

INTERFAZ PARA DISEÑAR Y CONFIGURAR PRUEBAS DE INVESTIGACIÓN NEUROLÓGICA



Daniel Alejandro Rodríguez López
Mario Felipe Soto Zambrano

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Línea de Investigación en Robótica Médica
Ingeniería en Automática Industrial

Popayán, Julio, 2019

INTERFAZ PARA DISEÑAR Y CONFIGURAR PRUEBAS DE INVESTIGACIÓN NEUROLÓGICA

Daniel Alejandro Rodríguez López
Mario Felipe Soto Zambrano

Director: Mg. Karin Correa Arana
Codirector: PhD. Oscar Andrés Vivas Albán

Monografía presentada como requisito parcial para obtener al
título de:

Ingeniero en Automática Industrial

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Línea de Investigación en Robótica Médica
Ingeniería en Automática Industrial

Popayán, Julio, 2019

Nota de Aprobación

Director

Jurado

Jurado

Fecha:

Agradecimientos

A Dios gracias por permitirnos alcanzar diferentes objetivos a lo largo de nuestra vida y sobre todo por la salud, fuerza y empeño recibidos durante el desarrollo de este proyecto.

A nuestras familias, por el gran apoyo brindado en todo momento y por confiar en nosotros siempre, sin su vital ayuda nada de lo presente hubiese sido posible.

Agradecer a nuestro director Oscar Andrés Vivas por la confianza depositada en nosotros para la realización de este trabajo.

A la empresa *Aura Innovative Robotics*, por permitirnos colaborar en el interesante proyecto de *Oscann desk100*, especialmente a Cecilia García Cena y a Erik Hernández Serrato, quienes estuvieron pendientes en todo momento durante nuestra estadía en España, proporcionando una oportuna asesoría en esta investigación.

A la Universidad del Cauca, especialmente a la Facultad de Ingeniería Electrónica la cual con sus docentes y administrativos siempre han estado a nuestro servicio.

A los evaluadores de este proyecto que con sus críticas constructivas incrementarán el alcance de esta investigación.

Resumen

La plataforma *Oscann desk100*, desarrollada por la empresa Española *Aura Innovative Robotics SL.*, es una herramienta que brinda soporte al diagnóstico diferencial de enfermedades neurodegenerativas, a partir, de los movimientos oculares registrados durante pruebas que estimulan visualmente al sujeto.

El banco de pruebas de la plataforma mencionada, a febrero de 2018, no presentaba aleatoriedad en el diseño y la construcción de estímulos, lo cual motiva el presente trabajo, con el objetivo de aportar mayor flexibilidad en este apartado, realizando la implementación de una interfaz gráfica para la configuración personalizada de experimentos.

Durante el desarrollo de este proyecto, se describen las diferentes áreas que se abordan a través del proyecto, como por ejemplo los conceptos generales en la anatomía del ojo, las principales técnicas de seguimiento ocular existentes y los software más difundidos para este tipo de aplicaciones en el campo científico.

Adicionalmente, se encuentran descritos cada uno de los elementos empleados para la implementación de la interfaz, incluyendo la descripción de las principales funcionalidades en la estructuración de la aplicación.

Es importante nombrar la herramienta software *Qt*, utilizada para el desarrollo de este trabajo, que permite realizar aplicaciones multiplataforma, a partir de la integración de diversos lenguajes de programación como: QML, C, C++, C# y JavaScript entre otros.

Finalmente se realiza la validación de la interfaz a partir de la experiencia de usuario, donde se analizan los datos obtenidos y se establecen trabajos futuros en esta rama.

Palabras clave: Interfaz gráfica, Enfermedades neurodegenerativas, Seguimiento ocular, Movimientos sacádicos, Software.

Abstract

The platform *Oscann desk100*, developed by the Spanish company *Aura Innovative Robotics SL.*, is a tool that provides support to the differential diagnosis of neurodegenerative diseases, from the ocular movements registered during tests that visually stimulate the subject.

The test bench of the mentioned platform, in February 2018, did not present randomness in the design and construction of stimuli, which motivates the present work, to provide greater flexibility in this section, implement a graphical interface for the personalized configuration of experiments.

During the development of this project, the different areas that are addressed through the project are described, such as the general concepts in the anatomy of the eye, the main existing eye tracking techniques and the most widespread software for this type of applications in the scientific field.

Additionally, each of the elements used for the implementation of the interface is described, including the description of the main functionalities in the structuring of the application.

It is important to name the software tool *Qt*, used for the development of this work, which allows multiplatform applications, from the integration of various programming languages such as QML, C, C++, C#, and JavaScript among others.

Finally, the interface is validated on the basis of the user experience, where the data obtained is analyzed and future work is established in this branch.

Keywords: GUI, Neurodegenerative diseases, Eye tracking, Saccadic movements, Software.

Índice general

Agradecimientos	VII
Resumen	IX
Abstract	XI
Índice	XIV
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Estructura del documento	2
2. Conceptos Generales	3
2.1. Anatomía del ojo humano	3
2.2. Funcionamiento del ojo humano	3
2.3. Movimientos Oculares	4
2.3.1. Movimientos voluntarios	6
2.3.2. Movimientos involuntarios	7
2.4. Alteraciones vinculadas a las patologías neurodegenerativas	9
3. Estado del arte	13
3.1. Repercusión de las enfermedades neurodegenerativas a nivel mundial	13
3.2. Tecnología <i>Eye Tracking</i>	15
3.2.1. Marco histórico de las técnicas de <i>Eye Tracking</i>	15
3.2.2. Estado actual de las técnicas de <i>Eye Tracking</i>	17
3.3. Estado actual en interfaces software	21
3.3.1. E-Prime	22
3.3.2. PsychoPy	24
3.3.3. Experiment Builder	25
3.3.4. DMDX	26
3.3.5. OpenSesame	27
3.3.6. Psychophysics Toolbox	28
3.3.7. Oscann desk100	29

4. Arquitectura Software	33
4.1. Requerimientos	33
4.1.1. Configuración y diseño de pruebas	33
4.1.2. Diseño de estímulos	39
4.2. Metodología	41
4.3. Historias de usuario	41
4.4. Casos de uso	48
4.5. Interfaz desarrollada	49
4.5.1. Interfaz para configuración y diseño de pruebas	50
4.5.2. Interfaz para el diseño de estímulos	56
5. Resultados	63
5.1. Proceso de validación de la interfaz	63
5.1.1. Perfiles de usuario	63
5.2. Protocolo	64
5.3. Resultados	65
5.3.1. Reacciones generales del sistema	65
5.3.2. Aprendizaje	67
5.3.3. Terminología e Información del sistema	72
5.3.4. Características del sistema	75
6. Conclusiones	77
6.1. Conclusiones	77
6.2. Desarrollos futuros	78
Bibliografía	79
Anexos	84
A. Manual para la configuración gráfica del estímulo	87
B. Manual de uso para el software de diseño de pruebas	99
C. Cuestionario para pruebas de validación	105
D. Desarrollo software	109

Índice de figuras

2.1. Anatomía del ojo humano	4
2.2. Musculatura extrínseca del ojo	5
2.3. Ejes de Fick y plano de Listing	6
2.4. Centro de rotación del ojo	6
2.5. Micromovimientos involuntarios durante la fijación [1]	9
3.1. Dispositivo <i>Oscann desk100</i> [2]	15
3.2. <i>Eye tracker</i> desarrollado por Edmund Huey [3]	16
3.3. Primer <i>eye tracker</i> no invasivo desarrollado por Dodge y Cline [4]	16
3.4. <i>Scleral search coil</i> [1]	17
3.5. <i>EyeContact</i> [5]	18
3.6. Dispositivo EOG [6]	19
3.7. Dispositivo de vídeo-oculografía montado, por <i>Ergoneers</i> [7]	20
3.8. Dispositivo <i>Tobii EyeX</i> ubicado en la parte inferior de un monitor [8]	21
3.9. Dispositivo no remoto <i>Oscann desk100</i> [9]	21
3.10. Entorno de configuración de pruebas de E-Prime [10].	22
3.11. Ejemplos de pruebas diseñadas con CAMCI-Research [11].	23
3.12. Prueba realizada con E-prime para estudio de movimientos trans-sacádicos [12]	23
3.13. Entorno de desarrollo integrado (IDE) ejecutando uno de los scripts de demostración [13]	24
3.14. Espacio de trabajo de la herramienta Experiment Builder [14]	25
3.15. Interfaz de configuración de pruebas en Visual DMDX [15]	26
3.16. Interfaz de OpenSesame [16]	27
3.17. Test para analizar la incidencia de un objeto distractor sobre la curvatura de una sacada [17]	28
3.18. Test diseñado con Psychophysics Toolbox de MatLab [18]	29
3.19. Puntos de fijación utilizados en la etapa de calibración de la prueba [9]	30
3.20. Ventana inicial de la plataforma <i>Oscann</i> [9]	30
3.21. Ambiente de la interfaz de configuración de pruebas <i>Oscann desk100</i> [9]	31
4.1. Secuencia de eventos de pruebas con antisacadas, sacadas y sacadas de memoria [19].	35
4.2. Resultados de test de seguimiento lento sinusoidal [20]	36
4.3. Esquema de funcionamiento de los paradigmas <i>Overlap</i> y <i>Gap</i> [21].	37
4.4. Diagrama de casos de uso de la interfaz	48
4.5. Diagrama de relaciones entre casos de uso de la interfaz	49

4.6.	Sección de configuración de pruebas	50
4.7.	Tabla de visualización de datos para configuración de pruebas	51
4.8.	Botonería auxiliar para la configuración de pruebas	52
4.9.	Cuadro de dialogo para repetición de elementos	52
4.10.	Cuadro de dialogo para finalizar con la configuración de pruebas	53
4.11.	Parámetros de configuración de los distintos test disponibles en la interfaz	54
4.12.	Ventana inicial de previsualización	55
4.13.	Botón para ejecutar previsualización de pruebas	55
4.14.	Modo demostración en un movimiento antisacádico	56
4.15.	Interfaz para el diseño de estímulos.	56
4.16.	Sección para seleccionar el aspecto del estímulo basado en formas geométricas o imágenes.	57
4.17.	Cuadro de dialogo auxiliar para carga las imágenes en formato JPG o PNG.	57
4.18.	Sección para modificar valores de tamaño y rotación.	58
4.19.	Sección para modificar tonos en el estímulo y el fondo de la aplicación.	58
4.20.	Paleta de colores empleada en la interfaz	59
4.21.	Sección para modificar brillo, contraste y saturación.	59
4.22.	Sección para aplicar el efecto de borrosidad o difuminado.	60
4.23.	Sección para configurar animaciones.	60
4.24.	Sección para configurar las trayectorias del estímulo.	61
4.25.	Tabla para visualizar los datos configurados, junto con los botones que permiten la exportación y restablecimiento de valores por defecto.	61
5.1.	Resultados 1.1: Terrible - Estupendo	65
5.2.	Resultados 1.2: Complejo - Intuitivo	66
5.3.	Resultados 1.3: Frustrante - Satisfactorio	66
5.4.	Resultados 1.4: Rígido - Flexible	67
5.5.	Resultados 2.1: Aprendiendo a manejar el sistema	67
5.6.	Resultados 2.2: Explorando nuevas características por prueba y error	68
5.7.	Resultados 2.3: Recordando nombres o uso de comandos	68
5.8.	Resultados 2.4: Toma poco tiempo aprender a manipular el software	69
5.9.	Resultados 2.5: Las tareas pueden ser realizadas de manera sencilla	69
5.10.	Resultados 2.6: Pienso que la mayoría de las personas aprenderá a usar la herramienta muy rápidamente	70
5.11.	Resultados 2.7: En un inicio aprender a utilizar esta herramienta es difícil	70
5.12.	Resultados 2.8: Mensajes de ayuda en la herramienta	71
5.13.	Resultados 2.9: Creo que necesitaría el apoyo de una persona técnica para poder usar este sistema	71
5.14.	Resultados 2.10: Necesitaba aprender muchas cosas antes de poder comenzar a usar el sistema	72
5.15.	Resultados 3.1: Uso de términos en todo el sistema	72
5.16.	Resultados 3.2: La terminología usada está relacionada con las tareas que está haciendo	73
5.17.	Resultados 3.3: Mensajes en la ventana que solicitan ingresar información por el usuario	73
5.18.	Resultados 3.4: Organización de la información en el sistema	74

5.19. Resultados 3.5: Puedo comprender y manejar la información proporcionada por este software	74
5.20. Resultados 4.1: Fiabilidad del sistema	75
5.21. Resultados 4.2: El software en algún momento se detuvo inesperadamente	75
5.22. Resultados 4.3: Velocidad del sistema	76
1. Pantalla principal de la interfaz para configuración gráfica del estímulo.	87
2. Definición de las secciones disponibles en la interfaz de configuración del estímulo.	88
3. Convención de colores de las secciones	88
4. Sección para elegir la forma representativa del estímulo	89
5. Cuadro de diálogo para cargar imágenes	89
6. Cambios en la rotación y tamaño del estímulo	90
7. Secciones disponibles en el estímulo y colores correspondientes	90
8. Sección para elegir la forma representativa del estímulo y los colores correspondientes.	91
9. Cuadro de diálogo auxiliar que despliega la paleta de colores.	91
10. Variaciones en los valores de brillo.	92
11. Variaciones en los valores de contraste.	93
12. Variaciones en los valores de saturación.	94
13. Variaciones en los valores de difuminación.	95
14. Sección de controles para configurar animaciones.	96
15. Sección de controles para configurar las trayectorias.	96
16. Espacio de visualización del estímulo.	97
17. Tabla disponible para la configuración de parámetros configurados.	97
18. Botones disponibles para agregar, restablecer y exportar datos.	97
19. Cuadro de diálogo auxiliar, solicitando un nombre para el archivo de exportación.	98
20. Sección 1 para diseñar test de sacadas visuales	101
21. Sección para diseñar tareas de seguimientos de estímulos	102
22. Sección para diseñar test de reacción pupilar	102
23. Sección para diseñar test de fijaciones	103
24. Botonería auxiliar	103
25. Cuadro de diálogo para iteración de elementos	104
26. Cuadro de diálogo para iteración de elementos	104
27. Formas definidas para el estímulo	111
28. Cambios en la escala y rotación del estímulo	113
29. Secciones iniciales de la configuración de pruebas	114
30. Árbol de archivos del proyecto	117
31. Cambios de saturación en el estímulo	119
32. Variación en el numero de muestreos del efecto de difuminación en el estímulo	120
33. Animación de parpadeo en el estímulo	123
34. Animación de contracción en el estímulo	124
35. Animación de giro en el estímulo	124

36.	Animación de pivote en el estímulo	125
37.	Animación del parpadeo en el fondo de las pruebas	126
38.	Sección de controles para configuración de animaciones	126
39.	Algunas trayectorias definidas para el desplazamiento del estímulo	127
40.	Sección de parámetros para test de movimientos sacádicos	128
41.	Elementos complementarios en test de movimientos sacádicos	129
42.	Graficet del funcionamiento de previsualización	131
43.	Controles de la animación de trayectorias	133
44.	Espacio definido en la interfaz de configuración para el desplazamiento del estímulo	134
45.	Tabla para visualización de parámetros configurados y controles para , restablecer y exportar datos	135
46.	Elementos complementarios para test de movimientos sacádicos en pruebas combinadas	136
47.	Ejemplo de relación entre elementos de una prueba	137
48.	Cuadro de diálogo indicando que la tabla se encuentra vacía	138
49.	Cuadro de diálogo para asignar nombre al fichero que se dispone a contener la exportación de los parámetros configurados	139
50.	Opciones de previsualización	141

Índice de tablas

2.1.	Anomalías del movimiento ocular en trastornos neurodegenerativos [22]	12
3.1.	Número estimado de personas que padecen enfermedades neurológicas	13
3.2.	Número estimado de personas que fallecen a causa de enfermedades neurológicas	14
3.3.	Principales características de los software de configuración de pruebas para investigación en neurología	32
4.1.	Historia de usuario para requerimiento sobre los test disponibles en la herramienta	42
4.2.	Historia de usuario para requerimiento sobre la definición de formas e imágenes para el diseño de estímulos	42
4.3.	Historia de usuario para requerimiento sobre inclusión de paradigmas <i>Gap y Overlap</i>	43
4.4.	Historia de usuario para requerimiento sobre definición de escala y rotación para el diseño de estímulos	43
4.5.	Historia de usuario para requerimiento sobre selección de colores y efectos gráficos en el diseño de estímulos	44
4.6.	Historia de usuario para requerimiento sobre combinación o integración de test en una prueba	44
4.7.	Historia de usuario para requerimiento sobre iteración de elementos en una prueba, para repetir una sección de la misma	45
4.8.	Historia de usuario para requerimiento sobre implementación de animaciones en el diseño de estímulos	45
4.9.	Historia de usuario para requerimiento sobre aleatoriedad en la presentación de estímulos de una prueba	46
4.10.	Historia de usuario para requerimiento sobre implementación de trayectorias en el diseño de estímulos	46
4.11.	Historia de usuario para requerimiento sobre previsualización de pruebas	47
4.12.	Historia de usuario para requerimiento sobre exportación de datos desde la interfaz hacia archivos XML externos	47
1.	Ciclos definidos para el desarrollo del proyecto, clasificados por requerimientos	109
2.	<i>Widgets</i> utilizados para el desarrollo de la interfaz	110

Capítulo 1

Introducción

Actualmente las enfermedades neurodegenerativas representan un problema latente que afecta a muchas personas en el mundo [23] y debe ser atendido con inmediatez, pero por diferentes motivos, tanto humanos, sociales y económicos no se han tratado correctamente [24]. Por otra parte, uno de los mayores problemas existentes es la carencia de pruebas específicas que permitan identificar efectivamente trastornos neurológicos [25], generalmente métodos invasivos y costosos, razón por la cual las investigaciones de los últimos años han impulsado la generación de nuevas técnicas de diagnóstico a través de diferentes herramientas con el objetivo de reducir tiempos, costos y aumentar la fiabilidad de los resultados.

1.1. Motivación

La investigación sobre nuevos métodos de diagnóstico ha llevado a los científicos y especialistas del tema, asociar dichas patologías con alteraciones en los movimientos del ojo [22], gracias a esto se han implementados nuevas técnicas que ayudan al diagnóstico de estas enfermedades. Nuestro trabajo nace como un aporte para el dispositivo *Oscann desk100*, creado por la empresa *AURA Innovative Robotics*, diseñado para la asistencia al especialista médico o investigador, en el diagnóstico de enfermedades neurológicas a partir del análisis de movimientos oculares.

Esta plataforma permite programar una serie de test para conformar una prueba estándar a la que se someten los pacientes, la limitación del sistema es que existen pocos parámetros que se pueden modificar para personalizar una prueba. La motivación de este trabajo surge con la necesidad de brindarle al especialista la posibilidad de configurar y diseñar pruebas con características diferentes, dependiendo de las necesidades del paciente.

El presente trabajo se desarrolló durante una estancia de investigación de 6 meses en la ciudad de Madrid, España, gracias al convenio con la Universidad Politécnica de Madrid (UPM). La estancia se realizó en la empresa *AURA Innovative Robotics*, nacida como un spinn-off del Centro de Automática y Robótica del alma máter, y en la cual se desarrollan proyectos de investigación aplicado a diferentes sectores e industria.

1.2. Objetivos

El objetivo general es de este trabajo es proponer una interfaz que le permita al especialista médico diseñar y configurar diversas pruebas para el apoyo en el diagnóstico enfermedades neurológicas en los pacientes. Como objetivos específicos tenemos los siguientes:

1. Diseñar una interfaz en QT/QML para la investigación y el diagnóstico neurológico.
2. Incluir en la interfaz diseñada herramientas para el diseño de diversas pruebas y su configuración por parte del especialista médico.
3. Validar el sistema diseñado a través de la experiencia de usuario.

1.3. Estructura del documento

A continuación, y para facilitar la comprensión de esta monografía, se detalla el contenido de cada capítulo.

- **Capítulo 1. Introducción**
Una breve descripción del contenido de este trabajo y los objetivos que se buscan alcanzar.
- **Capítulo 2. Conceptos Generales**
En este capítulo se describen los conceptos necesarios para la comprensión del escrito, correspondientes a la fisiología del ojo, movimientos oculares y su relación con las enfermedades neurológicas.
- **Capítulo 3. Estado del Arte**
Sección donde se describe el estado actual de las tecnologías para diagnóstico e investigación neurológica, además de características y datos relevantes de las patologías que aquí se abordan, al igual que de las pruebas y diagnósticos existentes para las mismas.
- **Capítulo 4. Arquitectura Software**
En este ítem se aborda la metodología del desarrollo de la interfaz y la descripción de los requerimientos necesarios para la satisfacción del usuario. Además, se muestra la interfaz generada y la descripción de su funcionamiento.
- **Capítulo 5. Resultados**
En esta sección se discute sobre los resultados obtenidos a través de la experiencia de usuario, además, se presentan los archivos que genera la interfaz para la comunicación con la plataforma utilizada.
- **Capítulo 6. Conclusiones**
Por último, en este apartado se plantean las conclusiones a partir de los resultados y se plasman algunas ideas para trabajos futuros.

Capítulo 2

Conceptos Generales

Para efectos en la comprensión del trabajo desarrollado, a continuación, se describen detalladamente conceptos relacionados con el funcionamiento del ojo, sus movimientos y como este órgano puede manifestar síntomas de enfermedades que no son propias de ellos o de la visión. Por otra parte, se describe brevemente el funcionamiento y características de la plataforma *Oscann desk100*.

2.1. Anatomía del ojo humano

El ojo es el órgano receptor de la visión y es el encargado de transformar la luz en señales eléctricas. Estas señales son procesadas por el cerebro, más exactamente por la corteza visual ubicada en el lóbulo occipital, a través de un complejo proceso neuronal; gracias a este proceso somos capaces de distinguir formas, percibir colores e identificar distancias y movimientos [26]. En la figura 2.1 se puede observar un esquema básico de la anatomía del ojo humano [27]. La estructura del globo ocular está formada por tres capas principales, la más externa que la conforman la córnea y la esclerótica (membrana de color blanco), esta última protege los elementos internos del ojo y le da forma al mismo. Sobre la esclerótica también se encuentra la conjuntiva, membrana que recubre la porción del ojo que coloquialmente se le llama "*parte blanca del ojo*" o "*blanco del ojo*" y también la superficie anterior de los párpados. En la capa media, denominada úvea, se encuentran el coroides, el cuerpo ciliar y el iris, es la capa vascular del ojo y además altamente pigmentada, dando así la coloración del iris y cumple la función de evitar rebotes de luz dentro del ojo gracias al coroides. La capa interna la conforman la retina, tejido que se encarga de transformar la luz en impulsos nerviosos que se envían al cerebro a través del nervio óptico. Al interior del ojo y en contacto con la retina, se encuentra el humor vítreo, un líquido que ayuda a mantener la forma del globo ocular. En promedio el diámetro del ojo es de 24 mm, mientras que el diámetro anteroposterior (desde la córnea hasta el polo posterior del ojo) es de 25 mm debido a la curvatura de la córnea [27][28].

2.2. Funcionamiento del ojo humano

El primer contacto con los estímulos lumínicos lo realiza la córnea, que es la parte más frontal del globo ocular, esta se encarga de refractar la luz procedente del entorno y

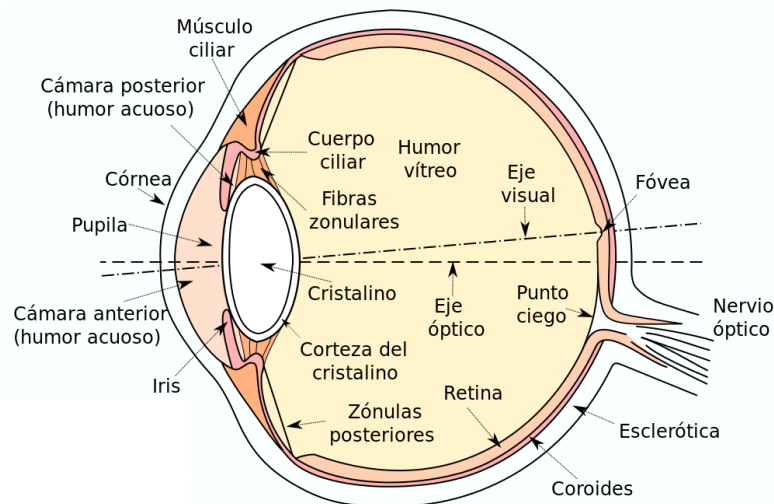


Figura 2.1: Anatomía del ojo humano

representa dos terceras partes de la capacidad de refracción del ojo [29][30]. El ajuste de la refracción la realiza el cristalino, proceso conocido como acomodación. El cristalino es un lente biconvexo capaz de cambiar su grosor para modificar su curvatura funcional y así redirigir la luz hacia la retina, este procedimiento se realiza para enfocar objetivos a diferentes distancias. Entre la córnea y el cristalino se encuentra el iris, una membrana circular que controla la cantidad de luz que ingresa al ojo a través de la pupila (orificio delimitado por el iris), si la intensidad de la luz es alta, la pupila se contrae, por el contrario, si la cantidad de luz es baja, la pupila se dilata permitiendo un mayor paso de luz. La luz que se proyecta sobre la retina da origen a reacciones químicas y eléctricas que se traducen en estímulos neuronales enviados a la corteza visual del cerebro [28]. En la parte posterior de la retina se encuentra la macula, zona donde la membrana es más sensible a la luz, en el centro de esta hay una hendidura llamada fóvea, esta depresión contiene la mayor cantidad de células fotosensibles y es responsable de nuestra visión aguda y detallada. Como se puede observar en la figura 2.1, la fóvea no se encuentra alineada con el eje óptico del ojo (definido por el centro de la pupila), sino con el eje de la vista, por lo tanto, lo que determina la fijación visual de una persona está definido por el eje de la vista y no el eje óptico, como normalmente se cree [31][27].

2.3. Movimientos Oculares

Fisiológicamente, los movimientos oculares son de naturaleza binocular, es decir, nunca los movimientos de un ojo son independientes del otro. Los movimientos de un ojo, en estado normal de salud, guardan congruencia con el otro, a esto se le denomina correspondencia motora, sin embargo, el estudio de movimientos monoculares es de gran importancia para comprender el funcionamiento de la visión [32].

Estos movimientos se conocen como ducciones y se estudian para comprender los movimientos fundamentales del ojo, provocados por la acción de la musculatura extrínseca del ojo. Como se puede observar en la Figura 2.2, el ojo tiene seis músculos efectores (siete si se considera el elevador del párpado), lo que se traduce en seis grados de libertad pa-

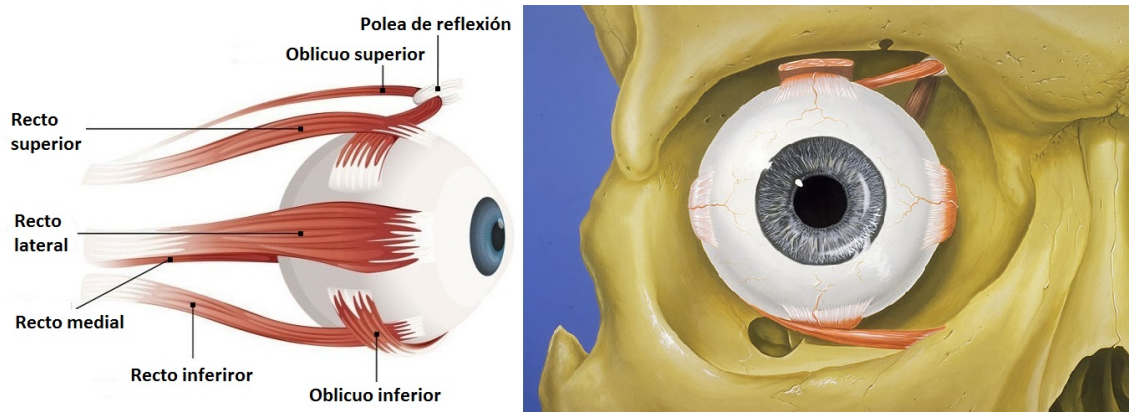


Figura 2.2: Musculatura extrínseca del ojo

ra realizar movimientos verticales, horizontales, oblicuos y torsiones [27]. El sistema de referencia de estos movimientos está definido por los ejes de *Fick* (Figura 2.3), descritos por el médico alemán *Adolf Eugen Fick* [32], que planteó tres ejes perpendiculares entre sí que convergen en el centro de rotación del ojo. El punto de rotación se encuentra sobre el eje anteroposterior del ojo, ubicado a 10 mm del polo posterior del globo ocular y a 13,5 mm de la córnea y está determinado por la acción de las fuerzas del aparato motor del ojo (Figura 2.4). El plano frontal que conforman los ejes *X* y *Z* de *Fick*, se conoce como el plano de *Listing*, este plano pasa por el centro de rotación del ojo dividiendo en dos hemisferios el globo acular, la parte posterior y anterior. Las ducciones se pueden definir como el desplazamiento que realiza el segmento anterior del ojo en torno a los ejes de *Fick* y alrededor de cualquiera de los ejes oblicuos situados en el plano de *Listing*. Los movimientos que describe cada eje son los siguientes,

- Eje horizontal o de cabeceo (*X*), permite la descripción de los movimientos verticales del ojo, elevación o descenso del ojo (arriba o abajo respectivamente). En la elevación del ojo entran en acción el recto superior y el oblicuo inferior y para realizar el descenso actúan el recto inferior y el oblicuo inferior.
- Eje vertical o de viraje (*Z*), describe los movimientos horizontales del ojo (derecha o izquierda), cuando el movimiento es hacia el interior (nariz), se denomina aducción, mientras que si es hacia el exterior (sien), se llama abducción. Los músculos que permiten la aducción son el recto medio y los dos rectos verticales, mientras que en la abducción están involucrados el recto lateral y los dos oblicuos.
- Eje anteroposterior o de torsión (*Y*), posibilita los movimientos rotacionales del globo ocular llamados ciclo-ducción o ciclo-rotación. Teniendo como referencia la parte superior de eje vertical de la córnea, si el movimiento es hacia el interior (dirección nasal), se le conoce como intorsión o ciclo-rotación interna y están a cargo del oblicuo superior y el recto superior. Por el contrario, cuando el movimiento es hacia afuera (dirección temporal o hacia la sien) se le llama ciclo-rotación externa o extorsión y son facultados por el oblicuo inferior y recto inferior.

Ahora bien, los movimientos en los que nos centraremos son los de tipo binocular, debido a que las investigaciones que se realizan en el ámbito neurológico se basan en

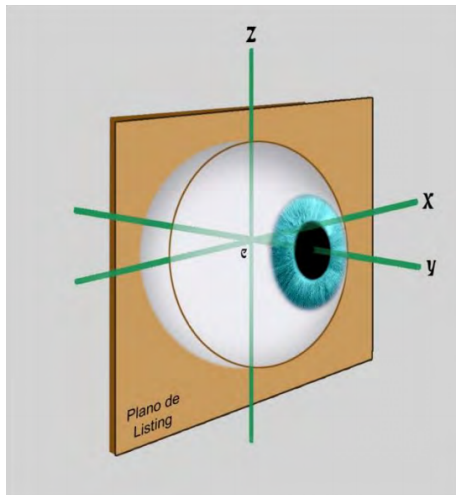


Figura 2.3: Ejes de Fick y plano de Listing

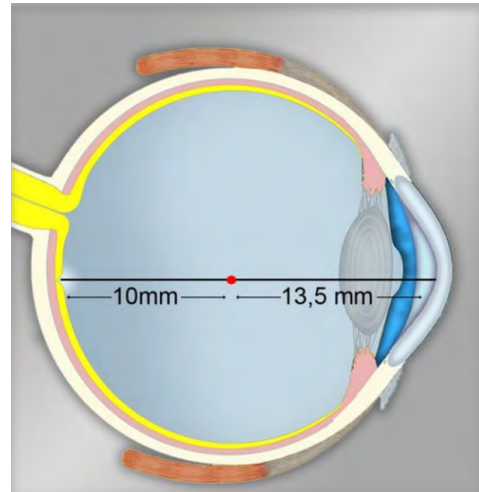


Figura 2.4: Centro de rotación del ojo

su gran mayoría en dichos desplazamientos. Como se mencionaba al inicio de esta sección, fisiológicamente los movimientos del ojo son de tipo binocular y pueden ser tanto reflexivos ó automáticos (involuntarios) como voluntarios. También se puede realizar una clasificación en función del tipo de desplazamiento, cuando están asociados a dirección se conocen como versiones, en los cuales los ejes de visión se mueven en el mismo sentido y sobre un plano frontal a distancia finita. Mientras que los desplazamientos asociados a distancias se denominan vergencias y su cometido es ajustar los ejes de visión cuando la fijación se realiza sobre un objeto que se encuentra a diferentes distancias de cada ojo, es decir, no hay un plano frontal sobre el cual se realiza la fijación porque el objeto u objetivo de la mirada, se ubica a una profundidad diferente con respecto a cada ojo. En las vergencias, los ejes de visión se mueven en sentido opuesto, en pocas palabras, mientras un ojo se mueve hacia la derecha, otro lo hace hacia la izquierda, con el fin de fijar el estímulo visual que se encuentra en un plano distinto para cada ojo. A continuación, se hace una clasificación más detallada según su accionamiento, movimientos voluntarios e involuntarios.

2.3.1. Movimientos voluntarios

El principal objetivo de estos movimientos es el de ubicar la mirada en un punto deseado (desplazamiento de la mirada), se pueden dividir en movimientos para búsqueda de objetivos y de seguimiento o persecución de los mismos.

Movimientos sacádicos o sacadas

Son rápidos desplazamientos voluntarios (también puede ser reflexivos) que realiza el ojo para inspeccionar continuamente el entorno. Son los movimientos que se realizan entre fijaciones, pequeños saltos entre dos puntos de interés que permiten tener una experiencia de una imagen continua en todo momento. El registro de una escena o del espacio que nos rodea, se realiza a través de movimientos secuenciales de este tipo, por ende, son los movimientos más frecuentes que ejecutan nuestros globos oculares. Se caracterizan

por tener una amplitud¹ máxima de 30° , la velocidad es directamente proporcional a la amplitud y su valor puede alcanzar los $(600 - 700)^\circ/s$ para amplitudes cercanas al valor máximo. La duración de estos movimientos oscila entre 30 a 120 milisegundos [33][32]. Las sacadas se pueden clasificar en prosacadas y antisacadas, las primeras son cuando el movimiento se realiza hacia el estímulo o punto de interés, mientras que las antisacadas se alejan de este.

Movimientos de seguimiento o persecución lenta

Son movimientos uniformes lentos que buscan perseguir un objeto o punto móvil del entorno, estos estabilizan la imagen que se proyecta sobre la fovea permitiendo al individuo una experiencia de visión continua de dicho objetivo. También se pueden realizar sobre un objeto fijo mientras la persona se desplaza manteniendo la mirada en el objeto o rotando únicamente la cabeza. La velocidad de los objetivos en el espacio no debe superar los $45^\circ/s$ [33], de lo contrario pasaran a ser movimientos sacádicos los que intenten seguir el movimiento del estímulo visual.

Vergencia

Como mencionamos anteriormente, son movimientos de persecución en los cuales los globos oculares se mueven en direcciones opuestas para conseguir la fijación de un objeto situado a diferentes distancias de cada ojo. La función de estas es proyectar la imagen en la fovea de cada ojo de forma sincronizada para experimentar una imagen unificada del objetivo de fijación. Estos movimientos pueden alcanzar velocidades de unos $10^\circ/s$ y amplitudes de 15° [32].

Fijación ocular

Ocurre cuando se establece la mirada sobre un objetivo en específico, sin realizar ningún movimiento² aparente [33], con el fin de dirigir la luz de dicho objetivo a la zona más sensible de la retina, la fovea, encargada de la visión aguda o detallada del sentido.

2.3.2. Movimientos involuntarios

La gran mayoría de estos movimientos tienen como función compensar o complementar los movimientos oculares de acción voluntaria. Se pueden diferenciar en dos categorías, movimientos compensatorios relativos al movimiento (bien sea del entorno o del cuerpo humano) y movimientos de compensación relativos a la fijación, conocidos como micromovimientos de fijación.

¹Hace referencia al tamaño del salto que debe hacer el ojo entre los puntos de fijación, su magnitud son los grados o minutos de arco.

²En realidad si hay movimientos durante una fijación pero son casi imperceptibles, de amplitudes muy pequeñas, dichos movimientos son abordados más adelante.

Reflejo vestibulo-oculares

Estos movimientos compensan la fijación a causa de la rotación de la cabeza o movimientos del cuello (tónico-cervicales), su despertar reflexivo es consecuencia del sistema vestibular que reacciona ante rotación o traslación cefálica (de la cabeza), enviando señales que se traducen en estímulos para los músculos del ojo, desplazando el globo ocular en el sentido contrario al movimiento de la cabeza [33]. Su función en el sentido de la visión, es mantener la luz del objetivo de fijación en la fovea de cada ojo.

Reflejo optocinético

Su función es estabilizar la mirada cuando el punto u objeto que se quiere fijar se encuentra en un campo visual en movimiento, es decir, mientras el objetivo de fijación puede estar estático los estímulos del campo periférico de la mirada se encuentran en movimiento [32]. El reflejo optocinético y el vestibulo-ocular se complementan entre sí, mientras esta sociedad funcione correctamente permiten que una persona pueda ver y moverse al mismo tiempo.

Fluctuaciones o derivas

Las fluctuaciones o derivas son pequeños, suaves e irregulares desplazamientos aleatorios del ojo en torno al punto de fijación. Se caracterizan por tener una velocidad promedio de $0,2^\circ/s$, una duración de entre 3 a 8 segundos y una amplitud de 2 a 5 minutos de arco. Su función es compensar los movimientos del ojo que no logran corregir las microsacadas o cuando hay ausencia de ellas, para mantener la posición del ojo en el punto de fijación [34] [33].

Microsacadas

Son movimientos rápidos de baja amplitud que buscan corregir la posición de la imagen retiniana cuando se ha alejado del centro de la fovea a consecuencia de las fluctuaciones. En síntesis, fluctuaciones y microsacadas se compensan entre sí para corregir la posición durante una fijación. Se caracterizan por tener una amplitud menor a 1° , una velocidad de 28 a 97 grados por segundo y una duración de 10 a 30 milisegundos [34] [35].

Temblor o tremor

Es un movimiento oscilatorio y constante de alta frecuencia, en promedio 80Hz y de amplitud muy baja de 24 segundos de arco ($< 0,017^\circ$), siendo el desplazamiento con menor amplitud de los movimientos oculares [34][33].

La figura 2.5 [1], representa el punto de fijación que se efectúa sobre la fovea y los diferentes micromovimientos que suceden durante este proceso. Los micromovimientos de fijación (fluctuaciones, microsacadas y temblores), en conjunto, evitan el desvanecimiento de la imagen sobre la retina. Este fenómeno se conoce como *fading* o desaparición de la imagen, antiguamente se pensaba que durante la fijación ocular dichos micromovimientos reducían la agudeza visual, por el contrario, se comprobó que en un estado de inmovilidad

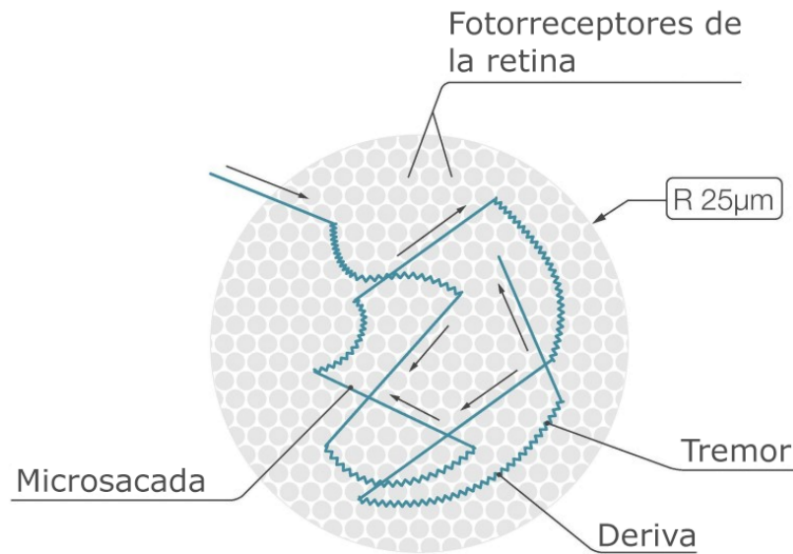


Figura 2.5: Micromovimientos involuntarios durante la fijación [1]

ocular, la imagen se desvanece debido a la saturación de los foto-receptores, los cuales tienen un gradiente de excitación, por lo tanto, si la luz que incide en estas células no se actualiza constantemente, se traduce en un desvanecimiento o desaparición de la imagen [36].

2.4. Alteraciones vinculadas a las patologías neurodegenerativas

Gracias a resultados de investigaciones publicadas en la literatura mundial, se conoce que algunas afecciones de carácter neurodegenerativas afectan la ejecución de movimientos oculares. Este hecho ha permitido que los especialistas se enfoquen en otro tipo de pruebas para investigar estas enfermedades, considerando los movimientos oculares como un marcador biológico para estos trastornos [37]. Movimientos específicos como las prosacadas, antisacadas, persecución y fijaciones, se encuentran entre los movimientos más analizados y de los cuales se encuentra mayor información en la literatura.

Enfermedades como el Parkinson, Alzheimer, epilepsia, esquizofrenia, encefalopatía hepática (cirrosis), síndrome de alcoholismo fetal, enfermedad de Huntington, entre otras, son afecciones que afectan en gran medida los procesos cognitivos y funciones motoras, esto se ve reflejado en algunas ocasiones en los movimientos oculares, permitiendo identificar comportamientos o desplazamientos anormales del globo ocular durante una prueba como las desarrolladas en este trabajo.

Algunas conclusiones de los resultados publicados concuerdan que existen comportamientos característicos de ciertos pacientes diagnosticados con estas patologías mencionadas anteriormente, como, por ejemplo:

- En estudios con pacientes esquizofrénicos se pueden encontrar varias conclusiones sobre distintas pruebas a las que se someten, por ejemplo, en pruebas de persecución lenta se ha identificado que sus movimientos no son suaves, sino que presentan la

aparición de saltos, en pruebas con movimientos antisacádicos se han identificado valores de latencia³ significativamente superiores [38], también se ha encontrado que en la mayoría de las ocasiones, los pacientes no pueden evitar mirar hacia el lado donde aparece el estímulo, para luego hacer la antisacada, lo que incrementa aún más el tiempo de reacción del movimiento antisacádico.[39]

- En pacientes con la enfermedad de Parkinson (EP), se ha observado un desempeño deficiente en algunas pruebas, por ejemplo en antisacadas, los pacientes realizan las tareas de manera más lenta frente a los controles y además cometen más errores, por otra parte, se han realizado estudios con pacientes de EP tanto moderada como avanzada y se encontró que el desempeño en pruebas con prosacadas de los pacientes con EP avanzada, es más pobre o ineficiente que el de los pacientes con EP moderada [40]. Otro aspecto importante que vale mencionar es que los pacientes tienen problemas para inhibir o reprimir las sacadas reflexivas, es decir que, durante las pruebas de sacadas visualmente guiadas, el paciente no tiene control total de sus movimientos oculares ante un estímulo nuevo, demostrando que hay un deterioro en los procesos neuronales que se encargan de realizar los movimientos sacádicos [41][42].
- En referencia a la encefalopatía hepática (a causa de la cirrosis), se encuentran resultados que describen anomalías en la persecución o seguimiento lento de estímulos, reflejando un deterioro tanto de la capacidad de seguimiento como de la precisión. El análisis del movimiento ocular durante las pruebas de persecución muestra que dicho seguimiento es muy pausado, es decir realizan interrupciones durante el desarrollo de las pruebas, esto para pacientes con encefalopatía hepática mínima, mientras que para pacientes con encefalopatía hepática manifiesta, se observa interrupción más pronunciada al punto de perder completamente el seguimiento del estímulo visual; además, ambos grupos de pacientes realizan sacadas de corrección o adelantamiento constantemente durante las pruebas [43].

En la tabla 2.1, ubicada al final de este capítulo, se muestran algunas de las enfermedades neurodegenerativas más estudiadas y sus principales características que varían en el grado en que afectan a los procesos motores, cognitivos y conductuales. Para mejorar la comprensión de la información consignada en la tabla, es importante tener claridad sobre los parámetros característicos de una sacada [39], descritos a continuación:

- Amplitud. Determina el tamaño de la sacada o salto. Usualmente se da en grados o minutos de arco.
- Ganancia. Es la relación entre la amplitud que obtiene una sacada y la amplitud esperada (referencia). Si esta relación es menor que 1, se dice que la sacada es hipométrica; mientras que si es mayor que 1, se dice que es hipométrica.
- Duración. Es el tiempo que se demora la persona en la realización de la sacada.
- Velocidad pico. Es la máxima velocidad alcanzada durante la sacada. La velocidad de estos movimientos se encuentra entre los 20°/seg y 700°/seg.

³Es el tiempo transcurrido entre la aparición de un estímulo visual y el inicio de la sacada en respuesta a este; tiene un valor promedio de 200 milisegundos.

2.4. ALTERACIONES VINCULADAS A LAS PATOLOGÍAS NEURODEGENERATIVAS11

- Latencia. Es el tiempo transcurrido entre la aparición de un estímulo visual y el inicio de la sacada en respuesta a este. A diferencia de los otros parámetros, el tiempo de latencia es extremadamente variable; toma valores en el rango de 100 a 350 milisegundos, con un valor promedio de 200 milisegundos.
- Impersistencia. Se define como una existencia transitoria o un acontecimiento que dura poco tiempo. La impersistencia en referencia con la habilidad motora, describe la incapacidad de sostener un movimiento [44].

Enfermedades Neurodegenerativas	Características del movimiento ocular bajo criterio clínico	Caracterización del movimiento ocular en laboratorio
Enfermedades parkinsonianas con cuerpos de Lewy		
Enfermedad de Parkinson	Hipometría leve en sacadas voluntarias hacia arriba. Deterioro leve en movimientos de persecución lenta.	Disminución de la ganancia en movimientos sacádicos, es decir, aumento de sacadas hipométricas.
Demencia por enfermedad de Parkinson	No reportado	Aumento de latencia y disminución en ganancia, en sacadas reflexivas y especialmente en sacadas voluntarias. Incremento de errores en antisacadas.
Demencia con cuerpos de Lewy	Parálisis de la mirada supranuclear en algunos pacientes.	Aumento de latencia en sacadas reflexivas y voluntarias. Reducción en la predicción de sacadas. Incremento de errores en antisacadas. Reducción de sacadas exprés en pruebas con gap.
Enfermedad de Huntington		
Pacientes sintomáticos	Apraxia de las sacadas (dificultad con la iniciación) con o sin movimiento cefálico. Sacadas lentas, especialmente en pacientes de edad temprana. Mirada distraída e impersistente.	Aumento de latencia sacádica. Incremento de la variabilidad en latencia sacádica. Reducción de la velocidad sacádica. Incremento de errores de dirección y de sincronización de antisacadas y sacadas de memoria. Distraición durante pruebas de persecución lenta.
Pacientes pre-sintomáticos	Normal	Aumento de latencia sacádica. Incremento de la variabilidad en latencia sacádica. Incremento de errores en antisacadas y sacadas de memoria.
Demencias		
DFT	No se ha reportado, pero los pacientes que padecen PSP o síndrome corticobasal presentan anomalías en el movimiento ocular características de estos trastornos.	Sacadas reflexivas normales. Aumento de errores antisacádicos (en pacientes con variante conductual DFT y APNF) que se corrigen normalmente. Antisacadas normales (en pacientes con demencia semántica). Aumento en la inestabilidad de la fijación.
Enfermedad de Alzheimer	Deterioro del reflejo visual en una prueba clínicas de movimientos antisacádicos.	Aumento de latencia en sacadas reflexivas y voluntarias. Incremento de errores en antisacadas. Disminución en la corrección de errores antisacádicos.
Abreviaciones: DFT, demencia frontotemporal; APNF, afasia progresiva no fluente; PSP, parálisis supranuclear progresiva.		

Tabla 2.1: Anomalías del movimiento ocular en trastornos neurodegenerativos [22]

Capítulo 3

Estado del arte

En este capítulo se hará una revisión de las tecnologías o plataformas actuales en el contexto de la investigación de enfermedades neurológicas, además, se hace un breve repaso sobre las principales enfermedades de esta clase y los niveles en los que estas patologías afectan a la humanidad. Por otra parte, se describe de manera breve la plataforma *Oscann desk100*, especialmente su parte software, la cual es nuestro mayor interés y motivo de este trabajo.

3.1. Repercusión de las enfermedades neurodegenerativas a nivel mundial

En la actualidad los trastornos neurológicos se han establecido como una preocupación de mayor importancia para la salud a nivel mundial, la cifra de personas que padecen estas enfermedades son alarmantes tal como lo presenta el informe de la Organización Mundial de la Salud OMS (WHO, por sus siglas en inglés) en 2004 [45], donde se realizó un estudio para estimar y proyectar la cantidad de personas afectadas con estas patologías para los años 2015 y 2030, dichas cifras se ilustran a continuación;

	Número	Por cada 1000	Número	Por cada 1000
	2015		2030	
Población	7 103 297 899		7 917 115 397	
Epilepsia	44 568 780	6.27	50 503 933	6.38
Alzheimer y otras demencias	31 318 923	4.41	44 016 718	5.56
Enfermedad de Parkinson	5 967 673	0.84	7 236 712	0.91
Esclerosis múltiple	2 823 092	0.40	3 279 199	0.41
Migraña	364 432 879	51.30	412 894 420	52.15
Accidentes cerebrovasculares	67 212 050	9.46	76 826 249	9.70
Neuro-infecciones	15 714 399	2.21	13 290 180	1.68
Neuropatías nutricionales	321 738 424	45.29	285 369 403	36.04
Lesiones neurológicas	197 627 526	27.82	242 728 912	30.66

Tabla 3.1: Número estimado de personas que padecen enfermedades neurológicas

Aquí se observa un aumento progresivo y considerable en las cifras ya que existen diferentes factores como: el aumento de fármacos que permite una mayor expectativa de

vida, el proceso de envejecimiento de la población (estas enfermedades son más comunes en personas mayores) y el estilo de vida. Cabe destacar que el avance y la incorporación de nuevas tecnologías de diagnóstico, han permitido identificar un mayor número de pacientes en países que pasan por situaciones socio-económicas complejas y anteriormente no tenían acceso a este tipo de servicios, razón por la cual no eran tenidos en cuenta y ahora aportan un alto porcentaje en estas estadísticas.

De vital importancia resulta realizar un diagnóstico oportuno y preciso que permita prevenir o ralentizar este tipo de enfermedades, ya que estas no causan el deceso de las personas a corto plazo, contrario a esto prevalecen una cantidad de tiempo considerable donde se ve reducida la calidad de vida de los pacientes y sus familiares, quienes adoptan un papel importante en estas situaciones. La cantidad de personas que sucumben por estos trastornos al año difiere en gran cantidad del número total de personas afectadas por la enfermedad como se ilustra en la siguiente tabla:

	Muertes	% total	Por cada 100 000	Muertes	% total	Por cada 100 000
	2015			2030		
Población	7 103 297 899			7 917 115 397		
Total DALYs	63 458 962			73 247 767		
Epilepsia	130 569	0.21	1.84	139 276	0.12	1.76
Alzheimer y otras demencias	513 230	0.81	7.23	671 372	0.92	8.48
Enfermedad de Parkinson	127 293	0.20	1.79	165 418	0.23	2.09
Esclerosis múltiple	16 669	0.03	0.23	17 012	0.02	0.21
Migraña	0	0.00	0.00	0	0.00	0.00
Accidentes cerebrovasculares	6 466 232	10.19	91.03	7 787 656	10.63	98.36
Poliomielitis (Polio)	654	0.00	0.01	577	0.00	0.01
Tétano	145 640	0.23	2.05	95 587	0.13	1.21
Meningitis	106 372	0.17	1.50	69 946	0.10	0.88
Encefalitis japonesa	7 282	0.01	0.10	4 318	0.01	0.05
Total	7 513 942	11.84	105.78	8 951 162	12.22	113.06

DALYs: Años de vida ajustados por discapacidad; es una medida de carga de la enfermedad global, expresado como el número de años perdidos debido a enfermedad, discapacidad o muerte prematura.

Tabla 3.2: Número estimado de personas que fallecen a causa de enfermedades neurológicas

Surge aquí la motivación de realizar un apoyo a los especialistas médicos para proveer herramientas de valoración que les permita tener un mayor grado de acierto a la hora de evaluar a sus pacientes, realizando un diagnóstico diferencial que permita tener un tratamiento específico y óptimo para cada una de las variaciones que presentan estos trastornos.

En el esfuerzo de contribuir soluciones para el diagnóstico de enfermedades neurológicas han surgido varias técnicas basadas en el seguimiento ocular (*Eye Tracking*) y en este campo la empresa española AURA Innovative Robotics SL., ha desarrollado la plataforma *Oscann desk100* (ver Figura 3.1 de la siguiente página), dispositivo especializado en el diagnóstico diferencial para el cual ha sido diseñada la interfaz de configuración de pruebas presente en este trabajo.

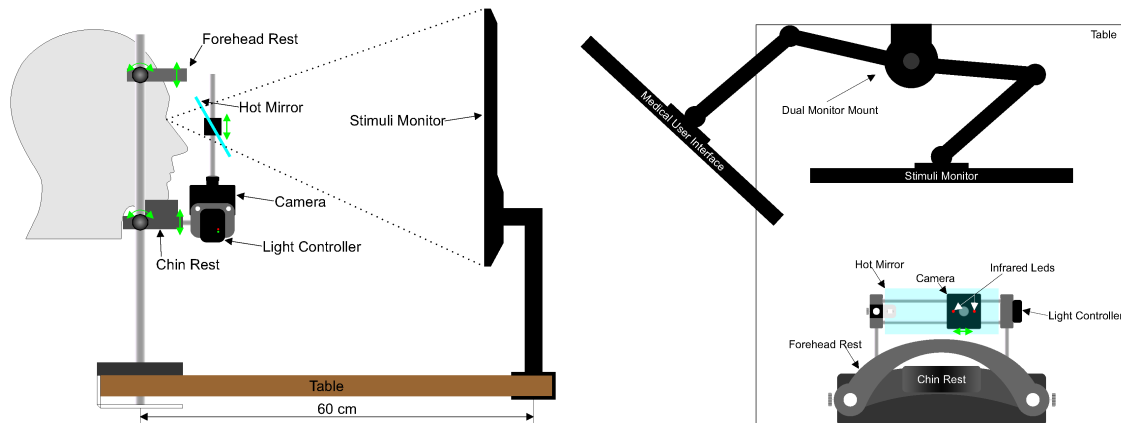


Figura 3.1: Dispositivo *Oscann desk100* [2]

3.2. Tecnología *Eye Tracking*

El seguimiento ocular más conocido como *Eye Tracking* por su denominación en inglés, es definido como el conjunto de tecnologías y técnicas que hacen posible la estimación de la posición del ojo de un individuo para saber hacia dónde se encuentra orientada su mirada en cualquier momento, estas técnicas miden la rotación del ojo dentro de su cavidad [1].

En la actualidad vemos el uso de los *eye tracker* entre otras cosas para realizar estudios en ciencias de la salud, más concretamente en el campo de la oftalmología, neurología y psiquiatría. También son utilizados para estudiar el impacto de la publicidad, estudios de mercadeo, análisis estadístico en el entrenamiento profesional de jugadores de videojuegos y para interactuar con interfaces a través del movimiento ocular. Esto resulta de especial utilidad para las personas con discapacidades físicas [46].

3.2.1. Marco histórico de las técnicas de *Eye Tracking*

El estudio de los movimientos oculares surge en el siglo XIX. Estudios y experimentos eran realizados bajo la observación visual directa, con la ayuda de espejos y lentes. Uno de los estudios más nombrados es el del oftalmólogo francés *Louis Émile Javal* quien en 1879 realizó observaciones de los movimientos oculares de personas durante la lectura [47]. Percatándose de que los ojos no realizaban un barrido suave y continuo por cada línea de texto, al contrario, estos mezclaban movimientos rápidos con cortas pausas, algo que en la actualidad se denomina sacadas y fijaciones.

El primer dispositivo denominado *eye tracker* lo realizó *Edmund Huey* en 1898 [3]. Este consistía de una lentilla anular de escayola, que permitía ver al paciente a la vez que se conectaba a un puntero de aluminio que trazaba los movimientos oculares en un papel de calco. Se indica el esquema del dispositivo en la Figura 3.2. Este sistema era altamente invasivo y requería anestesia local para su empleo, sin embargo, plantó los cimientos de investigación en este campo ya que desde aquel momento se lograron registrar de manera objetiva los movimientos.

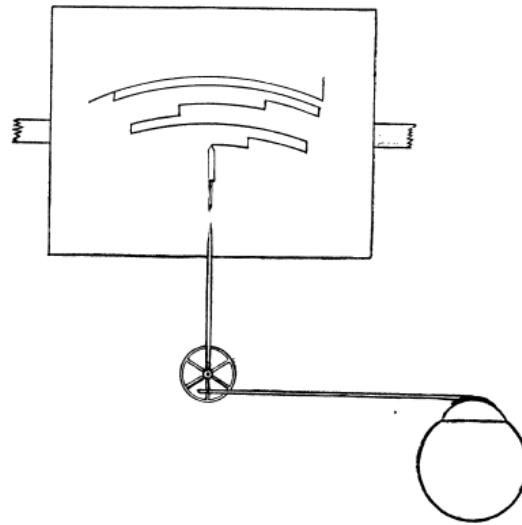


Figura 3.2: *Eye tracker* desarrollado por Edmund Huey [3]

Algunos años después, en 1901 apareció el primer sistema no invasivo de la mano de *Dodge y Cline* [4], donde realizan la introducción de la técnica de reflexión de la luz sobre la protuberancia de la córnea en el globo ocular. Esta técnica consistía en incidir un rayo de luz en el ojo para captar el reflejo en una placa fotovoltaica, la placa se desplazaba verticalmente mediante un sistema neumático y era capaz de captar la componente horizontal de los movimientos del ojo en función del tiempo. Esta configuración presentaba además una mejora considerable en la precisión de la medición frente al sistema anterior y por esta razón se convirtió en una técnica básica que se siguió usando hasta la época de los 70. Diferentes autores modificaron poco a poco el sistema para permitir grabaciones del eje vertical o de los movimientos de la cabeza.

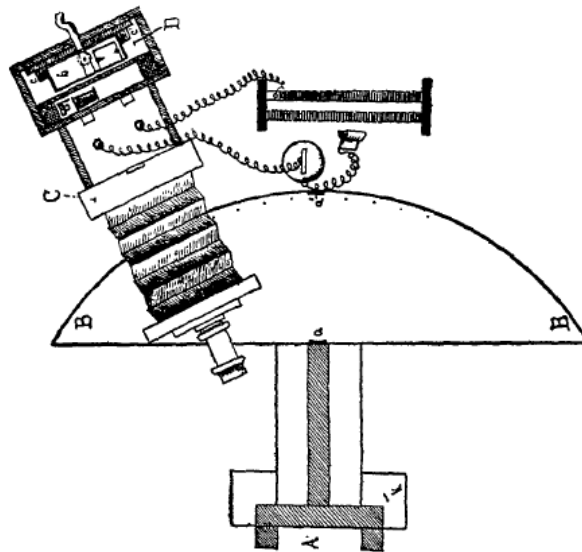


Figura 3.3: Primer *eye tracker* no invasivo desarrollado por Dodge y Cline [4]

Con el transcurso del tiempo estos sistemas fueron evolucionando, en los años 60 surgieron *eye tracker* de naturaleza invasiva como el primero, debido a que se produjo una alta popularidad gracias a la oportunidad de poder fijar elementos al ojo sin necesidad de usar escayola. Este hallazgo permitió la introducción de los sistemas *Coil*, detallados más adelante. Luego en los años 70 se empieza a utilizar los primeros sistemas electrónicos, los cuales incorporan nuevas tecnologías con la capacidad de grabar imágenes del ojo y a partir del procesamiento de estas capturas se detectan características del ojo como por ejemplo el borde del iris o la pupila, para cuantificar los movimientos oculares. Este tipo de sistemas son los que más se han popularizado en la actualidad.

3.2.2. Estado actual de las técnicas de *Eye Tracking*

En este apartado se encuentra un breve repaso de los sistemas de seguimiento ocular más extendidos en la actualidad. Se trata de los sistemas *Coil*, de *electro-oculografía* y *vídeo-oculografía*.

Sistemas Coil

Los sistemas *Coil* o *Scleral search coil* aparecen en los años 60 y su uso ha venido decreciendo considerablemente debido a su naturaleza invasiva. Estos sistemas permiten determinar la posición del ojo a partir de una lentilla de silicona que debe colocarse en el globo ocular del sujeto. Como se puede observar en la Figura 3.4, la lentilla contiene una bobina de alambre de cobre delgado, cuando el sujeto se sitúa en un campo magnético la posición del ojo puede ser determinada a partir de la amplitud de la corriente de inducción de la bobina. La lentilla puede contar con hasta tres conjuntos de bobinas para generar señales que corresponden a movimientos en horizontal, vertical y de torsión. Se necesita mantener fija la cabeza con el fin de que los movimientos registrados por la bobina sean exclusivamente generados por rotaciones en el ojo [1].

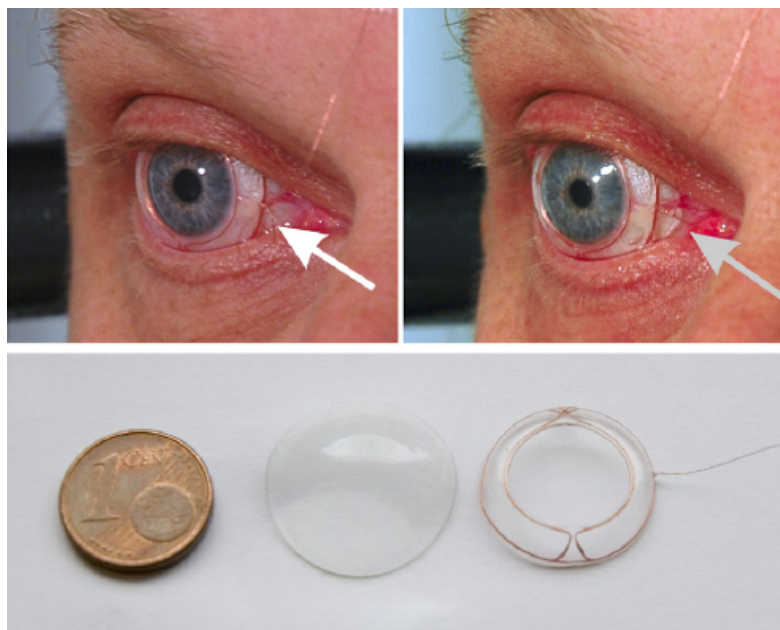


Figura 3.4: *Scleral search coil* [1]

La principal característica de este sistema es proporcionar resultados con niveles bastante bajos de ruido, debido a que permite resoluciones espaciales de 0,1 grados y temporales de un milisegundo [48]. La principal desventaja de este método, como ya se ha dicho, es que tiene una naturaleza invasiva y por lo tanto se puede usar con un tiempo limitado de treinta minutos [49]. Sin embargo, es posible incluir un lente de contacto de tipo vendaje, y así prolongar este tiempo hasta dos horas [50].

Una ventaja en su aplicación es que este dispositivo puede ser utilizado con los ojos cerrados, por lo que se pueden realizar experimentos de imágenes mentales y estudios del sueño entre otros [50].

Basados en el mismo concepto se han desarrollado en los últimos años nuevas tecnologías. En la Figura 3.5, se puede observar un ejemplo, siendo este un sistema denominado *EyeContact* [5], el cual es un dispositivo que usa cinco bobinas y que permite ser usado sin necesidad de fijar la cabeza. Es utilizado generalmente para aplicaciones de realidad virtual y realidad aumentada.



Figura 3.5: *EyeContact* [5]

Electro-oculografía (EOG)

Los sistemas de electro-oculografía permiten la medición de los movimientos oculares basados en el potencial eléctrico que existe entre la córnea y el fondo del ojo, cuyo valor oscila entre 10 y 30 milivoltios [51]. En este lugar se establece un campo eléctrico en los tejidos que rodean el ojo y la rotación de él ojo provoca un movimiento correspondiente al vector del campo. Para su aplicación es necesario el uso de un par de electrodos con el fin de medir ese cambio de potencial producido por movimientos oculares verticales y horizontales. El hecho de utilizar electrodos en contacto con la piel clasifica este método como invasivo, aunque en menor grado que los Coil. Su uso principalmente se da en ambientes médicos y de investigación

La ventaja de estos sistemas de medición es que permiten alcanzar elevadas tasas de muestreo al estar midiendo señales eléctricas analógicas. El hecho de obtener unos voltajes muy pequeños, hace que la amplificación y el filtrado de la señal se torne un poco

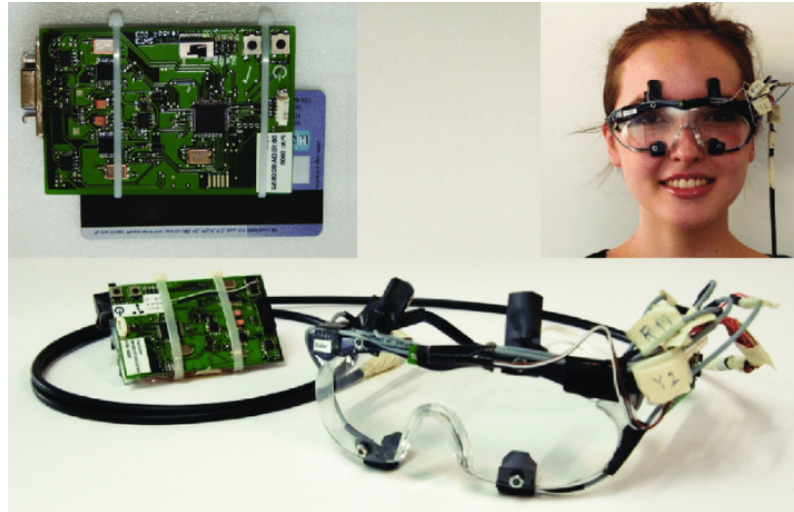


Figura 3.6: Dispositivo EOG [6]

más compleja. Esta dificultad conlleva a tener una baja exactitud junto con el ruido en la señal, producido por factores biológicos que no están relacionados con los movimientos oculares [1].

En la Figura 3.6 se puede ver un dispositivo basado en electro-oculografía [6]. Fue creado con el objetivo de ser un sistema portátil y tiene forma de gafas. Los electrodos se ubican en la estructura de las gafas para adquirir los datos que luego son enviados por Bluetooth a un computador para ser procesados.

Vídeo-oculografía (VOG)

Los sistemas VOG se basan en la grabación de uno o dos ojos simultáneamente por medio de cámaras con el objetivo de registrar los movimientos oculares. Actualmente es el método más extendido tanto en el campo de la investigación como en otros campos debido a su condición no invasiva. Existen dos grupos diferenciados de vídeo oculografía, dispositivos fijos y montados [52].

Dispositivos Montados: los dispositivos de vídeo-oculografía montados son aquellos cuya cámara o cámaras se encuentran sujetas a su estructura, siendo normalmente ésta un casco o unas gafas. También es muy útil para investigaciones con niños ya que permite que los niños se muevan y exploren [53]. La principal ventaja de este tipo de dispositivos es la libertad de movimiento, que permite usarse en actividades cotidianas. Su desventaja es la dificultad de relación entre la posición y orientación de la cabeza respecto a un sistema fijo, ocasionado errores en su medición. En la Figura 3.7 se muestra un dispositivo de esta categoría de la compañía *Ergoneers* [7], este *eye tracker* es capaz de registrar movimientos oculares con gafas graduadas.

Para registrar los movimientos del ojo se emplean varias regiones del mismo que puedan ser detectadas y seguidas. Los más comunes son el limbo (región del ojo que delimita la unión entre la esclerótica y la córnea) y la pupila. Detectar el limbo es relativamente fácil debido al alto contraste entre la esclerótica (blanco del ojo) y el iris. Usar esta técnica para movimientos verticales representa alguna dificultad ya que normalmente el



Figura 3.7: Dispositivo de vídeo-oculografía montado, por *Ergoneers* [7]

párpado cubre parte del iris, dificultando su detección, esto pueden conllevar a errores en la medición. Por otro lado, usar la detección de la pupila es más complejo debido al bajo contraste con el iris, pero en general tiene una precisión más alta ya que el párpado no cubre la pupila [31].

Para hacer más robusta la detección se utiliza el reflejo de la córnea [54]. El reflejo de la córnea ofrece un punto de referencia adicional para así, compensar los pequeños movimientos de la cabeza que pueden darse. Esta ventaja hace que los *eye tracker* basados en vídeo-oculografía que emplean métodos de detección de pupila y del reflejo de la córnea sean los más usados.

La unidad de medida de los datos extraídos de estos dispositivos viene dada en grados visuales, que representan el desplazamiento del ojo en grados respecto a la posición de él cuando este mira a un punto. Esta medida depende del tamaño del campo visual (área del monitor o pantalla donde se proyectan las pruebas) y de la distancia hacia el mismo.

Dispositivos Fijos: son aquellos *eye tracker* que no se mueven junto con la cabeza del individuo. Suelen ir acompañados de un monitor en el cual se muestran escenas de estimulación visual. Este tipo de dispositivos son los más comunes actualmente. Estos *eye tracker* fijos pueden ser remotos o no remotos.

- **Remotos.** Los dispositivos remotos son aquellos en los que la cabeza no permanece sujeta a ninguna estructura. La cámara está alejada del individuo, generalmente se encuentra en la parte inferior del monitor. Estos tipos de dispositivos son más usados en los campos de marketing y control de la interfaz mediante movimientos oculares. Estos dispositivos remotos suelen tener baja precisión, sin embargo, para el tipo de estudios que se realizan con esta técnica se consideran aceptables. En la Figura 3.8 se muestra un modelo de *Tobii EyeX*, el dispositivo más extendido de este tipo. No necesita calibraciones continuas y permite pequeños movimientos de la cabeza [8].



Figura 3.8: Dispositivo *Tobii EyeX* ubicado en la parte inferior de un monitor [8]

- **No remotos.** Los dispositivos no remotos son aquellos que tienen una estructura en la cual el sujeto se posiciona para evitar los movimientos de la cabeza. La cámara está dispuesta cerca del sujeto, fijada a la estructura. La precisión de este tipo de dispositivos es mayor que la anterior debido a la sujeción de la cabeza y la cercanía de la cámara [55]. En la Figura 3.9 se puede observar la disposición descrita, representada por el dispositivo *Oscann desk100*, el cual pertenece a esta categoría.



Figura 3.9: Dispositivo no remoto *Oscann desk100* [9]

3.3. Estado actual en interfaces software

En esta sección se nombran los software de configuración de pruebas para investigación en neurología más relevantes en la actualidad y sus respectivas características.

3.3.1. E-Prime

El software *E-Prime* de *Psychology Software Tools* es una de las herramientas con mayor uso a nivel mundial para diseñar pruebas y evaluar diferentes comportamientos cognitivos y psicológicos [10]. Combina interfaz gráfica con lenguaje de programación personalizada para construir los experimentos, donde los usuarios pueden crear diferentes escenarios a través de imágenes, diapositivas, textos, vídeos y sonidos para realizar la captura de datos con los diferentes *eye tracker* que soporta la aplicación, principalmente dispositivos de manufactura propia (*E-prime*) y otros como los de la compañía sueca *Tobii*.

Existe una extensión del software que profundiza en el área de la investigación neurológica y cognitiva denominada *CAMCI-Research*, en el cual se puede hacer uso de las nueve tareas predeterminadas para evaluar el rendimiento cognitivo mediante pruebas neuro-psicológicas, estándar de memoria, atención y capacidad ejecutiva, además de evaluar memoria incidental (que no se evalúa fácilmente mediante pruebas a lápiz y papel) [11]. Las capturas que se muestran a continuación (Figura 3.11), representan dos pruebas diseñadas en *CAMCI-Research*, y en ellas se puede observar un círculo verde que representa el punto de fijación del individuo, mientras cumple con la tarea que se la ha pedido.

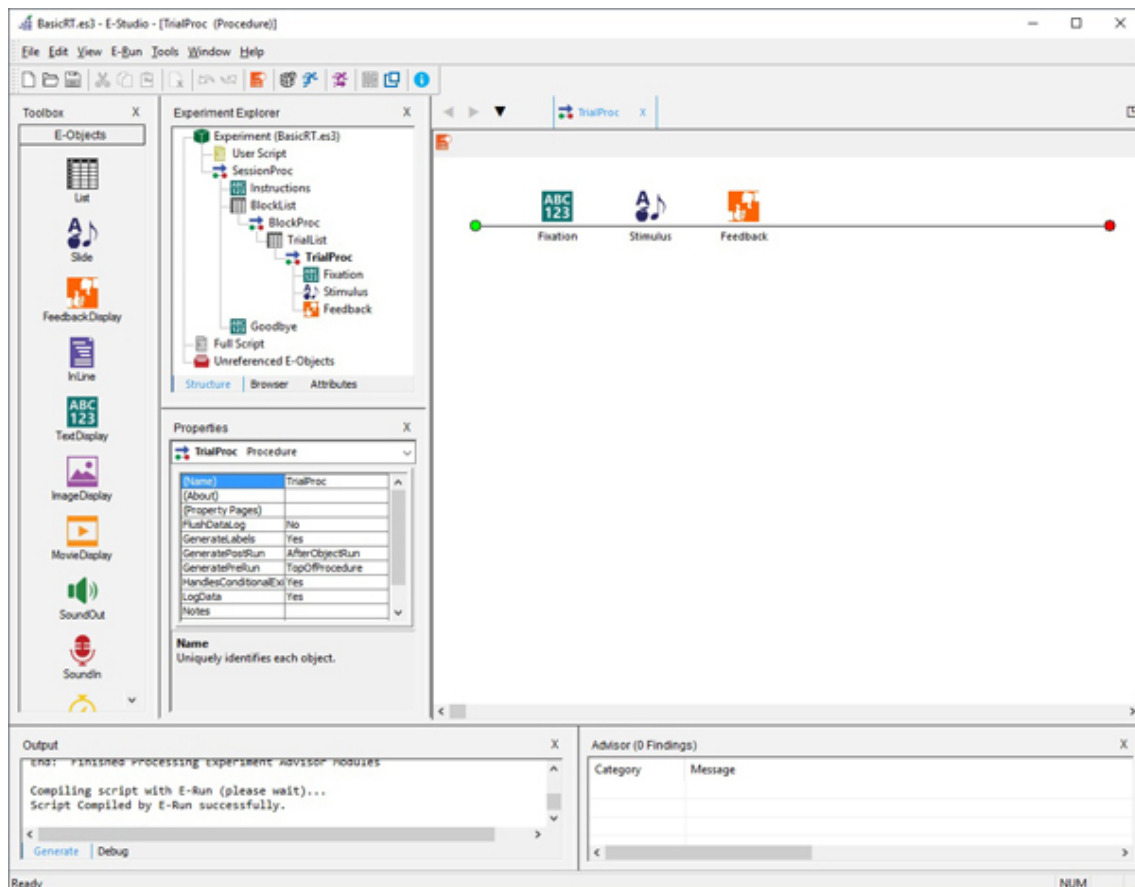


Figura 3.10: Entorno de configuración de pruebas de E-Prime [10].

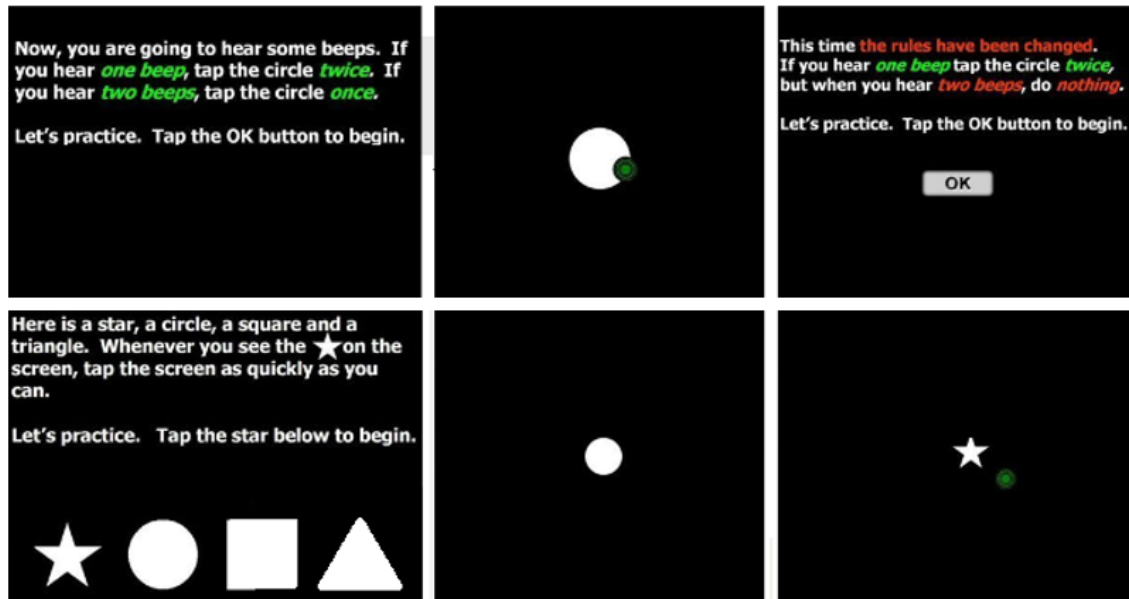
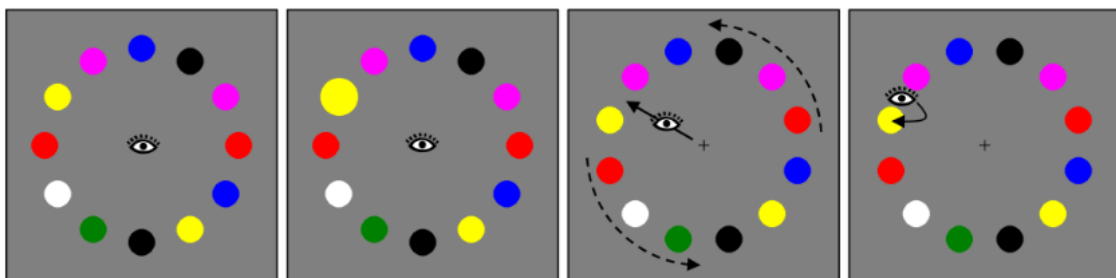


Figura 3.11: Ejemplos de pruebas diseñadas con CAMCI-Research [11].

Con este software se ha realizado el estudio de los movimientos *trans-sacádicos* basados en la memoria visual a corto plazo o VSTM por sus siglas en inglés, utilizando pruebas como la que se ilustra en la Figura 3.12, donde se proyectan círculos de diferentes colores para estimular a los usuarios a realizar movimientos oculares y con los que se demuestra que una función importante del sistema VSTM es establecer la correspondencia del objeto a través de las sacadas y permitir la corrección eficiente de la mirada cuando los ojos no apuntan en el objetivo real de la sacada.

Full-array condition (correction requires memory)



Single-object condition (correction does not require memory)

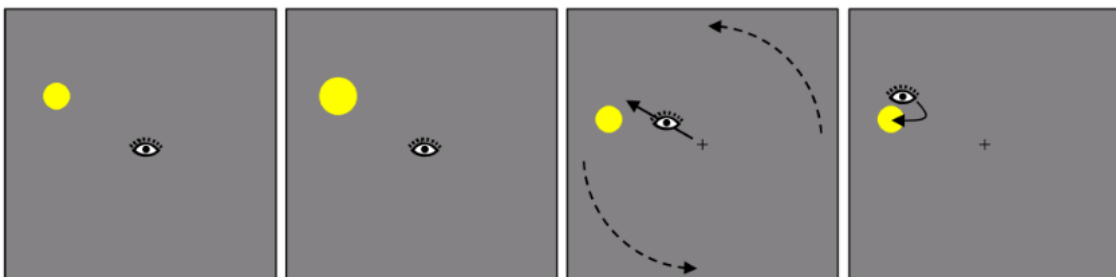


Figura 3.12: Prueba realizada con E-prime para estudio de movimientos trans-sacádicos [12]

3.3.2. PsychoPy

PsychoPy es un paquete de software de código abierto, escrito en secuencias de código Python para el diseño de pruebas de investigación en neurociencia y psicología [13]. Se pueden construir experimentos a partir de secuencias de código en lenguaje Python o a través de una interfaz gráfica basada en OpenGL que genera el código correspondiente, lo cual es de bastante utilidad para usuarios que no tienen conocimientos en lenguajes de programación [56]. A pesar de que su interfaz aún se está perfeccionando, el código es muy estable, compatible con versiones anteriores, lo suficientemente completo y libre de errores, que brinda seguridad en su uso. La independencia de plataformas es un objetivo primordial para *PsychoPy*, la constante actualización de las tecnologías informáticas, en particular de los sistemas operativos, representa un desafío para los desarrolladores, pero en este caso es diferente, los usuarios no necesitan aprender nuevos comandos o hacer uso de otras herramientas sólo porque han decidido cambiar de plataforma, además, dado el caso en que dos personas que trabajen con esta herramienta pero no compartan la misma plataforma, no habrá ningún problema para compartir sus proyectos.

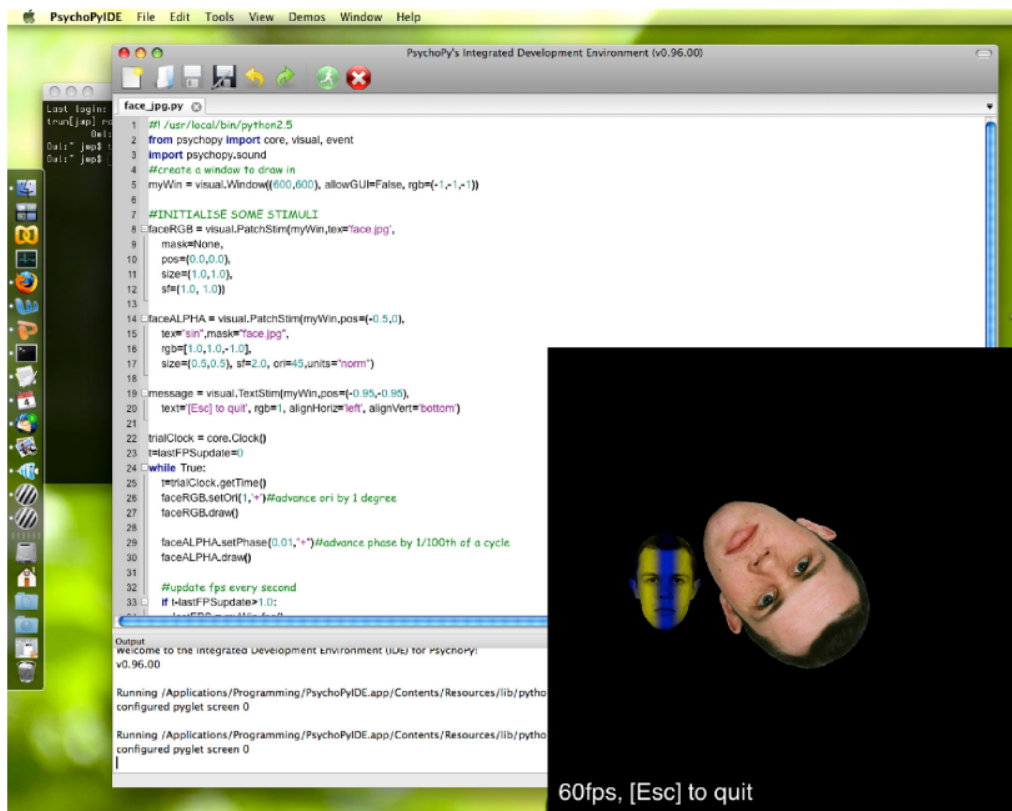


Figura 3.13: Entorno de desarrollo integrado (IDE) ejecutando uno de los scripts de demostración [13]

PsychoPy viene ahora con un editor de código incorporado (PsychoPyIDE), dotado de auto-completado de código, plegado de código y consejos de ayuda. Los scripts se pueden ejecutar directamente desde el editor y la salida o presentación de la prueba (*preview*) se muestra en otra ventana de la aplicación. En la Figura 3.13, se muestra el entorno de desarrollo integrado, se puede observar un *script* en el editor y la salida del mismo. En

el panel de la parte inferior de la ventana de diálogo se muestra la información de que el *script* está en ejecución, así como la ventana que permite visualizar la presentación de la prueba.

3.3.3. Experiment Builder

La compañía canadiense *SR-research* es propietaria del software *Experiment Builder*, en el cual es posible diseñar y configurar experimentos en las áreas de psicología y neurociencia por medio de la visualización de imágenes, formas, textos y vídeos que pueden estar acompañados de audio. Esencialmente cuenta con una interfaz gráfica donde es posible arrastrar y soltar las herramientas necesarias para construir una prueba, aunque también se encuentra la opción de introducir secuencias de código en lenguaje Python para configurar características más avanzadas; para generar una mayor flexibilidad se integra con los *eye tracker* denominados *EyeLink* que son de propia fabricación [14].

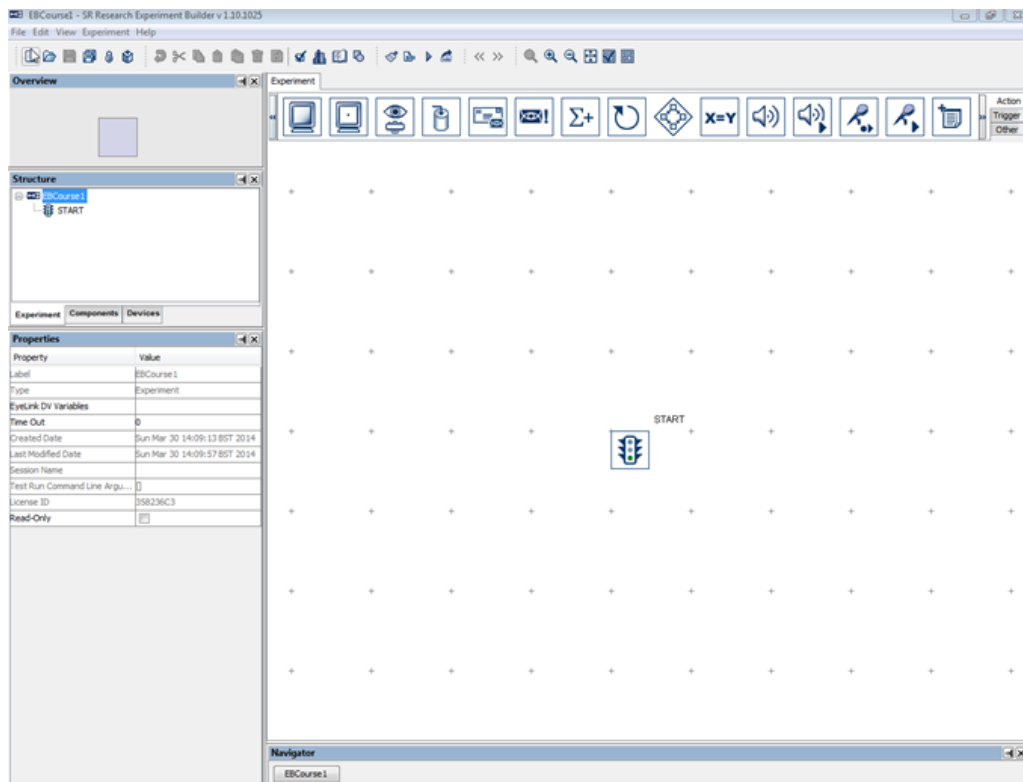


Figura 3.14: Espacio de trabajo de la herramienta Experiment Builder [14]

En la interfaz encontramos algo similar a un diagrama de flujo en el cual se pueden agregar las diferentes características que se han mencionado y dispone de instrucciones para realimentar el sistema a través de equipos hardware como teclados, además de dispositivos especiales de la propia marca o de otras como *Cedrus* y *VPixx* [57]. En la Figura 3.14, se puede apreciar el entorno de configuración de pruebas de la herramienta *Experiment Builder*.

En este software se han realizado diversas investigaciones basadas en la ejecución de pruebas para estimular movimientos sacádicos, por ejemplo, explorando la incidencia de la información textual de alto nivel, en la curvatura de la sacada. La prueba consiste en

presentar un texto en el cual se puedan comparar palabras periféricas irrelevantes versus palabras claves neutrales, los resultados se organizan en un paradigma de señales espaciales que permita medir las desviaciones de la trayectoria. El análisis de estas curvas ha permitido evidenciar que las sacadas presentan mayor desviación cuando los usuarios no conocen la semántica de las palabras periféricas irrelevantes en comparación con las palabras neutrales [58].

3.3.4. DMDX

DMDX es un software gratuito creado en la Universidad de Arizona y programado por *Jonathan Forster*, aunque cabe resaltar que el código fuente del programa no se encuentra disponible. Está basado en Windows y fue diseñado principalmente para realizar experimentos en neurociencia a través del procesamiento de lenguaje [59], se enfoca en la presentación de texto, audio, gráficos y videos para permitir la medición de los tiempos de reacción en las pantallas de un computador con una precisión de milisegundos. Utiliza las rutinas de biblioteca proporcionadas por *DirectX* para ofrecer una sincronización precisa de la salida visual y de audio.

El diseño de las pruebas en este software se realiza únicamente a través de un lenguaje de programación personalizado basado en *MS-DOS*. No obstante en el año 2013 *Pablo Garaizar* y *Ulf-Dietrich Reips* desarrollan una herramienta basada en tecnologías web denominada *Visual DMDX* [15], donde se dispone una interfaz gráfica para configurar los experimentos y luego exportarlos en archivos de extensión *RTF* que contienen las secuencias de código correspondientes al lenguaje personalizado del programa. Esta característica ha fomentado el uso de esta aplicación ya que facilita la implementación de pruebas a usuarios que no tienen conocimientos en programación. En la Figura 3.15, se puede observar el ambiente de configuración de la interfaz correspondiente a *Visual DMDX*.

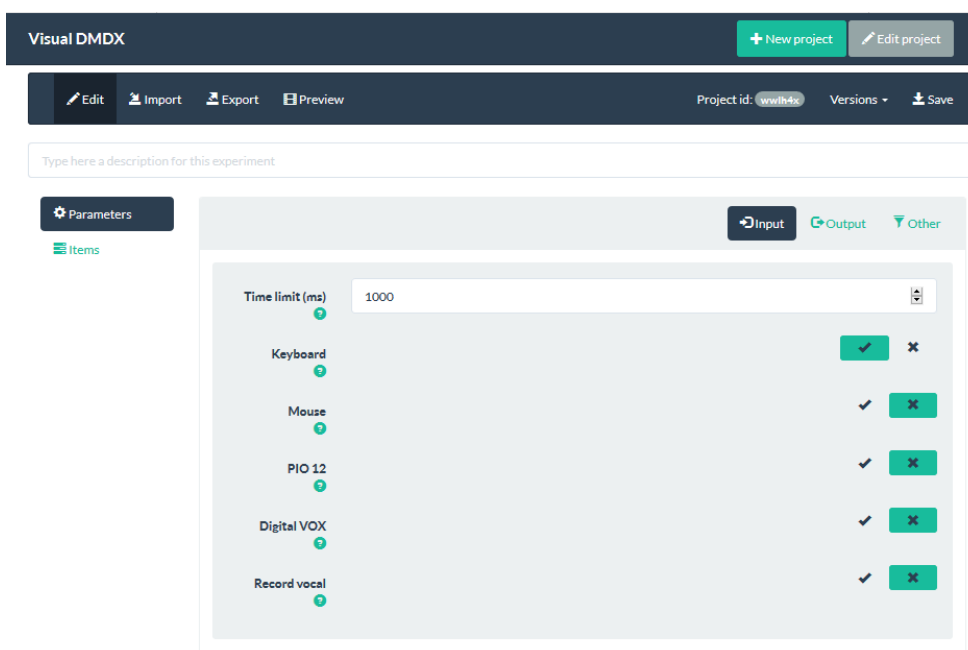


Figura 3.15: Interfaz de configuración de pruebas en Visual DMDX [15]

3.3.5. OpenSesame

Este software está orientado a la construcción de experimentos gráficos en las ciencias sociales. La aplicación es gratuita, de código abierto, con soporte multiplataforma y desarrollada en lenguaje Python, puntualmente basada en las librerías de *PyGame* y *PsychoPy* [60]. El apartado gráfico se despliega a través de OpenGL y en la interfaz de usuario es posible seleccionar y arrastrar formas, textos, imágenes, videos y sonidos para el diseño de las pruebas, además, se encuentran diferentes tipos de operaciones para realimentar el sistema a través del teclado o del puntero del ratón. En *OpenSesame* es posible introducir secuencias de código en lenguaje Python para realizar tareas más complejas, tales como realizar un seguimiento al puntero, o para obtener datos avanzados sobre la actividad de un usuario en una interfaz y así proveer mejoras en la navegabilidad. En la Figura 3.16, se muestra el ambiente de la aplicación *OpenSesame*.

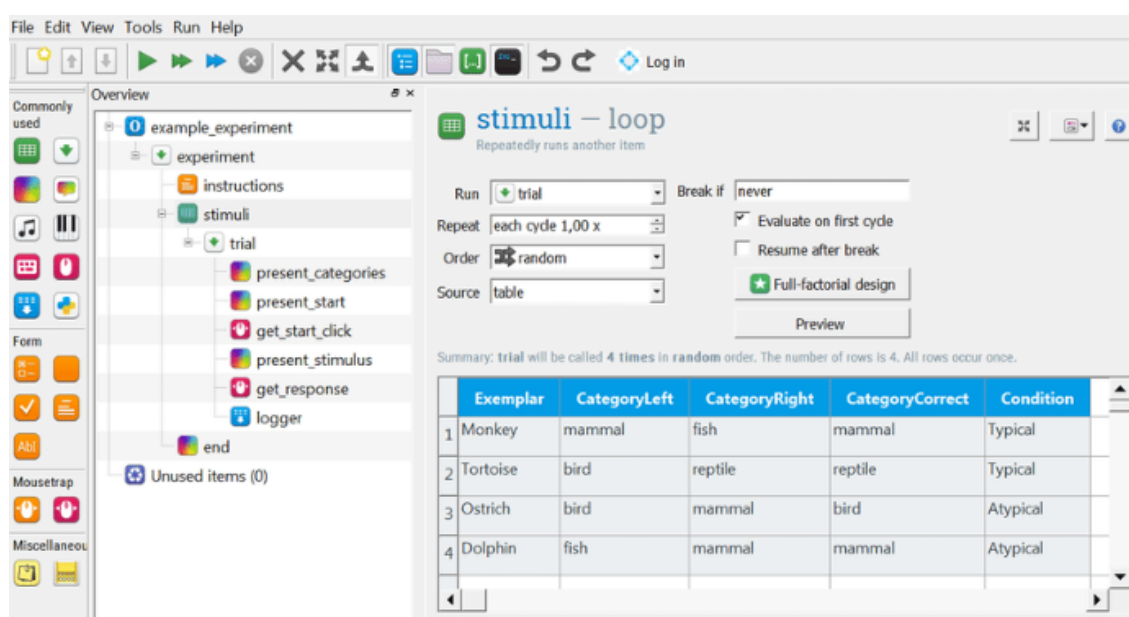


Figura 3.16: Interfaz de OpenSesame [16]

Gracias a la flexibilidad de este software y la integración con diferentes dispositivos de seguimiento ocular como los de la compañía *EyeLink*, se han realizado diversas investigaciones para determinar y analizar diferentes patrones en el sistema oculomotor.

Por ejemplo, en el año 2017 se realizó un estudio sobre el efecto de un objeto distractor sobre la curvatura del curso temporal de una sacada [17]. En los experimentos incluidos para aquella investigación se utilizaron formas circulares como estímulo, con el fin de que el usuario orientara su mirada hacia estos puntos, ejecutando una sacada. Adicionalmente durante el despliegue de estas formas, se incluía un segundo objeto (distractor), tal como se muestra en la Figura 3.17, este punto debía ser ignorado por los sujetos, donde, inicialmente el objeto estaba fijo y después en movimiento, esto dio como resultado que cuando el objeto distractor se encontraba en movimiento la curvatura temporal de la sacada era más prolongada que cuando se encontraba fijo y por tanto se concluye que esto se debe a que el sistema ocular intenta compensar todo el espacio visual al que está sometido.

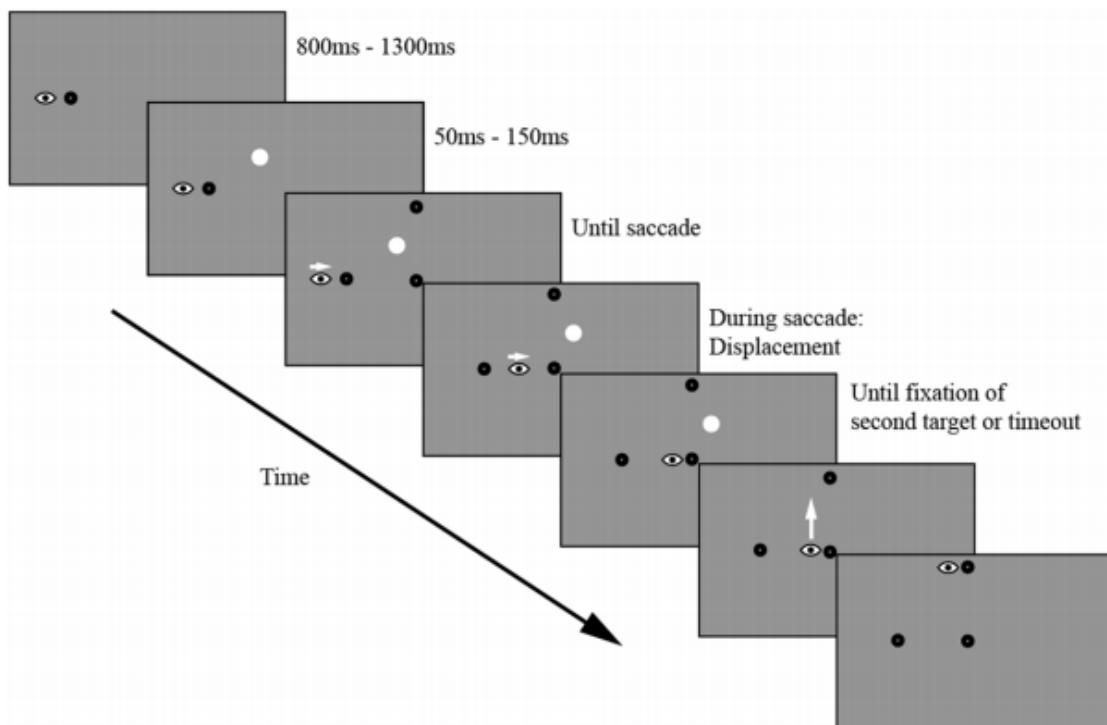


Figura 3.17: Test para analizar la incidencia de un objeto distractor sobre la curvatura de una sacada [17]

3.3.6. Psychophysics Toolbox

Son un conjunto de funciones y herramientas gratuitas que facilita la conexión entre *MATLAB* o *GNU Octave* y el hardware del computador para proporcionar una sincronización por debajo de los milisegundos. Utiliza los comandos OpenGL para permitir la reproducción y captura de video, así como las salidas de audio, además de facilitar la recopilación de respuestas de los usuarios a través de dispositivos de entrada regulares y especializados [61].

El principal uso de esta herramienta es la creación de experimentos de investigación en áreas como la psicología y neurología. Su flexibilidad en el desarrollo de pruebas y la popularidad del software *MATLAB* y *GNU Octave* en el ámbito científico, han posicionado esta herramienta como una de las más citadas en investigaciones de neurociencia pese a no contar con una interfaz de usuario y que su implementación sea únicamente por secuencias de código en lenguaje *MATLAB* u *Octave* [62].

En el año 2013 se realizó un estudio para evaluar el impacto del lenguaje en los movimientos oculares [18], el procedimiento consistía en presentar palabras que podían o no tener un significado concreto, las palabras con significado real implícitamente se vinculan con una posición en el espacio; así como un automóvil hace referencia al suelo, un avión hace referencia al cielo o aire. La instrucción que se le daba al sujeto era, mirar hacia el punto que se encontraba en la parte superior de la palabra cuando esta tenía un significado real, independientemente de su posible relación con una posición en el espacio, mientras que sí la palabra mostrada no tenía ningún sentido, debía mirar hacia el punto debajo de la palabra. En la Figura 3.18 se puede ver la secuencia de la prueba.

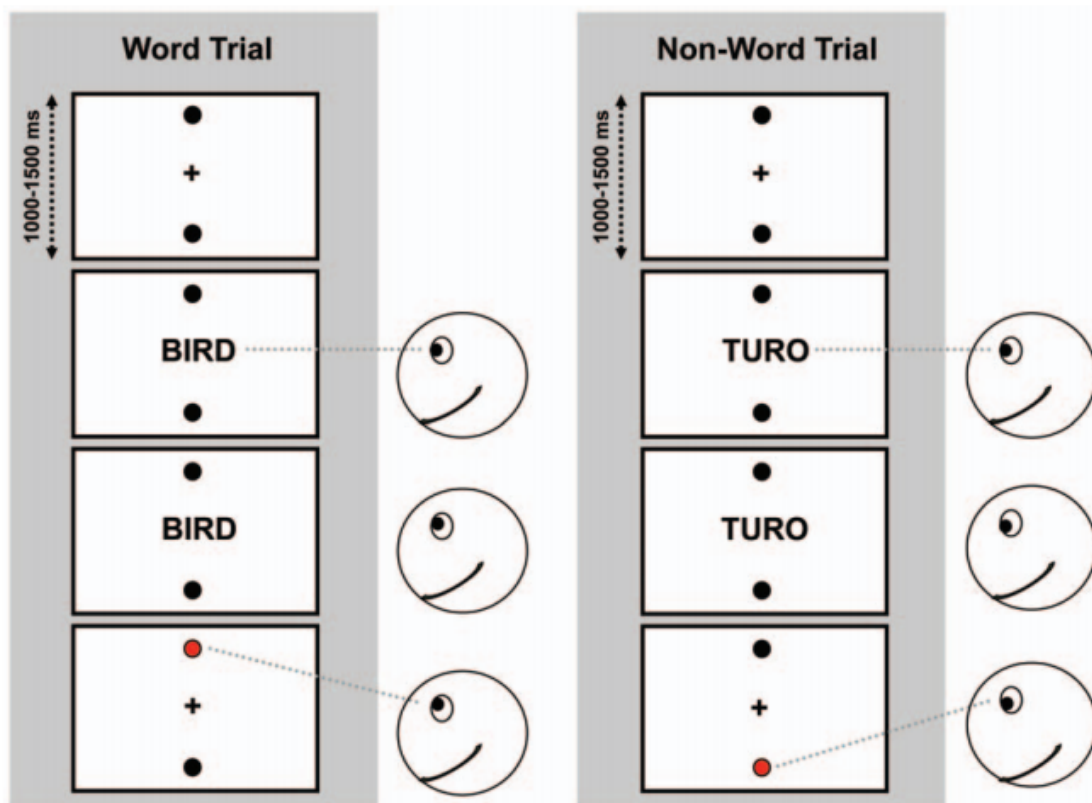


Figura 3.18: Test diseñado con Psychophysics Toolbox de MatLab [18]

A través de un *eye tracker* de la compañía *Eyelink* se realizaron las mediciones correspondientes cuyos resultados destacan la relación que existe entre las expresiones lingüísticas y la ubicación en el espacio que representan los objetos. Se observó que los sujetos realizaban una sacada hacia abajo si el objeto era representativo del suelo o hacia arriba si este hacía referencia a volar o al cielo, omitiendo la instrucción que se le indicó. Por otra parte, cuando las palabras no tenían algún significado no se presentaba este fenómeno y el sujeto orientaba su mirada hacia abajo con un mayor porcentaje de acierto.

3.3.7. Oscann desk100

La empresa *Aura Innovative Robotics SL.*, ha creado el software *Oscann desk100* con el fin de configurar pruebas de diagnóstico neurológico. La aplicación dispone de diferentes pruebas como, por ejemplo: test de sacadas, antisacadas, sacadas de memoria y antisacadas de memoria, entre otros.

En cada una de las pruebas se encuentra un estímulo visual que se conforma por un círculo y un punto de fijación en su interior, este estímulo se traslada a diferentes posiciones en el campo visual del monitor y dependiendo del tipo de paradigma de la prueba, el individuo debe cumplir con las tareas asignadas. La Figura 3.19 muestra los estímulos utilizados para realizar la calibración del dispositivo.



Figura 3.19: Puntos de fijación utilizados en la etapa de calibración de la prueba [9]

Oscann cuenta con una interfaz gráfica que ha sido desarrollada en *Qt Creator* en lenguaje QML y los gráficos son representados a través de OpenGL para garantizar la precisión de la integración con el *eye tracker* que se encarga de capturar los datos. En la Figura 3.20 se puede observar la ventana de inicio del software. En esta se encuentran las secciones para configurar las pruebas y capturar las imágenes del ojo.

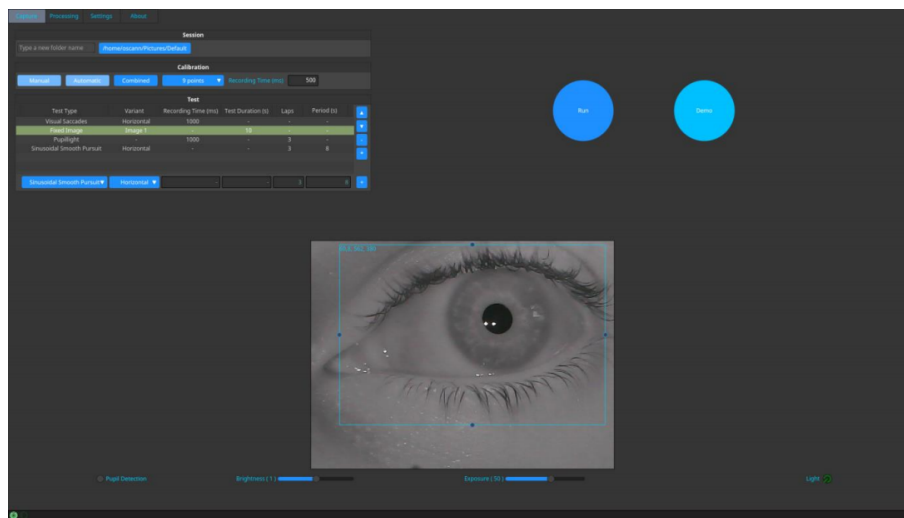


Figura 3.20: Ventana inicial de la plataforma *Oscann*[9]

El software de configuración de pruebas de *Oscann desk100* dispone de un banco de pruebas predefinidas con baja flexibilidad para la modificación de los parámetros de las mismas, en principio se permite seleccionar la cantidad de estímulos y el tiempo en que se desea que estos estímulos permanezcan visibles; esto es para pruebas de sacadas visuales, mientras que para las demás pruebas solo se permite modificar el tiempo de presentación de la prueba. En la Figura 3.21 se muestra un acercamiento de la Figura 3.20, en esta se puede apreciar de mejor manera la sección “*Calibration*” utilizada para configurar las pruebas.

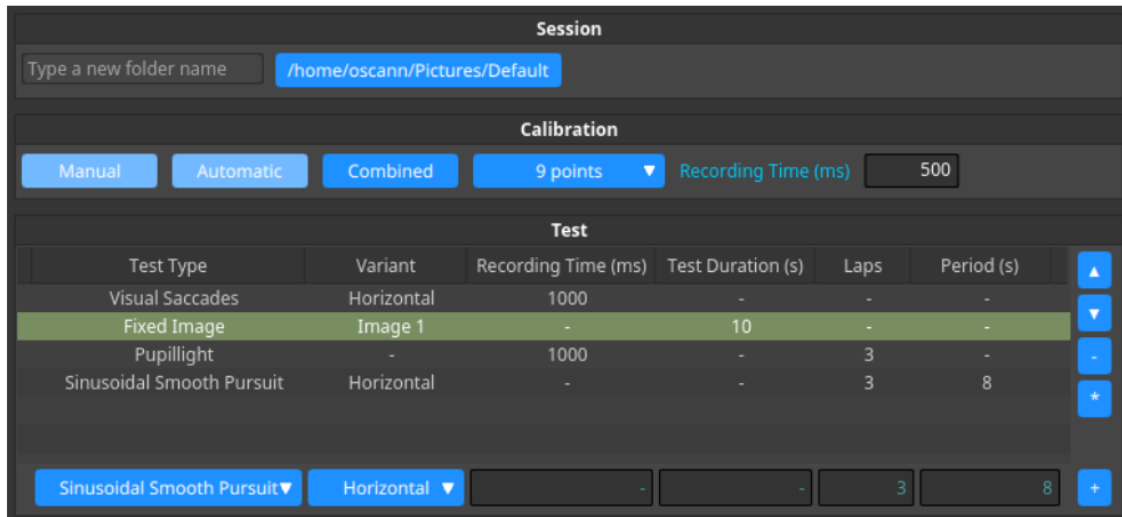


Figura 3.21: Ambiente de la interfaz de configuración de pruebas *Oscann desk100* [9]

Como síntesis de este capítulo, se propone la tabla (3.3.7) que muestra las principales características de las herramientas revisadas en esta sección, utilizada para identificar ciertas funciones que nos pueden interesar para el desarrollo de este trabajo. La tabla se consigna en la siguiente pagina debido a su tamaño.

	GUI	Gratuito	Escrito en	Codificación	Plataforma	Herramientas	Eye Tracker	Realimentación	Motor
E-prime	Si	No	Desconocido	E-Basic	Windows	Imágenes Diapositivas Textos Videos Sonidos	E-Prime Tobii	Si	DirectX
DMDX	Si*	Si**	DOS C++	Personalizado	Windows Web*	Imágenes Texto Videos Sonidos	Múltiple	Si	DirectX
Oscann Desk100	Si	No	C C++ QML	No	Linux	Formas Imágenes	Oscann	No	OpenGL
Experiment Builder	Si	No	Desconocido	Python	Windows Mac OS	Imágenes Formas Textos Videos Sonidos	EyeLink	Si	OpenGL
PsychoPy	Si	Si	Python	Python	Windows Mac OS Linux	Imágenes Formas Textos Videos Sonidos	Múltiples	Si	OpenGL
Psychophysics Toolbox	No	Si	C C++	MATLAB Octave	Windows Mac OS Linux	Imágenes Formas Textos Videos Sonidos	Múltiples	Si	OpenGL
OpenSesame	Si	Si	Python	Python	Windows Mac OS Linux	Imágenes Formas Textos Videos Sonidos	Múltiples	Si	OpenGL

*DMDX como tal, no tiene una interfaz de usuario, aunque la herramienta Visual DMDX basada en tecnologías Web fue creada para este propósito.

** Aunque es gratuita el código fuente de la aplicación no se encuentra disponible.

Tabla 3.3: Principales características de los software de configuración de pruebas para investigación en neurología

Capítulo 4

Arquitectura Software

En esta sección se encuentra todo lo concerniente al diseño y desarrollo de la interfaz lograda, partiendo desde su estado inicial o la interfaz original de la plataforma, los requerimientos pedidos, la estructura de funcionamiento y bases o fuentes para su diseño. También se describe la herramienta utilizada para realizar este proyecto, Qt; además, se exponen las secciones más importantes del código fuente necesario para el correcto funcionamiento de la interfaz.

4.1. Requerimientos

Los requerimientos se dividen en 2 partes, estratégicos, que son las necesidades generales o requerimientos principales con respecto al ámbito técnico y conceptual de la interfaz, y los requerimientos de alcance, que hacen referencia a las características funcionales de la interfaz, la finalidad de estas es brindar una experiencia agradable y fluida para el especialista médico. La mayor parte de los requerimientos fueron acordados por el equipo de trabajo de la empresa basándose en la opinión de los especialistas involucrados a lo largo del desarrollo del dispositivo *Oscann desk100*. También cabe resaltar que el trabajo se divide en 2 herramientas o interfaces diferentes, la primera está orientada al diseño de estímulos y la segunda a la configuración y diseño de pruebas, así también, los requerimientos varían en función de cada sección del trabajo.

4.1.1. Configuración y diseño de pruebas

Basados en el estado inicial del software *Oscann desk100* se requiere que la configuración de pruebas sea flexible y se ajuste al criterio que define el especialista, con la intención de proporcionar una herramienta completa para investigar sobre un mayor porcentaje de las patologías neurodegenerativas a partir de una recolección variada de datos con nuevos pacientes que anteriormente no se consideraban por falta de pruebas específicas o especiales.

Test o pruebas disponibles

A continuación, se describen los test que deben estar disponibles para el diseño de pruebas compatibles con la plataforma *OSCANN desk100*. Cabe resaltar que los test que se

mencionan aquí, se encuentran actualmente en el software de la plataforma, salvo algunas características que se detallan en la siguiente tabla.

- **Visual saccades:** En el test de sacadas visuales, el paciente deberá mirar los estímulos que se muestran en el monitor. Primero aparece un estímulo en el centro de la pantalla, luego estos cambian de posición según se hayan configurado durante el diseño de la prueba (ver Figura 4.1[19]), de esta manera se pueden caracterizar los movimientos sacádicos del paciente. Actualmente los objetivos se muestran sobre un solo eje sea horizontal o vertical, en ellos se encuentran posiciones específicas, (± 5 , ± 10 y ± 20) grados para la variante horizontal y (± 5 , ± 10) para la vertical. Además, los cambios de posición se hacen aleatoriamente. Nuestro trabajo para este tipo de test es brindar mayor flexibilidad para su diseño añadiendo funcionalidades como, permitir la selección de cualquier punto dentro del campo visual del monitor, sea en horizontal, vertical o con ambas componentes (posición oblicua), brindar la posibilidad de fijar el orden en que se presentan los estímulos y definir el tiempo en el que cada objetivo estará visible; en el software actual de la plataforma también es posible configurar dicho tiempo.
- **Visual antisaccades:** El test de antisacadas visuales es similar al anterior, con la diferencia que esta vez el paciente debe mirar al lado contrario donde el estímulo aparece, en la Figura 4.1 se muestra un esquema del test. Los requerimientos adicionales para este test son iguales que los del test de sacadas visuales.
- **Memory saccades:** Este test es similar a la prueba de sacadas, inicia con un estímulo central seguido de un cambio de posición para luego regresar al centro, hasta aquí un test de sacadas normal, ahora, cuando el estímulo central desaparece por segunda vez, el paciente debe llevar la mirada hacia la posición en la cual el estímulo apareció anteriormente, en la Figura 4.1 se ilustra el comportamiento del test. Actualmente solo se puede configurar el tiempo de visibilidad de los estímulos, por lo tanto, se deben adicionar las mismas funcionalidades mencionadas en el test de sacadas visuales, además, debe contar con un parámetro específico que permita seleccionar si el estímulo debe estar visible durante la prueba.
- **Memory antisaccades:** Este test es una combinación del test de antisacadas y el test de memoria, en esta prueba el paciente debe mirar al lado contrario donde el estímulo aparece y una vez el estímulo central ha desaparecido por segunda vez, debe recordar la misma zona donde miro anteriormente y fijar su mirada en ese punto.
- **Countermanding saccades:** Es un test diseñado para evaluar la capacidad de una persona para detener una sacada. La prueba consiste en mostrar estímulos a manera de un test de sacadas, durante la prueba se genera una señal de parada representada por un estímulo central, la instrucción que se le da al paciente es que si repentinamente aparece un estímulo en el centro de la pantalla, debe mirarlo inmediatamente, consiguiendo detener el movimiento sacádico del paciente.
- **Fixation:** Es un test muy sencillo en el cual se muestra un estímulo en el centro de la pantalla durante unos segundos. Para este tipo de pruebas solo se necesita configurar

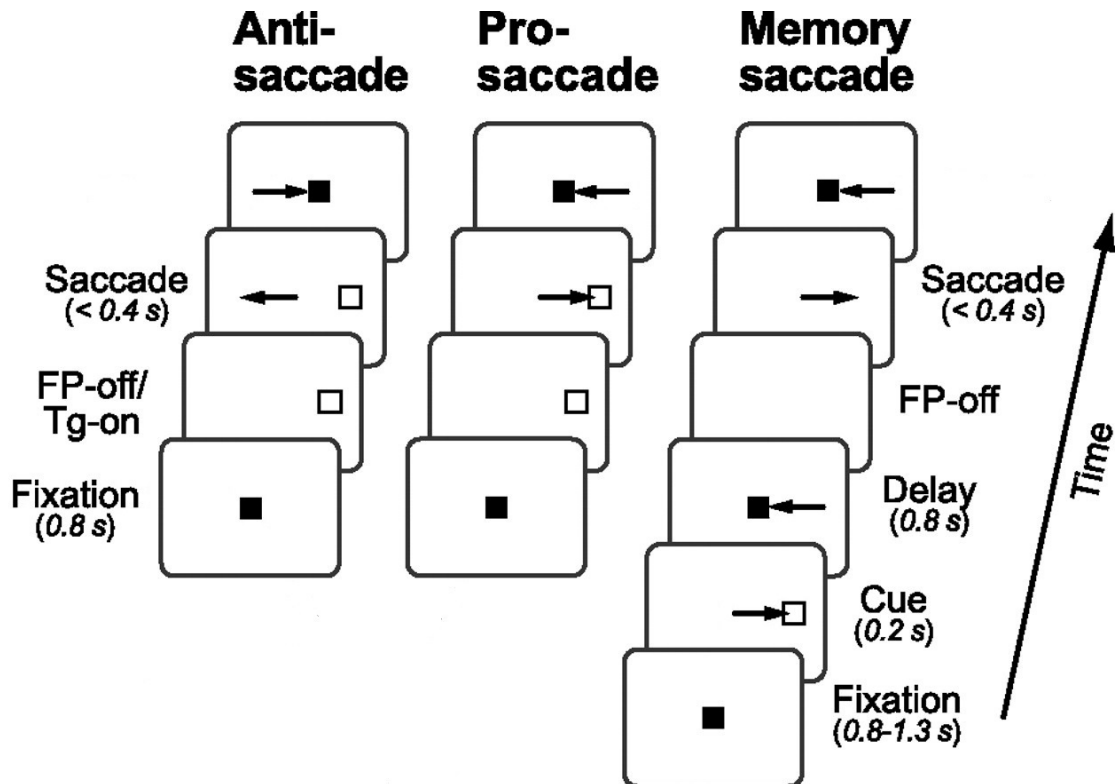


Figura 4.1: Secuencia de eventos de pruebas con antisacadas, sacadas y sacadas de memoria [19].

el tiempo que debe permanecer visible el estímulo, parámetro que actualmente está disponible. Adicionalmente, se puede seleccionar una posición diferente al centro de la pantalla, esta característica es un requerimiento para la interfaz y actualmente no está disponible.

- **Pupil light:** El test de reflejo pupilar consiste en iluminar (color blanco) y apagar (color negro) completamente la pantalla, comparable con la acción que realiza un médico cuando evalúa a un paciente encendiendo una linterna en su ojo. Se realiza para estudiar la dilatación y contracción de la pupila. Los intervalos para cada cambio de estado son fijos actualmente. El requerimiento principal es habilitar la opción de modificar el intervalo con el que cambian los colores. Como un requerimiento adicional, se propone que dichos contrastes de color se puedan modificar, es decir, proporcionar más colores además del blanco y negro.
- **Smooth pursuit:** Un test de persecución lenta consiste en que un estímulo se desplaza con relativa lentitud a lo largo (puede ser vertical u horizontal) de la pantalla. La velocidad se puede configurar para cada tipo de movimiento, actualmente existe el movimiento lineal y sinusoidal, pero se ha propuesto incrementar las opciones, añadiendo una gran variedad de trayectorias a la herramienta. En la Figura 4.2 se puede observar los resultados de una prueba de seguimiento lento sinusoidal realizadas por un paciente diagnosticado con Parkinson y un individuo control.

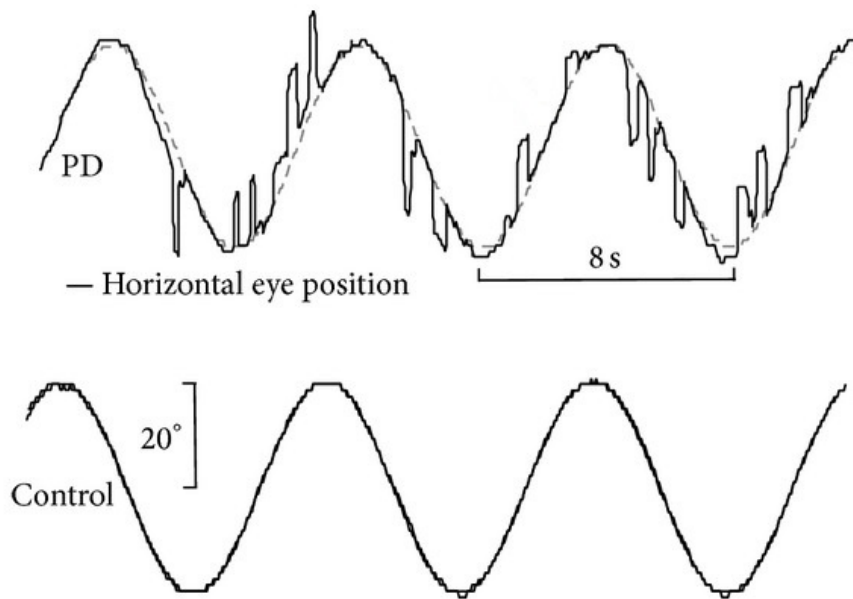


Figura 4.2: Resultados de test de seguimiento lento sinusoidal [20]

Inclusión de paradigmas *Gap* y *Overlap*

Estos parámetros se utilizan en test de sacadas o antisacadas y actualmente no están disponibles en la plataforma *OSCANN*. Los términos permiten modificar el tiempo en el que los estímulos objetivo (estímulos que aparecen en puntos distintos del centro $[0,0]$) aparecen durante la prueba, así, el parámetro *Gap* hace que el objetivo aparezca tiempo después de que el estímulo central ha desaparecido, mientras que el término *Overlap*, como lo indica su traducción, es un tiempo en el que se presenta una superposición, en este caso el objetivo aparece un momento antes de que el estímulo central desaparezca. En la Figura 4.3 se muestra un ejemplo de estas condiciones [21].

En el anterior esquema se puede observar el funcionamiento de los paradigmas. En el condicional de *Overlap* se puede apreciar como en la segunda escena se muestran 2 estímulos simultáneamente durante 200 milisegundos, mientras que con el condicional *Gap* se genera una brecha o espacio durante el cual no se visualiza ningún parámetro durante el mismo tiempo que el anterior. En este caso particular la asignación del tiempo se realiza con el mismo parámetro (*Gap*), dependiendo del signo que tenga el valor se pueden obtener ambos condicionales; positivo para *Overlap* y negativo para *Gap*. Esta configuración es un poco confusa y no se realizará de esta manera para nuestro trabajo. La interfaz desarrollada contará con campos independientes para ingresar estos valores, el cálculo para asignar los tiempos de visualización de cada estímulo se realizará internamente, así el especialista médico no se preocupará por signos o cálculos ambiguos.

Por otra parte también se integrará el paradigma *Gap* para los demás test, es decir, se podrá hacer uso de dicho parámetro cuando se diseñen pruebas combinadas con el fin de generar tiempos cortos de espera entre cada test, mientras no modifique el funcionamiento original de los mismos.

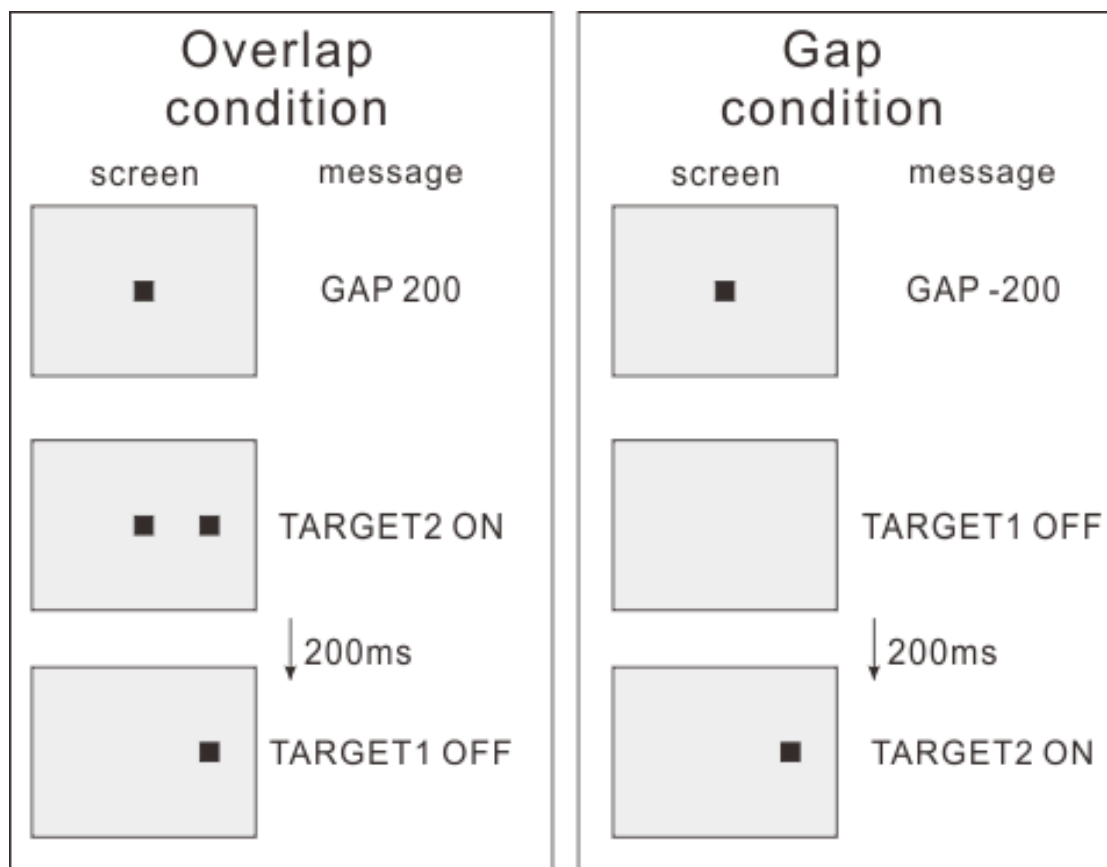


Figura 4.3: Esquema de funcionamiento de los paradigmas *Overlap* y *Gap* [21].

Integración de los test

Como se mencionó al final de la sección 3.3.7, en la cual se hace una breve descripción de la plataforma *Oscan desk100*, el método con el que se definen las pruebas actualmente se basa en un banco de pruebas y con algunos parámetros configurables. Para cumplir con el desarrollo de este requerimiento se debe modificar el procedimiento para diseñar una prueba. Como se observó en la Figura 4.1, los test se componen de uno o varios elementos o estímulos que cambian de posición, se desplazan, desaparecen, entre otras acciones. Al mes de febrero de 2018 cuando se inicia con este proyecto, la configuración individual de los estímulos no es posible debido a los parámetros preestablecidos de las pruebas, por lo tanto, se desea cambiar por un método que permita configurar cada uno de ellos y brindar mayor flexibilidad al diseño de las pruebas.

Inicialmente la presentación de los test se realizaba de manera separada, es decir, para realizar una prueba con varios de los test disponibles, se debe hacer por etapas, cada vez que finaliza un test se debe presionar una tecla para lanzar el próximo y así hasta terminar con la prueba. Con la integración de las pruebas se pretende que la presentación sea más fluida y que se puedan diseñar pruebas con componentes o test mixtos, además, de esta manera el especialista médico a cargo se evita la tarea de presionar una tecla cada vez que haya un cambio de test.

Este requerimiento se planteó en dos partes, como primera instancia se propone agrupar las pruebas con parámetros iguales, una vez se disponga de esta configuración se

pretende integrar las restantes.

- *Primer punto:* Para esta implementación se desea agrupar las secciones de diseño de los test de sacadas y antisacadas visuales, sacadas y antisacadas de memoria y sacadas de tipo *Countermanding*. Debido a la similitud en los parámetros de definición, es posible agrupar el diseño de estos test en una sola sección, ahorrando espacio visual en la interfaz y ampliando las oportunidades de personalizar una prueba gracias a la posibilidad de combinar elementos de varios test en una prueba.
- *Segundo punto:* Como requerimiento final y una vez se observe el correcto funcionamiento del punto anterior, se pretende integrar todas las secciones de diseño de los test, de esta manera habrá un único apartado para diseñar una prueba en la cual estarán disponibles los elementos de diseño de todos estos. Con esta configuración, aumentarán las opciones para desarrollar pruebas con diferentes componentes, buscando que los especialistas médicos personalicen sus pruebas con una mayor flexibilidad.

Para la implementación de este requerimiento es importante recordar que el comportamiento original de cada test se debe respetar, en otras palabras, cuando se diseñe una prueba con elementos de diferentes test, se debe garantizar que el funcionamiento de cada sección de la prueba sea el indicado. Cabe resaltar que durante el diseño de una prueba el especialista médico no debe preocuparse por esta condición ya que el código interno de la interfaz estará configurado para realizar ajustes de este tipo, así, el especialista puede realizar cualquier combinación de estímulos o elementos de distintos test y siempre se garantizará un correcto funcionamiento de las secciones de la prueba.

Iteración de elementos de una prueba

En pocas palabras significa que el especialista médico pueda reutilizar los elementos que ya haya ingresado durante el diseño de una prueba, o más simple, que pueda repetir una sección de la misma. Este requerimiento es necesario cuando se desarrollen pruebas que contengan muchos elementos o en ocasiones que el usuario vea necesario repetir los estímulos dentro de una prueba. Actualmente, para la definición de test de sacadas o antisacadas, este requerimiento se puede ver reflejado en el parámetro que permite definir la cantidad de estímulos para cada test.

Aleatoriedad de una prueba

Al igual que en el los test de sacadas visualmente guiadas del actual banco de pruebas, en los cuales los estímulos objetivo toman valores aleatorios de posición, se ve necesario incluir una opción para que la presentación de la prueba se realice aleatoriamente; distinto del orden en que se diseñó la prueba. También se propone un método para que esta organización aleatoria de la prueba, no altere el funcionamiento original de los test o las distintas secciones de la prueba. Para lograr esto se habilitará una opción que establezca un vínculo entre los elementos de una prueba y así respetar el orden de presentación de algunos de estos. Cabe resaltar que para test con movimientos sacádicos, se definirá una relación automática de sus componentes para garantizar que estas secciones se realicen en las 3 etapas que se describieron al final del requerimiento de integración de los test.

Previsualización de pruebas

Con esta función se pretende realizar una pequeña simulación del comportamiento de la prueba o también secciones de la misma, de esta manera será posible realizar una vista previa de los elementos antes de agregarlos definitivamente a esta. El “*Preview*” se implementa para hacer la configuración de las pruebas más dinámica y darle al especialista médico un tipo de realimentación adecuada para el diseño de las mismas. Esta visualización será de gran ayuda sobre todo cuando se desea observar el comportamiento de los paradigmas *Gap* u *Overlap*, que en un principio pueden ser parámetros complejos de usar y la manera más fácil de comprender es de manera visual y en general comprender el funcionamiento de las pruebas. Adicionalmente, se consideró agregar una opción para crear secciones de tipo demostración durante la prueba. Esta opción estará disponible para los test que pueden presentar mayor dificultad para los sujetos que se someten a las pruebas, presentando una pequeña demostración de los movimientos o lugares donde debe dirigir su mirada. Dichos test son los de movimientos sacádicos en los cuales los estímulos definidos como demostración contarán con un indicador visual que le indicará al sujeto cual es el estímulo que debe mirar.

Desarrollo de archivos XML

Este requerimiento es muy importante porque será el método de comunicación entre nuestra interfaz y la plataforma *Oscann desk100*, además para la comunicación entre las secciones de diseño de estímulos y diseño de pruebas. Esta última correspondencia es necesaria para habilitar el uso de los estímulos en la plataforma de diseño de pruebas. Se desarrollarán formatos diferentes para cada tipo de comunicación con algunas especificaciones individuales, los archivos estarán en la misma fuente o carpeta destino, desde los archivos destinados para *Oscann desk100*, hasta los de comunicación interna de la interfaz.

Una tarea importante que cumple esta implementación es integrar las dos herramientas de diseño a través de la inclusión de los estímulos en la herramienta de diseño de pruebas. En síntesis, los estímulos que se crean en la sección de diseño de estímulos estarán disponibles de manera inmediata en la herramienta de diseño de pruebas, esta rápida comunicación permite navegar de manera más fluida entre ambas herramientas.

4.1.2. Diseño de estímulos

Para la configuración del estímulo se debe establecer una variedad de opciones gráficas basadas en diferentes formas geométricas y colores que permitan al especialista realizar una elección que considere adecuada, incluyendo parámetros como: tamaño, orientación, brillo, contraste, saturación y desenfoque. Además, para estímulos con propiedades de movimiento se deben incluir parámetros como la velocidad, el tipo de trayectoria, el eje de movimiento, entre otros.

- El apartado de configuración visual del estímulo debe presentar diferentes controles para permitir la elección de un estímulo basado en figuras geométricas básicas, que corresponden a un círculo, cuadrado, estrella y triángulo. Estas formas deben incluir en su interior dos bordes, uno interno y otro externo, adicionalmente se

dispondrán dos áreas encerradas por estos bordes, generando así un área interna y externa. La finalidad de esta configuración es darle una mayor personalización a estos elementos a través de una paleta de colores que se dispondrá en otra sección para la selección y combinación de los tonos de preferencia para el aspecto del estímulo. De igual forma, en el presente apartado se debe incluir un botón que despliegue un cuadro de diálogo para que el especialista médico realice la elección de una imagen como configuración de aspecto en el estímulo si así lo desea.

- En otro apartado se establecen controles para la modificación del tamaño y la rotación tanto de las formas como de las imágenes que representan el estímulo, los cambios realizados a estas variables deben ser visibles de manera instantánea para generar fluidez en la interfaz.
- Se debe establecer un conjunto de controles donde sea posible visualizar una paleta de colores para modificar los tonos de color en el estímulo basado en formas y sus respectivas secciones, así como en el estímulo a partir de una imagen, generando una silueta, y también el cambio de color en el fondo de la aplicación.
- Definir una sección con opciones para animar el estímulo, donde se deben incluir controles para modificar el aspecto gráfico, con características como: saturación, brillo, contraste y desenfoque además de incluir en este mismo apartado cinco animaciones principales para darle dinámica a la presentación del estímulo. Estas animaciones se definen en los siguientes ítem.
 1. Parpadeo del estímulo, al activar esta animación se debe realizar un cambio de color en un tiempo predefinido. Para esta animación se disponen de controles para modificar los colores con los que se desea interactuar además de un control para definir el tiempo de muestreo de cada uno de ellos.
 2. Parpadeo en el fondo de la aplicación, esta animación al igual que la anterior, comprende el cambio de color en el fondo de la aplicación en un tiempo predefinido, se disponen también los respectivos controles para la selección de tonos y el tiempo de muestreo de los mismos.
 3. Pivotes en el estímulo, esta animación comprende el movimiento del estímulo ya sea en vertical u horizontal, como su nombre lo indica, la ejecución está basada en recrear un efecto de pivote sobre dos puntos que se definen a través de controles dispuestos en la sección.
 4. Contracción del estímulo, como su nombre lo indica en esta animación se recrea la contracción del estímulo, donde se definen dos valores correspondientes a la escala desde la cual el especialista desearía que se ejecute la escena.
 5. Giro del estímulo, la ejecución de esta animación genera un giro del estímulo de 360° continuamente, esta escena no tiene asociado ningún otro parámetro, solo se encuentra disponible el control para activarla o desactivarla.
 6. Para concluir, se deben disponer controles para la ejecución de movimiento en unas trayectorias predefinidas, estos movimientos se podrán realizar en vertical y horizontal, además las trayectorias disponibles deben contener un gráfico donde se pueda visualizar el tipo de movimiento que será reproducido, esto con el fin de guiar al especialista médico.

- Posteriormente, el estímulo debe contar con un espacio donde se puedan visualizar todos los cambios en los parámetros anteriormente nombrados, estas variaciones se deben reflejar de manera instantánea para que el especialista médico elija la configuración de su preferencia sin presentar retrasos en la navegabilidad de la interfaz. Se debe mostrar una tabla en la misma ventana, con la finalidad de que se añadan los datos que ha configurado sobre el aspecto del estímulo y se realice una verificación de los mismos, si alguno o todos los cambios que se han realizado no son de la conformidad del especialista médico debe establecerse también un botón que permita regresar a los datos predefinidos por la aplicación.
- Finalmente, debe definirse un botón que permita exportar todos los parámetros configurados para el estímulo en un archivo XML. Aquí se debe asignar una etiqueta a los ficheros para facilitar la integración con el área del banco de pruebas, ya que todos los estímulos que se pueden generar desde esta parte de la interfaz no serán compatibles con cada uno de los test disponibles. Esto facilitará la configuración de las pruebas en la siguiente fase.

4.2. Metodología

La implementación de la interfaz se realizó adoptando la metodología *SCRUM*. Este método plantea empezar con la recolección de los requerimientos del proyecto para una posterior clasificación y planificación de su desarrollo. La ejecución de estos objetivos se debe realizar en un periodo de tiempo acordado, normalmente de dos a cuatro semanas y dentro del mismo ciclo (*sprint*) se cumplen con objetivos de desarrollo diarios. Una vez cumplido el ciclo de ejecución principal se realiza la entrega del producto del desarrollo a las partes implicadas (supervisores o clientes) solicitando una realimentación sobre el trabajo realizado. Finalmente, las partes implicadas realizan la realimentación bajo la emisión de un concepto sobre el estado actual del desarrollo y se incorporan estos nuevos requerimientos al banco principal de requisitos para realizar nuevamente una iteración. Bajo esta metodología, se repite el ciclo el número de veces que sea necesario para lograr el resultado deseado. En la planificación de este trabajo se definieron 5 ciclos de desarrollo según la clasificación de los requerimientos impuestos. En el Anexo D. de esta monografía se encuentra toda la información respecto al desarrollo de los ciclos o *sprint's* del trabajo.

4.3. Historias de usuario

En metodologías de desarrollo como *SCRUM* se utilizan las historias de usuario para definir las funcionalidades o requisitos que deben implementarse en el sistema software de acuerdo a los requerimientos impuestos por el cliente o supervisor del trabajo, en pocas palabras, las historias de usuario se usan para establecer lo que se debe construir en las diferentes etapas de un proyecto para cumplir con las necesidades del cliente.

A continuación, se presentan las historias de usuario que representan los requerimientos de este trabajo y permiten definir las funcionalidades con las que debe contar la interfaz desarrollada en este proyecto.

Historia Usuario Nro:	1	Título:	Test disponibles en la herramienta
Programador:	Daniel	Usuario:	Supervisor del trabajo Sprint: 1
Descripción:	COMO	supervisor del trabajo	
	QUIERO	que la interfaz cuente con diferentes test	
	PARA	construir pruebas compatibles con la plataforma Oscann desk100	
Criterios de validación:	En caso que el especialista desee seleccionar uno de los test disponibles, cuando realice la configuración de una prueba, la interfaz debe contar con una lista que permita seleccionar cada uno de los test.		
	En caso que el especialista desee modificar los parámetros de una prueba, cuando elija uno de los test disponibles, la interfaz debe mostrar las opciones o parámetros que se pueden configurar correspondientes a cada test.		
	En caso que el especialista desee observar los datos que va ingresando y adicionalmente desee eliminar uno o varios elementos de la prueba, cuando realice la configuración de una prueba, la interfaz debe contar con una tabla que permita visualizar los datos y además permita modificar o eliminar dichos datos.		

Tabla 4.1: Historia de usuario para requerimiento sobre los test disponibles en la herramienta

Historia Usuario Nro:	2	Título:	Definición de las formas e imágenes
Programador:	Mario	Usuario:	Supervisor del trabajo Sprint: 1
Descripción:	COMO	supervisor del trabajo	
	QUIERO	tener la opción de escoger figuras geométricas o imágenes	
	PARA	cambiar el aspecto del estímulo	
Criterios de validación:	En caso de que el especialista quiera seleccionar alguna forma geométrica, cuando este configurando el aspecto del estímulo, la interfaz debe contar con una sección que disponga de diferentes controles, que permitan la elección entre un círculo, cuadrado, triangulo y estrella.		
	En caso de que el especialista quiera seleccionar una imagen desde el directorio local de archivos, cuando este configurando el aspecto del estímulo, la interfaz debe contar con un botón, que permita desplegar un cuadro de dialogo para la elección de ficheros en formato JPG o PNG.		

Tabla 4.2: Historia de usuario para requerimiento sobre la definición de formas e imágenes para el diseño de estímulos

Historia Usuario Nro:	3	Título:	Inclusión de paradigmas <i>Gap</i> y <i>Overlap</i>
Programador:	Daniel	Usuario:	Supervisor del trabajo Sprint: 2
Descripción:	COMO	supervisor del trabajo	
	QUIERO	que la interfaz permita trabajar con paradigmas <i>Gap</i> y <i>Overlap</i>	
	PARA	diseñar test de movimientos sacádicos especiales	
Criterios de validación:	En el caso que el especialista desee utilizar estos paradigmas, cuando realice la configuración de test con movimientos sacádicos, la interfaz debe disponer de un apartado para ingresar un valor de tiempo que permita modificar el comportamiento de los paradigmas en la prueba.		
	En el caso que el especialista desee usar el paradigma <i>Gap</i> , cuando configure una prueba con test diferentes, la interfaz debe contar con un apartado que permita ingresar un valor de tiempo para configurar el comportamiento de este paradigma entre cada transición de los diferentes test de la prueba diseñada.		

Tabla 4.3: Historia de usuario para requerimiento sobre inclusión de paradigmas *Gap* y *Overlap*

Historia Usuario Nro:	4	Título:	Definición de escala y rotación
Programador:	Mario	Usuario:	Supervisor del trabajo Sprint: 2
Descripción:	COMO	supervisor del trabajo	
	QUIERO	disponer de diferentes herramientas que me permitan modificar valores de píxeles y grados	
	PARA	cambiar el tamaño y la orientación del estímulo.	
Criterios de validación:	En caso que el especialista quiera cambiar el tamaño del estímulo, cuando seleccione la barra de desplazamiento o introduzca el valor de píxeles deseados, se debe indicar el cambio de manera instantánea para generar fluidez en la interfaz.		
	En caso de que el especialista quiera cambiar la rotación del estímulo, cuando seleccione la barra de desplazamiento o introduzca el valor de grados deseados, se debe indicar el cambio de manera instantánea para generar fluidez en la interfaz.		

Tabla 4.4: Historia de usuario para requerimiento sobre definición de escala y rotación para el diseño de estímulos

Historia Usuario Nro:	5	Título:	Paleta de colores y efectos gráficos
Programador:	Mario	Usuario:	Supervisor del trabajo Sprint: 2
Descripción:	COMO	supervisor del trabajo	
	QUIERO	disponer de herramientas que me permitan modificar los colores en el estímulo y el fondo de la aplicación, además de aplicar efectos gráficos	
	PARA	cambiar el aspecto del estímulo	
Criterios de validación:	En caso de que el especialista quiera cambiar los colores presentes en el estímulo o el fondo de la aplicación, la interfaz debe contar con una sección de controles, donde pulsando un botón se desplegará un cuadro de dialogo con la paleta de colores que permite seleccionar el tono de preferencia o introducirlo en formato hexadecimal.		
	En caso de que el especialista quiera aplicar efectos gráficos al estímulo, cuando seleccione la barra de desplazamiento o introduzca el valor deseado, se debe reflejar de manera instantánea la variación en el aspecto del estímulo contando con los siguientes efectos: brillo, contraste, saturación y borrosidad.		

Tabla 4.5: Historia de usuario para requerimiento sobre selección de colores y efectos gráficos en el diseño de estímulos

Historia Usuario Nro:	6	Título:	Combinación de test en una prueba
Programador:	Daniel	Usuario:	Supervisor del trabajo Sprint: 3
Descripción:	COMO	supervisor del trabajo	
	QUIERO	que la interfaz permita diseñar pruebas con distintos test	
	PARA	dar mayor flexibilidad y personalización a la configuración de pruebas	
Criterios de validación:	En el caso que el especialista desee utilizar varios test, cuando realice el diseño de una prueba, la interfaz debe permitir adicionar diferentes elementos o estímulos de los test disponibles en la herramienta a una misma prueba.		
	En el caso que el especialista desee verificar el tipo de elemento ingresado a la prueba, cuando realice el diseño de una prueba mixta, la tabla de visualización de datos debe contar con una sección que muestre el tipo al cual pertenece el elemento o estímulo de la prueba.		

Tabla 4.6: Historia de usuario para requerimiento sobre combinación o integración de test en una prueba

Historia Usuario Nro:	7	Título:	Iteración de elementos en una prueba
Programador:	Daniel	Usuario:	Supervisor del trabajo Sprint: 3
Descripción:	COMO	supervisor del trabajo	
	QUIERO	que la interfaz permita reutilizar los elementos ingresados a una prueba	
	PARA	repetir la sección de la misma, que comprenden los elementos mencionados	
Criterios de validación:	En el caso que el especialista desee repetir una sección de la prueba, cuando realice el diseño de un experimento que contenga muchos elementos, la interfaz debe contar con un apartado para especificar el número de veces que se repetirá dicha sección.		
	En el caso que el especialista desee comprobar la iteración de los elementos, cuando realice una repetición de una sección de la prueba, la tabla de visualización de datos debe mostrar los nuevos elementos creados a partir de la iteración realizada, respetando el orden en que los estímulos se agregaron inicialmente a la prueba.		

Tabla 4.7: Historia de usuario para requerimiento sobre iteración de elementos en una prueba, para repetir una sección de la misma

Historia Usuario Nro:	8	Título:	Implementación de animaciones
Programador:	Mario	Usuario:	Supervisor del trabajo Sprint: 3
Descripción:	COMO	supervisor del trabajo	
	QUIERO	tener la posibilidad de aplicar diferentes tipos de animaciones	
	PARA	darle una mejor dinámica a la presentación del estímulo.	
Criterios de validación:	En caso de que el especialista desee aplicar animaciones al estímulo, cuando este en proceso de diseño, la interfaz debe contar con diferentes controles que permiten activar o desactivar el estado de los siguientes comportamientos: parpadeo, contracción, pivote y giro además de parpadeo en el fondo de la aplicación.		
	En caso de que el especialista desee cambiar los colores definidos para la transición en los parpadeos, cuando este activado, se debe disponer de botones que desplieguen una paleta de colores y permita seleccionar los tonos de preferencia, además de permitir cambiar la velocidad de estas secuencias con el ingreso del tiempo.		
	En caso de que el especialista desee cambiar parámetros asociados a la contracción o al pivote, como los valores de escala y la distancia de pivote, la interfaz debe contar con casillas para modificar estos valores a partir de la introducción de datos.		

Tabla 4.8: Historia de usuario para requerimiento sobre implementación de animaciones en el diseño de estímulos

Historia Usuario Nro:	9	Título:	Aleatoriedad en la presentación de los estímulos de una prueba
Programador:	Daniel	Usuario:	Supervisor del trabajo Sprint: 4
Descripción:	COMO	supervisor del trabajo	
	QUIERO	que la interfaz permita definir un orden específico o aleatorio	
	PARA	la presentación de pruebas en la plataforma Oscann desk100	
Criterios de validación:	En caso que el especialista desee establecer un orden específico en la presentación de los estímulos de una prueba, cuando realice la configuración de los elementos, la interfaz debe contar con una opción que permita definir un vínculo entre cada elemento para garantizar un orden en la presentación de la prueba.		
	En caso que el especialista desee definir una prueba como aleatoria, cuando finalice la configuración del experimento, la interfaz debe mostrar un apartado que permita ingresar un nombre a la prueba y además, permita definir si la presentación de los estímulos durante la prueba realizada con la plataforma, se hará de forma aleatoria o con el orden establecido en la construcción del experimento.		

Tabla 4.9: Historia de usuario para requerimiento sobre aleatoriedad en la presentación de estímulos de una prueba

Historia Usuario Nro:	10	Título:	Implementación de trayectorias
Programador:	Mario	Usuario:	Supervisor del trabajo Sprint: 4
Descripción:	COMO	supervisor del trabajo	
	QUIERO	tener la posibilidad de seleccionar diferentes trayectorias que modifiquen la visualización del estímulo	
	PARA	ampliar el apartado de pruebas de seguimiento lento	
Criterios de validación:	En caso de que el especialista desee activar trayectorias para el estímulo, cuando quiera configurar un estímulo para seguimiento lento, la interfaz debe contar con un control que le permita activar esta opción.		
	En caso de que el especialista desee definir el eje de desplazamiento de las trayectorias, cuando esta configurando un estímulo para seguimiento lento, la interfaz debe contar con un control que permita elegir el eje X o Y.		
	En caso de que el especialista desee un comportamiento de trayectoria diferente, cuando esta configurando un estímulo para seguimiento lento, la interfaz debe contar con una lista para la elección de un tipo de movimiento diferente, que soportado el una gráfica debe describir el tipo de desplazamiento que el estímulo realizará.		

Tabla 4.10: Historia de usuario para requerimiento sobre implementación de trayectorias en el diseño de estímulos

Historia Usuario Nro:	11	Título:	Previsualización de pruebas
Programador:	Daniel	Usuario:	Supervisor del trabajo Sprint: 5
Descripción:	COMO	supervisor del trabajo	
	QUIERO	que la interfaz permita visualizar los experimentos diseñados	
	PARA	tener una realimentación adecuada de los datos ingresados a la prueba	
Criterios de validación:	En caso que el especialista desee visualizar el comportamiento de un estímulo o elemento, mientras está diseñando una prueba, la interfaz debe contar con un apartado que permita visualizar los elementos individualmente, antes de agregarlos a la tabla de datos de la prueba.		
	En caso que el especialista desee visualizar la prueba diseñada, cuando finalice con la construcción de esta, la interfaz debe contar con la opción de previsualizar el comportamiento de los datos que conforman el experimento diseñado.		
	En caso que el especialista desee mostrar una sección de la prueba a modo de demostración, cuando realice el diseño de la misma, la interfaz debe contar con una opción que le permita definir dichos elementos en modo demostración y, además, permitir que el especialista pueda observar esta demostración de la sección del experimento.		

Tabla 4.11: Historia de usuario para requerimiento sobre previsualización de pruebas

Historia Usuario Nro:	12	Título:	Exportación de datos en archivos XML
Programador:	Mario	Usuario:	Supervisor del trabajo Sprint: 5
Descripción:	COMO	supervisor del trabajo	
	QUIERO	exportar los datos de las pruebas y el estímulo configurados en archivos XML	
	PARA	facilitar la integración con la plataforma Oscann desk100	
Criterios de validación:	En caso de que el especialista haya configurado los parámetros correspondientes a la prueba y a los estímulos, cuando quiera exportarlos se debe contar con un botón que le permita visualizarlos en una tabla para verificarlos.		
	En caso de que el especialista haya verificado los datos correspondientes a la configuración de pruebas y del estímulo, se debe contar con un botón que permita realizar la exportación de parámetros.		
	En caso de que el especialista haya pulsado el botón de exportar parámetros, se desplegará un cuadro de dialogo que debe solicitar el ingreso de un nombre para asignarle al fichero donde se almacenaran los datos previamente configurados.		

Tabla 4.12: Historia de usuario para requerimiento sobre exportación de datos desde la interfaz hacia archivos XML externos

4.4. Casos de uso

En ingeniería, una de las partes más importantes en el desarrollo es el proceso de modelado, debido a que un modelo debe representar la estructura de un sistema a partir de la inclusión de ideas, y estas desarrollarse a través de diferentes herramientas que permitan alcanzar los objetivos del trabajo, adicionalmente el modelado conlleva a estandarizar los términos expuestos y facilitar la comunicación entre las diferentes áreas de conocimiento que se involucran en la implementación.

Los diagramas UML constituyen un lenguaje gráfico, donde es posible visualizar, especificar, construir y documentar un sistema software. En otras palabras, se puede decir que es una guía donde se incluyen conceptos referentes a procesos y funciones de un sistema.

Retomando la metodología adoptada para el presente desarrollo, la estructura de este trabajo se definió previamente a la implementación, haciendo uso de diagramas de “caso de uso” para definir los requerimientos de la interfaz y de esta forma construir un tipo de “bosquejo”, donde no se tenía mucho nivel de detalle, puesto que en el proceso de desarrollo se realizaron diferentes modificaciones, aunque a partir de esta herramienta se trazó el camino que se debía seguir.

El modelado basado en casos de uso, comprende efectividad y simplicidad en un mismo contexto, donde es posible estructurar los requerimientos del sistema desde el punto de vista del usuario. Desde estos diagramas es posible definir el comportamiento de un sistema de manera clara, coherente y fácil de comunicar.

La Figura 4.4 presenta el diagrama de casos de uso propuesto, donde se definen los requisitos del usuario al usar la interfaz.

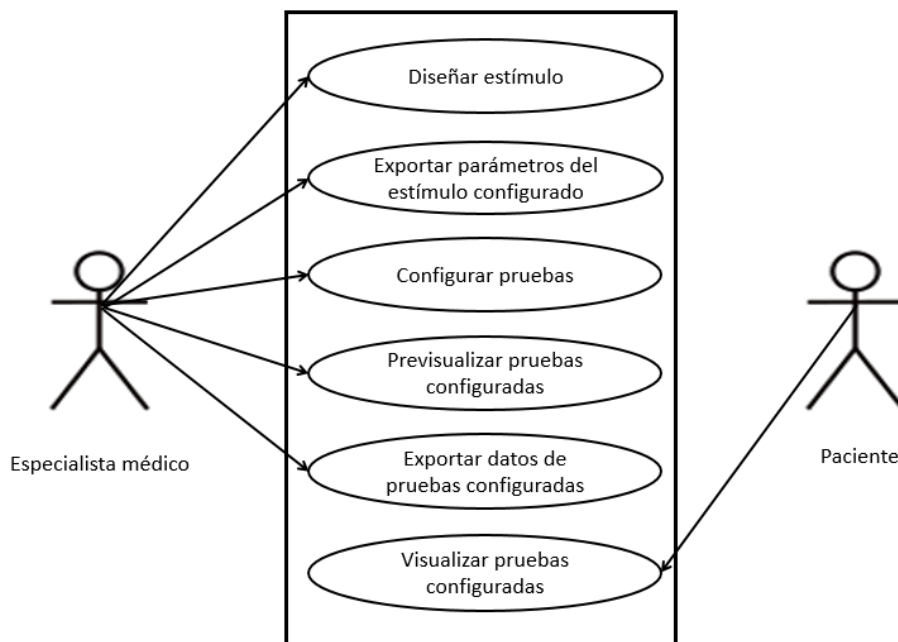


Figura 4.4: Diagrama de casos de uso de la interfaz

Las elipses representan los casos de uso, que especifican una funcionalidad, luego en el recuadro se incluyen las partes del sistema y los casos de uso que lo conforman.

El diagrama de la Figura 4.5 representa los casos de uso extendido, utilizado para el análisis y diseño del sistema, en este se describe la información que se va generando, en resumen las entradas al sistema, los procesos que ocurren dentro de él, y las salidas que se generan.

Las relaciones entre los casos de uso, se distinguen con una flecha de navegación que apunta hacia el actor que inicializa la interacción con la interfaz, como se observa en la Figura. Si no hay alguna flecha asociativa indica que no hay interacción entre el actor y el caso de uso.

Este diagrama indica cómo se relacionan los casos de uso, se observa que el actor únicamente interactúa sobre lo que se definió como funcionalidad del programa, para realizar configuraciones, donde se desprenden tareas adicionales de cada caso de uso. Esto permite que el sistema propuesto sea flexible y modular, facilitando la extensión de sus funciones, que implicaría incluir nuevos casos de uso lo cual es de absoluta transparencia para el usuario.

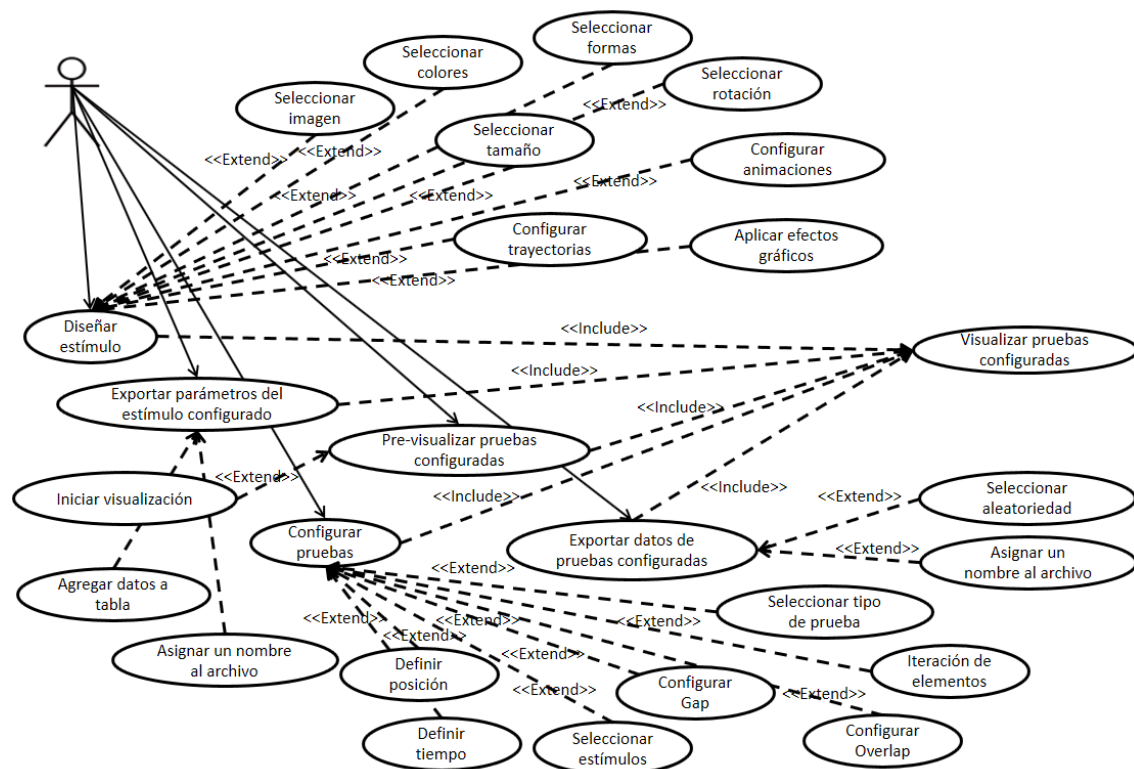


Figura 4.5: Diagrama de relaciones entre casos de uso de la interfaz

4.5. Interfaz desarrollada

Antes de mostrar los resultados del trabajo, es importante mencionar que la herramienta utilizada para el desarrollo de las interfaces fue *Qt*. Este software es ampliamente utilizado para desarrollar aplicaciones con interfaz gráfica de usuario. *Qt* tiene tres tipos de licencias para su uso, una gratuita (GPLv2/v3) para el desarrollo de aplicaciones de código abierto y libre, una licencia de pago (QPL) para el desarrollo de aplicacio-

nes comerciales y una licencia gratis (LGPL) pensada para aplicaciones comerciales. Los sistemas operativos con los que es compatible son: Windows, Mac OS X, Linux/X11 y Solaris. Además, es compatible con sistemas móviles como Android, iOS y Windows Phone.

En la librería *Qt* se utilizan principalmente secuencias de código en C++, aunque con la distribución del entorno de programación *Qt Creator* se permite la integración de diferentes lenguajes, como por ejemplo: QML, Python, HTML, Ruby, PHP, C# entre otros. Esta característica permite una mayor flexibilidad para desarrollar el código que se ejecuta bajo la interfaz. *Qt Creator* dispone de una extensión denominada *Qt Designer*, donde es posible crear interfaces de forma gráfica, aquí se encuentran diversas herramientas, donde se pueden seleccionar módulos para construir las aplicaciones.

La versión principal del software que contiene la plataforma *Oscann desk100* ha sido desarrollada en *Qt Creator*, por esta razón, este trabajo se realiza en dicha herramienta. Es importante resaltar que durante la implementación de la interfaz no se utilizó la extensión *Qt Designer*, debido al uso de componentes diseñados especialmente para la aplicación de *Oscann desk100*, los cuales se encuentran descritos en archivos QML y no se pueden visualizar gráficamente en la construcción de la interfaz, por lo cual la implementación se realizó completamente en el editor de código de *Qt Creator*.

Ahora bien, teniendo claridad sobre la herramienta utilizada para el desarrollo de este trabajo, a continuación, se muestran los resultados obtenidos en la implementación de la interfaz, buscando dar cumplimiento a los objetivos del proyecto en función de los criterios de validación de los requerimientos impuestos por la empresa *Aura Innovative Robotics*. Como se mencionó al inicio de esta sección, la interfaz se divide en dos secciones, por esta razón, la descripción de cada sección se realiza de manera individual.

4.5.1. Interfaz para configuración y diseño de pruebas

N°	Position X (Deg)	Position Y (Deg)	Initial Time (ms)	Visible time (ms)	Test Duration (seg)	Gap (ms)	Overlap (ms)	Countermanding (ms)	Reference	Stimulus Form	Type
1	0	0	0	2000	2.0	0	0	0	Null	estimulo1	Central fixation
2	12	0	2000	2000	4.0	0	0	0	Null	star0	Pro-saccade
3	0	0	4000	2000	6.0	0	0	0	2	estimulo1	Central fixation
4	0	0	6000	5000	11.0	0	0	0	Null	ordelLineal	Smooth Pursuit
5	0	0	11000	2000	13.0	0	0	0	Null	estimulo1	Central fixation
6	12	0	13000	2000	15.0	0	0	0	Null	star0	Pro-saccade
7	0	0	15000	2000	17.0	0	0	0	6	estimulo1	Central fixation
8	0	0	17000	5000	22.0	0	0	0	Null	ordelLineal	Smooth Pursuit

Figura 4.6: Sección de configuración de pruebas

En la Figura 4.6 se muestra la ventana principal de la sección de configuración de pruebas. En esta se existen 3 componentes principales, la indicada con el número 1 es la tabla de visualización de datos, la número 2 es la sección de configuración de parámetros y finalmente en la sección número 3 se encuentran las funciones auxiliares para modificar datos de la prueba o visualizar los experimentos.

Test Configuration											
Nº	Position X (Deg)	Position Y (Deg)	Initial Time (ms)	Visible time (ms)	Test Duration (seg)	Gap (ms)	Overlap (ms)	Countermanding (ms)	Reference	Stimulus Form	Type
1	0	0	0	2000	2.0	0	0	0	Null	estimulo1	Central fixation
2	12	0	2000	2000	4.0	0	0	0	Null	star0	Pro-saccade
3	0	0	4000	2000	6.0	0	0	0	2	estimulo1	Central fixation
4	0	0	6000	5000	11.0	0	0	0	Null	circleLineal	Smooth Pursuit
5	0	0	11000	2000	13.0	0	0	0	Null	estimulo1	Central fixation
6	12	0	13000	2000	15.0	0	0	0	Null	star0	Pro-saccade
7	0	0	15000	2000	17.0	0	0	0	6	estimulo1	Central fixation
8	0	0	17000	5000	22.0	0	0	0	Null	circleLineal	Smooth Pursuit

Figura 4.7: Tabla de visualización de datos para configuración de pruebas

En la Figura 4.7 se puede apreciar de mejor manera la tabla de datos de esta sección. En ella se pueden encontrar los siguientes apartados:

- *Position X*: En esta columna se muestran las posiciones para el eje X, describe el lugar donde se desplaza el estímulo de la fila correspondiente.
- *Position Y*: Al igual que el anterior, se consignan las posición para el eje Y donde se desplaza el estímulo. Los valores cambian únicamente para test de movimientos sacádicos y fijaciones, para otro tipo de test el valor por defecto es 0, tanto para X como para Y. Las posiciones se ingresan en grados con un rango de -20 a 20 para el eje X y de -12 a 12 para el eje Y.
- *Initial Time*: En esta columna se consigna el tiempo desde el cual el estímulo inicia su desplazamiento. Este valor se calcula internamente a medida que se van ingresando los elementos a la prueba. Este tiempo esta dado en milisegundos.
- *Visible Time*: Este corresponde al valor de tiempo ingresado en la configuración de cada elemento de la prueba. En la sección de parámetros se encuentra un apartado para ingresar este valor en milisegundos.
- *Test Duration*: En esta columna se muestra el tiempo total de la prueba, este se va actualizando conforme se agregan elementos al experimento y esta dado en segundos.
- *Gap/Overlap/Countermanding*: Cuando se habilita algún paradigma (*Gap/Overlap*) o se define un test de sacadas tipo *Countermanding*, se ingresa un valor de tiempo en milisegundos en el apartado de parámetros que se puede visualizar en esta columna.
- *Rerefence*: En esta columna se consignan los números asociados al orden de presentación de la prueba. Cuando hay un valor numérico (diferente a *Null*) quiere decir que durante la presentación del experimento, el estímulo de esa fila se mostrara **después** del elemento correspondiente al número de la casilla *Reference*.
- *Stimulus Form*: En esta columna se muestran los nombres de los estímulos utilizados para cada elemento de la prueba.

- *Type*: Esta sección permite definir el tipo de test al cual pertenece cada elemento de la prueba.

Con la implementación de esta tabla se busca dar cumplimiento a uno de los criterios de validación de la historia de usuario número 1 (Tabla 4.1), correspondiente al requerimiento de los test disponibles en la interfaz. En la Figura 4.7 también se puede observar que la combinación de test en una prueba es posible, esto se ve reflejado en la columna “*Type*” donde se puede observar elementos de distintos test para un misma prueba. Esta funcionalidad busca dar cumplimiento a los criterios de validación para el requerimiento sobre la combinación de test en una prueba, consignados en la Tabla 4.6 en la sección de Historias de usuario.

En la Figura 4.8 (marcado en rojo) se puede observar los botones que permiten eliminar elementos de la prueba. El botón que contiene el signo menos (-), elimina la última fila de la tabla de datos, es decir, el último elemento ingresado. Las instrucciones (código) que contiene este botón permiten actualizar los valores como la duración actual de la prueba, el apartado de referencias (“*Reference*”), entre otros. Mientras que el botón con el símbolo de la caneca de basura, elimina la totalidad de la prueba, es decir, limpia los elementos de la tabla de datos. Estos botones pertenecen a la sección 3 indicada en la Figura 4.6.

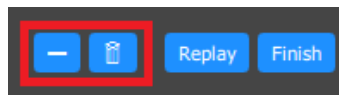


Figura 4.8: Botonería auxiliar para la configuración de pruebas

Por otra parte, el botón con la palabra “*Replay*” que también se observa en la imagen 4.8, cumple con la función de copiar los datos de la prueba que se encuentran en la tabla descrita anteriormente. Cuando se presiona este botón se despliega una ventana (ver Figura 4.9) que solicita al especialista ingresar el número de veces que quiere repetir los datos actuales de la tabla. Resulta adecuado contar con una función así cuando la cantidad de elementos de una prueba es muy grande o si el especialista médico considera que una sección de la prueba debe ser presentada nuevamente. De esta forma el usuario (especialista) no realiza un trabajo doble definiendo elementos que ya existen en la prueba.

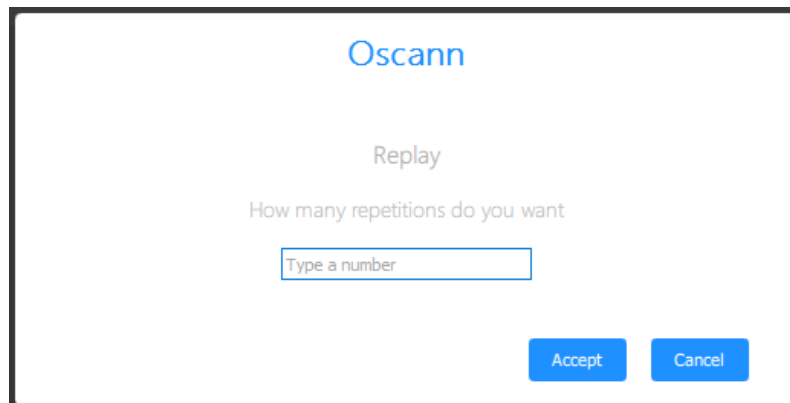


Figura 4.9: Cuadro de diálogo para repetición de elementos

Con esta funcionalidad se busca cumplir con los criterios de validación para el requerimiento de iteración de elementos en la configuración de pruebas, consignados en la Tabla 4.7 de la sección 4.3 de este capítulo, Historias de usuario.

Ahora bien, el botón “*Finish*” (que se observó en la Figura 4.8) permite asignar un nombre a la prueba cuando el especialista ha terminado de configurar la misma. Al presionar este botón se despliega una ventana (ver Figura 4.10) para ingresar el nombre deseado y además, permite definir si la presentación de los elementos de la prueba se hace de manera aleatoria o en el orden en que fueron agregados. Cuando el especialista marca la casilla “*Random Test*”, se agrega un dato auxiliar en el archivo XML de la prueba diseñada, este dato le dice a la plataforma *Oscann* el modo de presentación de la prueba, por esta razón, el modo de visualización aleatoria no está disponible en esta interfaz, la función en este trabajo es únicamente para definir el modo de presentación del experimento en la plataforma *Oscann*. Con esta funcionalidad se busca dar cumplimiento a uno de los criterios de validación del requerimiento de aleatoriedad en la presentación de pruebas, indicados en la Tabla 4.9 de la sección de Historias de usuario, además de los criterios de validación para el requerimiento de exportación de datos en archivos XML, consignados en la historia de usuario de la Tabla 4.12.

Figura 4.10: Cuadro de diálogo para finalizar con la configuración de pruebas

Continuando con la descripción de la interfaz para la configuración de pruebas, en la Figura 4.11 se muestra la sección de parámetros de los test disponibles en la herramienta. En esta se pueden diferenciar 4 diferentes apartados los cuales corresponden a los test de Sacadas visualmente guiadas o movimientos sacádicos (*Visual Guided Saccades*), Seguimiento o persecución lenta (*Smooth Pursuit*), Reflejo pupilar (*Pupil light*) y Fijaciones (*Fixation*). A continuación, se describen cada uno de los apartados mencionados.

- *Visual Guided Saccades*: En este apartado se configuran los test de sacadas, antisacadas, sacadas de memoria, antisacadas de memoria y sacadas de tipo *Countermanding*. Los parámetros principales son los de posición (X,Y) y tiempo en el cual el estímulo se desplaza a dicha posición (“*Duration*”). Con el control *Pro-saccade/Anti-saccade* se define si el elemento es de tipo sacadas o antisacadas respectivamente. Con el control *Visible/No visible*, se define si el estímulo es de tipo normal o de memoria respectivamente. Con el control *Test/Demo* se puede definir si el estímulo se presentará en modo normal o modo demostración respectivamente. Para elegir un paradigma *Gap* u *Overlap*, se marca la casilla correspondiente y

The image shows a software interface for configuring different tests. It is divided into four main sections:

- Visual Guided Saccades:** This section includes a table for test parameters:

	X	Y	Dur.	Gap	Ovlp.	Count.	Stimulus Form
Fixation:	0	0	2500				circle0
Target:	2	10	2500	0	0	0	circle0
Reference:	Null						

 Below the table are checkboxes for 'Gap', 'Overlap', and 'Countermanding', each with a 'ms' input field. There are also buttons for 'Set fixation time', 'Preview', and a '+' sign.
- Smooth Pursuit:** Includes 'Stimulus' (circleLineal), 'Reference' (Null), 'Duration' (ms), and 'Gap' (ms) fields. It has 'Preview' and '+' buttons.
- Pupil light:** Includes 'Stimulus' (bw), 'Reference' (Null), 'Duration' (ms), and 'Gap' (ms) fields. It has 'Preview' and '+' buttons.
- Fixation:** Includes 'Stimulus' (circle0), 'Reference' (Null), 'Horizontal' (Vertical), 'Oblique' checkbox, 'Position X' (-20° to 20°), 'Position Y' (0), 'Duration' (ms), and 'Gap' (ms) fields. It has 'Preview' and '+' buttons.

Figura 4.11: Parámetros de configuración de los distintos test disponibles en la interfaz

se ingresa un valor de tiempo deseado, de igual manera se realiza la configuración para test de tipo *Countermanding*. En la siguiente sección se encuentran las listas de estímulos disponibles para este tipo de pruebas (*Stimulus form*).

- *Smooth Pursuit / Pupil light*: En estos apartados se elije primero el estímulo que se desea utilizar para el test (*Stimulus*). Cuenta con un apartado para ingresar la duración del test y una opción para habilitar un *delay* entre cada cambio de test, es decir un espacio entre cada inicio de una sección de la prueba (*Gap*).
- *Fixation*: Para este tipo de test, se definieron parámetros similares a los de sacadas visualmente guiadas, como posición, duración y paradigma *Gap*.

Por otro lado, el botón con la palabra “*Preview*” presente en todas las secciones de los test, permite visualizar el comportamiento únicamente del elemento que se está diseñando en ese momento, es decir, la función toma los valores de la sección de parámetros y despliega una ventana donde se visualiza el elemento ejecutando las indicaciones ingresadas. Con el botón que contiene el signo (+), se agregan los datos a la tabla de la prueba. Con el apartado “*Reference*”, se elije el elemento con el cual se quiera establecer un orden específico en la presentación de la prueba.

Cabe resaltar que la lista de estímulos para cada test es diferente, los estímulos se clasifican en función de la extensión que hayan recibido en el diseño de estímulos. Los únicos test que comparten lista de estímulos son los de sacadas visualmente guiadas y los de fijaciones.

La implementación anterior busca cumplir con los criterios de validación restantes del requerimiento sobre los test disponibles en la herramienta (ver Tabla 4.1 en la sección 4.3 Historias de usuario). También se busca dar cumplimiento a los criterios de validación del requerimiento sobre la inclusión de los paradigmas Gap y Overlap para la configuración de movimientos sacádicos especiales (ver Tabla 4.3 en la sección 4.3 Historias de usuario).

Para finalizar con la descripción de esta sección de la interfaz, se muestra el apartado de previsualización. En la Figura 4.12, se puede apreciar la ventana de inicial de esta función, la grilla que se observa brinda una guía para comprobar la posición del estímulo.

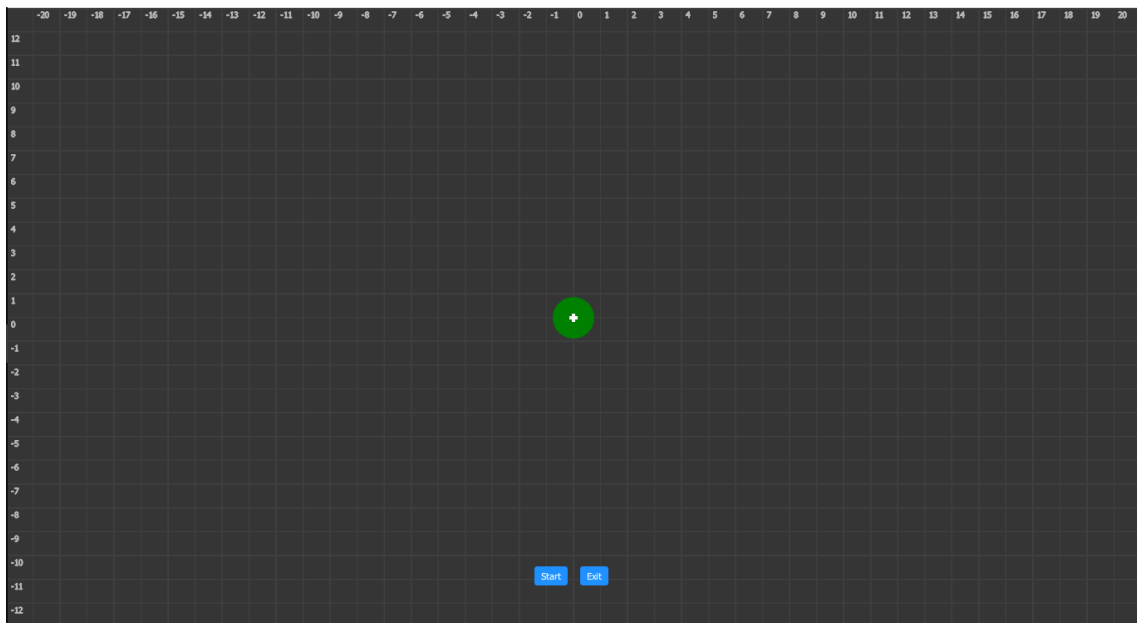


Figura 4.12: Ventana inicial de previsualización

Además de la previsualización de elementos individuales, esta función permite observar el comportamiento completo de la prueba, es decir, toma los datos consignados en la tabla y muestra una presentación general de la prueba. Con el botón que se muestra en la Figura 4.13, se ejecuta la ventana de previsualización para observar la prueba completa. Este se encuentra en la sección número 3 de la ventana principal de la interfaz.

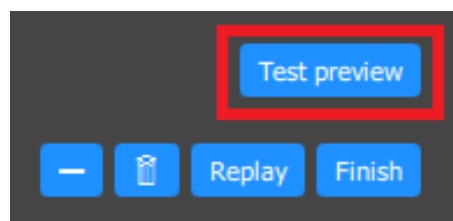


Figura 4.13: Botón para ejecutar previsualización de pruebas

El funcionamiento de la ventana es muy sencillo, una vez que se presiona el botón *Test preview*, se muestra la ventana de la Figura 4.12, en esta se encuentran dos botones, *Start* para ejecutar la presentación y *Exit* para salir de la ventana.

Otro aspecto importante que se vinculó con la previsualización es el modo demostración. El uso de esta propiedad es muy sencillo, para empezar, este modo solo se encuentra disponible para test de movimientos sacádicos. Para definir un estímulo como demostración, en la sección de parámetros se configuró una opción para habilitar o deshabilitar esta propiedad. En la Figura 4.14, se observa una secuencia de un test de antisacadas, se puede apreciar cómo actúa el modo demostración para indicarle al observador el área de la ventana que debe mirar, marcado con un círculo blanco.

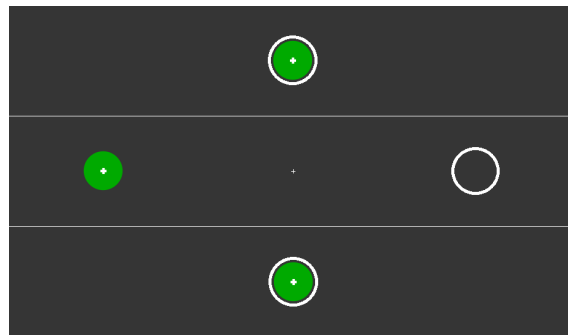


Figura 4.14: Modo demostración en un movimiento antisacádico

Con esta última implementación se busca dar cumplimiento a los criterios de validación del requerimiento sobre la previsualización de pruebas, condiciones consignadas en la historia de usuario de la Tabla 4.11.

4.5.2. Interfaz para el diseño de estímulos

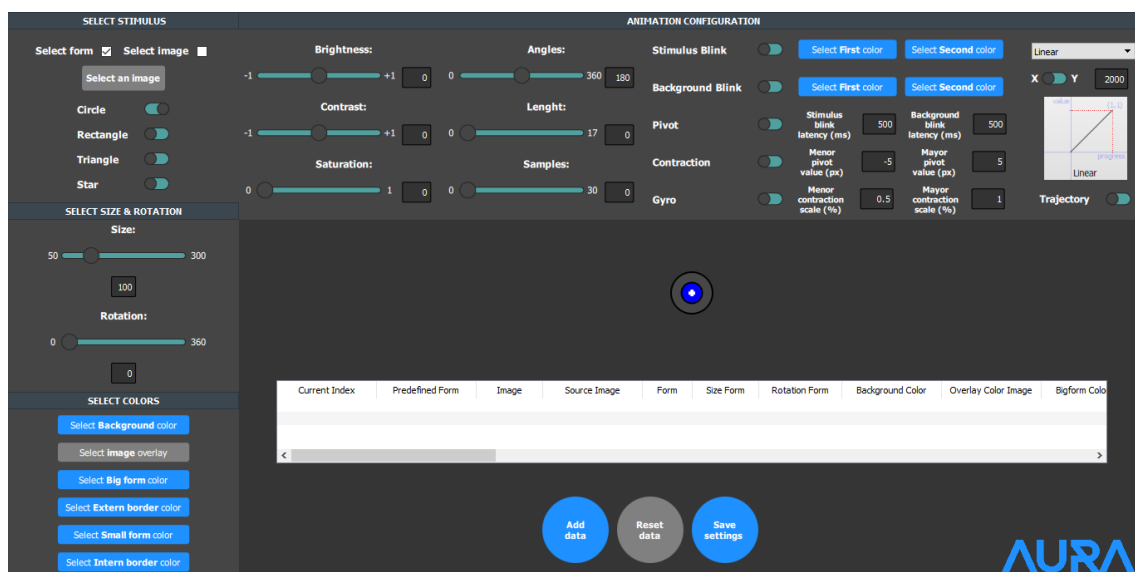


Figura 4.15: Interfaz para el diseño de estímulos.

Como resultado de la ejecución en cada uno de los *Sprint*, con los requerimientos funcionales basados en las historias de usuario y en los casos de uso, se obtuvo la interfaz para el apartado de configuración del estímulo que se muestra en la Figura 4.15.

Es importante reconocer cada una de las secciones con las que cuenta esta ventana, donde se da cumplimiento a los criterios de validación de las historias de usuario, por lo tanto se describe cada una de ellas a continuación.

Inicialmente se muestra en la Figura 4.16, la sección que permite escoger las formas geométricas, donde al activar o desactivar los *Switch* se representa el objeto correspondiente. Luego se encuentra un botón que permite desplegar un cuadro de dialogo que se muestra en la Figura 4.17, el cuál admite la carga de una imagen de archivo JPG o PNG como representación del estímulo.

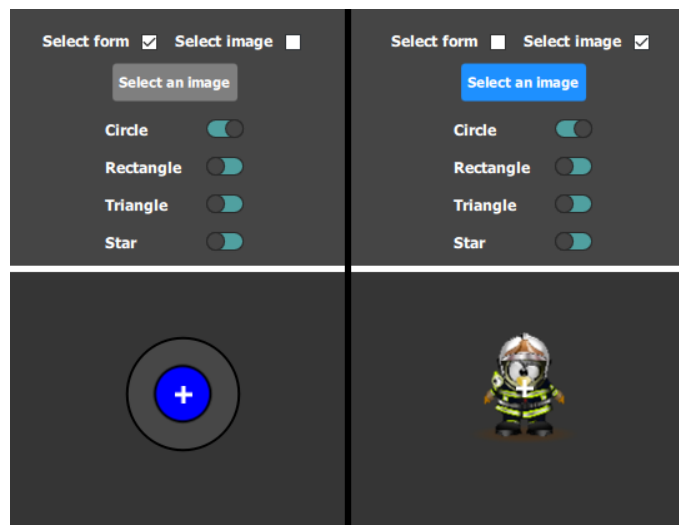


Figura 4.16: Sección para seleccionar el aspecto del estímulo basado en formas geométricas o imágenes.

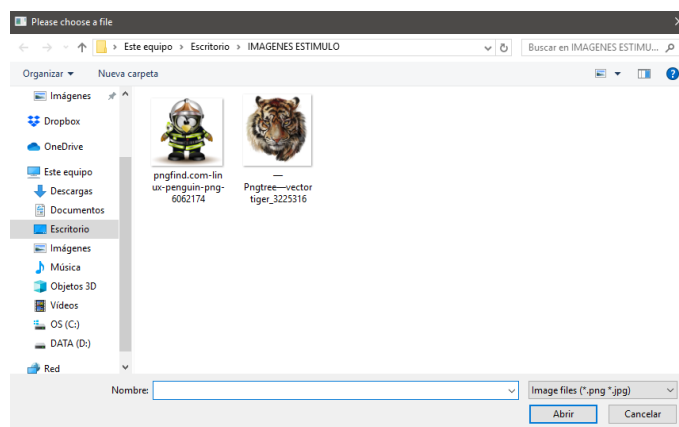


Figura 4.17: Cuadro de dialogo auxiliar para carga las imágenes en formato JPG o PNG.

Para los cambios de escala y orientación, fue reservada la sección que se muestra en la Figura 4.18, la cual contiene dos *Slider*, donde seleccionando el control se puede

variar los valores entre 50 a 300 píxeles para el tamaño y 0 a 360 grados para la rotación, además de verse reflejados en dos *TextView*, en los cuales también es posible modificar estos parámetros a partir del ingreso de texto por parte del usuario.

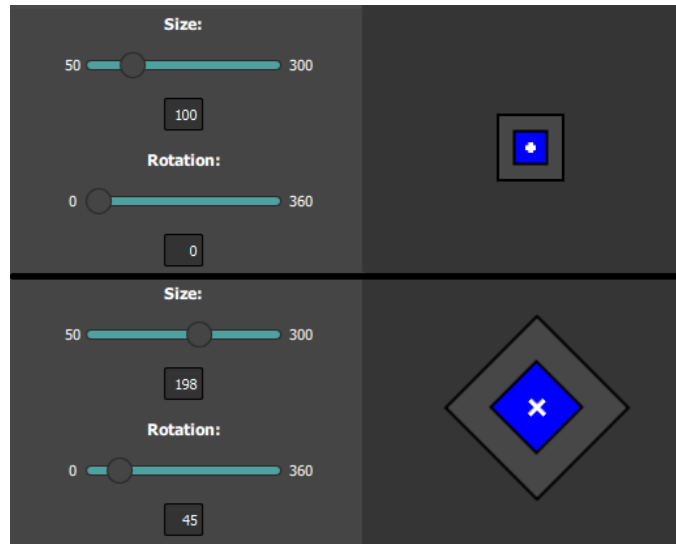


Figura 4.18: Sección para modificar valores de tamaño y rotación.

Luego se cuenta con la sección que se muestra en la Figura 4.19, que contiene diferentes botones para desplegar un cuadro de diálogo como el de la Figura 4.20, con la paleta de colores que permite elegir los tonos presentes en el estímulo y en el fondo de la aplicación, en la Figura 4.19, se muestran diferentes botones ya que el estímulo basado en formas geométricas cuenta con cuatro áreas que son denominadas de la siguiente manera: área interna, área externa, borde interno y borde externo. también se encuentran dos botones adicionales a los mencionados, que permiten elegir los tonos del fondo de la aplicación y el que se sobrepone a las imágenes convirtiéndolas en siluetas.

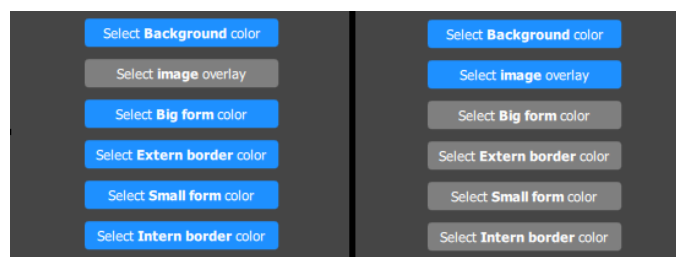


Figura 4.19: Sección para modificar tonos en el estímulo y el fondo de la aplicación.

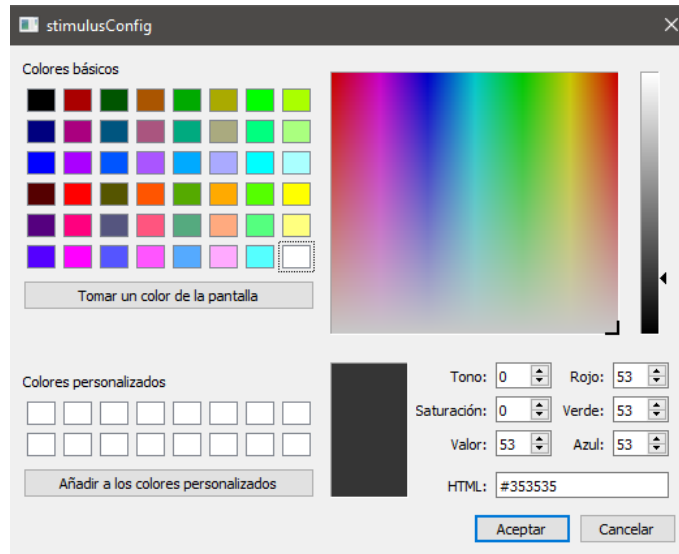


Figura 4.20: Paleta de colores empleada en la interfaz

En la historia de usuario numero 5 también se describe un requerimiento sobre la aplicación de efectos gráficos al estímulo, para lo cuál se implemento la sección que se muestra en la Figura 4.21, donde se cuenta con 3 *Slider* y 3 *TextView*, para realizar variaciones en el brillo, contraste y saturación.

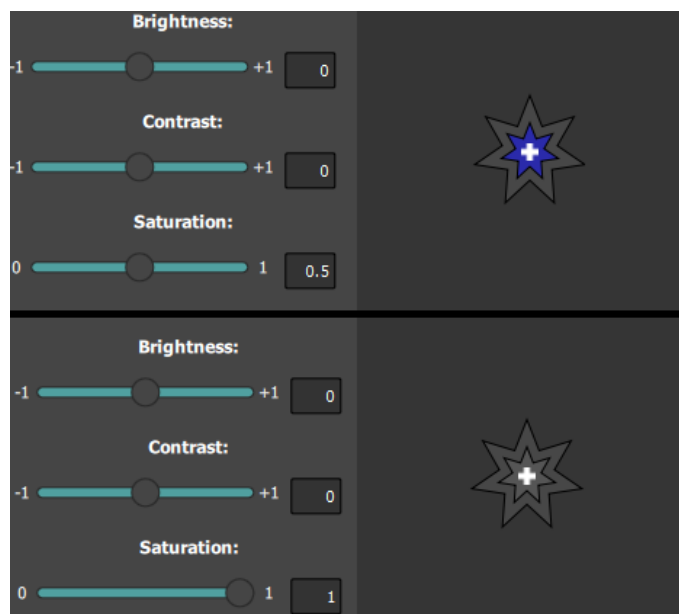


Figura 4.21: Sección para modificar brillo, contraste y saturación.

Luego se encuentran los otros 3 *Slider* y 3 *TextView*, como en la Figura 4.22, que se utilizan para el efecto de borrosidad o difuminado, donde se asocian los siguientes parámetros: ángulo, longitud y numero de muestreos.

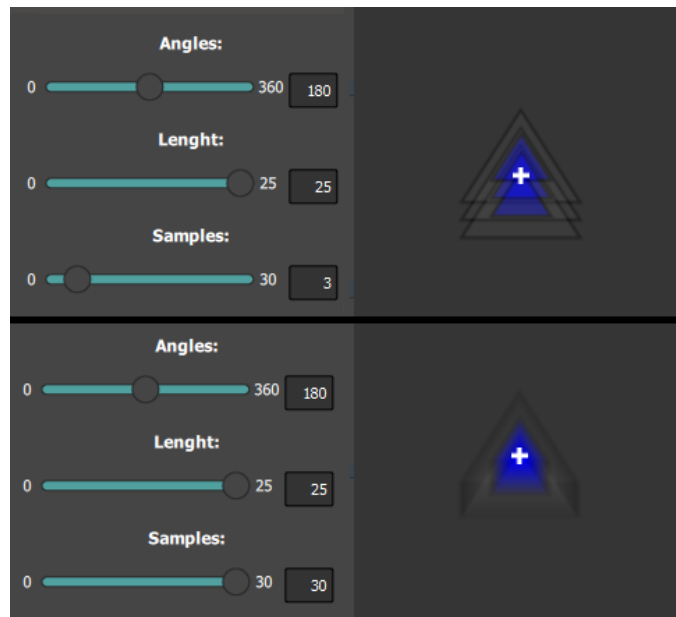


Figura 4.22: Sección para aplicar el efecto de borrosidad o difuminado.

Continuando con la descripción de las secciones que se implementaron en la interfaz para el diseño del estímulo, se muestra en la Figura 4.23, el área de configuración de animaciones correspondientes a: parpadeo en el fondo de la aplicación y parpadeo, contracción, pivote y giro del estímulo, donde se cumple con lo establecido en la historia de usuario numero 8.

Es necesario definir la funcionalidad de cada uno de los elementos disponibles en esta sección, inicialmente se encuentran 5 *Switch* que permiten activar o desactivar los comportamientos descritos, luego es posible seleccionar los dos colores entre los que se realiza la transición en los parpadeos del fondo de la aplicación y del estímulo, luego se encuentran dos *TextView* que permiten visualizar y modificar la latencia de dichas transiciones. Después aparecen 4 *TextViews*, de los cuales, los dos primeros sirven para establecer el valor máximo y mínimo del pivote, y los dos restantes para modificar la máxima y mínima escala de la contracción.

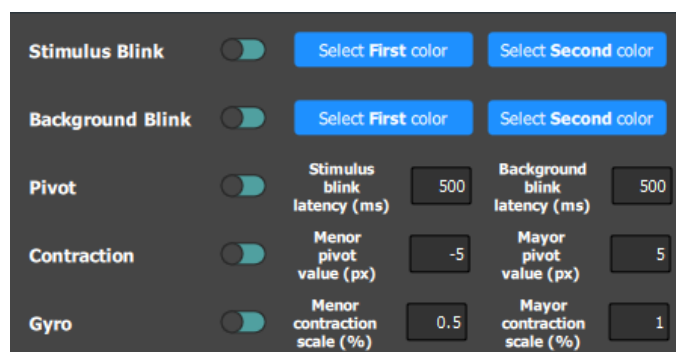


Figura 4.23: Sección para configurar animaciones.

La penúltima sección de la interfaz para configurar el estímulo se muestra en la Figura 4.24, donde están disponibles los controles que permiten realizar los movimientos del

estímulo basados en trayectorias, esto con el fin de complementar las pruebas de seguimiento lento, en esta área se dispone de dos *Switch* que sirven para activar o desactivar los movimientos de trayectoria y elegir el eje sobre el cual se realizaran los desplazamientos, también esta disponible una lista en la que se puede escoger el comportamiento de las trayectorias con base a una gráfica, la cual es representada en una imagen que se encuentra debajo de dicho elemento, para concluir con esta sección, se encuentra un *TextView* que permite visualizar y modificar el tiempo que se tarda el estímulo en realizar la mitad de un periodo en un desplazamiento.

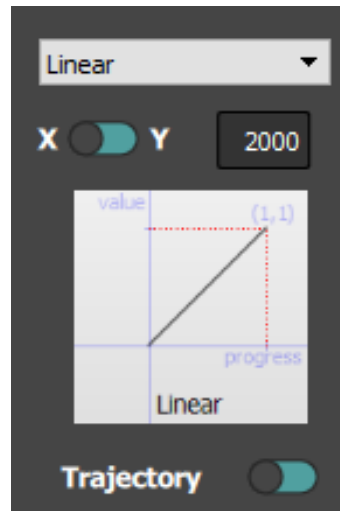


Figura 4.24: Sección para configurar las trayectorias del estímulo.

Finalmente para la exportación de datos en la interfaz de diseño del estímulo, se ha reservado la zona inferior para implementar esta sección, que se muestra en la Figura 4.25, donde se visualiza una tabla acompañada por tres botones que tienen una distinta funcionalidad, el primero de ellos agrega datos a la tabla para que el usuario pueda visualizarlos y verificarlos antes de realizar la exportación, el segundo permite restablecer los parámetros y las configuraciones por defecto del estímulo, y el ultimo para exportar los datos.

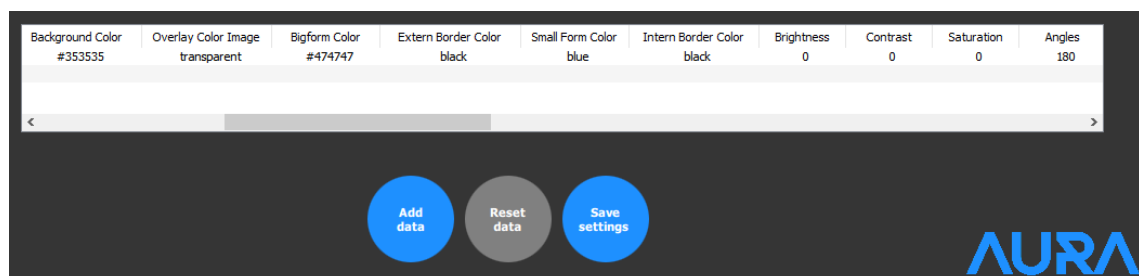


Figura 4.25: Tabla para visualizar los datos configurados, junto con los botones que permiten la exportación y restablecimiento de valores por defecto.

Capítulo 5

Resultados

En este capítulo se describe el procedimiento para la validación del trabajo realizado, el cual se lleva a cabo a partir de la apreciación de diferentes especialistas en el ámbito neurológico y la posterior discusión de estos resultados.

5.1. Proceso de validación de la interfaz

Como se definió en el objetivo específico número 3 del proyecto, la validación del trabajo se realiza a través de la experiencia de usuario con la interfaz. Inicialmente, se planteó la posibilidad de realizar estas pruebas directamente con la plataforma *Oscann*, una vez se haya integrado la interfaz desarrollada en este trabajo, de esta manera, se podría evaluar el desempeño del usuario frente a una prueba real con pacientes reales. Sin embargo, este proceso no se realizó de esta manera debido al estricto control del licenciamiento software para procedimientos médicos en España. La reglamentación indica que el control de versiones y actualización software de equipos médicos debe ser supervisado y aprobado por el Ministerio de Sanidad. Este es un proceso que tarda mucho tiempo y por esta razón resulta muy complicado proceder de esta manera.

Pese a lo anterior, las pruebas se realizaron con los miembros del equipo de trabajo de *Aura* y otros profesionales de la salud, especialistas en el ámbito neurológico y clínico. Las personas que participaron en la validación del trabajo, realizaron una exploración general de la herramienta (interfaz). Posteriormente, diligenciaron un formulario para conocer sus impresiones acerca del funcionamiento, navegabilidad, desempeño, terminología, gráficos, entre otros. En el Anexo C[6.2], se encuentra el formulario utilizado.

5.1.1. Perfiles de usuario

El público objetivo son profesionales en el área de la salud, específicamente, neurólogos, psicólogos con énfasis en neurología u oftalmólogos especialistas en Neuro-oftalmología. También, está dirigido a investigadores de cualquier rama profesional, interesados en el diagnóstico de enfermedades neurodegenerativas.

A continuación, se presentan las personas que participaron en la validación de este trabajo.

- **Dra. Cecilia García Cena**, doctora en Robótica y Automatización de la Universidad

Politécnica de Madrid, docente de la misma, co-fundadora de la empresa Aura y quien formó parte del equipo de desarrollo de la plataforma *Oscann*.

- **Dr. Erik Hernández**, doctor en Robótica y Automatización de la Universidad Politécnica de Madrid, miembro del equipo de trabajo de Aura como Director de Tecnología y quien supervisó constantemente el desarrollo del proyecto.
- **Dra. Marie Destarac**, miembro del equipo de trabajo de Aura, quien realiza actividades vinculadas al control de calidad y gestión de proyectos. Recientemente recibió su título de doctora en Robótica y Automatización de la UPM.
- **Dra. Viviana Martínez**, neuróloga clínica del Hospital Universitario Departamental de Nariño.
- **Dra. Mónica Rosero**, psicóloga clínica del Hospital Universitario Departamental de Nariño.
- **Dr. Luis Fernando Casanova**, neurocirujano del Hospital Universitario Departamental de Nariño.
- **Dr. Ricardo Moreno**, neurólogo clínico del Hospital Universitario Departamental de Nariño.

La opinión de las personas que hacen parte de la empresa resulta de gran importancia para validar el trabajo realizado, ya que ellos finalmente son los usuarios principales de la interfaz. Por otra parte, la participación de los profesionales en salud permite conocer una apreciación diferente, relacionada con el trabajo de campo.

5.2. Protocolo

El procedimiento para realizar la exploración de la interfaz fue el siguiente.

1. Inicialmente se explicó a los participantes el tema del trabajo y la finalidad de este, de esta manera se da contexto a los conceptos que se pueden ver en la interfaz.
2. Posteriormente, se realizó una demostración del funcionamiento de la interfaz explicando las dos secciones de la misma. En la demostración se muestra el procedimiento para crear estímulos y como utilizarlos en la sección de configuración de pruebas.
3. Una vez terminada esta pequeña capacitación, y después de aclarar las dudas presentadas, se le pidió al participante que utilice la interfaz para diseñar un estímulo y crear una prueba en la que utilice el estímulo diseñado.
4. Por último, finalizada la etapa de exploración de la herramienta, se le pidió al participante que conteste el cuestionario para conocer sus impresiones acerca de la interfaz.

Pese a no contar con el dispositivo *Oscann desk100* para complementar el uso de la interfaz desarrollada en este trabajo, los especialistas mencionados anteriormente aceptaron participar en la validación de este trabajo considerando las limitaciones del caso. La herramienta claramente se llevó hasta ellos, al Hospital Universitario Departamental de Nariño, en la ciudad de Pasto y en el cual se organizaron dos reuniones, la primera para socializar el trabajo y conocer si estaban dispuestos a participar, de esta manera se cumplía con el primer paso del protocolo. Posteriormente, si el especialista accedía a participar en esta validación, en la segunda reunión los participantes realizaron la exploración de la herramienta y dieron a conocer sus impresiones a través del cuestionario formulado al final del procedimiento.

5.3. Resultados

A continuación, se muestran los resultados obtenidos para cada una de las preguntas planteadas en el formulario con su respectiva interpretación. Cabe resaltar que el cuestionario se divide en 4 partes, la inicial corresponde a reacciones generales del sistema, la segunda trata sobre la terminología utilizada en la interfaz, la siguiente se centra en el aprendizaje de la herramienta y por último se plantea un apartado de características del sistema como velocidad y fiabilidad de la interfaz.

5.3.1. Reacciones generales del sistema

En esta sección se plantean 4 apreciaciones para calificar de una manera cualitativa el trabajo realizado.

En la gráfica presentada en la Figura 5.1, se puede observar que la impresión general para los participantes es muy buena, en una escala de 0 a 9, donde 0 corresponde a “Terrible” y 9 a “Estupendo”, el valor promedio es de 8.14, lo que permite afirmar que para la mayoría de los participantes la aplicación les parece estupenda.

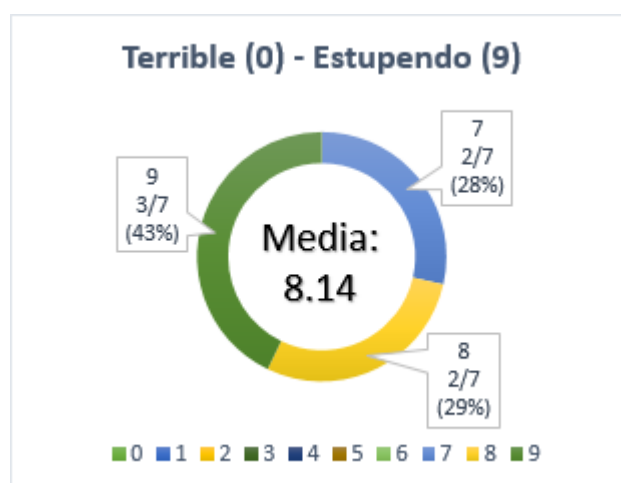


Figura 5.1: Resultados 1.1: Terrible - Estupendo

La Figura 5.2, muestra la apreciación de los participantes respecto a la complejidad del sistema, se puede observar que la tendencia se inclina hacia lo intuitivo con un valor

promedio de 7.14. En este caso, el promedio cae un poco debido a la experiencia previa con este tipo de herramientas por parte de los médicos participantes. Otra razón aparente puede ser el área de investigación para la cual se diseñó la interfaz, específicamente, la relación de los movimientos oculares con enfermedades neurodegenerativas.

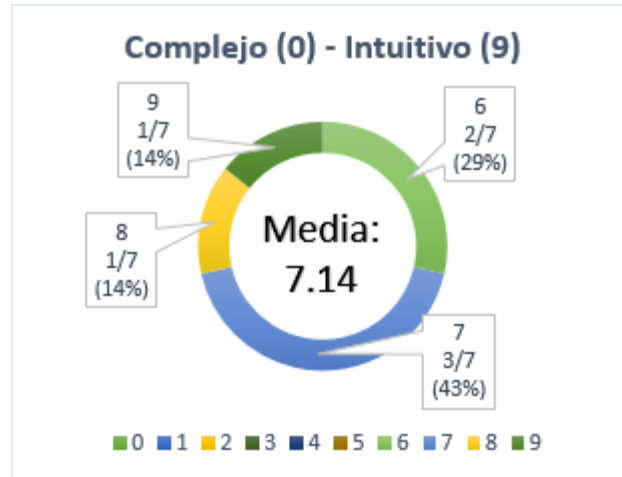


Figura 5.2: Resultados 1.2: Complejo - Intuitivo

Si bien, la exploración que realizaron los participantes no exigía alguna tarea específica, la experiencia con la interfaz resulta muy satisfactoria, esto se puede evidenciar en la Figura 5.3, la cual muestra los resultados frente a este concepto de satisfacción. Con un valor promedio de 8.42 se puede interpretar que los participantes se inclinan por una experiencia satisfactoria con la interfaz. Esta apreciación resulta muy relevante para la validación del trabajo debido al perfil de los participantes.

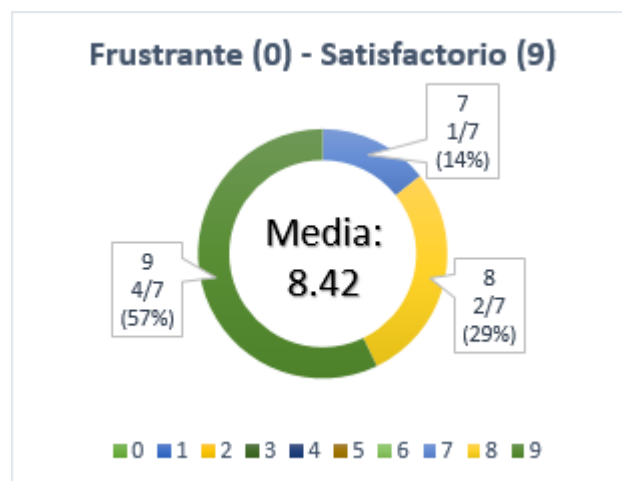


Figura 5.3: Resultados 1.3: Frustrante - Satisfactorio

En la Figura 5.4 se muestran los resultados frente al concepto de flexibilidad de la interfaz. En esta se puede apreciar una mayor deferencia en el criterio de los participantes, frente a los anteriores conceptos. Sin embargo, la opinión se inclina por calificar al sistema como "Flexible" y como se observa en la gráfica, 5 de 7 personas concuerdan con este

concepto. Aun cuando los datos se encuentran más dispersos, el valor promedio es de 8.14, siendo un número considerable para calificar el sistema como flexible.

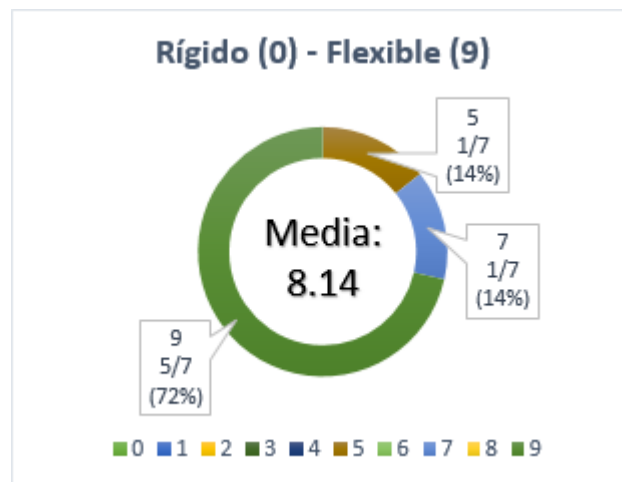


Figura 5.4: Resultados 1.4: Rígido - Flexible

5.3.2. Aprendizaje

Los resultados correspondientes al proceso de aprendizaje en la interfaz se muestran en la Figura 5.5. Se presenta la escala de 0 a 9, donde 0 corresponde a “Difícil” y 9 “Fácil”, aquí se observa una media correspondiente a 7.57, lo cual indica que el proceso de creación de experimentos de investigación a través del software resulta sencillo, aunque no de total facilidad. Esto se puede atribuir a que se deben tener algunos conocimientos previos para la configuración de pruebas.

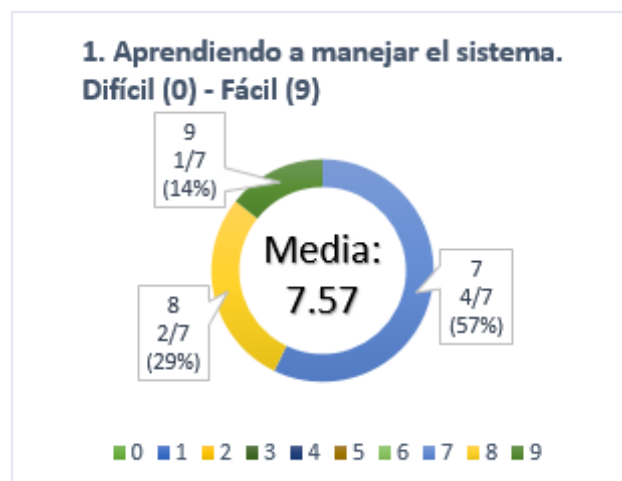


Figura 5.5: Resultados 2.1: Aprendiendo a manejar el sistema

En la Figura 5.6, se establecen los resultados de la opinión de los usuarios acerca de la exploración de nuevas características bajo el procedimiento de prueba y error, que consiste en familiarizarse con las herramientas de la interfaz para el desarrollo de pruebas a través de la exploración y manipulación de estos elementos. Se presenta una media de 7.28, la

cual refiere en este caso a una tendencia de facilitar la construcción de pruebas, aun en casos donde no existe pleno conocimiento de los criterios que se deben tener presentes y procedimientos en la presentación de estímulos.

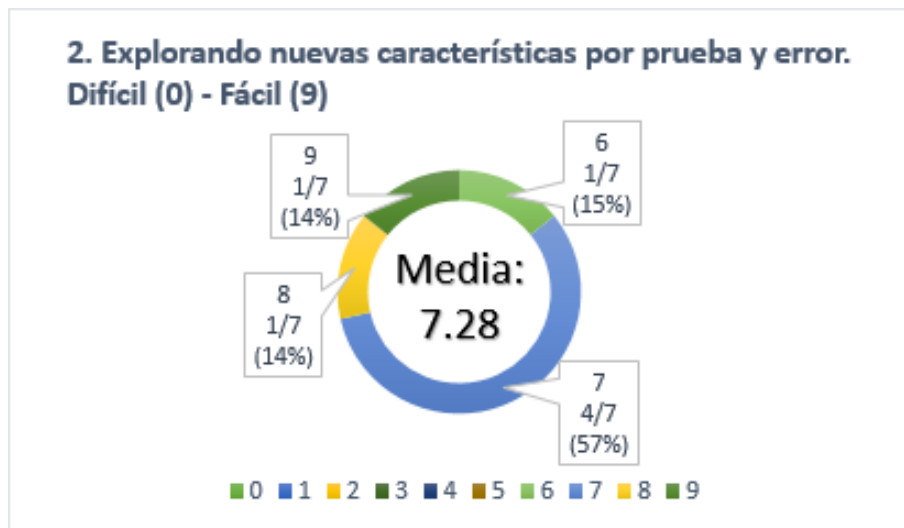


Figura 5.6: Resultados 2.2: Explorando nuevas características por prueba y error

En la Figura 5.7, se ilustra la gráfica correspondiente a los resultados de la opinión de los usuarios respecto a la ubicación de texto en la interfaz, el cual se encarga de recordar la finalidad de cada elemento presente en las distintas secciones. Por lo tanto, en el momento de realizar las pruebas, la información debe estar ubicada estratégicamente para disminuir los recorridos en la navegación y así minimizar los tiempos de configuración de pruebas. La media en este caso es de 7.71, que representa la aprobación de los usuarios acerca de la disposición de texto relevante en el entorno de la aplicación.

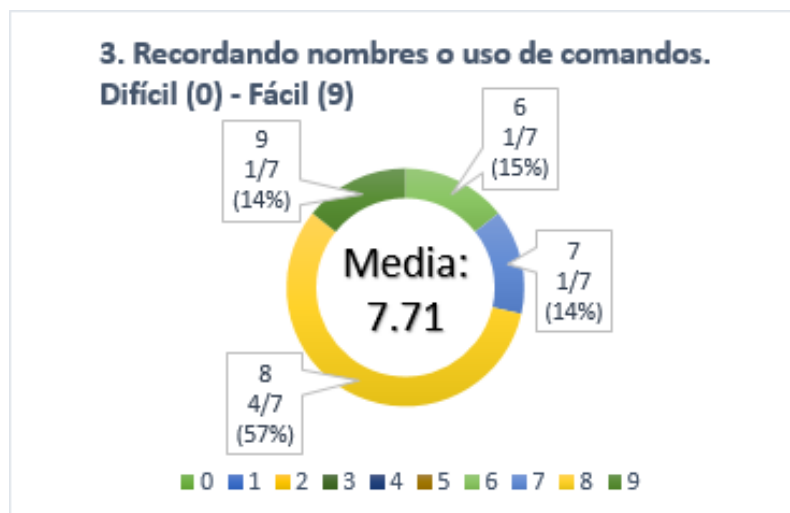


Figura 5.7: Resultados 2.3: Recordando nombres o uso de comandos

La figura 5.8, refleja la opinión de los usuarios sobre el tiempo que se tarda en aprender a manipular la herramienta software implementada, donde se presenta una media de 7.71,

por lo cual se considera que después de un tiempo prudente de exploración, se comienza a tener un mayor desempeño en el procedimiento para la configuración de pruebas.

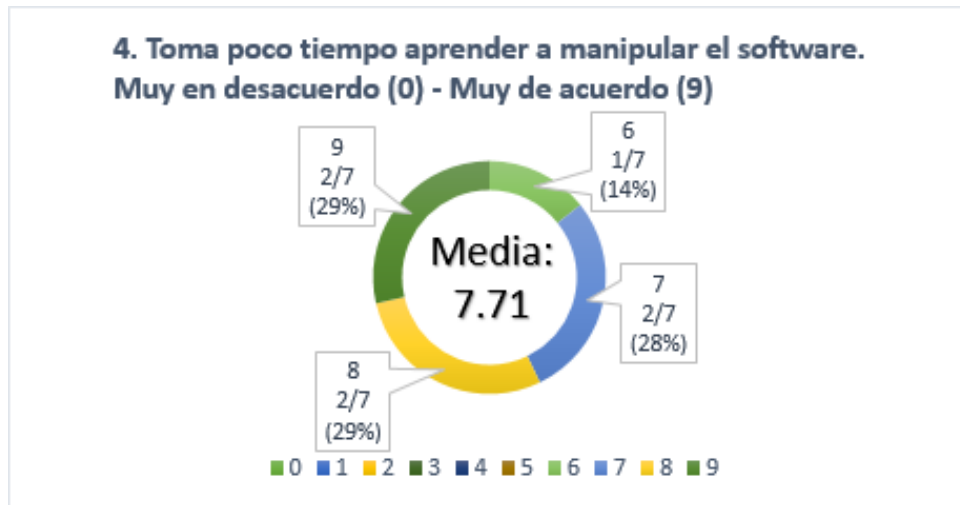


Figura 5.8: Resultados 2.4: Toma poco tiempo aprender a manipular el software

Ahora se presenta la opinión de los usuarios acerca de la facilidad con la que se pueden realizar tareas en la configuración de pruebas, el resultado se ve reflejado en la Figura 5.9, donde se calcula una media de 8.57, que resalta la objetividad que se presenta en la interfaz, ejecutando de forma correcta la finalidad que ha motivado el presente desarrollo.

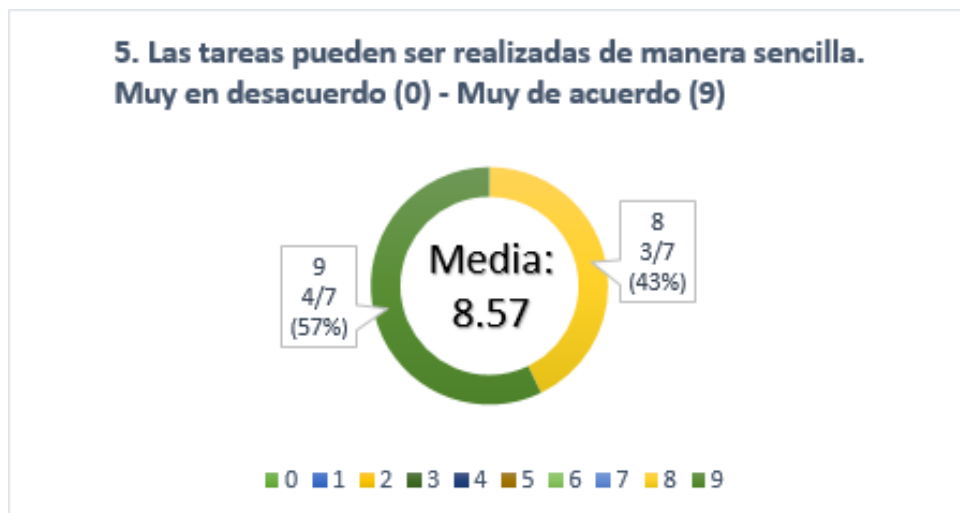


Figura 5.9: Resultados 2.5: Las tareas pueden ser realizadas de manera sencilla

Se ven reflejados en la Figura 5.10 los resultados acerca de la opinión que tienen los usuarios sobre el proceso que será llevado a cabo con otras personas en el aprendizaje del manejo de la interfaz, donde se presenta una media de 7.71, que supone una tendencia positiva sobre la fluidez y la claridad del contexto que se despliega a través de la herramienta propuesta.

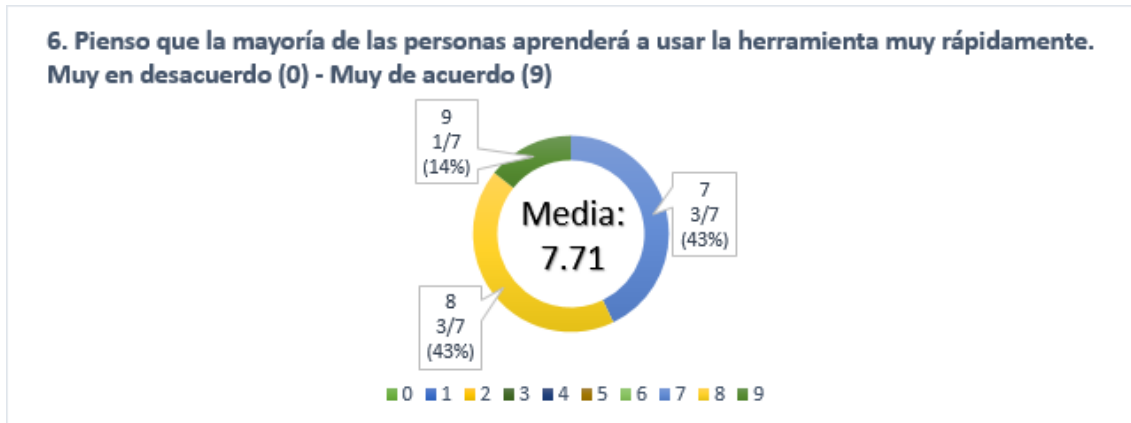


Figura 5.10: Resultados 2.6: Pienso que la mayoría de las personas aprenderá a usar la herramienta muy rápidamente

En la Figura 5.11, se muestran los resultados asociados a la familiarización inicial con la interfaz. La opinión para esta pregunta se encuentra muy dividida y se puede asociar a dos razones. La primera y menos probable, que la pregunta resulte ambigua para los participantes, o la segunda y más probable es que la opinión de los profesionales en salud se incline a estar de acuerdo con el enunciado, es decir, que concuerdan en que al principio resulta difícil aprender a usar la interfaz. Considerando como verdadera esta última razón, se puede comprender de mejor manera los resultados de esta pregunta. Es probable que para personas que no se encuentran en constante actualización con este tipo de herramientas o área de investigación, se presenten dificultades inicialmente para utilizar la herramienta. Sin embargo, el valor promedio para esta apreciación es de 4.43, por lo que, puede ser un punto que se deba atender para mejorar la interfaz.

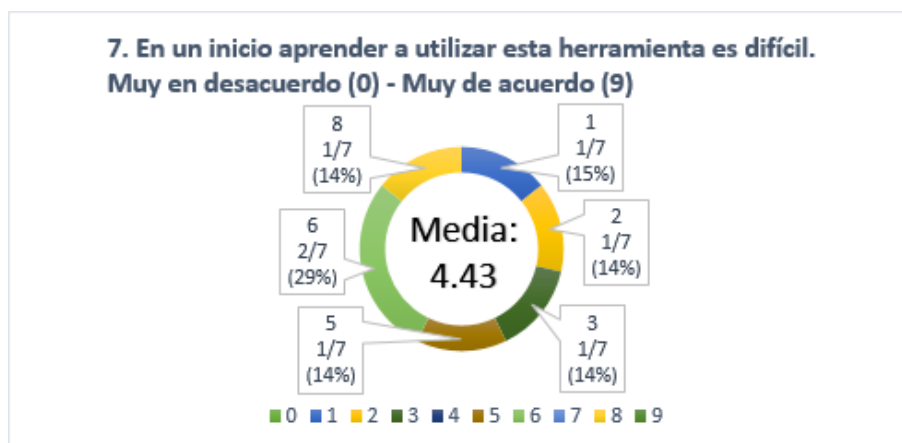


Figura 5.11: Resultados 2.7: En un inicio aprender a utilizar esta herramienta es difícil

La opinión de los usuarios acerca de la utilidad de mensajes de ayuda en la interfaz es positiva y se presenta en la Figura 5.12, donde se tiene una media de 7.42, que representa una adecuada asistencia en el desarrollo de las tareas en el software, esclareciendo en la mayoría de casos los valores disponibles en el rango de los parámetros a configurar.

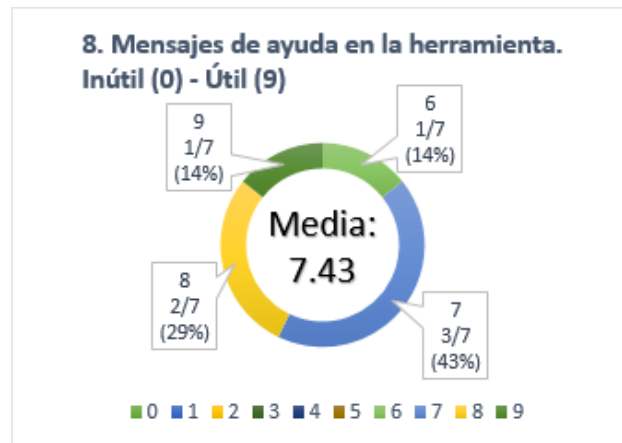


Figura 5.12: Resultados 2.8: Mensajes de ayuda en la herramienta

La Figura 5.13, muestra la opinión de los participantes frente a la posibilidad de contar con el apoyo de una persona especializada en el manejo de la herramienta. El criterio de los encuestados para este enunciado, refleja la misma situación de la pregunta 2.7 (Figura 5.11), donde la dispersión de las respuestas se puede interpretar como la dificultad inicial que pueden presentar ciertas personas para aprender a manipular una nueva herramienta como esta. Sin embargo para casos como estos se pueden plantear algunas soluciones, como por ejemplo, la creación de manuales de usuario o realizar capacitaciones a los nuevos usuarios de la herramienta. En cuanto a la tendencia en esta pregunta, el valor promedio es de 5.57, teniendo en cuenta que la respuesta esperada debería tender a “Muy en desacuerdo”, es un número aceptable considerando las soluciones sugeridas anteriormente.

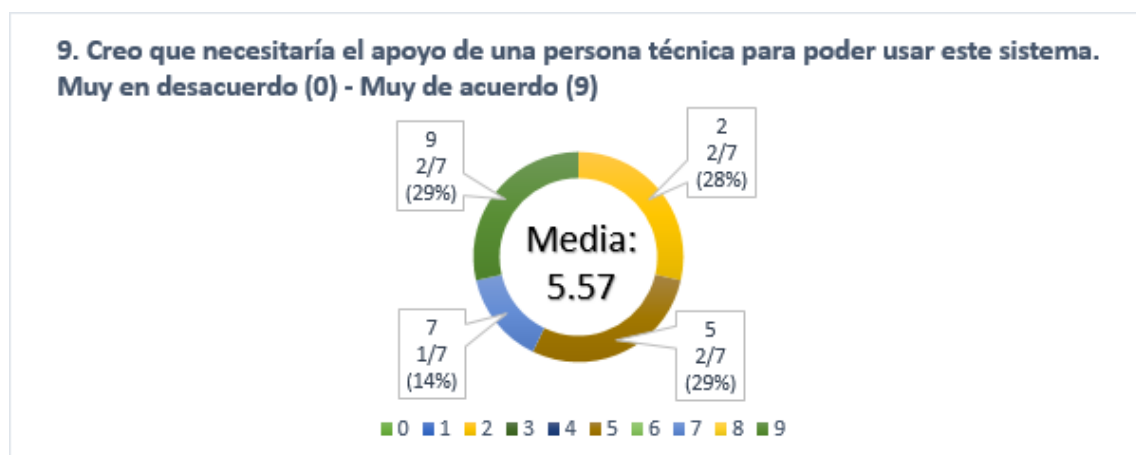


Figura 5.13: Resultados 2.9: Creo que necesitaría el apoyo de una persona técnica para poder usar este sistema

La Figura 5.14, muestra la percepción de los usuarios sobre los conocimientos previos que deben conocerse para comenzar a usar el sistema, donde, el promedio de los resultados obtenidos es de 5. Lo anterior evidencia que no existe una clara inclinación acerca de si se está de acuerdo o en desacuerdo con esta proposición, esto puede atribuirse principalmente a la experiencia por parte de los usuarios en este campo de investigación, debido

a que un mayor periodo de socialización con los conceptos vinculados a la herramienta, supone una competencia elevada al momento de construir pruebas y por tanto disminuye los tiempos de desarrollo, contrario al caso donde se realiza un primer acercamiento a este tipo de tecnologías y definiciones sobre la asociación de marcadores biológicos con los trastornos en cuestión, lo cual genera contratiempos en la configuración de los experimentos.

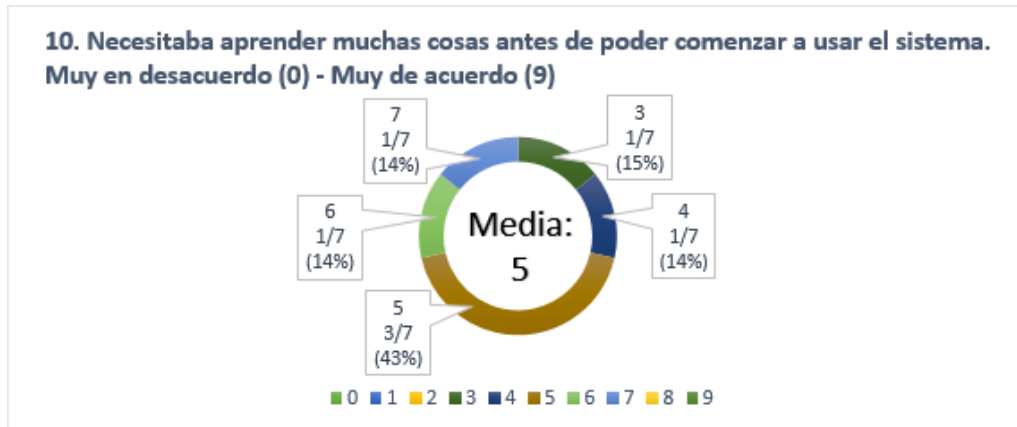


Figura 5.14: Resultados 2.10: Necesitaba aprender muchas cosas antes de poder comenzar a usar el sistema

5.3.3. Terminología e Información del sistema

Esta sección inicia consultando acerca de los términos utilizados en el desarrollo de la interfaz. En este contexto la consistencia puede relacionarse con el aspecto visual de la interfaz, puntualmente tipografía, colores, unidades o ventanas, entre otras. Como se puede observar en la Figura 5.15, la respuesta es muy aceptable. Sin embargo se aprecia que tan solo una persona no coincide completamente en que los términos utilizados en la interfaz son consistentes. Pese a lo anterior, el valor promedio es de 7.71, reflejando una inclinación por parte de los encuestados en coincidir que los términos del sistema son consistentes.

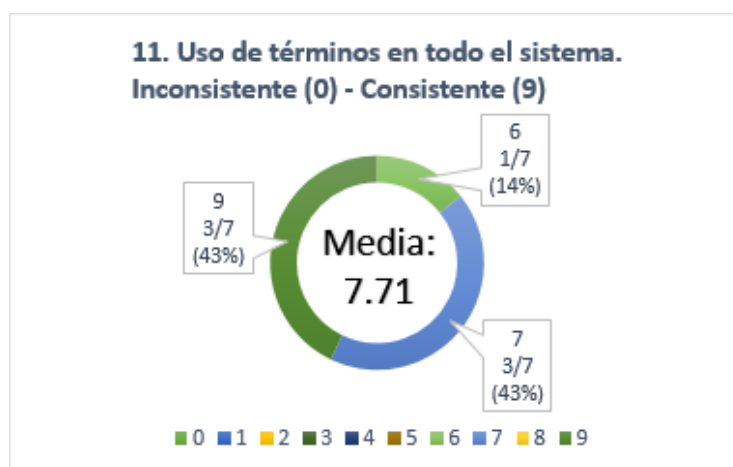


Figura 5.15: Resultados 3.1: Uso de términos en todo el sistema

La Figura 5.16, muestra la opinión de los participantes a cerca de si los términos (información textual) usados en la interfaz se relacionan con las tareas que la herramienta propone, en otras palabras, si el contexto de la información en el sistema es el correcto. En el gráfico se puede observar que la mayoría (5 de 7 participantes y un valor promedio de 8.43) de los participantes consideran que la terminología siempre se relaciona con las tareas que está realizando.

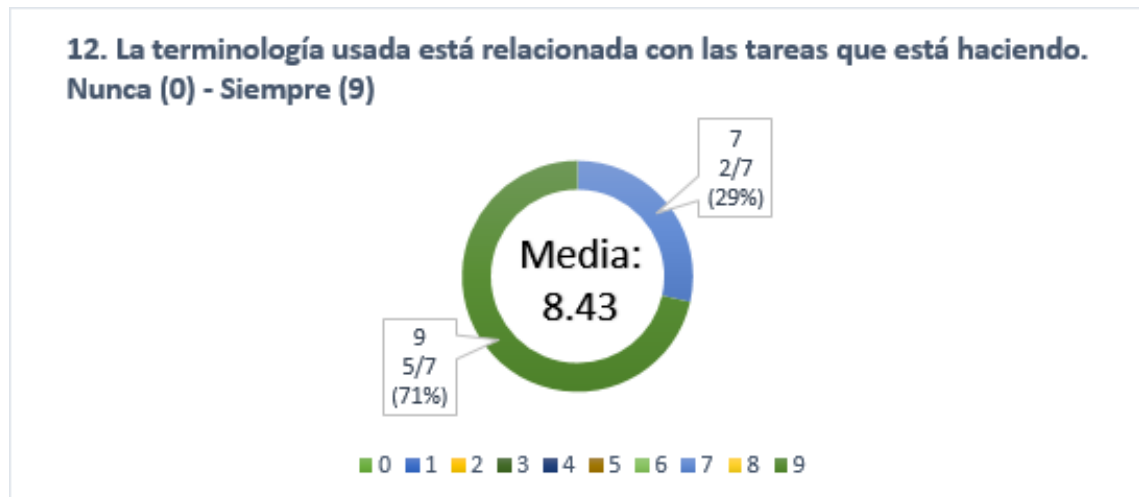


Figura 5.16: Resultados 3.2: La terminología usada está relacionada con las tareas que está haciendo

La opinión de los participantes acerca de los mensajes que solicitan ingresar información es buena, como se puede observar en la Figura 5.17, la opinión se inclina por calificar a estos mensajes como muy claros, tan solo una persona se mantiene imparcial frente a este concepto. Pese a lo anterior, el promedio se mantiene en 7.43, lo cual se considera bastante aceptable. Tanto esta cuestión como la de la Figura 5.12, se refieren a la utilidad de los cuadros de dialogo desarrollados para la interfaz, requerimiento que se describe detalladamente en el capítulo 4, en la sección de desarrollo software.

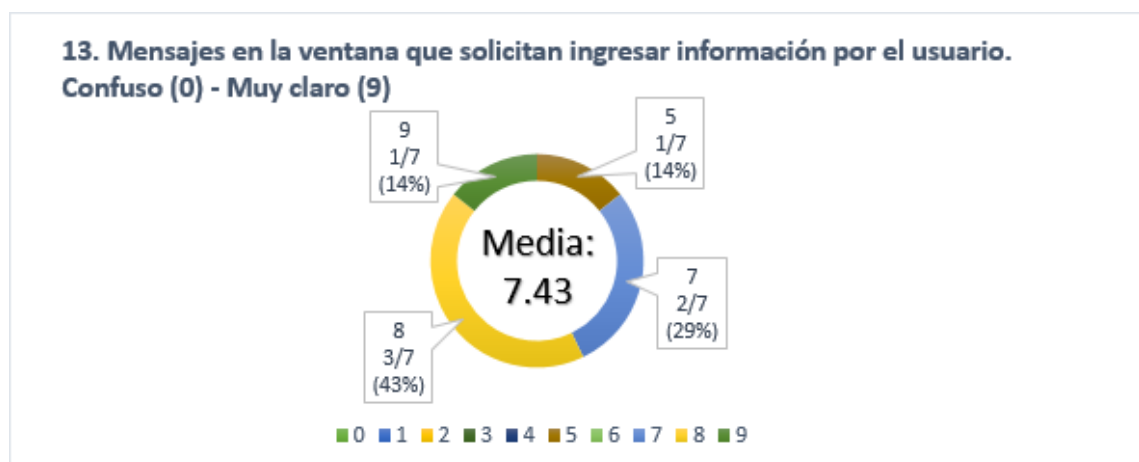


Figura 5.17: Resultados 3.3: Mensajes en la ventana que solicitan ingresar información por el usuario

La Figura 5.18, indica que los participantes consideran que la organización de la información en la interfaz, es clara, con un valor promedio de 8.14. Considerando que la organización es una característica fundamental para interfaces de usuario, estos resultados permiten afirmar que el desarrollo de la interfaz se logró adecuadamente, en términos de diseño gráfico.

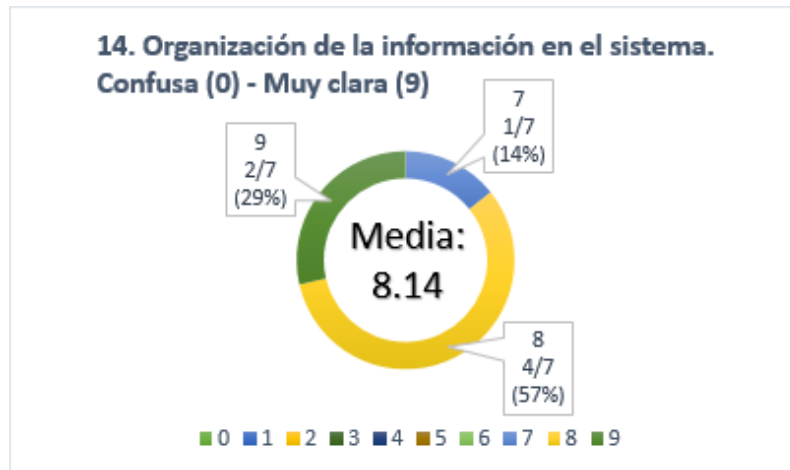


Figura 5.18: Resultados 3.4: Organización de la información en el sistema

Como apartado final de esta sección, se cuestiona a cerca de la comprensión de la información que proporciona la interfaz, obteniendo los resultados indicados en la Figura 5.19. La gráfica permite observar que los participantes concuerdan en que la información suministrada por la herramienta se comprende y maneja adecuadamente. Este hecho es comprensible debido al perfil de los encuestados, si bien, el área de investigación que se trata en este trabajo es muy específico, los participantes cuentan con los conocimientos necesarios para comprender la mayor parte de la información que la interfaz provee.

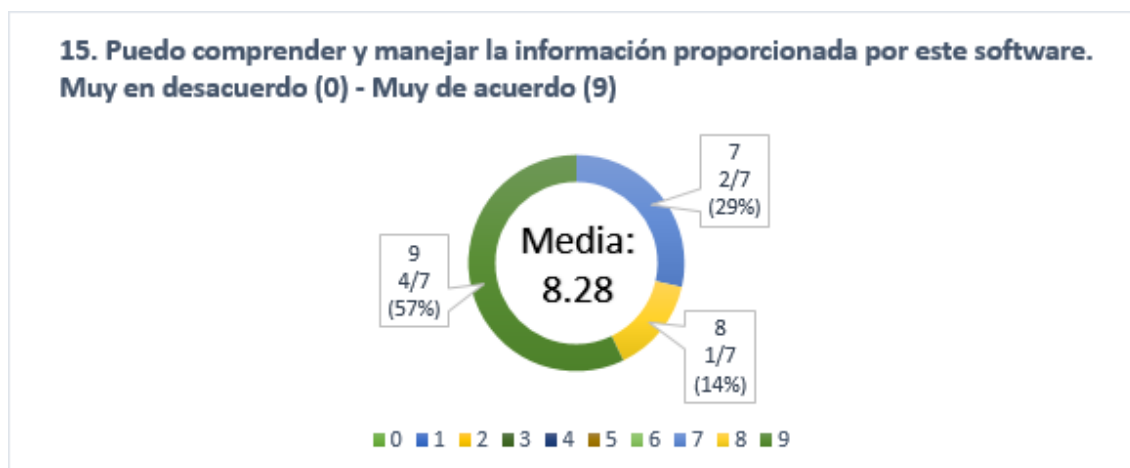


Figura 5.19: Resultados 3.5: Puedo comprender y manejar la información proporcionada por este software

5.3.4. Características del sistema

Acerca de la fiabilidad del sistema, los usuarios tienen una impresión positiva en términos generales, tal como se visualiza en la Figura 5.20, donde se tiene una media de 8.14 que refleja el desempeño objetivo de la temática tratada en el software, así como la orientación de cada uno de los elementos que se integran para la consecución de experimentos en la investigación neurológica.

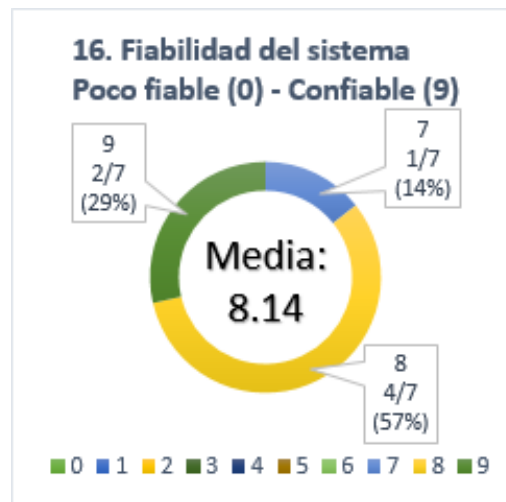


Figura 5.20: Resultados 4.1: Fiabilidad del sistema

En la Figura 5.21, se ven reflejadas las opiniones de los usuarios acerca de la detención imprevista del programa durante la configuración de experimentos en la interfaz, donde se obtiene que la mayoría tiende a estar en desacuerdo, expresando que no se presentó inconveniente alguno relacionado con este evento. Adicionalmente a este criterio se estipula por parte de los usuarios que el software nunca forzó su detención, pero si se retrasó en algunos momentos con la carga de elementos correspondientes a la pre-visualización y carga de estímulos, donde los parámetros utilizados tienen una repercusión directa por presentarse en considerables conjuntos de datos.

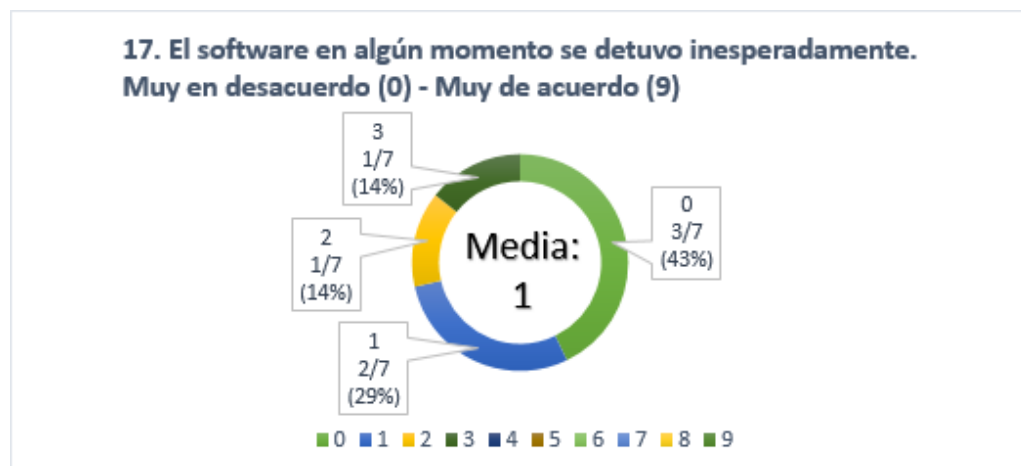


Figura 5.21: Resultados 4.2: El software en algún momento se detuvo inesperadamente

Por encima de los retrasos generados en la carga de los datos configurados en las secciones mencionadas, los usuarios manifiestan una correcta velocidad del sistema implementado, esto se visualiza en la Figura 5.22, donde la media es de 8.14, estableciendo una tendencia a considerar que la fluidez de la interfaz es lo suficientemente rápida para generar las tareas propuestas a través de los elementos y paradigmas propuestos.

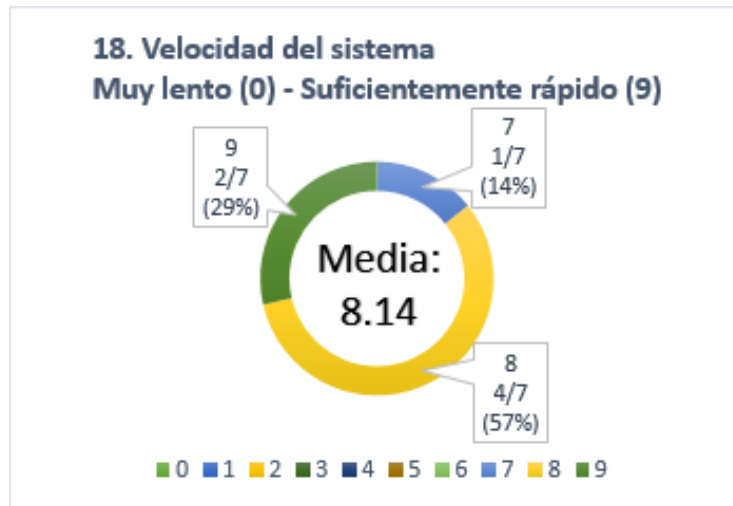


Figura 5.22: Resultados 4.3: Velocidad del sistema

Capítulo 6

Conclusiones

En este capítulo se realizan las conclusiones del trabajo y se proponen los trabajos futuros que pueden derivarse de este proyecto.

6.1. Conclusiones

Analizando los resultados anteriores y teniendo en cuenta el trabajo de investigación realizado para llevar a cabo este trabajo, se redactan las siguientes conclusiones.

- Los resultados en la validación de la interfaz a través de las opiniones de los participantes encuestados aprueban el trabajo realizado, con base en una experiencia de usuario satisfactoria. Pese a una baja convergencia en algunos de los criterios expuestos en el formulario, lo cual se debe a factores como el desconocimiento de los conceptos asociados y los paradigmas vinculados al desarrollo de pruebas en el ámbito neurológico, la respuesta general es bastante alentadora.
- La interfaz desarrollada en el presente trabajo cumple de manera correcta con los objetivos y requerimientos exigidos por la empresa *Aura Innovative Robotics*, realizando la respectiva validación en base al criterio profesional de sus miembros, quienes cuentan con una experiencia considerable gracias a la implementación de la plataforma *Oscann desk100* desarrollada para apoyar a los especialistas en las tareas de diagnóstico diferencial de enfermedades neurodegenerativas. Además, gracias al trabajo de investigación realizada durante el desarrollo de la interfaz, se puede concluir que los movimientos oculares son un marcador biológico notable para diagnóstico precoz de enfermedades neurodegenerativas. Por otra parte, los puntos de vista externos provenientes de los especialistas vinculados al sector clínico, reflejan una percepción positiva respecto al desempeño y la flexibilidad del software implementado.
- El principal aporte de la herramienta desarrollada a la plataforma *Oscann desk100*, es la flexibilidad y aleatoriedad asignada a la construcción de pruebas, algo que antes no era posible y reducía objetividad a la ejecución de estos procedimientos, donde los sujetos en cuestión podían memorizar los experimentos realizados y alterar de forma sustancial los resultados obtenidos.

- El uso del entorno Qt para la creación de interfaces gráficas permite realizar una implementación flexible en las diferentes plataformas existentes a día de hoy, por lo que provee elementos realmente útiles en distintas facetas, además de la integración con los diversos lenguajes de programación por medio de la inclusión de librerías, algo que hace posible desarrollar algoritmos en alto y bajo nivel, que cumplen con los diferentes objetivos estipulados en cualquier tipo de proyecto. Esta razón lo ha llevado a posicionarse en el ámbito de la programación como una de las herramientas más extendidas y empleada por reconocidas empresas en los distintos sectores de la industria, como por ejemplo: automatización, automotriz y servicios médicos entre otros.

6.2. Desarrollos futuros

Como trabajos futuros se proponen las siguientes actividades.

- Complementar la funcionalidad de la interfaz desarrollada, con base en los criterios establecidos por los participantes en la validación de este trabajo de grado.
- Actualizar la interfaz con un sistema de diseño *Drag & Drop*, en otras palabras, implementar un sistema que permita al usuario interactuar de manera aún más fluida con los elementos disponibles para el diseño de pruebas.
- La evolución de esta plataforma comprende trasladar las tecnologías a un dispositivo de seguimiento ocular montado (gafas), lo que propone varios retos para actualizar esta interfaz, por lo tanto, surge como un posible desarrollo futuro rediseñar esta herramienta para acoplarse con nuevas configuraciones de la plataforma.

Bibliografía

- [1] S. H. Román, “Caracterización de un dispositivo de medición de movimientos oculares y desarrollo software para el análisis de datos,” Master’s thesis, Universidad Politécnica de Madrid, 2017.
- [2] E. Hernández, S. Hernández, D. Molina, R. Acebrón, and C. E. García Cena, “Oscann: Technical characterization of a novel gaze tracking analyzer,” *Sensors*, vol. 18, no. 2, 2018.
- [3] E. B. Huey, “Preliminary experiments in the physiology and psychology of reading,” *American Journal of Psychology*, vol. 9, pp. 575–586, 1898.
- [4] R. Dodge and T. S. Cline, “The angular velocity of eye movements,” *Psychological Review*, vol. 8, pp. 145–157, 1901.
- [5] E. Whitmire, L. Trutoiu, R. Cavin, D. Perek, B. Scally, J. Phillips, and S. Patel, “Eyecontact: Scleral coil eye tracking for virtual reality,” in *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. New York, NY, USA: ACM, 2016, pp. 184–191.
- [6] A. Bulling, D. Roggen, and G. Tröster, “Wearable eog goggles: Seamless sensing and context-awareness in everyday environments,” *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 157–171, 2009.
- [7] “Dikablis eye tracking glasses,” Ergoneers, 2019. [Online]. Available: <https://www.ergoneers.com/en/hardware/dikablis-glasses/>[Accedido:18-03-2019]
- [8] A. Gibaldi, M. Vanegas, P. J. Bex, and G. Maiello, “Evaluation of the tobii eyex eye tracking controller and matlab toolkit for research,” *Behavior research methods*, vol. 49, no. 3, pp. 923–946, 2017.
- [9] *Manual de usuario - OSCANN desk100*, Aura Innovative Robotics S.L., Madrid, España, 2017.
- [10] W. Schneider, A. Eschman, and A. Zuccolotto, *E-Prime: User’s guide*, Psychology Software Incorporated, 2002.
- [11] “Camci-research,” Psychology Software Tools, 2019. [Online]. Available: <https://pstnet.com/products/camci-research/>[Accedido:07-03-2019]

- [12] A. Hollingworth, A. M. Richard, and S. J. Luck, "Understanding the function of visual short-term memory: transsaccadic memory, object correspondence, and gaze correction." *Journal of Experimental Psychology: General*, vol. 137, no. 1, p. 163, 2008.
- [13] J. W. Peirce, "Generating stimuli for neuroscience using psychopy," *Frontiers in neuroinformatics*, vol. 2, p. 10, 2009.
- [14] *SR Research Experiment Builder User Manual*, 2nd ed., SR Research Ltd., Mississauga, ON, Canada, 2017.
- [15] P. Garaizar and U.-D. Reips, "Visual dmdx: A web-based authoring tool for dmdx, a windows display program with millisecond accuracy," *Behavior Research Methods*, vol. 47, no. 3, pp. 620–631, 2015.
- [16] P. Kieslich, F. Henninger, D. Wulff, J. Haslbeck, and M. Schulte-Mecklenbeck, "Mouse-tracking: A practical guide to implementation and analysis," *A Handbook of Process Tracing Methods*. New York: Routledge, 2018.
- [17] J. Van Leeuwen and A. V. Belopolsky, "Distractor displacements during saccades are reflected in the time-course of saccade curvature," *Scientific reports*, vol. 8, no. 2469, 2018.
- [18] C. Dudschig, J. Souman, M. Lachmair, I. de la Vega, and B. Kaup, "Reading "sun" and looking up: The influence of language on saccadic eye movements in the vertical dimension," *PloS ONE*, vol. 8, no. 2, 2013.
- [19] J. Kunitatsu and M. Tanaka, "Roles of the primate motor thalamus in the generation of antisaccades," *Journal of Neuroscience*, vol. 30, no. 14, pp. 5108–5117, 2010.
- [20] M. Gorges, E. Pinkhardt, and J. Kassubek, "Alterations of eye movement control in neurodegenerative movement disorders," *Journal of ophthalmology*, vol. 2014, pp. 1–12, 2014.
- [21] M. Vernet, Q. Yang, M. Gruselle, M. Trams, and Z. Kapoula, "Switching between gap and overlap pro-saccades: Cost or benefit?" *Experimental Brain Research*, vol. 197, pp. 49–58, 2009.
- [22] T. J. Anderson and M. R. MacAskill, "Eye movements in patients with neurodegenerative disorders," *Nature Reviews Neurology*, vol. 9, no. 2, pp. 74–85, 2013.
- [23] S. Saxena, M. Funk, and D. Chisholm, "World health assembly adopts comprehensive mental health action plan 2013–2020," World Health Organization, Tech. Rep. 9882, 2013.
- [24] M. Garcés, "Estudio sobre las enfermedades neurodegenerativas en españa y su impacto económico y social," Universidad Complutense, Alianza Española de Enfermedades Neurodegenerativas, Madrid, Tech. Rep., 2016.
- [25] NN, "Neurological diagnostic tests and procedures," National Institute of Neurological Disorders and Stroke, Tech. Rep. 05-5380, 2005.

- [26] M. Alonso, A. Díaz, and A. Aguado, “Personas con discapacidad: perspectivas psicopedagógicas y rehabilitadoras,” in *Manuales (Siglo XXI de España Editores): Psicología*. Siglo XXI de España, 2005.
- [27] R. Warwick, P. L. Williams, and H. Gray, “The organs of the senses and the common integument,” in *Gray’s anatomy*. Longman, 1973, pp. 1000–1029.
- [28] F. J. Alañón Fernández, M. Cárdenas Lara, M. A. Alañón Fernández, and A. Martos Aguilera, “Anatomía y fisiología del aparato ocular,” *Oftalmología en atención primaria*. 2da ed. Alcalá la Real: Formación Alcalá, pp. 11–46, 2003.
- [29] E. Goldstein, *Sensation and Perception, 8th Edition*. Cengage Learning, 2009.
- [30] B. Cassin, S. Solomon, and M. Rubin, “Dictionary of eye terminology. gainesville,” *Florida: Triad Publishing*, 1990.
- [31] C. H. Morimoto and M. R. Mimica, “Eye gaze tracking techniques for interactive applications,” *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 4 – 24, 2005, special Issue on Eye Detection and Tracking.
- [32] J. Perea, “Fisiología motora,” in *Estrabismos*, 2nd ed., S. A. U. Artes Gráficas Toledo, Ed. Toledo, España: Diputacion de Toledo, 2008, ch. 2.
- [33] L. Gila, A. Villanueva, and R. Cabeza, “Fisiopatología y técnicas de registro de los movimientos oculares,” *Anales del Sistema Sanitario de Navarra*, vol. 32, pp. 9–26, 2009.
- [34] A. L. Yarbus, “Eye movements during fixation on stationary objects,” in *Eye Movements and Vision*. Boston, MA: Springer US, 1967, pp. 103–127.
- [35] F. Møller Møller, M. Laursen Laursen, J. Tygesen Tygesen, and A. K Sjølie, “Binocular quantification and characterization of microsaccades,” *Graefe’s archive for clinical and experimental ophthalmology*, vol. 240, pp. 765–70, 2002.
- [36] S. Martinez-Conde, S. L. Macknik, X. G. Troncoso, and T. A. Dyar, “Microsaccades counteract visual fading during fixation,” *Neuron*, vol. 49, no. 2, pp. 297 – 305, 2006.
- [37] S. D. Jamadar, B. P. Johnson, M. Clough, G. F. Egan, and J. Fielding, “Behavioral and neural plasticity of ocular motor control: Changes in performance and fmri activity following antisaccade training,” *Frontiers in Human Neuroscience*, vol. 9, p. 653, 2015.
- [38] L. Kleinedam, I. Frommann, S. Ruhrmann, J. Klosterkötter, A. Brockhaus-Dumke, W. Wölwer, W. Gaebel, W. Maier, M. Wagner, and U. Ettinger, “Antisaccade and prosaccade eye movements in individuals clinically at risk for psychosis: comparison with first-episode schizophrenia and prediction of conversion,” *European Archives of Psychiatry and Clinical Neuroscience*, 2019.
- [39] N. Leon Martinez, M. Blanco Jiménez, and A. Pinzón Amado, “Asociación entre la alteración de los movimientos oculares sacádicos y la esquizofrenia: un estudio de casos y controles,” *Revista Colombiana de Psiquiatría*, vol. XXXVI, pp. 628–642, 2007.

- [40] A. Srivastava, R. Sharma, S. K. Sood, G. Shukla, V. Goyal, and M. Behari, “Saccadic eye movements in parkinson’s disease,” *Indian journal of ophthalmology*, vol. 62, no. 5, p. 538, 2014.
- [41] Y. Terao, H. Fukuda, A. Yugeta, O. Hikosaka, Y. Nomura, M. Segawa, R. Hanajima, S. Tsuji, and Y. Ugawa, “Initiation and inhibitory control of saccades with the progression of parkinson’s disease - changes in three major drives converging on the superior colliculus,” *Neuropsychologia*, vol. 49, no. 7, pp. 1794 – 1806, 2011.
- [42] E. S. van Stockum, “Eye movement control and cognition in parkinson’s disease,” Master’s thesis, University of Canterbury, 2006.
- [43] S. Montagnese, H. M Gordon, C. Jackson, J. Smith, P. Tognella, N. Jethwa, R. Sherratt, and M. Morgan, “Disruption of smooth pursuit eye movements in cirrhosis: Relationship to hepatic encephalopathy and its treatment,” *Hepatology*, vol. 42, pp. 772–81, 2005.
- [44] B. Kuczynski and S. A. Kolakowsky-Hayner, “Impersistence,” in *Encyclopedia of Clinical Neuropsychology*, J. S. Kreutzer, J. DeLuca, and B. Caplan, Eds. New York, NY: Springer, 2011, pp. 1301–1301.
- [45] J. M. Bertolote, T. Dua, A. Janca, A. Muscetta, B. Saraceno, and S. Saxena, *Neurological disorders: public health challenges*. World Health Organization, 2006, ch. Conclusions and recommendations, pp. 189–194.
- [46] A. Poole and L. Ball, “Eye tracking in hci and usability research: Curent status and future prospect,” *Encyclopedia of Human Computer Interaction*, vol. 1, pp. 211–219, 2005.
- [47] E. Javal, “Essai sur la physiologie de la lecture,” *Annales d’Oculistique*, vol. 80, pp. 61–73, 1878.
- [48] S. Gómez Martín, “Análisis temporal de señales biológicas,” trabajo de fin de grado, Universidad Politécnica de Madrid, Madrid, España, 2018.
- [49] J. van der Geest and M. Frens, “Recording eye movements with video-oculography and scleral search coils: A direct comparison of two methods,” *Journal of neuroscience methods*, vol. 114, pp. 185–95, 2002.
- [50] A. Sprenger, B. Neppert, S. Köster, S. Gais, D. Kömpf, C. Helmchen, and H. Kimmig, “Long-term eye movement recordings with a scleral search coil-eyelid protection device allows new applications,” *Journal of neuroscience methods*, vol. 170, no. 2, pp. 305–309, 2008.
- [51] Q. Ding, K. Tong, and G. Li, “Development of an eog (electro-oculography) based human-computer interface,” in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. Shanghai, China: IEEE, 2006, pp. 6829–6831.
- [52] H. Singh and J. Singh, “Human eye tracking and related issues: A review,” *International Journal of Scientific and Research Publications*, vol. 2, no. 9, pp. 1–9, 2012.

- [53] J. M. Franchak, K. S. Kretch, K. C. Soska, and K. E. Adolph, “Head-mounted eye tracking: A new method to describe infant looking,” *Child development*, vol. 82, no. 6, pp. 1738–1750, 2011.
- [54] D. W. Hansen and Q. Ji, “In the eye of the beholder: A survey of models for eyes and gaze,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 478–500, 2010.
- [55] E. S. Dalmaijer, “Is the low-cost eyetribe eye tracker any good for research?” Department of Experimental Psychology, University of Oxford, Tech. Rep., 2014.
- [56] J. W. Peirce, “Psychopy - psychophysics software in python,” *Journal of neuroscience methods*, vol. 162, no. 1-2, pp. 8–13, 2007.
- [57] “Sr-research experiment builder,” SR Research Ltd, 2018. [Online]. Available: <https://www.sr-research.com/experiment-builder/>[Accedido:22-03-2019]
- [58] M. D. Weaver, J. Lauwereyns, and J. Theeuwes, “The effect of semantic information on saccade trajectory deviations,” *Vision research*, vol. 51, no. 10, pp. 1124–1128, 2011.
- [59] K. I. Forster and J. C. Forster, “Dmdx: A windows display program with millisecond accuracy,” *Behavior research methods*, vol. 35, no. 1, pp. 116–124, 2003.
- [60] S. Mathôt, D. Schreij, and J. Theeuwes, “Opensesame: An open-source, graphical experiment builder for the social sciences,” *Behavior research methods*, vol. 44, no. 2, pp. 314–324, 2012.
- [61] M. Kleiner, “Visual stimulus timing precision in psychtoolbox-3: Tests, pitfalls and solutions,” in *33rd European Conference on Visual Perception (ECPV 2010)*, Lausanne, Switzerland, 2010.
- [62] M. Kleiner, D. Brainard, D. Pelli, A. Ingling, R. Murray, C. Broussard *et al.*, “What’s new in psychtoolbox-3,” *Perception*, vol. 36, no. 14, 2007.

Anexos

A. Manual para la configuración gráfica del estímulo

En la Figura 1 se ilustra la pantalla principal que despliega la aplicación para la configuración del estímulo:

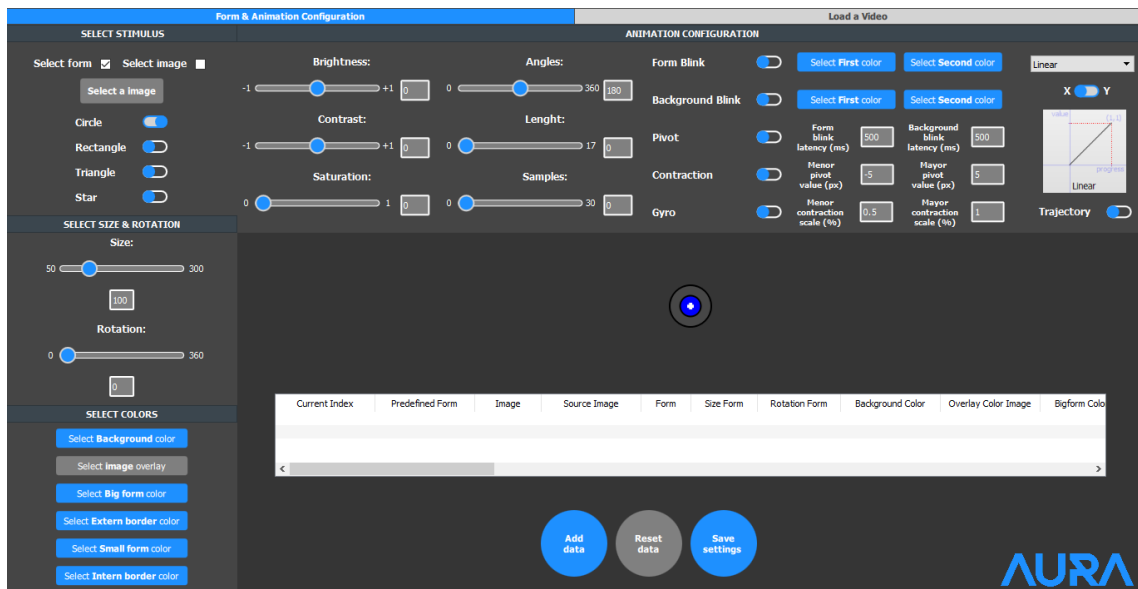


Figura 1: Pantalla principal de la interfaz para configuración gráfica del estímulo.

Es necesario identificar las secciones que están dispuestas en la pantalla para tener claro cómo se debe realizar una adecuada configuración del estímulo, por tanto se han señalado con distintos colores cada una de las secciones en la Figura 2, esto con el fin de facilitar el reconocimiento de las mismas.

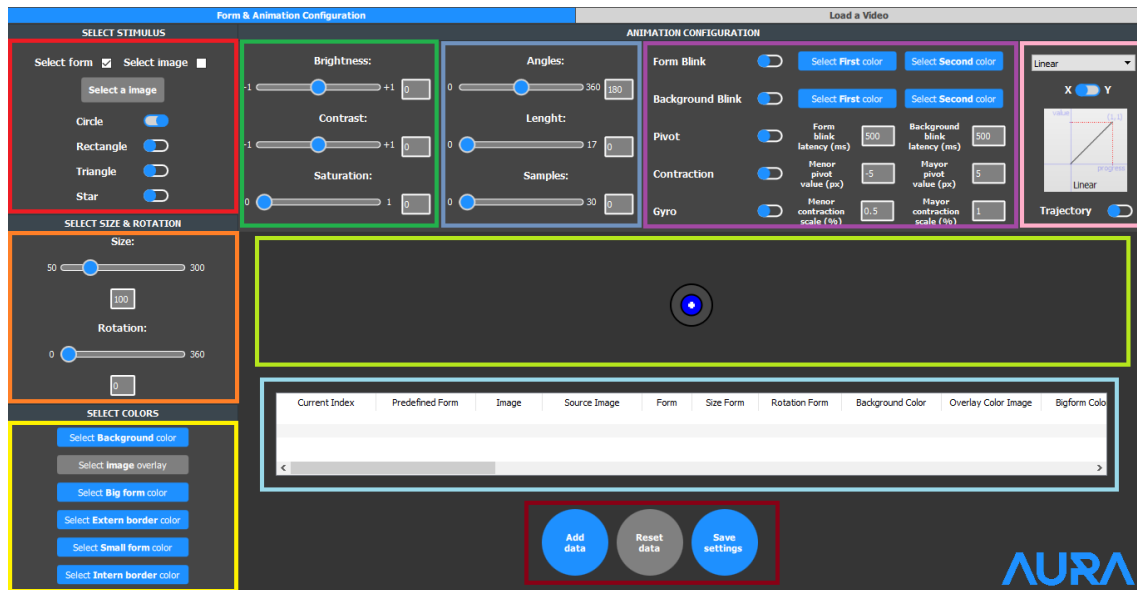


Figura 2: Definición de las secciones disponibles en la interfaz de configuración del estímulo.

Se ha definido cada sección señalada de acuerdo a la convención de colores que se indica en la Figura 3.

- -Selección del tipo de estímulo (Forma o imagen)
- -Configuración del tamaño y la rotación del estímulo
- -Configuración de colores del estímulo y fondo de la prueba
- -Configuración de brillo, contraste y saturación del estímulo
- -Configuración del efecto "blur" o difuminación del estímulo
- -Configuración de las animaciones y sus respectivos parámetros
- -Selección del tipo de movimiento y trayectoria según el eje
- -Visualización del estímulo con animaciones y trayectorias
- -Tabla para visualizar los parámetros configurados
- -Botones para agregar información a la tabla, exportarlos o reestablecer los parámetros de la aplicación

Figura 3: Convención de colores de las secciones

En la primera sección como se menciona anteriormente, se dispone de la posibilidad de escoger el tipo de estímulo, donde prevalecen dos opciones que consisten en seleccionar una forma predeterminada o cargar una imagen personalizada, la sección se detalla en la Figura 4.

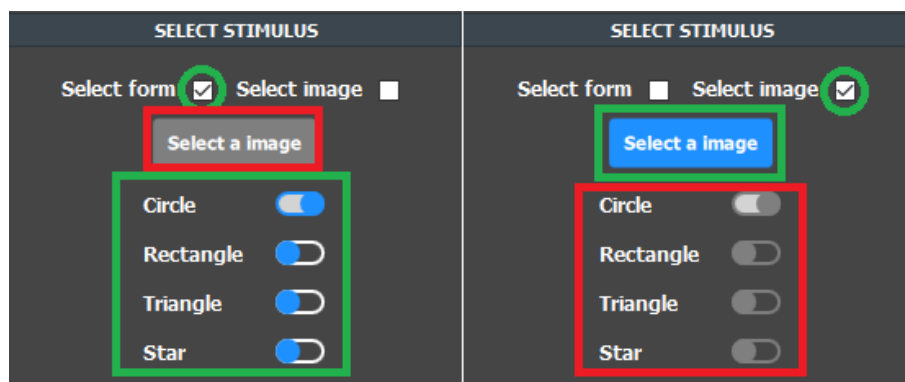


Figura 4: Sección para elegir la forma representativa del estímulo

Como se evidencia en la anterior imagen, existen cuatro formas predeterminadas que son: círculo, rectángulo, triángulo y estrella, opciones que están disponibles únicamente si el *CheckBox* que se indica en la parte superior izquierda se encuentra activado, de lo contrario al activar el *CheckBox* de la parte superior derecha estas opciones quedan deshabilitadas y se activa la opción de seleccionar una imagen presionando el botón “*Select an image*” y cargando el archivo a través del cuadro de diálogo que se ilustra en la Figura 5

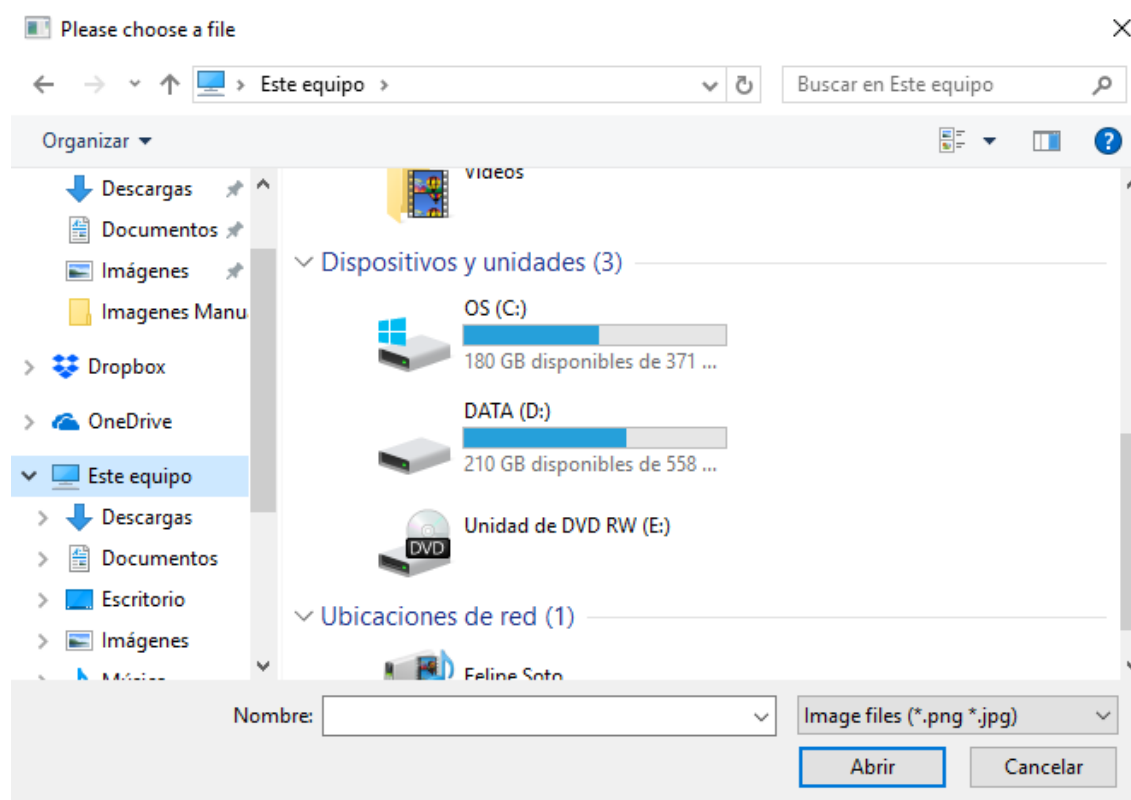


Figura 5: Cuadro de diálogo para cargar imágenes

Para la segunda sección están disponibles las opciones de cambiar el tamaño y la rotación del estímulo, en este caso el tamaño predeterminado es de 100 píxeles y la rotación

de 0 grados. En la Figura 6 se observa el cambio de tamaño a 200 píxeles en un triángulo y una rotación de 90 grados.

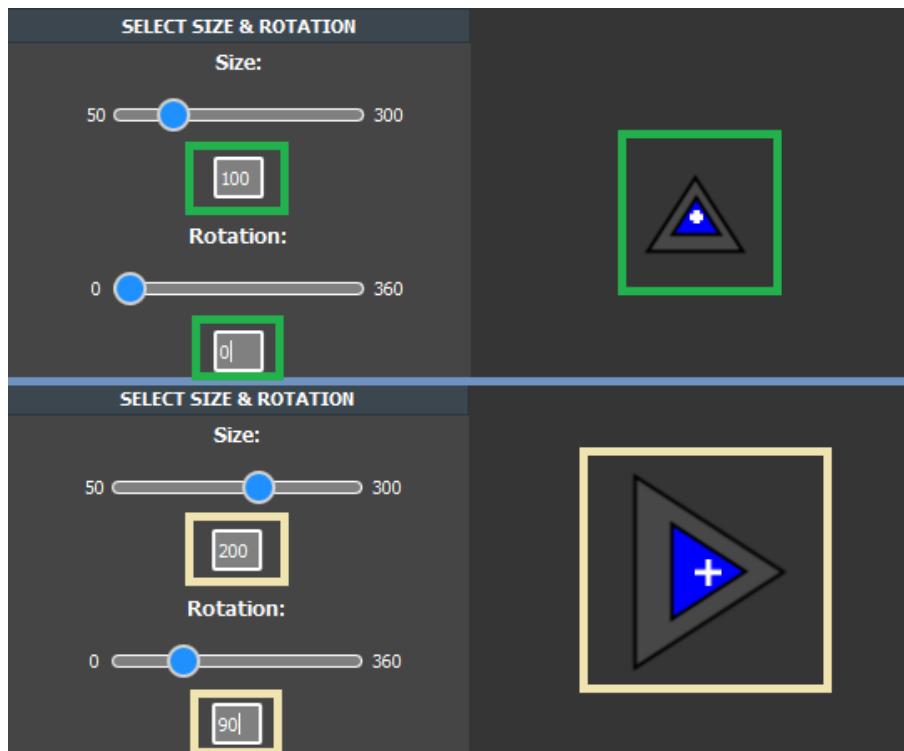


Figura 6: Cambios en la rotación y tamaño del estímulo

En la tercera sección se cuenta con la posibilidad de modificar los colores presentes en el estímulo, el color del fondo de la prueba y si se encuentra activo, un color para sobreponer a una imagen cargada, transformándose así en una silueta. Es necesario tener en cuenta que se han definido cuatro componentes de color para el estímulo, estos componentes corresponden a: color de forma interna (gris oscuro), color de forma externa (gris claro), color de borde interno (azul) y color de borde externo (verde), estas definiciones se evidencian en la Figura 7.

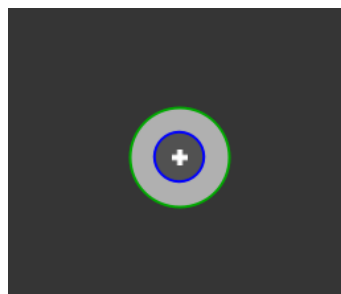


Figura 7: Secciones disponibles en el estímulo y colores correspondientes

Los colores en formato hexadecimal (#RRGGBB) por defecto para los parámetros de esta sección corresponden a los siguientes: color de forma interna (#0000ff), color de forma externa (#474747), color de borde interno (#000000), color de borde externo

(#000000), color de fondo (#353535) y color sobrepuesto de la imagen (ninguno, la condición inicial es transparente). La modificación de estos colores se realiza haciendo clic en los botones que se muestran en la Figura 8, que contienen el componente que se desea modificar, desplegando así un cuadro de diálogo con una paleta de colores para la elección como se indica en la Figura 9.

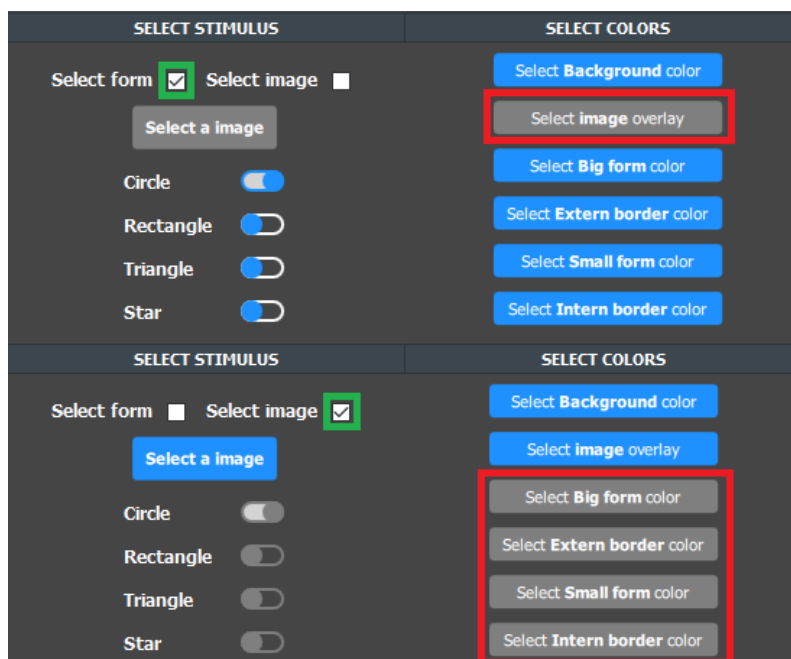


Figura 8: Sección para elegir la forma representativa del estímulo y los colores correspondientes.

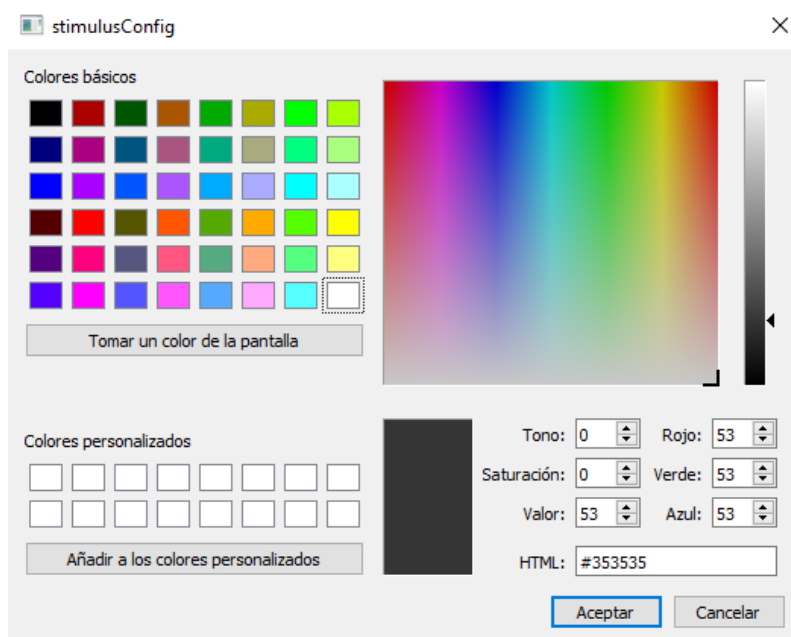


Figura 9: Cuadro de diálogo auxiliar que despliega la paleta de colores.

Para la cuarta sección se han definido tres *sliders* que permiten realizar cambios en el brillo, el contraste y la saturación del estímulo. Por defecto este valor es de 0 en los tres casos y a continuación se evidencian ejemplos de cambios que se producen al modificarlos. En la Figura 10 se presentan cambios en el brillo, en la Figura 11, cambios en el contraste y en la Figura 12 cambios en la saturación.

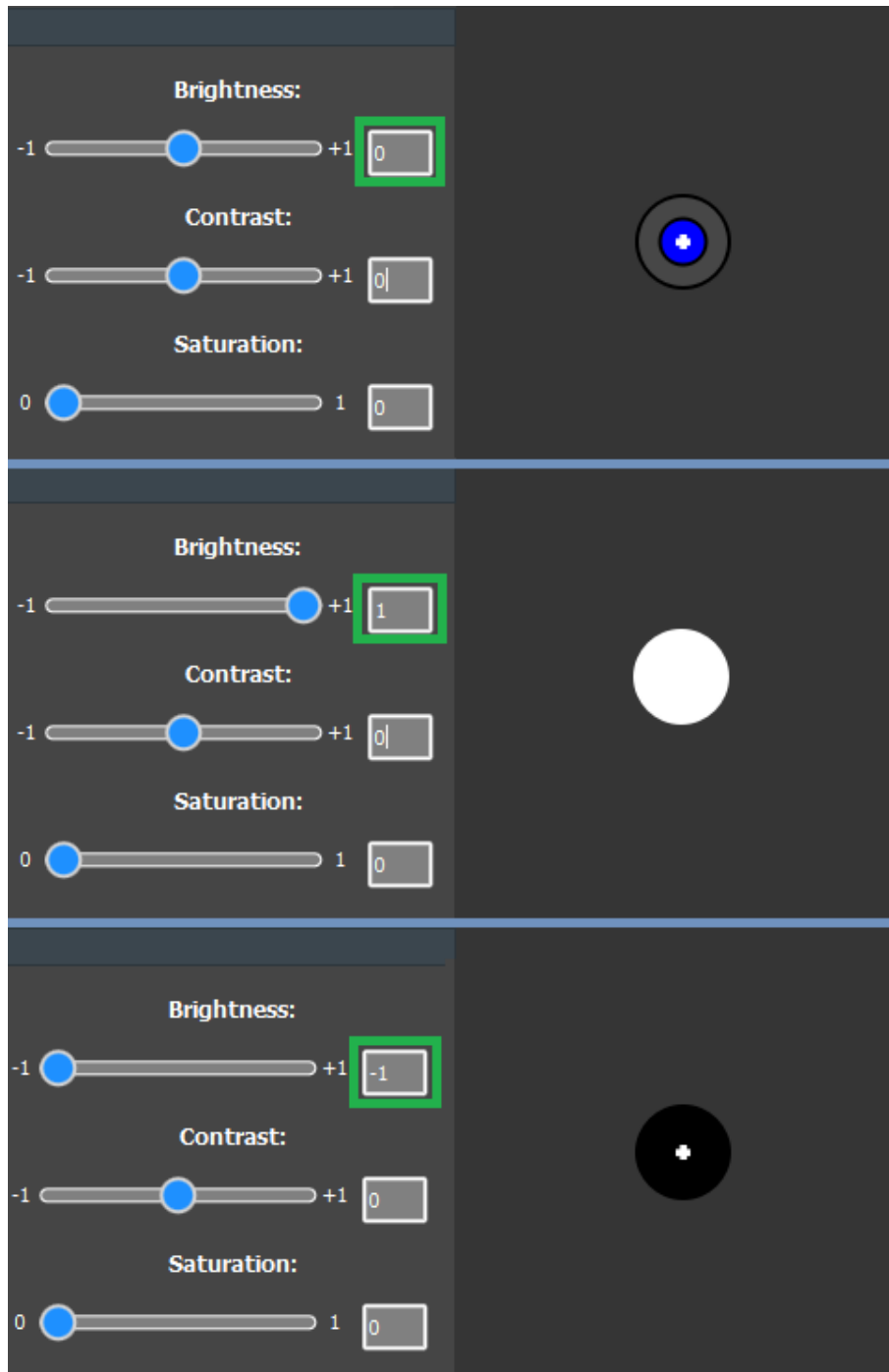


Figura 10: Variaciones en los valores de brillo.

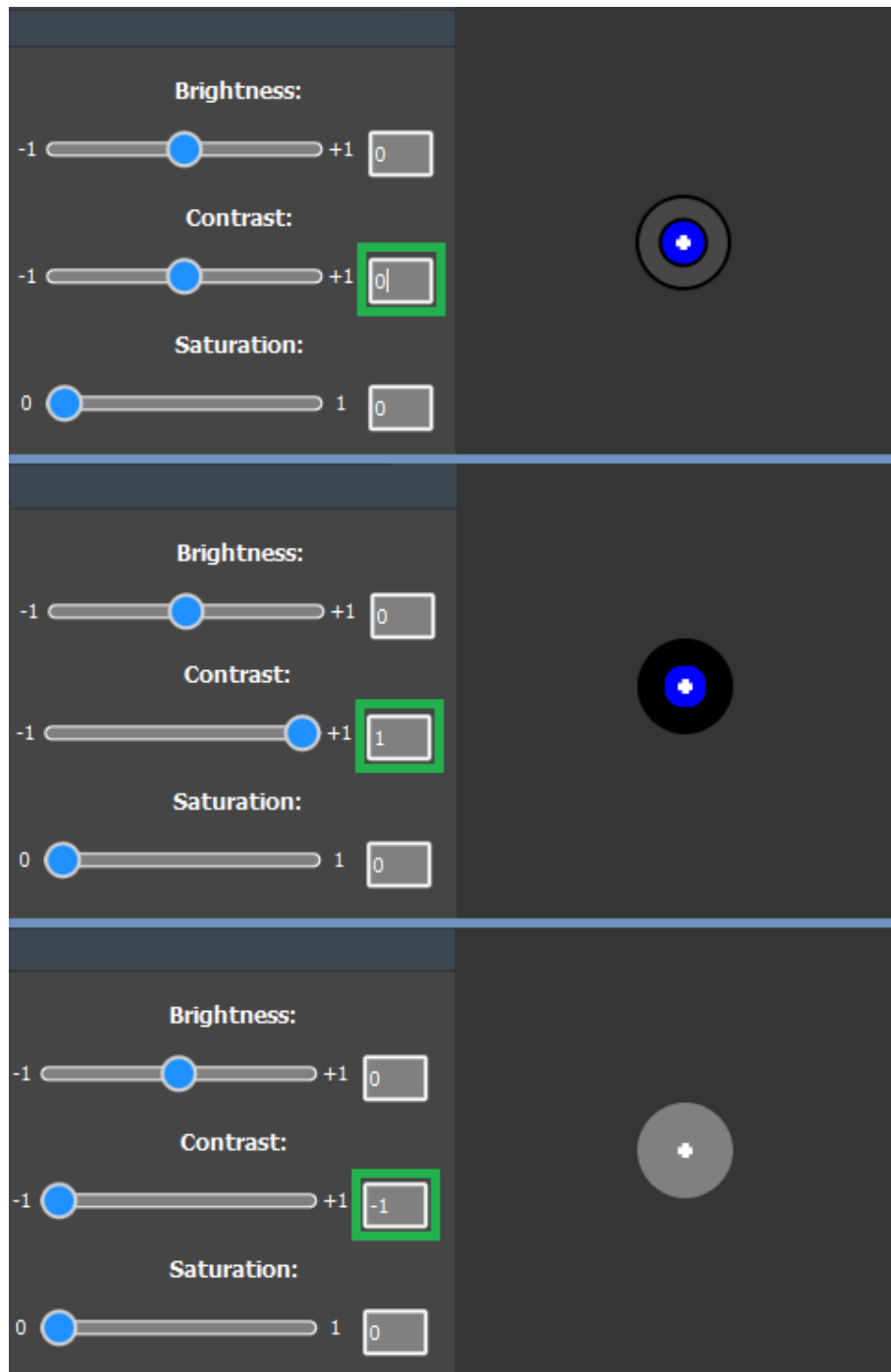


Figura 11: Variaciones en los valores de contraste.

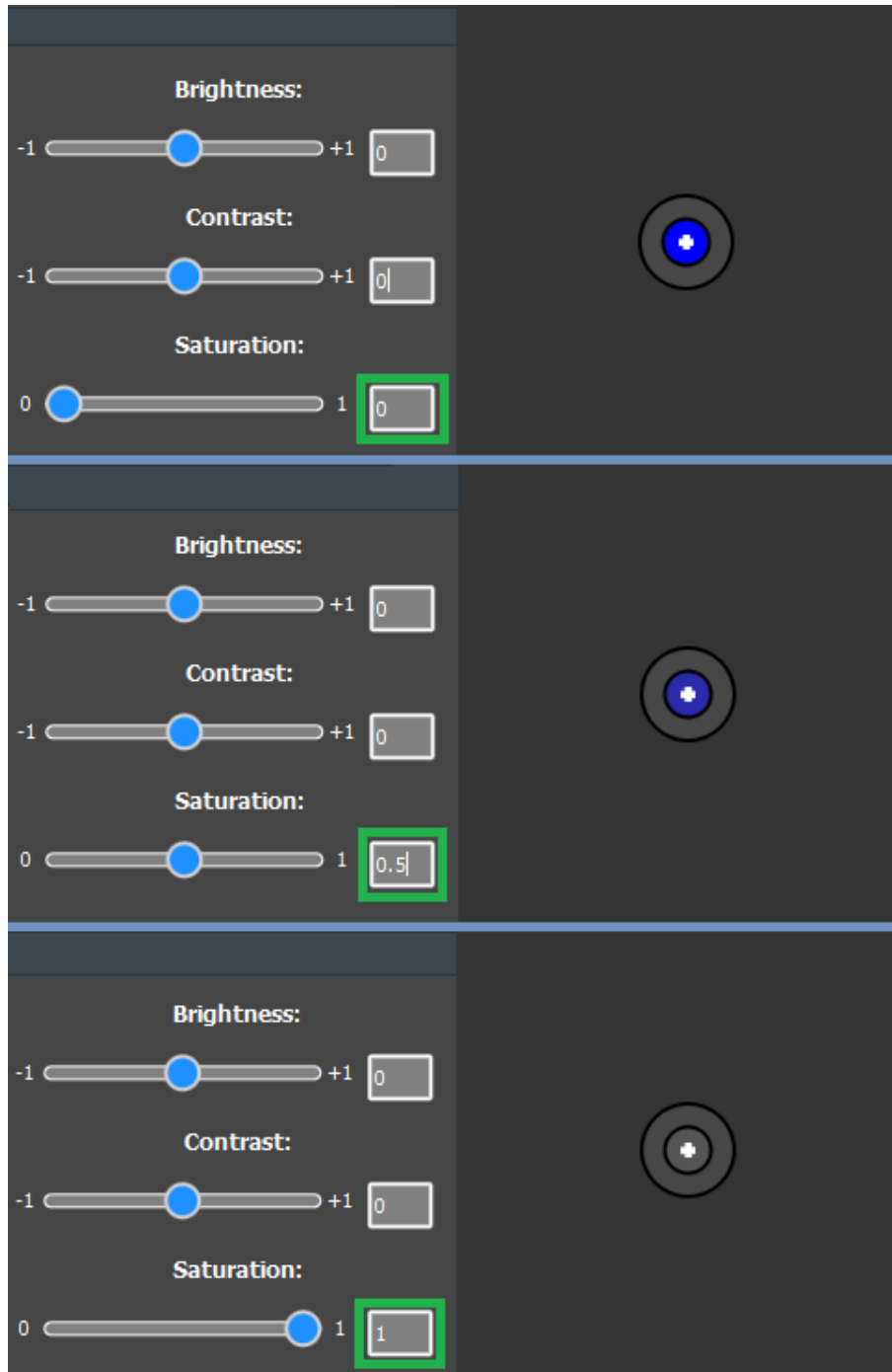


Figura 12: Variaciones en los valores de saturación.

En la quinta sección encontramos otros tres *sliders* adicionales, que permiten realizar efectos de difuminación tanto a las formas predeterminadas como a las imágenes. Aquí podemos modificar el ángulo hacia donde se dirige el efecto, la longitud del efecto y el número de muestreos para difuminar. En la Figura 13 se indican algunos cambios realizados en los parámetros asociados.

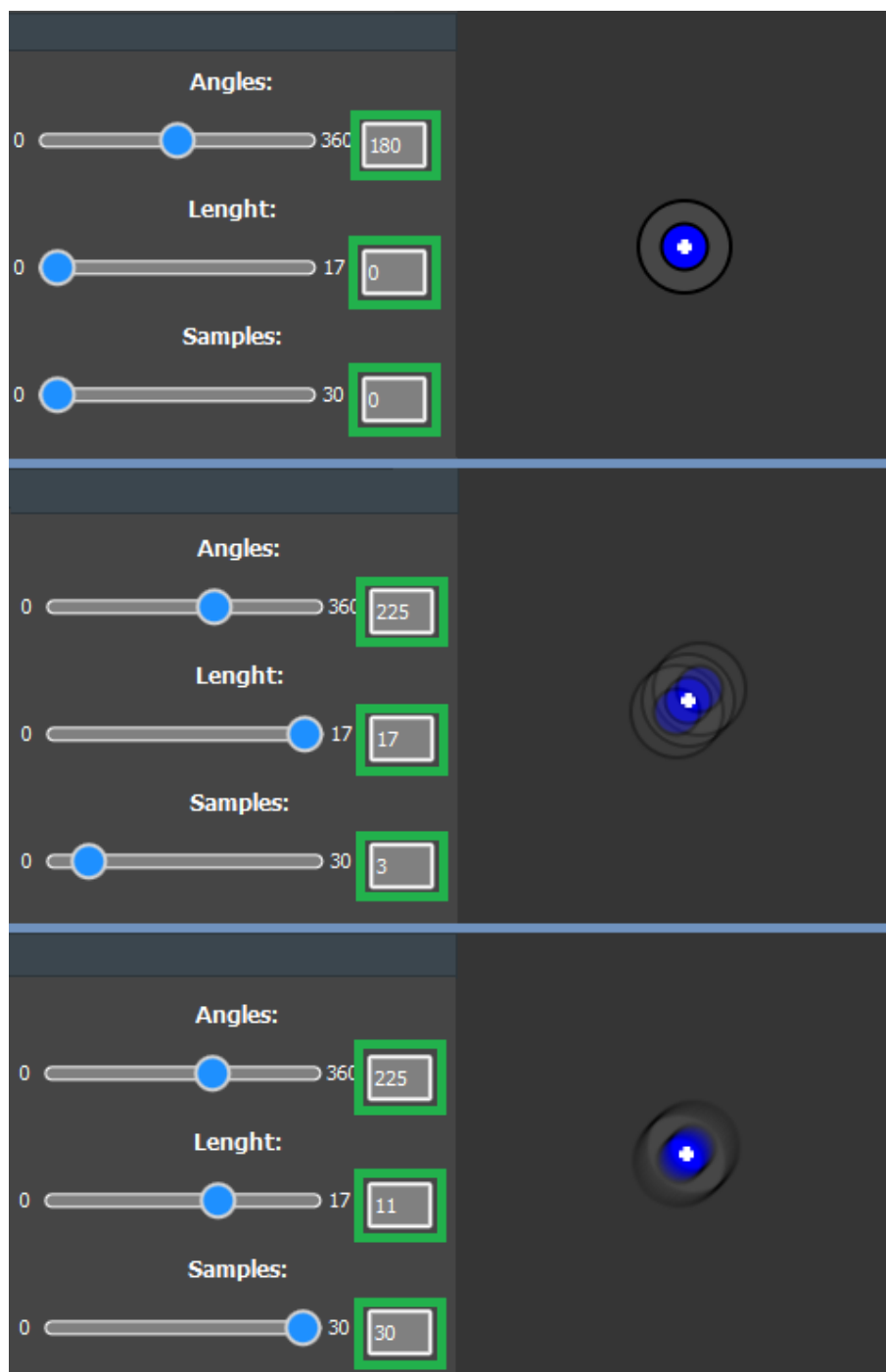


Figura 13: Variaciones en los valores de difuminación.

En la sexta sección encontramos cinco *switches* que permiten activar animaciones en el estímulo y en el entorno de la aplicación. Las animaciones que se encuentran disponibles son las siguientes: parpadeo en el color de fondo de la aplicación, parpadeo en el color de la forma externa o el color sobrepuesto de la imagen, efecto pivote, efecto de contracción y giro. Cada animación permite modificar parámetros asociados, en el caso de los parpadeos tanto del fondo como de la forma o imagen es posible seleccionar los dos colores que componen la animación y fijar el tiempo de latencia del parpadeo en un

rango de 10 a 5000 milisegundos. En el efecto pivote se puede fijar la distancia a la cual rebota el estímulo en un rango de -30 a 30 píxeles, y para la contracción se pueden fijar valores desde el 0 % al 100 % del tamaño inicial. En la Figura 14 se indica la distribución de esta sección.

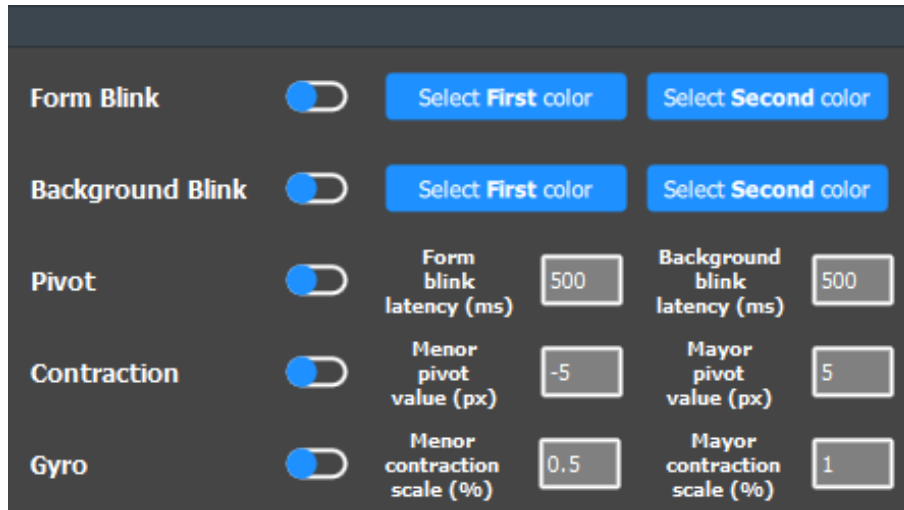


Figura 14: Sección de controles para configurar animaciones.

En la sección siete se tiene la posibilidad de activar trayectorias tanto en el eje x como en el eje Y. En el *ComboBox* hay una variedad de curvas bastante amplia donde al realizar una elección podemos visualizar una gráfica correspondiente al tipo de movimiento que se ejecutará. A continuación se observa un ejemplo de este tipo de trayectorias en la Figura 15.

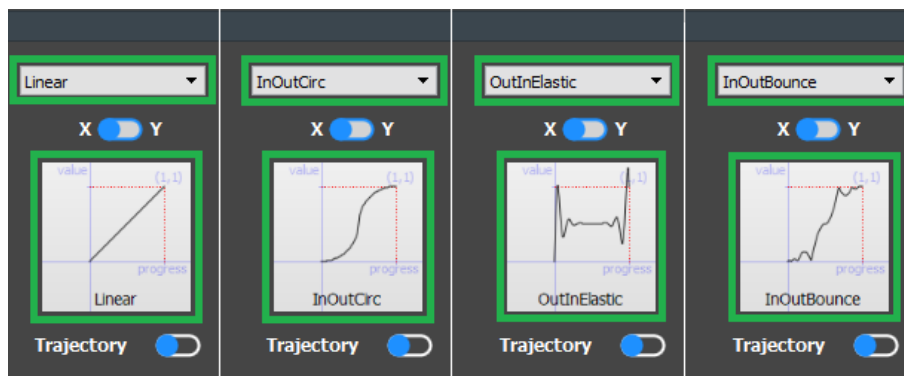


Figura 15: Sección de controles para configurar las trayectorias.

En la sección ocho se encuentra el espacio de visualización del estímulo, en este espacio se puede visualizar cada cambio realizado tanto en el aspecto del estímulo como en el de las animaciones, incluyendo la visualización de los diferentes tipos de trayectoria. En la Figura 16 se ilustra el espacio de trabajo, las direcciones hacia donde puede ir el estímulo y algunas variaciones en el aspecto del mismo.

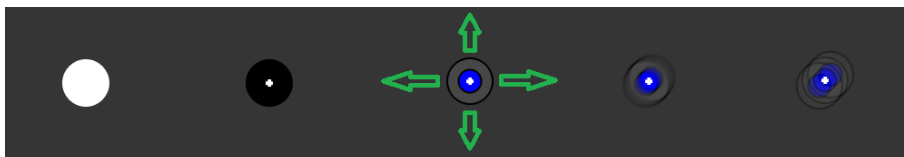


Figura 16: Espacio de visualización del estímulo.

En la sección nueve se observa una tabla a partir de la Figura 17, la cual nos permite visualizar cada uno de los parámetros que se han configurado para el estímulo y así tener claro que es lo que se debe exportar en el archivo XML. Estos datos solo se ven reflejados en la tabla al realizar la adición de datos con el botón “*Add Data*” disponible en la sección diez, lo que indica que cada cambio que se realice en la configuración del estímulo no se reflejará hasta que el usuario ejecute una especie de “*checkpoint*” con este botón.

Current Index	Predefined Form	Image	Source Image	Form	Size Form	Rotation Form	Background Color	Overlay Color Image	Bigform Colo
0	true	false	qrc:/qini/images/Mick...	Circle	100	0	#353535	transparent	#474747

Figura 17: Tabla disponible para la configuración de parámetros configurados.

Para finalizar, en la sección diez se encuentran disponibles tres botones, como se mencionó anteriormente. Uno de ellos es “*Add Data*” que nos permite añadir datos a la tabla y dejarlos listos para ser exportados, el segundo sirve para restablecer los parámetros por defecto de la configuración y el tercero tiene la finalidad de realizar la exportación de los datos en un archivo XML. Al pulsar el botón de exportar datos aparece un cuadro de diálogo solicitando un nombre para asignar al archivo de configuración que contiene los datos. Cabe destacar en este punto que la aplicación de acuerdo con los parámetros que se le especifiquen agrega una etiqueta a los archivos, esto se debe a que algunos estímulos no son compatibles con las diferentes pruebas que se pueden realizar por tanto van clasificados desde este apartado, tal como se especifica en el Sprint 5 en la configuración del estímulo. En la Figura 18 se ilustran los botones disponibles y en la Figura 19 el cuadro de diálogo ya mencionado.



Figura 18: Botones disponibles para agregar, restablecer y exportar datos.

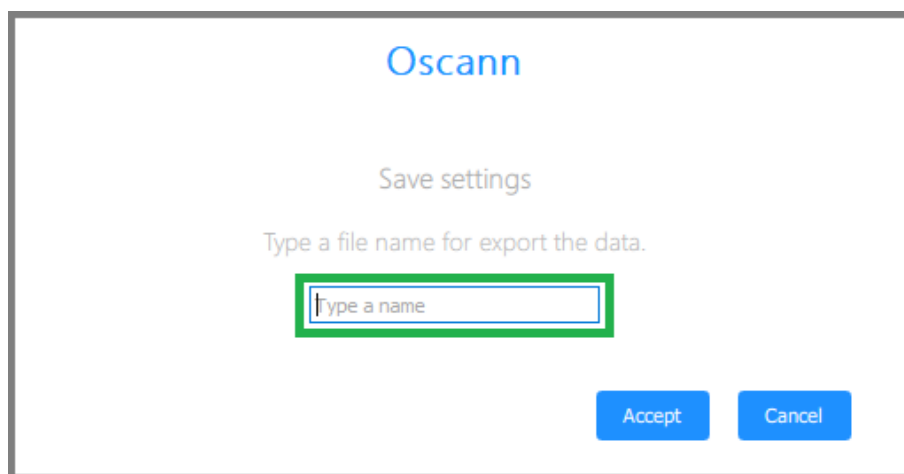


Figura 19: Cuadro de diálogo auxiliar, solicitando un nombre para el archivo de exportación.

B. Manual de uso para el software de diseño de pruebas

Este documento pretende dar una guía de uso del software de diseño de pruebas para investigación de enfermedades neurológicas, describiendo las funcionalidades de la herramienta y brindando claridad a ciertos términos que se manejan en este sistema. A continuación, se describen las tareas o pruebas que se pueden diseñar a partir de las funcionalidades del sistema.

- **Sacadas (*pro-saccades*):** Una prueba de sacadas consiste en que una persona que se encuentra frente a un monitor realiza el seguimiento de un estímulo visual, el cual en un principio se encuentra en el centro de la pantalla para después dirigirse hacia otro punto de la misma pero a manera de saltos, es decir el estímulo desaparece del centro y aparece en el punto donde se desee, para posteriormente regresar al centro de la misma manera.
- **Anti-sacadas (*anti-saccades*):** En esta prueba el paciente debe llevar su mirada o hacer una fijación en un punto simétrico en la dirección opuesta a la cual el estímulo se ha trasladado. Por ejemplo, si el estímulo se traslada hacia la derecha 15 grados, el paciente debe mirar al lado izquierdo sobre un área que él estime se encuentra a -15 grados.
- **Memoria (no visible):** Esta es similar a la prueba de sacadas, con la diferencia que se realiza en tres etapas. Mientras que en las sacadas el paciente inicialmente está mirando el estímulo en el centro de la pantalla para después llevar la mirada hacia donde se mueva el mismo y nuevamente llevar la mirada hacia el centro, ahí es donde termina una sacada. Ahora en esta prueba de memoria se le pide al paciente que lleve la mirada una vez más hacia donde se encontraba el estímulo anteriormente, sin que este aparezca, es decir, el paciente debe recordar el área de la pantalla donde se encontraba el estímulo.
- **Seguimiento del estímulo (*smooth pursuit*):** Esta prueba se trata de realizar un seguimiento al estímulo durante un periodo de tiempo, en términos generales es seguir un punto en la pantalla. Este tipo de pruebas pueden variar en cuanto a velocidad y el tipo de movimiento: lineal, sinusoidal, amortiguado, entre otros.
- **Reacción pupilar (*pupil light*):** Durante esta prueba el estímulo se encuentra fijo en el centro de la pantalla o simplemente se le indica al paciente que mantenga la mirada sobre el centro de esta, mientras el fondo o el color del “*background*”

cambia (generalmente blanco y negro) de manera intermitente durante un periodo definido.

- **Fijación (*fixation*):** Esta sencilla prueba se trata de mirar el estímulo fijo en una posición (generalmente en el centro) de la pantalla durante cierto tiempo.
- **Gap:** Paradigma presente cuando no hay estímulos visibles en la pantalla, este se da cuando hay un cambio en la posición del estímulo, es decir una sacada, pero esta se ve alterada por un tiempo en el cual desaparece (*Gap*), similar a un “*delay*” o pausa antes de realizar la sacada.
- **Overlap:** Al contrario del *Gap*, este paradigma describe la presencia de 2 estímulos a la vez, durante un periodo de tiempo muy corto.
- **Countermanding:** Es una prueba similar a una sacada o anti-sacada normal, pero en la cual reaparece el estímulo central después de un tiempo muy corto desde la sacada. Esta reaparición del estímulo central le dice al paciente que mantenga la mirada en el centro de la pantalla y no realice el movimiento sacádico.

Ahora, describiremos las funcionalidades del software para el diseño de los anteriores test.

Visual Guided Saccades

Con esta opción se pueden diseñar tareas de sacadas, anti-sacadas, sacadas de memoria, anti-sacadas de memoria y *countermanding*, además de hacer uso de los paradigmas de *Gap* y *Overlap*, y la capacidad de realizar tareas tipo demostración, para ejemplificar los movimientos que debe realizar el paciente durante la prueba. En la primera sección de esta opción (área delimitada en color azul de la Figura 20), procedemos seleccionando el eje en el cual se busca posicionar el estímulo, eje X o Y. Al seleccionar la opción ***Oblique*** se permite ingresar posiciones con ambas componentes, es decir, definir un valor para cada eje. Los grados están limitados a ciertos rangos que se muestran en cada campo de texto, establecidos para el tamaño del monitor donde se presentan las pruebas. ***Duration***, es el tiempo en el cual el estímulo se encuentra visible en la posición ingresada anteriormente. Para definir el comportamiento de la tarea, se marca la opción de *Pro-saccade* para una sacada normal o *Anti-saccade* para generar un movimiento tipo anti-sacada. La opción ***Visible/No visible***, permite realizar sacadas o anti-sacadas de memoria cuando el switch se encuentra en la posición de *No visible*. ***Test/Demo***, si el switch se encuentra en *Demo*, permite que durante la prueba se genere un indicador que realiza los movimientos que debe hacer el paciente, es decir una demostración de los movimientos correctos para dicha prueba. ***Gap u Overlap***, al seleccionar alguna de estas opciones se permite ingresar un valor entre 100-3000 milisegundos para un *delay*, en caso de *Gap* o un tiempo de sobre posición para *Overlap*, durante la transición de la sacada de ida, es decir el movimiento del centro de la pantalla a la posición con la que se haya programado. ***Countermanding***, permite que la prueba sea de este tipo, el rango de tiempo asignado es de 100-400 milisegundos, para este tipo de tarea se deben utilizar únicamente estímulos visibles.

Una vez ingresados todos los datos requeridos, se da clic sobre el botón de siguiente, ubicado en la parte inferior de la sección 1. Si hay algún error con los datos suministrados el sistema despliega un aviso informando cuál es el dato erróneo. Acto seguido

	X	Y	Dur.	Gap	Ovlp.	Count.
Fixation:	0	0	2000			
Target:	10	0	2000	0	0	0

Figura 20: Sección 1 para diseñar test de sacadas visuales

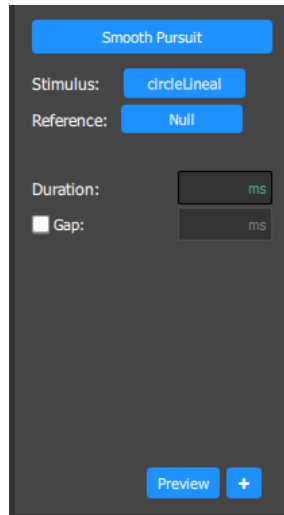
se despliega la sección dos donde se muestran los datos ingresados anteriormente (área delimitada en color rojo de la Figura 20), en esta sección se podrán elegir los estímulos para la prueba dentro de las posibilidades que se encuentran en la lista llamada “*Stimulus form*”. La opción de “*Set fixation time*” permite modificar el tiempo que permanecerá el estímulo en el centro de la pantalla (posición 0,0). Finalmente con los datos ingresados en su totalidad, se oprime el botón agregar (+) para llevar los datos a la tabla de información de la prueba. Cabe mencionar que una vez los datos se muestran en la sección dos, si se desea cambiar algún dato que provenga de la sección uno, recordar dar clic nuevamente sobre el botón de siguiente cada vez que se actualice información.

Smooth Pursuit

Esta opción es para añadir tareas de seguimiento de estímulos. Primero se procede a seleccionar el estímulo para dicha tarea en la lista de estímulos, a continuación se ingresa la duración de la presentación del estímulo en un rango de 500-5000 milisegundos y se procede a ingresar la tarea a la tabla de la prueba dando clic sobre el botón agregar (+). La opción de *Gap* se puede utilizar si la tarea que se está diseñando en ese momento no es la primera de la prueba, es decir deben existir tareas previas agregadas a la prueba, este tiempo de espera se agrega al final de la tarea anterior, es por eso que debe existir al menos una tarea previa.

Pupil Light

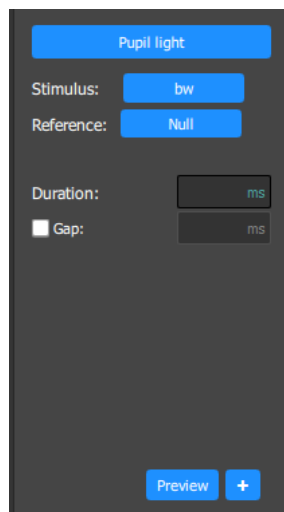
Se procede de manera similar a la anterior opción, primero se elige el estímulo. Cabe resaltar que para cada opción o tipo de prueba la lista de estímulos es diferente, a excepción de los estímulos para sacadas y fijaciones, donde son los mismos, en este caso



The screenshot shows a dark-themed interface for configuring a 'Smooth Pursuit' task. At the top, there is a blue header with the text 'Smooth Pursuit'. Below it, there are two rows of controls: 'Stimulus:' with a blue button labeled 'circleLineal', and 'Reference:' with a blue button labeled 'Null'. Underneath, there are two input fields: 'Duration:' with a text box and 'ms' to its right, and 'Gap:' with a checkbox, a text box, and 'ms' to its right. At the bottom, there are two blue buttons: 'Preview' and a '+' symbol.

Figura 21: Sección para diseñar tareas de seguimientos de estímulos

son los estímulos para generar reacción pupilar en el paciente. Ahora bien, ya seleccionado el estímulo, se ingresa el tiempo de duración de la tarea en un rango de 500 – 5000 milisegundos y se agrega dando clic sobre el botón agregar (+). La opción de *Gap* está disponible y funciona de igual manera que la señalada en la sección de *Smooth Pursuit*.



The screenshot shows a dark-themed interface for configuring a 'Pupil light' task. At the top, there is a blue header with the text 'Pupil light'. Below it, there are two rows of controls: 'Stimulus:' with a blue button labeled 'bw', and 'Reference:' with a blue button labeled 'Null'. Underneath, there are two input fields: 'Duration:' with a text box and 'ms' to its right, and 'Gap:' with a checkbox, a text box, and 'ms' to its right. At the bottom, there are two blue buttons: 'Preview' and a '+' symbol.

Figura 22: Sección para diseñar test de reacción pupilar

Fixation

Funciona de la misma manera que la primera sección de sacadas visuales a diferencia de la sección mencionada anteriormente, en esta se inicia seleccionando el estímulo para después definir la posición donde se presentará el estímulo. Paso siguiente se ingresa el tiempo de duración que el estímulo permanecerá en la pantalla y se finaliza presionando el botón de agregar (+). La opción de *Gap* está disponible y funciona de igual manera que la señalada en la sección de *Smooth Pursuit*.

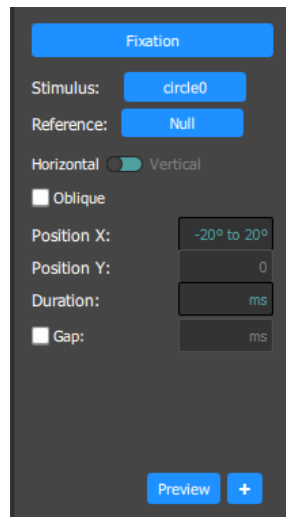


Figura 23: Sección para diseñar test de fijaciones

Acciones complementarias

A continuación, pasaremos a describir las funciones que cumplen ciertos botones que no se han tenido en cuenta anteriormente.

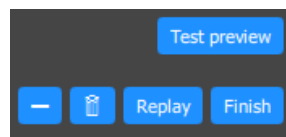
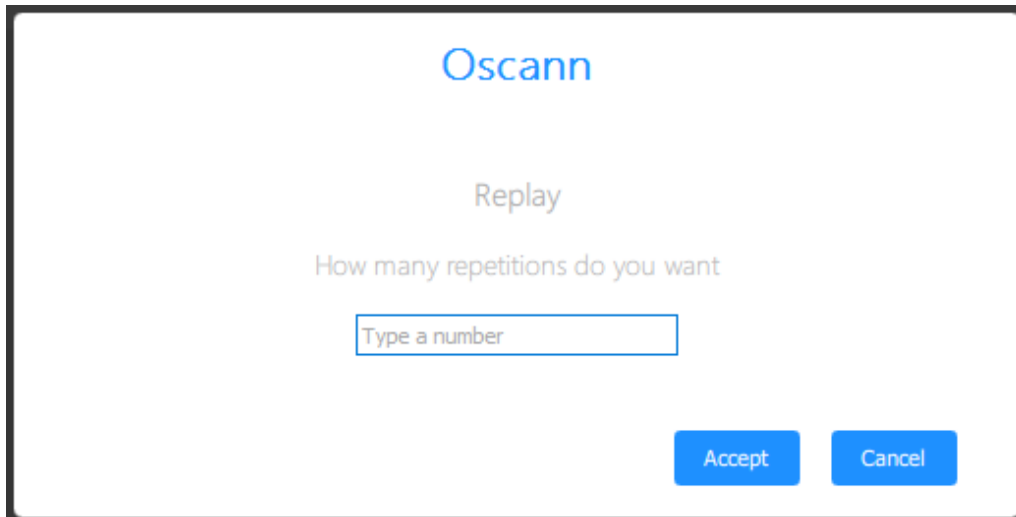


Figura 24: Botonería auxiliar

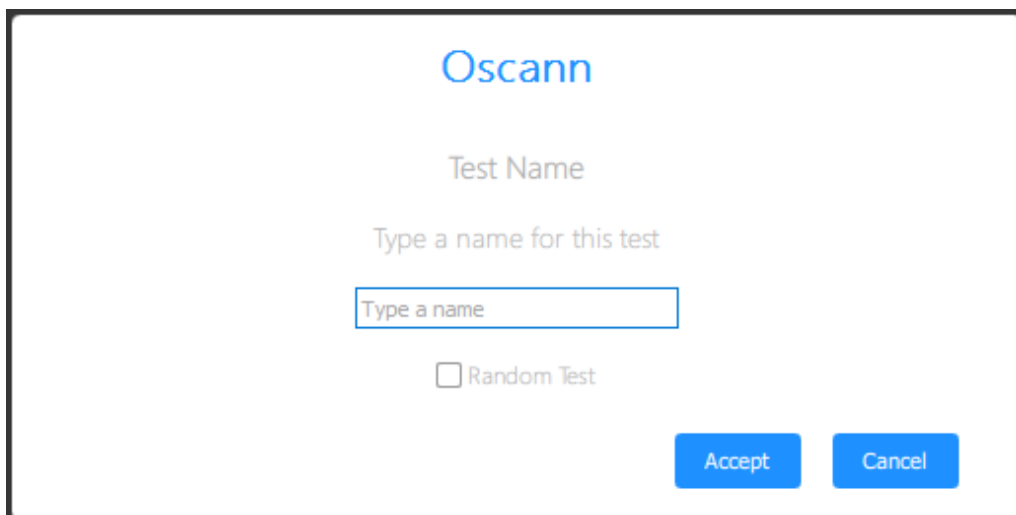
1. **Preview/Test preview:** Esta función permite visualizar el comportamiento tanto de una sección de la prueba, como la prueba completa. El botón *Preview* despliega una nueva ventana donde se puede observar únicamente la tarea que se está diseñando en ese momento, mientras que el botón de *Test preview* permite observar el funcionamiento de toda la prueba que hasta ese momento se haya configurado.
2. **Eliminar reciente:** Como su nombre lo indica, la función de este botón es eliminar la última tarea que se haya agregado. En la Figura 24, se muestra el botón que corresponde a un signo menos (-).
3. **Borrar prueba:** Elimina todos los datos de la prueba. En la Figura 24, se muestra el botón que corresponde a un icono con forma de cesta de basura.
4. **Replay:** La función de este es copiar los datos de la prueba y adicionarlos nuevamente, es decir, se replican los datos que hasta ese momento se hayan ingresado el número de veces que se indique en el cuadro de diálogo desplegado, ver Figura 25. Esta función se diseñó para evitar configurar todas las tareas de una prueba cuando estas son las mismas y deban repetirse durante toda la prueba.



The screenshot shows a dialog box titled "Oscann" with the subtitle "Replay". Below the subtitle, it asks "How many repetitions do you want". There is a text input field with the placeholder text "Type a number". At the bottom right, there are two blue buttons: "Accept" and "Cancel".

Figura 25: Cuadro de diálogo para iteración de elementos

5. **Reference:** Esta opción permite relacionar tareas para dar un orden estricto a la prueba si se desea, es decir, se puede relacionar la tarea que se está diseñando en ese momento con la última tarea ingresada, por esta razón en la lista solo se muestra el número de la última tarea que se encuentra en la tabla de información de la prueba. Dado el caso en que la prueba se defina como aleatoria, las tareas que se encuentran relacionadas no se verán alteradas en su orden de presentación.
6. **Finish:** Al finalizar el diseño de la prueba, se indica por medio de una ventana auxiliar ingresar un nombre o indicativo para guardar la prueba, además es posible seleccionar un orden aleatorio para la presentación de la prueba si así lo determina el usuario. En la Figura 26 se observa la ventana que permite realizar esta acción.



The screenshot shows a dialog box titled "Oscann" with the subtitle "Test Name". Below the subtitle, it asks "Type a name for this test". There is a text input field with the placeholder text "Type a name". Below the input field, there is a checkbox labeled "Random Test". At the bottom right, there are two blue buttons: "Accept" and "Cancel".

Figura 26: Cuadro de diálogo para iteración de elementos

C. Cuestionario para pruebas de validación

Cuestionario para la satisfacción de la interacción del usuario

El siguiente formulario busca medir el nivel de satisfacción de una interfaz de usuario, diseñada para realizar investigación en diagnóstico de enfermedades neurovegetativas.

Reacciones al software en general

Terrible	0	1	2	3	4	5	6	7	8	9	Estupendo
Complejo	0	1	2	3	4	5	6	7	8	9	Intuitivo
Frustrante	0	1	2	3	4	5	6	7	8	9	Satisfactorio
Rígido	0	1	2	3	4	5	6	7	8	9	Flexible

Aprendizaje

- 1. Aprendiendo a manejar la herramienta

Terrible 0 1 2 3 4 5 6 7 8 9 Fácil

- 2. Explorando nuevas características a prueba y error.

Difícil 0 1 2 3 4 5 6 7 8 9 Fácil

- 3. Recordando nombres o uso de comandos.

Difícil 0 1 2 3 4 5 6 7 8 9 Fácil

- 4. Toma poco tiempo aprender a manipular el software.

Muy en desacuerdo 0 1 2 3 4 5 6 7 8 9 Muy de acuerdo

- 5. Las tareas se pueden realizar de manera sencilla.

Muy en desacuerdo 0 1 2 3 4 5 6 7 8 9 Muy de acuerdo

- 6. Pienso que la mayoría de las personas aprenderá a usar la herramienta muy rápidamente.

Muy en desacuerdo 0 1 2 3 4 5 6 7 8 9 Muy de acuerdo

- 7. En un inicio aprender a utilizar esta herramienta es difícil.

Muy en desacuerdo 0 1 2 3 4 5 6 7 8 9 Muy de acuerdo

- 8. Mensajes de ayuda en la herramienta

Inútil 0 1 2 3 4 5 6 7 8 9 Útil

- 9. Creo que necesitaría el apoyo de una persona técnica para poder usar este sistema.

Muy en desacuerdo 0 1 2 3 4 5 6 7 8 9 Muy de acuerdo

- 10. Necesitaba aprender muchas cosas antes de poder comenzar a usar el sistema.

Muy en desacuerdo 0 1 2 3 4 5 6 7 8 9 Muy de acuerdo

Terminología e información del sistema

- 11. Uso de términos en todo el sistema.

Inconsistente 0 1 2 3 4 5 6 7 8 9 Consistente

- 12. La terminología usada está relacionada con las tareas que está haciendo.

Nunca 0 1 2 3 4 5 6 7 8 9 Siempre

- 13. Mensajes en la ventana que solicitan ingresar información por el usuario.

Confuso 0 1 2 3 4 5 6 7 8 9 Muy claro

- 14. Organización de la información en el sistema.

Confusa 0 1 2 3 4 5 6 7 8 9 Muy clara

- 15. Puedo comprender y manejar la información proporcionada por este software.

Muy en desacuerdo 0 1 2 3 4 5 6 7 8 9 Muy de acuerdo

Características del sistema

- 16. Fiabilidad del sistema.

Poco fiable 0 1 2 3 4 5 6 7 8 9 Confiable

- 17. El software en algún momento se detuvo inesperadamente.

Muy en desacuerdo 0 1 2 3 4 5 6 7 8 9 Muy de acuerdo

- 18. Velocidad del sistema.

Muy lento 0 1 2 3 4 5 6 7 8 9 Suficientemente rápido

Observaciones

- Con respecto a otra herramienta o software que conozca, similar a esta, que opinión puede suministrar.

- Opiniones o recomendaciones en general.

Nota: Las preguntas han sido tomadas de cuestionarios creados para medir satisfacción subjetiva y usabilidad.¹

¹Chin, J.P., Diehl, V.A., Norman, K.L. (1988) Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface. ACM CHI'88 Proceedings, 213-218.

D. Desarrollo software

Metodología

En la Tabla 1, se muestra la planificación del desarrollo del trabajo, distribuido en 5 etapas o ciclos, según la clasificación de los requerimientos.

Sprint	General	Configuración del estímulo	Configuración de pruebas
1	- Definición de características y diseño de la interfaz de usuario.	- Definición de las formas - Definición de colores - Definición de escala y rotación	- Definición de parámetros para los test - Integración de paradigmas Gap y Overlap
2	- Configuración de colores y estilo de la interfaz.	- Definición de propiedades gráficas - Implementación de la paleta de colores. - Configuración de imágenes como estímulos	- Definición de formato XML para guardar datos de pruebas - Configuración de sistema para guardar datos
3	- Configuración del área de trabajo definida para las herramientas empleadas.	- Implementación de animaciones - Definición de trayectorias	- Habilitar la iteración de elementos para repetir pruebas - Agrupación inicial de test (sacadas visuales) - Implementación inicial de previsualización
4	- Configuración de cuadros de dialogo auxiliares.	- Implementación de trayectorias en vertical y horizontal - Sincronización del diseño en tiempo real - Implementación de tabla y controles para exportación y visualización de parámetros	- Integración final de test - Definición y configuración de aleatoriedad
5	- Establecer el sistema de comunicación entre las dos secciones del trabajo.	- Configuración de la exportación de parámetros en archivos XML	- Configuración de sistema para selección de estímulos - Implementación final de previsualización

Tabla 1: Ciclos definidos para el desarrollo del proyecto, clasificados por requerimientos

Para el desarrollo de estas etapas se estableció un tiempo de 4 semanas para cada una. Al final de cada *sprint* se realiza una realimentación con base en el trabajo desarrollado, para redefinir o añadir nuevos objetivos a la siguiente etapa.

Desarrollo software

Widgets utilizados en la interfaz

Los *widgets* son recursos que el entorno *Qt Designer* ofrece para diseñar una interfaz de usuario. Estos elementos pueden ser personalizados tanto en el aspecto como en

funcionalidad. Como se mencionó anteriormente, además de estos *widgets* se utilizaron elementos diseñados especialmente para la plataforma *Oscann desk100*, los cuales se detallan más adelante. Estos elementos se diseñan como archivos QML que posteriormente se adjuntan al proyecto para ser utilizados.

QML	Descripción	Figura								
ComboBox	Botón con lista desplegable									
Button	Botón pulsador									
CheckBox	Casilla de verificación	<input checked="" type="checkbox"/> Breakfast <input type="checkbox"/> Lunch <input checked="" type="checkbox"/> Dinner								
Slider	Control de valores de tipo deslizador									
Switch	Botón interruptor									
Text	Ítem de texto plano o HTML	Hello World! Hello World!								
TextField	Campo o editor de texto de una sola línea	Enter name								
TableView	Lista de elementos en un entorno de tabla redimensionable.	<table border="1"> <thead> <tr> <th>Title</th> <th>Author</th> </tr> </thead> <tbody> <tr> <td>A Masterpiece</td> <td>Gabriel</td> </tr> <tr> <td>Brilliance</td> <td>Jens</td> </tr> <tr> <td>Outstanding</td> <td>Frederik</td> </tr> </tbody> </table>	Title	Author	A Masterpiece	Gabriel	Brilliance	Jens	Outstanding	Frederik
Title	Author									
A Masterpiece	Gabriel									
Brilliance	Jens									
Outstanding	Frederik									
FileDialog	Proporciona un selector de archivos básico: permite al usuario seleccionar o crear archivos en directorios locales.									
ColorDialog	Ventana de dialogo auxiliar para selección de colores.									

Tabla 2: *Widgets* utilizados para el desarrollo de la interfaz

En la tabla 2 que se presentan los principales elementos utilizados en el diseño de la interfaz, las imágenes adjuntas corresponden a elementos con el aspecto predeterminado de cada clase, algunos de estos se personalizaron según requerimientos del trabajo mientras que otros fueron reemplazados por elementos QML con características especiales, tanto visuales como funcionales. Los archivos correspondientes a estos elementos fueron

suministrados por el *Dr. Erik Hernández*, jefe técnico de la empresa y quien estuvo a cargo del desarrollo de este trabajo.

Desarrollo del *Sprint 1*

El diseño gráfico general debe respetar las siguientes características:

- Todo el diseño debe estar contenido en una sola ventana, de ser necesario se deben desplegar recuadros auxiliares en la ventana principal, esta disposición se realiza para mantener la compatibilidad con el software de la plataforma *OSCANN*, en la cual únicamente se permite vincular proyectos contenidos en una sola ventana o ítem. Por otra parte, esta configuración mantiene la información de las pruebas siempre visible y evita que el especialista médico deba navegar por otras ventanas que pueden generar distracción en él.
- La interfaz debe integrar controles específicos e intuitivos que proporcionen una configuración sencilla de las pruebas y un diseño fluido de los estímulos, además, deben ser visibles todos los parámetros configurados para la supervisión del especialista médico, de esta manera se facilita la exportación de los datos en archivos XML para la comunicación entre las interfaces del trabajo o entre la plataforma *OSCANN*.

Configuración del estímulo

- **Definición de las formas:**

Las formas que se habían definido previamente corresponden a: círculo, cuadrado, estrella y triángulo. Inicialmente el desarrollo de este requerimiento fue configurado basándose en imágenes que eran superpuestas una sobre otra, donde no se obtuvieron los resultados esperados debido a dificultades con la integración de animaciones y efectos gráficos, por consiguiente, se decidió optar por una alternativa distinta. Las formas se muestran en la Figura 27.



Figura 27: Formas definidas para el estímulo

Se recurrió a utilizar el entorno Canvas de HTML, el cual permite una variedad de características como: la generación de gráficos dinámicamente por medio de secuencias de código, renderización interpretada dinámica de gráficos 2D y mapas de bits, así como

animaciones basadas en los gráficos generados. Este elemento está enmarcado en un modelo de procedimiento de bajo nivel, que actualiza un mapa de bits y no tiene una gráfica de escena integrada.

Cabe destacar que en QML existe una personalización propia de este elemento, donde es posible realizar la implementación con instrucciones en lenguaje JavaScript, facilitando la integración con las animaciones disponibles en Qt, razón por la cual se valida el uso de esta herramienta.

En la secuencia de código 1, se realiza la implementación de un círculo, haciendo uso del elemento antes mencionado. Aquí se establece inicialmente una variable de contexto para preparar el lienzo o mapa de bits donde se realiza el trazo de la figura, en este caso es una imagen en 2D. Luego se define el grosor de la línea con la que será dibujada la forma y un centro para trazar un arco a partir de una función matemática, obteniendo así la circunferencia deseada. Finalmente se rellena la figura y se cierra el lienzo para no añadir más acciones de dibujo.

```

1 Canvas{
2     onPaint: {
3         var ctx = getContext("2d")
4         ctx.lineWidth = 2
5         var centreX = width/2
6         var centreY = height/2
7         ctx.beginPath()
8         ctx.strokeStyle = circleBig.cambiarCircleBigLine
9         ctx.fillStyle = circleBig.cambiarCircleBigFill
10        ctx.arc(centreX,centreY,width/4,0,2*Math.PI)
11        ctx.closePath()
12        ctx.fill()
13        ctx.stroke()
14        ctx.restore()
15    }
16 }

```

Código 1: Instrucciones para crear una forma con el elemento Canvas

■ Definición de escala y rotación:

Para la presentación del estímulo es necesario permitir la edición del tamaño y la orientación del objeto, por lo tanto, en la interfaz se disponen de controles para configurar estos parámetros. La escala del estímulo se define en un rango de 50 a 300 píxeles y la rotación se realiza de 0 a 360 grados.

Los controles implementados para esta tarea corresponden a: *Slider*, que permite mover un control en un rango definido y asignar valores dentro de sus límites. *TextField*, que permite la visualización e introducción de texto por parte del especialista médico. Aquí se realizan las restricciones pertinentes, con el fin de no admitir valores inferiores o superiores a los estipulados anteriormente. A continuación se indica el código 2, donde se puede observar la restricción antes mencionada, además de la asignación y visualización de los valores de *TextField* hacia el *Slider*. En la Figura 28, se visualiza la sección que contiene estos controles y una comparación de cambios en sus valores.

```

1  Slider {
2      id: sliderSize
3      minimumValue: 50
4      maximumValue: 300
5  }
6  TextField {
7      id: textSize
8      text: sliderSize.value
9      validator: IntValidator{bottom: 50;top:300}
10 }

```

Código 2: Instrucciones para crear una forma con el elemento Canvas

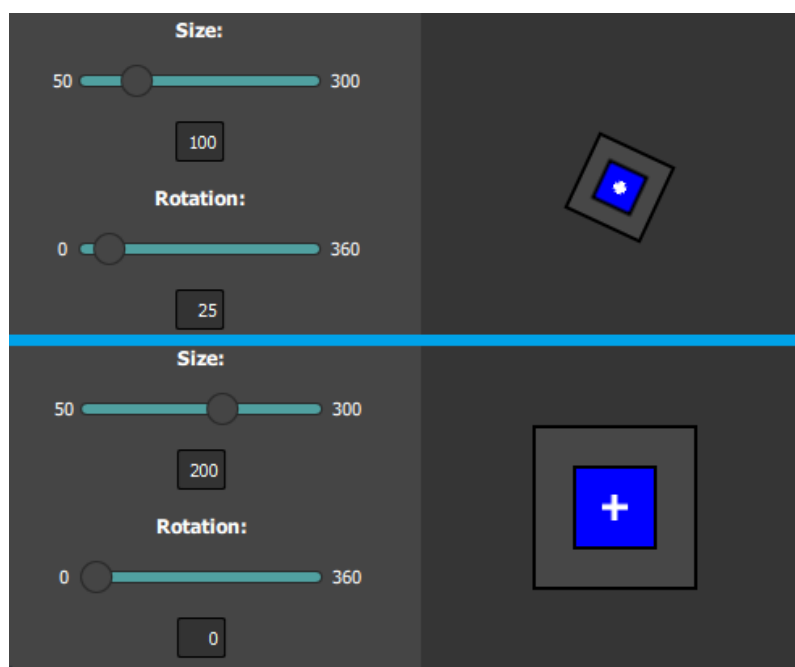


Figura 28: Cambios en la escala y rotación del estímulo

Configuración de pruebas

Teniendo en cuenta las recomendaciones del apartado visual para la interfaz, el desarrollo inicial realizado para esta sección fue el diseño de la ventana principal. Inicialmente para cada test se despliegan dos recuadros para la configuración de la prueba; uno de ellos contiene los espacios disponibles para ingresar la información de los parámetros a configurar y en la siguiente se encuentra una tabla de visualización donde se despliega la información o datos que se van ingresando para la prueba.

En la Figura 29 se puede observar el apartado para configurar test de sacadas visualmente guiadas. En la sección 1 se agrupan las opciones para cambiar los parámetros de configuración según el tipo de test. En la segunda sección 2 se despliegan los parámetros correspondientes a cada test, en este caso test de sacadas visualmente guiadas. En la

imagen se pueden observar variables como posición, duración (tiempo de visibilidad), la dirección para una sacada, entre otros. Conforme se adicionan elementos se incrementan los datos de la tabla (3) hasta que finalice la configuración de la prueba. Si el especialista médico desea modificar algún dato de los que haya guardado anteriormente, basta con eliminar la información anterior y escribir los nuevos datos.

En la sección 2 también se encuentran disponibles los paradigmas Gap y Overlap, inicialmente estas variables solo son habilitadas para este tipo de test. Cabe resaltar que esta Figura (29) corresponde a la interfaz desarrollada hasta el sprint 3, por esta razón la ventana ya cuenta con los colores y botones indicados, botón de previsualización y botón para guardar los datos de la prueba. Estos avances se describirán en su respectiva sección.

The screenshot shows the 'Visual Guided Saccades' configuration interface. On the left, a sidebar labeled 'Test Type' contains buttons for: Visual Guided Saccades, Countermanding, Linear Smooth Pursuit, Sinusoidal Smooth Pursuit, Pupil light, Fixation, Optokinetic, and Fixed Image. The main area displays a table with columns: N°, Position X (Deg), Position Y (Deg), Initial Time (ms), Visible time (ms), Test Duration (seg) (ms), Gap (ms), Overlap (ms), and Task type. The table contains 16 rows of test data. At the bottom, a configuration panel allows setting: Horizontal/Vertical orientation, Oblique checkbox, Position X (range: -20° to 20°), Position Y (0), Duration (ms), Pro-saccade/Anti-saccade toggle, Visible/No visible toggle, Gap (ms), and Overlap (ms). A blue arrow button labeled 'Previsualización' is at the bottom of the panel. Red numbers 1, 2, and 3 indicate the sidebar, configuration panel, and table respectively.

N°	Position X (Deg)	Position Y (Deg)	Initial Time (ms)	Visible time (ms)	Test Duration (seg) (ms)	Gap (ms)	Overlap (ms)	Task type
1	0	0	0	1500	---	---	---	Fixation
2	1	3	1500	1400	2.9	2900	0	Pro-saccade
3	0	0	2900	1500	---	---	---	Fixation
4	1	3	4400	1400	5.8	5800	0	Pro-saccade
5	0	0	5800	1500	---	---	---	Fixation
6	1	3	7300	1400	8.7	8700	0	Pro-saccade
7	0	0	8700	1500	---	---	---	Fixation
8	1	3	10200	1400	11.6	11600	0	Pro-saccade
9	0	0	11600	1500	---	---	---	Fixation
10	1	3	13100	1400	14.5	14500	0	Pro-saccade
11	0	0	14500	1500	---	---	---	Fixation
12	1	3	16000	1400	17.4	17400	0	Pro-saccade
13	0	0	17400	1500	---	---	---	Fixation
14	1	3	18900	1400	20.3	20300	0	Pro-saccade
15	0	0	20300	1500	---	---	---	Fixation
16	1	3	21800	1400	23.2	23200	0	Pro-saccade

Figura 29: Secciones iniciales de la configuración de pruebas

Ahora bien, el código desarrollado para lograr esta interfaz resulta muy extenso para adjuntar en esta memoria, pero sí es necesario consignar un segmento importante de este, debido a su funcionalidad para agregar los datos de la sección 2 hacia la tabla. Las instrucciones realizan cálculos para establecer una secuencia de tiempo correcta entre cada elemento agregado. En este también se consideran los paradigmas Gap y Overlap, que cuando se utilizan, se deben realizar ajustes a los tiempos mencionados anteriormente.

Una vez se presiona el botón de agregar datos de la sección 2 (línea hacia la derecha), se ejecuta el código 3. La primera línea actualiza el tiempo de duración de la prueba (**drtn**), esto se realiza sumando los tiempos de duración de visibilidad del estímulo

más el paradigma Gap menos el paradigma Overlap. La siguiente instrucción establece el tiempo en el que el estímulo inicia su periodo de visibilidad y se guarda en la variable **to**, tiempo de inicio. A continuación, se convierte la duración actual de la prueba en segundos, adicionalmente se agrega una función para fijar la cantidad de decimales, en este caso 1. En la línea 5, la función **append** permite agregar información a una tabla a través de un modelo de listas, en este caso llamado **libraryModel**. Este modelo se establece como un parámetro a la tabla y se actualiza de esta manera, así la información contenida en la lista se mostrará en el entorno de tabla. Cada ítem corresponde a un elemento de una fila mostrada en la tabla y los datos se obtienen de los campos de texto editables. Por ejemplo en la línea 7, "**posx**", corresponde a una columna de la tabla para ingresar la posición en X del estímulo y la línea **posx_target.text**, hace referencia al texto disponible en el TextField llamado *posx_target*. De esta manera se ingresan los datos de toda una fila para establecer los parámetros de un elemento de la prueba.

Como se puede apreciar en el código, para realizar operaciones matemáticas se debe multiplicar cada variable por 1 para convertir texto en escalares, este ajuste es importante porque los campos de texto no distinguen de número o letras. Una característica importante de esos TextField es su propiedad de restricción, gracias a esta función se puede limitar un campo de texto para recibir caracteres específicos o números en un rango limitado. Por ejemplo, para el campo disponible para ingresar la posición en X, se restringe para ingresar valores entre -20 a 20 grados. En la declaración de las propiedades del TextField se establece la restricción a través de la propiedad *validator*, de la siguiente manera, **validator: IntValidator bottom: -20; top:20**, así, el editor de texto queda restringido de la manera descrita anteriormente.

```

1 drtn=drtn*1+dur_target.text*1+gap_target.text*1-over_target.text
   *1
2 to=drtn*1-dur_target.text*1
3 drtn_seg=(drtn*1/1000*1).toFixed(1)
4 drtn_seg=(drtn_seg*1).toFixed(1)
5 libraryModel.append({
6     "number":libraryModel.count*1+1,
7     "posx": posx_target.text,
8     "posy": posy_target.text,
9     "to": to,
10    "time": dur_target.text,
11    "durttime_seg": drtn_seg,
12    "durttime_ms":drtn,
13    "gap": gap_target.text,
14    "over": over_target.text,
15    "count":count_target.text,
16    "posX_aux": posxBan,
17    "posY_aux": posyBan,
18    "visibleOpc": ban4_1,
19    "type":"stml"
20 })

```

Código 3: Instrucciones para agregar datos a la tabla de visualización

Es importante mencionar que en este punto del trabajo, la interfaz de configuración de pruebas permitía ingresar datos únicamente del test de sacadas visuales, desarrollo que se complementa en la siguiente etapa cuando se muestra el método para guardar la información en los archivos XML. Los valores de Gap y Overlap también se encontraban disponibles únicamente para test de sacadas visuales, hasta ese momento. En la Figura 29 se pueden observar dos espacios en los cuales se puede ingresar estos valores, pero inicialmente estos se determinaban a partir de una diferencia entre tiempo de inicio y duración de la visualización del estímulo, es decir, para el cálculo de un paradigma se debía tener presente la duración actual de la prueba para establecer un valor de tiempo antes (Overlap) o después (Gap) de este. Durante el desarrollo de este Sprint, se fueron tomando decisiones que llevaron a cambiar este método para establecer estos paradigmas, se definió una manera más sencilla en la cual el especialista médico no deba realizar cálculos o estar pendiente del tiempo actual de la prueba, instrucción que se indicó en el código 3, en la línea 1.

En síntesis, el trabajo realizado para el apartado de configuración de pruebas para esta etapa fue en su gran mayoría desarrollo gráfico y definición de parámetros de los distintos test. Una vez evaluada esta etapa, se determinó que la sección 1 mostrada en la Figura 29, debía ser actualizada para la fase de integración de los test. La configuración propuesta es cambiar todo el recuadro por un ComboBox que contenga la lista de los test disponibles y agregarlo a la sección 2. Esta modificación se realiza en la etapa 3 de desarrollo, en esta se describe detalladamente el cambio y las ventajas que representa.

Desarrollo del *Sprint 2*

En esta etapa se configuró la interfaz con los colores y diseño original de la plataforma *OSCANN*. En esta sección se describe el proceso de integración de los archivos necesarios para esta personalización.

Cada archivo debe tener una extensión `.qml` para ser compatible con el proyecto. Los archivos se agregan a la carpeta principal del proyecto directamente y para habilitar su funcionamiento se debe modificar el archivo `.QRC`, en este se encuentra la lista de los recursos del proyecto, desde los archivos principales hasta las imágenes utilizadas en éste. En este archivo, que lleva el nombre del proyecto, se ingresa el nombre del nuevo archivo QML así, `<file>qml/AuraButton.qml</file>`. Una vez agregada esta línea, se guarda el archivo y el nuevo componente estará listo para su uso. En la Figura 30 se pueden observar los archivos utilizados para el desarrollo de este trabajo.

Los archivos con el nombre de “*Aura*” y “*Oscann*”, son los componentes QML utilizados para personalizar la interfaz diseñada. La declaración de cada *widget* en el editor de QtCreator es muy sencilla gracias a la función de autocompletado de la herramienta, basta con ingresar la primera palabra para encontrar el elemento que se pretende utilizar. Las propiedades de cada elemento son configurables, pero para mantener una simetría en la forma y colores de estos, se estableció no modificar ciertas características. Cuando los componentes permiten ingresar texto, el tamaño del elemento se adapta a estos datos, aunque también se puede establecer un tamaño fijo. Los colores de la interfaz son iguales a los de la plataforma original de *Oscann desk100*, de esta manera cuando se realice la integración al sistema, se respetan las mismas características visuales del sistema actual. A continuación, se realiza una breve descripción de cada componente utilizado.

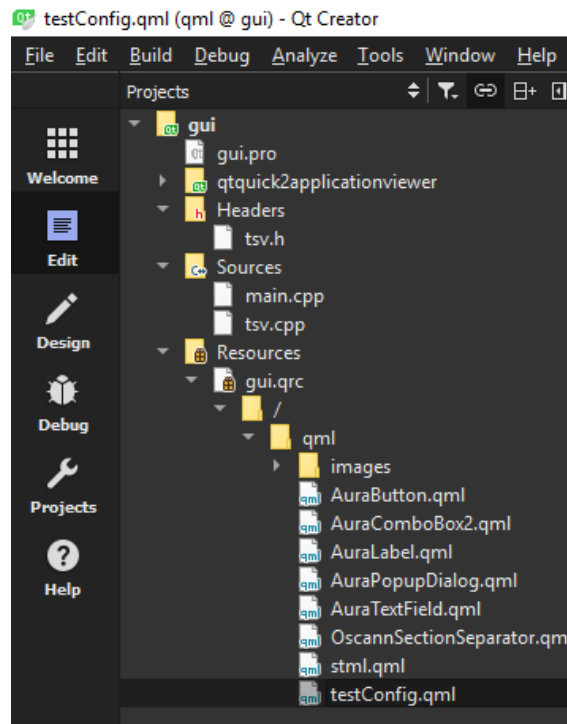


Figura 30: Árbol de archivos del proyecto

- **AuraButton:** Es un botón tipo pulsador con una apariencia distinta al estilo estándar, en la imagen 29, se pueden observar los botones utilizados para este apartado de la interfaz. Todos los botones que se aprecian en la sección 1 son de este tipo. Se construye a partir de un entorno rectangular, posteriormente se configura para darle un aspecto más común de botón y se asignan los colores respectivos.
- **AuraComboBox2:** Este se diseñó a partir de un *widget* ComboBox estándar, es decir, se creó un componente con las mismas propiedades del ComboBox pero con un estilo diferente. Esta personalización resulta un poco extensa en su implementación, por esta razón se decide crear un componente QML para reutilizar fácilmente.
- **AuraLabel:** Al igual que un *widget* tipo *Text*, permite mostrar un texto plano no editable. De ahí su nombre *Label*, refiriéndose a un ítem o etiqueta ideales para nombrar componentes dentro de una interfaz gráfica. A diferencia de un Text o Label común, este cuenta con colores distintos en sus propiedades.
- **AuraPopupDialog:** Este componente no tiene un *widget* similar, se trata de una ventana de diálogo auxiliar para varias tareas. Para este trabajo se utilizó como mensajes de ayuda, advertencia o para ingresar información cuando se finaliza la configuración de la prueba. Está diseñado con varios componentes o entornos, entre ellos, *ComboBox*, *CheckBox*, *TextField*, animaciones e incluso los botones son de tipo *AuraButton*. La implementación de este es un poco más compleja, el proceso se explicará en la etapa respectiva (Sprint 4).
- **AuraTextField:** Al igual que el componente *AuraComboBox*, se crea a partir del *widget* original, en este caso un *TextField*. La configuración de colores elegida tam-

bién permite que su apariencia cambie dependiendo su estado, una configuración de colores distinta cuando se encuentra habilitado o deshabilitado.

- ***OscannSectionSeparator***: Se trata de un encabezado especial para diferencias varias secciones mediante un recuadro con un nombre. Se puede observar un ejemplo en la Figura 29, en la parte superior de las secciones 1 y 3, en los que se muestra el título, *Test Type* y *Visual Guided Saccades*, respectivamente. Este se diseñó por la falta de un componente específico para definir un encabezado para cada elemento en una interfaz.

Los anteriores componentes reemplazan a sus similares en la implementación de la interfaz, componentes indicados en la tabla 2. Además de estos, *widgets* de tipo *CheckBox* y *Slider*, se personalizaron intentando seguir la línea de diseño original de la plataforma a falta de componentes de esta clase diseñados específicamente, como los de tipo “*Aura*”.

Por otra parte, el componente *stml.qml* es un elemento diseñado especialmente para este trabajo. Este elemento es el que permite realizar el enlace entre las secciones de diseño de estímulo y la sección de configuración de pruebas a través de un componente que permite visualizar los estímulos previamente diseñados, en el apartado de configuración de pruebas. En la última fase de desarrollo se detalla el funcionamiento de este (Sprint 5).

Configuración del estímulo

- **Definición de propiedades gráficas:**

Además de la configuración de colores que el estímulo contiene en su interior, se define aplicar efectos gráficos al estímulo, correspondientes a: brillo, contraste, saturación y difuminación. Para el brillo y el contraste se dispone un rango de -1 a 1 y para la saturación de 0 a 1, con la finalidad de modificar los valores aplicables en estas propiedades. Los controles implementados en este caso son idénticos a los que se mencionaron en la configuración de escala y rotación del estímulo. Para brillo y contraste, se hace uso del elemento *BrightnessContrast*, y en la saturación se emplea el elemento *Desaturate*. En el código 4, se observa la implementación de estos elementos, donde se asignan los valores de los controles a las variables de brillo, contraste y saturación. En la Figura 31 se muestran modificaciones en la saturación.

Es necesario destacar que los efectos gráficos se aplican a través de capas, adquiriendo una nueva propiedad sobre un trazado anterior.

```

1  BrightnessContrast {
2      id: effectIm2
3      anchors.fill: effectIm
4      source: effectIm1
5      brightness: sliderBrightness.value
6      contrast: sliderContrast.value
7      antialiasing: true
8      visible: true
9  }
10
11  Desaturate {

```

```

12   id: effectIm3
13   anchors.fill: effectIm2
14   source: effectIm2
15   desaturation: sliderSaturation.value
16   visible: true
17   scale: 1
18 }

```

Código 4: Configuración de brillo, contraste y saturación

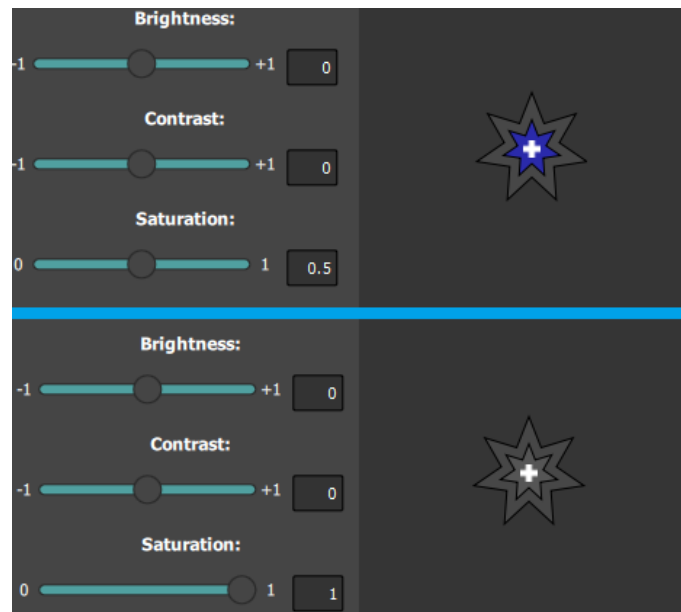


Figura 31: Cambios de saturación en el estímulo

Para la difuminación se utilizan tres parámetros diferentes que corresponden a: Ángulo, longitud y muestras, donde, ángulo representa la orientación del efecto y su valor se puede variar en un rango de 0 a 180 grados. La longitud define la distancia en que se desea representar el efecto, para este parámetro se definieron dos rangos, un rango mínimo que va de 0 a 8 píxeles cuando el tamaño del estímulo presenta el valor más bajo, que corresponde a 50 píxeles, y un rango máximo que va de 0 a 50 píxeles cuando el tamaño del estímulo presenta el valor más alto, que corresponde en este caso a 300 píxeles.

La definición anterior se implementa en la aplicación de forma proporcional al aumento o disminución en el tamaño del estímulo, lo que garantiza la correcta visualización del efecto de difuminado sobre las formas implementadas con el entorno *Canvas* de HTML.

```

1 DirectionalBlur {
2   id: effectIm1
3   anchors.fill: imageOverlay
4   source: imageOverlay
5   angle: sliderAngle.value
6   length: sliderLenght.value
7   samples: sliderSamples.value
8   transparentBorder: true

```

```

9   visible: true
10  }

```

Código 5: Configuración del efecto de difuminación

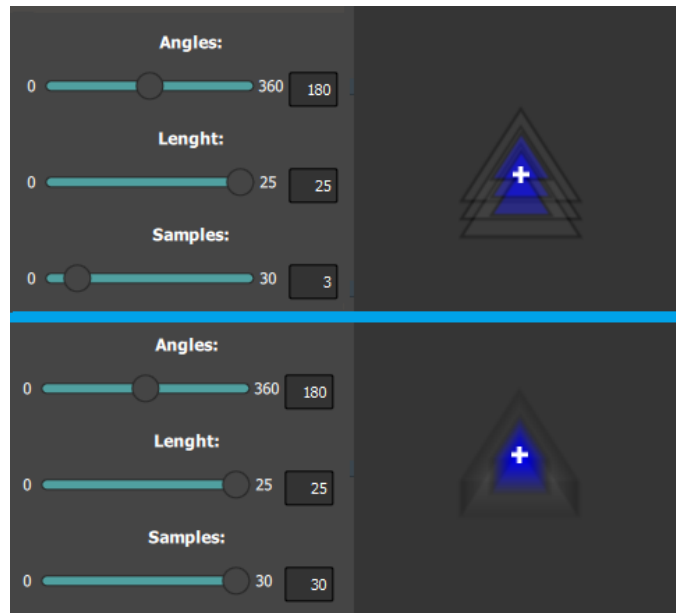


Figura 32: Variación en el numero de muestreos del efecto de difuminación en el estímulo

Finalmente, las muestras definen el número de veces que se desea replicar la imagen inicial. Cuando el número de muestras incrementa, la visualización del efecto de difuminado en el estímulo tiende a presentar una textura más suavizada y borrosa. El rango asignado a este parámetro va de 0 a 30 muestreos. La implementación de estas propiedades se realiza a través del elemento *DirectionalBlur*. Previamente, se muestra el código 5, en el que se desarrolla este requerimiento con las respectivas variables asociadas que permiten la aplicación del efecto de difuminado. Luego en la Figura 32, se presentan cambios en los parámetros de este efecto gráfico, específicamente en el número de muestreos.

Configuración de pruebas

Para lograr lo impuesto para esta etapa se estableció la comunicación entre el archivo principal del proyecto y una nueva clase C++, con el objetivo de guardar la información de la tabla de datos de la prueba en un archivo XML. Esta integración se realiza para ejecutar tareas más avanzadas, posibles con C++, en estos archivos se definen los parámetros y el procedimiento para exportar datos de la interfaz a un archivo externo de una manera más simple y eficiente. En la Figura30, consignada al principio de este *srpint*, se pueden observar los archivos `tsv.h` y `tsv.cpp` que hacen posible esta comunicación entre QML y C++. En el archivo de cabecera se establece la clase de este, en este caso se define como un *QObject* ideal para manipular objetos QML desde C++. También se definen los métodos de la clase y el número de parámetros que se importarán del archivo QML. Mientras que en el archivo fuente se define la estructura tanto de los métodos de la clase como del

archivo XML. Se definieron 3 métodos para la clase, uno para inicializar o preparar el archivo, otro que exporta la información y el último que finaliza la comunicación.

Ahora, la estructura definida para el archivo XML es la siguiente.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <data name="name">
3   <duracion_test></duracion_test>
4   <random></random>
5   <stimulus id="n">
6     <x_mostrar></x_mostrar>
7     <y_mostrar></y_mostrar>
8     <x_guardar></x_guardar>
9     <y_guardar></y_guardar>
10    <t_duracion></t_duracion>
11    <t_gap></t_gap>
12    <t_overlap></t_overlap>
13    <t_count></t_count>
14    <visible></visible>
15    <demo></demo>
16    <tipo></tipo>
17    <estimulo></estimulo>
18    <link></link>
19  </stimulus>
20 </data>

```

Código 6: Estructura de archivos XML, generados en la configuración de pruebas.

En este se pueden observar las etiquetas que definen el comportamiento de cada estímulo de la prueba. Cuando el test está conformado por varios elementos, las etiquetas que se replican son desde la línea 5 hasta la 19. Cada estímulo se identifica por un id que corresponde al orden en que se ingresaron a la tabla de datos de la prueba y se complementa con la información respectiva de cada estímulo. Cabe mencionar que esta estructura (6), es la versión definitiva, inicialmente se trabajó únicamente con datos de posición y duración, mientras se perfeccionaban los métodos de la clase.

Por otra parte, para hacer uso de la clase *tsv* se debe configurar una propiedad de esta en el archivo fuente del proyecto (*main.cpp*). Esto se logra agregando la siguiente línea, **`viewer.rootContext () ->setContextProperty("TSV", new tsv());`**, de esta manera se define el comando "TSV", para llamar los métodos de la clase en el archivo QML del proyecto. El siguiente código muestra su implementación.

```

1 TSV.open(qml_listName, drtn, random_opc);
2   for (var i = 0; i < libraryModel.count; i++) {
3     TSV.append(
4       libraryModel.get(i).number,
5       libraryModel.get(i).posx, libraryModel.get(i).posy,
6       libraryModel.get(i).posX_aux, libraryModel.get(i).
7         posY_aux,
8       libraryModel.get(i).time,
9       libraryModel.get(i).gap, libraryModel.get(i).over,

```

```

9         libraryModel.get(i).count,
10        libraryModel.get(i).visibleOpc,
11        libraryModel.get(i).demo,
12        libraryModel.get(i).type,
13        libraryModel.get(i).stml,
14        libraryModel.get(i).link
15    );
16    }
17    TSV.close();

```

Código 7: Instrucciones para usar los métodos de la clase *tsv*

Este código se encuentra contenido en una función utilizada cuando se finaliza la configuración de una prueba. Esta función se utiliza en un cuadro de dialogo auxiliar, que permite ingresar un nombre o identificador para el archivo XML y también se puede definir si la prueba se desea realizar en orden aleatorio o no. Como se puede apreciar en la línea 1, el método `open` utiliza los datos mencionados anteriormente y los asigna a las etiquetas `data_name` y `random` del archivo XML. La duración de la prueba `drtn` y los datos ingresados en el método `append`, se obtienen de la tabla de datos de la prueba. Como se observa en la línea 2, este método se llama un número de veces igual a la cantidad de estímulos contenidos en la prueba. Finalmente, el método `close` cierra el documento y el archivo se guarda en el directorio indicado en el método `open`.

El desarrollo de este sistema no se logró por completo en esta etapa, debido a la constante actualización de los datos que generaba la interfaz durante el desarrollo del trabajo. Hasta este punto la integración de los test no estaba completa y el proceso para guardar datos solo era habilitada para pruebas únicamente de sacadas visuales. Una vez se finalizó con la integración de los test (Sprint 4), se retomó el desarrollo de este sistema de guardado, obteniendo como resultado el código consignado anteriormente (7). Además, debido a la falta de pericia con las librerías de *Qt*, el desarrollo de la comunicación entre QML y C++ fue un poco dispendioso al principio, implementando funciones poco eficientes y que generaban muchos errores. El resultado final después de varias semanas de trabajo, por la razón comentada anteriormente, fue muy satisfactorio gracias al constante trabajo de optimización del código y pese a su simplicidad es un método eficiente que logra una comunicación ideal entre ambos entornos.

Desarrollo del *Sprint 3*

Configuración del estímulo

- **Implementación de animaciones:**

Se integran en esta etapa cinco animaciones correspondientes a: Parpadeo, contracción, giro y pivote del estímulo, además de animar el parpadeo del fondo de las pruebas. Cada una de las animaciones se encuentran descritas a continuación.

Parpadeo del estímulo: Esta animación corresponde a un cambio de colores en el área externa del estímulo, se define para su ejecución el tiempo de duración, donde se realiza una animación secuencial del cambio de estado entre dos colores. Aquí se presentan

dos *AuraButton* que permiten desplegar el cuadro auxiliar de la paleta de colores para la elección correspondiente, y un *AuraTextField* para la introducción del tiempo que el especialista médico desea para la duración de la transición entre un color y otro. En el código 8, se encuentra la configuración del parpadeo, y en la Figura 33, una representación de la animación.

```

1 SequentialAnimation on cambiarStarBigFill {
2     id: colorSBBlink
3     loops: Animation.Infinite
4     running: false
5     ColorAnimation {
6         id: colorSBBlinka
7         from: colorBlink1Dialoga.currentColor
8         to: colorBlink1Dialoga.currentColor
9         duration: sliderTimeBlink1.value
10    }
11    ColorAnimation {
12        id: colorSBBlinkb
13        from: colorBlink2Dialoga.currentColor
14        to: colorBlink2Dialoga.currentColor
15        duration: sliderTimeBlink1.value
16    }
17 }

```

Código 8: Configuración del parpadeo en el estímulo

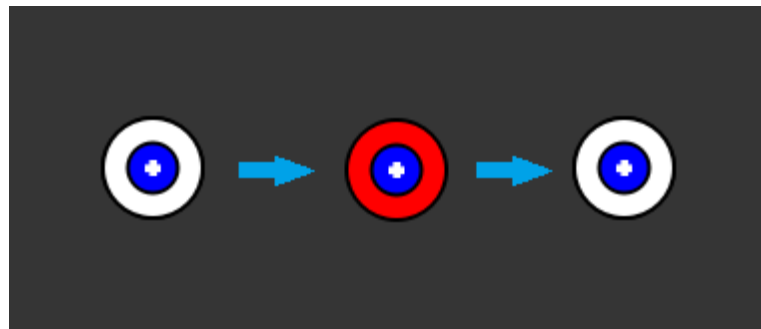


Figura 33: Animación de parpadeo en el estímulo

Contracción del estímulo: Para ejecutar la animación de contracción es necesario establecer dos escalas diferentes del estímulo, donde se define un valor superior y uno inferior, asignando un rango para el cambio en el tamaño, esta animación se lleva a cabo secuencialmente, a través de una variación sinusoidal, que permite la transformación uniforme de los valores contenidos en estos límites, durante un periodo de tiempo de un segundo para su ejecución, se debe resaltar que el tiempo predefinido no es modificable. Se dispone en esta sección de dos *AuraTextField* para la introducción del valor de las escalas de contracción, dadas en porcentaje. En el código 9, se encuentra la configuración de la contracción, y en la Figura 34, una secuencia similar a la animación.

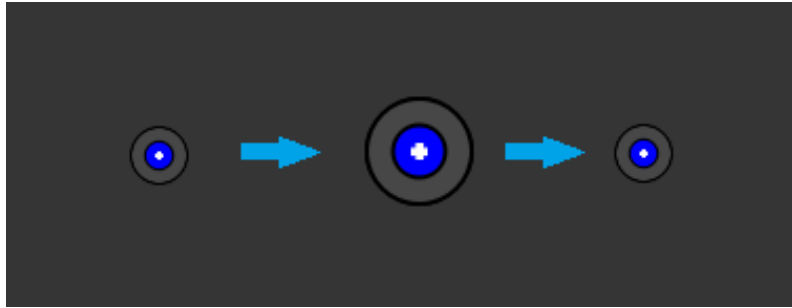


Figura 34: Animación de contracción en el estímulo

```

1 SequentialAnimation on cambiarStarBigFill {
2   id: colorSBblink
3   loops: Animation.Infinite
4   running: false
5   ColorAnimation {
6     id: colorSBblinka
7     from: colorBlink1Dialoga.currentColor
8     to: colorBlink1Dialoga.currentColor
9     duration: sliderTimeBlink1.value
10  }
11  ColorAnimation {
12    id: colorSBblinkb
13    from: colorBlink2Dialoga.currentColor
14    to: colorBlink2Dialoga.currentColor
15    duration: sliderTimeBlink1.value
16  }
17 }

```

Código 9: Configuración de la contracción en el estímulo

Giro del estímulo: El efecto de giro se visualiza a través de una animación secuencial, donde en un periodo de dos segundos se realiza una variación lineal de los valores de orientación comprendidos entre 0 y 360 grados. Esta configuración se encuentra reflejada en el código 10, donde ninguna de estas variables se puede modificar. En la interfaz solamente se dispone de un *Switch* para ejecutar la animación. La Figura 35 representa la secuencia de la animación de giro.

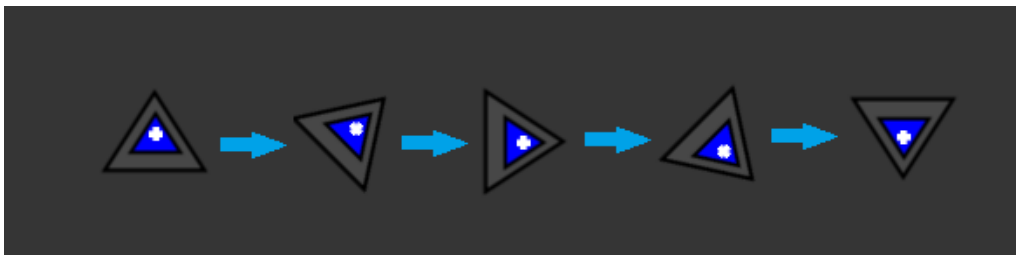


Figura 35: Animación de giro en el estímulo


```

1 SequentialAnimation on rotation {
2   id: rotationCBMovement
3   loops: Animation.Infinite
4   running: false
5   NumberAnimation {
6     duration: 2000
7     from: 0
8     to: 360
9     easing.type: "Linear"
10  }
11 }

```

Código 10: Configuración del giro en el estímulo

Pivote: En este caso se establecen dos límites para la ejecución de esta animación, cuyos valores corresponden a -30 y 30 píxeles. Se realiza la variación sinusoidal de los valores comprendidos en este rango, donde la duración de esta animación secuencial es de 500 mili-segundos. Pueden modificarse los valores de pivote a través de dos *AuraTextField*, además de la orientación del pivote que se relaciona de forma opuesta al eje definido para realizar trayectorias. En el código 11, se indica la configuración de la animación de pivote, y en la Figura 36, la demostración del movimiento.

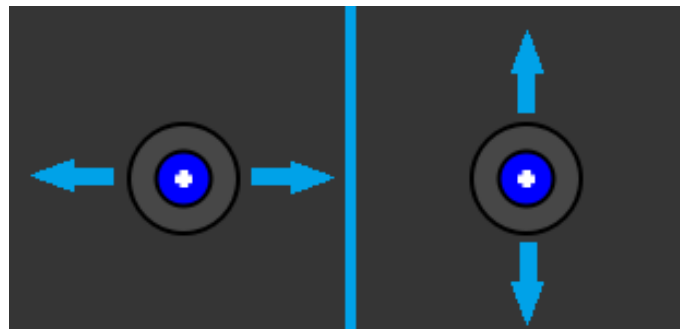


Figura 36: Animación de pivote en el estímulo

```

1 SequentialAnimation on nuevo2 {
2   id: pivotMovement
3   loops: Animation.Infinite
4   running: false
5   NumberAnimation {
6     id: pivotMovementa
7     duration: 500
8     from: sliderPivot1.value
9     to: sliderPivot2.value
10    easing.type: "SineCurve"
11  }
12 }

```

Código 11: Configuración del pivote en el estímulo

Parpadeo del fondo de las pruebas: Esta animación tiene una configuración idéntica a la de parpadeo en el estímulo, aunque los controles y secciones para configurar operan independientemente. En la Figura 37, se ilustra la secuencia realizada en esta animación, y en la Figura 38, la sección de controles definida en la interfaz para la configuración de animaciones.

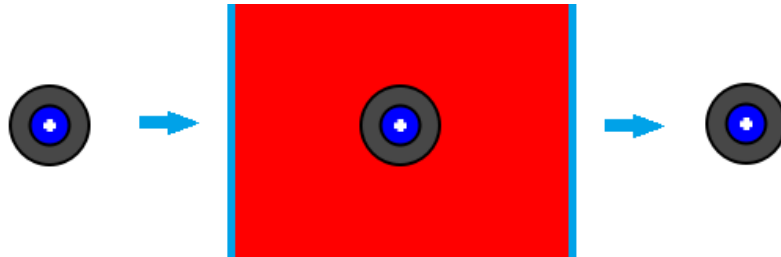


Figura 37: Animación del parpadeo en el fondo de las pruebas



Figura 38: Sección de controles para configuración de animaciones

■ Definición de trayectorias:

Las trayectorias a implementar se han definido como todos los movimientos disponibles a través del elemento *Animator* de QML, específicamente con *EasingTypes*, donde se presenta una cantidad considerable de funciones y sus gráficas correspondientes. Algunos de estos movimientos se muestran en la Figura 39. Estos elementos se pueden incluir en las animaciones secuenciales de alguna variable, en el caso de las trayectorias modifican la posición del estímulo en cualquiera de los dos ejes.

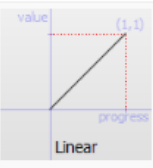
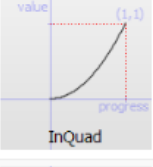
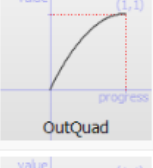
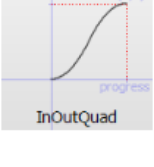
Easing.Linear	Easing curve for a linear (t) function: velocity is constant.	
Easing.InQuad	Easing curve for a quadratic (t^2) function: accelerating from zero velocity.	
Easing.OutQuad	Easing curve for a quadratic (t^2) function: decelerating to zero velocity.	
Easing.InOutQuad	Easing curve for a quadratic (t^2) function: acceleration until halfway, then deceleration.	

Figura 39: Algunas trayectorias definidas para el desplazamiento del estímulo

Configuración de pruebas

■ Agrupación inicial (sacadas visualmente guiadas)

Como se menciona anteriormente, los test de sacadas, antisacadas, sacadas y antisacadas de memoria, y sacadas de tipo *countermending*, comparten muchos parámetros para su definición. Durante la definición de los test en la primera etapa, la configuración de todos los test se realizaba de manera independiente, es decir, no se permitía diseñar pruebas que combinen varios tipos de test. Por ejemplo sí el especialista médico estaba configurando una prueba de antisacadas, la tabla de visualización de datos únicamente permitía ingresar datos de este tipo de pruebas, entonces, para continuar con la configuración de otro tipo de test, el usuario debía finalizar la prueba de antisacadas para liberar la tabla de datos y pasar a configurar otro tipo de test. De esta manera, se buscaba mantener la misma metodología utilizada actualmente por la plataforma, definir pruebas de manera independiente. Este enfoque se cambió con el fin de brindar mayor flexibilidad en el diseño de pruebas, permitiendo combinar componentes de todos los test en una sola prueba, adaptándose a las necesidades que el especialista médico determine para el paciente. Ahora, para la implementación de esta agrupación de test inicial, se configuró la sección 2 de la interfaz o apartado de parámetros del test, como se muestra en la Figura 40.

El *widget* marcado con el cuadro rojo es el *ComboBox* que despliega la lista de los test disponibles. Este reemplaza toda la sección 1 que se mostraba en la Figura 29, de esta manera se obtiene mayor espacio para desplegar la información de la tabla y los parámetros de cada test. Los elementos restantes se detallan a continuación.

- Con el switch *Horizontal/Vertical* y la casilla de verificación *Oblique*, se define la orientación del movimiento sacádico. Cuando la casilla de verificación se encuentra seleccionada, el switch se deshabilita, debido a que esta opción permite ingresar

Visual Guided Saccades

Horizontal Vertical

Oblique

Position X:

Position Y:

Duration:

Pro-saccade Anti-saccade

Visible No visible

Gap:

Overlap:

Countermanding:

	X	Y	Dur.	Gap	Ovlp.	Count.
Fixation:	0	0	1200			
Target:	15	0	1200	0	0	0

→

Set fixation time Preview +

Figura 40: Sección de parámetros para test de movimientos sacádicos

posiciones tanto para X como para Y. Como se puede apreciar en la imagen, cuando el switch se encuentra en posición *Horizontal*, únicamente se habilita el campo de texto de la posición en X. El caso opuesto se presenta cuando se selecciona la componente *Vertical*.

- Los primeros campos de texto corresponden a la posición del estímulo y la duración de visibilidad del mismo. Estos se encuentran restringidos a ciertos valores, para la posición en X se establece un rango de valores entre -20 y 20, que representan los grados en los que se distribuye la componente horizontal del monitor. Para la posición en Y es de -12 a 12 grados, una distribución menor por la longitud vertical del monitor, y por último la duración está limitada entre 500 y 5000 milisegundos.
- Los switch de *Pro-saccade/Anti-saccade* y *Visible/No visible*, permiten configurar el tipo de sacada del estímulo, todas las combinaciones se detallan en el manual de usuario para la configuración de pruebas consignado en el Anexo B.
- Las siguientes opciones permiten agregar un tipo de paradigma Gap u Overlap al estímulo o bien definir este como uno de tipo *countermanding*. Cuando se selecciona este último, los paradigmas Gap y Overlap no se habilitan, debido al comportamiento que representa un estímulo de tipo *countermanding*.
- El botón marcado con el cuadro amarillo cumple con la función de validar la información ingresada en los campos anteriores y despliega los elementos de la sección derecha de la figura. En la parte superior se pueden observar unos campos de textos que contienen la información suministrada en la sección anterior, los datos extra

que se indican con el nombre de *Fixation* se refieren al estímulo central que precede al estímulo que realiza movimiento sacádico. Esto se refiere a lo mencionado en la sección de requerimientos, cuando se habla sobre garantizar el comportamiento original de cada test. En este caso, un test de movimientos sacádicos se realiza siempre en 3 etapas (centro, desplazamiento y centro nuevamente), por lo tanto, para evitar que el especialista médico deba ingresar datos de un estímulo que siempre lleva las mismas coordenadas (0,0), este se genera automáticamente. Así pues, cada vez que se define un estímulo se ingresan 2 estímulos al test, el estímulo objetivo y el estímulo central, de esta manera, cada estímulo definido estará acompañado por un elemento que complementa al anterior, en la Figura 41 se ilustra este proceso.

Test Configuration									
Nº	Position X (Deg)	Position Y (Deg)	Initial Time (ms)	Visible time (ms)	Test Duration (seg)	Gap (ms)	Overlap (ms)	Countermanding (ms)	Type
1	0	0	0	2000	2.0	0	0	0	Central fixation
2	10	0	2000	2000	4.0	0	0	0	Pro-saccade

Test Configuration									
Nº	Position X (Deg)	Position Y (Deg)	Initial Time (ms)	Visible time (ms)	Test Duration (seg)	Gap (ms)	Overlap (ms)	Countermanding (ms)	Type
1	0	0	0	2000	2.0	0	0	0	Central fixation
2	10	0	2000	2000	4.0	0	0	0	Pro-saccade
3	0	0	4000	1500	5.5	0	0	0	Central fixation
4	0	7	5500	1500	7.0	0	0	0	Pro-saccade

Figura 41: Elementos complementarios en test de movimientos sacádicos

En la tabla superior, la primera fila marcada en azul, muestra los datos del estímulo de fijación central que acompaña al estímulo objetivo definido en la fila dos. En la tabla inferior se aprecia como los datos de la fila tres representan otro estímulo central que complementa al estímulo de la fila dos y también acompaña al elemento de la fila 4. Reiterando, el objetivo de realizar esta configuración representa un menor tiempo para diseñar una prueba y además, se garantiza que el número de elementos para un test de este tipo sea el correcto. Si el especialista médico desea puede establecer un tiempo de duración diferente para el estímulo de fijación central, pulsando el botón *Set fixation time*, que se puede observar en la parte inferior de la Figura 40.

El botón con el símbolo “+”, es el que ingresa los datos a la tabla de visualización. El botón de *Preview*, como su nombre lo indica es el que permite observar el comportamiento de los estímulos antes de ser agregados a la tabla de datos.

■ Previsualización inicial

La vista previa inicial se desarrolló únicamente para visualizar test de movimientos sacádicos, correspondiente con la agrupación inicial de test, descrita anteriormente. Para

esta etapa, la configuración de estímulos no existía y únicamente se presentaban circunferencias de color verde. Para la ejecución de los movimientos, las propiedades de posición eran modificadas y de esta manera se lograban representar movimientos sacádicos.

Los estímulos se diseñaron con el entorno *Canvas*, en este es posible diseñar figuras geométricas más precisas. A pesar de que estos círculos son muy simples, la herramienta *QtCreator* no cuenta con muchas figuras predeterminadas, únicamente rectángulos. Con estos lienzos de la herramienta *Canvas*, también se definió la cruz o indicador en su interior. El código que permite configurar el movimiento de los estímulos es muy simple, como se puede observar en las líneas 12, se definen unas propiedades de posición y de visibilidad para cada elemento, con estas se puede establecer la lógica del movimiento de cada tipo de test.

```

1 Rectangle{
2     property int position_X: 0
3     property int position_Y: 0
4     property int vsbl: 1
5     id:stimulus
6     width: 50
7     height: 50
8     radius: 50
9     color: "green"
10    anchors.centerIn: parent
11    anchors.horizontalCenterOffset: parent.width*position_X/42
12    anchors.verticalCenterOffset: parent.height*position_Y/26
13    z:vsbl
14    Canvas {
15        ...
16        ...
17    }
18 }

```

Código 12: Definición de estímulos para ventana de previsualización

En las líneas 2, 3 y 4 del Código 12, se definen las propiedades de posición y visibilidad del estímulo. Con las siguientes instrucciones se define el tamaño y color del estímulo, así como un nombre para identificar el elemento. Las propiedades de las líneas 11 y 12, hacen posible el desplazamiento del estímulo en la ventana de previsualización, en estas se encuentran las funciones que permiten ubicar al elemento según las posiciones establecidas. El parámetro *z*, asociado con la visibilidad del estímulo (*vsbl*), permite establecer el orden de apilamiento de un objeto. Se refiere al orden en que se dibujan los elementos en la ventana, por ejemplo, un rectángulo con valor de *z*=1 se dibujará sobre otro rectángulo con valor menor, en pocas palabras, es una propiedad para superponer o quitar un elemento. Ya que la ventana de visualización es un elemento de tipo rectángulo, posee propiedades iguales a los estímulos, por lo tanto, define un valor de apilamiento *z*=0 para la ventana y se modifican los valores de los estímulos entre -1 o 1 para lograr desaparición o visualización del elemento.

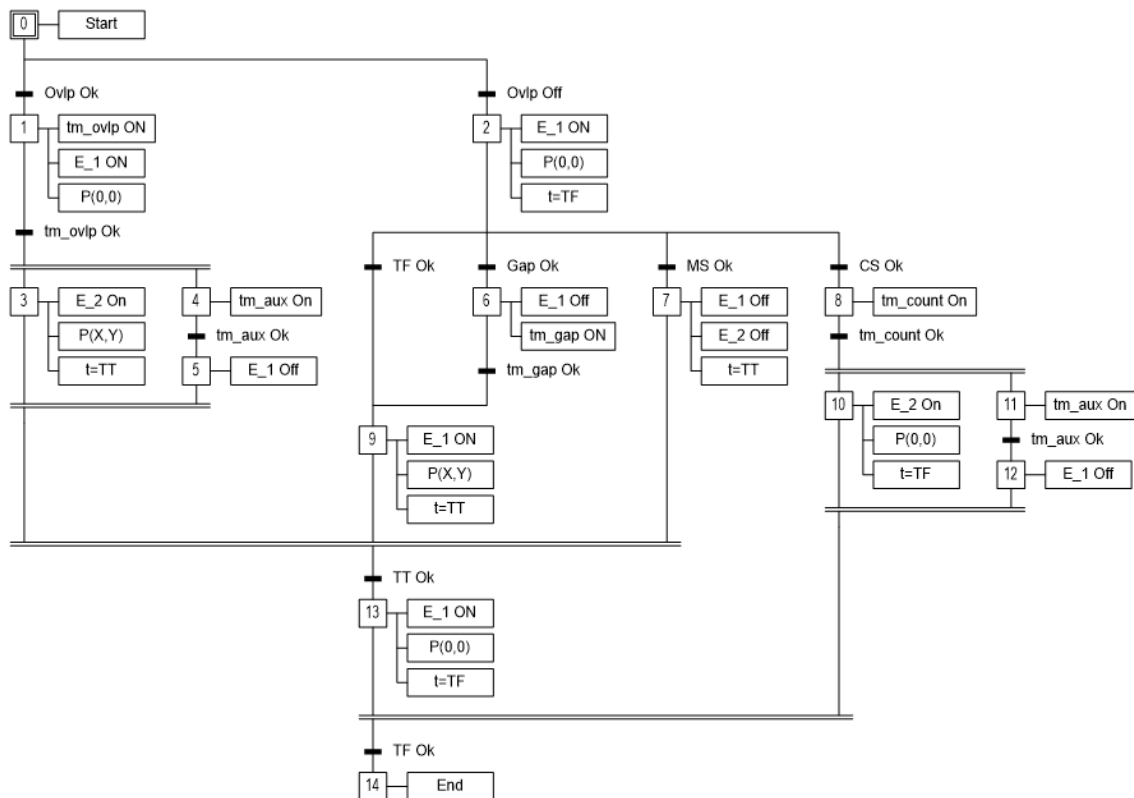


Figura 42: Grafcet del funcionamiento de previsualización

El anterior esquema describe el funcionamiento del algoritmo encargado de realizar la previsualización de los test. En este se indican los diferentes tipos de comportamientos que pueden tomar los estímulos, ya sea cuando contienen paradigmas Gap u Overlap (Ovlp), o cuando son sacadas de tipo *Conuntermanding* (CS) o de memoria (MS). Un movimiento sacádico normal está descrito por las etapas 2, 9 y 13 del diagrama, como se puede apreciar guarda congruencia con el funcionamiento natural de un movimiento sacádico realizado en 3 pasos, centro (0,0), desplazamiento del estímulo (X, Y), centro nuevamente. Las transiciones *Ovlp Ok*, *Gap Ok*, *MS Ok* y *CS Ok* representan las condiciones que debe cumplir un estímulo para ejecutar las instrucciones correspondientes de cada tipo de comportamiento. Mientras que las transiciones restantes representan algún cumplimiento de tiempo, por ejemplo, de un *Timer*. *E_X*, representa al estímulo número X, cuando se hace visible (ON) o desaparece (Off); *P(n,n)*, representa la posición del estímulo en el punto (n,n); TF, es el tiempo asignado para los estímulos centrales o de fijación en el punto (0,0) y TT es el tiempo que se ingresa para los estímulos objetivo o que se desplazan a los puntos P(X,Y).

Desarrollo del *Sprint 4*

La integración de los cuadros de diálogo correspondientes con el elemento QML *AuraPopupDialog*, resulta un poco más compleja que los *widgets* restantes. Como se detalló anteriormente, este *widget* cuenta con distintos elementos en su construcción y por esta razón se deben configurar ciertas funciones para su implementación. Para empezar, se

definieron 2 usos diferentes para este *widget*. El primero como ventana de aviso o advertencia, y el segundo como cuadro de dialogo para ingresar algún tipo de información. Para emplear este elemento es necesario definir ciertas funciones dependiendo del uso que se quiera, en el código se desarrolló una función para crear los cuadros de diálogo que recibe varios argumentos correspondientes a cada tipo de ventana.

```

1  property var qml_compPopUp: Qt.createComponent("AuraPopupDialog.
    qml");
2  function createPopupDlg(type, issue, info, details, btnType,
    emitOkSignal, fun){
3      ...
4  }

```

Código 13: Definición de estímulos para ventana de previsualización

Las anteriores líneas pertenecen al código del archivo principal del proyecto, con las cuales se define la variable que permite crear un componente a partir de un elemento externo, en este caso `AuraPopupDialog.qml` y la función `createPopupDialog` para crear los cuadros de dialogo. De igual forma, se deben definir muchas más variables para habilitar todas las funciones del *widget*. Los argumentos asociados a esta función, permiten definir, entre otras características, el estilo del cuadro, el texto y las funciones. El argumento `fun`, permite elegir la función definida para cada cuadro de dialogo, estas se encuentran agrupadas en una sentencia *Switch* donde cada caso permite llamar las funciones correspondientes al argumento `fun`.

Configuración del estímulo

■ Implementación de trayectorias en vertical y horizontal:

Una vez establecidas las trayectorias a implementar, se realiza la configuración correspondiente para visualizar los movimientos.

Se emplea un modelo de lista, donde se dispone el nombre de cada uno de los componentes de *EasingTypes*. Luego, a través de esta lista es posible visualizar los elementos en un *ComboBox*, donde se elige el comportamiento deseado con base en las gráficas de diferentes funciones matemáticas que expresan el movimiento.

Para esta animación se dispone de dos *Switch*, donde uno se encarga de activar el movimiento de trayectoria, y otro sirve para especificar en qué eje se desea realizar el desplazamiento, esta configuración de sección se indica en la Figura 43. En esta configuración, cuando se deciden ejecutar los desplazamientos en el eje X la duración de la animación es de 2,5 segundos, contrario a cuando se efectúa el movimiento en el eje Y, donde es de 1 segundo.

En el código 14 se presenta la configuración de este requerimiento, donde a través de una animación secuencial se realiza la variación de la posición, ya sea horizontal o vertical. Esta variación obedece a la asignación del *EasingType* que define el valor que toman estas posiciones durante el tiempo de ejecución. Los valores máximos y mínimos de cada eje para la posición final del estímulo en una trayectoria se definen con el ancho y alto del display donde serán ejecutadas las pruebas. En este caso debido a la configuración de secciones disponibles, se establece la posición inicial del estímulo en 0, donde tiene

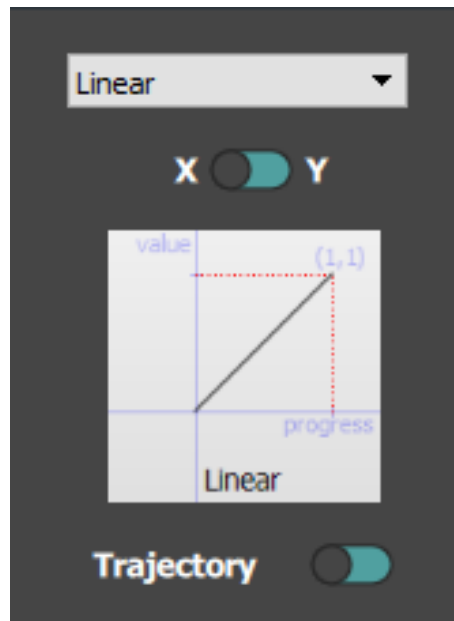


Figura 43: Controles de la animación de trayectorias

libertad de desplazamiento de -50 a 50 píxeles en el eje Y, y de -325 a 600 píxeles en el eje X. La última consideración se define de esta manera debido a un desplazamiento horizontal inicial del estímulo, que permite una visualización justificada de las secciones y el área de configuración en la interfaz. En la Figura 44 se ilustra el espacio definido en la aplicación para realizar los desplazamientos del estímulo basados en las trayectorias.

```

1 SequentialAnimation on nuevo {
2   id: xMovement
3   loops: Animation.Infinite
4   running: false
5   NumberAnimation {
6     duration: 2500
7     from: -325
8     to: 600
9     easing.type: movementType.currentText
10  }
11  NumberAnimation {
12    duration: 2500
13    from: 600
14    to: -325
15    easing.type: movementType.currentText
16  }
17 }

```

Código 14: Configuración del desplazamiento del estímulo basado en trayectorias

■ Sincronización del diseño en tiempo real:

Se realiza un ajuste a cada uno de los parámetros asignados a las animaciones y a los

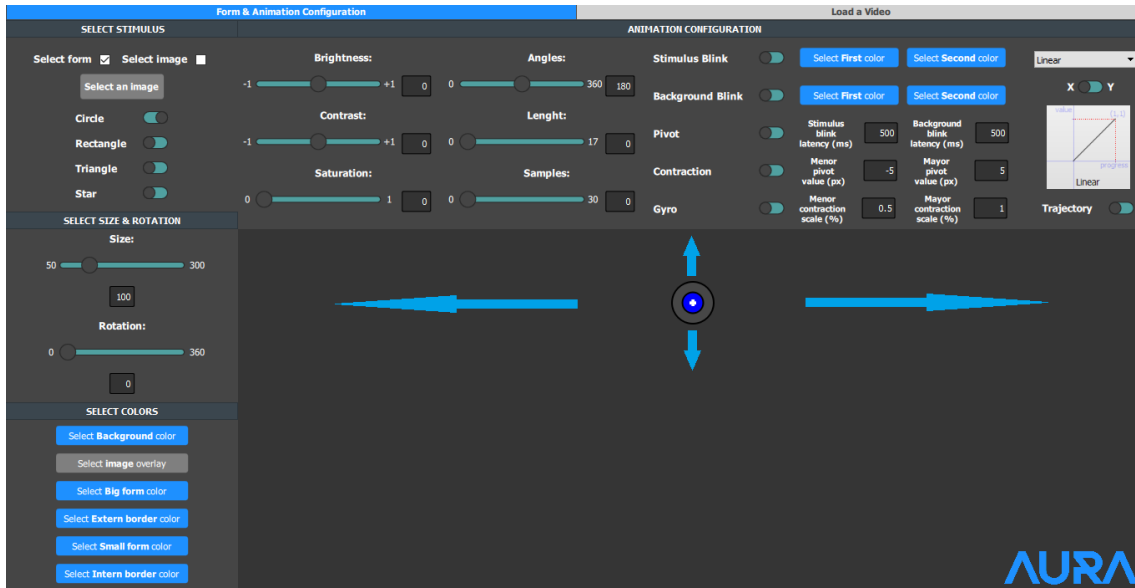


Figura 44: Espacio definido en la interfaz de configuración para el desplazamiento del estímulo

recursos estipulados en la interfaz, con el fin de que se vean reflejados en el estímulo de manera inmediata y esto no genere contratiempos en el desarrollo de las pruebas. De esta manera se establecen diversas instrucciones que permiten reiniciar las animaciones en el momento en que se registre el cambio en los parámetros asociados, de igual forma en la parte visual, el cambio de la configuración es consecuente de un cambio a nivel gráfico.

■ Implementación de tabla y controles para exportación y visualización de parámetros:

Se define una tabla en la interfaz con la finalidad de que el especialista médico tenga visibilidad de los parámetros que ha configurado, esta funcionalidad facilita la exportación de archivos en formato XML para la posterior integración con otras secciones en la configuración de pruebas.

Utilizando el elemento *ListModel* de QML, se pueden definir las etiquetas correspondientes a cada uno de los parámetros que serán añadidos a esta tabla de visualización. Inicialmente se utiliza el método “*clear*” para vaciar los elementos de un estado anterior en la tabla y luego con el método “*append*” se asigna cada uno de los elementos configurados a su respectivo ítem en el modelo de lista, luego este elemento se carga en un *TableView* como se muestra en el código 15, donde se visualiza la configuración generada. En la Figura 45 se presenta la tabla implementada, acompañada de los controles que permiten realizar el flujo de datos.

```

1 TableView {
2     id: tabla
3     visible: true
4     anchors.centerIn: parent
5     anchors.horizontalCenterOffset: 140
6     anchors.verticalCenterOffset: 150

```

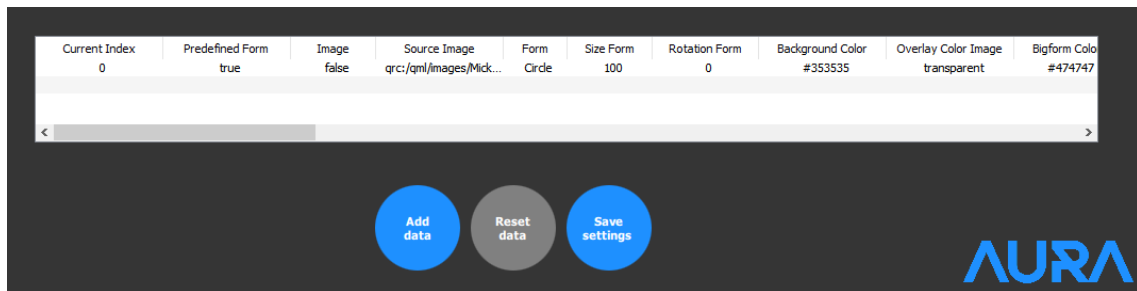


Figura 45: Tabla para visualización de parámetros configurados y controles para , restablecer y exportar datos

```

7   width: 1000
8   height: 100
9   model: datos
10  TableViewColumn {
11      role: "CurrentIndex"
12      title: "Current Index"
13      width: 120
14      horizontalAlignment: Text.AlignHCenter
15  }
16  TableViewColumn {
17      role: "PredefinedForm"
18      title: "Predefined Form"
19      width: 120
20      horizontalAlignment: Text.AlignHCenter
21  }
22 }
```

Código 15: Configuración de la visualización de parametros en la tabla

Configuración de pruebas

■ Integración total del test

Para esta etapa se actualizó el algoritmo que ingresa los datos a la tabla de visualización. Inicialmente el funcionamiento original de cada test no se veía comprometido porque no existía combinación de pruebas, por lo tanto, los elementos de una prueba siempre pertenecían a una clase de test. Ahora, gracias a este requerimiento, los elementos de una prueba pueden ser de varios test, en consecuencia el método para ingresar información a la prueba debe actualizarse. El ajuste más relevante tiene que ver con garantizar las tres etapas de los test de movimientos sacádicos. Como se expuso en la agrupación inicial (Sprint 3), la definición de un estímulo genera automáticamente su predecesor, en otras palabras, se generan 2 elementos de manera simultanea. La Figura 41, muestra este procedimiento con test de movimientos sacádicos, luego, la Figura 46, describe como se realiza el complemento de estos elementos, cuando se realizan pruebas con test de diferentes tipos. En la tabla superior se indican los elementos de un test de sacadas, en esta se puede observar que únicamente cuenta con dos elementos. En la siguiente tabla

(inferior), se agrega un elemento de un test de persecución o seguimiento lento, cuando este se ingresa a la prueba automáticamente agrega un elemento que complementa al test de sacadas, indicado en color azul. Así pues se garantiza que la sección de movimientos sacádicos cuente con el número correcto de elementos para la presentación de la prueba.

Test Configuration									
Nº	Position X (Deg)	Position Y (Deg)	Initial Time (ms)	Visible time (ms)	Test Duration (seg)	Gap (ms)	Overlap (ms)	Countermanding (ms)	Type
1	0	0	0	2000	2.0	0	0	0	Central fixation
2	7	6	2000	2000	4.0	0	0	0	Pro-saccade
Test Configuration									
Nº	Position X (Deg)	Position Y (Deg)	Initial Time (ms)	Visible time (ms)	Test Duration (seg)	Gap (ms)	Overlap (ms)	Countermanding (ms)	Type
1	0	0	0	2000	2.0	0	0	0	Central fixation
2	7	6	2000	2000	4.0	0	0	0	Pro-saccade
3	0	0	4000	2000	6.0	0	0	0	Central fixation
4	0	0	6000	2000	8.0	0	0	0	Smooth Pursuit

Figura 46: Elementos complementarios para test de movimientos sacádicos en pruebas combinadas

Para lograr esta integración final y garantizar la condición anterior, se actualizó el código para agregar datos con una función que ingresa un elemento de tipo fijación central, cada vez que haya un cambio entre movimientos sacádicos y cualquier otro. La ejecución de la función está controlada por una variable que se activa cuando se cumplen ciertas condiciones. La función se ejecuta si el último elemento de la tabla de datos es de tipo “*stml*”, esta información se encuentra oculta en la tabla de visualización. En otras palabras hace parte de cada elemento en la tabla, pero no se muestra al especialista médico. Las instrucciones en la función, copian los datos anteriores al último elemento de la tabla y lo agregan a la posición actual de la misma. Posteriormente se agrega la información del nuevo elemento que el usuario ha configurado en ese momento, en el caso de la Figura 46, un elemento de un test de persecución lenta. Las instrucciones que agregan los datos son iguales al código 3, indicado en el desarrollo del Sprint 1, con la diferencia que el tiempo de inicio se calcula de otra manera. En el código `finSac`, la línea 3 permite apreciar que la variable `time_opc` toma el valor de tiempo anterior al estímulo objetivo, en el caso de la Figura 46, el elemento de la posición 0. En la línea 4, la variable `drtn`, es la duración actual de la prueba, por lo tanto, a esta se suma el tiempo de duración correspondiente al nuevo elemento y se realizan los ajustes necesarios. Los elementos (...) en la función `append` son similares a los observados en el código 3, por esta razón no se indican aquí. Finalmente, con la línea 10, se actualiza el tiempo de duración de la prueba, adicionando el tiempo de duración del nuevo elemento.

```

1 function finSac() {
2     var dur_opc = 0
3     var time_opc = libraryModel.get(libraryModel.count-2).time

```

```

4     drtnf = drtn*1+time_opc*1
5     dur_opc = drtnf
6     drtnf = (drtnf*1/1000*1) .toFixed(1)
7     libraryModel.append({
8         ...
9     })
10    drtn=drtn*1+time_opc*1
11 }

```

Código 16: Definición de estímulos para ventana de previsualización

■ Aleatoriedad de pruebas

Esta característica simplemente permite definir un orden aleatorio para la presentación de la prueba realizada en la plataforma *Oscann desk100*. Esta información se consigna en el archivo XML de la prueba y así la plataforma puede definir el orden de presentación de los elementos de la prueba. La opción *Reference* permite al especialista médico establecer un orden estricto de presentación para algunos o elementos de la prueba. El procedimiento es simple, este *widget* muestra una lista con dos elementos; *Null* y el valor de la última posición correspondiente a un elemento de la tabla de datos, es decir una opción nula y otra válida para el último elemento agregado a la prueba. Cuando el especialista médico se encuentre diseñando una prueba podrá establecer, si así lo desea, una relación de posición entre el elemento que ingresará y el último elemento de la prueba. De este modo, si la prueba se define como aleatoria, la plataforma agrupará en el orden establecido los elementos que contengan un parámetro de relación y los restantes los presentará de manera aleatoria. Esta información se agrega de igual manera que los parámetros restantes como tiempo y posición, la función *append* vista anteriormente toma el texto actual del *ComboBox Reference*, y lo agrega a la tabla de datos. Esta relación se establece de manera automática cuando se diseñan pruebas que combinan varios tipos de test de movimientos sacádicos para garantizar las 3 etapas normales del movimiento. En este sentido, cada estímulo ingresado a la prueba se relaciona con el anterior para mantener el orden en que se ingresan, así, cuando se realice la presentación se mantiene un orden correcto para movimientos sacádicos. En la Figura 47 la casilla delimitada de color rojo indica la relación que se establece automáticamente para los estímulos de test de movimientos sacádicos, mientras que la casilla amarilla muestra la relación que se asigna cuando se ingresa un elemento de diferente test al de sacadas, en este caso un estímulo de persecución lenta.

Test Configuration										
Nº	Position X (Deg)	Position Y (Deg)	Initial Time (ms)	Visible time (ms)	Test Duration (seg)	Gap (ms)	Overlap (ms)	Countermanding (ms)	Reference	Type
1	0	0	0	2000	2.0	0	0	0	Null	Central fixation
2	7	6	2000	2000	4.0	0	0	0	1	Pro-saccade
3	0	0	4000	2000	6.0	0	0	0	2	Central fixation
4	0	0	6000	2000	8.0	0	0	0	Null	Smooth Pursuit

Figura 47: Ejemplo de relación entre elementos de una prueba

Desarrollo del *Sprint 5*

Configuración del estímulo

■ Configuración de la exportación de parámetros en archivos XML:

Una vez se han añadido los datos al modelo de lista descrito anteriormente, y estos elementos se encuentran visibles para el especialista médico, se puede realizar la exportación de estas configuraciones utilizando un archivo XML, el cual proporciona organización de los parámetros asignándoles una etiqueta correspondiente. Para este requerimiento se asigna el comando “SCD” desde el archivo fuente (main.cpp), donde con el método *save* se guardan los datos en un fichero de extensión XML.

En la Figura 48, se indica un cuadro de dialogo alertando que no existen elementos añadidos en la tabla, por lo tanto, no sería posible exportar ningún dato. Luego en la Figura 49, se realiza la petición de introducir un nombre para el archivo que almacenará los parámetros configurados en esta parte de la interfaz.

Cabe destacar que desde la interfaz de configuración, se asignan etiquetas a los archivos de exportación donde se han definido tres variantes, que corresponden a:

SP: Esta etiqueta se asigna cuando se activan las trayectorias en la configuración del estímulo, destinado por tanto a la integración de las pruebas de seguimiento lento.

STML: Se asigna esta marca cuando la visualización del estímulo se encuentra activada, y el parpadeo del fondo junto con las trayectorias desactivadas, por tanto, se orienta hacia pruebas de comportamientos sacádicos por medio de la variación de las posiciones.

PL: Se fija esta etiqueta cuando se activa la animación de parpadeo en el fondo de las pruebas, donde el estímulo es destinado a pruebas de reacción pupilar.

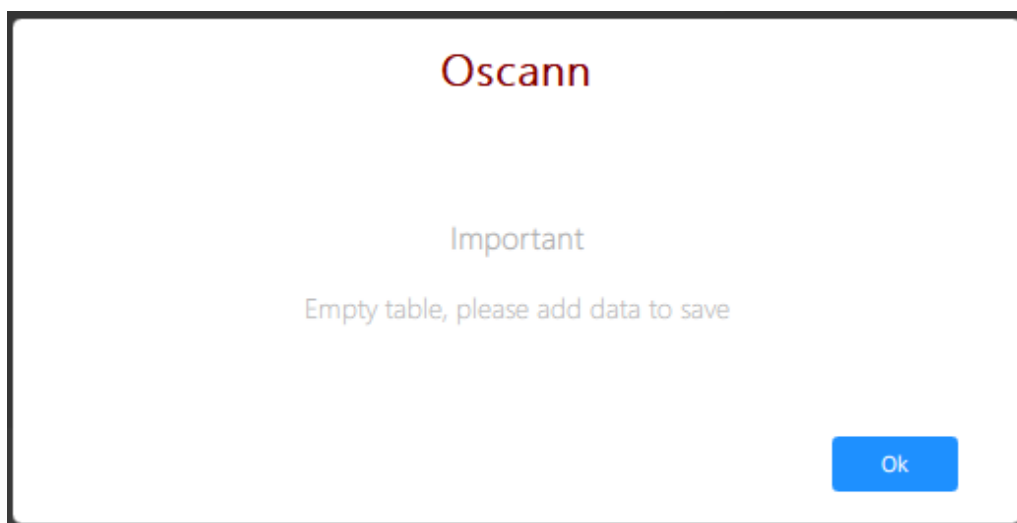


Figura 48: Cuadro de diálogo indicando que la tabla se encuentra vacía

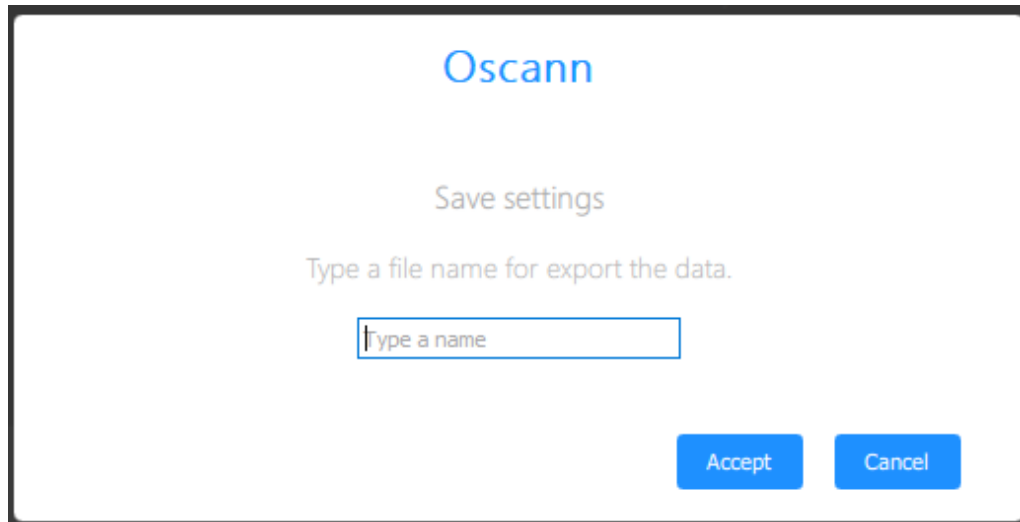


Figura 49: Cuadro de diálogo para asignar nombre al fichero que se dispone a contener la exportación de los parámetros configurados

Configuración de pruebas

Durante la etapa final se desarrolló el sistema para selección de estímulos y su respectiva visualización. Con la construcción del archivo XML para la creación de estímulos, indicada en el anterior apartado, se procede a diseñar el método para habilitar la selección de estos en la interfaz de configuración de pruebas. Para lograr esta comunicación se diseñó el elemento *stml.qml*, el cual contiene toda la información para crear cualquier elemento diseñado en la sección de diseño de estímulos. De esta forma, los archivos XML provenientes de esta sección contienen los datos que definen el estímulo y es posible recrearlos en la sección de configuración de pruebas. Esta implementación funciona de forma similar a los cuadros de dialogo descritos en la etapa anterior, donde cada estímulo se utiliza una vez y se destruye cuando cumple su función. Los *widgets Stimulus* y *Stimulus form*, contienen los estímulos específicos para cada test, de esta manera, no se presentarán problemas debido a la elección de un estímulo incorrecto para un tipo de test. Esta clasificación se logra gracias a la integración con C++, que, como se mencionó anteriormente, permite desarrollar tareas más avanzadas, como, por ejemplo, definir una ruta en el sistema de archivos locales y ubicar los elementos en esta y así usarlos en el archivo QML principal del proyecto. En los archivos *tsv.h* y *tsv.cpp* se definen elementos de la clase `QStringList`, las cuales se agregan a los *widgets* correspondientes como un modelo de lista normal. Estas listas se actualizan cada vez que se agrega un nuevo elemento a la ruta de archivos definida, por lo tanto, los estímulos se invocan automáticamente en la sección de configuración de la prueba a través de los *ComboBox* denominados *Stimulus* y *Stimulus form*, según sea el test. Es importante destacar las funciones empleadas para este procedimiento las cuales se indican en el Código 17.

```

1  QStringList tsv::stml_List() {
2      QDir().mkpath(QStandardPaths::standardLocations(
3          QStandardPaths::DataLocation)[0]);

```

```

        QStandardPaths::DataLocation)[0], QStringList() << "stml
        *.xml", QDir::Files);
4   QStringList list;
5   QString name;
6   while(mainFolder.hasNext()){
7       name = mainFolder.next();
8       name = name.split("stml")[1];
9       name = name.split(".")[0];
10      list.push_back(name);
11  }
12  return list;
13  }

```

Código 17: Método para actualización de lista de estímulos

Este método para el elemento `stml_List()`, permite actualizar la lista de estímulos correspondientes al test de movimientos sacádicos. La clase `DDir`, permite acceder al directorio de archivos que se indica en el argumento, en concreto `QStandardPaths` proporciona el método de acceso a través de `standardLocations`, el cual retorna la ruta de acceso de la constante asignada, en este caso `DataLocation`. Esta variable corresponde a la carpeta local de archivos del sistema operativo. La primera vez que el proyecto se ejecuta se crea una carpeta en esta ruta. En un SO Windows, la carpeta se crea en la siguiente ruta, `C:/Users/AppData/Local/proyecto`. En la línea 3, la clase `QDirIterator` proporciona un iterador para navegar por los archivos del directorio especificado. Para esta lista se especifica que busque los archivos empezados con la línea *stml* y con la extensión *.xml*. De este modo, con la sentencia `while` y las instrucciones en su interior, se extraen los nombres de los estímulos que cumplan estas condiciones para retornarlos a la lista de cada *widget*.

■ Actualización de previsualización

Ahora bien, una vez implementado el sistema para seleccionar los estímulos, se procede con el desarrollo final de la previsualización de pruebas. Recordemos que inicialmente este requerimiento permitía visualizar únicamente test de movimientos sacádicos y contaba con tan solo una clase de estímulo. Como se mencionó anteriormente, tanto la definición como la creación de estos estímulos se realiza de forma similar a los cuadros de diálogo, tratados en el anterior apartado. Ahora, el código necesario para la definición es demasiado extenso para incluirlo en esta memoria, sin embargo, se incluyen las siguientes líneas de código que muestran la función definida para crear los estímulos.

```

1  property var componente : Qt.createComponent("stml.qml")
2  function createStml(posx, posy, time){
3      if(componente.status === Component.Ready){
4          var objeto = componente.createObject(root, {
5              //Aqui se obtiene la informacion del estimulo
6              //desde el archivo xml
7          })
8      }
9      objeto.destroy(time)

```


10 | }

Código 18: Función para la creación de estímulos en ventana de vista previa

En la línea 1, se define la variable que permite crear un objeto a partir del archivo `stim1.qml`. En la función definida como `createStiml`, se carga toda la información proveniente del archivo xml y se asignan a los distintos parámetros del objeto `componente` para crear el estímulo correspondiente. En la línea 2, se puede observar que la función acepta 3 argumentos, dos de posición y uno de tiempo. En este caso el tiempo que se asigna también se utiliza para destruir el estímulo, así, cuando el objeto ya ha cumplido con su tarea en el tiempo señalado, se destruye para continuar con los siguientes estímulos. En la línea 4, el parámetro `root`, define el componente donde se dibujan los objetos, es decir la ventana de visualización. Ahora, los archivos XML se cargan en una lista de la clase `XmlListModel` que permite extraer información del archivo y asignarlo a la lista de parámetros del objeto.

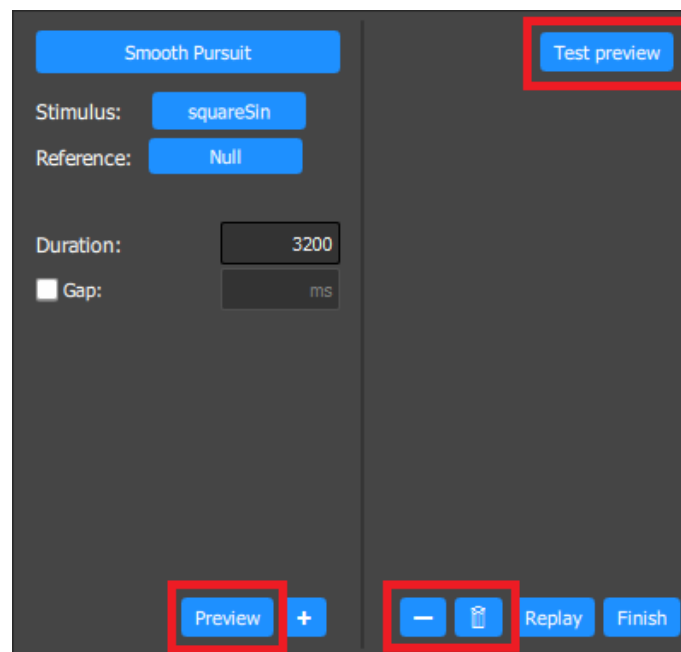


Figura 50: Opciones de previsualización

Por otra parte, el algoritmo para realizar la previsualización con este tipo de objetos, sigue la lógica explicada en la implementación inicial de este requerimiento (Sprint 3), a través de la Figura 42. La diferencia con esta actualización está en la manera que se suprimen los estímulos. Inicialmente se manipulaba la visibilidad del objeto, es decir el objeto sigue existiendo, simplemente se oculta del observador. Ahora, gracias a la instancia `destroy()` en la función para crear los estímulos, el objeto utilizado se elimina una vez termina su presentación. Cuando existen muchos objetos creados dinámicamente, resulta beneficioso para el rendimiento eliminar los objetos no utilizados. En cuanto a la presentación de una prueba completa o bien tomando todos los elementos de la tabla de datos de esta, a diferencia de la implementación inicial, los contadores se reinician cada vez que hay un cambio de estímulo y de esta manera se recorren todas las posiciones de

la tabla, realizando una presentación completa de la prueba. En la Figura 50, se pueden observar las dos opciones para realizar una vista previa, mientras el botón de la parte inferior denominado *Preview* realiza la previsualización del elemento que se encuentra en la sección de parámetros (en el caso de la Figura, un estímulo de tipo persecución lenta), el botón *Test preview*, permite visualizar el test completo, es decir, muestra los elementos que se encuentran en la tabla de datos de la prueba. Además, en la Figura 50, se puede observar los botones que permiten eliminar elementos de la prueba. El botón que contiene el signo menos (-), elimina la última fila de la tabla de datos, es decir, el último elemento ingresado. Las instrucciones que contiene el botón permiten actualizar los valores como la duración actual de la prueba, el apartado de referencias ("*Reference*"), entre otros. Mientras que el botón con el símbolo de la caneca de basura, elimina la totalidad de la prueba, es decir, limpia los elementos de la tabla de datos. Cuando se realiza esta acción, se reinician todas las variables del sistema, dispone el tiempo en 0 y se eliminan los elementos de la lista "*Reference*". Finalmente, con la implementación completa de estos requerimientos, se cumple con el objetivo principal de esta etapa; establecer el sistema de comunicación entre las dos interfaces del trabajo.