

UNIVERSIDAD DE VALLADOLID



E.T.S.I.T TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS
DE TELECOMUNICACIÓN

**Diseño e implementación de un
sistema de captura de datos de
entrenamiento para la realización
de eye-tracking sin hardware
específico**

Autor: Raúl Barba Alonso

Tutores: Rodrigo de Luis García y Federico Simmross
Wattenberg

Septiembre 2021

Resumen

El seguimiento ocular o eye-tracking es un método para obtener información sobre los movimientos oculares, lo que a su vez proporciona información sobre los procesos cerebrales que los controlan. El estudio de los movimientos oculares permite a esta tecnología obtener información, por ejemplo, sobre procesos cognitivos, patologías neurológicas o alteraciones psicológicas. Tradicionalmente, se han utilizado para realizar seguimiento ocular tecnologías que requerían de un hardware específico, teniendo por lo tanto un elevado coste económico y poca disponibilidad. Este trabajo pretende dar un primer paso hacia la realización de seguimiento ocular simplemente utilizando la webcam presente en todos los ordenadores portátiles actuales. Para ello se ha construido una plataforma web capaz de capturar datos de los movimientos oculares de voluntarios mientras realizan una serie de tareas, con el objetivo de construir una base de datos con la que entrenar un sistema de aprendizaje automático para determinar la posición de la mirada del usuario sobre la pantalla a partir de una imagen de su rostro y ojos.

Este trabajo se centra en el diseño, desarrollo y despliegue de dicha plataforma web.

Abstract

The term eye-tracking refers to the technology employed to capture and analyze the eye movements of a person while he/she performs a certain task. These movements are related to the neural processes that control them, and thus this technology allows the study of cognitive processes, neurological pathologies or psychological alterations, among others. Traditionally, eye-tracking has been performed using specific hardware, which is expensive and hinders its widespread use. This work aims at providing a first step towards performing eye-tracking using only the webcam that is incorporated in every modern laptop. To that end, we have created a web platform that is able to capture the ocular movements of volunteers while they perform a series of tasks, in order to build a dataset that will be used to train a machine learning system that can determine the position of the gaze within the screen from an image of the face and eyes of the user.

This work is focused on the design, development and deployment of such a web platform.

Palabras clave

Eye-tracking, experimentos, aplicación web, captura de datos.

Agradecimientos

En estas líneas me gustaría agradecer a todas las personas que me han acompañado y ayudado a llegar hasta aquí.

En primer lugar a mis padres y mi hermano, quienes me han apoyado en todo momento a lo largo de estos años de carrera.

En segundo lugar agradecer a mis amigos, tanto los conocidos en la carrera como los que tenía en el instituto que me han hecho seguir hacia delante en los momentos más duros, pero también he disfrutado mucho a su lado.

Doy las gracias también a los tutores de este proyecto, Rodrigo y Federico, por ayudarme en el desarrollo de este proyecto resolviendo mis dudas.

Índice general

1. Introducción	13
1.1. Motivación	13
1.2. Objetivos	14
1.3. Fases y métodos	15
1.4. Estructura de la memoria	16
2. Estado del arte	17
2.1. Introducción	17
2.2. Sistema visual humano	18
2.2.1. Estructura anatómica del ojo humano	18
2.2.2. Psicofísica visual	20
2.2.3. Modelos de movimientos oculares	22
2.3. Sistemas de eye-tracking	23
2.3.1. Electrooculografía (EOG)	24
2.3.2. Lentes de contacto combinadas con una bobina	25
2.3.3. Foto-oculografía y Vídeo-oculografía	25
2.4. Eye-tracking a partir de video natural	28
2.4.1. GazeCapture	28
2.4.2. MPIIGaze	29
3. Tecnologías base	31
3.1. Introducción	31
3.2. HTML5	31
3.3. CSS	31
3.4. Javascript	32
3.5. PHP	32
3.6. Apache HTTP Server	32
3.7. MySQL	33
3.8. Astah	34
4. Análisis y diseño de la aplicación	35
4.1. Introducción	35
4.2. Análisis de la aplicación	35
4.2.1. Especificación de requisitos	35
4.2.2. Casos de uso	37
4.3. Diseño	43
4.3.1. Diseño de la interfaz de usuario	43
4.3.2. Diseño de la base de datos	47

5. Implementación y resultados	53
5.1. Introducción	53
5.2. Interfaz de usuario	53
5.2.1. Formulario de la aplicación	53
5.2.2. Obtención de los parámetros de la cámara	56
5.2.3. Realización del experimento 1	56
5.2.4. Realización del experimento 2	61
5.2.5. Mostrar puntuaciones	63
5.3. Interfaz de administrador	65
5.4. Pruebas	65
5.4.1. Pruebas realizadas en el servidor local	65
6. Manuales	69
6.1. Manual de usuario de la aplicación	69
6.1.1. Ventana principal	69
6.1.2. Registro, modificación de usuario e inicio de sesión	70
6.1.3. Realización de experimentos	75
6.1.4. Ventana de puntuaciones y errores	79
6.2. Manual de administrador de la aplicación	79
6.2.1. Listar experimentos	81
6.2.2. Descargar experimentos	81
6.3. Manual de mantenimiento de la aplicación	83
6.3.1. Instalar apache, mysql y php	83
6.3.2. Crear un host virtual para el sitio web	84
6.3.3. Probar la conexión con la base de datos desde php	86
7. Conclusiones	89
7.1. Conclusiones	89
7.2. Líneas futuras	90

Índice de figuras

2.1. Anatomía del ojo humano	19
2.2. Diagrama esquemático de las conexiones neuronales entre las células ganglionar, bipolar, horizontales y amacrinas. [1]	19
2.3. Distribución de los conos y bastones en la retina en función del ángulo visual. [1]	20
2.4. Gráfica en la que se muestra la sensibilidad al contraste en función de la frecuencia temporal. [1]	22
2.5. Estructura muscular del ojo. 1-Recto superior; 2- Elevador del párpado superior; 3- Recto lateral; 4- Recto medial; 5- Oblicuo superior;6- Tendón reflejado del oblicuo superior. 7- Anillo de Zinn; 8- Recto inferior; 9- Oblicuo inferior. [1]	22
2.6. Ejemplo de medición de electrooculografía (EOG). [1]	24
2.7. Ejemplo de bobina incrustada en lente de contacto y el marco de campos electromagnéticos para la medición del movimiento ocular. [1]	24
2.8. Ejemplo de inserción de una lente de contacto para medición de movimiento ocular. [1]	25
2.9. Ejemplos de mediciones del movimiento ocular por reflejo de la pupila, el limbo y la córnea. [1]	26
2.10. Ejemplo de un antiguo rastreador de ojos basado en vídeo montado sobre una mesa. [1]	27
2.11. Ejemplo de un rastreador ocular basado en vídeo montado en la cabeza [1]	27
2.12. Ejemplo de un antiguo rastreador de ojos basado en vídeo montado sobre una mesa. [1]	27
2.13. Gafas Tobbi pro. [2]	28
2.14. Línea de tiempo de visualización de un punto. [3]	29
3.1. Resumen gráfico de las tecnologías utilizadas.	34
4.1. Diagrama de casos de uso de la aplicación Eye-tracking	38
4.2. Diagrama de de secuencia de iniciar sesión	41
4.3. Diagrama de de secuencia de realizar el experimento número 1.	42
4.4. Diagrama de de secuencia de realizar el experimento número 2.	42
4.5. Diagrama de de secuencia de ver lista de experimentos	43
4.6. Diagrama de de secuencia de descargar datos de los experimentos.	43
4.7. Estructura del sitio web	44
4.8. Diseño de la página principal de la aplicación.	44
4.9. Cabeceras interna y externa de la aplicación web.	45
4.10. Diseño de formulario del registro de la aplicación	45

4.11. Diseño de la página de inicio de sesión de la aplicación	46
4.12. Diseño de la página de información de un usuario de la aplicación	46
4.13. Diseño de la página de experimentos de la aplicación	47
4.14. Diseño de la página de información de experimentos de la aplicación	47
4.15. Diseño de la página de ranking de las puntuaciones de los usuarios de la aplicación	48
4.16. Diseño de la página de error de experimentos de la aplicación	48
4.17. Base de datos de la aplicación Eye-tracking.	49
5.1. Formulario de la aplicación web	54
5.2. Tarjeta de identificación del registro que sirve para obtener las dimensiones de la pantalla	55
5.3. Ejemplo de obtención de los parámetros de la cámara.	56
5.4. Círculo negro con aura azul principal elemento de los experimentos.	57
5.5. Fragmento de código asociado a la generación del código de experimento.	58
5.6. Fragmento de código asociado a la función del temporizador.	59
5.7. Alerta que salta al usuario en el caso de no haber pulsado ninguna vez el ratón en el experimento 1	60
5.8. Alerta que salta al usuario en el caso de haber pulsado la tecla escape [ESC] durante el experimento.	60
5.9. Capturas de las curvas de la circunferencia y de la elipse del experimento 2.	62
5.10. Capturas de las curvas de Lissajous generadas en el experimento 2.	62
5.11. Puntuaciones personales y globales de uno de los experimentos realizados	64
5.12. Fichero con los datos del experimento 2021-8-10_13_38.37.1_YFDR con formato json.	67
5.13. Imágenes capturadas en una de las pruebas realizadas del experimento 1	68
5.14. Imágenes que representan las pruebas que se han realizado respecto a la simultaneidad de la posición y el vídeo respecto al experimento de tipo 2.	68
5.15. Captura del mensaje recibido por parte de la aplicación.	68
6.1. Ventana principal del usuario de la aplicación.	69
6.2. Barra de navegación de un usuario que ya se ha identificado en la página.	70
6.3. Ventana principal del usuario de la aplicación en la que se describe quiénes somos y en qué consiste.	70
6.4. Ventana principal del usuario de la aplicación en la que se muestran las instituciones que han ayudado en el proyecto.	71
6.5. Sección del registro donde el usuario debe introducir la información de la cuenta de usuario.	71
6.6. Sección del registro donde el usuario debe introducir la información personal.	72
6.7. Sección del registro donde el usuario debe introducir las dimensiones de la pantalla.	73
6.8. Sección del registro donde el usuario debe introducir la posición de la webcam.	74

6.9. Mensaje de alerta informando al usuario que no puede realizar los experimentos con una tablet o móvil	74
6.10. Sección del registro donde el usuario debe leer el consentimiento y aceptación de los términos de la aplicación.	75
6.11. Ventana para mostrar la información al usuario	75
6.12. Inicio de sesión de los usuarios de la aplicación	76
6.13. Ventana principal de experimentos	76
6.14. Ventana en la que se muestra la información y las instrucciones del experimento 1.	77
6.15. Ventana en la que el usuario puede probar la cámara web del ordenador.	77
6.16. Ejemplo de la realización experimento 1	78
6.17. Representación del círculo siguiendo una trayectoria de una elipse sobre la pantalla durante el experimento 2.	79
6.18. Ventana para mostrar al usuario las puntuaciones globales y personales del experimento.	80
6.19. Ventana para mostrar al usuario que ha habido un error en la realización del experimento.	80
6.20. Ventana para mostrar al administrador el panel de control de buscador de experimentos.	81
6.21. Ventana para mostrar al administrador la lista del experimento 1 con fecha 2021-8-10.	82
6.22. Descarga del experimento 2021-8-10-10_13_38_37_1_YFDR, en el que aparece un fichero .json y un directorio con las imágenes capturadas	82
6.23. Descarga de los experimentos con fecha 2021-08-10	83
6.24. Fichero de configuración de apache	85
6.25. Tablas de la base de datos.	87
6.26. Sentencia para añadir al usuario privilegiado de la aplicación.	87

Capítulo 1

Introducción

1.1. Motivación

La mirada es el indicador observable desde el exterior de la atención visual humana, y muchos han intentado registrarla desde finales del siglo XVIII. Hoy en día, el estudio de esta actividad humana, por ejemplo durante la interacción con un ordenador, es de gran interés en múltiples disciplinas, ya que puede explicar aspectos importantes del comportamiento humano y de los procesos cerebrales que subyacen a él. Una de las técnicas que permiten analizar el comportamiento de la mirada se conoce como seguimiento ocular o eye-tracking, y consiste esencialmente en determinar la posición de la mirada de un individuo a lo largo del tiempo mientras realiza una cierta tarea. Este procedimiento tiene un gran potencial en una amplia variedad de disciplinas y áreas de estudio, desde el marketing digital hasta la investigación médica o la psicolingüística, porque la posición de la mirada y su evolución es un resultado (medible) de un conjunto de procesos cerebrales que pueden por lo tanto ser estudiados indirectamente de esta manera.

Existen varios tipos de soluciones tecnológicas para la realización de eye-tracking, pero en general requieren de la utilización de hardware específico, lo que involucra un alto coste (por ejemplo, Tobii Pro Glasses 3, Eye Tribe) o inexactitud en condiciones del mundo real [3]. Por este motivo el eye-tracking no es una tecnología muy generalizada, lo que dificulta su uso tanto en investigación como para otras posibles aplicaciones (interacción hombre-máquina para personas con discapacidad, por ejemplo).

En este contexto, se plantea la realización de eye-tracking sin el empleo de hardware específico, utilizando solamente la webcam que está presente en cualquier ordenador portátil en la actualidad. A partir de imágenes del rostro y los ojos de un usuario capturadas con este dispositivo, se pretende determinar la posición de la mirada del usuario sobre la pantalla. Una de las vías más prometedoras para realizar esta determinación es la utilización de sistemas de aprendizaje automático, en particular de Deep Learning. Esto requiere la construcción de un conjunto de datos de entrenamiento, y ese es precisamente el foco de este trabajo. Se plantea la construcción de un sistema web capaz de generar un conjunto de datos de entrenamiento para conseguir un sistema de eye-tracking sin hardware específico.

El uso de la webcam incorporada en los ordenadores portátiles tiene varias ventajas sobre otras alternativas:

- Uso generalizado: gran parte de la población tiene la posibilidad de acceder a un ordenador con facilidad.
- Alta tasa de actualizaciones de hardware: gran proporción de personas tiene hardware reciente, que permite el uso de métodos computacionalmente costosos como las redes neuronales.
- El uso masivo de las cámaras en los dispositivos ha llevado al gran desarrollo y despliegue de la tecnología de estas cámaras, que proporcionan imágenes de gran calidad.
- Permite el desarrollo de seguimiento ocular sin calibración de alta precisión ya que al tener la posición fija de la cámara sobre la pantalla reduce el número de parámetros desconocidos.

Con este proyecto se pretende realizar una aplicación que sea robusta y escalable. Se espera que esta aplicación tenga éxito y se recopilen gran cantidad de imágenes y vídeos para su posterior análisis.

1.2. Objetivos

El principal objetivo de este proyecto es crear una aplicación web que permita al usuario realizar experimentos relacionados con el seguimiento ocular o eye-tracking, y que un usuario administrador pueda recopilar los datos que se han ido capturando durante los mismos para la creación del conjunto de datos necesario para el entrenamiento de un sistema de aprendizaje automático capaz de determinar la posición de la mirada del usuario sobre la pantalla a partir de las imágenes capturadas por la webcam.

La aplicación pretende ser lanzada para su uso por parte de el mayor número posible de voluntarios, procedentes de cualquier parte del mundo. Para ello debe ser desplegada en un servidor de la Escuela Técnica Superior de Ingenieros de Telecomunicación. Así cualquier persona que quiera acceder sólo tendrá que conectarse a internet y añadir la URL específica.

El objetivo general del proyecto puede dividirse en los siguientes subobjetivos:

- Implementar una aplicación capaz de capturar datos, imágenes y vídeos de sujetos que utilicen la aplicación. Los sujetos deben poder registrarse y aceptar las condiciones de privacidad que se presentan, además de dar su consentimiento para poder analizar sus datos, imágenes y vídeos que se capturarán durante los experimentos.
- Desarrollar una base de datos relacional en el servidor de la escuela mencionado anteriormente, ya que se necesita guardar todos los datos que se capturan durante los experimentos para su posterior exportación por parte del administrador.

- Capturar imágenes y vídeos de buena calidad para poder realizar un buen análisis de los datos.
- Funcionar en diferentes navegadores como son Google Chrome y Mozilla Firefox.
- Generar tareas dentro de la aplicación en las que se consiga, por una parte, que los usuarios tengan un movimiento ocular que explore la mayor parte posible de la pantalla de su dispositivo, y por otro lado que pueda garantizarse que se conoce la posición de la mirada del usuario en los momentos en los que se capture su imagen (esto es necesario para que el conjunto de datos generado pueda servir para el entrenamiento del sistema de aprendizaje automático previsto). Para ello se crearán varios juegos o experimentos.
- Incentivar el uso del sistema por parte de voluntarios a través de una estrategia de gamificación, proporcionando al usuario una puntuación alcanzada después de la realización de cada experimento.
- Exportar los datos almacenados en la base de datos por parte del administrador de la aplicación para su análisis y la construcción del conjunto de datos deseado.

1.3. Fases y métodos

En una primera fase de la realización del proyecto fue necesaria una lectura de la documentación sobre la tecnología de seguimiento ocular o eye-tracking y las diferentes técnicas y tecnologías que se emplean para ello. Además, se consultó bibliografía especializada en la programación de páginas web, familiarizándose con el lenguaje recurriendo a pequeños ejemplos.

En segundo lugar, se realizó un diagrama de flujo con las diferentes partes que debía tener la aplicación desde el punto de vista del usuario, así como el diseño de su aspecto visual.

En un tercer punto se han desarrollado los principales casos de uso que debería tener la aplicación impuestos por los requisitos que se demandaban. A partir de este punto, ya se estuvo en disposición de poder empezar a desarrollar la aplicación.

La siguiente fase aborda el desarrollo de las primeras páginas de la aplicación web. Tras una primera descripción del software empleado, se llevó a cabo un análisis de los experimentos a implementar, los cuales involucran buena parte del código desarrollado. Además, durante esta fase también se implementó la base de datos relacional en un servidor local.

En la penúltima fase se realizaron diferentes pruebas en dicho servidor local. En esta fase se observaron diferentes errores que había en la aplicación y los aspectos que se debían modificar para su correcto funcionamiento.

Por último, se realizó un despliegue de la aplicación en el servidor de la escuela con las configuraciones necesarias en la máquina virtual que albergaba nuestra aplicación.

1.4. Estructura de la memoria

A continuación se define la distribución de capítulos escogida para este documento, especificando la información que se detallará en cada uno de ellos.

En este primer capítulo, se pretende dar una imagen global del proyecto desarrollado, exponiendo las fases en las que ha consistido y los objetivos que se han perseguido.

En el segundo capítulo, se describe detalladamente el estado del arte de las tecnologías utilizadas en el eye-tracking. Se empieza con una breve introducción sobre el sistema visual humano (HVS), después se realiza una descripción histórica de los sistemas más utilizados en el seguimiento ocular, y una última parte dedicada a las aplicaciones más significativas que se han desarrollado sin dispositivos externos, sólo a partir del vídeo natural.

A continuación, el tercer capítulo expone las tecnologías base implementadas en este proyecto. Se describen principalmente los diferentes lenguajes de programación utilizados para la realización de la página web y el gestor de base de datos utilizado.

El capítulo cuatro detalla el análisis de la aplicación. En primer lugar, se exponen los requisitos tanto funcionales como no funcionales, y después se presenta un análisis de la aplicación y de la base de datos.

En el quinto capítulo se presenta la implementación y los resultados que se han obtenido de la aplicación. Las dos primeras secciones describen las implementaciones de los casos de uso más significativos de la interfaz del usuario y de la interfaz del administrador. En la última parte de este capítulo se presentan algunas pruebas para comprobar la correcta operatividad de algunas funcionalidades de la aplicación y para la realización de las pruebas del despliegue en el servidor de la escuela.

El sexto capítulo describe y detalla los manuales de la aplicación, tanto del usuario como del administrador, y el manual de mantenimiento de la aplicación en el servidor remoto.

En el último capítulo expone las conclusiones obtenidas de este proyecto y las posibles líneas futuras en el análisis de los datos obtenidos en la aplicación.

Capítulo 2

Estado del arte

2.1. Introducción

El eye-tracking es una tecnología que pretende capturar información sobre el movimiento ocular de un usuario. Las aplicaciones de esta tecnología van desde el estudio de la atención y cognición humana en psicología hasta el diagnóstico temprano de enfermedades neurodegenerativas, o el marketing digital a través de análisis de la atención de los individuos a determinados estímulos.

Un ejemplo que permite comprender el potencial del análisis de los movimientos oculares es la lectura de un texto. Esta tarea requiere que diferentes sistemas cognitivos estén activos, desde la atención hasta la memoria. En personas sanas, se ha observado que palabras frecuentes y predecibles son procesadas con mayor facilidad y en menor tiempo. También se ha observado que la predictibilidad de las oraciones tiene un peso en el procesamiento de la información. Las oraciones que son más predecibles o proverbios son leídas con mayor rapidez que otras oraciones con baja predictibilidad. Mientras que esta predictibilidad facilita el proceso de lectura en personas sanas, no ocurre lo mismo con aquellos lectores que presentan alguna enfermedad cognitiva respecto de la memoria o de la atención. Algunos experimentos desarrollados establecen que la visión periférica no es la que facilita el proceso de lectura, si no que a través de la memoria semántica o la memoria de trabajo pueden empezar a procesar palabras antes de que sus ojos se muevan hacia ellas.

La aparición de enfermedades neurodegenerativas, como el Alzheimer, puede tener un impacto sobre la forma en la que las personas mueven los ojos en tareas como la lectura, y por lo tanto estas patologías son susceptibles de ser analizadas (y potencialmente detectadas, incluso de manera temprana) a través del eye-tracking. El Alzheimer se caracteriza por una pérdida de neuronas y sinapsis en la corteza cerebral, y los pacientes con deterioros leves suelen presentar deterioros en la memoria ejecutiva. La práctica de experimentos en que estén involucrados los movimientos oculares implican procesos donde participan diferentes partes del cerebro; si alguna de esas partes está dañada o deteriorada las respuestas motoras que conducen los movimientos oculares reflejan alteraciones y respuestas que no son normales [4].

A lo largo de este capítulo se elabora una descripción del sistema visual humano, desde su estructura y los tipos de visión del ojo hasta los movimientos oculares. En

segundo lugar se resumen los diferentes sistemas de eye-tracking más utilizados a lo largo de la historia. Por último, se hará una comparación con diferentes sistemas desarrollados sin hardware específico.

2.2. Sistema visual humano

En este apartado se describen algunos conceptos generales acerca del sistema visual humano (HVS) necesarios para entender el funcionamiento del seguimiento ocular o eye-tracking.

2.2.1. Estructura anatómica del ojo humano

El globo ocular tiene una forma ovalada y mide en torno a 24-25 mm de diámetro. Está formado por tres capas diferenciadas: la capa externa, intermedia e interna. Se puede ver su estructura y las partes de las que consta en la figura 2.1.[5]

- Capa externa. Esta capa está formada por la esclerótica y la córnea. La esclerótica es la parte blanca y lisa del ojo; una membrana opaca y rígida cuya función es la de proteger la parte interna del ojo. En cambio, la córnea presenta una forma convexa, sin vasos sanguíneos y es la responsable de dejar pasar la luz al interior del globo ocular.
- Capa intermedia. En esta capa se ubican la coroides, el cuerpo ciliar, el iris y el cristalino. La coroides es una membrana de color oscuro encargada de transportar alimentos, que contiene células pigmentarias que ayudan a absorber el exceso de luz y prevenir la visión borrosa. El iris es una membrana circular coloreada, debido al número y distribución de las células pigmentadas que contiene. Se encuentra entre la córnea y el cristalino. El cuerpo ciliar es una prolongación del iris y se une con la coroides. Produce un líquido en el interior del ojo, denominado humor acuoso y también contiene el músculo ciliar que se encarga de la acomodación del cristalino en las tareas de enfoque. Por último, el cristalino es la lente natural del ojo que tiene una forma biconvexa, transparente y elástica. Se localiza detrás del iris y está soportado por los músculos ciliares.
- Capa interna. En esta capa aparece la retina, que se extiende desde los músculos ciliares hasta la parte posterior del ojo, donde se encuentran los fotorreceptores (conos y bastones) y las estructuras nerviosas. Su función principal es la capacidad de detectar estímulos luminosos. En la parte central de la retina se encuentra la fóvea, especializada en la visión nítida [6]. La mayoría de los conos se concentran en esa parte de la retina para perfeccionar el enfoque y así poder leer, discriminar colores o sentir profundidad tridimensional.

Como se ha mencionado anteriormente, es en la retina es donde se encuentran las células fotorreceptoras, que son esencialmente los conos y los bastones. Los bastones son sensibles a la luz tenue y acromática (visión nocturna) y los conos a la luz cromática (visión diurna). La retina contiene aproximadamente 120 millones de bastones y 7 millones de conos.

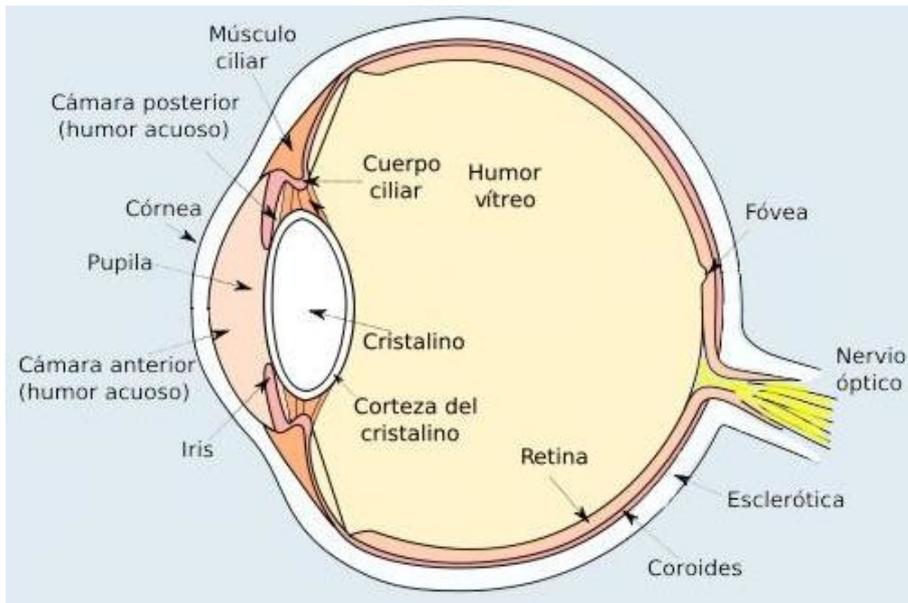


Figura 2.1: Anatomía del ojo humano

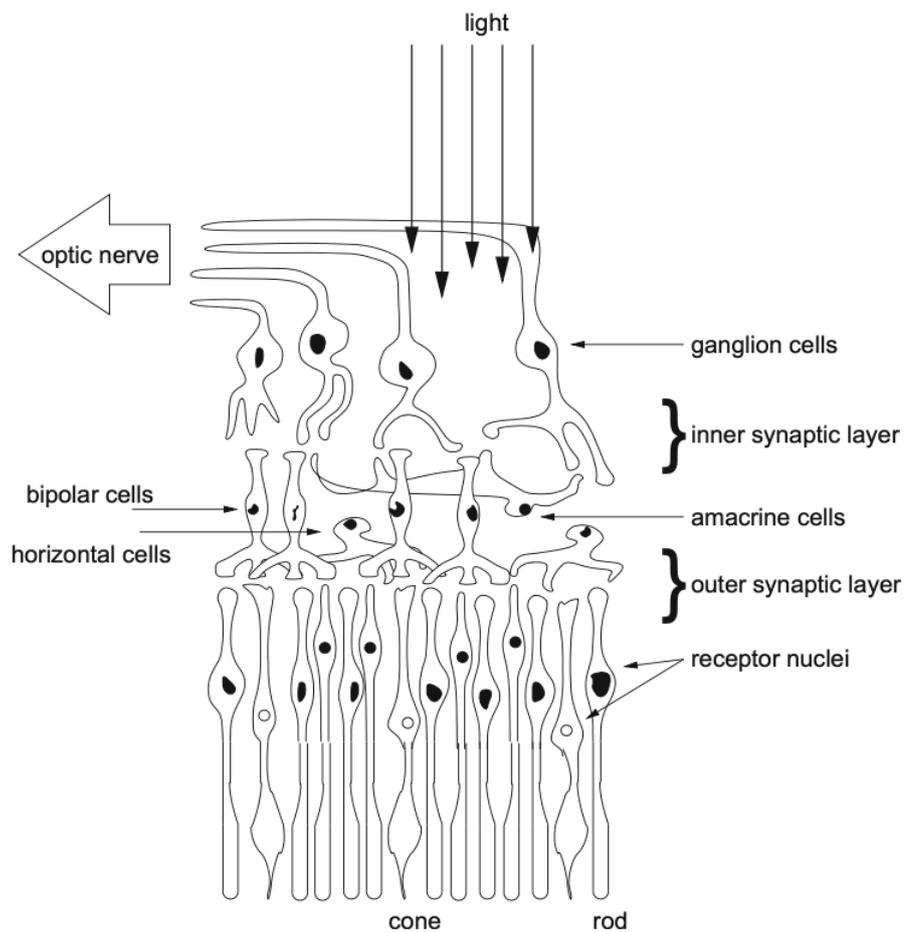


Figura 2.2: Diagrama esquemático de las conexiones neuronales entre las células ganglionar, bipolar, horizontales y amacrinas. [1]

La retina consta de tres capas de células con conexiones entre ellas, y en la capa inferior se ubican los bastones y los conos. Estas capas de conexión se denominan capas sinópticas y se muestra un esquema en la figura 2.2. La capa por donde entra la luz es la capa nuclear externa, en la que se encuentran los bastones y conos. La primera capa de conexión es la capa plexiforme externa que conecta los receptores con las células bipolares. La capa nuclear interna es la capa central de células, se localizan las células bipolares, amacrinas y horizontales. La tercera y última capa es la capa ganglionar, la cual alberga las células ganglionares y se conectan con la capa central a través de la capa de conexión plexiforme interna.

2.2.2. Psicofísica visual

El objetivo de esta sección es estudiar diferentes aspectos de la visión como son la visión espacial, la visión temporal y la percepción del color.

Visión espacial

Las dimensiones de la retina se miden en proporción de la imagen proyectada en unidades de grado de ángulo visual. La fovea centralis (o foveola) mide $400\ \mu\text{m}$ de diámetro y contiene 25.000 conos. La fovea mide $1500\ \mu\text{m}$ de diámetro y tiene una capacidad de 100.000 conos. La mácula, parte central de la retina, mide $5.000\ \mu\text{m}$ y almacena 650.000 conos. El ángulo visual de un grado corresponde a $300\ \mu\text{m}$, por tanto, la foveola que mide $400\ \mu\text{m}$ corresponde a 1.3° , la fovea 5° y la mácula 16.7° .

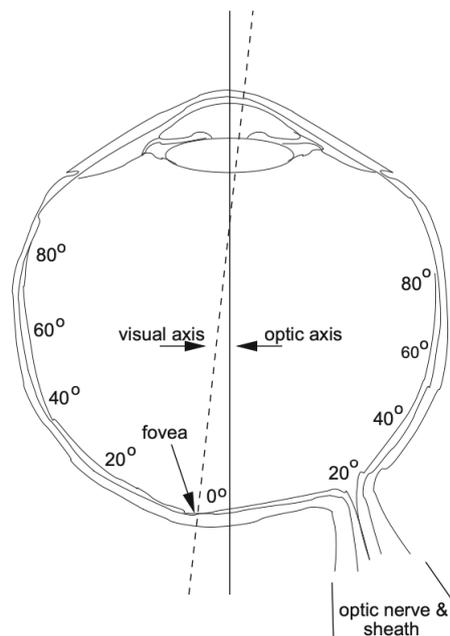


Figura 2.3: Distribución de los conos y bastones en la retina en función del ángulo visual. [1]

En la figura 2.3 se muestra una distribución de los conos y bastones en función del ángulo visual. La fovea contiene $147.000\ \text{conos}/\text{mm}^2$ y un número ligeramente menor de bastones. A 10° el número de conos cae a menos de $20.000\ \text{conos}/\text{mm}^2$ y a 30° el número de bastones es $100.000\ \text{bastones}/\text{mm}^2$. El campo visual corresponde

a un área de 23.400 grados cuadrados de una elipse con el eje horizontal mayor de 180° y el menor de 130° . La zona de mayor agudeza visual se expande hasta los 4° o 5° y desciende bruscamente a medida que se hace más grande, de forma que a solo 5° la agudeza cae hasta 50%. A medida que aumentamos el ángulo, los conos van aumentando su tamaño y son los que más información aportan a los centros neuronales más profundos.

Visión temporal

La respuesta visual humana al movimiento se caracteriza por la persistencia visual y el conocido como fenómeno phi. La persistencia describe la frecuencia de muestreo temporal del HVS y el fenómeno phi un umbral a partir del cual el HVS detecta movimiento.

Estos fenómenos se pueden apreciar en el cine o la televisión para obtener percepción de movimiento a partir de imágenes fijas que se muestran sucesivamente. El sistema visual es incapaz de obtener información de imágenes que cambian muy rápidamente, y por ello aparece un parpadeo a unos 50-60Hz (se suele llamar frecuencia crítica de parpadeo). Como podemos ver en la figura 2.4, cuando la frecuencia temporal presenta un valor bajo, la sensibilidad al contraste entre las imágenes es muy grande. Sin embargo, a medida que aumentamos la frecuencia la sensibilidad disminuye, hasta el valor de frecuencia crítica (en torno a 50Hz), impidiendo apreciar el contraste. Esto sucede en el cine, en el que los fotogramas se muestran a una frecuencia de 24Hz; pero con un obturador de tres hojas hace que aumente la frecuencia a 72Hz. En la televisión sucede algo similar: la frecuencia es de 30Hz pero utiliza un esquema entrelazado y logra tener una frecuencia de 60Hz. El segundo hecho, el fenómeno phi, explica el movimiento estroboscópico. Este es un efecto que se produce al iluminar un objeto que se mueve de forma rápida y periódica con destellos de luz. Para que se produzca el movimiento estroboscópico, la frecuencia entre destellos debe ser de 62Hz.

Por otra parte, la respuesta temporal del HVS no es homogénea en todo el campo visual. El movimiento se percibe de manera uniforme en todo el campo visual, pero la región foveal es más receptiva a movimientos más lentos. La periferia de la fovea es más sensible a objetivos en movimiento que los estacionarios. En esencia, el sistema visual está diseñado para alertar de la entrada de objetos en el campo visual a través de la periferia.

Visión del color

La visión del color se consigue gracias a los tres tipos de fotorreceptores de cono retiniano. Las curvas de sensibilidad espectrales de los tres tipos de receptores presentan un pico en 450, 520 y 555 nm de longitudes de onda, y corresponden a cada uno de los fotorreceptores azul, verde y rojo respectivamente. La mayoría de los conos están concentrados en la zona central de la fovea, por lo que la distribución de los conos se empobrece cuando nos alejamos de la parte central. Esto implica que la visión periférica sea también menos sensible al color.

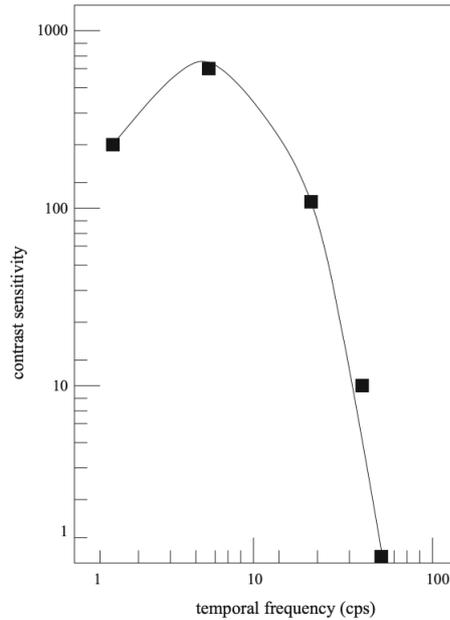


Figura 2.4: Gráfica en la que se muestra la sensibilidad al contraste en función de la frecuencia temporal. [1]

2.2.3. Modelos de movimientos oculares

Los ojos se mueven dentro de seis grados de libertad: tres traslaciones y tres rotaciones. Son también seis los músculos responsables de los movimientos oculares: para movimientos laterales encontramos los rectos medial y lateral; para movimientos arriba y abajo, los rectos superior e inferior; y para movimientos de giro los oblicuos superior e inferior. El sistema neuronal responsable de los movimientos oculares es la planta oculomotora. Los diferentes músculos oculares se ilustran en la figura 2.5.

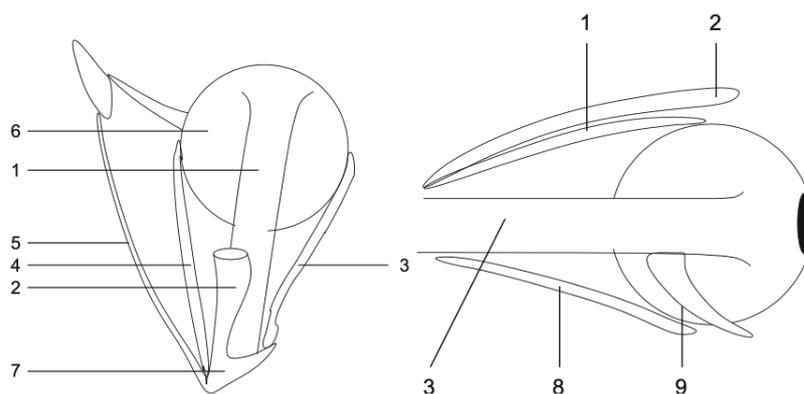


Figura 2.5: Estructura muscular del ojo. 1-Recto superior; 2- Elevador del párpado superior; 3- Recto lateral; 4- Recto medial; 5- Oblicuo superior; 6- Tendón reflejado del oblicuo superior. 7- Anillo de Zinn; 8- Recto inferior; 9- Oblicuo inferior. [1]

Habitualmente se distinguen varios tipos de movimientos (o ausencias de movimientos) oculares: las sacadas, las fijaciones y los movimientos de persecución. A continuación nos ocupamos de ellos.

Sacadas

Los movimientos sacádicos son movimientos rápidos de los ojos cuyo objetivo es posicionar la fovea en una nueva ubicación del campo visual. Estos movimientos pueden ser voluntarios pero también reflexivos, y su duración varía entre 10 y 100 ms. También se caracterizan por ser movimientos estereotipados (se repiten ciertos patrones de movimiento) y balísticos (el destino de los movimientos está preprogramado).

En el control de los movimientos sacádicos intervienen múltiples áreas y circuitos cerebrales, por lo que su análisis es especialmente importante en muchos contextos. Esto implica que los sistemas de eye-tracking deben tener una resolución temporal suficiente para poder capturar estos eventos.

Movimientos de persecución

Los movimientos de persecución ocurren cuando se realiza un seguimiento visual a un objeto que se está moviendo. Se trata de movimientos relativamente lentos, al menos en comparación con los movimientos sacádicos. Dependiendo de la velocidad del objeto perseguido, el movimiento de persecución trata de igualar dicha velocidad.

Fijaciones

Se llaman fijaciones a los periodos de tiempo en los que la retina se estabiliza sobre un objeto. Se podría pensar que las fijaciones se pudiesen modelar como un movimiento de persecución caracterizado por tener una velocidad cero, pero esto no es realmente cierto, ya que las fijaciones se caracterizan por tres movimientos oculares en de pequeña envergadura: el temblor, la deriva y las microsacadas.

Las microsacadas se producen debido a la sensibilidad del movimiento del tejido unicelular del sistema visual; son señales que se suceden aleatoriamente y con diferente amplitud (tics oculares). Son los responsables del 80 % de la percepción visual.

2.3. Sistemas de eye-tracking

El origen del eye-tracking se remonta al siglo XVII, cuando Louise Emile Javal, oftalmólogo francés, se dio cuenta de que los ojos de los lectores no se movían de manera fluida, sino que realizaban movimientos rápidos, que ahora denominamos sacadas, y pausas, que denominamos fijaciones. En esta época no se desarrolló ningún dispositivo para rastrear el movimiento de los ojos, sólo se basaron en la observación.

Muchos años después, en 1908, fue cuando Edmund Huey construyó un dispositivo capaz de rastrear el movimiento ocular durante la lectura. Este dispositivo contaba con una lente de contacto con la pupila del lector y además iba unida a un puntero que cambiaba de posición a medida que se iba moviendo el ojo. Los hallazgos que realizó se encuentran en un libro que publicó llamado *The Psychology and Pedagogy of Reading*.

Charles H. Judd, psicólogo educativo estadounidense, fue el primero en utilizar una cámara para grabar el movimiento ocular en una película y permitir su posterior estudio. Guy Thomas Buswell y Judd analizaron los movimientos oculares y obtuvieron grandes avances en el estudio de la lectura. Gracias a sus trabajos se describió que hay una diferencia en el comportamiento ocular entre la lectura oral y en silencio.



Figura 2.6: Ejemplo de medición de electrooculografía (EOG). [1]

Unas décadas más tarde se empezaron a desarrollar nuevos métodos, como la electrooculografía (EOG), el método de lentes de contacto combinadas con una bobina, fotooculografía (POG) o videooculografía (VOG) y la reflexión pupilar/corneal combinada basada en vídeo.

2.3.1. Electrooculografía (EOG)

En 1939, Jung midió los movimientos oculares verticales y horizontales a través de electrodos situados en la piel alrededor de los ojos. Este método recibe el nombre de electrooculografía (EOG), fue el método que más se utilizaba hace aproximadamente 40 años para el estudio de los movimientos oculares y se sigue utilizando en algunos casos en la actualidad. Se basa en la medición de la diferencia de potencial eléctrico de las zonas de la piel donde se encuentran los electrodos. En la figura 2.6 se puede ver una imagen de un dispositivo de este tipo empleado sobre un voluntario. Los potenciales eléctricos que se aplicaban a los electrodos eran valores muy bajos del rango de 15 a $200\mu\text{V}$, y con este sistema se pueden conseguir sensibilidades del orden de $20\mu\text{V}/\text{grado}$ de movimiento ocular.

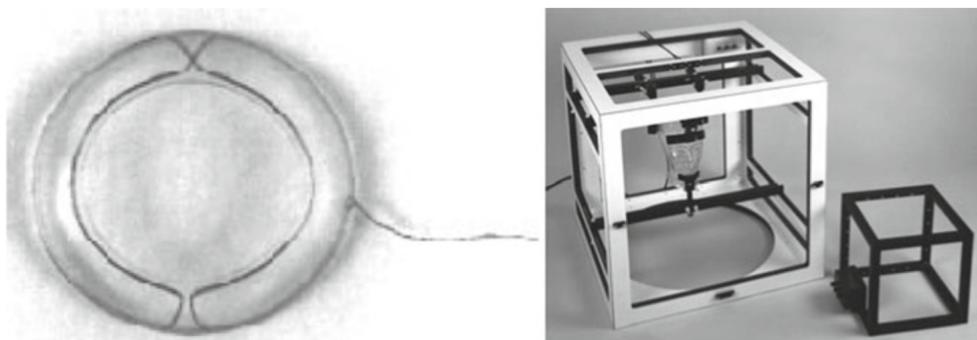


Figura 2.7: Ejemplo de bobina incrustada en lente de contacto y el marco de campos electromagnéticos para la medición del movimiento ocular. [1]

2.3.2. Lentes de contacto combinadas con una bobina

Otro de los métodos de medición ocular más precisos que se desarrollaron a lo largo del siglo XX consistía en colocar un objeto mecánico u óptico sobre una lente de contacto que se colocaba sobre el ojo. En un primer momento se utilizaba un anillo de yeso unido directamente a la córnea y a mecanismos mecánicos conectados a sus bolígrafos de grabación. Esta técnica fue avanzando hasta llegar a una lente de contacto suficientemente grande para cubrir toda la córnea y la esclerótica. Ha habido diferentes dispositivos mecánicos que se han unido a la lente, como los fósforos reflectantes, los diagramas de líneas y las bobinas de alambre. Este método principalmente empleaba una bobina que se movía sobre un campo electromagnético, pudiendo por lo tanto medirse dicho movimiento. En la figura 2.7 se puede ver la bobina insertada en la lente de contacto y el marco de campos electromagnéticos para la medición del movimiento ocular. La forma de inserción de la lente de contacto se puede ver en la figura 2.8. Este es un método que requiere cuidado y práctica a la hora de insertar la lente, ya que puede llegar a ser muy incómodo. También es capaz de medir la posición del ojo en relación con la cabeza.



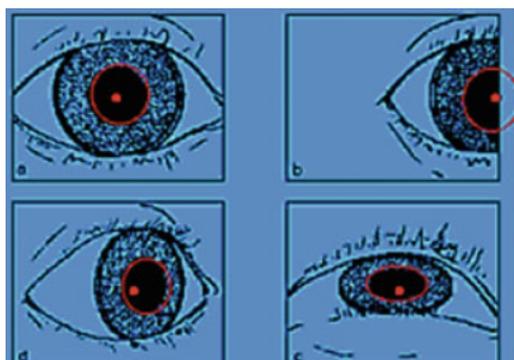
Figura 2.8: Ejemplo de inserción de una lente de contacto para medición de movimiento ocular. [1]

2.3.3. Foto-oculografía y Vídeo-oculografía

En lugar de utilizar sistemas invasivos para detectar los movimientos de los ojos como hemos visto con las técnicas anteriores, también es posible determinar la posición de la mirada, así como los movimientos oculares, simplemente a partir de imágenes o vídeos de los ojos. Esto puede hacerse tratando de hacer un seguimiento de diferentes puntos específicos de los ojos, como pueden ser la pupila, el limbo (parte del ojo que presenta el límite entre el iris y la esclerótica), o analizando los reflejos corneales a una fuente de luz. En general, la luz infrarroja es la más adecuada para este tipo de sistemas, porque por una parte permite iluminar los ojos sin que el sujeto reaccione a ello o se vea incomodado, y por otra parte la detección de los puntos de interés del ojo es más sencilla en imágenes capturadas en este rango de frecuencias. El problema, claro, es que estos sistemas requieren por lo tanto de un hardware específico para la emisión y la grabación de este tipo de luz.

En la figura 2.9(a) se puede ver unas imágenes de un ojo en el que se ve las diferentes posiciones del limbo; además en la figura 2.9(b) y figura 2.9(c) también se presentan dos aparatos de luz infrarroja para captar el seguimiento del limbo.

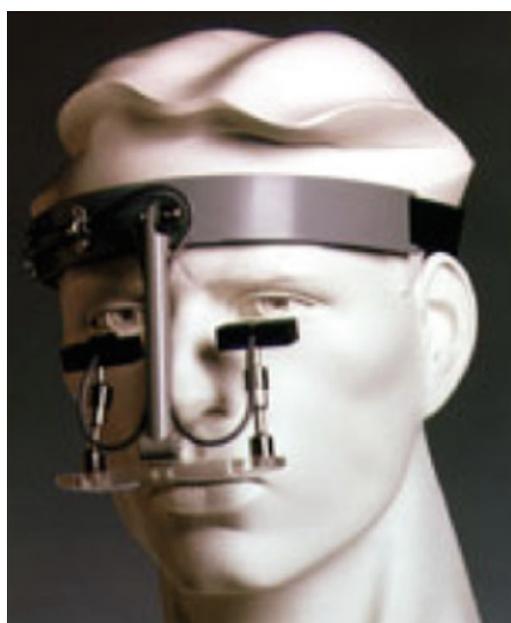
Las técnicas anteriores son adecuadas para la medición de movimientos oculares, pero no proporcionan mediciones del punto de visión. Para obtener estas mediciones se debe fijar la posición del ojo respecto de la cabeza y al punto de mirada, o deben medirse múltiples características para diferenciar el movimiento de rotación



(a) Ejemplo de imágenes generadas de la pupila.



(b) Ejemplo de un aparato de infrarrojos.



(c) Ejemplo de un aparato de infrarrojos sobre un sujeto

Figura 2.9: Ejemplos de mediciones del movimiento ocular por reflejo de la pupila, el limbo y la córnea. [1]

del ojo y el movimiento de la cabeza. Dos de estas características son el reflejo corneal, adquirido con una fuente de luz infrarroja y el centro de la pupila. Para la realización de estas mediciones se emplean rastreadores de vídeo que utilizan cámaras y hardware para el procesamiento de imágenes. El aparato puede estar fijado en una mesa como aparece en la figura 2.10 o puede ir montado en la cabeza como en la figura 2.11, pero el funcionamiento es idéntico.

El reflejo corneal de la fuente de luz se mide en relación a la ubicación de la pupila. Estos reflejos se denominan reflejos de Purkinje o imágenes de Purkinje. Debido a la estructura ocular se forman cuatro reflejos o imágenes de Purkinje, y los rastreadores oculares suelen adquirir la primera de esas imágenes. Se pueden observar unas imágenes de los primeros reflejos de Purkinje en la figura 2.12, en las



Figura 2.10: Ejemplo de un antiguo rastreador de ojos basado en vídeo montado sobre una mesa. [1]



Figura 2.11: Ejemplo de un rastreador ocular basado en vídeo montado en la cabeza [1]

que el círculo blanco corresponde con el reflejo y el círculo negro representa la pupila. Hay otros tipos de rastreadores, llamados rastreadores oculares de generación V, que también miden la cuarta imagen de Purkinje. Al medir los dos reflejos separan los movimientos oculares de rotación y traslación.

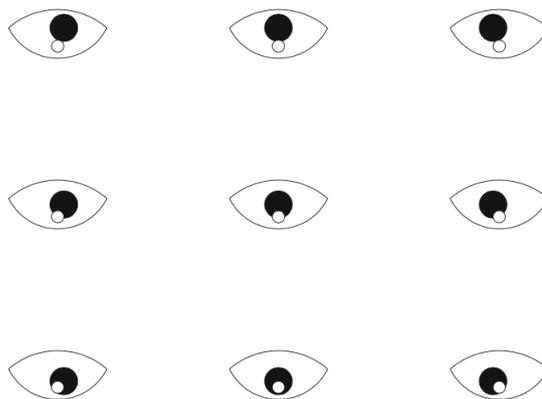


Figura 2.12: Ejemplo de un antiguo rastreador de ojos basado en vídeo montado sobre una mesa. [1]

Muchos de los sistemas de eye-tracking más habituales en la actualidad se basan en el concepto de video-oculografía con luz infrarroja, tanto en su versión remota (sobre una mesa) como en su versión montada en la cabeza. Un buen ejemplo son las gafas Tobbi pro, que se pueden ver en la figura 2.13. Estas gafas presentan 16

iluminadores y 4 cámaras de ojos integradas en lentes resistentes [2].



Figura 2.13: Gafas Tobbi pro. [2]

2.4. Eye-tracking a partir de video natural

Como hemos visto anteriormente, la mayor parte de los sistemas de eye-tracking usados en la actualidad se basan en el concepto de video-oculografía con luz infrarroja, de forma que requieren el uso de hardware específico tanto para emitir como para recibir luz infrarroja. En los últimos años, sin embargo, se han producido varios intentos de realizar eye-tracking utilizando vídeo natural a partir de dispositivos no específicos como pueden ser teléfonos móviles, tablets u ordenadores.

2.4.1. GazeCapture

GazeCapture es un sistema de seguimiento ocular utilizando dispositivos móviles. A partir de datos adquiridos de unos 1500 sujetos de diferentes orígenes se entrenó una red neuronal convolucional (CNN) con un modelo basado en la apariencia, no dependiente de características diseñadas a mano como la posición de la cabeza o la ubicación del centro del ojo.

El sistema se diseñó con el objetivo de que fuese escalable, confiable y que soportara una gran variabilidad [3]. Para ello se tomaron las siguientes decisiones en la construcción de la base de datos empleada para el entrenamiento del sistema:

- Escalable: Los autores crearon una aplicación móvil, llamada GazeCapture, capaz de registrar y cargar datos de seguimiento de la mirada de los usuarios, y usaron *Amazon Mechanical Turk (AMT)* para darse a conocer y obtener sujetos y crear datos de entrenamiento.
- Confiable: El funcionamiento de la aplicación consistía en mostrar a los usuarios unos puntos en la pantalla con ubicaciones aleatorias. Los usuarios debían seguir esos puntos con la mirada. La figura 2.14 muestra la estrategia seguida para asegurar que el usuario estaba mirando a la posición del punto. Tras la aparición de dicho círculo rojo en un punto de la pantalla, la grabación de la imagen comenzaba 0.5 segundos después. Tras un segundo y medio, aparecía una R o una L dentro del círculo, durante un breve periodo de tiempo de 0.05 segundos. Esta letra indicaba al usuario que debía tocar la parte izquierda o derecha de la pantalla, verificando así que el usuario estaba prestando atención al experimento y mirando al círculo. Además, se integró un detector facial en

tiempo real, para asegurar que las imágenes que entraban a formar parte del conjunto de entrenamiento incluían el rostro y los ojos del usuario.

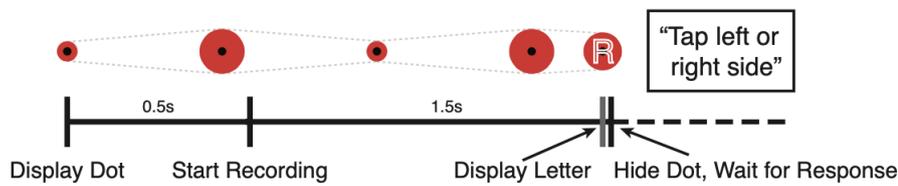


Figura 2.14: Línea de tiempo de visualización de un punto. [3]

- Variable: Para conseguir una buena variabilidad en el conjunto de datos, se obligó también a los usuarios a cambiar la orientación del dispositivo móvil cada 60 puntos (utilizando sensores integrados en el dispositivo).

En total se adquirieron datos de 1440 sujetos, lo que dio lugar a un conjunto de 2.445.504 imágenes con sus respectivas ubicaciones de la fijación de la mirada sobre la pantalla.

2.4.2. MPIIGaze

MPIIGaze es un conjunto de datos para la estimación de la posición de la mirada basada en la apariencia. Se obtuvo a partir de 15 sujetos durante varios meses, hasta conseguir un total de 213.659 imágenes. Este sistema se diseñó con dos objetivos principalmente. El primero era registrar imágenes de los usuarios fuera de las condiciones comunes, es decir, fuera del laboratorio de trabajo. El segundo pretendía seguir obteniendo datos de los mismos sujetos durante varios meses, recogiendo datos de diferentes lugares y tiempos de grabación. Para llevarlo a cabo optaron por ordenadores, en vez de dispositivos móviles o tablets.

Los usuarios tenían que realizar los experimentos cada 10 minutos. Durante la prueba los usuarios debían mirar una secuencia aleatoria de 20 posiciones visualizada en círculo gris con un punto blanco en su interior. El punto blanco del interior era para que el usuario fijase la mirada en ese punto. [7]

Estos sistemas son similares a los implementados en nuestro proyecto. Nuestro proyecto pretende conseguir un conjunto de datos para el entrenamiento de un sistema de eye-tracking con ordenadores y sus web-cams. En lugar de una aplicación nativa, como en el caso de GazeCapture, en nuestro caso se trata de una aplicación web.

Capítulo 3

Tecnologías base

3.1. Introducción

En este capítulo se analizan las tecnologías más importantes involucradas en el desarrollo de este proyecto. Se empieza explicando los principales lenguajes que se han utilizado: HTML5 dotó de estructura a las diferentes páginas; CSS generó el estilo; Javascript y JQuery fueron utilizados para realizar todo tipo de eventos y la programación de los diferentes elementos de la aplicación; y Php fue implementada para establecer la conexión con la base de datos. Tras explicar los principales lenguajes utilizados, se expuso una discusión sobre idoneidad de la utilización de MySQL para el desarrollo de la base de datos. Por último, se ha analizado la herramienta utilizada para el desarrollo del modelado UML.

3.2. HTML5

HTML es un lenguaje de programación empleado para determinar la estructura de la página web. Este lenguaje ha ido actualizándose con los años creando nuevas versiones, progresivamente más sencillas y con más funcionalidades para el usuario. En este proyecto se ha utilizado entre otras cosas, porque ya la había utilizado en otros proyectos.

La principal característica por la que se utilizó HTML5 es porque dispone de un elemento llamado `<canvas>`, el cual permite definir un área de la página web para dibujar líneas, formas, importar imágenes u otros elementos. Es algo que facilitó el poder crear los experimentos o juegos que se realizan en la aplicación.

En un principio también se pensó la alternativa de utilizar Unity para el desarrollo del videojuego o experimento, pero a la larga iba a dar más problemas a la hora de incorporarlo a la página web. Por tanto, se decidió utilizar HTML para la realización de los experimentos además de ser el lenguaje que da la estructura a toda la página web.

3.3. CSS

CSS u hojas de estilo en cascada mejoraron la presentación de la página web. Es un lenguaje para ayudar al usuario, ya que hace que la página web sea mas atractiva.

Principalmente, se encarga de diseñar y dar estilo a la página web alterando el color, la fuente, el tamaño, el espaciado y muchas más características de los diferentes elementos que constituyen la estructura de la página web [8].

Este lenguaje ha sido útil para realizar el elemento del círculo de los experimentos, ya que nos ha proporcionado la animación del círculo (haciéndose más grande y más pequeño) para hacer la página más amigable para el usuario.

3.4. Javascript

Javascript es un lenguaje de secuencia de comandos diseñado para dotar a la página web de cierto grado de interactividad. Puede, entre otras cosas, proporcionar mecanismos de detección y tratamiento de eventos, como un click con un dispositivo apuntador, generalmente un ratón. Es un lenguaje interpretado por el cliente, es decir, procesado por el navegador web y no por el servidor web. Este elemento fue importante porque, entre otras funciones, pudimos verificar formularios web antes de enviarlos al servidor, asegurándonos de que todos los campos obligatorios del formulario hayan sido rellenados. [9]

Este lenguaje fue empleado con frecuencia en la página web. Sobre todo en la realización de los experimentos, ya que la posición del ratón se recogía gracias a los eventos que javascript proporcionaba. Se escogió esta tecnología principalmente porque los eventos que se utilizan son parecidos a los utilizados en el lenguaje Java y pude aprender más rápido como se implementaban.

3.5. PHP

PHP es un lenguaje de programación interpretado, y que se procesa en el lado del servidor, a diferencia de los lenguajes que hemos visto en los apartados anteriores. Es un lenguaje que se ha utilizado para la realización de las peticiones y las respuestas de la base de datos. Es el lenguaje elegido ya que es un lenguaje que se ha utilizado en una de las asignaturas de la carrera y daba muy buenos resultados a la hora de la realizar esas peticiones. Además, se ha podido utilizar para crear variables de sesión dentro de la página web una vez que el usuario iniciaba sesión. Esto ha hecho que fuese mucho más fácil la realización del código.

3.6. Apache HTTP Server

En la realización de este proyecto se ha tenido que usar un servidor web y por ello surge el problema de cual es el más adecuado para nuestra página web. Se barajaron tres opciones: Nginx, Tomcat y Apache HTTP Server.

Nginx es un tipo de servidor web bastante actual y es popular por la optimización de las conexiones de red, es decir, puede gestionar gran cantidad de clientes al mismo tiempo. Dado que la aplicación web no va a tener tantas conexiones al mismo tiempo no sería necesario tener estas prestaciones.

Tomcat se pensó como posible servidor web ya que también es bastante popular,

pero se descartó ya que se creó específicamente para aplicaciones Java y aunque puede servir páginas web no es tan eficiente como el servidor de Apache.

Por tanto, dado que las anteriores opciones no son las más adecuadas para servir nuestra página web, se eligió Apache como servidor web, entre otras cosas porque tiene una fácil configuración y para sitios web pequeños o medianos es la mejor opción. Además, Apache HTTP Server es un servidor web de código abierto, popularizado entre los programadores por sus contribuciones para la mejora de funcionalidades y recursos.

3.7. MySQL

Uno de los principales problemas que surgen al realizar una página web es el sistema de gestión de base de datos que se va a utilizar. Por tanto, se tuvo que analizar las características que debía tener la aplicación y escoger el sistema más apropiado.

Se pensó en tres sistemas de gestión de bases de datos: Oracle (ya que es el más utilizado según el ranking de sistemas de gestión de bases de datos [10]), PostgreSQL y MySQL.

Oracle es un sistema de gestión de los más completos actualmente, destaca por su estabilidad, escalabilidad y por su soporte multiplataforma. Una de las razones por las que se descartó fue que es un sistema que consume muchos recursos ya que presenta muchas funcionalidades, además de ser un sistema que se utiliza para implementaciones a gran escala.[11]

PostgreSQL es un sistema que tiene buena reputación dentro de los sistemas de gestión de bases de datos y destaca por su gran fiabilidad e integridad de datos, pero también se suele utilizar para bases de datos grandes, y en nuestro proyecto no vamos a tener una base de datos de grandes dimensiones.[11]

Por último, MySQL es uno de los sistemas más sencillos para la iniciación en los sistemas de gestión de base de datos. Destaca por su velocidad y bajo consumo de recursos. Uno de los principales inconvenientes de usar MySQL es que tiene menos funciones que las bases de datos mencionadas anteriormente.

Por tanto, debido a la simplicidad y los pocos recursos que consume se ha optado por elegir MySQL como sistema de gestión de bases de datos. Además, es un sistema que se ha estudiado durante la carrera y se puede encontrar documentación de forma sencilla y fiable. Aunque se podría pensar que es un sistema que destaca por su sencillez y parece no ser estable, se utiliza para proyectos y empresas muy conocidas como Facebook, Youtube, GitHub, Tesla o incluso NASA.

MySQL es un gestor de base de datos que aparecen organizado en tablas para poder relacionar los elementos o parámetros de forma muy sencilla, logrando establecer relaciones entre registros, como nos sucede en nuestra base de datos, que se analizará en los siguientes capítulos.

3.8. Astah

Astah es un programa para realizar diagramas UML, es fácil de aprender y muy intuitivo. También se ha utilizado este programa para otros trabajos realizados durante la carrera en cuanto a ingeniería de software y daba muy buenos resultados. En el siguiente capítulo podemos ver el uso de este programa con el desarrollo de diagramas.

Los logos de las tecnologías se pueden observar en la figura 3.1.



Figura 3.1: Resumen gráfico de las tecnologías utilizadas.

Capítulo 4

Análisis y diseño de la aplicación

4.1. Introducción

En este proyecto se pretende diseñar e implementar una aplicación web para la obtención de un conjunto de datos de entrenamiento de un sistema de eye-tracking sin utilizar ningún hardware específico. Hasta hace unos pocos años el eye-tracking se realizaba siempre con ciertos sistemas específicos, como un sistema de luz infrarroja o lentes de contacto.

En este capítulo se elabora el diseño y el análisis de la aplicación, además de una recopilación de los requisitos que necesita. Además, se hará un diseño de la base de datos.

4.2. Análisis de la aplicación

En este apartado se va a realizar un análisis de la aplicación desarrollada. Primero, se empezará analizando los requisitos funcionales y no funcionales y a continuación el desarrollo de los casos de uso.

4.2.1. Especificación de requisitos

Los requisitos se pueden dividir en dos grupos: funcionales y no funcionales. Los funcionales engloban las características requeridas del sistema que expresan una capacidad de acción del mismo, es decir, una funcionalidad. En cambio, los requisitos no funcionales suponen restricciones que afectan a los servicios o funciones del sistema, tales como accesibilidad, internalización, aspectos éticos y morales, etc.

Requisitos funcionales

En primer lugar se realizó un diseño de la estructura básica de la aplicación con las funcionalidades que debía de tener la página web en cuestión. Por ello, se redactaron los siguientes requisitos funcionales.

- El sistema deberá permitir a un usuario registrarse en la aplicación.
- El sistema deberá permitir iniciar sesión a los usuarios registrados previamente, a través de un código de 4 letras generado aleatoriamente y una contraseña.

- El sistema deberá enviar un correo electrónico al usuario una vez acabado su registro con el código de inicio de sesión generado.
- Durante el registro, el sistema dejará al usuario escoger las dimensiones de la pantalla y la posición en la que se encuentra la webcam en nuestro ordenador, a no ser de que se se esté accediendo desde un móvil o tablet. En ese caso, el sistema permitirá registrar al usuario sin ambos parámetros y antes de la realización del experimento el sistema deberá dejar que se introduzcan.
- El sistema permitirá a los usuarios registrados escoger alguno de los dos experimentos para su realización.
- El sistema deberá mostrar una página principal donde se expone el propósito de la página web, quiénes somos y las empresas o asociaciones participantes en el proyecto.
- El sistema deberá mostrar a los usuarios una explicación de los experimentos antes de su realización.
- El sistema deberá mostrar a los usuarios su información personal y les permitirá modificar aquellos parámetros que consideren oportunos.
- El sistema deberá dejar a los usuarios borrar su cuenta si así lo desean.
- El sistema deberá mostrar a los usuarios las puntuaciones obtenidas tras las realización de cada experimento. Deberá mostrar dos clasificaciones, una para las puntuaciones obtenidas globalmente de ese tipo de experimento y otra con las puntuaciones personales.
- El sistema podrá diferenciar entre usuarios y administrador a través de un campo de privilegio que viene prefijado cuando realizas el registro.
- El sistema deberá permitir al administrador iniciar sesión en la aplicación.
- El sistema deberá mostrar al administrador una clasificación de los experimentos realizados. Además, el sistema deberá permitir al administrador filtrar los experimentos por las fechas que desee.
- El sistema permitirá al administrador descargar los ficheros correspondientes a los experimentos realizados.
- El sistema deberá permitir a los usuarios contactar con los propietarios de la aplicación vía correo electrónico en el caso de que tengan alguna duda o sugerencia.
- El sistema deberá mostrar a los usuarios las políticas de privacidad que el usuario debe acepta siempre que quiera registrarse y participar en el proyecto.
- El sistema deberá mostrar una página de error en el caso de que durante el experimento no se cumplan con las especificaciones expuestas.
- El sistema deberá dejar al usuario registrado probar la cámara para que se almacenen los datos de la misma.

Requisitos no funcionales

Los requisitos no funcionales que se expusieron fueron los siguientes.

- El sistema mostrará la página en español.
- El sistema no deberá permitir que se realicen experimentos desde navegadores que se estén ejecutando en móviles o tablets.
- El sistema no almacenará en la base de datos ningún dato personal, ya sea nombre, apellidos, correo electrónico, etc. El correo electrónico sólo se usará para mandar al usuario el código de usuario pero no se almacenará en la base de datos.
- El sistema deberá visualizarse y funcionar correctamente en las últimas versiones de los navegadores Mozilla Firefox y Google Chrome.
- El sistema no almacenará la contraseña de los usuarios en texto plano sino que se almacenará encriptada.
- El sistema permitirá interactuar con la página de una forma fiable, sencilla y rápida.

Una vez vistos los requisitos funcionales y no funcionales pasamos a analizar las funcionalidades de la aplicación.

Para ello se van utilizar los diagramas de casos de uso, en los que se muestra la interacción entre los actores y los casos de uso del sistema. Los casos de uso representan los requisitos más potenciales del sistema, además de ser la técnica que permite capturar información acerca de cómo un sistema trabaja o se desea que trabaje. En los diagramas de casos de uso se representa el sistema mediante una caja rectangular con el nombre en su interior, y los actores fuera. Cada actor está unido a los casos de uso, en los que participa, mediante una línea.

Los actores son entidades externas al sistema que guarda una relación con éste y demandan una funcionalidad. Comúnmente son operadores humanos aunque también pueden ser sistemas externos, máquinas, organizaciones, etc.

Los casos de uso representan una unidad funcional que pueden ejecutar un determinado actor o actores, por tanto, existe una comunicación entre ellos y el sistema. Un caso de uso se representa mediante una elipse dentro del rectángulo que engloba al sistema y que se une con algún actor.

Además, se mostrarán diagramas de secuencia de los casos de uso más útiles en la aplicación. Los diagramas de secuencia representan cómo se llevan a cabo los casos de uso que generan los actores sobre el sistema y cómo el sistema reacciona realizando acciones solicitadas para proporcionar una respuesta adecuada.

4.2.2. Casos de uso

Podemos ver en la figura 4.1 que tenemos dos actores, administrador y usuario. El primer actor es el usuario, el sujeto que va a colaborar en el proyecto y va a

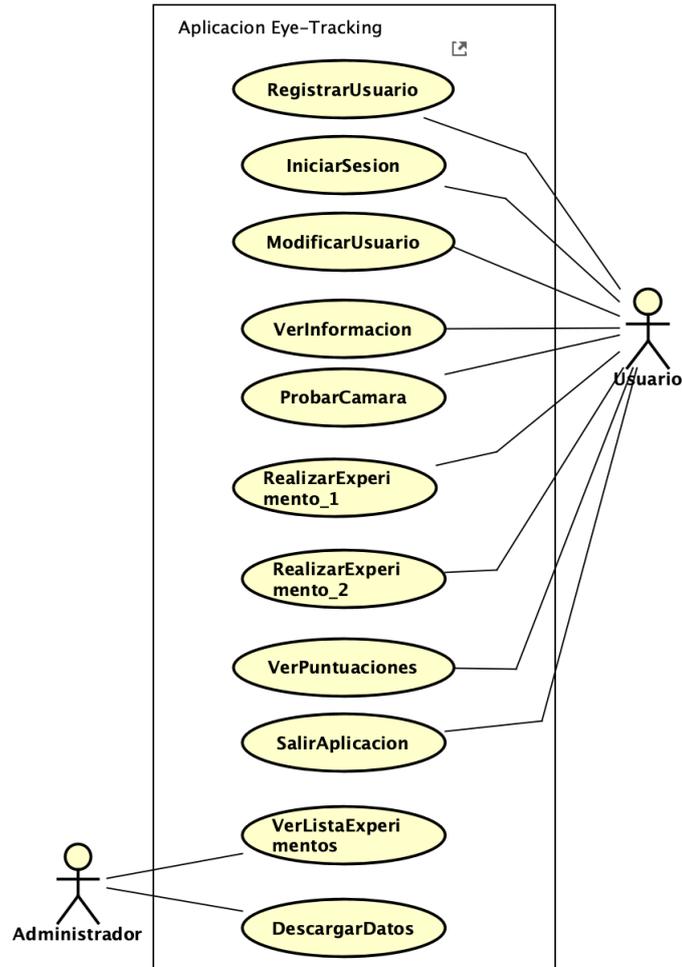


Figura 4.1: Diagrama de casos de uso de la aplicación Eye-tracking

realizar los experimentos. A continuación, se van a detallar los diferentes casos de uso del usuario.

- RegistrarUsuario. Cualquier sujeto que entre en nuestra página web tiene la posibilidad de registrarse para poder realizar los experimentos. el usuario debe cumplimentar un formulario indicando su información personal: color de ojos, edad, sexo, si el usuario realizará los experimentos con gafas o con lentillas; parámetros relacionados con el ordenador que va a utilizar para la realización de los experimentos como las dimensiones de la pantalla o la posición de la webcam que va a utilizar; y parámetros que se usaran para el inicio de sesión de los usuarios: el código de usuario, la contraseña y el correo electrónico.

Hay que destacar que el código de usuario es único para cada usuario y es un código de 4 letras creado de forma aleatoria. Este código se enviará por correo electrónico a los usuarios una vez que se hayan registrado, y no se almacenará el correo en la base de datos, así aseguramos no tener ningún dato personal de los usuarios.

En cuanto a los parámetros del ordenador, el usuario podrá introducir en un

cuadro de texto el valor de la dimensión de la pantalla en pulgadas, si es que lo conoce. En cualquier caso, tanto si lo conoce como si no, se comprobará si las dimensiones introducidas son correctas a través de un dibujo de una tarjeta de la biblioteca, de identificación o de crédito, que aumentará o disminuirá de tamaño hasta coincidir con nuestra tarjeta física. La posición de la webcam se elegirá en base a unas ilustraciones. En el capítulo de Resultados se verán más detenidamente estos aspectos.

- **IniciarSesion.** El usuario puede iniciar sesión en la aplicación, a partir de su código de usuario enviado a su correo electrónico facilitado en el registro y de la contraseña. Una vez iniciada la sesión el usuario puede modificar su información si así lo desea, ver las especificaciones de cada experimentos, probar la cámara de su ordenador para obtener sus parámetros, realizar los experimentos, etc.
- **ModificarUsuario.** Tiene la posibilidad de ver su información del registro, y modificar los parámetros que desee, excepto el código de usuario que no se puede cambiar. En este caso aparecerá una tabla con todos los datos del usuario y si quiere modificar algún parámetro le llevará a un formulario como el que se cumplimentó en el registro.
- **VerInformacion.** En a página principal, el usuario puede ver toda la información específica acerca del proyecto, en qué consiste, quiénes somos, política de privacidad, organizaciones que han colaborado en el proyecto, etc. El usuario debe saber para qué sirve este proyecto y las finalidades que puede llegar a tener.
- **ProbarCamara.** El usuario deberá probar la cámara antes de la realización de los experimentos para que el sistema recoja información sobre ella. Es importante recoger estos parámetros antes de la realización de los experimentos, sobre todo por la resolución de la cámara web con las que estemos trabajando, ya que queremos asegurar la mejor calidad de nuestras imágenes y vídeos.
- **RealizarExperimento1.** Este experimento consiste en captar imágenes con la webcam a partir de un círculo negro que se dispone en la pantalla y que cambia de posición cada 4 segundos. El cambio de posición se realiza de forma aleatoria. Cada vez que el círculo cambia de posición el usuario debe realizar una fijación sobre el objeto en cuestión y mover el ratón del ordenador hacia éste. Una vez que el puntero del ratón esté encima del círculo, deberá pulsar con el click izquierdo. En el momento que se realiza la pulsación el sistema tiene que captar las posiciones tanto del círculo como del ratón, el instante temporal y la imagen. Estos parámetros serán almacenados en la tabla de posicion_tiempo de la base de datos cada vez que se pulse el ratón.

Cuando el experimento haya finalizado el sistema calculará una puntuación del experimento y se almacenará en la tabla de experimento de la base de datos junto a otros parámetros que se quieren almacenar, como pueden ser la fecha de realización del experimento, dimensiones de la pantalla del usuario en centímetros y en píxeles o el identificador de la cámara. Al final del experi-

mento se deberá tener 15 imágenes con sus respectivas posiciones e instantes temporales.

- RealizarExperimento2. Este segundo experimento consiste en seguir la trayectoria continua del círculo negro sobre la pantalla con una duración de 30 segundos. Durante ese tiempo, la webcam estará grabando la posición de la mirada y al final del experimento obtendremos un vídeo y no unas imágenes como en el experimento anterior. Además se van a capturar las posiciones del círculo y del puntero del ratón. Por lo general, las cámaras web graban el vídeo a una velocidad de 30 imágenes por segundo, entonces también es razonable que capturaremos 30 posiciones por segundo con sus respectivos instantes temporales. Estos valores se almacenarán en la tabla de posicion_tiempo de la base de datos.

Una vez acabado el experimento se deberá obtener alrededor de 900 posiciones con sus instantes temporales y el vídeo. En este experimento el usuario también obtiene una puntuación en base a la cercanía entre la posición del círculo y del puntero del ratón. Esa puntuación junto con los demás parámetros del experimento, fecha, hora, dimensiones de la pantalla, etc, también serán almacenados en la tabla de experimento de la base de datos.

- VerPuntuaciones. Una vez finalizado el experimento, el usuario podrá ver la puntuación obtenida del experimento realizado. Además, se ha realizado un ranking con las puntuaciones personales y globales de ambos experimentos, es decir, las clasificaciones son independientes para cada experimento.
- SalirAplicacion. El usuario puede salir de la aplicación siempre que lo desee.

El administrador es el que va a obtener los datos de los experimentos que se realicen y, por ello, tiene dos funciones dentro de la aplicación.

- VerListaExperimentos. Una vez que ha iniciado sesión, tiene la capacidad de poder ver el listado de experimentos que se han realizado en un panel de control que sólo él puede ver. En este panel puede buscar los experimentos según la fecha en la que se han realizado.
- DescargarDatos. Tiene la posibilidad de descargar los datos de los experimentos realizados. En el listado de experimentos podrá descargar un sólo experimento o todos los que le salgan en el día filtrado. El administrador obtendrá un fichero .zip con las imágenes o el vídeo correspondiente al experimento y un fichero .JSON con los parámetros relativos al experimento, al usuario y a la cámara.

A continuación, se va a explicar a través de diagramas de secuencia los casos de uso más relevantes de la aplicación.

Diagramas de secuencia

- Inicio de sesión. El inicio de sesión se realiza tanto si eres usuario como si eres el administrador. En la figura 4.2 se puede observar el diagrama de secuencia de este caso de uso, que en este caso aparece un usuario pero sería igual

para el administrador. El usuario introduce el código de sesión que obtiene al registrarse en la aplicación y la contraseña; el sistema pasa los datos a la base de datos para que verifique a quién corresponden esas credenciales. Una vez que el sistema recibe confirmación, muestra la página de mostrar experimentos para que el usuario escoja el experimento que desee. Una vez que han iniciado

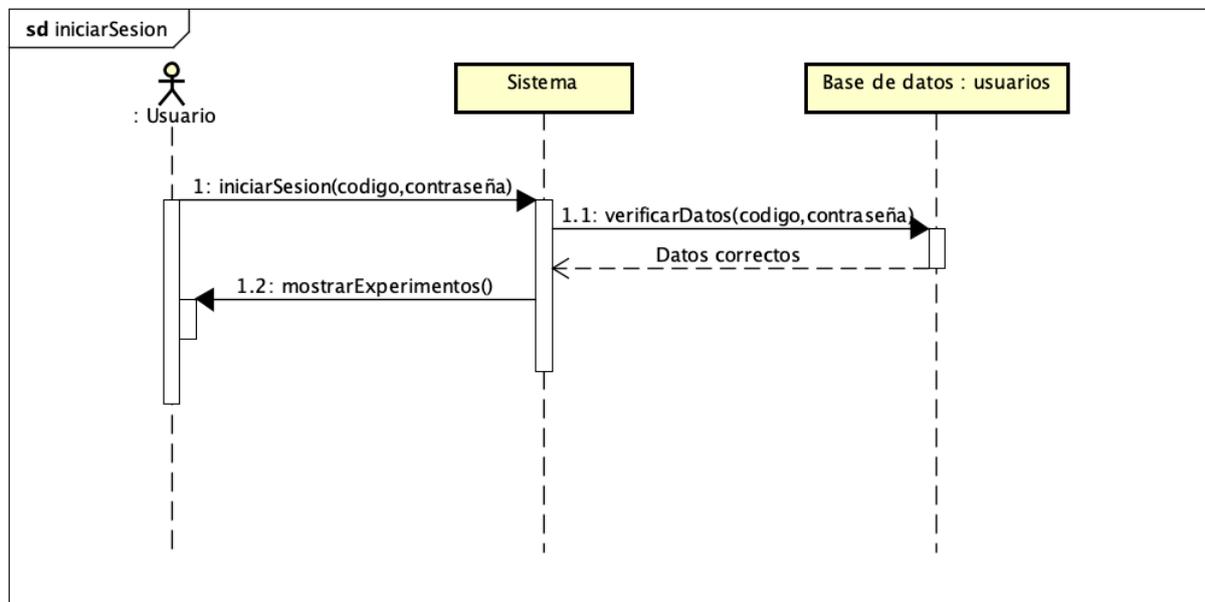


Figura 4.2: Diagrama de de secuencia de iniciar sesión

sesión en el sistema, el usuario tiene diferentes funciones dentro del sistema, como modificar su información si ha habido algún fallo o probar la cámara web antes de la realización de los experimentos, pero la más importante es la de realizar los experimentos. Por ello van a ser los dos casos de uso que se van a exponer a continuación.

- Realizar experimento 1. En la figura 4.3 se presenta el diagrama de secuencia de la realización del experimento número 1 de la aplicación. Una vez que empieza el experimento se van realizando pulsaciones en el ratón cada 4 segundos que es cuando cambia de posición el círculo, cada vez que se pulsa el ratón se guardan los datos en la tabla de posición tiempo de la base de datos. Transcurridos 60 segundos el experimento finaliza y se guardan los datos de la puntuación, junto con otros parámetros, en la base de datos del experimento.
- Realizar experimento 2. En la figura 4.4 se presenta el diagrama de secuencia de la realización del experimento número 2 de la aplicación. El diagrama de secuencia es el mismo que para el primer experimento con la particularidad de que este experimento dura 30 segundos y que los parámetros se guardan en tabla de posición tiempo con mucha más frecuencia dado que el círculo está en continuo movimiento. Además tiene la particularidad de que se almacena un vídeo en la base de datos y no imágenes.

En cuanto al administrador, sólo tiene una función, que es la de listar y descargar los datos de los experimentos que se han realizado. Por tanto, éstos son los diagramas de secuencia que se van a analizar.

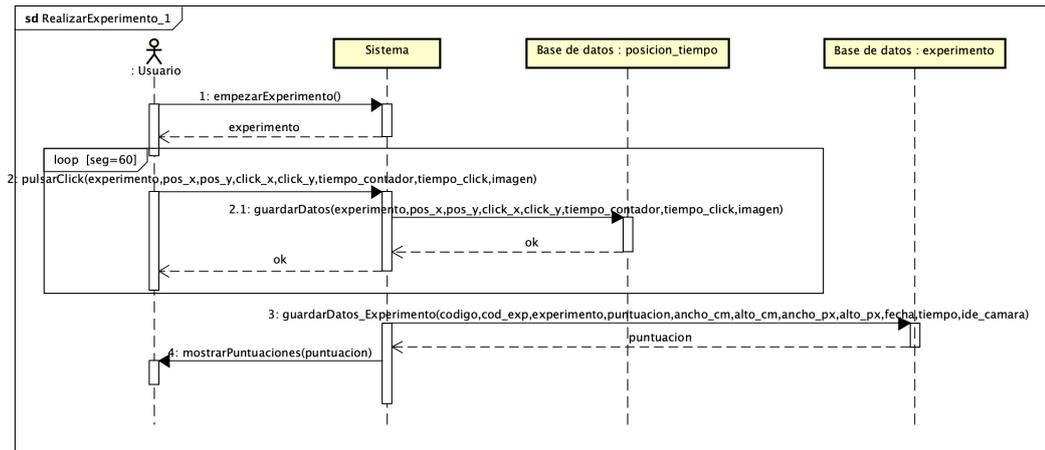


Figura 4.3: Diagrama de de secuencia de realizar el experimento número 1.

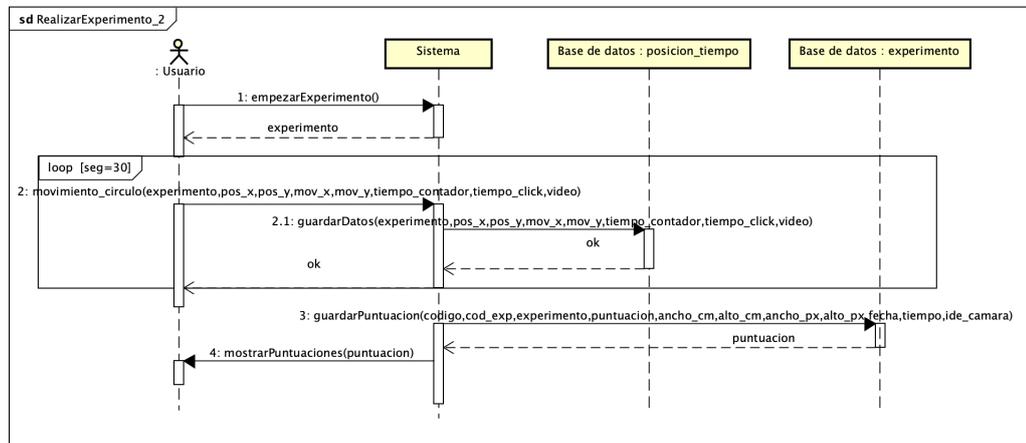


Figura 4.4: Diagrama de de secuencia de realizar el experimento número 2.

- Lista de experimentos. El administrador está dentro del sistema y la única función que puede hacer es la de descargar los datos de los experimentos, pero para ello debe obtener un listado de los mismos. Debe introducir una fecha y el sistema buscará en la tabla de experimentos de la base de datos la fecha que coincida con la introducida. La base de datos le devolverá al sistema el listado de los experimentos que coincidan con esa fecha. En la figura 6.21 se puede apreciar como el administrado envía al sistema la fecha para que éste la mande a la base de datos.
- Descargar datos de los experimentos. El administrador llega a su objetivo que es el de descargar los datos que se han almacenado en la base de datos. A partir de la lista que se obtiene del caso de uso anterior, el administrador tiene la posibilidad de descargar los experimentos que quiera según la fecha introducida. Cuando le llegue la petición al sistema como vemos en la figura 4.6 recogerá los datos asociados a ese experimento o grupo de experimentos y generará un fichero .zip que se descargará automáticamente.

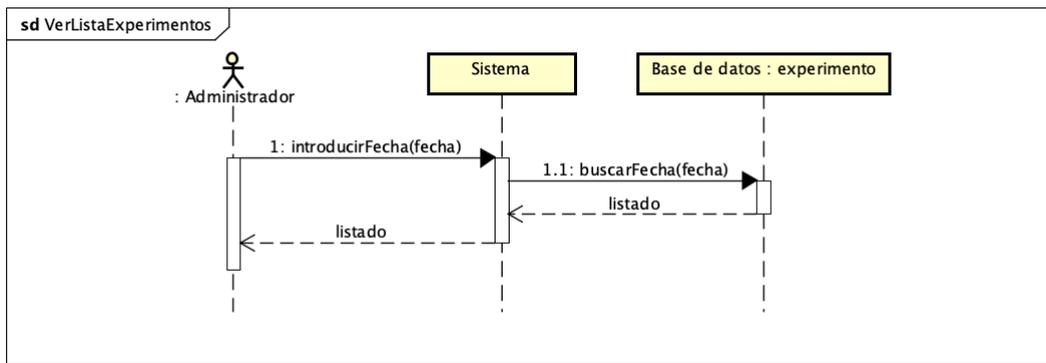


Figura 4.5: Diagrama de de secuencia de ver lista de experimentos

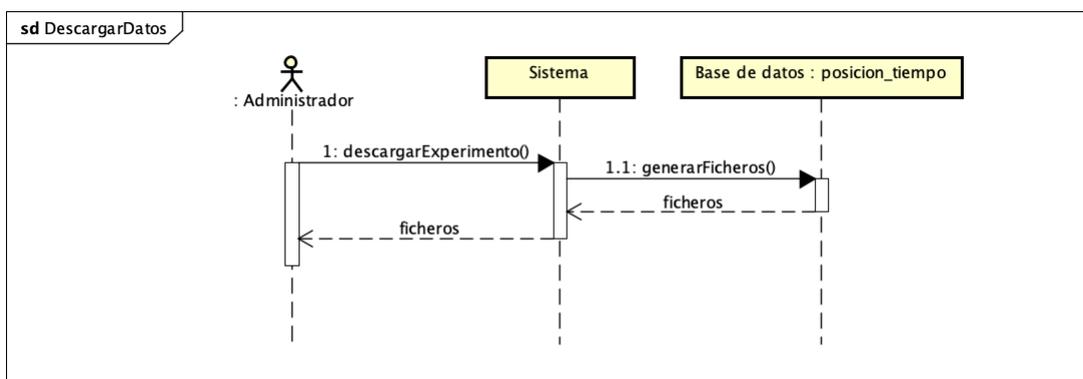


Figura 4.6: Diagrama de de secuencia de descargar datos de los experimentos.

4.3. Diseño

En esta sección se va a realizar una explicación del diseño desarrollado en la interfaz del usuario y de la base de datos.

4.3.1. Diseño de la interfaz de usuario

En este apartado se va a realizar una explicación del diseño que se ha seguido a la hora de realizar la interfaz del usuario. Para ello contamos con bocetos que se realizaron antes de la realización de a página.

En primer lugar, la estructura que debía tener la página es la que se muestra en la figura 4.7 dado que necesitamos una cabecera o header para que el usuario pueda interactuar con las demás ventanas de la página, el cuerpo de la página en la que mostraremos la información necesaria y un footer en el que esencialmente se mostrarán las políticas de privacidad y la información de contacto.

La página principal de la aplicación web debe tener una estructura en la que se explique el propósito del proyecto y quién está detrás de esta aplicación. Por ello, se han realizado varias partes en el cuerpo de la página, incluyendo esta información, y además los logotipos de las empresas e instituciones colaboradoras en el proyecto, como podemos ver en la figura 4.8.

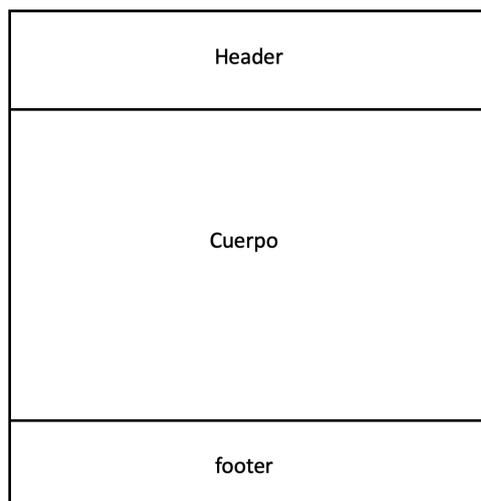


Figura 4.7: Estructura del sitio web

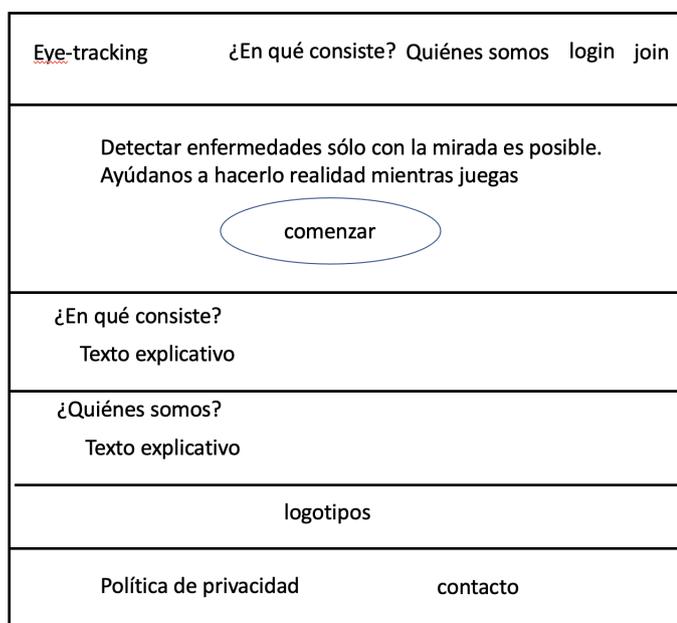


Figura 4.8: Diseño de la página principal de la aplicación.

En cuanto a las cabeceras o headers, hay que tener en cuenta que los usuarios se va a registrar en la aplicación, por tanto, debemos tener una cabecera diferente para cada uno de los dos casos: si el usuario ha iniciado sesión o si el usuario aún no lo ha hecho. Esto es bastante importante sobre todo la hora de realizar los experimentos. Un usuario, sin haber iniciado sesión, no puede realizar experimentos ya que no tenemos datos sobre él. Por ello se crearon estas dos cabeceras: interna (figura 4.9(b)) y externa (figura 4.9(a)).

El usuario al entrar en la página lo primero que debe hacer es realizar el registro en la aplicación. Estos datos se le piden al usuario con un formulario como el que aparece en la figura 4.10. En este formulario se le piden los datos personales, de

Eye-tracking ¿En qué consiste? Quiénes somos login join

(a) Diseño de la cabecera externa de la aplicación.

Eye-tracking ¿En qué consiste? Quiénes somos Info Experimentos Log out
--

(b) Diseño de la cabecera interna de la aplicación.

Figura 4.9: Cabeceras interna y externa de la aplicación web.

la cuenta, y relativos a la webcam de su ordenador. El código de usuario aparece en la figura en rojo ya que es un valor que no se va a poder modificar.

Header externo								
Información personal <table border="1"> <tr><td>Edad</td></tr> <tr><td>Sexo</td></tr> <tr><td>Color ojos</td></tr> <tr><td>Lentillas</td></tr> <tr><td>Gafas</td></tr> </table>	Edad	Sexo	Color ojos	Lentillas	Gafas	Cuenta de usuario Código usuario <table border="1"> <tr><td>Contraseña</td></tr> <tr><td>email</td></tr> </table>	Contraseña	email
Edad								
Sexo								
Color ojos								
Lentillas								
Gafas								
Contraseña								
email								
Tamaño pantalla Dimensiones: <input type="text"/> <table border="1"> <tr><td>tarjeta</td></tr> </table>		tarjeta						
tarjeta								
Posición webcam Cuatro posiciones de la webcam indicando donde se encuentra Aceptacion de los términos <input type="radio"/> <table border="1"> <tr><td>Crear usuario</td></tr> </table>		Crear usuario						
Crear usuario								
footer								

Figura 4.10: Diseño de formulario del registro de la aplicación

Cuando un usuario se ha registrado en la aplicación, ya puede iniciar sesión en la aplicación, para ello debe tener el código de usuario (se manda por correo electrónico al usuario, por si se pierde o si se le olvida) y la contraseña que se introdujo en el registro. El esquema de inicio de sesión se muestra en la figura 4.11.

Una vez que el usuario ha iniciado sesión puede realizar dos acciones. En primer lugar, ver la información de su cuenta, y modificar datos o borrar la cuenta (figura 4.12). En segundo lugar, puede ir a la página de experimentos, como se muestra en la figura 4.13. En el primer caso el usuario puede modificar sus datos al clicar en modificar datos, de forma que aparecerá un formulario muy similar al del registro, o puede borrar su cuenta del sistema. La segunda opción es la de realizar los experimentos y

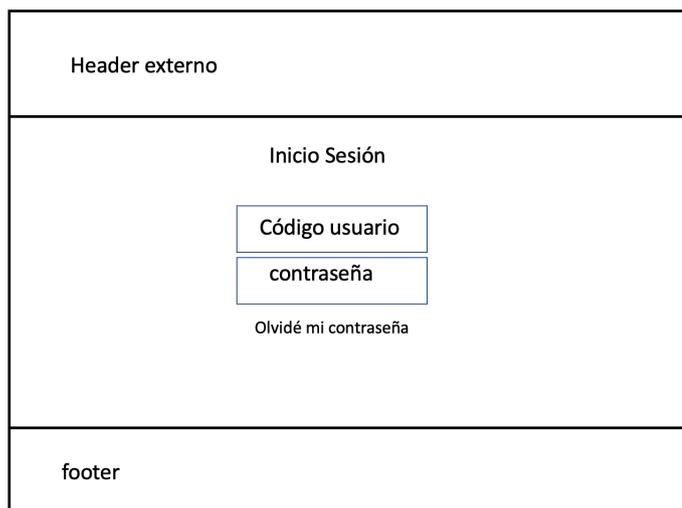


Figura 4.11: Diseño de la página de inicio de sesión de la aplicación

en la figura 4.13 se realizó un esquema con dos posibles experimentos y una breve explicación de lo que consisten. A partir de aquí se inicia uno de los dos experimentos.

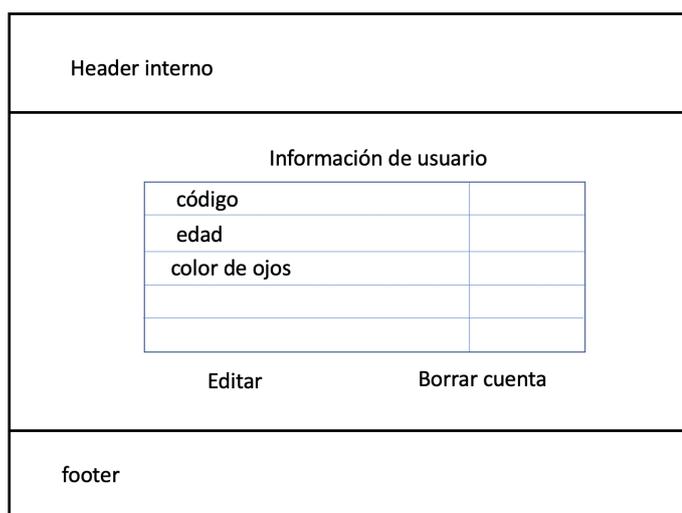


Figura 4.12: Diseño de la página de información de un usuario de la aplicación

El usuario entra en uno de los experimentos y tienen que aparecer las instrucciones de cómo se realiza dicho experimento, por ello, se realizó un esquema de la página de información de experimentos como la que aparece en la figura 4.14.

Cuando el usuario ya ha realizado el experimento pueden ocurrir dos cosas: el experimento ha ido correctamente, y en ese caso se pasa a la ventana de fin de experimentos donde aparecen las puntuaciones cuyo esquema se hizo como aparece en la figura 4.15, o si ha habido algún fallo en la realización del experimento aparecerá una ventana de fallo de experimento, cuyo diseño aparece en la figura 4.16.

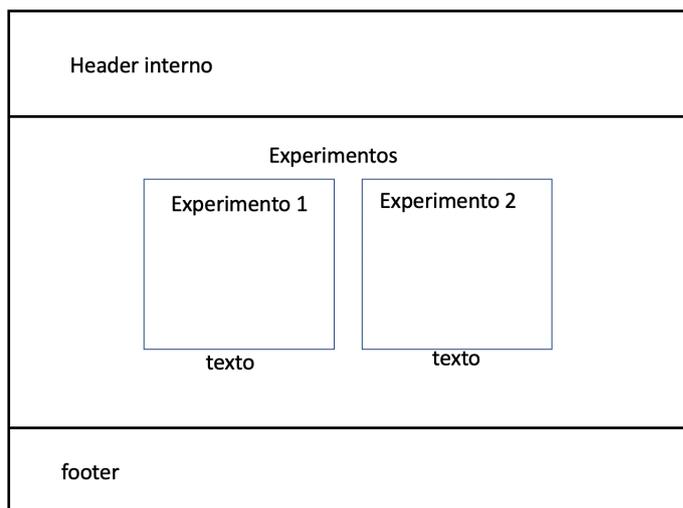


Figura 4.13: Diseño de la página de experimentos de la aplicación

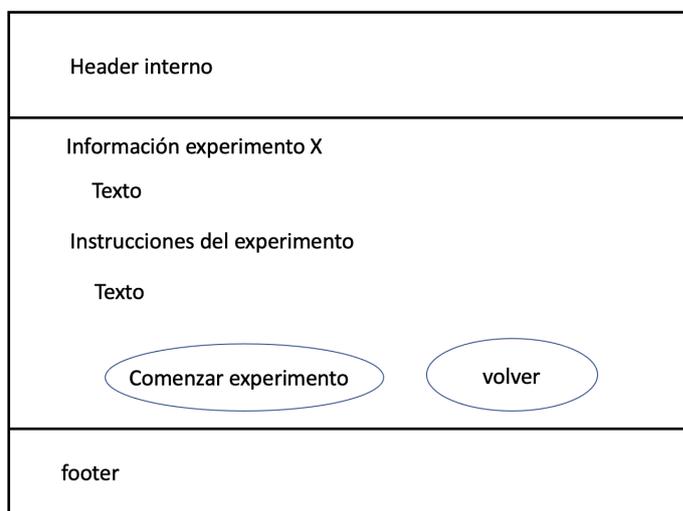


Figura 4.14: Diseño de la página de información de experimentos de la aplicación

4.3.2. Diseño de la base de datos

Este apartado está dedicado al diseño de la base de datos utilizada en la aplicación desarrollada. A continuación, se explicará cuál es el sistema de gestión de base de datos que se va a utilizar; posteriormente se detallará la estructura implementada en este proyecto y los datos que alberga.

Sistemas de gestión de bases de datos

A la hora de realizar una aplicación web es esencial elegir el sistema de gestión de base de datos que se va a utilizar. Como ya se explicó en el Capítulo 3, el sistema de gestión de bases de datos que se va a utilizar es MySQL.

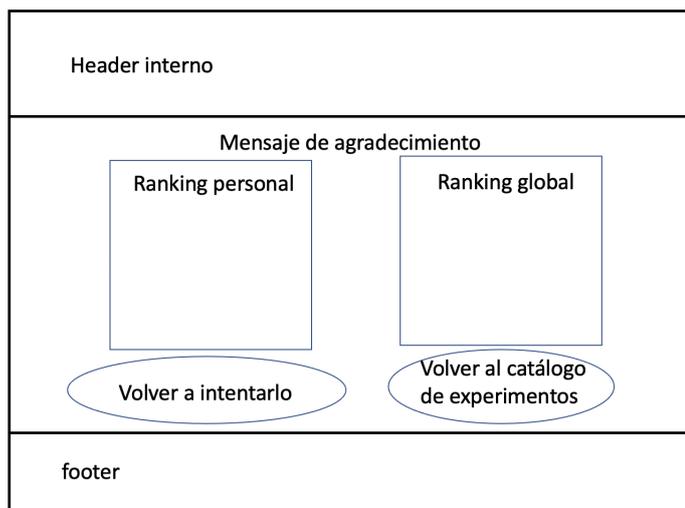


Figura 4.15: Diseño de la página de ranking de las puntuaciones de los usuarios de la aplicación

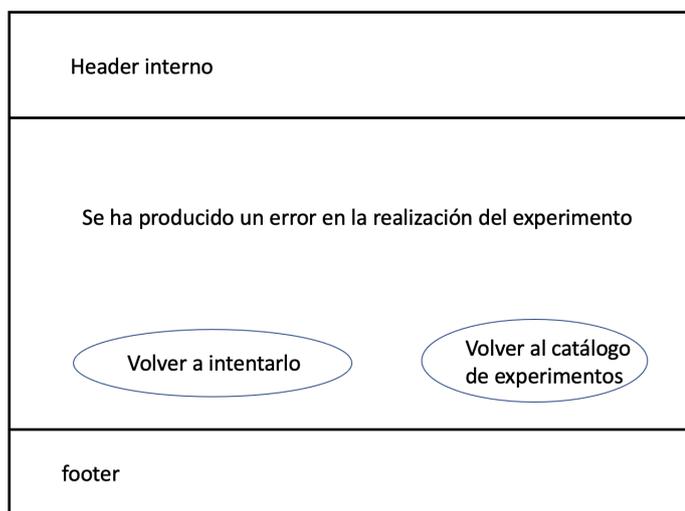


Figura 4.16: Diseño de la página de error de experimentos de la aplicación

Estructura de la base de datos

La base de datos de nuestro proyecto presenta 4 tablas: usuario, experimento, posicion_tiempo e info_camara. En la figura 4.17 se pueden ver las diferentes tablas con sus respectivas cardinalidades y con sus parámetros. A continuación, se van a ir describiendo los diferentes parámetros de las tablas que se muestran.

- Tabla de usuario. Relativa a los datos del usuario que se almacenan tras el proceso de registro.
 - **codigo**: valor de 4 letras generado aleatoriamente al realizar el registro. Es único para cada usuario y por eso es la clave primaria de la tabla de usuarios. Este valor es enviado por correo electrónico al usuario.
 - **pass**: contraseña que el usuario introducirá en el registro y que le servirá

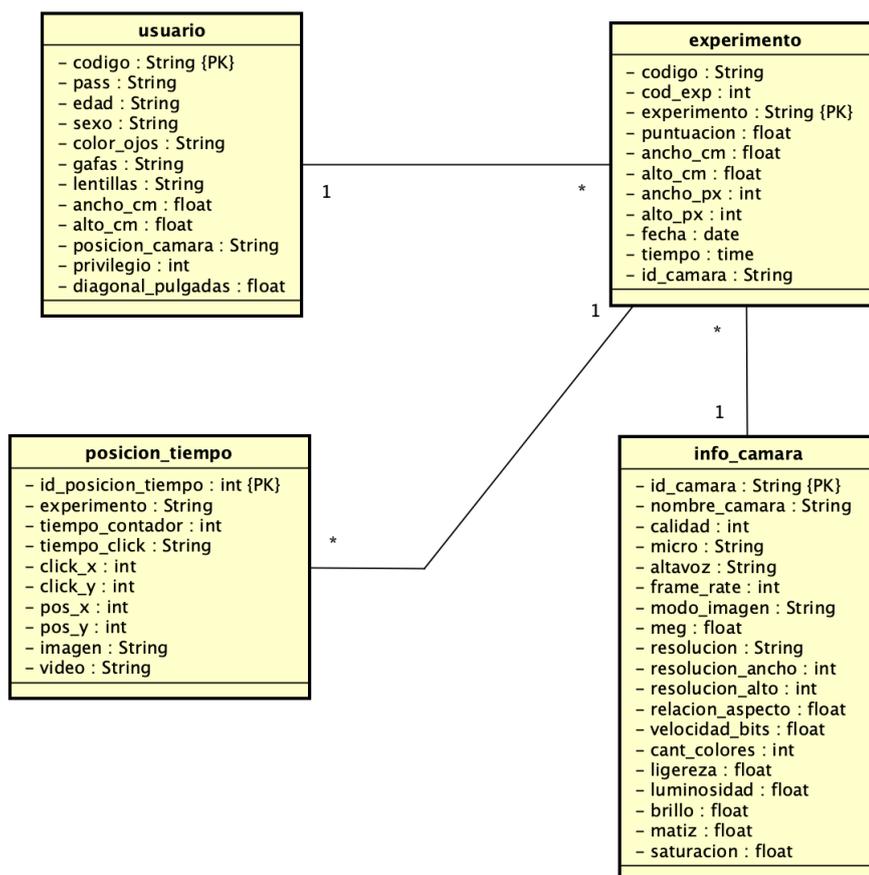


Figura 4.17: Base de datos de la aplicación Eye-tracking.

para iniciar sesión en la aplicación web. Esta contraseña debe contener más de 6 caracteres y menos de 16 para que sea válida.

- **edad:** edad del usuario que se registra en la página. Debe tener 18 años o más para poder registrarse y realizar los experimentos.
- **sexo:** especificación de sexo, si es masculino o femenino.
- **color_ojos:** especificación del color de los ojos, el usuario deberá elegir entre: verdes, marrones, azules, grises, miel y negros.
- **gafas:** especificación de si el usuario va a realizar los experimentos con gafas o sin ellas.
- **lentillas:** especificación de si el usuario va a realizar los experimentos con lentillas o sin ellas.
- **ancho_cm:** ancho en centímetros de la pantalla del ordenador con el que va a realizar el experimento.
- **alto_cm:** alto en centímetros de la pantalla del ordenador con el que va a realizar el experimento.
- **diagonal-pulgadas:** diagonal en pulgadas de la pantalla del ordenador con el que va a realizar el experimento.

- **posicion_camara:** posición en la que se encuentra la webcam del ordenador, pueden ser: estándar (la webcam esta en la parte central superior de la pantalla del ordenador), a la derecha de la pantalla, a la izquierda de la pantalla y en otra posición.
- Tabla de `posicion_tiempo`. Los datos se almacenan en esta tabla según transcurre un experimento
 - **id_posicion_tiempo:** identificador de cada posición, tiempo e imagen que se almacena en la tabla. Es un identificador autoincremental y es la clave primaria de esta tabla.
 - **experimento:** código de identificación del experimento que se está realizando. Éste se genera a partir de la fecha y la hora de la realización del experimento y el código del usuario que lo está realizando.
 - **tiempo_contador:** contador del número de posiciones e imágenes que se almacenan en un experimento.
 - **click_x:** coordenada x de la posición del ratón en el momento de la captura de la imagen o del vídeo.
 - **click_y:** coordenada y de la posición del ratón en el momento de la captura de la imagen o del vídeo.
 - **pos_x:** coordenada x de la posición del círculo.
 - **pos_y:** coordenada y de la posición del círculo.
 - **imagen:** imagen capturada en el momento que se realiza la pulsación del ratón. Será un parámetro NULL cuando se realice el experimento 2.
 - **video:** vídeo capturado tras finalizar el experimento. En este caso almacenaremos este campo en el experimento 2, en el caso de que se realice el experimento 1 este valor será NULL.
- Tabla de `experimento`. Los datos del experimento se almacenan tras la finalización del mismo.
 - **cod_exp:** código que diferencia el experimento que se ha realizado. En el caso de haberse realizado el experimento 1, obtendrá un valor de 1, por el contrario, si se realiza el experimento 2, obtendrá un valor de 2.
 - **puntuacion:** La puntuación se obtiene tras la realización del experimento. Sirve para mostrar al usuario una realimentación de cómo lo ha realizado. Esta puntuación se calcula en función de la distancia que haya entre el puntero del ratón y la posición del círculo en el momento de la captura de los datos. Esta puntuación comprende el rango de 1 a 100, siendo 1 la peor puntuación y 100 la mejor.
 - **ancho_px:** ancho de la pantalla en píxels.
 - **alto_px:** alto de la pantalla en píxels.
 - **fecha:** fecha de la realización del experimento. El formato es AAAA-MM-DD, donde AAAA corresponde al año, MM al mes y DD al día.
 - **tiempo:** hora de la realización del experimento. El formato es HH:MM:SS, donde HH es la hora, MM los minutos y SS los segundos.

- **id_camara:** identificador de la cámara del usuario. Este valor se genera antes de realizar el experimento cuando se prueba la cámara. Está compuesto por el nombre de la cámara del usuario
- Tabla `info_camara`. Tabla relacionada con los parámetros de la cámara web del usuario. Los datos se almacenan antes de la realización del experimento, cuando se prueba la cámara.
 - **nombre_camara:** nombre de la cámara del usuario.
 - **calidad:** calidad de la cámara del usuario.
 - **micro:** parámetro que indica si se está utilizando el micrófono del usuario al activar la cámara.
 - **alravez:** parámetro que indica si se está utilizando el altavoz del usuario al activar la cámara.
 - **frame_rate:** velocidad de imágenes por segundo que puede capturar la cámara.
 - **modo_imagen:** modo de la imagen a capturar, pueden ser RGB o CMYK entre otras.
 - **meg:** resolución de la cámara en megapíxels.
 - **resolucion:** resolución de la cámara. El formato de este campo es NNNNxMMMM, donde NNNN es la `resolucion_ancho` y MMMM es la `resolucion_alto`.
 - **resolucion_ancho:** es el ancho de las imágenes y el vídeo que se capturen. Dado por el parámetro anterior.
 - **resolucion_alto.** es el alto de las imágenes y el vídeo que se capturen. Dado por el campo de `resolucion`.
 - **relacion_aspecto.** la relación de aspecto es la proporción entre el ancho y el alto.
 - **velocidad_bits:** es la velocidad con la que se capturan las imágenes de la cámara en bits.
 - **cant_colores:** cantidad de colores que puede recoger la cámara del usuario.
 - **ligereza:** se mide en %.
 - **luminosidad:** también llamada claridad, es la cantidad de luz emitida o reflejada por un objeto, en nuestro caso de la imagen o del vídeo. Se mide en %.
 - **brillo:** parámetro de sensación visual por la zona que parece mostrar más o menos luz. Se mide en %.
 - **matiz:** parámetro que diferencia un color de otro. Se mide en %.
 - **saturation:** intensidad o grado de pureza del color. Se mide en %.

Capítulo 5

Implementación y resultados

5.1. Introducción

En este capítulo se va a exponer una descripción de la implementación y un análisis de los resultados obtenidos de la realización de la aplicación web. Tendremos tres secciones, la interfaz del usuario, la interfaz del administrador y pruebas. En la primera de ellas, se hará una explicación de los aspectos más importantes sobre la interfaz del usuario. En la segunda sección se mostrará la interfaz del administrador y cómo se ha implementado para poder descargar los datos, ya que es la principal función que tiene este usuario. Por último, se mostrarán las pruebas que se han realizado para asegurar que la aplicación funciona correctamente.

Esta aplicación web se ha desarrollado en un servidor local, ya que es más sencillo realizar pruebas de como funciona y más cómodo que tener que ir modificando y conectándonos al servidor de la escuela. Pero el objetivo de esta aplicación es que llegue a gran cantidad de usuarios y tener un servidor que guarde todos los datos que se van generando. Por ello, se ha proporcionado una máquina virtual del grupo de investigación de la escuela para poder conectarnos con su servidor y hacer el despliegue.

Por tanto, en esta sección se va a explicar la implementación que se ha realizado en base al servidor local. En el último apartado de la sección se verán las pruebas que se han realizado en cuanto al despliegue en el servidor de la escuela.

5.2. Interfaz de usuario

La interfaz de usuario es la que se encarga de: registrar a los usuarios, iniciar sesión, hacer pruebas de la cámara, realizar los experimentos y mostrar las puntuaciones. Dentro de estas funcionalidades en este apartado se van a exponer las implementaciones más relevantes.

5.2.1. Formulario de la aplicación

En primer lugar el usuario debe registrarse en el sistema a través del formulario que aparece en la figura 5.1. En este formulario se deberá registrar la información personal del usuario, las dimensiones de la pantalla y la posición de la webcam.

Además, se deberá aceptar los términos y condiciones establecidas para la realización de dichos experimentos.

(a) Primera parte del formulario.

(b) Segunda parte del formulario.

Figura 5.1: Formulario de la aplicación web

El formulario se ha realizado en base al elemento *form* de HTML, y lo más relevante es de este formulario es la obtención de las dimensiones de la pantalla del usuario que se explica a continuación.

Obtención de las dimensiones de la pantalla del usuario

Uno de los principales retos del proyecto consiste en cómo obtener las dimensiones de la pantalla del usuario. El primer paso era encontrar cómo poder encontrar el ancho y el alto de la pantalla en píxeles. Para ello se han utilizado una funciones que vienen predefinidas en javascript a partir de los objetos *document*, *window* y *screen*, dependiendo del navegador que estemos usando [12]. Una vez que tenemos las dimensiones de la pantalla en píxeles hace falta tenerlas en centímetros ya que nos interesa tener dimensiones reales para luego poder realizar el análisis de los datos.

Encontrar las dimensiones de la pantalla en centímetros no es tan inmediato como en píxeles y hay que realizar cálculos. Para poder obtener los valores en centímetros hay que calcular esos valores en pulgadas y una vez que tenemos la proporción en pulgadas ya podremos pasarlo a centímetros.

- Conversión de píxeles a pulgadas

La conversión de píxeles a pulgadas se realiza a partir de un cuadro de 1 pulgada x 1 pulgada y las dimensiones de una tarjeta de identificación. El cuadro de 1 pulgada x 1 pulgada se dibuja en una zona de la pantalla que no se ve y nos servirá para ver la proporción de píxeles que hay en 1 pulgada cuadrada, es decir, obtendremos los píxeles por pulgada en las coordenadas x e y. Por

otro lado necesitamos saber una medida real de la pantalla, por eso se dibuja una tarjeta de identificación como la que aparece en la figura 5.2, que será una modificación de la que aparece en la referencia [13]. El usuario pulsará los botones de *Aumentar tamaño* y *Disminuir tamaño* para hacer coincidir las dimensiones de la tarjeta dibujada con la tarjeta real que tiene; cada vez que el usuario pulsa estos botones aumenta o disminuye la escala por la que se dividirán las dimensiones en píxeles. Podemos apreciar las expresiones las dimensiones en pulgadas de la pantalla en 5.1, 5.2 y 5.3, donde dpi_x y dpi_y son los píxeles por pulgada que describíamos anteriormente.



Figura 5.2: Tarjeta de identificación del registro que sirve para obtener las dimensiones de la pantalla

$$ancho_pulgadas = \frac{ancho_px}{dpi_x * escala} \quad (5.1)$$

$$alto_pulgadas = \frac{alto_px}{dpi_y * escala} \quad (5.2)$$

$$diagonal_pulgadas = \sqrt{(ancho_pulgadas^2 + alto_pulgadas^2)} \quad (5.3)$$

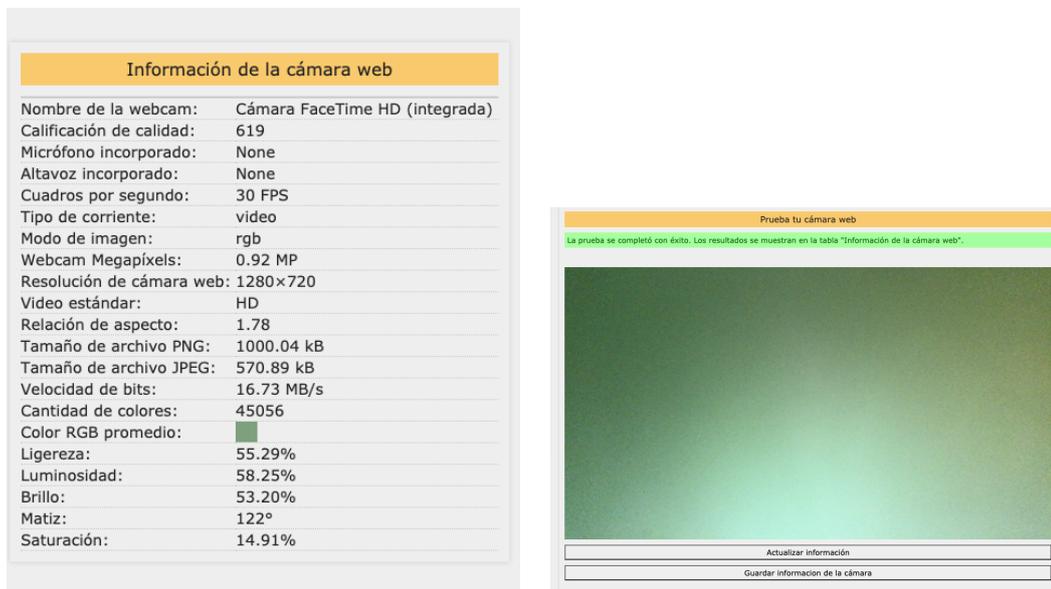
- Conversión de pulgadas a centímetros.

Una vez que hemos calculado las dimensiones en pulgadas, realizar el cálculo de las dimensiones en centímetros es inmediato. Sabiendo que una pulgada equivale a 2.54 centímetros, las expresiones de las dimensiones en centímetros quedarían multiplicadas por ese factor. Estos valores serán almacenados en la base de datos en la tabla de usuarios.

Cabe destacar que los experimentos se deben realizar en un ordenador portátil o de sobremesa, no es compatible para tablets o móviles. Por tanto, antes de que el usuario realice los experimentos el sistema debe cerciorarse de que se está realizando en un ordenador. Para ello, se ha realizado una comprobación de los agentes de usuario que se están utilizando y así ver si nos encontramos en una tablet o un móvil [14]. En ese caso no se podrá realizar el experimento y deberá iniciar sesión con un ordenador.

5.2.2. Obtención de los parámetros de la cámara

Uno de los problemas que se presentaban era que al generar las imágenes del primer experimento no se determinaba ninguna resolución para ellas. Es decir, podíamos generar imágenes de cualquier tamaño pero no se sabía cual podía ser el adecuado para que las imágenes fuesen de buena calidad. Por tanto, se encontró una página web en la que se realizaba una prueba de la webcam y se obtenían muchos parámetros, entre los que se encontraba la resolución de la cámara web [15]. Además, se obtuvieron parámetros secundarios para la aplicación, pero aún así se han guardado en la base de datos por si en el futuro pueden ayudar a analizar datos. Un ejemplo de la obtención de estos parámetros se puede ver en la siguiente figura 5.3. En la figura 5.3(a) se puede ver un ejemplo de los diferentes parámetros que se recogen de la cámara y en la 5.3(b) se muestra un lienzo con la grabación que se esta haciendo al usuario. Además al terminar esta prueba disponemos dos botones uno para guardar la información en la base de datos y otra por si nuestros resultados nos han sido satisfactorios volver a realizar la prueba.



(a) Ejemplo de información de los parámetros de la cámara. (b) Ejemplo de la visualización de la pantalla de probar la cámara.

Figura 5.3: Ejemplo de obtención de los parámetros de la cámara.

5.2.3. Realización del experimento 1

En este apartado se va a explicar los diferentes pasos que se han ido realizando para el desarrollo del experimento 1. Este experimento se basa fundamentalmente en seguir con la mirada y con el ratón un círculo negro que aparece y desaparece en lugares aleatorios de la pantalla. El círculo permanece durante 4 segundos estático en una parte de la pantalla, cuando el usuario esté encima del círculo con el ratón debe clicar con el botón izquierdo del ratón y se realizará la captura con la webcam. El experimento tiene una duración de 60 segundos. A partir de esta pequeña explicación se exponen las diferentes componentes del experimento.

Generación del canvas y del círculo

El experimento se pensó en realizar sobre un lienzo o canvas, ya que es un elemento HTML que se utiliza para dibujar gráficos, cuadros, círculos, textos o imágenes [12]. El experimento debía tener un círculo sobre el canvas que fuese apareciendo y desapareciendo, y en una primera instancia se pensó en representar directamente el círculo sobre el mismo lienzo a través de los métodos predefinidos del canvas. Después, se pensó en cambiar la forma del círculo, ya que éste era demasiado básico y se creó una estructura `div` encima del lienzo para que fuese más atractivo para el usuario. Este círculo contaba con un pulsación alrededor del círculo y el código se obtuvo de una de las referencias que se exponen [16]. Se puede ver una imagen de este círculo en la figura 5.4.



Figura 5.4: Círculo negro con aura azul principal elemento de los experimentos.

Una vez que se implementó el canvas y el círculo se necesitaba que las dimensiones del canvas fuesen las dimensiones de la pantalla. Las dimensiones de la pantalla ya se calcularon antes de empezar el experimento, por tanto, se le asignaron esos valores al ancho y largo del canvas. Aunque se habían definido esas dimensiones al lienzo, surgía el problema de que había partes de la página web que no debían aparecer en la pantalla como la barra de navegación, ya que no se estaba optimizando toda la pantalla del ordenador. Por ello, se decidió realizar una función con los métodos de los elementos de los diferentes navegadores con las solicitudes para que se pudiese en pantalla completa. Algunos de estos métodos son `elem.requestFullscreen()`, `elem.mozRequestFullScreen()` o `elem.webkitRequestFullscreen()` [8].

Generación del código de experimento

El código del experimento se genera cada vez que se inicia un experimento y está asociado a cada usuario. Éste se genera a partir de la fecha y la hora de inicio del experimento, el tipo de experimento y el código de usuario. Así, se está garantizando que no haya dos códigos de experimento iguales. La generación de este código se puede ver en la figura 5.5 en la que se calcula la fecha y la hora a partir del objeto `Date` [8].

Este código también nos servirá para crear el nombre del directorio donde guardaremos las imágenes, por tanto, una vez esté generado crearemos el directorio con ese nombre a partir de la función `mkdir` [17].

Generación de posiciones del círculo

En este experimento se generan posiciones del círculo cada 4 segundos. Para ello se ha realizado una función llamada `dibujar_circulo` que llama a la función

```

//Calculamos la fecha y la hora con el objeto date que hemos creado anteriormente
fecha=hoy.getFullYear()+'-'+(hoy.getMonth() + 1)+'-'+hoy.getDate();
hora=hoy.getHours()+'_'+hoy.getMinutes()+'_'+hoy.getSeconds();

//generamos el código de experimento
codigo_experimento=fecha+'_'+hora+'_1_'+usuario;

```

Figura 5.5: Fragmento de código asociado a la generación del código de experimento.

nueva_posicion para que calcule la nueva posición. La función *nueva_posicion* se encarga de calcular la nueva posición del círculo a partir de las dimensiones de la pantalla y las dimensiones del círculo, ya que el círculo siempre debe quedar dentro de la pantalla para que el usuario pueda verlo completamente y pulsarlo en su parte central. Para generar la posición aleatoria utilizamos funciones aleatorias como *Math.random()* multiplicado por el valor del ancho o el alto de la pantalla menos el radio del círculo. Esto nos dará un valor decimal, y esto no nos interesa ya que nuestras posiciones deben ser posiciones enteras; por tanto, redondearemos ese valor al valor de las unidades con la función *Math.floor()* [8]. Una vez que tenemos la nueva posición se dibuja el círculo sobre el canvas cada 4000 milisegundos como dijimos anteriormente, con la ayuda de la función *setInterval()*.

Generación de clicks e imágenes

Una vez que cambia de posición el círculo, el usuario tiene 4 segundos para realizar la pulsación del click sobre él. Cuando el usuario pulsa el click del ratón, salta un evento y se llama a la función *posicion* para calcular las coordenadas en píxeles del ratón. Cada vez que se pulsa el botón izquierdo del ratón salta ese evento, pero debemos controlar si ha sido la primera pulsación que se ha realizado en los 4 segundos ya que sólo se debe de almacenar esa y no las demás. Por tanto, se propuso un contador para verificar si es la primera, segunda, tercera o n-ésima vez que se pulsaba el ratón. Además, para ayudar al usuario a saber si pulsa el ratón o no, se ha implementado otro círculo de las mismas dimensiones pero con la pulsación de color rojo. En el caso de ser la primera vez que se pulsa el ratón en ese intervalo de tiempo se generará la imagen asociada.

Las imágenes de la webcam se generan a partir de otro canvas que esta superpuesto al que tenemos para visualizar el experimento. En el instante de pulsación del click del ratón se parará la grabación de la webcam para realizar la captura de la imagen y copiarla sobre el canvas. Después debemos guardar la imagen en base 64 con formato png para guardarla en el servidor. Las imágenes se guardan en un directorio llamado *capturas*, y dentro de éste en otro con el nombre del código del experimento asociado. Las imágenes en sí se guardan con un identificador único que es el identificador de la tabla de posición.tiempo que es autoincremental.

Temporizador

El experimento tiene una duración de 60 segundos, por tanto, se creó un temporizador para poder parar la grabación de la webcam y terminar el experimento tras ese tiempo. El temporizador empieza a contar cuando empieza la grabación

de la webcam y llamamos al objeto `Date()` para que quede registrado ese instante. Se puede ver el código en la figura 5.6, en el que se calcula el tiempo transcurrido entre el comienzo del experimento y el instante actual. Una vez obtenido ese valor, realizamos una conversión a milisegundos, segundos y minutos. Cuando se cumpla el tiempo necesario se termina de contar llamando a la función `parar`. Este código es una modificación del código encontrado en la referencia [18]. Al terminar el experimento ya se podrá realizar el cálculo de las puntuaciones.

```
//funcion del temporizador
function timer() {
  actual = new Date() //fecha en el instante actual
  cro = actual - emp; //tiempo transcurrido en milisegundos desde el instante actual al comienzo del experimento
  cr = new Date() //fecha donde guardamos el tiempo transcurrido
  cr.setTime(cro);
  cs = cr.getMilliseconds() //conversión a milisegundos
  sg = cr.getSeconds(); //conversión a segundos
  mn = cr.getMinutes(); //conversión a minutos

  if (cs < 10) { cs = "00" + cs; }
  else if (cs < 100) { cs = "0" + cs; }
  if (sg < 10) { sg = "0" + sg; }
  if (mn < 10) { mn = "0" + mn; }

  //cuando llevemos un minuto se termina el experimento y se llama a la función parar
  if (mn == 1) {
    parar();
  }
}
```

Figura 5.6: Fragmento de código asociado a la función del temporizador.

Cálculo de puntuaciones

Cuando ha terminado el experimento se realizan unos cálculos para obtener la puntuación del experimento. Esto fue una idea que se pensó para que el usuario supiese si ha realizado el experimento bien, mal o regular, además de motivarlo a seguir jugando. El cálculo se realiza en función de la distancia entre el punto central del círculo (coordenadas del círculo en píxeles) y el punto de pulsación del ratón (coordenadas del click del ratón). Dado que las posiciones se recogen en vectores, se ha realizado un bucle para recorrer los valores e ir calculando las distancias entre los puntos. Se ha utilizado la expresión 5.4 para ese cálculo, donde `click_x` y `click_y` son las coordenadas del click y `pos_x` y `pos_y` son las coordenadas del círculo.

$$distancias = \sqrt{(click_x - pos_x)^2 + (click_y - pos_y)^2} \quad (5.4)$$

Las distancias se van sumando en una variable para hacer una media ponderada de todas las distancias. Sin embargo, cabe la posibilidad de que el usuario no pulse el click del ratón en alguno de los intervalos de 4 segundos en los que el círculo está parado. En esos casos se ha realizado una distancia aproximada contando con las distancias totales. Esta distancia aproximada viene definida a partir del producto de la media de la suma de las distancias, de un contador con el número de veces que no hemos pulsado el ratón y de un factor elevado ya que esta distancia siempre tiene que ser más elevada que cualquier otra. Cuando tenemos la suma de todas las distancias se ha realizado una media total de todas las distancias, en el caso de haber pulsado el ratón en todos los intervalos de tiempo será la misma que la calculada.

La puntuación final será el inverso de la media total multiplicada por un factor elevado y se ha aplicado una transformación logarítmica para que haya valores de puntuaciones relativamente altos. Si no aplicamos esta transformación las puntuaciones obtenidas serían medio-bajas incluso si se realiza el experimento correctamente pulsando las 15 veces, y esos valores medio-bajos se convierten en altos. Estas puntuaciones van desde 1 hasta 100.

Excepciones

Hay que considerar algunas excepciones a la hora de realizar el experimento ya que hay casos en los que el experimento se ha realizado mal y no se pueden guardar esos datos en la base de datos. Estos casos son los siguientes.

- Se produce un error en el caso de que no se pulse ninguna vez el click izquierdo del ratón durante el experimento. Si no se pulsa ninguna vez quiere decir que no se ha obtenido ninguna coordenada del ratón y por tanto, tampoco ninguna imagen del sujeto en cuestión. En este caso se mostrará al usuario un mensaje de error y un link para volver a intentarlo. Este mensaje se puede ver en la figura 5.7



Figura 5.7: Alerta que salta al usuario en el caso de no haber pulsado ninguna vez el ratón en el experimento 1

- Se produce un error en el caso de que el usuario salga de pantalla completa ya que no estaríamos utilizando toda la pantalla del ordenador para realizar el experimento y no coincidirían las coordenadas con las imágenes. Por tanto, si se sale de pantalla completa también mostrará al usuario un mensaje de error.

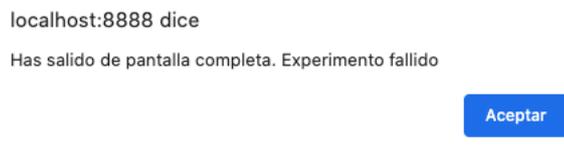


Figura 5.8: Alerta que salta al usuario en el caso de haber pulsado la tecla escape [ESC] durante el experimento.

En el caso de que aparezcan uno de los dos mensajes, aparecerá una pantalla de fallo para indicar que el experimento se ha abortado. Esta pantalla se puede ver en la sección del manual del usuario en la figura 6.19

5.2.4. Realización del experimento 2

En este apartado se van a exponer las directrices que se han seguido para el desarrollo del experimento 2. Hay bastantes aspectos comunes entre ambos experimentos y por tanto, no se van a explicar de nuevo. Las diferencias respecto al experimento 1 son: el movimiento del círculo es continuo y no estático, se guarda un vídeo de todo el experimento, no imágenes y las puntuaciones se calculan de manera diferente.

Generación de curvas

El experimento 2 consiste en el movimiento continuo del círculo sobre el canvas. El movimiento se ha generado en base a una trayectoria que debe seguir el círculo en función de diferentes curvas. En primer lugar, se desarrollaron dos curvas sencillas como son una circunferencia y una elipse, y después se desarrollaron otras dos curvas de Lissajous. Para cada curva se ha realizado una función y en cada función debe tener: el dibujo de la trayectoria que debe seguir el círculo, la parametrización de la curva y un temporizador para volver a llamar a la función cada cierto tiempo.

- Circunferencia y elipse. En primer lugar, se dibujó la trayectoria de la curva sobre el canvas con la ayuda de las funciones definidas *ctx.arc* y *ctx.ellipse*, donde *ctx* es la variable que contiene el canvas. Después, se realizaron las parametrizaciones de las curvas en función de las expresiones 5.5 y 5.6. [19].

$$\begin{cases} x = R \cdot \cos(t) \\ y = R \cdot \text{sen}(t) \end{cases} \quad t \in [0, 2\pi] \quad (5.5)$$

$$\begin{cases} x = \alpha + a \cdot \cos(t) \\ y = \beta + b \cdot \text{sen}(t) \end{cases} \quad t \in [0, 2\pi] \quad (5.6)$$

La expresión 5.5 hace referencia a la circunferencia y la *R* es el radio de la misma. Para nuestro experimento, este radio no puede ser mayor que la mitad del ancho de la pantalla ya que si no se saldría de ésta y habría momentos que el círculo desaparecería. Igual ocurre para la elipse, la expresión 5.6 corresponde a una elipse de centro (α, β) que no pueden ser mayores de la mitad del alto de la pantalla y semiejes *a, b*. El radio de la circunferencia y de la elipse se ha definido para que sea un valor aleatorio para que sea como máximo la mitad del alto de la pantalla y como mínimo un tercio de la mitad del alto de la pantalla. En las siguientes figuras 5.10(a) y 5.10(b) se pueden ver las dos curvas que se han mencionado: la circunferencia y la elipse.

- Curvas de Lissajous. Estas curvas se empezaron a investigar en 1815 y más tarde por Jules Antoine Lissajou en 1857. Se describen mediante las siguientes ecuaciones paramétricas 5.7.

$$\begin{cases} x = A \cdot \text{sen}(at + \delta) \\ y = B \cdot \text{sen}(bt) \end{cases} \quad (5.7)$$

Son formas creadas cuando las coordenadas *x* e *y* están descritas por ondas sinusoidales. Al ajustar la frecuencia de cada onda, la fase y sus amplitudes

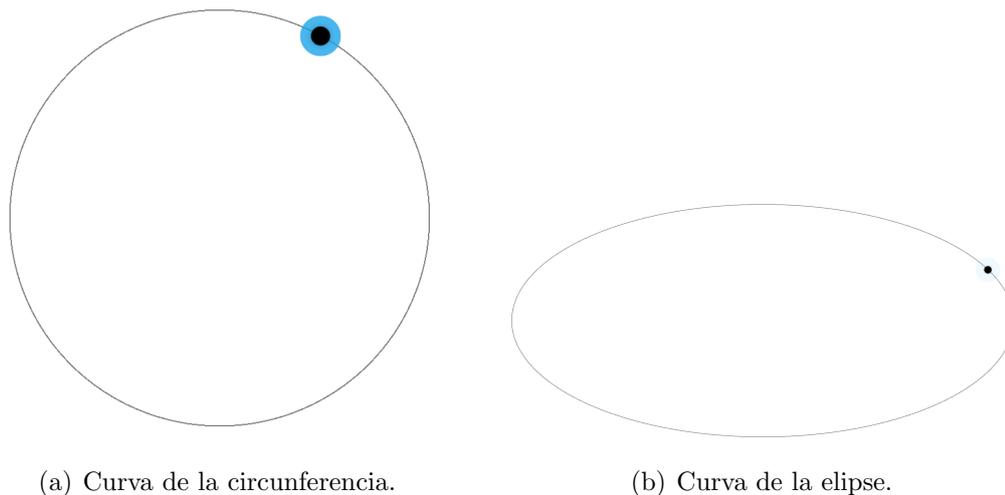


Figura 5.9: Capturas de las curvas de la circunferencia y de la elipse del experimento 2.

relativas, surgen patrones interesantes. En función de la relación entre las frecuencias a y b se generan formas cerradas y pueden aparecer diferentes lóbulos en el plano vertical y horizontal. En nuestro caso hemos realizado dos curvas de Lissajous, con relaciones 1:3 y 7:9 y se pueden ver en las figuras 5.10(a) y 5.10(b).

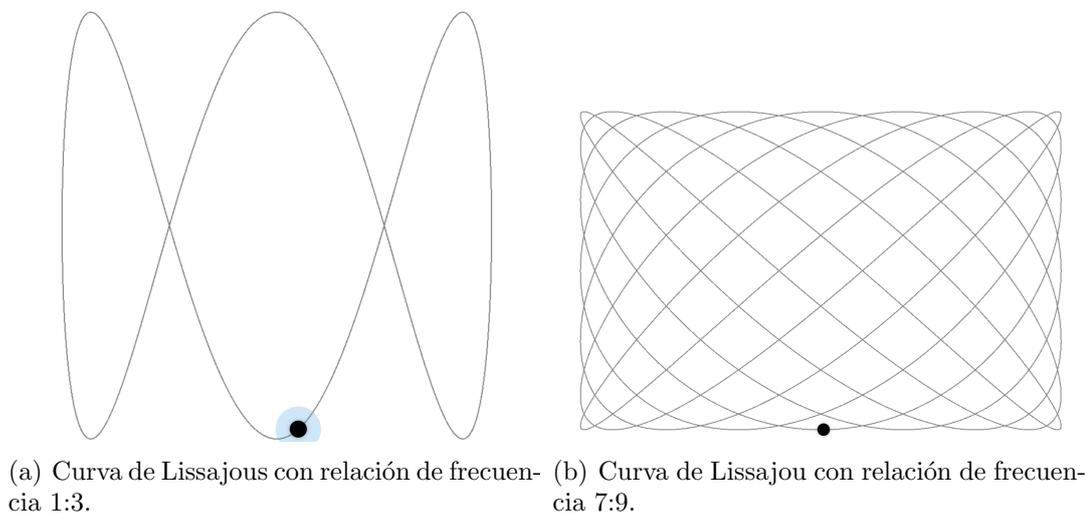


Figura 5.10: Capturas de las curvas de Lissajous generadas en el experimento 2.

Al igual que en las dos curvas anteriores, también se ha puesto un límite al radio de las curvas, el cual el máximo es la mitad del alto de la pantalla y el mínimo el tercio de éste aproximadamente.

Dado que se han construido 4 curvas diferentes, el usuario puede realizar el experimento de alguna de ellas de forma aleatoria. Es decir, antes de empezar el experimento, la aplicación decide aleatoriamente que curva se va a escoger.

Generación de posiciones del ratón y vídeo

Las posiciones del ratón ya no pueden ser realizando clicks sobre el círculo, ya que el movimiento del círculo ahora es continuo. Por tanto, lo que se ha hecho es recoger el valor de las coordenadas del ratón a través de un evento del mismo cada vez que cambia de posición el círculo. Así, tendremos unas coordenadas del ratón sólo cuando se mueva el círculo. El círculo tiene un período de cambio cada 33.33 ms, ya que así es como si se generasen 30 imágenes por segundo que es el valor típico de la velocidad de captura de una cámara web.

En unas primeras pruebas, se pensó en no realizar una captura de vídeo y guardar imágenes durante los 30 segundos que duraba el experimento, pero no tuvo éxito. Era complicado generar tantas imágenes en tan poco tiempo sin tener algún fallo temporal. Dado que se tenían que generar 30 imágenes por segundo, al final del experimento debíamos tener 900 imágenes con las respectivas posiciones, además de los otros parámetros. El principal problema fue que se tardaba mucho en procesar la información y en la base de datos no se guardaban todas las entradas correctamente.

Por tanto, se pensó en grabar un vídeo y ver si coincidía en el tiempo de manera razonable las posiciones del círculo y las posiciones de la mirada del ojo en el vídeo. Esta será una de las pruebas que se han realizado y que se describen más adelante con más detalle. El vídeo se guardará en un directorio denominado vídeos y en la base de datos se guardará el nombre del video en cuestión. Al igual que las imágenes el nombre del vídeo viene definido por el instante temporal en el que se inicia el experimento y el código de usuario, así se asegura que no se tiene un vídeo con nombres iguales.

Generación de puntuaciones

En este caso las puntuaciones no se calculan como en el primer experimento. El ratón siempre va a tener una posición dentro de la pantalla no como en el experimento 1 que había que realizar un click, entonces no tiene sentido que se realicen estimaciones sobre distancias en el caso de que no se obtenga la posición. Cada vez que se genera una posición del ratón se verifica si esa posición se encuentra cerca o lejos, en función de la distancia entre ambos, de la posición del círculo. Si la distancia entre ellas es mayor al valor impuesto se contabiliza como posición incorrecta y si es menor como posición correcta.

Cuando finaliza el experimento se dividen las posiciones correctas entre las posiciones totales y se multiplica por 100 para así obtener una puntuación sobre 100. La expresión utilizada es la que se ve en la ecuación 5.8.

$$puntuacion = (correcto/total) * 100 \quad (5.8)$$

5.2.5. Mostrar puntuaciones

Una vez que el experimento ha finalizado debe aparecer una lista con las puntuaciones tanto personales, como globales del experimento en cuestión. Para cada tipo de experimento habrá una lista independiente, el experimento tiene una clasificación y el experimento 2 tiene otra diferente. Para mostrar las puntuaciones se

realiza una petición a la base de datos con el tipo de experimento. Nos devolverá esas puntuaciones por orden de mayor a menor. En el caso del ranking personal también se indicará el código de usuario para filtrar en la base de datos y obtener sólo las puntuaciones de ese usuario.

Una vez que tenemos todos los datos se plasma una lista con las 10 mejores puntuaciones en la página en cuestión y aparece en negrita la puntuación del último experimento realizado independientemente del lugar en el que se encuentre. Quiere decir que si mi puntuación del experimento realizado no está entre las 10 mejores también saldrá en esa lista en el lugar que ocupe. Así, estamos dando a entender al usuario cuanto tiene que mejorar para llegar a mejores posiciones. En el ranking global también se indicará con la palabra *Usted* al lado de las puntuaciones si estas pertenecen al usuario que ha realizado el experimento.

El sistema también clasifica las puntuaciones entre *Mala*, *Regular* y *Buena* en función de unos umbrales. Si la puntuación es menor de 60 se considera que el experimento no se ha realizado del todo bien y que es probable que se pueda mejorar con facilidad; si la puntuación está entre 60 y 85 se considera que la puntuación es aceptable pero se puede mejorar y ser más preciso en el seguimiento; y por último, si la puntuación es mayor de 85 se considera que el experimento se ha realizado correctamente y el usuario debe seguir así en sus siguientes intentos.

En la siguiente figura 5.11 se muestra un ejemplo con las puntuaciones de unos de los ejemplos realizados en el experimento 1. En esta figura se muestran ambos rankings, tanto el ranking personal como el global, siempre dentro del mismo tipo de experimento, en este caso es un ranking del experimento 1.

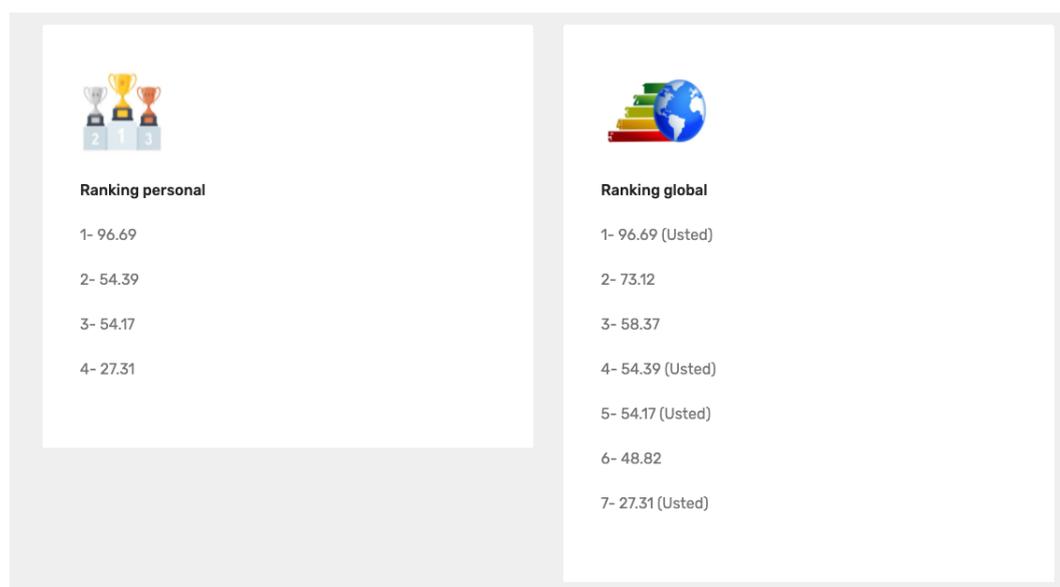


Figura 5.11: Puntuaciones personales y globales de uno de los experimentos realizados

5.3. Interfaz de administrador

La interfaz de administrador es la encargada de listar y descargar los datos de los experimentos. En este apartado se va a explicar cómo se han realizado estas funciones y el formato de los ficheros que el administrador va a generar.

Listar y descargar experimentos

El administrador tiene que poder descargar los datos de los experimentos que desee, por ello se pensó en realizar un panel de control donde se pudiesen elegir los experimentos en función su fecha de realización. Una vez introducida una fecha, el sistema realiza una petición a la base de datos y devuelve los códigos de dichos experimentos y los muestra en una tabla.

Cuando el administrador pulsa el botón de descargar un experimento, el sistema genera un fichero .zip con las imágenes/vídeo del experimento dependiendo si es el número 1 o 2, y los datos de la base de datos asociado ha dicho experimento en un fichero .json. La creación del fichero .zip se creó a partir del objeto *ZipArchive()*, en el que se añadió un directorio donde almacenaremos las imágenes/vídeo. Cuando tenemos creado el directorio adjuntamos estas imágenes/vídeos con la función *zip-files()*. [17]

Además se adjunta el fichero .json a partir de los datos almacenados en la base de datos. La estructura de este fichero debe ser fácil de interpretar para el administrador ya que posteriormente tendrá que realizar su análisis. El formato del fichero .json es el mostrado en la figura 5.12. [8].

5.4. Pruebas

En este apartado se han realizado varias pruebas en el servidor local del ordenador personal, que es el utilizado durante todo el proyecto.

5.4.1. Pruebas realizadas en el servidor local

Las pruebas que se han realizado en el servidor local, son pruebas de funcionamiento de la aplicación que son las que se van a exponer a continuación.

Captura de imágenes en el experimento 1

Uno de los principales objetivos de la aplicación es el de capturar imágenes de buena calidad, por ello se ha estado probando muchas formas de capturar las imágenes hasta que se ha llegado a las imágenes que se muestran en la figura 5.13. En la figura 5.13(a) se puede ver como la posición del círculo se encuentra en un lado de la pantalla, ya que los ojos apuntan hacia allí, sin embargo, en la figura 5.13(b) se puede ver como el usuario mira al frente porque el círculo se encuentra en la parte central de la pantalla. Las imágenes necesitaban tener un mínimo de calidad para que se apreciase donde estaba mirando el usuario, ya que si no se podrían analizar los datos posteriormente.

Prueba del temporizador del experimento 2

El experimento 2 consiste en la captura de vídeo y de posiciones del ratón. El experimento dura 30 segundos y se capturan 30 posiciones del ratón por segundo. Por tanto, es difícil ver si cada posición generada coincide en el punto exacto en el que estamos grabando el vídeo. Esto se puede ver a partir de la posición de los ojos en el vídeo del usuario y la posición que ocupa el círculo en un instante concreto. Es una prueba aproximada para ver si en realidad estamos cometiendo un error muy grande, ya que si no, no nos servirían estos datos capturados.

Esta prueba se ha realizado sobre la curva de la circunferencia ya que es más fácil detectar las posiciones. Lo que se ha realizado es una disminución de la variable dependiente para que el círculo diese saltos más grandes, y el periodo con el que se dibujaba el círculo también se ha disminuido a 1 segundo, por tanto, para esta prueba tendremos 30 posiciones, una cada segundo. Dado que la duración es 30 segundos, se ha ajustado el valor de la variable dependiente a $\pi/15$, así aseguramos que el círculo debe terminar donde empezó en 0, ya que ha dado una vuelta completa a la circunferencia (2π).

Se pueden ver en las imágenes capturadas que coinciden aproximadamente la posición de la mirada hacia el círculo, figura 5.14(a), y la posición del círculo en el instante final, figura 5.14(b). Si no fuesen simultáneamente se vería como la posición de la mirada estaría en otra posición y no la que se muestra.

Prueba envío de correo

Al realizar el registro en la aplicación se manda un correo al usuario que se registra. Este correo contiene un mensaje con el código de usuario necesario para iniciar sesión en la aplicación. Se ha realizado una prueba al realizar un registro de un usuario y he rellenado el campo de correo electrónico con un correo personal y el mensaje recibido es el que se muestra en la figura 5.15. En ésta se puede ver que el código de correo que se envía es PYWL.

```

{
  "0": "Datos del experimento:",
  "Código": "YFDR",
  "Tipo de experimento": "1",
  "Código de experimento": "2021-8-10_13_38_37_1_YFDR",
  "Puntuación": "58.37",
  "Ancho pantalla cm": "28.40",
  "Alto pantalla cm": "17.80",
  "Ancho pantalla px": "1440",
  "Alto pantalla px": "900",
  "Fecha": "2021-08-10",
  "Tiempo": "13:38:37"
},
{
  "0": "Datos del usuario:",
  "Edad": "23",
  "Sexo": "Femenino",
  "Color de ojos": "Marrones",
  "Gafas": "No",
  "Lentillas": "No",
  "Posición": "Estándar"
},
{
  "0": "Datos de la cámara:",
  "Nombre de la cámara": "Cámara FaceTime HD (integrada)",
  "Id de la cámara": "Cámara FaceTime HD (integrada)_39",
  "Calidad": "3438",
  "Micrófono": "None",
  "Altavoz": "None",
  "Frame rate": "30FPS",
  "Modo de la imagen": "rgb",
  "Resolución ancho": "1280 puntos",
  "Resolución alto": "720 puntos",
  "Relación de aspecto": "1.78",
  "Velocidad en bits": "19.89 MB/s",
  "Cantidad de colores": "183630",
  "Ligereza": "45.29%",
  "Luminosidad": "47.29%",
  "Brillo": "45.62%",
  "Matiz": "33.00%",
  "Saturación": "20.35%"
},

```

```

"Posición x de la bolita": [
  "1195",
  "1359",
  "734",
  "491",
  "747",
  "1373",
  "1166",
  "1009",
  "922",
  "815",
  "807",
  "188",
  "1210",
  "1384",
  "914",
  "640"
],
"Posición y de la bolita": [
  "826",
  "286",
  "329",
  "119",
  "69",
  "757",
  "63",
  "391",
  "311",
  "199",
  "655",
  "365",
  "159",
  "305",
  "655",
  "17"
],
"Tiempo del click": [
  "00:01:200",
  "00:06:136",
  "00:09:549",
  "00:14:928",
  "",
  "00:21:889",
  "00:25:669"
],

```

(a) Datos del experimento, del usuario y de la cámara de un experimento en formato json. (b) Datos de la posición del círculo y del tiempo del click de un experimento en formato json.

```

"00:33:385",
"00:37:741",
"00:41:925",
"00:45:877",
"00:49:713",
"00:53:438",
"00:57:541",
"00:57:686"
],
"Click x de la bolita": [
  "413",
  "1356",
  "732",
  "490",
  "-1",
  "1372",
  "1170",
  "1088",
  "924",
  "814",
  "807",
  "189",
  "1210",
  "1383",
  "912",
  "638"
],
"Click y de la bolita": [
  "151",
  "286",
  "326",
  "119",
  "-1",
  "758",
  "62",
  "390",
  "310",
  "200",
  "655",
  "365",
  "159",
  "305",
  "656",
  "17"
],

```

(c) Datos de la posición del ratón en el momento del click de un experimento en formato json.

Figura 5.12: Fichero con los datos del experimento 2021-8-10_13_38_37_1_YFDR con formato json.



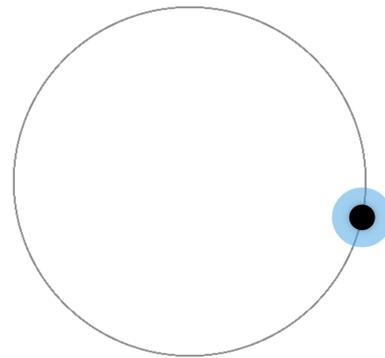
(a) Imagen generada en la primera posición del círculo.

(b) Imagen generada en la segunda posición del círculo.

Figura 5.13: Imágenes capturadas en una de las pruebas realizadas del experimento 1



(a) Imagen generada a partir del vídeo de uno de los experimentos de tipo 2 realizado



(b) Posición del círculo en el instante final de uno de los experimentos de tipo 2 realizado.

Figura 5.14: Imágenes que representan las pruebas que se han realizado respecto a la simultaneidad de la posición y el vídeo respecto al experimento de tipo 2.

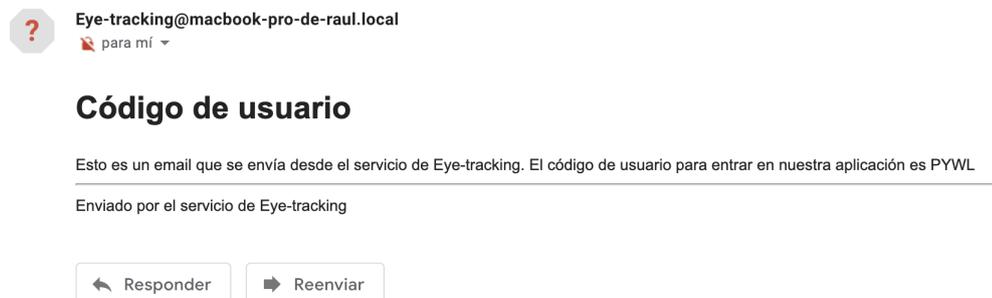


Figura 5.15: Captura del mensaje recibido por parte de la aplicación.

Capítulo 6

Manuales

6.1. Manual de usuario de la aplicación

En este manual de usuario se dará una explicación de cada una de las partes de la página web. Será una guía para los usuarios sin experiencia en la web.

6.1.1. Ventana principal

En esta ventana principal podemos tener dos casos posibles: el usuario está identificado o el usuario no está identificado.

- Usuario no identificado. El usuario no está identificado en la aplicación no tiene la posibilidad de realizar los experimentos. En este caso, se puede apreciar en la figura 6.1, que la barra de navegación tiene dos botones en la parte superior derecha, uno para registrarse y otro para iniciar sesión en la aplicación. El botón de comenzar llevará al usuario a la página de inicio de sesión ya que no está identificado.



Figura 6.1: Ventana principal del usuario de la aplicación.

- **Usuario identificado.** El usuario está identificado y tiene la posibilidad de realizar algún experimento. En la barra de navegación se suprimirán los botones de registrarse y de iniciar sesión y aparecerán uno de experimentos, otro de información y otro de cerrar sesión como se puede ver en la figura 6.2. Además, si en este caso se pulsa el botón de comenzar de la parte centrar no nos llevará a la página de experimentos ya que hemos iniciado sesión anteriormente.



Figura 6.2: Barra de navegación de un usuario que ya se ha identificado en la página.

En la parte inferior de la ventana principal tenemos una parte común para ambos usuarios, tanto si han sido identificados como si no. En esta parte encontramos dos secciones: en qué consiste la aplicación y quiénes son los creadores de la aplicación, como se puede observar en la figura 6.3. Además, en la figura 6.4 se puede ver la última parte de la ventana principal con las instituciones que han participado en el proyecto, y un pie de página con botones que nos llevan a la página de contacto y a la página donde se describen las políticas de privacidad.

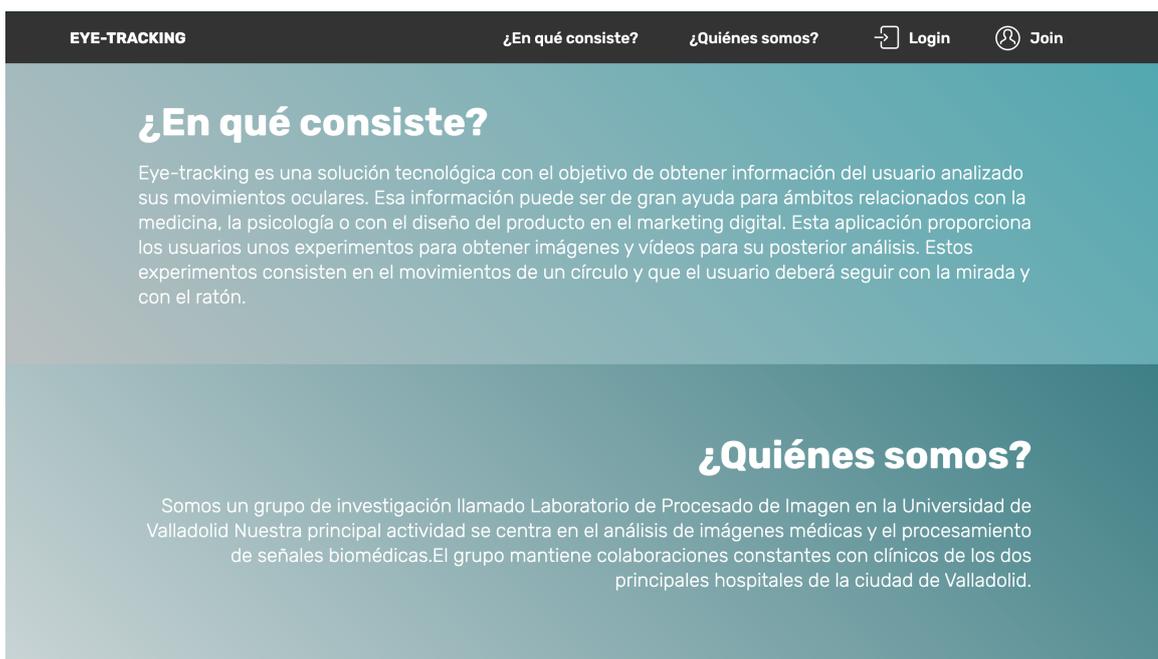


Figura 6.3: Ventana principal del usuario de la aplicación en la que se describe quiénes somos y en qué consiste.

6.1.2. Registro, modificación de usuario e inicio de sesión

Registro

El registro debe llevarse a cabo antes de la realización de los experimentos y constará de 4 partes.

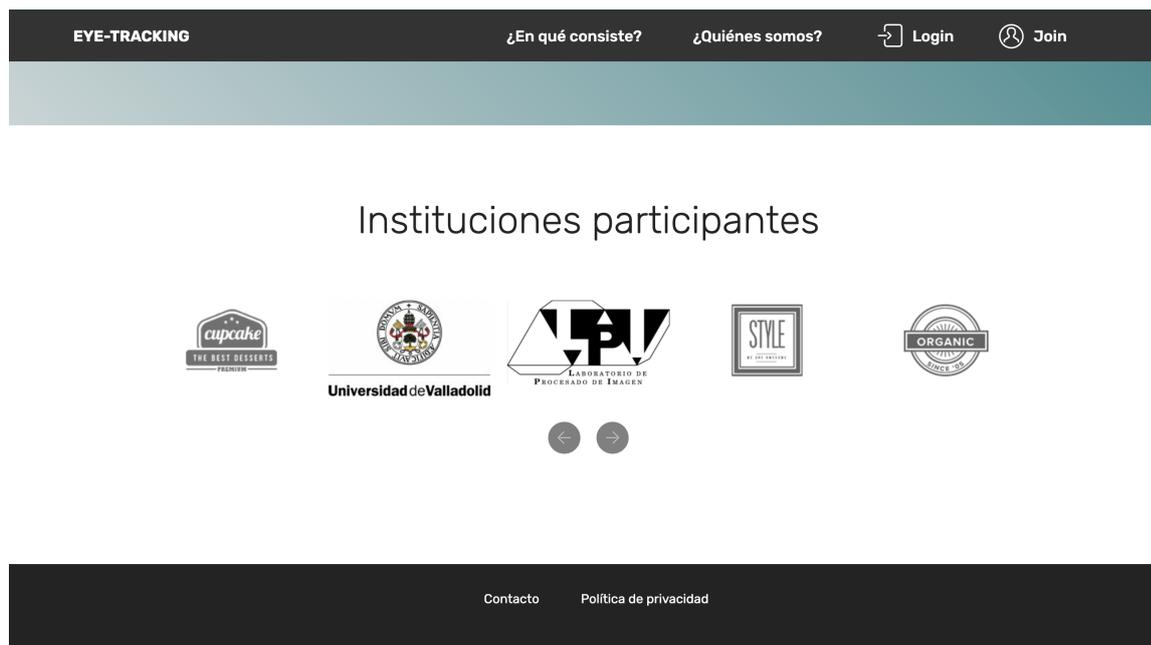


Figura 6.4: Ventana principal del usuario de la aplicación en la que se muestran las instituciones que han ayudado en el proyecto.

- Cuenta de usuario. Son los parámetros que se deben introducir para iniciar sesión en la aplicación siempre que se desee. Como se puede ver en la figura 6.5 consta de un código de usuario, una contraseña y un correo electrónico. El código usuario es generado por el sistema y será enviado al usuario registrado vía correo electrónico.

The screenshot shows a registration form titled 'Cuenta de usuario'. It contains three input fields: 'Código de usuario:' with the value 'IMMZ', 'Contraseña:' with the placeholder 'Password', and 'Correo electrónico:' with the placeholder 'Email'. Below the 'Correo electrónico:' field, there is a note: 'El correo electrónico sólo se utilizará para mandar el código del usuario, no se almacenará en la base de datos'.

Figura 6.5: Sección del registro donde el usuario debe introducir la información de la cuenta de usuario.

- Información personal. En esta sección se recaudará información sobre la edad, el sexo, el color de ojos y si el usuario va a realizar los experimentos con gafas o con lentillas y se puede observar en la figura 6.6. Esta información solo se va a utilizar para fines científicos.



Figura 6.6: Sección del registro donde el usuario debe introducir la información personal.

- Dimensiones de la pantalla. Esta parte del registro es de las más importante para nuestro proyecto, ya que se necesita saber las dimensiones de la pantalla para poder analizar los datos que vamos a generar. Por tanto, se debe introducir las dimensiones de la pantalla en pulgadas en el campo que se indica y pulsa el botón de *Aplicar*, en el caso de que sepamos ese dato. Si el usuario que se va a registrar no sabe ese dato, deberá indicarlo en el campo *No sé cual es la dimensión de mi pantalla*.

En cualquiera de los dos casos aparecerá una tarjeta dibujada en la pantalla representando un carné de la biblioteca de la Universidad de Valladolid que se puede ver en la figura 6.7. Esta tarjeta sirve para comprobar si hemos introducido bien el valor de nuestra pantalla en pulgadas o para saberlo en caso de que no lo hayamos introducido. Debemos tener a mano una tarjeta de una biblioteca, de identidad o similar y ponerla sobre la tarjeta de la pantalla y ver si las dimensiones coinciden. Si no coinciden debemos pulsar los botones de *Aumentar tamaño* o *Disminuir tamaño* hasta que coincida. Cuando los tamaños coincidan no debemos hacer nada más en esta parte del registro y pasar a la siguiente.

- Posición de la webcam. El usuario debe introducir la posición e la que se encuentra la webcam de su ordenador, ya que esa posición importa a la hora de analizar los datos obtenidos. Esta sección se puede ver en la figura 6.8

Las secciones de cumplimentar las dimensiones de la pantalla y la posición de la webcam no se podrán completar si no se está accediendo a través de

Dimensiones de la pantalla del usuario

Si sabes cuál es la dimensión en pulgadas de tu pantalla, introduce ese valor en el recuadro posterior, si no es así, marca la casilla que indica que no sabes las dimensiones de la pantalla.

Aplicar »

No sé cual es la dimensión de mi pantalla en pulgadas.

Puedes comprobar que el tamaño de tu pantalla es correcto con una tarjeta de la biblioteca, de crédito, de identidad o similar. Coloca la tarjeta sobre la pantalla y comprueba que los tamaños de ambas tarjetas coinciden. Por defecto sale la medida de la tarjeta a escala 1, pulsa los botones de **augmentar tamaño** y **disminuir tamaño** hasta que coincida con su tarjeta física.



TARJETA DE BIBLIOTECA

Universidad de Valladolid

Aumentar tamaño Reducir tamaño

La diagonal de tu pantalla es: 13.2" (33.5 cm)

Figura 6.7: Sección del registro donde el usuario debe introducir las dimensiones de la pantalla.

un ordenador portátil o de sobremesa. En el caso de que se acceda desde una móvil o tablet, se indicará al usuario con un mensaje de alerta como el que aparece en la figura 6.9. En este caso podrá terminar el registro pero sin estos valores. Cuando se acceda desde un ordenador, cumplimentará la parte del registro que falta antes de realizar los experimentos.

Además, presenta una última sección en la que el usuario deberá leer detenidamente el consentimiento y aceptación de los términos que se exponen. Una vez leídos, tendrá que pulsar el botón de aceptación de los términos y de verificación de edad. Una vez terminado el registro se deberá pulsar el botón de *Crear usuario* como el que aparece en la figura 6.10.

Modificación de usuario

El usuario tiene la capacidad de poder cambiar información de su cuenta, si así lo considera. Para ello, debe estar identificado en la aplicación web. Como se puede ver en la figura 6.2 hay un botón de información que es el que nos da la información de los parámetros del usuario. Al pulsar ese botón nos mostrará una página como la que se presenta en la figura 6.11. Se muestran los valores que se han introducido en el registro a excepción de la contraseña que es un valor que no se puede enseñar y el correo electrónico que es un campo que no se almacena en la base de datos y que sólo se usa para enviar un correo al usuario. En esta ventana el usuario puede editar el perfil o borrar los datos y la cuenta del sistema. Si decide editar el perfil



Figura 6.8: Sección del registro donde el usuario debe introducir la posición de la webcam.

Los experimentos de esta web necesitan ser realizados desde un ordenador y no desde una tablet. Puedes continuar con el registro, pero no podrás realizar los experimentos. Si quieres, puedes completar el registro y acceder más tarde desde un ordenador.

Aceptar

Figura 6.9: Mensaje de alerta informando al usuario que no puede realizar los experimentos con una tablet o móvil

podrá modificar todos los campos que aparecen en la figura 6.11 menos el código de usuario ya que es un parámetro único y que no se puede modificar.

Inicio de sesión

El usuario debe introducir el código que se ha generado en el registro y que se ha enviado a su correo electrónico; y la contraseña que se introdujo entonces. Si no estás registrado en el momento que vas a iniciar sesión, puedes entrar en el registro pulsando el link que aparece debajo de los campos de inicio de sesión. Se puede ver la interfaz de inicio de sesión en la figura 6.12.

Consentimiento y aceptación de los términos:

Lorem ipsum dolor sit amet consectetur adipisicing elit. Iste reprehenderit commodi perferendis, modi ducimus, deleniti, porro soluta culpa quasi facere repudiandae fugiat cum tenetur neque fuga natus? Pariatur, eum repellat! Lorem ipsum dolor sit amet consectetur adipisicing elit. Iste reprehenderit commodi perferendis, modi ducimus, deleniti, porro soluta culpa quasi facere repudiandae fugiat cum tenetur neque fuga natus? Pariatur, eum repellat! Lorem ipsum dolor sit amet consectetur adipisicing elit. Iste reprehenderit commodi perferendis, modi ducimus, deleniti, porro soluta culpa quasi facere

Tienes 18 años o más y aceptas las condiciones y términos para la realización de los experimentos

Crear usuario

Figura 6.10: Sección del registro donde el usuario debe leer el consentimiento y aceptación de los términos de la aplicación.

EYE-TRACKING
¿En qué consiste?
¿Quiénes somos?
Información
Experimentos
Cerrar sesión

Información de usuario

Código:	IEGX
Edad:	34
Color de ojos:	Verdes
Gafas:	Si
Lentillas:	No
Tamaño de pantalla en pulgadas:	13.6
Ancho de la pantalla en cm:	29.20
Alto de la pantalla en cm:	18.30
Posición de la webcam:	Estándar

Editar perfil
Borrar datos y mi cuenta

Contacto
Política de privacidad

Figura 6.11: Ventana para mostrar la información al usuario

6.1.3. Realización de experimentos

Ventana principal de experimentos

En esta ventana se indica al usuario identificado los dos tipos de experimentos que se pueden realizar. Disponen de una breve explicación del experimento debajo de la imagen como se puede ver en la figura 6.13. El usuario escogerá uno de los dos pulsando el botón de *Empezar*.

Explicación de experimentos

Después de elegir uno de los dos experimentos se mostrará al usuario la información de los experimentos con las instrucciones necesarias para que el experimento se realice correctamente. Se muestra en la figura 6.14 una captura realizada del primer experimento. Para el segundo experimento es similar pero cambiando la información

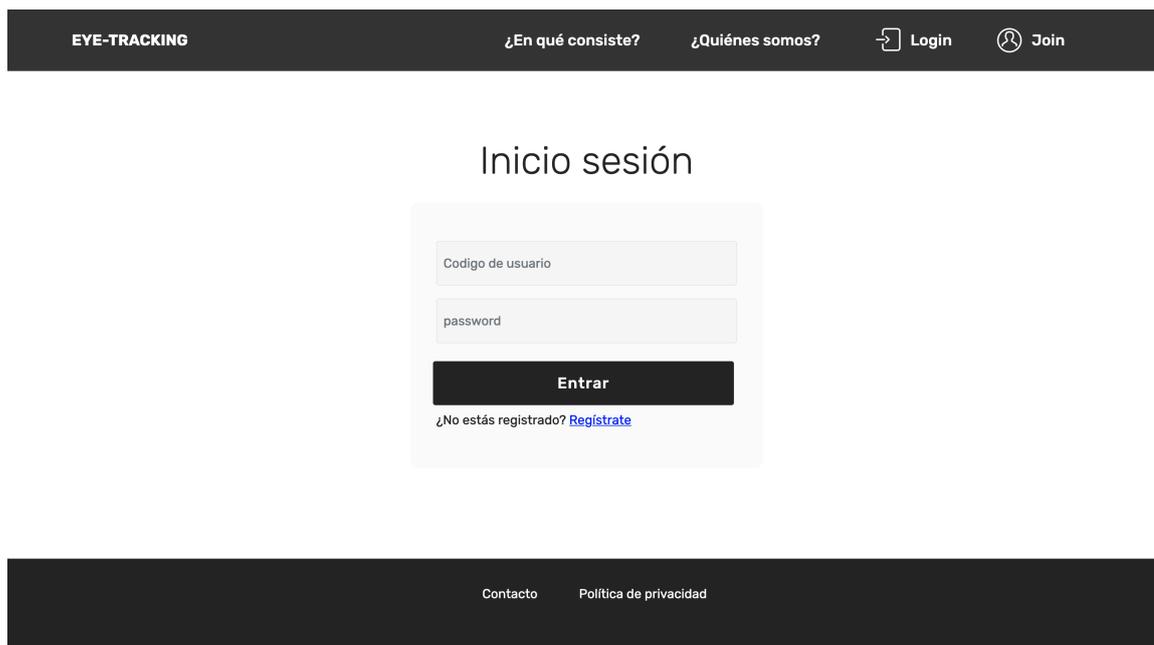


Figura 6.12: Inicio de sesión de los usuarios de la aplicación

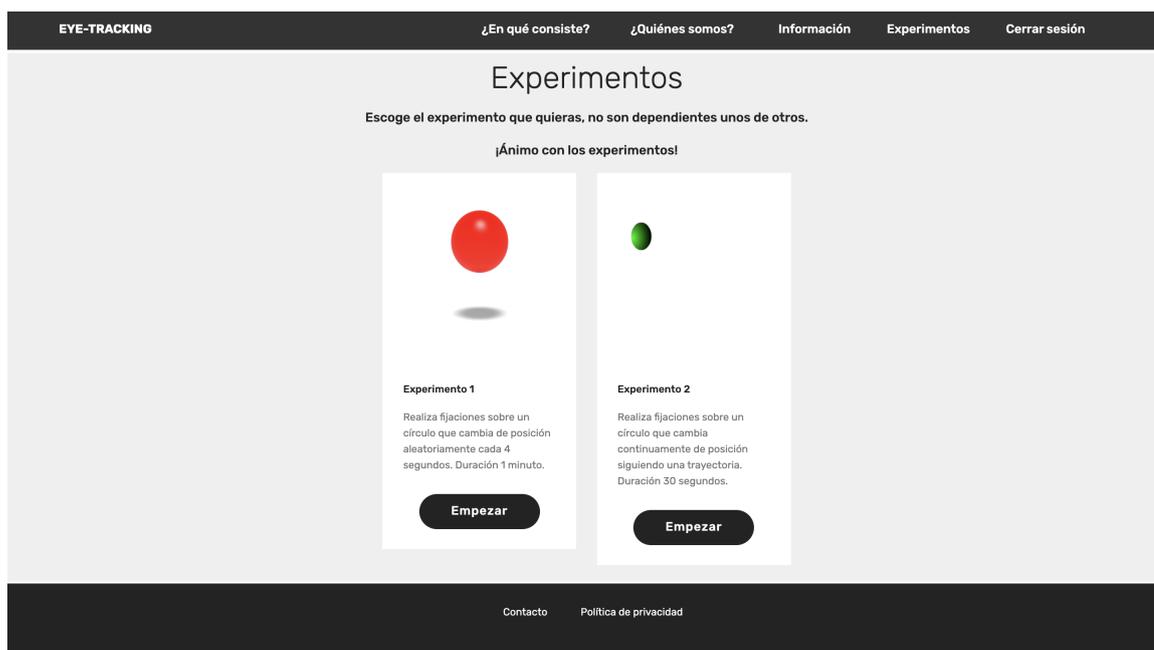


Figura 6.13: Ventana principal de experimentos

y las instrucciones del experimento. Cuando hayamos leído todas las instrucciones debemos pulsar el botón de *Comenzar*.

Probar cámara web

Antes de iniciar el experimento debemos hacer una prueba de la cámara web que se va a usar en el experimento. En la figura 6.15 se muestra la interfaz de esta ventana, en la que aparecerán a la izquierda los parámetros recogidos por el sistema, que serán datos que se almacenarán en la base de datos y en la parte central



Figura 6.14: Ventana en la que se muestra la información y las instrucciones del experimento 1.

se mostrará al usuario el vídeo que está recogiendo en la cámara. Este vídeo no se va almacenar en la base de datos. Tardará unos 10 segundos en obtenerse los datos. Aparece una secuencia de números en % hasta llegar a 100%. Una vez que haya terminado se podrán actualizar los datos de la cámara, si el usuario considera que debe de hacer otra prueba, o guardar esa información en la base de datos y pasar a la pantalla del experimento. Esta ventana ha sido una modificación de una página web de tests de webcams ya existente. [15]

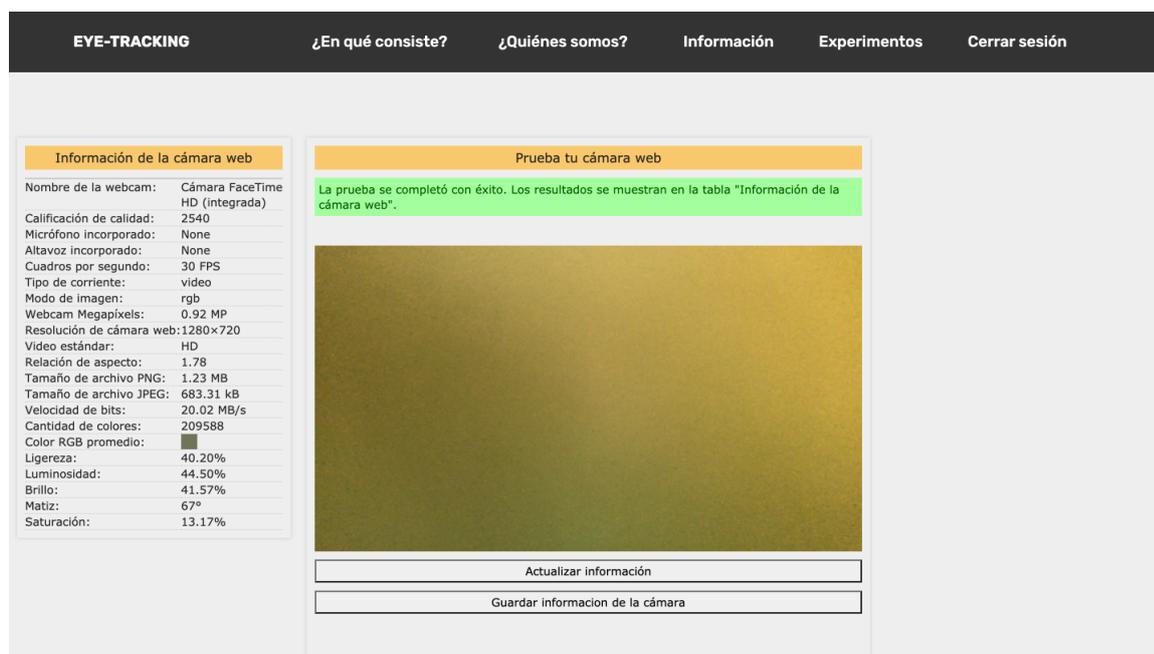


Figura 6.15: Ventana en la que el usuario puede probar la cámara web del ordenador.

Experimento 1

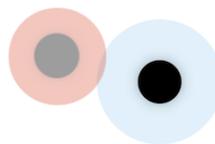
En la figura 6.16(a) se puede ver la pantalla de inicio del experimento, una vez que le demos al botón de *Empezar* iniciaremos el experimento. Este experimento 1 se basa en la aparición y desaparición de un círculo negro con una aura azul que disminuye y aumenta su tamaño como aparece en la figura 6.16(b). El objetivo del usuario es seguir el círculo con la mirada y realizar una pulsación con el click izquierdo del ratón cuando se esté mirando al círculo y el puntero del ratón esté encima del mismo. Al realizar esa pulsación, aparecerá en la pantalla un círculo igual que el otro pero con la aura roja, como el que se puede apreciar en la figura 6.16(c). El experimento tendrá una duración de 1 minuto.



(a) Inicio del experimento 1



(b) Posición aleatoria del círculo en la pantalla



(c) Posición aleatoria del círculo en la pantalla con pulsación por parte del usuario

Figura 6.16: Ejemplo de la realización experimento 1

Experimento 2

En la figura 6.16(a) se puede ver la pantalla de inicio del experimento, una vez que le demos al botón de *Empezar* iniciaremos el experimento, este botón es igual que el del experimento 1. El experimento 2 se basa en el movimiento continuo de un círculo negro que sigue una trayectoria. El objetivo del usuario es seguir el círculo con el puntero del ratón. en este caso no tendrá que pulsar ningún botón del ratón. En el ejemplo expuesto en la figura 6.17 se puede ver una elipse, pero como hemos dicho en el anterior capítulo hay muchas más. El experimento tendrá una duración de 60 segundos.

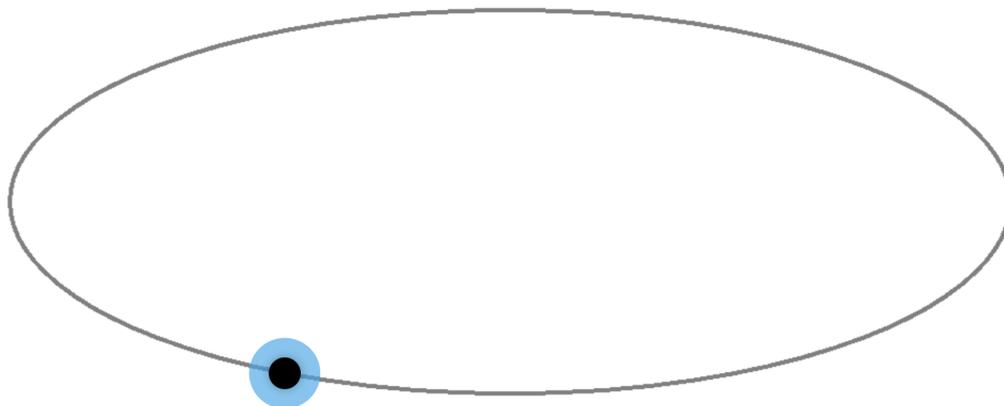


Figura 6.17: Representación del círculo siguiendo una trayectoria de una elipse sobre la pantalla durante el experimento 2.

6.1.4. Ventana de puntuaciones y errores

Cuando se termina el experimento podemos tener dos opciones: el experimento se ha realizado correctamente y por tanto se mostrará una pantalla de fin de experimento con la puntuación asociada, o una pantalla error si no se ha seguido algunas de las instrucciones que se han marcado al inicio del experimento.

Ventana de fin de experimento

En el caso de que el experimento se haya completado correctamente, se mostrará una ventana de fin de experimentos como la que se indica en la figura 6.18. En ella podemos ver que hay dos rankings de puntuaciones, un con el ranking personal, es decir, las puntuaciones obtenidas en ese experimento por ese usuario y un ranking global, con las puntuaciones obtenidas con todos los usuarios de ese experimento. Este ranking es independiente de si es el experimento 1 o 2. Puede volver al catálogo de experimentos para elegir otro experimento o puede volver a intentar ese mismo experimento.

Ventana de error de experimento

Si el usuario no ha cumplido con las instrucciones expuestas antes de la realización del experimento se mostrará una página de error de experimento como la que se indica en la figura 6.19. Puede volver al catálogo de experimentos para elegir otro experimento o puede volver a intentar ese mismo experimento.

6.2. Manual de administrador de la aplicación

En este apartado se van a dar las explicaciones necesarias para que el administrador sepa como tiene que actuar cuando interactúe con la aplicación. La principal

EYE-TRACKING ¿En qué consiste? ¿Quiénes somos? Información Experimentos Cerrar sesión

La realización del experimento ha sido Mala. Seguro que puedes hacerlo mejor. [Volver a intentarlo](#)

Ranking de puntuaciones del experimento

Ranking personal	Ranking global
1- 73.12	1- 96.69
2- 58.37	2- 73.12 (Usted)
3- 48.82	3- 58.37
	4- 54.17
	5- 48.82 (Usted)
	6- 27.31

[Volver a intentarlo](#) [Volver al catálogo de experimentos](#)

Figura 6.18: Ventana para mostrar al usuario las puntuaciones globales y personales del experimento.

EYE-TRACKING ¿En qué consiste? ¿Quiénes somos? Información Experimentos Cerrar sesión

 Ha habido un error en la realización del experimento

[Volver a intentarlo](#) [Volver al catálogo de experimentos](#)

[Contacto](#) [Política de privacidad](#)

Figura 6.19: Ventana para mostrar al usuario que ha habido un error en la realización del experimento.

función que tiene el administrador de la aplicación es descargar los datos de los experimentos y para ello necesita iniciar sesión, listar los experimentos y descargarlos. La función de iniciar sesión no se va a exponer, ya que es la misma que la de un usuario normal.

6.2.1. Listar experimentos

Una vez que el administrador ha iniciado sesión en la aplicación, le aparecerá una ventana de panel de control para realizar las búsquedas de los experimentos realizados. El panel de control, visualizado en a figura 6.20, presenta dos buscadores:

- Buscador por fecha. El administrador debe introducir una fecha en el formato adecuado y escoger el experimento del que quiera elegir los datos. Esto se debe a que en el experimento 1 se almacenan imágenes y en el 2 vídeos, así el administrador puede guardar los experimentos independientemente del tipo que sean.
- Buscador por intervalo de fechas. El administrador debe introducir dos fechas. La primera de ellas siempre más antigua que la segunda, si no aparecerá un error. Además, deberá escoger experimento como en el caso anterior.

En cualquiera de los dos casos las fechas deben ser introducidas en el formato correcto. Este formato debe ser AAAA-MM-DD, donde AAAA representa el año, MM el mes, y DD el día de la realización del experimento. Los dos buscadores son independientes entre sí, y deben introducirse los parámetros correspondientes al buscador o mostrará un error. Cuando se haya terminado de introducir los valores se pulsará el botón de *Buscar* de buscador correspondiente.

Panel de control

Buscar por:

- Fecha: retorna los experimentos que se hayan realizado en esa fecha

Fecha

Elige el experimento del que quieras descargar los datos: Experimento 1 Experimento 2

- Intervalo de fechas: retorna los experimentos que se hayan realizado entre las dos fechas

Primera fecha:

Segunda fecha:

Elige el experimento del que quieras descargar los datos: Experimento 1 Experimento 2

Figura 6.20: Ventana para mostrar al administrador el panel de control de buscador de experimentos.

Una vez que se ha pasado a la siguiente página aparecerá una lista con el código de los experimentos que coincidan con la fecha buscada. Por ejemplo, en la figura 6.21 se puede observar una lista de los experimentos realizados del experimento 1 con fecha 2021-08-10.

6.2.2. Descargar experimentos

El administrador dispone de una lista de experimentos y puede descargar los datos de un experimento concreto o puede descargar todos los experimentos de una

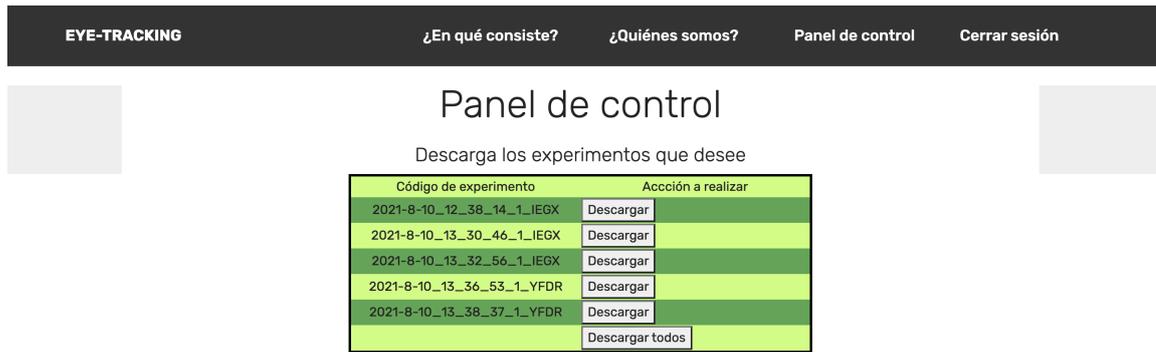


Figura 6.21: Ventana para mostrar al administrador la lista del experimento 1 con fecha 2021-8-10.

fecha. Se van a exponer ambas situaciones y que se obtendría de cada una de ellas.

- Descargar un experimento concreto. El administrador pulsa el botón *Descargar* de uno de los experimentos expuestos y se descargará un fichero .zip cuyo nombre es el código de de experimento, por ejemplo 2021-8-10-10_13_38_37_1_YFDR.zip. Una vez que hemos descomprimido el fichero obtendremos un fichero .json con la información detallada de dicho experimento y dependiendo si es el experimento 1, un directorio con las fotos correspondientes y si es el experimento 2, un directorio con el vídeo, como aparece en las figura 6.22.

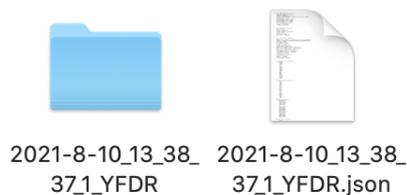


Figura 6.22: Descarga del experimento 2021-8-10-10_13_38_37_1_YFDR, en el que aparece un fichero .json y un directorio con las imágenes capturadas

- Descargar todos los experimentos de una fecha concreta. El administrador pulsa el botón de *Descargar todos* y se descargará un fichero con el nombre de la fecha en la que se está buscando, por ejemplo 2021-08-10.zip. Dentro de ese directorio habrá varios directorios con los experimentos individuales de la fecha asignada como se muestra en la figura 6.23. Esos 5 experimentos son los correspondientes al experimento 1 con fecha 2021-08-10. Dentro de cada una de los directorios estarán los datos correspondientes a los experimentos individuales explicados anteriormente.

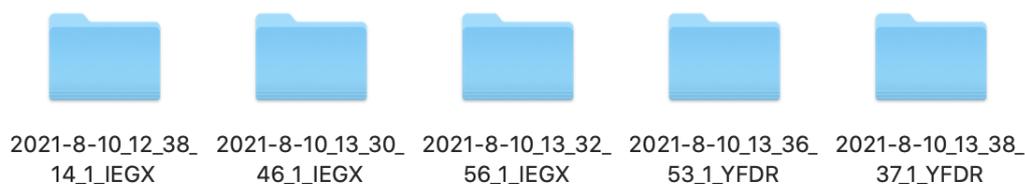


Figura 6.23: Descarga de los experimentos con fecha 2021-08-10

El análisis de los datos por parte del administrador será una tarea posterior que no se ha implementado en el proyecto.

6.3. Manual de mantenimiento de la aplicación

La aplicación se ha montado en un servidor de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid. Para ello se ha tenido acceso remoto a una máquina virtual para poder desplegar el proyecto. En este apartado se va a realizar una explicación de los pasos que se han seguido para su despliegue y como poder actuar en caso de se tenga que realizar cambios en la aplicación.

Los primeros pasos fueron conectarnos a la máquina virtual a través de ssh, para ello se utilizaba la siguiente sentencia en la línea de comandos.

```
ssh -Y -p 467 raul@cairo.lpi.tel.uva.es
```

Para realizar todas las instalaciones y configuraciones en la máquina virtual se necesita permisos de root o administrador. Hay que tener en cuenta que la máquina virtual presenta un sistema operativo de linux, así que es importante saber comandos de linux para poder entender y realizar estos pasos correctamente. Una vez que hemos entrado en la máquina y entrado en la sesión del administrador debemos realizar varias cosas: instalar apache, mysql y php; crear un host virtual para el sitio web; y probar la conexión con la base de datos desde php [20]. A continuación se va a detallar como se han realizado cada uno de los pasos hasta llegar a desplegar la página web.

6.3.1. Instalar apache, mysql y php

En primer lugar se instaló el servidor web de apache ya que es un de los servidores más populares del mundo. Para realizar esta instalación se ha utilizado el comando:

```
sudo apt install apache2
```

En segundo lugar se realizó la instalación de mysql que será el sistema de gestión de base de datos para poder almacenar y gestionar lo datos del sitio web. El comando utilizado para esta instalación es el que aparece a continuación:

```
sudo apt install mysql-server
```

Una vez que se ha instalado se ha ejecutado una secuencia de comandos de seguridad que viene preinstalada en mysql. Con este comando se eliminan algunos parámetros predeterminados y bloqueará el acceso a su sistema de base de datos.

```
sudo mysql_secure_installation
```

A partir de aquí se le harán una serie de preguntas para validar la contraseña que desea imponer para cada vez que se desee acceder a la base de datos como root. Una vez que se ha realizado esta validación podrá acceder a mysql a través del comando:

```
mysql
```

y podrá salir con el comando

```
exit
```

Lo último que queda por instalar es php, el componente de la configuración que procesará el código para mostrar el contenido al usuario final. En este caso se necesitan tres paquetes php, php-mysql para poder conectarse con la base de datos y libapache2-mod-php que habilita a apache para gestionar ficheros php. El comando utilizado para instalar los paquetes es el siguiente:

```
sudo apt install php libapache2-mod-php php-mysql
```

Una vez instalado podrá confirmar su versión de php con el comando

```
php -v
```

6.3.2. Crear un host virtual para el sitio web

A partir del servidor web de apache se pueden crear host virtuales para más de un dominio en un único servidor. En este manual se va a configurar el dominio cairo.lpi.tel.uva.es. La máquina virtual de linux nos proporciona una configuración en la que encontramos ficheros html en el directorio /var/www/html. En ese directorio es donde deberíamos añadir nuestros ficheros de la página web, sin embargo, éste se dejará como directorio predeterminado en el caso de que no coincida con la petición a nuestro sitio web y se creó otro directorio con nuestro dominio, para ello utilizamos el siguiente comando:

```
sudo mkdir /var/www/cairo.lpi.tel.uva.es
```

Después se cambiaron los permisos del directorio:

```
sudo chmod 755 /var/www/your_domain
```

Después se creará un nuevo fichero en el directorio sites-available de apache utilizando el editor de línea de comandos que prefiera, en nuestro caso hemos utilizado nano:

```
sudo nano /etc/apache2/sites-available/cairo.lpi.tel.uva.es.conf
```

Así estaremos creando un fichero de configuración para ese dominio del que podemos ver su contenido en la figura 6.24 Con la configuración de este fichero estamos configurando el host virtual para que apache proporcione cairo.lpi.tel.uva.es usando /var/www/cairo.lpi.tel.uva.es como directorio root. Para habilitar el nuevo host virtual debe introducir el siguiente comando

```

<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    ServerName cairo.lpi.tel.uva.es
    ServerAlias www.cairo.lpi.tel.uva.es
    DocumentRoot /var/www/cairo.lpi.tel.uva.es
    DirectoryIndex index.php
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

```

Figura 6.24: Fichero de configuración de apache

```
sudo a2ensite cairo.lpi.tel.uva.es
```

Es recomendable deshabilitar el sitio web predeterminado que viene instalado en apache, ya que en algunos caso puede que se sobrescriba en nuestro host virtual. Para deshabilitar el sitio web se utilizó el comando:

```
sudo a2dissite 000-default
```

Para verificar que hemos escrito bien el fichero de configuración del host virtual y que se carguen los cambios que hemos realizado deberemos utilizar la siguiente secuencia de comandos.

```
sudo apache2ctl configtest
sudo systemctl reload apache2
```

El sitio web ya se ha activado, pero el directorio root web está vacío, para comprobar que funciona correctamente se creó un fichero *index.html* con código html para verificar el funcionamiento del sitio web. Una vez creado el fichero se accedió desde un navegador a partir del nombre de dominio o de la dirección ip del servidor:

```
http://servidor_de_dominio_o_IP
```

Una vez que se ha probado ya se puede añadir todos los ficheros de la página web creada al directorio */var/www/cairo.lpi.tel.uva.es*.

6.3.3. Probar la conexión con la base de datos desde php

Se va a realizar una prueba para establecer la conexión con mysql y ejecutar consultas a la base de datos. Primero debemos crear la base de datos ya que ahora se tiene que poder guardar datos en el nuevo servidor, no como en las pruebas que se realizaban en la máquina personal que teníamos un servidor local. Para ello iniciamos mysql y creamos la base de datos con el comando:

```
CREATE DATABASE eye_tracking;
```

Una vez creada la base de datos se puede crear un usuario nuevo y dar privilegios a este nuevo usuario. Esto hace que la base de datos sea más segura ya que tiene un método de autenticación predeterminado.

```
CREATE USER 'example_user'@'%' IDENTIFIED WITH mysql_native_password BY 'password';
```

Para darle permiso a este usuario a la base de datos:

```
GRANT ALL ON eye_tracking.* TO 'example_user'@'%';
```

Se puede verificar si el usuario nuevo tiene los permisos adecuados al iniciar sesión de nuevo en mysql con las credenciales del usuario:

```
mysql -u example_user -p
```

A continuación crearemos las tablas de la base de datos, para ello, lo que se ha realizado es generar todas las tablas en un fichero y después agregarlas a la consola de mysql, ya que modificar parámetros en la consola es más complicado que hacerlo sobre un editor de texto. Las tablas creadas se pueden ver en la siguiente figura 6.25. Además, dado que la aplicación tiene un administrador que ya viene prefijado se ha añadido en la base de datos con el código de usuario ROOT y el campo de privilegio a 1. Como podemos ver en la sentencia de la figura 6.26 se inserta en la tabla de usuarios.

```

create table usuarios(
codigo char(4) not null primary key,
pass char(50) not null,
edad char(3) not null,
sexo char(20) not null,
color_ojos char(15) not null,
gafas char(3) not null,
lentillas char(3) not null,
diagonal_pulgadas float,
ancho_cm float(8,2),
alto_cm float(8,2),
posicion char(25),
privilegio int not null
);

create table experimentos(
codigo char(4) not null,
cod_exp int not null,
experimento char(50) not null primary key,
puntuacion float(6,2),
ancho_cm float (4,2),
alto_cm float (4,2),
ancho_px int,
alto_px int,
fecha date,
tiempo time,
id_camara char(140)
);

create table posicion_tiempo(
id_posicion_tiempo int AUTO_INCREMENT not null primary key,
experimento char(50) not null,
tiempo_contador int,
tiempo_click char(40),
click_x int,
click_y int,
pos_x int,
pos_y int,
imagen varchar(240),
video varchar(240)
);

create table info_camara(
id_camara char(140) primary key,
nombre_camara char(100),
calidad int,
micro char(5),
altavoz char(5),
frame_rate int,
modo_imagen char(7),
mega float (3,2),
resolucion char(20),
resolucion_ancho int,
resolucion_alto int,
relacion_aspecto float(4,2),
velocidad_bits float(4,2),
cant_colores int,
ligereza float(5,2),
luminosidad float(5,2),
brillo float(5,2),
matiz float(5,2),
saturacion float(5,2)
);

```

(a) Tablas de posicion_tiempo e info_camara de la base de datos. (b) Tablas de usuarios y experimentos de la base de datos.

Figura 6.25: Tablas de la base de datos.

```

insert into usuarios values('ROOT','dc76e9f0c0006e8f919e0c515c66dbba3982f785','23','Masculino','Marrones','No','No',0,0,0,'Estándar',1);

```

Figura 6.26: Sentencia para añadir al usuario privilegiado de la aplicación.

Capítulo 7

Conclusiones

En este capítulo se comentan las conclusiones obtenidas de este trabajo y las líneas futuras que se seguirán a partir de la aplicación creada.

7.1. Conclusiones

Como resultado de este trabajo, se ha implementado un sistema de captura de datos para la realización de eye-tracking sin hardware específico. La evaluación de la aplicación tenía principalmente los siguientes objetivos: crear una aplicación web amigable y escalable, capturar imágenes y vídeos de los usuarios, exportar todos los datos disponibles en la base de datos por parte del administrador para su posterior análisis y desplegar la aplicación a una gran cantidad de sujetos.

El primero de los objetivos es la implementación de la aplicación web, se puede ver en los diferentes manuales realizados, tanto del usuario como del administrador, que la aplicación es intuitiva y fácil de usar. Se pretende que llegue a gran cantidad de público para obtener el mayor número de datos posibles para su posterior análisis.

En cuanto al segundo objetivo queda comprobado que la captura de las imágenes y vídeos son de buena calidad como se presenta en el apartado de pruebas del capítulo 5. Las imágenes sobre todo presentan un buen enfoque al ojo que es lo que más nos interesa, ya que a la hora de realizar el análisis se valorará la posición y dirección a la que está mirando.

El tercero de los objetivos es el de descargar los datos capturado en la aplicación. Se puede observar que tanto en el apartado de pruebas del capítulo 5 como en el manual de administrador del capítulo 6, se ha realizado con éxito la descarga de los datos capturados durante las pruebas realizadas en la aplicación. Los ficheros siguen una estructura organizada para que no haya errores entre los experimentos.

A nivel más general, las principales conclusiones obtenidas de la realización de este trabajo son las siguientes. En primer lugar, uno de aspectos más importantes de este trabajo es la decisión de no utilizar un dispositivo de seguimiento ocular para la realización de los experimentos. Como ya se comentó, uno de las particularidades de realizar el seguimiento ocular era que los dispositivos utilizados eran bastante costosos y no todo el mundo podía obtener uno, esto hace que sólo se pueda reali-

zar esta técnica en ciertos lugares como laboratorios específicos. Con la aplicación creada sólo bastaría con adquirir un ordenador portátil o de sobremesa con cámara web. En segundo lugar, es importante destacar las características ya incorporadas a la aplicación para su posterior análisis, entre ellas, la información incluida en el formulario del usuario, la creación de un código propio de cada usuario, los diferentes directorios creados de cada experimento y que se podrían clasificar en función del usuario, etc.

7.2. Líneas futuras

A lo largo del proyecto hemos intentado demostrar las potenciales utilidades de esta tecnología, así como la diversidad de funcionalidades para la detección de enfermedades y la mejora de la calidad de vida de los afectados. En esta cuestión, nos avala la multitud de estudios al respecto en las últimas décadas, siempre al amparo de los incesantes avances tecnológicos que acompañan este proceso. En el caos concreto de esta investigación, existe una posible línea futura del proyecto, encargada de realizar el análisis de los datos capturados en los experimentos de la aplicación. Este análisis consistiría en realizar un aprendizaje automático de todas las imágenes para detectar la posición y dirección en la que miran los usuarios. Podría ser una implementación en un posible trabajo de fin de máster o tesis doctoral.

Bibliografía

- [1] Andrew T. Duchowski. *Eye Tracking Methodology*. Springer, Clemson, SC USA, 2017.
- [2] Tobii pro. URL <https://www.tobiipro.com/>.
- [3] Petr Kellnhofer Harini Kannan Suchendra Bhandarkar Wojciech Matusik Antonio Torralba University of Georgia Massachusetts Institute of Technology MPI Informatik Kyle Krafka, Aditya Khosla. *Eye tracking for everyone*. 2016.
- [4] DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y DE COMPUTADORAS UNIVERSIDAD NACIONAL DEL SUR. Evaluación cognitiva mediante eye tracking (ecet).
- [5] Oftalvist. URL <https://www.oftalvist.es/blog/globo-ocular-partes-ojo/>.
- [6] Macula-retina. URL <https://www.macula-retina.es/la-fovea/>.
- [7] Mario Fritz Andreas Bulling Perceptual User Interfaces Group Scalable Learning Xucong Zhang, Yusuke Sugano and Germany Perception Group Max Planck Institute for Informatics, Saarbrücken. *Appearance-based gaze estimation in the wild*. 2015.
- [8] Mdn web docs. URL <https://developer.mozilla.org/es/docs/Web/CSS>.
- [9] Tech terms. URL <https://techterms.com>.
- [10] Db engines. URL <https://db-engines.com/en/ranking>.
- [11] Javatpoint. URL <https://www.javatpoint.com/>.
- [12] w3schools. URL <https://www.w3schools.com/>.
- [13] infobyip. URL <https://es.infobyip.com/detectdisplaysize.php>.
- [14] Detección de móviles y tablets con php. URL <https://mobiforge.com/design-development/tablet-and-mobile-device-detection-php>.
- [15] webcamtests. URL <https://es.webcamtests.com/>.
- [16] codepen.io. URL <https://codepen.io/Yario/pen/bQZMBM>.
- [17] php.net. URL <https://www.php.net/>.

- [18] aprende-web.net. URL <https://aprende-web.net/jspracticass/tiempo/tiempo3.php>.
- [19] Álgebra y geometría analítica. URL <https://aga.frba.utn.edu.ar/ecuaciones-parametricas-de-las-conicas/>.
- [20] Instalación de mysql, apache y php. URL <https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-20-04-es>.