



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Maestría

Método para analizar el movimiento ocular de usuarios para
generar métricas y correlación con estados emocionales y
cognitivos

presentada por

Ing. Derick Axel Lagunes Ramírez

como requisito para la obtención del grado de
Maestro en Ciencias la Computación

Directores de tesis

Dr. Juan Gabriel González Serna

Dr. Máximo López Sánchez

Cuernavaca, Morelos, México. Julio de 2020.



“2020, Año de Leona Vicario, Benemérita Madre de la Patria”

Cuernavaca, Mor., 17/agosto/2020

OFICIO No. DCC/110/2020
Asunto: Aceptación de documento de tesis
CENIDET-AC-004-M14-OFICIO

C. DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del **C. Ing. Derick Axel Lagunes Ramírez**, con número de control M18CE078, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado **“Método para analizar el movimiento ocular del usuario para generar métricas y correlación con estados emocionales y cognitivos”** y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

Dr. Juan Gabriel González Serna
Doctor en Ciencias de la Computación
7820329
Director de tesis

Dr. Máximo López Sánchez
Doctor en Ciencias de la Computación
7498547
Co-director de tesis

Dr. Noé Alejandro Castro Sánchez
Doctor en Ciencias de la Computación
08701806
Revisor 1

Dr. Dante Mújica Vargas
Doctor en Comunicaciones y Electrónica
09131756
Revisor 2

C.c.p. Depto. Servicios Escolares
Expediente / Estudiante
JGGS/lmz



“2020, Año de Leona Vicario, Benemérita Madre de la Patria”

Cuernavaca, Morelos **18/agosto/2020**

OFICIO No. SAC/ 244/2020

Asunto: Autorización de impresión de tesis

DERICK AXEL LAGUNES RAMÍREZ
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
DE LA COMPUTACIÓN
PRESENTE

Por este conducto tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado “*Método para analizar el movimiento ocular del usuario para generar métricas y correlación con estados emocionales y cognitivos*”, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE

Excelencia en Educación Tecnológica®
“Conocimiento y tecnología al servicio de México”

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



**CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA**

C.c.p. M.E. Guadalupe Garrido Rivera. Jefa del Departamento de Servicios Escolares
Expediente
GVGR/CHG

DEDICATORIAS

A mis padres, Sonia y Derick, que siempre me han apoyado en el transcurso de mi vida. A mi hermano Alexis que por mucho tiempo me ha dado una razón para seguir adelante. A Nancy quien me ha dado todo, me ha apoyado y acompañado durante toda la maestría.

A todos mis amigos y compañeros que me han acompañado a lo largo de mi carrera, quienes enriquecen mi vida con opiniones y conocimientos en otras áreas. Y a todos mis profesores quienes han dejado huella en mí con los conocimientos que me han enseñado durante toda mi vida académica.

A la generación 2018-2020 del departamento de Ciencias de la Computación, de las diferentes líneas de investigación.

AGRADECIMIENTOS

Muchas gracias a todos los que me han apoyado en el transcurso del desarrollo de esta tesis.

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por su apoyo económico, sin éste no hubiera sido posible el desarrollo de esta tesis.

Agradezco al Tecnológico Nacional de México campus Centro Nacional de Investigación y Desarrollo Tecnológico (TecNM/CENIDET) y a todo su personal por darme la oportunidad de realizar mis estudios de maestría en Ciencias de la Computación.

Agradezco también a mi Director de Tesis el Dr. Juan Gabriel González Serna por el tiempo empleado para guiarme y apoyarme durante todo el periodo de mi estancia en el TecNM/CENIDET. A mi Codirector, el Dr. Máximo López Sánchez, y mis revisores, el Dr. Noé Alejandro Castro Sánchez y el Dr. Dante Mújica Vargas, quienes me compartieron su valioso tiempo para revisiones, correcciones, consejos y sugerencias.

Resumen

La experiencia de usuario (UX, por sus siglas en inglés) trata de tomar en cuenta las reacciones de los usuarios al interactuar con un producto o servicio, los datos recabados de los usuarios son utilizados después para determinar si este producto o servicio tuvo un impacto positivo en los usuarios finales. De este modo los responsables pueden tomar acciones sobre el diseño y mejorar la aceptación general del producto con el público final.

Las tecnologías de seguimiento ocular permiten entender varios comportamientos de los usuarios que interactúan con otros objetos, al estudiar y comprender estos patrones se pueden reconocer los estados mentales que influyeron en el usuario durante el uso de, por ejemplo, un producto digital.

Con el fin de hacer uso de estas técnicas de análisis de seguimiento ocular para ser aplicadas a un contexto de evaluación de la UX, se desarrolló la presente investigación. El trabajo de tesis incluye el desarrollo de algoritmos de extracción de datos de sensores de seguimiento ocular y generación de métricas oculares, durante la investigación se encontraron muy pocas herramientas de código abierto o acceso público que permitieran extraer datos oculares o generar métricas. Por lo que se ha desarrollado una solución que permite estructurar los datos de seguimiento ocular (puntos de visión, fijaciones y parpadeos) de cualquier sensor y generar videos y análisis de métricas oculares (mapas de calor, mapas de rutas, áreas de interés, videos de puntos, análisis del parpadeo).

Estos algoritmos fueron implementados en una plataforma para evaluar la UX de productos digitales llamada UXLab, la cual ha sido desarrollada con funcionalidades que permiten grabar y observar datos de sensores fisiológicos que aportan otro enfoque a analizar del comportamiento de los usuarios durante su interacción con el producto. Al integrar los algoritmos de seguimiento ocular a la suite de UXLab se pudieron correlacionar las métricas oculares con los demás datos fisiológicos del usuario creando así un algoritmo que permite correlacionar los movimientos oculares con los estados mentales de los usuarios, durante pruebas UX.

Los algoritmos desarrollados fueron probados e implementados en UXLab aportando valor agregado a las evaluaciones UX con las métricas oculares.

Abstract

User experience (UX) tries to take into account the reactions of users when interacting with a product or service, the data collected from the users helps to determine if this product or service had a positive impact in the end users group. In this way, those responsible for the product or service can act on the design and improve the general acceptance of the product with the final public.

Eye tracking technologies allow us to observe and understand the user's behavior when he interacts with other objects. By studying and understanding these patterns, the mental states that influenced the user during the use of, for example, a digital product can be recognized.

This investigation was developed to make use of eye tracking analysis techniques in a UX evaluation context. The thesis work includes the development of algorithms for data extraction from eye tracking sensors and generation of eye metrics. During the research, very few open source or public access tools were found that allowed eye data extraction or metric generation. So a solution has been developed that allows structuring the eye tracking data (points of view, fixations and blinks) of any sensor and generating videos and analysis of eye metrics (heat maps, route maps, areas of interest, videos dot analysis, flicker analysis).

These algorithms were implemented in a platform to evaluate the UX of digital products called UXLab, which has been developed with functionalities that allow recording and observing data from physiological sensors that provide another approach to analyze user behavior during their interaction with the product. By integrating eye tracking algorithms into the UXLab suite, eye metrics could be correlated with the user's other physiological data, thus creating an algorithm that allows eye movements to be correlated with users' mental states during UX tests.

The algorithms developed were tested and implemented in UXLab adding value to UX evaluations with eye metrics.

Contenido

CAPÍTULO 1. INTRODUCCIÓN.....	1
1.1 ANTECEDENTES.....	4
1.2 PLANTEAMIENTO DEL PROBLEMA	5
1.3 OBJETIVOS.....	6
1.3.1 <i>Objetivo general</i>	6
1.3.2 <i>Objetivos particulares</i>	6
1.4 JUSTIFICACIÓN.....	7
1.5 ALCANCES Y LIMITACIONES	8
1.5.1 <i>Alcances del proyecto</i>	8
1.5.2 <i>Limitaciones del proyecto</i>	8
1.6 ESTRUCTURA DEL DOCUMENTO	9
CAPÍTULO 2. MARCO TEÓRICO	10
2.1 EXPERIENCIA DE USUARIO	10
2.2 ESTADO MENTAL	10
2.3 ESTADOS COGNITIVOS	11
2.4 ESTADOS AFECTIVOS	11
2.5 SENSORES FISIOLÓGICOS	12
2.6 SEGUIMIENTO OCULAR.....	13
2.6.1 <i>Puntos de fijación y visión</i>	13
2.6.2 <i>Métricas de fijación y visión</i>	14
2.6.2.2 <i>Mapas de rutas</i>	16
2.6.2.3 <i>Áreas de interés</i>	18
2.6.2.4 <i>Parpadeos</i>	19
2.6.2.5 <i>Otras métricas del seguimiento ocular</i>	19
CAPÍTULO 3. ANÁLISIS	23
3.1 ANÁLISIS DEL ESTADO DEL ARTE	21
3.1.1 <i>Trabajos relacionados con la evaluación de la UX</i>	21
3.1.2 <i>Trabajos relacionados con los estados mentales</i>	22
3.1.3 <i>Discusión sobre trabajos del estado del arte</i>	23
3.1.4 <i>Tabla comparativa de técnicas analizadas del estado del arte</i>	23
3.2 ANÁLISIS DE LAS MÉTRICAS DE SEGUIMIENTO OCULAR EN LA EVALUACIÓN UX	25

3.2.1 Mapas de atención.....	25
3.2.2 Áreas de interés	26
3.2.3 Análisis del parpadeo	26
3.3 ANÁLISIS DE LA HERRAMIENTA PARA EVALUAR LA EXPERIENCIA DE USUARIO	
UXLAB	27
3.3.1 Sistema de grabación de UXLab	28
3.3.2 Sistema Reproductor de UXLab	31
CAPÍTULO 4. METODOLOGÍA DE SOLUCIÓN.....	34
4.1 FASE 1: EXTRACCIÓN DE CARACTERÍSTICAS DEL SENSOR	35
4.2 FASE 2: VALIDACIÓN Y GENERACIÓN DE MÉTRICAS	35
4.3 FASE 3: RELACIÓN ENTRE COMPORTAMIENTO VISUAL Y ESTADO COGNITIVO	36
4.4 FASE 4: IMPLEMENTACIÓN A UXLAB Y PRUEBAS.....	36
4.5 ALGORITMO DE EXTRACCIÓN DE DATOS DEL SENSOR DE SEGUIMIENTO OCULAR.....	37
4.5.1 Extracción de datos de visión.....	38
4.5.2 Extracción de datos de fijaciones	39
4.5.3 Algoritmo de extracción de parpadeos.....	41
4.5.3.1 Precisión del algoritmo de extracción de parpadeos	42
4.5.4 Herramienta de extracción de datos EyeWidget	43
4.6 ALGORITMOS DE GENERACIÓN DE MÉTRICAS DEL SEGUIMIENTO OCULAR.....	48
4.6.1 Video de puntos de visión	48
4.6.2 Mapas de calor	51
4.6.3 Mapas de rutas	55
4.6.4 Generación de áreas de interés dinámicas.....	60
4.6.5 Análisis del parpadeo	64
4.7 INTEGRACIÓN CON UXLAB	67
4.7.1 Funcionamiento de UXLab.....	69
4.7.2 Integración y mejoras del módulo de seguimiento ocular.....	72
4.7.2.1 Modificaciones en grabador UXLab	72
4.7.2.1 Reproductor UXLab	75
4.7.2.3 Mejoras generales a UXLab	76
4.8 MÉTODO DE CORRELACIÓN DE MÉTRICAS DEL SEGUIMIENTO OCULAR Y ESTADO	
MENTAL DEL USUARIO	78
CAPÍTULO 5. PRUEBAS Y RESULTADOS	80

5.1 PLAN DE PRUEBAS	82
5.1.1 <i>Pruebas funcionales de UXLab</i>	82
5.1.2 <i>Pruebas emocionales</i>	85
5.1.2.1 Contexto de pruebas	102
5.1.3 <i>Pruebas de estados cognitivos</i>	104
5.2 EJECUCIÓN DE LAS PRUEBAS Y REPORTE DE RESULTADOS	106
5.2.2 <i>Pruebas emocionales</i>	107
5.2.2.1 <i>Primera iteración</i>	108
5.2.2.2 <i>Segunda iteración</i>	117
5.2.2.1 Conclusiones de las pruebas emocionales	130
5.2.3 <i>Pruebas cognitivas</i>	131
5.2.3.1 Conclusiones de las pruebas cognitivas	137
CAPÍTULO 6. CONCLUSIONES Y TRABAJOS FUTUROS.....	141
6.1 APORTACIONES	135
6.1.1 <i>Integración a la herramienta UXLab</i>	135
6.1.2 <i>Reconocimiento facial</i>	136
6.1.3 <i>Generación de las métricas oculares</i>	136
6.1.3 <i>Publicaciones</i>	136
6.2 TRABAJOS FUTUROS	137
6.2.1 <i>Integración de pupilometría para medir estados mentales</i>	137
6.2.2 <i>Experimentos de correlación de parpadeos con estado cognitivo</i>	137
CAPÍTULO 7. REFERENCIAS	138
CAPÍTULO 8. ANEXOS	141

Lista de figuras

<i>FIGURA 1:</i> ARQUITECTURA DE UXLAB.....	5
<i>FIGURA 2:</i> CLASIFICACIÓN DE ESTADOS MENTALES	12
<i>FIGURA 3:</i> ESQUEMA DEL FUNCIONAMIENTO DEL SEGUIMIENTO OCULAR CON UN SENSOR INFRARROJO.....	13
<i>FIGURA 4:</i> ESQUEMA DE COORDENADAS DE VISIÓN EN UNA PANTALLA.	14
<i>FIGURA 5:</i> MAPA DE CALOR: REALIZADO CON LOS DATOS DE SEGUIMIENTO OCULAR DE UNA PRUEBA DE EVALUACIÓN UX.	15
<i>FIGURA 6:</i> MAPA DE RUTAS: REALIZADO CON LOS DATOS DE SEGUIMIENTO OCULAR DE UNA PRUEBA DE EVALUACIÓN UX.	17
<i>FIGURA 7:</i> ÁREAS DE INTERÉS: REALIZADO CON LOS DATOS DE SEGUIMIENTO OCULAR DE UNA PRUEBA DE EVALUACIÓN UX	18
<i>FIGURA 8:</i> REPRESENTACIÓN GRÁFICA DE UN PARPADEO.	19
<i>FIGURA 9:</i> ESQUEMA DE TIEMPO HACIA LA PRIMERA FIJACIÓN.	20
<i>FIGURA 10:</i> ESQUEMA DE RECUENTO DE VISIÓN.....	20
<i>FIGURA 11:</i> ESQUEMA DE TIEMPO GASTADO.....	21
<i>FIGURA 12:</i> REPRESENTACIÓN DE LA MEDICIÓN DEL TAMAÑO DE LA PUPILA.	22
<i>FIGURA 13:</i> ESQUEMA DE LA DISTANCIA ENTRE EL USUARIO Y LA PANTALLA.....	22
<i>FIGURA 14:</i> ARQUITECTURA DE UXLAB. EL TRABAJO SE CENTRA EN LAS CAPAS DE MÉTRICAS Y CAPA DE CAPTURA	27
<i>FIGURA 15:</i> PROCESO DE REGISTRO DE DATOS EN UXLAB.....	29
<i>FIGURA 16:</i> CONTROL DE SENSORES FISIOLÓGICOS DE UXLAB	29
<i>FIGURA 17:</i> MONITOREO EN TIEMPO REAL DE SENSORES EN UXLAB.....	30
<i>FIGURA 18:</i> GRABACIÓN DE DATOS DE SENSORES EN UXLAB.....	30
<i>FIGURA 19:</i> ARCHIVOS GENERADOS EN LA ETAPA DE GRABACIÓN DE UXLAB.	31
<i>FIGURA 20:</i> INTERFAZ GRÁFICA DEL REPRODUCTOR DE UXLAB.....	32
<i>FIGURA 21:</i> ELEMENTOS DE LA INTERFAZ DEL REPRODUCTOR UXLAB.....	33
<i>FIGURA 22:</i> ESQUEMA CONCEPTUAL DE LA METODOLOGÍA DE SOLUCIÓN.	35
<i>FIGURA 23:</i> PSEUDOCÓDIGO PARA OBTENCIÓN DE PUNTOS DE VISIÓN.....	38
<i>FIGURA 24:</i> EJEMPLO DE ARCHIVO DE PUNTOS DE VISIÓN	39
<i>FIGURA 25:</i> PSEUDOCÓDIGO PARA OBTENCIÓN DE PUNTOS DE FIJACIÓN.	40
<i>FIGURA 26:</i> EJEMPLO DE ARCHIVO DE PUNTOS DE FIJACIÓN.....	40
<i>FIGURA 27:</i> PSEUDOCÓDIGO PARA OBTENCIÓN DE PUNTOS DE FIJACIÓN.	41
<i>FIGURA 28:</i> EJEMPLO DE ARCHIVO DE PUNTOS DE PARPADEOS.	42

<i>FIGURA 29: VIDEOS CAPTURADOS DE CADA PARTICIPANTE.</i>	43
<i>FIGURA 30: EYEWIDGET ORIGINAL</i>	45
<i>FIGURA 31: EYEWIDGET MODIFICADO</i>	46
<i>FIGURA 32: ESQUEMA DEL PROCESO COMPUTACIONAL DE EYEWIDGET MODIFICADO PARA EXTRAER DATOS DE VISIÓN, FIJACIONES Y PARPADEO.</i>	48
<i>FIGURA 33: ESQUEMA DEL PROCESO DE GENERACIÓN DE VIDEOS DE PUNTOS.</i>	49
<i>FIGURA 34: PSEUDOCÓDIGO PARA LA GENERACIÓN DE VIDEOS DE PUNTOS.</i>	51
<i>FIGURA 35: EJEMPLO DE VIDEO DE PUNTOS PROCESADO SOBRE UNA PRUEBA DE SEGUIMIENTO OCULAR.</i>	51
<i>FIGURA 36: ESQUEMA DEL PROCESO DE GENERACIÓN DE VIDEOS DE MAPA DE CALOR.</i>	52
<i>FIGURA 37: PSEUDOCÓDIGO PARA LA GENERACIÓN DE MAPAS DE CALOR.</i>	54
<i>FIGURA 38: EJEMPLO DE MAPA DE CALOR PROCESADO SOBRE UNA PRUEBA DE SEGUIMIENTO OCULAR.</i>	54
<i>FIGURA 39: ESQUEMA CONCEPTUAL DE LAS MANCHAS DE COLORES EN LOS MAPAS DE CALOR.</i> 55	
<i>FIGURA 40: PSEUDOCÓDIGO PARA LA GENERACIÓN DE SÁCADAS A PARTIR DE FIJACIONES.</i>	56
<i>FIGURA 41: ESQUEMA DEL PROCESO DE GENERACIÓN DE VIDEOS DE MAPA DE RUTAS.</i>	57
<i>FIGURA 42: EJEMPLO DE MAPA DE RUTAS UTILIZANDO LA DURACIÓN DE LAS FIJACIONES PARA REPRESENTAR EL TAMAÑO DE LA FIJACIÓN</i>	58
<i>FIGURA 43: ESQUEMA DE MAPA DE RUTAS</i>	58
<i>FIGURA 44: EJEMPLO DE MAPA DE RUTAS PROCESADO SOBRE UNA PRUEBA DE SEGUIMIENTO OCULAR</i>	59
<i>FIGURA 45: PSEUDOCÓDIGO PARA LA GENERACIÓN DE MAPAS DE RUTAS</i>	59
<i>FIGURA 46: DEFINICIÓN DE 3 CLÚSTERES DADA LA CONCENTRACIÓN DE DATOS CON MEANSHIFT</i>	60
<i>FIGURA 47: ESQUEMA DEL PROCESO DEL ALGORITMO MEANSHIFT</i>	62
<i>FIGURA 48: PSEUDOCÓDIGO PARA LA GENERACIÓN DE ÁREAS DE INTERÉS DINÁMICAS.</i>	63
<i>FIGURA 49: EJEMPLO DE ÁREAS DE INTERÉS PROCESADAS SOBRE UNA PRUEBA DE SEGUIMIENTO OCULAR.</i>	64
<i>FIGURA 50: ESQUEMA DEL PROCESO DEL ALGORITMO DE ANÁLISIS DEL PARPADEO</i>	65
<i>FIGURA 51: EJEMPLO DE GRÁFICA DE ANÁLISIS DE PARPADEO RESULTANTE DE ARCHIVO DE FRECUENCIAS.</i>	65
<i>FIGURA 52: DIAGRAMA DE FLUJO DE INTERPRETACIÓN DE ESTADO COGNITIVO MEDIANTE GRÁFICA DE ANÁLISIS DEL PARPADEO.</i>	66

<i>FIGURA 53: PSEUDOCÓDIGO PARA LA GENERACIÓN DE ARCHIVO DE FRECUENCIA DE PARPADEO</i>	67
<i>FIGURA 54: ESQUEMA DE PROCESO GENERALIZADO DE UXLAB</i>	68
<i>FIGURA 55: ESQUEMA CONCEPTUAL DE FUNCIONAMIENTO DEL MÓDULO <i>OUTLET</i> DEL GRABADOR UXLAB</i>	69
<i>FIGURA 56: ESQUEMA CONCEPTUAL DE FUNCIONAMIENTO DEL MÓDULO <i>INLET</i> DEL GRABADOR UXLAB</i>	70
<i>FIGURA 57: ESQUEMA DE FUNCIONAMIENTO DE UXLAB CON <i>EYEWidget</i></i>	71
<i>FIGURA 58: ESQUEMA DE FUNCIONES EXTENDIDAS E IMPLEMENTADAS EN UXLAB, RELACIONADAS CON EL MÓDULO DE SEGUIMIENTO OCULAR</i>	72
<i>FIGURA 59: ESQUEMA DE FUNCIONAMIENTO DE LA COMUNICACIÓN POR SOCKET ENTRE UXLAB Y <i>EYEWidget</i></i>	73
<i>FIGURA 60: ESQUEMA DE PROCESO DE POSTPROCESAMIENTO DE DATOS OCULARES AL FINAL DE UNA GRABACIÓN EN EL GRABADOR DE UXLAB</i>	74
<i>FIGURA 61: ESQUEMA DE PROCESO DE VISUALIZACIÓN DE PRUEBAS UX EN EL REPRODUCTOR DE UXLAB</i>	75
<i>FIGURA 62: ESQUEMA DE PROCESO DE CAMBIO DE MÉTRICAS OCULARES EN EL REPRODUCTOR DE UXLAB</i>	76
<i>FIGURA 63: ESQUEMA DE PROCESO DE UXLAB</i>	77
<i>FIGURA 64: DIAGRAMA DE FLUJO DEL ALGORITMO DE CORRELACIÓN DE MÉTRICAS DEL SEGUIMIENTO OCULAR Y ESTADO MENTAL DEL USUARIO</i>	78
<i>FIGURA 65: ESQUEMA DE FUNCIONES DE SEGUIMIENTO OCULAR IMPLEMENTADAS EN UXLAB. LAS FUNCIONES RELACIONADAS CON EL MÓDULO DE SEGUIMIENTO OCULAR SE MUESTRAN DENTRO DE LA LÍNEA PUNTEADA</i>	83
<i>FIGURA 66: DIAGRAMA DEL PROCESO DE INDUCCIÓN EMOCIONAL <i>PIE</i></i>	88
<i>FIGURA 67: ESQUEMA DE LA ETAPA 1 DEL PROCESO <i>PIE</i> PARA LA INDUCCIÓN EMOCIONAL</i>	89
<i>FIGURA 68: ESQUEMA DE LA ETAPA 2 DEL PROCESO <i>PIE</i> PARA LA INDUCCIÓN EMOCIONAL</i>	90
<i>FIGURA 69: ESQUEMA DE PROCESO DE IMPLEMENTACIÓN DE INDUCCIÓN EMOCIONAL <i>PIE</i></i>	90
<i>FIGURA 70: ESQUEMA DE LA ETAPA 3 DEL PROCESO <i>PIE</i> PARA LA INDUCCIÓN EMOCIONAL</i>	93
<i>FIGURA 71: PROCESO DE PRUEBAS EMOCIONALES ADAPTADO DEL <i>PIE</i></i>	94
<i>FIGURA 72: ALGUNOS ESTÍMULOS DE BASE DE DATOS <i>IAPS</i></i>	95
<i>FIGURA 73: ALGUNOS ESTÍMULOS DE BASE DE DATOS <i>OASIS</i></i>	96
<i>FIGURA 74: VALORES DE VALENCIA Y ACTIVACIÓN PARA LOS ESTÍMULOS DE IMÁGENES DEL REPOSITORIO <i>OASIS</i></i>	97

<i>FIGURA 75: VALORES DE VALENCIA Y ACTIVACIÓN PARA LOS ESTÍMULOS A USAR EN LAS PRUEBAS.....</i>	<i>98</i>
<i>FIGURA 76: GRÁFICA DE RESPUESTAS A LAS PREGUNTAS DEL CUESTIONARIO PHQ-9.....</i>	<i>100</i>
<i>FIGURA 77: RESULTADOS DE LA PRUEBA PHQ-9 PARA LOS 10 PARTICIPANTES.....</i>	<i>101</i>
<i>FIGURA 78: GRÁFICA DE RESPUESTAS A LAS PREGUNTAS DEL CUESTIONARIO STAI</i>	<i>101</i>
<i>FIGURA 79: PANTALLA DE GRABADOR UXLAB MONITOREANDO USUARIO.</i>	<i>104</i>

Lista de tablas

<i>TABLA 1:</i> CLASIFICACIÓN DE LOS ESTADOS AFECTIVOS, DE SORIANO, 2018, TODOS LOS DERECHOS RESERVADOS 2018. TOMADO CON PERMISO DEL AUTOR.	12
<i>TABLA 2:</i> TABLA COMPARATIVA DE TÉCNICAS Y MÉTODOS DEL ESTADO DEL ARTE.....	24
<i>TABLA 3:</i> SENSORES FISIOLÓGICOS Y BIOMÉTRICOS SOPORTADOS POR UXLAB.....	28
<i>TABLA 4:</i> PRUEBAS DE EFICIENCIA DE EXTRACCIÓN DE PARPADEOS CON EYEWIDGET.	43
<i>TABLA 5:</i> PRUEBAS DE EFICIENCIA DE EXTRACCIÓN DE PARPADEOS CON EYEWIDGET.	60
<i>TABLA 6:</i> PLAN DE PRUEBAS UXLAB, MÓDULO DE SEGUIMIENTO OCULAR.	83
<i>TABLA 7:</i> EJEMPLO DE ESTÍMULOS DE UNA SOLA IMAGEN PARA LAS 5 EMOCIONES (ALEGRÍA, TRISTEZA, NEUTRA, RELAJACIÓN Y ANSIEDAD).....	86
<i>TABLA 8:</i> EJEMPLO DE ESTÍMULOS DE IMÁGENES MIXTAS PARA 4 EMOCIONES (ALEGRÍA, TRISTEZA, RELAJACIÓN Y ANSIEDAD).....	87
<i>TABLA 9:</i> ADAPTACIÓN DE LA ETAPA 2 DEL PROCESO PIE PARA ENTORNOS NO INMERSIVOS.	91
<i>TABLA 10:</i> PLAN DE PRUEBAS EMOCIONALES UXLAB, MÓDULO DE SEGUIMIENTO OCULAR.....	94
<i>TABLA 11:</i> IMÁGENES DE IAPS SELECCIONADAS PARA LA INDUCCIÓN.	95
<i>TABLA 12:</i> IMÁGENES DE OASIS SELECCIONADAS PARA LA INDUCCIÓN.	97
<i>TABLA 13:</i> USUARIOS PARA PRUEBAS EMOCIONALES.....	99
<i>TABLA 14:</i> PLAN DE PRUEBAS COGNITIVAS UXLAB, MÓDULO DE SEGUIMIENTO OCULAR.....	105
<i>TABLA 15:</i> REPORTE DE PRIMERA ITERACIÓN DE PRUEBAS EMOCIONALES UXLAB.	108
<i>TABLA 16:</i> REPORTE DE SEGUNDA ITERACIÓN DE PRUEBAS EMOCIONALES UXLAB.	117
<i>TABLA 17:</i> REPORTE DE PRUEBAS COGNITIVAS UXLAB.....	131
<i>TABLA 18:</i> RESULTADOS DE PRUEBA T PARA EL NÚMERO DE PARPADEOS CAPTURADOS EN LOS NIVELES DE CARGA COGNITIVA BAJA Y ALTA.....	138
<i>TABLA 19:</i> RESULTADOS DE PRUEBA T PARA LA FRECUENCIA DE PARPADEO DE LOS USUARIOS EN LOS NIVELES BAJO Y ALTO.	139
<i>TABLA 20:</i> CONTRASTE DE FRECUENCIA DE PARPADEOS EN ESTE TRABAJO Y OTROS REPORTADOS EN LA LITERATURA.....	139

Capítulo 1. Introducción

En el pasado era complicado realizar estudios que analizaran el movimiento ocular de las personas, los dispositivos que se utilizaban eran primitivos, con grandes cámaras de video que necesitan apuntar a los ojos del usuario que debía mantener la cabeza inmóvil con la ayuda de un reposacabezas. El reciente avance digital ha mejorado la precisión y la usabilidad de los sensores de seguimiento ocular (Ho, 2014) y hoy en día la tecnología se ha simplificado considerablemente para estar al alcance de todos, lo cual ha permitido que las investigaciones con esta tecnología aumenten de manera significativa.

El seguimiento ocular (eye-tracking) analiza el movimiento de los globos oculares de un usuario mientras se encuentra observado un estímulo visual (una imagen, un video, una aplicación móvil, una página web o un objeto convencional). Observar los puntos de visión, las fijaciones, las sácadas (desplazamientos rápidos de entre 30 a 120 milisegundos de los ojos entre dos puntos de fijación) y los parpadeos, proporciona información sobre el comportamiento, la atención y el estado cognitivo de usuarios. Los estudios del seguimiento ocular se realizan en varios campos científicos, como la psicología, la medicina, la evaluación de la experiencia de usuario, entre otros. Las señales de movimiento ocular se han utilizado ampliamente en la investigación de la Interacción Humano Computadora (HCI por sus siglas en inglés) para el análisis y la evaluación de la usabilidad, ya que pueden proporcionar una forma natural y eficiente de observar el comportamiento de los usuarios (Lu, et al., 2015).

La experiencia de usuario (UX por sus siglas en inglés) es un área de investigación que se centra en la evaluación subjetiva de la experiencia del usuario. El proceso permite identificar el conjunto de experiencias que tiene el usuario final al utilizar un producto digital, estos estudios consideran las percepciones y respuestas de las personas que utilizan un producto, sistema o servicio, la experiencia del usuario incluye todas las emociones, las creencias, las preferencias, las percepciones, las respuestas físicas, las respuestas psicológicas, los comportamientos y logros que ocurren antes, durante y después de su uso. Por lo tanto, la UX es de naturaleza subjetiva, dependiente del contexto y está ligada al beneficio potencial que se obtiene del producto, sistema o servicio (Hussain, et al., 2018).

Los diseñadores y expertos de UX tratan de que los usuarios finales tengan una percepción positiva de los productos que utilizan ya que, una mala experiencia con un

producto digital puede provocar la pérdida de usuarios finales. Por lo tanto, la UX ayuda a generar productos finales de alta calidad que se perciben positivamente por el público objetivo.

Históricamente, la evaluación de la experiencia de usuario ha utilizado instrumentos de recolección de información convencionales, tales como cuestionarios o entrevistas después de que cada participante interactúa con un producto, sistema o servicio. En ese contexto, los datos que se obtienen se ven fácilmente afectados por el entorno, la dificultad de las comparaciones entre productos, emociones falsas y opiniones subjetivas. Resulta difícil precisar la retroalimentación de los usuarios, debido a que las respuestas dependen de la voluntad y de la habilidad del usuario para poder expresar cómo se sintió cuando fue expuesto al producto, sistema o servicio (Guo, Ding, Liu, Liu, y Zhang, 2016), además, la información de estos instrumentos es relativamente simple y carente de evidencia de datos objetivos (Qu, Zhang, Chao, y Duffy, 2016), puede ser fácilmente manipulada y la interpretación es relativa para cada persona.

Actualmente los investigadores de la UX utilizan mediciones fisiológicas para evaluar la experiencia de usuario con métricas cuantitativas (Hussain, et al., 2018). Estos nuevos procesos de UX utilizan modelos y patrones de datos recolectados mediante sensores fisiológicos y biométricos, con los cuales se pueden identificar los estados cognitivos y estados emocionales de los participantes durante el proceso UX de un producto, sistema o servicio. De este modo se pueden analizar datos biométricos y fisiológicos reales sobre la impresión que dejan los productos digitales en los participantes. Los datos de sensores son más útiles cuando se analizan en combinación con otros ya que al correlacionar las lecturas de dos o más sensores se puede validar un estado emocional o cognitivo de forma más precisa.

Los datos biométricos incluyen:

- Análisis facial
- Seguimiento ocular
- Análisis de voz (Volumen, cadencia y tono)
- Análisis de postura corporal y de cabeza

Los datos fisiológicos incluyen:

- Electrocardiograma (ECG)
- Electromiografía (EMG)
- Respuesta Galvánica de la Piel (GSR)
- Electroencefalografía (EEG)

Los movimientos oculares comienzan con una reacción que se procesa en el subconsciente cuando se presenta algo nuevo y terminan en un procesamiento consciente cuando se observa. Por lo que el seguimiento ocular proporciona datos valiosos para la UX sobre la atención del participante a un nivel instintivo y consciente, lo cual resuelve el problema de los métodos de reporte de la UX tradicional, los cuales dependen de la interpretación, los recuerdos y juicios del participante (Hussain, et al., 2018).

Otras disciplinas utilizan la tecnología de seguimiento ocular para medir la atención en lectura, psicolingüística, uso del sitio web, juegos en línea, escritura y adquisición del lenguaje (Bačić, 2017). Y se continúa con la investigación en el campo del comportamiento humano. El uso del seguimiento ocular en el área de UX se ha extendido con el tiempo hasta el punto en donde es un requerimiento para pruebas de diseño de páginas web, aplicaciones y marketing. De forma general, el seguimiento ocular revela:

- ¿Cuáles elementos atraen atención inmediata?
- ¿Cuáles elementos atraen más atención que otros?
- Si algunos elementos son ignorados o pasados por alto
- El orden en que los elementos son observados
- ¿Cuánto tiempo son observados los elementos?

Todos son datos importantes y relevantes para la UX, además se pueden obtener otras métricas relevantes a partir del seguimiento ocular las cuales se explicarán más adelante.

1.1 Antecedentes

Dentro del grupo de investigación de Sistemas Híbridos Inteligentes del TecNM/CENIDET se investigan los temas de la usabilidad y la experiencia de usuario ya que están directamente relacionados con el área de Interacción Humano Computadora. En el centro de investigación se han trabajado métodos e investigaciones relacionadas a la medición de la calidad de la experiencia de usuario y el comportamiento humano.

En el trabajo para grado doctoral de Alejandres (2017) “Evaluación centrada en el usuario de sistemas de recomendación sensibles al contexto: efecto de interfaces multimodales interactivas y esquemas de explicación en la experiencia del usuario”, se describen experimentos utilizando sistemas de recomendación sensibles al contexto, que utilizan interfaces de realidad aumentada, El trabajo buscaba obtener una valoración cuantitativa de aspectos subjetivos de la experiencia del usuario (efectividad, confianza y satisfacción) empleando cuestionarios como instrumentos de evaluación. Los experimentos muestran que realizar explicaciones incrementa la confianza del usuario pero que el tipo de explicación a utilizar depende del factor a potenciar, además, que el uso de una interfaz de realidad aumentada ejerce un esfuerzo positivo sobre la satisfacción y efectividad del usuario y que esos aspectos mantienen una relación proporcional con la calidad percibida.

La tesis doctoral “Metodología UXEEg para la evaluación de la Experiencia del Usuario en personas con discapacidad a partir de Interfaces Cerebro Computadora” de González (2017) desarrolla una metodología de evaluación de experiencia del usuario llamada UXEEg, para ser usada en tecnologías de asistencia/rehabilitación de personas con discapacidades. La metodología ayuda a desarrollar experimentos que permitan identificar correlaciones entre estados mentales y la experiencia del usuario haciendo uso del sensor de EEG. La investigación concluyó que es posible utilizar datos EEG para estimar la cognición de un usuario con discapacidad debido a que se identificaron patrones de actividad cerebral asociados a los estados de concentración y distracción.

Los siguientes antecedentes están relacionados con el software UXLab el cual comenzó con la tesis “Método para integrar y sincronización de datos EEG y multimedia para su aplicación en un ambiente de evaluación para la experiencia del usuario” por Fouilloux (2018). En esta tesis se desarrolló un método y sistema para sincronizar y grabar flujos

multimedia y datos de diferentes sensores fisiológicos y biométricos resultado del proceso de evaluación de la experiencia del usuario (Figura 1).

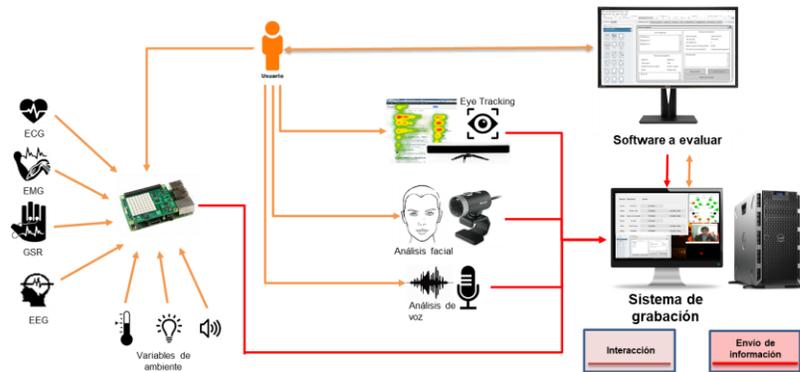


Figura 1: Arquitectura de UXLab, de D. Fouilloux, 2018, Todos los derechos reservados 2018. Tomado con permiso del autor.

En el trabajo de Fouilloux en 2018, UXLab incluye un módulo de seguimiento ocular donde se utiliza el sensor “the eye tribe” modelo ET1000 y por medio de una secuencia de imágenes tomadas por éste, se obtiene la trayectoria de la mirada en forma de puntos. También se generan mapas de calor a partir de los puntos de visión y la utilización de una fórmula de distribución gaussiana multivariable.

Posteriormente, el trabajo de Maestría de García (2020) “Procesamiento de datos fisiológicos para detectar estados afectivos en el proceso de evaluación de la experiencia de usuario” Se incluyen nuevos módulos de sensores fisiológicos y depuración del código de UXLab.

1.2 Planteamiento del problema

En la actualidad dentro de la evaluación de la experiencia de usuario (UX por sus siglas en inglés), el análisis del movimiento ocular del usuario se limita solo a observar lo que está viendo en una interfaz de usuario (UI) de un producto digital, de este modo, se identifica qué le llama más la atención de los componentes que conforman la UI. Los datos de seguimiento ocular por sí solos pueden ofrecer información acerca del comportamiento, por ejemplo, el nivel de atención e interés de los usuarios, sin embargo, al utilizar estas métricas en combinación con otras obtenidas de diferentes tipos de sensores fisiológicos se puede extender la comprensión del comportamiento del usuario. El potencial de los

datos de seguimiento ocular radica en que se pueden relacionar con el estado cognitivo de los usuarios que realizan la evaluación UX, el cual se puede contrastar con los datos obtenidos con algunos de los sensores fisiológicos resultando en un análisis más preciso del estado cognitivo y que apoye a la evaluación de UX.

Para aprovechar por completo el análisis del seguimiento ocular en el campo de la evaluación de la UX se debe analizar y diseñar una metodología que extraiga datos y pueda correlacionar estos con el estado cognitivo de los usuarios. Esto permitiría un mejor entendimiento del comportamiento humano durante la interacción con productos digitales y agrega valor a las evaluaciones de UX.

Esta metodología debe estar implementada en un software que permita correlacionar el seguimiento ocular con el estado cognitivo del usuario, observar el comportamiento de movimiento ocular, el comportamiento del parpadeo y su relación con estados emocionales y cognitivos obtenidos de datos fisiológicos, este sistema también debe guardar y sincronizar las grabaciones de los datos de sensores fisiológicos y biométricos para su posterior procesamiento.

1.3 Objetivos

1.3.1 Objetivo general

Evaluar e implementar una metodología para analizar el movimiento ocular de un usuario durante el proceso de evaluación de la experiencia de usuario para obtener datos relevantes y generar mapas de calor y de trazo de trayectorias correlacionados con el estado cognitivo que experimenta durante el proceso de evaluación.

1.3.2 Objetivos particulares

- Comparar diferentes algoritmos de análisis de seguimiento ocular.
- Analizar el movimiento ocular mediante un sensor biométrico para generar:
 - Puntos de atención.
 - Puntos de fijación.
 - Rutas sacádicas.
 - Áreas de interés.
 - Frecuencia de parpadeo.

- Relacionar los movimientos oculares con el estado cognitivo del usuario.
- Generar mapas de atención (de calor y de rutas).
- Desarrollar e incorporar el módulo de seguimiento ocular de la plataforma de grabación de experiencia de usuario UXLab.
- Evaluar el algoritmo de análisis de seguimiento ocular para validar los resultados con pruebas experimentales mediante la plataforma UXLab.

1.4 Justificación

Realizar un análisis del seguimiento ocular completo junto con la observación de otros datos del comportamiento humano, proporciona una vista más amplia de los estados mentales y preferencias de los usuarios durante la interacción con el producto digital. Lo cual nos permite mejorarlo y adaptarlo en función de los gustos de los usuarios finales, para lograr que el producto tenga una mejor aceptación en el mercado objetivo.

Con el análisis de seguimiento ocular se puede medir: las fijaciones en un área (utilizando mapas de calor), áreas de interés, secuencias de fijaciones, el tiempo a la primera fijación, el tiempo gastado, conteo de respuesta, tamaño de pupila y parpadeos.

Una fijación se define como un punto en la pantalla (x, y) en donde el usuario fija la mirada. Las fijaciones en función del tiempo dejan un rastro que puede ser representado como un mapa de calor o un mapa de rutas sacádicas, estos mapas son la forma más común para analizar y visualizar el comportamiento del movimiento ocular de un usuario. Las áreas de interés (AOI, por sus siglas en inglés) son secciones de la interfaz de usuario (UI, por sus siglas en inglés) de un producto digital que ayudan a identificar las áreas en las que se concentra la mayor atención de la mirada del usuario.

Por otro lado, otra métrica utilizada es la dilatación de la pupila del usuario, aunque es sensible a la luz ambiente y al brillo de pantallas, está correlacionada a los estados emocionales y de concentración del usuario. Otra métrica utilizada es el conteo de los parpadeos, los cuales se pueden relacionan con la carga de trabajo en el proceso cognitivo. Al observar la frecuencia de los parpadeos de un usuario se puede inferir si lo que ve le genera interés o aburrimiento.

El análisis de estas métricas de seguimiento ocular revela datos valiosos del comportamiento del usuario que pueden ser utilizados en el proceso de evaluación de la

experiencia de usuario (UX, por sus siglas en inglés). Por ejemplo, al analizar las fijaciones de un usuario en una página web se pueden generar AOI, a partir de esta métrica se puede establecer cuánto tiempo visita un área y en función de esta métrica modificar la vista de la página web para colocar información relevante en las áreas más visitadas.

1.5 Alcances y limitaciones

A continuación, se presentan los alcances y limitaciones del proyecto de tesis propuesto.

1.5.1 Alcances del proyecto

- Se utilizó la API del sensor de seguimiento ocular “Tobii Eye Tracker 4C” para obtener datos.
- Se desarrolló una metodología para extraer de la información de seguimiento ocular: los puntos de visión, los puntos de fijación y los parpadeos de los usuarios. Y en conjunto con datos obtenidos a partir de sensores fisiológicos obtener información acerca del comportamiento del usuario que sea útil para la evaluación de la experiencia de usuario.
- Integrar la captura de información del seguimiento ocular a la plataforma de evaluación de experiencia de usuario UXLab.
- Se obtuvo el estado cognitivo general del usuario relacionando el estado cognitivo evaluado a partir de los datos de seguimiento ocular y el estado cognitivo evaluado de los datos de sensores fisiológicos (disponibles en el módulo de evaluación fisiológica de UXLab). Mediante la aplicación de algoritmos de aprendizaje o máquinas de clasificación.
- Implementar mapas de calor, mapas de rutas, el estado cognitivo y la emoción percibida por el usuario sobre los videos de pruebas de experiencia de usuario utilizando los datos obtenidos del sensor de seguimiento ocular.
- Se evaluaron y validaron los resultados con métodos estadísticos.

1.5.2 Limitaciones del proyecto

- No se obtendrán métricas a partir de la pupila.
- No se detectarán emociones.

1.6 Estructura del documento

Este documento será dividido en capítulos, comenzando con el capítulo actual que es la introducción, seguido del marco teórico donde se contextualizará al lector sobre los temas que trata el documento, el capítulo 3 presenta una recopilación del estado del arte, el capítulo 4 describe el análisis de las métricas oculares, el capítulo 5 muestra el trabajo de integración de algoritmos de seguimiento ocular en el software UXLab, en el capítulo 6 se presentan las pruebas y análisis de la correlación de las métricas oculares con los estados mentales reportados por UXLab, el capítulo 7 trata las conclusiones del trabajo de investigación y finalmente el capítulo 8 muestra las referencias.

Capítulo 2. Marco teórico

A continuación, se presenta la teoría fundamental en la cual se basa el contenido de este documento.

2.1 Experiencia de usuario

Según la Organización Internacional de Normalización (ISO, 2010) la experiencia de usuario, (UX por sus siglas en inglés), se define como “las percepciones y respuestas de una persona que resultan del uso de un producto, sistema o servicio”. El término de experiencia de usuario se refiere entonces a la sensación, buena o mala, de la interacción de los usuarios con un producto, sistema o servicio. Ésta depende de muchos factores del producto o servicio como lo son el hardware, la accesibilidad, el diseño gráfico y visual, la calidad de los contenidos, la utilidad, etc. Y además los aspectos relativos a las emociones y sentimientos de la persona que lo utiliza.

La evaluación de la experiencia de usuario se realiza con el fin de observar la percepción de los usuarios ante un producto, sistema o servicio, en función de las observaciones positivas o negativas de la evaluación, se pueden tomar decisiones para mejorar el objeto de la prueba. La experiencia de usuario expande la usabilidad en el sentido de que estudia no solo la eficiencia de las interfaces sino también al usuario final como humano (Tzafilkou & Protogeros, 2017)

En los últimos años las investigaciones sobre experiencia de usuario han tomado gran importancia para las empresas ya que se busca cambiar el enfoque de desarrollo de productos para que gire en torno a los comentarios de usuarios reales y por lo tanto se logre un mayor nivel de aceptación en el grupo de los usuarios finales que utilizaran el producto.

2.2 Estado Mental

Estado o proceso que corresponde con el pensar y sentir de un sujeto, y que además posee propiedades espaciotemporales, causas y efectos distinguibles (Goldstein, 2000). Los estados mentales se clasifican en dos tipos: estados cognitivos y estados afectivos.

2.3 Estados cognitivos

Un estado cognitivo es una condición neuropsicológica que presenta un sujeto durante la realización de una o más tareas cognitivas, las cuales contemplan los procesos necesarios para el cálculo, la concentración y la memoria, además de las funciones ejecutivas, de las cuales se consideran las siguientes:

- Concentración.
- Planificación de tareas.
- Resolución de problemas.
- Formular conceptos abstractos.
- Autoconciencia.

2.4 Estados Afectivos

Se entiende por estados afectivos a la condición neuropsicológica que presenta un sujeto. Los estados afectivos se clasifican en emociones y sentimientos. Los estados afectivos se diferencian principalmente por el tiempo de duración, los sentimientos tienen una duración mayor que las emociones. En la

Tabla 1 se presenta la clasificación de los estados afectivos y algunos ejemplos que permiten diferenciarlos.

Clasificación de los estados afectivos	
Emociones	Sentimientos
Miedo	Vergüenza
Ira	Resentimiento
Tristeza	Amor
Alegría	Simpatía
Sorpresa	Orgullo
Asco	Celos

Clasificación de los estados afectivos	
Emociones	Sentimientos
Miedo	Vergüenza
Ira	Resentimiento
Tristeza	Amor
Alegría	Simpatía

Sorpresa	Orgullo
Asco	Celos

Tabla 1: Clasificación de los estados afectivos, de Soriano, 2018, Todos los derechos reservados 2018.

Tomado con permiso del autor.

Los estados emocionales se clasifican en dos tipologías: estados emocionales positivos y estados emocionales negativos. En la Figura 2 se muestra una categorización de estados mentales que según la neurociencia.

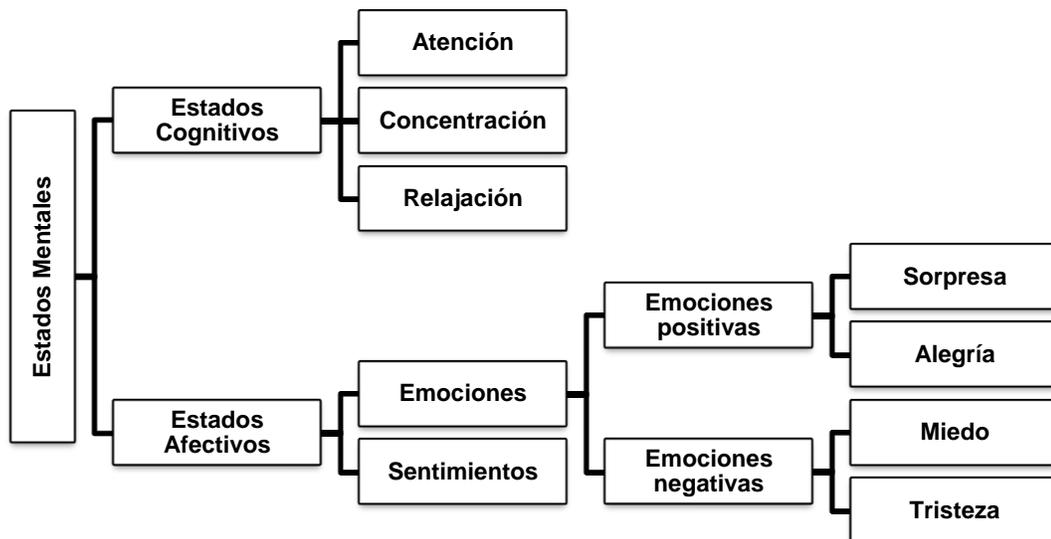


Figura 2: Clasificación de estados mentales, de Soriano, 2018, Todos los derechos reservados 2018.

Tomado con permiso del autor.

2.5 Sensores fisiológicos

Los sensores fisiológicos permiten medir ciertas características de las funciones y mecanismos que tienen los seres vivos, por ejemplo, los sensores de electrocardiograma (ECG), respuesta galvánica en piel (GSR), los sensores de electroencefalograma (EEG), entre otros. Ayudan a los investigadores a obtener pistas del comportamiento humano cuando las personas se enfrentan a ciertos estímulos.

2.6 Seguimiento ocular

El seguimiento ocular es una tecnología que ayuda a entender la atención visual. Existen dos tipos de sensores que pueden extraer la información del movimiento ocular de un usuario, el primero utiliza cámaras de video y analizar el movimiento de los ojos. Mientras que el segundo utiliza tecnología infrarroja para procesar la reflexión de la luz en la pupila del usuario (ver Figura 3). La reflexión de la luz sobre la pupila es registrada por un sensor que calcula la posición en coordenadas (x, y) del punto que representan la mirada del usuario en el plano espacial de la pantalla. Los datos en este trabajo fueron obtenidos con un sensor de seguimiento ocular infrarrojo.

El seguimiento ocular detecta en dónde está mirando el usuario en un punto del tiempo, cuánto tiempo mira algo y la ruta que sus ojos siguen. En el campo de la experiencia de usuario, el seguimiento ocular ayuda a entender de manera más completa la percepción del usuario (Romano Bergstrom & Schall, 2014).

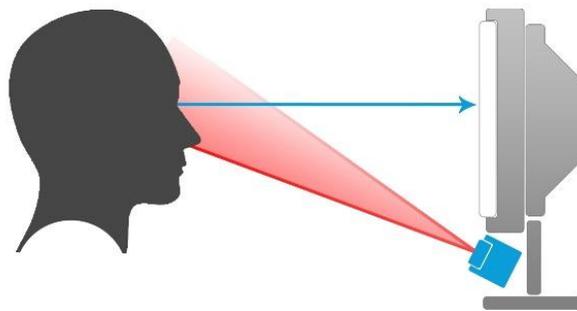


Figura 3: Esquema del funcionamiento del seguimiento ocular con un sensor infrarrojo, de D. Fouilloux, 2018, Todos los derechos reservados 2018. Tomado con permiso del autor.

2.6.1 Puntos de fijación y visión

Los puntos de visión son la unidad básica de medición del seguimiento ocular, se obtienen directamente de los sensores de seguimiento ocular y representan puntos, en forma de coordenadas (x, y) (Figura 4). Las coordenadas del punto de visión representan un punto en el plano de visión del usuario, comúnmente este plano es la pantalla o monitor donde está colocado un sensor de seguimiento ocular, pero también puede representar el plano de visión en lentes con tecnología de seguimiento ocular.

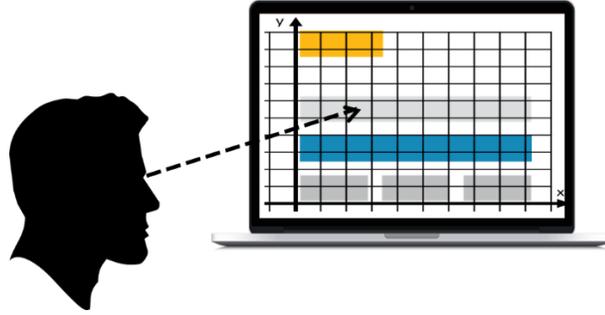


Figura 4: Esquema de coordenadas de visión en una pantalla.

Un punto de visión (1) es un conjunto que contiene tres elementos: una coordenada x , una coordenada y y una estampa de tiempo t .

$$\text{punto de visión} = \{x, y, t\} \quad (1)$$

Donde:

- x es un punto en el eje x del plano espacial de la pantalla.
- y es un punto en el eje y del plano espacial de la pantalla.
- t es la estampa de tiempo que define el momento en que ocurrió el punto de visión.

2.6.2 Métricas de fijación y visión

Utilizando los puntos de visión del usuario se obtienen las coordenadas de fijación visual, las fijaciones se representan como muestras discretas de puntos estables donde mira el ojo (Borys y Plechawska Wójcik, 2017). Los puntos de observación se agregan según un área y un intervalo de tiempo específicos. El área de agregación suele ser de unos 20 a 50 píxeles, en un intervalo de tiempo entre 200 y 300 ms (Holmqvist, et al., 2010).

Un punto de fijación se obtiene calculando la diferencia entre la velocidad de un primer punto de visión al siguiente o como grupos de puntos consecutivos dentro de una dispersión específica, o separación máxima (Salvucci y Goldberg, 2000). Y se representan como un conjunto de cinco elementos (2): coordenada x , coordenada y , tiempo de inicio t_i , tiempo de fin t_f y duración de la fijación d .

$$\text{punto de fijación} = \{x, y, t_i, t_f, d\} \quad (2)$$

Donde:

- x es un punto en el eje x del plano espacial de la pantalla.
- y es un punto en el eje y del plano espacial de la pantalla.
- t_i es la estampa de tiempo que define el momento en que ocurrió la fijación.
- t_f es la estampa de tiempo que define el momento en que termino la fijación.
- d es la diferencia de t_i y t_f , representa el tiempo que duro la fijación.

Actualmente la mayoría de los sensores de seguimiento ocular calcula directamente los puntos de fijación y pueden ser distinguidos de puntos de visión simples. Los puntos de fijación son la base para obtener mapas de atención que son formas de representar la atención visual de un usuario. Dos de los mapas de atención más utilizados, ya que son fáciles de interpretar, son los mapas de calor y los mapas de rutas sacádicas.

2.6.2.1 Mapas de calor

Representan las agregaciones de los puntos de fijación y/o de visión de uno o varios usuarios, revelan la distribución de la atención visual utilizando una serie de colores (azul = frio, verde = tibio y rojo = caliente) para representar temperatura, las temperaturas altas, de color rojo, indican mayor tiempo de atención en el área (Figura 5).



Figura 5: Mapa de calor: Realizado con los datos de seguimiento ocular de una prueba de evaluación UX.

Para realizar un mapa de calor se debe tratar cada píxel disponible en la imagen de la captura como un espacio en un arreglo de dos dimensiones, utilizando el conjunto de puntos de fijaciones se aumentará la intensidad de un píxel (i, j) relacionado con la coordenada (x, y) de cada uno de los puntos de fijación (en varias iteraciones) y además, se aumenta el valor de los píxeles alrededor de éste en una menor medida utilizando α (ver formula 3), normalmente, utilizando un kernel gaussiano (Duchowski, et al., 2012), esto para asignarle un peso de manera más uniforme a los píxeles vecinos. Mientras más alto sea el valor del píxel se le asignará un color más caliente.

$$intensidad\ píxel(i, j) = exp - \frac{(x-i)^2 + (y-j)^2}{2\alpha^2} \quad (3)$$

Donde:

- x es un punto capturado en el eje x del plano espacial de la pantalla.
- y es un punto capturado en el eje y del plano espacial de la pantalla.
- i es un punto de un píxel en el eje x del plano espacial de la pantalla.
- j es un punto de un píxel en el eje y del plano espacial de la pantalla.
- α es la desviación estándar del conjunto de datos capturados, se puede modificar el valor para variar la intensidad.
- La función es iterativa se calcula para cada píxel (i, j) disponible en la pantalla.

Al final se procesa el arreglo como una imagen con los colores de temperatura y se coloca sobre la imagen o video capturado durante la prueba.

2.6.2.2 Mapas de rutas

Los mapas de rutas (ver Figura 6), al igual que los de calor, sirven para representar la atención visual de un usuario o de un conjunto de ellos, se definen como una secuencia de fijaciones alternas y sácadas (Holmqvist, et al., 2010). Los elementos visuales de los mapas de rutas son tres: círculos para representar una fijación y el diámetro del círculo que representa la duración de la fijación. Cada círculo está numerado, para representar la secuencia en el que ocurrieron las fijaciones. Finalmente, el último elemento es un

conjunto de vectores que unen los círculos, esto representa la ruta que siguieron los ojos para llegar a cada una de las fijaciones.



Figura 6: Mapa de rutas: Realizado con los datos de seguimiento ocular de una prueba de evaluación UX.

Para generar el mapa de rutas se debe utilizar el conjunto de puntos de fijación para obtener el conjunto de sácadas. Éstas describen un movimiento ocular rápido de una fijación a otra. Suelen durar entre 30 y 80 ms y son el movimiento más rápido que puede realizar el cuerpo humano (Holmqvist, et al., 2010). Representan el movimiento de transición entre dos fijaciones y se describen con cinco elementos (4): el punto x de la primera y segunda fijación, el punto y de la primera y segunda fijación y la duración de la sácada d (que es calculada restando el tiempo de fin t_f de la primera fijación con el tiempo de inicio t_i de la segunda fijación).

$$sacada = \{ x_1, y_1, x_2, y_2, d \} \quad (4)$$

Donde:

- x_1 es un punto de la primera fijación en el eje x del plano espacial de la pantalla.
- y_1 es un punto de la primera fijación en el eje y del plano espacial de la pantalla.
- x_2 es un punto de la segunda fijación en el eje x del plano espacial de la pantalla.
- y_2 es un punto de la segunda fijación en el eje y del plano espacial de la pantalla.
- d es la diferencia de t_i de la primera fijación y t_f de la segunda fijación, representa el tiempo que duro la sácada.

Una vez obtenido el conjunto de sácadas se procesa la imagen de la prueba para sobreponer los puntos de fijación en orden con las sácadas que los unen de forma que el par de coordenadas de la primera fijación forma el origen de la flecha y el par de coordenadas de la segunda fijación forma el final del trazo de la flecha.

2.6.2.3 Áreas de interés

Otra de las métricas del seguimiento ocular son las áreas de interés también conocidas como AOI's o ROI's (Areas Of Interest, y Regions Of Interest, por sus siglas en inglés). En la práctica se pueden definir manualmente áreas de interés donde los investigadores observan el comportamiento ocular dentro de las zonas establecidas manualmente. También es posible obtener áreas de interés automáticamente mediante algoritmos de agrupamiento, el principio es obtener n número de clústeres de forma automática utilizando todo el conjunto de puntos de visión que fueron capturados durante una prueba de seguimiento ocular. Al agrupar los puntos se obtiene de forma dinámica un conjunto de clústeres a los cuales pertenecen n puntos de visión, el centro de cada clúster representa también el centro del área de interés mientras que los demás puntos se utilizan para determinar el área que abarca la región de interés (ver Figura 7).

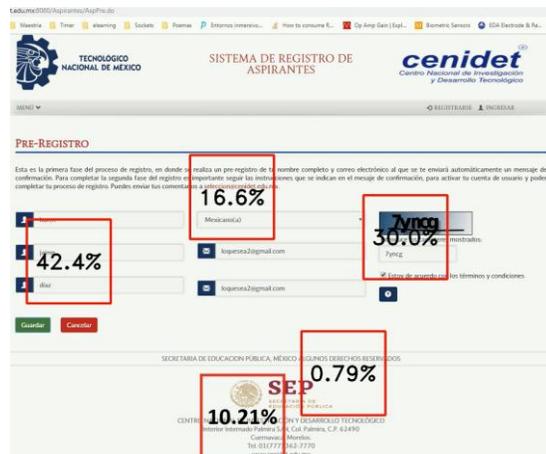


Figura 7: Áreas de interés: Realizado con los datos de seguimiento ocular de una prueba de evaluación UX, los porcentajes que se muestran en los rectángulos representan la proporción de puntos de visión en ese clúster, del total del conjunto de puntos de visión del usuario.

Uno de los algoritmos de agrupamiento que obtiene un número dinámico de clústeres de manera automática y que ha sido utilizado para obtener las áreas de interés con datos de seguimiento ocular es el algoritmo Meanshift (Fuhl, et al., 2018; Santella y DeCarlo, 2004).

2.6.2.4 Parpadeos

Los parpadeos (Figura 8) son una fuente de información sobre la carga cognitiva, se consideran como el cierre total de los párpados en los dos ojos y duran aproximadamente de 100 a 150 ms bajo condiciones normales (Chen, et al., 2011), cualquier tiempo mayor se conoce como un microsueño.

En varios trabajos encontrados en la literatura (Zagermann, et al., 2018; Sakai, et al., 2017; Zagermann, et al., 2016; Mallick, et al., 2016; McIntire, et al., 2014; Chen, et al., 2011; Smilek, et al., 2010) el número de parpadeos se correlaciona con la carga cognitiva del usuario que realiza una tarea visual. Esto demuestra que la frecuencia de los parpadeos se puede utilizar para obtener información acerca de la carga cognitiva que le genera un estímulo y para obtener una idea general de su nivel de atención.

Los estudios demuestran que, si la frecuencia de parpadeos es baja, el usuario se enfrenta a una carga cognitiva alta y prestan más atención a las tareas, por ejemplo, al ver una película o jugar un videojuego. Por otro lado, una frecuencia de parpadeo alta denota lo contrario, una carga cognitiva baja y falta de atención o aburrimiento.



Figura 8: Representación gráfica de un parpadeo.

2.6.2.5 Otras métricas del seguimiento ocular

En función de los datos extraídos se pueden obtener otras métricas que proporcionan información del comportamiento ocular del usuario.

Tiempo hacia la primera fijación

Cuando nos interesa un objeto o área en específico, el tiempo que toma, desde el inicio de la prueba, a un usuario observar por primera vez ese objeto o área en la pantalla (ver Figura 9), proporciona una idea de que tan accesible se encuentra el objeto o área para ser encontrado, normalmente aplica cuando se busca saber si un segmento de información de una página web es llamativo. Esta métrica identifica los objetos que más llaman la atención del usuario. La métrica mide la diferencia de tiempo entre el inicio de la prueba y el tiempo en el que se registra la primera fijación (Guo, Ding, Liu, Liu, y Zhang, 2016).



Figura 9: Esquema de tiempo hacia la primera fijación.

Recuento de visión

En el mismo contexto de una prueba en página web, esta métrica utiliza un simple conteo de puntos de visión dentro de un área de interés definida, para obtener información acerca de la concentración de visión y cuántos usuarios guiaron su mirada a un área de interés específica (Figura 10). Un conteo más elevado, en relación con otras áreas o contenidos, puede indicar que los datos están siendo influenciados por algún material dentro del área de interés, de esta forma se obtienen las áreas más populares para los usuarios. (Guo, et al., 2016; Shojaeizadeh, Djamasbi, Paffenroth, y Trapp, 2019)

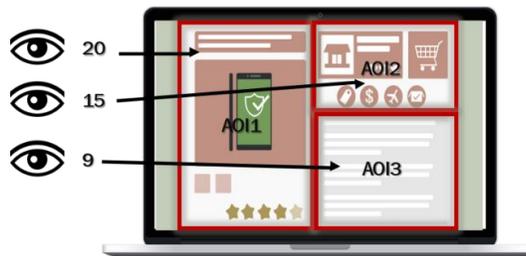


Figura 10: Esquema de recuento de visión.

Tiempo gastado

El tiempo gastado cuantifica la cantidad de tiempo que los usuarios pasan en un área de interés, realizando la sumatoria de la duración de las fijaciones que se encuentren dentro del rango de coordenadas que definen el perímetro del área de interés. Mientras más tiempo se mire un área se puede inferir un mayor nivel de interés, mientras que un menor tiempo puede indicar que otras áreas en la pantalla son más llamativas (Figura 11) (Shojaeizadeh, et al., 2019). En el contexto de diseño web esto ayuda a los diseñadores a definir las áreas en donde poner la información relevante del sitio.

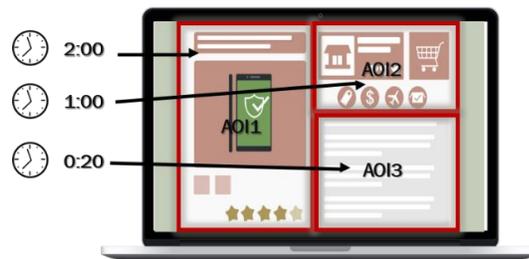


Figura 11: Esquema de tiempo gastado.

Tamaño de pupila o dilatación

El tamaño de la pupila responde a factores como la iluminación, heridas, enfermedades, medicamentos, o a estímulos visuales. Si se controlan los factores de iluminación y de salud de una persona, del diámetro que presenta su pupila (Figura 12) se pueden extraer la excitación emocional y la carga de trabajo cognitiva del usuario (Wang, Yang, Liu, Cao y Ma, 2014; Zheng, Dong y Lu, 2014; Lu, Zheng, Li y Lu, 2015; Guo, et al., 2016; Mallick, Slayback, Touryan, 2016; Shojaeizadeh, et al., 2019). Esta métrica medida en cambios de milímetros de diámetro de la pupila ayuda a conocer la reacción de los usuarios a ciertos estímulos o elementos visuales.

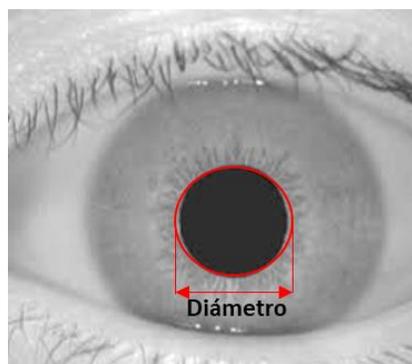


Figura 12: Representación de la medición del tamaño de la pupila (CASIA-IrisV3, 2005).

Distancia a la pantalla

Los sensores de seguimiento pueden calcular la distancia entre el usuario y la pantalla (Figura 13), este dato sirve para observar el comportamiento de acercamiento-rechazo del usuario ante diferentes estímulos (Farnsworth, 2018). Esto se mide en centímetros desde la captura del sensor de seguimiento a los ojos del usuario.

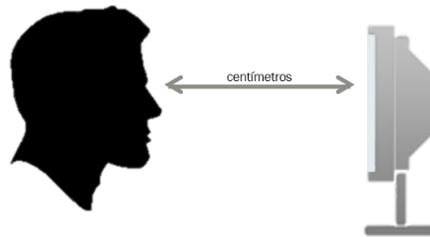


Figura 13: Esquema de la distancia entre el usuario y la pantalla.

Capítulo 3. Análisis

3.1 Análisis del estado del arte

En esta sección se presenta un breve resumen del estado actual de las investigaciones que utilizan el seguimiento ocular para estudios relacionados con la evaluación de la experiencia de usuario y de trabajos que traten el seguimiento ocular para la detección de estados mentales.

3.1.1 Trabajos relacionados con la evaluación de la UX

Dentro del estado del arte existen múltiples trabajos que han investigado la viabilidad de utilizar el seguimiento ocular para pruebas relacionadas a la evaluación de la experiencia de usuario (UX, por sus siglas en inglés). Varios de estos trabajos utilizan métricas de seguimiento ocular como mapas de atención y áreas de interés para observar el comportamiento ocular de los usuarios.

Por ejemplo, el trabajo de (Ho, 2014), estudia el comportamiento visual de las mujeres cuando visitan el catálogo de productos de una tienda en línea, los datos de seguimiento ocular, en especial las áreas de interés revelan las preferencias en características de los productos, esta información se utiliza para cambiar la forma en que los productos son mostrados a los clientes.

Otros estudios utilizan el seguimiento ocular directamente para proponer mejoras de productos digitales, en un estudio de (Qu, Zhang, Chao y Duffy, 2016) se utilizan los datos oculares de los usuarios que utilizan aplicaciones de teléfonos móviles para evaluar la experiencia de los usuarios y así poder mejorar las interfaces. En el trabajo de (Gaspar, Kompan, Simko y Bielikova, 2018) se investigó el comportamiento visual y los patrones que presentaron las personas al escoger diferentes películas, esto con el objetivo de identificar los factores de objetos en páginas web que pueden causar un sesgo en la atención visual de sus usuarios.

La publicación de (Burger, Guna y Pogacnik, 2018) buscaba comprobar si los dispositivos de seguimiento ocular comerciales de bajo costo aportan mejoras relevantes al análisis de experiencia de usuario. Se compararon los datos obtenidos del sensor con los resultados de encuestas tradicionales de evaluación UX. Los resultados indican que los datos objetivos del sensor tienen una correlación significativa con aquellos de los

métodos tradicionales y además los mapas de calor, las áreas de interés, las fijaciones y las transiciones entre diferentes puntos proveen más información sobre la experiencia de usuario.

Uno de los trabajos más destacados es el de (Hussain et al., 2018), donde se desarrolló una plataforma completa que utiliza diferentes métodos y sensores para evaluar la percepción de los usuarios ante productos digitales. Es una plataforma multimodal cuya arquitectura se construyó en 4 capas: la capa de datos, la capa de medición de UX, la capa de análisis y la capa de visualización. Al evaluar la experiencia se obtuvieron diferentes datos del usuario durante la prueba: La grabación de video y audio del usuario, la respuesta galvánica en piel, el electroencefalograma, el seguimiento ocular y encuestas de experiencia de usuario al final de la prueba. Con estos datos utilizando métodos de extracción de características y fusión de éstas, se logró identificar las emociones de los usuarios.

Dentro de esta fusión de datos se utilizó, de la información obtenida del sensor de seguimiento ocular, el tamaño de la pupila de los usuarios en combinación con las señales del sensor de EEG para mejorar el reconocimiento de emociones: positivas, negativas o neutrales. Además, para medir la atención visual de los usuarios, se utilizaron los puntos de fijaciones para crear mapas de calor.

3.1.2 Trabajos relacionados con los estados mentales

A continuación, se describen algunos trabajos que relacionan movimientos oculares con estados mentales de los usuarios.

Los trabajos de (Zheng, Dong y Lu, 2014) y (Lu, Zheng, Li y Lu, 2015), se relacionan con utilizar clasificadores con una combinación multimodal de variables de datos EEG y de seguimiento ocular, en este caso la dilatación de la pupila, para detectar emociones positivas, negativas o neutrales de las personas en sus experimentos. Según los resultados publicados en sus artículos, realizar la fusión de datos de pupila con EEG aumenta la precisión de predicciones emocionales cuando solo se utiliza EEG.

Otros trabajos buscan correlacionar la carga de trabajo cognitiva con algunas características oculares como lo es la dilatación de la pupila, el número de fijaciones, sácadas y parpadeos. El estudio de (Wang, Yang, Liu, Cao y Ma, 2014) buscaba medir la

complejidad de páginas web desde una perspectiva de carga cognitiva utilizando las métricas oculares en conjunto con el tiempo que tomo realizar diferentes tareas.

Un trabajo que sirvió como base para experimentos en la investigación fue el realizado por (Mallick, Slayback, Touryan, 2016) donde se planteó observar los cambios en las métricas de seguimiento ocular: parpadeos, fijaciones, sacadas y tamaño de la pupila durante la interacción con el video juego de Tetris en diferentes niveles de complejidad, el estudio concluye que mientras la carga de trabajo aumenta, la frecuencia de los parpadeos disminuye, la frecuencia de las fijaciones aumenta, la velocidad de las sacadas aumenta y que el tamaño de la pupila también aumenta. Otros trabajos también utilizan diferentes experimentos para correlacionar las diferencias de movimientos oculares mientras los usuarios realizaron diversas tareas, como actividades de vigilancia de control de tráfico aéreo (McIntire, McKinley, Goodyear y McIntire, 2014), ejercicios de multiplicaciones (Kosch, Hassib, Buschek y Schmidt, 2018), entre otros.

3.1.3 Discusión sobre trabajos del estado del arte

Todos los trabajos investigados tienen en común la utilización de tecnología de seguimiento ocular para obtener y analizar el comportamiento humano, dependiendo del caso de estudio, se utilizan distintas formas de visualizar la información obtenida de los sensores para interpretar la atención visual o el estado cognitivo de las personas. La información obtenida en muchos casos es utilizada para apoyar las evaluaciones de UX o para reconocer patrones de estados mentales combinando datos de otros sensores.

3.1.4 Tabla comparativa de técnicas analizadas del estado del arte

A continuación, se presenta una tabla comparativa que muestra las técnicas y métodos utilizados en el estado del arte para obtener información del comportamiento de los usuarios con métricas de seguimiento ocular (Tabla 2), en esta tabla se puede observar cómo se ha utilizado la tecnología de los sensores de seguimiento ocular, en ocasiones combinándola con otro tipo de datos de sensores fisiológicos, para atender las necesidades de las investigaciones. En la última fila de la tabla se muestran las técnicas y métodos que fueron utilizados en esta tesis, diferenciando el trabajo realizado con otros del estado del arte.

Tabla 2: Tabla comparativa de técnicas y métodos del estado del arte.

Trabajo	Visualización de datos				Tipo de estímulo			Sensor		Métricas extraídas					Combinación de sensores				Evaluación de estado							
	Mapas de calor	Mapas de rutas	Áreas de interés	Otro	Estático	Dinámico	Interactivo	Infrarrojo	Video	Puntos de visión	Fijaciones	Parpadeos	Tamaño de pupila	Sacadas	Otros	EEG	GSR	Audio	Reconocimiento facial	Otros	Emocional			Cognitivo		
																					Emoción positiva	Emoción neutra	Emoción negativa	Atención	Carga de trabajo	
(Burger, Guna, & Pogacnik, 2018)	✓		✓		✓		✓	✓		✓	✓			✓	✓											
(Gaspar, Simko, Kompan, & Bielikova, 2018)		✓	✓		✓			✓			✓				✓											✓
(Ho, 2014)		✓	✓		✓			✓		✓	✓															✓
(Hussain, et al., 2018)	✓				✓		✓	✓		✓		✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
(Kosch, Hassib, Buschek, & Schmidt, 2018)				✓		✓	✓	✓				✓														✓
(Lu, Zheng, Li, & Lu, 2015)				✓		✓		✓				✓			✓						✓	✓	✓			
(Mallick, Slayback, Touryan, Ries, & Lance, 2016)				✓		✓	✓	✓		✓	✓	✓	✓												✓	✓
(McIntire, McKinley, Goodyear, & McIntire, 2014)				✓		✓	✓	✓			✓			✓					✓						✓	
(Qu, Zhang, Chao, & Duffy, 2016)	✓	✓	✓		✓		✓	✓						✓											✓	
(Wang, Yang, Liu, Cao, & Ma, 2014)	✓				✓		✓	✓			✓			✓												✓
(Zheng, Dong, & Lu, 2014)				✓		✓		✓				✓			✓						✓	✓	✓			
Tesis	✓	✓	✓		✓		✓	✓		✓	✓	✓			✓										✓	✓

3.2 Análisis de las métricas de seguimiento ocular en la evaluación UX

Las técnicas de visualización de datos del seguimiento ocular permiten a los investigadores analizar diferentes niveles y aspectos de la información registrada de forma exploratoria y cualitativa. Ayudan a analizar el aspecto espaciotemporal de los datos de seguimiento ocular y sus relaciones con los datos (Blascheck, et al., 2014).

3.2.1 Mapas de atención

Los mapas de atención revelan las secciones del producto digital que más observaron los usuarios, los elementos más observados pueden ser interpretados como más llamativos para los usuarios o como elementos difíciles de comprender. Un sitio web complejo distraerá la atención de los usuarios y activará en consecuencia más fijaciones. Pero los efectos de la complejidad del sitio web pueden ser sensibles a la existencia de otros factores, como la complejidad de la tarea (Wang, et al., 2014). Observar más tiempo un elemento representa más atención, pero no representa una percepción positiva o negativa del elemento del sitio web, el mapa de atención puede ser apoyado por cuestionarios de percepción para cada usuario y/o de sensores fisiológicos como de electroencefalografía, que puedan detectar estados mentales positivos o negativos.

Otro de los usos de los mapas de atención es observar si alguno de los elementos del sitio web, como menús, imágenes, publicidad o información importante, están siendo observados por el usuario final. Los mapas de calor pueden mostrar si una página web es adecuada para ayudar a los usuarios a encontrar puntos de entrada importantes de manera efectiva y si anima a los usuarios a inspeccionar la página a fondo (Djamasbi, 2014). Los diseñadores del software pueden utilizar esta información para cambiar el diseño del producto en función de la atención que se desee para cada elemento del sitio web.

En el caso específico de los mapas de rutas se puede observar el camino que sigue la mirada del usuario, esto permite identificar desde dónde comienza a navegar el usuario en el sitio web (elemento de entrada), el tiempo que permanece la mirada en la sección y el orden de los cambios entre cada sección después de la primera, de modo que se puede observar qué secciones se visualizan en función de haber visto otra anteriormente.

Algunos de los datos que pueden ser analizados para los mapas de atención son el número de fijaciones y su duración, estos datos representan la carga cognitiva sobre un elemento o en toda la prueba. Y el tiempo a la primera fijación, que representa el tiempo que le toma al usuario realizar su primera fijación cuando es expuesto a un nuevo estímulo, determina qué tan rápido encuentra algo que le llama la atención.

Los mapas de atención deben ser interpretados por un experto evaluador de la UX, quien decidirá, en conjunto con respuestas de cuestionarios de evaluación y datos de sensores fisiológicos, si los elementos observados generan una percepción positiva o no en el usuario.

3.2.2 Áreas de interés

Las áreas de interés proporcionan información acerca de las secciones específicas de la interfaz del producto digital que están recibiendo más atención visual de parte del usuario y en consecuencia realizar análisis similares a aquellos que se hacen en los mapas de atención (tiempo de permanencia, número de fijaciones, entre otras). Además, estas secciones pueden ser aisladas del resto para generar nuevas áreas de interés y mapas de atención para estudiar el comportamiento centrado en el límite de estas regiones.

En este trabajo se describe cómo se crearon áreas de interés dinámicas utilizando algoritmos de agrupamiento, sin embargo, también se pueden definir áreas de interés manualmente seleccionando áreas específicas de la interfaz para obtener datos de comportamiento por cada una, por ejemplo, un diseñador quiere observar el comportamiento visual de sus usuarios solo en la sección de publicidad de su página web.

3.2.3 Análisis del parpadeo

Finalmente, el análisis del parpadeo también aporta métricas interesantes a la evaluación de la UX, por ejemplo, un histograma de parpadeos por minuto puede mostrar si la frecuencia de parpadeo aumenta o se reduce. Un aumento en la frecuencia de los parpadeos puede indicar falta de interés en el contenido del producto digital mientras que una reducción de la frecuencia del parpadeo puede significar que el usuario está interesado, concentrado y/o prestando atención a los contenidos del producto.

3.3 Análisis de la herramienta para evaluar la experiencia de usuario UXLab

UXLab es un producto digital resultado de la investigación de la tesis “Método para integrar y sincronizar datos EEG y multimedia para su aplicación en el proceso de evaluación de la experiencia de usuario “(Fouilloux, 2018). En la tesis de maestría se utilizaron algoritmos de sincronización de datos, necesarios para tomar las señales de diferentes sensores y alinear el flujo de datos durante la grabación de la evaluación UX. UXLab ha evolucionado desde la versión inicial con el trabajo de otras investigaciones que han agregado funcionalidades al software. La figura 14 muestra la arquitectura de 4 capas de UXLab donde se muestra con recuadros de línea punteada los módulos que se integraron a este software resultado de este trabajo de tesis.

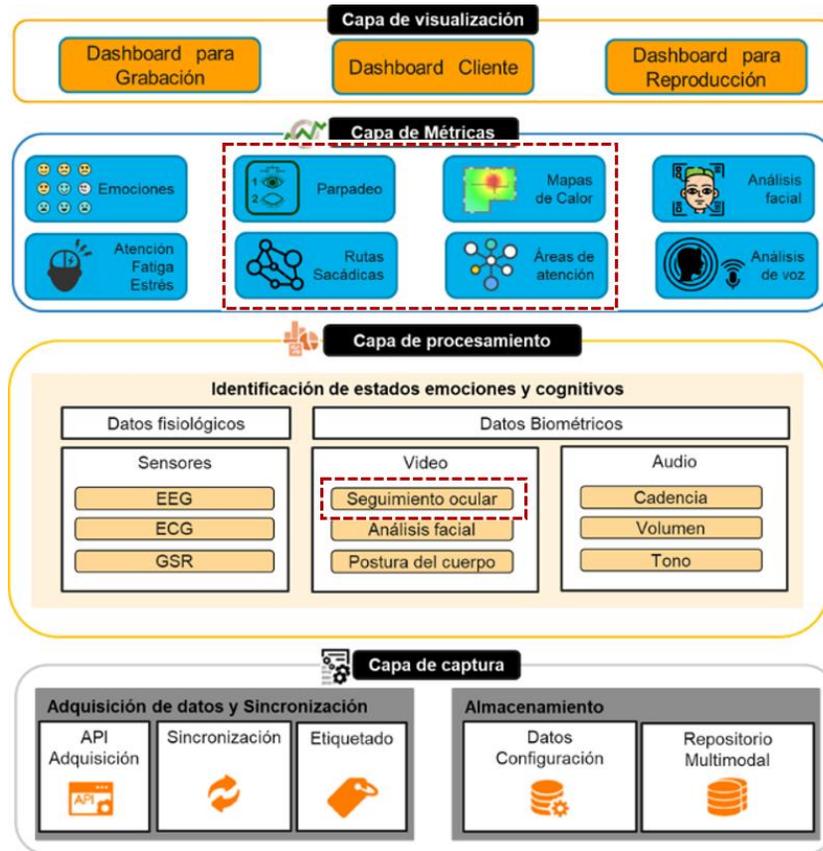


Figura 14: Arquitectura de UXLab. El trabajo se centra en las capas de métricas y capa de captura (Rodeado con rojo). Fuente: UXLab Arquitectura (Diapositiva 6), de G. González, 2019, Todos los derechos reservados 2019. Adaptado con permiso del autor.

UXLab permite grabar simultáneamente y de manera sincronizada diferentes sensores fisiológicos y de comportamiento mientras un usuario interactúa con algún estímulo. La herramienta está desarrollada en el lenguaje de programación Python en su versión 2.7 y el lenguaje C# utilizando el framework de interfaces WPF.

Para evaluar el método desarrollado en esta tesis se implementó, en UXLab, un módulo para registrar el movimiento ocular de un usuario mediante un sensor con tecnología infrarroja especializado para capturar el movimiento de los ojos. A continuación, se describen los componentes de UXLab.

3.3.1 Sistema de grabación de UXLab

El sistema de grabación de UXLab, está diseñado para soportar diferentes sensores fisiológicos y biométricos, ver Tabla 3, para registrar y guardar datos multimodales del usuario.

Tabla 3: Sensores fisiológicos y biométricos soportados por UXLab.

Fisiológicos	Biométricos	Computadora
Electroencefalograma (EEG)	Audio	Captura de la pantalla
Electrocardiograma (ECG)	Expresión facial	
Respuesta Galvánica en Piel (GSR)	Seguimiento ocular	

En la figura 15 se representa del proceso de captura de datos que permite grabar datos multimodales durante la interacción de un usuario con un producto digital.

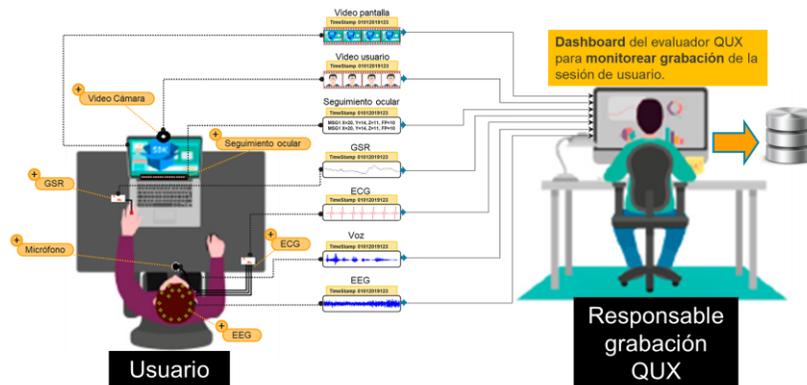


Figura 15: Proceso de registro de datos en UXLab. Fuente: UXLab Arquitectura (Diapositiva 7), de G. González, 2019, Todos los derechos reservados 2019. Tomado con permiso del autor.

El sistema de grabación de UXLab tiene configurados tres sensores fisiológicos (ver figura 16) de los siguientes fabricantes: Emotiv®, Seed Studio® y MySignals®, dos se conectan con una placa (Shield) para Arduino® y en función del sensor conectado se habilita la opción para seleccionarlo en la interfaz del sistema de grabación.

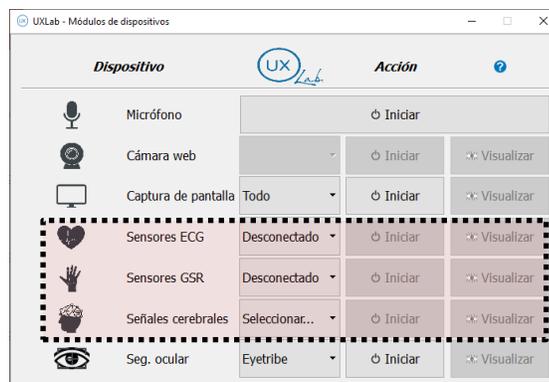


Figura 16: Control de sensores fisiológicos de UXLab. Se muestra al interior de la línea punteada los sensores fisiológicos en el software de UXLab.

Para verificar que los sensores fisiológicos y biométricos funcionen correctamente, UXLab permite el monitoreo en tiempo real de los sensores conectados, con el fin de detectar problemas de lectura antes de iniciar el proceso de grabación, ver figura 17.



Figura 17: Monitoreo en tiempo real de sensores en UXLab. Se muestra al interior de la línea punteada las opciones que habilitan el monitoreo de los sensores en UXLab.

El software UXLab recolecta y sincroniza los datos multimodales de una sesión de usuario, ver figura 18, posteriormente el responsable de monitorear esta sesión detiene la grabación y guarda los archivos de datos (texto, video y/o audio) en una ruta especificada en la parte superior de la ventana del integrador de grabación agrupando los archivos en carpetas con el *id* del usuario.

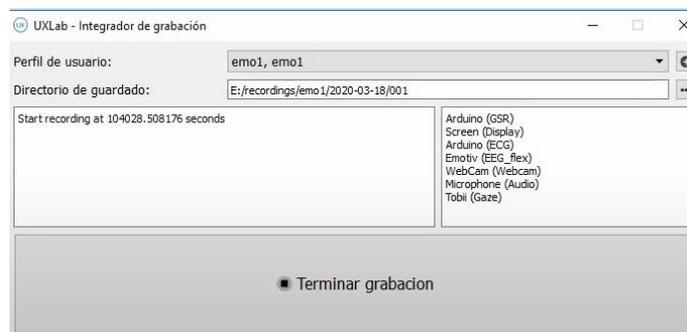


Figura 18: Grabación de datos de sensores en UXLab.

Al terminar la grabación del proceso de evaluación UX del producto digital, el encargado de monitorear el proceso debe verificar los datos registrados, los guarda y se inicia la etapa de postprocesamiento. Durante este proceso se utilizan los archivos de datos guardados en diferentes formatos: .txt, .csv, .wav y .mp4 (ver figura 19), para generar las métricas UX. Por ejemplo, con los datos de seguimiento ocular y de captura de pantalla, se generan videos que muestran las métricas relacionadas con la atención visual del

usuario, mapas de calor, mapas de rutas, áreas de interés; con los datos fisiológicos EEG, ECG y GSR se identifican estados mentales del usuario.

Nombre	Fecha de m...	Tipo	Tamaño
eyetracking_data	18/03/202...	Carpeta de archivos	
001blinkAnalysis	19/03/202...	Archivo de valores separados por comas de Microsoft Excel	1 KB
Audio_6	18/03/202...	Archivo WAV	61,399 KB
caracteristicas	18/03/202...	Archivo de valores separados por comas de Microsoft Excel	234 KB
ecg_002_emo4	18/03/202...	Archivo de valores separados por comas de Microsoft Excel	0 KB
eeg_002_emo4	18/03/202...	Archivo de valores separados por comas de Microsoft Excel	4,340 KB
events	18/03/202...	Documento de texto	1 KB
Gaze_7	18/03/202...	Archivo de valores separados por comas de Microsoft Excel	23 KB
gsr_002_emo4	18/03/202...	Archivo de valores separados por comas de Microsoft Excel	1,509 KB
resultados_labels	18/03/202...	Archivo de valores separados por comas de Microsoft Excel	1 KB
timestamp	18/03/202...	Archivo de valores separados por comas de Microsoft Excel	145 KB
Video_display	18/03/202...	Archivo MP4	30,262 KB
Video_webcam	18/03/202...	Archivo MP4	81,028 KB
VideoFull_AOI	19/03/202...	Archivo MP4	40,928 KB
VideoFull_Heatmap	19/03/202...	Archivo MP4	55,088 KB
VideoFull_Raw	19/03/202...	Archivo MP4	32,611 KB
VideoFull_Scanpath	19/03/202...	Archivo MP4	34,614 KB

Figura 19: Archivos generados en la etapa de grabación de UXLab.

El resultado de la etapa de postprocesamiento de los datos fisiológicos y biométricos genera archivos que son almacenamos en el repositorio gestionado por UXLab. Estos archivos se utilizan en la tercera etapa, que requiere la reproducción de la sesión de usuario con todos los datos multimodales guardados y procesados por el software Reproductor de UXLab.

3.3.2 Sistema Reproductor de UXLab

El reproductor de UXLab (ver figura 20) permite tomar los datos guardados de cualquier prueba UX grabada para mostrar los datos de los archivos gráficamente en una interfaz que permite visualizarlos de manera sincronizada en una línea de tiempo.

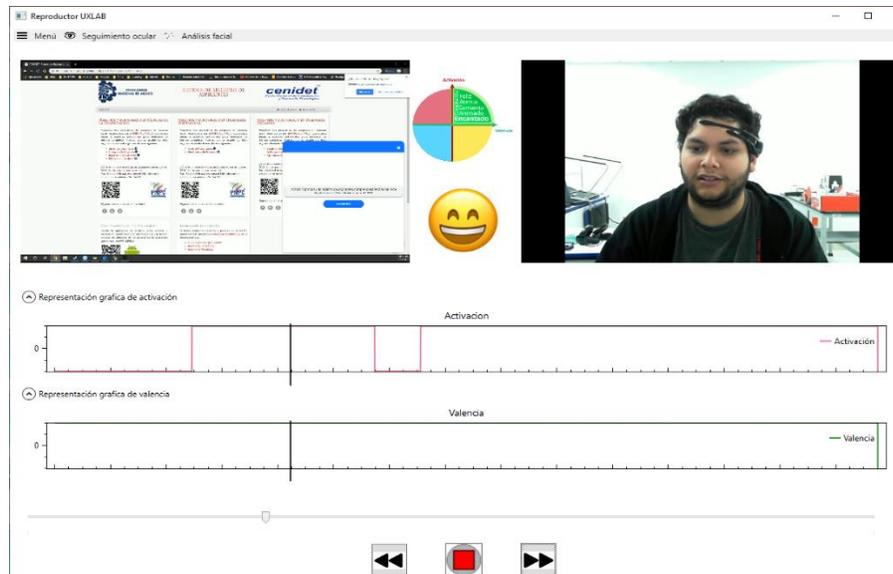


Figura 20: Interfaz gráfica del reproductor de UXLab.

El software lee todos los archivos disponibles en el directorio especificado durante la grabación, donde se mantiene una sesión de UX grabada. El reproductor mostrará en la interfaz los datos que hayan sido grabados (en caso de no haber grabado algún dispositivo multimedia o sensor, el programa mostrará vacío el lugar donde se representa esa información).

En la figura 21 se muestran los elementos de la interfaz de usuario del reproductor de UXLab, estos elementos son utilizados por expertos evaluadores de UX que analizan las pruebas para determinar la experiencia del usuario.

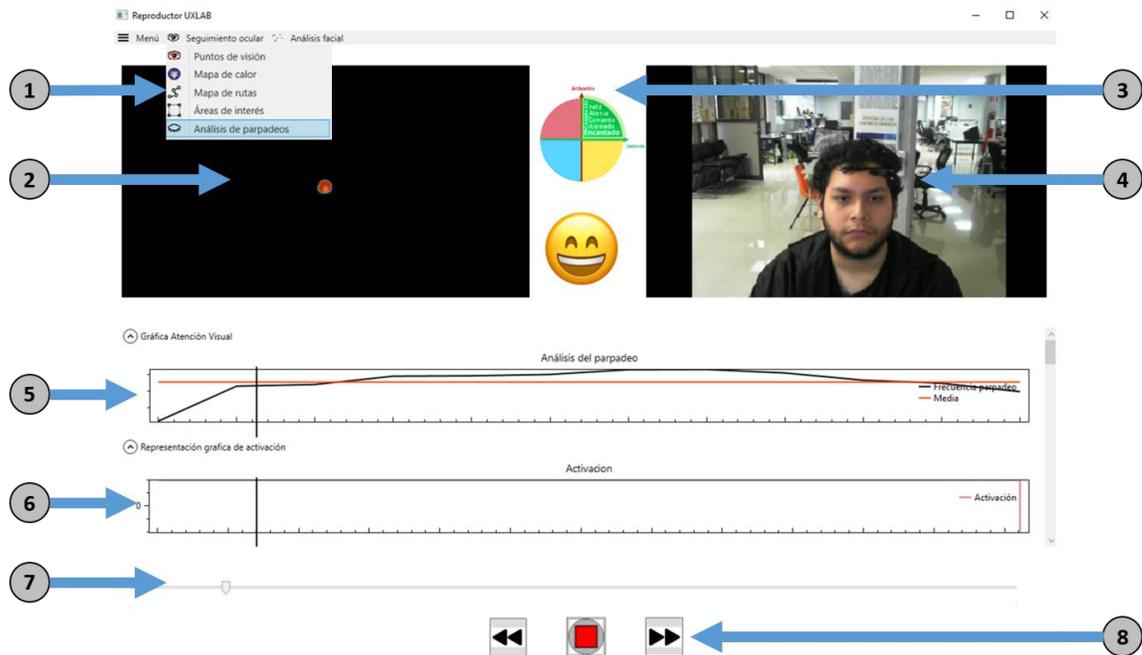


Figura 21: Elementos de la interfaz del reproductor UXLab. 1) Menú de métricas oculares, 2) Memoria de video del usuario durante su experiencia, 3) Indicadores emocionales, 4) Video del rostro del participante de la prueba, 5) Gráfica de atención visual utilizando la frecuencia de los parpadeos, 6) Gráficas de sensores y diadema EEG, 7) Línea de tiempo, indica el fragmento de tiempo que se está reproduciendo, 8) Botones de control, reproducir, adelantar, regresar.

El reproductor identifica los datos multimodales que se grabaron (debido a que se pueden grabar datos de diferentes sensores fisiológicos y biométricos), se tiene la opción de habilitar o deshabilitar la visualización para mostrar la información de cada sensor, por ejemplo: si en una prueba UX no se grabó video del rostro del usuario, cuando se reproduzca la prueba, la visualización del rostro se mostrará en un cuadro negro durante la reproducción.

Capítulo 4. Metodología de solución

De acuerdo con la metodología de solución que se planteó en esta tesis se desarrolló un algoritmo que relaciona los estados mentales del usuario con los datos del seguimiento ocular, el cual se incluyó en la plataforma de evaluación de la experiencia de usuario UXLab, este nuevo componente realiza la captura y análisis de datos de seguimiento ocular del usuario. Aplicando la siguiente metodología de solución en 4 fases (ver figura 22).

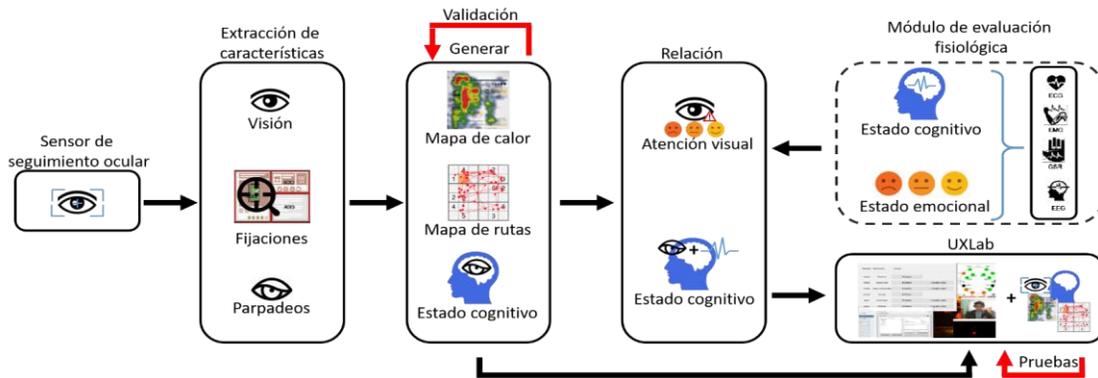


Figura 22: Esquema conceptual de la metodología de solución.

4.1 Fase 1: Extracción de características del sensor

En esta fase se obtuvieron los datos del sensor de seguimiento ocular que se generan durante una prueba de evaluación UX, estos datos son los puntos de visión, los puntos de fijación y los parpadeos que realizó el usuario. Cada una de estas variables se registran y almacenan como un conjunto de datos separados (coordenadas en x , coordenadas en y , tiempo, tipo de punto, entre otras variables) en un archivo en formato CSV para tener un registro de los datos de la prueba y posteriormente utilizarlos en la fase 2.

4.2 Fase 2: Validación y generación de métricas

En esta fase se utilizaron los archivos obtenidos en la fase 1 para generar los mapas de atención (mapas de calor y mapas de rutas), además con los datos del parpadeo del usuario, se puede identificar el estado cognitivo (nivel de atención) relacionando su frecuencia de parpadeo durante la prueba UX.

Para validar la generación de los mapas de atención se realizó una comparación entre mapas generados en diferentes pruebas con el software UXLab donde se comparan con la reacción de los usuarios ante diferentes estímulos.

Utilizando la frecuencia de los parpadeos, se puede inferir el nivel de atención del usuario, este nivel es inversamente proporcional a la frecuencia de parpadeos por minuto, la atención aumenta cuando la frecuencia de parpadeos por minuto disminuye. Una frecuencia muy alta indica falta de atención, aburrimiento y adormecimiento. Para comprobar esta correlación entre los movimientos oculares (parpadeos) y el estado cognitivo (nivel de interés) se desarrollaron pruebas y experimentos.

4.3 Fase 3: Relación entre comportamiento visual y estado cognitivo

En la fase de relación se preparó la información para que el usuario de UXLab pueda interpretar los mapas de atención y la frecuencia de parpadeo de la fase 2 con los resultados del módulo de evaluación fisiológica, este último determina el estado emocional del usuario con base a sensores fisiológicos. Al combinar los dos procesos de análisis de los estados mentales (seguimiento ocular y fisiológico), se puede mejorar la precisión del resultado y correlacionar las acciones del usuario.

Para realizar esto se desarrolló el módulo que presenta las métricas del seguimiento ocular en la interfaz del software UXLab. Con esto, el usuario se apoya de la visualización de los datos oculares de la prueba de evaluación UX en tiempo real sobre la interfaz. Los mapas de calor, los mapas de rutas y la frecuencia de parpadeo se sobreponen sobre el video registrado en la prueba de evaluación UX (de forma opcional para el usuario) y cambian de acuerdo con la escala de tiempo del video, la atención visual del usuario en un punto de tiempo se visualiza en conjunto con la emoción sentida en el mismo lapso.

4.4 Fase 4: Implementación a UXLab y pruebas

En la fase 4 se integraron todos los módulos en el software UXLab para que la plataforma puede complementar los reportes de evaluación UX con información más precisa y objetiva sobre el comportamiento de los usuarios que realizan las evaluaciones UX de productos digitales, utilizando información de la atención visual.

El procesamiento de la información obtenida con el sensor de seguimiento ocular se realiza después de haber completado la grabación de una prueba de evaluación UX ya que la generación de archivos no permite su procesamiento en tiempo real.

Una vez realizada la implementación del módulo de seguimiento ocular en la plataforma se realizaron pruebas funcionales del software y de integración con la plataforma.

4.5 Algoritmo de extracción de datos del sensor de seguimiento ocular

En la etapa inicial de esta tesis se desarrolló un algoritmo para la extracción de métricas del seguimiento ocular, utilizando un kit de desarrollo de software (SDK) en lenguaje C# el cual cuenta con licencia gratuita. El SDK contiene el código necesario para comunicarse con el sensor conectado en la computadora y leer la información que éste proporciona, entre estos datos se encuentran los puntos de visión y de fijación.

El SDK proporciona un método para obtener en tiempo real los puntos de fijación y de visión, los cuales se pueden representar como un objeto que contiene 3 variables:

- La coordenada en X
- La coordenada en Y
- Una estampa de tiempo

Además, los eventos de fijación del sensor se dividen en 3 tipos:

- Inicio: Se marca un evento de fijación con el tipo *Begin* cuando es el inicio de la fijación del usuario.
- Datos: Se marcan con el tipo *Data* las líneas de captura del sensor que ocurren durante la fijación del usuario. La cantidad de líneas de este tipo dependen de cuanto dure la fijación.
- Fin: Se marca un evento con el tipo *End* cuando el sensor detecta que el usuario ha dejado de observar/fijar la mirada.

Para obtener los datos (puntos de fijación, visión y parpadeos) de este sensor se desarrolló un programa que utiliza hilos para poder obtener en tiempo real y en paralelo los diferentes datos disponibles.

El programa, desarrollado en C#, utiliza el código del SDK para obtener los puntos de visión y parpadeo en hilos separados, además utiliza los datos de visión para calcular

tiempos muertos entre las lecturas, los puntos que no se registran lecturas de visión son filtrados mediante condiciones de tiempo para reconocer los parpadeos.

En total se utilizaron dos hilos diferentes, ejecutados en paralelo para obtener estos datos, cada hilo registra y almacena los datos en un archivo de texto, los tres hilos utilizan una estampa de tiempo global definida al principio de la ejecución del programa.

4.5.1 Extracción de datos de visión

La figura 23 muestra el pseudocódigo para obtener los puntos de visión del sensor de seguimiento ocular. Se define la función *obtenerPuntosVisión* la cual recibe una lista de las lecturas del sensor de seguimiento ocular (línea 1), posteriormente se define una nueva lista que contendrá la estructura de datos de puntos de visión (línea 2). Se utiliza un ciclo para obtener de la lista de lecturas las coordenadas (x , y) con la estampa de tiempo, estos datos son agregados a la lista de puntos (líneas 3-8). Finalmente, la función devuelve la lista de Puntos para ser utilizada (línea 9).

```

1  Función obtenerPuntosVisión(lecturas):
2  |   listaPuntos = []
3  |   mientras(sensor.siguietePunto() != vacio):
4  |       x = sensor.siguietePunto().coordenadaX
5  |       y = sensor.siguietePunto().coordenadaY
6  |       tiempo = sensor.siguietePunto().tiempo
7  |       listaPuntos.agregar([x, y, tiempo])
8  |   fin mientras
9  |   regresa listaPuntos
10 fin función

```

Figura 23: Pseudocódigo para obtención de puntos de visión.

El código formatea las variables del flujo continuo de datos registrados por el sensor y las agrega a una lista de datos de visión, posteriormente la lista se utiliza para escribir el archivo de texto en formato CSV, a continuación, se muestra la estructura de los datos contenidos en el archivo de visión (ver Figura 24):

- Punto en X
- Punto en Y

- Estampa de tiempo en formato “*hh:mm:ss:ffff*” (Hora, Minuto, Segundo, Milisegundo).

	A	B	C
1	X Gaze Data	Y Gaze Data	Timestamp
2	336	692	00:00:00:0250
3	336	693	00:00:00:0339
4	337	693	00:00:00:0460

Figura 24: Ejemplo de archivo de puntos de visión.

4.5.2 Extracción de datos de fijaciones

Para obtener las fijaciones se hace uso de los tipos de eventos capturados (Inicio, Datos y Fin), estos definen el inicio, la duración y el fin de una fijación. La figura 25 muestra el pseudocódigo para obtener las fijaciones del sensor de seguimiento ocular.

La primera línea define la función, esta recibe una lista de las lecturas del sensor de seguimiento ocular (línea 1), posteriormente se definen una serie de lista que se usaran como auxiliares y dos variables de tiempo (líneas 2-4). La lógica de la función comienza con un ciclo *para cada* (foreach), este tipo de ciclo toma cada elemento de la lista de lecturas hasta que se recorran todos los elementos, la función revisa si los eventos de las tuplas son de tipo *begin*, *data* o *end* dependiendo del tipo de evento la función separa la información de los puntos de fijación en las listas auxiliares (líneas 5-23). Cada vez que se detecta un evento de tipo *end* se obtiene el promedio de los puntos de fijación almacenados en las listas auxiliares para definir el centro de la fijación y también se usan las variables de tiempo para calcular el inicio, el fin y la duración de la fijación (líneas 6-17). Finalmente, cuando todas las fijaciones son definidas, la función retorna la lista de fijaciones para ser utilizada en otros procesos (línea 24).

```

1  Función obtenerFijaciones(lecturas):
2  |   lista = [], puntosX = [], puntosY = [], fijaciones = []
3  |   inicio = tiempo.ahora()
4  |   fin = tiempo.ahora()
5  |   para cada lectura en lecturas:
6  |       si (lectura.evento = "Fin"):
7  |           lista.agregar(lectura)
8  |           fin = lectura.tiempoFin
9  |           duración = (fin - inicio)
10 |           para cada fix en lista:
11 |               puntosX.agrega(fix.coordenadaX)
12 |               puntosY.agrega(fix.coordenadaY)
13 |           fin para cada
14 |           X = promedio(puntosX)
15 |           Y = promedio(puntosY)
16 |           fijaciones.agregar([X, Y, duracion, inicio, fin])
17 |           reiniciarListas(lista, puntos, puntosY)
18 |       si no:
19 |           lista.agregar(lectura)
20 |           si (lectura.evento = "Inicio"):
21 |               inicio = lectura.tiempoInicio
22 |       fin si
23 |   fin para cada
24 |   retorna fijaciones
25 | fin función

```

Figura 25: Pseudocódigo para obtención de puntos de fijación.

Al igual que con la obtención de puntos de visión, el código formatea las variables del flujo continuo de datos devueltos del sensor y las agrega a una lista de datos, posteriormente la lista se utiliza para escribir el archivo de texto en formato CSV, a continuación, se muestra la estructura de los datos contenidos en el archivo de fijaciones (ver Figura 26):

- Tipo de evento (Inicio, Datos, Fin)
- Punto en *X*
- Punto en *Y*
- Estampa de tiempo en formato “*hh:mm:ss:ffff*” (Hora, Minuto, Segundo, Milisegundo).

	A	B	C	D
1	Event	X Fixation Data	Y Fixation Data	Timestamp
2	Data	386	375	00:00:00:0489
3	Data	368	481	00:00:00:0580
4	Data	357	550	00:00:00:0680

Figura 26: Ejemplo de archivo de puntos de fijación.

4.5.3 Algoritmo de extracción de parpadeos

El algoritmo que extrae los parpadeos se ilustra con la figura 27. Esta función recibe como parámetro la lista de puntos de visión de la sección 4.5.1 (línea 1), se utiliza una función *para cada* con el fin de recorrer cada elemento de la lista (línea 3). La lógica de esta función es comparar la diferencia en el tiempo del punto actual del ciclo con el siguiente (líneas 4-5), si existe una diferencia en tiempo mayor a 100 milisegundos y menor a 400 milisegundos, se agrega un parpadeo a la lista junto a su estampa de tiempo (líneas 6-9). Finalmente se devuelve la lista de parpadeos (línea 11).

```

1  Función obtenerParpadeos(puntosVisión):
2  |   siguientePunto = [], parpadeos = 0, listaParpadeos = []
3  |   para cada punto en puntosVisión:
4  |       siguientePunto = puntosVision.next()
5  |       diferencia = siguientePunto.tiempo() - punto.tiempo()
6  |       si (diferencia > 100 yy diferencia < 400 ):
7  |           |   parpadeos = parpadeos + 1
8  |           |   listaParpadeos.Add(parpadeos + "," + tiempoActualPrueba);
9  |           |   fin si
10 |   fin para cada
11  |   retorna listaParpadeos

```

Figura 27: Pseudocódigo para obtención de puntos de fijación.

El algoritmo compara la diferencia de tiempo entre la captura de un punto de visión y el siguiente, el tiempo de duración normal de un parpadeo es de 100ms a 400 ms (Schiffman, 2001). Se trata de identificar el tiempo muerto donde el sensor no está leyendo datos oculares (debido a que se bloquea la lectura cuando el parpadeo del usuario se cierra).

La lista de parpadeos resultante se utiliza para escribir el archivo de texto en formato CSV, a continuación, se muestra la estructura de los datos contenidos en el archivo de parpadeos (ver Figura 28):

- Número de parpadeo
- Estampa de tiempo en formato “*hh:mm:ss:ffff*” (Hora, Minuto, Segundo, Milisegundo).

	A	B
1	Blink #	Timestamp
2	1	00:00:53:5795
3	2	00:01:19:6397
4	3	00:01:19:9387

Figura 28: Ejemplo de archivo de puntos de parpadeos.

Posteriormente se calcula la frecuencia de parpadeo cada minuto utilizando la siguiente formula (5):

$$frecuencia = \frac{\sum \text{parpadeos en un minuto}}{\text{minutos totales}} \quad (5)$$

Donde:

- $\sum \text{parpadeos en un minuto}$, es la sumatoria de todos los parpadeos que se presenten en cada minuto durante la prueba.
- minutos totales , es el número de minutos que dura la prueba.
- Por ejemplo, si el usuario parpadeo 5 veces en el primer minuto y 7 en el segundo entonces la frecuencia seria = $(5 + 7) / 2$.

4.5.3.1 Precisión del algoritmo de extracción de parpadeos

Se comparó la precisión de la captura de parpadeos de la función de extracción de datos del sensor haciendo un conteo “manual” de los parpadeos en los videos de una muestra de seis pruebas de captura de datos oculares grabadas con una cámara web. La idea fue observar cuántos parpadeos captura la función contra los parpadeos que en verdad ocurren.

Para realizar esta prueba se tuvo una muestra de seis personas, a las cuales se les pidió ver dos videos, con una duración aproximada de tres minutos cada uno; durante el tiempo de exposición de los videos se registró y grabó el movimiento ocular de casa usuario y se grabó video de su rostro (ver Figura 26), una vez finalizadas las pruebas se calculó automáticamente el número de parpadeos y se comparó con el conteo manual observado en el video de cada usuario.

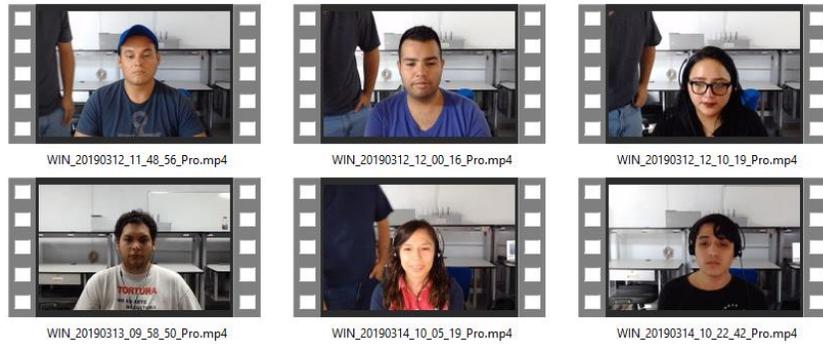


Figura 29: Videos capturados de cada participante.

A continuación, en la Tabla 4, se muestran los resultados obtenidos:

Tabla 4: Pruebas de eficiencia de extracción de parpadeos con EyeWidget.

Participante	Conteo manual (Parpadeos)	Captura del Sensor (Parpadeos)	Porcentaje de capturas efectivas
1	18	18	100.00%
2	5	5	100.00%
3	27	24	88.88%
4	11	10	90.90%
5	15	15	100%
6	10	8	80.00%
Promedio/Eficacia:			93.29%

En promedio el método de captura de los parpadeos en el software desarrollado tiene una eficacia del 93.29%, este porcentaje de precisión se consideró aceptable ya que sobrepasa el 90%.

4.5.4 Herramienta de extracción de datos EyeWidget

Antes de implementar los algoritmos de extracción de datos oculares en una interfaz, se realizó una investigación de herramientas software similares, por ejemplo, se encontró la herramienta PyGaze Analyzer (Dalmaijer, 2013). PyGaze Analyzer es una suite de herramientas de software libre, escritas en Python, para procesamiento de datos de seguimiento ocular. Con estas herramientas se pueden leer conjuntos de datos para generar

mapas de atención sobre imágenes y animaciones que permiten visualizar los datos del seguimiento ocular (Mapas de calor, mapas de rutas sacádicas, áreas de interés) utilizando la librería *matplotlib*. PyGaze fue considerado para adaptar los algoritmos de extracción de datos oculares, sin embargo, el programa resultó ser muy complicado de implementar con el SDK que proporciona el sensor de seguimiento ocular por lo que se decidió buscar otra alternativa.

Después de la búsqueda se encontró la herramienta EyeWidget, desarrollada por (King, 2017), como se muestra en la figura 30, este software se está estructurado en 3 capas. La primera capa (de abajo hacia arriba) se encarga de capturar la información del sensor de seguimiento ocular por medio de un SDK para lenguaje C# y estructura la información obtenida para ser mostrada posteriormente. La capa de métricas se encarga de la interpretación de los datos en la primera capa, en este caso EyeWidget solo presenta las métricas de seguimiento ocular y presencia del usuario. Finalmente la capa de visualización hace uso de las ventanas WPF (Windows Presentation Foundation, por sus siglas en inglés) para crear una interfaz gráfica para visualizar la información capturada, en tiempo real, las métricas de la segunda capa, se muestra la posición de la vista del usuario en la pantalla de la computadora (en coordenadas x, y) y muestra si el sensor está siguiendo la mirada, si el usuario está presente y el tiempo que lleva corriendo el programa. EyeWidget resultó ser compatible con la interfaz hardware utilizada en este trabajo ya que la implementación de UXLab también utiliza ventanas WPF.

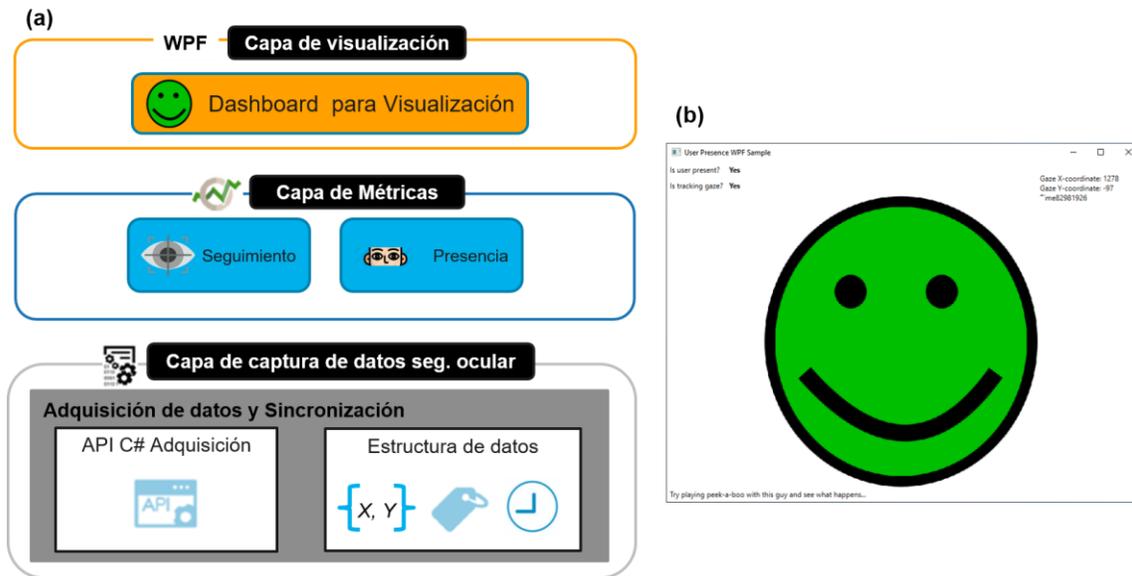


Figura 30: EyeWidget original, (a) Estructura de 3 capas de la aplicación EyeWidget original, (b) Interfaz del programa EyeWidget original.

El autor de EyeWidget puso a disposición el código fuente en un repositorio web público, sin embargo, como el autor no adjuntó una licencia de uso, se solicitó su permiso vía correo electrónico, para hacer uso de este código como base de los algoritmos de extracción de datos. Al obtener una respuesta positiva por parte del autor se pudo utilizar y modificar el código sin restricciones.

La figura 31 muestra la arquitectura del programa EyeWidget modificado en 4 capas, la primera capa (de abajo hacia arriba) se encarga de extraer los datos del sensor y de estructurarlos, también se implementó el almacenamiento de esta información en archivos de texto formato CSV (valores separados por comas). La segunda capa se implementó para procesar los datos provenientes de la capa de captura, este procesamiento genera los datos de puntos de visión, puntos de fijación y de parpadeos. En la capa de métricas se puede observar la interpretación de datos del software que dan información al usuario sobre el seguimiento ocular en tiempo real, de la presencia del usuario, la captura de parpadeos y la posición de los ojos del usuario con relación a la pantalla. Por último, en la capa de visualización se tiene la interfaz mejorada de EyeWidget la cual permite visualizar las métricas y contiene elementos de control para realizar grabaciones de seguimiento ocular.

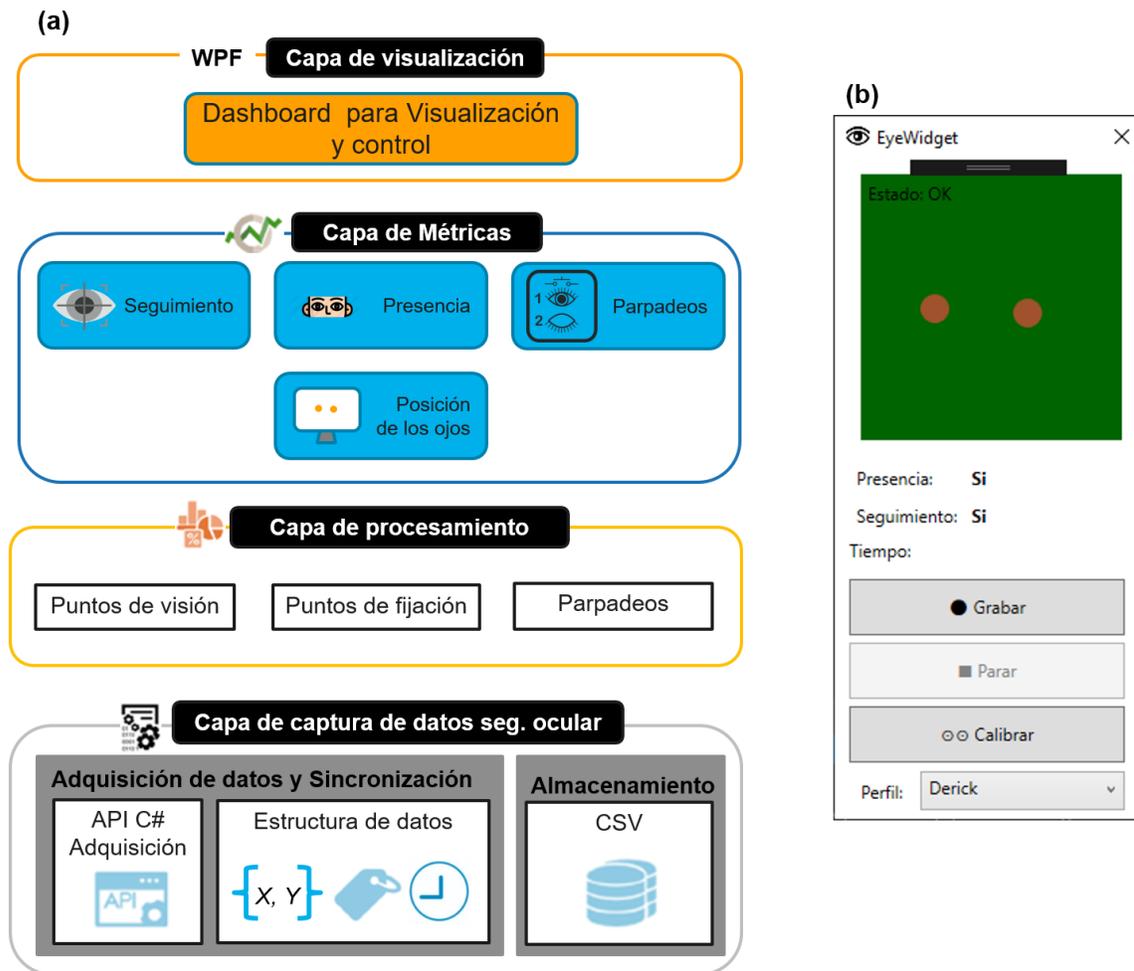


Figura 31: EyeWidget modificado, (a) Estructura de 4 capas de la aplicación EyeWidget modificado, (b) Interfaz del programa EyeWidget modificado.

A continuación, se listan las modificaciones que extienden la funcionalidad del programa original:

- Se cambió el tamaño de la interfaz a una más pequeña (con el fin de adaptarse mejor a la información que se muestra en un contexto donde se realizan pruebas de evaluación UX).
- Se eliminó la salida de coordenadas en tiempo real, ya que no era necesario.
- Se agregó la obtención de los puntos de fijación y visión implementando los algoritmos de extracción de datos.

- Se agregó una visualización en tiempo real de la posición de los ojos del usuario en relación con la pantalla.
- Se agregó un botón para iniciar la grabación de los datos.
- Se agregó un botón para detener la ejecución del programa.
- Se agregó un botón para lanzar el asistente de calibración del sensor.
- Se agregó una lista para seleccionar diferentes perfiles de calibración del sensor.
- Se incluyó la funcionalidad de conteo y obtención de la frecuencia de parpadeo.

Básicamente se tomaron los algoritmos de extracción de métricas y se desarrollaron funciones adicionales que ayudan a realizar grabaciones en tiempo real y se adaptaron a la interfaz WPF de EyeWidget.

En la figura 32 se describe el funcionamiento de la herramienta EyeWidget, utilizando la conexión entre el sensor y la computadora por medio de los controladores y el API en C#, se obtienen las muestras de datos más actuales del flujo de información que registra el sensor de seguimiento ocular. Se itera cada muestra de datos oculares y es etiquetada con la estampa de tiempo obtenida de un contador definido en 0 al inicio del proceso.

Posteriormente el programa divide el procesamiento en 2 hilos diferentes, donde se procesan las muestras de datos para definir el conjunto de donde corresponden las muestras: puntos de visión o puntos de fijación. Dentro del primer hilo, se utilizan las estampas de tiempo de los puntos de visión para que, mediante un algoritmo, que se describe más a detalle en la sección 4.6.5, se procesan los parpadeos capturados.

Finalmente, luego de haber terminado de estructurar los datos, estos son agregados a un registro de salida que corresponde al tipo de dato procesado (puntos de visión, puntos de fijación o parpadeos), los registros se recuperan y se almacenan en archivos CSV que serán utilizados posteriormente en otra etapa del procesamiento.

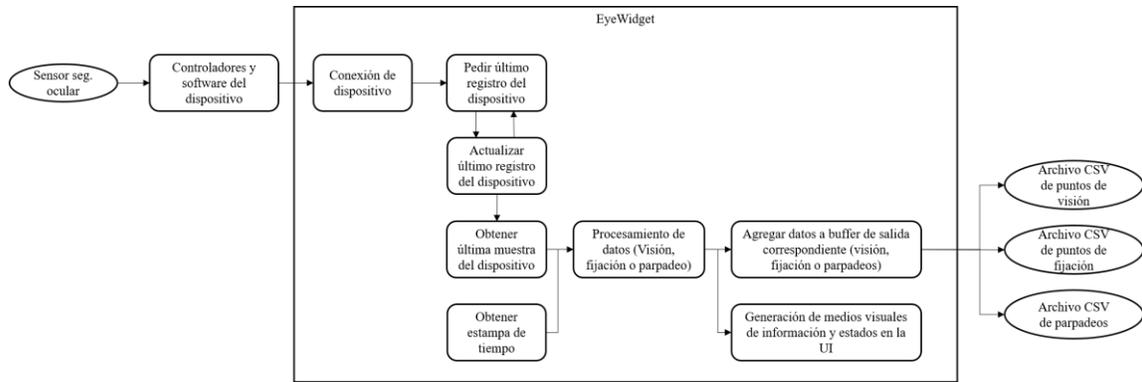


Figura 32: Esquema del proceso computacional de EyeWidget modificado para extraer datos de visión, fijaciones y parpadeo.

Cabe mencionar que posterior a la primera versión se modificó la interfaz de usuario de esta herramienta para incluirla en el dashboard de grabación de UXLab. De igual manera se conservó el nombre EyeWidget de la herramienta para mantener los derechos del autor del software que permite la grabación de los datos del seguimiento ocular.

4.6 Algoritmos de generación de métricas del seguimiento ocular

Para generar las métricas visuales se trabajó con los archivos de texto mencionados en la sección 4.3.2 en combinación con videos que fueron grabados al mismo tiempo que los datos oculares. Se utilizaron diferentes librerías y procesos para cada una de estas métricas, las cuales fueron integradas al código de UXLab para generarlas automáticamente después de una grabación de evaluación UX.

4.6.1 Video de puntos de visión

Los videos de puntos utilizan el conjunto de datos del archivo CSV que contiene los puntos de visión. En esta métrica se dibujan los puntos grabados sobre el video que representa la pantalla que observó el usuario durante la evaluación UX (representando un plano de dos dimensiones x, y).

Para dibujar estos puntos en el video se utilizó el lenguaje de programación Python y la librería de OpenCV (Bradski, 2000). OpenCV (Open Source Computer Vision Library) es una librería de código abierto para aplicaciones de visión artificial y aprendizaje

automático, contiene muchas herramientas y algoritmos optimizados que permiten la manipulación de imágenes y videos. En este caso, se utilizó la funcionalidad de OpenCV que permite trabajar cada fotograma de un video utilizando la función *cv2.videoCapture*. Al tener cada fotograma disponible se creó una función que lee cada punto contenido en el archivo CSV y lo dibuja sobre el fotograma del video que corresponde con su estampa de tiempo.

La figura 33 muestra el proceso de generación de videos, el cual utiliza como entrada un archivo de puntos de visión en formato CSV y un video (El archivo de video debe ser tomado al mismo tiempo que se grabó el seguimiento ocular para que exista una correspondencia entre los eventos registrados y el comportamiento ocular del usuario).

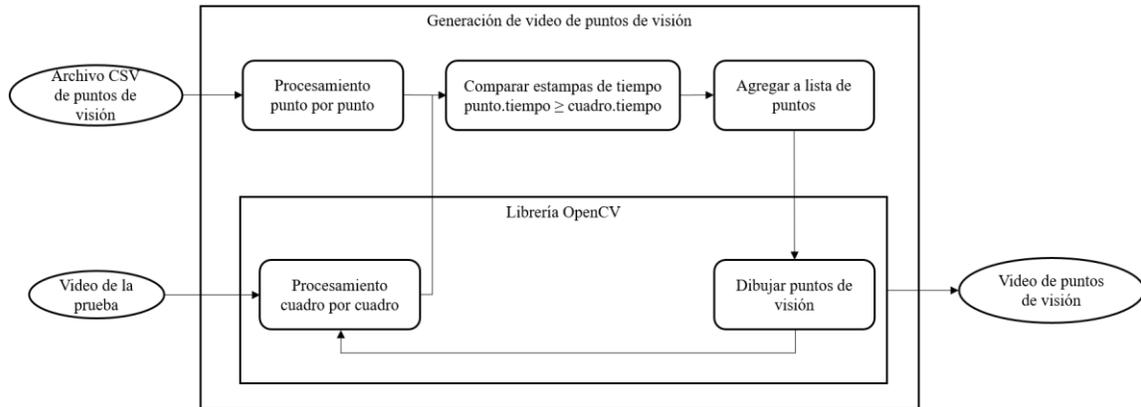


Figura 33: Esquema del proceso de generación de videos de puntos.

Tanto los puntos de visión como los fotogramas del video cuentan con estampas de tiempo, en el caso de los datos oculares indican el momento en el que se realizó el punto de visión, mientras que, en el caso de los fotogramas de video, indican la marca de tiempo. Por cada punto de visión y fotograma, las estampas de tiempo se procesan para realizar una comparación, si el tiempo del punto es mayor o igual al del fotograma entonces ese punto debe ser representado en el fotograma.

Los puntos que cumplen la condición se agregan a una lista de puntos que son dibujados posteriormente, se sobreponen elementos gráficos en forma de círculo sobre el fotograma que se está iterando. El fotograma del video representa un plano espacial de coordenadas *x*, *y* (por ejemplo, de 1200 x 720 pixeles) la función dibuja la figura en la posición del plano donde existen las coordenadas del punto de visión. La función *cv.circle* permite

especificar el tamaño del círculo definido por el radio, el color definido por un arreglo de valores y el tipo de relleno.

Finalmente, cuando se han iterado todos los puntos de visión y todos los fotogramas de video, se recupera el conjunto de fotogramas donde se dibujaron los puntos de visión para crear el video en formato *mp4*.

En el caso específico de los videos de puntos se puede generar la representación visual de todo el conjunto de puntos capturados, sin embargo, cuando el número de puntos es elevado se puede llegar a confundir el rastro de estos puntos sobre el video. Por este motivo el algoritmo limita los puntos en pantalla para que solo se muestren 10 puntos por cada fotograma del video para facilitar la interpretación del comportamiento ocular. La figura 34 muestra el pseudocódigo utilizado para realizar los videos de puntos.

La función recibe como parámetro de entrada el archivo CSV de puntos de visión tomados de una grabación de seguimiento ocular (línea 1), la variable *listaPuntos* se utiliza para leer y guardar el conjunto de puntos de visión del archivo de datos (línea 2), la variable *fotogramas* guarda el conjunto de fotogramas del video de la prueba y la variable *listaDibujo* es utilizada como auxiliar para guardar el conjunto de puntos que serán dibujados sobre el video (líneas 3-4). Se utilizan dos ciclos iterativos para manejar cada fotograma del video y luego cada punto de visión (líneas 5-6), dentro de los ciclos se comparan las estampas de tiempo de cada fotograma con cada punto de visión, si el punto de visión se encuentra antes de la estampa de tiempo del fotograma entonces se agrega a la variable *listaDibujo* (líneas 7-9), como se mencionó anteriormente, este algoritmo limita la cantidad de puntos en *listaDibujo* para no saturar el fotograma con muchos puntos, se limita la lista a 10 puntos (líneas 11-13). Se dibujan los puntos de *listaDibujo* sobre el fotograma actual (línea 14) y finalmente al terminar de iterar los puntos y fotogramas se retorna el video de la prueba con los fotogramas que incluyen los puntos de visión.

```

1  Función videoPuntos(csvPuntos):
2  | listaPuntos = leer(csvPuntos)
3  | fotogramas = cv2.videoCapture('ruta al video de la prueba')
4  | listaDibujo = []
5  | para cada fotograma en fotogramas:
6  |   | para cada punto en listaPuntos:
7  |   |   | si punto.tiempo < fotograma.tiempo:
8  |   |   |   | listaDibujo.agregar(cv.circle(fotograma,(punto.x,punto.y),1,(0,255,0),1))
9  |   |   |   | fin si
10 |   |   | fin para cada
11 |   | mientras listaDibujo.longitud > 10:
12 |   |   | listaDibujo.cut() // limita la lista a 10 elementos
13 |   |   | fin mientras
14 |   |   | dibujar(listaDibujo)
15 |   | fin para cada
16 |   | fin para cada
17 |   | fin función

```

Figura 34: Pseudocódigo para la generación de videos de puntos.

La figura 34 muestra un ejemplo de una imagen procesada de esta forma sin limitar el número de puntos en la pantalla. Cabe mencionar que los videos generados con OpenCV se guardan en formato mp4.

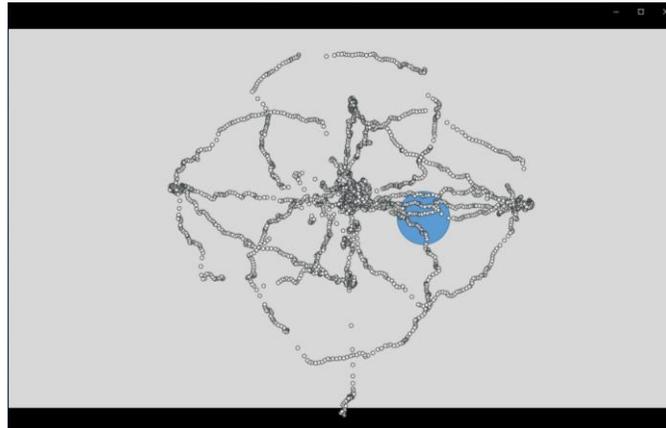


Figura 35: Ejemplo de video de puntos procesado sobre una prueba de seguimiento ocular.

4.6.2 Mapas de calor

El algoritmo, descrito por la figura 36, muestra el proceso computacional de generación de videos de mapas de calor, se necesita realizar una grabación de los datos de fijación y memoria de video para obtener las entradas del proceso.

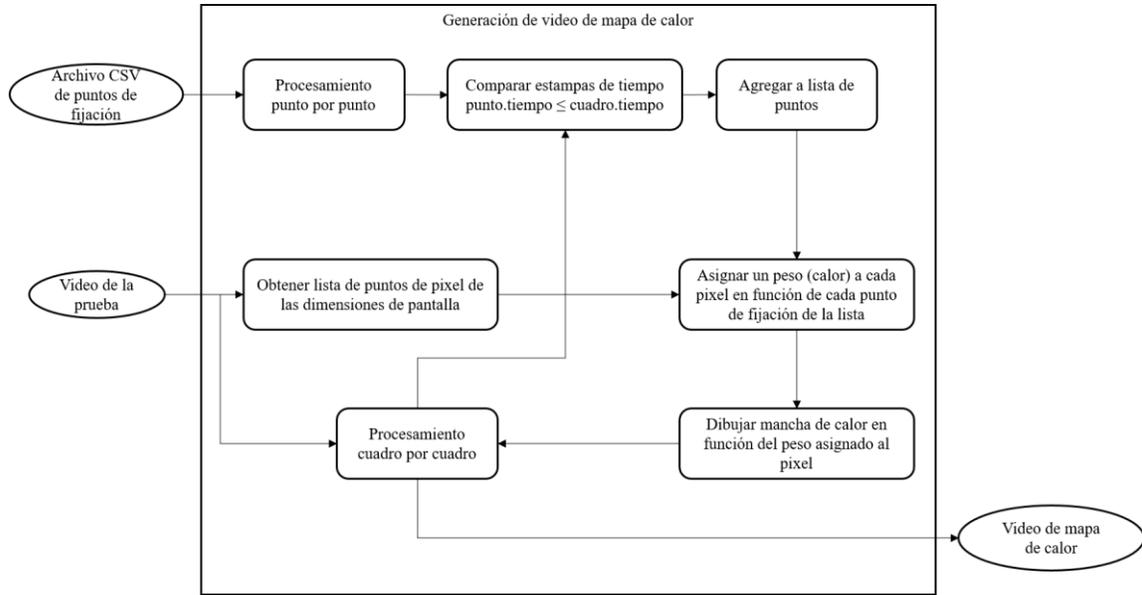


Figura 36: Esquema del proceso de generación de videos de mapa de calor.

El algoritmo itera los fotogramas del video para utilizar las estampas de tiempo en una comparación con el tiempo donde ocurrieron las fijaciones, de este modo se obtiene una lista de puntos de fijación que corresponden a cada fotograma del video. Al mismo tiempo se obtienen las dimensiones en pixeles del video, para obtener así el plano espacial de coordenadas x , y a los que se ligaran las fijaciones como manchas de calor. Una vez que se obtiene la lista de puntos de fijación y de pixeles en cada fotograma, estos últimos se iteran y se calcula el peso de cada píxel en función de la distancia o diferencia entre el píxel y la fijación utilizando un kernel gaussiano (3).

$$intensidad\ pixel(i, j) = exp - \frac{(x-i)^2 + (y-j)^2}{2\alpha^2} \quad (3)$$

Donde:

- x es un punto capturado en el eje x del plano espacial de la pantalla.
- y es un punto capturado en el eje y del plano espacial de la pantalla.
- i es un punto de un píxel en el eje x del plano espacial de la pantalla.
- j es un punto de un píxel en el eje y del plano espacial de la pantalla.
- α es la desviación estándar del conjunto de datos capturados, se puede modificar el valor para variar la intensidad.
- La función es iterativa se calcula para cada píxel (i, j) disponible en la pantalla.

Finalmente se dibuja sobre el fotograma la mancha de calor que representa el valor del peso del píxel en colores que representan calor desde azul (frio) a rojo (caliente).

En este trabajo, los mapas de calor son generados con la ayuda de una librería de Python llamada *Heatmappy* (Lumen Research, 2019), la librería funciona con el mismo principio de generación de mapas de calor (figura 36) pero facilita y optimiza la creación del video con el mapa, esta librería trabaja sobre objetos de imágenes creadas con la librería PIL (Python Imaging Library), la cual otorga herramientas para manipulación de imágenes en Python y videos en formatos mp4 y avi.

Heatmappy contiene funciones que reciben y tratan arreglos de puntos de visión en un formato (x, y, *tiempo*), *Heatmappy* toma este arreglo de puntos y dibuja los mapas de calor sobre los videos. La función trabaja con los siguientes elementos: un arreglo de puntos de visión, la ruta hacia el archivo de video, la intensidad de las marcas de calor, el tipo de mapa de calor (colores) y la ruta del archivo de salida generado.

Cabe mencionar que, debido a la complejidad de realizar la comparación de cada punto de visión con el resto de los puntos en toda la prueba, el proceso de generación de mapas de calor requiere alto costo computacional. Por ejemplo, realizar el mapa de calor de una prueba con duración de 10 minutos, el procesamiento de ésta tarda aproximadamente entre 50 y 60 minutos.

La figura 37 muestra el seudocódigo que describe la generación de los mapas de calor sobre una imagen, La función recibe los datos del archivo CSV de puntos (línea 1), se definen las variables que contienen la lista de puntos (línea 2), la imagen donde se colocaran las manchas de calor (línea 3), una lista auxiliar donde se guardaran las manchas de calor para ser dibujadas posteriormente (línea 4) y el valor de alfa que será utilizado (línea 5), nótese que alfa es un valor variable que afecta la intensidad del valor de las manchas de calor. Luego se itera cada punto disponible en la variable *listaPuntos* (línea 6), cada uno de estos puntos debe ser comparado con cada píxel en la imagen para determinar el valor de calor del píxel, por lo que se define otro ciclo iterativo dentro del primero para trabajar cada píxel de la imagen (línea 7). Se utiliza la formula (3) para calcular y asignar un valor a cada píxel y se agrega a la variable *listaCalor* (línea 8), después de haber iterado cada píxel de la imagen, se realiza la sumatoria del valor de cada uno con el siguiente punto de fijación de la lista hasta haber iterado cada punto de fijación

disponible. Posteriormente se utilizan funciones de dibujo para colocar las manchas de calor sobre la imagen, y asignar un color dependiendo de su valor (línea 13), finalmente se retorna la imagen con el mapa de calor generado (línea 14).

```

1  Función mapaCalor(csvPuntos):
2      listaPuntos = leer(csvPuntos)
3      imagen = abrir('ruta a la imagen de la prueba')
4      listaCalor = []
5      alfa = 1 // valor variable que aumenta o reduce la intensidad de calor de cada punto
6      para cada punto en listaPuntos:
7          para cada pixel en imagen:
8              listaCalor.agregar(exp(-((punto.x-pixel.x)^2 + (punto.y-pixel.y)^2) / (2*alfa)^2 )
9              // cada punto le será asignado un peso dependiendo del valor del peso será
10             // la intensidad del color
11         fin para cada
12     fin para cada
13     dibujar(listaDibujo)
14     retorna imagen
15 fin función
    
```

Figura 37: Pseudocódigo para la generación de mapas de calor.

La figura 38 muestra un ejemplo de mapa de calor generado mediante el algoritmo:

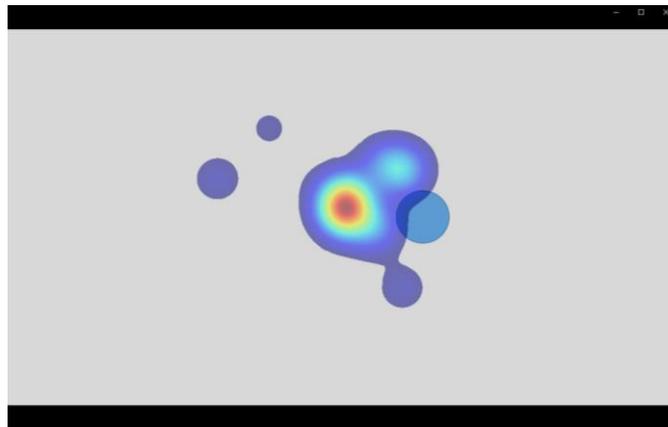


Figura 38: Ejemplo de mapa de calor procesado sobre una prueba de seguimiento ocular.

Los mapas de calor ayudan a identificar las áreas donde se encuentra la mayor concentración de fijaciones de los usuarios, la figura 39 muestra cómo identificar la concentración de fijaciones en el mapa de calor con un esquema de 5 colores.

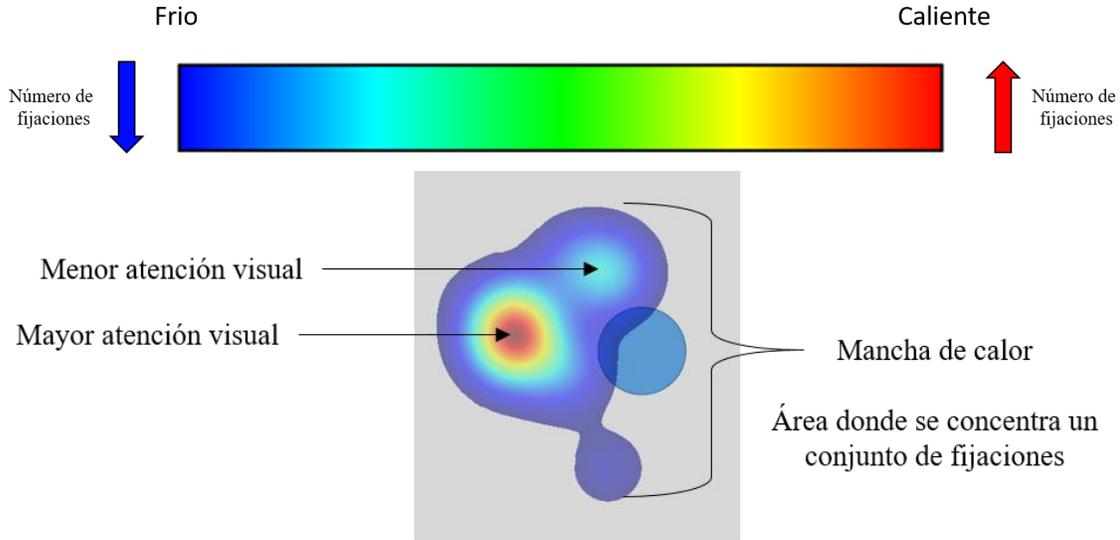


Figura 39: Esquema conceptual de las manchas de colores en los mapas de calor.

4.6.3 Mapas de rutas

Los mapas de rutas entran en la categoría de mapas de atención, se utilizan para observar el comportamiento ocular del usuario mediante la representación del camino que siguen los ojos entre las fijaciones. Para representar este camino se utilizan las sácadas, que son los movimientos de los ojos entre cada fijación.

Para calcular las sácadas se debe utilizar el archivo de datos de las fijaciones, con cada par de fijaciones se obtiene el inicio de una sácada (x, y), la duración de la sácada (diferencia entre la estampa de tiempo de una fijación con la segunda) y el final de la sacada (x, y) (4).

$$sacada = \{x_1, y_1, x_2, y_2, d\} \quad (4)$$

Donde:

- x_1 es un punto de la primera fijación en el eje x del plano espacial de la pantalla.
- y_1 es un punto de la primera fijación en el eje y del plano espacial de la pantalla.
- x_2 es un punto de la segunda fijación en el eje x del plano espacial de la pantalla.
- y_2 es un punto de la segunda fijación en el eje y del plano espacial de la pantalla.
- d es la diferencia de t_i de la primera fijación y t_{fde} la segunda fijación, representa el tiempo que duro la sácada.

La figura 40 muestra el pseudocódigo utilizado para calcular las sácadas a partir de las fijaciones. La función recibe como parámetro la lista de fijaciones grabada de una prueba de seguimiento ocular (línea 1), luego se definen variables, *listaPuntos* que contiene la lista de puntos de fijaciones leídas del archivo CSV, *siguiente* es una variable auxiliar que se utiliza más adelante, *listaSacadas* es la lista donde se guardan las sacadas obtenidas (líneas 2-4). Se utiliza un ciclo iterativo para cada punto de fijación en *listaPuntos* (línea 5) donde se utilizará la variable auxiliar para obtener la referencia al siguiente punto de la lista (línea 6), esto se realiza para obtener la sácada de cada par de puntos de fijación (línea 7). Al terminar el ciclo iterativo se retorna la lista de las sacadas (línea 9). Cabe mencionar que para generar una sácada debe de existir un par de fijaciones continuas, por lo que la longitud de la variable *listaPuntos* siempre será mayor por una unidad a la variable *listaSacadas*.

```

1  Función sacadas(csvFijaciones):
2      listaPuntos = leer(csvFijaciones)
3      siguiente = []
4      listaSacadas = []
5      para cada punto en listaPuntos:
6          siguiente = listaPuntos.next()
7          listaSacadas.agregar([punto.x,punto.y,siguiente.x,siguiente.y,siguiente.tiempo-punto.tiempo])
8      fin para cada
9  retorna listaSacadas
10 fin función

```

Figura 40: Pseudocódigo para la generación de sácadas a partir de fijaciones.

Luego de obtener la lista de sácadas de la prueba, se utiliza en conjunto con la lista de fijaciones para dibujar el mapa de rutas sobre la imagen o video de la prueba.

La figura 41 muestra el proceso computacional que se aplica para crear los videos de mapas de rutas, al igual que otros procesos que generan videos de métricas oculares, el algoritmo recibe como parámetros de entrada dos archivos: un de fijaciones y un video.

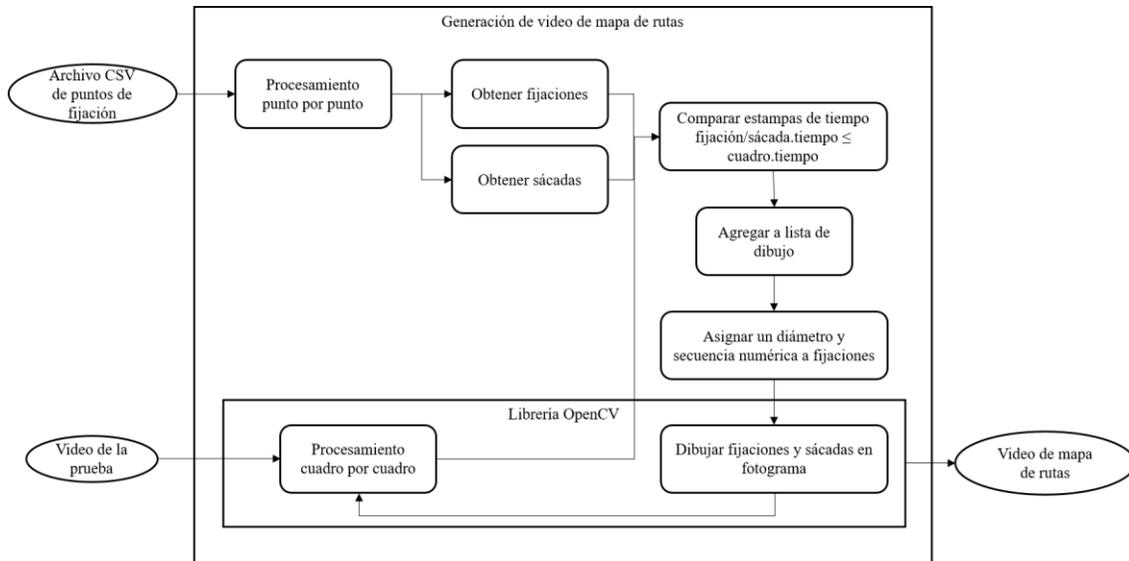


Figura 41: Esquema del proceso de generación de videos de mapa de rutas.

El proceso utiliza el archivo de fijaciones para obtener cada fijación y sácada, Estos objetos contienen una estampa de tiempo que corresponde al tiempo en el que fueron capturados los datos oculares en la grabación. Se utiliza la librería de OpenCV para iterar cada fotograma del video, las estampas de tiempo de las fijaciones y sácadas son comparadas con aquellas que corresponden al fotograma actual del video, si los objetos de seguimiento ocular se presentan al mismo tiempo que la estampa de tiempo del fotograma entonces son agregados a una lista de objetos a dibujar.

Las fijaciones son dibujadas como círculos numerados (de acuerdo al orden de aparición de la fijación) y de diferente diámetro en función de la duración del punto de fijación (esto para representar visualmente la cantidad de tiempo que el usuario enfocó su mirada en ese punto) (ver figura 42), mientras que las sácadas se representan como flechas cuyo punto inicial se encuentra en las coordenadas x, y de la primera fijación y de forma similar su punto final se representa en las coordenadas de la segunda fijación (ver figura 43), de tal manera que cada a cada sácada le corresponden 2 fijaciones.



Figura 42: Ejemplo de mapa de rutas utilizando la duración de las fijaciones para representar el tamaño de la fijación.

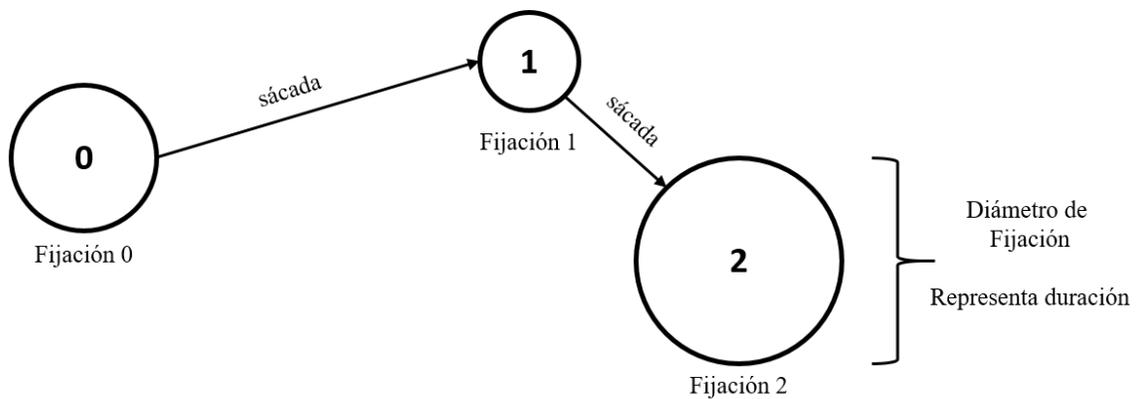


Figura 43: Esquema de mapa de rutas, se muestran 3 fijaciones numeradas en orden de aparición (del 0 al 2) y 2 sacadas donde a cada una le corresponde un par de fijaciones en su punto de inicio y final. El diámetro de las fijaciones representa la duración de la mirada del usuario en ese punto.

Este proceso se repite para cada objeto en la lista de fijaciones y de sacadas. Finalmente, los fotogramas se integran para generar el video con los objetos dibujados.

La figura 44 muestra un ejemplo de imagen con mapa de rutas procesado sin limitar el número de puntos en la pantalla.

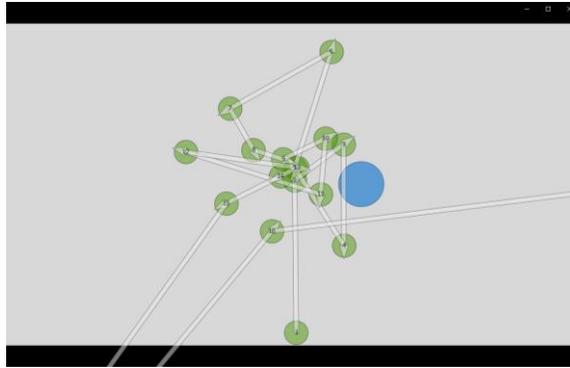


Figura 44: Ejemplo de mapa de rutas procesado sobre una prueba de seguimiento ocular.

A continuación, en la figura 45, se muestra el pseudocódigo de generación de videos de mapas de rutas utilizado en el trabajo de tesis.

```

1  Función mapaRutas(listaFijaciones, listaSacadas):
2      fotogramas = cv2.videoCapture('ruta al video de la prueba')
3      listaDibujo = []
4      fijación = listaFijaciones.pop() //toma el primer elemento de la lista
5      sacada = listaSacadas.pop() //toma el primer elemento de la lista
6      para cada fotograma en fotogramas:
7          si fijación.tiempo <= fotograma.tiempo:
8              listaDibujo.agregar(fijación)
9              fijación = listaFijaciones.pop() // toma la siguiente fijación
10             listaDibujo.agregar(sacada)
11             sacada = listaSacadas.pop() // toma la siguiente sácada
12         fin si
13     si listaDibujo.longitud > 10:
14         listaDibujo.cut(10) // limita la lista a 10 elementos
15     fin si
16     dibujar(listaDibujo)
17     fin para cada
18     regresa video(fotogramas)
19     fin función

```

Figura 45: Pseudocódigo para la generación de mapas de rutas.

Dentro del contexto de la evaluación del comportamiento humano, el número de sácadas en el mapa de rutas (cada flecha del mapa representa una sacada) representa el nivel de atención visual del usuario.

Un número elevado de sácadas indica que la persona ha escaneado el contenido visual de manera muy activa y por ende representa también un estado mental de atención. En cambio, un número pequeño de sácadas podría representar falta de interés en el contenido del estímulo visual (representa una vista rápida, no detallada del contenido), sin embargo,

es posible que represente un análisis más profundo de ciertos elementos visuales específicos del estímulo, si la duración de las fijaciones es mayor (ver tabla 5).

Tabla 5: Pruebas de eficiencia de extracción de parpadeos con EyeWidget.

Elemento visual	Más Sácadas	Menos Sácadas
Más fijaciones	Mayor atención visual	Mayor atención visual en puntos específicos
Menos fijaciones	Mayor escaneo visual	Menor atención visual

4.6.4 Generación de áreas de interés dinámicas

Para la generación de áreas de interés (AOI) dinámicas se utiliza el algoritmo de agrupamiento Meanshift (Fuhl, et al., 2018; Santella y DeCarlo, 2004), este algoritmo se utiliza en el campo de la visión artificial para la segmentación de imágenes, suavizado de bordes y seguimiento de objetos. El método Meanshift no es paramétrico, por lo que el clúster puede agrupar diferentes puntos de visión sin tener que definir el número de grupos con anterioridad (ver figura 46).

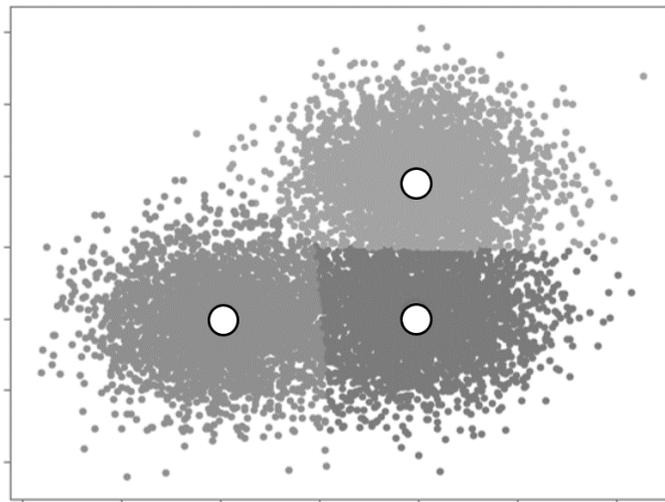


Figura 46: Definición de 3 clústeres dada la concentración de datos con Meanshift (Pedregosa et al., 2011).

El algoritmo Meanshift se basa en el concepto de estimación de densidad del núcleo (KDE), que es un método para estimar la distribución de probabilidad (también llamada función de densidad de probabilidad) para un conjunto de datos.

Para calcular el KDE se debe utilizar un kernel en cada punto del conjunto de datos. Un kernel es una función de ponderación (le da un valor a cada dato). Existen muchos tipos de kernel, el más popular es el kernel gaussiano (6) (Haddad y Akansu, 1991; Nixon, y Aguado 2008).

$$K(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (6)$$

Donde:

- x es el valor del punto de visión en la coordenada x .
- y es el valor del punto de visión en la coordenada y .
- σ es la desviación estándar.
- e es la función exponente.

Al agregar todos los valores de kernel individuales se genera la superficie de probabilidad KDE (7).

$$f(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (7)$$

Donde:

- K es una función no negativa.
- h es un parámetro llamado *ancho de banda*, que sirve para definir el alcance de la comparación entre puntos, si se usa un kernel gaussiano, la desviación estándar funciona como h .
- n es el número de muestras.
- x son los datos de entrada (en este caso los puntos de visión).

El algoritmo funciona cambiando de forma iterativa vectores que mueven los puntos de datos hacia las modas del conjunto de datos discretos muestreados. Comienza con una estimación inicial x . Y utiliza la función kernel para determinar el peso de los puntos cercanos para la reestimación de la media. La media ponderada de la ventana donde se encuentra el punto es determinada por (8):

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x)x_i}{\sum_{x_i \in N(x)} K(x_i - x)} \quad (8)$$

Donde:

- x es el punto seleccionado.
- x_i es un punto que pertenece a $N(x)$.
- $N(x)$ es la vecindad de puntos, los cuales son un conjunto para los cuales $K(x_i)$ resulta diferente de 0.
- $K(x_i - x)$ es la función kernel utilizada para aproximar una función de densidad de probabilidad subyacente del conjunto de datos.

El algoritmo pasa a la siguiente iteración, ahora estableciendo $x \leftarrow m(x)$, y repite la estimación hasta que $m(x)$ converja. La diferencia de $m(x)-x$ se llama desplazamiento medio (Meanshift, en inglés) (Fukunaga y Hostetler, 1975).

La figura 47 ilustra el proceso del algoritmo Meanshift, éste se centra en encontrar la moda más cercana para cada punto del conjunto de datos, para hacerlo, se define una región alrededor de ese punto y busca la media de los puntos en la región, luego cambia la media actual a la nueva (shift, cambio en inglés) y repite el proceso hasta que converja, de este modo el punto del conjunto de datos se le asigna un centro o grupo nuevo al que pertenece.

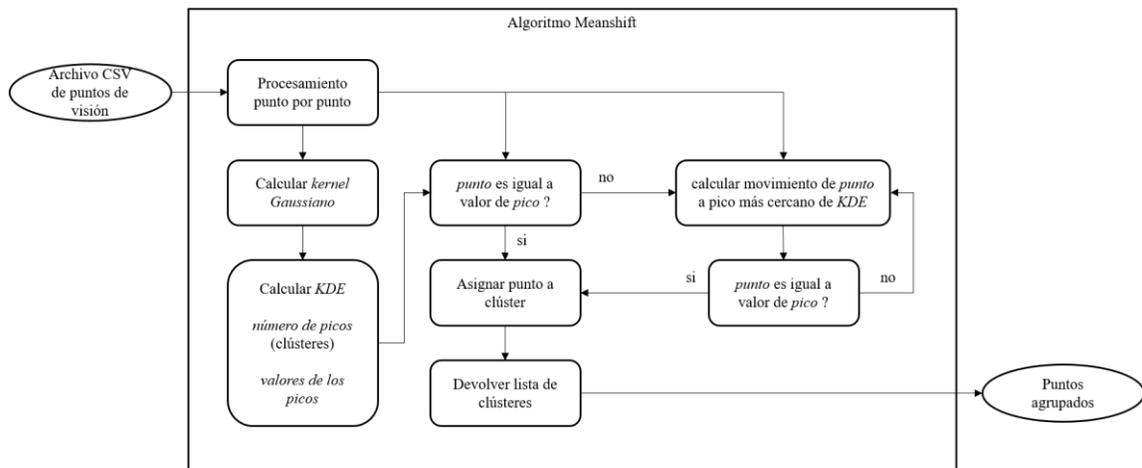


Figura 47: Esquema del proceso del algoritmo Meanshift.

En esta tesis se utilizó una implementación de este algoritmo para el lenguaje de programación Python, en la librería de scikit-learn (Pedregosa, et al., 2011), esta función utiliza el algoritmo de Meanshift con un kernel plano para agrupar los puntos de visión de los usuarios en grupo dinámicos. Dentro de los centros de los grupos calculados con el algoritmo se encuentran una mayor cantidad de fijaciones que se denominan áreas de interés.

La figura 48, se muestra el pseudocódigo de generación de videos de áreas de interés dinámicas utilizado en el trabajo de tesis, éste utiliza una lista de los clústeres devueltos por la función de la librería scikit-learn (línea 1) y son dibujados utilizando las herramientas de OpenCV (línea 13). El código utiliza dos variables, *fotogramas* para guardar los cuadros de video y *listaDibujo* para ir guardando los objetos de AOI a dibujar (líneas 2-3). Al igual que la generación de otros videos de métricas oculares, por cada clúster (AOI) se realiza la comparación de estampas de tiempo de los clústeres contra el tiempo del cuadro del video para determinar en qué fotogramas se dibujarán las AOI (líneas 5-9). Cada vez que se dibujan las AOI (línea 10) se libera la variable de *listaDibujo* (línea 11) esto para refrescar las regiones de interés en la pantalla.

```

1  Función aoi(listaClusteres):
2      fotogramas = cv2.videoCapture('ruta al video de la prueba')
3      listaDibujo = []
4      para cada fotograma en fotogramas:
5          para cada cluster en listaClusteres:
6              si cluster.tiempo < fotograma.tiempo:
7                  listaDibujo.agregar(cluster)
8              fin si
9          fin para cada
10         dibujar(listaDibujo)
11         listaDibujo = [] // se restablece la lista de dibujo
12     fin para cada
13     regresa video(fotogramas)
14     fin función

```

Figura 48: Pseudocódigo para la generación de áreas de interés dinámicas.

El programa se codificó de tal manera que se utilizan bloques de 500 puntos de datos de visión (aproximadamente 10 segundos de datos oculares) para encontrar los centros de las áreas de interés con Meanshift, de esta manera se puede observar de manera dinámica

en función del tiempo, los cambios de atención del usuario en el video de la prueba. La figura 49 muestra un ejemplo de video con áreas de interés procesado.

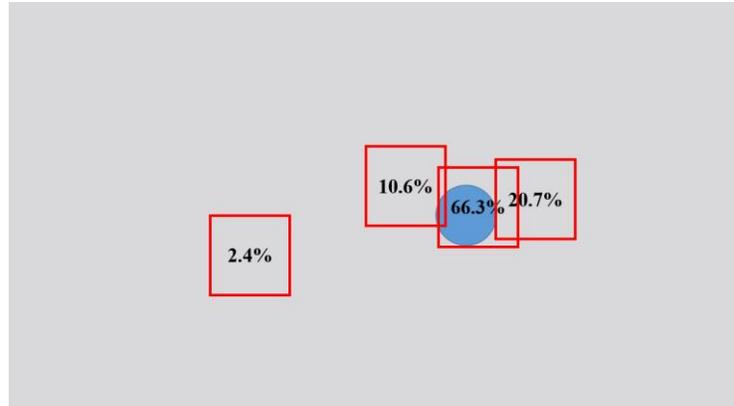


Figura 49: Ejemplo de áreas de interés procesadas sobre una prueba de seguimiento ocular.

4.6.5 Análisis del parpadeo

Para realizar un análisis de la frecuencia del parpadeo se realizó un proceso de obtención de promedio del parpadeo del usuario en función del tiempo de la prueba, esto con el objetivo de ir observando el cambio en la frecuencia del parpadeo dependiendo de los minutos transcurridos. Para ello se utilizó la formula (8), que representa la sumatoria de los promedios de los conjuntos de parpadeos que corresponden a cada minuto que transcurre durante la grabación P_{mi} , estos valores se registran y etiquetan con cada minuto de la prueba en otro archivo en formato CSV, el cual es usado posteriormente para observar las diferencias en el comportamiento ocular durante la grabación.

$$Frecuencia\ parpadeo = \frac{\sum_{i=0}^m \overline{P_{mi}}}{m} \quad (8)$$

Donde:

$\overline{P_{mi}}$ = promedio de parpadeos del minuto i .

m = minutos grabados de la prueba.

La figura 50 muestra el proceso que se sigue para obtener el archivo de frecuencias de parpadeo. Primero se procesa cada línea del archivo que contiene el registro de los parpadeos capturados en la prueba de seguimiento ocular, con el fin de cuantificar cada

parpadeo que corresponda al conjunto de parpadeos que ocurrieron en el mismo minuto de la grabación (se hace uso de las estampas de tiempo para clasificar los conjuntos). Por cada conjunto se agrega el valor del promedio de la cantidad de parpadeos entre el número de minutos transcurridos en la prueba, este valor x se agrega a una lista de frecuencias de parpadeos que son etiquetadas con el minuto de la prueba donde se calculó la frecuencia. Se realiza este proceso hasta que ya no quedan parpadeos para procesar y finalmente, se retorna un archivo en formato CSV con la lista de frecuencias de parpadeos etiquetadas.

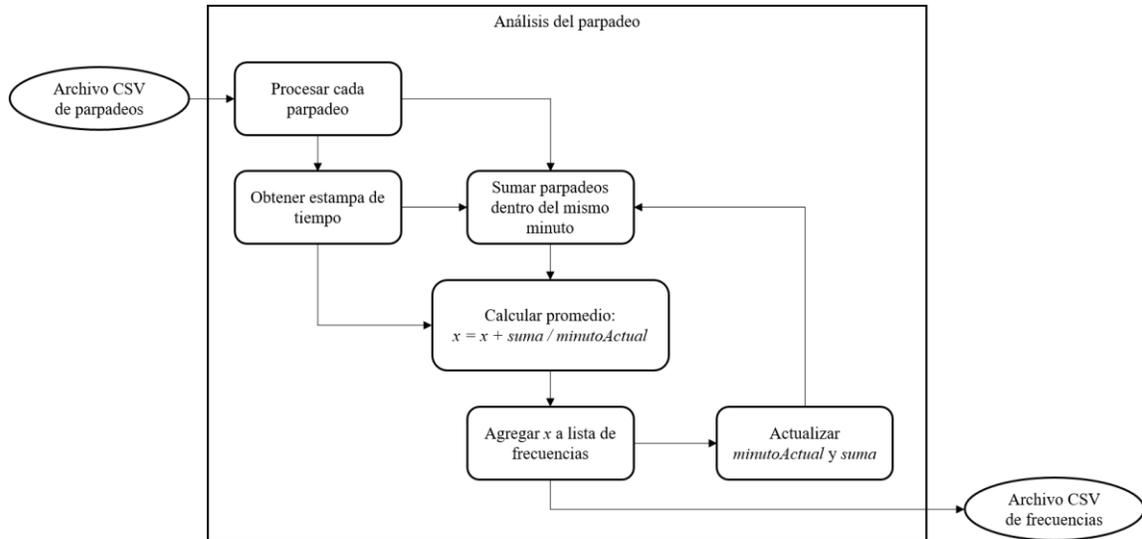


Figura 50: Esquema del proceso del algoritmo de análisis del parpadeo.

El archivo de frecuencias CSV se utiliza para visualizar una gráfica de frecuencia de parpadeo en función del tiempo de la prueba (ver figura 51).

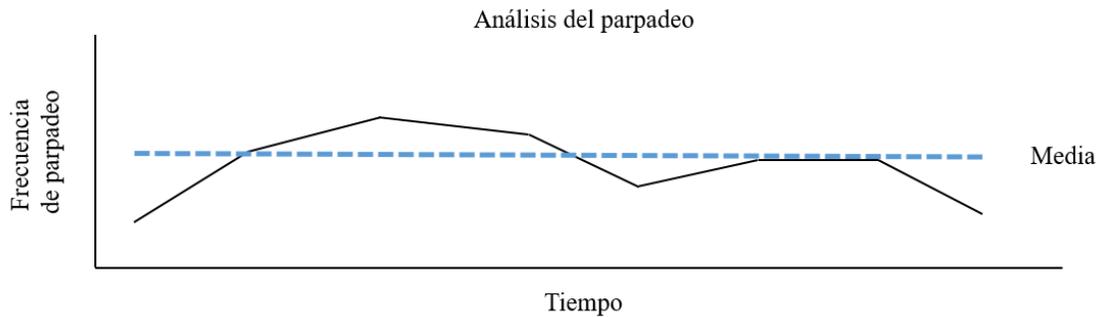


Figura 51: Ejemplo de gráfica de análisis de parpadeo resultante de archivo de frecuencias.

De acuerdo con la literatura revisada en este trabajo, la frecuencia del parpadeo se correlaciona con el estado cognitivo del usuario, es decir, una mayor cantidad de parpadeos representa menor carga cognitiva y una menor cantidad de parpadeos representan mayor carga cognitiva. La figura 52 describe la interpretación de la gráfica de análisis de parpadeos de la siguiente manera:

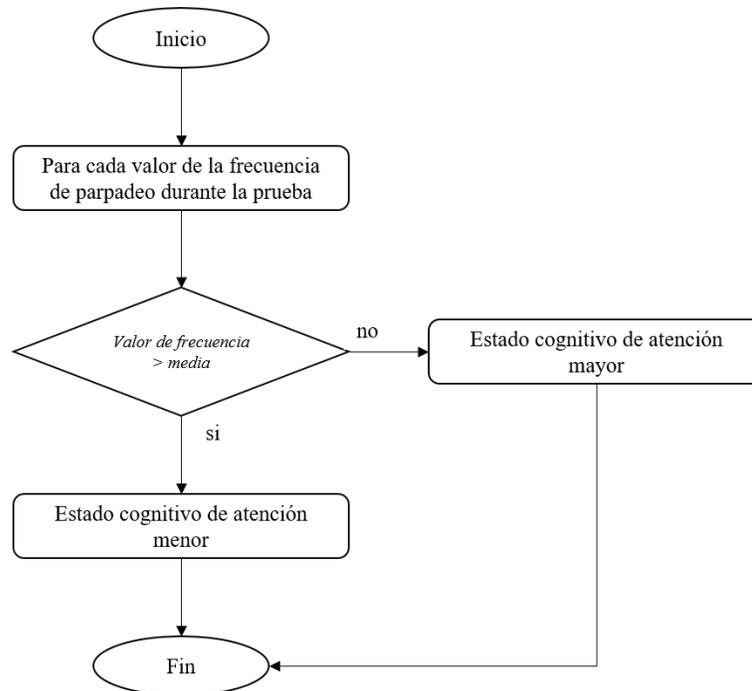


Figura 52: Diagrama de flujo de interpretación de estado cognitivo mediante gráfica de análisis del parpadeo.

La figura 53 muestra el pseudocódigo para realizar el cálculo de la frecuencia del parpadeo con el archivo de conteo de parpadeos. La función recibe como parámetro la variable *listaParpadeos* (línea 1), se definen variables auxiliares, *minutoActual* que representa el minuto de las estampas de tiempo que se maneja en el momento, *suma* es una variable auxiliar para realizar el conteo de parpadeos, *frecuencia* es la variable que guarda la sumatoria de promedios y la variable *frecuencias* se utiliza para el registro de frecuencias en función del tiempo (líneas 2-5). Se itera cada *parpadeo* en *listaParpadeos* (línea 6), si la estampa de tiempo del parpadeo corresponde al *minutoActual* se agrega un parpadeo a la variable *suma* (línea 8), cuando se detecta un parpadeo fuera del *minutoActual* se realiza el cálculo del promedio del parpadeo en la variable *frecuencia*, la

cual sumará el valor del nuevo promedio a su valor actual, posteriormente la frecuencia será agregada a la variable *frecuencias* y etiquetada con el *minutoActual*. Finalmente, se reinicia el valor de la variable *suma* y se incrementa en 1 unidad el valor de *minutoActual* (líneas 9-13). Después de iterar por cada parpadeo disponible en los datos de entrada, la función retorna el archivo CSV conteniendo los datos etiquetados de la variable *frecuencias* (línea 16).

```

1  Función blinkFrec(listaParpadeos):
2  |   minutoActual = 0
3  |   suma= 0
4  |   frecuencia = 0
5  |   frecuencias = []
6  |   para cada parpadeo en listaParpadeos:
7  |       si parpadeo.tiempo < minutoActual+1:
8  |           |   suma = suma+1
9  |       si parpadeo.tiempo > minutoActual+1:
10 |           |   frecuencia = (frecuencia + suma) / (minutoActual + 1)
11 |           |   frecuencia.agregar([minutoActual, frecuencia])
12 |           |   suma = 0
13 |           |   minutoActual = minutoActual + 1
14 |       fin si
15 |   fin para cada
16 |   regresa archivo(frecuencias)
17 |   fin función

```

Figura 53: Pseudocódigo para la generación de archivo de frecuencia de parpadeo.

4.7 Integración con UXLab

Uno de los objetivos de esta tesis fue integrar un módulo de seguimiento ocular en el software UXLab para proporcionar métricas de análisis de la atención visual en las pruebas de evaluación de la experiencia de usuario (UX). El software UXLab (ver Figura 54) proporciona herramientas para extraer y sincronizar información de diferentes tipos de sensores fisiológicos y biométricos, de esta manera se pueden obtener datos de los sensores mientras el usuario interactúa con productos digitales durante una evaluación UX.

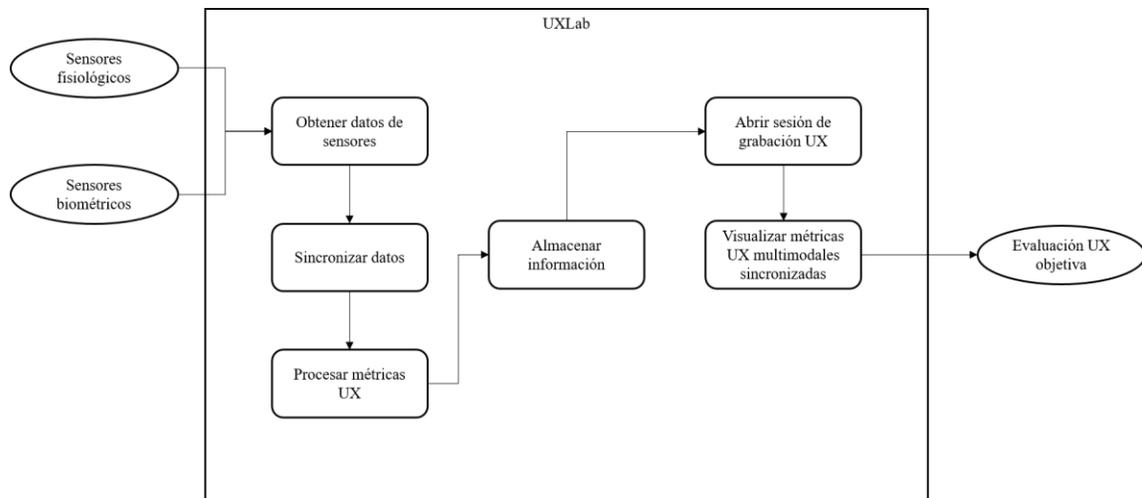


Figura 54: Esquema de proceso generalizado de UXLab.

UXLab incluye métodos para identificar el estado emocional de los usuarios. Puede grabar el rostro del usuario, la memoria de video de la pantalla, datos de electrocardiografía, datos de respuesta galvánica en piel, datos EEG y como resultado de esta tesis se incorporó el módulo de seguimiento ocular para complementar las métricas del comportamiento del usuario.

La primera versión de UXLab incluía soporte de seguimiento ocular limitado ya que utilizaba un sensor con poca precisión que posteriormente fue descontinuado del mercado, era propenso a ruidos y el usuario no tenía mucha libertad para mover la cabeza durante las sesiones de grabación.

Por lo anterior, se requería incorporar tecnología y algoritmos de seguimiento ocular más robustos en la plataforma UXLab utilizando un sensor más eficiente. El sensor de tecnología infrarroja que se utilizó permite al usuario mover la cabeza con mayor libertad sin perder el seguimiento de los ojos y además permite la manipulación de los datos crudos mediante un SDK. Durante el proceso de implementación de estos módulos en UXLab se solucionaron problemas técnicos debido a las licencias del software, finalmente se logró implementar el módulo de seguimiento desarrollando con el programa EyeWidget, ver Figura 32 en sección 4.5.4, el cual extrae la información relevante del sensor cuando UXLab lo requiere.

4.7.1 Funcionamiento de UXLab

UXLab es una suite conformada por dos sistemas, un sistema de grabación y un sistema reproductor. El trabajo realizado en esta tesis incluyó un tercer componente de software (EyeWidget) dentro del sistema de grabación de UXLab, además de agregar y extender funcionalidades en los sistemas del grabador como del reproductor.

Los dos componentes de UXLab se utilizan de manera independiente, sin embargo, los datos de las grabaciones de evaluaciones UX son la entrada del sistema reproductor.

La primera parte, llamada *outlet*, se utiliza en para cargar e iniciar los sensores para extraer la información, etiquetarla, empaquetarla y enviarla a través de un buffer compartido de datos (ver Figura 55), mientras que la segunda parte, llamada *inlet*, se encarga de sincronizar, procesar y almacenar los datos de los diferentes sensores (ver Figura 56).

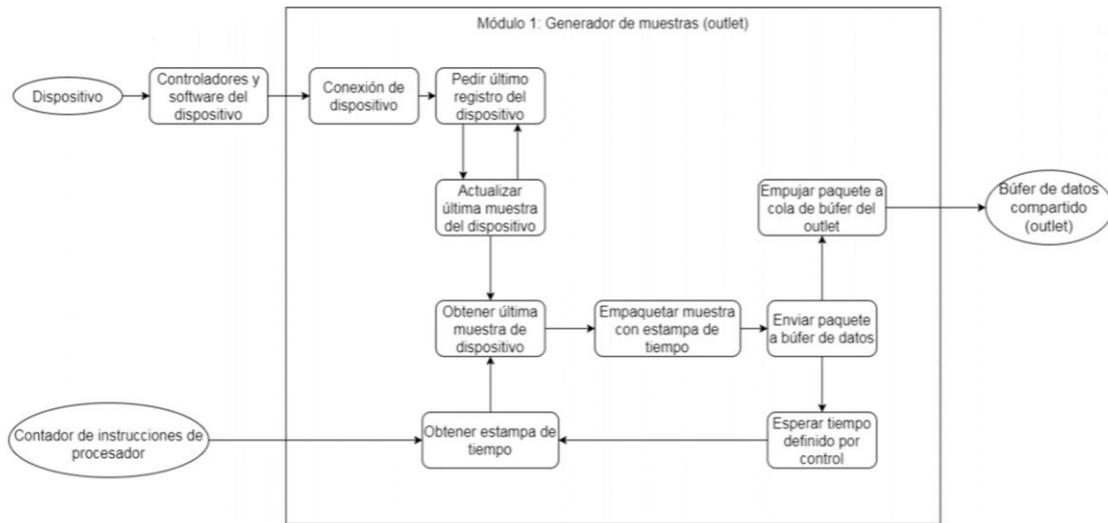


Figura 55: Esquema conceptual de funcionamiento del módulo *outlet* del grabador UXLab, de D.

Fouilloux, 2018, Todos los derechos reservados 2018. Tomado con permiso del autor.

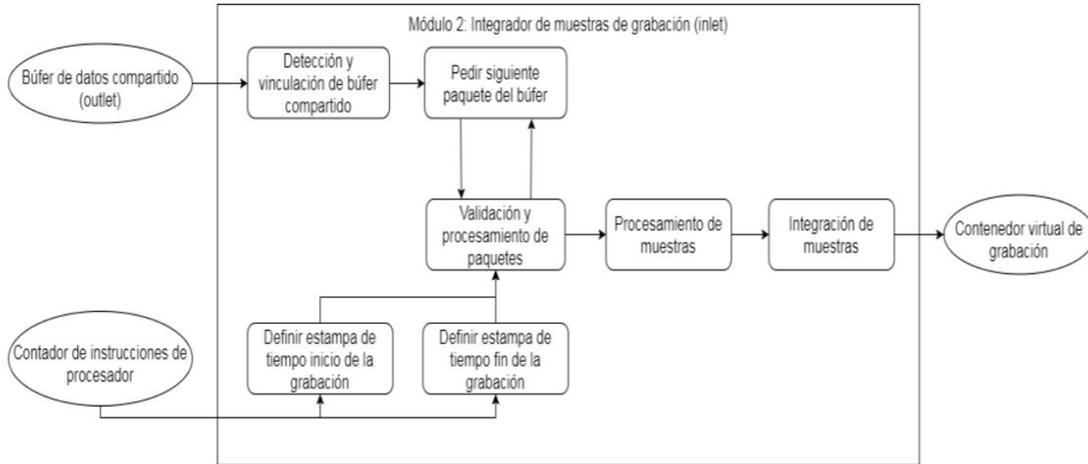


Figura 56: Esquema conceptual de funcionamiento del módulo *inlet* del grabador UXLab, de D. Fouilloux, 2018, Todos los derechos reservados 2018. Tomado con permiso del autor.

En este trabajo de tesis se implementaron y extendieron los algoritmos de extracción y de generación métricas mediante la inclusión de EyeWidget en UXLab. La figura 57 muestra el esquema conceptual del funcionamiento de UXLab incluyendo EyeWidget, el cual se comunica con el grabador de UXLab mediante una conexión con socket.

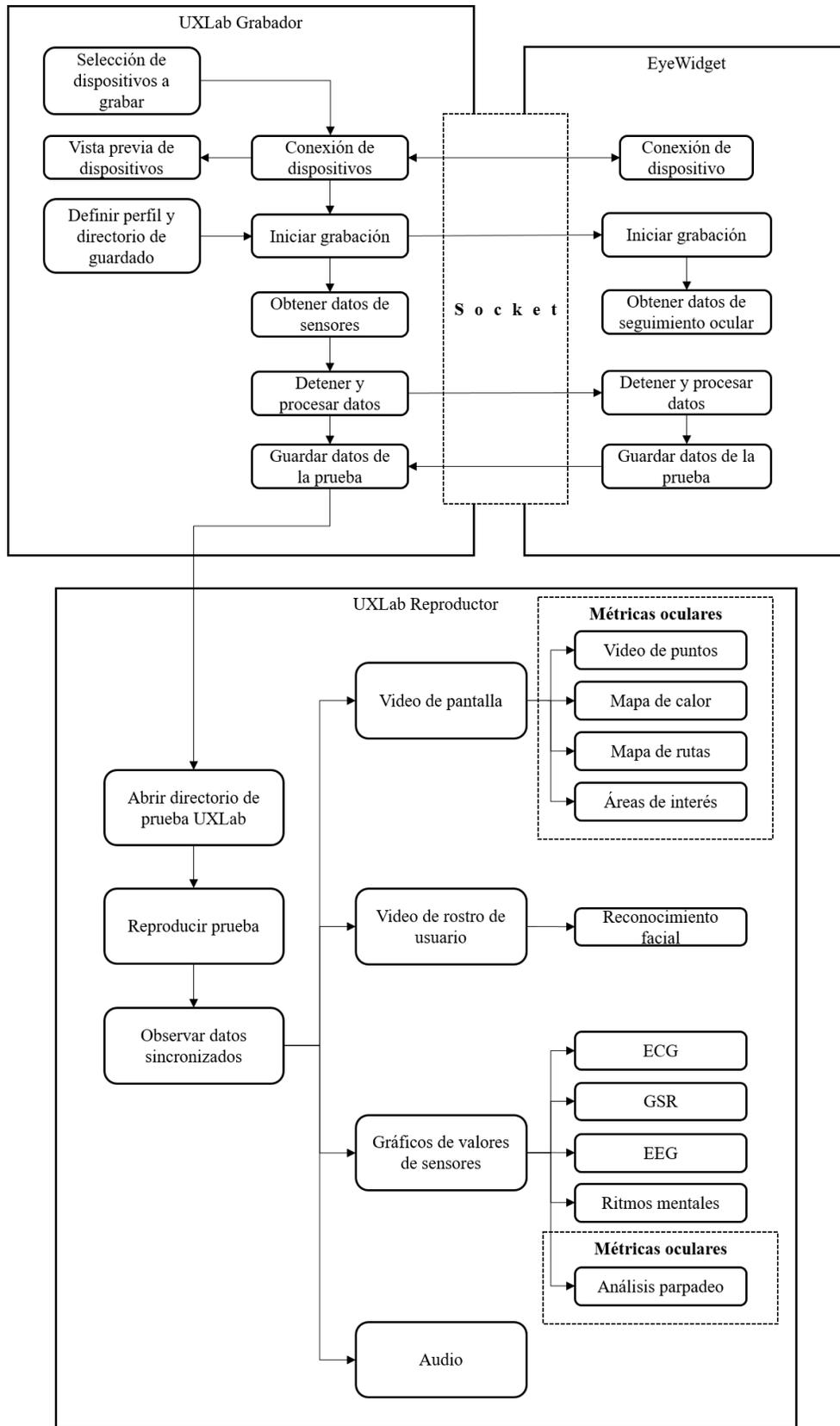


Figura 57: Esquema de funcionamiento de UXLab con EyeWidget.

4.7.2 Integración y mejoras del módulo de seguimiento ocular

La figura 58 resume las características que fueron incorporadas al software de UXLab en el trabajo de tesis, el postprocesamiento de datos oculares y de detección facial utilizando el video de la cámara web. El reconocimiento facial se trabajó como un componente adicional en la investigación, podría ser útil para apoyar la extracción de datos del parpadeo y de dilatación de pupila.

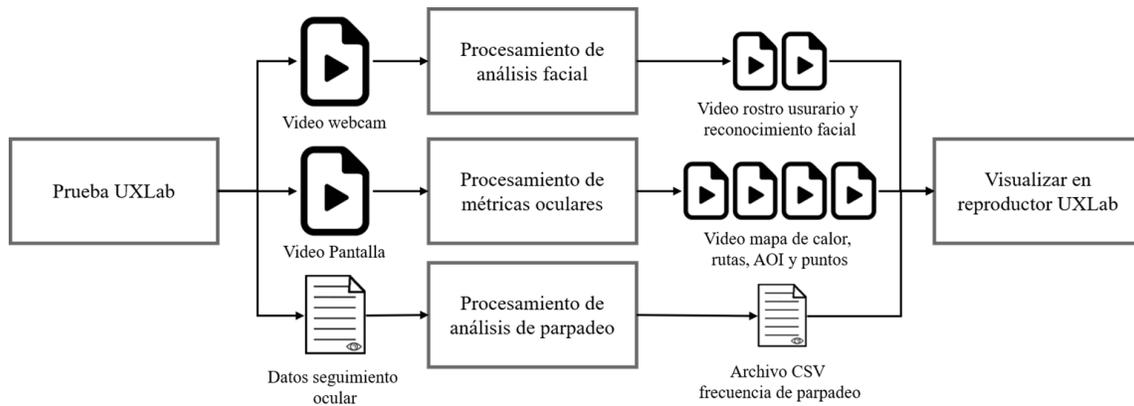


Figura 58: Esquema de funciones extendidas e implementadas en UXLab, relacionadas con el módulo de seguimiento ocular.

A continuación, se describe a detalle la integración de funciones y mejoras del módulo de seguimiento ocular de UXLab.

4.7.2.1 Modificaciones en grabador UXLab

Después de contemplar diferentes tipos de comunicación entre procesos (por ejemplo, señales de procesos) se decidió implementar el intercambio de mensajes entre UXLab (Python) y EyeWidget (C#) mediante sockets. EyeWidget es controlado por el módulo de dispositivos de UXLab mediante una variedad de instrucciones enviadas desde UXLab a EyeWidget a través de la comunicación por sockets entre los dos sistemas (ver figura 59). Estas señales incluyen las instrucciones de grabar, detener y cerrar el programa.

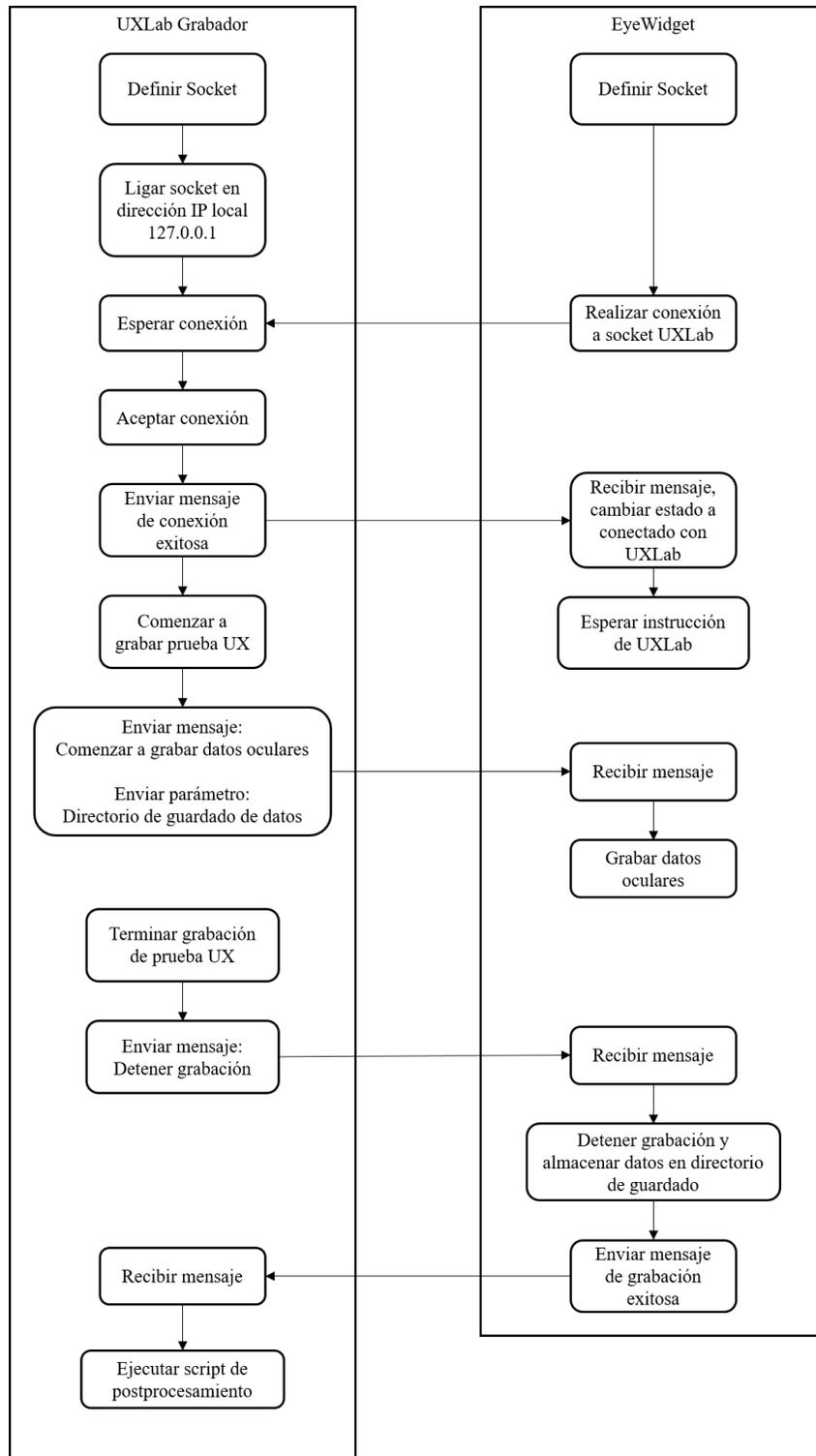


Figura 59: Esquema de funcionamiento de la comunicación por socket entre UXLab y EyeWidget.

Dentro de la función de UXLab donde se inicia el sensor de seguimiento ocular se implementó una función que primero ejecuta el programa EyeWidget, inmediatamente se

crea un socket. Mientras EyeWidget inicia, automáticamente intenta conectarse al socket de UXLab, estableciendo la comunicación entre los dos programas. La versión con conexión por socket de EyeWidget requiere de UXLab para funcionar ya que, éste envía las instrucciones e información a EyeWidget sobre donde guardar los datos del seguimiento ocular después de las grabaciones.

Además de la implementación de la herramienta de extracción de datos, también se implementaron los algoritmos de generación de métricas oculares (ver sección 4.6) durante el postprocesamiento de los datos al final de las grabaciones UX (ver figura 60).

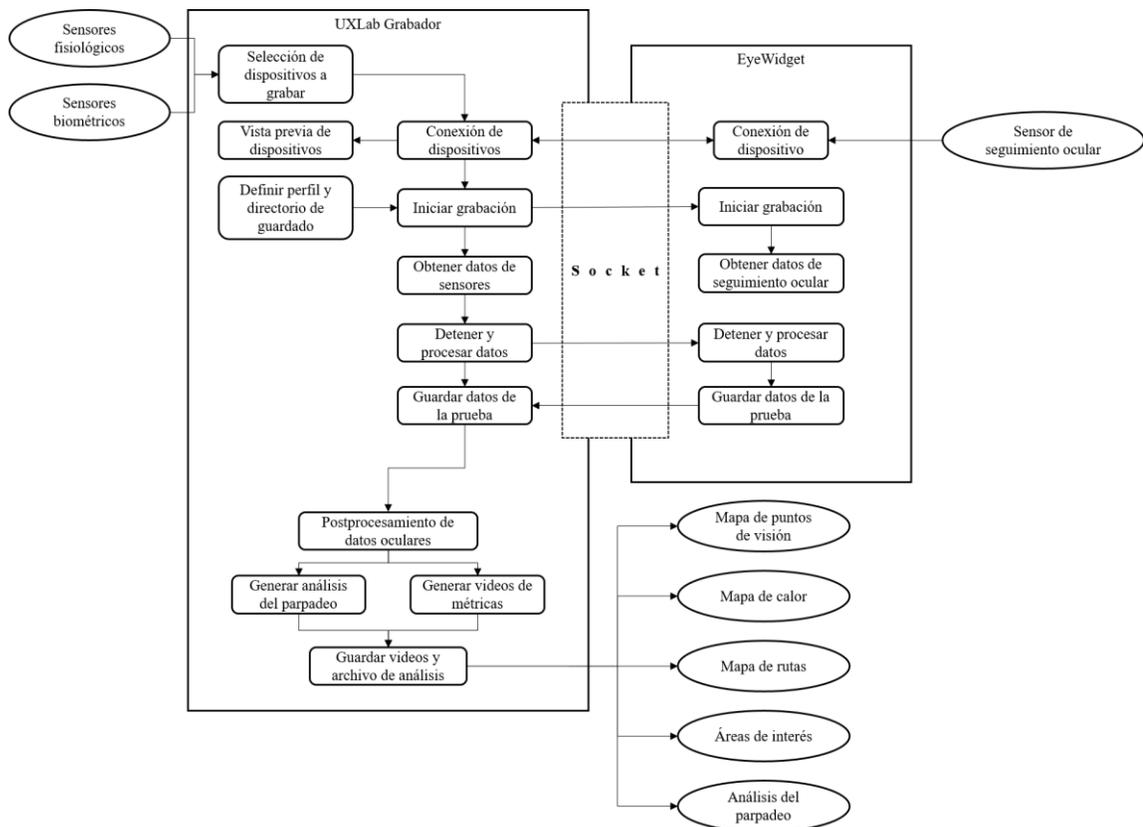


Figura 60: Esquema de proceso de postprocesamiento de datos oculares al final de una grabación en el grabador de UXLab.

Para generar estas métricas se incluyó en la parte inferior derecha del módulo de dispositivos, una casilla que activa el postprocesamiento de los videos de métricas oculares y el análisis del parpadeo (debido a que este proceso tarda mucho tiempo y puede entorpecer un flujo de grabación de varias pruebas UX).

4.7.2.1 Reproductor UXLab

El software reproductor de UXLab permite observar los datos de una prueba, se muestran de manera sincronizada y reproducen en conjunto en función de una línea de tiempo, cuando esto sucede se puede observar en un mismo instante de tiempo los videos, datos y estados del usuario en ese instante de tiempo.

La figura 61 ilustra el proceso del funcionamiento del reproductor UXLab, las áreas con líneas punteadas muestran las métricas oculares generadas.

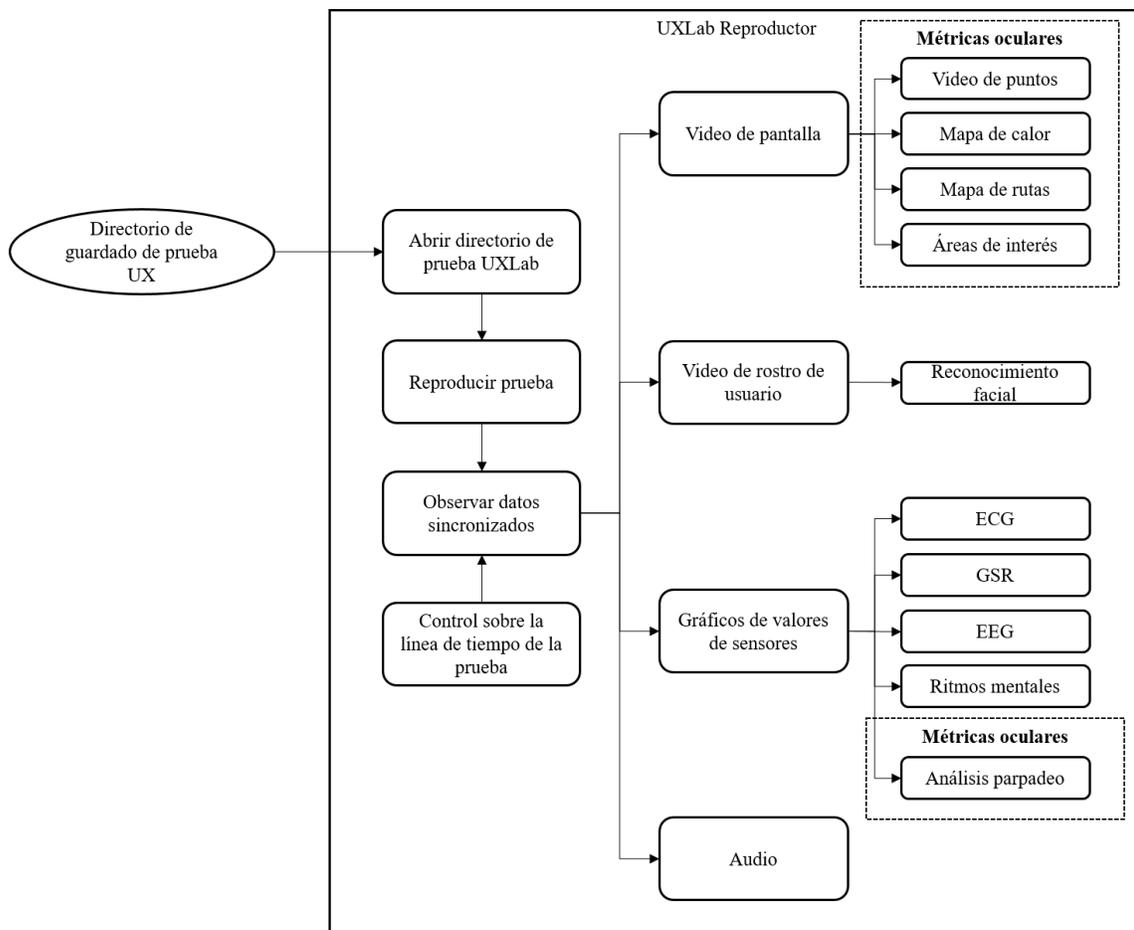


Figura 61: Esquema de proceso de visualización de pruebas UX en el reproductor de UXLab.

Se incluyó mediante una barra de menús la implementación de la visualización de las métricas oculares, esto permite activar o desactivar la visualización de las opciones del seguimiento ocular y el reconocimiento facial (ver figura 62).

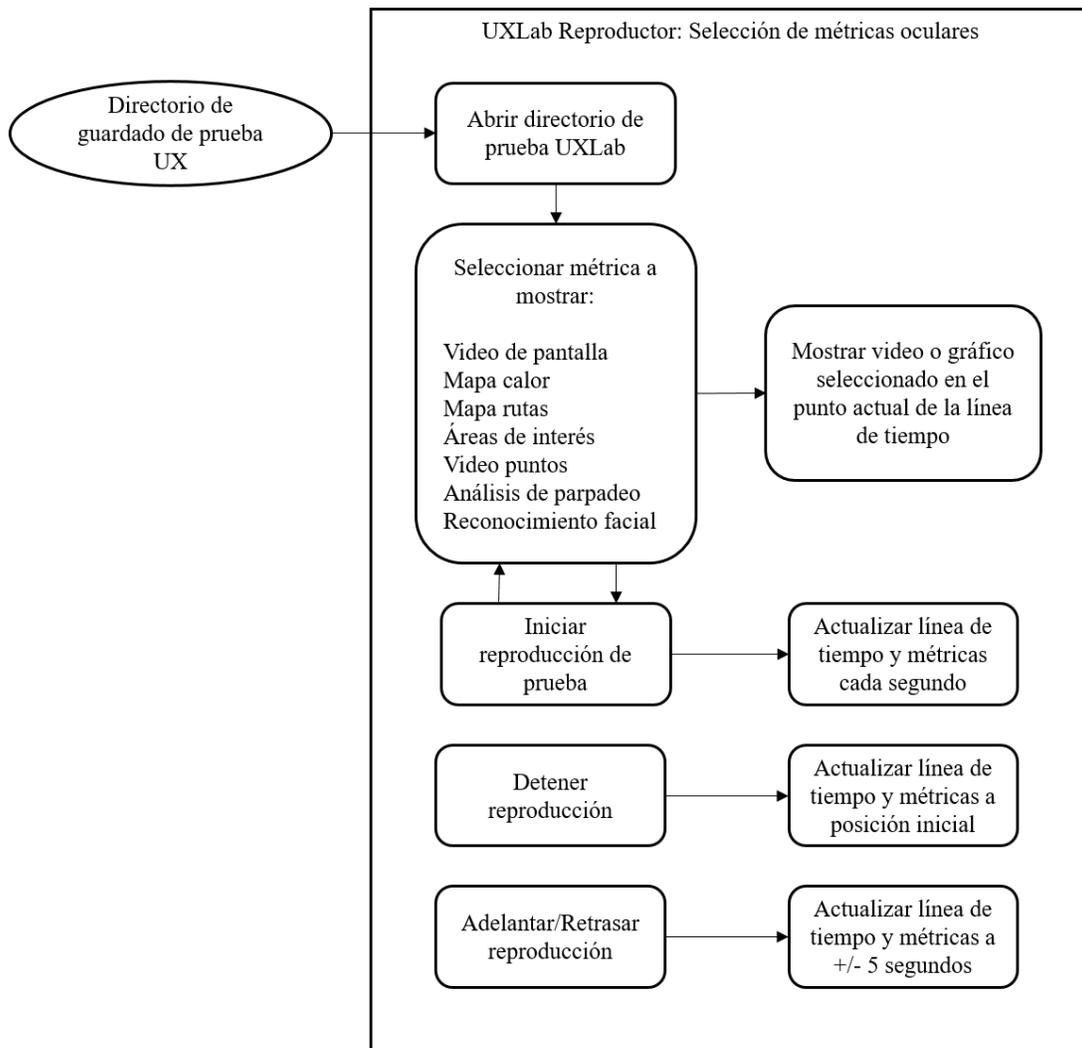


Figura 62: Esquema de proceso de cambio de métricas oculares en el reproductor de UXLab.

4.7.2.3 Mejoras generales a UXLab

Además de integrar nuevas funcionalidades en el software de UXLab, se trabajó en mejoras dentro del código de la interfaz gráfica, con el fin de mejorar su usabilidad. Las mejoras fueron las siguientes:

- Visuales
 - Iconos
 - Tooltips
 - Botones consistentes
 - Menús de ayuda

- Validación de campos
- Tamaños, posición y consistencia de ventanas
- Funcionales
 - Arreglo de bugs variados en módulos de video
 - Implementación de código para realizar un reconocimiento facial
 - Implementación de postprocesamiento de las métricas oculares de manera automática
 - Generación del paquete de instalación completo (Antes de la generación de este paquete de instalación se debía buscar cada componente que requería UXLab e instalarlo por separado):
 - Python 2.7 y 3.7
 - Librerías de Python
 - EyeWidget
 - SDK de sensor de seguimiento ocular
 - Reproductor
 - Grabador

Finalmente, la figura 63 describe el proceso que sigue UXLab desde la grabación de datos (incluyendo datos oculares con EyeWidget) hasta la visualización de las métricas.

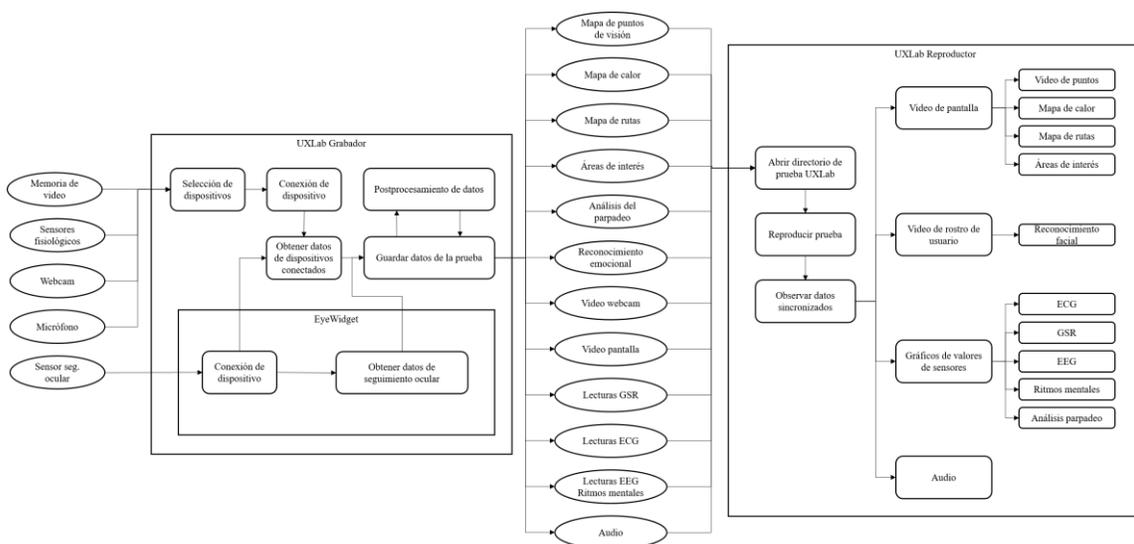


Figura 63: Esquema de proceso de UXLab.

4.8 Método de correlación de métricas del seguimiento ocular y estado mental del usuario

Al utilizar el conjunto de herramientas que ofrece UXLab y los algoritmos de métricas oculares que fueron implementados en el software reproductor de pruebas UX, se obtiene una herramienta que permite correlacionar el seguimiento ocular del usuario con los estados mentales que experimenta: emocionales utilizando los algoritmos de predicción emocional de UXLab y cognitivos utilizando el análisis del parpadeo.

En la figura 64 se muestra el diagrama de flujo del algoritmo que se sigue para correlacionar las métricas del seguimiento ocular y los estados mentales.

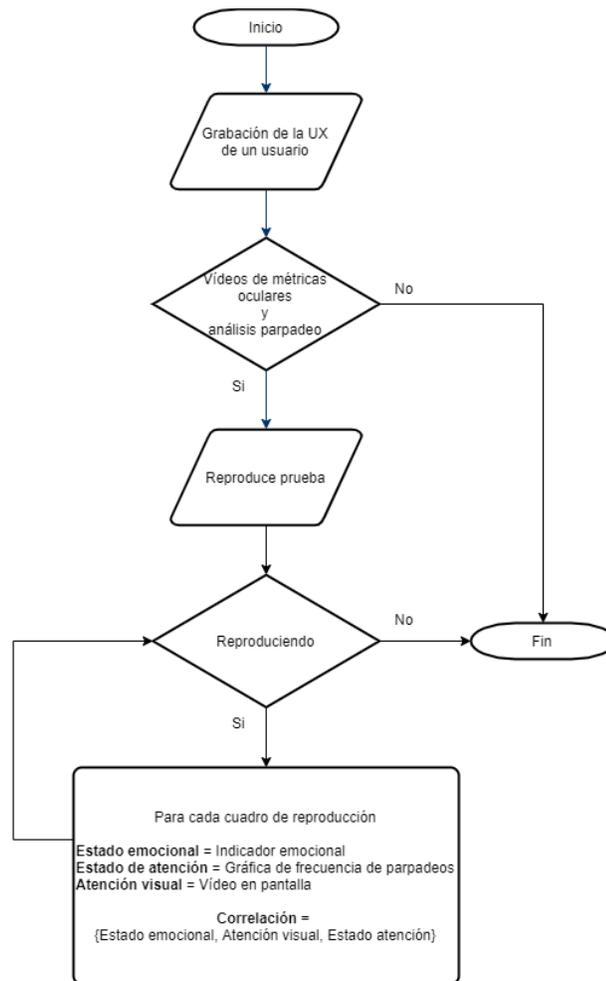


Figura 64: Diagrama de flujo del algoritmo de correlación de métricas del seguimiento ocular y estado mental del usuario.

Al iniciar la reproducción se pueden seleccionar las diferentes métricas del seguimiento ocular para que se muestren sobre el video de la pantalla del usuario, al mismo tiempo seleccionar la opción del análisis de los parpadeos muestra una gráfica con los cambios de la frecuencia del parpadeo (parpadeos por minuto) la cual da un indicador del estado cognitivo de atención del usuario.

El indicador de línea de tiempo de la reproducción muestra al evaluador los datos de sensores y de parpadeos por cada segundo de reproducción de la prueba, en conjunto con los indicadores de emociones y los videos con métricas de seguimiento ocular y el reconocimiento facial, se puede observar, por ejemplo, si una parte de la interacción de un programa llamó la atención del usuario y qué estado emocional sintió en ese instante de tiempo. Agregando así un valor adicional a las pruebas de la UX, el cual, fue uno de los objetivos de esta investigación.

Cabe mencionar que el hecho de que un usuario enfoque su mirada en algún elemento visual puede denotar un estado cognitivo de atención por defecto, sin embargo, debe ser analizado con cuidado debido a que una fijación no solo puede ser sinónimo de atención si no que a bajo nivel, representa un análisis de información visual por parte del usuario. Un estímulo visual si bien puede ser observado porque llama la atención, también puede ser observado porque el usuario está tratando de comprender el estímulo, como, por ejemplo, una fórmula matemática o una figura abstracta.

Capítulo 5. Pruebas y resultados

En este capítulo se reportan las pruebas que se implementaron sobre la integración de los algoritmos de correlación de métricas del seguimiento ocular con el estado mental del usuario.

5.1 Plan de pruebas

El plan de pruebas propuesto para este trabajo de tesis se dividió en tres grupos, cada uno de los cuales corresponde a la verificación y validación de los objetivos planteados en esta tesis, a continuación, se describe cada una de las pruebas realizadas:

- **Pruebas funcionales de UXLab**, el objetivo de estas pruebas fue evaluar y validar el correcto funcionamiento de todos los componentes implementados en la plataforma UXLab, los cuales forman parte de los objetivos definidos en esta tesis, los detalles de las pruebas realizadas se describen en la sección 5.1.1.
- **Pruebas emocionales**, el objetivo de estas pruebas fue evaluar y validar el método implementado para realizar el análisis de seguimiento ocular, en estas pruebas se valida la generación de las métricas establecidas en los objetivos: mapas de calor, rutas sácaras y áreas de interés mostrando su correlación con las emociones identificadas por los algoritmos implementados en la plataforma UXLab. los detalles de las pruebas realizadas se describen en la sección 5.1.2.
- **Pruebas cognitivas**, el objetivo de estas pruebas fue evaluar y validar el algoritmo implementado en esta tesis para analizar el parpadeo del usuario durante la realización de una actividad con diferentes niveles de carga cognitiva y validar la identificación de estados cognitivos de concentración o falta de atención en base al análisis de parpadeos. los detalles de las pruebas realizadas se describen en la sección 5.1.3.

5.1.1 Pruebas funcionales de UXLab

Las pruebas funcionales se enfocaron en las funcionalidades asociadas al módulo de seguimiento ocular implementado en la plataforma UXLab, sistema de grabación y sistema reproductor; estas funcionalidades se describen en la figura 65. Se incluye el

módulo de grabación de la memoria de video del sistema de grabación ya que se utiliza para generar métricas de seguimiento ocular.

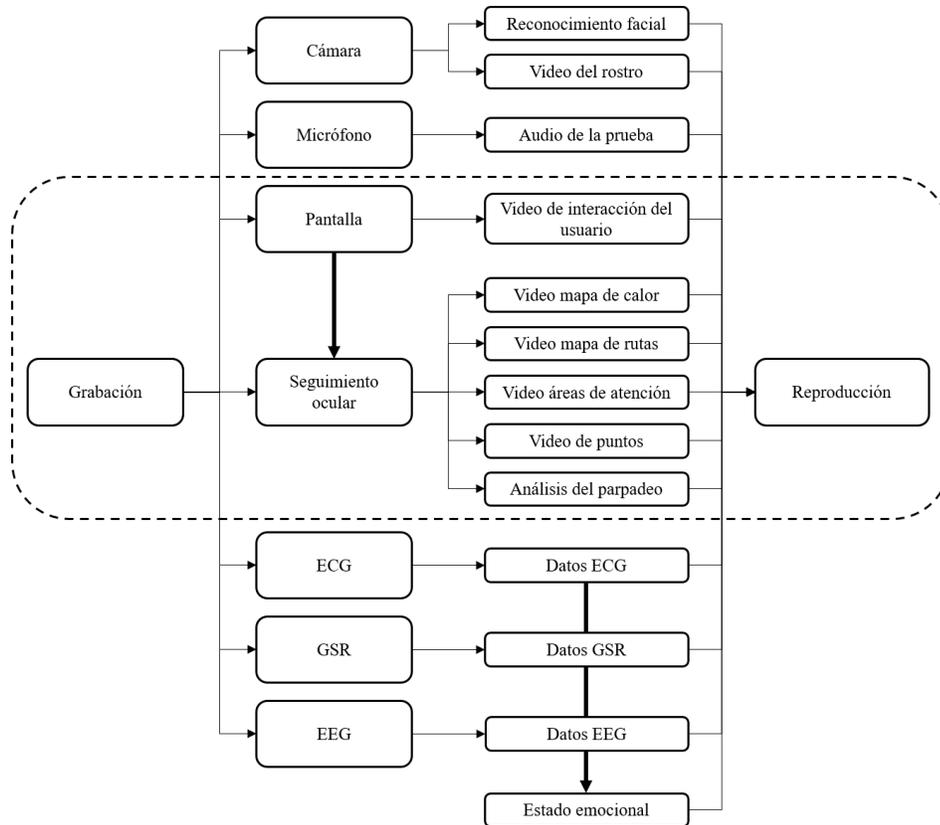


Figura 65: Esquema de funciones de seguimiento ocular implementadas en UXLab. Las funciones relacionadas con el módulo de seguimiento ocular se muestran dentro de la línea punteada.

Las pruebas aplicadas al módulo de seguimiento ocular contemplan cada una de las funciones y su integración con el software de grabación y de reproducción de UXLab (ver tabla 6).

Tabla 6: Plan de pruebas UXLab, módulo de seguimiento ocular.

ID de prueba	Función que se evalúa
UX-G-1	Conexión de UXLab con EyeWidget
UX-G-2	Representación de los ojos del usuario
UX-G-3	Comportamiento de los ojos en movimiento
UX-G-4	Comportamiento de los ojos con parpadeo

UX-G-5	Comportamiento de los ojos observando fuera de la pantalla
UX-G-6	Comportamiento con los ojos cerrados
UX-G-7	Grabación de archivo de puntos de visión
UX-G-8	Grabación de archivo de puntos de fijación
UX-G-9	Grabación de archivo de parpadeos
UX-G-10	Detener grabación
UX-G-11	Grabar de nuevo
UX-G-12	Desconexión entre UXLab y EyeWidget
UX-G-13	Procesamiento de video mapa de calor
UX-G-14	Procesamiento de video mapa de rutas
UX-G-15	Procesamiento de video áreas de interés
UX-G-16	Procesamiento de archivo de análisis del parpadeo
UX-G-17	Procesamiento de video de puntos
UX-I-1	Realizar grabación solo con seguimiento ocular y memoria de video
UX-I-2	Realizar grabación con todos los sensores
UX-R-1	Visualizar mapa de calor
UX-R-2	Visualizar mapa de rutas
UX-R-3	Visualizar puntos de visión
UX-R-4	Visualizar áreas de interés
UX-R-5	Visualizar gráfica de parpadeos
UX-R-6	Visualizar estados mentales sincronizados con métricas oculares
UX-I-3	Visualizar todas las métricas oculares en una sesión UX
UX-I-4	Grabar con todos los sensores y reproducir la sesión. Observando la sincronización de estados mentales con métricas oculares.

Nota. G = Grabador, R = Reproductor e I = Integración.

De acuerdo con las especificaciones de UXLab, para realizar pruebas de evaluación de la UX se requiere de un equipo de cómputo con ciertas características hardware y software. Para realizar las pruebas funcionales que se presentan en este capítulo se utilizó una computadora con las siguientes características:

- Sistema operativo Windows 10
- 64 gigabytes de memoria RAM
- Procesador Intel Core i7
- Más de 500 gigabytes de espacio en el disco duro

Las pruebas se realizaron utilizando los sensores adecuados para cada caso de prueba (Ver Tabla 6), finalmente se analizaron los resultados con el objetivo de encontrar y corregir posibles errores en el software.

Durante la ejecución de las pruebas se tomó cada caso de prueba (Ver Tabla 6) de manera individual, se ingresaron las entradas necesarias para probar el caso, posteriormente se observó el comportamiento del sistema y se documentaron los resultados. Después de ejecutar cada caso de prueba se reinició el sistema a un estado inicial (donde no se ha ingresado ninguna entrada) para iniciar el siguiente caso de prueba.

5.1.2 Pruebas emocionales

El objetivo de estas pruebas fue identificar la correlación que existe entre el movimiento ocular del usuario y las emociones que experimenta durante la exposición a estímulos visuales, cabe señalar que la identificación de emociones la realiza el sistema UXLab mediante el procesamiento de señales fisiológicas (ECG, GSR y EEG), este proceso no está considerado en los objetivos de este trabajo de tesis, los algoritmos se desarrollaron en los trabajos de (García, 2020) y (Morales, 2020) y se integraron en la plataforma UXLab. Las pruebas que se realizaron consistieron en presentar estímulos visuales durante cierto tiempo, tomando como base la Metodología de Proceso de Estados Emocionales (PIE), desarrollado por (Soriano, 2018), esta metodología se diseñó originalmente para ser aplicada en plataformas de realidad virtual inmersiva (RVI) que requieren como dispositivo de salida un visor (Headset) de RV, por lo que se realizó una adaptación para utilizar esta metodología con una plataforma convencional de escritorio utilizando como dispositivo de salida un monitor convencional, eliminando el visor (headset) de RV, con respecto al proceso de exposición de estímulos visuales, en la metodología se utilizan cinco imágenes en las cuatro etapas de inducción, para estas pruebas solo se utilizó una imagen por etapa, lo cual permitió identificar de manera más precisa el comportamiento del movimiento ocular del usuario.

Las pruebas de exposición a estímulos visuales se dividieron en dos grupos, cada uno de ellos define diferentes objetivos en el análisis del movimiento ocular del usuario y de la correlación con el estado emocional, los cuales se describen a continuación:

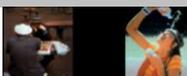
- Pruebas con una sola imagen con valencia y diferentes niveles de activación** (ver tabla 7). El objetivo de esta prueba es la observación del comportamiento ocular y la evaluación de las métricas generadas con los datos oculares en las pruebas. Se utilizó UXLab para obtener los archivos de datos de visión fijación y parpadeos de los ojos de cada usuario, además de los datos de todos los módulos de sensores de UXLab (EEG, GSR, ECG, video de rostro, video de pantalla y audio de la prueba), los datos fisiológicos fueron utilizados para identificar las emociones del usuario durante los diferentes lapsos de la prueba. Mediante la generación de las métricas oculares se observa el comportamiento del usuario relacionado a sus estados emocionales con el reproductor de UXLab. Al realizar estos experimentos se evalúan los algoritmos desarrollados de extracción y generación de métricas del seguimiento ocular.

Tabla 7: Ejemplo de estímulos de una sola imagen para las 5 emociones (Alegría, tristeza, neutra, relajación y ansiedad).

Emoción	Ejemplo de estímulo emocional
Alegría	
Tristeza	
Neutro	
Ansiedad	
Relajación	

- **Pruebas con imágenes mixtas con valencias diferentes y niveles de activación** (ver tabla 8). Las imágenes mixtas constan de 2 estímulos de las emociones: alegría, relajación, tristeza y ansiedad, estas imágenes son ordenadas al azar de tal manera que el orden en que parecen las emociones no afecte el comportamiento visual. El objetivo de esta prueba es igual al de la prueba anterior, correlacionar los movimientos oculares de los usuarios con el reporte de los demás sensores de UXLab en cada tipo de estímulo diferente. Mediante la generación de las métricas oculares se observa el comportamiento del usuario relacionado a sus estados emocionales con el reproductor de UXLab. Al realizar estos experimentos se evalúan los algoritmos desarrollados de extracción y generación de métricas del seguimiento ocular.

Tabla 8: Ejemplo de estímulos de imágenes mixtas para 4 emociones (Alegría, tristeza, relajación y ansiedad).

Emoción	Alegría	Tristeza	Ansiedad	Relajación
Alegría	-			
Tristeza		-		
Ansiedad			-	
Relajación				-

La Metodología PIE (Soriano, 2018), utilizada en las pruebas emocionales, se conforma de tres etapas o procesos, las cuales se describen en la figura 66.

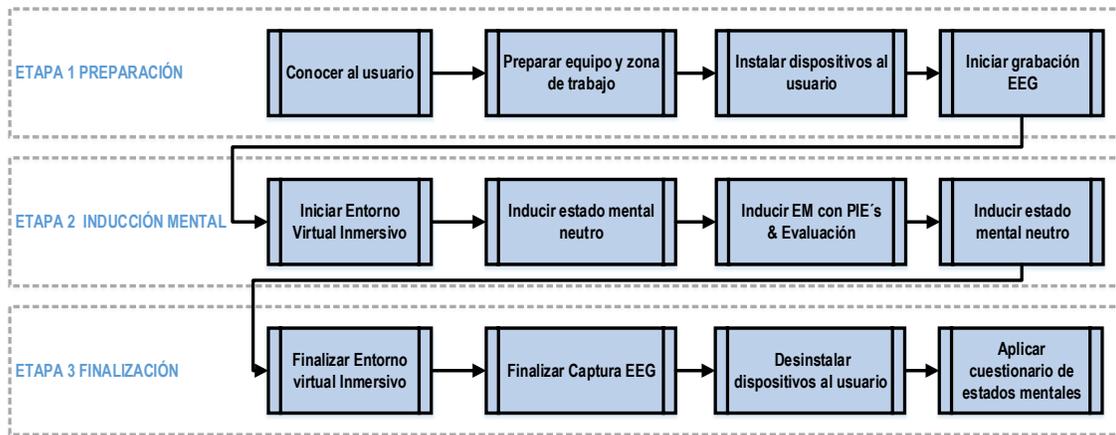


Figura 66: Diagrama del proceso de inducción emocional PIE. Se aplicó la metodología PIE adaptada a pantalla de computadora (sin visor de RVI) para poder grabar la atención visual del usuario en los estímulos visuales. Fuente: de Metodología para Caracterizar e Inducir Estados Cognitivos y Emocionales Mediante Realidad Virtual Inmersiva, J. Soriano, 2018, Todos los derechos reservados 2018, Tomado con permiso del autor.

Etapa de preparación

En esta etapa, se perfila al sujeto al que se le aplicará la inducción mental y se prepara la zona de trabajo donde se realizarán las pruebas de inducción de estados emocionales, ver figura 67. En esta etapa se realizan cinco actividades, 1) Conociendo al usuario, se aplican diferentes instrumentos para perfilar al usuario, cuestionario de perfil demográfico, cuestionario PHQ9 para identificar estados de ansiedad y el cuestionario STAI para identificar estados de depresión. 2) Se prepara el equipo y zona de trabajo, revisando las condiciones ergonómicas de la zona de trabajo, como iluminación, temperatura, nivel de ruido, entre otros. 3) Se instalan los sensores fisiológicos al sujeto de prueba verificando su correcto funcionamiento y se verifica que el sistema de grabación UXLab funcione correctamente y detecte todos los sensores utilizados en la prueba, entre los sensores instalados se encuentra el de seguimiento ocular con el que se registra el movimiento de los ojos del usuario, 4) Una vez que se verifican todas las condiciones se inicia la grabación de la sesión y 5) se inicia el ambiente de inducción PIE.

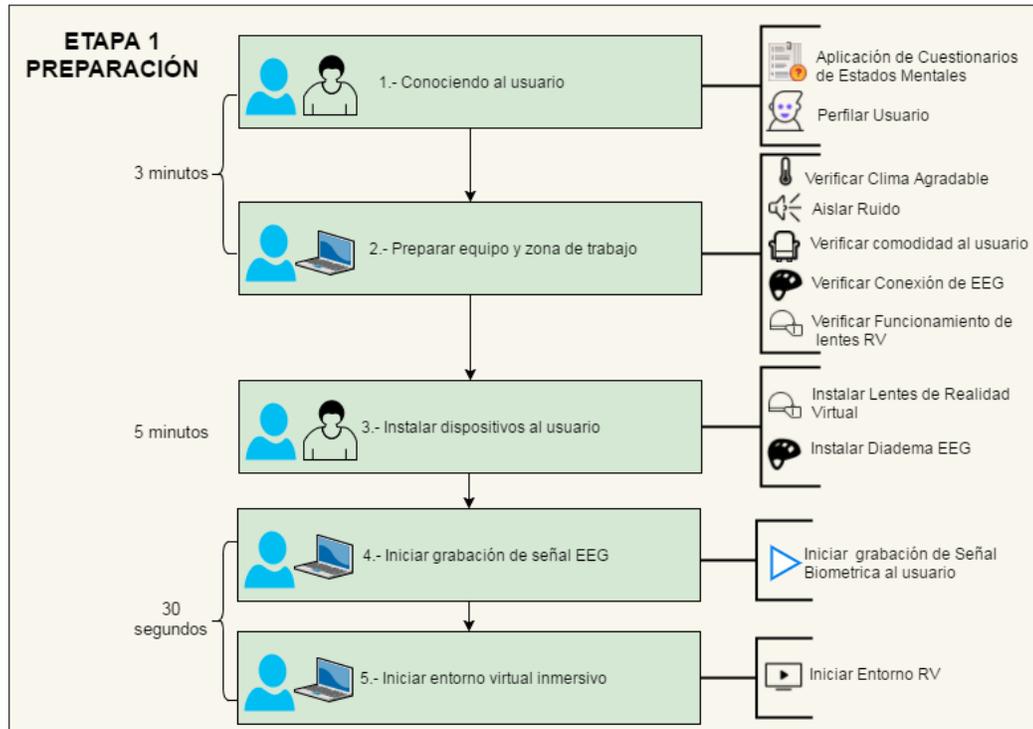


Figura 67: Esquema de la etapa 1 del proceso PIE para la inducción emocional. Fuente: Metodología para Caracterizar e Inducir Estados Cognitivos y Emocionales Mediante Realidad Virtual Inmersiva, J. Soriano, 2018, Todos los derechos reservados 2018, Tomado con permiso del autor.

Etapa de inducción mental

Esta etapa consiste en exponer al usuario a imágenes proporcionados por los repositorios IAPS (Lang, Bradley, y Cuthbert, 2008) y OASIS (Kurdi, Lozano, y Banaji, 2016) las cuales están etiquetadas con una valencia, emoción positiva o negativa y el nivel de intensidad, baja, media o alta, con tiempos de exposición establecidos por el PIE, como se describe en la figura 68. Las actividades que se realizan en esta etapa son: 6) Se induce un estado neutral al usuario mediante una línea base, con duración de dos minutos, 7) Se expone al usuario a un estímulo visual durante 15 segundos, dependiendo del estado emocional que se requiera inducir, esto se realiza cuatro veces, cada una con diferente intensidad, y finalmente, 8) se termina el proceso de estímulo visual con un estado neutro.

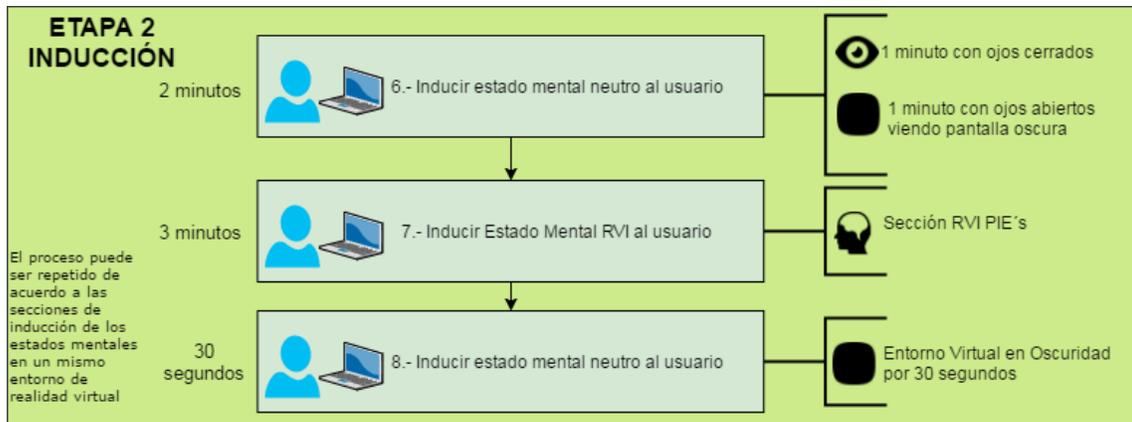


Figura 68: Esquema de la etapa 2 del proceso PIE para la inducción emocional. Fuente: Metodología para Caracterizar e Inducir Estados Cognitivos y Emocionales Mediante Realidad Virtual Inmersiva, J. Soriano, 2018, Todos los derechos reservados 2018, Tomado con permiso del autor.

La figura 69 describe la etapa 2 de la inducción emocional.

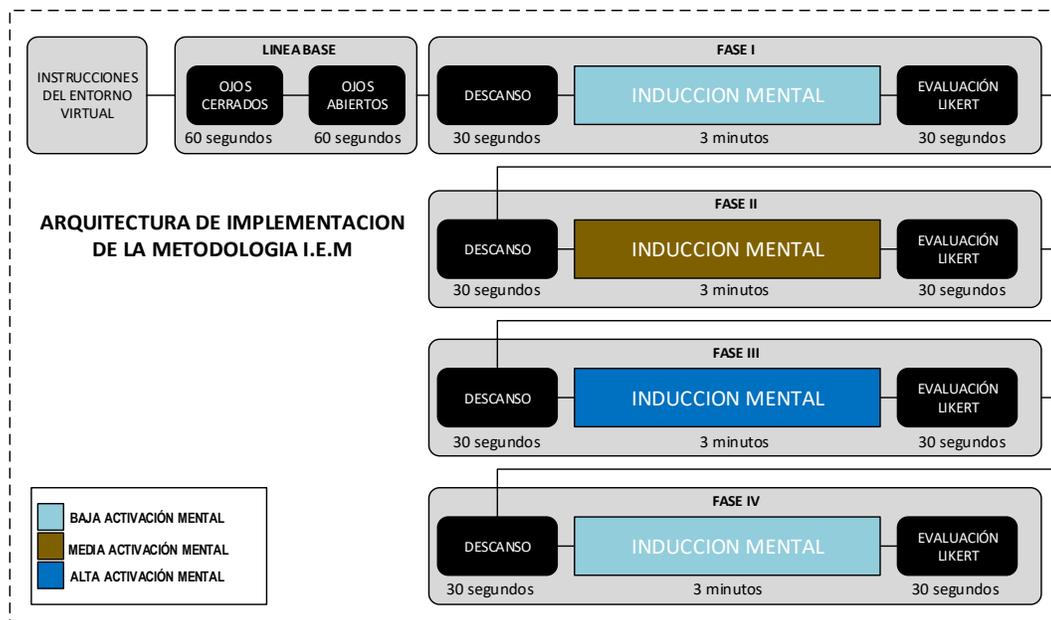


Figura 69: Esquema de proceso de implementación de inducción emocional PIE. Fuente: Metodología para Caracterizar e Inducir Estados Cognitivos y Emocionales Mediante Realidad Virtual Inmersiva, J. Soriano, 2018, Todos los derechos reservados 2018, Tomado con permiso del autor.

Para esta tesis se modificó y adaptó este procedimiento para realizar el análisis de movimiento ocular del usuario en pantalla, como se describe en la tabla 9, en esta etapa

los sujetos de prueba realizan las tres actividades descritas en la figura 68, con tiempos y contenidos adaptados.

Tabla 9: Adaptación de la etapa 2 del proceso PIE para entornos no inmersivos.

Actividad	Descripción
1	Instrucciones del entorno (38 segundos) , esta actividad es igual en tiempo y características al proceso original.
2	Línea base (1 minuto) , en esta actividad se reduce el tiempo de la línea base a 1 minuto, 30 segundos de ojos cerrados y 30 segundos de ojos abiertos.
3	<p>Inducción mental en cuatro fases con diferente intensidad:</p> <p>Descanso (15 segundos), en esta sección no hay estímulos visuales, se utiliza para realizar la transición entre las fases y sus intensidades.</p> <p>A diferencia del proceso original cada fase muestra solo un estímulo visual durante 15 segundos, en cada fase la intensidad varía: bajo, medio, alto. El tiempo de 15 segundos por estímulo es suficiente para generar un análisis de los datos oculares en cada estímulo.</p> <ul style="list-style-type: none"> • Fase I (15 segundos), se presenta el primer estímulo emocional, de intensidad baja. • Fase II (15 segundos), se presenta el segundo estímulo emocional, de intensidad media. • Fase III (15 segundos), se presenta el primer estímulo emocional, de intensidad alta. • Fase IV (15 segundos), se presenta el primer estímulo emocional, de intensidad baja, para devolver al usuario a línea base (neutralidad). <p>Evaluación Likert (20 segundos), Al final de cada fase el usuario responde una evaluación SAM (Bradley y Lang, 1994), esta utiliza maniqués para la auto evaluación del estado emocional en tres dimensiones: Valencia, Activación y Dominio:</p>

¿Cómo te sentiste cuando viste o escuchaste el contenido multimedia?

En esta sección de la prueba el usuario cuenta con 20 segundos para realizar la autoevaluación, estos datos son recopilados y usados para correlacionar el tipo de emoción inducida con el sentimiento del usuario.

Etapa de finalización

La etapa de inducción de emociones finaliza, cuando el escenario virtual indica que la prueba ha concluido, en este momento el evaluador tiene que realizar la etapa de finalización, en la figura 70, se describe la etapa 3, en esta etapa se realizan cuatro actividades, 9) Se finaliza el entorno de inducción, 10) Se finaliza la grabación con el módulo de UXLab, 11) Se retiran todos los sensores fisiológicos instalados en el sujeto de prueba (EEG, ECG y GSR) y finalmente 12) Se aplican cuestionario PANAS (Watson, D., Clark, L. A., y Tellegen, A., 1988) para registrar los estados emocionales inducidos en el sujeto de prueba.

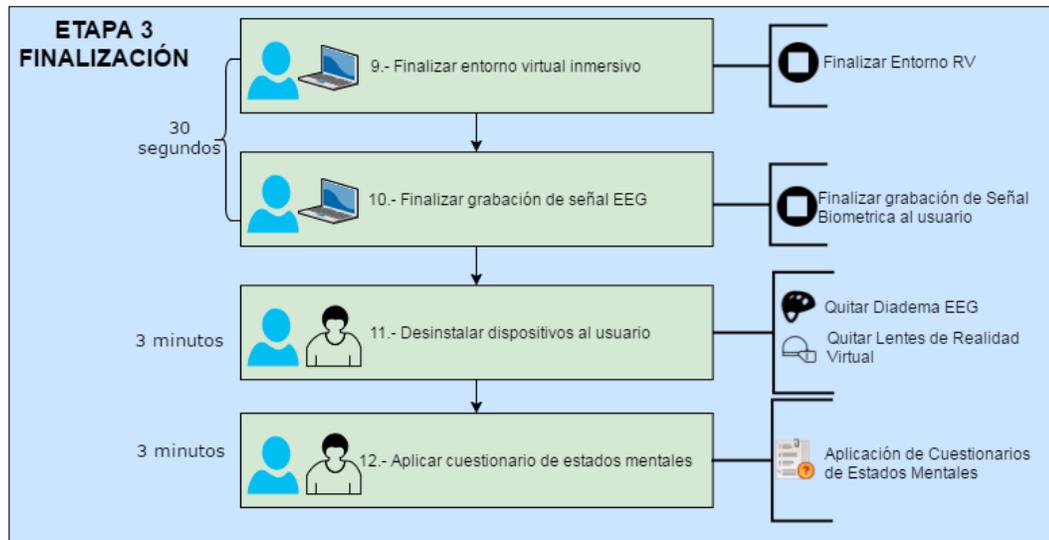


Figura 70: Esquema de la etapa 3 del proceso PIE para la inducción emocional. Fuente: Metodología para Caracterizar e Inducir Estados Cognitivos y Emocionales Mediante Realidad Virtual Inmersiva, J. Soriano, 2018, Todos los derechos reservados 2018, Tomado con permiso del autor.

Conformación del repositorio de datos de prueba

Para la generación de los datos de las pruebas emocionales utilizando la metodología PIE, se tomaron las siguientes consideraciones para la conformación del grupo de sujetos de prueba:

- La muestra de sujetos de prueba fue de 10, 5 hombres y 5 mujeres. Con un rango de edad de entre 20 a 40 años y nivel de escolaridad de educación superior y postgrado.
- Los estímulos emocionales mixtos deben aparecer en orden aleatorio para cada participante.
- Para la conformación del conjunto de imágenes que se utilizaron en el PIE se probaron cinco estímulos, dos emociones con valencia alta (o emociones positivas), dos emociones con valencia baja (o emociones negativas) y estímulos con valencia neutra.
- Cada estímulo estará compuesto de cuatro imágenes emocionales, cada fase del estímulo aumentará en 1 el nivel de intensidad del estímulo desde nivel bajo luego medio luego alto y terminará con nivel bajo.

La duración de la prueba fue de aproximadamente 12 minutos, la figura 71 muestra el proceso de la prueba de inducción emocional con los cinco estímulos.

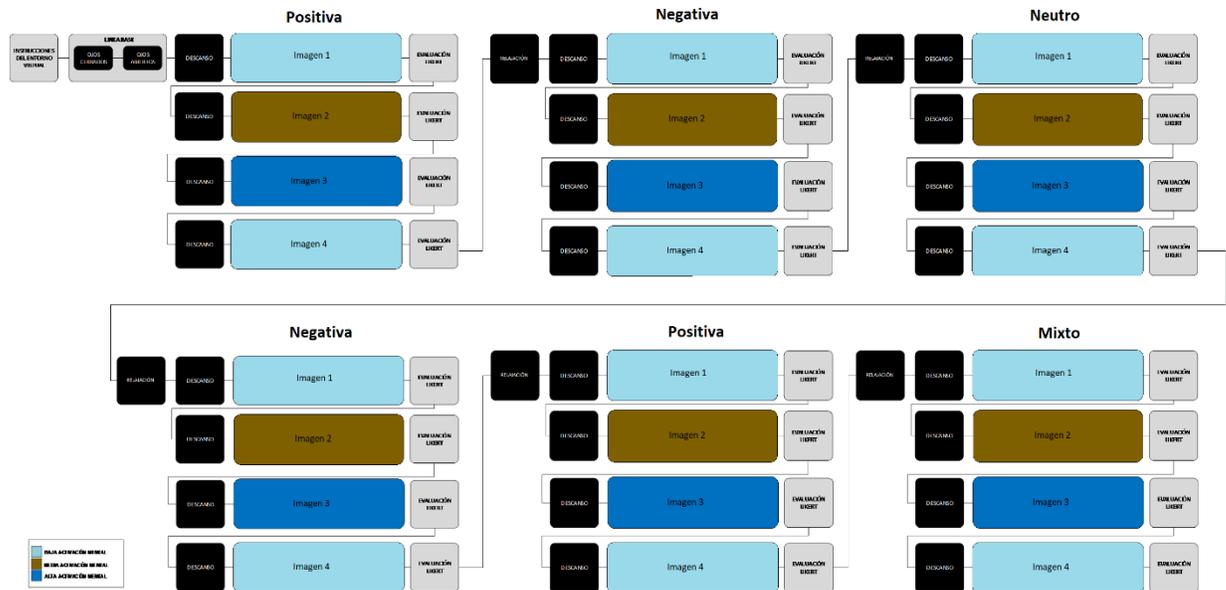


Figura 71: Proceso de pruebas emocionales adaptado del PIE. Fuente: Metodología para caracterizar e inducir estados cognitivos y emocionales mediante realidad virtual inmersiva, J. Soriano, 2018, Todos los derechos reservados 2018, Tomado con permiso del autor.

A continuación, en la tabla 10, se describen las pruebas a realizar.

Tabla 10: Plan de pruebas emocionales UXLab, módulo de seguimiento ocular.

ID de prueba	Prueba
UX-E-1	Estímulo felicidad
UX-E-2	Estímulo ansiedad
UX-E-3	Estímulo tristeza
UX-E-4	Estímulo neutro
UX-E-5	Estímulo relajación
UX-E-6	Estímulo mixto

Se utilizaron dos repositorios de imágenes con carga emocional, IAPS (Moltó et al., 2013; Lang, et al., 2008) y OASIS (Kurdi S., Benedek B. y Lozano M. 2016).

La base de datos de estímulos emocionales IAPS (Lang, et al., 2008) contiene estímulos en imágenes que inducen distintas emociones. Originalmente este repositorio de imágenes fue probado en países de habla inglesa y se adaptó en España. Y se utiliza en el laboratorio de Sistemas Híbridos Inteligentes del TecNM/CENIDET para diferentes pruebas e investigaciones.

Los estímulos IAPS (Lang, et al., 2008) son un conjunto de imágenes (ver figura 72). con diferentes categorías emocionales clasificadas en tres dimensiones, valencia, activación y dominio.



Figura 72: Algunos estímulos de base de datos IAPS. Fuente: International affective picture system (IAPS): Affective ratings of pictures and instruction manual, por Lang, P.J., Bradley, M.M., y Cuthbert, B.N., 2008.

El repositorio IAPS cuenta con un cerca de 700 estímulos visuales cada uno definido por un valor en valencia, activación y dominio. Los estímulos IAPS seleccionados se tomaron en función de su intensidad desde baja hasta alta, para las pruebas con IAPS fueron seleccionados 4 estímulos de cada emoción: alegría, tristeza, neutro, ansiedad (ver tabla 11)

Tabla 11: Imágenes de IAPS seleccionadas para la inducción.

Imágenes IAPS	Emoción	Valencia	Activación
	Alegría	Positiva	Baja

	Tristeza	Negativa	Baja
	Ansiedad	Negativa	Alta
	Neutro	Neutra	Neutra

Nota. Imágenes tomadas de International affective picture system (IAPS): Affective ratings of pictures and instruction manual, por Lang, P.J., Bradley, M.M., y Cuthbert, B.N., 2008.

El repositorio de imágenes OASIS (Kurdi S., Benedek B. y Lozano M. 2016) cuenta con una gran variedad de estímulos divididos en cuatro clases y con diversidad de valores para valencia y activación (ver figura 73).



Figura 73: Algunos estímulos de base de datos OASIS. Fuente: Open Affective Standardized Image Set, por S. Kurdi, B. Benedek y M. Lozano, 2016 (<http://www.benedekkurdi.com/#oasis>), Repositorio de Dominio Público.

Oasis contiene más de 900 estímulos clasificados en cuatro clases: personas, animales, objetos y escenarios, como se muestra en la figura 74. Cada uno de los estímulos cuenta con un valor de dos dimensiones: valencia y activación, lo cual pone a cada estímulo en un sector del modelo de valencia activación: Alegría, Relajación, Neutro y Tristeza.

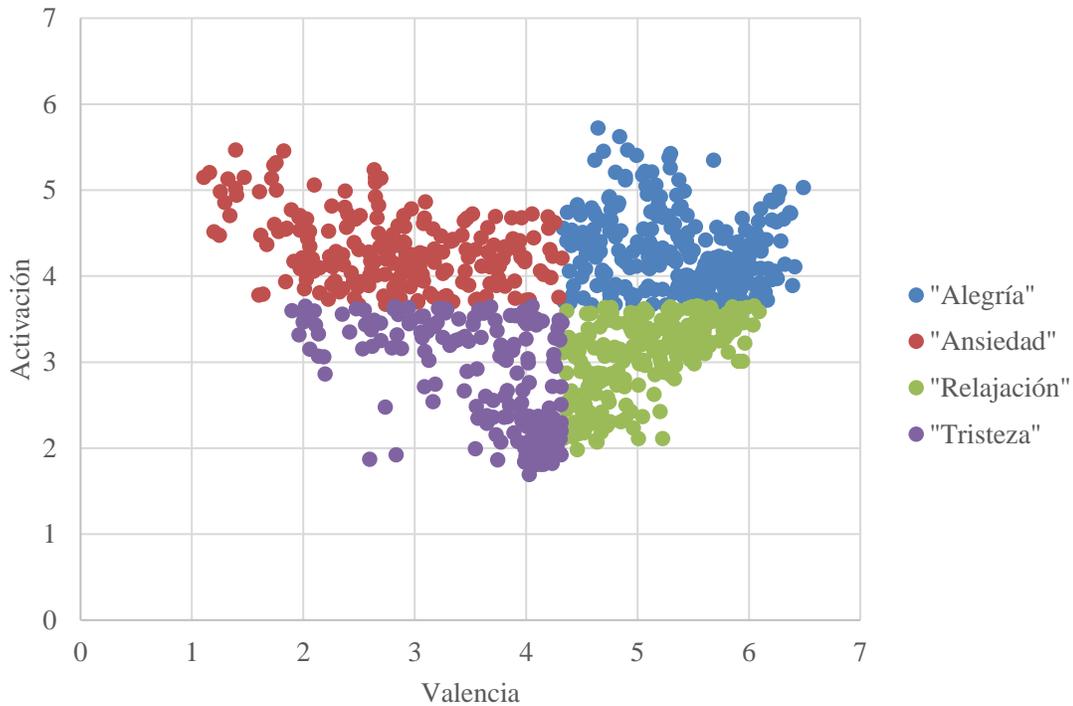


Figura 74: Valores de valencia y activación para los estímulos de imágenes del Repositorio OASIS. La gráfica representa los valores de valencia- activación de cada uno de los estímulos visuales del repositorio OASIS. Adaptado de Open Affective Standardized Image Set, por S. Kurdi, B. Benedek y M. Lozano, 2016 (<http://www.benedekkurdi.com/#oasis>), Repositorio de Dominio Público.

En la tabla 12, se describen los estímulos que se eligieron para las pruebas; se le dio prioridad al valor de la valencia debido a que el algoritmo de clasificación implementado en UXLab identifica dos clases de emociones: positivas y negativas. Las imágenes seleccionadas se clasificaron en función del valor de la valencia desde “baja” (2 imágenes) a “media” para terminar con un valor “alto” (como se especifica en el proceso de inducción emocional PIE).

Tabla 12: Imágenes de OASIS seleccionadas para la inducción.

Imágenes Oasis	Clase	Emoción	Valencia	Activación
	Persona	Relajación	Positiva	Alta

	Animal	Alegría	Positiva	Baja
	Objeto	Tristeza	Negativa	Baja
	Animal	Ansiedad	Negativa	Alta
	Variado	Neutro	Neutra	Neutra

Nota. Imágenes tomadas de Open Affective Standardized Image Set, por S. Kurdi, B. Benedek y M. Lozano, 2016 (<http://www.benedekkurdi.com/#oasis>), Repositorio de Dominio Público.

En la figura 75 se muestran los estímulos OASIS en un plano de valencia – activación:

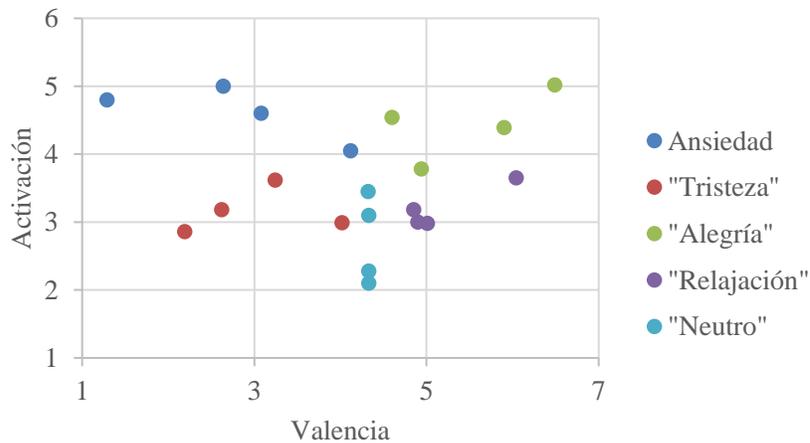


Figura 75: Valores de valencia y activación para los estímulos a usar en las pruebas. La gráfica representa el valor de valencia – activación de las imágenes seleccionadas para realizar las pruebas emocionales, ver tabla 12, los valores valencia –activación fueron definidos en el estudio Introducing the Open Affective Standardized Image Set (OASIS), de S. Kurdi, B. Benedek y M. Lozano, 2016.

Para realizar las pruebas se había considerado inicialmente un grupo de prueba conformado por diez usuarios, sin embargo, debido a la contingencia derivada del COVID-19 solo se logró la participación de ocho sujetos de prueba (Ver Tabla 13). Aunque la muestra se redujo a ocho, cada uno de los 10 usuarios seleccionados en un principio respondió los cuestionarios PHQ-9 y STAI (etapa 1 de la metodología PIE) para conocer el estado de depresión y ansiedad de los usuarios además de responder un breve cuestionario sobre información básica perfil sociodemográfico, de acuerdo la metodología PIE (Soriano, 2018).

Tabla 13: Usuarios para pruebas emocionales.

Número	Edad	Sexo
1	24	Femenino
2	26	Femenino
3	26	Femenino
4	30	Femenino
5	27	Masculino
6	26	Masculino
7	25	Masculino
8	34	Masculino

Nota. Todos los participantes son estudiantes de postgrado en ciencias de la computación.

La figura 76 muestra los resultados del cuestionario PHQ-9 (Kroenke, Spitzer y Williams, 2001), el cual trata de medir la depresión de los usuarios a través de preguntas donde se autoevalúa el sentir del usuario durante la semana, mientras más preguntas sean contestadas con “casi todos los días” mayor es el nivel de depresión, en la gráfica se puede observar que los sujetos de prueba contestaron más veces “Ningún día” y “Varios días” lo cual no indica problemas de depresión en general.

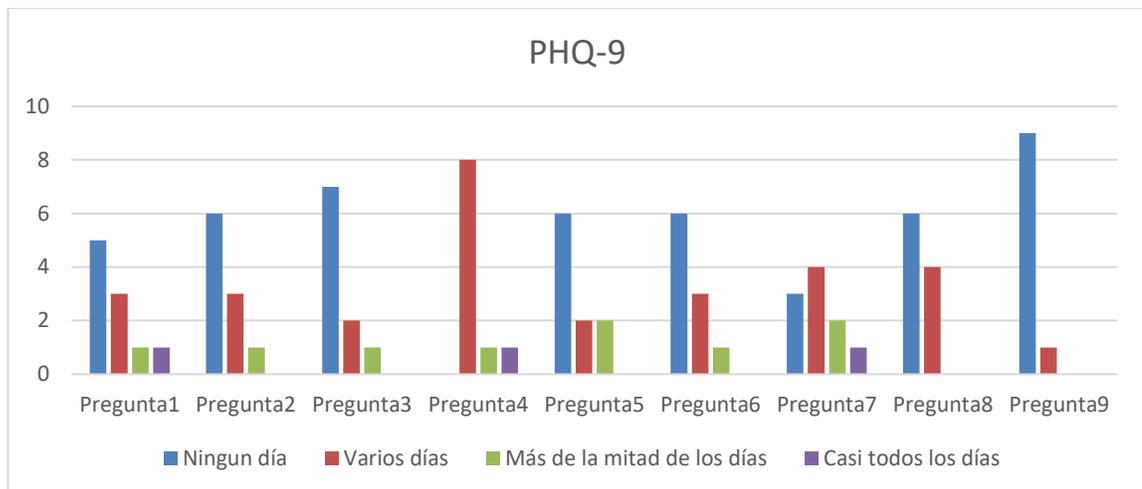


Figura 76: Gráfica de respuestas a las preguntas del cuestionario PHQ-9.

Cada respuesta de los usuarios es ponderada:

- Ningún día = 0
- Varios días = 1
- Más de la mitad de los días = 2
- Casi todos los días = 3

Finalmente se realiza la sumatoria de las respuestas para determinar un valor:

- 0 – 4 = sin depresión
- 5 – 9 = síntomas mínimos
- 10 – 14 = síntomas presentes
- 15 – 19 = síntomas moderados
- 20 o más = síntomas de depresión severos

La figura 77 muestra los resultados de los 10 participantes, Algunos participantes mostraron síntomas de depresión presentes. Para este tipo de estudio se buscaba filtrar usuarios con una puntuación de 15 o más lo cual no fue el caso.

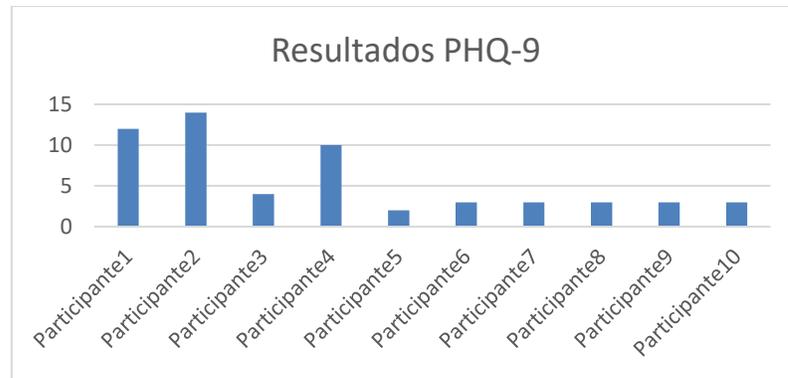


Figura 77: Resultados de la prueba PHQ-9 para los 10 participantes. Esta gráfica representa la puntuación para cada participante en la prueba PHQ-9.

El cuestionario STAI (Spielberger, Gorsuch, Lushene, Vagg, y Jacobs, 1983) es una herramienta para medir el nivel de ansiedad basada en una escala Likert y consta de preguntas positivas y negativas, en forma de autoinforme. Los puntajes más altos se correlacionan positivamente con niveles más altos de ansiedad.

La figura 78 muestra los resultados del cuestionario para los 10 usuarios de las pruebas, puede observarse que la mayoría de las respuestas negativas fueron contestadas con *casi nunca* y *a veces*, mientras que las respuestas positivas tienen respuestas *A menudo* y *siempre*.

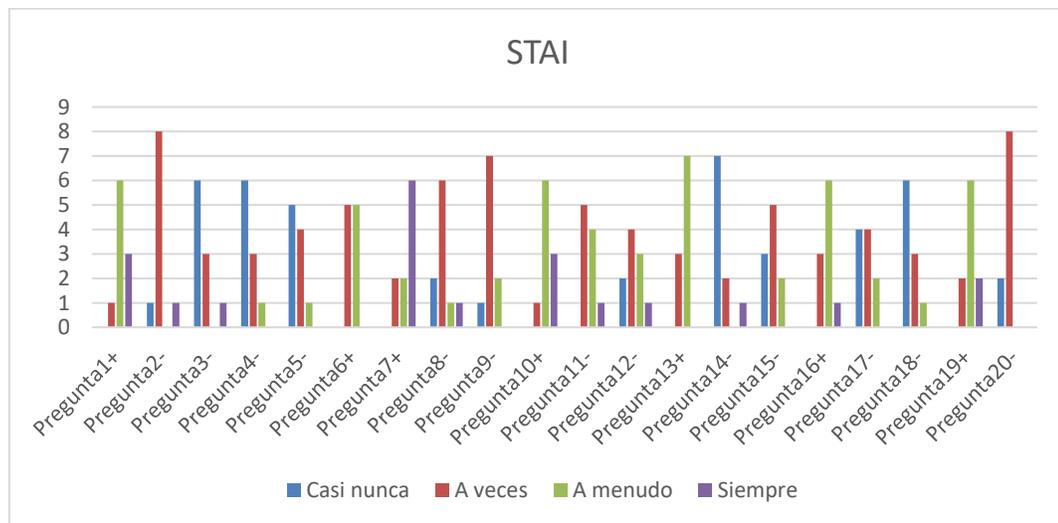


Figura 78: Gráfica de respuestas a las preguntas del cuestionario STAI. Se espera que las preguntas positivas (+) sean contestadas con “a menudo” o “siempre” mientras que las preguntas negativas (-) sean contestadas con “casi nunca” o “a veces”.

No se identificaron problemas en ninguno de los usuarios, en los datos demográficos se muestra que todos los usuarios tienen experiencia con tecnologías. De acuerdo con la metodología PIE (Soriano, 2018), se establecen las siguientes consideraciones para las pruebas emocionales:

5.1.2.1 Contexto de pruebas

Consideraciones ambientales y área de pruebas:

- Se recomienda utilizar una habitación aislada de ruido e interferencias de personas.
- Se recomienda que el usuario que realizará la evaluación sea el único presente en el espacio/lugar para realiza la prueba/evaluación.
- Se recomienda evitar todo ruido externo o interno que distraiga al usuario.
- Se recomienda que la temperatura ambiente mínima sea de 22°C.
- Se recomienda utilizar la misma silla con descansabrazos para todas las evaluaciones.
- Se recomienda garantizar un nivel de luz apropiado y controlado.
- El evaluador debe evitar el contacto físico o visual con el usuario durante la prueba.
- Para el monitoreo de la grabación del usuario el evaluador deberá utilizar otro monitor o computadora de ser necesario.

Consideraciones de software:

- Tener instalado y funcionando el programa EyeWidget.
- Tener instalado y funcionando el programa UXLab grabador.
- Tener instalado y funcionando el programa UXLab reproductor.
- Tener instalado y funcionando el producto digital a evaluar.

Consideraciones de hardware:

- Se debe utilizar una pantalla entre las 22 a 32 pulgadas.
- La pantalla debe tener de preferencia una resolución de 1920 x 1080 pixeles.
- Se debe ajustar la intensidad de brillo del monitor para que sea cómodo para el sujeto de pruebas.

- Para el sensor de seguimiento ocular la distancia entre recomendada debe ser entre 45 a 100 centímetros, los ojos del usuario deben estar de frente al sensor, evitar el uso de lentes.
- Verificar que no se encuentre un modem, router, antena, etc., en la misma habitación donde se realizarán las pruebas, para evitar que los sensores fisiológicos reciban campos magnéticos externos.
- Para los sensores de datos fisiológicos se debe tomar en cuenta las siguientes consideraciones:
 - Para el uso adecuado del BCI EMOTV Flex se recomienda medir el diámetro encefálico del usuario, se cuentan con tres medidas: 54, 56 y 58 cms.
 - Realizar una carga completa del BCI EMOTV Flex antes de realizar las pruebas.
 - Verificar que no estén dañados los electros, las terminales de conexión y cables de los diferentes sensores.
 - Verificar en la pantalla del software UXLab que todos los sensores a utilizar estén correctamente conectados y recibiendo datos.

Consideraciones del sujeto de prueba:

- Previo a realizar las pruebas, al usuario se le debieron aplicar las pruebas y cuestionarios definidos en el PIE (Soriano, 2018).
- Verificar que el usuario esté de acuerdo en realizar las pruebas y de haber firmado el consentimiento de estas en el formato definido por la metodología.
- El usuario debe estar informado de las características y del entorno a utilizar durante las pruebas.
- Se debe solicitar al usuario que acuda a las pruebas en un lapso de dos a tres horas después del desayuno o de la comida.
- Se debe solicitar al usuario no utilizar ningún producto para el cabello antes de la prueba (gel, ceras, cremas, aerosol, etc.)
- El cabello del usuario debe estar seco y de preferencia corto o un muy grueso.
- Tratar que el usuario se encuentre relajado y tranquilo antes de iniciar la prueba.

Se utilizó el sistema de grabación UXLab (Ver figura 79) para registrar los datos de los usuarios, se registraron datos fisiológicos para identificar su estado emocional y su comportamiento ocular (utilizando el sensor de seguimiento ocular).

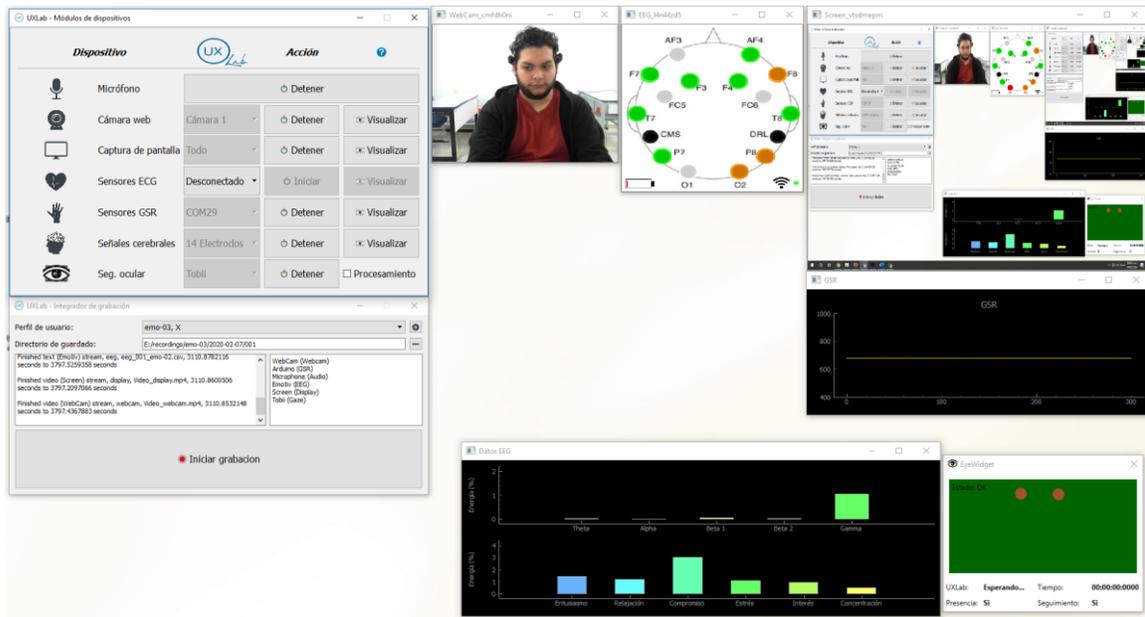


Figura 79: Pantalla del módulo de grabación UXLab monitoreando Usuario.

5.1.3 Pruebas de estados cognitivos

Para comprobar el cambio en el estado cognitivo y de atención con el software UXLab, se evaluó el análisis del parpadeo de los usuarios. Se utilizó como base de la prueba, una interacción con un videojuego que tiene diferentes niveles de dificultad, los diferentes niveles de complejidad permiten generar escenarios con carga de trabajo cognitiva.

El videojuego que se utilizó fue “Osu” (Herbert, 2007), el cual está clasificado como un juego de ritmo en donde el usuario debe seguir patrones dando clic en el momento justo siguiendo el ritmo de una melodía o canción. Se eligió este juego ya que es fácil de utilizar, cuenta con diferentes modos de juego que se adaptan al estilo de los jugadores, permite personalizar el nivel de dificultad mediante modificaciones y además jugar un nivel del juego es relativamente corto (de 2 a 3 minutos).

En la tabla 14 se especifican las 3 pruebas cognitivas planeadas para las pruebas del software UXLab, el objetivo de estas pruebas fue comprobar la correlación entre el estado cognitivo de atención de los usuarios utilizando UXLab.

Tabla 14: Plan de pruebas cognitivas UXLab, módulo de seguimiento ocular.

ID de prueba	Prueba
UX-C-1	Estímulo de baja carga cognitiva
UX-C-2	Estímulo de alta carga cognitiva
UX-C-3	Estímulo de baja carga cognitiva a alta carga cognitiva

De acuerdo con los estudios de carga cognitiva desarrollados por Zagermann, Pfeil, y Reiterer (2018), Mellick et al. (2016) y Wang, Yang, Liu, Cao, y Ma (2014), se realizaron experimentos para observar distintos comportamientos oculares de sujetos de prueba mientras realizan algún tipo de tarea.

Los autores reportan cambios entre algunas métricas oculares conforme aumenta el nivel de carga cognitiva de las tareas a realizar. Dentro de la investigación realizada para este trabajo de tesis, se encontró una relación inversa entre el nivel de carga cognitiva y la frecuencia de parpadeos de una persona. Por lo anterior, se diseñó un modelo para analizar la frecuencia del parpadeo de los usuarios y medir su nivel de atención de acuerdo con las métricas identificadas en la literatura.

Con el desarrollo de estas pruebas se observaron las diferencias entre la frecuencia del parpadeo en diferentes niveles de dificultad. Las pruebas cognitivas consistieron en observar el estado de atención que experimentó un usuario en función de donde se encuentra su atención visual en ese momento, por ejemplo, cuando el usuario realiza una actividad que requiere de mayor atención visual se verá impactado en la reducción de su frecuencia de parpadeo.

Durante la ejecución de las pruebas se consideró cada caso de prueba de manera individual, se definieron las entradas necesarias para probar el caso, posterior a eso se observó el comportamiento del sistema y se registraron los resultados.

Después de ejecutar cada caso de prueba se reinició el sistema a un estado inicial o neutro (donde no se ha ingresado ninguna entrada) para iniciar el siguiente caso de prueba.

Durante la ejecución de las pruebas se decidió utilizar un modo de juego especial de Osu! (Herbert, 2007), el cual es más simple de jugar que el modo normal, ya que la mayoría de los usuarios tuvieron dificultades para aprender y adaptarse al juego. Este modo de juego es Osu Catch (Herbert, 2007), consiste en mover un personaje en la parte inferior de la pantalla de lado a lado para atrapar diferentes frutas de colores al ritmo de la música.

La mayoría de los usuarios escogió una canción diferente y jugaron 2 o más niveles de dificultad. Antes del experimento los usuarios jugaron un nivel para entender las mecánicas del juego.

La forma en que el videojuego calcula el desempeño de los jugadores es mediante dos métricas: desempeño y precisión.

El desempeño se calcula en función a la barra de vida del jugador durante todo el tiempo que dure el nivel. La barra de vida a su vez se calcula en función del error (dos valores que representan cuán lejos estaban los primeros puntos en promedio y cuán lejos estaban los puntos finales en promedio) y la tasa de inestabilidad, que representa la consistencia de tiempo de los puntos.

La precisión (9) se calcula tomando el total de objetos (frutas) recogidos que no sean plátanos dividido por el número total de objetos que no sean plátanos. Todos los objetos tienen el mismo valor, a excepción de los plátanos ya que son objetos de puntos extras.

$$Precisión = \frac{n_{objetos\ atrapados}}{n_{fallos} + n_{objetos\ atrapados}} \quad (9)$$

Donde:

- $n_{objetos\ atrapados}$ = Número de objetos atrapados (frutas o pedazos de frutas, llamados gotas dentro del juego) por el jugador.
- n_{fallos} = Número de objetos (frutas o pedazos de frutas, llamados gotas dentro del juego) que no fueron atrapados por el jugador.

5.2 Ejecución de las pruebas y reporte de resultados

El resultado de las pruebas se presenta en tres secciones correspondientes a 1) pruebas funcionales del sistema UXLab, 2) Pruebas emocionales y 3) Pruebas cognitivas.

5.2.1 Pruebas funcionales

Las pruebas funcionales, se realizaron para evaluar y validar el correcto funcionamiento de los módulos de seguimiento ocular implementados en la plataforma UXLab, los resultados y evidencias de las pruebas funcionales se describen en el anexo 1.

El resultado obtenido en estas pruebas fue satisfactorio, no se identificaron errores de implementación en los algoritmos y tampoco se encontraron errores funcionales con la integración de estos componentes en los dos componentes, el sistema de grabación y el sistema de reproducción de UXLab. Finalmente, como conclusión de las pruebas funcionales aplicadas a los componentes de UXLab en los que se implementaron los módulos de seguimiento ocular funciona correctamente, los datos generados son válidos y la integración con UXLab se realizó de manera adecuada.

5.2.2 Pruebas emocionales

Se realizaron dos iteraciones de pruebas emocionales aplicando la metodología PIE descrita en la sección 5.1.2 , 1) en la primera iteración se utilizaron imágenes de estímulos emocionales del repositorio IAPS (Lang, Bradley, y Cuthbert, 2008) y 2) en la segunda iteración se utilizaron imágenes de estímulos emocionales del repositorio OASIS (Kurdi, Lozano, y Banaji, 2016). Las dos iteraciones de pruebas son similares, difieren en algunas consideraciones del contexto de pruebas y en la duración de exposición a los estímulos visuales. Sin embargo, los resultados de las dos pruebas muestran el comportamiento ocular de los usuarios al ser expuestos a distintos estímulos con cargas emocionales diferentes.

Como se especifica en el proceso del PIE, al final de cada serie de imágenes los usuarios respondieron una evaluación SAM (Bradley y Lang, 1994) para registrar las emocionales percibidas después de haber recibido la inducción emocional mediante estímulos visuales.

5.2.2.1 Primera iteración

Al realizar las pruebas emocionales se tomaron muestras de cuatro usuarios diferentes, dos hombres y dos mujeres, todos los usuarios son estudiantes de posgrado con una media de edad de 28 años.

Debido a que los estímulos utilizados en esta iteración son parte del repositorio de imágenes emocionales IAPS (Lang, Bradley, y Cuthbert, 2008), las imágenes tienen restricción para su publicación, sin embargo, en esta tesis se incluirán algunas imágenes con ciertas modificaciones de sobreposición de mapas de calor y áreas de interés con la intención de mantener la poca exposición de estos contenidos visuales. La tabla 15 muestra los resultados obtenidos de las pruebas.

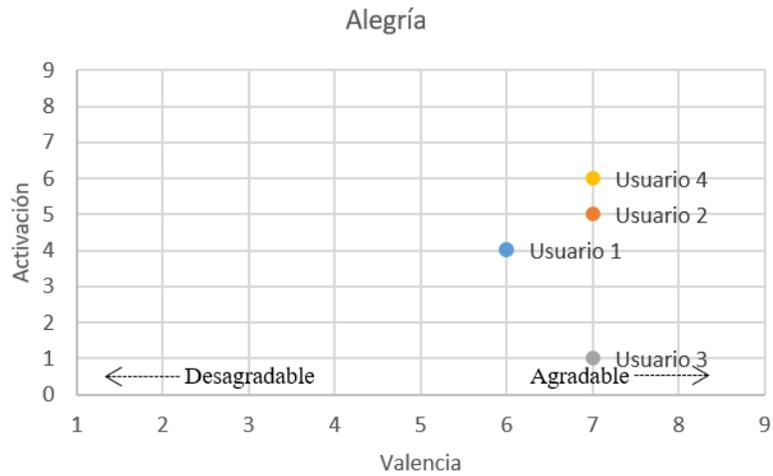
La recopilación de los datos oculares de estas pruebas se obtiene mediante la herramienta UXLab. Mientras que, los datos de las evaluaciones SAM fueron recopilados directamente de los usuarios durante el proceso de inducción de emociones, como se menciona en la metodología PIE (Soriano, 2018).

Tabla 15: Reporte de primera iteración de pruebas emocionales UXLab.

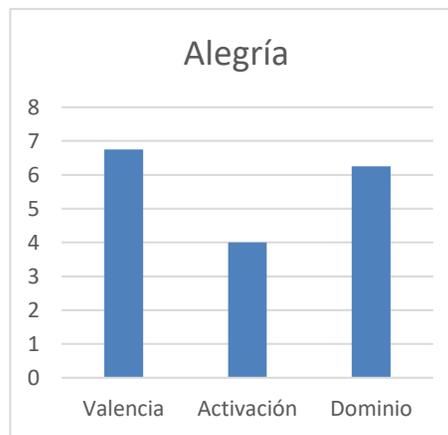
<p>UX-E-1</p>	<p>Estímulo de felicidad:</p> <p>Un análisis de los patrones oculares (utilizando mapas de calor y áreas de interés) muestra que los usuarios concentran su mirada en rostros de personas y animales y en algunos objetos, antes que en los alrededores o en el paisaje de fondo lo cual empata bien con lo establecido en la literatura (Iskra y Gabrijelčić Tomc, 2016).</p>
<p>Evidencia</p>	<div data-bbox="574 1291 1256 1747" data-label="Image"> </div> <p>Los usuarios reportaron una emoción positiva, más alta del valor neutro en la escala de la auto evaluación SAM, la siguiente tabla muestra los valores del instrumento SAM para los participantes:</p>

Alegría			
Usuario	Valencia	Activación	Dominio
1	6	4	6
2	7	5	7
3	7	1	7
4	7	6	5

Dispersión de datos de valencia y activación por usuario:

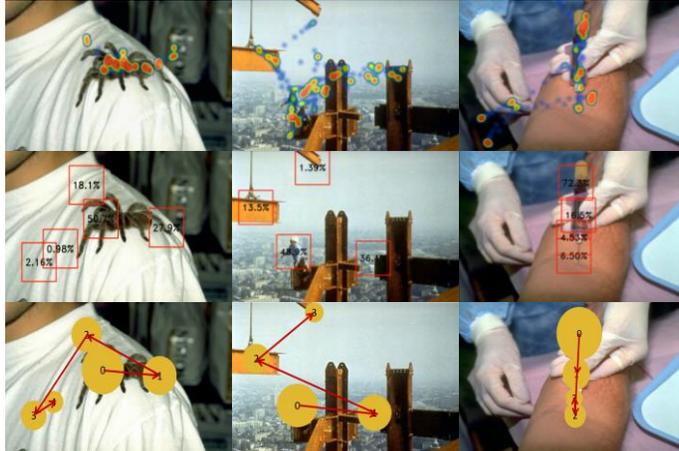


Promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio):



La gráfica muestra que, en la dimensión de la valencia, los estímulos fueron tomados de manera positiva, mientras que en la dimensión de la activación y dominio se muestra que estos estímulos no tuvieron un gran impacto en los usuarios.

UX-E-2
Estímulo de ansiedad:
 El comportamiento de los usuarios durante este tipo de estímulo solo dos de los cuatro usuarios mostró signos de nerviosismo o ansiedad (movimientos repetidos de los dedos de las manos y movimientos en las piernas).

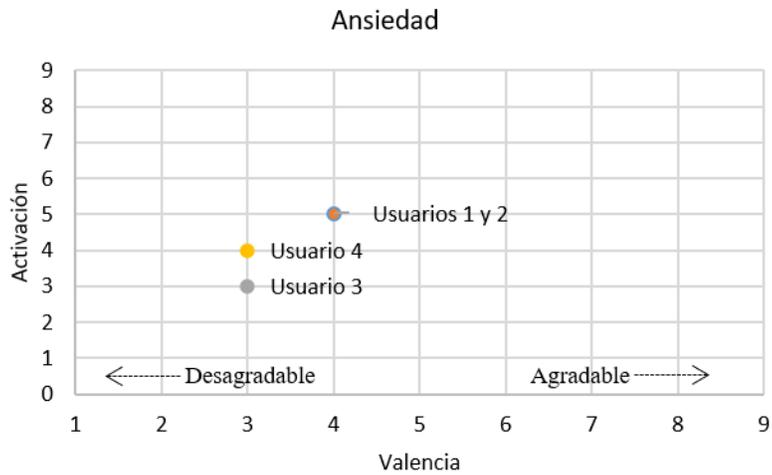


Los usuarios reportaron una emoción negativa, más alta del valor neutro en la escala de la auto evaluación SAM, la siguiente tabla muestra los valores del instrumento SAM para los participantes:

Evidencia

Ansiedad			
Usuario	Valencia	Activación	Dominio
1	4	5	4
2	4	5	5
3	3	3	3
4	3	4	3

Dispersión de datos de valencia y activación por usuario:



Promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio):



La gráfica muestra que, en la dimensión de la valencia, los estímulos fueron tomados de manera negativa, al mismo tiempo, en la dimensión de la activación se muestra que estos estímulos no tuvieron un gran impacto en los usuarios sin embargo el dominio de los usuarios sobre sus emociones en esta prueba se redujo.

Estímulo de tristeza:

UX-E-3

El comportamiento de los usuarios durante este tipo de estímulo se ha notado expresiones faciales “tristes” durante la observación de los estímulos y falta de movimiento corporal.

Evidencia



Los usuarios reportaron una emoción negativa un poco menor a la media de la escala SAM (menor que 6), la siguiente tabla muestra los valores del instrumento SAM para los participantes:

Tristeza			
Usuario	Valencia	Activación	Dominio

1	5	4	5
2	3	7	3
3	5	1	5
4	5	2	8

Dispersión de datos de valencia y activación por usuario:



Promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio):



La gráfica muestra que, en la dimensión de la valencia, los estímulos fueron tomados de manera negativa, mientras que en la dimensión de la activación y dominio se muestra que estos estímulos no tuvieron un gran impacto en los usuarios.

UX-E-4 Estímulo neutro:

En el comportamiento de los usuarios durante este tipo de se han notado algunos bostezos y parpadeos.

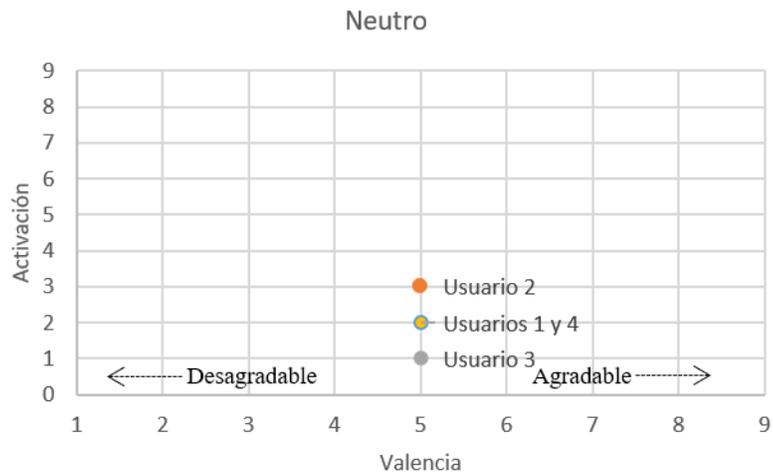


Los usuarios reportaron el estímulo con un valor neutro, media en la escala de la auto evaluación SAM, la siguiente tabla muestra los valores del instrumento SAM para los participantes:

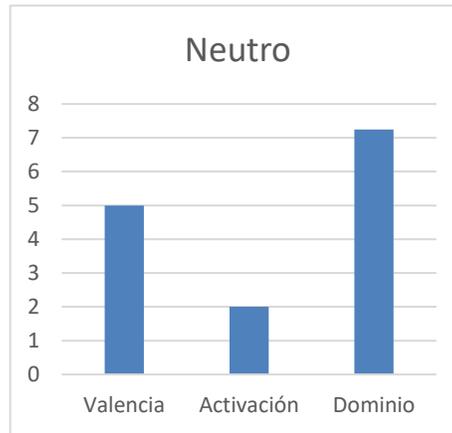
Evidencia

Neutro			
Usuario	Valencia	Activación	Dominio
1	5	2	8
2	5	3	9
3	5	1	5
4	5	2	7

Dispersión de datos de valencia y activación por usuario:



Promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio):



La gráfica muestra que, en la dimensión de la valencia, los estímulos fueron en el valor medio de la escala, en la dimensión de la activación se muestra que los estímulos no tuvieron casi ningún impacto en los usuarios y en la dimensión del dominio se muestra que los usuarios lograron dominar sus sentimientos ante los estímulos.

UX-E-6

Estímulo Mixto:

Se crearon 4 estímulos mixtos para este experimento: tristeza/neutro, alegría/ansiedad, ansiedad/tristeza, alegría/neutro. Para estas pruebas se analizaron los patrones oculares de los usuarios.

Evidencia

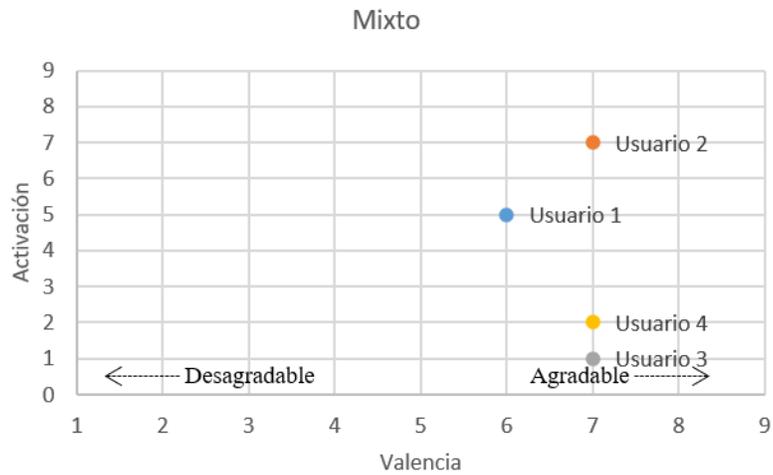


Los usuarios reportaron una emoción positiva, más alta del valor neutro en la escala de la auto evaluación SAM, la siguiente tabla muestra los valores del instrumento SAM para los participantes:

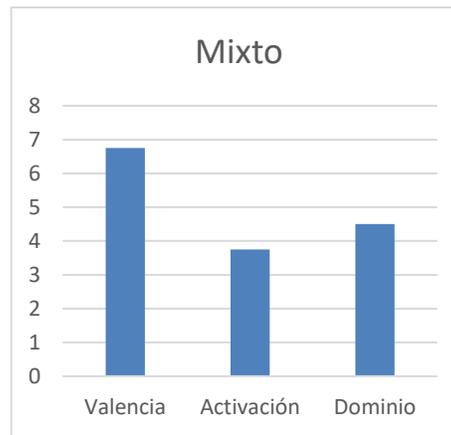
Mixto			
Usuario	Valencia	Activación	Dominio

1	6	5	5
2	7	7	4
3	7	1	1
4	7	2	8

Dispersión de datos de valencia y activación por usuario:

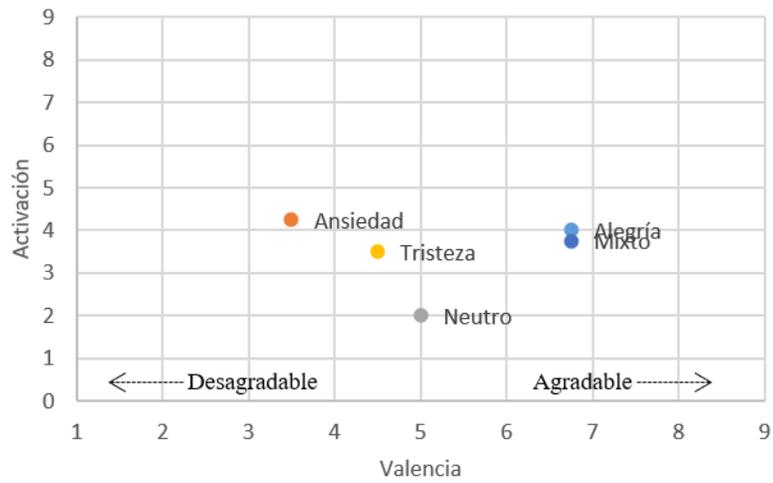


Promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio):

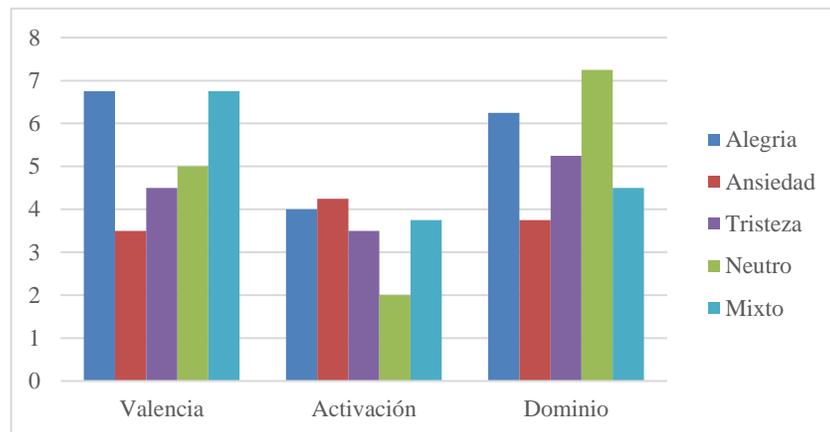


La gráfica muestra que, en la dimensión de la valencia, los estímulos fueron tomados de manera positiva con un valor promedio de 6.75 en la escala SAM, se sospecha que esto se debe a que los últimos estímulos muestran la emoción alegría. Mientras que en la dimensión de la activación y dominio se muestra que estos estímulos no tuvieron un gran impacto en los usuarios.

Dispersión de datos de valencia y activación por emoción:



La siguiente gráfica muestra el promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio) en todos los estímulos emocionales probados:



En esta prueba se analizaron, por medio de la observación de las métricas oculares en el reproductor UXLab, los patrones oculares entre los estímulos que componen las imágenes mixtas, se reporta el primer estímulo que vieron los usuarios en cada uno de los estímulos mixtos:

Usuario	Imagen mixta 1	Imagen mixta 2	Imagen mixta 3	Imagen mixta 4
	Tristeza – Neutro	Alegría – Ansiedad	Ansiedad – Tristeza	Alegría - Neutro
1	Tristeza	Alegría	Tristeza	Alegría
2	Tristeza	Ansiedad	Tristeza	Neutro
3	Neutro	Alegría	Ansiedad	Alegría
4	Tristeza	Alegría	Tristeza	Neutro

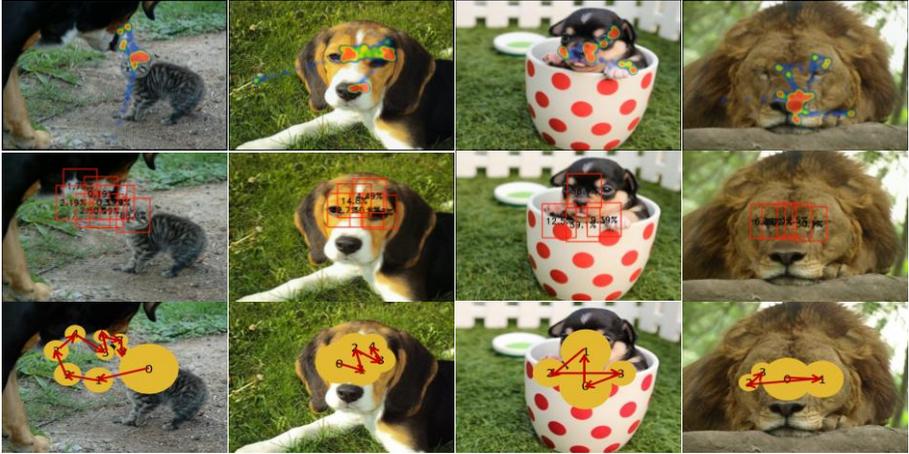
Y el tiempo que pasó la visión de cada usuario en los diferentes sectores de los estímulos mixtos:

Usuario	Imagen mixta 1		Imagen mixta 2		Imagen mixta 3		Imagen mixta 4	
	Tristeza	Neutro	Alegría	Ansiedad	Ansiedad	Tristeza	Alegría	Neutro
1	9.14	15.95	9.63	15.36	14.00	10.99	15.87	9.12
2	12.26	12.73	15.70	9.29	12.34	12.68	9.54	15.45
3	9.25	15.5	8.46	16.61	10.41	14.60	9.87	14.33
4	6.55	18.49	17.73	7.26	11.32	13.67	3.86	21.13

5.2.2.2 Segunda iteración

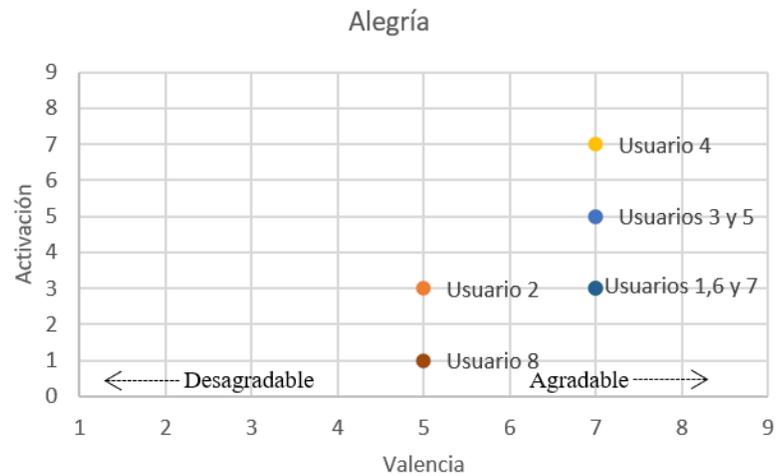
En la segunda iteración de las pruebas se reportan los resultados de ocho usuarios considerados en el plan de pruebas. Los siguientes resultados de comportamiento visual se encuentran agrupados por los diferentes estímulos emocionales en la tabla 16, en esta iteración se utilizó el repositorio de estímulos OASIS (Kurdi, Lozano, y Banaji, 2016).

Tabla 16: Reporte de segunda iteración de pruebas emocionales UXLab.

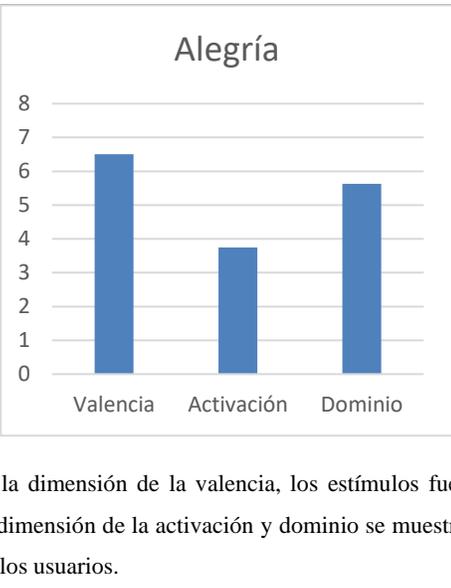
<p>UX-E-1</p>	<p>Estímulo de felicidad: Puede observarse que la atención principal se enfoca en los animales, este fenómeno es equivalente al ya antes reportado donde las personas centran su atención en el rostro de las personas primero antes de observar pequeños detalles del entorno.</p>
<p>Evidencia</p>	 <p>Los usuarios reportaron una emoción positiva, más alta del valor neutro en la escala de la auto evaluación SAM, la siguiente tabla muestra los valores del instrumento SAM para los participantes:</p>

Alegría			
Usuario	Valencia	Activación	Dominio
1	7	3	5
2	5	3	5
3	7	5	5
4	7	7	5
5	7	5	5
6	7	3	7
7	7	3	5
8	5	1	8

Dispersión de datos de valencia y activación por usuario:

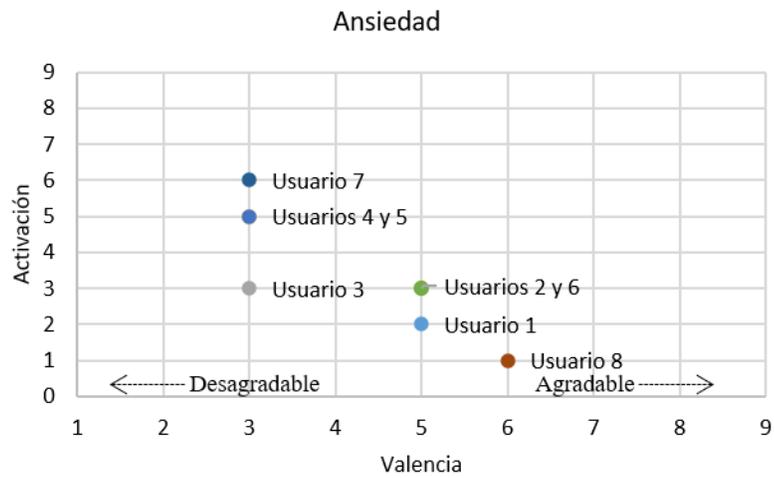


Promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio):

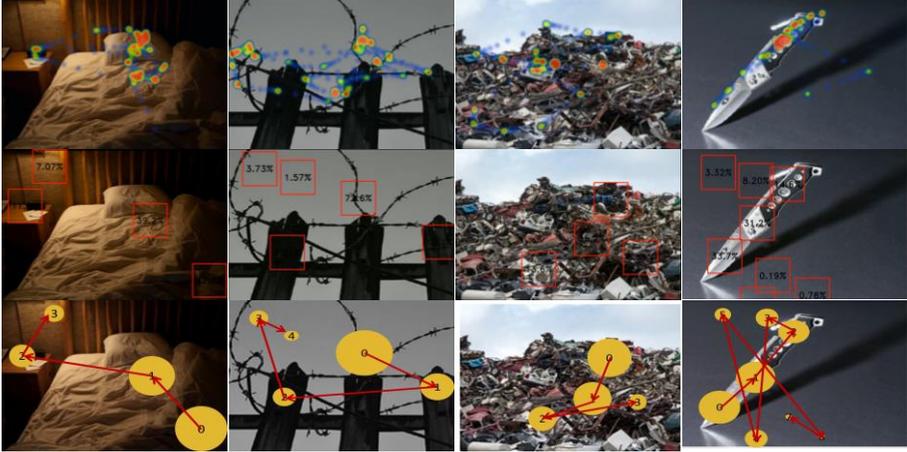
		 <p style="text-align: center;">Alegría</p> <table border="1"> <thead> <tr> <th>Categoría</th> <th>Puntuación</th> </tr> </thead> <tbody> <tr> <td>Valencia</td> <td>~6.5</td> </tr> <tr> <td>Activación</td> <td>~3.8</td> </tr> <tr> <td>Dominio</td> <td>~5.5</td> </tr> </tbody> </table>	Categoría	Puntuación	Valencia	~6.5	Activación	~3.8	Dominio	~5.5	
Categoría	Puntuación										
Valencia	~6.5										
Activación	~3.8										
Dominio	~5.5										
<p>UX-E-2</p>	<p>Estímulo de ansiedad: En el estímulo de ansiedad, los usuarios reportaron sentirse incomodos observando el estímulo:</p>										
<p>Evidencia</p>	 <p>Los usuarios reportaron una emoción positiva, más alta del valor neutro en la escala de la auto evaluación SAM, la siguiente tabla muestra los valores del instrumento SAM para los participantes:</p>										

Ansiedad			
Usuario	Valencia	Activación	Dominio
1	5	2	5
2	5	3	7
3	3	3	3
4	3	5	7
5	3	5	6
6	5	3	7
7	3	6	4
8	6	1	8

Dispersión de datos de valencia y activación por usuario:

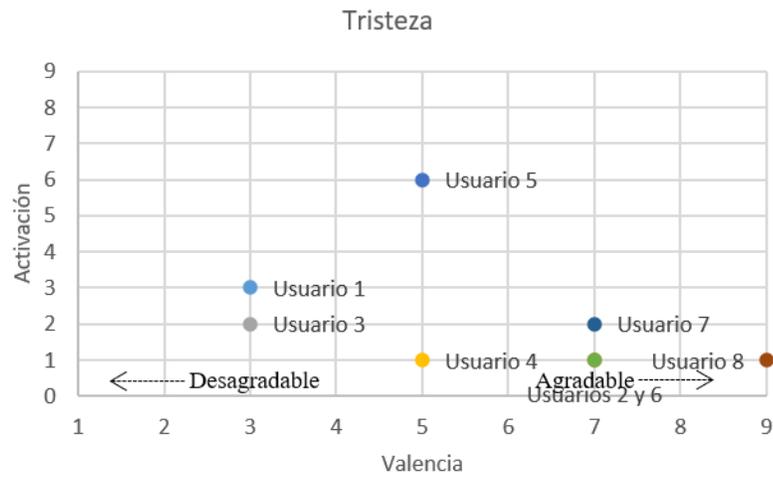


Promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio):

	 <p>La gráfica muestra que, en la dimensión de la valencia, los estímulos fueron tomados de manera negativa, mientras que en la dimensión de la activación y dominio se muestra que estos estímulos no tuvieron un gran impacto en los usuarios.</p>	
<p>UX-E-3</p>	<p>Estímulo de tristeza:</p> <p>Se puede notar una fijación visual sobre los elementos como la cama y el cuchillo, mientras que los otros estímulos presentan fijaciones visuales más esparcidas debido a la naturaleza de las imágenes (tipo paisajes).</p>	
<p>Evidencia</p>	 <p>Los usuarios reportaron una emoción positiva, más alta del valor neutro en la escala de la auto evaluación SAM, la siguiente tabla muestra los valores del instrumento SAM para los participantes:</p>	

Tristeza			
Usuario	Valencia	Activación	Dominio
1	5	2	5
2	7	1	7
3	4	3	3
4	5	1	5
5	5	3	5
6	7	5	7
7	5	3	7
8	9	1	9

Dispersión de datos de valencia y activación por usuario:



Promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio):

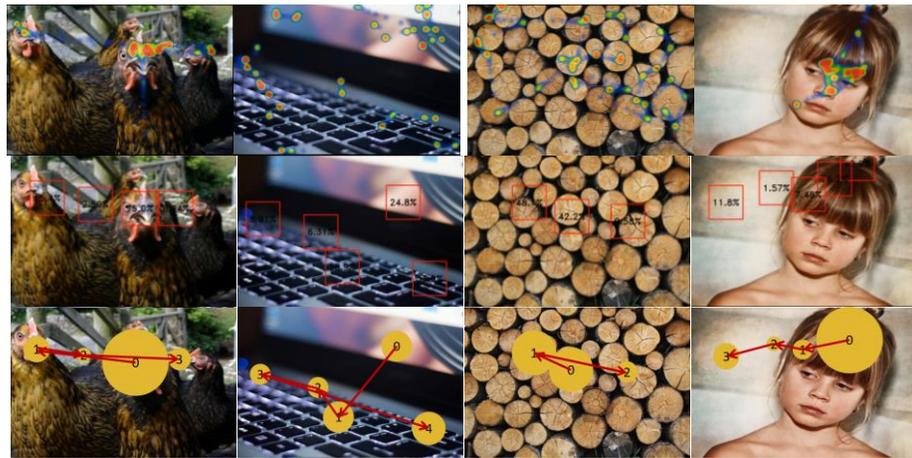


La gráfica muestra que, en la dimensión de la valencia, los estímulos fueron tomados de manera un poco positiva (mayor a 5 pero menor a 6), lo cual indica que la selección de estímulos para la emoción tristeza debe ser reconsiderada. En la dimensión de la activación y dominio se muestra que estos estímulos no tuvieron un gran impacto en los usuarios.

UX-E-4

Estímulo neutro:

En los estímulos neutros las fijaciones visuales de los usuarios se concentran en los rostros de la niña y las gallinas mientras que en los estímulos de la computadora y los troncos las fijaciones se encuentran esparcidas por la imagen ya que no existe algo que capte la atención de los usuarios como se muestra en los mapas de atención.

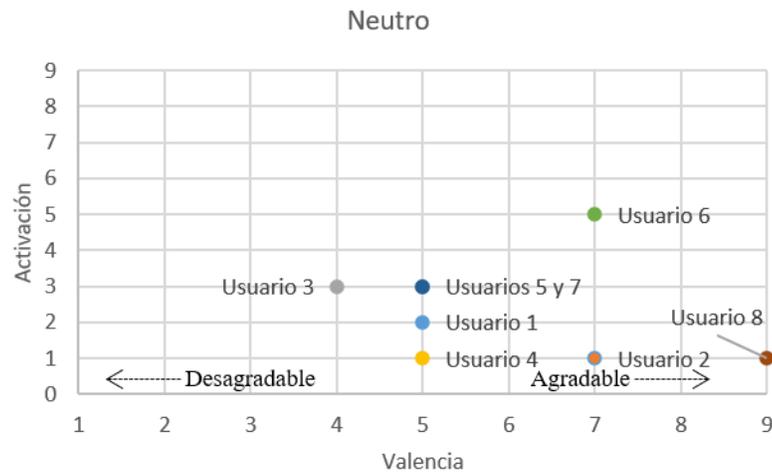


Evidencia

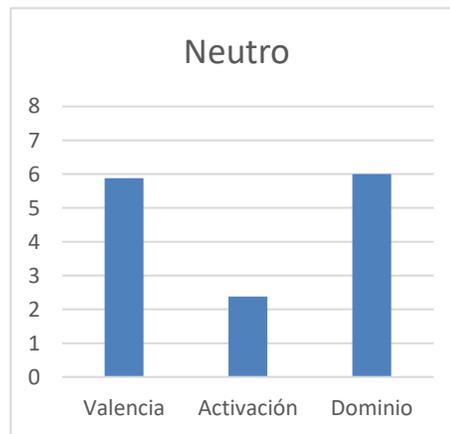
Los usuarios reportaron una emoción positiva, más alta del valor neutro en la escala de la auto evaluación SAM, la siguiente tabla muestra los valores del instrumento SAM para los participantes:

Neutro			
Usuario	Valencia	Activación	Dominio
1	5	2	5
2	7	1	7
3	4	3	3
4	5	1	5
5	5	3	5
6	7	5	7
7	5	3	7
8	9	1	9

Dispersión de datos de valencia y activación por usuario:



Promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio):



La gráfica muestra que, en la dimensión de la valencia, los estímulos fueron tomados de manera un poco positiva (mayor a 5 pero menor a 6), mientras que en la dimensión de la activación y dominio se muestra que estos estímulos no tuvieron un gran impacto en los usuarios.

UX-E-5

Estímulo de relajación:

Nuevamente se puede notar el comportamiento de enfocar la mirada en los rostros de las personas antes de comenzar a notar pequeños detalles como la ropa y las herramientas que usa la doctora en el primer estímulo.

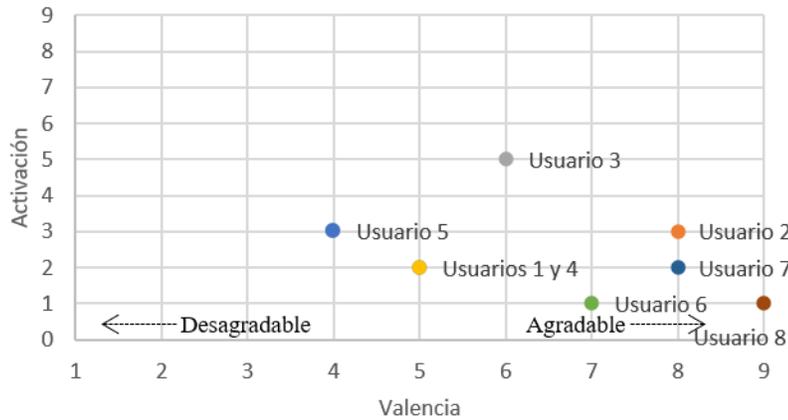


Evidencia

Los usuarios reportaron una emoción positiva, más alta del valor neutro en la escala de la auto evaluación SAM, la siguiente tabla muestra los valores del instrumento SAM para los participantes:

Relajación			
Usuario	Valencia	Activación	Dominio
1	5	2	5
2	8	3	7
3	6	5	4
4	5	2	5
5	4	3	4
6	7	1	7
7	8	2	7
8	9	1	9

Dispersión de datos de valencia y activación por usuario:

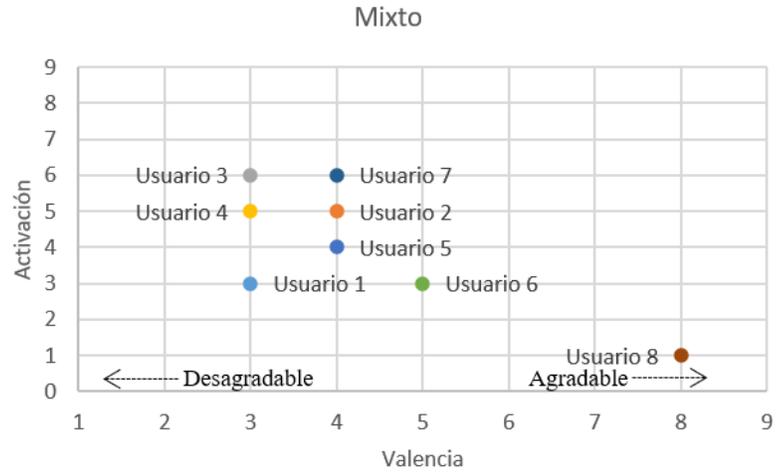
	<p style="text-align: center;">Relajación</p>  <p>Promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio):</p>  <p>La gráfica muestra que, en la dimensión de la valencia, los estímulos fueron tomados de manera positiva, mientras que en la dimensión de la activación y dominio se muestra que estos estímulos no tuvieron un gran impacto en los usuarios.</p>
<p>UX-E-6</p>	<p>Estímulo Mixto: En el estímulo mixto se utilizó una combinación de dos imágenes con diferentes cargas emocionales para observar el comportamiento ocular del usuario. Además de observar el comportamiento típico de ver los rostros de las personas, se puede notar (observando las áreas de interés) que la visión de las personas se concentró en estímulos positivos la mayor parte del tiempo, esto apoya lo ya antes mencionado en la primera iteración.</p>
<p>Evidencia</p>	



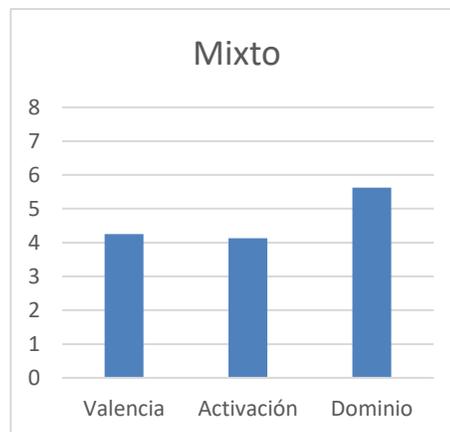
Los usuarios reportaron una emoción positiva, más alta del valor neutro en la escala de la auto evaluación SAM, la siguiente tabla muestra los valores del instrumento SAM para los participantes:

Mixto			
Usuario	Valencia	Activación	Dominio
1	3	3	5
2	4	5	5
3	3	6	5
4	3	5	6
5	4	4	5
6	5	3	7
7	4	6	4
8	8	1	8

Dispersión de datos de valencia y activación por usuario:



Promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio):

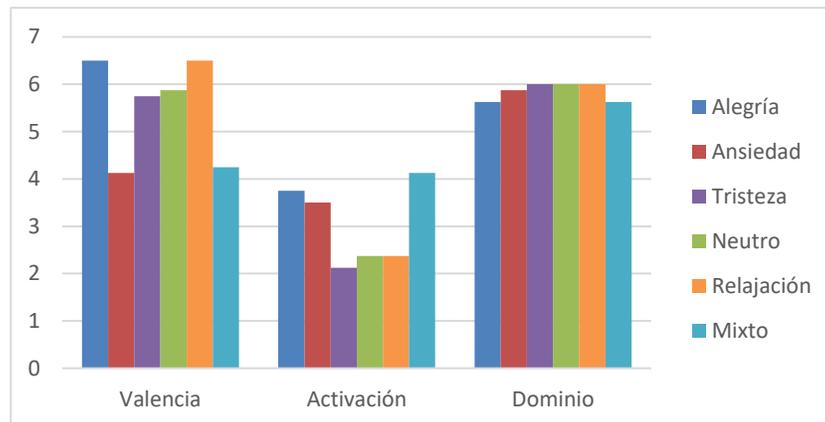


La gráfica muestra que, en la dimensión de la valencia, los estímulos fueron tomados de manera negativa, mientras que, en la dimensión de la activación, esta serie de estímulos mixto obtuvieron la calificación de activación más alta. En el dominio se muestra que estos estímulos no tuvieron un gran impacto en los usuarios.

Dispersión de datos de valencia y activación por emoción:



Promedio para las 3 dimensiones de la escala SAM (Valencia, Activación y dominio) en todos los estímulos emocionales:



En la escala de valores SAM la media es 5 para las 3 dimensiones. Puede observarse que los estímulos positivos, tienen una valencia superior a 5 mientras que los negativos tienen un valor menor a 5. En cuanto a la activación los estímulos caen en valores menores a 5 por lo que el impacto emocional no fue demasiado, esto también se ve reflejado en el dominio con valores de dominio mayores a 5.

En esta prueba se analizaron, por medio de la observación de las métricas oculares en el reproductor UXLab, los patrones oculares entre los estímulos que componen las imágenes mixtas, se reporta el primer estímulo que vieron los usuarios en cada uno de los estímulos mixtos:

Usuario	Alegría Ansiedad	Alegría Relajación	Alegría Tristeza	Ansiedad Tristeza	Relajación Ansiedad	Relajación Tristeza
1	Ansiedad	Alegría	Tristeza	Tristeza	Ansiedad	Tristeza
2	Alegría	Alegría	Tristeza	Ansiedad	Relajación	Tristeza
3	Alegría	Alegría	Alegría	Ansiedad	Ansiedad	Relajación
4	Alegría	Relajación	Alegría	Tristeza	Relajación	Relajación
5	Ansiedad	Alegría	Alegría	Ansiedad	Ansiedad	Tristeza
6	Ansiedad	Alegría	Alegría	Tristeza	Ansiedad	Tristeza
7	Ansiedad	Alegría	Tristeza	Tristeza	Relajación	Tristeza
8	Alegría	Relajación	Alegría	Tristeza	Relajación	Relajación
Tend.	Ale / Ans	Alegría	Alegría	Tristeza	Ans/Rela	Tristeza

Además, se realizó un análisis de los tipos de estímulos con mayor concentración de fijaciones, por sexo.

Sexo	Alegría Ansiedad	Alegría Relajación	Alegría Tristeza	Ansiedad Tristeza	Relajación Ansiedad	Relajación Tristeza
Mujer	Ansiedad	Alegría	Alegría	Tristeza	Ansiedad	Tristeza
Hombre	Alegría	Alegría	Tris / Ale	Ans / Tris	Relajación	Tris / Rela

En lo anterior se puede observar una tendencia por observar estímulos más positivos en hombres, y lo contrario para las mujeres quienes tienden a observar más los estímulos negativos.

5.2.2.1 Conclusiones de las pruebas emocionales

Al terminar las pruebas emocionales con éxito, se puede concluir que la generación de métricas oculares fue exitosa y facilitó en gran medida el análisis del comportamiento humano durante las pruebas. Los mapas de calor y áreas de interés proporcionan una idea clara de qué partes de las imágenes llamaban más la atención de los usuarios. Los mapas de rutas fueron utilizados para observar la ruta que seguían los ojos de los usuarios, sin embargo, no se encontró un patrón específico ya que cada usuario seguía una ruta diferente.

Se pueden hacer varias conclusiones sobre el comportamiento de los sujetos de prueba mientras estaban expuestos a estímulos de imágenes:

- Los usuarios comienzan por observar los rostros de las personas antes que cualquier objeto en los estímulos.

- Los usuarios prefieren observar estímulos neutros o de alegría antes de estímulos con temas de tristeza o ansiedad.
- Las autoevaluaciones SAM de los usuarios corresponden con el tipo de emoción que los estímulos IAPS deberían hacer sentir a los usuarios (por ejemplo, estímulos de alegría dieron como resultado un valor de casi 7 en la escala SAM)
- Algunos estímulos, sobre todo los de ansiedad y tristeza, afectan los movimientos corporales y expresiones faciales de los usuarios expuestos.

5.2.3 Pruebas cognitivas

En esta sección se presentan los resultados de las pruebas cognitivas, en las que participaron 10 usuarios con un rango de edad de entre 24 y 30 años, todos estudiantes de postgrado en ciencia de la computación. Las pruebas fueron realizadas en el laboratorio de Sistemas Híbridos Inteligentes, siguiendo las recomendaciones básicas de ambiente controlado de pruebas con seguimiento ocular, esto incluye ambiente silencioso, sin corrientes de aire, iluminación adecuada, posición del monitor y del usuario, entre otras consideraciones que aseguran que los datos se graben de la mejor manera posible.

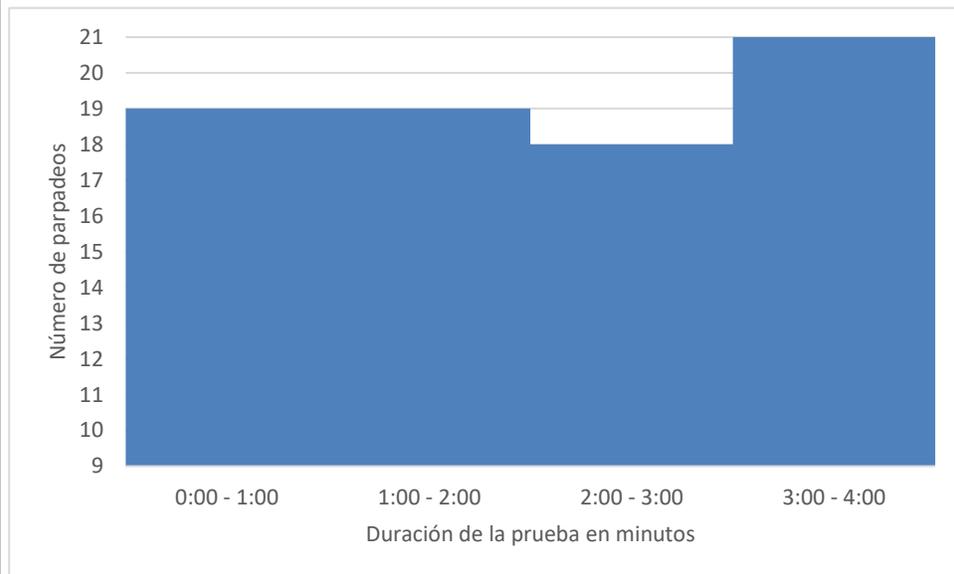
La tabla 17 muestra los resultados de la ejecución del plan de pruebas cognitivas.

Tabla 17: Reporte de pruebas cognitivas UXLab.

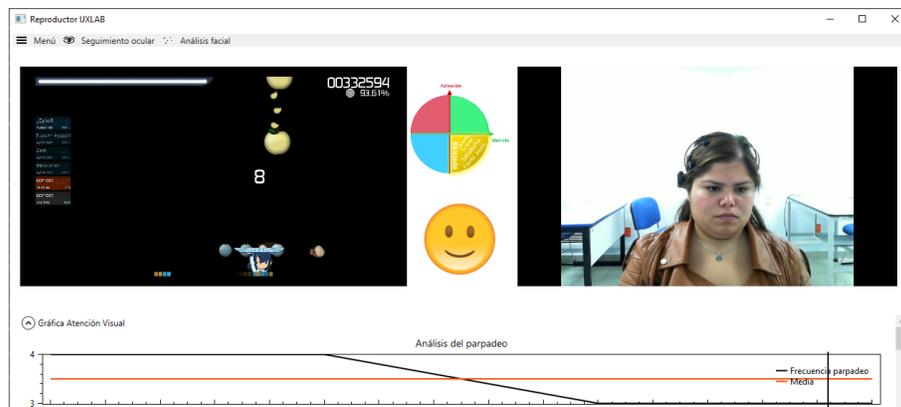
<p>UX-C-1</p>	<p>Estímulo de baja carga cognitiva:</p> <p>Los resultados preliminares de la prueba muestran una frecuencia de parpadeo media bastante reducida, si se toma en cuenta que cada experiencia con un nivel del videojuego es en promedio de 3 minutos, entonces se puede observar que los usuarios parpadearon máximo una vez cada minuto. La frecuencia normal de una persona viendo algún estímulo en pantalla es de 15 a 20 parpadeos por minuto (Nakano, Kato, Morito, Itoi, y Kitazawa, 2013).</p>												
<p>Evidencia</p>	<p>Datos capturados durante las pruebas para cada usuario:</p> <table border="1" data-bbox="623 1686 1219 1900"> <thead> <tr> <th>#</th> <th>Frecuencia de parpadeo promedio</th> <th>Cantidad de parpadeos</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0.25</td> <td>1</td> </tr> <tr> <td>2</td> <td>3.5</td> <td>12</td> </tr> <tr> <td>3</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	#	Frecuencia de parpadeo promedio	Cantidad de parpadeos	1	0.25	1	2	3.5	12	3	1	1
#	Frecuencia de parpadeo promedio	Cantidad de parpadeos											
1	0.25	1											
2	3.5	12											
3	1	1											

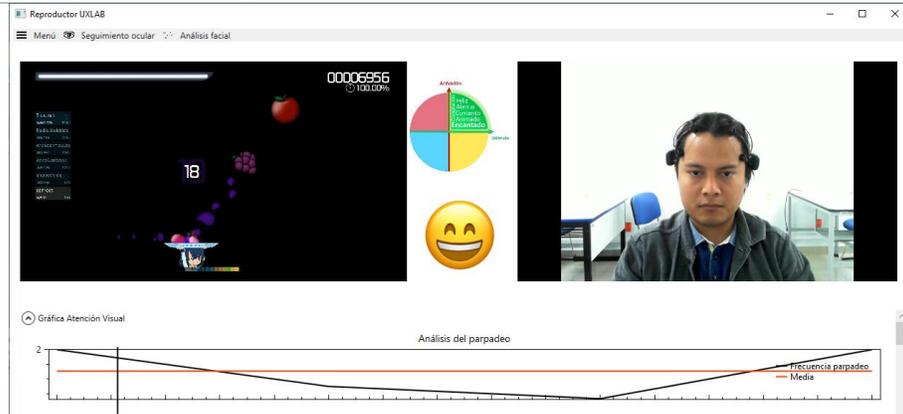
4	1.70	8
5	2.5	9
6	3.75	15
7	2.5	6
8	2.333	7
9	1.5	6
10	3	9

La siguiente gráfica muestra la sumatoria de los parpadeos capturados para los 10 usuarios en función de la duración de la prueba:



Visualización de las pruebas para su análisis con UXLab:





Desempeño de los usuarios mostrados anteriormente:



UX-C-2

Estímulo de alta carga cognitiva:

El estímulo de alta carga cognitiva se caracteriza por tener una complejidad mayor (el nivel del videojuego se vuelve más complicado al aumentar el número de frutas a atrapar y la velocidad con las que estas caen). En este caso los usuarios reportaron un incremento significativo de la dificultad y de la velocidad de los niveles del juego.

A diferencia de la prueba anterior los usuarios tuvieron más problemas para completar el nivel del juego, esto se debe al aumento de la complejidad y de la carga de trabajo. Se puede observar cómo disminuye el desempeño de los usuarios en esta dificultad.

En cuanto a la cantidad y frecuencia de parpadeos no se nota una diferencia significativa entre la prueba anterior y esta, algunos usuarios aumentaron su número de parpadeos y en otros casos ocurre lo contrario.

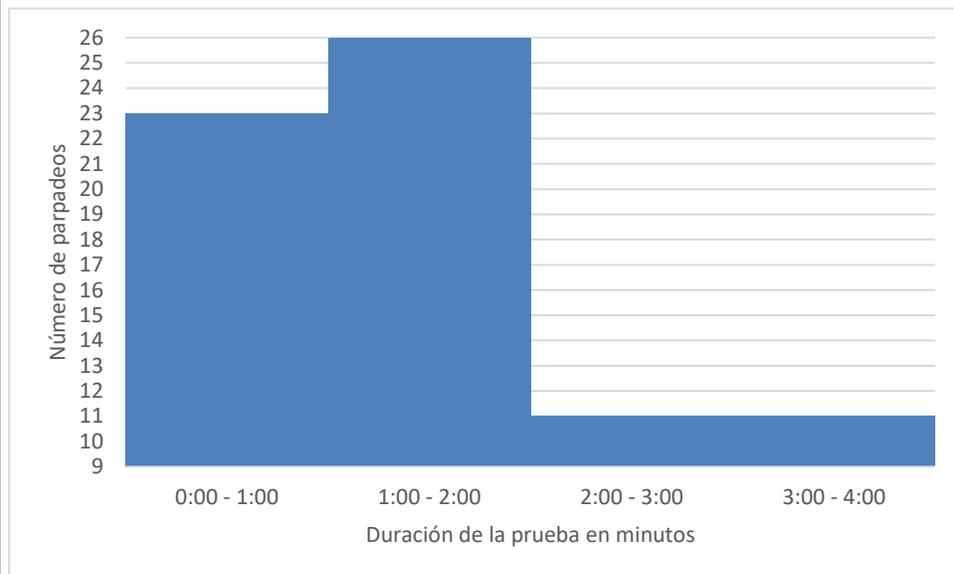
Evidencia

Datos capturados durante las pruebas para cada usuario:

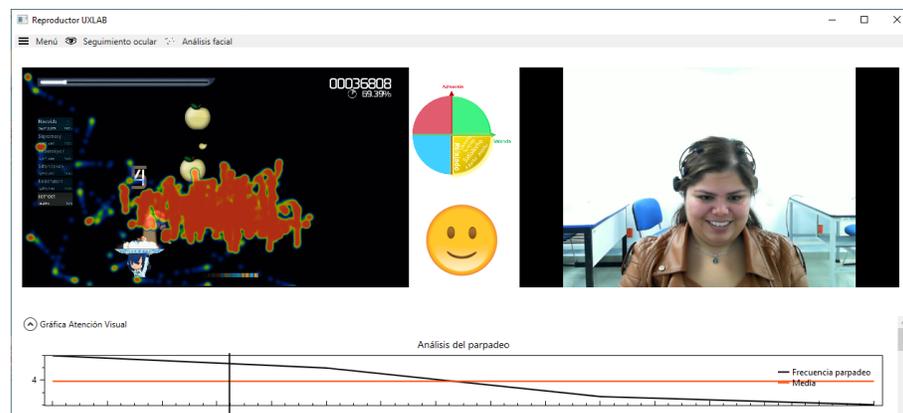
#	Frecuencia de parpadeo promedio	Cantidad de parpadeos
1	2.75	2
2	3.95	12
3	3	3
4	1.60	7
5	1.66	5
6	4	16

7	1.25	5
8	3.333	10
9	0.75	3
10	2.66	8

La siguiente gráfica muestra la sumatoria de los parpadeos capturados para los 10 usuarios en función de la duración de la prueba:



Visualización de las pruebas para su análisis con UXLab:





Desempeño de los usuarios mostrados anteriormente:



UX-C-3

Estímulo de baja carga cognitiva a alta carga cognitiva:

Se tomaron las pruebas UX-C-1 y UX-C-2 para comparar los resultados de ambas y observar diferencias entre el comportamiento cognitivo y ocular de los usuarios en distintos niveles de carga cognitiva (bajo y alto).

En todas las pruebas, la diferencia entre los parpadeos capturados es mínima y no es significativa. Al calcular la frecuencia de parpadeo por minuto, se obtienen valores menores a 4, 4 parpadeos por minuto es una cifra menor al valor normal de parpadeos por minuto establecidos en la literatura:

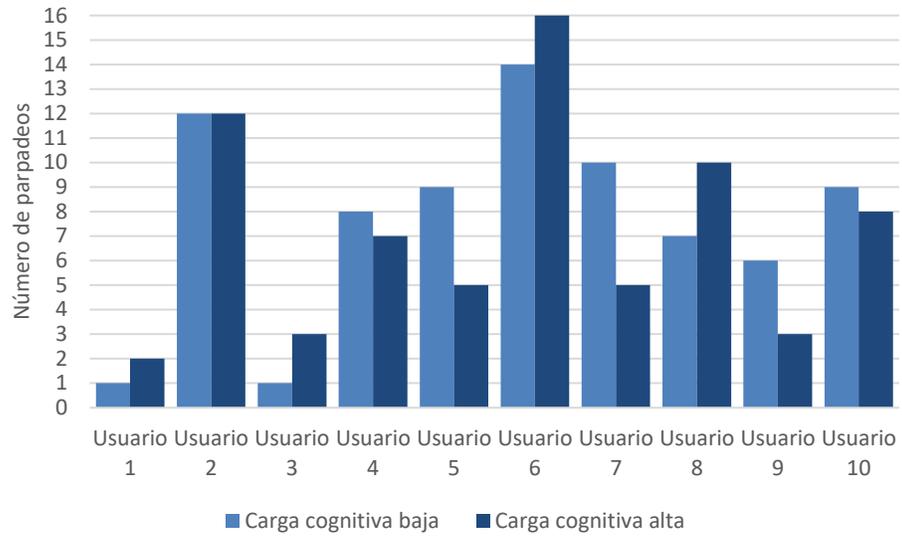
- 15 a 20 parpadeos por minuto (Nakano et al., 2013).
- En promedio 10 parpadeos por minuto en ambiente de laboratorio (Bentivoglio et al., 1997)

Según Bentivoglio et al., La frecuencia del parpadeo puede bajar hasta un promedio de 3 o 4 veces por minuto cuando una persona se concentra en un objeto, como por ejemplo al leer. Este último número de frecuencia se asemeja más con los resultados obtenidos en las pruebas cognitivas.

Sin embargo, con estos los datos de las pruebas no se puede inferir que la frecuencia del parpadeo se reduce mientras la tarea de una persona aumente su carga cognitiva o complejidad.

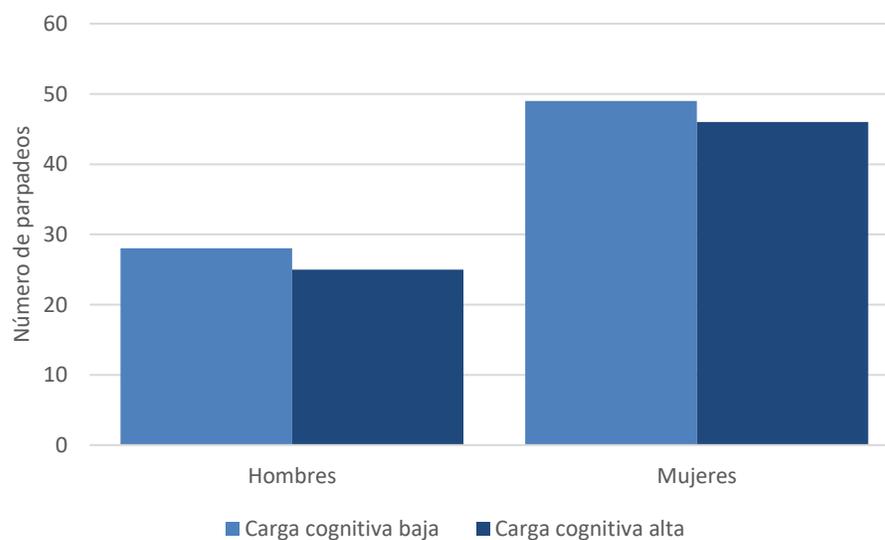
Evidencia

Comparación de datos capturados durante las dos pruebas para cada usuario:



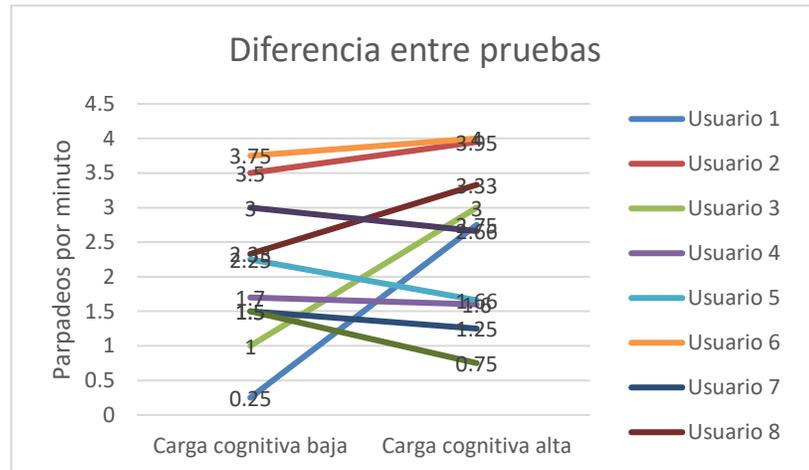
Diferencia entre la captura de parpadeos entre sexo de los usuarios y niveles de prueba:

	Carga cognitiva baja		Carga cognitiva alta	
	Mujeres	Hombres	Mujeres	Hombres
Número de parpadeos	12	1	12	2
	15	9	16	5
	6	1	5	3
	7	8	10	7
Promedio	9.2	5.6	9.2	5



Se puede observar en la gráfica anterior la diferencia entre los parpadeos de hombres y mujeres, donde las mujeres registraron muchos más parpadeos que su contraparte.

Hay 5 casos donde la frecuencia de parpadeo se ve reducida (usuario 4, 5, 7, 9 y 10), quienes bajan su frecuencia del nivel bajo al nivel alto de complejidad en las pruebas:



Se puede observar entonces como las dos tareas no contrastan demasiado en cuanto a concentración o complejidad. Este fenómeno se discutirá en las conclusiones de las pruebas.

5.2.3.1 Conclusiones de las pruebas cognitivas

Luego de realizar un estudio de los datos obtenidos en el experimento para comprobar que se trataba de una distribución normal de datos, se aplicó una prueba de Shapiro Wilk (Shapiro y Wilk, 1965), que permite determinar si un conjunto de datos tiene una distribución normal, sobre los 4 grupos de datos obtenidos del experimento, se obtuvo un valor $p > 0.05$ para cada grupo por lo que no existe suficiente evidencia para concluir que la distribución de datos de cada grupo no es normal. Al saber que los datos siguen un patrón normal, se pueden realizar análisis paramétricos para comprobación de hipótesis como el análisis de la varianza o la prueba t , que permiten realizar una comparación entre varios grupos de datos de tal manera que se comprueba si existe una diferencia estadística significativa entre los conjuntos de datos.

Con el análisis paramétrico se trata de rechazar la hipótesis nula (H_0), la cual establece que los grupos de datos analizados son diferentes entre ellos, por la hipótesis alternativa (H_1) que establece que existe una diferencia significativa entre los dos grupos de datos.

Para el caso de las pruebas realizadas se puede definir la hipótesis nula, en términos coloquiales, como H_0 : “El comportamiento de parpadeos de los usuarios no presenta cambios cuando juegan un nivel de videojuego más difícil”, mientras que la hipótesis alternativa se define como H_1 : “Los usuarios cambian su comportamiento de parpadeos cuando juegan un nivel de videojuego más difícil”

Para este análisis de los datos se optó por utilizar una prueba t para medias de dos muestras emparejadas, teniendo como primer conjunto de datos el número de parpadeos capturados en la prueba UX-C-1 y como segundo conjunto aquellos capturados en la prueba UX-C-2, dio como resultado, $t(9) = 0.43$, $p = 0.67$ (ver tabla 18).

Tabla 18: Resultados de prueba t para el número de parpadeos capturados en los niveles de carga cognitiva baja y alta.

	<i>Baja</i>	<i>Alta</i>
Media	7.4	7.1
Varianza	18.93333333	20.1
Observaciones	10	10
Coefficiente de correlación de Pearson	0.880552817	
Grados de libertad	9	
Estadístico t	0.438633161	
$P(T \leq t)$ dos colas	0.671268049	
Valor crítico de t (dos colas)	2.262157163	

Como el estadístico t no es mayor al valor crítico de t , no hay suficiente evidencia de que las dos pruebas sean significativamente diferentes. Esto quiere decir que la dificultad de los niveles del videojuego para la prueba UX-C-1 (nivel bajo) y la prueba UX-C-2 (nivel alto) no arrojan resultados significativamente diferentes y que la cantidad de parpadeos capturados en las dos pruebas son similares.

Así mismo, el valor p de la prueba es demasiado alto ($p > 0.05$) esto significa que, si se rechaza H_0 , la posibilidad de error tipo I es de 67.12% por lo tanto los datos no son estadísticamente significativos y no se rechaza la hipótesis nula, no hay suficiente evidencia de que las dos pruebas sean significativamente diferentes.

En el segundo análisis con prueba t para medias de dos muestras emparejadas, se utilizó como conjuntos de datos la frecuencia de parpadeo de los usuarios en la prueba UX-C-1 y la prueba UX-C-2, dio como resultado, $t(9) = 0.75$, $p = 0.47$ (Ver tabla 19).

Tabla 19: Resultados de prueba t para la frecuencia de parpadeo de los usuarios en los niveles bajo y alto.

	<i>Baja</i>	<i>Alta</i>
Media	2.2033	2.4953
Varianza	1.2125089	1.2805409
Observaciones	10	10
Coefficiente de correlación de Pearson	0.393189704	
Grados de libertad	9	
Estadístico t	0.750651768	
$P(T \leq t)$ dos colas	0.472030988	
Valor crítico de t (dos colas)	2.262157163	

Al igual que el caso anterior el estadístico t es menor al valor crítico de t , esto quiere decir que no existe suficiente evidencia de que las dos pruebas sean significativamente diferentes. Por lo que la frecuencia de los parpadeos de los usuarios en la prueba UX-C-1 (nivel bajo) y la prueba UX-C-2 (nivel alto) tampoco son significativamente diferentes.

Al mismo tiempo, el valor p de la prueba es demasiado alto ($p > 0.05$) esto significa que, si se rechaza H_0 , la posibilidad de error tipo I es de 47.20% por lo tanto los datos no son estadísticamente significativos y no se rechaza la hipótesis nula (no hay suficiente evidencia de que las dos pruebas sean significativamente diferentes).

Con este análisis de los datos se puede concluir que las dos pruebas no son demasiado diferentes entre ellas, esto apunta a que no existe mucha diferencia en el número de parpadeos capturados y las frecuencias de parpadeos durante las dos tareas de jugar un nivel de dificultad fácil y uno difícil.

Sin embargo, se puede contrastar la frecuencia del parpadeo en las pruebas con lo reportado en la literatura (Ver tabla 20).

Tabla 20: Contraste de frecuencia de parpadeos en este trabajo y otros reportados en la literatura.

	<i>Media de frecuencia parpadeo</i>
Este trabajo	2.34
Nakano et al., 2013	15
Bentivoglio et al., 1997	10

Esto denota efectivamente que la frecuencia de parpadeo se ve reducida ante actividades que requieren el uso de un proceso de concentración como es el caso de jugar un videojuego.

De las posibles razones en las que existe tan poca variación entre los dos niveles de dificultad de las pruebas puede deberse a los intentos de minimizar la pérdida de información de los jugadores en los niveles más difíciles, Fogarty y Stern (1989) explican que existe una relación entre la tasa de parpadeo y las demandas de las tareas. Por lo tanto, puede que el aumento de la carga cognitiva en tareas (niveles más difíciles) en realidad aumenten el número de parpadeos en vez de reducirlo como se teorizaba al inicio del trabajo.

Los parpadeos no ocurren al azar (Stern, Walrath y Goldstein, 1984), sino que los estallidos de parpadeo siguen a una carga cognitiva alta (Fukuda, 1994, 2001; Ohira, 1996) o procesamiento de la información (Ichikawa y Ohira, 2004), citando a Ohira, Winton y Oyama (1998) esto sugiere que al aumentar la dificultad de una tarea las personas necesitan parpadear más para poder liberar los recursos utilizados en la cognición relaciona al estímulo (en este caso el video juego).

Utilizando los datos obtenidos y analizando lo establecido en la literatura y experimentos realizados durante la investigación, se concluye que, si existe una correlación entre la frecuencia del parpadeo de las personas y la carga cognitiva que enfrentan, teniendo que mientras aumenta la carga cognitiva, la frecuencia de parpadeos se reduce y mientras que decrementa la carga cognitiva, aumenta la frecuencia de parpadeo.

Esta conclusión es empírica y de preferencia debería ser probada analizando los datos de un experimento que defina más de 3 diferentes niveles de carga cognitiva y se tomen al menos 20 muestras de los parpadeos de los usuarios. Utilizando el análisis de la varianza de un factor.

Capítulo 6. Conclusiones y trabajos futuros

El objetivo general establecido para el proyecto, el cual trata de “Evaluar e implementar algoritmos para analizar el movimiento ocular de un usuario durante el proceso de evaluación de la experiencia de usuario para obtener datos relevantes y generar mapas de calor y de trazo de trayectorias correlacionados con el estado cognitivo que experimenta”, se cumplió al definir las métricas del seguimiento ocular e implementarlas en la plataforma para la evaluación de la experiencia de usuario UXLab. Estos algoritmos que generan las métricas de análisis de datos oculares (Mapas de atención, Áreas de interés y Análisis de parpadeos) se correlacionan con el estado cognitivo del usuario directamente, usando el análisis del parpadeo, y con la ayuda de las funciones de visualización de datos fisiológicos de UXLab.

De igual forma los objetivos particulares definidos para la tesis fueron completados en su totalidad:

- Dentro del trabajo se compararon varias técnicas de extracción de datos de seguimiento ocular y generación de métricas.
- Los algoritmos desarrollados han sido utilizados para:
 - Extraer datos de puntos de visión, puntos de fijación y parpadeos
 - Generar mapas de atención, rutas sacádicas, áreas de interés y la frecuencia de parpadeo.
- Estas métricas fueron incorporadas al módulo de seguimiento ocular de UXLab, donde se evaluó el estado cognitivo del usuario relacionando sus datos fisiológicos con sus movimientos oculares.
- Se realizaron pruebas de los algoritmos desarrollados.

Durante la investigación se buscaron herramientas y documentación sobre la extracción o generación de métricas, sin embargo, no se logró encontrar información de acceso libre a estos temas que se consideran básicos para trabajar con seguimiento ocular. Derivado de la revisión de estado del arte, se logró la definición de varias fórmulas y algoritmos sobre seguimiento ocular, logrando obtener los conocimientos necesarios para definir la estructura de datos y las métricas generadas a partir de estas.

En el transcurso del trabajo se pudo interactuar con bastantes usuarios de los cuales se registraron datos de seguimiento ocular para realizar análisis de los algoritmos

implementados, pero también se han utilizado estas pruebas para obtener una visión general del comportamiento humano durante la interacción con productos digitales y multimedia. Como se describe en el capítulo 5 el comportamiento de los usuarios durante las pruebas emocionales y cognitivas denota ciertos patrones que se encuentran registrados en el estado del arte, por ejemplo, el hecho de observar los rostros de las personas y la tendencia de un mayor número de parpadeo en mujeres que en hombres.

Las métricas obtenidas a partir de los datos del seguimiento ocular son una forma de observar el comportamiento de la atención visual del usuario y este comportamiento puede aportar datos útiles para evaluar la experiencia de los usuarios cuando utilizan productos digitales. Las métricas proporcionan información sobre la percepción que genera en el usuario a un nivel cognitivo todo el contenido o partes aisladas del producto digital.

Al considerar estas métricas, en conjunto con los datos fisiológicos, gracias a las plataformas de evaluación UXLab, se obtienen pistas objetivas sobre el estado mental y emocional de los usuarios, la evaluación de la UX ayudará a mejorar los niveles de satisfacción y aceptación de los productos digitales ya desarrollados o en desarrollo.

6.1 Aportaciones

6.1.1 Integración a la herramienta UXLab

La inclusión del módulo de seguimiento ocular en UXLab aporta nuevas formas de observar el comportamiento del usuario durante pruebas de evaluación de la UX, la implementación de las métricas visuales ayuda a los evaluadores a observar la atención del usuario en los elementos de la pantalla. Además, como el software UXLab reporta la información de otros sensores de tipo fisiológico, se puede realizar una correlación con el estado emocional del usuario durante la observación de un cierto estímulo.

UXLab permitirá a los desarrolladores de software evaluar de forma objetiva sus productos, los cuales podrán enfocarse en un desarrollo centrado en el usuario en vez del desarrollo clásico por requerimientos funcionales. Se podrá entonces generar un software de mejor calidad y con mayor aceptación entre el grupo de los usuarios finales.

6.1.2 Reconocimiento facial

Como aportación adicional (debido a que no se encontraba dentro de los alcances del proyecto) se desarrolló e implementó un algoritmo de reconocimiento facial en UXLab, este algoritmo puede reconocer el rostro de los usuarios que han sido grabados con la cámara web.

Esto puede ayudar a los evaluadores UX a identificar expresiones representativas de ciertas emociones, además la implementación del código de reconocimiento facial puede obtener una representación de datos de la posición de puntos en la cara por lo que sería posible entrenar un algoritmo de aprendizaje para detectar las emociones de manera automático.

6.1.3 Generación de las métricas oculares

La generación de métricas oculares a partir de datos de puntos de visión y fijaciones usando OpenCV se considera una aportación ya que no existe hasta ahora una librería de acceso libre que haga esto, debido a que cada sensor de seguimiento ocular utiliza diferentes métodos de extracción de datos, licencias y software restringido. Los métodos desarrollados en esta tesis se pueden utilizar como base para generar la estructura de datos y métricas de seguimiento ocular sin la necesidad de usar software restringido.

6.1.3 Publicaciones

Se realizó una publicación de artículo en el volumen 2 de la revista Jornada de Ciencia y Tecnología Aplicada. Este artículo, Métricas UX mediante análisis de seguimiento ocular (Lagunes, González, López, Mujica y Castro, 2019), describe varias métricas oculares, su generación y su aplicación en un contexto de evaluación de la experiencia de usuario.

6.2 Trabajos futuros

6.2.1 Integración de pupilometría para medir estados mentales

Debido a las limitaciones físicas del sensor de seguimiento ocular utilizado durante la tesis, no fue posible utilizar los datos de pupila de los usuarios. Sería posible integrar los datos de la pupilometría al contexto del trabajo de tesis, varios trabajos en el estado del arte relacionan los cambios en la dilatación de la pupila con estímulos tanto emocionales como cognitivos, por lo que se podría realizar un estudio de la relación de esta métrica con la experiencia del usuario, haciendo énfasis especial en productos digitales.

Es importante mencionar que, si bien es cierto muchos trabajos mencionan la relación de los datos de la pupila con estados mentales, también se mencionan varias consideraciones que deben considerarse al trabajar con la pupila humana ya que es susceptible a cambios debido a muchos factores ambientales, como la iluminación.

6.2.2 Experimentos de correlación de parpadeos con estado cognitivo

Como trabajo futuro podría utilizarse UXLab para probar mediante experimentos estadísticos (parecidos a los mencionados en el capítulo 5) la correlación de la frecuencia de los parpados con los estados cognitivos de atención, concentración y aburrimiento. Una vez probada la correlación, estos estados cognitivos serían un gran aporte a las pruebas de UX.

Capítulo 7. Referencias

Alejandres Sánchez, H. O. (2017). Evaluación centrada en el usuario de sistemas de recomendación sensibles al contexto: efecto de interfaces multimodales interactivas y esquemas de explicación en la experiencia del usuario (Tesis de doctorado, TecNM/Cenidet).

Bačić, D. (2017). Understanding Business Dashboard Design User Impact: Triangulation Approach Using Eye-tracking, Facial Expression, Galvanic Skin Response and EEG Sensors. Twenty-third Americas Conference on Information Systems.

Bentivoglio, A. R.; Bressman, S. B.; Cassetta, E.; Carretta, D.; Tonali, P.; Albanese, A. (November 1997). "Analysis of blink rate patterns in normal subjects". *Movement Disorders*. 12 (6): 1028–1034. doi:10.1002/mds.870120629.

Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D. y Ertl, T. (2014). State-of-the-art of visualization for eye tracking data. *Eurographics Conference on Visualization (EuroVis)*. pp. 1-20

Borys, M. y Plechawska Wójcik, M. (2017). Eye-tracking metrics in perception and visual attention research. *European journal of medical technologies*, pp. 11-23.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Burger, G., Guna, J., y Pogacnik, M. (2018). Suitability of Inexpensive Eye-Tracking Device for User Experience Evaluations. *Sensors*, 18(6), 1822. Doi: 10.3390/s18061822

CASIA-IrisV3. (2005). CBSR, Center for Biometrics and Security Research. Recuperado de <http://www.cbsr.ia.ac.cn/IrisDatabase.htm>

Chen, S., Epps, J., Ruiz, N., y Chen, F. (2011). Eye activity as a measure of human mental effort in HCI. En *Proceedings of the 15th international conference on Intelligent*

user interfaces - IUI '11 (p. 315). New York, New York, USA: ACM Press. doi:10.1145/1943403.1943454

Coral, M. (2011). Analyzing cognitive workload through eye-related measurements: a meta-analysis, Tesis de maestría, Wright State University.

Djamasbi, S. (2014). Eye Tracking and Web Experience. AIS Transactions on Human-Computer Interaction, 6(2), págs. 37-54.

Duchowski, A., Price, M., Meyer, M. y Orero, P. (2012). Aggregate gaze visualization with real-time heatmaps. Proceedings of the Symposium on Eye Tracking Research and Applications. pp. 13-20.

Farnsworth B. (2018). Eye Tracking: The Complete Pocket Guide, Recuperado de <https://imotions.com/blog/eye-tracking/>

Fogarty, C., y Stern, J. A. (1989). Eye movements and blinks: Their relationship to higher cognitive processes. International Journal of Psychophysiology, 8(1), 35- 42. Doi: 10.1016/0167-8760(89)90017-2.

Fogarty, C., y Stern, J. A. (1989). Eye movements and blinks: Their relationship to higher cognitive processes. International Journal Of Psychophysiology, 8(1), 35- 42. Doi: 10.1016/0167-8760(89)90017-2.

Fouilloux Quiroz, D. E. (2018). Método para integrar y sincronizar datos EEG y multimedia para su aplicación en el proceso de evaluación de la experiencia del usuario (Tesis de maestría, TecNM/Cenidet).

Fuhl, W., Kuebler, T., Brinkmann, H., Rosenberg, R., Rosentiel, W. y Kasneci, E. (2018). Region of interest generation algorithms for eye tracking data. Proceedings of the 3rd workshop on eye tracking and visualization. pp. 1-9.

Fukuda, K. (1994). Analysis of eyeblink activity during discriminative tasks. *Perceptual and Motor Skills*, 79, 1599–1608.

Fukuda, K. (2001). Eye blinks: New indices for the detection of deception. *International Journal of Psychophysiology*, 40, 239–245.

Fukunaga, Keinosuke; Larry D. Hostetler (January 1975). "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition". *IEEE Transactions on Information Theory*. 21 (1), págs: 32–40. Doi: 10.1109/TIT.1975.1055330

García Pinzón, G. A. (2020). Procesamiento de datos fisiológicos para detectar estados afectivos en el proceso de evaluación de la experiencia de usuario (Tesis de maestría, TecNM/Cenidet).

Gaspar, P., Simko, J., Kompan, M., y Bielikova, M. (2018). Analysis of User Behavior in Interfaces with Recommended Items: An Eye-tracking Study. 5th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems co-located with ACM Conference on Recommender Systems (RecSys 2018). Vancouver.

Goldstein, I. (2000). Intersubjective properties by which we specify pain, pleasure, and other kinds of mental states. *Philosophy*, 89-04.

González Franco, N. (2017). Metodología UXEEg para la evaluación de la Experiencia del Usuario en personas con discapacidad a partir de Interfaces Cerebro Computadora (Tesis de doctorado, TecNM/Cenidet).

González, G. (2019). QuEXLabArquitectura [Diapositiva de PowerPoint]. Repositorio Google Drive del Centro de Investigación y Desarrollo Tecnológico.

Guo, F., Ding, Y., Liu, W., Liu, C., y Zhang, X. (2016). Can eye-tracking data be measured to assess product design?: Visual attention mechanism should be considered. *International Journal of Industrial Ergonomics*, 53, pp. 229-235. doi:10.1016/j.ergon.2015.12.001

Haddad, R., y Akansu, A. (1991) "A Class of Fast Gaussian Binomial Filters for Speech and Image Processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 39, pp 723-727.

He, S., R. Tavakoli, H., Borji, A., y Pugeault, N. (2019). A Synchronized Multi-Modal Attention-Caption Dataset and Analysis.

Herbert, D. (2007). *Osu!* [Programa de computadora]. Australia: Wembley.

Ho, H. (2014). The effects of controlling visual attention to handbags for women in online shops: Evidence from eye movements. *Computers in Human Behavior*, 30, 146-152. doi:10.1016/j.chb.2013.08.006

Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H. y van de Weijer, J. (2010). *Eye tracking: a comprehensive guide to methods and measures*. UK: Oxford University Press.

Hussain, J., Ali Khan, W., Hur, T., Bilal, H., Bang, J., Hassan, A., Lee, S. (2018). A Multimodal Deep Log-Based User Experience (UX) Platform for UX Evaluation. *Sensors*, 18(5). Doi: 10.3390/s18051622

Ichikawa, N., y Ohira, H. (2004). Eyeblink activity as an index of cognitive processing: Temporal distribution of eyeblinks as an indicator of expectancy in semantic priming. *Perceptual and Motor Skills*, 98, 131–140.

Iskra, A., y Gabrijelčič Tomc, H. (2016). Eye-tracking analysis of face observing and face recognition. *Journal of Graphic Engineering and Design*, 7(1), 5–11. <https://doi.org/10.24867/JGED-2016-1-005>

K. Muthumanickam, P., Vrotsou, K., Nordman, A., Johansson, J., y Cooper, M. (2019). Identification of Temporally Varying Areas of Interest in Long-Duration Eye-Tracking Data Sets. *IEEE transactions on visualization and computer graphics*, 25(1). doi:10.1109/TVCG.2018.2865042

Kim, J., Sunil Kumar, Y., Yoo, J., y Kwon, S. (2018). Change of Blink Rate in Viewing Virtual Reality with HMD. *Symmetry*. Doi: doi: 10.3390/sym10090400

Kosch, T., Hassib, M., Buschek, D., y Schmidt, A. (2018). Look into My Eyes: Using Pupil Dilation to Estimate Mental Workload for Task Complexity Adaptation. (pp. LBW617:1--LBW617:6). New York: ACM. doi:10.1145/3170427.3188643

Kurdi, B., Lozano, S., y Banaji, M. (2016). Introducing the Open Affective Standardized Image Set (OASIS). *Behavior Research Methods*, 49. <https://doi.org/10.3758/s13428-016-0715-3>

Kroenke, K., Spitzer, R. L., y Williams, J. B. W. (2001). The PHQ-9. *Journal of General Internal Medicine*, 16(9), 606–613. <https://doi.org/10.1046/j.1525-1497.2001.016009606.x>

Lagunes Ramírez, D. A., González Serna, G., López Sánchez, M., Mujica Vargas, D., y Castro Sánchez, N. (2019). Métricas UX mediante análisis de seguimiento ocular. *Jornada de Ciencia y Tecnología Aplicada*, 2(2), 33–37. [https://jcyta.cenidet.tecnm.mx/revistas/jcyta/Revista_JCyTA_Vol-2-Num-2_Nov_2019\(120320\).pdf](https://jcyta.cenidet.tecnm.mx/revistas/jcyta/Revista_JCyTA_Vol-2-Num-2_Nov_2019(120320).pdf)

Lang, P.J., Bradley, M.M., y Cuthbert, B.N. (2008). International affective picture system (IAPS): Affective ratings of pictures and instruction manual. Technical Report A-8. University of Florida, Gainesville, FL.

Lu, Y., Zheng, W.-L., Li, B., y Lu, B.-L. (2015). Combining Eye Movements and EEG to Enhance Emotion Recognition. 24th International Conference on Artificial Intelligence (pp. 1170--1176). Buenos Aires: AAAI Press.

Lumen Research. (2019). Heatmappy, Draw image and video heatmaps in python, Recuperado de <https://github.com/LumenResearch/heatmappy>

Mallick, R., Slayback, D., Touryan, J., Ries, A., y Lance, B. (2016). The Use of Eye Metrics to Index Cognitive Workload in Video Games. IEEE Second Workshop on Eye Tracking and Visualization (ETVIS) (pp. 60-64). Baltimore: U.S. Government. doi:10.1109/ETVIS.2016.7851168

McIntire, L., McKinley, R., Goodyear, C., y McIntire, J. (2014). Detection of vigilance performance using eye blinks. *Applied Ergonomics*, 45(2), 354-362. doi:10.1016/j.apergo.2013.04.020

Moltó, J., Segarra, P., López, R., Esteller, Àngels, Fonfría, A., Pastor, M. C., y Poy, R. (2013). Spanish adaptation of the "International Affective Picture System" (IAPS). Third part. *Anales De Psicología / Annals of Psychology*, 29(3), 965-984. <https://doi.org/10.6018/analesps.29.3.153591>

Morales Morante, J., A. (2020). Máquina de aprendizaje para el reconocimiento de emociones mediante datos fisiológicos multimodales. (Tesis de maestría, TecNM/Cenidet).

Nakano, T., Kato, M., Morito, Y., Itoi, S., y Kitazawa, S. (2013). Blink-related momentary activation of the default mode network while viewing videos. *Proceedings of the National Academy of Sciences*, 110(2), 702-706. <https://doi.org/10.1073/pnas.1214804110>

Nixon, M., y Aguado, A. (2008) *Feature Extraction and Image Processing*. Academic Press, p. 88.

Ohira, H. (1995). Analysis of eyeblink activity during self-referent information processing in mild depression. *Perceptual and Motor Skills*, 81, 1219–1229.

Ohira, H. (1996). Eyeblink activity in a word-naming task as a function of semantic priming and cognitive load. *Perceptual and Motor Skills*, 82, 835–842.

Ohira, H., Winton, W. M., y Oyama, M. (1998). Effects of stimulus valence on recognition memory and endogenous eyeblinks: Further evidence for positive-negative asymmetry. *Personality and Social Psychology Bulletin*, 24, 986–993.

OpenCV. (2015). Open Source Computer Vision Library.

Qu, Q.-X., Zhang, L., Chao, W.-Y., y Duffy, V. (2016). User experience design based on eye-tracking technology: a case study on smartphone APPs. *Applied Human Factors and Ergonomics International*. Orlando, Florida: ResearchGate.

Romano Bergstrom, J., y Schall, A. J. (2014). *Eye Tracking in User Experience Design*. Amsterdam: Morgan Kaufmann.

Russell, James. (1980). A Circumplex Model of Affect. *Journal of Personality and Social Psychology*. 39. 1161-1178. 10.1037/h0077714.

Sakai, T., Tamaki, H., Ota, Y., Egusa, R., Inagaki, S., Kusunoki, F., Sugimoto, M. y Mizoguchi, H. (2017). EDA-BASED ESTIMATION OF VISUAL ATTENTION BY OBSERVATION OF EYE BLINK FREQUENCY. *International Journal on Smart Sensing and Intelligent Systems*, 10(2), 296–307. <https://doi.org/10.21307/ijssis-2017-212>

Salvucci, D. D. y Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. *Proceedings of the 2000 symposium on eye tracking research y applications*. pp. 71-78

Santella, A. y DeCarlo, D. (2004). Robust clustering of eye movement recordings for quantification of visual interest. *Proceedings of the eye tracking research y applications symposium on eye tracking research y applications*. pp. 27-34

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. y Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python, pp. 2825-2830.

Shapiro, S. S., y Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4), 591-611. <https://doi.org/10.1093/biomet/52.3-4.591>

Shojaeizadeh, M., Djamasbi, S., Paffenroth, R. C., y Trapp, A. C. (2019). Detecting task demand via an eye tracking machine-learning system. *Decision Support Systems*, 116, 91-101. <https://doi.org/10.1016/j.dss.2018.10.012>

Smilek, D., Carriere, J. S. A., y Cheyne, J. A. (2010). Out of Mind, Out of Sight. *Psychological Science*, 21(6), 786-789. <https://doi.org/10.1177/0956797610368063>

Smith, P., Shah, M., y Lobo, N. da V. (2003). Determining driver visual attention with one camera. *IEEE Transactions on Intelligent Transportation Systems*, 4(4), 205-218. <https://doi.org/10.1109/TITS.2003.821342>

Spielberger, C. D., Gorsuch, R. L., Lushene, R., Vagg, P. R., & Jacobs, G. A. (1983). *Manual for the State-Trait Anxiety Inventory*. Palo Alto, CA: Consulting Psychologists Press.

Soriano Terrazas, J. (2018). Metodología para caracterizar e inducir estados mentales a través de realidad virtual inmersiva e interfaz cerebro computadora (tesis de maestría). Morelos, México: CENIDET.

Stern, J. A., Walrath, L. C., y Goldstein, R. (1984). The endogenous eyeblink. *Psychophysiology*, 21, 22–33.

Student Sealy, W., (1908), THE PROBABLE ERROR OF A MEAN. *Biometrika*, 6(1), 1–25. <https://doi.org/10.1093/biomet/6.1.1>

Tzafilkou, K., y Protogeros, N. (2017). Diagnosing user perception and acceptance using eye tracking in web-based end-user development. *Computers in Human Behavior*, 72, 23-37. doi:10.1016/j.chb.2017.02.035

Wang, Q., Yang, S., Liu, M., Cao, Z., y Ma, Q. (2014). An eye-tracking study of website complexity from cognitive load perspective. *Decision Support Systems*, 62, 1-10. doi:10.1016/j.dss.2014.02.007

Watson, D., Clark, L. A., & Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: The PANAS scales. *Journal of Personality and Social Psychology*, 54(6), 1063–1070. <https://doi.org/10.1037/0022-3514.54.6.1063>

Zagermann, J., Pfeil, U., y Reiterer, H. (2018). Studying Eye Movements as a Basis for Measuring Cognitive Load. En *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18* (pp. 1–6). New York, New York, USA: ACM Press. <https://doi.org/10.1145/3170427.3188628>

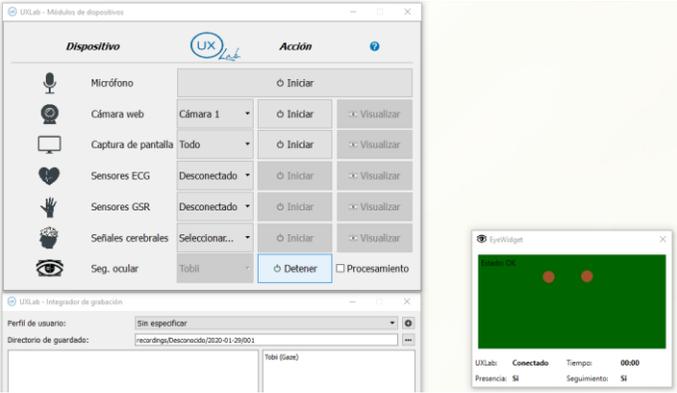
Zagermann, J., Pfeil, U., y Reiterer, H. (2016). Measuring Cognitive Load using Eye Tracking Technology in Visual Computing. *Proceedings of the Sixth Workshop on*

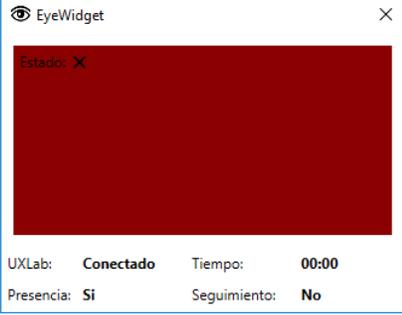
Beyond Time and Errors on Novel Evaluation Methods for Visualization, BELIV'16, págs.78-85. doi:10.1145/2993901.2993908

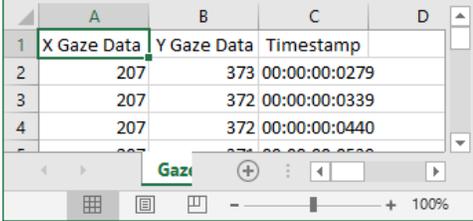
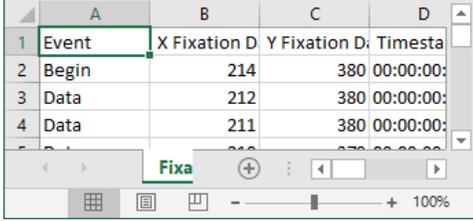
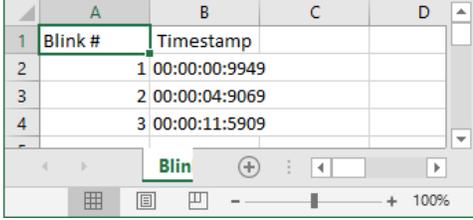
Zheng, W.-L., Dong, B.-N., y Lu, B.-L. (2014). Multimodal Emotion Recognition using EEG and Eye Tracking Data. 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 5040-5043). Chicago: IEEE. doi:10.1109/EMBC.2014.6944757

Capítulo 8. Anexos

Anexo 1. Reporte de Pruebas UXLab, Módulo de Seguimiento Ocular

<p>UX-G-1</p>	<p>Conexión de UXLab con EyeWidget: Al iniciar el módulo de seguimiento ocular con el sensor Tobii se abre una ventana del programa EyeWidget el cual muestra el estado de UXLab como “conectado”</p>
<p>Evidencia</p>	
<p>UX-G-2</p>	<p>Representación de los ojos del usuario: Inmediatamente cuando se abre EyeWidget se pueden observar 2 puntos de color café en la pantalla los cuales representan los ojos del usuario. Si el sensor de seguimiento ocular no está detectando los ojos la pantalla no muestra los puntos de color café.</p>
<p>Evidencia</p>	
<p>UX-G-3</p>	<p>Comportamiento de los ojos en movimiento: Cuando el usuario mueve su cabeza esto se ve reflejado en el programa, se demuestra como el sensor de seguimiento identifica la posición de los ojos y mientras aparece en pantalla el color verde y los ojos, no existe ningún problema con el seguimiento ocular.</p>
<p>Evidencia</p>	

	
<p>UX-G-4</p>	<p>Comportamiento de los ojos con parpadeo: Cuando el usuario cierra los ojos estos desaparecen de la pantalla del programa EyeWidget sin embargo la pantalla se mantiene en verde a menos que cierre los ojos por más de 1 segundo.</p>
<p>Evidencia</p>	
<p>UX-G-5</p>	<p>Comportamiento de los ojos observando fuera de la pantalla: Cuando el usuario observa un lugar fuera de los límites de la pantalla el programa EyeWidget se mostrará en rojo indicando que no se está realizando un seguimiento. Se puede observar que el programa muestra que hay presencia de usuario, pero no seguimiento ocular.</p>
<p>Evidencia</p>	
<p>UX-G-6</p>	<p>Comportamiento con los ojos cerrados: Cuando el usuario cierra los ojos o el sensor es bloqueado, el programa indica que no hay seguimiento y no se detecta la presencia del usuario.</p>

<p>Evidencia</p>	
<p>UX-G-7</p>	<p>Grabación de archivo de puntos de visión: Cuando se inicia la grabación, comienza a correr el tiempo en el programa EyeWidget, al finalizar la grabación se puede leer el archivo CSV de datos de visión que se grabaron.</p>
<p>Evidencia</p>	
<p>UX-G-8</p>	<p>Grabación de archivo de puntos de fijación: Cuando se inicia la grabación, comienza a correr el tiempo en el programa EyeWidget, al finalizar la grabación se puede leer el archivo CSV de datos de fijación que se grabaron.</p>
<p>Evidencia</p>	
<p>UX-G-9</p>	<p>Grabación de archivo de parpadeos: Cuando se inicia la grabación, comienza a correr el tiempo en el programa EyeWidget, al finalizar la grabación se puede leer el archivo CSV de datos de parpadeo que se grabaron.</p>
<p>Evidencia</p>	
<p>UX-G-10</p>	<p>Detener grabación:</p>

Cuando el usuario detiene la grabación, el programa EyeWidget reinicia el tiempo y su estado cambia a “esperando”. UXLab bloqueara el botón de grabar hasta que se terminen de procesar los archivos de la grabación anterior. Finalmente, al terminar se podrán visualizar los archivos en la carpeta de guardado de la prueba.

Evidencia

The evidence consists of three screenshots. The top screenshot shows the EyeWidget window with a green background and two red dots, displaying 'Estado: OK' and status information: 'UXLab: Esperando...', 'Tiempo: 00:00:00:0000', 'Presencia: Si', and 'Seguimiento: Si'. The middle screenshot shows the 'UXLab - Integrador de grabación' window with fields for 'Perfil de usuario' (Sin especificar) and 'Directorio de guardado' (E:/recordings/Desconocido/2020-01-29/002), and a list of recording details including 'Finished text (Tobi) stream, gaze, Gaze_1.csv, 3712150.6757 seconds to 3712166.69166 seconds' and a 'Terminar grabacion' button. The bottom screenshot shows a file explorer window for 'eyetracking_data' containing three CSV files: 'BlinkDataOutput.csv' (1 KB), 'FixationDataOutput.csv' (172 KB), and 'GazeDataOutput.csv' (173 KB).

UX-G-11 Grabar de nuevo:
 Cuando se termina una grabación en UXLab y se terminan de procesar los archivos al dar clic de nuevo en el botón grabar, UXLab comenzara a grabar los sensores que se encuentren iniciados en ese momento. Finalmente, cuando termine la nueva grabación los archivos de esa sesión se encontrarán guardados en la carpeta especificada en UXLab.

Evidencia

The evidence consists of two screenshots. The left screenshot shows the 'UXLab - Módulos de dispositivos' window with a list of sensors (Microfono, Cámara web, Captura de pantalla, Sensores ECG, Sensores GSR, Señales cerebrales, Seg. ocular) and an 'EyeWidget' window overlaid on top. The right screenshot shows the 'UXLab - Integrador de grabación' window with recording details: 'Stop recording, 3712290.68939 seconds to 3712323.71973 seconds, 33.0303447 total', 'Finished text (Tobi) stream, gaze, Gaze_1.csv, 3712290.81502 seconds to 3712322.84706 seconds', and 'Start recording at 3712386.44938 seconds', with a 'Terminar grabacion' button.

UX-G-12 Desconexión entre UXLab y EyeWidget:

Al presionar el botón de detener el módulo de seguimiento ocular se envía una señal al software de EyeWidget que termina el programa realizando una desconexión de UXLab. Como se realiza la desconexión del socket este proceso tarda uno o dos segundos

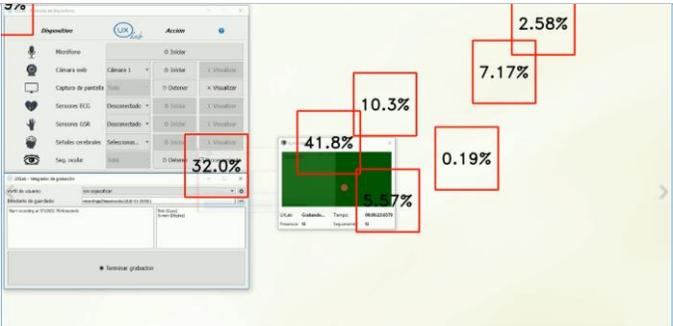
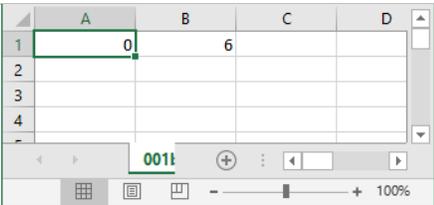
Evidencia

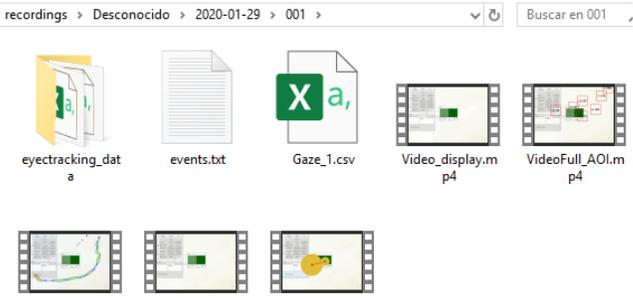
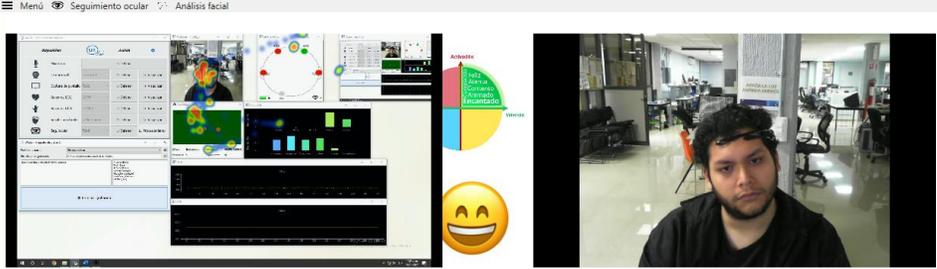
UX-G-13 **Procesamiento de video mapa de calor:**
Al terminar una grabación con seguimiento ocular se genera el mapa de calor del video capturado en la pantalla del usuario.

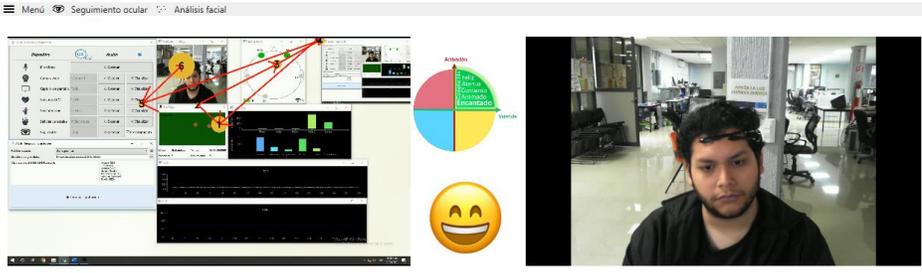
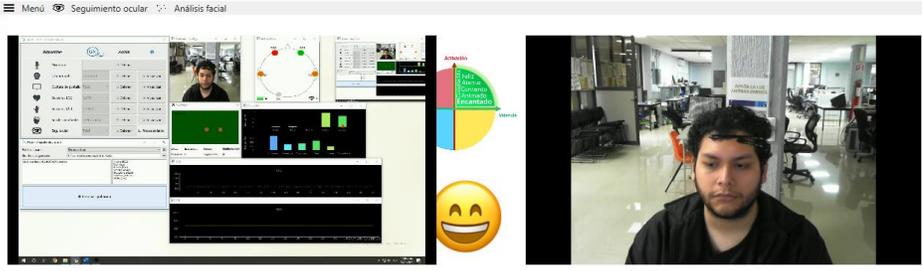
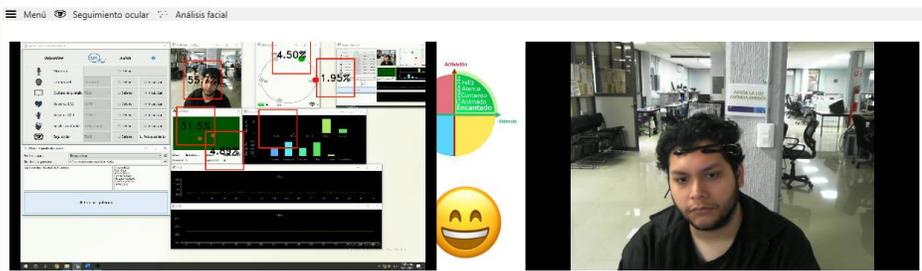
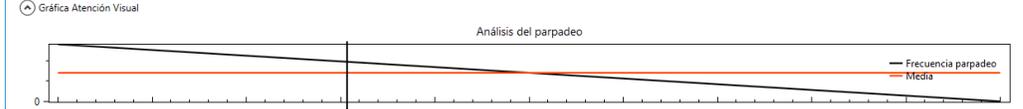
Evidencia

UX-G-14 **Procesamiento de video mapa de rutas:**
Al terminar una grabación con seguimiento ocular se genera el mapa de rutas del video capturado en la pantalla del usuario.

Evidencia

<p>UX-G-15</p>	<p>Procesamiento de video áreas de interés: Al terminar una grabación con seguimiento ocular se generan las áreas de interés del video capturado en la pantalla del usuario.</p>
<p>Evidencia</p>	
<p>UX-G-16</p>	<p>Procesamiento de archivo de análisis del parpadeo: Al terminar una grabación con seguimiento ocular se genera el archivo de parpadeos del usuario. La primera columna representa el minuto de la grabación, la segunda el número de parpadeos capturados en ese minuto.</p>
<p>Evidencia</p>	
<p>UX-G-17</p>	<p>Procesamiento de video de puntos: Al terminar una grabación con seguimiento ocular se generan los puntos en el video capturado en la pantalla del usuario.</p>
<p>Evidencia</p>	
<p>UX-I-1</p>	<p>Realizar grabación solo con seguimiento ocular y memoria de video: Al realizar una prueba completa grabando solo la pantalla y el seguimiento ocular (además de activar la casilla de postprocesamiento) se generan todos los archivos de video con métricas oculares y archivos de datos oculares.</p>

<p>Evidencia</p>	
<p>UX-I-2</p>	<p>Realizar grabación con todos los sensores: Al realizar una prueba completa con todos los sensores disponibles de UXLab (además de activar la casilla de postprocesamiento) se generan todos los archivos de video con métricas oculares, archivos de datos oculares y demás archivos que apoyan la parte emocional.</p>
<p>Evidencia</p>	
<p>UX-R-1</p>	<p>Visualizar mapa de calor: Utilizando el reproductor de UXLab y la grabación resultante de la prueba anterior se puede seleccionar del menú de métricas de seguimiento ocular, la opción de ver el mapa de calor. Al seleccionar la opción se mostrará el video en la pantalla de video de la prueba.</p>
<p>Evidencia</p>	
<p>UX-R-2</p>	<p>Visualizar mapa de rutas: Utilizando el reproductor de UXLab y la grabación resultante de la prueba anterior se puede seleccionar del menú de métricas de seguimiento ocular, la opción de ver el mapa de rutas. Al seleccionar la opción se mostrará el video en la pantalla de video de la prueba.</p>

<p>Evidencia</p>	
<p>UX-R-3</p>	<p>Visualizar puntos de visión: Utilizando el reproductor de UXLab y la grabación resultante de la prueba anterior se puede seleccionar del menú de métricas de seguimiento ocular, la opción de ver el video de puntos. Al seleccionar la opción se mostrará el video en la pantalla de video de la prueba.</p>
<p>Evidencia</p>	
<p>UX-R-4</p>	<p>Visualizar áreas de interés: Utilizando el reproductor de UXLab y la grabación resultante de la prueba anterior se puede seleccionar del menú de métricas de seguimiento ocular, la opción de ver el video de áreas de interés. Al seleccionar la opción se mostrará el video en la pantalla de video de la prueba.</p>
<p>Evidencia</p>	
<p>UX-R-5</p>	<p>Visualizar gráfica de parpadeos: Utilizando el reproductor de UXLab y la grabación resultante de la prueba anterior se puede seleccionar del menú de métricas de seguimiento ocular, la opción de ver la gráfica de parpadeos. Al seleccionar la opción se mostrará en la parte de debajo de la interfaz, la gráfica que muestra el análisis del parpadeo.</p>
<p>Evidencia</p>	

<p>UX-R-6</p>	<p>Visualizar estados mentales sincronizados con métricas oculares:</p> <p>Mientras se reproduce una prueba UX el software UXLab reproductor muestra una línea de tiempo y una línea que muestra el punto actual, la evidencia muestra la información que capturaban los sensores fisiológicos en ese momento específico de la prueba (al mismo tiempo o sincronizados con la información de los demás sensores).</p>
<p>Evidencia</p>	
<p>UX-I-3</p> <p>UX-I-4</p>	<p>Visualizar todas las métricas oculares en una sesión UX</p> <p>Grabar con todos los sensores y reproducir la sesión. Observando la sincronización de estados mentales con métricas oculares:</p> <p>En esta última prueba se realizó la prueba desde la grabación de una prueba UX completa (con todos los sensores disponibles) y después de la grabación se guardaron todos los archivos de la prueba en la ruta especificada. No se encontraron problemas ni en la visualización o en la grabación de la prueba.</p>

Evidencia

