por ello, la única vía de negocio que podemos explotar es proporcionar servicios de desarrollo de características específicas a laboratorios o individuales que lo requieran puntualmente o mediante un contrato de larga duración con mantenimiento incluido.

Para ello en un principio contaríamos con el mismo personal que ha formado parte del desarrollo de este proyecto y posibles incorporaciones en el futuro.

Aplicaremos un 20% adicional a nuestros costes de ejecución en relación al número de horas necesarias para cada desarrollo, ajustando este porcentaje según las necesidades y requisitos.

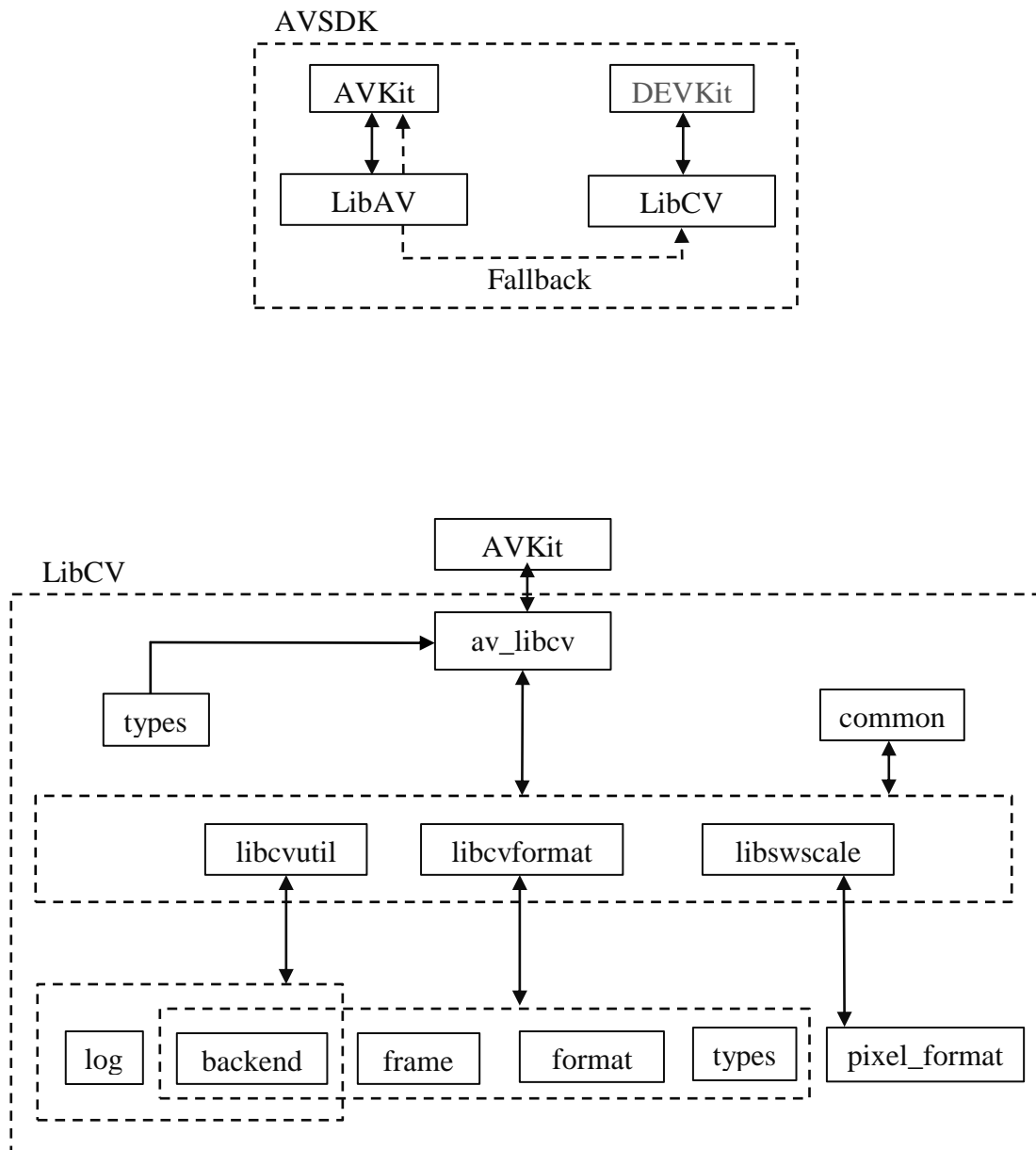
Apéndice 1

Estructura interna de la librería LibAV



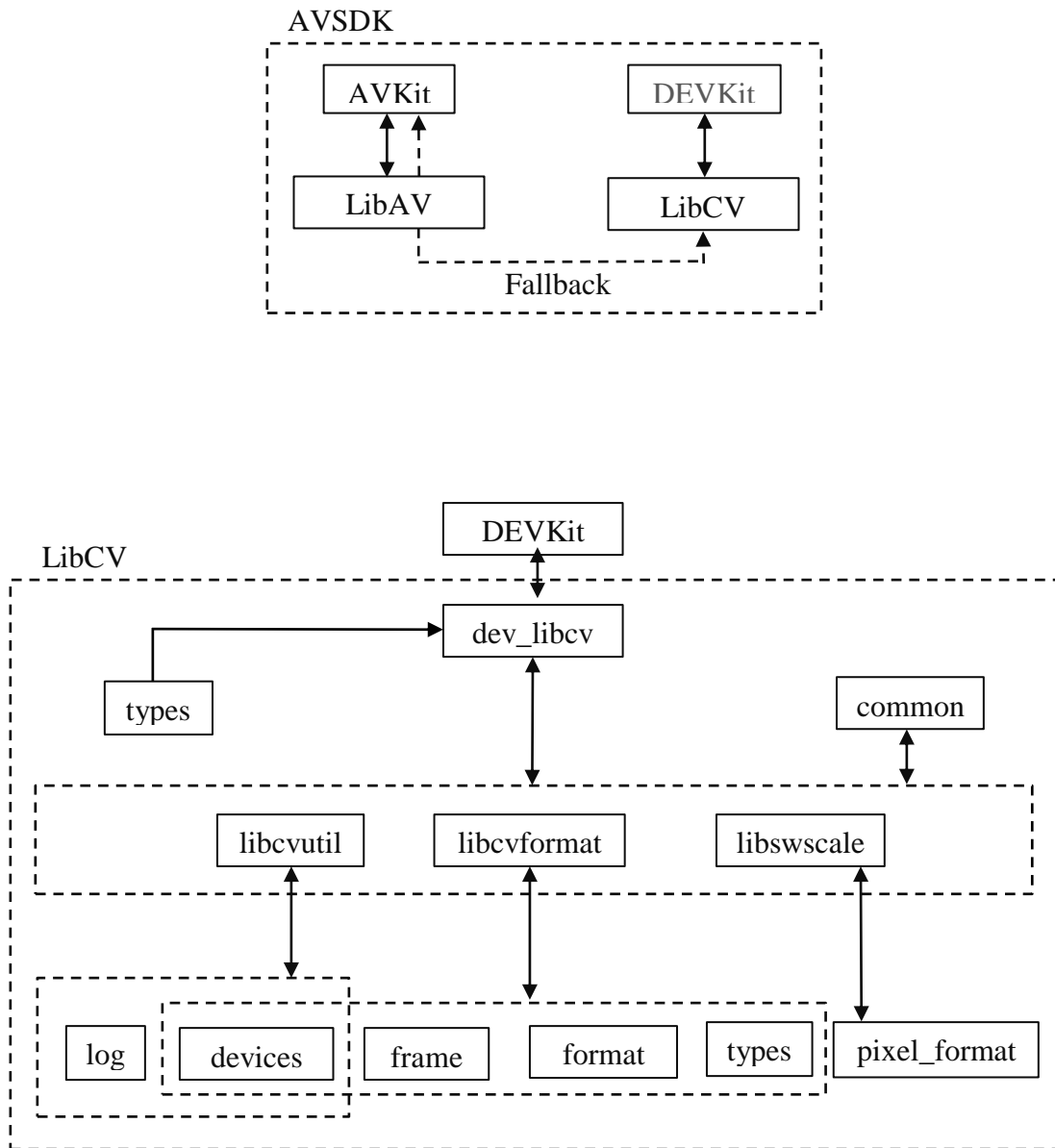
Apéndice 2

Estructura interna de la librería LibCV



Apéndice 3

Estructura interna de la librería DEVKit



Bibliografía

- [1] Wikipedia (2019, Mayo 31), “Neurociencia” [Online].
Disponibile: <https://es.wikipedia.org/wiki/Neurociencia>
- [2] Wikipedia (2019, Mayo 31), “T-Maze” [Online].
Disponibile: <https://en.wikipedia.org/wiki/T-maze>
- [3] Wikipedia (2019, Mayo 31), “Elevated plus maze” [Online].
Disponibile: https://en.wikipedia.org/wiki/Elevated_plus_maze
- [4] Wikipedia (2019, Mayo 31), “Radial arm maze” [Online].
Disponibile: https://en.wikipedia.org/wiki/Radial_arm_maze
- [5] Wikipedia (2019, Mayo 31), “Morris water navigation task” [Online].
Disponibile: https://en.wikipedia.org/wiki/Morris_water_navigation_task
- [6] FFmpeg Project (2019, Mayo 31), “FFmpeg Documentation 4.1” [Online].
Disponibile: <https://ffmpeg.org/doxygen/4.1>
- [7] OpenCV Team (2019, Mayo 31), “OpenCV Online Documentation 4.1” [Online].
Disponibile: <https://docs.opencv.org/4.1.0>
- [8] Dranger (2019, Mayo 31), “An ffmpeg and SDL Tutorial” [Online].
Disponibile: <http://dranger.com/ffmpeg/tutorial01.html>
- [9] Wikipedia (2019, Mayo 31), “Filtro de Kalman” [Online].
Disponibile: https://es.wikipedia.org/wiki/Filtro_de_Kalman
- [10] Geeksforgeeks (2019, Mayo 31), “How to check if a given point lies inside or outside a polygon?” [Online]
Disponibile: <https://www.geeksforgeeks.org/how-to-check-if-a-given-point-lies-inside-a-polygon>
- [11] Wikipedia (2019, Mayo 31), “Point in polygon” [Online].
Disponibile: https://en.wikipedia.org/wiki/Point_in_polygon
- [12] Hongjun Choi, Mingi Kim, Onseok Lee “An extended Kalman filter for mouse tracking”, Medical & Biological Engineering & Computing, vol. 56, no 11, pp. 2103-2123, Mayo 2018.
- [13] James Teow (2018, Mayo 4), “Understanding Kalman Filters with Python” [Online].
Disponibile: <https://medium.com/@jaems33/understanding-kalman-filters-with-python-2310e87b8f48>
- [14] Effbot (2005, Noviembre), “An introduction to tkinter” [Online].
Disponibile: <http://effbot.org/tkinterbook>

[15] John W. Shipman (2013, Diciembre 31), “Tkinter 8.5 reference: a GUI for Python” [Online].

Disponible: <http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html>

[16] John MacNamara (2019, Mayo 31), “XlsxWriter documentation 1.1.8” [Online].

<https://xlsxwriter.readthedocs.io/contents.html>

[17] Ministerio de Empleo y Seguridad Social (2018, Marzo 6), “XVII Convenio colectivo estatal de empresas de consultoría y estudios de mercado y de la opinión pública”, B.O.E., no 57, pp. 26951-26981.

[18] Calculadoraconversor.com (2019, Mayo 22), “Calculadora sueldo neto 2019” [Online].

<https://www.calculadoraconversor.com/calculadora-sueldo-neto>

Glosario de términos

Glosario de definiciones

Background subtraction Conjunto de técnicas de visión artificial para la detección de objetos en primer plano.

Background subtractor Técnica basada en la diferencia de dos imágenes (Referencia o fondo e Imagen) para extraer los objetos de primer plano.

Background subtractor CNT Técnica basada en la definición dinámica del fondo de un conjunto de imágenes para extraer los objetos de primer plano.

Binding Ligadura o referencia. En informática hace referencia a la ligadura de una o varias funciones desde un lenguaje distinto al usado para definir dichas funciones.

BGR Hace referencia a la codificación de los canales de color de una imagen. B: Azul, G: Verde y R: Rojo, en ese orden.

Controlador En el patrón de diseño Modelo-Vista-Controlador, hace referencia al conjunto de funciones que se encargan de gestionar la lógica de la aplicación y es nexa entre la Vista y el Modelo.

CTypes Librería para lenguaje Python que incluye definiciones para el lenguaje C/C++ y permite realizar llamadas desde un código programado en Python a código programado en C/C++.

Detección de primer plano También denominado Background subtraction, es el conjunto de técnicas de visión artificial para la detección de objetos de primer plano

Fallback Método de contingencia que permite recuperar el control o funcionalidad ante un error previsto.

FFMPEG Conjunto de librería de código abierto que permiten la decodificación de casi cualquier formato o códec de video.

Frame Fotograma. Cada una de las imágenes que conforman un video.

Kalman Filtro Kalman. Inventado por Rudolf, E. Kalmán en 1960. Algoritmo recursivo con múltiples aplicaciones en tecnología, como guía, navegación o control de vehículos.

MeasurementMatix En el filtro Kalman es la matriz que indica la relación entre mediciones y el vector de estado al momento k , en el supuesto ideal de que no hubiera ruido en las mediciones.

Modelo En el patrón de diseño Modelo-Vista-Controlador, hace referencia al conjunto de funciones que se encargan de gestionar los datos de una aplicación.

Morris Water Maze Desarrollado en 1984 por Richard Morris, es un laberinto acuático de forma circular que se utiliza para el aprendizaje espacial en ratas y ratones.

Neurociencia Campo de la ciencia que estudia el sistema nervioso a todos los niveles y todos sus aspectos.

Neurocientífico Según la R.A.E. especialista en neurociencia.

Numpy Librería para computación científica de Python.

OpenCV Librería de código abierto para el desarrollo de aplicaciones basadas en visión artificial.

Openfield Desarrollado por Calvin S. Hall es un laberinto usado en pruebas con ratas o ratones para evaluar los niveles de actividad, ansiedad y deseos de exploración del animal.

Plus-Maze También conocido como Elevated Plus Maze, es un laberinto que consta de cuatro brazos para su uso en experimentos con roedores y evaluar la ansiedad o el uso de compuestos ansiolíticos entre otros.

ProcessNoiseCov En el filtro Kalman es la matriz de covarianza del ruido de las mediciones

Radial-Maze Diseñado por Oton y Samuelson en 1976, es un laberinto originalmente de ocho brazos y usado en experimentos con ratas para evaluar la memoria y medir el aprendizaje espacial.

RGB Hace referencia a la codificación de los canales de color de una imagen. R: Rojo, B: Azul, G: Verde, en ese orden.

Segmentación Conocido como Segmentación de imagen es el proceso de dividir una imagen digital en varias partes (grupos de píxeles) u objetos.

Super-pixel Conjunto de píxeles adyacentes que contienen niveles de gris parecidos o color.

T-Maze Laberinto en forma de T usado en experimentos con roedores para el estudio de la memoria cuando se les aplica ciertos estímulos.

Test En el ámbito de este proyecto, representa cada prueba realizada a un animal en un aparato, Normalmente coinciden número de Test y número de videos.

Threshold Valor umbral usado en visión artificial por un grupo de algoritmos para separar los píxeles de una imagen que no interesan.

Tigmotaxia Tendencia o preferencia de algunos animales a permanecer en contacto con superficies de objetos.

Tkinter Librería de Python para la creación de interfaces de usuario.

TransitionMatrix En el filtro Kalman es la matriz de transición de estados.

Trial En el ámbito de este proyecto representa un intento o prueba.

Visión artificial Disciplina científica que usa métodos para procesar y analizar imágenes del mundo real tratadas mediante un ordenador.

Vista En el patrón de diseño Modelo-Vista-Controlador, hace referencia al interfaz de usuario.

Y-Maze Laberinto con forma de Y usado en experimentos con roedores para el estudio de la memoria cuando se les aplica ciertos estímulos. Muy parecido al T-Maze.

Glosario de acrónimos y abreviaturas

GB GigaByte.

Ghz Gigahercio.

Mhz Megahercio.

MVC Modelo-Vista-Controlador

PNG Portable Network Graphics

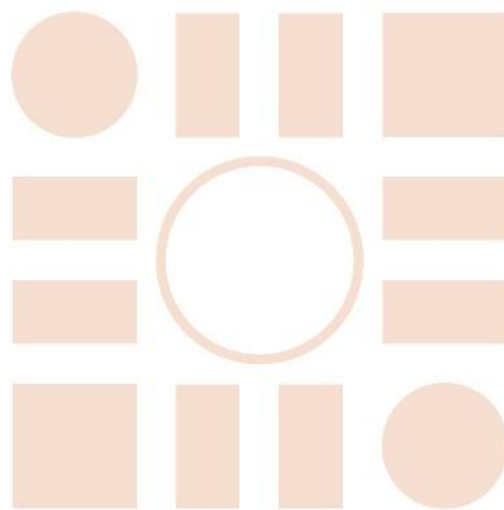
PTS Presentation Time Stamp

RAM Random Access Memory

SQL Structure Query Language

SSD Solid State Drive

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá