

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



PROYECTO FIN DE CARRERA

Ingeniería de Telecomunicación

**MONITORIZACIÓN SIMULTÁNEA DE
COMPORTAMIENTO Y SEÑALIZACIÓN BIOLÓGICA
ELÉCTRICA**

Alicia García Sastre

Junio 2016

**MONITORIZACIÓN SIMULTÁNEA DE
COMPORTAMIENTO Y SEÑALIZACIÓN BIOLÓGICA
ELÉCTRICA**

**Autor: Alicia García Sastre
Tutor: Pablo Varona Martínez**

Junio 2016

Grupo de neurocomputación biológica (GNB)

Dpto. de Ingeniería informática

Escuela Politécnica Superior

Universidad Autónoma de Madrid

RESUMEN

La monitorización simultánea del comportamiento de un sistema biológico y el registro de las señales eléctricas que genera su sistema nervioso brinda nuevas posibilidades en el contexto de la biomedicina y en concreto en neurociencia y neuroetología. Es por ello que se ha generado la necesidad de una herramienta para monitorizar, sincronizar y caracterizar de forma simultánea señales biológicas y de comportamiento registradas con electrodos y cámaras.

En este proyecto de fin de carrera se documenta el diseño y la creación de esta herramienta. Para ello se ha implementado una aplicación en C++ y que usa librerías de OpenCV y Qt para el sistema operativo Linux. Permite visualizar simultáneamente el vídeo y las señales electrofisiológicas, desplazarse de forma coordinada en la serie temporal y adaptarse a las distintas frecuencias de muestreo del vídeo y del registro. La herramienta se ha validado con registros de una neurona electrónica, registros de actividad de un pez eléctrico y con registros de video electroencefalografía.

PALABRAS CLAVE

Sincronización de vídeo y señalización biológica, electrofisiología, serie temporal, electroencefalografía (EEG).

ABSTRACT

Monitoring simultaneously the behavior of a biological system and the electrical signals that its own nervous system is generating opens new possibilities on the field of biomedicine and specifically on neuroscience and neuroetology. This is why there is a need for a tool to monitor, synchronize, and characterize in a simultaneous manner biological and behavior signals recorded with electrodes and cameras.

In this thesis it is explained in detail the design and creation of this tool. An desktop application in c++ has been implemented using OpenCV and Qt libraries, for the Linux operative system. It allows to visualize simultaneously video and electrophysiology signlas, navigate on a coordinated manner on the time series and adapt to the different sampling frequencies of the video and biological signal. The tool has been validated with the recording of and electronic neuron, the recording of electrical fish activity and the recording of video electroencephalography.

KEYWORDS

Video and biological signaling synchronization, biological signaling, electrophysiology, time series, electroencephalography (EEG).

Agradecimientos

A Danny, que me acompaña allá donde voy y es irremplazable. Tan vital, imaginativo y amable que es difícil describirlo con palabras.

A mis amigos de los comienzos, a Indri, Sofi y Nuria. Cuantas cosas geniales compartidas y cuan cerca os siento siempre aunque estemos lejos. A mis compañeros del club, que me enseñaron mundos enteros. A Daniel, Sergio, Andrea, Pablo, Javier, Nayeli, Álvaro y Antonio.

A mis amigos de la UAM, que son para toda la vida, un apoyo constante y un soplo de aire fresco, la vida es mejor con vosotros. A Javi, que es el responsable de que eligiera esta uni y me acompañó sin saberlo en mis primeros años hasta que nos conocimos en persona. A Bruno, las tardes siempre con café y la conversación inmejorable. A Raquel, Álvaro, Carolina, Eva, Tamara, Alfredo, Gustavo y Almudena.

Y muy especialmente a mi familia. A mi hermana Patricia, que me enseñó a programar entre tantas otras cosas, a Miriam que me alegra la vida, a mi madre que siempre está para mí. A mi padre, que me inspiró para ser ingeniera.

CONTENIDO

1	Introducción	18
1.1	Motivación.....	18
1.2	Objetivos.....	19
1.3	Organización de la memoria	21
2	Estado del arte.....	22
2.1	Cowlog.....	22
2.1.1	Descripción de objetivos y funcionalidad	22
2.1.1	Análisis de cumplimiento de requerimientos.....	24
2.2	AcqKnowledge.....	24
2.2.1	Descripción de objetivos y funcionalidad	24
2.2.2	Análisis de cumplimiento de requerimientos.....	25
2.3	Datagraph.....	26
2.3.1	Descripción de objetivos y funcionalidad	26
2.3.2	Análisis de cumplimiento de requerimientos.....	26
2.3.3	<i>WinEEG</i>	26
2.3.4	Descripción de objetivos y funcionalidad	26
2.3.5	Análisis de cumplimiento de requerimientos.....	27
2.4	Otras posibilidades	28
3	Diseño	28
3.1	Lenguajes de programación y librerías consideradas	28
3.1.1	Matlab	29
3.1.2	OpenCV	29
3.1.3	Qt.....	31

3.1.4 Diseño elegido	31
3.2 Estructura principal	32
3.2.1 Signal y slots	36
3.2.2 Carga de ficheros de video y señal biológica y reproducción	37
3.2.3 Reproducción	38
3.2.4 QTimer	39
3.2.5 Navegación	41
3.2.6 Sincronización	42
4 Desarrollo	43
4.1 Apertura y lectura de ficheros	43
4.1.1 Señal de video	44
4.1.2 Señal biológica	46
4.2 Reproducción, pausa y parada	49
4.3 Navegación	54
4.4 Sincronización	54
4.5 Información del programa y características de los ficheros cargados	55
4.6 Hardware utilizado en las pruebas	56
5 Integración, pruebas y resultados	57
5.1 Sincronización del video de una neurona eléctrica y su serie temporal	57
5.2 Registros simultáneos de EEG y de actividad ocular en vídeo	60
5.3 Monitorización en video del comportamiento del <i>Gnathonemus Petersii</i> y su actividad eléctrica	63
6 Conclusiones y trabajo futuro	71
6.1 Conclusiones	71
6.2 Trabajo futuro	71
Referencias	75

7 Glosario	77
8 Anexos.....	79
8.1 Presupuesto.....	79
8.2 PLIEGO DE CONDICIONES.....	80

ÍNDICE DE FIGURAS

FIGURA 2-1: EJEMPLO DE USO DE LA MONITORIZACIÓN EN VÍDEO DE COWLOG	23
FIGURA 2-2: EJEMPLO DE USO DE ACQKNOWLEDGE	25
FIGURA 2-3: EJEMPLO DE MONITORIZACIÓN DE SEÑALES DE EEG (32 CANALES) Y VIDEO CON WINEEG	27
FIGURA 3-1: PROTOTIPO DE REPRODUCCIÓN DE VÍDEO A PANTALLA COMPLETA PROGRAMADO CON FUNCIONES DE OPENCV	29
FIGURA 3-2: CÓDIGO DEL PROTOTIPO DE REPRODUCCIÓN DE VÍDEO CON LIBRERÍAS DE OPENCV	30
FIGURA 3-3: ESQUEMA DE DISEÑO DE LA VENTANA PRINCIPAL DEL PROGRAMA	32
FIGURA 3-4: BARRA DE MENÚ	32
FIGURA 3-5: CUADRO DE DIÁLOGO DE CARACTERÍSTICAS DEL VÍDEO	33
FIGURA 3-6: CUADRO DE DIÁLOGO DE CARACTERÍSTICAS DE LA SEÑAL BIOLÓGICA	33
FIGURA 3-7: SOBRE EL PROGRAMA	34
FIGURA 3-8: EJEMPLO DE MENSAJE DEN LA BARRA DE ESTATUS. EN ESTA OCASIÓN SE INFORMA AL USUARIO DE QUE SE ESTÁ LEYENDO UN FICHERO PARA PINTAR LA SEÑAL TEMPORAL..	35
FIGURA 3-9: EJEMPLO DE USO DE SIGNALS Y SLOTS	37
FIGURA 3-10: CUADRO DE DIÁLOGO PARA AÑADIR SEÑALES DE VIDEO O DE SEÑAL BIOLÓGICA	38
FIGURA 3-11: FLUJO DE HABILITACIÓN DE LOS BOTONES	38
FIGURA 3-12: RELACIÓN ENTRE NÚMERO DE MUESTRAS DE LA SEÑAL TEMPORAL QUE SE DIBUJAN POR CADA FOTOGRAMA	38
FIGURA 3-13: EJEMPLO DE APLICACIÓN DE LA RELACIÓN ENTRE MUESTRAS DE LA SEÑAL TEMPORAL Y NÚMERO DE FOTOGRAMAS	39

FIGURA 3-14: RELACIÓN ENTRE LA DURACIÓN DEL CONTADOR Y LOS FPS DE LA SEÑAL DE VIDEO	39
FIGURA 3-15: EJEMPLO DE APLICACIÓN DE LA DURACIÓN DEL CONTADOR RESPECTO A LOS FPS DE LA SEÑAL DE VIDEO	39
FIGURA 3-16: EJEMPLO DE APLICACIÓN DE LA DURACIÓN DEL CONTADOR RESPECTO A LOS FPS DE LA SEÑAL DE VIDEO	40
FIGURA 3-17: CONEXIÓN DE LA SEÑAL QUE MARCA QUE EL CONTADOR SE HA AGOTADO CON LAS DOS FUNCIONES DE REPRODUCCIÓN DE VIDEO Y SEÑAL TEMPORAL RESPECTIVAMENTE..	40
FIGURA 3-18: PUESTA EN MARCHA DEL CONTADOR.....	40
FIGURA 3-19: CONEXIÓN DE LA SEÑAL QUE MARCA QUE EL CONTADOR SE HA AGOTADO CON LAS DOS FUNCIONES DE REPRODUCCIÓN DE VIDEO Y SEÑAL TEMPORAL RESPECTIVAMENTE..	41
FIGURA 3-20: ETIQUETAS PARA LA VISUALIZACIÓN DE AVANCE DEL VIDEO EN SEGUNDOS Y FOTOGRAMAS RESPECTIVAMENTE	41
FIGURA 3-21: CUADRO DE DIÁLOGO PARA LA SINCRONIZACIÓN DE LA SEÑAL DE VIDEO Y DE SEÑAL TEMPORAL.....	42
FIGURA 4-1: CREACIÓN DE UN OBJETO TIPO FILEDIALOG Y LANZADO DE LA VENTANA	43
FIGURA 4-2: ESQUEMA DE FUNCIONES DE LA APERTURA Y LECTURA DEL ARCHIVO DE VIDEO ..	44
FIGURA 4-3: CÓDIGO DE LA FUNCIÓN LEERFICHERO()	45
FIGURA 4-4: ESTRUCTURA DE LA APERTURA Y LECTURA DE UN FICHERO DE SEÑAL BIOLÓGICA	47
FIGURA 4-5: FORMATO DEL FICHERO DE SEÑAL BIOLÓGICA.....	47
FIGURA 4-6: CÓDIGO DE INITIALIZEBIOGRAPH().....	48
FIGURA 4-7: CÓDIGO DE LA SEÑAL INTIALIZEBIOVECTOR(INT N).....	49
FIGURA 4-8: ESQUEMA DE REPRODUCCIÓN	50
FIGURA 4-9: CÓDIGO DE PLAYVIDEO()	51

FIGURA 4-10: CÓDIGO DE PLAYBIO()	53
FIGURA 4-11: ESTRUCTURA DE NAVEGACIÓN	54
FIGURA 4-12: ESTRUCTURA DE LA SINCRONIZACIÓN	55
FIGURA 4-13: ESTRUCTURA DE LOS CUADROS DE DIÁLOGO DE CARACTERÍSTICAS DE LA SEÑAL Y SOBRE EL PROGRAMA	56
FIGURA 5-1: LAS DOS SEÑALES SON IGUALES, PERO LA SEÑAL DEL VIDEO ESTÁ RETRASADA RESPECTO A LA SEÑAL TEMPORAL.....	58
FIGURA 5-2: SINCRONIZACIÓN DE LA SEÑAL DE VIDEO A 11 FOTOGRAMAS DEL COMIENZO DE LA SEÑAL DE VIDEO.....	58
FIGURA 5-3: SEÑALES SINCRONIZADAS	59
FIGURA 5-4: LAS SEÑALES SIGUEN SINCRONIZADAS DESPUÉS DE UN TIEMPO.....	59
FIGURA 5-5: LA DISTORSIÓN EN LA SEÑAL TEMPORAL SE OBSERVA DESDE EL PRINCIPIO DE SU REPRODUCCIÓN. SIN EMBARGO, EL USUARIO TODAVÍA NO HA PARPADEADO EN EL FOTOGRAMA 57.	60
FIGURA 5-6: PRIMER MOMENTO EN EL QUE SE APRECIA EL PARPADEO EN EL VIDEO. ES EN EL FOTOGRAMA 133 APROXIMADAMENTE.	61
FIGURA 5-7: VALOR DE SINCRONIZACIÓN DE LA SEÑAL DE VIDEO.....	61
FIGURA 5-8: SEÑALES SINCRONIZADAS	62
FIGURA 5-9: EL SUJETO ABRE LOS OJOS DESPUÉS DE TENERLOS CERRADOS.	62
FIGURA 5-10: EL SUJETO PARPADEA Y SE OBSERVA UN PICO EN LA SEÑAL TEMPORAL.....	63
FIGURA 5-11: ARRIBA, FOTOGRAMA DEL VIDEO CON LA LUZ APAGADA. ABAJO, FOTOGRAMA DEL VIDEO CON LA LUZ ENCENDIDA. LAS DIFERENCIAS SON MÁS VISIBLES EN LAS ZONAS MARCADAS.	64
FIGURA 5-12: SEÑAL TEMPORAL REPRESENTADA EN SU TOTALIDAD. AL PRINCIPIO Y AL FINAL DEL VIDEO SON APRECIABLES SENDAS DISTORSIONES POR EL ENCENDIDO DE LA LUZ.....	65

FIGURA 5-13: INICIO DE REPRODUCCIÓN DE SEÑAL DE VIDEO Y SEÑAL BIOLÓGICA.....	66
FIGURA 5-14: CAPTURA DE PANTALLA DE LA APLICACIÓN EN EL FOTOGRAMA 115. SE APRECIA UNA DISTORSIÓN EN LA SEÑAL TEMPORAL DESPUÉS DEL MOMENTO 4400 PERO SE APRECIA QUE LA LUZ TODAVÍA NO SE HA ENCENDIDO EN EL VÍDEO.....	67
FIGURA 5-15: CAPTURA DE PANTALLA DE LA APLICACIÓN EN EL FOTOGRAMA 839. LA LUZ SE HA ENCENDIDO Y ES APRECIABLE EN EL VÍDEO, NO SE APRECIA NINGÚN ARTEFACTO EN LA SEÑAL TEMPORAL.	67
FIGURA 5-16: SINCRONIZACIÓN DE LA SEÑAL DE VIDEO A 800 FOTOGRAMAS DEL COMIENZO DE LA SEÑAL DE VIDEO Y A 100 FOTOGRAMAS PARA LA SEÑAL TEMPORAL RESPECTO AL INICIO DEL VÍDEO TAMBIÉN.....	68
FIGURA 5-17: EMPIEZA LA REPRODUCCIÓN DEL VIDEO DESPUÉS DE LA SINCRONIZACIÓN.....	68
FIGURA 5-18: EN LA SEÑAL DE VIDEO LA SEÑAL ESTÁ ENCENDIDA Y SE APRECIA UNA DISTORSIÓN EN LA SEÑAL TEMPORAL. LA SEÑAL ESTÁ SINCRONIZADA.	69
FIGURA 5-19: SEÑALES REPRODUCIÉNDOSE JUSTO ANTES DE QUE SE ENCIENDA LA LUZ POR SEGUNDA VEZ.....	70
FIGURA 5-20: LA LUZ ESTÁ ENCENDIDA COMO SE PUEDE APRECIAR EN LA SEÑAL DE VIDEO Y TAMBIÉN SE APRECIA EL ARTEFACTO EN LA SERIE TEMPORAL POR EL ENCENDIDO DE LA LUZ.	70

1 Introducción

1.1 Motivación

La monitorización simultánea del comportamiento de un sistema biológico y el registro de las señales eléctricas que genera su sistema nervioso brinda nuevas posibilidades en el contexto de la biomedicina y en concreto en neurociencia y neuroetología. El desarrollo y abaratamiento de las tecnologías de vídeo ha permitido que hayan sido aprovechadas e introducidas paulatinamente en los citados ámbitos [1], [2], [3].

Es por ello que haya surgido la necesidad de aplicaciones que faciliten la visualización de ambas señales de manera sincronizada en una interfaz específicamente diseñada para ello. A continuación se detallan algunas líneas de investigación que registran ambas señales y podrían beneficiarse de esta implementación.

En el diagnóstico de pacientes con crisis epilépticas psicógenas, es especialmente crucial un análisis detallado de la sintomatología del sujeto, ya que es frecuente que su patología sea erróneamente diagnosticada como epilepsia o crisis electroencefalográficas. Mediante la monitorización simultánea en vídeo de los movimientos del paciente y de su electroencefalograma (EEG) durante largos periodos de tiempo, se consigue caracterizar su actividad en momentos previos, durante y posteriormente a un ataque. Los resultados de esta prueba, permiten establecer una clara distinción entre crisis psicógenas no epilépticas y epilepsia, lo que evitará que sean tratados por error con fármacos epilépticos, de los que experimentarán todos los efectos secundarios y ninguno de sus beneficios [3].

Actualmente se están llevando a cabo experimentos científicos en el laboratorio del Grupo de Neurocomputación Biológica (de ahora en adelante GNB) de la UAM con objeto de estudiar los mecanismos de comunicación, electro-localización y comportamiento de la especie *Gnathonemus petersii*, también conocido como pez elefante. Este pez utiliza la emisión y recepción de impulsos eléctricos para comunicarse con otros especímenes y situarse en entornos de reducida visibilidad, propiedad conocida como “electro localización”. En este ámbito se han realizado mediciones de su actividad eléctrica del pez y se han empezado a correlacionar con su comportamiento para esclarecer el tipo de señales emitidas en cada contexto. [2], [4]–[6]

En el GNB también se está desarrollando una interfaz cerebro-maquina con monitorización de la actividad ocular, con el objeto de detectar artefactos en el EEG y el filtrado de ruido producido por el movimiento de los ojos, o de caracterizar el foco de observación durante el registro. La monitorización simultánea de las señales adquiridas por el lector de ondas cerebrales y la grabación de los ojos del usuario permite afinar el sistema y mejorar su eficacia, sobre todo si se combina técnicas de estimulación adaptadas al usuario [7], [8], [9].

Asimismo, existen otras aplicaciones en las que se registra el entorno y el comportamiento, así como la señalización biológica del sistema, con el objeto de un estudio de varios factores simultáneos que pueden influir en un experimento científico. [10], [11] Particularmente podemos destacar el reconocimiento en tiempo real de emociones aplicadas a las terapias que utilizan música. Este análisis permite ajustar los estímulos al paciente basándose en su estado de ánimo, y se vale de señales EEG [12],

Igualmente esta monitorización simultánea de actividad eléctrica y comportamiento se puede utilizar en el diseño de ciclos cerrados de estimulación respuesta atendiendo a eventos detectados tanto en los registros electrofisiológicos como en el vídeo [2]. Existe un gran número de posibilidades de aplicación de esta tecnología en el ámbito biomédico para integrar la información de comportamiento y la información proveniente de los registros de electrofisiología.

1.2 Objetivos

En este proyecto se propone el diseño de una herramienta para monitorizar, sincronizar y caracterizar de forma simultánea señales biológicas y de comportamiento registradas con electrodos y cámaras.

Se propone la validación de esta herramienta en el estudio de la neuroetología de los peces eléctricos. Los peces de la especie *Gnathonemus Petersii* poseen emisores y receptores tanto para la electrocomunicación como para la electrorecepción [13]. En este trabajo, se utilizará esta especie animal como objeto para la validación de la herramienta de monitorización simultánea en vídeo de las acciones del pez y también las señales eléctricas que emite con el objetivo de estudiar su correlación. Una vez obtenidas las dos señales,

se procederá al desarrollo de una aplicación software para la sincronización y visualización simultánea de ambas debidamente sincronizadas. Hay que tener en cuenta que la frecuencia de adquisición del vídeo será diferente de la de la actividad eléctrica, siendo esta última muy superior.

Para la validación de la herramienta, también se utilizarán registros simultáneos de EEG adquiridos con un casco de electrofisiología y de la actividad ocular registrada con una cámara de vídeo para su sincronización y estudio. Estos dos ejemplos en dos contextos de investigación distintos ilustrarán la utilidad de la herramienta desarrollada y su potencial impacto en diversas disciplinas.

Es importante recalcar la importancia de la sincronización de las señales de vídeo y de los registros electrofisiológicos. Esto requiere que haya un evento común en ambas señales. Por ejemplo, en el caso de los registros del pez se puede utilizar una luz visible en el archivo de video, y que provocará también una alteración reconocible en el campo eléctrico medido. En los registros de video EEG se puede utilizar una señal de audio o directamente utilizar la detección de la amplitud del artefacto provocado por el movimiento ocular.

Resumiendo, se requiere un método de sincronización de ambas señales (video y señal eléctrica) para que las dos empiecen a visualizarse en el mismo punto temporal. Posteriormente, se ha de garantizar que su reproducción sea simultánea aunque las dos señales tengan frecuencias de adquisición diferentes. Para un aprovechamiento mayor de la herramienta, se requiere una funcionalidad de navegación a través de los ficheros, esto es, que se pueda adelantar o retroceder en el tiempo, e incluso pausar en algún momento de la visualización. Se hará un especial hincapie en generar una interfaz sencilla y de fácil manejo para el usuario, enfatizando la usabilidad, una calidad de visualización buena y la optimización de los recursos de memoria y procesamiento. El código de este programa se distribuirá bajo licencia open source.

1.3 Organización de la memoria

La memoria se estructura en torno a los siguientes capítulos:

Introducción: En este apartado se dará una visión global de las motivaciones de la realización de este proyecto, posibles campos de aplicación y una explicación detallada de los objetivos establecidos para el prototipo.

Estado del arte: Hay diferentes herramientas que desempeñan un papel similar aunque no tan general como el esperado para este proyecto, se hará un repaso de los distintos sistemas operativos en los que se pueden usar, las licencias de utilización y sus capacidades.

Diseño: Se abordan distintas posibilidades de implementación para la herramienta y se argumentarán las decisiones de diseño que se han tomado, siempre poniendo como prioridad la sencillez y usabilidad de cara al usuario. Se argumenta la decisión final respecto al lenguaje de programación y librerías a utilizar, y se expone un diseño de alto nivel de la herramienta a nivel funcional y estético.

Desarrollo: Se explicará la estructura del código, como interactúan las diferentes clases entre sí y también se detallará la separación entre la capa de presentación y la capa de acceso a datos. También se explicarán los diferentes hitos de implementación por los que se ha ido pasando y las dificultades encontradas.

Validación de los requisitos de la aplicación: Se detallarán las pruebas llevadas a cabo para probar la funcionalidad de la aplicación. En este caso se utilizarán tanto los registros realizados con el *Gnathonemus Petersii* como los obtenidos con en los registros simultáneos de EEG y de vídeo de la actividad ocular.

Conclusiones: En esta sección se hará una conclusión del trabajo realizado y una revisión de los objetivos cumplidos.

Trabajo futuro: Por último se abordarán los posibles caminos de mejora de la herramienta en futuros trabajos de investigación.

2 Estado del arte

Existen distintos programas con funcionalidades similares a las definidas para este proyecto. A continuación, se expondrá un resumen conciso de sus capacidades, sistemas operativos compatibles, licencias de uso y otros factores. Se hará también un análisis de cumplimiento de los requerimientos definidos en el capítulo anterior.

2.1 Cowlog

2.1.1 Descripción de objetivos y funcionalidad

El objetivo principal del programa Cowlog (<http://cowlog.org/>), es la monitorización por vídeo de animales con la posibilidad de generar marcadores manualmente cuando ocurre un tipo de evento de interés en el experimento científico. Para ello, el investigador anota un número con cada comportamiento que se quiera estudiar, disponiendo también de accesos directos en teclado para un rápido manejo. Posteriormente, debido a que los marcadores están numerados, se puede visionar los ficheros de video y revisar específicamente los momentos relevantes en la grabación. Dado que permite que se visualicen varias pantallas a la vez, se maximiza la productividad del investigador al distinguir sucesos de interés en la monitorización [10].

La capacidad de navegación a través de los ficheros de video es muy alta, pudiendo pausar, retroceder y avanzar en el tiempo, cambiar la velocidad de reproducción, saltar a determinados momentos en el video y deshacer la última acción. La última versión de este software se puede utilizar desde Windows, Linux, MAC e incluso está disponible a través de una interfaz gráfica web. Ver figura 2-1.

Uno de las motivaciones más importantes de los creadores a la hora de diseñar este software era ofrecer una herramienta de uso sencillo y gratuita, por lo que se puede descargar de su página web de forma gratuita bajo licencia open source [14].

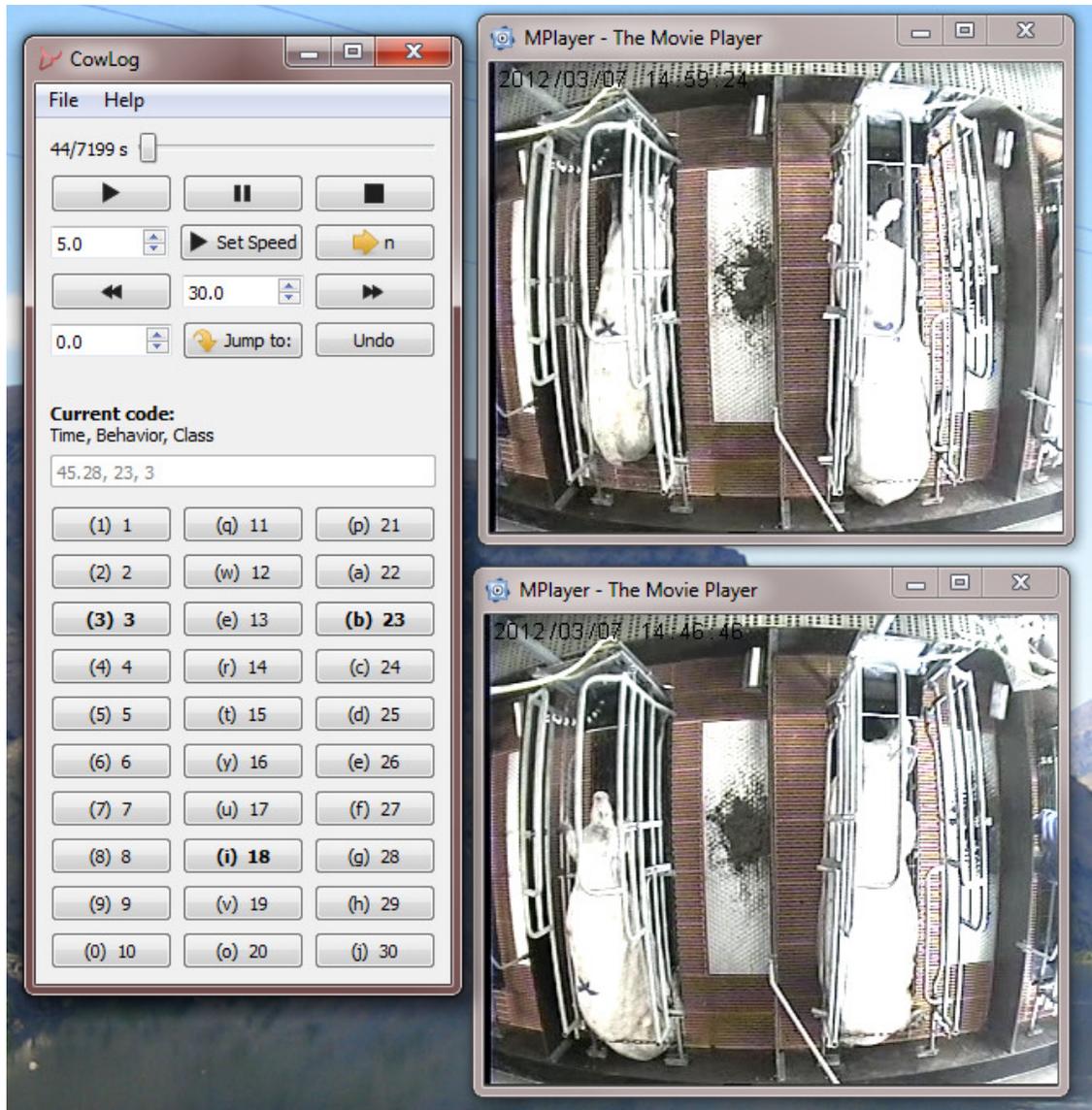


Figura 2-1: Ejemplo de uso de la monitorización en vídeo de Cowlog

2.1.1 Análisis de cumplimiento de requerimientos

Este software ofrece la posibilidad de visualizar varios ficheros de video a la vez pero no series temporales relacionadas como el experimento, así como tampoco ofrece ningún mecanismo de sincronización temporal de ambas señales, ni cuenta con la posibilidad de mostrar señales con distintas frecuencias de adquisición.

2.2 AcqKnowledge

2.2.1 Descripción de objetivos y funcionalidad

Este programa (<http://www.biopac.com/product/acqknowledge-software/>) está enfocado a la visualización en tiempo real, medida, transformación y análisis de datos. Está integrado con varias plataformas hardware de adquisición de datos (DAQ) y también se puede usar de manera independiente. Puede ejecutar adquisiciones de datos complejas, lanzar estímulos y analizar datos con una sencilla interfaz gráfica que no requiere que el usuario sepa programar.

Se pueden aplicar de manera online algunas opciones de análisis, filtrado, transformaciones, escalado de las señales con adquisición desde varios elementos de hardware al mismo tiempo. Tiene múltiples opciones de visualización disponibles durante y después de la adquisición como el pintado de las series temporales, la construcción de histogramas, gráficas de FFT, entre otras. Ofrece una gran capacidad de personalización de la interfaz gráfica, pudiendo ordenar los botones y mostrar barras de herramientas según las preferencias del usuario (ver figura 2-2).

Este software está enfocado a las señales de fisiología por lo que ofrece gran cantidad de opciones específicas para visualización de señales de electrocardiografía, EDA, EEG, EMG y señales respiratorias, así como características mejoradas para la detección de convulsiones en EEG. Se puede ejecutar en sistemas operativos Windows y MAC con licencia de software privada [15].

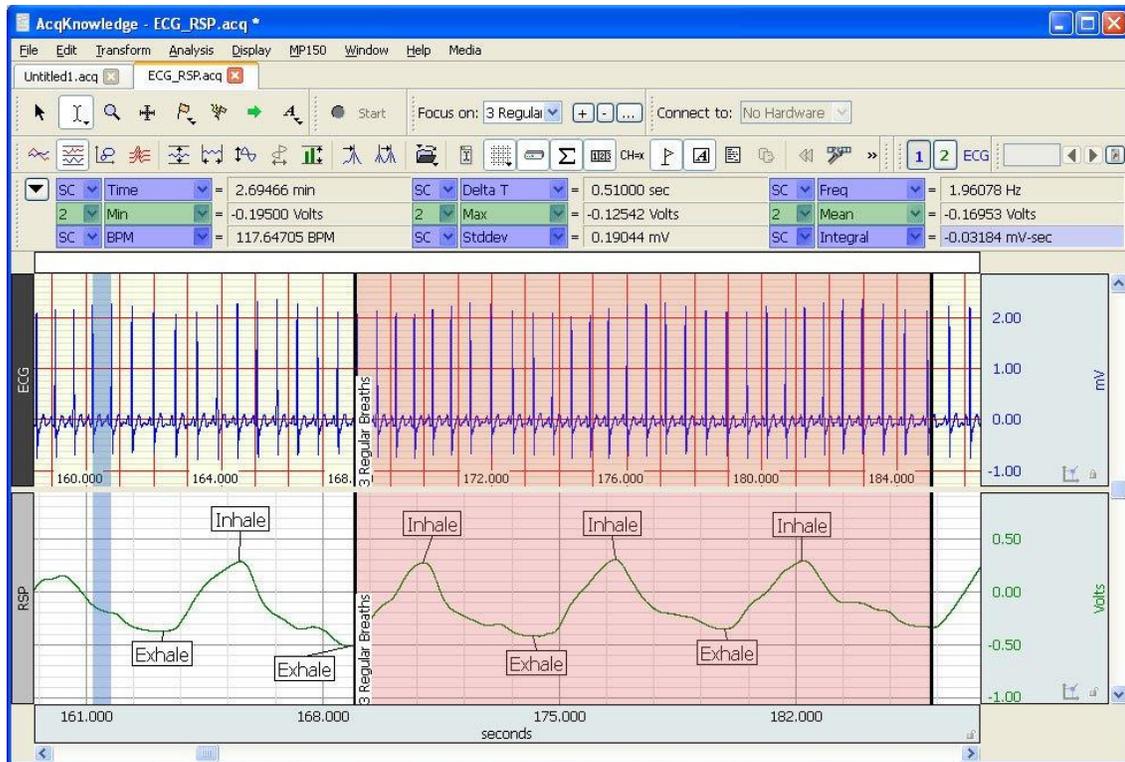


Figura 2-2: Ejemplo de uso de AcqKnowledge

2.2.2 Análisis de cumplimiento de requerimientos

En primer lugar se toma en consideración la versatilidad del software en términos de manejo de señales fisiológicas. Sin embargo, dado que no es posible la visualización simultánea de señales de vídeo, este programa no reúne los requerimientos necesarios para la aplicación buscada. Además, su licencia es de pago por lo que esta será una desventaja añadida al programa.

2.3 Datagraph

2.3.1 Descripción de objetivos y funcionalidad

Esta aplicación está enfocada a la representación en dos dimensiones de gráficos, con una interfaz que permite de manera sencilla exportar e importar datos, cambiar sus formatos y tipos de representación [16].

Las características que facilitan más el manejo de gráficos son la capacidad de combinar gráficos, aplicar métodos matemáticos como regresiones lineales, polinómicas y exponenciales, dibujar gráficos de dispersión, cálculo y representación de la media, desviación estándar y otras. Los gráficos modificados se pueden exportar en formato pdf, eps, sag, tiff, jpg y png.

La aplicación sólo está disponible para el sistema operativo MAC y su licencia de uso es privativa.

2.3.2 Análisis de cumplimiento de requerimientos

Las opciones de representación de señales temporales y las distintas capacidades de edición, formateado de la señal y aplicación de fórmulas pueden resultar útiles, pero son muy básicas y la herramienta no permite representar video, y mucho menos sincronizarlo con la serie temporal, por lo que sus características son insuficientes. Además, sólo está disponible para sistemas operativos MAC y su licencia de uso es privativa.

2.3.3 WinEEG

2.3.4 Descripción de objetivos y funcionalidad

WinEEG (<http://www.mitsar-medical.com/eeg-software/qeeg-software/>) es un programa que permite la grabación, edición y análisis de EEG grabado de forma continua, usando para ello un hardware específico, un amplificador Mitsar (<http://www.mitsar-medical.com/eeg-system/video-eeg/>).

La versión avanzada permite al usuario la posibilidad de importar ficheros no grabados en los formatos creados por los amplificadores Mitsar, ficheros de potencial relacionado con evento (ERP), y realizar comparaciones de bases de datos.

La interfaz gráfica permite mostrar hasta 32 canales de EEG de manera simultánea y la señal de video en la que se graba el comportamiento del sujeto. Entre otras funcionalidades, se puede visualizar la señal en tiempo real o reproducir las señales grabadas para un análisis de las mismas. También es capaz de calcular la media y de calcular tendencias de densidad espectral para hacer el análisis de la grabación a largo plazo más rápidamente.

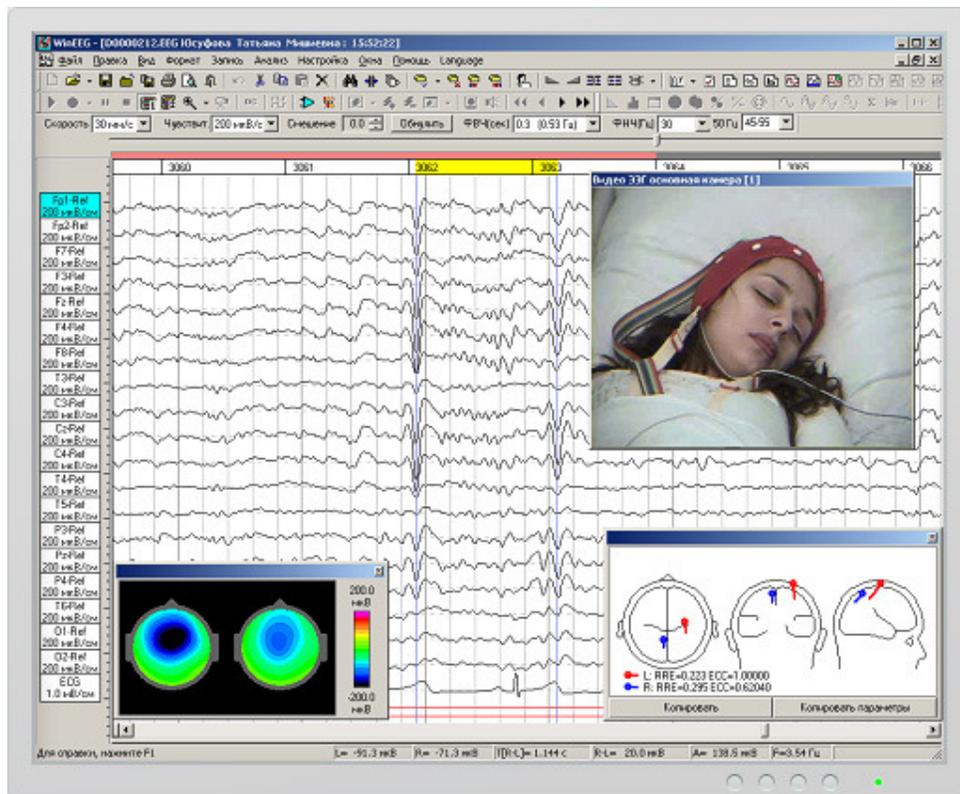


Figura 2-3: Ejemplo de monitorización de señales de EEG (32 canales) y video con WinEEG

2.3.5 Análisis de cumplimiento de requerimientos

En la GUI se puede visualizar a la vez video y señales fisiológicas, lo que se ajusta a los objetivos buscados. Sin embargo, el programa está diseñado para ser integrado con un

hardware que graba de manera síncrona todas las señales, por lo que no es capaz de sincronizar el mismo las señales se han empezado a grabar en distintos momentos del tiempo. Es compatible con el sistema operativo Windows y su licencia de uso es privada.

2.4 Otras posibilidades

Cabe reseñar que hay otras posibilidades utilizadas en investigación para monitorizar señales de video y encefalografía (VEEG). Hay varias líneas de investigación para correlar estas dos señales, y por tanto existen otros programas para la monitorización de ambas señales que han sido desarrollados ad-hoc para determinados experimentos [17], [18].

3 Diseño

En este apartado se detallan las decisiones de diseño que se han considerado y tomado, desde el lenguaje de programación, los requisitos funcionales y la estructura principal del programa, los retos que se han encontrado a la hora de implementar las funcionalidades definidas y un modelo de alto nivel de funcionamiento del programa.

3.1 Lenguajes de programación y librerías consideradas

A continuación, se presentan los lenguajes de programación y librerías que se han considerado para la implementación del programa. El criterio de selección inicial ha sido la existencia de librerías de gestión de video, la facilidad para la representación de señales y su sincronización con el vídeo, y la velocidad de la interfaz gráfica, los tres principales componentes del programa. Se ha tenido en cuenta que los vídeos y las correspondientes señales electrofisiológicas pueden ser muy largas, y por tanto su gestión tiene que estar optimizada tanto en el uso de memoria como en carga computacional.

3.1.1 Matlab

Matlab es un lenguaje de programación de cálculo técnico de alto nivel y también un entorno de desarrollo. Está especializado en la visualización y análisis de datos y pertenece a la categoría de lenguajes de scripting [19]. Es una muy buena opción para la representación de series temporales, pudiendo personalizar el trazo, los ejes, y con una alta eficiencia y control sobre los vectores que se usarían para representar la señal biológica. Sin embargo, Matlab no tiene un gran abanico de funciones dedicadas a la gestión de video adecuadas para este proyecto y no está demasiado optimizado para la una sencilla creación de interfaces gráficas, pues su potencia principal reside en su capacidad de cálculo matricial.

3.1.2 OpenCV

OpenCV es un conjunto de librerías en C++ especializadas en la gestión, análisis y procesado de imágenes en tiempo real. Debido a que OpenCV tiene unas librerías muy optimizadas para la reproducción de video, con el tipo de datos VideoCapture y la función imshow, en un principio se desarrolló un piloto que gestionaba toda la reproducción de video con librerías OpenCV. [20]

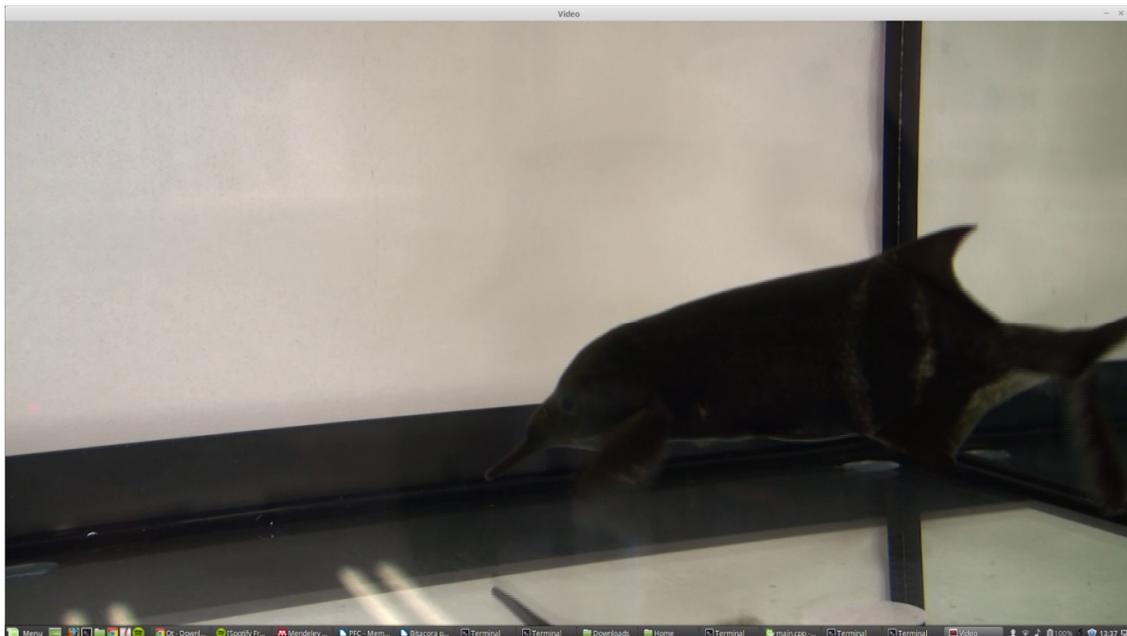


Figura 3-1: Prototipo de reproducción de vídeo a pantalla completa programado con funciones de OpenCV

Este primer acercamiento fue exitoso y se reproducía la señal de vídeo a pantalla completa, pudiendo pausar y navegar de manera fluida. Con el código detallado a continuación se hicieron pruebas con ficheros en formato MTS y MOV de alta resolución y de larga duración con resultados positivos. Ver figura 3-1.

En primer lugar se guardaba el fichero de video en formato VideoCapture y se procedía a la lectura. OpenCV tiene un sistema de gestión de ventanas y este se usaba para mostrar fotograma a fotograma el video mediante la función imshow(). La reproducción continuaba a no ser que el usuario pulsara alguna tecla (ver figura 3-2).

```
using namespace cv;

int main(int argc, char *argv[])
{
    String filename = "/home/asastre/Dropbox/proyecto/video tests/00023.MTS";

    VideoCapture cap(filename);
    if(!cap.isOpened()){ // check if we succeeded
        std::cout << "Error reading video" << std::endl;
        return -1;
    }
    namedWindow("Video",1);
    for(;;)
    {
        Mat frame;
        cap >> frame; // get a new frame
        imshow("Video", frame);
        if(waitKey(30) >= 0) break;
    }
    // the VideoCapture will be deinitialized automatically in VideoCapture destructor

    waitKey(0);
    return 0;
}
```

Figura 3-2: Código del prototipo de reproducción de vídeo con librerías de openCV

Más adelante, al integrar el widget de video con la señal eléctrica para mostrar los dos en la misma ventana, se constató que la interfaz gráfica de OpenCV no es totalmente compatible con Qt en una configuración estándar, es decir, que la gestión de una ventana no puede ser compartida por OpenCV y Qt. Por tanto, todos los elementos gráficos que están preprogramados en Qt, como barras de navegación, barras de herramientas, botones, y otros elementos gráficos no podían ser usados fácilmente en la misma ventana que el video. Asimismo, esto impedía que se representara con funciones de Qt la señal eléctrica en la misma ventana.

3.1.3 Qt

Qt es una biblioteca multiplataforma disponible para C++ y Python. Se utiliza principalmente en la creación de interfaces de usuario, dado su amplio catálogo de elementos gráficos listos para usar como botones, barras de navegación, diálogos... También cabe destacar su gestión de ventanas y su mecanismo de comunicación entre objetos. Cuenta con soporte para distintas plataformas, incluidos dispositivos móviles con sistema operativo Android [21].

3.1.4 Diseño elegido

Con la información recopilada, se tomó la decisión de usar una solución híbrida para la reproducción de video, basada solo parcialmente en OpenCV, quien gestiona la captura de fotogramas y los parámetros del video (se utilizan funciones de OpenCV para obtener los fotogramas por segundo del video) y el resto de la gestión del video se hace con librerías de Qt. La señal biológica eléctrica y la interfaz del usuario se han programado íntegramente en C++ con librerías de Qt.

Matlab resulta útil para la representación de la señal temporal, pero se desechó la idea de usarlo debido a que es en sí mismo un lenguaje de programación distinto y su integración con el resto de la implementación sería más complejo y quizás no muy eficiente.

3.2 Estructura principal

Para facilitar su visualización sincronizada, se decidió diseñar una interfaz gráfica que mostrara el vídeo y de las señales biológicas con los siguientes componentes, ver figura 3-3.

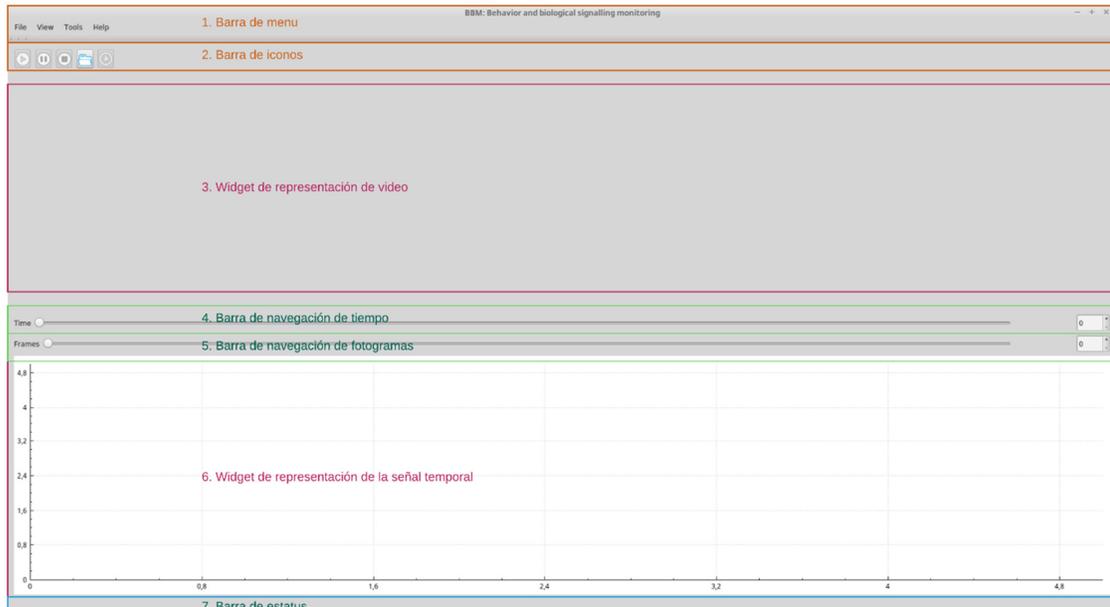


Figura 3-3: Esquema de diseño de la ventana principal del programa

1. Barra de menú

El menú está formado por varios submenús, ver figura 3-4:

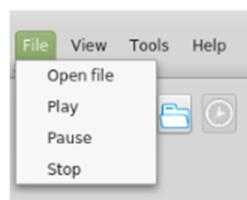


Figura 3-4: Bara de menú

A) Submenú "file". Opciones disponibles: Abrir fichero, reproducir, pausar, parar.

En este menú se pueden cargar los ficheros, tanto de video como el fichero de señal biológica. El resto de opciones de este submenú conciernen a la reproducción del video.

B) Submenú “view menú”. Opciones disponibles: Ver parámetros del fichero y ver parámetros de la señal biológica.

B.1) Características del video: Aquí se puede ver el nombre del fichero elegido y su árbol de directorios, el número de fotogramas por segundo, la duración de segundos y el número de fotogramas total. Este último parámetro resultará de especial interés cuando se vaya a realizar la sincronización entre los ficheros de video y señal biológica. Ver figura 3-5.

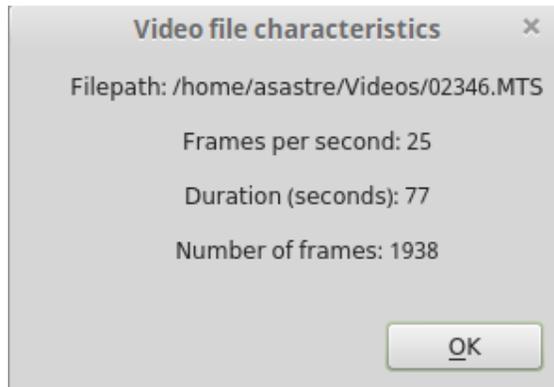


Figura 3-5: Cuadro de diálogo de características del video

B.2) Características de la señal biológica: Esta ventana muestra la frecuencia de muestreo y el número de muestras total del fichero leído. Ver figura 3-6.

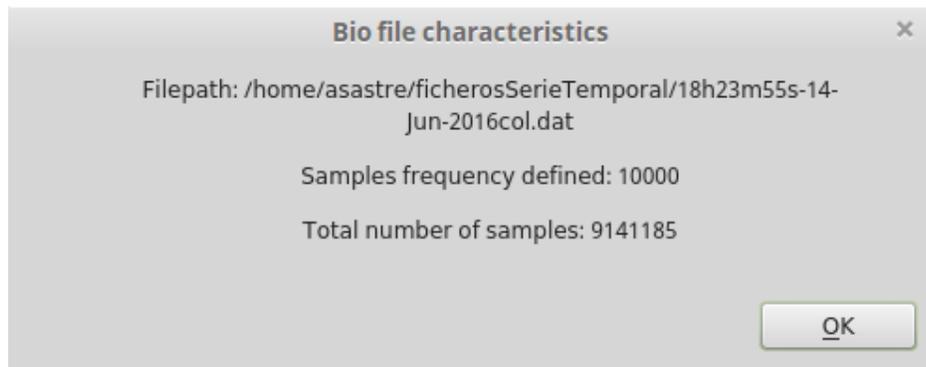


Figura 3-6: Cuadro de diálogo de características de la señal biológica

C) Submenú Tools. Opciones disponibles: Sincronización

El comando sincronización lanzará una nueva ventana que pide al usuario introducir el fotograma al que se quiere saltar en el fichero de video y de forma separada el instante del fichero de serie temporal. El mecanismo de sincronización se explicará con más detenimiento en el apartado 3.2.6.

D) Submenú “Help”. Opciones disponibles: Información sobre el programa

Una nueva ventana se abre al pulsar esta opción, donde se muestra un resumen de la finalidad del programa, la versión y la autora. Ver imagen 3-7.



Figura 3-7: Sobre el programa

2. Barra de iconos

Esta barra es la versión rápida y resumida de la barra de menús, con los comandos más usados. En ella se pueden encontrar los botones para abrir fichero, reproducción, pausa, parada y sincronización.

3. Widget de reproducción de video

En este espacio se reproducirá la señal de video una vez que se haya seleccionado un fichero.

4. Barra de navegación de tiempo

Moviendo el cursor de esta barra el usuario se puede desplazar al segundo de tiempo deseado del video y/o señal eléctrica, mostrándose a la derecha de la barra el segundo actual de reproducción versus el total de tiempo en segundos.

5. Barra de navegación de fotogramas

Moviendo el cursor de esta barra el usuario se puede desplazar al fotograma del video y/o señal eléctrica, mostrándose a la derecha de la barra el fotograma actual de reproducción versus el total de fotogramas.

6. Widget de reproducción de la señal temporal

Aquí se reproducirá la serie temporal una vez que se haya cargado el fichero correspondiente. Se mostrarán 10000 muestras respecto al tiempo, y a medida que se reproduzcan más, las muestras más antiguas desaparecerán de la pantalla por la izquierda. Este parámetro es configurable en la aplicación.

7. Barra de estatus

La barra se utilizará para informar al usuario de eventos que están sucediendo en el programa, como la lectura de un fichero, la reproducción, etc. Ver figura 3-8.



Reading bio file: /home/asastre/ficherosSerieTemporal/12h05m07s-29-Jul-2015.dat

Figura 3-8: Ejemplo de mensaje den la barra de estatus. En esta ocasión se informa al usuario de que se está leyendo un fichero para pintar la señal temporal

3.2.1 Signal y slots

El mecanismo central de funcionamiento de Qt se basa en la gestión de eventos y la comunicación entre los diferentes elementos del programa, lo cual realiza con signals y slots.

En la mayoría de lenguajes de programación, cuando hay un evento en el programa (un valor ha cambiado, se ha pulsado un botón, ha llegado a su fin un temporizador...) se realiza una llamada a una función pasándole un parámetro por referencia. Esta metodología tiene dos problemas principales, puede haber problemas con el tipo del valor pasado por puntero, y en segundo lugar se tiene que saber de antemano a que función se va a querer llamar cuando el evento ocurra.

En Qt, sin embargo, estas notificaciones se realizan con señales. Cuando una determinada acción ocurre, se emite una señal que puede ser predefinida o creada por el programador. Qt de hecho cuenta con un gran abanico de señales por defecto que facilitan en gran medida la creación de las interfaces y evitan realizar trabajo redundante. Cada señal puede estar conectada a uno o varios slots, que son básicamente funciones a las que se las llama cuando este evento ocurre.

Por ejemplo, en la figura 3-9 se aprecia que cuando se emite la señal 1, se llama a dos funciones distintas del objeto 2. En nuestro programa, si el usuario está en la ventana para seleccionar un fichero de video y aprieta el botón de selección, este evento podría emitir una señal que en la ventana principal llamaría a dos funciones, reproducir video y pintar un mensaje en la barra de estatus para comunicar al usuario que la lectura del fichero de video ha sido exitosa y se procede a la reproducción.

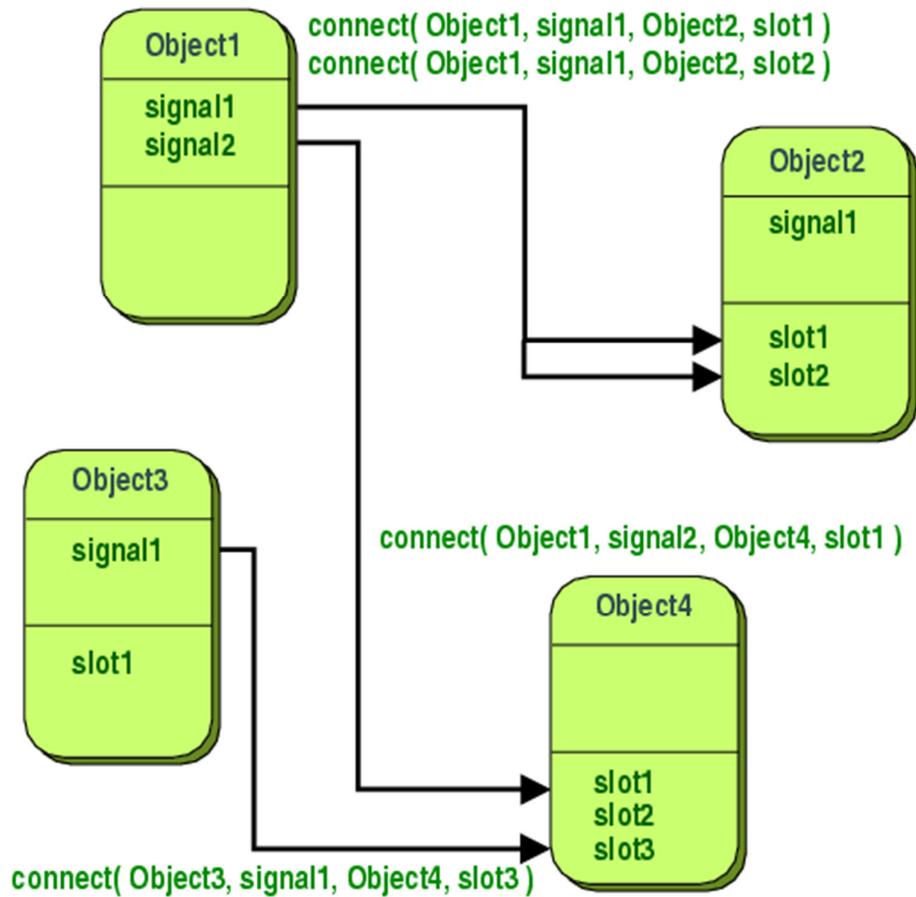


Figura 3-9: Ejemplo de uso de signals y slots

En el programa se utiliza ampliamente las signals y slots para gestionar los eventos, se explicará con más detalle en el capítulo 4.

3.2.2 Carga de ficheros de video y señal biológica y reproducción

El primer paso del flujo del programa es cargar ficheros de video y de señal temporal. Al pulsar el botón de abrir fichero, se abre una ventana en la que se puede navegar por los directorios del ordenador y elegir ambos ficheros. Una vez elegido el fichero biológico, se pregunta al usuario la frecuencia de muestreo de la señal. Respecto a la señal de video no hace falta puesto que se obtiene el número de fotogramas por segundo directamente del fichero usando una función de OpenCV. Ver figura 3-10.

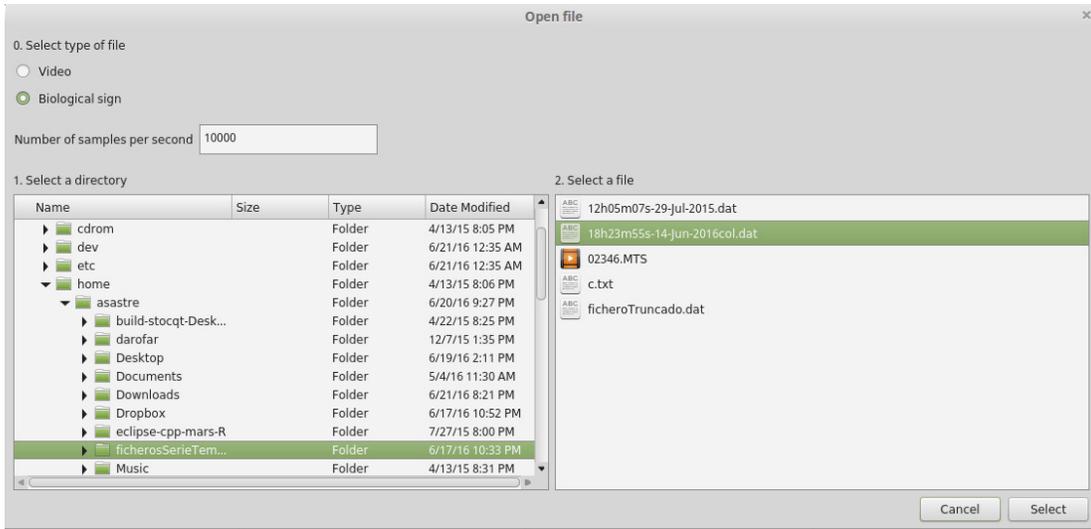


Figura 3-10: Cuadro de diálogo para añadir señales de video o de señal biológica

Se han realizado pruebas con éxito con los siguientes formatos para la reproducción de la señal de vídeo: mp4, avi y mkv, mov y mts.

3.2.3 Reproducción

Una vez los ficheros están en memoria, se desbloquean los botones de reproducción y de sincronización. Ver figura 3-11.

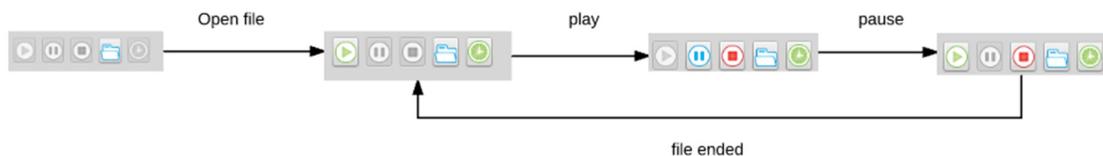


Figura 3-11: Flujo de habilitación de los botones

Debido a que la frecuencia de adquisición del video es distinta que la de la señal biológica, se hace la siguiente conversión:

$$\frac{f_{\text{señal temporal}}}{f_{\text{video}}} \frac{\text{muestras/s}}{\text{fotogramas/s}} = \frac{\text{muestras}}{\text{fotograma}}$$

Figura 3-12: Relación entre número de muestras de la señal temporal que se dibujan por cada fotograma

Es decir, que si por ejemplo tenemos una frecuencia de video de 25 fotogramas/s y una frecuencia de señal eléctrica de 15 kHz:

$$\frac{15000 \text{ muestras/s}}{25 \text{ fotogramas/s}} = 600 \frac{\text{muestras}}{\text{fotograma}}$$

Figura 3-13: Ejemplo de aplicación de la relación entre muestras de la señal temporal y número de fotogramas

Por cada fotograma, se tienen que mostrar 600 muestras de la señal biológica.

Para evitar errores, se restringe la libertad del usuario a la hora de elegir qué acciones quiere tomar en el programa. Por ejemplo, no tiene sentido que se intente parar la reproducción si no se ha empezado a reproducir antes. Se sigue la siguiente secuencia de activación: Si el usuario ha cargado una señal temporal y un fichero de video, se permite reproducir y/o sincronizar. Si los ficheros se están reproduciendo, entonces es posible pausar o parar la reproducción.

3.2.4 QTimer

Toda la reproducción está basada en un contador de tiempo de la librería Qt, del tipo QTimer. En el momento que se carga un fichero de video, el programa extrae el número de fotogramas por segundo (fps) y se establece la siguiente proporción:

$$\text{Duración de QTimer} = \frac{1 \text{ fotograma}}{\text{fps}_{\text{video}}}$$

Figura 3-14: Relación entre la duración del contador y los fps de la señal de video

En el ejemplo que nos ocupa:

$$\text{Duración de QTimer} = \frac{1 \text{ fotograma}}{\text{fps}_{\text{video}}} = \frac{1 \text{ fotograma}}{25 \text{ fps}} = 40 \text{ ms}$$

Figura 3-15: Ejemplo de aplicación de la duración del contador respecto a los fps de la señal de video

Es decir, que cada 40 ms se va a pintar un fotograma, y según nuestros cálculos anteriores, se pintarán 600 muestras de la señal temporal también. A continuación en la figura 3-16 se puede ver la definición del contador y la asignación de la fórmula expresada en la referencia 3-12.

```
QTimer* timer;  
timer = new QTimer(this);  
qTimerDelay=(1000/FPS_open);
```

Figura 3-16: Ejemplo de aplicación de la duración del contador respecto a los fps de la señal de video

Cuando este contador se agote, se emitirá una señal timeout() que está conectada a su vez con dos slots, las dos funciones principales del programa, playVideo() y playBio(), como se muestra en la figura 3-17.

```
connect(timer, SIGNAL(timeout()), this, SLOT(playVideo()));  
connect(timer, SIGNAL(timeout()), this, SLOT(playBio()));
```

Figura 3-17: Conexión de la señal que marca que el contador se ha agotado con las dos funciones de reproducción de video y señal temporal respectivamente

Es decir, que cada vez que pasen 40 ms, se llamará a playVideo(), que mostrará un nuevo fotograma por pantalla, y también se llamará a playBio(), que cargará el número de valores necesarios para ese fotograma desde la memoria y los mostrará.

Una vez definido la duración del contador, cuando el usuario pulse el botón de reproducir, se pondrá en marcha este contador con el comando de la figura 3-18:

```
timer->start(qTimerDelay);
```

Figura 3-18: Puesta en marcha del contador

Este contador es clave para la gestión de la lógica de todo el programa, puesto que es el método que se usa para pausar el video. Cuando el usuario pulsa el botón de pausa, simplemente este contador dejará de ser llamado y por tanto no emitirá más señales de timeout(), dejando todos los demás parámetros del programa intactos y listos para reanudar la reproducción en cuanto el usuario apriete el botón de reproducción de nuevo. Por

supuesto, cuando el usuario para el video completamente, el contador también se parará en esta ocasión. Ver figura 3-19.

```
timer->stop();
```

Figura 3-19: Conexión de la señal que marca que el contador se ha agotado con las dos funciones de reproducción de video y señal temporal respectivamente

3.2.5 Navegación

Una de las utilidades más básicas es la posibilidad de navegación en los ficheros. Esta brinda la posibilidad de avanzar o retroceder a un determinado segundo del tiempo o fotograma.

Debido a la correspondencia que se ha establecido entre los fotogramas del video y el número de muestras, se ha procedido a la simplificación de la solución y no se permite saltar a determinadas muestras de la señal temporal, pudiendo hacerlo solo a un determinado fotograma. Al saltar a un fotograma, se empezará a reproducir el número de muestras asignado a él. Del mismo modo, si por cada segundo mostramos por ejemplo 25 fotogramas, al saltar a un segundo, se empezará a reproducir el primer fotograma asignado a ese segundo del tiempo, y así sucesivamente.

A la derecha de las barras de navegación se han definido unas etiquetas que muestran la posición actual en segundos y fotogramas respecto al total, que pueden apreciarse en la figura 3-20.

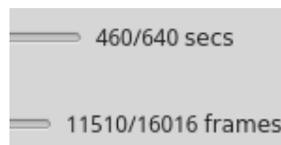


Figura 3-20: Etiquetas para la visualización de avance del video en segundos y fotogramas respectivamente

3.2.6 Sincronización

Sin duda una de las partes más cruciales del programa, la opción de sincronización nos permite elegir en qué fotograma exacto debe empezar la señal de video, y se hará de forma análoga para la señal biológica.

Debido a que para la grabación de ambas señales se utilizan equipos de adquisición diferentes (por ejemplo en el caso del video del pez se utilizarán una cámara y para la señal eléctrica se utiliza el circuito de electrodos de la pecera) [4] es necesario realizar una sincronización de ambas señales.

Para ello, el investigador utilizará un artefacto en visible en ambas señales. Por ejemplo, para el caso del pez, se utilizará una señal lumínica que se encenderá y se apagará, lo que quedará reflejado en el video, y también provocará una distorsión visible en la señal temporal.

Para realizar la sincronización, el usuario procederá al visionado de ambas señales y decidirá en que fotograma de la señal de video es visible por primera vez ese artefacto, y de forma análoga para la señal temporal. A continuación, utilizando el menú de sincronización del programa mostrado en la figura 3-21, se elegirá el fotograma absoluto en el que la señal debería de empezar, haciendo esto de forma separada para las dos señales.

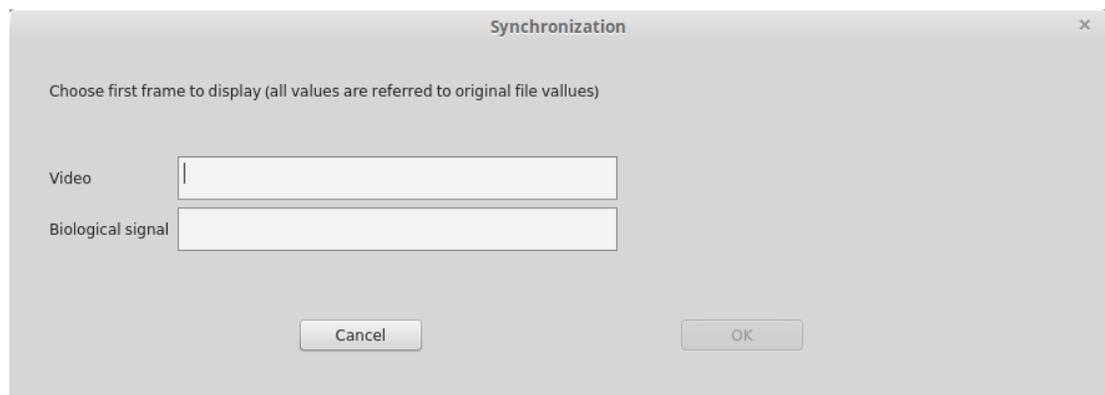


Figura 3-21: Cuadro de diálogo para la sincronización de la señal de video y de señal temporal

Es importante destacar que si se realiza la sincronización varias veces sobre el mismo fichero, cada vez que se abra el cuadro de diálogo para preguntar al usuario a qué fotograma quiere desplazar la señal, estos fotogramas siempre serán definidos respecto a la señal original, desechando los cambios realizados con anterioridad en la sincronización de la señal.

4 Desarrollo

Dado que se ha explicado en el capítulo anterior todo lo pertinente a la GUI, en este apartado se profundizará en la implementación de esas funcionalidades.

4.1 Apertura y lectura de ficheros

Nada más abrir el programa, la única acción disponible para el usuario es cargar los ficheros. Para ello se abre el cuadro de diálogo de la figura 3.9. Al pulsar el botón de abrir fichero, se genera una señal que está conectada con un slot en la función `on_open_fileToolButton_clicked()`. Esta función creará un objeto de `fileDialog` y lanzará la ventana con el comando `exec()`. Este comando hace que se muestre la ventana y no se pueda interactuar con la ventana padre que la ha llamado hasta que se cierre (se puede cerrar de tres formas, dando a cancelar, seleccionando un fichero y pulsando OK o cerrando la ventana pulsando en la X de la esquina superior derecha), ver figura 4-1.

```
QString filePath="";  
FileDialog w;  
  
//exec won't return control to the main window until the open file dialog window is closed  
w.exec();
```

Figura 4-1: Creación de un objeto tipo `fileDialog` y lanzado de la ventana

No se permite al usuario pulsar OK en el menú de este diálogo hasta que haya elegido si quiere abrir un fichero de video o de señal biológica, seleccionado un directorio válido y un fichero. Así pues, al cerrar este diálogo (siempre que no se haya elegido la opción cancelar o se haya cerrado la ventana) se tendrá como parámetro una cadena con el directorio del fichero que se quiere cargar, y, en el caso de la señal biológica, también la frecuencia de adquisición de la señal introducida por el usuario.

4.1.1 Señal de video

Una vez seleccionado un fichero de video, se detallan a continuación las acciones que siguen en el programa,

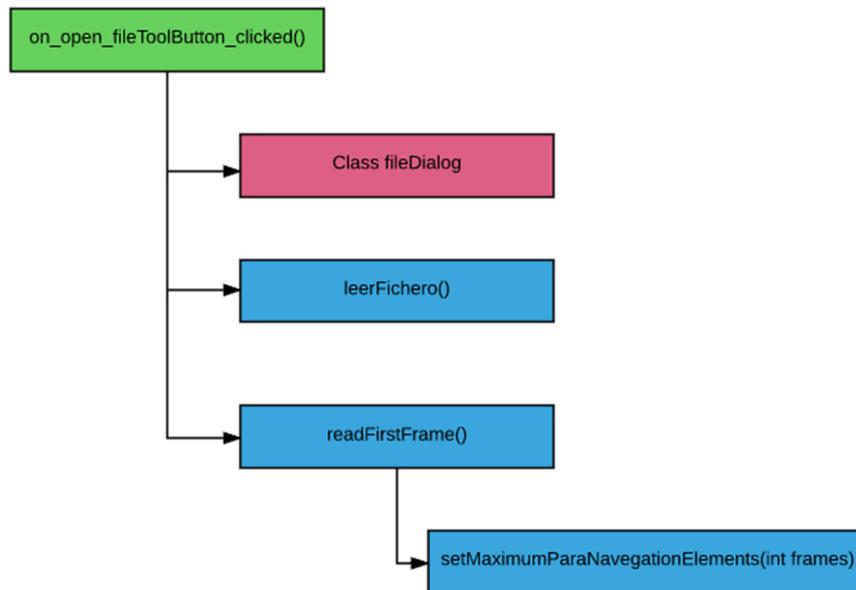


Figura 4-2: Esquema de funciones de la apertura y lectura del archivo de video

- **leerFichero()**

Este método identifica si el fichero es MTS extrayendo esta información de la string que contiene el nombre del fichero a leer y su path. Esta información se usará luego puesto que se ha demostrado empíricamente que con las funciones de openCV utilizadas para obtener los fps, se obtiene el doble de los fps que realmente existen. Es decir, una señal de 25 fps será detectada como si tuviera 50 fps.

Después, simplemente se guarda el primer frame en una variable de tipo VideoCapture. Ver figura 4-3.

```
VideoCapture MainWindow::leerFichero()
{
    if(videoFileWithPath.contains("MTS"))
    {
```

```

    videolsMTS=1;
}
VideoCapture cap(videoFilePath.toString());
if(!cap.isOpened()){ // check if we succeeded
    std::cout << "Error reading video" << std::endl;
}
return cap;
}

```

Figura 4-3: Código de la función leerFichero()

- **readFirstFrame()**

En esta función, se pasa la variable del tipo VideoCapture a una de tipo Mat. Se obtienen los fps del video (con la salvedad de MTS que además este número se divide por dos, como ya se ha explicado en la función leerFichero()) y también el número total de fotogramas. También se hallará el número de ms al que se inicializa el contador QTimer. Desde aquí se llamará a setMaximumParamNavigationElements(int n), ver figura 4-4.

```

void MainWindow::readFirstFrame()
{
    Mat currentFrame;
    cap>>currentFrame;

    if(videolsMTS)
    {
        FPS_open=cap.get(CV_CAP_PROP_FPS)/2;
        totalNumFrames=cap.get(CV_CAP_PROP_FRAME_COUNT)/2;
    }
    else
    {
        FPS_open=cap.get(CV_CAP_PROP_FPS);
        totalNumFrames=cap.get(CV_CAP_PROP_FRAME_COUNT);
    }
}

```

```
totalNumFramesWithSync=totalNumFrames; //Initially all the signal frames will be
displayed, unless the sync makes the video sign to jump ahead
syncDiff = totalNumFrames-totalNumFramesWithSync;
qTimerDelay=(1000/FPS_open);

setMaximumParamNavegationElements(totalNumFrames);

statusBar()->clearMessage();
statusBar()->showMessage("Video loaded: "+ videoFileWithPath, 7000);

videoLoaded=1;
ui->syncToolButton->setEnabled(true);
}
```

Figura 4-4: Código de la función readFirstframe()

- **setMaximumParamNavegationElements(int n)**

Las barras de navegación y las etiquetas que muestran el progreso del video se inicializan.

4.1.2 Señal biológica

La parte de selección del fichero es análoga a la parte de video. Una vez que se tiene el fichero, se llama a readBioFile(), ver figura 4.4.

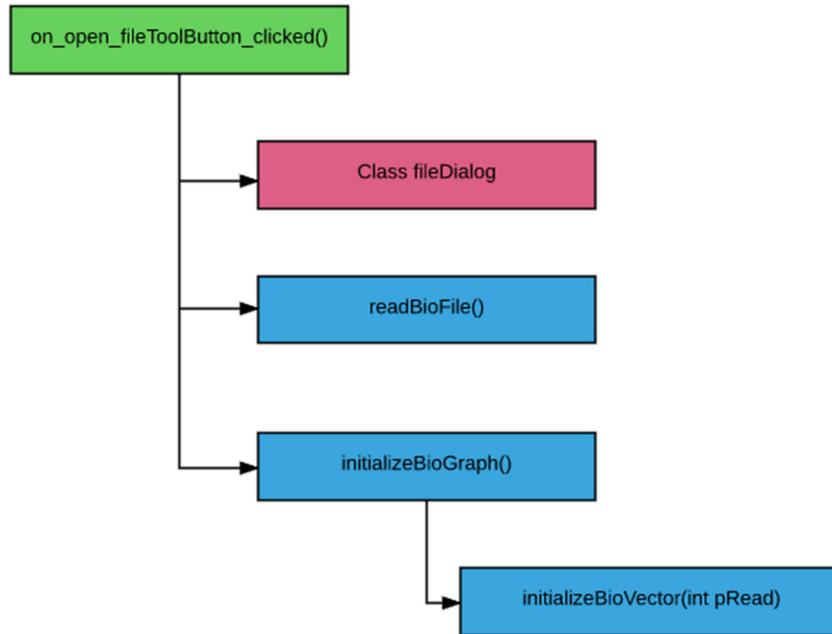


Figura 4-4: Estructura de la apertura y lectura de un fichero de señal biológica

- **readBioFile()**

Aquí se guardan todas las muestras leídas en memoria, en un vector del tipo QVector. Hay que tener en cuenta que en la versión actual el sistema solo admite ficheros de dos columnas, en los que los valores de la primera columna se considerarán como abcisas y la segunda columna serán las ordenadas, ver figura 4-5. También se calcula el número de pares x-y de muestras leídas y los valores mínimos y máximos de y para la posterior determinación del rango del eje y.

```

1 -5.423
2 -5.41
3 -5.40
4 -5.423
5 -5.41
6 -5.40
7 -5.423
8 -5.41
9 -5.40
  
```

Figura 4-5: Formato del fichero de señal biológica

- **initializeBioGraph()**

Se inicializa el gráfico para la serie temporal. Se definen los títulos de los ejes, sus valores máximos y mínimos y se llama a initializeBioVector(0). Ver figura 4-6

```
void MainWindow::initializeBioGraph()
{
    ui->bioWidget->addGraph();

    ui->bioWidget->xAxis->setLabel("Time");
    ui->bioWidget->yAxis->setLabel("Amplitude");
    ui->bioWidget->xAxis->setRange(minX,maxX);
    ui->bioWidget->yAxis->setRange(minY,maxY);

    initializeBioVector(0);
}
```

Figura 4-6: Código de initializeBioGraph()

- **initializeBioVector(int value)**

Esta función puede llamarse en otros momentos de la ejecución del programa para que cargue los vectores de representación de la señal temporal a partir de un determinado valor. Es decir, que “salte” a un valor distinto de 0. En este caso simplemente queremos que este vector empiece desde 0.

Para ello, se vacían los vectores que se usan para la representación, displayX y displayY (hay otros dos vectores para x e y que contienen todas las muestras de la señal pero son diferentes). El tamaño de estos vectores depende del parámetro bioSamplesDisplayed, que es el número de muestras que se representan en el gráfico a la vez. Debido a que en esta ocasión estamos saltando al valor 0, se llenan estos dos vectores de 0 para que siempre se represente el mismo número de muestras en el widget y no se distorsione la señal al representar un número mayor o menor de muestras.

```

void MainWindow::initializeBioVector(int value)
{
    int j;
    pRead=value;
    displayOnTheLeft=0;

    if(totalElectricalSamples>(pRead+bioSamplesDisplayedPerFrame))
    {
        bioFinished=0;
    }

    displayX.clear();
    displayY.clear();
    for(j=pRead; j<displayedSamples+pRead; j++)
    {
        displayX.append(vectorX[j]);
        displayY.append(0);
    }
}

```

Figura 4-7: Código de la señal initializeBioVector(int n)

4.2 Reproducción, pausa y parada

Las funciones `on_playButton_clicked()`, `on_pauseButton_clicked()`, `on_stopButton_clicked()` de reproducción, pausa y parada respectivamente se llaman cuando el usuario pulsa el botón pertinente y se activa una señal que está conectada con un slot en sus respectivas funciones.

Una vez que se ha pulsado el botón play, este reanuda el contador que llamará periódicamente a `playVideo()` y `playBio()`, ver figura 4.8. Cuando se pulsa el stop, se llama a `jumpToVideoBeginning()`, que reseteará a 0 el iterador que va extrayendo un frame cada vez que se llama a `playBio()`. En el caso de la señal biológica, se reseteará a 0 el contador de muestras que se utiliza para marcar el progreso de la señal y leer las

siguientes muestras pertinentes. Aparte, se limpia el widget de la señal temporal para que aparezca vacío.

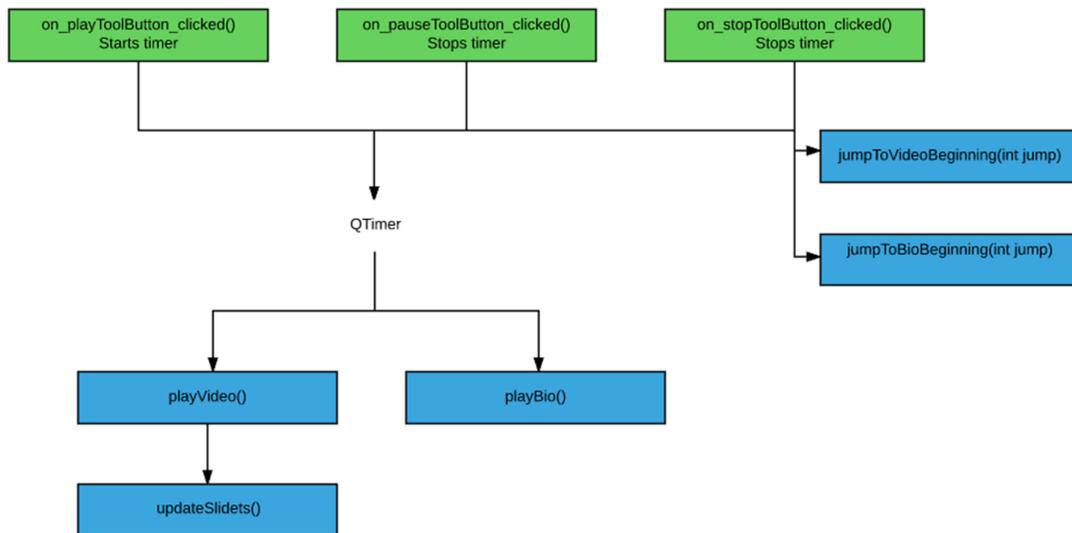


Figura 4-8: Esquema de reproducción

- **playVideo()**

Primero se evalúa si hay un video cargado y no se ha llegado al final de sus fotogramas (QTimer seguirá llamando a esta función aunque no se cumplan las dos condiciones citadas). Después, se carga un nuevo fotograma, se utilizan las funciones cvtColor para definir los colores y otras transformaciones.

A continuación, se realiza una transformación del fotograma QPixmap, y se pasa a la QLabel donde se va a representar, y se llama a updateSliders() que actualizará los valores de las barras de navegación y las etiquetas, ver figura 4-9. Si el video ha llegado a su fin, se para su reproducción.

```

void MainWindow::playVideo()
{
    if((videoLoaded==1)&&(frameReadCounter<=totalNumFrames)){
        if (frameReadCounter!=0)
        {
            cap>>currentFrame;
        }
        if (currentFrame.channels()== 3){
  
```

```

        cv::cvtColor(currentFrame, currentFrame, CV_BGR2RGB);
        qtFrame = QImage( (const unsigned char*)
(currentFrame.data),currentFrame.cols, currentFrame.rows,
currentFrame.step1(),QImage::Format_RGB888 );
    }
    else
    {
        qtFrame = QImage((const unsigned
char*)(currentFrame.data),currentFrame.cols, currentFrame.rows,
QImage::Format_Indexed8);
    }

    QPixmap pix;
    pix= QPixmap::fromImage(qtFrame);

    ui->videoLabel->setPixmap(pix.scaled(ui->videoLabel->size(),
Qt::KeepAspectRatioByExpanding, Qt::SmoothTransformation));
    ui->videoLabel->adjustSize();

    frameReadCounter++;

    updateSliders();
}
else
{
    on_stopToolButton_clicked();
    cout<<"ya no se puede reproducir mas señal de video"<<endl;
    statusBar()->showMessage("Video file ended ", 7000);
}
}
}

```

Figura 4-9: Código de playVideo()

- **playBio()**

En primer lugar, se comprueba si hay un fichero cargado y si lo hay, que no se ha terminado de reproducir, con el objetivo de evitar incurrir en errores en el sistema.

A continuación, si quedan muestras suficientes para mostrar el número establecido para cada fotograma, se sigue adelante. Se comprueba si se han reproducido un número menor de muestras que el total que se muestra en el widget. Si es el caso, habrá ceros rellenando los vectores de representación, por lo que se desplazan para hacer sitio para la nueva señal.

Se calculan los valores mínimos y máximos de x para actualizar los ejes y se vuelve a pintar el gráfico, lo que lo actualizará y mostrará los nuevos valores.

```
void MainWindow::playBio()
{
    int i;
    if(bioLoaded==1 && bioFinished!=1){
        if(totalElectricalSamples>(pRead+bioSamplesDisplayedPerFrame))
        {
            for(i=0; i<bioSamplesDisplayedPerFrame; i++)
            {
                if(displayOnTheLeft<displayedSamples)
                {
                    displayY[displayOnTheLeft]=vectorY.value(displayOnTheLeft);
                    displayOnTheLeft++;
                }
                else
                {
                    displayX.append(vectorX[pRead]);
                    displayY.append(vectorY[pRead]);
                }
            }
        }
    }
}
```

```

        displayX.removeFirst();
        displayY.removeFirst();
    }

    minX=displayX.first();
    maxX=displayX.last();

    pRead++;
}
}
else
{
    cout<<"ya no se puede reproducir mas bio"<<endl;
    bioFinished=1;
    statusBar()->showMessage("Bio file ended ", 2000);
}
ui->bioWidget->graph(0)->setData(displayX,displayY);

ui->bioWidget->xAxis->setRange(minX,maxX);
ui->bioWidget->yAxis->setRange(minY,maxY);
ui->bioWidget->replot();
cout<<"Reproduzcome. QTimerdelay "<<qTimerDelay<<endl;
}
}

```

Figura 4-10: Código de playBio()

4.3 Navegación

Como ya se ha visto, las barras de navegación se actualizan cada vez que la señal de video pinta un nuevo fotograma. A continuación se describe qué ocurre cuando se mueve el cursor de una de las barras para desplazarse por la señal.

Hay dos funciones que se están conectadas por un slot y detectan cuando la señal de que el usuario ha movido las barras es emitida, son `timeChanged()` y `frameChanged()`, para sus respectivas barras de navegación.

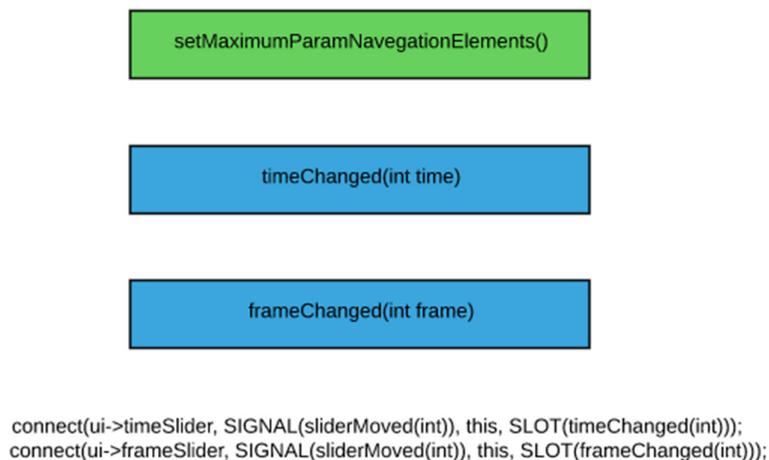


Figura 4-11: Estructura de navegación

4.4 Sincronización

Cuando se pulsa el botón de sincronización, se crea una clase del tipo `syncDialog` que mostrará el cuadro de diálogo para que el usuario introduzca a que fotograma quiere saltar de la señal de video y/o fotograma para la señal biológica. No se da la opción al usuario de saltar a una determinada muestra de la señal temporal. Esta decisión se ha tomado por razones de implementación y porque normalmente las señales de sincronización aparecen con más naturalidad en el vídeo. Se ha creado una correlación directa entre un fotograma y el número de muestras eléctricas que se representarán para ese solo fotograma, número que variará según las frecuencias de adquisición de ambas señales. Así pues, se saltará a un determinado fotograma y a partir de ahí se empezará a reproducir la señal eléctrica como de costumbre para ese determinado fotograma.

Como se ha comentado ya anteriormente, los valores introducidos por el usuario son siempre respecto a la señal original, si se sincroniza varias veces sucesivamente, el número de fotograma al que se salta se considerará respecto a la señal que se cargó inicialmente.

Después de obtener los datos de sincronización, se pausará la señal y según si los valores de video o de señal eléctrica se han cambiado, o los dos, se llamará a las funciones `jumpToVideoBeginning(int jump)` y `jumpToBioBeginning(int jump)` que ya se explicaron anteriormente, ver figura 4-12.

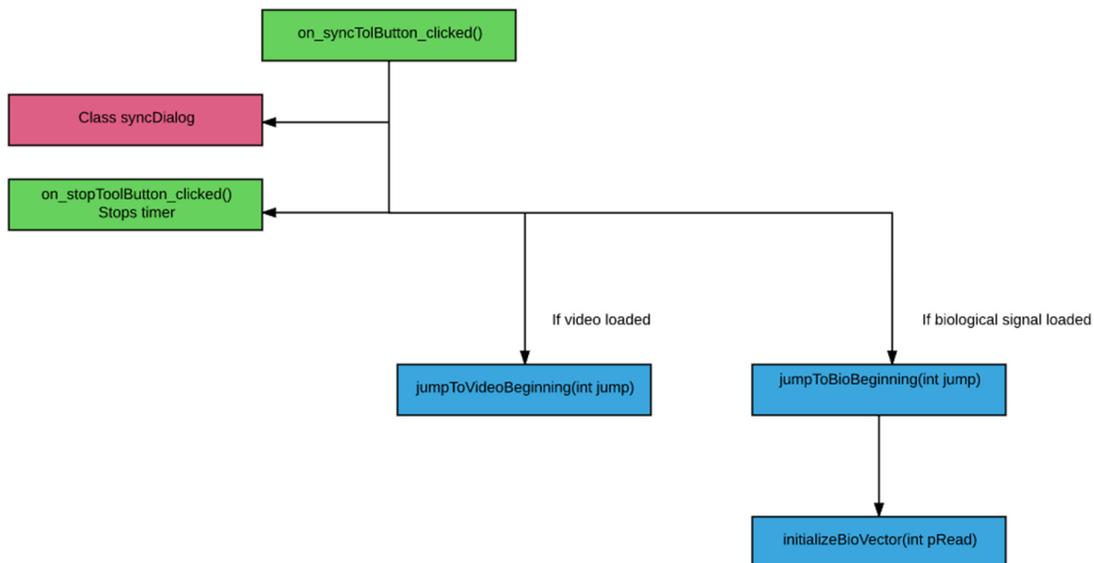


Figura 4-12: Estructura de la sincronización

4.5 Información del programa y características de los ficheros cargados

Al seleccionar las opciones “video characteristics”, “bio characteristics” y “about” se mostrarán las ventanas pertinentes, descritas en el capítulo 3.2 Ver figura 4-13.



```
connect(ui->actionVideo_characteristics, SIGNAL(triggered()), this, SLOT(videoSignalCharacteristics()));  
connect(ui->actionBio_signal_characteristics, SIGNAL(triggered()), this, SLOT(bioSignalCharacteristics()));  
connect(ui->actionAbout, SIGNAL(triggered()), this, SLOT(aboutBBM()));
```

Figura 4-13: Estructura de los cuadros de diálogo de características de la señal y sobre el programa

4.6 Hardware utilizado en las pruebas

Para adquirir los vídeos para las pruebas de la aplicación se han utilizado dos tipos de cámaras: una cámara Nikon SLR 5100 y una cámara de vídeo SONY HDR-XR520VE. Los registros de actividad del pez se han obtenido con una tarjeta de adquisición de datos NI PCI-6521 y los de EEG con un amplificador g.tec g.USBAMP.

5 Integración, pruebas y resultados

Para hacer las pruebas de funcionamiento del programa, se utilizan dos sets distintos de datos. En concreto los dos casos de uso que se detallaron anteriormente. La monitorización simultánea del pez *Gnathonemus Petersii* en video y su correspondiente señal eléctrica, y también la señal de EEG de una persona conectada con un casco de EEG a un amplificador g.USBAMP monitorizando la actividad ocular en vídeo.

5.1 Sincronización del video de una neurona eléctrica y su serie temporal

Como primera demostración de la sincronización del programa, se utiliza una neurona electrónica que genera pulsos periódicos similares a la actividad en ráfagas de los circuitos generadores centrales de patrones [2]. La actividad de la neurona ha sido grabada en video directamente de la pantalla desde el programa de adquisición de datos del GNB con una frecuencia de 25 fotogramas por segundo. Al mismo tiempo, se ha guardado en un fichero los valores de la señal temporal representada, esta señal se ha adquirido con una frecuencia de 10 KHz.

Para proceder a la sincronización, se generan distorsiones mediante la inyección de corriente en la neurona que serán visibles en ambas señales. Aunque las dos señales se han empezado a grabar casi al mismo tiempo hay una ligera variación entre ellas. En concreto, la señal del vídeo está retrasada como se aprecia en la imagen 5-1.

Es importante destacar que aunque las dos señales son idénticas, los valores de tiempo representados en el eje x de las señales no coinciden, puesto que los valores que se aprecian en el vídeo son respecto a un momento mucho anterior en el tiempo, cuando la neurona empezó a generar pulsos, mucho antes de que se adquirieran las dos señales.

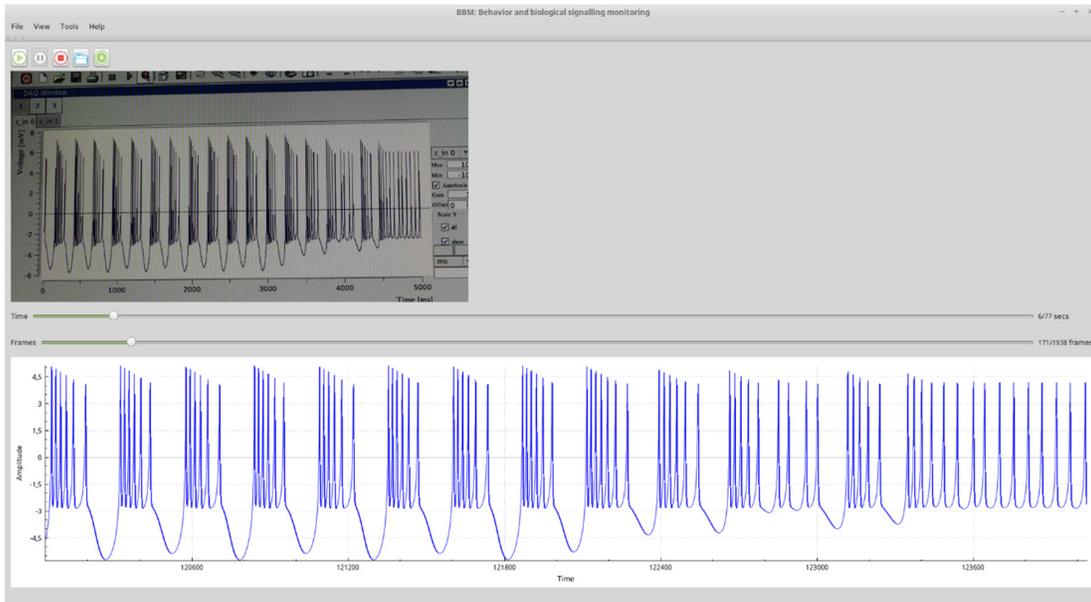


Figura 5-1: Las dos señales son iguales, pero la señal del video está retrasada respecto a la señal temporal

Ahora vamos a proceder a la sincronización. Debido a que la señal de video está atrasada, hay que moverse a un momento posterior en el video para que las dos señales empiecen a la misma vez. En concreto, se ha elegido el valor de 11 fotogramas para la señal de video, como se aprecia en la figura 5-2.

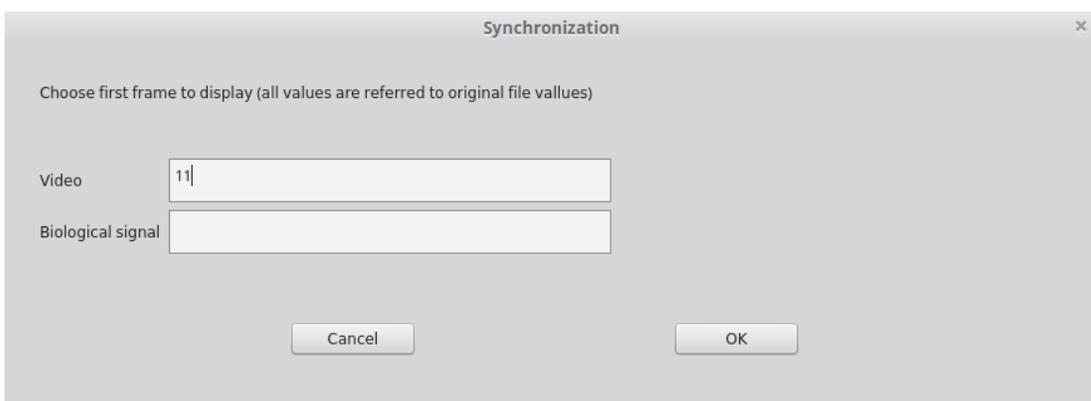


Figura 5-2: Sincronización de la señal de video a 11 fotogramas del comienzo de la señal de video

Al reproducir las dos señales de nuevo, se aprecia que están sincronizadas (Ver figura 5-3). En concreto, centrándose en los pulsos de amplitud constante que se observa a la derecha del video y de la señal temporal se ve que hay el mismo número de estos pulsos para la señal de video y la señal temporal, 10 pulsos en total.

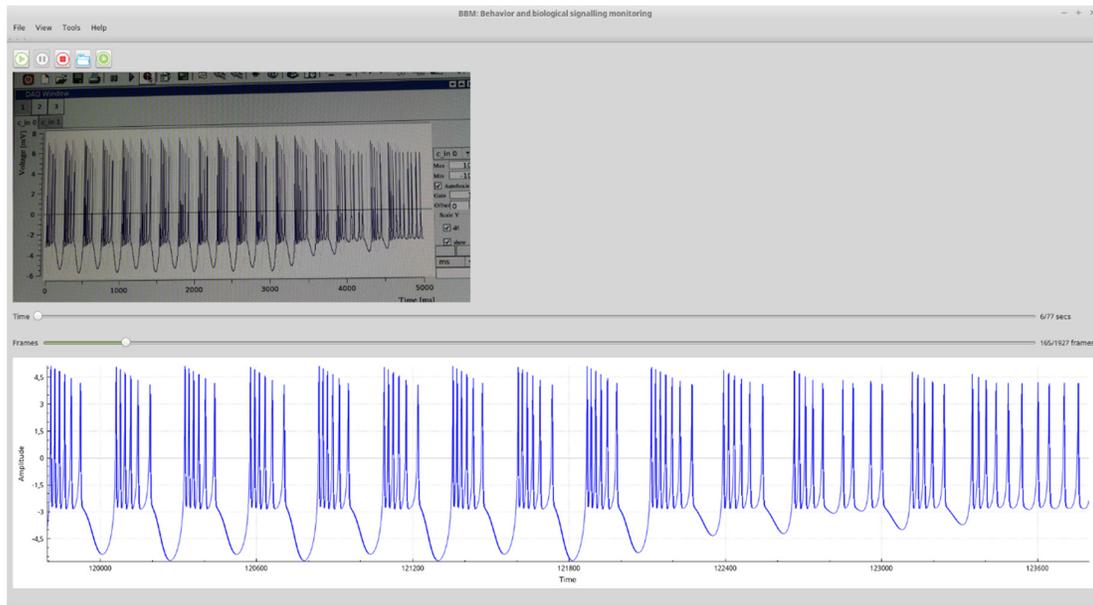


Figura 5-3: Señales sincronizadas

Si la señal sigue avanzando, se puede observar que siguen sincronizadas:

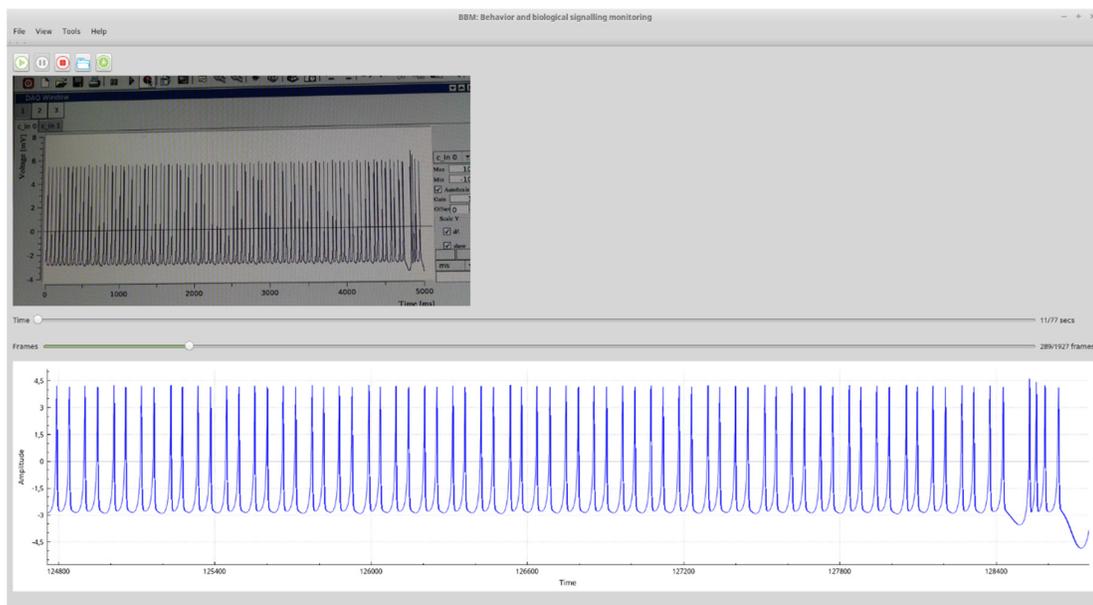


Figura 5-4: Las señales siguen sincronizadas después de un tiempo

5.2 Registros simultáneos de EEG y de actividad ocular en vídeo

En esta ocasión se utilizará la señal temporal de un electrodo en la parte occipital de la cabeza procedente registrada con el amplificador a una frecuencia de adquisición de 256 Hz. Simultáneamente se registró la actividad ocular con la cámara de vídeo. Para la sincronización de la señal de EEG y del vídeo se consideraron inicialmente los artefactos en los registros de EEG provocados por el movimiento ocular y muscular de la mandíbula.

En el vídeo, el sujeto cierra los ojos y parpadea varias veces seguidas, lo que provoca una gran distorsión en el registro de encefalografía. Más adelante, mueve la mandíbula y parpadea levemente, lo que provoca una distorsión menor en la señal temporal. Es por eso que se utilizó el primer parpadeo para sincronizar la señal.

En un principio, se aprecia que las señales no están sincronizadas. Al empezar a reproducir, en el video el sujeto todavía no ha parpadeado, pero se observa desde el principio la distorsión en la señal temporal. Pasan unos segundos en el video hasta que el sujeto parpadea. En concreto, en el fotograma 117 se ha capturado la pantalla para mostrar que el usuario todavía sigue con los ojos abiertos y la distorsión es claramente visible desde el inicio de la reproducción de la señal temporal. Ver figura 5-5.

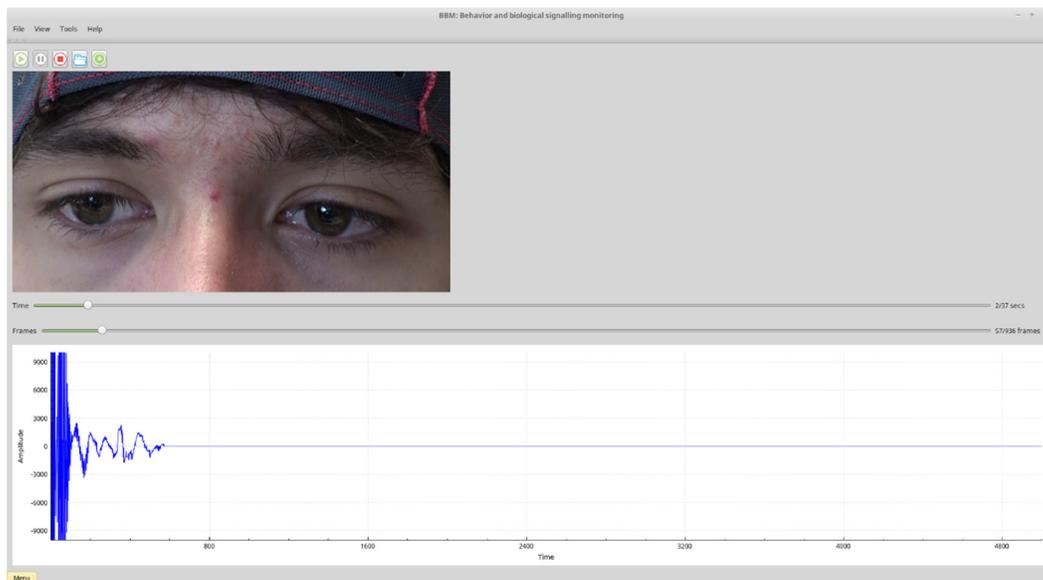


Figura 5-5: La distorsión en la señal temporal se observa desde el principio de su reproducción. Sin embargo, el usuario todavía no ha parpadeado en el fotograma 57.

El primer parpadeo se produce aproximadamente en el frame 133, con el usuario cerrando momentáneamente los ojos y seguido de unos parpadeos muy rápidos. Ver figura 5-6.

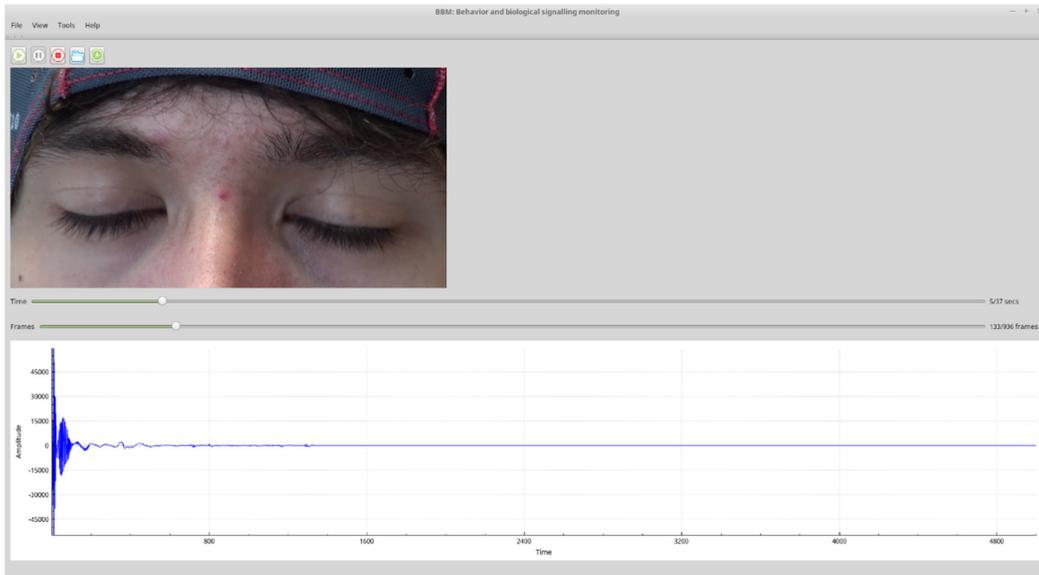


Figura 5-6: Primer momento en el que se aprecia el parpadeo en el video. Es en el fotograma 133 aproximadamente.

Vamos a proceder pues a la sincronización de la señal. Se escoge el valor 133 para la señal de video, como se aprecia en la figura 5-7.

Figura 5-7: Valor de sincronización de la señal de video.

A continuación se reproducen las dos señales juntas y como se puede visualizar en la figura 5-8, el primer fotograma que se muestra en el video es en el que el sujeto acaba de empezar a parpadear, y esta sincronizado con la distorsión de la señal temporal.

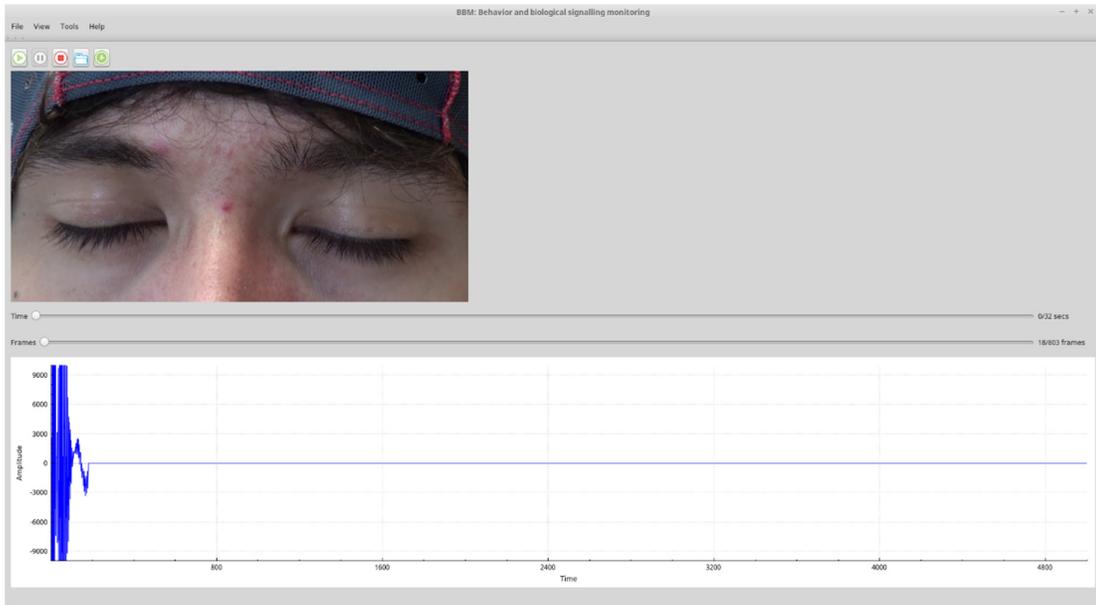


Figura 5-8: Señales sincronizadas

A continuación, el sujeto abre los ojos y se aprecia como lo hace justo en el último pico de la distorsión de la señal temporal, ver figura 5-9.

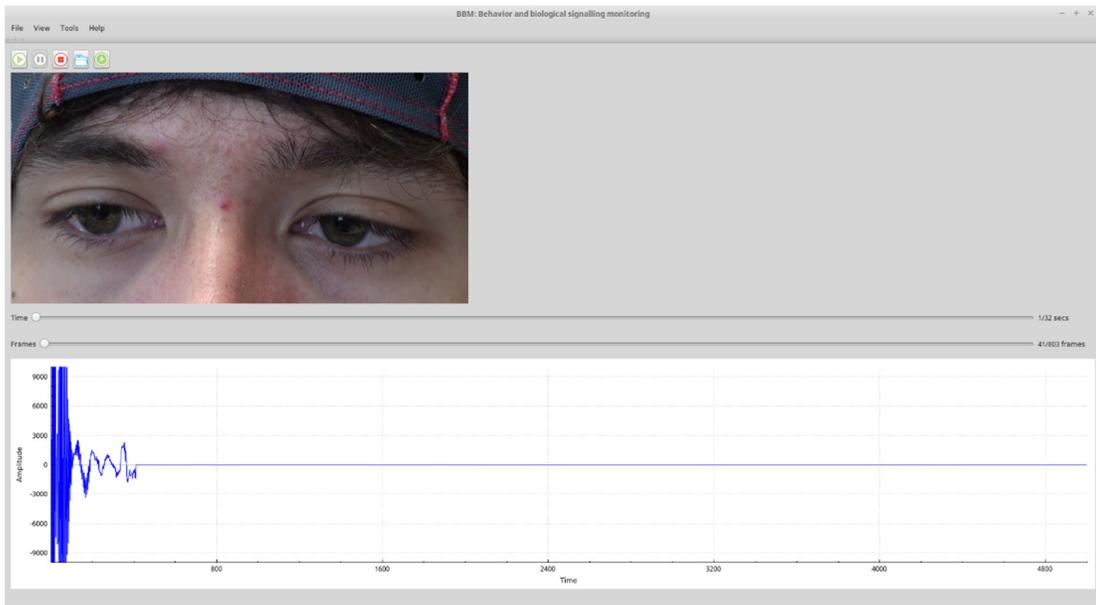


Figura 5-9: El sujeto abre los ojos después de tenerlos cerrados.

Justo después de eso, el sujeto vuelve a parpadear y se observa otro pico en la señal apreciable en la figura 5-10.

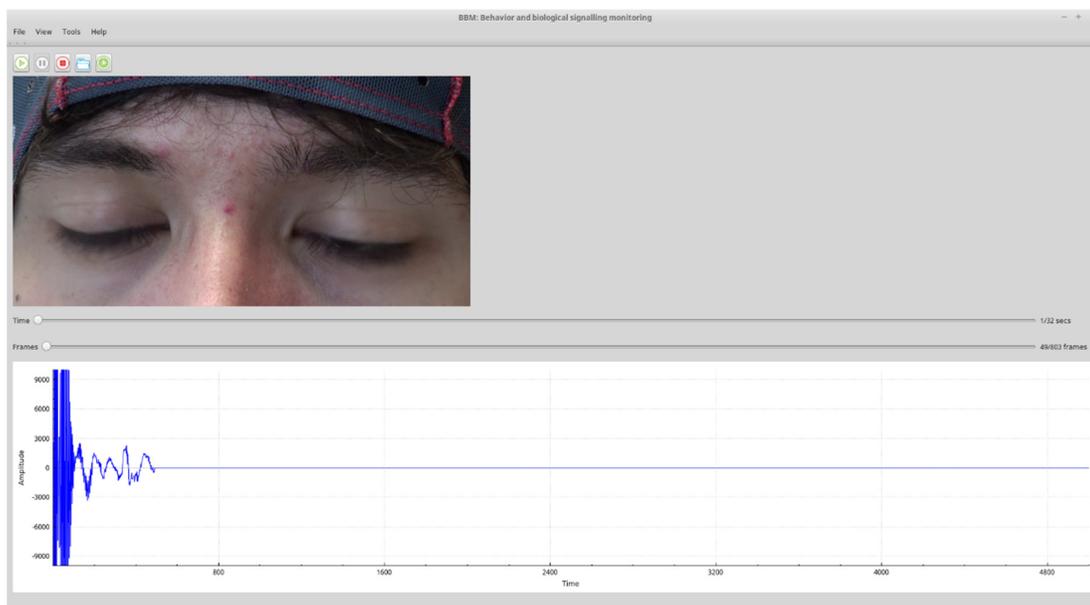


Figura 5-10: El sujeto parpadea y se observa un pico en la señal temporal

5.3 Monitorización en video del comportamiento del *Gnathonemus Petersii* y su actividad eléctrica

Para esta prueba, se ha grabado el comportamiento del pez durante un periodo de 636 segundos (10 minutos y 36 segundos). Se ha utilizado un formato de reproducción mkv con frecuencia de adquisición 25 fps y una frecuencia de 15KHz para la señal eléctrica. Por tanto, se representarán 600 muestras eléctricas por cada fotograma.

Como eventos para sincronizar la señal del video con la señal fisiológica del pez, en el experimento se enciende y se apaga una luz al principio de la señal y otra vez casi al final del video. Esta acción, es visible en el video como se puede apreciar en la imagen 5-11. Se ha marcado el área de la imagen donde es más fácil distinguir a simple vista la diferencia entre luz apagada y luz encendida:

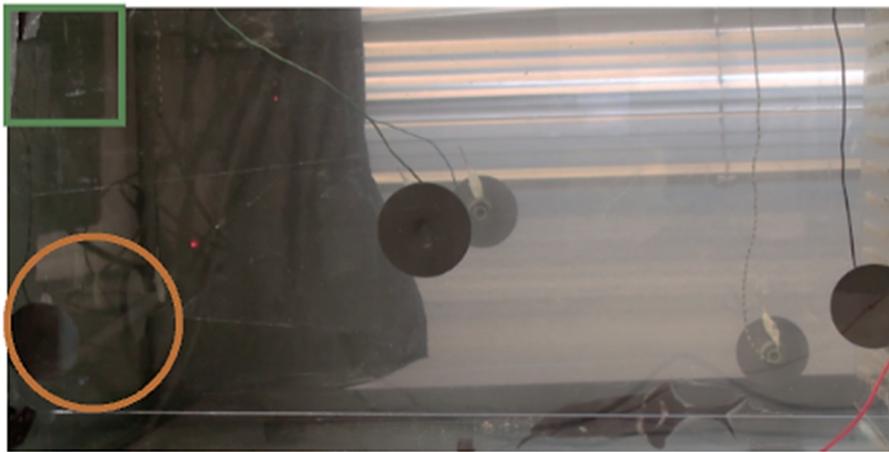
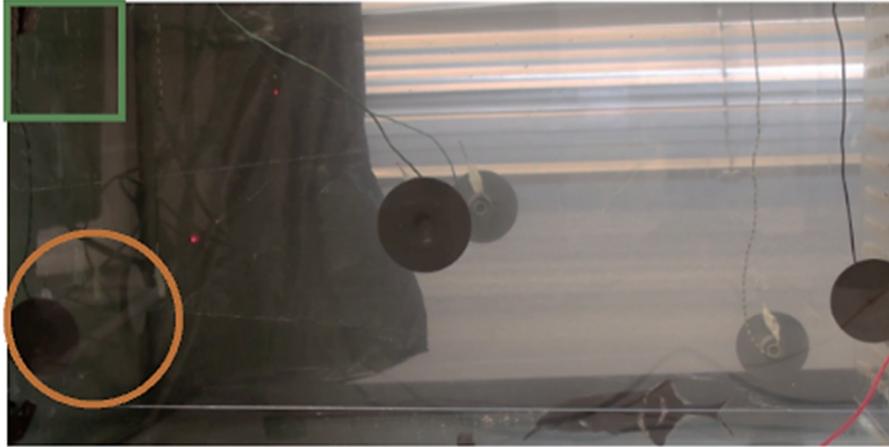


Figura 5-11: Arriba, fotograma del video con la luz apagada. Abajo, fotograma del video con la luz encendida. Las diferencias son más visibles en las zonas marcadas.

El encendido de la luz también provoca una distorsión en la señal eléctrica, como se puede apreciar en la imagen 5-12. Dado que se ha encendido y apagado la luz dos veces, una al principio de la adquisición de la señal y otra al final, se pueden apreciar dos artefactos en la señal fisiológica.

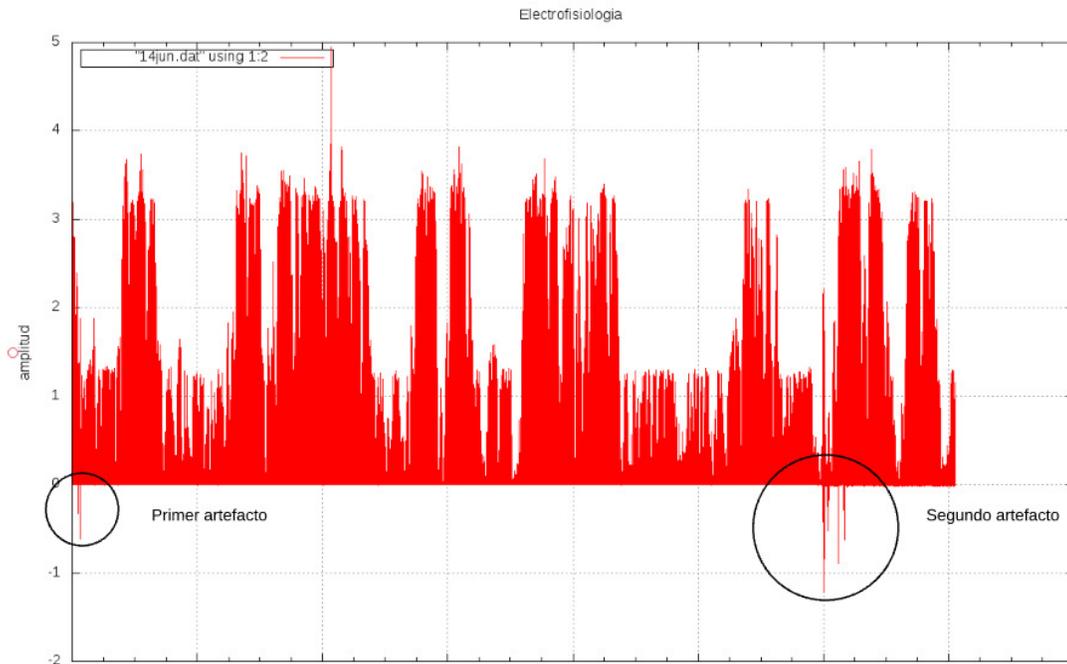


Figura 5-12: Señal temporal representada en su totalidad. Al principio y al final del video son apreciables sendas distorsiones por el encendido de la luz.

Para estos dos ficheros, la señal de video empieza a ser grabada antes y un poco más tarde empieza la adquisición de la señal eléctrica. Hay una diferencia de varios segundos entre los inicios de las dos señales, por lo que están claramente desincronizadas.

Si las dos señales se reproducen a la vez, se pueden apreciar dos cosas durante el primer segundo. Una, que la luz está apagada, y la segunda es que el tiempo empieza en cero. En la figura 5-13 no se ve exactamente el valor cero porque han pasado unos milisegundos desde que se pulsa reproducir y se muestra el primer fotograma hasta que se pausa la imagen para poder hacer la captura de pantalla.

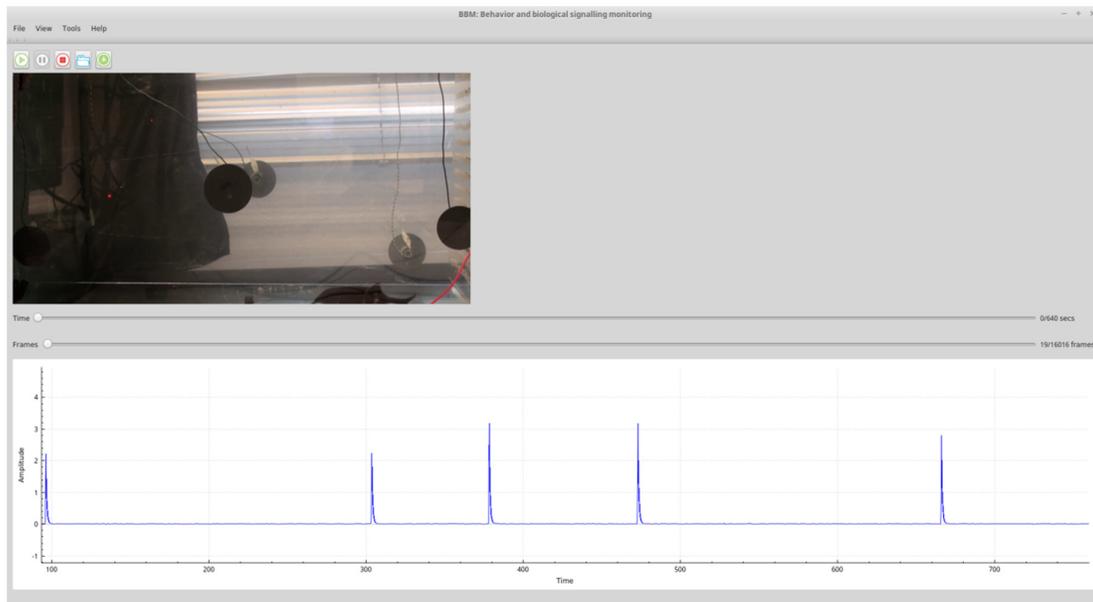


Figura 5-13: Inicio de reproducción de señal de video y señal biológica.

Ahora bien, si se reproduce el video un poco más, se ve el primer artefacto en la señal eléctrica pero la luz todavía está apagada. Como se ha explicado antes, la señal temporal se empezó a grabar en un instante de tiempo posterior, Si las dos señales empiezan a la vez es de esperar que el artefacto sea visible primero en la señal eléctrica y en un instante posterior en el video. En concreto, el artefacto es visible en la señal temporal alrededor del fotograma 115 (figura 5-14) y la luz se enciende en el video alrededor del fotograma 839 (figura 5-15).

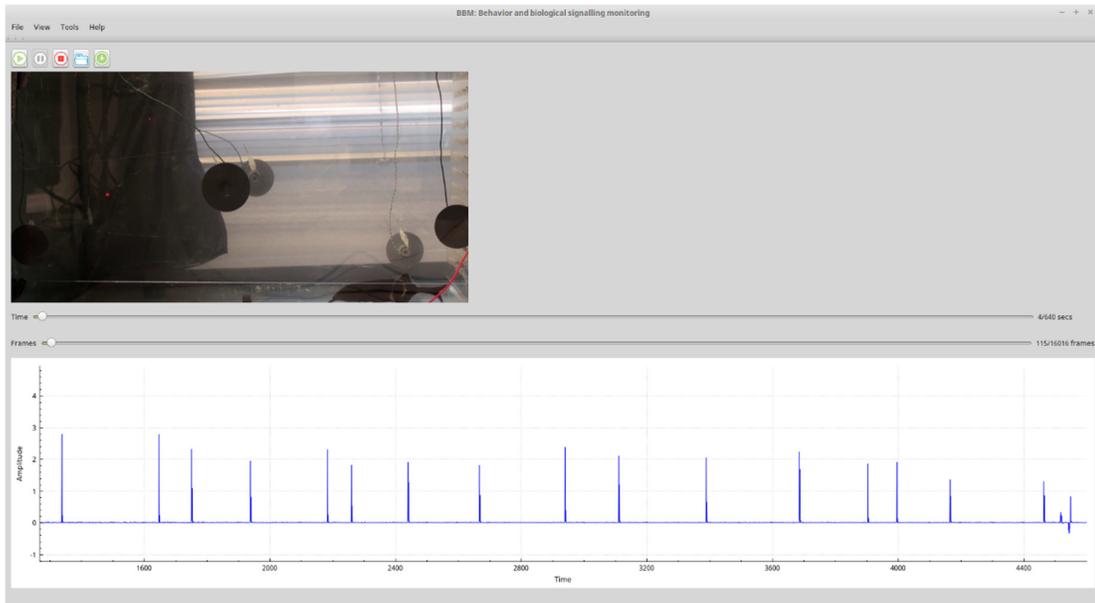


Figura 5-14: Captura de pantalla de la aplicación en el fotograma 115. Se aprecia una distorsión en la señal temporal después del momento 4400 pero se aprecia que la luz todavía no se ha encendido en el vídeo.

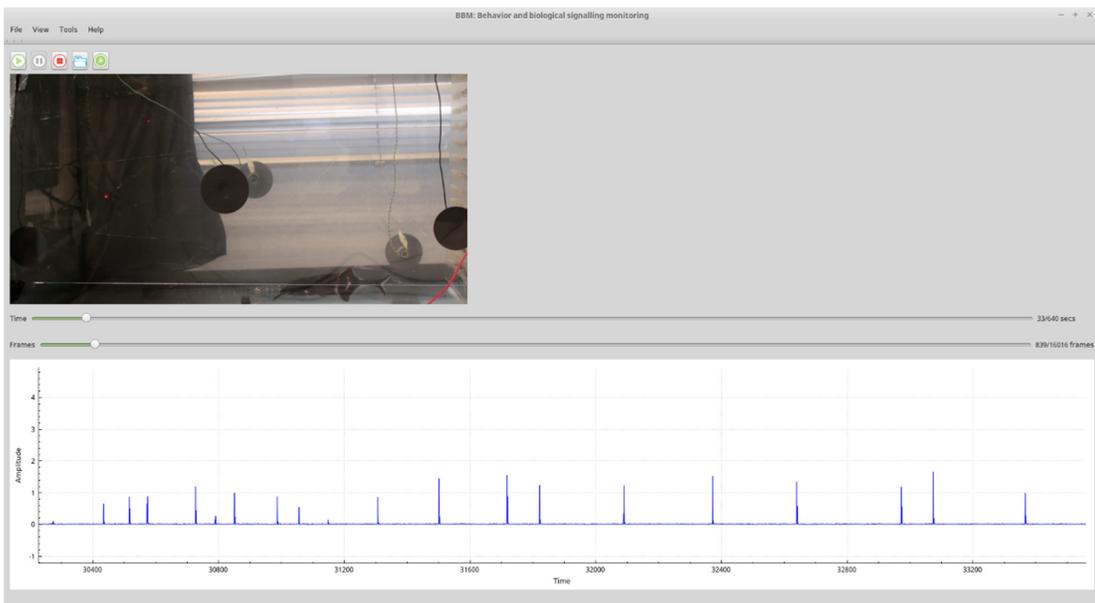


Figura 5-15: Captura de pantalla de la aplicación en el fotograma 839. La luz se ha encendido y es apreciable en el vídeo, no se aprecia ningún artefacto en la señal temporal.

Procedemos pues a sincronizar la señal, para ello se escogen unos valores de fotogramas ligeramente inferiores al momento de los eventos, para que dé tiempo a apreciar en que momento exacto se enciende la luz en el video y evitar perderse el artefacto en la señal eléctrica. En este caso se han utilizado los valores de fotograma 800 para el video y fotograma 100 para la señal eléctrica. Esto se realiza desde el cuadro de diálogo de sincronización, apreciable en la figure 5-16.

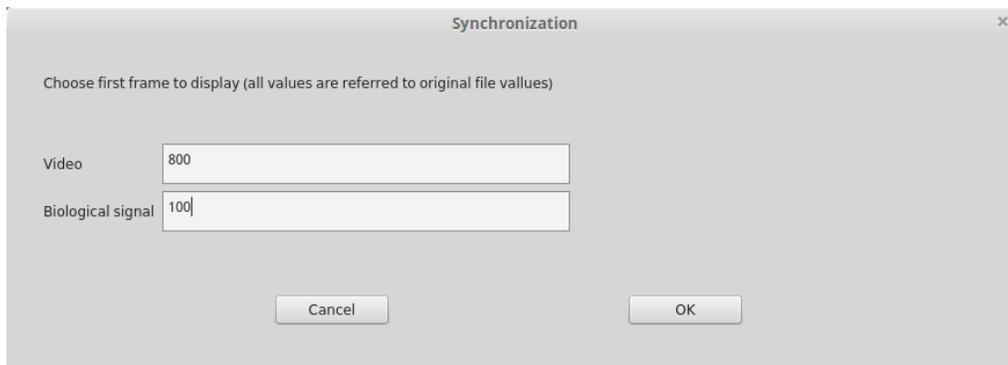


Figura 5-16: Sincronización de la señal de video a 800 fotogramas del comienzo de la señal de video y a 100 fotogramas para la señal temporal respecto al inicio del video también.

Cuando empieza la reproducción, se observa que la luz está apagada y no se observan artefactos en la señal eléctrica. También cabe destacar que la señal eléctrica está empezando en el momento 4000 en vez de en el instante 0 de tiempo. Ver figura 5-17.

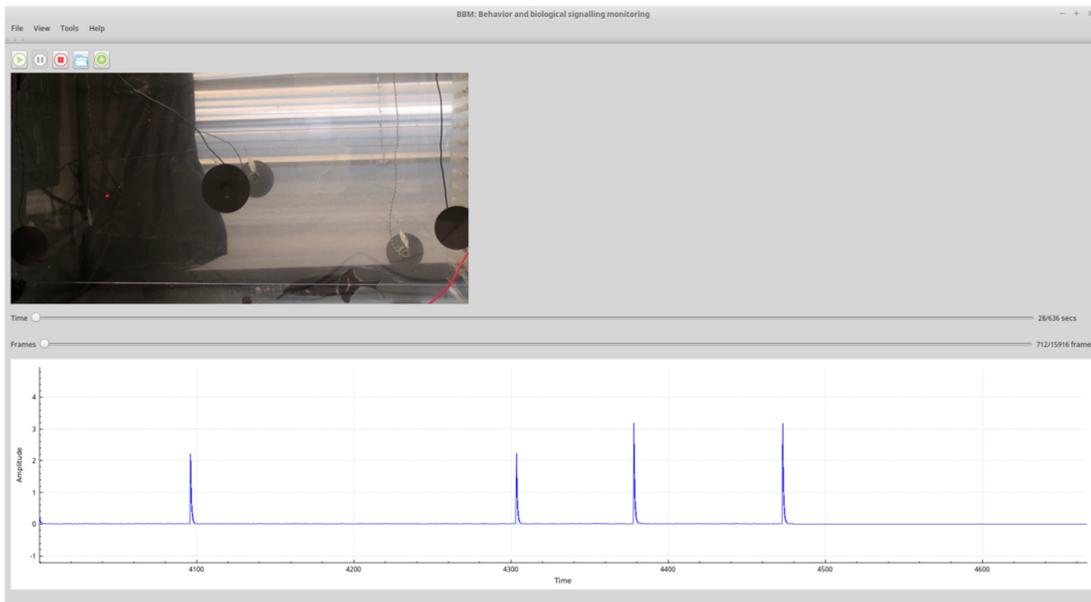


Figura 5-17: Empieza la reproducción del video después de la sincronización.

Si se sigue reproduciendo la señal, unos fotogramas más adelante la luz se enciende y se observa la distorsión en la señal eléctrica como se puede apreciar en la figura 5-18.

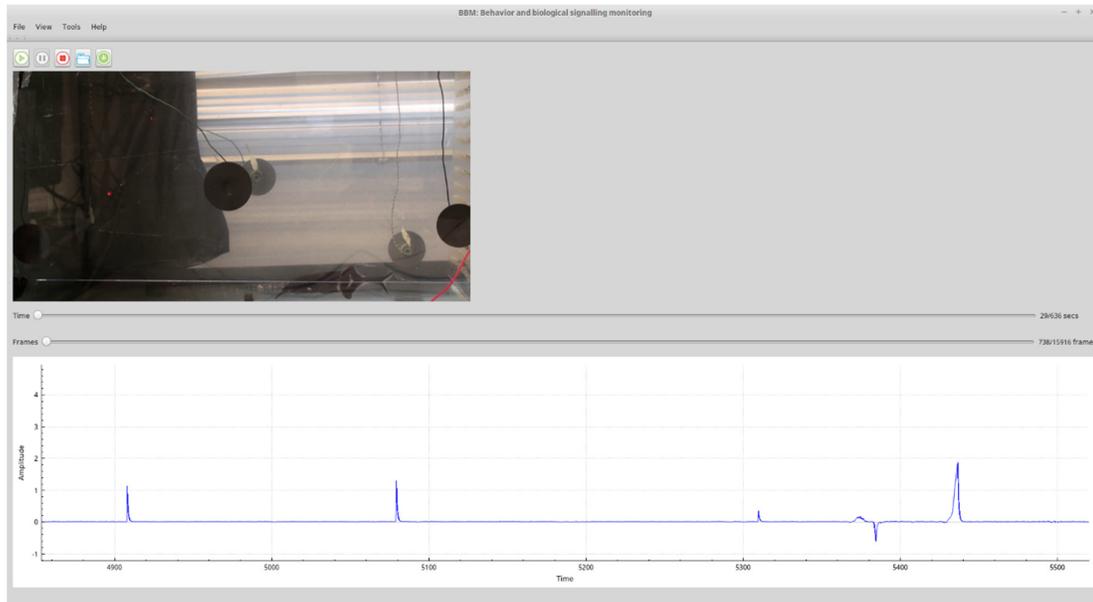


Figura 5-18: En la señal de video la señal está encendida y se aprecia una distorsión en la señal temporal. La señal está sincronizada.

Al continuar reproduciendo el video, también se comprueba que la señal está sincronizada al final del experimento, cuando la luz se vuelve a encender.

En la figura 5-19 se muestran las dos señales reproduciéndose justo antes de que se encienda la luz. Se puede apreciar que en la señal de video la luz está apagada como se espera y la señal eléctrica no muestra el evento todavía. Se puede apreciar en la referencia de tiempo de la señal eléctrica que tiene un valor de 515700, es decir, 515,7 segundos.

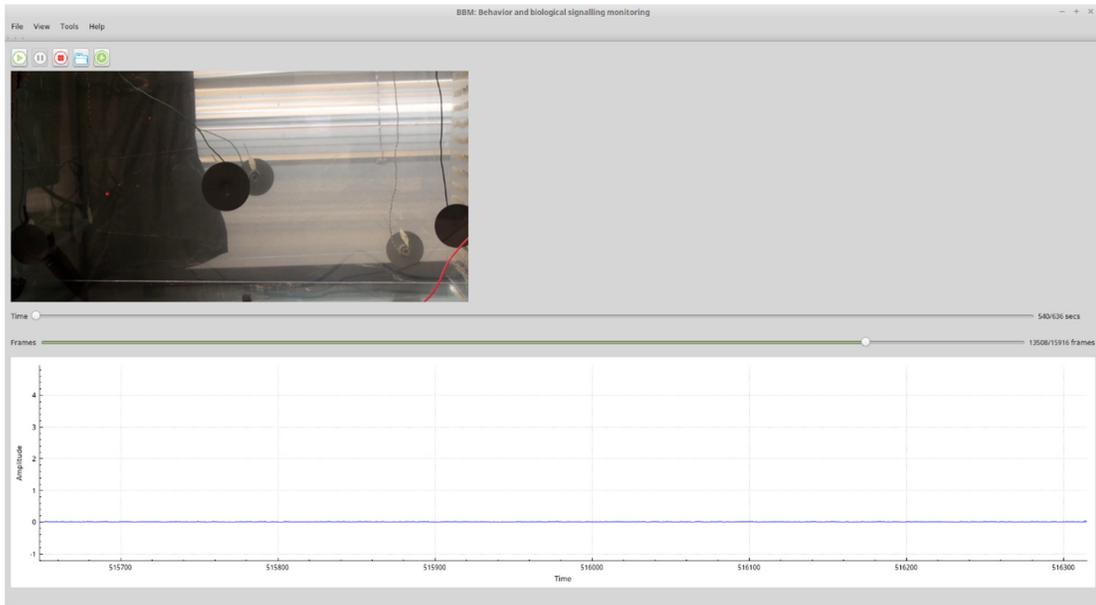


Figura 5-19: Señales reproduciéndose justo antes de que se encienda la luz por segunda vez.

Unos segundos más tarde, en concreto en la referencia temporal de la señal eléctrica 518200 (518,2 segundos) se ve que la luz está encendida y al mismo tiempo es observable en la señal temporal. Ver figura 5-20.

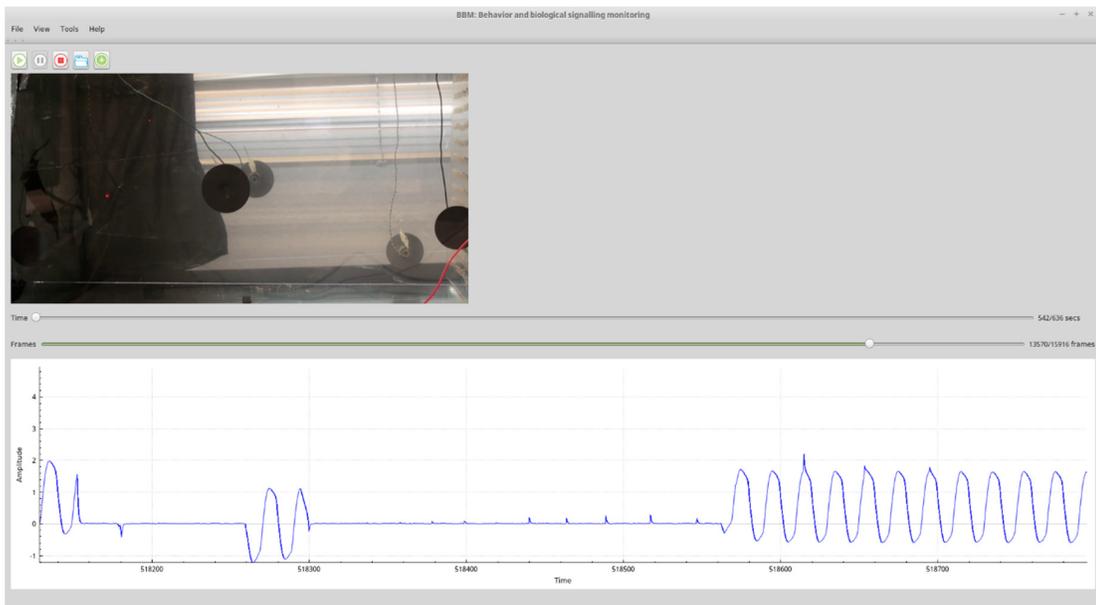


Figura 5-20: La luz está encendida como se puede apreciar en la señal de video y también se aprecia el artefacto en la serie temporal por el encendido de la luz.

La sincronización, por tanto, ha sido exitosa, coinciden los dos artefactos en ambas señales.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

La monitorización simultánea de comportamiento y actividad electrofisiológica puede contribuir en gran medida al entendimiento de numerosos fenómenos observados en investigación en neurociencia básica, neuroetología y en situaciones clínicas. En este trabajo se ha desarrollado una herramienta para realizar la sincronización y monitorización simultáneas de este tipo de señales. Como se ha demostrado, el programa cumple con la funcionalidad esperada. Se puede reproducir una señal de video y una señal temporal simultáneamente y de forma sincronizada considerando la diferencia de muestreo en las señales, y pudiendo pausar, o parar completamente la visualización. Además, se puede avanzar o retroceder en la señal utilizando las barras de navegación.

Se ha creado un cuadro de diálogo para que el usuario introduzca los ficheros y otro para la sincronización de la señal. La sincronización permite empezar la reproducción de los ficheros en momentos distintos del primer fotograma y con ello que las dos señales tengan una referencia temporal común. Las pruebas realizadas con registros simultáneos de actividad eléctrica de un pez y de su comportamiento en vídeo, así como el registro simultáneo de EEG y de actividad ocular en vídeo han validado su uso en los contextos de neuroetología y de vídeo-EEG.

6.2 Trabajo futuro

Respecto a las mejoras que se pueden aplicar al programa, son principalmente de cara a la integración con otras plataformas y también para añadir funcionalidad al sistema. A continuación se presenta un resumen de las más relevantes:

1. Implementación para Windows y macOS

La integración de este programa en distintos sistemas operativos requiere la modificación de algunos parámetros del código como por ejemplo el árbol de directorios. Dado que el sistema está pensado para ejecutarse en Linux, se utiliza la notación “/directorio/archivo.mkv” para acceder a los ficheros de video y señal de fisiología, con los consecuentes chequeos para ver si es un directorio válido, mientras que en Windows la

estructura sería del tipo “C:\directorio\archivo.mkv”. La empaquetación del código también sería diferente, entre otros. Las opciones de implementación con OpenCV y Qt facilitan que la aplicación pueda ser multiplataforma.

2. Pantalla completa para la señal de video y/o de fisiología

Esta opción permitiría observar en pantalla completa ambas señales en distintas resoluciones, lo que puede ser útil para discernir con claridad algunos detalles del experimento mostrado.

3. Aplicar zoom a la señal de fisiología

Con el objeto de estudiar más al detalle la señal biológica, agrandar y empequeñecer la señal mientras la reproducción está en pausa puede dar una visión más clara al usuario de que cambios se están produciendo en la señal. Se recuerda que la frecuencia de muestreo en el vídeo siempre es muy inferior a la frecuencia de muestreo de las señales electrofisiológicas. De la misma manera se podrían aplicar distintas transformaciones y operaciones de análisis en la señal de fisiología.

4. Reproducción de varios videos simultáneamente

Esta funcionalidad puede resultar altamente útil para el tipo de casos de uso posibles de este programa en el contexto de aplicaciones en investigación básica y clínica. Por ejemplo, a la hora de grabar el comportamiento del pez, se puede grabar el experimento desde varios ángulos y posteriormente visualizar de manera sincronizada ambos videos y la señal de fisiología. Para tal efecto, dado que el video está actualmente alineado a la izquierda, se podría cómodamente representar otro vídeo en el espacio que hay a la derecha de este.

5. Incluir marcadores para eventos

Debido a que puede haber varios momentos de especial interés en las señales, se podría implementar un sistema de marcación de determinados momentos en los ficheros, de

manera que se puedan guardar los ficheros con esta nueva información y posteriormente visualizarlos y poder saltar hasta los eventos pulsando un botón.

6. Introducción de atajos en la reproducción

Todas las acciones disponibles en el programa, son susceptibles de tener un atajo de teclado, como por ejemplo reproducción, pausa, parada, abrir el diálogo para cargar un archivo... Lo que aportaría usabilidad y rapidez de manejo.

7. Reproducción de varias señales de fisiología a la vez

Esta nueva característica, podría ser de especial utilidad en el caso de la interfaz cerebro máquina. Debido a que se toman medidas simultáneamente con varios electrodos, representar estas señales a la vez en distintos colores puede ser de especial utilidad para el investigador.

Referencias

- [1] and S. M. Ching-Wei Wang*, Member, IEEE, Andrew Hunter, Member, IEEE, Neil Gravill, “Unconstrained Video Monitoring of Breathing Behavior and Application to Diagnosis of Sleep Apnea,” *IEEE Trans. Biomed. Eng.*, vol. 61, 2014.
- [2] P. Chamorro, C. Muñiz, R. Levi, D. Arroyo, F. B. Rodríguez, and P. Varona, “Generalization of the dynamic clamp concept in neurophysiology and behavior,” *PLoS One*, vol. 7, no. 7, pp. 1–10, 2012.
- [3] D. Press, “Long-term video EEG monitoring for diagnosis of psychogenic nonepileptic seizures,” pp. 1979–1986, 2014.
- [4] V. H. García García, “Desarrollo de una ‘toolbox’ para condicionar el comportamiento de peces eléctricos, basado en la codificación de las señales eléctricas emitidas por éste y su comportamiento.” 01-Jul-2014.
- [5] C. G. Forlim, C. Muñiz, R. D. Pinto, F. B. Rodríguez, P. Varona, “Behavioral driving through on line monitoring and activity-dependent stimulation in weakly electric fish,” *BMC Neurosci. 2013 141*, vol. 14, no. 1, p. e40887, 2013.
- [6] C. G. Forlim, R. D. Pinto, P. Varona, F. B. Rodríguez, “Delay-Dependent Response in Weakly Electric Fish under Closed-Loop Pulse Stimulation,” *PLoS One*, vol. 10, no. 10, p. e0141007, Oct. 2015.
- [7] Á. M. García, “Diseño de interfaces cerebro-máquina controlados mediante registros de EEG,” 2015.
- [8] M. Marcos, “Diseño y control de un robot mediante interfaz cerebro-máquina usando la tecnología SSVEP,” 2015. [Online]. Available: <http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20150716ManuelMarcosGarcia.pdf>.
- [9] J. Fernandez-Vargas, H. U. Pfaff, F. B. Rodríguez, and P. Varona, “Assisted closed-loop optimization of SSVEP-BCI efficiency.,” *Front. Neural Circuits*, vol. 7, p. 27, 2013.

- [10] L. Hänninen and M. Pastell, "CowLog: open-source software for coding behaviors from digital video.," *Behav. Res. Methods*, vol. 41, no. 2, pp. 472–476, 2009.
- [11] A. Ojeda, N. Bigdely-Shamlo, and S. Makeig, "MoBILAB: an open source toolbox for analysis and visualization of mobile brain/body imaging data.," *Front. Hum. Neurosci.*, vol. 8, no. March, p. 121, 2014.
- [12] O. Sourina, Y. Liu, and M. K. Nguyen, "Real-time EEG-based emotion recognition for music therapy," *J. Multimodal User Interfaces*, vol. 5, no. 1–2, pp. 27–35, Mar. 2012.
- [13] G. von der Emde and S. Fetz, "Distance, shape and more: recognition of object features during active electrolocation in a weakly electric fish.," *J. Exp. Biol.*, vol. 210, no. Pt 17, pp. 3082–95, Sep. 2007.
- [14] Cowlog, "<http://cowlog.org/download/>." .
- [15] AcqKnowledge, "<http://www.biopac.com/product/acqknowledge-software/>." .
- [16] Datagraph, "<http://www.visualdatatools.com/DataGraph/>." .
- [17] A. Magaudda, A. Laganà, A. Calamuneri, T. Brizzi, C. Scalera, M. Beghi, C. M. Cornaggia, and G. Di Rosa, "Validation of a novel classification model of psychogenic nonepileptic seizures by video-EEG analysis and a machine learning approach.," *Epilepsy Behav.*, vol. 60, pp. 197–201, Jul. 2016.
- [18] T. Kobulashvili, J. Höfler, J. Dobesberger, "Current practices in long-term video-EEG monitoring services: A survey among partners of the E-PILEPSY pilot network of reference for refractory epilepsy and epilepsy surgery.," *Seizure*, vol. 38, pp. 38–45, May 2016.
- [19] Matlab, "<http://www.mathworks.com/>."
- [20] "OpenCV | OpenCV." [Online]. Available: <http://opencv.org/>. [Accessed: 19-Feb-2015].
- [21] Qt, "<https://www.qt.io/>."

7 Glosario

Artefacto	Distorsión en una señal de fisiología que en este proyecto se utiliza para sincronizar la señal con el vídeo.
AVI	(Audio Video Interleave) Formato contenedor de audio y video lanzado por Microsoft en 1992.
ECG	El electrocardiograma (ECG) es la representación gráfica de la actividad eléctrica del corazón
EEG	La electroencefalografía (EEG) es una medida neurofisiológica que se basa en el registro de la actividad eléctrica cerebral. Puede realizarse en distintos periodos: reposo, vigilia, o sueño
Electrofisiología	Es el estudio de las propiedades eléctricas de células y tejidos biológicos. Incluye medidas de cambio de voltaje o corriente eléctrica en una variedad amplia de escalas, desde el simple canal iónico de proteínas hasta órganos completos como el corazón. En neurociencia, se incluyen las medidas de la actividad eléctrica de neuronas, y particularmente actividad de potencial de acción. Registros a gran escala de señales eléctricas del sistema nervioso como Electroencefalografía, también se pueden clasificar como registros electrofisiológicos
ERP	Un potencial relacionado con evento (ERP) es la medida de la respuesta cerebral que es resultado directo de un evento específico sensorial, cognitivo o motriz. De manera más formal, es cualquier respuesta electrofisiológica estereotipada a un estímulo
Etología	Ciencia que estudia el comportamiento de los animales
FPS	Fotogramas por segundo
GUI	Interfaz gráfica de usuario
Linux	Sistema operativo basado en Unix
MacOS	Sistema operativo basado en Unix. Creado por Apple.
Matlab	Lenguaje de programación de cálculo técnico de alto nivel y también un entorno de desarrollo. Está especializado en la visualización y análisis de datos y pertenece a la categoría de lenguajes de scripting
Mkv	Matroska o mkv es un formato contenedor de estándar abierto, un archivo informático que puede contener una cantidad ilimitada de vídeo, audio, imagen o pistas de subtítulos dentro de un solo archivo. ¹ Su finalidad es la de servir como formato universal para el almacenamiento de contenidos audiovisuales y multimedia, como películas o programas de televisión, videojuegos, imágenes y textos. No es un códec en sí sino un contenedor de formatos
Mov	Extensión de archivo usada por el formato QuickTime

Mp4	MPEG-4 Parte 14 es un formato AAC de compresión de datos de audio desarrollado conjuntamente con algunas empresas privadas como AT&T, Nokia, Sony y Dolby. El formato MPEG-4 Parte 14 tiene diversos nombres populares y está relacionado a nivel de programación, con otros formatos AAC anteriores
MTS	Formato de archivo que se utiliza para reproducir vídeos en alta definición. Es comúnmente conocido como AVCHD (Advanced Video Coding High Definition). Esta extensión de archivo soporta resoluciones de 720p y 1080i.
OpenCV	Conjunto de librerías en C++ especializadas en la gestión, análisis y procesado de imágenes en tiempo real
Signal and slot	Las señales (signal) son notificaciones de un evento en un programa. Cuando una determinada acción ocurre, se emite una señal que puede ser predefinida o creada por el programador. Cada señal puede estar conectada a uno o varios slots, que son básicamente funciones a las que se las llama cuando este evento ocurre.
Qt	Qt es una biblioteca multiplataforma disponible para C++ y Python. Se utiliza principalmente en la creación de interfaces de usuario, dado su amplio catálogo de elementos gráficos listos para usar como botones, barras de navegación, diálogos... También cabe destacar su gestión de ventanas y su mecanismo de comunicación entre objetos
QTimer	Clase perteneciente a Qt y tipo también tipo de datos. Su finalidad es la implementación de funciones para un uso básico de contadores.
Widget	Un widget es una pequeña aplicación. Entre sus objetivos está el dar fácil acceso a funciones frecuentemente usadas. Puede tener herencia, aplicándose ciertos parámetros a todos los widgets hijo.

8 Anexos

8.1 Presupuesto

	Coste
Hardware	
Ordenador con tarjeta DAQ NI PCI-6521	3000 €
Amplificador USBAMP y casco de EEG	12.350 €
Circuitería electrodos pecera	310 €
<hr/>	
Gastos relacionados con el pez elefante	
Adquisición de los ejemplares de <i>Gnathonemus Petersii</i>	35 €
Alimentación y mantenimiento del acuario	50 €
<hr/>	
Software	
SDK para USBAMP	250 €
<hr/>	
Ejecución material y honorarios	
Subtotal de ejecución material (hardware y software)	15995 €
Gastos generales (16% del subtotal)	2559,2 €
Beneficio industrial (6% del subtotal)	959,7 €
Honorarios (600 horas a 15€/hora)	9000 €
<hr/>	
Material fungible	
Agua ionizada	10 €
Electrodos para el casco de EEG	2916€
Fotocopias	10 €
Gastos de impresión	120 €
Encuadernaciones	50 €
<hr/>	
Totales	
Subtotal del presupuesto	28101 €
I.V.A. Aplicable (21%)	4496,16 €
Total presupuesto	32597,16 €

Madrid, Junio de 2016

El ingeniero jefe de proyecto

Fdo.: Alicia García Sastre

Ingeniera de Telecomunicación

8.2 PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de una aplicación de monitorización simultánea de comportamiento y señalización biológica eléctrica. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partidaalzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción

provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.

