



Centro de Ciencias Básicas

Dpto. de Ciencias de la Computación

Tesis

**Modelo para la Aplicación del Análisis del Comportamiento de Mirada
Mediante Eyetracking de la Interacción Usuario- Sistema para Mejorar la
Calidad de la UX y Mejorar los Procesos de Desarrollo de Software.**

Presenta

Angel Eduardo Villegas Ortíz

Para obtener el grado de: Maestría en Ciencias de la Computación,
Rama de Investigación: Ingeniería de Software.

Comité Tutorial:

Tutor: Dr. Francisco Javier Álvarez Rodríguez

Co-tutor: MCC. Eduardo Emmanuel Rodríguez López

Asesor: Dr. Julio César Ponce Gallegos

Aguascalientes, Ags, México.

25 de julio de 2024.

M. EN C. JORGE MARTÍN ALFÉREZ CHÁVEZ
DECANO DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE

Por medio del presente como **Miembros del Comité Tutorial** designado del estudiante **ANGEL EDUARDO VILLEGAS ORTIZ** con ID **235190** quien realizó la tesis titulada: **MODELO PARA LA APLICACIÓN DEL ANÁLISIS DEL COMPORTAMIENTO DE MIRADA MEDIANTE EYETRACKING DE LA INTERACCIÓN USUARIO-SISTEMA PARA MEJORAR LA CALIDAD DE LA UX Y MEJORAR LOS PROCESOS DE DESARROLLO DE SOFTWARE**, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia damos nuestro consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que nos permitimos emitir el **VOTO APROBATORIO**, para que el pueda proceder a imprimirla así como continuar con el procedimiento administrativo para la obtención del grado.

Ponemos lo anterior a su digna consideración y sin otro particular por el momento, le enviamos un cordial saludo.

ATENTAMENTE

"Se Lumen Proferre"

Aguascalientes, Ags., a 27 de mayo de 2024.



Dr. Francisco Javier Álvarez Rodríguez
Tutor de tesis



MCC. Elinora Guzmán Rodríguez López
Co-Tutor de tesis



Dr. Yulín César Ponce Gallegos
Asesor de tesis

c.c.p.- Interesado

c.c.p.- Secretaría Técnica del Programa de Posgrado

Elaborado por: Depto. Apoyo al Posgrado.
Revisado por: Depto. Control Escolar/Depto. Gestión de Calidad.
Aprobado por: Depto. Control Escolar/ Depto. Apoyo al Posgrado.

Código: DO-SEE-FO-16
Actualización: 00
Emisión: 17/05/19



DICTAMEN DE LIBERACIÓN ACADÉMICA PARA INICIAR LOS TRÁMITES DEL EXAMEN DE GRADO



Fecha de dictaminación dd/mm/aaaa: 22/07/2024

NOMBRE: Angel Eduardo Villegas Ortiz **ID** 235190

PROGRAMA: Maestría en Ciencias con opciones a la Computación, Matemáticas Aplicadas. **LGAC (del posgrado):** Computación -Ingeniería de software

TIPO DE TRABAJO: () Tesis () Trabajo Práctico

TÍTULO: Modelo para la Aplicación del Análisis del Comportamiento de Mirada Mediante Eyetracking de la Interacción Usuario- Sistema para Mejorar la Calidad de la UX y Mejorar los Procesos de Desarrollo de Software.

IMPACTO SOCIAL (señalar el impacto logrado):

Una de las principales características del trabajo de investigación es la implementación de una tecnología innovadora como lo es el eye-tracking en la ingeniería de software. Conforme más pruebas se realizaron fue posible proponer conceptos, métodos y modelos que proponen un paradigma muy poco usado en el campo de la ingeniería de software. El desarrollo de este proyecto busca generar nuevos enfoques para el diseño de interfaces para perfiles de personas poco consideradas en la mayoría de procesos de desarrollo de software como lo son personas de la tercera edad, débiles visuales, daltónicos, etc.

INDICAR	SI	NO	N.A.	(NO APLICA)	SEGÚN CORRESPONDA:
<i>Elementos para la revisión académica del trabajo de tesis o trabajo práctico:</i>					
SI					El trabajo es congruente con las LGAC del programa de posgrado
SI					La problemática fue abordada desde un enfoque multidisciplinario
SI					Existe coherencia, continuidad y orden lógico del tema central con cada apartado
SI					Los resultados del trabajo dan respuesta a las preguntas de investigación o a la problemática que aborda
SI					Los resultados presentados en el trabajo son de gran relevancia científica, tecnológica o profesional según el área
SI					El trabajo demuestra más de una aportación original al conocimiento de su área
SI					Las aportaciones responden a los problemas prioritarios del país
SI					Generó transferencia del conocimiento o tecnológica
SI					Cumple con la ética para la investigación (reporte de la herramienta antiplagio)
<i>El egresado cumple con lo siguiente:</i>					
SI					Cumple con lo señalado por el Reglamento General de Docencia
SI					Cumple con los requisitos señalados en el plan de estudios (créditos curriculares, optativos, actividades complementarias, estancia, predoctoral, etc)
SI					Cuenta con los votos aprobatorios del comité tutorial, en caso de los posgrados profesionales si tiene solo tutor podrá liberar solo el tutor
N.A					Cuenta con la carta de satisfacción del Usuario
SI					Coincide con el título y objetivo registrado
SI					Tiene congruencia con cuerpos académicos
SI					Tiene el CVU del Conacyt actualizado
N.A					Tiene el artículo aceptado o publicado y cumple con los requisitos institucionales (en caso que proceda)
<i>En caso de Tesis por artículos científicos publicados</i>					
N.A					Aceptación o Publicación de los artículos según el nivel del programa
N.A					El estudiante es el primer autor
N.A					El autor de correspondencia es el Tutor del Núcleo Académico Básico
N.A					En los artículos se ven reflejados los objetivos de la tesis, ya que son producto de este trabajo de investigación.
N.A					Los artículos integran los capítulos de la tesis y se presentan en el idioma en que fueron publicados
N.A					La aceptación o publicación de los artículos en revistas indexadas de alto impacto

Con base a estos criterios, se autoriza se continúen con los trámites de titulación y programación del examen de grado: Sí No

Elaboró:

* NOMBRE Y FIRMA DEL CONSEJERO SEGÚN LA LGAC DE ADSCRIPCIÓN: Dr. Francisco Javier Álvarez Rodríguez

NOMBRE Y FIRMA DEL SECRETARIO TÉCNICO: Dr. Heriberto Sánchez Cruz

* En caso de conflicto de intereses, firmará un revisor miembro del IAB de la LGAC correspondiente distinto al tutor o miembro del comité tutorial, asignado por el Decano

Revisó:

NOMBRE Y FIRMA DEL SECRETARIO DE INVESTIGACIÓN Y POSGRADO: Dr. Alejandro Padilla Díaz

Autorizó:

NOMBRE Y FIRMA DEL DECANO: M. en. C. Jorge Martín Alférez Chávez

Nota: procede el trámite para el Depto. de Apoyo al Posgrado
 En cumplimiento con el Art. 105C del Reglamento General de Docencia que a la letra señala entre las funciones del Consejo Académico: ... Cuidar la eficiencia terminal del programa de posgrado y el Art. 105F las funciones del Secretario Técnico, llevar el seguimiento de los alumnos.

Elaborado por: D. Apoyo al Postg.
 Revisado por: D. Control Escolar/D. Gestión de Calidad.
 Aprobado por: D. Control Escolar/ D. Apoyo al Postg.

Código: DO-SEE-FO-15
 Actualización: 01
 Emisión: 28/04/20

Agradecimientos

Primeramente, quiero agradecer a la Benemérita Universidad Autónoma de Aguascalientes por permitirme desde el año 2017 pertenecer a una institución en la que además de adquirir conocimientos, conocí la calidad humana de cada uno de sus integrantes y que me formaron para ser una persona que pueda representar los valores de la institución dentro y fuera de ella. También extendiendo un reconocimiento a CONAHCYT, institución que mediante su apoyo a la maestría de Ciencias de la Computación me permitió utilizar mejores herramientas y obtener mejores resultados.

Dentro de la institución agradezco en particular al Dpto. de Ciencias de la Computación, el cuál entre docentes, asesores, administrativos e incluso estudiantes siempre han ofrecido su ayuda y orientación para resolver cualquier duda o llevar a cabo los procedimientos necesarios para cumplir en tiempo y forma con los procesos del plan de estudios de posgrado. También reconozco al Dpto. de Apoyo a Posgrados ya que me ayudaron a cubrir gastos de proyectos de productividad científica.

También deseo extender un muy especial reconocimiento a mis tutores Eduardo Rodríguez López y Francisco Álvarez Rodríguez los cuáles en todo momento me ayudaron a forjarme un camino dentro de la investigación y la docencia, y me permitieron desarrollar trabajos de investigación de una gran calidad.

De manera muy especial le doy gracias a mis padres Esmeralda Ortiz Monreal y Renato Villegas Arroyo por cada uno de los sacrificios que han hecho durante tantos años que me han permitido realizar cada proyecto de mi vida y me han hecho la persona que soy.

También quisiera agradecer a mi amiga, confidente y motivación Dalia, quién desde hace muchos años ha estado a un lado mío dándome el regalo de su compañía.

Finalmente quisiera agradecer y felicitar por su trabajo a los estudiantes de 9no semestre del ciclo escolar Ago-Dic del 2023 de la carrera de ICI que participaron en el desarrollo de uno de los casos de estudio que se abordan en este trabajo: Abraham, Adrián, Carolina, Jorge y Luis.

Dedicatoria

Dedico este trabajo a la persona que día con día representa mi más grande deseo de ser una inspiración y un ejemplo a seguir, a la única persona por la que dejaría atrás todo por verlo feliz y cumpliendo cada una de sus metas sean cual sean estas, dedico este trabajo a Jorge Renato, mi hermano y, sin duda, la persona que más quiero en este mundo.



Índice

Resumen.....	11
Abstract	12
Introducción	13
1. Planteamiento del problema	15
1.1 Antecedentes	15
1.2 Problemática a resolver	19
1.3 Justificación	20
1.4 Preguntas de Investigación	22
1.4.1 Pregunta general	22
1.4.2 Preguntas específicas	22
1.5 Objetivos de la Investigación.....	22
1.5.1 Objetivo general.....	23
1.5.2 Objetivos particulares	23
2. Marco Teórico	24
2.1 Interacción Humano-Computadora (HCI)	24
2.2 Ingeniería de Software	26
2.2.1 Planeación y diseño de Software	27
2.2.2 Marcos de Trabajo (Frameworks)	27
2.2.3 Metodologías de Desarrollo.....	28
A) Dos enfoques diferentes: Metodologías Ágil vs. Tradicional.....	29
B) Metodología incremental y en espiral	30
2.2.4 Diseño de Interfaces.....	34
A) Elementos de una interfaz	36
B) Patrones de diseño de interfaces	37
2.2.5 Testing de software.....	39
A) Pruebas Unitarias y de Integración.....	39
2.3 Eye-tracking.....	40
2.3.1 Visión humana.	42
2.3.2 Proceso de atención visual.....	44
2.3.3 Intervención de métodos cognitivos en el proceso de atención visual	46

2.3.4 Hardware y Software de eye-tracking en la actualidad	48
A) Eye-tracker en el mercado y características relevantes	48
B) Hardware y software de eye-tracking usado en la investigación.	49
3. Metodología de la Investigación	51
3.1 Metodología de experimentación.....	52
3.2 Metodología de desarrollo de software tradicional propuesta.....	54
3.3 Metodología de desarrollo de software ágil propuesta.....	56
3.4 Metodología de diseño de interfaces: Interfaces paralelas.....	58
3.5 Metodología de selección e interpretación de métricas de <i>eye-tracking</i>	61
3.5.1 Fundamentación de las métricas de eye-tracking seleccionadas	62
3.5.2 Formalización de los elementos del análisis de uso de interfaces mediante eye-tracking propuesto.....	63
A) Interfaz y elementos	64
B) Métricas de eye-tracking	66
C) Interpretaciones aplicables a las métricas de eye-tracking.....	68
3.6 Índices de Uso de Interfaces mediante Eye-Tracking (INUSINET).....	72
3.6.1 Índice de Complejidad de Entendimiento (ICE)	73
A. Valor Cuantitativo del ICE	74
B. Ecuación del ICE	74
3.6.2 Índice de Complejidad de Recorrido (ICR)	74
A. Valor cuantitativo de ICR.....	75
B. Ecuación del ICR.....	75
3.6.3 Índice de Complejidad de Uso (ICU).....	75
A. Valor cuantitativo del ICU	76
B. Ecuación del ICU.....	76
3.6.4 Implementación de INUSINET	77
3.7 Marco de trabajo para la Evaluación del Uso de Interfaces mediante Eye-Tracking (EVUSINET)	81
3.7.1 Análisis de Uso Particular y General (Análisis de uso de bajo y alto nivel)	83
3.7.2 EVUSINET: Pruebas de Uso.....	85
A) Diseño Experimental.....	86
B) Pruebas con Usuarios.....	88
3.7.3 EVUSINET: Uso de Eye-tracking.....	89
A) Preparación del análisis: Eventos	91

B) Preparación del análisis: Áreas de Interés: Áreas de Interés.....	93
C) Exportación de datos	97
3.7.4 EVUSINET: Evaluación	98
A) Análisis de Patrones de Uso Generales: Visualizaciones	101
B. Análisis de Patrones de Uso Particulares	110
4. Desarrollo	117
4.1. Caso de estudio 1: Componente Login.....	118
4.1.1 Características del sistema	119
4.1.2 Proceso de desarrollo de software del componente login	120
A. Objetivos	120
B. Requerimientos.....	121
C. Diagramas.....	122
4.1.3 Propiedades de las interfaces	127
A. Propuesta de Interfaz Dinámica.....	128
B. Propuesta de Interfaz de Formulario Convencional	130
4.1.4 Implementación de técnicas y conceptos de eye-tracking	132
A. Diseño experimental del componente login mediante EVUSINET.	133
B. Implementación de eye-tracking en el análisis del componente login mediante EVUSINET.....	134
C. Evaluación mediante eye-tracking del componente login.....	140
4.2 Caso de estudio 2: Easy-Bank Vision	149
4.2.1 Características del sistema	149
4.2.2 Proceso de desarrollo de software.	152
A. Objetivos	152
B. Requerimientos.....	153
C. Tipos de usuarios.....	153
4.2.3 Propiedades de las interfaces	154
4.2.4 Diseño de la secuencia de interacción	160
4.2.5 Implementación de técnicas y conceptos de eye-tracking	163
A. Diseño experimental de Easy-Bank Vision mediante EVUSINET.....	164
B. Implementación de eye-tracking en el análisis de Easy-Bank Vision mediante EVUSINET	166
C. Evaluación mediante eye-tracking de Easy-Bank Vision	171
5. Resultados y Discusiones.....	178

5.1 Resultados de Caso de Estudio 1: Componente Login	179
5.1.1 Resultados de patrones de uso generales	179
5.1.2 Resultados de patrones de uso particulares.	183
A. Evaluación de Interfaz dinámica	184
B. Evaluación de interfaz de formulario convencional.....	186
5.1.3 Análisis de resultados del componente login.....	190
5.2 Resultado de Caso de Estudio 2: Easy-Bank Vision	194
5.2.1 Resultados de Patrones de Uso Generales	194
A. Login Detallado	195
B. Login Simple	196
C. Home	197
D. Saldos y Movimientos	197
E. Pago Tarjeta.....	198
F. Transferencias	199
5.2.2 Resultados de patrones de uso particulares	200
A. Login Detallado	201
B. Login Simple	203
C. Home	205
D. Saldos y Movimientos	207
E. Pagar Tarjeta	209
F. Transferencias	210
5.2.3 Análisis de resultados por interfaz de Easy-Bank Vision	212
A. Login Detallada y Simple	213
B. Home	214
C. Saldos y Movimientos	215
D. Pagar Tarjeta	216
E. Transferencias	216
5.2.4 Análisis de la experiencia general de Easy-Bank Vision	217
5.3 Discusiones.....	218
Conclusiones y Trabajo Futuro.....	220
Productividad Científica	222
Artículos científicos	222
Referencias.....	225

Índice de Tablas

Tabla 1: Unidades de medida de variables de ET, (Elaboración propio, 2024).....	67
Tabla 2: Tabla de métricas de eye-tracking y sus reglas de interpretación, (Elaboración propia, 2024).	71
Tabla 3: Expectativas de uso del caso de uso hipotético, (Elaboración propia, 2024).	78
Tabla 4: Tabla de resultados de cada prueba.	79
Tabla 5: Tabla de resultados de promedios de elementos.	80
Tabla 6: Tabla de resultados mediante índices.....	80
Tabla 7: Tabla de expectativas de uso para la prueba simulada de evaluación de eye-tracking, (Elaboración propia, 2024).....	111
Tabla 8: Recopilación resultados para cada prueba realizada mediante eye-tracking de la interfaz de estudio, (Elaboración propia, 2024).....	114
Tabla 9: Resultados por elementos de la interfaz de estudio, (Elaboración propia, 2024).	114
Tabla 10: Valores cuantitativos de pruebas de eye-tracking usando INUSINET, , (Elaboración propia, 2024).	115
Tabla 11: Objetivos del componente login, (Elaboración propia, 2024).....	120
Tabla 12: Requerimientos funcionales del componente login, (Elaboración propia, 2024).....	121
Tabla 13: Requerimientos no funcionales del componente Login, (Elaboración propia, 2024).	122
Tabla 14: Asociación de íconos y colores, (Elaboración propia, 2023).	128
Tabla 15: Tabla de estados de etiqueta de control de errores, (Elaboración propia, 2023)	130
Tabla 16: Roles y actividades del equipo del caso de estudio, (Elaboración propia, 2024)-	150
Tabla 17: Objetivos del caso de estudio Easy-Bank Vision, (Elaboración conjunta con el equipo de desarrollo, 2023).	152
Tabla 18: Tabla de requerimientos funcionales de Easy-Bank Vision, (Elaboración conjunta con equipo de desarrollo, 2023).	153
Tabla 19: Tabla de requerimientos no funcionales de Easy-Bank Vision, (Elaboración conjunta con equipo de desarrollo, 2023).	153
Tabla 20: Resultados de métricas de eye-tracking de Interfaz Dinámica, (Elaboración propia, 2024).	185
Tabla 21: Resultados de la aplicación de INUSINET en la interfaz dinámica, (Elaboración propia, 2024).	185
Tabla 22: Resultados de métricas de eye-tracking de Interfaz Convencional, (Elaboración propia, 2024).	188
Tabla 23: Resultados de la aplicación de INUSINET en la interfaz de formulario convencional.	189
Tabla 24: Resultados de la aplicación de INUSINET en la interfaz dinámica remarcando elementos que el evaluador detectó como puntos de mejora.....	192
Tabla 25: Resultados de la aplicación de INUSINET en la interfaz convencional remarcando elementos que el evaluador detectó como puntos de mejora.....	193
Tabla 26: Resultados de Interfaz Login Detallado de Easy-Bank Vision, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).....	202

Tabla 27: Resultados de INUSINET para la interfaz de Login Detallado, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023)..... 202

Tabla 28: Resultados de Interfaz Login Simple de Easy-Bank Vision, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023). 204

Tabla 29: Resultados de INUSINET para la interfaz de Login Simple, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023). 204

Tabla 30: Resultados de Interfaz Home de Easy-Bank Vision,, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023)..... 206

Tabla 31: Resultados de INUSINET para la interfaz Home, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023)..... 206

Tabla 32: Resultados de Interfaz Saldos y Movimientos de Easy-Bank Vision, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023). 208

Tabla 33: Resultados de INUSINET para la interfaz de Saltos y Movimientos,, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023). 208

Tabla 34: Resultados de Interfaz Pagar Tarjeta de Easy-Bank Vision, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023). 210

Tabla 35: Resultados de INUSINET para la interfaz Pagar Tarjeta,, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023)..... 210

Tabla 36: Resultados de Interfaz Transferencias de Easy-Bank Vision, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023). 211

Tabla 37: Resultados de INUSINET para la interfaz de Transferencias, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023). 212

Tabla 38: Resultados de la aplicación de INUSINET en la interfaz Home remarcando elementos que peores y mejores resultados obtuvieron, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023). 215

Tabla 39: Resultados de la aplicación de INUSINET en la interfaz saldos y Movimientos remarcando el índice con peores resultados generale, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023). 215

Tabla 40: Resultados de la aplicación de INUSINET en la interfaz Pago Tarjeta remarcando elementos que peores resultados obtuvieron, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023). 216

Tabla 41: Resultados de la aplicación de INUSINET en la interfaz Transferencias remarcando elementos que peores resultados obtuvieron, , (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023)..... 217

Índice de Figuras

Figura 1: Diagrama de Metodología Incremental, (Zumba Gamboa & León Arreaga, 2018).....	31
Figura 2: Diagrama de metodología de desarrollo de software en espiral, (Elaboración propia, 2024).	33
Figura 3: Diagrama de métodos de entrada y salida de una computadora, (Elaboración propia, 2024).	34
Figura 4: Diagrama propuesto para el proceso de interacción dentro de un entorno de uso, (Elaboración propia, 2024).....	36
Figura 5: Ojo humano, (Duchowski, 2017).....	44
Figura 6: Eye-tracker TobiiProNano, (Sitio Oficial de TobiiTechnologys).....	49
Figura 7: Diagrama de desarrollo de software y evaluación mediante eye-tracking, (Elaboración propia, 2023).	54
Figura 8: Metodología de desarrollo en espiral-iterativa, (Elaboración propia, 2024.).....	56
Figura 9: Diagrama de la metodología ágil de desarrollo de software, (Elaboración propia, 2024). ..	57
Figura 10: Metodología ágil de desarrollo de software con varios sprint's, (Elaboración propia, 2024).	58
Figura 11: Diseño hipotético de interfaz A, (Elaboración propia, 2023).	59
Figura 12: Diseño Hipotético de Interfaz B, (Elaboración propia, 2023).....	59
Figura 13: Propuesta de diseño con elementos mínimos para la identificación de elementos n en la interfaz I, (Elaboración propia, 2024).....	66
Figura 14: Diseño visual de propuesta INUSINET, (Elaboración propia, 2024).	73
Figura 15: Diseño propuesto para identificar EVUSINET visualmente, (Elaboración propia, 2024). ..	82
Figura 16: Diagrama integral de la propuesta EVUSINET, (Elaboración propia, 2024).	83
Figura 17: Sección de pruebas de uso del marco de trabajo EVUSINET, (Elaboración propia, 2024).	86
Figura 18: Diagrama de protocolo experimental general para la implementación correcta de las técnicas de eye-tracking en el análisis de interfaces, (Elaboración propia, 2024).	89
Figura 19: Sección de pruebas de uso del marco de trabajo EVUSINET, (Elaboración propia, 2024).	91
Figura 20: Procedimiento experimental para una propuesta de interfaces paralelas, (Elaboración propia, 2024).	91
Figura 21: Ejemplo de interfaz de tienda WEB, (Elaboración propia, 2024).....	92
Figura 22: Eventos establecidos para la interfaz, (Elaboración propia, 2024).....	93
Figura 23: Ejemplo de Interfaz de tienda WEB con elementos catalogados y diferenciados, (Elaboración propia, 2023).....	95
Figura 24: Propuesta A de áreas de Interés de figura 21, (Elaboración propia, 2023).	96
Figura 25: Propuesta B de áreas de Interés de figura 21, (Elaboración propia, 2023).	96
Figura 26: Escala de Likert en usabilidad de sistemas (Lukito Edi et al., 2018).....	99
Figura 27: Sección de EVUSINET: Evaluación, (Elaboración propia, 2023).	100
Figura 28: Visualización de recorrido de uso por parte del usuario hipotético en una interfaz con elementos idénticos, (Elaboración Propia, 2024).	103
Figura 29: Visualización de recorrido de uso por parte del usuario hipotético en una interfaz con elementos de diferentes dimensiones, (Elaboración propia, 2024).	104

Figura 30: Visualización de recorrido de uso por parte del usuario hipotético en una interfaz con elementos de diferentes dimensiones y títulos reducidos, (Elaboración propia, 2024). 104

Figura 31: Visualización del recorrido de un usuario ideal en una interfaz con elementos idénticos, pero de diferente color, (Elaboración propia, 2024). 106

Figura 32: Visualización del recorrido del usuario ideal uno en una interfaz con elementos idénticos distribuidos de forma triangular, (Elaboración propia, 2024). 107

Figura 33: Visualización del recorrido del usuario ideal dos en una interfaz con elementos idénticos distribuidos de forma triangular, (Elaboración propia, 2024). 108

Figura 34: Visualización del recorrido de ambos usuarios ideales en una interfaz con elementos idénticos distribuidos de forma triangular, (Elaboración propia, 2024). 108

Figura 35: Resultados de tres pruebas de eye-tracking de fijaciones y movimientos sacádicos sobre la interfaz de estudio, (Elaboración propia, 2024). 113

Figura 36: Diagrama de alto nivel del componente login, (Elaboración propia, 2023). 123

Figura 37: Diagrama de contexto del componente login, (Elaboración propia, 2023). 124

Figura 38: Diagrama de contendores del componente login, (Elaboración propia, 2023). 125

Figura 39: Diagrama de componentes y subcomponentes del componente login, (Elaboración propia, 2024). 126

Figura 40: Diagrama de clases del componente login, (Elaboración propia, 2024). 127

Figura 41: Interfaz Dinámica de Creación de Usuario Fuente: (Elaboración propia, 2023). 130

Figura 42: Diseño de Interfaz de Formulario Convencional 132

Figura 43: Diseño Experimental Interno de TobiiProLab, (Elaboración propia, 2024). 133

Figura 44: Instrucciones para el usuario de la prueba Fuente: (Elaboración propia, 2023) 133

Figura 45: Herramienta de Áreas de Interés de TobiiProLab, (Elaboración propia, 2024). 135

Figura 46: Áreas de interés para la interfaz de estudio con diseño dinámico, (Elaboración propia, 2024). 136

Figura 47: Áreas de interés para la interfaz de estudio con diseño de formulario convencional, (Elaboración propia, 2024). 136

Figura 48: Diagrama de eventos de las pruebas de uso del par de propuestas de diseño de interfaces del caso de estudio del componente login, (Elaboración propia, 2024). 137

Figura 49: Delimitación de eventos para las interfaces de estudio (Elaboración propia, 2023). ... 137

Figura 50: Grabación completa dividida en periodos (Elaboración propia, 2023). 138

Figura 51: Tiempos de uso de cada evento de una única prueba del componente login, (Elaboración propia, 2024). 138

Figura 52: Elección de las métricas, eventos y pruebas que forman el data-set para la evaluación objetiva del uso de interfaces mediante eye-tracking del caso de estudio del componente login, (Elaboración propia, 2024). 140

Figura 53: Mapa de calor de la interfaz 1 obtenida de una prueba aleatoria del componente login, (Elaboración propia, 2024). 142

Figura 54: Mapa de calor de la interfaz 2 obtenida de una prueba aleatoria del componente login, (Elaboración propia, 2024). 142

Figura 55: Visualización del camino de abeja del recorrido de mirada de la interfaz 1 de un único sujeto de prueba, Fuente: (Elaboración propia, 2023) 143

Figura 56: Visualización del camino de abeja del recorrido de mirada de la interfaz 2 de un único sujeto de prueba Fuente: (Elaboración propia, 2023) 144

Figura 57: Visualización de ruta de recorrido con fijaciones de la interfaz 1 de un único sujeto de prueba, (Elaboración propia, 2024). 145

Figura 58: Visualización de ruta de recorrido con fijaciones de la interfaz 1 de un único sujeto de prueba, (Elaboración propia, 2024). 145

Figura 59: Resultados de calibración de prueba retirada del análisis de eye-tracking, (Elaboración propia, 2024). 147

Figura 60: Login en modo detallado de Easy-Bank Vision, (Elaboración propia, 2024). 155

Figura 61: Login en modo simple del Easy-Bank Vision, (Elaboración propia, 2024). 156

Figura 62: Interfaz Home de Easy-Bank Vision, (Elaboración propia, 2024). 157

Figura 63: Interfaz de Transferencias de Easy-Bank Vision, (Elaboración propia, 2024). 158

Figura 64: Interfaz Saldos y Movimientos de Easy-Bank Vision, (Elaboración propia, 2024). 159

Figura 65: Interfaz Pago Tarjeta de Easy-Bank Vision, (Elaboración propia, 2024). 159

Figura 66: Diagrama de casos de uso para Easy-Bank Vision, (Elaboración propia, 2024). 160

Figura 67: Diagrama de navegación de Easy-Bank Vision, (Elaboración propia, 2024). 161

Figura 68: Diagrama de protocolo experimental de Easy-Bank Vision, (Elaboración propia, 2024). 163

Figura 69: Diseño experimental para la evaluación de Easy-Bank Vision en la herramienta de Diseño de TobiiProLab, (Elaboración propia, 2024). 165

Figura 70: Uso de la herramienta para desplegar URL como un estímulo de TobiiProLab, (Elaboración propia, 2024). 165

Figura 71: Áreas de Interés para la interfaz simple del Login de Easy-Bank Vision, (Elaboración, propia, 2024). 166

Figura 72: Áreas de Interés para la interfaz detallada del Login de Easy-Bank Vision, (Elaboración, propia, 2024). 167

Figura 73: Áreas de Interés para la interfaz de saldos y movimientos de Easy-Bank Vision, (Elaboración, propia, 2024). 168

Figura 74: Áreas de Interés para la ventana emergente del caso de uso Pagar Tarjeta de Easy-Bank Vision, (Elaboración propia, 2024). 168

Figura 75: Áreas de Interés para la interfaz Transferir de Easy-Bank Vision, (Elaboración propia, 2024). 169

Figura 76: Diagrama de secuencia de eventos de las pruebas de Easy-Bank Vision, (Elaboración propia, 2024). 170

Figura 77: Eventos establecidos en una prueba de Easy-Bank Vision, (Elaboración propia, 2024). 170

Figura 78: Mapas de calor de una sola prueba del conjunto de interfaces de estudio de Easy-Bank Vision, (Elaboración propia, 2024). 172

Figura 79: Mapa de recorrido de abeja de una sola prueba del conjunto de interfaces de estudio de Easy-Bank Vision, (Elaboración propia, 2024). 173

Figura 80: Mapas de recorrido de fijaciones de una sola prueba del conjunto de interfaces de estudio de Easy-Bank Vision, (Elaboración propia, 2024). 174

Figura 81: Ventana de generación de exports de la herramienta TobiiProLab para obtener resultados de la interfaz de estudio Login Detallado, (Elaboración propia, 2024). 176

Figura 82: Áreas de Interés de Home y Saldos y Movimientos congregados para analizar ambas interfaces individualmente, (Elaboración propia, 2024). 176

Figura 83: Visualización en recorrido de abeja del patrón de uso de la interfaz 1 de la población de sujetos de prueba, (Elaboración propia, 2024). 180

Figura 84: Visualización en recorrido de abeja del patrón de uso de la interfaz 2 de la población de sujetos de prueba, (Elaboración propia, 2024). 181

Figura 85: Mapa de calor de los resultados de la población de sujetos de prueba en la interfaz 1, (Elaboración propia, 2024). 181

Figura 86: Mapa de calor de los resultados de la población de sujetos de prueba en la interfaz 2, (Elaboración propia, 2024). 182

Figura 87: Visualización de ruta de recorrido con información de fijaciones de la interfaz 1 de la población de sujetos de prueba, (Elaboración propia, 2024. 182

Figura 88: Visualización de ruta de recorrido con información de fijaciones de la interfaz 2 de la población de sujetos de prueba, (Elaboración propia, 2024). 183

Figura 89: Elementos peor evaluados en ICE (azul), ICU (naranja) e ICR (amarillo) de la interfaz dinámica, (Elaboración propia, 2024). 186

Figura 90: Elementos peor evaluados en ICE (azul), ICU (naranja) e ICR (amarillo) de la interfaz convencional, (Elaboración propia, 2024). 190

Figura 91: Comparativa de las figuras 83 y 84 respectivamente, (Elaboración propia, 2024) 191

Figura 92: Comparativa de las figuras 63 y 64 con los elementos peor evaluados y, remarcado en rojo, elementos que son detectados por el evaluador como puntos débiles de uso, (Elaboración propia, 2024). 193

Figura 93: Visualizaciones de los patrones de uso generales de la interfaz Login Detallado (de izquierda a derecha mapa de calor, mapa de fijaciones y recorrido de abeja), (Elaboración propia, 2024). 196

Figura 94: Visualizaciones de los patrones de uso generales de la interfaz Login Simple (de izquierda a derecha mapa de calor, mapa de fijaciones y recorrido de abeja), (Elaboración propia, 2024). 196

Figura 95: Visualizaciones de los patrones de uso generales de la interfaz Home (de izquierda a derecha mapa de calor, mapa de fijaciones y recorrido de abeja), (Elaboración propia, 2024). ... 197

Figura 96: Visualizaciones de los patrones de uso generales de la interfaz Saludos y Movimientos (de izquierda a derecha mapa de calor, mapa de fijaciones y recorrido de abeja), (Elaboración propia, 2024). 198

Figura 97: Visualizaciones de los patrones de uso generales de la interfaz Pago Tarjeta (de izquierda a derecha mapa de calor, mapa de fijaciones y recorrido de abeja), (Elaboración propia, 2024). 199

Figura 98: Visualizaciones de los patrones de uso generales de la interfaz Transferencias (de izquierda a derecha mapa de calor, mapa de fijaciones y recorrido de abeja), (Elaboración propia, 2024). 200

Figura 99: Comparación entre mapas de recorrido de fijaciones de las interfaces Login Detallado y Login Simple respectivamente, (Elaboración propia, 2024). 214

Resumen

Uno de los primeros puntos que revisa este trabajo son las características de los procesos de desarrollo de software y como, si bien existe mucha documentación al respecto, hay muchos nuevos tipos de enfoque de diseño que cambian radicalmente el producto desarrollado.

De entre los aspectos más influyentes en el diseño de interfaces son los patrones de diseño y, más recientemente, el diseño centrado en el usuario. En este trabajo se revisan las metodologías de desarrollo de software y diseño y evaluación de interfaces para implementar dentro de los conceptos de ingeniería de software la tecnología novedosa y precisa del eye-tracking.

Para poder combinar la ingeniería de software con el eye-tracking primero se repasa a detalle la teoría de atención visual para posteriormente proponer una selección de métricas de eye-tracking que respalden un conjunto de índices para evaluar el uso de una interfaz de forma objetiva mediante esta tecnología.

Ahora bien, cada proceso de desarrollo de software es único y las propiedades de las interfaces también, para que más procesos de desarrollo de software e investigaciones puedan implementar las técnicas y propuestas de este trabajo sobre las técnicas eye-tracking en la ingeniería de software se propone un marco de trabajo referente a este tema.

Para compartir la implementación de las técnicas de eye-tracking sobre un proceso de desarrollo de software y en el diseño y evaluación de interfaces se exponen dos casos de estudio experimentando con las propuestas de eye-tracking y las características del software a producir.

Finalmente, los resultados abordan una novedosa manera de interpretar los datos de una prueba de uso de interfaces mediante mapas de visualización y la aplicación de los índices propuestos en este trabajo, los cuáles ofrecen valores cuantitativos de origen biométrico que son objetivos y que es una propiedad muy poco vista en evaluaciones más convencionales.

Abstract

One of the first points that this paper reviews are the characteristics of software development processes and how, although there is a lot of documentation on the subject, there are many new types of design approaches that radically change the developed product.

Among the most influential aspects of interface design are design patterns and, more recently, user-centered design. In this paper we review software development and interface design and evaluation methodologies to implement within software engineering concepts the novel and accurate eye-tracking technology.

To combine software engineering with eye-tracking, we first review in detail the theory of visual attention and then propose a selection of eye-tracking metrics that support a set of indices to evaluate the use of an interface objectively through this technology.

Now, each software development process is unique and so are the properties of the interfaces, so that more software development processes, and research can implement the techniques and proposals of this work on eye-tracking techniques in software engineering a framework is proposed regarding this topic.

To share the implementation of eye-tracking techniques in a software development process and in the design and evaluation of interfaces, two case studies are presented, experimenting with eye-tracking techniques in a software development process and in the design and evaluation of interfaces.

Introducción

Han pasado bastantes décadas desde que el ser humano ha aprovechado las capacidades de los sistemas computacionales para resolver problemas o usarlas como herramientas. Pasando desde aplicaciones de grado militar hasta hoy en día usarlas para los fines más triviales. Durante todo este proceso de constante y vertiginosa evolución han surgido una cantidad tal de conceptos que transformaron una rama de las matemáticas de combinatoria y permutaciones a una ciencia de estudio y pilar de muchos aspectos de la vida moderna.

Durante todo este periodo se han probado una extensa cantidad de métodos para el funcionamiento de cualquier sistema. Desde tarjetas perforadas hasta mecanismos de interacción táctiles, pasando por otra infinidad de periféricos y el teclado y ratón más convencionales. Todos estos cambios en las formas de uso representan hoy en día una complejidad a la hora de estandarizar ciertos los elementos que conforman una interfaz.

Además de los efectos que los elementos de hardware provocan en el modo de usar un software, también existen muchas investigaciones sobre como los diseños intangibles de un programa también afectan en cómo el usuario percibe el programa (Taniş et al., 2022; Tullis, 2019).

En este trabajo se abordan los efectos de las propiedades de diseño de un sistema en la forma en la que el usuario hace uso de este. Analizando e infiriendo los patrones de uso e interacción y cotejándolas con el rendimiento esperado al desplegar el producto.

Como se ha visto en tiempos modernos, el éxito o fracaso de un proceso de desarrollo de software de alguna empresa o institución depende mucho de cómo y con qué fines se usará. Es evidente que las compañías de *e-commerce* buscan mejor alcance y facilidades a la hora de atraer y retener usuarios, sin embargo, en un ambiente académico las prioridades pasan a ser diferentes.

La gran variedad de ámbitos, contextos y objetivos con las que se diseña un proceso de desarrollo de software generalmente impiden un desarrollo perfecto y que cumpla todas las cuotas de eficiencia, calidad y costos. Este último punto ha estimulado la creación de paradigmas bien definidos que encasillan ciertos patrones de desarrollo e incluso de uso. Sin embargo, hoy en día, estos paradigmas están lejos de ser infalibles y a prueba de errores o retrasos de desarrollo.

La principal pregunta que surge entre los desarrolladores sería: “¿Cómo podemos minimizar los riesgos de producción de software y acercar el producto al modelo que más agrade y mejore el rendimiento del usuario?”. Si bien para responder esta pregunta se necesitaría un trabajo muy extenso que abarque miles de casos de estudio por cada tipo de desarrollo, en esta investigación se propone un análisis práctico de las capacidades de un sistema y sus interfaces para mejorar su uso a la par que se proponen tipos de pruebas para evaluar de mejor manera cómo se está usando.

Aunque la idea anterior ya se ha analizado desde muchos puntos de vista con anterioridad, es preciso mencionar que la aparición de nuevas técnicas de obtención de datos mediante seguimiento de mirada expande ampliamente las capacidades de desarrolladores y diseñadores de software para analizar el producto desde nuevas perspectivas.

En los contenidos de esta investigación se aborda de manera teórica y práctica las técnicas y conceptos que en la actualidad son ampliamente usados y que, aunque pueden parecer estándares ya muy definidos, aún tienen un margen de mejora bastante considerable. Uno de los aspectos que más se ha estudiado buscando mejorar las condiciones de uso ha sido el diseño de interfaces, sin embargo, este estudio busca también considerar el diseño con el cual se concibe el software, así como los procesos de desarrollo y las pruebas que se aplican para probar el desempeño de un programa y sus características.

Después de repasar los antecedentes que sirven como un contexto histórico que permiten al lector conocer las primeras formas con las que se interactuaba con una computadora y cómo el concepto fue cambiando, dependiendo del mercado o los paradigmas de cada época, se exponen los conceptos clave que permiten compartir y proponer los experimentos y hallazgos del trabajo.

El apartado de metodologías de desarrollo establece los procedimientos que se llevaron a cabo para estudiar los procesos de desarrollo de software y mejorarlos aplicando las nuevas tecnologías de análisis de uso de sistemas. A su vez, se revisa la teoría de pruebas de usabilidad para de igual forma buscar la mejora tanto de los procesos como de los resultados sujetos a la aplicación de estas técnicas de desarrollo de software.

Dado que la investigación sugiere el análisis de algunas técnicas de desarrollo y pruebas de software es conveniente establecer un caso de estudio que permita aplicar en este las propuestas que plantea esta investigación. En el apartado de desarrollo es donde se puede apreciar los análisis mencionados anteriormente.

1. Planteamiento del problema

La manera en la que se utiliza un equipo de cómputo y sus eventuales interfaces es un proceso mucho más complejo de lo que pueda parecer. Si bien en el año en el que se publica este trabajo la mayoría de las personas de país y del mundo ya se exponen frecuentemente a miles de productos de software, siguen existiendo plataformas con diseños de interacción poco agradables, poco eficientes o que no logren que el usuario complete su tarea.

Lo anterior hace evidente la necesidad de generar mejores evaluaciones ya que siguiendo métodos convencionales, la información está principalmente inclinada hacia valores cualitativos y, por ende, subjetivos. Pero antes de abordar a detalles las características de las pruebas de uso de un software y la aplicación de nuevas tecnologías se hace un repaso histórico de la evolución de las características más buscadas en la interacción humano computadora a lo largo de la historia.

Posteriormente, se aborda el problema a resolver con la justificación y preguntas y objetivos de investigación correspondientes con el fin de exponer la relevancia de este trabajo de investigación, el cual busca en todo momento compartir cada hallazgo sobre la implementación de la novedosa técnica de eye-tracking en el campo de la ingeniería de software como un primer paso para proponer un nuevo nivel de análisis de un producto de software.

1.1 Antecedentes

En los principios de la programación los estándares de desarrollo eran un aspecto que no era tomado en cuenta para la construcción del software. Las capacidades de un sistema estaban limitadas por los algoritmos, el especialista que se encargaba de interactuar con la máquina y sobre todo el equipo donde se está ejecutando el software (Sammet & Mahoney, 1995). Como se menciona en el trabajo de (Barra, 1998) tomó bastante tiempo para que un mismo sistema pudiera funcionar en más de dos equipos sin tener que hacer cambios importantes al código base y, posteriormente, a que un sistema pudiera ser usado por otras organizaciones, incluso otra persona.

La idea anterior expresa la poca relevancia que tenía la planificación de la interacción humano computadora con que los primeros softwares que fueron desarrollados. Por consiguiente, los aspectos actuales con los que se puede medir grosso modo la calidad del software, no estaban estipulados siquiera. Uno de los primeros puntos que resalta si se comparan los métodos actuales

TESIS TESIS TESIS TESIS TESIS

con los métodos usados en los años sesenta es la forma de comunicarse con el sistema (Barra, 1998). Como se aborda en la investigación de (Lawson, 2018) una de las principales dificultades al usar el software eran los mecanismos de interacción que proporcionaba el hardware. Durante varias décadas se buscó cambiar el paradigma que hasta el momento impedía un desarrollo planificado y sobre todo estandarizado para poder ejecutar un mismo software en otro hardware (Lawson, 2018).

También hay que tomar en cuenta que hoy en día al usuario promedio de equipos de cómputo le podrá parecer una obviedad que un mismo sistema operativo pueda servir en hardwares muy diferentes entre sí, pero en la década de los setenta este concepto apenas se empezaba a explorar. Los primeros equipos personales de cómputo fueron posibles gracias a la unión de varios componentes que hasta hace muy poco estaban aislados unos de los otros. Con la necesidad de unir varias tecnologías en el menor espacio posible surgieron nuevos problemas y soluciones (Tullis, 2019).

Como se mencionó en párrafos anteriores, hasta los setenta, la interacción estaba limitada a tarjetas perforadas o comandos que sólo el programador del sistema conocía, con el nuevo paradigma de un equipo que pudiera resolver diferentes problemas y tareas se podía empezar a manejar el término de “entorno” (Ribera Turró, 2005).

En un entorno conviven varios componentes que fueron diseñados por separado pero que en conjunto expanden ampliamente las capacidades de los equipos para resolver pequeños problemas para un usuario. Ya con un punto de partida establecido, durante los años ochenta los diseñadores de los equipos de cómputo afrontaron el reto de mejorar las capacidades de sus equipos en todos los aspectos. Nuevas teorías surgieron, así como inventos prácticos que fueron poco a poco moldeando y estandarizando todo lo que compone al equipo y por extensión, al entorno, así como sus métodos de procesamiento, almacenamiento, despliegue de interfaces y, por supuesto, los procesos de interacción con el equipo (Bødker, 2015; Ribera Turró, 2005).

Al estallido de las innovaciones durante los ochenta se le denomina “la era del equipo personal de cómputo”. En este periodo se hizo evidente que aprovechar las posibilidades que permitían los nuevos equipos dependía de la capacidad del usuario para hacer uso de ellas. Es decir, conforme sea mejor la interacción entre el usuario y el equipo de cómputo mejor será la experiencia y productividad al usar el entorno (Ribera Turró, 2005).

El concepto de usabilidad surgió para identificar las características que cambian la capacidad del entorno para ser usado de una forma eficiente por el usuario (Marques et al., 2021). Sin embargo, conforme los entornos se fueron complejizando también surgió la necesidad de ofrecer nuevas herramientas para mejorar la interacción con éste. Como se mencionó en ideas anteriores, durante la era del equipo personal de cómputo, existían bastantes enfoques sobre lo que era la calidad de uso de un entorno. Mientras unos se enfocaban en las interfaces del software (Clement & Gotlieb, 1986) otros mencionaban que las herramientas de interacción eran el factor más importante al usar el entorno (Eastman, 1964).

Al concepto anterior que hace referencia a que es posible mejorar un entorno gracias a los instrumentos en aquel momento se le llamó “pre-usabilidad” (o también llamado contexto de uso). Este concepto se empezó a considerar cuando el mercado se llenó de equipos de cómputo y fue fácil encontrar fallos tanto del funcionamiento del sistema como de dificultades de los usuarios a la hora de usar el equipo. Para la gran mayoría de usuarios que durante décadas habían realizado sus tareas de oficina a mano era bastante complicado remplazar sus actividades directamente con un equipo de cómputo (Tullis, 2019).

Según (Barra, 1998) el concepto de software no fue correctamente entendido por el grueso de la población fuera del gremio de los desarrolladores e ingenieros hasta alrededor de dos o tres décadas desde que surgió el concepto. Debido al constante rechazo al cambio, la industria prefirió actuar activamente en generar marcos establecidos de desarrollo, manejo de datos, despliegue de interfaces, entre otros métodos. Además de lo anterior, empresas consideraron la capacitación del usuario mediante manuales de uso impresos guiar a este en el uso del sistema (Abbott, 1977; Priestley & Rintjema, 1995).

Si bien se han abordado las primeras décadas de los procesos de software por corrientes o tendencias, es conveniente en este punto virar la atención hacia los lenguajes de programación específicos que, gracias a los paradigmas que proponían, lograron mejorar el software desde un punto de vista técnico en lugar del uso. En otras palabras, los desarrolladores de software no se limitaron únicamente al uso comercial de equipo, en su lugar, buscaban llevar al límite las capacidades del hardware para mejorar las capacidades del software, algo opuesto al planteamiento de pre-usabilidad.

El lenguaje de programación C fue un pilar para los primeros paradigmas de programación y procesos de desarrollo a mediados de la década de los ochenta, el lenguaje tuvo carencias a la hora

de abstraer ciertos conceptos de la realidad al enfoque estructurado y por lotes que manejaba C. Incluso con las implementaciones que tuvo C++ (entendiéndose que para esta investigación no es relevante hacer la distinción si C++ es un lenguaje independiente o un continuación directa de C) enfocadas al manejo módulos, el lenguaje seguía teniendo problemas para manejar grandes volúmenes de información de una forma cómoda y de fácil interpretación (Stroustrup, 1993).

Una de las principales características que hizo de Java un lenguaje de programación ampliamente usado fue precisamente su capacidad de abstracción y su paradigma de funcionar envuelto en una máquina virtual (Sammet & Mahoney, 1995). Como se mencionó anteriormente, una de las primeras desventajas que tuvo el software es que eran desarrollados como soluciones a problemas muy específicos en ambientes no estandarizados. Con una máquina virtual era posible (a costo de carga de procesamiento) generalizar medianamente el software para entornos diferentes entre sí. A pesar de que en sus inicios a algunos desarrolladores le parecía preocupante emplear buena parte de la capacidad de procesamiento solamente en ejecutar el entorno del lenguaje, la constante evolución de procesadores, mecanismos de almacenamiento y de los propios sistemas operativos permitió que Java y su máquina virtual fueran vistas con optimismo por la otra parte de los desarrolladores (Reinholtz, 2000).

Con la portabilidad a su favor, Java poco a poco fue ganando terreno a otros lenguajes de programación sobre todo en sistemas pensados para su venta al público. Llegó a tal el éxito el lenguaje que incluso su modelo fue estudiado dentro del gremio de desarrolladores de sistemas operativos. El nuevo paradigma Java Virtual Machine (Máquina Virtual de Java traducción al español de las siglas JVM) adoptó el lema de "Write Once, Run Everywhere", presumiendo las capacidades del lenguaje para ejecutarse independientemente del sistema (Tyma, 1998).

Si bien en el lenguaje de programación Java es un caso de estudio en sí, la popularidad que tuvo a partir de la década de los noventa y su uso ya ha sido ampliamente estudiado por muchos autores. El enfoque orientado a objetos del lenguaje permitió una capacidad de abstracción que hizo del lenguaje una herramienta muy potente para estandarizar conceptos y sobre todo representarlos.

Ya con un ambiente de ejecución generalizado para cada vez más sistemas se renovó el interés por la estandarización, el reúso y la optimización de procesos de desarrollo. Como se expresa en la investigación de (Carr et al., 2004) el "Lenguaje Unificado de Modelado" (o mejor conocido por las siglas en inglés de "Unified Modeling Language": UML) no es solo una herramienta de representación del mismo sistema a programar, es un análisis complejo de todos los aspectos de este.

Se tiende a debatir sobre las capacidades del UML como un enfoque que permita mejorar las capacidades de un sistema para ser usado con comodidad y eficiencia (Glinz, 2000), sin embargo, es imposible negar la influencia que tuvo el paradigma para formalizar procesos de desarrollo, así como estándares de programación y la fomentación de un análisis que incluyera un análisis de procesos, contexto y usabilidad que mejoraran el software independientemente del hardware provisto.

Una vez que el UML se estableció como un estándar para la programación incluso fuera del lenguaje Java se alcanzó un auge del desarrollo de sistemas para organizaciones y desarrollos específicos. Fue mucho más sencillo desarrollar el software y a su vez, los usuarios sentían cada vez menos rechazo al cambio debido a que muchos sistemas compartían características en sus interfaces, así como en sus mecanismos de interacción.

1.2 Problemática a resolver

Ya se ha planteado el desarrollo de software como una secuencia de procesos de construcción que tratan de optimizar los procesos de producción del sistema. Dependiendo de las necesidades del cliente y los paradigmas de los programadores es posible que la elaboración del producto pueda complicarse durante el desarrollo, también es posible que la recolección de requerimientos haya sido tan pobre que el resultado final no cumpla los estándares de calidad, agrado o eficiencia de uso del cliente.

Como toda rama de la ingeniería es preciso tener flexibilidad para afrontar un problema. En el caso de la ingeniería de software se busca reducir costos y tiempos de producción, a la par que se vigila la calidad de uso y diseño del sistema para acercar lo más posible el producto final al producto planteado al inicio del desarrollo.

Las metodologías de desarrollo son útiles para verificar el cumplimiento de los requerimientos lógicos de un sistema, pero el aparatado de uso es mucho más difícil de evaluar. Las pocas metodologías que aplican la retroalimentación del usuario como un mecanismo de medición de usabilidad o de calidad de diseño tienen que apegarse a pruebas que, además de que el resultado puede cambiar dependiendo del contexto de uso, también las experiencias previas del usuario añaden un respaldo subjetivo a sus evaluaciones que a largo plazo genera ambigüedad en los diseños propuestos y lanzados.

La dificultad para garantizar el uso de un sistema representa un freno en sistemas destinados a la docencia, inclusión, comunicación y con fines de lucro. En otras palabras, si existe incertidumbre de las capacidades de un software para resolver un problema específico mediante su uso, es posible que en la práctica este sistema no funcione de la manera esperada.

1.3 Justificación

El diseño de interfaces no es una disciplina que se encasille en algún ámbito específico de uso, por lo contrario, interviene directamente en la percepción y sensaciones que se generan en el usuario al usar el sistema.

Cuando una persona interactúa con un software, tanto los ojos como el cerebro comienzan una complicada cadena de procesos que buscan la interpretación de la información que se le está mostrando. La vista tiene movimientos oculares que en muchas ocasiones no son perceptibles para la persona; Como lo son los movimientos sacádicos, fijaciones, enfoques (diámetro de la pupila), entre otros. De acuerdo con (Duchowski, 2017) cuando los estímulos son presentados al usuario, este recorre rápidamente la interfaz con la mirada para brindarle a la memoria de trabajo una “primera impresión” de los contenidos del estímulo, la memoria de trabajo hace uso de experiencias previas para darle una interpretación rápida de lo que se espera observar (Tupikovskaja-Omovie & Tyler, 2022). Cuando se realiza una primera interpretación, inmediatamente la memoria de trabajo identifica las partes que menos pudo identificar para darle más detalle a la información en pantalla y poder reaccionar de mejor manera ante el estímulo.

Tomando en cuenta que el proceso anteriormente mencionado es solo la punta del iceberg de los procesos oculares y cognitivos que intervienen en la forma de percibir una interfaz o software completo, además de los eventos tienen duración en el orden de los milisegundos, durante muchos años ha sido prácticamente imposible estudiar el comportamiento de la mirada en pantallas el tiempo suficiente para obtener información reveladora de su uso. Es gracias a las nuevas técnicas de seguimiento de mirada que es posible plantear nuevas métricas e hipótesis de los datos biométricos del usuario.

Si bien la interacción de un usuario con un sistema activa en el primero complicadísimo sistemas de atención visual, interpretación cognitiva y reacción ante los estímulos visuales que difieren entre cada usuario, es posible con las tecnologías de seguimiento de mirada estimar los patrones de uso.

Mencionado lo anterior se puede decir que conforme un sistema es más complejo, existe una mayor diferencia entre los patrones de uso de cada uno de los usuarios.

A pesar de que resulta muy complicado proponer la estandarización de los diseños de interfaces para todos los sistemas desarrollados, es posible analizar a detalle los elementos en común que conforman el diseño e interpretar cómo la dimensión, colores y disposición de los elementos están influyendo en la experiencia del usuario. Con este enfoque es posible dividir una interfaz en bloques de contenido de los cuales se pueden determinar un conjunto de objetivos de uso.

Retomando a (Duchowski, 2017) y también consultando el trabajo de (Johnson, 2010) es importante destacar la relevancia que tiene el sentido de la vista para los procesos de reacción e interacción en muchos procesos de la vida diaria de una persona. Mencionado lo anterior, se hace evidente la necesidad de generar diseños que sean atractivos sin descuidar en ningún momento la funcionalidad de este.

Tomando en cuenta que un desarrollo de software correctamente implementado tiene una etapa de recolección de requerimientos y objetivos, es preciso proponer que se recaben objetivos específicos y contextuales de uso para cohesionar de mejor manera la funcionalidad del sistema con el diseño de uso.

La necesidad de la industria de software para mejorar la interacción con el usuario no es un aspecto que solo las empresas de tiendas en línea, redes sociales o de *streaming* desean (Martínez Barros & Vásquez Muriel, 2021). Otros campos como la docencia en medios digitales o rehabilitaciones a distancia tienen deficiencias para generar evaluaciones en los primeros y diagnósticos en los segundos (Balladares-Burgos, 2018; Martínez, 2017). Si bien en los campos anteriormente mencionados es necesario considerar un contexto mucho más amplio para asegurar un ambiente académico o de consulta médica mediante medios digitales aptos para su fin, es conveniente mejorar los diseños que rodean dicho contexto. Es decir, aunque la mejora de los diseños de interfaces y flujos de interacción pueden parecer un análisis pequeño comparado con todos los aspectos que intervienen en los procesos de interacción con el usuario, no deja de ser un aspecto vital para el análisis de uso de un sistema.

1.4 Preguntas de Investigación

Tomando como punto de partida la intención de generar un análisis innovador que mejore las técnicas que se usan en la industria de software para generar sistemas en los que la experiencia de uso sea lo más agradable posible se plantean a continuación un conjunto de preguntas de investigación.

1.4.1 Pregunta general

- ¿Cómo se pueden involucrar las técnicas de eye-tracking en los procesos de desarrollo de software, y más específicamente en el diseño y evaluación de interfaces?

1.4.2 Preguntas específicas

- ¿Qué métricas de eye-tracking permiten una primera aproximación al análisis objetivo del uso de una interfaz?
- ¿Cómo se puede elaborar un marco de trabajo para involucrar las técnicas de *eye-tracking* en los procesos de diseño, desarrollo y *testing* de software?
- ¿Qué índices se proponen para relacionar el uso de una interfaz con las métricas de *eye-tracking* en los procesos de diseño y *testing* de software?

1.5 Objetivos de la Investigación

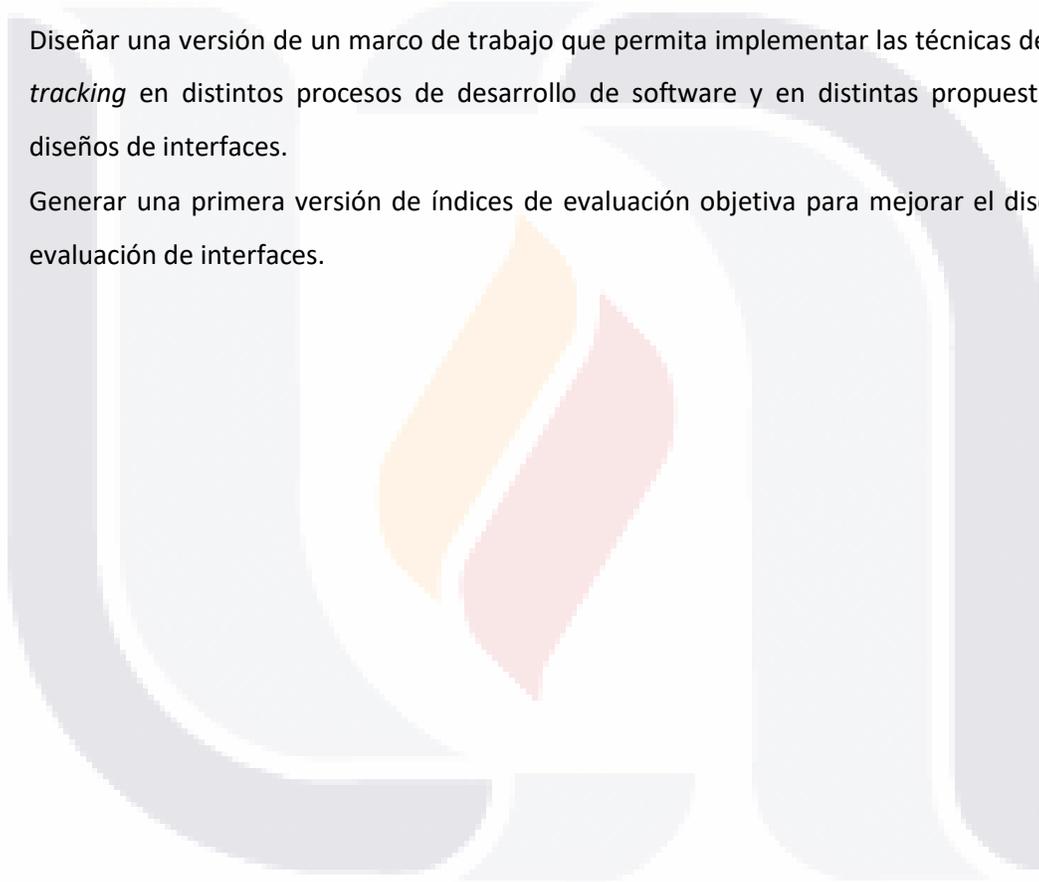
A pesar de los miles de procesos de desarrollo de software llevados a cabo, con su respectiva documentación, planeación y todos los aspectos que lo envuelven, sigue habiendo productos que no logran resolver un problema específico o que simplemente fallan de manera estrepitosa al no ser evaluados por los usuarios de manera objetiva. Para poder mejorar las probabilidades de un desarrollo exitoso se busca no solo proponer un tipo de pruebas de uso objetivas, también se busca analizar diferentes marcos de trabajo mientras se aplica el análisis mediante eye-tracking para encontrar las metodologías o diseños de proceso que mejor adapten la técnica y que puedan aplicarla en muchos más ambientes y tipos de proyecto.

1.5.1 Objetivo general

Implementar las técnicas de eye-tracking en procesos de desarrollo de software, enfocándose en los procesos de diseño y evaluación de interfaces.

1.5.2 Objetivos particulares

- Identificar las métricas obtenidas por *eye-tracking* que mejor se ajusten a la evaluación objetiva del uso de interfaces.
- Diseñar una versión de un marco de trabajo que permita implementar las técnicas de *eye-tracking* en distintos procesos de desarrollo de software y en distintas propuestas de diseños de interfaces.
- Generar una primera versión de índices de evaluación objetiva para mejorar el diseño y evaluación de interfaces.



2. Marco Teórico

Durante bastantes años muchos conceptos referentes al desarrollo y uso de software no estaban delimitados por alguna definición exacta ya que cada proceso de desarrollo de software y sus resultados divergían tanto el uno del otro que fue complicado para los desarrolladores del momento llegar a consensos o establecer conceptos y tendencias que permitieran proponer desarrollos más complejos, eficientes y con mejores resultados. Los conceptos propuestos en esta época de ambigüedad en el estudio de los procesos de desarrollo de software trajeron propuestas que incluso hoy en día son debatidas y probadas.

Antes de exponer las propuestas y enfoques aplicados en la investigación, es necesario definir varios conceptos, así como desarrollar en los diferentes paradigmas que se han aplicado con anterioridad o actualmente generan debate entre los teóricos del campo de la computación. Cabe destacar que los conceptos elegidos además de tener una influencia directa en los procedimientos aplicados en este trabajo, también se busca revisarlos en la búsqueda de áreas de mejora mediante la implementación de tecnologías innovadoras.

Previamente a la definición de conceptos ya establecidos y para fines de esta investigación se usa el concepto “software como herramienta/solución” el cuál responde al objetivo fundamental de la programación: ofrecer un mecanismo que mejore, en todo lo amplio de la palabra, las capacidades de un usuario al realizar sus tareas u operaciones. Ahora bien, el cómo se mejoran dichas capacidades conforma parte de los tópicos que se abordarán a profundidad en este marco teórico para establecer los paradigmas que mejor permitan la aplicación de nuevas tecnologías y parámetros de evaluación en el objetivo de ofrecerle al usuario mejores herramientas y solucionar sus problemas.

2.1 Interacción Humano-Computadora (HCI)

Las posibilidades actuales de la tecnología permiten que esta pueda integrarse en ámbitos a los que años antes no se pensaría que pueden implementarse. Conforme las aplicaciones de la tecnología se expanden, es necesario estandarizar las acciones o interacciones del sistema al interactuar con un equipo de cómputo. Citando la investigación de (Stephanidis et al., 2019): “Tradicionalmente, la HCI se ha centrado en el ser humano y en cómo garantizar que la tecnología satisfaga las

TESIS TESIS TESIS TESIS TESIS

necesidades de los usuarios de la mejor manera posible.” El concepto anterior, aunque a primera impresión pueda parecer ambiguo, presenta una sentencia importante. El hecho de que las propiedades de las computadoras estén dadas en medida de las características del ser humano y no al revés indica un enfoque específico de diseño y orientación del uso. En otras palabras, la interacción humano-computadora debe entenderse como la adaptación de los métodos de interacción del equipo al humano en lugar de acostumbrar al usuario a los métodos que propone el equipo.

Además de los métodos de comunicación con el equipo, la interacción humano-computadora también involucra los aspectos del entorno físico o digital del sistema. También involucra en gran medida las capacidades del software para ser usado de una forma que mejore el desempeño de los usuarios en sus casos de uso. Para lo anterior es necesario prever las necesidades del usuario y diseñar modelos que favorezcan y prioricen el contexto de uso para una mejor HCI del sistema (Ruiz et al., 2018).

Una de las grandes dificultades de la medición del HCI es que muchos de los factores que afectan directamente la experiencia no pueden ser correctamente cuantificados o fundamentados. Lo anterior aunado al incremento constante de los tipos de dispositivos a los cuales se aplican conceptos de calidad en HCI dificulta la estandarización de métodos de alta profundidad para mencionados conceptos (Cairns & Cox, 2008).

Como se menciona en el apartado de antecedentes, en varias etapas y corrientes de diseño e implementación de periféricos se priorizó la estandarización de estos con el fin de que el usuario no tuviera que adaptarse a nuevos mecanismos de interacción y permitiendo sistemas más complejos ya que el usuario tenía experiencia en el sistema. Si bien lo anterior permitió extender el alcance de la industria de desarrollo de software, pronto la necesidad de mercado forzó a las empresas a innovar (con más o menos éxito) en los mecanismos de interacción, implementación de páginas web o incluso desarrollando dispositivos nuevos como los teléfonos celulares o laptops. Esta divergencia en los modelos no fue precisamente una ventaja para los desarrolladores. Como se aborda en la investigación de (Charland & Leroux, 2011) en la época del auge de los dispositivos móviles dónde cada compañía desarrollaba tanto el hardware y el software era muy conveniente revisar las especificaciones técnicas de cada dispositivo para analizar el alcance que tendría un sistema antes de lanzarse.

Conforme algunas tecnologías iban retirándose poco a poco del mercado al no cumplir las características de innovación y calidad de uso y otras se fueron refinando, la industria volvió a buscar la estandarización de al menos algunos aspectos de sus dispositivos (Dehlinger & Dixon, 2011). Y una vez que muchos de los dispositivos usaron un sistema operativo general y por versiones muy delimitadas, los paradigmas para el diseño y desarrollo de aplicaciones se centró en el diseño estándar de interfaces (permitiendo desarrollar una sola interfaz para diferentes dispositivos) y la adopción de la metodología de desarrollo basado en componentes (Nagappan & Shihab, 2016).

2.2 Ingeniería de Software

La ingeniería de software comprende el estudio, planeación, desarrollo, revisión, evaluación, mantenimiento, lanzamiento, entre otras, de cada una de las etapas por las que tiene que pasar un producto para realizarse. Antes de establecerse la definición tan amplia que atañe a la ingeniería de software como una ciencia amplia y rigurosa existió un tipo de programación enfocado al hardware, dónde los programadores aprendían a prueba y error (Barra, 1998).

Hoy en día el éxito de cualquier producto de software depende totalmente de la aplicación de este concepto. Cada una de las ramas que se desprenden de la ingeniería de software por sí sola se pueden considerar un campo de estudio ya que afectan directamente en el desempeño del producto, así como los costos de tiempo y producción que demandó el proyecto.

Dicho lo anterior, la correcta implementación y jerarquización de cada una de las ramas en el proyecto son responsabilidad del ingeniero de software. Si bien existen varios enfoques sobre el diseño de procesos de desarrollo de software (que serán revisados en secciones posteriores más a detalle) la gran mayoría coinciden en que una buena planeación permite llevar a buen rumbo el proyecto (Engels & Whittle, 2012).

A continuación, se presentan varias de las categorías de la ingeniería de software que de manera más directa se involucran en esta investigación. Además, se debe tomar en cuenta que se busca implementar tecnologías innovadoras que permitan nuevas perspectivas en el diseño, desarrollo y evaluación de un producto de software.

2.2.1 Planeación y diseño de Software

Si bien cada una de las etapas en el proceso de desarrollo de un producto de software son relevantes, es de vital importancia mencionar la capacidad de la planeación como herramienta para mejorar la toma de decisiones, apuntar objetivos y estimar costos en tiempo y esfuerzos. Sin embargo, ningún plan está exento de imprevistos, por lo que también se contempló la mejora en los métodos para verificar, retroalimentar y evaluar el proceso.

De manera convencional, el diseño de software contempla aspectos como los requerimientos del usuario, análisis de los procesos de la organización, flujo de datos, entre otros aspectos más relacionados con el funcionamiento técnico del sistema. Si bien este enfoque permite controlar rigurosamente los procesos y estados de un sistema, existen otros muchos factores que cambian la manera en la que el usuario usa el sistema (Nagappan & Shihab, 2016).

Conforme los sistemas se hacen cada vez más abiertos al público general se hace evidente que los softwares deben contemplar el mayor número de usuarios posibles, pero no siempre fue de esta manera. Por muchos años los principales contratistas de industrias de software fueron medias y grandes empresas que buscaban hacer más eficientes sus procesos.

Hoy en día, un proyecto de software debe de manera estricta contemplar desde el tipo de tecnologías a implementar, arquitecturas de servidores, un proceso de desarrollo adecuado, pruebas que se adapten a los objetivos a cumplir, mantenimiento multiplataforma, actualizaciones, entre otras, y es la planeación del software lo que permite mantener un rumbo fijo en el proceso de desarrollo (Sommerville, 2011).

2.2.2 Marcos de Trabajo (Frameworks)

Un marco de trabajo hace referencia al estudio y estipulación de las características de un proceso de desarrollo de software que se ha realizado anteriormente. Dado que un proceso de desarrollo de software contempla muchos factores y la mayoría de las veces, aunque se apliquen los mismos métodos no se garantiza que los resultados sean los mismo. Por lo anterior, muchos desarrolladores han generado pautas para que futuros proyectos no reincidan en los errores del pasado. Lo que permite, a largo plazo ahorrar tiempo y esfuerzo en el desarrollo de un producto (Nokleberg & Hawkes, 2021).

Los marcos de trabajo tienden a ser modelos de desarrollo específicos para un problema o aplicabilidad, es decir, tienen la capacidad de expresar un enfoque de solución que se apega a ciertas características particulares, pero que también sirven como un caso de estudio que un desarrollador puede consultar para prever escenarios o anticipar los problemas en su proyecto.

Las características de un marco de trabajo o framework dependen directamente de las dificultades o enfoques aplicados en el proceso de desarrollo. Para poner un ejemplo, muchos frameworks priorizan las etapas de diseño para generar una aplicación enfocada en el usuario, mientras otras buscan desarrollos más dinámicos y menos propensos a errores de codificación. Al final, el objetivo es aumentar la capacidad del proceso para ser exitoso (Ruparelia, 2010).

2.2.3 Metodologías de Desarrollo

Un proceso de desarrollo de software se puede entender como el estudio de todo proceso que interviene en el diseño, elección y aplicación de metodología de desarrollo, pruebas, despliegue, mantenimiento y principalmente el análisis de alto nivel de un software con el fin de mejorar las capacidades de este para cumplir los objetivos establecidos o resolver un problema específico (Zumba Gamboa & León Arreaga, 2018).

Hablar de las metodologías de software es introducirse en un tópico muy entrelazado entre la teoría y la práctica. El de desarrollo de software, como se mencionó antes, es un conjunto de etapas bien diferenciadas que dotan al sistema de un orden lógico, capacidades de uso y características del diseño. Todos estos aspectos son planteados con más o menos rigurosidad en las etapas de concepción del software. De acuerdo con (Sommerville, 2011) se aborda el tema de la ingeniería de requerimientos como un punto clave para generar un producto que responda a las necesidades del usuario. Independientemente de la metodología usada, la etapa de recolección de requerimientos es el pilar con el que sistema es planificado. A su vez, aspectos de funcionalidad del sistema deben estar validados por el cliente para orientar al diseñador de software en su tarea.

Una vez que se analizan los procesos de alguna organización o se especifican a detalle las funcionalidades que se espera permitan resolver un problema mediante el uso del sistema, se debe elegir una metodología de desarrollo específica. Es importante señalar que cada sistema es tan diferente en sí que una metodología de desarrollo no es una pauta fija de instrucciones de construcción de software, es un modelo que permite orientar el software a buen rumbo, controlado

los tiempos, esfuerzos, costos y siguiendo siempre los requerimientos como puntos de control que definen la capacidad del sistema para ser usado con eficiencia y calidad.

Para declarar exitoso un proceso de desarrollo de software se debe considerar tanto el tiempo, costo, incluso ámbito de uso (académico, comercial, médico) también se deben tomar en cuenta las ventajas y desventajas de las metodologías de software.

A) Dos enfoques diferentes: Metodologías Ágil vs. Tradicional

Como se menciona en el apartado anterior, existe una gran variedad de tipos de sistemas y también son muy distantes entre sí los objetivos y problemas específicos que buscan resolver. Para abordar esta problemática la teoría propone dos variantes ampliamente diferenciadas de tipos de metodologías de desarrollo.

Las metodologías tradicionales como se le denomina actualmente se concibieron principalmente para estructurar de una manera más planificada un producto de un proceso de desarrollo de software. El problema que se buscó solucionar al diseñar modelos tan rígidos pero generalizados era que los desarrolladores de software tuvieran una línea de producción y objetivos visibles en todo momento (Velásquez Restrepo et al., 2019). Lo anterior permitió a largo plazo ahorrar tiempo y costos en los procesos de producción. Cuando los sistemas prácticamente podían salir en serie de la línea de producción, los líderes de proyecto aprovecharon para experimentar con las características del proceso de desarrollo.

Mientras algunos tomaron un camino de desarrollo riguroso que instauraba que cada fase tenía que ser diseñada, completada y probada antes de continuar con la siguiente etapa (denominada metodología en cascada) (Vite Ceballos et al., 2018), otros prefirieron optar por dotar al desarrollo de una flexibilidad tal que permitiera corregir aspectos del software que no estuvieran considerados en la planificación inicial (metodologías en espiral) (Velásquez Restrepo et al., 2019). Parecidas a las anteriores, las metodologías iterativas o por prototipos (también denominadas RUP por las siglas del inglés Proceso Racional Unificado) permiten lanzar versiones que el cliente o los desarrolladores pueden usar y evaluar antes de gastar tiempo y costos en desplegar una versión a gran escala.

Sin embargo, y aunque parezca que las metodologías RUP tienen muchos aspectos parecidos a las metodologías ágiles, en la teoría, las primeras siguen perteneciendo al enfoque de las metodologías tradicionales. Las metodologías ágiles se destacan por su gran flexibilidad para poder cambiar

muchos elementos del sistema en cualquier momento del proceso de desarrollo. Otra de las ventajas que se le atribuye a este tipo de proceso ágil o también llamado ligero, es que en lugar de detallados procesos de documentación se utiliza la revisión de prototipos por parte del cliente para asegurar que el proyecto agrade a este último. A pesar de los puntos positivos, ninguna metodología es capaz de cubrir todos los rubros para un proceso de desarrollo perfecto. Este enfoque, al inmiscuir más al cliente, le otorgó propiedades subjetivas al sistema, es decir, sin un documento de cumplimiento de objetivos o de estándares de calidad, es difícil tener la certeza de que la versión nueva es mejor que la anterior. Puede que la versión nueva sea más agradable para el usuario final, sin embargo, eso no garantiza que sea el diseño que optimice su tarea (Navarro Cadavid et al., 2013).

Si bien los estándares prácticos de desarrollo de software de la actualidad abogarían por el uso de metodologías ágiles para la elaboración de cualquier tipo de proyecto, es un paradigma que ha sido más estudiado en la práctica que en la bibliografía. La metodología ágil sigue siendo un campo experimental que puede desviar la objetividad de un proyecto.

Como opuesto al concepto de metodologías ágiles, como se mencionó anteriormente, las metodologías tradicionales tienen características objetivas que son invaluable en los procesos de investigación, análisis o formulación de nuevas técnicas de desarrollo y testing de software, verificación de procesos, entre muchos otros aspectos relevantes.

Una vez estipulado que las metodologías tradicionales poseen características más aptas para llevar a cabo pruebas experimentales sobre ellas, es conveniente explorar dos metodologías específicas que dan paso a propuestas de mejoras de los procesos, así como la búsqueda de modelos de diseño o pruebas respaldados por análisis objetivos y que optimicen tanto el agrado de uso, como la eficiencia del usuario.

B) Metodología incremental y en espiral

Es bien conocido que en la rama de industria de software existe una infinidad de procesos de desarrollo y, a su vez, otro gran número de metodologías. La eficiencia de los procesos de desarrollo está ampliamente condicionada al diseño y objetivo final del producto. Conforme se han estudiado dichos procesos de desarrollo han surgido corrientes de metodologías que plantean ser mucho más dinámicas en sus capacidades de producción y pruebas de software.

Como se menciona en las investigaciones de (Velásquez Restrepo et al., 2019; Zumba Gamboa & León Arreaga, 2018), una de las metodologías que mejor permiten tener un desarrollo controlado y que a largo plazo permite tanto una planeación como una revisión de objetivos de una forma más rigurosa es el modelo incremental.

Cómo se representa en la figura 1, la metodología de desarrollo incremental permite analizar el nivel de cumplimiento de los objetivos y requerimientos del sistema con cada pequeño avance que sea realizado (Zumba Gamboa & León Arreaga, 2018). Si bien es fácil realizar el seguimiento, una de las principales limitantes de esta metodología es que puede demorar el desarrollo de un proceso debido a la regla de cumplir los objetivos de manera secuencial. En otras palabras, la metodología incremental puede sufrir retrasos importantes si surgen problemas ya sea en la etapa de diseño dónde no se consideraron aspectos importantes del sistema o en la etapa de desarrollo en sí.

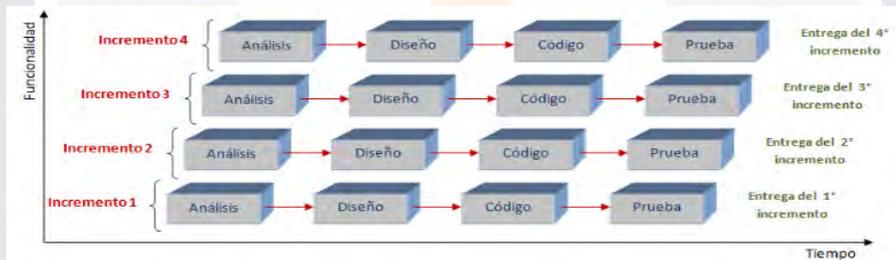


Figura 1: Diagrama de Metodología Incremental, (Zumba Gamboa & León Arreaga, 2018).

Otro gran inconveniente del modelo incremental es su apartado de pruebas. Como se mencionó antes, esta metodología prioriza el diseño previo del proceso de desarrollo, sin embargo, a la hora de hacer pruebas pueden ocurrir dos escenarios: el primero es cuando se utiliza demasiado tiempo de desarrollo, pruebas y solución de errores para garantizar el uso de un apartado del sistema, retrasando el desarrollo de los demás componentes. El segundo caso es cuando se completa un elemento del sistema, pero al realizar otros componentes se encuentran fallas de interacción o comunicación entre ambos elementos. Este último problema puede plantear modificaciones fuertes al diseño inicial del sistema, por lo que inmediatamente se deben replantear los costos, tiempos de entrega, etc.

Si bien no es un problema como tal, la rigurosidad de la planeación del modelo incremental debe delimitarse tomando en cuenta el tamaño del programa a desarrollar. Si el software propone abarcar muchas ramas de uso, escenarios u opciones, es posible que el modelo incremental no sea la opción adecuada para desarrollarlo. Por otro lado, en sistemas pequeños es una herramienta que revisa cada pequeño apartado con rigurosidad para que, de manera independiente, cada elemento funcione correctamente, pero sacrificando la integración entre estos.

Para analizar otra metodología importante en la rama de ingeniería de software se retoma el trabajo de (Zumba Gamboa & León Arreaga, 2018) donde se exponen a profundidad los cambios que han tenido ciertas metodologías para ajustarse a nuevas corrientes de desarrollo o problemáticas. En el caso particular de la metodología de desarrollo incremental, se puede entender que su evolución resultó en cierto grado al desarrollo del concepto de metodología en espiral.

La metodología en espiral (o también denominado modelo de desarrollo evolutivo) busca dotar de una mayor flexibilidad a la hora de proponer cambios en un software y reducir los riesgos de fallos importantes en el proceso de desarrollo (Prieto Álvarez et al., 2015). Por otro lado, la metodología en espiral puede llegar a ser caótica si no se planifican adecuadamente las iteraciones de desarrollo. Para poder expresar mejor la idea anterior, es importante definir el uso de una iteración en la metodología en espiral.

Como su nombre lo indica, un modelo iterativo es aquel que utiliza como fundamento la repetición de procesos para poco a poco aproximarse al objetivo inicial planteado. En este paradigma se permite que una primera versión desplegada puede ser muy alejada del diseño final, dado que el objetivo es analizar las características del sistema y reaccionar en la siguiente interacción dependiendo de los resultados de la primera. Cabe destacar que, si bien existe un riesgo de iterar más veces de las necesarias, esto solo se convierte en un retraso temporal mientras que la funcionalidad es cada vez mejor. En la figura 2 se muestra un diagrama de la propuesta de una metodología en espiral.

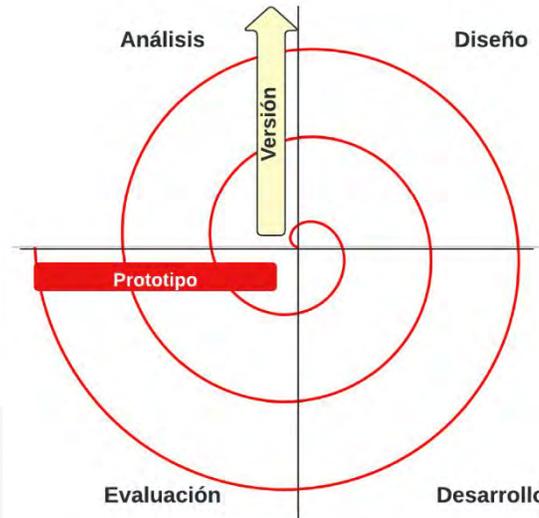


Figura 2: Diagrama de metodología de desarrollo de software en espiral, (Elaboración propia, 2024).

Aunque se menciona que la metodología en espiral es capaz de tener cambios en medio del desarrollo, esto no la convierte en un desarrollo improvisado, es decir, que el diseño del software pueda ser modificado no significa que no deba tener una planificación de desarrollo, por el contrario, un buen diseño será la pauta para planear las evaluaciones del estado actual del software en cada iteración (Prieto Álvarez et al., 2015).

Si bien la planificación del software no es tan rigurosa como en el modelo incremental, es preciso tener experiencia en los procesos que conforman cada iteración para sistematizar dichos procesos y reducir tiempos y costos de producción, que al final, es el objetivo de toda metodología.

Habiendo dicho lo anterior, para esta investigación resulta conveniente explorar y aplicar características de ambas metodologías. Si bien puede parecer que existen grandes diferencias entre las metodologías, los procesos particulares que envuelven las fases de desarrollo de ambas contemplan en general los apartados de diseño, desarrollo, pruebas y evaluación para generar resultados de calidad.

Una de las tareas de esta investigación es experimentar con las metodologías propuestas para encontrar qué elementos de cada una de ellas puede implementarse en la otra mientras se usan técnicas de eye-tracking en una de las etapas más importantes y estudiadas pero que en la práctica tienden a perder su objetividad, dificultando la planeación de la siguiente iteración o incremento.

2.2.4 Diseño de Interfaces

Como se ejemplifica en la Fig. NN una de las visualizaciones más usadas para ilustrar el método de interacción con un equipo de cómputo es, en teoría acertada, ya que es necesario un mecanismo físico para activar un cambio de estado en la máquina, sin embargo, la separación tan notoria de los métodos de entrada y salida en el proceso de interacción invitan a pensar que existe cierto retardo en la ejecución. Es decir, equivalente a que existiera un desfaseamiento entre el movimiento del cursor realizado por el usuario y la visualización en la pantalla. Otro símil que podría dársele sería parecido al juego de mesa “batalla naval”, dónde debes indicar la siguiente ubicación de tu ataque utilizando coordenadas específicas en una zona del tablero.

El concepto anteriormente mencionado sigue una tendencia que fue abordada en la sección de Antecedentes que hace referencia a la estandarización del hardware como método para mejorar el software. Es por esta razón que existe una diferencia tan marcada entre los métodos físicos de interacción con su contraparte de retroalimentación.

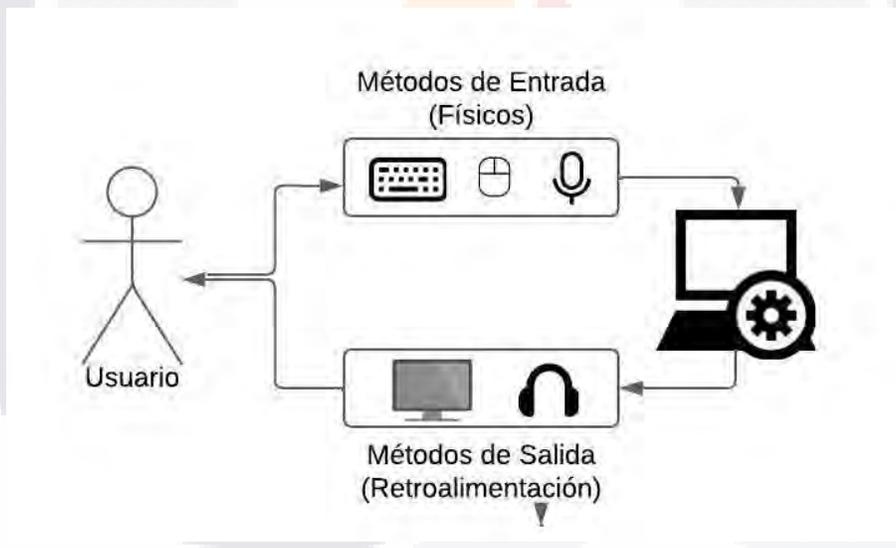


Figura 3: Diagrama de métodos de entrada y salida de una computadora, (Elaboración propia, 2024).

Habiendo mencionado lo anterior, el concepto de interfaz puede definirse cómo los métodos visuales y físicos que intervienen en el proceso de interacción del usuario con el hardware y software (Velasco Mamián & Varón Torres, 2018). Para poder definir el concepto de interfaz en esta investigación y tomando como base la figura 3, en la figura 4 se propone un proceso de interacción

que permite limitar los conceptos de la interacción humano-computadora con el diseño de las interfaces.

Como se empieza a abordar en la idea anterior, la figura 4 se envuelve dentro de un entorno de uso. Este entorno contempla cuatro principales elementos:

- Métodos Físicos: Todo mecanismo que permite introducir información al proceso de interacción.
- Métodos de Retroalimentación: Corresponde a toda salida de datos que muestre el proceso de interacción. El usuario usa esta información en tiempo real para poder planificar su siguiente acción física, acompañada del uso de un método físico.
- Interpretación: Proceso que sincroniza el entendimiento de los estímulos recibidos por el equipo y el uso de los mecanismos físicos que, en conjunto permiten realizar una interacción con un propósito y que se retroalimentan en tiempo real.
- Interacción física: Más que una simple interacción mecánica por parte del usuario en uno de los dispositivos de entrada, en esta investigación se entiende la interacción física como una interacción motivada por la necesidad del usuario para cumplir una de las etapas de del uso de un sistema. Un ejemplo podría entenderse cómo mover el cursor del ratón a un espacio de texto, en esta interacción el usuario identifica la posición inicial del cursor y se orienta en el plano bidimensional del monitor para, de manera física, introducir mediante el ratón el movimiento necesario para completar la tarea.

Dentro de los elementos que se contemplan en el entorno de uso, son los primeros tres los que se encapsulan en la interfaz de usuario. Es responsabilidad del equipo de desarrollo asegurar que el proceso de interacción física sea el adecuado para el usuario.

Una vez definido el concepto de interfaz, es preciso señalar que a día de hoy tanto los mecanismos de entrada físicos y de retroalimentación están en su mayoría estandarizados, por lo que, el proceso de interacción tiende a ser más sencilla para el usuario, sin embargo, es el proceso de interacción física el que hay que tener en cuenta a la hora de planear, diseñar, desarrollar y probar una interfaz, sin embargo, por motivos de esta investigación, cada que se haga referencia al concepto interfaz se refiere al concepto de generar la retroalimentación más adecuada para que el proceso de interacción sea efectivo y agradable.

Mencionado lo anterior, es posible definir el diseño de una interfaz como el estudio, planeación, aplicación de metodologías y diseño de las propiedades de una interfaz que permiten al usuario completar una tarea o resolver un problema (Morejón Labrada, 2020). Anteriormente el diseño de interfaces también contenía diversas propuestas de hardware, sin embargo, esa tendencia no es muy adecuada para procesos de desarrollo de software pequeños, quienes por su parte, necesitan ofrecer soluciones mediante el uso de los productos ya usadas en hogares y oficinas mientras no se disparan los costos de operación (Darejeh & Singh, 2013; Oppermann, 2002).

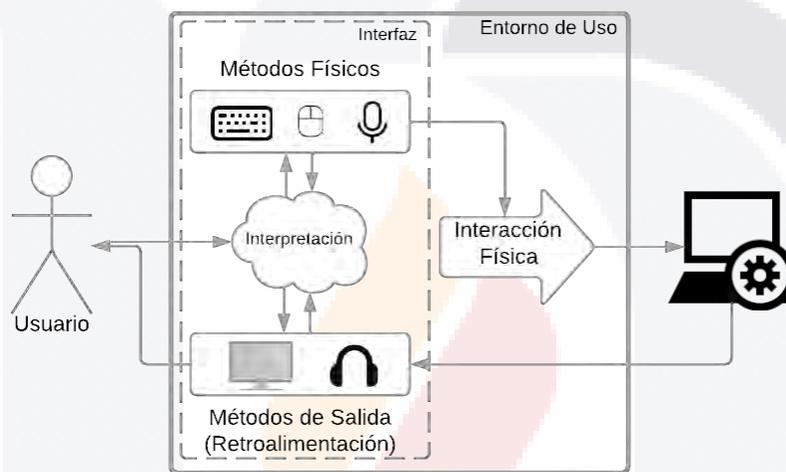


Figura 4: Diagrama propuesto para el proceso de interacción dentro de un entorno de uso, (Elaboración propia, 2024).

A) Elementos de una interfaz

Como se mencionó en la sección anterior, parte importante del diseño de un sistema debe considerar el entorno de uso, por ende, el desarrollador debe buscar en todo momento que el usuario pueda realizar una acción específica o muy general.

Pero surge la pregunta: ¿cómo podemos completar que el usuario realice una tarea por más simple que esta sea? (Morejón Labrada, 2020). Para intentar responder este cuestionamiento se debe tomar en cuenta que el estudio que sería necesario para revisar cada una de las propiedades de lenguajes de programación, sistemas operativos y dispositivos móviles y sus métodos gráficos es un trabajo demasiado extenso, se opta por tomar en cuenta la función general de los elementos que

TESIS TESIS TESIS TESIS TESIS

pueden haber sido implementados para desarrollo web o de lenguajes de programación como Java, Python, C, entre otros.

La “función general” es un concepto que esta investigación implementa para representar el papel de cada uno de los elementos que conforman un diseño. Expresándolo de otra forma, puede verse cómo un mismo elemento que, si bien la codificación difiere drásticamente, la utilización de este y su rol en el diseño son idénticos.

Cumplir de manera cabal los objetivos de un elemento individual es sencillo a nivel de diseño y desarrollo, es decir, evaluar a un elemento como lo puede ser un cuadro de texto o una etiqueta basta con teclear dentro del primero o visualizar la segunda para el aspecto de diseño, mientras que en el aspecto de desarrollo puede resultar más minucioso el procedimiento para verificar si el elemento se comporta de la manera esperada, pero en ambos casos es un procedimiento que requiere poco tiempo y esfuerzo de desarrollo (Paucar Ati, 2017).

La evaluación se complica cuando se tiene que corroborar si dos o más elementos están cumpliendo su labor en conjunto para permitirle al usuario completar su tarea de una manera efectiva, secuencial y acertada.

Ahora bien, la funcionalidad no es el único elemento que interviene en el desempeño de un elemento de interfaz, según (Albornoz, 2014), además de lo anterior, también se debe tomar en cuenta la experiencia previa del usuario, que tan sencillo es para el usuario identificar cada elemento y familiarizarlo para un siguiente encuentro, mantener las sorpresas al mínimo, informar al usuario de errores o validar su información, guiarlo a través de su tarea, entre otros.

El estudio intensivo y extensivo de los conceptos anteriores dejó evidente la necesidad de asegurar en medida de lo posible el siguiente movimiento de un usuario al usar la interfaz (Darejeh & Singh, 2013), por lo que, conforme las interfaces tenían más y más herramientas de diseño se generaron convenciones de uso y acomodo de los elementos a una pantalla. A estas convenciones se les denominó Patrones de Diseño de Interfaces.

B) Patrones de diseño de interfaces

A pesar de que existen diferentes enfoques en los que se puede aplicar un patrón de diseño, se tiende a definir este concepto como el análisis, planeación y verificación de la secuencia de

TESIS TESIS TESIS TESIS TESIS

interacción del usuario con los elementos que los diseñadores implementan en cada estado del uso del sistema. Aún con la estandarización de los métodos de interacción los patrones de diseño no buscan eliminar o sustituir la creatividad inherente al diseño, por lo contrario, son pautas que permiten al diseñador tener un punto de partida esperable sobre el uso del producto por parte del usuario (Blancarte, 2016).

El concepto de patrón de diseño invita a pensar que se insta al diseñador a seguir instrucciones acartonadas y rigurosas de cómo plantear un diseño, sin embargo, existen diferentes tipos de aplicabilidad en los patrones de diseño de interfaces que hacen a los distintos enfoques a plantear y resolver problemas desde diferentes puntos de partida. Un ejemplo de lo anterior y uno de los paradigmas más recurrentes en los patrones de diseño es analizar a detalle las características del equipo en dónde el producto será desplegado y decidir qué métodos se usarán para controlar de manera más directa los estados del proceso de interacción(Blancarte, 2016). Para generar este proceso se recolectan datos de las dimensiones de la pantalla, los métodos de entrada y salida de datos, así como las propiedades y ventajas del sistema operativo del componente. Este estilo de implementación de patrones de desarrollo es especialmente útil para desarrollos amplios, multiplataforma y mayormente en entornos WEB (Cortes-Camarillo et al., 2016).

Otro enfoque es intentar resolver problemas particulares que sufren tipos de usuario específicos. Una de las ventajas de lo anterior es que, si el tipo de usuario está correctamente modelado, será relativamente sencillo generar una secuencia de interacción menos complicada a medida que cada vez se hace más y más específica para el uso de un cierto tipo de usuario (Luna-García et al., 2015).

También existen los patrones de diseño que buscan generar herramientas agradables a proveer información clara, precisa y ordenada en distintos ámbitos. Este tipo de aplicabilidad de los patrones de uso orienta sus métodos para evitar que el usuario se distraiga y retenga su atención al realizar una actividad muy extensa y con cargas de estrés altas. Adicionalmente, se desea que a largo plazo el usuario complete un curso sin perder la motivación e interés a lo largo de varias sesiones (Gros et al., 2016). Los conceptos anteriores definen generalmente los ambientes académicos y más en particular caso las evaluaciones en medios digitales, las cuales son mucho más complejos de realizar que sus contrapartes físicas debido a que en ciertos campos de estudio la lexicografía necesaria del curso y de la evaluación es tan particular que se dificulta o imposibilita representar ideas adecuadamente en formato digital (Martínez, 2017; Sancho-Vinuesa & Pérez-Navarro, 2009).

Sin embargo, existe una peculiaridad en los patrones de diseño, ya que al plantear un uso ideal del producto dónde el usuario es capaz de completar una tarea de manera secuencial, ordenada y efectiva incluso sin haber usado antes el sistema y lo hace de manera perfecta toda vez que use el sistema, por definición, no sería necesarios los patrones de uso, ya que se da por sentado que el usuario responderá exactamente como lo espera el diseñador. Por consiguiente, al estudiar la diversidad de usuarios, contextos y sobre todo los procesos cognitivos que se involucran en la interacción con el equipo, se hace más que evidente el porqué de utilizar los patrones de uso como una primera línea de batalla para intentar converger las interpretaciones de cada usuario en un uso común para todos ellos (Johnson, 2010).

2.2.5 Testing de software

Las pruebas de software en la actualidad son una de las herramientas más importantes para medir la calidad de sistema. No solo ayudan a verificar el cumplimiento de los objetivos si no que pueden ser usados para medir la calidad del producto en casi todos sus apartados. Ahora bien, el simple hecho de someter al software a pruebas de cualquier tipo no mejora inmediatamente el software. Parte vital del apartado de pruebas o Testing de software es conocer los fundamentos con los que han planteado dichas pruebas y aplicarlas en los momentos importantes dentro del proceso de desarrollo para mejorar la calidad del software (Paucar Ati, 2017).

Como se mencionó en el apartado de metodologías de software, el apartado de pruebas es una de las etapas clave de un proceso de desarrollo de software. Sin embargo, esto no quiere decir que se aplique una sola técnica de Testing de software y mucho menos sólo en una ocasión durante el proceso. Si bien algunas metodologías en la actualidad proponen etapas de pruebas delimitadas al final de un proceso de desarrollo para evaluar el cumplimiento de objetivos (Silvis-Cividjian et al., 2021) y resolución de errores, otros enfoques están más encaminados a la división de los aspectos de un software y su correspondiente apartado de pruebas.

A) Pruebas Unitarias y de Integración

La aplicación de pruebas durante el desarrollo de un proyecto no solo influye en el apartado estético también brinda al desarrollador la certeza de que sus esfuerzos están encaminados correctamente para llegar a un producto de calidad.

Tomando en cuenta la teoría de (Sommerville, 2011), existen tres grandes grupos de tipos de pruebas de un sistema: unitarias, de integración y de usabilidad. Cada una cuenta con características que al aplicarlas correctamente convierten un proceso de pruebas caótico y un sistema poco funcional a un sistema resistente a fallas, con mejor calidad de uso y que se ciña a los presupuestos de tiempo, costos y esfuerzos del proceso

Las pruebas unitarias permiten analizar las propiedades de un elemento o un conjunto de estos teniendo como punto en común y de evaluación la funcionalidad que brindan independientemente del sistema (Gómez et al., 2013). Por otro lado, las pruebas de integración ponderan el grado de cohesión que tienen los bloques de elementos establecidos en las pruebas unitarias, es decir, revisan que los contenidos sean congruentes unos con otros independientemente de la funcionalidad de estos. El análisis de integración también revisa los algoritmos que se usan, además de los datos de entrada o salida que generan los bloques de elementos y cómo se están usando a través de toda la aplicación.

Como se puede inferir después de revisar muy grosamente los dos tipos de prueba anteriores, es mucho más sencillo evaluar el funcionamiento del sistema primero analizando las propiedades individuales de los conjuntos de elementos y después revisando cómo interactúan entre ellos.

Este tipo de enfoque está muy dirigido a buscar la eficiencia de los procesos, en otras palabras, el desarrollador es el que evalúa la calidad de un sistema en medida de lo funcional que es. Como se menciona en ideas anteriores, el otro apartado más importante al interactuar con el sistema además de la funcionalidad es la manera en cómo el usuario lo percibe. En muchos casos no basta con la calidad funcional con la que el programador diseña un sistema y para encontrar un punto medio entre estos dos tópicos se usan las pruebas de usabilidad o de uso.

2.3 Eye-tracking

Como se menciona en el apartado de Interacción Humano-Computadora, se ha teorizado durante muchos años y mediante diferentes enfoques las condiciones y contextos que influyen en el uso de un sistema. Es a partir del auge del software de investigación que se analizaron las métricas y herramientas que intervenían en los procesos experimentales (Cartwright & Shepperd, 2000). Rápidamente se hizo evidente que la complejidad de los análisis que podían hacerse a un software es muy limitados o incluso alterados por los periféricos como la pantalla, bocinas o incluso el lugar

TESIS TESIS TESIS TESIS TESIS

dónde se realizaba la prueba. Cuando estos impedimentos fueron corregidos el análisis se centró en el análisis de uso y eficiencia al usar un software (Coleman & Connor, 2008). A partir de este punto, el paradigma de software de investigación abogó por crear experiencias mucho más complejas, generando emociones o reacciones que usualmente no se ligaban a la interacción humano computadora.

Si bien el campo de la biomédica fue pionero en el aprovechamiento de microsensores que pueden registrar señales biométricas del cuerpo humano, las posibilidades de este análisis emigraron rápidamente a campos de análisis de frecuencias cognitivas (Gramfort et al., 2014). El análisis e inferencia de los datos brindados por estas herramientas siguen en discusión en la actualidad y tanto para las instituciones académicas o industrias de marketing es de un gran valor el entendimiento y aplicación de estas nuevas tecnologías.

En la investigación de (Gomolka et al., 2020) se especifica que la vista es el sentido dominante del ser humano. La cantidad y complejidad de la información obtenida por parte de los mecanismos de observación, interpretación y reacción que el sistema ocular envía al cerebro son una fuente de información bastante valiosa para el entendimiento de cualquier estímulo presentado. Gracias a las técnicas de seguimiento ocular es posible analizar los procesos visuales anteriormente mencionados y sus reacciones entre usuarios experimentados y no experimentados al usar un sistema, por mencionar uno de los enfoques solamente.

Si bien la aplicación de las técnicas de eye-tracking combinadas con el análisis de datos es un enfoque reciente, el estudio de los movimientos oculares documentado data de finales del siglo XIX (Chaplin & Bradford, 2011). Incluso anteriormente análisis anatómicos ya habían dado un primer paso en el entendimiento de la vista en humanos y otros animales. A pesar de la antigüedad de las investigaciones sobre la visión humana, es hasta la década de 1950 que se diseñaron tecnologías no invasivas para el participante, una de ellas fue la electrooculografía que, como su nombre lo indica, capta las pequeñas variaciones eléctricas de los músculos faciales alrededor de las cuencas oculares (Duchowski, 2017). Décadas después, alrededor de los años ochenta, surgió un nuevo paradigma denominado “reflejo de Púpila/Córnea combinado con Base de Video”. El término Base de Video hace referencia a que el sujeto apoya su cabeza sobre una plataforma en la cual, mediante cámaras fijas, graban a corta distancia los ojos estudiando las reacciones a estímulos. Es para la segunda mitad de la década de los noventa que se empiezan a diseñar hardwares específicos para este fin.

Abriendo el camino a que muchas más investigaciones pudieran aplicar la tecnología en diferentes campos y con diferentes enfoques.

Ahora bien, los estudios realizados mediante técnicas de eye-tracking han sido influenciados por las teorías y conceptos que se han estipulado durante el siglo XX referentes a la atención visual. La relación del sistema de la vista con la actividad neuronal es ampliamente aceptada desde hace muchos años (Duchowski, 2017), durante los últimos cincuenta años se ha buscado detallar los procesos cognitivos específicos que ven, analizan y reaccionan ante los estímulos del día a día.

Aunque la rama de la neurología ha realizado muchas de investigaciones y teorías para entender al cerebro humano, para el campo de la biotecnología, biometría incluso ciencia de datos es un campo prácticamente nuevo y poco explorado.

Uno de los primeros retos que se presentó a la hora de estudiar los datos obtenidos mediante herramientas de seguimiento de mirada fue cómo dividir las etapas de la atención humana. Una vez que se conjeturó que la atención visual está dada de forma general por los factores de fijación y movimiento sacádico es que se pudo dar un primer análisis de bajo nivel sobre la mirada de un ser humano. En el apartado siguiente se abordan de manera más amplia estos conceptos.

2.3.1 Visión humana.

Antes de profundizar en los movimientos del ojo humano es preciso repasar levemente la fisionomía de este órgano tan importante para el ser humano. No se abordará extensamente este tópico dado que la gran mayoría de librerías, sistemas o hardwares de eye-tracking ya aplican directamente los estándares establecidos al investigar minuciosamente las características de la visión humana en las métricas y funciones que brindan.

La visión humana es uno de los mecanismos más complejos dado que los movimientos del órgano son, además del más rápido del cuerpo humano, los que más costo cognitivo significa para la persona (Ruiz-Contreras & Cansino, 2005). En otras palabras, el cerebro humano usa buena parte de su capacidad para analizar más y de mejor manera los estímulos visuales que el resto de los estímulos sensoriales del cuerpo. Los estímulos, que le llegan en el orden de milisegundos, además de ser captados y procesados el cerebro en la mayoría de los casos se envía información por todo el cuerpo para reaccionar ante estos. Una de las primeras partes que entra en funcionamiento es el iris que regula la entrada de luz al contraerse o dilatarse, este movimiento también afecta a la pupila

TESIS TESIS TESIS TESIS TESIS

quién envía la imagen invertida a la parte posterior del ojo (para las siguientes explicaciones de las partes del ojo humano consúltese la figura 5).

En la parte posterior del ojo se encuentran la retina y la fovea. La primera utiliza receptores neurales conectados al nervio óptico que invierten la imagen generada por la pupila. La segunda es la encargada de percibir los colores de la imagen (Chaplin & Bradford, 2011; Duchowski, 2017).

Cabe destacar que existen dos ejes importantes para la visión: El óptico y el visual. El primero cambia dependiendo de los músculos del globo ocular, mientras que la pupila hace de filtro para reducir o ampliar el eje visual. Expresado de otra manera, el cruce entre estos dos ejes en la parte posterior del globo y el ángulo que generan es la cantidad de información que tanto la retina como la fovea reciben. Entre menor sea el diámetro de la pupila al observar un punto de atención específico la cantidad de información enviada a través del nervio óptico será más detallada y precisa, a costa de reducir la visión periférica e ignorar automáticamente algunos otros elementos del campo de visión (Duchowski, 2017).

El concepto anterior puede entenderse cómo los parámetros óptimos para analizar un campo de visión, dado que si se tiene información detallada de cada elemento se puede reaccionar de mejor manera a los estímulos, sin embargo, se debe recordar que la “captura” de la imagen está en orden de los milisegundos. Para el cerebro humano la necesidad de reaccionar lo más rápido y mejor posible no es opcional, durante la evolución de las capacidades cognitivas del hombre el cerebro tenía que buscar maneras de sobrevivir, comunicarse y aprender habilidades, por lo que el constante trabajo entre la visión humana y el cerebro fueron optimizándose para permitir tener una coordinación tal que le permitiera imponerse sobre sus depredadores y presas.

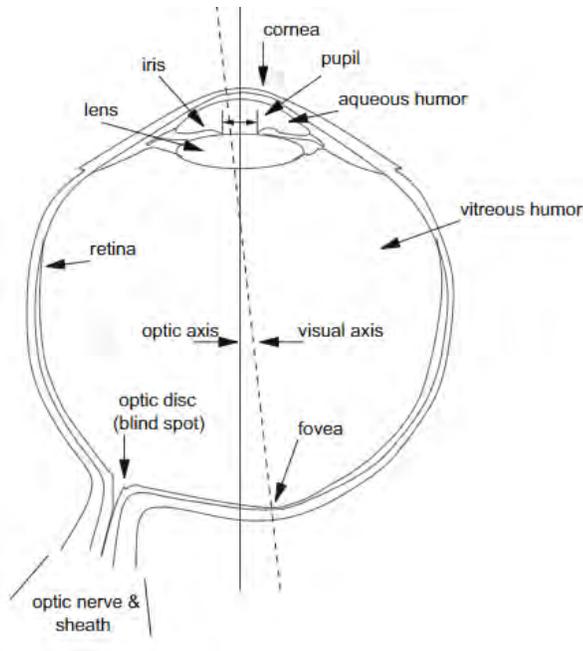


Figura 5: Ojo humano, (Duchowski, 2017).

2.3.2 Proceso de atención visual

Cómo se mencionó en el apartado anterior las reacciones cognitivas, sensoriales y visuales, entre otros elementos que se activan en el cuerpo humano estaban orientados para la supervivencia. Reaccionar o no ante un peligro rápidamente podía significar la diferencia entre la vida y la muerte. Por tales razones, tanto el ojo humano como el cerebro desarrollaron mecanismos que permitieran obtener información de manera rápida y detallada sobre un punto focal específico mientras que la información se completaba por la visión periférica y otras herramientas como lo son la conciencia espacial, estimación de las distancias y la interpretación de los colores (Nathans, 1999).

A pesar de las investigaciones que se han realizado hasta la fecha, las características de la visión humana, sus peculiaridades y sus métodos siguen descubriéndose actualmente. Una de las convenciones más divulgadas en los últimos años es que la visión hace uso de diversos procesos para entender una imagen que se le presenta y el más relevante de ellos es el filtrado de la información (Evans et al., 2011).

El proceso del filtrado de la información se puede entender como todo el recorrido que realiza ojo para analizar un estímulo apuntando su eje óptico varias direcciones, apoyándose de los músculos oculares y la rotación de la cabeza para tener una imagen amplia de la escena. Este análisis visual

está poco enfocado en buscar puntos muy altamente reconocibles, como una cara humana, un objeto inanimado o incluso una representación gráfica de algo dentro de nuestras experiencias previas. El nombre que se le otorga al efecto anterior, es decir, a la búsqueda de experiencias o conocimientos previos por parte del cerebro para usarlos como base al entendimiento del estímulo expuesto se le denomina memoria esquemática o de largo plazo (Andrade-Lotero, 2012; Gutiérrez de Piñeres Botero, 2019).

Retomando el proceso de atención visual, una vez que se asocia un estímulo a una idea anteriormente experimentada, el ojo vira en búsqueda de los puntos que le permitan confirmar su teoría. Usando uno de los ejemplos anteriores, el sujeto al ver una cara y reconocerla como tal, pasará a ver los puntos que le permitan confirmar su primera idea del estímulo cómo lo serían los ojos, nariz o boca de la otra persona. A este proceso de focalización en puntos específicos para recolectar información relevante para reaccionar y entender el estímulo es denominada fijación (Drusch et al., 2011; Mondal et al., 2022).

Una fijación puede definirse como un punto de la atención visual que permanece fijo en el tiempo hasta que considera que ha reconocido al elemento o, en su defecto, necesita recurrir a otro punto para completar o ampliar la información. Al movimiento entre un punto de fijación y otro se le denomina movimiento sacádico. Recordando levemente la anatomía del ojo humano (figura 5), la fijación hace uso del iris y la pupila para enfocar en una zona específica, permitiendo al nervio óptico tener más receptores libres para poca información; Complementariamente, los movimientos sacádicos hacen uso de los músculos oculares para pivotar el eje óptico y ampliar la imagen a costo del detalle del análisis (Duchowski, 2017; Nathans, 1999; Ruiz-Contreras & Cansino, 2005).

Los movimientos sacádicos pueden variar dependiendo de diversos factores, desde el orden de los elementos que debe visitar para lograr entender la información completa, colores, movimientos, logos o advertencias son características que pueden modificar el recorrido visual ante un estímulo. El proceso de atención visual continúa alternando entre fijaciones y movimientos sacádicos hasta que se completa todo el análisis de información que la persona considera suficiente para reaccionar, entender o reconocer algún elemento que le fue presentado (Sharafi et al., 2016).

A pesar de que el proceso de atención visual parece estar optimizado, no todos los estímulos resultan ser tan fáciles de entender que otros, o se les pueda asignar mucho tiempo. La mente humana, a pesar de su increíble capacidad tiene sus pequeños trucos bajo la manga para poder

reaccionar rápidamente a los estímulos. En el siguiente apartado se exponen las convenciones de los métodos de la mente humana para analizar la información visual.

2.3.3 Intervención de métodos cognitivos en el proceso de atención visual

Cómo se empieza a abordar en la sección anterior, parte importante de la atención visual es la etapa del procesamiento de la información. Para desarrollar este tema de manera más precisa se debe tomar en cuenta que el enfoque de esta investigación se limita a estudiar al comportamiento de la mirada en una interfaz o sistemas específicos.

Hoy en día hay muchas experiencias inmersivas que logran generarle al usuario una sensación de miedo, vértigo o una percepción de aceleración, sin embargo, estos aspectos están más relacionados al campo de la gamificación. Dicho lo anterior, centraremos el análisis de la memoria de trabajo y esquemática únicamente con su relación a la interpretación visual de elementos digitales, dejando de lado la retroalimentación sensorial del cuerpo humano o la aplicación de otros métodos como la percepción de profundidad que es parte importante para que el sujeto comprenda su entorno.

Mencionado lo anterior, se procede a explicar los conceptos relacionados a cómo la memoria humana es capaz de reconocer objetos y actuar en consecuencia de ellos. El primer punto es la recolección de información, recordando el proceso de atención visual existe una etapa de reconocimiento inicial, se genera un barrido general que brinda poca información detallada. En esta etapa actúa directamente la memoria de trabajo almacenando por cortos periodos de tiempo esa información. La memoria de trabajo es un mecanismo que ayuda a dar una primera estructura de la información, analiza distancias entre elementos, colores y formas básicas, hay que tener en cuenta que hasta que se realiza una fijación es que se obtiene información detallada del elemento que se observa, mientras tanto la memoria de trabajo sigue haciéndose una idea general del contenido (Ustároz & Grandi, 2016). La memoria de trabajo o de corto plazo es llamada así por su tarea de congregar toda la información recabada por los transmisores y sentidos del cuerpo y ponerla a disposición de la memoria esquemática.

La memoria esquemática es toda información almacenada y organizada en el cerebro lista para ser usada en el momento que la memoria de trabajo la necesite (Blake & Lee, 2005). De manera neurológica, la memoria esquemática o a largo plazo son un conjunto de pulsos electrónicos que

TESIS TESIS TESIS TESIS TESIS

recorren el cerebro por rutas específicas (Sakamoto & Love, 2004). Lo que comúnmente entendemos como aprendizaje es la capacidad con la que la memoria esquemática logra recorrer las rutas que están destinadas en reconocer o recordar la información. Ejemplificando lo anterior, cuando un guitarrista recién está aprendiendo una composición las rutas son poco claras, pues estas se abren camino buscando las partes del cerebro que manejan la memoria muscular, otras del oído, etcétera. Entre más veces el guitarrista practique la composición el cerebro encontrará la ruta que necesitó para comprender y ejecutar la obra de manera más automática y rápida. Este proceso se extiende para cada actividad del ser humano, desde el habla, pasando por caminar o reconocer olores y sabores, hasta leer una obra compleja o resolver un problema matemático. La experiencia previa es el mecanismo más usado del cerebro para lograr reaccionar a un estímulo (Hess & Slaughter, 1990; Sakamoto & Love, 2004).

Como se puede notar después de repasar los conceptos de memoria a corto y largo plazo es evidente que ambas requieren de la otra para poder funcionar. A pesar de que ambas logran reaccionar a estímulos que pueden ser muy rápidos, la memoria humana tiene sus limitantes. De manera general se le denomina carga cognitiva a la capacidad de información que la memoria, en sus fases a corto y largo plazo, puede manejar. La carga cognitiva es un mecanismo para evitar la saturación de información que procesa el cerebro, dentro de sus herramientas más utilizadas es el filtro de elementos en la primera etapa de inspección del estímulo (Tarouco, 2006). De manera automática, y de nuevo mediante las experiencias previas, el ojo humano en su recorrido pondera cada uno de los elementos y los cataloga en un rango desde interesante para volverlo a visitar, hasta inútil para obtener información de valor que le permita el estímulo o la pantalla en el caso de esta investigación (Andrade-Lotero, 2012; Tarouco, 2006).

Esta valuación de los elementos es muy subjetiva para el análisis de interfaces, pues la hipótesis más natural es que un usuario experimentado puede sobreentender de manera más sencilla una aplicación, es decir, puede dar por sentado que el botón con una gran equis puede significar “salir” cuándo es un ícono de fallo en el sistema. Se sobrentiende puede variar las experiencias del sistema o el agrado que el usuario percibe.

2.3.4 Hardware y Software de eye-tracking en la actualidad

En las secciones anteriores se abordaron los procesos que componen el proceso de atención visual y su relación con algunos mecanismos de la mente humana. Si bien estos tópicos han sido revisados ampliamente, el campo de la recolección de datos biométricos sigue en una etapa temprana todavía.

Al inicio de la sección de 2.3 referente al eye-tracking se hizo referencia a los diferentes enfoques que se utilizaron durante la segunda mitad del siglo. Es partir de las fundamentaciones teóricas y algunos experimentos que empresas vieron la oportunidad de mercado para desarrollar tecnologías de seguimiento de mirada que muchas organizaciones pudieran utilizar para diversos fines.

A) Eye-tracker en el mercado y características relevantes

Más allá de empresas como *Tobii Technology* (de la cual se especificarán dos productos que fueron usados en partes de pruebas de uso en esta investigación) es relevante mencionar otros enfoques sin ánimos de lucro que fueron en su momento, y son aún, partes muy importantes del estado del arte del eye-tracking. Uno de los casos de investigación más relevantes es la librería de Python PyGaze desarrollado por el Dr. Edwin Dalmaijer. PyGaze está enfocado en métricas e investigación, por lo que en ambientes académicos es uno de los pioneros en la transformación de datos biométricos crudos en métricas más interpretables para los estudios. Hoy en día (2024) el progreso de la investigación parece pausado indefinidamente, sin embargo, las bases que estableció para el análisis de los resultados de eye-tracking y su aplicación en investigaciones fue muy relevante para estos últimos cinco años.

Entrando en materia de *Tobii Technologys*, esta organización con fundación y sede en Suecia se ha dedicado desde 2001 ha desarrollar tecnologías relacionadas a la visión humana y su estudio. Para enfocar la información se abordarán únicamente los productos que tienen un impacto directo en la investigación. Uno de estos productos es el software de *TobiiProLab*, en esta aplicación de escritorio es posible diseñar, aplicar y analizar pruebas experimentales en las que se desee involucrar el registro de seguimiento de mirada. El software está pensando para el investigador, ya que brinda herramientas como la creación de participantes, exposiciones controladas de estímulos, selección de áreas y eventos de interés. También posee métodos de exportación de métricas y del registro en crudo para permitir al usuario aplicar técnicas de ciencias de datos o visualizaciones. Si bien existen

métodos para simular una experiencia de un usuario al exponerlo a un estímulo, la verdadera potencia de *TobiiProLab* se consigue al usar alguno de los eye-trackers que ofrece la misma compañía.

Existen muchos tipos de eye-tracker incluso dentro de la misma compañía de *Tobii Technology*. Algunos están enfocados para escenarios más amplios como lo podrían ser la conducción o en marketing, un ejemplo de estos son los *TobiiProGlasses*. Por otro lado, el *TobiiProSpectrum* y *TobiiProFusion* están enfocados en el análisis de movimientos micro-sacádicos, por lo que está pensando para el campo biomédico o para pruebas en exteriores con condiciones variables de entorno. Por último, el *TobiiProNano* está desarrollado para pruebas de laboratorio, es muy portátil, fácil de instalar y usar. Otra de las ventajas de este último eye-tracker mencionado es que puede ser usado en la parte inferior cualquier pantalla, siendo esta característica lo que lo hace ideal para realizar pruebas de uso de sistemas, diseños de interfaces, test psicológicos, entre otros.

A diferencia de las *TobiiProGlasses* y *TobiiProSpectrum*, la mayoría de eye-trackers de la compañía usan tecnología de infrarrojo para estimar tanto la distancia del usuario, cómo los movimientos del párpado y cuenca ocular. Posteriormente utiliza las posiciones del iris, así como la amplitud de la pupila, para estimar el foco de atención.

B) Hardware y software de eye-tracking usado en la investigación.

El *TobiiProNano* es una herramienta compacta pensada para analizar estímulos reproducidos en pantallas mientras se usa un equipo de cómputo de escritorio o laptop portátil, por las características compactas de este dispositivo es posible adherirse cualquier pantalla o monitor (véase figura 6).



Figura 6: Eye-tracker TobiiProNano, (Sitio Oficial de TobiiTechnologys, 2023).

Para complementar el análisis, el software de TobiiProLab permite generar análisis y pruebas experimentales para diversos estímulos de eye-tracking. Este software contiene secciones para el control de pruebas, diseño de exposición de estímulos al usuario, áreas de interés, ventos, reproducción de grabaciones, visualizaciones de los resultados, una herramienta de generación de *datasets* muy potente, entre otras funcionalidades, que lo hacen una herramienta muy valiosa para que sirva como base experimental de diferentes propuestas que se abordan en este trabajo de investigación.



3. Metodología de la Investigación

En esta sección se desarrollan varias propuestas que, primeramente, son abordadas desde la perspectiva de la ingeniería de software, y de manera adicional y de lo general a lo particular, se van introduciendo los conceptos de eye-tracking en el desarrollo, diseño y evaluación de interfaces.

Se debe tomar en cuenta en todo momento que este trabajo de investigación, además de los objetivos propios de este, es compartir el uso de la tecnología de eye-tracking, ya que, como se aborda brevemente los pocos trabajos en los que se implementan estas técnicas sus resultados no son viables de replicar ya que, en caso de hacerlo, se tendría que duplicar en realizar el caso de investigación idéntico de la fuente, por lo que, al analizar los datos, lo máximo que se puede aspirar de estos es repetirlos, en otras palabras, existen muy pocas estandarizaciones o guías de implementación de estas tecnologías que puedan ser usadas en múltiples procesos de desarrollo o experimentación.

La idea anterior, desde el punto de vista de este trabajo, se considera además de un área de oportunidad gigantesca, se considera un paso adelante en el entendimiento no solo de la tecnología de seguimiento de mirada, sino también de los procesos cognitivos del ser humano.

Ampliando lo anterior, compartir los resultados de los descubrimientos y técnicas que mejor se adapten es una de las tareas principales de este trabajo, por lo que, si bien muchas de las propuestas puedan necesitar modificaciones para poder ser implementadas en otros procesos de desarrollo de software, se busca definir un mapa de ruta para los que busquen implementar esta tecnología a sus proyectos.

Pero antes de generar las propuestas, se describe a continuación, las características que dan forma y rigurosidad a la investigación comenzando por abordar la metodología de la investigación, posteriormente se seleccionan y especifican las metodologías más adecuadas para implementar sobre ellas las técnicas de eye-tracking. Para completar lo anterior, se establece una sección referente a una metodología propuesta de diseño de interfaces basada totalmente en las técnicas de eye-tracking y, una vez, en este punto, se aborda ampliamente las propuestas metodológicas más relevantes de este trabajo.

3.1 Metodología de experimentación

Los procesos de desarrollo de software son muy diferentes el uno del otro, aunque usen la misma metodología o incluso sean las mismas personas que lo llevan a cabo. Sin embargo, hay que tener en cuenta que el objetivo final de todos y cada uno de ellos es resolver un problema para el usuario. Tomando en cuenta este enfoque queda claro que se debe orientar todo el esfuerzo en lograr ofrecer una herramienta a la persona que usará el producto y no solo una plataforma.

Si se deseara limitar esta investigación al diseño de interfaces ya se estaría innovando al usar técnicas de eye-tracking para este fin, pero tomando el punto anterior, existirían casos dónde se aplicarían las técnicas y los procesos planteados y de todas formas no sería exitoso. Uno de los más grandes retos de la industria de software es encontrar la manera de generalizar una solución para todos los desarrollos. De ser posible cumplir esta premisa no sería necesaria una investigación sobre los procesos de desarrollo de software, pero al no poder generar una solución única es que se deben estudiar y proponer marcos de trabajo que mejoren las posibilidades de éxito de estos.

Ya se mencionó que el análisis de interfaces mediante eye-tracking por sí solo puede aportar a despejar las dudas de diseñadores y desarrollares sobre la utilidad, agrado y usabilidad de sus diseños de interfaces. Sin embargo, estos dos roles no son los únicos que intervienen en el proceso. Para poder analizar diferentes escenarios es preciso generar varios casos de estudio que estén planteados tanto por diferentes personas como con diferentes enfoques. Los casos de uso deben estar pensados para simular alguna característica que muchos de los sistemas actuales estén usando ampliamente. Recordemos que la aplicación de eye-tracking debe ayudar a buscar diseños optimizados con cada iteración, por lo que al seleccionar casos de uso relativamente parecidos se está promoviendo que el análisis de las métricas de eye-tracking sean bastante precisas y detalladas.

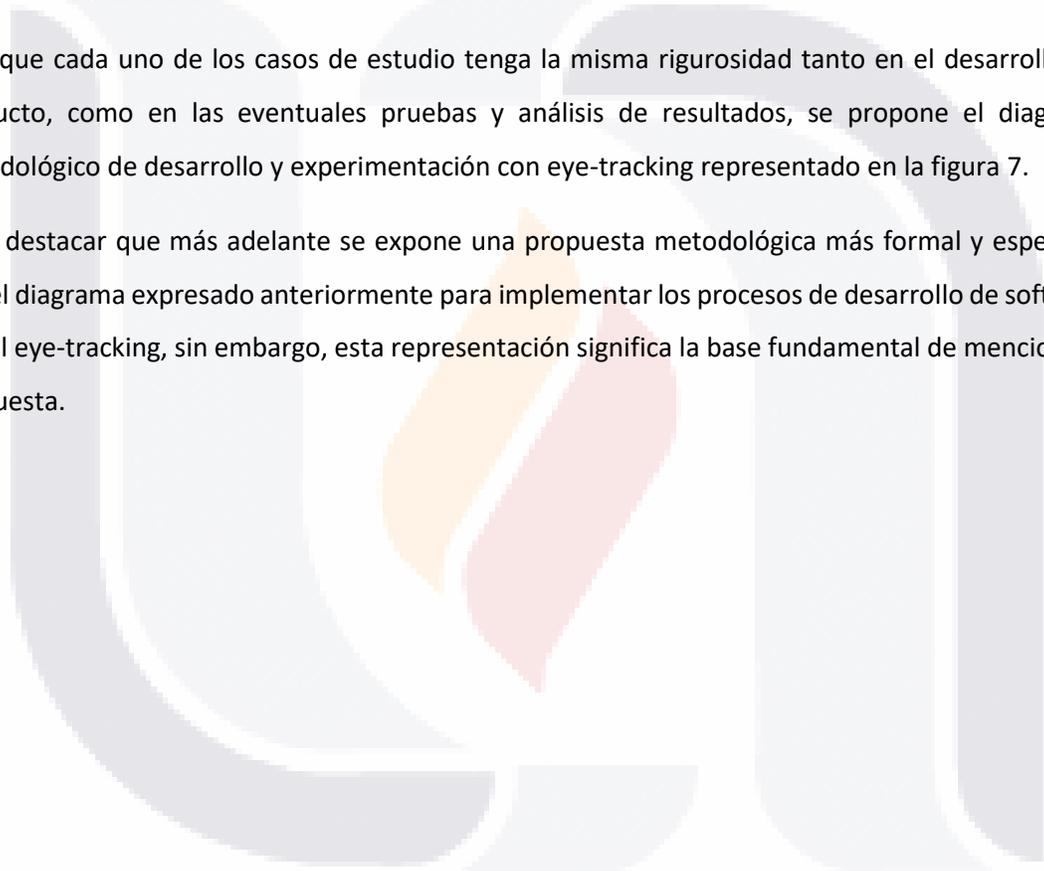
Una de las palabras clave del párrafo anterior es la palabra “iterativo”. Esta especificación permite acotar la cantidad y variedad de metodologías de desarrollo de software que se han propuesto hasta el día de hoy. La capacidad de un proceso de desarrollo de ser iterativo dota al producto de más etapas de evaluación que, al menos en el particular caso de esta investigación resultan muy valiosas para aplicar en estas etapas el análisis mediante eye-tracking. Una vez dentro de las metodologías de desarrollo iterativas se pueden dividir en dos grandes enfoques, metodologías ágiles y tradicionales. Las primeras están casi en su totalidad enfocadas a separar los procesos en tiempos muy definidos y aplicar en cada una de sus iteraciones al menos una evaluación del estado actual

del software, algunas veces por miembros del equipo de desarrollo y otras por tester's o usuarios de prueba. Las segundas varían mucho más su tiempo, más no sus etapas, que están secuenciadas de manera más estática.

Habiendo mencionado todo lo anterior se proponen dos desarrollos, uno aplicando una metodología tradicional en espiral y otro mediante la metodología ágil enfocada en Sprint's. Deben estar implementadas de manera que se pueda identificar con certeza la etapa en la que se aplican las pruebas de eye-tracking y, posteriormente, analizar el beneficio que tuvieron ambos desarrollos al aplicar las pruebas de usabilidad objetivas.

Para que cada uno de los casos de estudio tenga la misma rigurosidad tanto en el desarrollo del producto, como en las eventuales pruebas y análisis de resultados, se propone el diagrama metodológico de desarrollo y experimentación con eye-tracking representado en la figura 7.

Cabe destacar que más adelante se expone una propuesta metodológica más formal y específica que el diagrama expresado anteriormente para implementar los procesos de desarrollo de software con el eye-tracking, sin embargo, esta representación significa la base fundamental de mencionada propuesta.



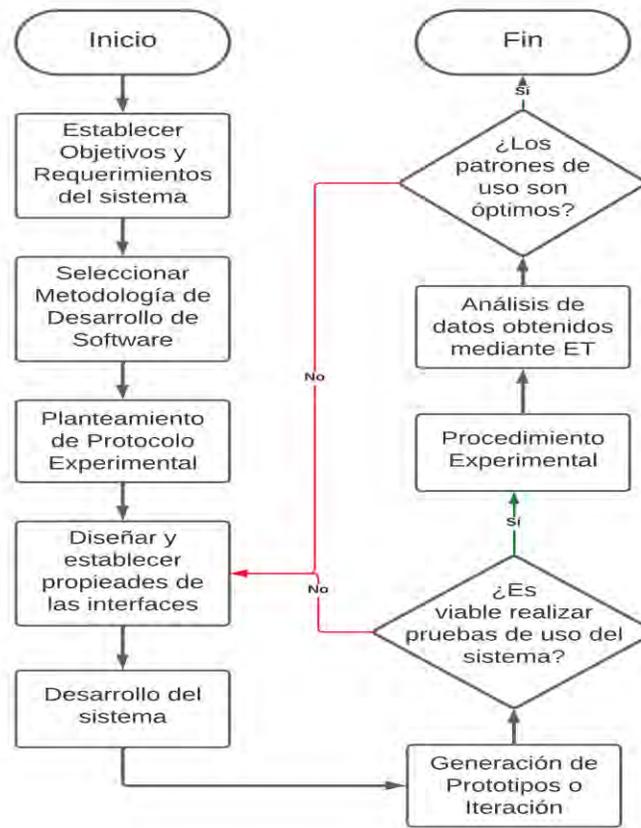


Figura 7: Diagrama de desarrollo de software y evaluación mediante eye-tracking, (Elaboración propia, 2023).

3.2 Metodología de desarrollo de software tradicional propuesta.

En la sección de fundamentación teórica se abordan las características de las metodologías en cascada y en espiral a detalle. En esta sección se plantea generar una versión diferente de las convenciones de la metodología para aumentar la preponderancia de la etapa de pruebas de uso mediante eye-tracking. El modelo iterativo cuenta con etapas definidas concretamente, sin embargo, cada proceso de desarrollo, dependiendo de los objetivos que se establecieron se le otorga más o menos relevancia a ciertas etapas.

Algunas convenciones señalan que la etapa más relevante es la etapa de requerimientos, pues es en este momento que se toman las decisiones más determinantes para el producto final, sin embargo, al seguir este enfoque muchas de las ventajas de un modelo en espiral se pierden

levemente. Recordando que el modelo en espiral está pensando para ser flexible a cambios, resulta poco efectivo pensar en sobre medida en la planeación del proyecto entero.

Contraria a la idea anterior, una convención ha manejado que la etapa de evaluación debe ser la más importante a la hora de desarrollar un producto por prototipos. Si bien la metodología en espiral es catalogada como tradicional, cuenta con varios conceptos que posteriormente fueron fundamentales para las metodologías ágiles. Entre esos conceptos se encuentra que los diseños propuestos y su calidad están en función de las evaluaciones que se les realicen. Expresado de otra manera, en lugar de proponer diseños “estáticos” se busca anticiparse a las evaluaciones orientando y escogiendo los elementos de la pantalla que mejoren tanto el uso, utilidad y agrado por parte del usuario.

En la figura 8 es posible observar las etapas de la metodología tradicional propuesta haciendo énfasis en la etapa de evaluación mediante eye-tracking. Cabe destacar que las etapas que preceden y anteceden a la evaluación de eye-tracking son fases que permiten analizar el punto actual del software y, en medida de los objetivos, generar correcciones necesarias para mejorar las posibilidades de éxito del proceso. Otra especificación que es preciso realizar sobre la metodología es la etapa de levantamiento de requerimientos. De manera convencional, los requerimientos son una de las primeras formas de darle un entendimiento a los procesos del sistema, sin embargo, en este enfoque se busca que los requerimientos funjan como un conjunto de nuevos objetivos o pautas claras de la iteración. Si bien conforme el proceso se itere la cantidad de requerimientos disminuya, la especificidad de estos irá aumentando.



Figura 8: Metodología de desarrollo en espiral-iterativa, (Elaboración propia, 2024.)

3.3 Metodología de desarrollo de software ágil propuesta

Como se aborda en la revisión de los conceptos más relevantes de esta investigación, las metodologías ágiles permiten procesos de desarrollo de software con mayor control de los avances a corto plazo y promueven evaluaciones en periodos mucho más cortos de tiempo.

Este tipo de procesos requieren una principal retroalimentación entre el líder del proyecto con el grupo de desarrolladores, ya que, generalmente, implementar una metodología ágil mejora sus resultados al separar diversas funciones en roles de trabajo específicos para cada miembro del equipo.

En un desarrollo en el que el equipo de trabajo es reducido implementar una metodología ágil puede llegar a ser contraproducente debido a que al evaluar de forma tan interna, se puede caer en sesgos de diseño y, en general, menos ideas o contrapuntos para mejorar el producto, sin embargo, en un proceso de desarrollo que cuente con un equipo multidisciplinario o con diferentes especialistas en diferentes ramas del diseño, desarrollo, implementación y evaluación implementar una metodología ágil es el camino a seguir recomendado.

Si se define una metodología ágil todo aquel proceso en el que cada etapa del desarrollo de un producto está ligada a dos principales elementos que son los periodos de tiempos limitados para presentar avances (*sprint's*) y la retroalimentación constante tanto del líder del equipo como del resto del equipo de desarrollo, es preciso determinar en específico que etapas de del desarrollo queremos añadir a nuestra secuencia de producción. Como se observa en la figura 9, existe una completa relación entre los procesos de desarrollo delimitados con los *sprint's* a las etapas que conforman este, y a su vez, las etapas de retroalimentación, si bien están gobernadas por las retroalimentaciones e indicaciones del líder de proyecto, es mediante las reuniones del equipo de trabajo que realizan cada etapa del software.

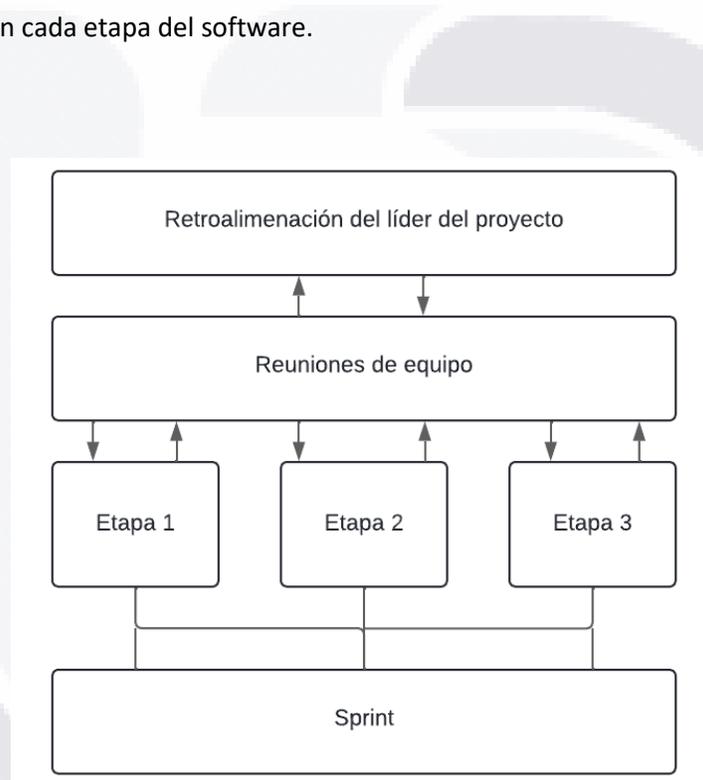


Figura 9: Diagrama de la metodología ágil de desarrollo de software, (Elaboración propia, 2024).

A pesar de que de manera teórica cada etapa genera aspectos diversos del producto de software, en mucho de los casos, y tomando como bases técnicas de metodologías tradicionales, la etapa de análisis y diseño puede tener más relevancia en etapas temprana de proceso de producción, por lo contrario, resultaría complicado generar pruebas de uso de una interfaz si el proceso de desarrollo de estas no se ha completado hasta un punto que sea prudente evaluar las capacidades de las interfaces. Es por esto por lo que, como se indica en la figura 10, una secuencia de *sprints* puede

contener las mismas o diferentes etapas de desarrollo, establecer estas etapas es una de las tareas más importantes del líder del proyecto, ya que debe de permitir congregar distintos procesos de desarrollo de software en los que, además de tener una secuencia lógica, planteen tempranamente las posibles evaluaciones del producto, o el despliegue, o manejo de errores, etc.

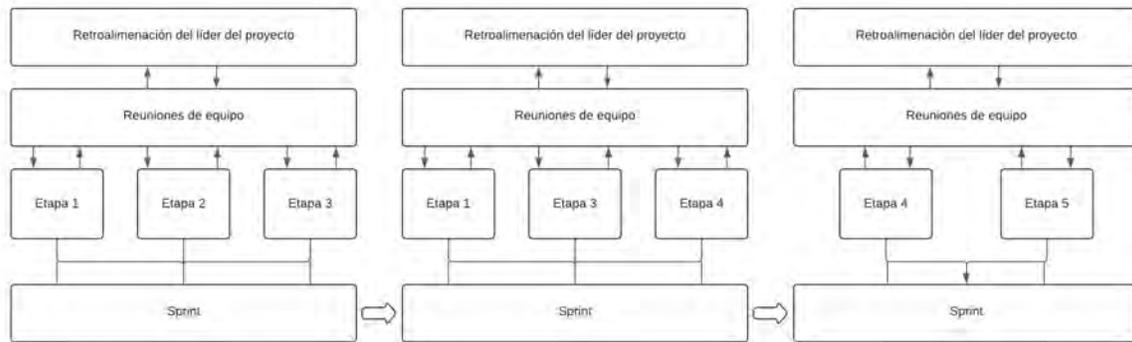


Figura 10: Metodología ágil de desarrollo de software con varios sprint's, (Elaboración propia, 2024).

3.4 Metodología de diseño de interfaces: Interfaces paralelas

Cómo se ha abordado en el apartado teórico del diseño de interfaces, existen muchos enfoques posibles para proponer los elementos que la conforman y establecer la distribución, dimensión y colores de estos. Retomando brevemente los enfoques más comunes de las metodologías tradicionales y ágiles, se encuentra que las primeras son muy rígidas a la hora de proponer diseños de interfaces, ya que se tiende a limitar cada diseño a un caso de uso por separado. Si bien las metodologías ágiles se han centrado en ser flexibles en la mayoría de los apartados, los desarrolladores tienden a estancarse en una idea preconcebida de un diseño (Varakin et al., 2004).

Para la correcta evaluación de una interfaz se deben contemplar varios aspectos que cambian altamente los resultados de las pruebas de uso, sin embargo, existen escenarios donde el desarrollo de una interfaz puede aparentar a vista de los diseñadores que se retrocede en lugar que se avanza. En estos casos puede estar fallando el método para obtener información por parte de los usuarios, los objetivos de diseño no son claros, los elementos no son una solución para el usuario, entre muchos otros factores.

Con el fin de potenciar las capacidades de los diseñadores para evaluar el punto actual de su diseño se propone el concepto de generar dos propuestas de diseño para resolver un mismo objetivo en particular. Debe hacerse la precisión de que este concepto puede tener complicaciones para adaptarse a interfaces ya funcionales y con contenidos muy diversos. El concepto se potencializa en etapas tempranas del desarrollo cuándo en cualquier tipo de metodología convencional o ágil se obtienen ventajas de una correcta planeación del proceso.

En secciones posteriores se abordarán los conceptos de análisis de alto y bajo nivel de una interfaz, sin embargo, las evaluaciones que pueden realizarse sobre este tipo de diseño de interfaces son del segundo tipo, es decir, de bajo nivel. Esto hace referencia a que la evolución del diseño de las propuestas paralelas está en función de la distribución, dimensión, colores y fuentes de cada uno de los elementos del par de propuestas.

Para proponer diseños paralelos de interfaces en un proceso de desarrollo se aplican las las siguientes condiciones (véase figuras 11 y 12):

- Que la interfaz esté pensada para cumplir una única tarea.
- Que las interfaces paralelas cumplan la misma tarea.
- Que las interfaces cuenten con los mismos elementos, pero diferentes características en distribución, dimensión, colores y fuentes.



Figura 11: Diseño hipotético de interfaz A, (Elaboración propia, 2023).



Figura 12: Diseño Hipotético de Interfaz B, (Elaboración propia, 2023).

Gracias a la capacidad de este tipo de diseño de interfaces es que se puede experimentar fácilmente con las condiciones fundamentales del uso de estas, desde los cuadros de texto, botones de confirmación, imágenes de advertencia o anuncios. Para determinar los eventos que ocurren a este nivel del análisis se deben priorizar acciones bastante específicas, ya que el análisis correcto de las

ambas propuestas de interfaces debe plantearse como una competencia encarnizada de ambos diseños. Haciendo un símil con las teorías de la supervivencia del más apto, ambos individuos (diseños) comparten una misma presa (objetivo de la interfaz) y comparten el mismo hábitat (elementos y entorno de uso), ambos deben competir para sobrevivir y lo harán si son más astutos (diseños más fáciles de entender) y más fuertes (diseños más fáciles de usar) que su rival.

Es a partir de este punto que las interfaces se mejoran con cada iteración o etapa del proceso de desarrollo. Tomando en cuenta que “mejorar una interfaz” por mera definición considera un campo bastante amplio e incluso ambiguo, como se menciona anteriormente, en secciones posteriores se complementa con la evaluación sugerida para este tipo de interfaces. Pero de manera general, es posible realizar cualquier tipo de evaluación de UX para cada una de las interfaces, siempre y cuando se aplique el mismo método y proceso experimental. Los resultados que se obtienen permiten resultados a la mano de la evolución de los diseños.

A pesar del símil de párrafos anteriores con una competencia de supervivencia, en la aplicación real de esta metodología de diseño ambas propuestas se retroalimentan entre sí para encontrar los mejores métodos de interacción para cumplir con una tarea específica. El concepto de diseño de interfaces paralelas se fundamenta en la expresión: “Diferentes caminos, mismas soluciones”. Lo que hace referencia a que el propósito de generar dos diseños es que se puede encontrar una convergencia de las soluciones a pesar del punto de partida. Si los objetivos de diseño son claros y las evaluaciones de las interfaces son de calidad las probabilidades del éxito del diseño aumentan ya que se coteja con mayor rigor las características de uso de cada diseño uno frente al otro.

Una de las debilidades de las interfaces paralelas, y como se ha mencionado anteriormente, es que las evaluaciones e investigaciones de UX que se le aplican a cualquier interfaz están sujetas a ciertos grados de subjetividad por parte del *tester*. Extendiendo este punto, al analizar el uso del diseño puede ser bastante complicado encontrar los puntos críticos que entorpecen al usuario para completar la tarea. Esto limita altamente el análisis de bajo nivel de una interfaz. Sin embargo, si se plantea un proceso de desarrollo que contemple tanto el diseño paralelo como una evaluación objetiva se establecería una mejora notoria en la calidad de los diseños de interfaces para ser entendidos y usuarios.

Una vez abordado el concepto de interfaces paralelas y habiendo establecido las desventajas que puede tener la metodología de diseño de interfaces propuesta si se usa en metodologías de desarrollo tradicional o ágil, se procede a establecer un marco de trabajo que, además de permitir

la completa y correcta implementación de la teoría de interfaces paralelas, también brinda una primera aproximación a una evaluación objetiva de interfaces mediante la implementación de técnicas de seguimiento de mirada. Así como el establecimiento de la teoría de análisis de alto y bajo nivel para el diseño de interfaces.

3.5 Metodología de selección e interpretación de métricas de *eye-tracking*

Gracias a las tecnologías de eye-tracking de la actualidad es posible acceder al análisis biométrico de la visión humana mediante métricas precisas. Sin embargo, existe poca fundamentación teórica en cuanto al significado de mencionadas métricas en aspectos particulares de la atención visual. Como se maneja en el apartado 2.3 referente a los conceptos y panorama general del eye-tracking existe una primera condición que permite realizar pruebas de uso para interfaces, y es que una pantalla es capaz de representar en un plano bidimensional una gran cantidad de elementos desde lineales hasta tridimensionales aprovechando diferentes técnicas visuales.

La idea anterior de manera práctica define las condiciones que permiten al análisis de eye-tracking en monitores o pantallas como una herramienta que permite analizar una variedad muy amplia de posibles elementos, entornos, escenarios, imágenes, interpretaciones, entre otras interacciones sin tener que considerar tan a detalle el contexto ya que todos los contenidos, por complejos que sean usan técnicas visuales ya anteriormente estudiadas por campos como la fotografía, cinematografía, efectos visuales digitales, entre muchos otros campos, por lo que para una gran parte de los usuarios ya cuentan con antecedentes de los cuales apoyarse para proceder con la interacción en elementos en distintas dimensiones.

Sin embargo, la complejidad del análisis de elementos tridimensionales por largos periodos de tiempo es una tarea altamente complicada para las capacidades de la inferencia de métricas de eye-tracking que existe actualmente. Es por esa razón que se estima que para poder proponer un análisis en experiencias mucho más complejas y tridimensionales es necesario escalar paulatinamente en las interpretaciones que pueden realizarse sobre las métricas que ofrece el eye-tracking en un conjunto de elementos definidos, estáticos y más precisamente, bidimensionales.

3.5.1 Fundamentación de las métricas de eye-tracking seleccionadas

En la sección 2.3 referente a los conceptos que permiten respaldar las propuestas que se expresan en esta investigación referentes al eye-tracking es posible encontrar dos principales variables que, de manera muy general, permiten reducir a valores medibles algunos de los procesos de atención visual del ser humano: fijaciones y movimientos sacádicos. Estas dos variables hacen referencia a la secuencia más comúnmente observada en la mayoría de los usuarios ya que son procesos cognitivos que, independientemente de ser voluntarios o involuntarios siempre se rigen por la búsqueda de información más eficiente (Sharafi et al., 2016).

La primera etapa del proceso mencionado en el párrafo anterior hace referencia a las fijaciones, si bien las visualizaciones pueden aparentar que los puntos de retención de mirada en una interfaz indican únicamente la posición de los ojos, en un análisis más minucioso de eye-tracking es posible encontrar información de valor en la cantidad y duración de las fijaciones.

La cantidad de fijaciones sobre un elemento está asociado a la eficacia de este para brindar información relevante al usuario, esto significa que si un elemento es capaz de ser entendido fácilmente será visitado con menor asiduidad. A su vez, la duración de las fijaciones indica que la capacidad de un elemento para proveer información útil para planear y ejecutar la siguiente interacción del usuario hará que este retenga su mirada durante más tiempo en determinadas áreas particulares, esta métrica acumula los milisegundos de las fijaciones sobre un área en particular (Colmenero et al., 2001; Gutiérrez de Piñeres Botero, 2019).

Antes de pasar a la posible inferencia de las métricas de los movimientos sacádicos es preciso definir los procesos visuales y cognitivos que impulsan dichos movimientos. Primeramente, existe una amplia discusión sobre si el movimiento sacádico es mayormente voluntarios o involuntarios. La diferencia entre la voluntariedad radica en que existe un grado de subjetividad por parte de diferentes usuarios en los elementos que se reconocen como más interesantes para realizar una siguiente fijación, mientras que síncronamente y de manera involuntaria se aplican patrones de recorrido de mirada que son más generales en la recolección de información (Duchowski, 2017).

Sin embargo, las técnicas de eye-tracking no son capaces de determinar si un movimiento es voluntario o involuntario ya que este movimiento que transcurre en el orden de milisegundos contempla procesos cognitivos bastante complejos que no es posible verificar mediante la señal infrarroja del dispositivo de seguimiento de mirada.

Por lo anterior, la información que es posible analizar a detalle son los movimientos en sí con la información inherente que se genera, primeramente, el conteo de la cantidad de movimientos sacádicos en un espacio determinado puede representar que tan eficiente es un elemento para impartir información al usuario (Martínez Gutiérrez, 2008).

Además de la cantidad de movimientos sacádicos es posible obtener el promedio de la aceleración de estos, esta métrica hace referencia a la eficiencia del patrón de recolección de información sobre todo en procesos de lectura (Referencias Parodi). Un promedio de aceleración altos es posible relacionarlo con un proceso de lectura o de recolección de información poco eficientes, contraparte de esto, los promedios bajos se relacionan con una facilidad intrínseca del área en particular para ofrecer información (Ramírez Coronel, 2018).

Parecido a la métrica de aceleración de los movimientos sacádicos, la amplitud de estos también representa información relevante sobre la eficiencia de los recorridos en la obtención de información, si se promedian las aceleraciones de los movimientos sacádicos dentro de un área en particular es posible aumentar la capacidad del análisis que se aborda en la sección siguiente sobre la complejidad del recorrido dentro de los elementos (Sharafi et al., 2016).

3.5.2 Formalización de los elementos del análisis de uso de interfaces mediante eye-tracking propuesto

En la sección anterior se abordaron las teorías que mejor suportan la propuesta de que es posible mejorar las evaluaciones del uso de interfaces mediante el entendimiento de los movimientos oculares. Es de destacar que en diferentes trabajos que esta investigación consulta se encontraron puntos en común sobre el entendimiento que es posible aplicarse sobre las métricas de eye-tracking.

Mencionado lo anterior, esta investigación propone establecer una nomenclatura y formalización de los posibles entendimientos que pueden abstraerse del uso de interfaces mediante métricas de eye-tracking separando y relacionando elementos de interfaces, métricas de eye-tracking e interpretaciones del uso.

A) Interfaz y elementos

Antes de entrar de lleno a los elementos, interpretaciones y ecuaciones del uso de interfaces mediante eye-tracking es importante mencionar que el análisis propuesto de las capacidades de una interfaz está dada por cada uno de los elementos que la componen, de manera que el conjunto de elementos debe trabajar en equipo para ofrecer herramientas para el usuario, sin embargo, también es conveniente examinar las propiedades inherentes al elemento como lo pueden ser su dimensión, ubicación en pantalla, color de fondo, tipo de fuente, color de fuente, entre otras características para verificar si la información contenida está correctamente expuesta para que le brinde información útil al usuario.

Para proponer el análisis de los elementos se comienza con la asignación de un número que lo identifique, en la sección 3.7.3 apartado B referente a las áreas de interés de una interfaz se aborda de manera más detallada la metodología y efectos de analizar únicamente un elemento o congregar varios de ellos en un área mayor. Para abordar esta sección se usará un caso hipotético de una interfaz con elementos finitos y que estos no cambian drásticamente su ubicación, dimensión, colores de fondo o tipos y tamaños de fuentes.

Una vez identificados los elementos que conforman la interfaz se procede a identificarlos con un valor único o bien, letras u otro tipo de nomenclatura específica con el fin de mejorar las capacidades del análisis. A continuación, se propone una nomenclatura que permite encapsular el concepto anteriormente descrito sobre la identificación de elementos contenidos en una interfaz.

a) Nomenclatura para interfaces y elementos

Sea una interfaz de estudio con una cantidad finita de elementos que conforman el diseño y tomando el símbolo “ \rightarrow ” como significa, la interfaz y sus elementos se denotan:

$I \rightarrow$ Interfaz

$n \rightarrow$ Elemento de interfaz

$N \rightarrow$ Conjunto de elementos de una interfaz

$I(n) \rightarrow$ Elemento de la Interfaz I

$I|N| \rightarrow$ Longitud del conjunto de elementos n en interfaz I

$I\{N\} = \{I(n), I(n + 1), \dots, I(|N|)\} \rightarrow$ Conjunto de elementos de la Interfaz I

En el caso de tener que identificar distintas interfaces:

$I_A \rightarrow$ Interfaz A

$I_B \rightarrow$ Interfaz B

$I_A(n) \rightarrow$ Elemento de la Interfaz A

$I_A|N| \rightarrow$ Longitud del conjunto de elementos n en interfaz A

$I_A\{N\} = \{I_A(n), I_A(n + 1), \dots, I_A(|N|)\} \rightarrow$ Conjunto de elementos n en interfaz A

b) Ejemplificación de nomenclatura de interfaces y elementos

La figura 13 representa una propuesta de diseño de interfaz muy sencilla que permite ejemplificar el enfoque de la identificación de los elementos de esta. En este ejemplo se cuenta con 3 elementos: un título para un contenedor, el contenedor y un botón. Como se observa en la figura antes mencionada se identifican los elementos mediante una numeración del 1 al 3. Estos elementos coexisten dentro de la interfaz I. Con lo anterior se establece que:

$I(1) =$ Título del contenedor

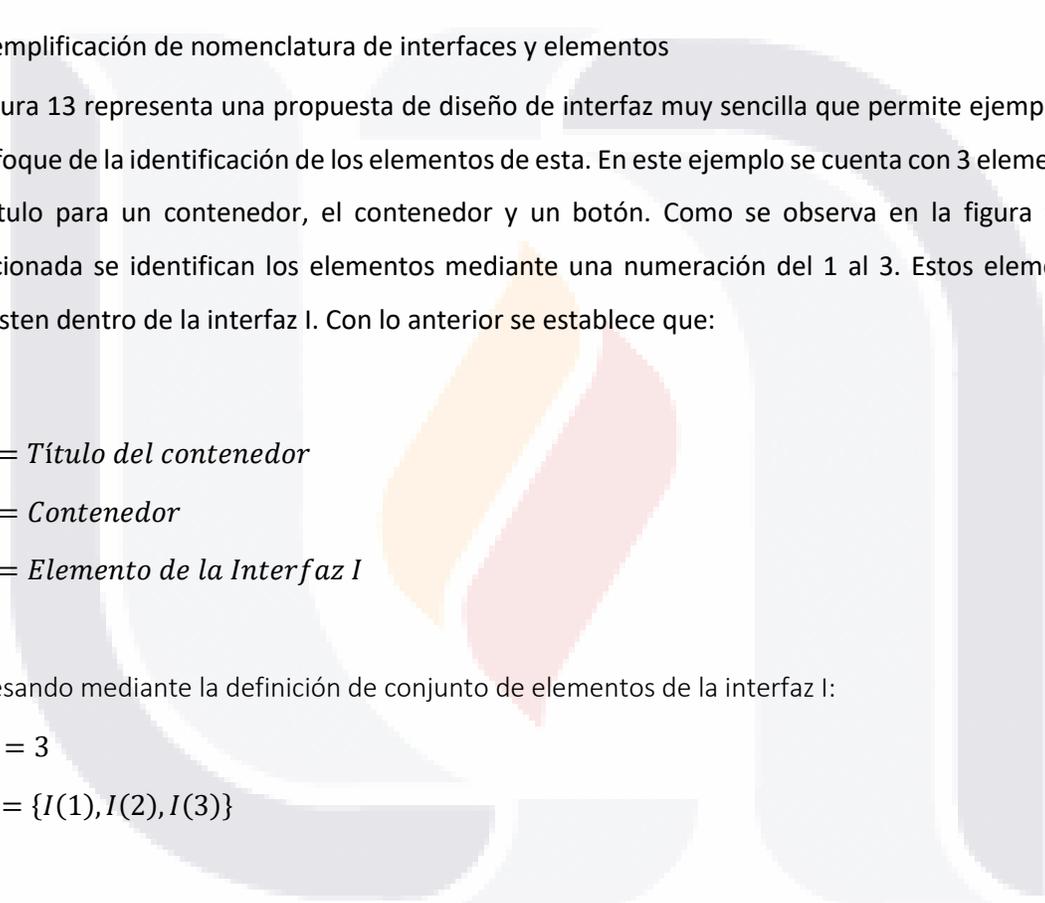
$I(2) =$ Contenedor

$I(3) =$ Elemento de la Interfaz I

Expresando mediante la definición de conjunto de elementos de la interfaz I:

$I|N| = 3$

$I\{N\} = \{I(1), I(2), I(3)\}$



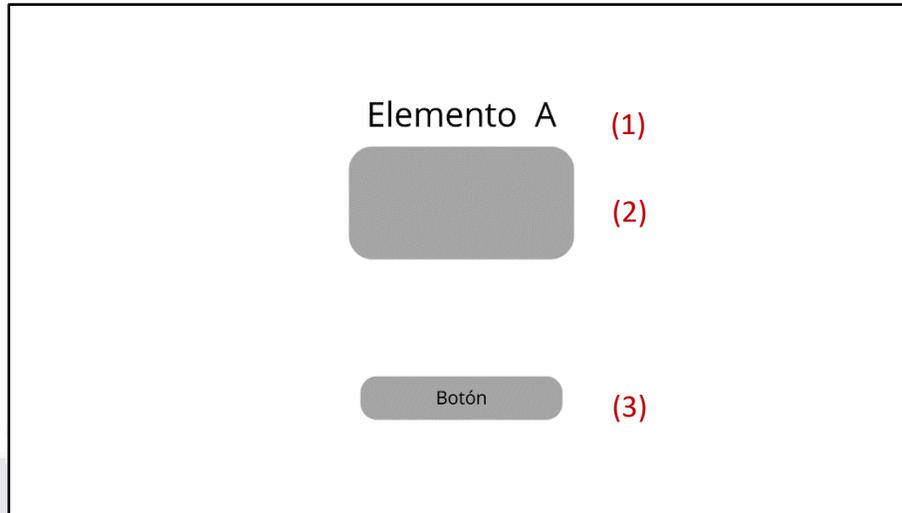


Figura 13: Propuesta de diseño con elementos mínimos para la identificación de elementos n en la interfaz I, (Elaboración propia, 2024).

B) Métricas de eye-tracking

Como se abordó en la anterior es posible relacionar patrones de atención visual con las métricas que ofrecen dispositivos de seguimiento de mirada. También es preciso retomar que las principales métricas otorgadas por las técnicas de eye-tracking son las fijaciones y los movimientos sacádicos.

Con el propósito de generar un primer enfoque objetivo en la evaluación de elementos de una interfaz mediante técnicas de eye-tracking se propone un conjunto de variables que permiten cuantificar patrones de atención visual que de no aplicarse esta tecnología sería imposible detectar en evaluaciones convencionales de software. Mencionando lo anterior, se formalizan a continuación las variables que se proponen en esta investigación para abordar la idea anteriormente mencionada y sentar las bases de un análisis robusto y fundamentado de los elementos que componen la interfaz.

a) Nomenclatura propuesta para métricas de interfaces

Sean un conjunto de variables que expresen métricas de eye-tracking y usando como sufijo “f” para fijaciones y el sufijo “s” para movimientos sacádicos, tomando el símbolo “→” como significa, las variables propuestas de métricas de eye-tracking se denotan:

$Cf(n) \rightarrow$ Cantidad de fijaciones en elemento $I(n)$

$Df(n) \rightarrow$ Duración de las fijaciones en elemento $I(n)$

$Int(n) \rightarrow$ Interacciones (clicks) en elemento $I(n)$

$Cs(n) \rightarrow$ Cantidad de movimientos sacádicos en elemento $I(n)$

$Vs(n) \rightarrow$ Velocidad promedio de movimientos sacádicos de elemento $I(n)$

En el caso de tener que nombrar un conjunto de interfaces se añade a cada variable un prefijo “I” añadiendo un superíndice como identificador único:

Nota: El superíndice puede ser una palabra o un número si no complica la legibilidad de las ecuaciones relacionadas.

$I_A Cf(n) \rightarrow$ Cantidad de fijaciones en elemento n de la interfaz $A(n)$

$I_{OpcionB} Cf(n) \rightarrow$ Cantidad de fijaciones en elemento n de la interfaz $OpcionB(n)$

b) Tabla de unidades de medida de eye-tracking

En la tabla 1 se expresan las unidades de medida de cada variable abordada en la sección anterior.

Tabla 1: Unidades de medida de variables de ET, (Elaboración propio, 2024).

Variable de ET propuesta	Métrica de ET	Unidad de medida
$Cf(n)$	Cantidad de fijaciones en n.	Conteo num. entero (cant).
$Df(n)$	Duración de fijaciones en n.	Milisegundos (ms.)
$Int(n)$	Interacciones (Clicks) en elemento n.	Conteo num. entero (cant).
$Cs(n)$	Cantidad de mov. Sacádicos en n.	Conteo num. entero.
$Vs(n)$	Velocidad promedio de mov. Sacádicos en n.	Promedio de distancia en pixeles sobre el tiempo empleado (px/ms) para todo mov. sacádico en n.

C) Interpretaciones aplicables a las métricas de eye-tracking.

Parte de los objetivos de esta investigación es encontrar métricas que sean relevantes para el análisis de interfaces, sin embargo, aunque las métricas son una herramienta efectiva para la medición de la atención visual las propuestas llevadas por otros investigadores tienen un enfoque para funcionar particularmente en sus trabajos, dificultando la implementación de los descubrimientos y capando las posibilidades del análisis mediante eye-tracking en otras ramas de estudio.

Cabe destacar que si bien las propuestas abordadas en esta sección 3.5.2 referente a la selección e interpretación de métricas de eye-tracking están ampliamente influenciadas por el análisis, diseño y evaluación de interfaces (y de manera general a la aplicación de eye-tracking en la ingeniería de software), están diseñadas para ser implementadas en cualquier estudio de eye-tracking que desee aplicar los conceptos tratados y formalizados anteriormente, permitiendo también la discusión, difusión, ampliación y mejora de estos con la intención de responder a la idea del párrafo anterior.

Mencionado lo anterior, se proponen una serie de ecuaciones que permiten generar una relación mucho más intuitiva, clara y aplicable para las métricas de eye-tracking y la interpretación que se le puede asignar a cada una de estas (Las métricas se abordan en la sección 3.5.2 apartados B y C).

a) Regla de interpretación para la cantidad de fijaciones Cf(n).

En un área específica de un elemento n de interfaz I se puede entender a la cantidad de fijaciones como la eficiencia en que el elemento es capaz de transmitir información para que el usuario lo identifique y le asigne un rol dentro de la interfaz, es decir, determina si en ese espacio se requiere una lectura, una interacción o una fijación duradera.

$$Cf(n) = \begin{cases} + \text{Cantidad de fijaciones} \rightarrow - \text{Eficiencia de obtención de información} \\ - \text{Cantidad de fijaciones} \rightarrow + \text{Eficiencia de obtención de información} \end{cases}$$

También es posible expresarlo mediante sus variables relacionadas:

$$Cf(n) = \begin{cases} + Cf(n) \rightarrow - E \\ - Cf(n) \rightarrow + E \end{cases}$$

b) Reglas de interpretación para la duración de fijaciones $Df(n)$.

Durante el barrido que se aplica para cada interfaz el proceso de atención visual genera una ponderación de las distintas áreas de la interfaz en función de su capacidad para brindar información relevante en el uso de esta. Expresándolo de otra forma, se puede establecer que, si la fijación responde a un mayor tiempo en un lugar, quiere decir que mientras dure la fijación no considera necesario visitar otra área de la pantalla, ya que está recibiendo información relevante en el lugar donde ubica la mirada en ese momento

Ahora bien, dado que una sola área puede contener distintas fijaciones dependiendo del modelo experimental, es preciso recurrir a la herramienta de sumatoria de los milisegundos en los que se recurre a este elemento durante el uso de una interfaz.

$$Df(n) = \begin{cases} + \sum \text{Duración de fijaciones} \rightarrow + \text{Relevancia de obtención de información} \\ - \sum \text{Duración de fijaciones} \rightarrow - \text{Relevancia de obtención de información} \end{cases}$$

De manera simplificada se expresa de la siguiente manera:

$$Df(n) = \begin{cases} + \sum Df(n) \rightarrow +R \\ - \sum Df(n) \rightarrow -R \end{cases}$$

c) Reglas de interpretación para las interacciones $Int(n)$.

Como se ha abordado en la sección 4.6.1 las interacciones son un proceso complejo para la atención visual, ya que requiere en tiempo y atención grandes esfuerzos por parte del usuario. Ahora bien, dependiendo de los requerimientos de la interfaz un elemento puede ser diseñado para recibir más o menos interacciones, por ejemplo, en un formulario sencillo se esperaría que si las instrucciones son claras y los campos están correctamente diseñados el usuario pulsaría únicamente una vez el botón de enviar formulario, esto indicaría que el diseño de la interfaz es eficiente, por otro lado, si se está probando un menú desplegable y se ejecutan muchas interacciones significa que la información necesaria para realizar una tarea está oculta o el proceso para llevarla a cabo es muy rebuscado. La idea anterior se refleja en la siguiente ecuación que relaciona las interacciones con la facilidad de uso.

Ahora bien, en muchas interfaces existen elementos que no están pensados para ser usados, o que en la práctica no se usaron ya que había otras opciones de navegación o interacción. Este tipo de casos es denominado un “caso cero” de Interacciones, denotado de la siguiente manera: $Int(n) \rightarrow \emptyset$. Esta propiedad permite al evaluador desestimar elementos que no son diseñados para recibir una interacción o que no cumplen ninguna labor en el estudio de una tarea específica contenida en la interfaz. Es por esta misma razón que depende del criterio del evaluador determinar si un elemento debe ser considerado como apto para establecer un “caso cero” o, por lo contrario, el “caso cero” es una anomalía de uso de la interfaz que hay que corregir.

$$Int(n) = \begin{cases} + \text{ Interacciones (clicks)} \rightarrow - \text{ Facilidad de uso} \\ - \text{ interacciones (clicks)} \rightarrow + \text{ Facilidad de uso} \end{cases}$$

También se puede expresar mediante los nombres de las variables relacionadas:

$$Int(n) = \begin{cases} + Int(n) \rightarrow - F \\ - Int(n) \rightarrow + F \end{cases}$$

d) Reglas de interpretación para la cantidad de movimientos sacádicos $Cs(n)$.

Como se ha mencionado anteriormente, los movimientos sacádicos son un tema de debate sobre el deseo cognitivo que representan para el usuario, ampliando lo anterior, algunos trabajos indican que el movimiento sacádico es producto del descarte de la zona en la que se posa la atención visual previo al movimiento, contraparte de eso se teoriza que el proceso de barrido es más eficiente de lo que parece a simple vista y que cada movimiento sacádico responde a la búsqueda ordenada y secuencial de la información. En ambos casos, es posible relacionar la cantidad de movimientos sacádicos con la capacidad del elemento para proveer información de valor mediante las propiedades de este en sus aspectos de dimensión, color y coherencia en el espacio que ocupa en la pantalla. Lo anterior significa que si el elemento es interpretado fácilmente desde el principio requerirá menos movimientos sacádicos para identificar las propiedades del elemento.

$$Cs(n) = \begin{cases} + \text{ Cantidad de mov. sacádicos} \rightarrow - \text{ Eficiencia de obtención de información} \\ - \text{ Cantidad de mov. sacádicos} \rightarrow + \text{ Eficiencia de obtención de información} \end{cases}$$

Mediante sus variables se expresa:

$$Cs(n) = \begin{cases} + Cs(n) \rightarrow - P \\ - Cs(n) \rightarrow + P \end{cases}$$

e) Reglas de interpretación para la velocidad promedio de sacádicos Vs(n).

Ya se ha establecido que los movimientos sacádicos están relacionados con las áreas que en las que se espera obtener información que permita ahorrar tiempo y esfuerzo cognitivo. Aunado a lo anterior el proceso de lectura que puede parecer muy natural e intuitivo representa todo un reto cognitivo y está compuesto por múltiples etapas desde el reconocimiento de caracteres, análisis semántico del texto, aplicación de experiencia previa, hasta sesgos culturales. Este proceso es tan amplio que se podría generar un trabajo de investigación contemplando cada aspecto que influye en el proceso de lectura. De manera general se establece que las etapas mencionadas anteriormente pueden cuantificarse mediante la velocidad de los movimientos sacádicos ya que expresa la naturalidad del recorrido, así como el esfuerzo requerido para leer e interpretar el elemento dónde se posa la mirada.

$$Vs(n) = \begin{cases} + Velocidad prom. de mov. sacádicos \rightarrow - Esfuerzo de lectura requerido \\ - Velocidad prom. de mov. sacádicos \rightarrow + Esfuerzo de lectura requerido \end{cases}$$

La idea anterior es posible expresarla también de la siguiente manera:

$$Vs(n) = \begin{cases} + Vs(n) \rightarrow - H \\ - Vs(n) \rightarrow + H \end{cases}$$

f) Tabla de variables de eye-tracking y sus reglas de interpretación.

En la siguiente tabla (tabla 2) se congregan las métricas de eye-tracking que han sido abordadas en las secciones anteriores y sus interpretaciones relacionadas para facilitar la lectura de esta información.

Tabla 2: Tabla de métricas de eye-tracking y sus reglas de interpretación, (Elaboración propia, 2024).

Variable	Interpretación relacionada	Regla de interpretación
<i>Cf(n)</i>	<i>E</i> → Eficacia de obtención de información	$Cf(n) = \begin{cases} + Cf(n) \rightarrow - E \\ - Cf(n) \rightarrow + E \end{cases}$
<i>Df(n)</i>	<i>R</i> → Relevancia en la obtención de información	$Df(n) = \begin{cases} + \sum Df(n) \rightarrow + R \\ - \sum Df(n) \rightarrow - R \end{cases}$

$Int(n)$	$F \rightarrow$ <i>Facilidad de uso</i>	$Int(n) = \begin{cases} + Int(n) \rightarrow - F \\ - Int(n) \rightarrow + F \end{cases}$
$Cs(n)$	$P \rightarrow$ <i>Eficiencia en la obtención de información</i>	$Cs(n) = \begin{cases} + Cs(n) \rightarrow - P \\ - Cs(n) \rightarrow + P \end{cases}$
$Vs(n)$	$H \rightarrow$ <i>Esfuerzo de lectura requerido</i>	$Vs(n) = \begin{cases} + Vs(n) \rightarrow - H \\ - Vs(n) \rightarrow + H \end{cases}$

3.6 Índices de Uso de Interfaces mediante Eye-Tracking (INUSINET)

En las secciones anteriores se han fundamentado y formalizado mediante una nomenclatura a tres diferentes elementos: interfaces, métricas de eye-tracking e interpretaciones propuestas. Estos aspectos ofrecen nuevas capacidades para generar análisis de interfaces utilizando mediciones biométricas objetivas de los patrones de atención visual de los usuarios.

Ahora bien, el análisis de interfaces comprende distintos aspectos como lo son la identificación de los elementos que mejor se adapten al contenido que se desea expresar, la selección de las propiedades de los elementos (distribución, dimensión, fuente y color), la corrección de elementos pobremente diseñados o la detección de patrones de uso poco intuitivos.

Las características cualitativas anteriormente descritas permiten evaluar la interfaz en sus aspectos de diseño, más no de su uso. Para proponer un conjunto de variables cuantitativas se aplica una unión de las variables de las métricas de eye-tracking propuestas anteriormente para aprovechar las interpretaciones asociadas.

Ahora bien, retomando la teoría de atención visual y sus dos principales variables (fijaciones y movimientos sacádicos) es preciso mencionar que estas variables son complementarias, es decir, para que exista un movimiento sacádico debe tener como nodos un par de fijaciones, y viceversa, entre cada fijación existe una distancia física que se entiende como el movimiento sacádico. Esta característica restringe la unión de variables que involucren a las fijaciones con las variables que involucren los movimientos sacádicos.

Por lo anterior, se propone un modelo de índices que permite cuantificar el uso de los patrones de atención visual al usar interfaces denominado índices de uso de interfaces mediante eye-tracking (INUSINET). En la figura 14 se propone un diseño para representar de manera visual la propuesta del análisis de datos biométricos obtenido por técnicas de seguimiento de mirada. El diseño cuenta con una serie de elementos que simbolizan las fijaciones y los movimientos sacádicos, mientras que

la dimensión y distancia entre cada elemento puede entenderse como el estudio de estas pequeñas variaciones para la interpretación de interfaces.

Los índices que componen INUSINET están planteados en función del entendimiento, recorrido y uso que es posible encontrar en una interfaz. Entonces, los índices se denominan: Índice de Complejidad de Entendimiento (ICE), Índice de Complejidad de Recorrido (ICR) e Índice de Complejidad de Uso (ICU). A continuación, se aborda cada uno de estos índices, así como las variables de eye-tracking que permiten obtener un valor cuantitativo del uso de interfaces.

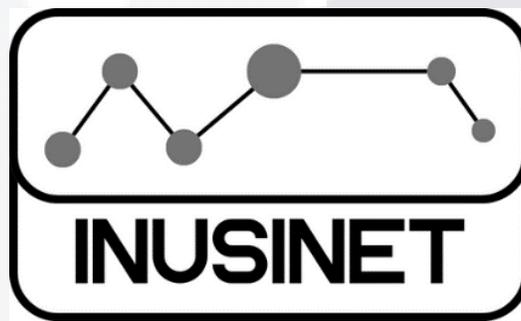


Figura 14: Diseño visual de propuesta INUSINET, (Elaboración propia, 2024).

3.6.1 Índice de Complejidad de Entendimiento (ICE)

Al analizar una interfaz cada elemento que la compone forma parte de una interpretación que le permite al usuario identificar las posibles interacciones o procesos que puede realizar para cumplir su objetivo. La interpretación parte de la información contenida dentro de cada elemento, también en este punto se refleja la relevancia que tienen varios elementos en la obtención de información. Por esta razón es que las variables elegidas para este índice comprenden la cantidad de fijaciones (Cf) y la duración acumulada de las fijaciones (Df) que corresponden a las interpretaciones E y R . La idea anterior es posible expresarla de la siguiente manera:

Sean las variables de eye-tracking Cf y Df , la unión de las variables conforma el ICE, el cual representa la unión de las interpretaciones E y R :

$$Cf \cup Df = ICE = \{Cf, Df\}$$

$$ICE = \{Cf \rightarrow E, Df \rightarrow R\}$$

A. Valor Cuantitativo del ICE

Una vez planteada la relación entre ambas variables se busca un obtener un valor que indique la razón correspondiente entre la interpretación relacionada a la variable y los valores obtenidos mediante eye-tracking del uso que le da un usuario a la interfaz. Esta razón se plantea dividiendo la cantidad de fijaciones (*Cf*) entre la duración acumulada de fijaciones (*Df*) para obtener el índice de eficacia de obtención de información (*E*) sobre la relevancia en la obtención de información (*R*).

Este índice hace referencia a la capacidad de las características del elemento de ser identificado fácilmente debido a que posee propiedades que lo hacen fácilmente reconocible ya que su dimensión, color, ubicación en la pantalla y fuentes son más o menos adecuadas para que el usuario complete la tarea u objetivo dentro de la interfaz. Entre más elevado sea el valor del índice más sencillo es de entender el elemento.

B. Ecuación del ICE

Mencionado lo anterior, la ecuación que permite asignar un valor cuantitativo de la capacidad de entender un elemento dentro de una interfaz es la siguiente:

$$ICE = \frac{\text{Cantidad de fijaciones } (Cf)}{\sum \text{Duración de fijaciones } (Df)}$$

3.6.2 Índice de Complejidad de Recorrido (ICR)

Dado que se ha teorizado sobre la relación complementaria de los movimientos sacádicos y las fijaciones es que se propone un índice que contemple las variables *Cs* y *Vs* relacionadas a los movimientos sacádicos para obtener un valor cuantitativo de las interpretaciones *P* y *H* propuestas en las secciones anteriores.

Sean las variables de eye-tracking Cs y Vs, la unión de las variables conforma el ICR, el cual representa la unión de las interpretaciones P y H:

$$Cs \cup Vs = ICR = \{Cs, Vs\}$$

$$ICR = \{Cs \rightarrow P, Vs \rightarrow H\}$$

A. Valor cuantitativo de ICR

El recorrido sobre un elemento es posible relacionarlo con la información que es capaz de recibir el usuario de forma eficiente (P), además, es una herramienta para analizar el esfuerzo de lectura requerido (H).

El índice aborda la interacción que toma como base la capacidad de un elemento para brindar información de manera eficaz. La diferencia de los términos entre eficiencia y eficacia hace referencia a que los movimientos sacádicos están regidos por la memoria de trabajo. Este concepto responde al proceso de barrido de la interfaz en busca de los elementos que permitan identificar la secuencia esperada para el uso de esta. Por esta razón se aplican la cantidad de movimientos sacádicos (C_s) y se relaciona directamente con el esfuerzo de lectura requerido para analizar el documento, es decir, si el elemento cuenta con pocos movimientos sacádicos y cada uno de estos fue lento se puede interpretar que el elemento no cuenta con la claridad necesaria en cuanto a dimensión, color, distribución en pantalla y fuentes que permitan al usuario acceder a la información contenida a mencionado elemento.

B. Ecuación del ICR

La siguiente ecuación permite obtener un valor cuantitativo de los procesos de obtención de información para usar un elemento mientras que se divide entre la complejidad de leer el texto o imagen contenida en el elemento.

$$ICR = \frac{\text{Cantidad de mov. sacádicos } (C_s)}{\text{Velocidad prom. de mov. sacádicos } (V_s)}$$

3.6.3 Índice de Complejidad de Uso (ICU)

El uso de un sistema es una tarea que puede parecer muy sencilla para un usuario que utilice softwares constantemente, sin embargo, la facilidad del uso de un sistema está relacionado con la experiencia previa del usuario a distintos niveles. Por ejemplo, una persona que recién está usando un computador tendrá que realizar interpretaciones más complejas para llevar a cabo una acción, si este usuario reincide en el uso del software poco a poco su uso será más intuitivo, preciso y le tomará menos esfuerzo reconocer cada elemento. Para proponer una primera parametrización de

este proceso de uso se implemente el índice de complejidad de uso, que toma en cuenta la interpretación F referente a la facilidad de uso.

Además de la facilidad de uso, también se considera relevante estudiar cómo el usuario apunta la mirada a un elemento para después utilizar los periféricos para, de manera voluntaria y consiente, efectuar una interacción en el software, provocando un cambio de estado del sistema.

El cambio de estado en el sistema no es un aspecto menor en el uso de una interfaz ya que representa un paso en la secuencia que interpreta el usuario para completar la tarea de forma eficiente. Entonces, si el diseñador es capaz de verificar que los elementos están siendo usados de manera que permitan al usuario completar su tarea fácilmente se está cumpliendo el objetivo más fundamental de la usabilidad.

A. Valor cuantitativo del ICU

En el párrafo anterior se enfatiza la importancia del reconocimiento de los patrones de uso y para eso, es conveniente utilizar una variable antes mencionada que permite identificar la capacidad de un elemento para ofrecer información de uso para el usuario: la cantidad de fijaciones (*Cf*).

La razón entre la capacidad de visualizar un elemento y entenderlo está estrechamente relacionada con la información proporcionada por este. Expresándolo de otra manera, para un usuario incluso sin experiencia que está completando un formulario, teóricamente, le sería más sencillo usar un botón que contenga el texto “Enviar respuestas” que un botón que contenga el texto “Avanzar”. Lo anterior se hace más evidente si la distribución, dimensión, color y fuentes de los demás elementos que conforman la interfaz no brindan de manera coherente una estructura que le permita al usuario reconocer de manera intuitiva el proceso requerido para completar una tarea.

B. Ecuación del ICU

Para cuantificar la idea anterior mediante las métricas e interpretaciones relacionadas se propone la siguiente ecuación:

$$ICU = \frac{\text{Cantidad de fijaciones (Cf)}}{\text{Cantidad de interacciones (Clicks) (Int)}}$$

3.6.4 Implementación de INUSINET

De manera convencional los análisis de eye-tracking analizan estímulos muy cortos en duración de exposición. Uno de los retos con los que se enfrenta el establecimiento de métodos de implementación de técnicas de seguimiento de mirada es precisamente que cada diseño experimental dista mucho uno del otro. Así mismo, la evaluación de software tiende a estar relacionada al cumplimiento de requerimientos específicos establecidos por un analista y diseñador del sistema, de modo que estos actores son generalmente el juez, jurado y verdugo de las propiedades de cada uno de los elementos.

Desarrollando lo anterior, cada proceso de desarrollo de software es sometido a una infinidad de variables en cuanto a la selección de metodología, cronogramas de desarrollo, tipo, cantidad y características de pruebas, público objetivo, problemática a resolver, entre otra infinidad de características que envuelven el proceso. Es en cada una de estas etapas que el analista y diseñador de software debe tomar decisiones que permitan generar un producto que sea “bueno” en función a las necesidades que este busca cubrir.

La implementación de los índices comprendidos en INUSINET propone, mediante las diversas nomenclaturas y formalizaciones establecidas anteriormente, ofrecer un punto de apoyo para el cotejamiento de los requerimientos y cualidades deseadas del uso de un producto con el uso real que los usuarios efectúan en este último.

Establecer las cualidades deseadas de uso de una interfaz es una tarea muy compleja de realizar ya que el diseñador de software requiere una gran experiencia en los efectos que los patrones de diseño de interfaces provocan en los distintos tipos de usuarios a los que está pensado el producto, a la par de los anterior, se deben contemplar aspectos técnicos de codificación y diseño y, sobre todo, garantizar que el producto sea una herramienta que mejora los procesos o permita realizar una tarea fácilmente.

Concretando la idea anterior, los índices propuestos en este trabajo de investigación requieren un manejo detallado y sensible por parte del analista, diseñador y evaluador de software para determinar si las características de la interfaz responden a los objetivos planteados.

Para proponer un análisis más acercado al uso que se le daría en un entorno real al producto y mantener el análisis de eye-tracking lo más accesible para implementarlo en diferentes procesos de

desarrollo de software se sugiere establecer antes de aplicar cualquier técnica de seguimiento de mirada una serie de parámetros y características que permitirá evaluar, a entender del diseño y evaluador, el uso esperado del producto. Los parámetros esperados de evaluación recomendados son los siguientes.

Tabla 3: Expectativas de uso del caso de uso hipotético, (Elaboración propia, 2024).

Parámetro esperado de evaluación	Descripción
Interfaces de Estudio	Diseños que serán evaluados y la secuencia de navegación entre interfaces.
Cantidad de elementos	Elementos que conforman la interfaz. Se sugiere realizar una lista detallada de cada uno de estos para posteriormente identificarlos mediante la notación propuesta para interfaces y elementos.
Tiempo estimado de la interacción completa.	Este apartado puede estimarse tomando en cuenta un valor máximo de tiempo que una interacción pueda realizarse hasta que se determine que la tarea no pudo ser completada con éxito. Ahora bien, entre mayor sea la complejidad de las interfaces a analizar y la cantidad de estas será mucho más complicado establecer este límite máximo. También debe tomarse en cuenta que someter al usuario a un tiempo cronometrado puede influir drásticamente en el uso que realiza en el sistema.
Tipo de tarea	Establecer si el tipo de interfaz es un formulario, sitio de e-commerce, módulo para un operario, sitio de entretenimiento, entre otros tipos de enfoques.
Relevancia esperada de cada elemento	La relevancia esperada es posible determinarla identificando los elementos que se diseñaron para que sean más visitados, usados o que más se espera que retengan la atención al usuario por tiempos prolongados. Se recomienda

	determinar una jerarquía de relevancia esperada para cada elemento ya que con esto es posible generar una evaluación mediante INUSNET que, en el caso de coincidir con el criterio del diseñador y evaluador, indica que el diseño está más próximo al uso que se desea conseguir del diseño de la interfaz.
Requerimientos de la interfaz	Objetivos principales de uso de la interfaz, este parámetro funge como la pauta tanto para el diseñador como evaluador para tener siempre en mente las necesidades que debe cubrir el diseño. De igual manera, al definir estos objetivos de uso es posible implementar INUSNET para obtener valores objetivos del uso del diseño, y posteriormente cotejar si se encuentran resultados que se están cumpliendo los objetivos y expectativas del diseño de uso propuesto.

Una vez determinados previamente los parámetros de las pruebas y se realizan pruebas válidas de interfaces mediante eye-tracking se procede a congregar los resultados en una tabla de resultados. La tabla de resultados es posible expresarla de diferentes maneras dependiendo del tipo de prueba que se realice, en la tabla 4 se representan los valores de las métricas de eye-tracking de cada elemento que conforma la interfaz separando la información en cada prueba independiente. Cada prueba generalmente está relacionada a un usuario, por lo que este tipo de ordenamiento puede, para un evaluador experimentado en las técnicas de INUSNET, reconocer a simple vista el comportamiento que un usuario específico realizó.

Tabla 4: Tabla de resultados de cada prueba.

Prueba	Elemento de I	Cf(n)	$\Sigma Df(n)$	Int(n)	Cs(n)	Vs(n)
Prueba 1	I(1)					
	I(2)					
	...					
	I(n)					
Prueba 2	I(1)					
	I(2)					
	...					

	$I(n)$					
--	--------	--	--	--	--	--

La segunda forma de representar la información es posible observarla en la tabla 5, en esta se establecen los valores promedio para cada elemento que conforma la interfaz de estudio sin tomar en cuenta cada prueba de manera individual. Esta tabla permite generar resultados para poblaciones de usuarios.

Tabla 5: Tabla de resultados de promedios de elementos.

Elemento de I	Cf(n)	Df(n)	Int(n)	Cs(n)	Vs(n)
$I(1)$					
$I(2)$					
...					
$I(n)$					

Ya con los datos organizados y clasificados es posible aplicar los índices de ICE, ICU e ICR para evaluar las capacidades de uso de cada elemento. Es por esta razón que la tabla de resultados mediante índices permite obtener una gran cantidad de información de manera resumida. Para un ojo experto en evaluaciones mediante INUSINET interpretar estos resultados puede ser una tarea automatizada. En la tabla 6 es posible reconocer el formato propuesto para representar la información de uso de interfaces mediante los índices propuestos de INUSINT.

Tabla 6: Tabla de resultados mediante índices.

Elemento de I	ICE	ICU	ICR
$I(1)$			
$I(2)$			
...			
$I(3)$			

Los formatos propuestos en esta sección pueden variar dependiendo de las características de la interfaz. Uno de los casos donde esto es más evidente es si se despliega un menú, esto significa que los elementos en pantalla pueden cambiar dependiendo de las interacciones que se realicen sobre la interfaz. Para mejorar las capacidades de los procesos de evaluación de interfaces mediante eye-tracking se propone un marco de trabajo que mejore la estandarización de los métodos aquí presentados.

3.7 Marco de trabajo para la Evaluación del Uso de Interfaces mediante Eye-Tracking

(EVUSINET)

Se ha abordado la importancia de las pruebas de uso para generar interfaces que respondan a las necesidades del usuario. Sin embargo, la capacidad de las pruebas de uso se limita a análisis subjetivos y las variables que involucran, si bien buscan tener enfoques cuantitativos (Florián et al., 2010) no siempre logran relacionar adecuadamente los diferentes patrones de uso y sus causas a profundidad.

Mencionado lo anterior, se puede considerar que dentro de la etapa de pruebas de uso la fase que más puede ofrecer información de valor para el diseñador o investigador de uso de la interfaz es la evaluación. Sin embargo, antes de evaluar hay que tener muchas consideraciones. Como todo proceso experimental se pueden presentar variaciones o sesgos de los métodos que se aplican para obtener datos. La validez de las métricas que se propongan está sujetas a condiciones de aplicación específicas, así como la estandarización de las características de los estímulos que se presentan al usuario.

Ahora bien, una de las principales desventajas que se encuentran al revisar el estado del arte de las pruebas de uso con eye-tracking es que se limitan a proponer análisis específicos de sus casos de uso. Ejemplo de lo anterior son los trabajos de (Djamasbi, 2014; Drusch et al., 2011; Martínez Barros & Vásquez Muriel, 2021; Sulikowski et al., 2022; Wong et al., 2014). Lo anterior complica enormemente la aplicabilidad de estos trabajos en otros procesos experimentales. Desde hace algunos años el eye-tracking y el marketing han ido de la mano adaptando los tipos de pruebas a esta rama, sin embargo, para el caso de la ingeniería de software resultan muy distantes los objetivos finales del marketing como para aprovecharlos en pruebas de usabilidad o de análisis de interés de algún producto o una interfaz.

Ya se mencionó una debilidad que presenta el estado del arte de las pruebas de uso mediante eye-tracking, además, en secciones anteriores se aborda la extensa cantidad de metodologías y tipos de procesos de desarrollo de software que hacen imposible tener diseños o productos “universales”. Por consiguiente, uno de los principales objetivos de la propuesta de las pruebas de uso mediante eye-tracking es que debe ser planteada como un componente que se “inserta” en los diferentes procesos de desarrollo dotándolos de una ventaja y facilidad de análisis de interfaces.

Para mejorar las capacidades de la propuesta de aplicarse en diferentes procesos, así como tener la capacidad de analizar interfaces de distintos patrones de diseño se propone un marco de trabajo denominado: “Marco de trabajo para la evaluación del uso de interfaces mediante eye-tracking” o también “Marco de trabajo EVUSINET” y está representado en la figura 16. Esta propuesta se divide en tres etapas correlacionadas entre sí que son: Pruebas de Uso, Uso de eye-tracking y Evaluación de interfaces. Externamente a estas etapas o secciones del marco de trabajo se le añade el proceso de desarrollo de software inherente al desarrollo de un producto.

Uno de los conceptos importantes del marco de trabajo EVUSINET es que está diseñado en formato modular, es decir, cada una de las tres etapas puede ser usada u omitida dependiendo del proceso de desarrollo de software. Sin embargo, y como se abordará de manera mucho más amplia a continuación, el resultado de las evaluaciones está en función tanto del estímulo que se pretende estudiar, cómo de la adecuada aplicación de las técnicas de eye-tracking. Si bien el marco de trabajo busca ser robusto y contemplar una gran cantidad de escenarios, su alteración puede resultar en poca información de valor obtenida por los análisis. Mencionado lo anterior se procede a seccionar el marco de trabajo para explicar cada uno de sus elementos.

Para generar un identificador visual se propone el siguiente diseño que hace referencia a la propuesta EVUSINET. La figura 15 hace referencia al título del marco de trabajo, pero también hace referencia al dispositivo de eye-tracking implementado para generar las pruebas, usando el diagrama en forma de proyección circular para simular los sensores que permiten registrar los patrones de atención visual.



Figura 15: Diseño propuesto para identificar EVUSINET visualmente, (Elaboración propia, 2024).



Figura 16: Diagrama integral de la propuesta EVUSINET, (Elaboración propia, 2024).

3.7.1 Análisis de Uso Particular y General (Análisis de uso de bajo y alto nivel)

La diversidad de los diseños y elementos de las interfaces obligan a plantear una infinidad práctica de escenarios. Cada uno de ellos difiere en objetivo, herramientas para ofrecerle al usuario, colores de la paleta, incluso pueden diferir en el método para darle indicaciones al usuario. Con este panorama tan abrumador se puede llegar a la conclusión que una evaluación objetiva de una interfaz sólo es posible realizarse por medio de una rúbrica específica para el diseño, sin embargo, es precisamente la gran cantidad de patrones de uso e interacciones lo que enriquecen la experiencia del usuario. En otras palabras, en lugar de entender las interfaces como un estático y limitado cajón dónde se encajan elementos dónde menos espacio quiten al resto de elementos, se

TESIS TESIS TESIS TESIS TESIS

propone el enfoque del diseño como un lienzo dónde es posible moldear los recorridos y las acciones específicas del usuario, influyendo así en sus percepción y sensación al usar el producto.

A pesar de la idea anterior, y cómo se aborda en el marco teórico en su sección 2.2.4 apartado B referente a Patrones de Diseño de interfaces, existen muchos tipos de enfoques para ofrecer diseños válidos y funcionales que no vale la pena desestimar por completo, sin embargo, la experimentación de los elementos y su disposición permiten probar con una amplia gama de reacciones por parte del usuario que pueden guiar o mejorar las propuestas de diseño de interfaz.

Ahora bien, el cambio arbitrario de la disposición y características de los elementos no mejora automáticamente la capacidad del producto para cumplir un objetivo. Para lo anterior EVUSINET está planteado para ofrecer métodos aplicables a una gran diversidad de softwares, tomando en cuenta desde casos muy pequeños y específicos hasta experiencias más complejas dentro de una plataforma grande. Para ofrecer un método de estudio para la mayor cantidad de escenarios es que se proponen los conceptos de Análisis de Uso Particular y Análisis de uso General, haciendo referencia al análisis de bajo y alto nivel respectivamente.

El análisis de uso particular comprende específicamente los elementos de una interfaz. De manera ideal este análisis está pensando para aplicarse en un proceso de desarrollo que contemple interfaces paralelas, ya que las características de esta metodología de diseño se enfocan en la comparación directa de cada elemento de las propuestas, ya que ambas comparten mismos elementos y objetivos.

Por otro lado, el análisis de uso general aborda la evaluación del uso de una interfaz cómo el desempeño de un conjunto de los elementos que permiten al usuario completar una tarea que puede realizarse en muchas o una interfaz. Es en este tipo de análisis dónde más es posible jugar con las distintas distribuciones que se le puede dar a los elementos de una interfaz para analizar los recorridos en una plataforma que puede ya estar en el mercado y se busque optimizar algún servicio o módulo en específico.

Independientemente de los enfoques de ambos tipos de análisis no son excluyentes el uno del otro, ambos pueden aplicarse en una sola propuesta de interfaz, así como en un par de propuestas paralelas e interfaces cargadas de muchos elementos. Una parte irremplazable de la evaluación de interfaces es el criterio del investigador ya que es este rol el que establece las secciones específicas

a estudiar de un software y debe dar entendimiento e inferencia a los resultados obtenidos. El marco de trabajo EVUSINET intenta ser una herramienta para validar dichas inferencias del evaluador.

Para generar análisis de alto y bajo nivel del uso de una interfaz a continuación se abordan los procesos que pueden aplicarse en un proceso de desarrollo de software que busque generar diseños deseablemente óptimos o que brinden mayor certidumbre del uso que los usuarios darán una vez se lance el producto.

3.7.2 EVUSINET: Pruebas de Uso

Sobre el tópico de las pruebas de uso existe una gran cantidad de enfoques al respecto. Cómo se mencionó en la revisión teórica del concepto, uno de los enfoques más usados sugiere llevar a cabo entrevistas o encuestas a distintos usuarios de la aplicación (Florián et al., 2010). Otro enfoque como el de (Landa-Durán et al., 2017) no se centra en comentarios sueltos de los usuarios, si no que propone una serie de tareas a completar o acciones específicas de interacción que deben realizarse sobre la interfaz para mejorar las posibilidades de que el diseño ayude al usuario a cumplir una tarea. Este enfoque anteriormente mencionado es parecido al propuesto para el marco de trabajo EVUSINET.

Cabe destacar que las pruebas de uso son una herramienta importante para todo proceso de desarrollo, la fundamentación y fundamentación de las teorías que rodean el concepto han sido abordadas ampliamente, pero incluso con el estudio siguen existiendo aplicaciones que proponen diseños de interfaces o experiencias poco optimizadas o pobres. Una de las hipótesis que pueden surgir sobre cómo pueden generarse diseños de interfaces malos es que las investigaciones de las características del diseño y uso de las interfaces está sesgado o no se aplican herramientas pertinentes que le permitan al diseñador obtener en claro los elementos o acciones que son un factor negativo al interactuar con el producto.

A continuación, se expresan a detalle los elementos que se sugiere tomar en cuenta para generar pruebas de uso que, además de ser válidas para evaluaciones de *eye-tracking*, mejoran los resultados de estas incluso en pruebas convencionales. Para dar seguimiento de una manera más cómoda de la jerarquía de los elementos de esta sección del marco de trabajo EVUSINET se usa la representación parcial de la figura 17.



Figura 17: Sección de pruebas de uso del marco de trabajo EVUSINET, (Elaboración propia, 2024).

A) Diseño Experimental

El primer paso para realizar pruebas de uso de una interfaz es generar un diseño experimental y una de las partes más importantes de esta etapa es generar perfiles de usuarios. La divergencia entre los perfiles de usuarios propuestos no es necesariamente buena o mala, ya que depende de las características del análisis que se pretenda llevar a cabo del sistema. Expresándolo de otra manera, es posible generar análisis de una interfaz tomando en cuenta dos tipos de usuarios completamente diferentes, por poner casos extremos, un perfil de usuario representa a un joven con grado académico en el uso de tecnologías computacionales, mientras que el otro usuario representa a un adulto mayor que tiene nula experiencia utilizando equipos de cómputo. En este caso extremo, los patrones al usar la aplicación pueden ser totalmente divergentes, sin embargo, si el objetivo del análisis es proponer diseños de interfaces especializados en adultos mayores con nula experiencia resulta de mucho valor encontrar las diferencias de uso entre el usuario con experiencia contra la persona que no usa la tecnología asiduamente. Por otro lado, la convergencia entre los tipos de perfiles permite un cotejamiento más estricto y minucioso de las características de uso de una aplicación o sistema más amplio.

Ahora bien, antes de confrontar los tipos de usuarios y sus patrones de uso, deben establecerse en función de las necesidades del análisis que el investigador considere pertinentes. En general un proceso de desarrollo de software robusto y que al ser lanzado pueda ser usado de manera agradable por la mayoría de los usuarios contempla varios tipos de perfiles de usuario. En trabajos

posteriores se plantea generar diseños pensados para perfiles de usuarios específicos usando los patrones de uso divergentes que se observen en personas con alguna limitación cognitiva, motriz, visual, entre otras características que dificulten el uso de una aplicación. Lo anterior para generar propuestas de diseño inclusivas, equitativas y que sus herramientas de accesibilidad no solo se implementen por cumplir el estándar, si no que se garantice que las personas con dificultades de verdad sean ayudadas y guiadas para completar su objetivo al usar la aplicación.

Mencionado lo anterior y ya contando con perfiles de usuario pertinentes para la interfaz o plataforma de estudio, se debe considerar una muestra estadística válida para llevar a cabo el proceso experimental y que este arroje información de valor y patrones reconocibles del uso del software. Sin embargo, el tamaño de muestra en el área de ingeniería de software parece diferir del propuesto en el área de estadística. Como es bien sabido, la teoría estadística propone una serie de parámetros para calcular tamaños de muestra como lo son el nivel de confianza, tamaño de la población, margen de error, desviación estándar, entre otros. Caso contrario, en el apartado de pruebas de uso hasta hace no muchos años se usa un número pequeño de muestra (cinco) para generar las evaluaciones, sin embargo, citando a (Bastien, 2010): “Con una muestra de 5 participantes solo se puede descubrir el 35% de los problemas de uso. Adicionalmente, algunos problemas fueron descubiertos hasta el treceavo y quinceavo participante”. Ahora bien, retomando un concepto que se ha tratado en esta investigación en reiteradas ocasiones la calidad de una evaluación del uso de una interfaz depende de diversos factores, si los mecanismos de evaluación no son buenos el proceso de evaluación avanzará a marchas forzadas, obligando al investigador a realizar más pruebas para descubrir errores o puntos débiles de uso. Si bien la aplicación de técnicas de eye-tracking en la evaluación de interfaces está en una etapa de “primera aproximación”, los datos que brinda la herramienta ya permiten una mejor interpretación general de los patrones de uso que envuelven la experiencia, por lo que, de manera tentativa se propone doblar la cantidad convencional de tamaño de muestra de cinco a diez pruebas de uso para cada interfaz o escenario a verificar.

El otro aspecto necesario para terminar de plantear un diseño experimental es establecer un estímulo. Para escogerlo es preferible proponer interfaces en las que se puedan hacer acciones simples o muy delimitadas, sin embargo, es posible proponer análisis más extensos de uso de una aplicación o sitio web completo. En un primer escenario donde sólo se busque analizar una interfaz se debe centrar el análisis en elementos particulares o acciones simples, por ejemplo: “¿se usó el

botón de enviar correctamente?, ¿el diseño para el ingreso de algún dato es cómodo para el usuario?, “es esta etiqueta o mensaje de advertencia lo suficientemente visible para el usuario?”. Para el segundo escenario como lo puede ser el análisis de una tienda en línea la evaluación se recomienda que se centre en acciones más amplias como: “¿el usuario accedió a la interfaz que recién se implementó?, ¿los botones de regresar o avanzar están permitiéndole al usuario navegar por el sitio web?, ¿logró el usuario llegar de punto A al punto B del sitio web cómo se había planeado?”.

B) Pruebas con Usuarios

Para completar el apartado de Pruebas de Uso se debe tener la certeza de que las pruebas sean replicables y en ambiente controlados. Si bien existe la desventaja de que probar un software en ambientes tan rigurosos dista del uso en el mundo real, el conjunto de usuarios y las variaciones en el uso del software se disipan al aplicar las otras dos etapas del marco de trabajo EVUSINET. Es por lo anterior que las Pruebas con Usuarios deben estipular protocolos experimentales muy detallados para facilitar tanto la replicación de la prueba cómo la aplicación del proceso experimental en otros procesos de desarrollo.

Cómo se observa en el protocolo experimental de la figura 18 para tener un protocolo experimental bien detallado se deben tomar en cuenta los siguientes puntos:

- Momento de selección de la interfaz o estímulo de estudio.
- Etapas de calibración del dispositivo de eye-tracking.
- Condicionales de verificación de ambiente controlado.
- Momento dónde inicia el registro de datos.
- Inicio y final de la interacción con el estímulo.
- Toda condición necesaria para replicar el experimento.

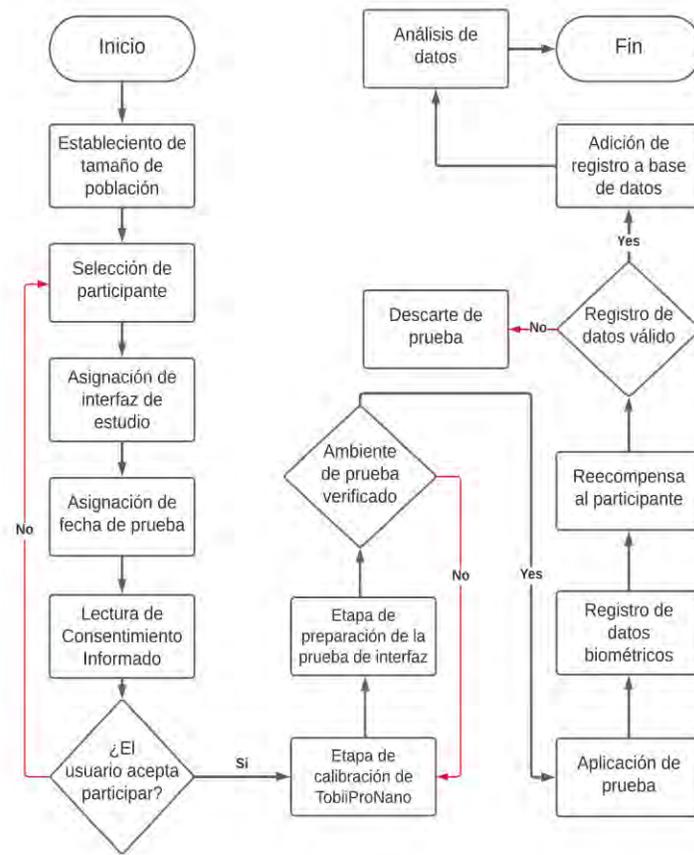


Figura 18: Diagrama de protocolo experimental general para la implementación correcta de las técnicas de eye-tracking en el análisis de interfaces, (Elaboración propia, 2024).

Una vez mencionado lo anterior la primera etapa recomendada del marco de trabajo asienta las condiciones necesarias para la replicación de las pruebas de uso, así como la implementación de éstas en otros procesos de desarrollo. Tomando en cuenta que la calidad de la información está ligada al proceso de recolectarla de manera adecuada para mejorar las capacidades del análisis. En la siguiente sección se detallan las características de la etapa de la implementación de técnicas de eye-tracking.

3.7.3 EVUSINET: Uso de Eye-tracking

El eye-tracking es una herramienta que está ligada a la calidad y complejidad de las métricas que ofrece. Incluso en los productos de un mismo fabricante las características de la información brindada varían levemente, a su vez, métricas como las fijaciones o movimientos sacádicos están calculados por ecuaciones ligeramente diferentes.

Cómo se aborda en (Sharafi et al., 2016) la obtención de los valores que rigen las métricas tiene una fundamentación matemática basadas en los ángulos del eje óptico y el eje visual, además del diámetro de la pupila y la aceleración del diferencial del ángulo de visión. En otras palabras, el simple hecho de obtener las métricas ya está brindando una información relevante de los eventos visuales y en parte cognitivos que el usuario aplica al interactuar con un estímulo.

Sin embargo, al llevar la teoría a la práctica los estímulos no se presentan en periodos controlados o incluso delimitables, es decir, al usar una interfaz los escenarios de uso son tantos como el software utilice. Para lo anterior es que se busca generar análisis que permitan inferir tanto los patrones más específicos y abstractos, así como los patrones de usos reconocibles y deseados.

Si bien hoy es posible generar estudios válidos de eye-tracking con herramientas sin costo, la capacidad del análisis que se genera al usar estas herramientas se reduce bruscamente al usar cámaras web o análisis con librerías de código abierto. Para dar solución a este problema empresas como *Tobii Technology* se dieron a la tarea de generar el software de *TobiiProLab*. En esta herramienta convergen varias de las teorías y estándares más aceptados en los últimos años referentes al eye-tracking. A su vez genera un entorno apto para la experimentación y pruebas de todo tipo.

Cómo primer punto para el uso de las técnicas de eye-tracking se debe diferenciar entre la etapa de preparación o producción del análisis que posteriormente deriva en la fase de exportación de datos, lo anterior se expresa en la figura 19.



Figura 19: Sección de pruebas de uso del marco de trabajo EVUSINET, (Elaboración propia, 2024).

A) Preparación del análisis: Eventos

La preparación del análisis se centra en dividir al estímulo de estudio en alto y bajo nivel. Como se menciona en secciones anteriores presentar una imagen estática para evaluar el uso de una interfaz es muy poco aplicable en el mundo real, en su lugar, es recomendable seleccionar un objetivo global a analizar. Tomando en cuenta la propuesta de interfaces paralelas establecida en esta investigación, la propuesta toma nuevas dimensiones al permitir el análisis secuencial de ambas interfaces, es decir, es posible evaluar el par de propuestas de diseño en un mismo procedimiento experimental. Como se observa en la figura 20 los eventos pueden determinarse como cada una de las interfaces paralelas ya que comparten una mismas instrucciones y objetivos de la interfaz.

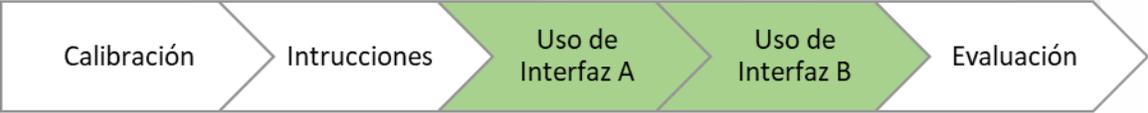


Figura 20: Procedimiento experimental para una propuesta de interfaces paralelas, (Elaboración propia, 2024).

Si bien una de las principales contras que tiene el diseño de interfaces paralelas es que puede costar tiempo y esfuerzo el diseño de dos propuestas, se debe recordar que las interfaces están pensadas para resolver un mismo problema, por lo que las diferencias del diseño están principalmente en la distribución, dimensión, fuentes y paleta de colores, simplificando enormemente el tiempo de

diseño. Posteriormente en la etapa de desarrollo los tiempos y esfuerzos al implementar un único diseño revisado permiten mejores resultados a largo plazo.

El tipo de selección de eventos es conveniente para interfaces paralelas que sean propuestas en una etapa temprana del proceso de desarrollo de software, sin embargo, muchas de las pruebas de uso de interfaces buscan implementar nuevos elementos o verificar el estado actual del software. Para estos casos es necesario relacionar los requerimientos o casos de uso con los criterios de evaluación.

Para ejemplificar lo anterior se ilustra un interfaz de una tienda WEB (figura 21). En esta representación de una página web se contemplan los siguientes elementos: título, barra para buscar producto, carrito de la tienda, banner de ofertas y tres contenedores para productos. Este caso de estudio puede parecer un caso particular, sin embargo, toda interfaz tiene elementos reconocibles y bien diferenciados, independientemente si están enfocados para ser interfaces visuales, de texto, informativas, de algún servicio, etc.



Figura 21: Ejemplo de interfaz de tienda WEB, (Elaboración propia, 2024).

Continuando con el ejemplo de la tienda WEB hay que establecer los diferentes casos de uso que una tienda de este tipo pretende ofrecer a los usuarios. El más evidente es añadir un producto al carrito, en ese caso de uso el evento se determina de esa manera, es decir, la delimitación de los

eventos está en acciones concretas y se parametriza principalmente el tiempo que le costó realizar mencionada tarea, botones o interacciones necesarias para llevar a cabo una tarea y la facilidad con la que se completa esta. Como es posible verlo en la figura 22 la etapa de “Caso de Uso de Investigación” está rodeada de dos etapas de “Uso de Sitio Web o Producto”, lo anterior indica que las acciones del usuario no serán evaluadas desde el primer momento que interactúa con el sistema, si no, el investigador lo delimita conforme el caso de estudio de interés.



Figura 22: Eventos establecidos para la interfaz, (Elaboración propia, 2024).

B) Preparación del análisis: Áreas de Interés: Áreas de Interés

En la sección anterior se abordó las separaciones temporales del uso de una interfaz, con la intención de aumentar las capacidades del marco de trabajo EVUSINET también se hace una revisión de las características de la disposición de los elementos de manera posicional en la pantalla (distribución física). Antes de pasar a la estructuración de las áreas de interés es preciso repasar levemente algunas características de los elementos digitales y las abstracciones que el usuario tiene que lograr para usar el producto.

Anteriormente se mencionó el concepto de distribución “física” en un diseño de interfaces digitales, por contradictorio que aparenta ser esta afirmación, la visión humana siempre genera relaciones físicas de su ambiente (Duchowski, 2017). Si bien los procesos de atención visual han evolucionado para analizar e interactuar con ambientes tridimensionales, a partir del uso de sistemas computacionales con interfaces gráficas, y hasta el día de hoy, es necesario un nuevo tipo de abstracción para interactuar con los elementos de pantalla ya que, si bien el televisor como tal es un objeto tridimensional, la pantalla (o display) es bidimensional, y en ella, se pueden representar objetos de una, dos o tres dimensiones. En otras palabras, el ser humano necesita sobreentender que el estímulo visual que observa es una representación digital de la realidad y usa diferentes niveles y tipos de abstracciones para entender e interactuar con el dispositivo.

Poniendo un ejemplo de este tipo de abstracción, al observar un dibujo animado realizado en dos dimensiones puede sobreentenderse que, si la animación de una animación de una persona camina a la derecha, de facto, recorre su posición en el plano bidimensional, pero si la animación tiene un fondo y la persona se anima para que dé la sensación de alejarse en la profundidad, el espectador da por sentado que la persona de la animación se desplaza en un entorno tridimensional. Esta capacidad cognitiva puede ser muy útil para análisis mediante eye-tracking ya que algunos de los procesos de atención visual pueden ser únicos para analizar un ambiente en dos dimensiones.

Una vez analizado brevemente la importancia del análisis de los elementos en un ambiente bidimensional, se propone el estudio de las interfaces de manera que se divida el conjunto de elementos en secciones rectangulares o que representen áreas de interacción separadas físicamente unas de otras.

Las separaciones físicas no hacen referencia únicamente a la distancia de los elementos entre sí, también refleja una distinción marcada de los objetivos y responsabilidades que tiene cada objeto como parte de un equipo para que el usuario logre una tarea. EL investigador debe ponderar el objetivo de la evaluación con la cantidad y tipo de elementos. Para ejemplificar lo anterior se hará referencia a la figura 21 de ejemplo de interfaz de tienda WEB, en esta figura se representan varios elementos reconocibles como lo son: logo de la tienda en formato de texto, barra de búsqueda de productos, carrito de la tienda, tres productos y etiquetas con el nombre del producto. Para poder identificar de mejor manera los elementos se enlistan y se les añade un identificador de la siguiente manera (Nota: el prefijo “A” de los identificadores hace referencia a una interfaz paralela que tiene como nomenclatura “Interfaz A”):

- [A1]-Logo de la tienda
- [A2]-Barra de búsqueda de productos
- [A3]-Carrito de la tienda
- [A4]-Etiqueta de Descuentos y promociones
- [A5]-Producto 1
- [A6]-Producto 2
- [A7]-Producto 3
- [A8]-Etiqueta Producto 1
- [A9]-Etiqueta Producto 2
- [A10]-Etiqueta Producto 3

Visualmente los elementos serían identificados de la siguiente manera en la figura 23 del ejemplo de Interfaz de tienda WEB:



Figura 23: Ejemplo de Interfaz de tienda WEB con elementos catalogados y diferenciados, (Elaboración propia, 2023).

Agregando a lo anterior, la identificación de los elementos debe contemplar rigurosamente los criterios de evaluación, por lo que, las áreas de interés que se seleccionen deben comprender un solo elemento reconocible o un conjunto de elementos que cumplan una misma función. Para demostrar lo anterior se generaron dos posibles áreas de interés para la figura 21 de ejemplo de interfaz de tienda WEB. En la propuesta A (figura 24) las áreas de interés envuelven cada uno de los elementos reconocibles. Nótese que incluso las etiquetas de cada producto están separadas por un área de interés específica, lo anterior permite un análisis detallado de cada elemento y su papel en la disposición de la interfaz. Como se menciona en la sección de Interfaces paralelas el aprovechamiento de esta técnica se debe a la comparación estricta de cada uno de los elementos de las dos propuestas. Sin embargo, si se desea analizar aspectos específicos del uso de una interfaz es posible conjuntar varios elementos reconocibles dentro de una sola área de interés con el objetivo de evaluar la capacidad de mencionado conjunto de elementos para completar una tarea específica. Lo anterior se ejemplifica en la figura 25 donde los elementos para el producto están comprendidos en una sola área de interés mientras que las etiquetas se mantienen en tres áreas de interés. Lo anterior permitiría evaluar que tan buenos son los elementos para los productos,

mientras que las etiquetas indicarían cuál elemento fue más identificado. La idea anterior ilustra la importancia del investigador en la selección de las áreas de interés.



Figura 24: Propuesta A de áreas de Interés de figura 21, (Elaboración propia, 2023).



Figura 25: Propuesta B de áreas de Interés de figura 21, (Elaboración propia, 2023).

C) Exportación de datos

Una vez que se han establecido los tipos de análisis mediante eye-tracking se debe tomar en cuenta que independientemente de las técnicas y herramientas para llevar a cabo pruebas de uso mediante seguimiento de mirada las métricas brindadas son bastante extensas y con diferenciaciones fundamentales marcadas para cada método.

Si bien el eye-tracking contempla dos principales variables que son fijaciones y movimientos sacádicos, la parametrización de cada una de ellas es diferente para cada herramienta que se usa. Si bien existen varias convenciones cada mecanismo tiene sus documentaciones para poder usar las métricas antes mencionadas. Dentro del software de TobiiProLab las métricas se dividen en fijaciones, movimientos sacádicos, interacciones o clicks, visitas y vistazos, métricas de lectura y textos, entre otras. Cada una de ellas tiene varias categorías que permiten profundizar en las mediciones que se hacen.

Durante todas las etapas de la fundamentación del marco de trabajo EVUSINET se ha hecho evidente la importancia del investigador y su capacidad para encontrar los puntos de uso críticos del uso de una interfaz con el fin de parametrizar los comportamientos tan divergentes del ser humano y lograr condensarlos en un solo patrón de uso y en etapa del proceso de Exportación de Datos es de vital importancia escoger las métricas que permitan realizar una inferencia correcta de los patrones de uso observados.

En la propia documentación de TobiiProLab se exponen ejemplos como el siguiente, parafraseando al artículo obtenido de la página oficial: En el caso de querer evaluar el grado de excitación que siente un usuario al observar un nuevo modelo de automóvil se recomienda usar métricas como fijación, visita y amplitud de la pupila para estimar que tanta sorpresa sintió mencionado usuario. Lo anterior es uno de los muchos escenarios que pueden surgir a la hora de realizar evaluaciones. Esta gran cantidad de escenarios complican ampliamente la capacidad de un investigador de generar análisis a una gran cantidad de procesos de desarrollo de software.

Para poder reducir la divergencia de las métricas en la sección de evaluación se profundizará en las métricas que pueden ayudar al investigador a inferir el uso de una interfaz, por el momento, para pruebas más generales se recomienda usar sumatorias de la cantidad de fijaciones dentro de áreas de interés, promedios de la duración de las fijaciones, promedio de aceleración del movimiento

sacádico, promedio de dilatación de la pupila, tiempo para la primera fijación, duración de tiempos de interés o eventos.

Todas las métricas antes mencionadas son únicamente un atisbo de las posibilidades que permiten las técnicas de eye-tracking y no debe pensarse que son éstas las que permiten el análisis de todos los estímulos y objetivos de investigación que se plantean. Una vez que se toma en cuenta la idea anterior en una evaluación se recomienda generar diferentes versiones de formatos de *data sets* al realizar una o varias pruebas. Ahora bien, en esta investigación no se profundizará a detalle en la estructuración y diseño de las bases de datos que podrían ser útiles para posibles análisis con inteligencia artificial u otras técnicas de análisis de datos por la gran complejidad de este tema y los posibles enfoques que se pueden implementar para los datos biométricos dado que la sola fundamentación y experimentación práctica del tópico anteriormente mencionado con los aspectos de la UX de un sistema daría material suficiente para una tesis completa.

Mencionado lo anterior, la generación de un documento que contiene los datos biométricos, para este tipo de análisis de uso de interfaces, se limita a la aglomeración de las pruebas y sus resultados directos en cuestión de las métricas de interés. Una vez revisadas las dificultades a la hora de elección de métricas y las características de los datos y cómo es su análisis se abordará a continuación la evaluación de las interfaces.

3.7.4 EVUSINET: Evaluación

La capacidad de la evaluación en la actualidad está ampliamente basada en cuestionarios de carácter subjetivo (Florián et al., 2010) o cuándo son cuantitativos estos se limitan a realizar una escala de Likert (Lukito Edi et al., 2018). En la Figura 26 se observa una de las aplicaciones de la escala de Likert en usabilidad y como, a pesar de tener en cuenta muchos factores que comprenden a la UX, no termina de ser una herramienta que permita estimar fehacientemente el uso del software.

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	enjoyable	1						
not understandable	<input type="radio"/>	understandable	2						
creative	<input type="radio"/>	dull	3						
easy to learn	<input type="radio"/>	difficult to learn	4						
valuable	<input type="radio"/>	inferior	5						
boring	<input type="radio"/>	exciting	6						
not interesting	<input type="radio"/>	interesting	7						
unpredictable	<input type="radio"/>	predictable	8						
fast	<input type="radio"/>	slow	9						
inventive	<input type="radio"/>	conventional	10						
obstructive	<input type="radio"/>	supportive	11						
good	<input type="radio"/>	bad	12						
complicated	<input type="radio"/>	easy	13						
unlikable	<input type="radio"/>	pleasing	14						
usual	<input type="radio"/>	leading edge	15						
unpleasant	<input type="radio"/>	pleasant	16						
secure	<input type="radio"/>	not secure	17						
motivating	<input type="radio"/>	demotivating	18						
meets expectations	<input type="radio"/>	does not meet expectations	19						
inefficient	<input type="radio"/>	efficient	20						
clear	<input type="radio"/>	confusing	21						
impractical	<input type="radio"/>	practical	22						
organized	<input type="radio"/>	cluttered	23						
attractive	<input type="radio"/>	unattractive	24						
friendly	<input type="radio"/>	unfriendly	25						
conservative	<input type="radio"/>	innovative	26						

Figura 26: Escala de Likert en usabilidad de sistemas (Lukito Edi et al., 2018)

Mencionado lo anterior, la principal característica de EVUSINET es proveer herramientas objetivas para el análisis de interfaces y por extensión del uso que se hace de estas. En la sección se hace referencia a los diferentes niveles de alto y bajo nivel del análisis de los patrones de atención visual. Este punto de vista propone una primera evaluación sobre la usabilidad de un sistema tomando en cuenta diferentes niveles de análisis.

Al usar una interfaz el proceso de evaluación debe contemplar desde las interacciones más pequeñas hasta los flujos completos que conforman la experiencia. El investigador debe tomar en cuenta en todo momento los objetivos del sistema y las posibilidades que provee el producto para ser una herramienta/solución, con base en eso, se puede inferir el uso que se le está dando al sistema y apoyarse de visualización y aplicación de índices para confirmar sus hipótesis. Como se observa en la figura 27 la evaluación de interfaces contempla precisamente los análisis generales y particulares, comprendiendo así una herramienta de estimación e inferencia de las características de la interacción.



Figura 27: Sección de EVUSINET: Evaluación, (Elaboración propia, 2023).

Antes de abordar las técnicas para evaluar objetivamente el uso de un sistema es conveniente repasar lo que se entiende por “Patrón de Uso”. Para fines de esta investigación se propone entender el patrón de uso como una conducta reconocible o secuencia de interacciones por parte de un conjunto de usuarios. Los comportamientos pueden ser influenciados por el tipo de elementos, su distribución, dimensión, diseño y colores, por lo que, una de las tareas del diseñador es proponer diseños en los cuáles se pueda inducir, hasta cierto punto, el comportamiento del usuario. Una vez que el patrón es detectado, la idea es generar una unión del comportamiento de los usuarios para comprender un patrón de uso. Una vez detectado el patrón, es posible modelar la experiencia para que empate con los objetivos del sistema al modificar la dimensión, disposición, diseño y color.

Al procedimiento anterior se le denomina en esta investigación: modelo convergente de uso, este modelo está altamente enfocado a la generación de diseños en los que los objetivos son la prioridad de la aplicación, tratando de inducir altamente en los patrones de uso por parte del grueso poblacional de los usuarios.

Ya estudiados de manera general los patrones de uso y sus posibles enfoques en la usabilidad se procede a estudiar las metodologías que permiten la evaluación de los procesos de interacción del usuario.

A) Análisis de Patrones de Uso Generales: Visualizaciones

Una de las propuestas de esta investigación es establecer una relación entre la usabilidad de un sistema y los patrones de uso que los usuarios presentan al interactuar con el producto. Ampliando lo anterior, durante la sección de metodologías se ha profundizado en cómo los elementos que conforman la interfaz son los protagonistas para la obtención de mencionados patrones.

Un patrón de uso puede diferir de usuario en usuario, sin embargo, y sobre todo cuando la interfaz presenta varios componentes que funcionan estrechamente entre sí para generar una interacción es posible evaluar la secuencia de atención aplica el usuario para seguir cada paso de la tarea. Es en este punto dónde las técnicas de eye-tracking permiten un registro muy detallado de los procesos de atención visual que gobiernan las acciones consiguientes.

Cómo se menciona en la sección 2.3.3 referente a los métodos que intervienen en los procesos cognitivos de atención visual existen reacciones que son muy complicadas de diagnosticar a la hora de responder a un estímulo. Aun así, es posible definir un comportamiento intermedio de los recorridos y las áreas que se visitan con mayor asiduidad para estimar promedios de uso. Al conjunto de promedios de uso se le puede denominar patrón de uso general. Mencionado lo anterior, una representación gráfica que represente los promedios de uso es considerada una visualización de los patrones de alto nivel.

De manera intuitiva, e incluso de manera teórica, resulta complicado asimilar o suponer que todos los usuarios generarán las mismas interpretaciones de los elementos en pantalla, ya que, de manera expandida se ha propuesto que la experiencia al usar un producto es uno de los elementos más preponderantes en la eficiencia del usuario. Sin embargo, uno de los enfoques que tiene esta investigación es encontrar mecanismos que aumenten la certidumbre de los diseñadores y desarrolladores a la hora de proponer e implantar las propuestas.

Para abordar la metodología relacionada con las visualizaciones de un patrón de uso se plantea un usuario hipotético que se caracteriza por ser un usuario ideal, es decir, completa la tarea de manera inequívoca cada vez independientemente de si tenga experiencia previa o no, pero para ello necesita evaluar cada elemento de la interfaz para comprender el papel de cada uno de ellos y estimar el mejor recorrido. Para generar las visualizaciones se realizaron pruebas inducidas para ejemplificar teóricamente las propuestas presentadas en esta sección, si bien las imágenes que ejemplifican el uso de una interfaz con el mínimo de elementos formaron parte de un diseño

experimental de eye-tracking, las pruebas fueron realizadas simulando posibles comportamientos de uso.

a) Reacciones cognitivas regidas por el contexto cultural y evolutivo.

Tomando en cuenta que, si bien se plantea un usuario ideal, no se desprende de la subjetividad e influencias inherentes del humano. Para representar lo anterior la figura 28 ilustra un primer enfoque en el que sólo se deja un par de bloques con elementos idénticos en la interfaz. En este caso se puede teorizar que la secuencia de atención de mirada por parte del usuario hipotético está directamente influenciada por el lenguaje, es decir, si el usuario es hispano parlante u de manera más amplia, occidental, es muy posible que realice un recorrido de izquierda a derecha, tomando en cuenta que en la figura el punto de color azul es el punto de partida de las fijaciones (Parodi & Julio, 2016).

Además de la interpretación realizada anteriormente sobre la imagen, es preciso añadir que el proceso de atención primeramente reconoce un par de elementos generales en la imagen gracias a la visión periférica. En este ejercicio en particular es muy relevante que los elementos estén a la izquierda o derecha del punto central de la interfaz. Esto activa un proceso cognitivo-visual en el que se pondera la relevancia de un elemento dentro de la estructura general de la interfaz para en primer lugar, entender el tipo de contenido se le está presentando, y en segundo, la manera más coherente de dar solución a la tarea.

Mencionado lo anterior, en el tiempo cero de interacción, el usuario ideal tiene exactamente la misma posibilidad de observar primeramente el elemento A o elemento B, sin embargo, el proceso de reconocimiento mediante la visión periférica de los elementos permite reconocer un elemento con texto, al reconocer este tipo de elemento automáticamente se activa el proceso cognitivo y visual inherente a la lectura al que el ser humano ha sido sometido desde su infancia, por lo que, de manera natural es casi imposible resistirse a este impulso (Parodi & Julio, 2016).

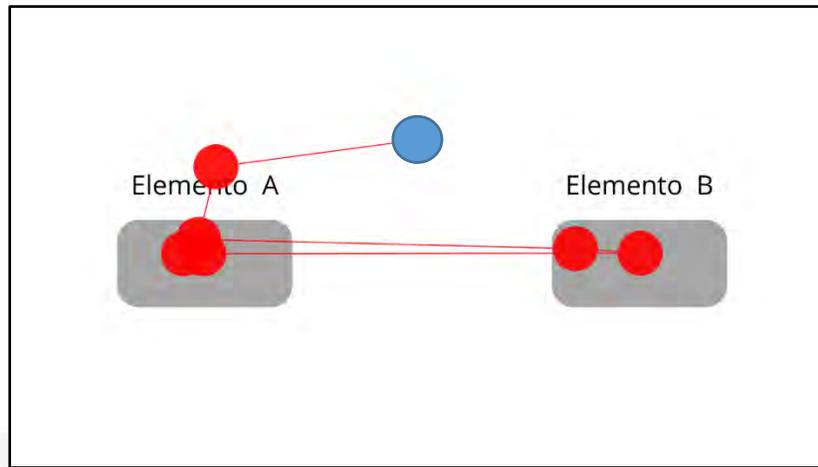


Figura 28: Visualización de recorrido de uso por parte del usuario hipotético en una interfaz con elementos idénticos, (Elaboración Propia, 2024).

b) Dimensión de los elementos.

Continuando con las posibles reacciones por parte de un usuario ideal en ambientes de uso específicos, toca turno de analizar una de las principales propiedades de un elemento: su dimensión. De manera evolutiva existen procesos de atención visual que han ayudado al ser humano a dominar su entorno, dentro de ellos, la capacidad de generar una relación entre la distancia de un objetivo y su posible tamaño ayudó a los primeros seres humanos a calcular distancias o reconocer posibles peligros en la lejanía.

Esta capacidad no pasa desapercibida en el ser humano actual, ya que, en el proceso de análisis de una interfaz, antes de centrarse a detalle en algún elemento, primero compara las características más sencillas del elemento (Colmenero et al., 2001). Cómo se comprobará a lo largo de esta sección, las reacciones del usuario requieren contemplar ampliamente cada elemento, reconocer si tienen algún detalle en particular y, de no tenerlo, se recurre inmediatamente a la dimensión para jerarquizar cada uno de ellos y recorrerlos en un orden de lo general a lo particular (véase figura 29).

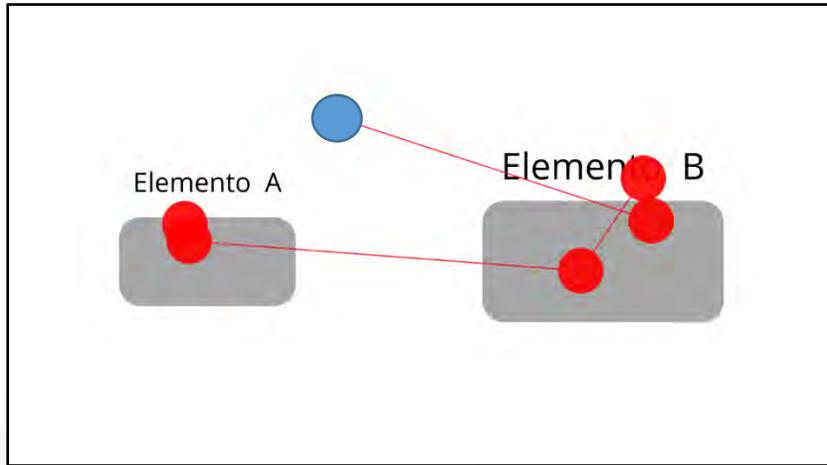


Figura 29: Visualización de recorrido de uso por parte del usuario hipotético en una interfaz con elementos de diferentes dimensiones, (Elaboración propia, 2024).

Lo anterior se hace cada vez más notable mientras la diferencia de dimensiones se amplía. Comparando la figura 29 con la figura 30 es notorio que el conteo de fijaciones en el elemento B es mayor que el elemento A debido a que el segundo es mucho más pequeño que su contraparte. Esta diferencia de dimensiones obliga al usuario ideal a gastar más tiempo recorriendo el elemento B ya que entiende que es el elemento más relevante, además al cubrir más espacio en pantalla, por ende, se requiere más tiempo y fijaciones para terminar de entender el elemento.

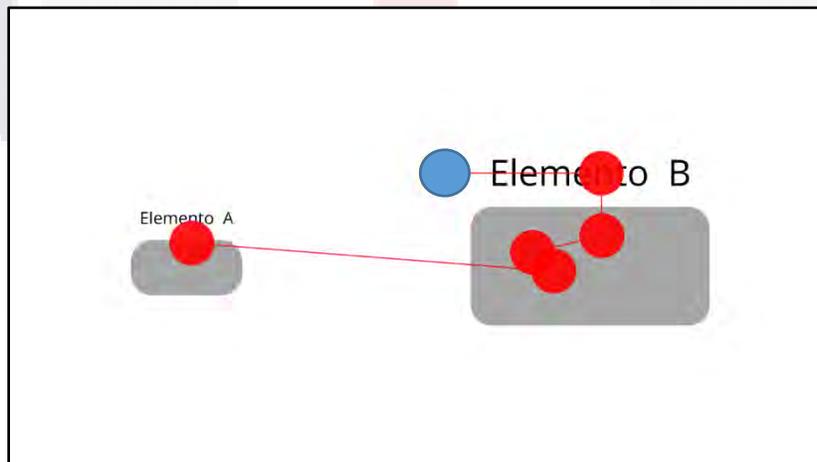


Figura 30: Visualización de recorrido de uso por parte del usuario hipotético en una interfaz con elementos de diferentes dimensiones y títulos reducidos, (Elaboración propia, 2024).

c) Color de los elementos.

Una de las características de los elementos en la que más se involucran los procesos cognitivos evolutivos de la atención visual es el color. Durante el proceso de evolución el ser humano fue mejorando sus métodos preventivos para estar lejos del peligro principalmente. Hoy en día muchos de los mecanismos de reflejos con los que cuenta el ser humano están ampliamente influenciados por el color que el objeto presenta (Tarazona Evangelista et al., 2018).

También conceptos como la señalética (ciencia que estudia las señales de tránsito, entre otras) han usado la información mencionada en el párrafo anterior para proponer e implementar diseños, pero también ha contribuido a propagar la utilización de colores específicos para dar a entender ideas o conceptos más fácilmente para un gran número de personas.

Volviendo al ejemplo evolutivo, para el ser humano es sencillo relacionar colores rojizos y tintos con la sangre, incluso en vexilología (ciencia que estudia banderas y estandartes de países o culturas históricas) se relaciona automáticamente el color rojo este líquido vital. Sin embargo, los estudiosos de este campo de estudio antes mencionado no asignaron esta interpretación de forma arbitraria al color, existe, desde los albores del hombre, un fuerte impulso de supervivencia que permite reconocer este tipo de tonalidad más fácilmente, por lo que, la fuerza y relevancia de este tono ha sido representado a lo largo de la historia en muchas ocasiones.

Como es posible observarlo en la figura 31 en el comportamiento de un usuario ideal, pero que no ignora sus comportamientos más primitivos, su visión se verá altamente atraída al reconocer en su recorrido inicial un elemento con características tan diferentes, además de eso, el tono rojo automáticamente dispara una señal de alerta imposible de ignorar para el cerebro humano.

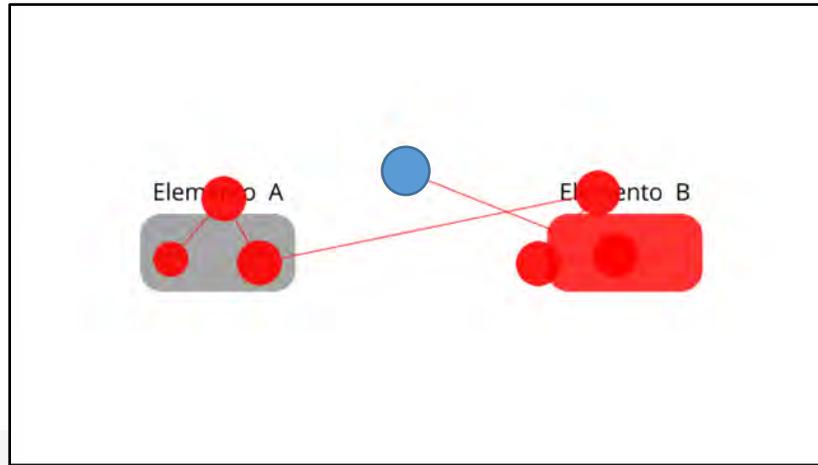


Figura 31: Visualización del recorrido de un usuario ideal en una interfaz con elementos idénticos, pero de diferente color, (Elaboración propia, 2024).

Sin embargo, las reacciones naturales ante ciertos estímulos no son estáticas. Según (Tarazona Evangelista et al., 2018) existen determinantes internos que influyen directamente en el resultado de los procesos de atención visual. Por ejemplo, si el usuario está buscando un producto o imagen específica dentro de un conjunto de otros elementos, es más sencillo reconocer los tonos referentes a la idea que el usuario preconibió en su imaginación. También existen otros factores que alteran los procesos de atención visual como lo pueden ser necesidades fisiológicas (como el cansancio) o deseos o preferencias personales como al observar el color favorito de una persona en una interfaz o reconocer una marca que es de agrado personal.

d) Distribución de los elementos.

Uno de los aspectos que más se revisa en el diseño de una interfaz es cómo distribuir los elementos en pantalla. Si se revisa la historia del diseño de interfaces y tomamos uno de los lenguajes de programación que permitió establecer un método de contenedores para agrupar y jerarquizar elementos, Java utiliza mecanismos para distribuir los elementos de manera uniforme dentro de los anteriormente mencionados paneles. Dentro de este tipo de distribuciones uniformes existen el tipo malla (*Grid*), *Flow Layout*, *Box Layout*, *Border Layout*, entre otros (Glinz, 2000).

Si bien este tipo de distribuciones siguen siendo utilizadas en el diseño ya no representan los elementos principales de la interfaz, ya que existen otros paradigmas como el diseño web, o el

diseño centrado en el usuario los cuáles proponen otros métodos para seleccionar los elementos que conforman la propuesta (Ruiz et al., 2018).

Para generar una visualización del concepto anterior se toma en cuenta un diseño de interfaz con tres elementos idénticos distribuidos de manera equitativa y triangular. Hasta este punto, ya se ha manejado brevemente las posibles influencias que tienen un usuario ideal ante elementos idénticos, sin embargo, en este caso es preciso abordar que, si bien dos usuarios pueden tener recorridos específicos diferentes, las transiciones entre cada elemento son las mismas.

Mencionado lo anterior, las figuras 32 y 33 representan dos usuarios ideales que recorren los elementos. El primero caso al tener un punto inicial aleatorio más cercano al elemento B el usuario ideal uno realiza su recorrido particular a partir del elemento mencionado, contraparte a lo anterior, el usuario ideal dos inicia su recorrido a partir del elemento A.

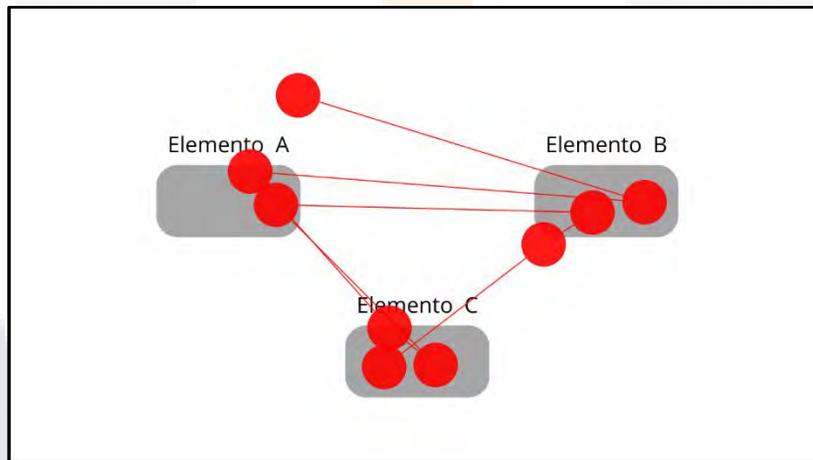


Figura 32: Visualización del recorrido del usuario ideal uno en una interfaz con elementos idénticos distribuidos de forma triangular, (Elaboración propia, 2024).

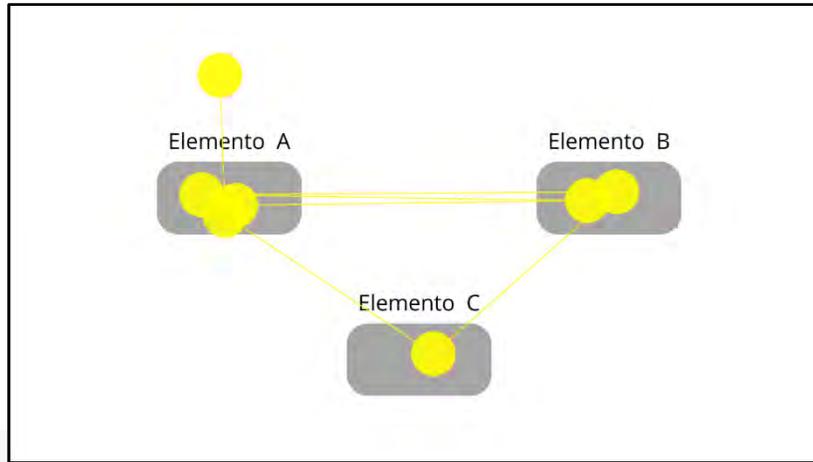


Figura 33: Visualización del recorrido del usuario ideal dos en una interfaz con elementos idénticos distribuidos de forma triangular, (Elaboración propia, 2024).

Lo más representativo de la distribución de los elementos surge al representar ambas visualizaciones de cada usuario ideal en una misma figura. La figura 34 representa los dos recorridos de cada usuario ideal. Es posible observar que, si bien el orden de recorrido fue diferente para cada elemento, e incluso el conteo de fijaciones por elemento difiere, las transiciones entre cada elemento son muy evidentes y están reclinadas a seguir minuciosamente cada elemento de la manera más efectiva posible, es decir, en ningún momento la mirada del usuario se retiene en un punto medio del recorrido.

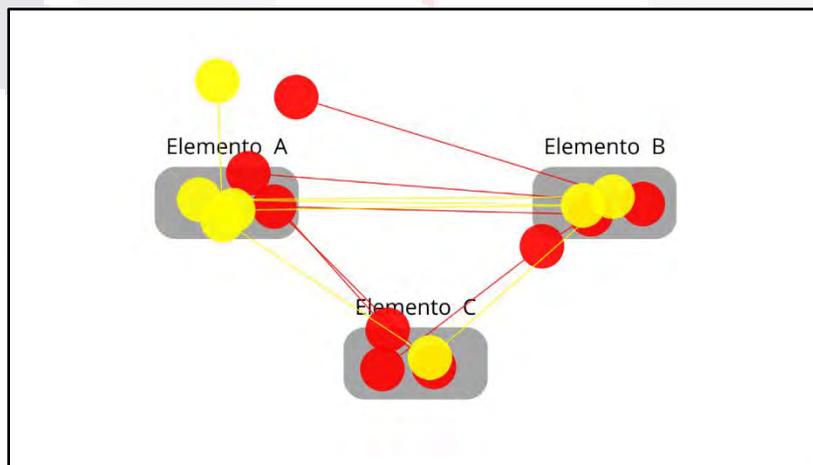


Figura 34: Visualización del recorrido de ambos usuarios ideales en una interfaz con elementos idénticos distribuidos de forma triangular, (Elaboración propia, 2024).

f) Inferencia y análisis de los recorridos de atención visual.

La característica anteriormente descrita e ilustrada en la figura 32 corresponde al movimiento ocular y sus mecanismos de rastreo, el cual está íntimamente relacionado con la identificación de objetos. Según investigaciones, los movimientos oculares tienden a saturarse después de los 30 grados sobre segundo (Parra, 2004), sin embargo, este valor hace referencia a la velocidad con que el sistema de músculos oculares, así como los demás elementos del ojo humano como la fovea, retina e iris, ya no tienen la capacidad de reaccionar y reconocer el objeto en movimiento.

Ahora bien, cuando se presentan elementos estáticos en una pantalla bidimensional es mucho más sencillo realizar los movimientos oculares ya que no existe una presión inherente al reconocimiento de una interfaz, es por esto por lo que las fijaciones indican en un análisis de uso de interfaces los puntos que el usuario decidió usar más tiempo para retener la mirada en búsqueda de información que le permita encontrar el mejor recorrido.

Conforme las propuestas de interfaces se hacen cada vez más complejas, existe la necesidad hacer una unión de todos los posibles enfoques que pueden afectar el comportamiento de un usuario al usar un diseño. Anteriormente, cuando los sitios web tendían a ser estáticos sería muy sencillo evaluar las capacidades de los elementos para ser usados, ya que se conformaban principalmente de secciones de texto o imágenes. Hoy en día cada página web contiene decenas de elementos, cada uno con diferente dimensión, distribución, color y tipo de fuente, entonces surge la pregunta: ¿cómo podemos analizar tan distintos tipos de elementos?

A pesar de lo prudente que suena la pregunta anteriormente formulada, es posible generar una evaluación mucho más interesante si se plantea la pregunta desde un punto de vista enfocado a preguntar sobre los objetivos de uso de la interfaz, en lugar de detenerse en los elementos individuales.

El enfoque anteriormente mencionado permite generar una diferenciación entre observar los distintos patrones de uso a reconocer si los elementos están realizando su labor en conjunto para indicar al usuario una secuencia clara y le permita completar procesos específicos dentro del uso de la interfaz.

El “cómo” se está utilizando la interfaz resulta ser información más valiosa que el preguntarse “¿qué se puede hacer para mejorar?”, ya que representa de manera general el entendimiento que un grupo de usuarios muestra en la interfaz. Ya sea si existe una divergencia enorme entre los comportamientos de uso, así como sean todos homogéneos, conocer el objetivo de la interfaz en el sistema permite inferir de mejor manera cómo el usuario está interpretando la información contenida en ella.

Al final de una interacción, y sobre todo en interacciones largas, es preciso generar en el usuario la sensación que se han cumplido un montón de objetivos, más no de obstáculos para completar una tarea. De ahí la importancia de verificar que las posibles secuencias de uso estén respaldadas de un recorrido claro, repetible, diferenciable e, intentar, garantizar el cumplimiento del objetivo del usuario.

Una vez revisados los elementos más generales de las posibles visualizaciones que se tienen al usar una interfaz cabe destacar que muchos de los procesos más complejos de la atención visual en interfaces se siguen estudiando en muchos trabajos de investigación. Parte del trabajo futuro de esta investigación es ampliar el reconocimiento de patrones a modelos estandarizados.

En la siguiente sección se aborda un enfoque mucho más individual para el análisis de los elementos en pantalla, lo cual busca ampliar la capacidad del marco de trabajo EVUSINET para la evaluación objetivas de interfaces.

B. Análisis de Patrones de Uso Particulares

El marco de trabajo EVUSINET busca aplicar la mayor cantidad de técnicas revisadas para implementar las técnicas de eye-tracking en la mayor cantidad posible de procesos de desarrollo de software. Ya se ha mencionado anteriormente que parte importante de la validez de un análisis de eye-tracking depende del criterio por parte del investigador.

Para representar de mejor manera la implementación de los índices propuestos, los resultados y la manera en que es posible congregar la información se expone a continuación un proceso de experimentación mediante técnicas de eye-tracking sobre un diseño de interfaz con elementos mínimos, este diseño es posible consultarlo en la figura 13 contenida en la sección 3.5.2 apartado A.

Para generar un análisis de una interfaz el primer paso es establecer los parámetros esperados de prueba, es decir, se debe indicar la interfaz de estudio, la cantidad de elementos e identificarlos de manera única, tiempo estimado de la interacción de la interfaz, el tipo de tarea a resolver, la relevancia esperada de cada elemento y los objetivos principales que busca resolver el diseño.

Por lo tanto, la tabla 7 recopila los parámetros esperados del diseño de interfaz de la figura 13:

Tabla 7: Tabla de expectativas de uso para la prueba simulada de evaluación de eye-tracking, (Elaboración propia, 2024).

Parámetro esperado de evaluación	Descripción
Interfases de Estudio	La interfaz de estudio es denominada Interfaz con elementos mínimos. Expresándolo usando la notación propuesta de INUSINET: <i>I</i> → <i>Interfaz con elementos mínimos</i>
Cantidad de elementos	3 elemento reconocibles: Título del contenedor. Contenedor. Botón. Usando la notación INUSINET: <i>I</i> (1) = <i>Título del contenedor</i> <i>I</i> (2) = <i>Contenedor</i> <i>I</i> (3) = <i>Botón</i>
Tiempo estimado de la interacción completa.	Ya que se plantea una interfaz de estudio que no comprende parte de una tarea mayor se establece un tiempo de interacción fijo de 30 segundos.
Tipo de tarea	Formulario/Diseño de interfaz propuesto con fines de análisis y pruebas de eye-tracking.
Relevancia esperada de cada elemento	Los elementos de este diseño no cuentan con grandes diferencias entre dimensión o color, sin embargo, tienen una distribución particular. En este caso se espera que el título del elemento sea el menos visitado ya que el recorrido se estima que sea descendente. Posteriormente, el elemento de contenedor sea el más visitado por ser ligeramente más grande que el botón. Por lo tanto, la relevancia esperada de los elementos es la siguiente: – Relevancia de <i>I</i> (1) esperada: 20% de las fijaciones. – Relevancia de <i>I</i> (2) esperada: 50% de las fijaciones.

	– Relevancia de $I(3)$ esperada: 30% de las fijaciones.
Requerimientos de la interfaz	La interfaz está planteada como un formulario de experimentación en la que se supone que el contenedor tendrá algún tipo de información, sin embargo, el requerimiento a cubrir en este análisis es verificar si la relevancia real obtenida por eye-tracking corresponde a la esperada.

La tabla 7 contiene una hoja de ruta para que el evaluador sesgue la interpretación de resultados que surge al exponer la interfaz de estudio a las pruebas de eye-tracking. En la sección 3.7.1 se aborda a mayor profundidad las técnicas que permiten implementar las técnicas de seguimiento de mirada sobre interfaces. Para esta sección que busca ejemplificar la implementación de INUSINET se generaron tres pruebas de eye-tracking utilizando una alta rigurosidad en el control de calidad en cuánto a protocolos y en el uso de las herramientas de eye-tracking con las que se realiza la investigación (TobiiProNano y TobiiProLab).

Como se mencionó anteriormente, se realizaron tres pruebas sobre la interfaz de estudio de la figura 11. Estas pruebas se realizaron con un único usuario que tenía la indicación de recorrer cada elemento de la manera más detallada posible en el orden que más le pareciera conveniente, al no mencionarle el límite de tiempo ni poner a la vista ningún cronómetro el usuario no tenía ninguna noción del momento en que concluiría la prueba, por lo que se puede estimar que las métricas de eye-tracking no están sesgadas por algún tipo de influencia en el usuario al presionarlo con un reloj o, de manera no voluntaria, repartir la mirada entre los elementos dependiendo del tiempo asignado.

En la figura 35 se observan los resultados de manera visual de las métricas que fue posible obtener después de las tres pruebas de 30 segundos cada una. Cada una de las pruebas está remarcada por un color diferente, también cabe destacar que si bien es complicado representar fácilmente la cantidad y calidad de información proporcionada por las técnicas de eye-tracking, mediante la visualización es posible seguir la secuencia de fijaciones, así como la duración de estas y las características de los movimientos sacádicos.

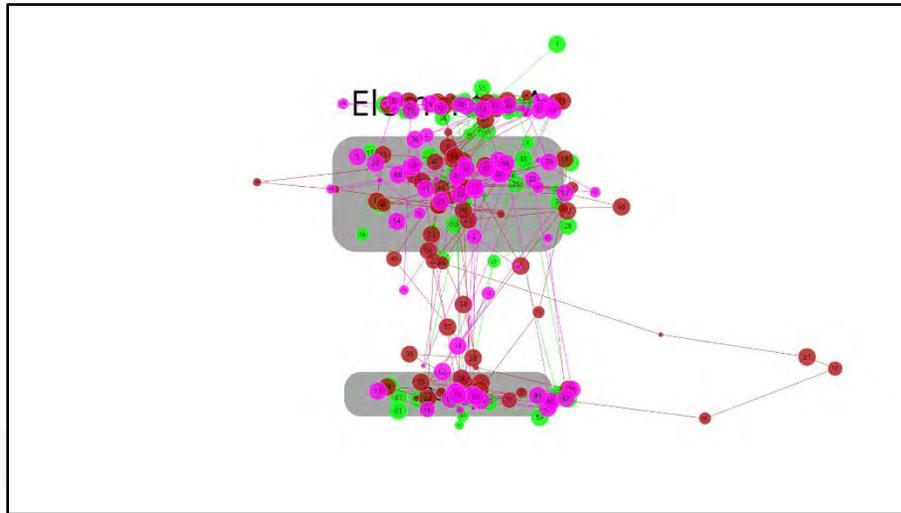


Figura 35: Resultados de tres pruebas de eye-tracking de fijaciones y movimientos sacádicos sobre la interfaz de estudio, (Elaboración propia, 2024).

A pesar de que la información contenida en la figura 35 es muy útil para que el evaluador se dé una idea general sobre el uso del sistema, la mejor manera de analizar las propiedades de los elementos es agrupar los valores obtenidos por eye-tracking de cada elemento mediante una tabla. Dado que la prueba fue realizada con una interfaz en la cual no era posible interactuar se generan los siguientes supuestos para establecer el número de interacciones de cada elemento:

- Siguiendo la teoría del caso cero de interacciones el elemento del título del contenedor se determina que este no está diseñado para recibir interacciones, por lo que $Int(1) \rightarrow \emptyset$.
- Se interactuó dos veces con el contenedor, por lo que $Int(2) = 2$.
- Se interactuó dos veces con el botón, por lo que $Int(3) = 2$.

Es preciso destacar que esta implementación de las interacciones no es válida en una interfaz real, ya que como se ha mencionado anteriormente, el proceso de interacción requiere un esfuerzo mayor en cuanto a la atención visual y cognitiva que influye en el registro obtenido para las variables de cantidad de fijaciones (Cf), cantidad de sacádicos (Cs), duración de fijaciones (Df) y más evidentemente en la cantidad de interacciones (clicks) (Int).

a) Tablas de resultados

A continuación, se despliegan los resultados en diferentes tablas, en la tabla 8 se expresan los resultados de cada prueba y cada elemento, este tipo de registro de los datos tan particular pueden ayudar a encontrar un patrón de uso que sea muy divergente de los demás usuarios.

Tabla 8: Recopilación resultados para cada prueba realizada mediante eye-tracking de la interfaz de estudio, (Elaboración propia, 2024).

Prueba	Elemento de I	Cf(n)	Σ Df(n)	Int(n)	Cs(n)	Vs(n)
1	I(1)	12	5379	0	7	96.295
	I(2)	30	10459	2	19	139.86
	I(3)	12	3904	2	7	127
2	I(1)	20	7095	0	15	76.055
	I(2)	24	8394	2	16	218.625
	I(3)	18	11009	2	14	334.37
3	I(1)	20	5679	0	15	129.945
	I(2)	32	10538	2	19	127.745
	I(3)	20	7528	2	14	114.85

Por otra parte, la tabla 9 representa el promedio de los resultados obtenidos para cada elemento de las tres pruebas. Cabe resaltar que recopilar los datos de esta manera más resumida resulta mucho más objetiva para analizar el promedio de uso de una población de usuarios.

Tabla 9: Resultados por elementos de la interfaz de estudio, (Elaboración propia, 2024).

Elemento de I	Cf(n)	Df(n)	Int(n)	Cs(n)	Vs(n)
I(1)	17.33	6051	0	12.33	100.77
I(2)	28.67	9797	1	18	162.08
I(3)	16.67	7480.33	2	11.67	192.07

Una vez que se han obtenido los valores de las métricas de eye-tracking se aplican los índices propuestos de INUSINET para conocer las características de manera objetiva de cada elemento de la Interfaz I. En la tabla 10 es posible encontrar los valores generados al evaluar la interfaz de estudio.

Tabla 10: Valores cuantitativos de pruebas de eye-tracking usando INUSINET, , (Elaboración propia, 2024).

Elemento de I	ICE	ICU	ICR
I(1)	0.00286454	Caso cero (\emptyset)	0.122396996
I(2)	0.002926066	14.33333333	0.111058553
I(3)	0.002228065	8.333333333	0.060740689

b) Interpretación de resultados.

Hasta este punto del análisis se tienen valores cuantitativos de las pruebas de eye-tracking realizadas usando los índices de INUSINET. Es a partir de este punto que el evaluador debe realizar interpretaciones referentes a mencionados resultados. Si se observa el ICE, que hace referencia a la complejidad de entender un elemento en la pantalla) se distingue que el elemento más bajo en cuanto al valor numérico es el I(3). Esto hace referencia a que el objetivo del elemento no fue del todo claro, ya que el proceso de obtención de la información no fue claro para el usuario.

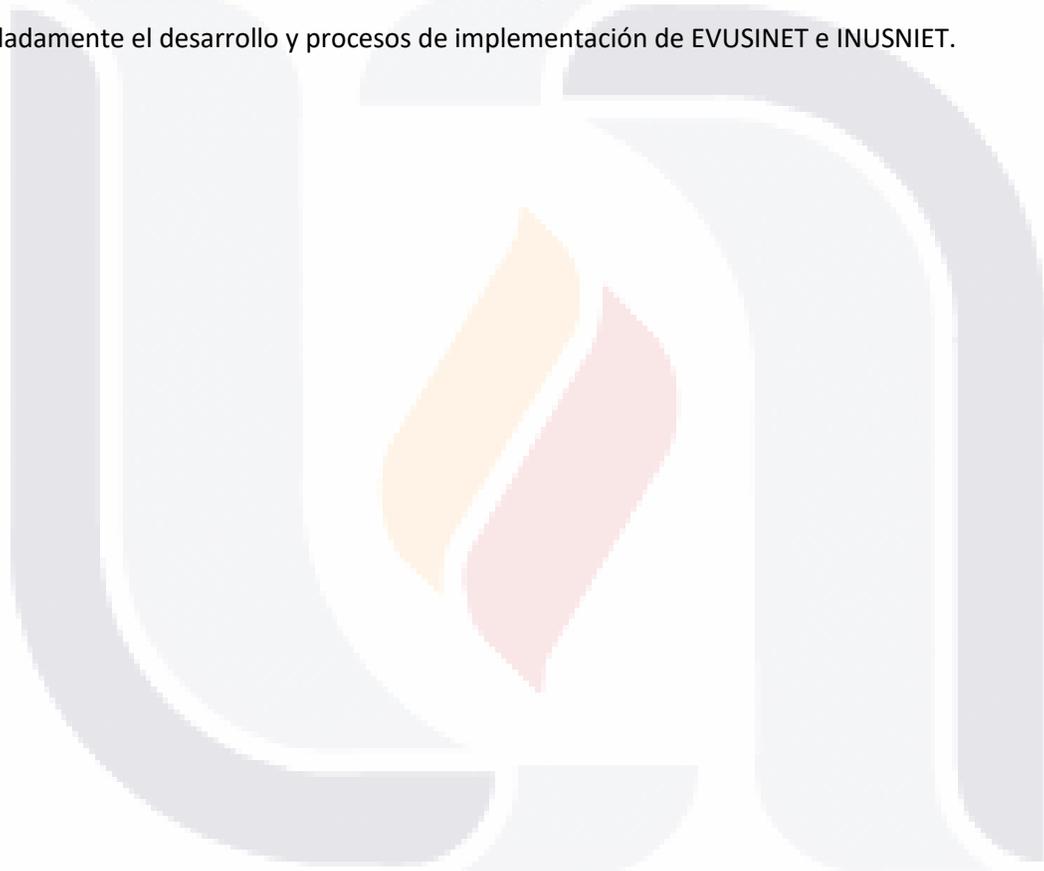
Otro caso interesante es el ICU, en el cual, si bien en los elementos I(2) e I(3) se estableció un mismo valor para la cantidad de interacciones, el índice muestra valores muy contrarios. En este caso, el elemento en el que se puede distinguir un peor uso es el elemento I(3).

Por último, el ICR presenta una información muy relevante que puede relacionarse con los resultados obtenidos en los demás índices, en este caso, el valor cuantitativo del ICR de I(3) es el mayor, seguido del elemento I(1), ya que en estos elementos contienen texto, lo cual, cambia drásticamente la lectura obtenida en la variable Vs, relacionada al esfuerzo de lectura. En este valor indica que el elemento I(3) contienen un texto que es fácil de leer, ya que el movimiento rápido de los movimientos sacádicos indica que el recorrido realizado fue eficaz, por otro lado, el elemento con peor ICR es el I(2) ya que al no contener texto el usuario mueve de manera arbitraria la vista, reduciendo la velocidad de los movimientos sacádicos.

Como se constató en la última sección, las propuestas de EVUSINET e INUSINET tienen diferentes mecanismos para utilizar las técnicas de eye-tracking en distintos entornos de desarrollo, tipos de pruebas, características de diseño, entre otros aspectos de la ingeniería de software. Si bien las

propuestas están fundamentadas de manera rigurosa no se descarta la modificación y experimentación de las propuestas y conceptos aquí abordados. Se considera que para avanzar realmente en el análisis objetivo de interfaces y poder hacer pruebas válidas requiere de muchos casos de estudio que permitan obtener resultados más precisos, mientras se expanden los entornos de implementación de estas tecnologías para fomentar un nuevo tipo de evaluación de software.

Abordando lo anterior, alrededor de esta investigación se han desarrollado casos de estudio en los cuales se probaron las técnicas, métodos y se establecieron los procedimientos que se encontró que mejor se adaptaran al tipo de análisis que la investigación plantea. En la sección siguiente se aborda detalladamente el desarrollo y procesos de implementación de EVUSINET e INUSNIET.



4. Desarrollo

Después de repasar a detalle las distintas metodologías de desarrollo y posibles enfoques para generar un análisis objetivo de interfaces de manera teórica, es conveniente aplicar las teorías para revisar la validez que estas tienen en la práctica.

En la sección de metodologías se propusieron distintos conceptos, todos relacionados con la implementación de las técnicas de eye-tracking en los procesos de diseño, desarrollo y evaluación de las interfaces, es por esta misma razón que los casos de estudio están planteados para poner a prueba las propuestas antes desarrolladas.

Cada caso de estudio busca identificar las fortalezas y debilidades de los modelos propuestos, sin embargo, al ser una primera implementación de los métodos propuestos, estos aún están sujetos a cambios y mejoras para futuros trabajos de investigación.

Retomando lo abordado en la sección 3.4 referente a la metodología propuesta de interfaces paralelas se hace hincapié en las capacidades del eye-tracking como una herramienta que se adhiere al proceso de diseño y desarrollo a la par que se aplican evaluaciones para comprobar el comportamiento del usuario de manera objetiva. Recordando levemente el proceso, se propone que al iniciar el proceso de desarrollo de una interfaz se propongan dos diseños que tengan el mismo propósito de uso y los mismos elementos, sin embargo, la distribución, dimensión, diseño, colores y fuentes de cada elemento difiere en cada propuesta. El diseño de este modelo de diseño de interfaces permite al diseñador conocer de primera mano el impacto que cada elemento tiene en la experiencia del usuario sobre el diseño.

La evaluación iterativa de las interfaces paralelas permite homogeneizar los diseños con cada iteración aplicada, es decir, si el diseñador y evaluador notan que un elemento responde de manera positiva ante los usuarios, este elemento pasa el filtro y permanece en la siguiente propuesta de diseño. Extendiendo las capacidades del concepto anterior, la interfaz contraria (o gemela) de igual manera puede proveer elementos que resultan exitosos en la prueba de uso con los usuarios, de tal manera que el diseño de interfaz final contiene los mejores elementos de cada propuesta de diseño.

Una vez mencionado el enfoque de evaluación preventiva o activa en el proceso de diseño de interfaces, es preciso mencionar que el segundo caso de estudio se centra en la evaluación

correctiva de las capacidades de la interfaz para ofrecer una experiencia efectiva en distintos tipos de usuarios.

El segundo enfoque de aplicación de las metodologías de análisis objetivo de interfaces permite conocer a detalle el comportamiento de un grupo de usuarios específicos, con el fin de generar un modelo de interfaz que cubra sus necesidades de uso.

Para revisar las metodologías propuestas para cada caso de estudio revise la sección 3.2 referente a la metodología tradicional relacionada al primer caso de estudio y la sección 3.3 que contempla la metodología ágil para el segundo caso de estudio.

4.1. Caso de estudio 1: Componente Login

Para encontrar un caso de estudio que pueda cumplir las características de un componente y permita generar un desarrollo independiente y flexible a cambios es preciso encontrar una funcionalidad que se use en la mayoría de los sistemas actualmente. Estableciendo lo anterior, durante varios años se ha manejado el concepto de formulario para definir un proceso muy específico de interacción con el usuario que obliga a este a introducir datos específicos en un conjunto de elementos de recolección de información (Seckler et al., 2014).

Llenar un formulario debe entenderse como un escenario que el usuario preferiría evitar. Es en este punto que el usuario debe tener todas las facilidades de uso para completar la tarea en el menor tiempo posible. La dificultad agregada a un proceso de llenado de campos de información puede influir muy negativamente en la impresión general del sistema (Saquero Ros & Berná Martínez, 2019). Es importante destacar que el usuario tiende a perder el interés con facilidad sobre todo si los procesos son rutinarios, poco claros y con un flujo de interfaces caótico.

Ampliando lo anterior, en la sección 2.2.4 se hace referencia al proceso de interacción con sistemas. En esta sección se enfatiza en que este proceso es complejo y requiere un gran conjunto de procesos cognitivos y visuales para poder completar una tarea correctamente. Prever y verificar que el usuario cumple las características de uso esperadas es muy complicado utilizando métodos de evaluación de interfaces convencionales.

Ahora bien, el método propuesto de diseño de interfaces mediante las interfaces paralelas busca específicamente proveer una herramienta de constante escrutinio de las capacidades de cada

elemento en pantalla. Es por lo anterior, y aunado a la idea de evaluar un componente muy general en la mayoría de los sistemas que se propone desarrollar un componente de inicio de sesión en el cual se apliquen las propuestas de interfaces paralelas, EVUSINET e INUSINET.

4.1.1 Características del sistema

Una vez seleccionado el caso de estudio, y tomando en cuenta las ventajas estipuladas en el apartado de antecedentes, se seleccionó para el desarrollo del sistema el lenguaje de programación *Java*. Además, cabe resaltar que varias de las metodologías abordadas en esta investigación fueron diseñadas basadas en el desarrollo del lenguaje de programación orientado a objetos. Es importante recordar que el sistema de ingreso de usuarios pasará por varias iteraciones para proponer diseños hasta que se verifique que las características de las interfaces sean las más agradables para el usuario a la par que optimicen el uso de cada elemento.

Tomando en cuenta que las pruebas que se aplicarán son principalmente mediante eye-tracking, se espera que los patrones de uso sean altamente diferenciables entre sí. Expandiendo la idea de un desarrollo en interfaces paralelas, la primera etapa de desarrollo general debe proponer diseños con dos funcionamientos esperados idénticos, pero con diferentes propiedades de cada elemento.

El primero tomará un enfoque de interacción dinámica y que minimice el texto en pantalla. Este aproximamiento a interfaces dinámicas se propone después de revisar los sistemas de login de varias empresas como *Google*, *Zoom*, *DeepL translator*, entre otras cuyo enfoque es reducir al mínimo la dificultad del proceso de verificación. Una de las observaciones que se hicieron al analizar las plataformas antes mencionadas, es que este tipo de interacción tiene mayor inclinación por métodos de interacción visuales. Disminuyen los lugares al usuario debe prestar cambiando el texto que despliega la siguiente instrucción o paso del proceso. Para informar al usuario que los datos que están siendo ingresados son correctos o erróneos hacen uso de íconos o advertencias que, dependiendo del contenido en tiempo real pueden ser visibles o desaparecer de pantalla en “tiempo real”.

La segunda interfaz representa un modelo más convencional y apegada a estándares de uso. Antes probados, sin embargo, el diseño propuesto para la interfaz del tipo formulario propone elementos estáticos y acompañados por texto que indiquen al usuario los contenidos de cada bloque de texto. También se destaca que la confirmación al usuario para indicarle que los datos son válidos o

incorrectos es a través de una etiqueta de notificación y cambia únicamente al intentar enviar el formulario. Este componente de texto permitirá comparar los íconos visuales y que tan claros son para el usuario.

4.1.2 Proceso de desarrollo de software del componente login

Una vez planteados las características principales que conforman el contexto del caso de estudio se comparten en esta sección tablas, herramientas y listados de los procesos por a los que se sometió el software para ser producido. Es preciso mencionar que en esta sección en la que se desarrolla un sistema con pocos módulos e interfaces se opta por utilizar una metodología de desarrollo convencional y de manera más específica, la descrita en la sección 3.2.

A. Objetivos

Una de las principales características del sistema es que está planteado como un software de experimentación que le permita al investigador tener control sobre cada una de las herramientas, elementos y secuencias de interacción. Si bien, muchas de las teorías propuestas en los marcos de trabajo EVUSINET y el modelo de evaluación de interfaces INUSINET fueron implementados en este proceso de desarrollo, también el proceso permitió fundamentar y probar con diferentes mecanismos de aplicación de la tecnología, observando las ventajas, desventajas puntos fuertes y débiles y también estimar tiempo y esfuerzos requeridos para llevar a cabo un análisis de eye-tracking. Es por esta razón, que parte de los objetivos que se plantearon alrededor de este desarrollo corresponden a lograr una experimentación homogénea, válida y que permita eliminar sesgos tanto del investigador, como del usuario al generar las pruebas. En la tabla 11 se expresan los objetivos generales y particulares del componente login.

Tabla 11: Objetivos del componente login, (Elaboración propia, 2024).

Objetivo general	
Desarrollar un componente que permita probar técnicas de implementación de eye-tracking para obtener datos biométricos para evaluar el uso de la interfaz y proponer mejoras en función de los resultados obtenidos.	
Objetivos específicos	Prioridad.

Desarrollar una interfaz para la interacción humano-sistema clasificando los elementos y sus propiedades.	Obligatorio
Aplicar técnicas de eyetracking para estudiar el comportamiento de la mirada al usar el sistema y la relación que existe entre las propiedades de la interfaz.	Obligatorio
Encontrar el análisis de datos de eyetracking óptimo para establecer las relaciones con los elementos y propiedades de la interfaz.	Deseable
Desarrollar un esquema de protocolo experimental y procedimiento de registro de pruebas tomando como experiencia previa los problemas técnicos o de análisis vistos en el análisis del módulo.	Obligatorio

B. Requerimientos

Consiguiente a los objetivos, los requerimientos del componente login están enfocados en la elaboración de un software que permita experimentar con las tecnologías de eye-tracking. Por lo que, además de seguir la convención de estipular requerimientos, funcionales y no funciones, el desarrollador debe tomar en cuenta el objetivo principal del software. En las tablas 12 y 13 se expresan los requerimientos que se establecieron para el producto de software.

Tabla 12: Requerimientos funcionales del componente login, (Elaboración propia, 2024).

Clave.	Descripción del requerimiento funcional.	Fecha.
R1	Acceso al sistema mediante la validación de usuario con los registros de una base de datos.	12/02/23
R2	Interfaz para la creación de un usuario y contraseña (Alta en base de datos).	Idem.
R3	Cada elemento cumple una función específica dentro de la interfaz.	“
R4	Los elementos deben ser claros, con buena dimensión y con una propuesta definida para las fuentes y tamaños de texto.	“
R5	Proporcionar un selector de imagen de perfil en cada propuesta de diseño.	“

Tabla 13: Requerimientos no funcionales del componente Login, (Elaboración propia, 2024).

Clave.	Descripción del requerimiento no funcional.	Fecha.
NF1	La interfaz responde al diseño óptimo para la interacción humano-sistema.	13/02/23
NF2	El componente debe ser totalmente independiente y su diseño puede ser implementado a una amplia variedad de sistemas.	Idem.
NF3	La base de datos puede ser simulada.	"

C. Diagramas

Ya definidos los requerimientos del software, es preciso generar una serie de diagramas que permitan al desarrollador planear detalladamente los esfuerzos necesarios para producir el producto.

a) Diagrama de arquitectura de alto nivel.

Siguiendo las convenciones de UML, el primer diagrama generado es el de alto nivel, el cual, y de manera particular del caso de estudio, muestra el entorno en dónde se plantea ejecutar el sistema. Como se observa en el elemento de color verde de la figura 36, el caso de estudio del componente login se plantea como un software externo que se consume desde distintos sistemas.

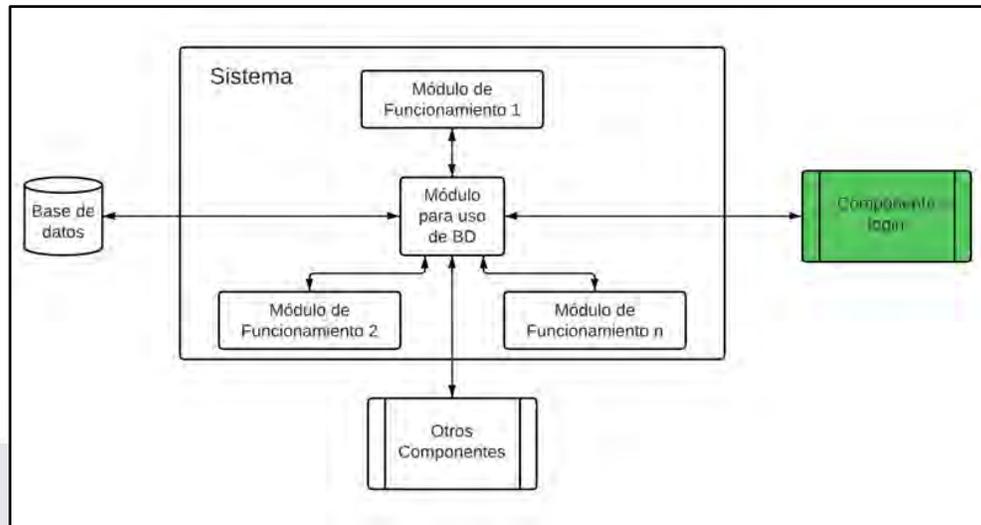


Figura 36: Diagrama de alto nivel del componente login, (Elaboración propia, 2023).

b) Diagrama de contexto.

Este diagrama establece que este propio sistema externo puede tener su propio módulo para manejo de base datos que el mismo componente login puede consumir para facilitar la tarea de validación y reducir los tiempos de respuesta. También se establece que el tipo de validaciones que es posible determinar mediante el componente login responde al concepto de “Entidad: Usuario” que puede responder a las sesiones de algunos lenguajes de desarrollo web, una clase en POO o incluso un correo electrónico ya que cuenta con los campos mínimos para validar al usuario. Esta entidad permite estandarizar los parámetros de entrada y respuesta mínimos para poder resolver la petición del sistema que usa el componente login (véase figura 37).

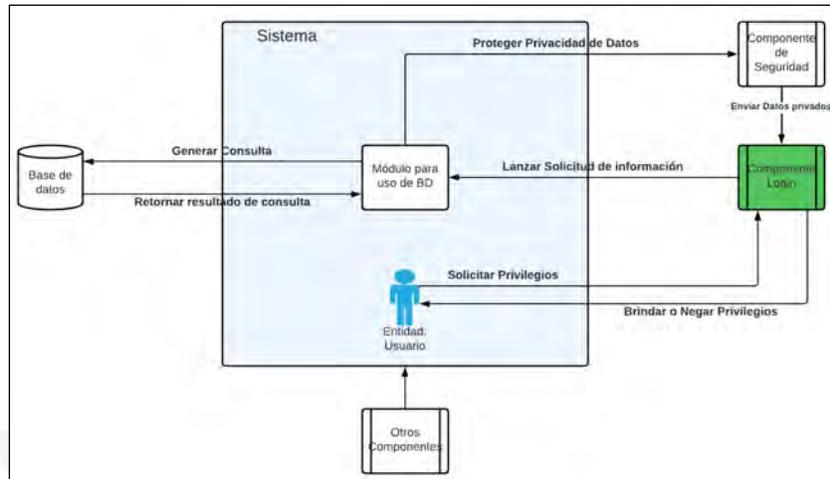


Figura 37: Diagrama de contexto del componente login, (Elaboración propia, 2023).

c) Diagrama de Contenedores

Como se observa en la figura 38, la idea de que el componente se comunica con los demás servicios de la manera en que se transfieren la menor cantidad de datos e información entre componentes externos. También en este diagrama se representan a grandes rasgos los subcomponentes que forman parte del caso de estudio. En esta misma figura, en la parte superior izquierda es posible observar el concepto de entidad usuario que, como se menciona anteriormente, representa la abstracción de los datos mínimos requeridos por el solicitante del servicio para validar y dotar de privilegios a su usuario.

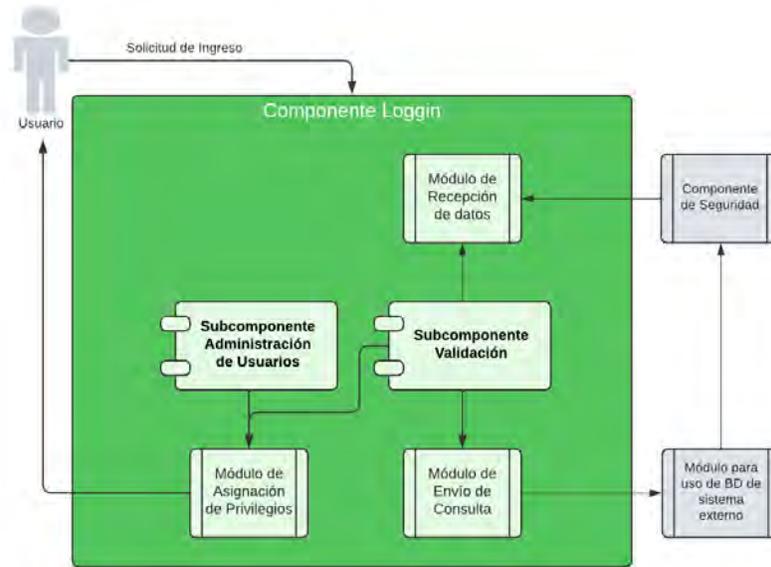


Figura 38: Diagrama de contenedores del componente login, (Elaboración propia, 2023).

d) Diagrama de Componentes

En el diagrama de componentes se establecen de manera más detalladas las relaciones y flujo de datos entre los distintos subcomponentes, además de que se expresan las interfaces que son desplegadas para los mencionados subcomponentes (consúltese figura 39).

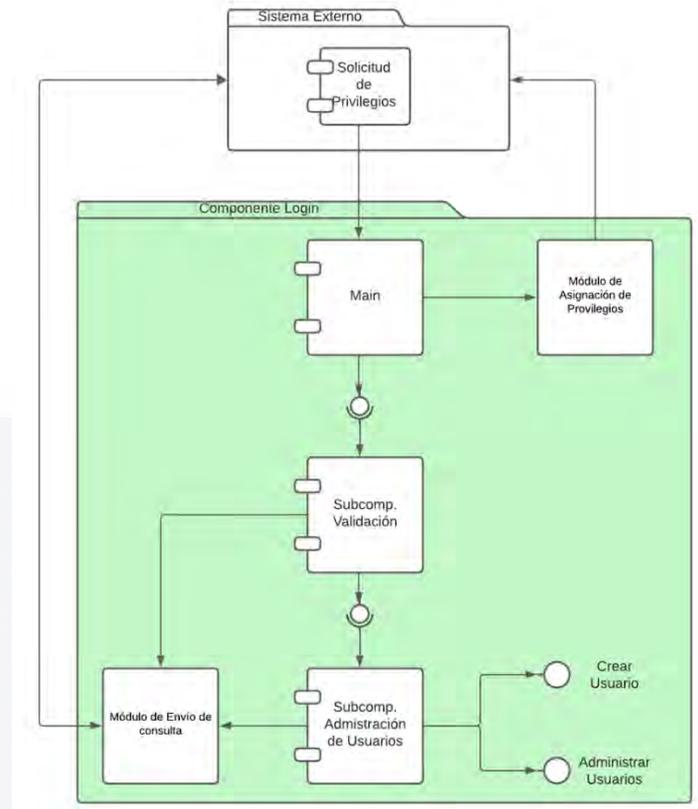


Figura 39: Diagrama de componentes y subcomponentes del componente login, (Elaboración propia, 2024).

e) Diagrama de clases

Para un desarrollo utilizando el paradigma orientado a objetos es muy útil generar el diagrama de clases. En el caso del componente login se generó el diagrama que está representado en la figura 40 en dónde, además de especificar las clases concretas que dan forma al proceso para generar las validaciones, se denotan a grandes rasgos las variables involucradas en la codificación de los elementos. Cabe resaltar que muchos de los submódulos de funcionamiento con las bases de datos son, para este caso de estudio, simuladas. Contrario a lo anterior, el diseño de las propuestas de interfaces y los patrones de uso esperado están mucho más trabajados para replicar las condiciones de uso de formularios de creación de usuario que existen en el mercado actualmente.

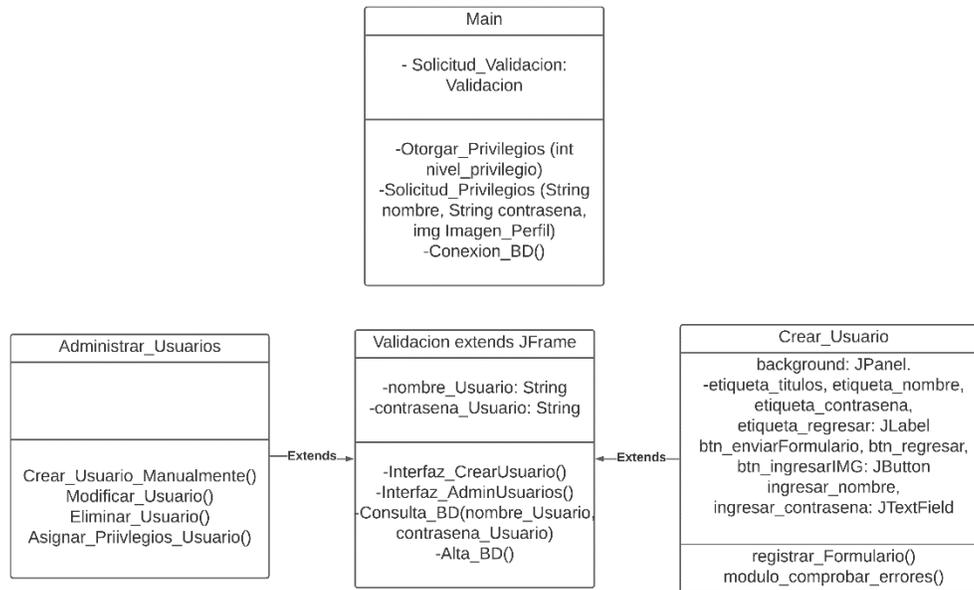


Figura 40: Diagrama de clases del componente login, (Elaboración propia, 2024).

Una vez abordados algunos de los diagramas y procesos en la etapa de análisis del componente login, se procede a diseccionar las características de las interfaces que formarán parte de las pruebas experimentales. Como se ha observado en los diferentes diagramas, es imprescindible que el componente login sea una herramienta que pueda consumir una aplicación, lo que enfatiza que las características de uso deben ser las adecuadas para que un mayor número de personas puedan usarlas.

4.1.3 Propiedades de las interfaces

En esta sección se establece la fundamentación de los diseños propuestos para generar un análisis siguiendo la metodología de diseño de interfaces paralelas. Retomando que el componente login fue planteado como un servicio que puede ser consumido por distintas plataformas o sistemas externos es un caso de estudio que orienta el uso de sus interfaces para que gran cantidad de tipos de usuarios puedan usarlas.

En los siguientes apartados se describe las características más destacables de las interfaces que serán sometidas a pruebas para evaluar la capacidad de éstas para ser usadas, con el fin de encontrar patrones de uso y atención visual que indiquen que tan agradable para el usuario es la

disposición, dimensión y diseño de los elementos, entre otros factores que nos ayudarán a estimar el uso del software mediante las propuestas de EVUSINET e INUSINET mediante.

A. Propuesta de Interfaz Dinámica

La interfaz 1 se enfoca en reducir el texto en pantalla. Cambiando o reaccionando en tiempo real a los datos introducidos por el usuario en cada momento. Cabe destacar que los íconos, además de un significado inherente también se usa la teoría de asociación de colores para reforzar el estímulo. Cómo se observa en la tabla 14, el tipo de ícono está relacionado con un color en específico.

Tabla 14: Asociación de íconos y colores, (Elaboración propia, 2023).

Ícono	Significado de color
!	Las advertencias suelen expresarse de un color rojo, siendo este tono el que más resalta en la vista humana.
✓	Los tonos claros son aplicados en confirmaciones o validaciones debido a que este color no se relaciona a escenarios de alerta o peligro.
◀◀Regresar	El tono gris se considera un tono neutral o de espera por lo que, inconscientemente el cerebro ignore este elemento a primera vista. Dado que la función esperada del elemento "Regresar" es que sea usada solo en caso de error de navegación o falla en el recabado de datos, no es necesario brindarle un color llamativo, por lo que el tono gris es el que más se apega al comportamiento esperado para el botón.

Recordando que se busca generar un formulario dinámico y reducir al mínimo los elementos de texto del sistema, se usará una disposición central de ingreso de datos, mientras que, en el apartado

izquierdo, encapsulado en otro panel se encuentra una vista previa del usuario a generar. Este diseño propuesto trata de evaluar la cantidad mínima de información necesaria para que el usuario comprenda y confirme los datos que ingresó.

El bloque central de ingreso de datos cuenta con los elementos mínimos indispensables de interacción, contando únicamente con un texto de instrucciones dinámico, la barra de ingreso de nombre de usuario y el par de recuadros de texto para ingresar y confirmar la contraseña. Una vez que todos los campos son llenados, en el apartado de confirmación de datos del usuario se despliega el ícono de confirmación indicando que los datos son congruentes y válidos. Es en este punto que se habilita el botón de enviar formulario. Una vez explicado el funcionamiento esperado de la Interfaz Dinámica de Creación de Usuario, el diseño planteado se muestra en la figura 41 indicando con la siguiente numeración los nombres de los elementos a los que hacen referencia.

Numeración de Elementos:

1. Título dinámico
2. Campo de texto de Nombre de Usuario.
3. Campo de texto Ingreso de contraseña.
4. Campo de texto de Confirmación de contraseña.
5. Botón de Enviar Formulario.
6. Área de retroalimentación dinámica de estado del usuario.
7. Botón regresar.

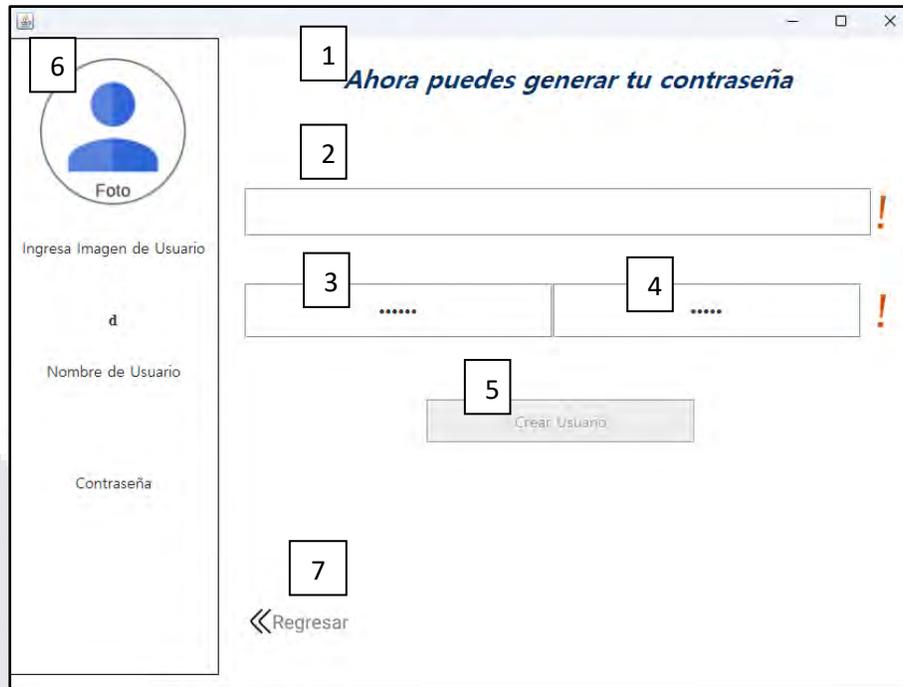


Figura 41: Interfaz Dinámica de Creación de Usuario
 Fuente: (Elaboración propia, 2023).

B. Propuesta de Interfaz de Formulario Convencional

Por otro lado, la interfaz basada en los formularios convencionales tiene un diseño mucho más sencillo y, a su vez, el número de elementos no está pensado para minimizarse sino para acompañar a los cuadros de texto. A diferencia de la interfaz dinámica, el único elemento que cambia dependiendo de la interacción con el usuario es la etiqueta de control de errores. En la tabla 15 se representan los diferentes mensajes que muestra esta etiqueta para validar o informar de errores al usuario.

Tabla 15: Tabla de estados de etiqueta de control de errores,
 (Elaboración propia, 2023)

Mensaje en etiqueta de control de errores	Casos de uso que despliega mensaje en etiqueta de control de errores.
“No se ha introducido usuario ni contraseña válidos”	Los datos que ha ingresado el usuario no son válidos en los apartados de nombre de usuario y contraseña.
“Su nombre de usuario no es válido”	El nombre introducido no cumple la condición: $4 \leq \text{longitud} \leq 15$

“Su contraseña no es válida”	La contraseña introducida no cumple la condición: $6 \leq \text{longitud} \leq 12$
“Las contraseñas no coinciden”	Validada la longitud de la contraseña, la confirmación no coincide con el otro campo de ingreso de contraseña.
“Correcto!”	Los datos han sido validados y se procede a enviar el formulario.

Una vez que se han establecido los casos de uso para la validación de los datos introducidos en cada elemento, en la figura 42 se puede apreciar el diseño propuesto para analizar la capacidad de cada uno de estos para ser usados de manera óptima ya sea de manera individual o de integración, a la vez que se genera una disposición que facilite el recorrido de la mirada por parte del usuario. De igual manera, se enlistan los elementos que conforman la interfaz a continuación:

1. Título de interfaz.
2. Título para ingresar nombre de usuario.
3. Campo de ingreso de usuario.
4. Título para ingreso de contraseña.
5. Campo de ingreso de contraseña.
6. Título para conformación de contraseña.
7. Campo para confirmar contraseña.
8. Botón de selección de imagen de perfil.
9. Botón de creación de usuario.
10. Etiqueta de mensaje de error.
11. Botón Regresar

1 Crear Usuario

2 Nombre Usuario: 3

4 Contraseña: 5

6 Confirma Contraseña: 7

Selecciona una imagen  8

9

Error No se ha introducido usuario ni contraseña válidos

10

11

Figura 42: Diseño de Interfaz de Formulario Convencional

4.1.4 Implementación de técnicas y conceptos de eye-tracking

Como se ha mencionado anteriormente, la creación de este caso de estudio implementará los conceptos de eye-tracking generados en esta investigación. El primer concepto por aplicar es el marco de trabajo denominado EVUSINET, que es una herramienta para diferentes procesos de desarrollo, diseño y evaluación de interfaces puedan implementar el análisis objetivo del uso de interfaces mediante eyetracking.

Retomando el marco de trabajo EVUSNIET, este consta de tres secciones principales: pruebas de uso, implementación de eyetracking y evaluación. Cada una de estas secciones permite generar un procedimiento estandarizado para que cada interfaz pueda ser sometido al análisis. Sin embargo, y como se aborda de manera mucho más amplia en la sección 3.7, tanto el marco de trabajo EVUSINET, como el análisis de INUSINET requieren un alto grado de sensibilidad y conocimiento por parte del diseñador y evaluador para determinar si el uso del sistema esperado está siendo validado por los datos ofrecidos por el eyetracking. En otras palabras, si bien el análisis mediante eyetracking

elimina sesgos de uso muy importantes, es responsabilidad del diseñador generar los cambios pertinentes en los diseños propuestos de la interfaz para generar mejores experiencias de uso.

Mencionado lo anterior, se aborda a continuación, sección por sección, los pasos para implementar EVUSINET en el diseño del componente login.

A. Diseño experimental del componente login mediante EVUSINET.

El primer paso para generar un diseño experimental es determinar el tipo de estímulo de estudio. En este particular caso, y debido a que se usará el concepto de interfaces paralelas, se tienen dos estímulos, uno por cada interfaz. En la figura 43 se establece la secuencia de los elementos que conforman el diseño experimental. Este proceso cuenta con una etapa de calibración, una etapa de instrucciones (véase figura 44) y la grabación del par de estímulos.

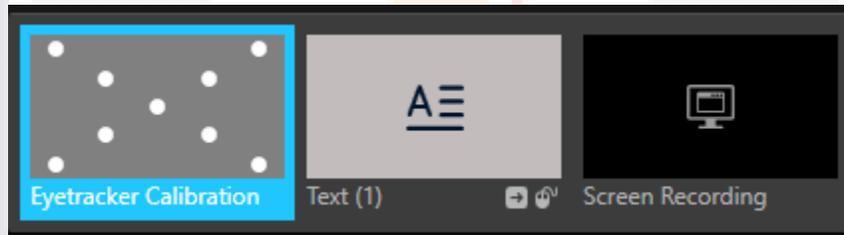


Figura 43: Diseño Experimental Interno de TobiiProLab, (Elaboración propia, 2024).

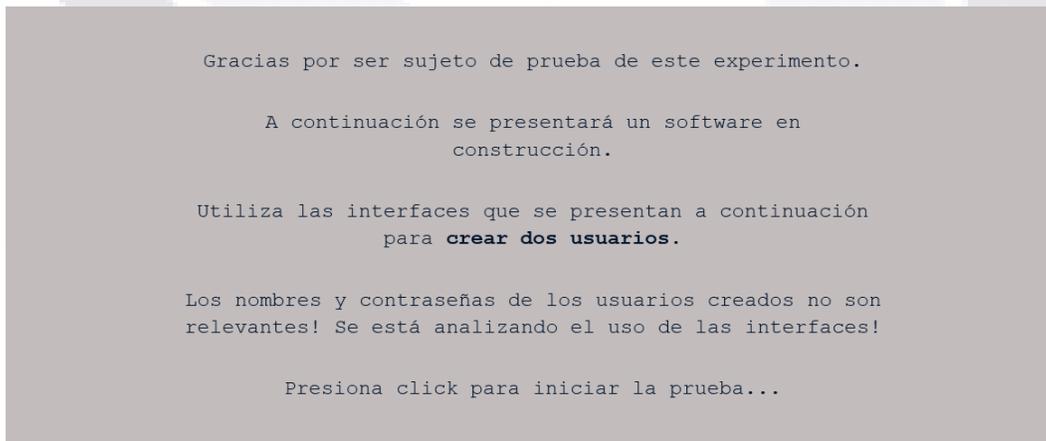


Figura 44: Instrucciones para el usuario de la prueba Fuente: (Elaboración propia, 2023)

Ya seleccionados los tipos de estímulos se establece la muestra estadística necesaria para contemplar un análisis profundo y preciso del uso de una interfaz y los tipos de usuarios que se plantea usen este software. Para responder al primer requisito, y para este caso de estudio se plantean diez pruebas para cumplir con la teoría de que cinco usuarios representan el uso de toda la población. Si bien la evaluación está planteada para que un mismo usuario genere un usuario para cada interfaz en la misma prueba, se consideró llevar a cabo más pruebas de las mínimas (5 pruebas) para generar resultados más precisos mediante eyetracking y reducir el riesgo del descarte de pruebas al implementar las técnicas de eye-tracking.

Para el tipo de usuario al que se enfoca la interfaz se consideró a jóvenes entre 18 y 25 años que tuvieran cualquier tipo de experiencia al usar equipos de cómputo. En caso específico de un componente login es de un uso tan general que, en este caso, no se optó por considerar usuarios de edad avanzada o con características de uso específicas ya que se busca conocer las capacidades de los conceptos propuestos para la implementación de técnicas de eye-tracking en la evaluación de interfaces.

La siguiente sección en el apartado de pruebas de uso de EVUSINET es generar mecanismos para que el proceso de pruebas sea lo más preciso posible, por lo cual se diseñan protocolos experimentales y se establece un lugar fijo para realizar todas las pruebas. Si bien las tecnologías de eye-tracking no son tan sensibles a parámetros como la luz de la habitación o las condiciones del espacio, el usuario sí lo es, por lo que se debe garantizar que los resultados obtenidos respondan únicamente al estímulo y algunas variables de uso del entorno (como lo podría ser la configuración de las teclas de un teclado o la resolución de la pantalla) sin que afecten la experiencia base de cada usuario.

En este caso, se usa el protocolo experimental de la figura 18 ya que las condiciones del experimento no requieren pasos adicionales a los establecidos en la mencionada figura.

B. Implementación de eye-tracking en el análisis del componente login mediante EVUSINET.

La etapa de implementación de técnicas de eyetracking consta de dos principales fases, la etapa de selección de eventos y áreas de interés y la exportación de los valores de cada métrica obtenidas durante las pruebas.

La selección de las áreas de interés en este momento facilita enormemente en el proceso de evaluación, así como la objetividad del análisis, ya que desde la etapa de planteamiento de las pruebas se definen los elementos que formarán parte del análisis particular, así como su función en las tareas que debe completar el usuario (para más detalles consúltese la sección 3.7.3 apartado B). Para realizar lo anterior se utiliza la herramienta de AOI's del software de *TobiiProLab*, utilizado para generar el análisis de interfaces para este caso de estudio (véase figura 45).

Tomando como base los elementos expresados en la sección 4.1.3 se establecen las áreas de interés para elemento de las propuestas de diseño. En la figura 46 se perciben las áreas de interés para la interfaz de diseño dinámico, mientras en la figura 47 a de diseño de formulario convencional.

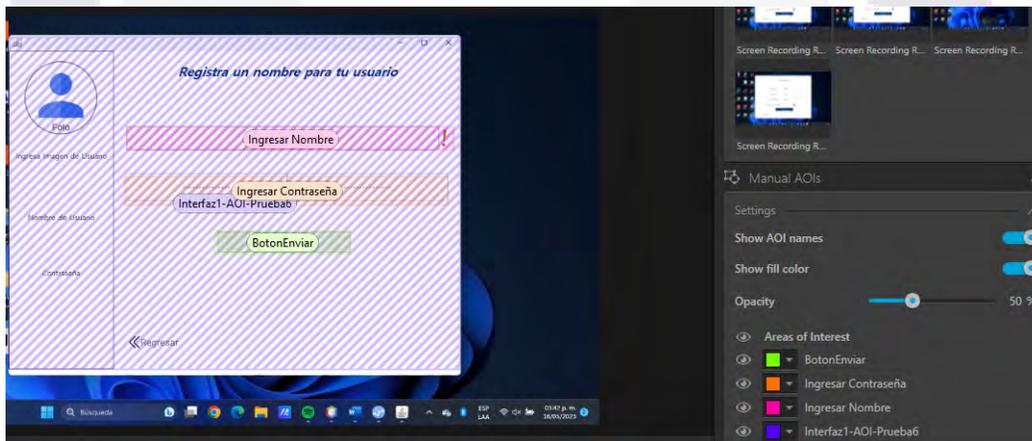


Figura 45: Herramienta de Áreas de Interés de TobiiProLab, (Elaboración propia, 2024).

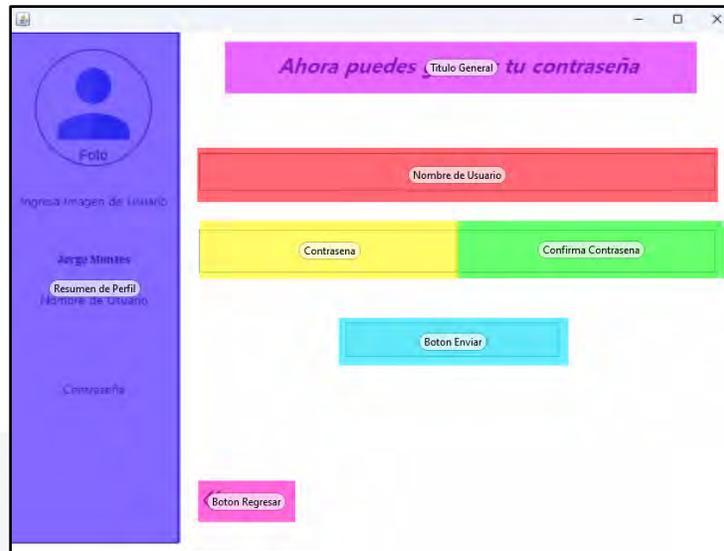


Figura 46: Áreas de interés para la interfaz de estudio con diseño dinámico, (Elaboración propia, 2024).



Figura 47: Áreas de interés para la interfaz de estudio con diseño de formulario convencional, (Elaboración propia, 2024).

Si bien en la formulación del marco del trabajo se indica que el primer paso en la etapa de aplicación de las técnicas de eye-tracking es la asignación de eventos, este puede ser posterior a la identificación de las áreas de interés. Para este caso de estudio, se establece que existen 5 fases por las cuáles pasará el usuario para completar una prueba. Lo anterior es posible visualizarlo en la figura 48, indicando en color verde las fases en dónde se aplicará el análisis mediante eye-tracking.

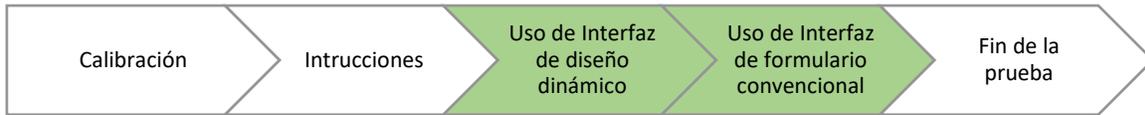


Figura 48: Diagrama de eventos de las pruebas de uso del par de propuestas de diseño de interfaces del caso de estudio del componente login, (Elaboración propia, 2024).

Mediante el software de TobiiProLab se expresan los eventos de la siguiente manera: en la figura 49 se establecen los tipos de eventos que se usarán en la prueba, en este caso solo se indica el inicio de uso de la interfaz y el fin de uso de esta. Posteriormente, en la figura 50, y para cada prueba realizada, se asignan los momentos en los que se alterna el uso de una interfaz a la otra de manera manual. Finalmente, en la figura 51 se expresan los tiempos de cada fase de la prueba. Cabe destacar que este tipo de análisis en el que se tiene que especificar manualmente el tiempo del evento es para pruebas en los cuáles sea imposible definir un tiempo exacto para la realización de esta.

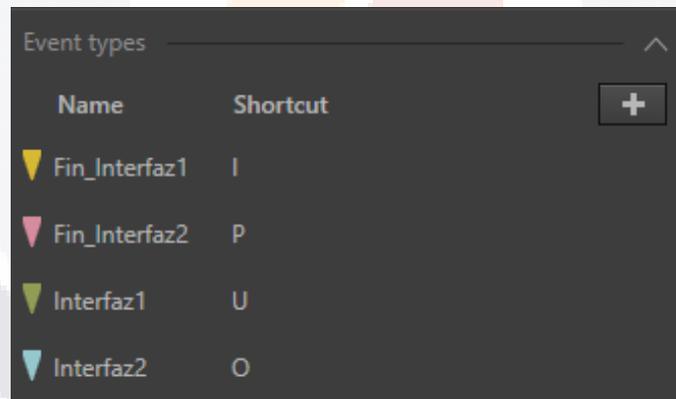


Figura 49: Delimitación de eventos para las interfaces de estudio (Elaboración propia, 2023).

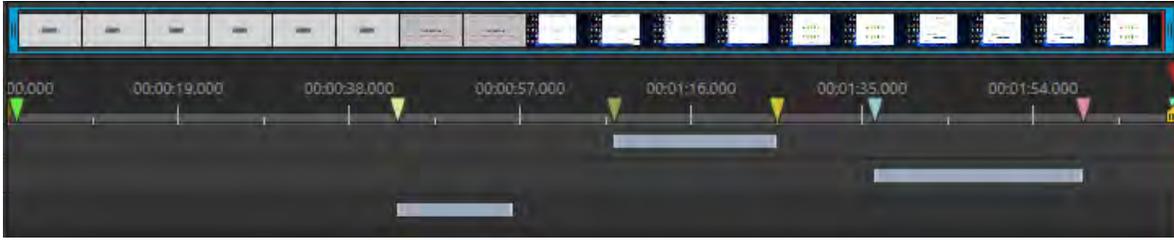


Figura 50: Grabación completa dividida en periodos
(Elaboración propia, 2023)

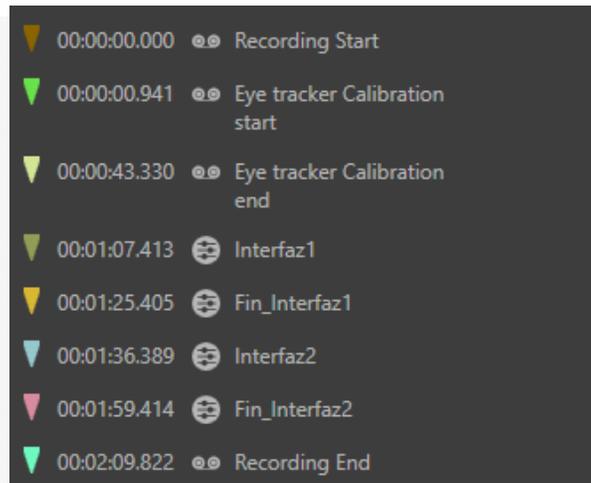


Figura 51: Tiempos de uso de cada evento de una única prueba del componente login,
(Elaboración propia, 2024).

La segunda sección de la implementación de las técnicas de eye-tracking corresponde a la exportación de los datos resultado de las pruebas. Es en esta etapa que el evaluador debe generar un data set que permite el fácil análisis de los datos obtenidos durante las pruebas. En la figura 52 se visualiza un data-set que busca agrupar los resultados de la interfaz de formulario convencional, este proceso debe repetirse para la otra propuesta de diseño. Esta etapa es lenta y minuciosa, más no es compleja ya que el software de TobiiProLab, aligera el trabajo ampliamente.

Una de las ventajas de seguir el marco de trabajo EVUSINET es que el evaluador no tiene que decidir una a una las métricas que ofrecen diversas tecnologías del mercado o librerías de código abierto, si no que “activa” únicamente las variables que serán utilizadas siguiendo el modelo de INUSINET. Sin embargo, es responsabilidad del evaluador generar data-set válidos y que permitan analizar los resultados sin que tome mucho tiempo.

Ampliando lo anterior y enfocándolo en el caso de uso, es posible generar un único data-set que contenga la información de ambos estímulos y de todos los sujetos de prueba, sin embargo, si se realiza de esta manera el manejo de los datos podría ser confuso, tardado y muy susceptible a truncar los resultados.

En la figura 52 se muestra una captura del software *TobiiProLab* en su apartado de *Metrics Export*, el cual permite filtrar cada una de las variables que conforman el data-set que contiene los datos biométricos de los sujetos de prueba. Para que la interpretación de la figura 52 sea más sencilla se procede a explicar brevemente las diferentes secciones que conforman la interfaz del software de experimentación: la ventana de previsualización del data-set, el menú desplegable para las diferentes categorías de métricas que la herramienta de seguimiento de mirada registra y los parámetros de la prueba que se desea exportar.

La ventana de previsualización permite verificar si algunas de las etapas en el proceso de análisis no fueron realizadas correctamente, también facilita agrupar visualmente los datos para que estos puedan ser analizados fácilmente. La sección de métricas es un menú donde se describe a grandes rasgos las características e información que brinda la métrica y la unidad de medida en que se expresa. Por último, en el costado derecho de la pantalla se encuentran las opciones que permiten filtrar la información que se desea almacenar en el documento del data-set. Es posible dividir por estímulo mostrado, por sujeto de pruebas, incluso por alguna área de interés en específico que se desee analizar. El conjunto de estas herramientas permite generar documentos que tengan una lectura definida por algún algoritmo que consuma esta información, o permita al evaluador, de manera “manual”, generar un resultado de evaluación con los datos.

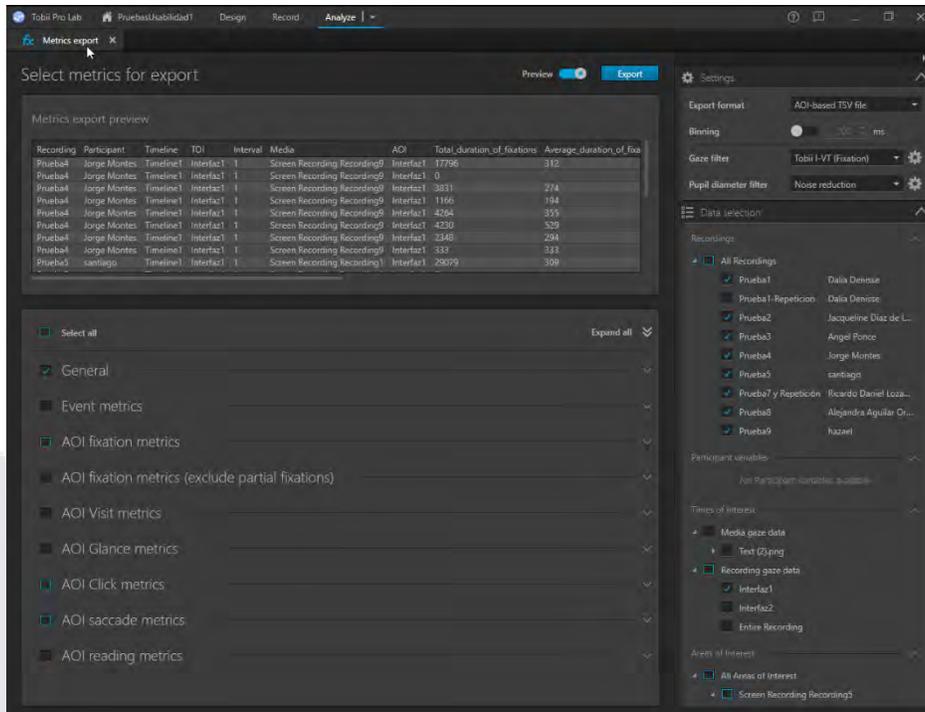


Figura 52: Elección de las métricas, eventos y pruebas que forman el data-set para la evaluación objetiva del uso de interfaces mediante eye-tracking del caso de estudio del componente login, (Elaboración propia, 2024).

C. Evaluación mediante eye-tracking del componente login.

Como se menciona en la sección 3.7.1 y como se refuerza en el marco de trabajo EVUSINET, el análisis de interfaces mediante eye-tracking permite dos niveles de análisis con respecto al patrón de uso. El primer caso de patrón de uso es el general o de alto nivel y permite visualizar al diseñador las áreas o recorridos que realizan los usuarios para completar una tarea. Este análisis general de los elementos permite identificar si la distribución de estos genera un patrón de uso repetible en la población de sujetos de prueba (a este patrón en el que la mayoría de los sujetos repite el uso se le denomina patrón de uso convergente). El planteamiento anterior cobra mayor fuerza cuándo se implementa la metodología de diseño de interfaces paralelas ya que mediante la comparación de las capacidades de ambas propuestas se genera una competencia de cada diseño para que sus elementos sean lo más “usables” posibles.

Como se menciona en la idea anterior, los patrones de uso generales son una herramienta principalmente visual en la que los evaluadores identifican a simple viste el comportamiento del

usuario. Mediante las herramientas de eye-tracking actuales es posible representar y filtrar la información que se desea analizar en la graficación.

a) Mapas de calor

Convencionalmente, los mapas de calor son la herramienta más usada en el análisis mediante eye-tracking para conocer las áreas más visitadas. En la figura 53 es posible visualizar un conjunto de lecturas en colores amarillo, rojo y verde. Al observar exclusivamente la figura antes mencionada no existe mucho contexto sobre el uso de la interfaz ya que las regiones rojas (percibidas como las más visitadas durante la prueba) parecen indicar un lugar exacto de interés, sin embargo, las muestras en verde (que indican lugares visitados brevemente) están repartidas por todo el gráfico de manera casi arbitraria.

Como se mencionó en ideas anteriores, la potencia del análisis de alto nivel mejora al confrontar una propuesta de diseño que busque resolver el mismo problema mediante la variación de las propiedades de los elementos que conforman la interfaz de referencia. Es por esta razón, que una vez observada la figura 54 es posible contrastar la información que, anteriormente, parecía arbitraria en la figura 53. Esta comparación tan estricta de las propiedades e influencias de los elementos que son los mismos entre ambas propuestas, pero que poseen diferentes propiedades de diseño, posibilita al diseñador determinar si la experiencia observada de manera objetiva responde a las expectativas previamente planteadas.

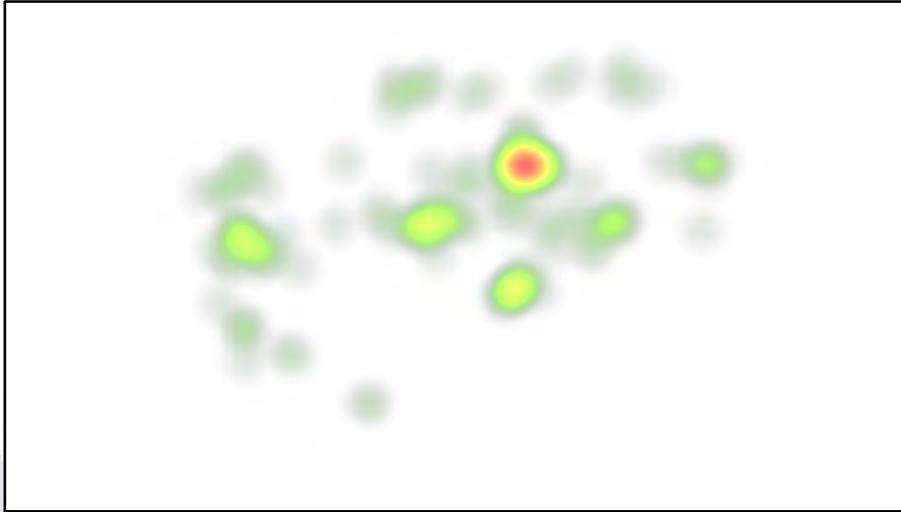


Figura 53: Mapa de calor de la interfaz 1 obtenida de una prueba aleatoria del componente login, (Elaboración propia, 2024).

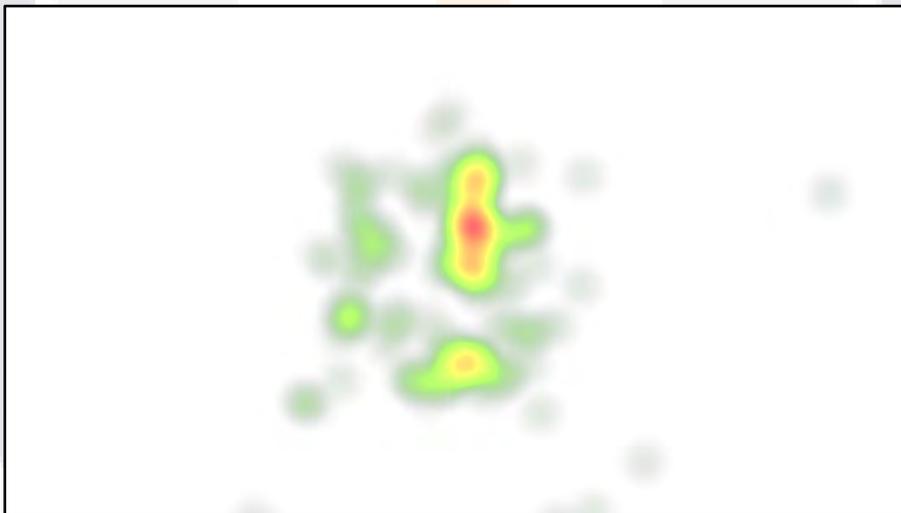


Figura 54: Mapa de calor de la interfaz 2 obtenida de una prueba aleatoria del componente login, (Elaboración propia, 2024).

b) Mapas de recorrido (camino de abeja)

El patrón de camino de abeja es denominado así ya que es una línea continua que registra secuencialmente la mirada del sujeto que realiza la prueba haciendo un símil con el vuelo de este insecto en particular. Este tipo de visualización permite identificar patrones de movimiento ocular entre distintos puntos específicos de la interfaz. En el caso del componente login al ser necesario que el sujeto de pruebas ingresara información a través del teclado es que existen las líneas que se

proyectan a la parte inferior de la pantalla (véase figura 55). Ahora bien, comparando la figura 55 con la figura 56 es notoria la diferencia de la línea que representa el recorrido de uso: mientras que en la primera hay movimientos mucho más horizontales, en la segunda los recorridos son principalmente verticales.

Como se demuestra al utilizar este tipo de visualización, esta contiene mucha más información para el evaluador ya que no indica únicamente puntos reincidentes de atención visual, si no que indica las relaciones y efectos que tienen los elementos en los patrones de recorrido de la interfaz.

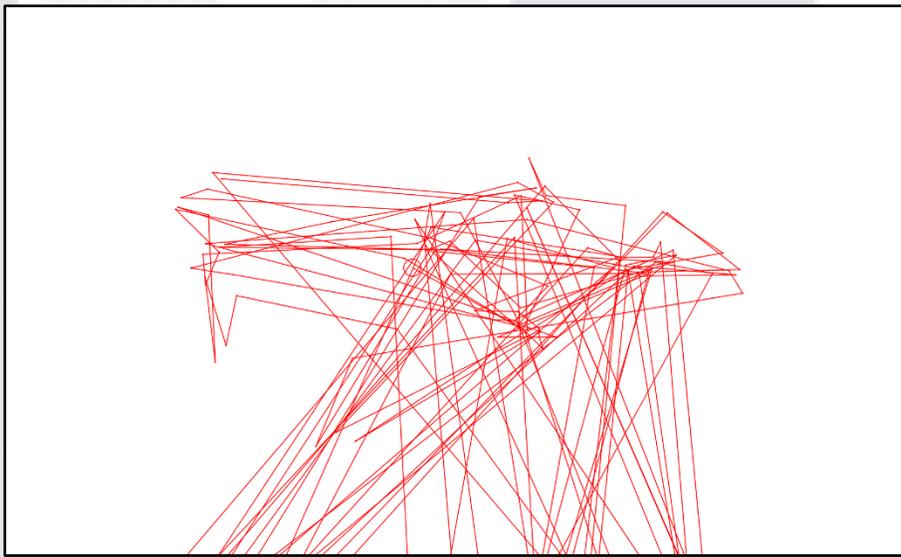


Figura 55: Visualización del camino de abeja del recorrido de mirada de la interfaz 1 de un único sujeto de prueba, Fuente: (Elaboración propia, 2023)

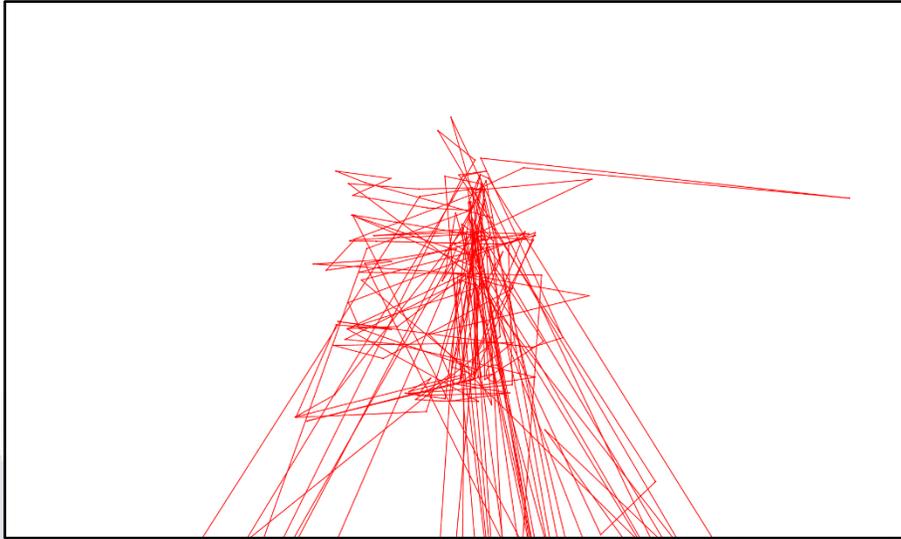


Figura 56: Visualización del camino de abeja del recorrido de mirada de la interfaz 2 de un único sujeto de prueba

Fuente: (Elaboración propia, 2023)

c) Diagrama de recorridos con fijaciones.

Parecido a los patrones de recorrido de abeja, la visualización de fijaciones permite reconocer el recorrido realizado en la interfaz, sin embargo, se añade información adicional de las fijaciones. Esta información adicional representa principalmente la secuencia ordenada de las fijaciones, sin embargo, si se observa detallada la figura 57 o la figura 58 cada círculo rojo, que representa la fijación, tiene un diámetro diferente. Este diámetro hace referencia a la duración de la fijación hasta un máximo de 200 milisegundos (valor recomendado por la herramienta *TobiiProLab*).

Esta información sobre las fijaciones permite al diseñador darse una idea general de los lugares o elementos que más son visitados, sin embargo, una fijación larga no siempre hace referencia a un mejor lugar para obtener información, también puede estar relacionada con un contenido confuso. También existen fijaciones que son muy pequeñas con respecto a las demás, si bien estas pueden relacionarse con la lectura de algún elemento, también puede representar un movimiento ocular que busque información casi de manera desesperada, lo que podría indicar que el elemento no tiene las características de diseño adecuadas para ser usado o visto.

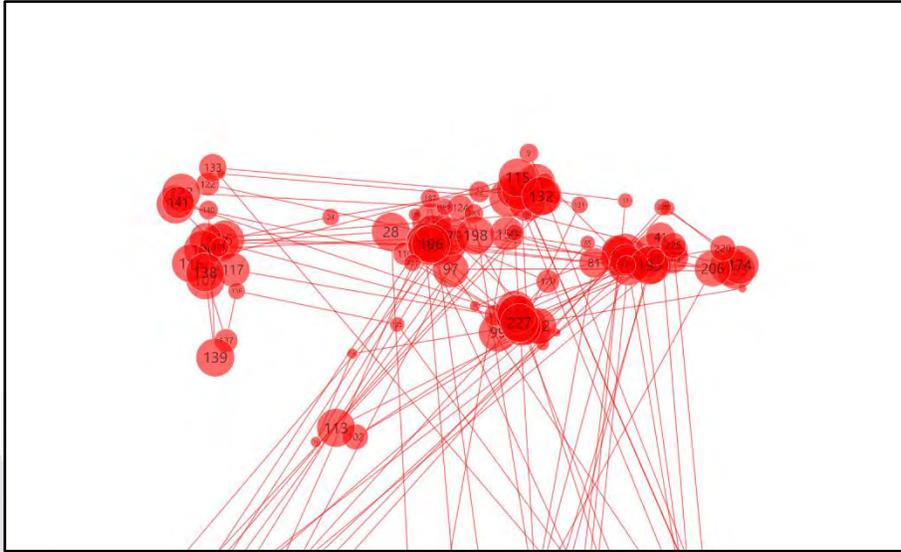


Figura 57: Visualización de ruta de recorrido con fijaciones de la interfaz 1 de un único sujeto de prueba, (Elaboración propia, 2024).

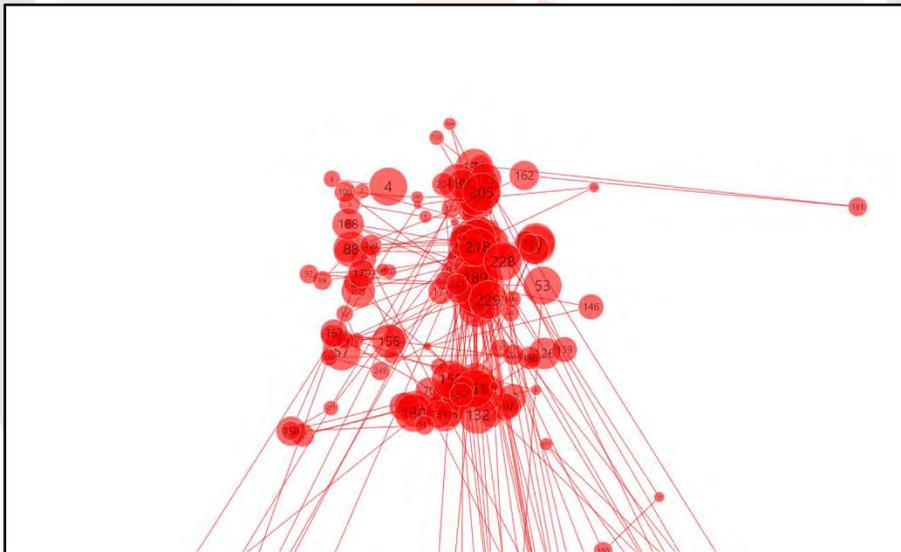


Figura 58: Visualización de ruta de recorrido con fijaciones de la interfaz 1 de un único sujeto de prueba, (Elaboración propia, 2024).

e) Data-set para el análisis de patrones de uso particulares.

Ya se han abordado las diferentes visualizaciones que hacen posible el análisis de uso generales. Ahora es turno de permitir el análisis particular de cada uno de los elementos de la interfaz. Como se aborda en la sección de EVUSINET, el primer paso para generar un análisis válido de la interfaz es

TESIS TESIS TESIS TESIS TESIS

tener los mecanismos para controlar que todas las pruebas sean lo más parecidas posibles, y en su defecto, retirar del estudio aquellos datos sesgados o poco precisos.

Como se menciona en apartados anteriores, idealmente se buscaba registrar una cantidad de diez pruebas para duplicar la cantidad mínima de pruebas de usabilidad. Y si bien se llevaron a cabo todas las pruebas, existieron dos casos en los que se tuvo que descartar la prueba.

El primer caso en el que se tuvo que cancelar el registro de los datos (véase figura 18) fue debido a que la secuencia de interfaces prevista para el experimento no pudo ser satisfecha ya que el usuario cerró la interfaz 1, concluyendo el proceso y dificultando enormemente el proceso de análisis de eye-tracking, por lo que se decidió por descartar la prueba.

El segundo caso estuvo relacionado con una falla en la calibración del dispositivo, como se indica en la figura 59 toda prueba contiene una etapa de calibración, sin embargo, en esta prueba en particular no se detectó que tenía una calibración pobre y que imposibilitaba analizar los datos hasta que se generaron las primeras versiones del documento de resultados de las pruebas de eye-tracking. Al buscar información adicional en el software de TobiiProLab para intentar identificar el origen de la falla se encontraron los resultados de la calibración de la prueba. En la figura 59 se observan un conjunto de círculos cuyo diámetro expresa el error aproximado de la fijación en el área de la pantalla en donde se ubica. Si bien existen círculos cuyo diámetro parece no ser muy grande, el error más pequeño de esta prueba es más elevado que el error más grande de la segunda prueba peor calibrada. Es por estas razones que se optó por retirar la prueba definitivamente al no tener una precisión tal que permitiera realizar un análisis específico de cada elemento.



Figura 59: Resultados de calibración de prueba retirada del análisis de eye-tracking, (Elaboración propia, 2024).

Mencionado lo anterior, y tomando como base lo expresado en la figura 52 el data-set generado para analizar las pruebas tiene los siguientes atributos para los resultados de ambas propuestas de diseño.

- Nombre de la grabación de la prueba.
- Participante.
- Diseño Experimental.
- Intervalo de estudio.
- Nombre del Estímulo
- Área de Interés.
- Duración Total de las fijaciones.
- Cantidad de fijaciones.
- Cantidad de Clicks.
- Cantidad de sacádicos
- Velocidad de entrada de los sacádicos.
- Velocidad de salida de los sacádicos.
- Promedio de velocidad de sacádicos.

En esta sección se abordaron los procesos más relevantes que, primeramente, dieron forma al software denominado como el componente login, luego se analizaron los fundamentos del diseño de interfaces y se detuvo en explicar las capacidades de la metodología de diseño de interfaces y, finalmente se expresaron los procedimientos para la integración de las técnicas de eye-tracking en la evaluación del producto. En el siguiente capítulo, en la sección relacionada a los resultados de la evaluación se continúa con el análisis de interfaces y se determina detalladamente el significado de estos con respecto a las expectativas de diseño planteadas y desarrolladas para este caso de estudio.



4.2 Caso de estudio 2: Easy-Bank Vision

El segundo caso de estudio que se realiza en esta investigación está planteado para analizar mediante las métricas, metodologías y conceptos de eye-tracking en un conjunto mayor de interfaces y con más sujetos de prueba que respondan a diversos perfiles de usuario.

Para este caso de estudio se aplica la metodología ágil expresada en la sección 3.3 ya que, para este caso de estudio se cuenta con un equipo desarrollo mucho más amplio que en el caso de estudio del componente login, lo que hace que implementar una metodología ágil es mucho más acertado que una convencional.

Una de las principales características del equipo de desarrollo es que, si bien todos pertenecen al mismo perfil general, cada uno tiene fortalezas en diferentes ramas del desarrollo que permiten plantear un equipo amplio y con roles bien planteados.

También otra de las características generales del desarrollo es que, tomando como base los conceptos de diseño centrado en el usuario, se genera un apartado más específico de los perfiles de usuario que usarán la interfaz.

4.2.1 Características del sistema

Como se menciona anteriormente, una de las características que moldearon las dimensiones del caso de estudio y sus posibilidades es que el grupo de personas implicadas en el proyecto es de 6 participantes incluyendo al investigador principal de este trabajo.

Una de las primeras tareas fue identificar los roles de cada miembro del equipo, en el caso particular de este proceso de desarrollo, se establecieron cuatro roles cada uno con sus funciones específicas: líder de proyecto, diseñador de interfaces, desarrollador y evaluador.

Además de los roles anteriores, también existe la entidad de un investigador encargado, el cuál provee las herramientas de hardware y software de eye-tracking, así como en la orientación del uso de estos elementos en el proceso de desarrollo de software.

Ahora bien, el desarrollo de este caso de estudio, al igual que componente login, está ampliamente influenciado por las tecnologías y técnicas de eye-tracking. Para que, independientemente de sus roles y actividades, cada miembro del equipo estuviera en la misma sintonía y conocieran las propiedades de las tecnologías de seguimiento de mirada, se les brindó un curso y demostraciones de otras investigaciones que estaban en etapas mucho más avanzadas y que permitieron aproximar

de manera más precisa los tiempos, esfuerzos y características de diseño que mejor se adaptasen a la implementación del eye-tracking.

Es por esta razón que las actividades directas en el desarrollo del caso de estudio, como otras complementarias están especificadas desde el primer momento en la creación de los roles en la tabla 16:

Tabla 16: Roles y actividades del equipo del caso de estudio, (Elaboración propia, 2024)-

<p>Líder del Proyecto</p>	<ul style="list-style-type: none"> • Definir los objetivos del proyecto y la visión general. • Realizar investigación sobre eye-tracking, UI/UX y diseño de páginas web bancarias. • Investigar las referencias de las investigaciones. • Coordinar las actividades del equipo y asignar tareas. • Realizar pruebas de eye-tracking con usuarios y analizar los resultados. • Diseñar y conducir pruebas de usabilidad. • Documentar el proceso y los resultados del proyecto. • Encargado de tecnologías de eye-tracking.
<p>Equipo de Desarrollo</p>	<ul style="list-style-type: none"> • Desarrollar la página web simulada del banco basada en el diseño y las especificaciones proporcionadas. • Integrar el eye-tracking en la página web para recopilar datos sobre la mirada de los usuarios. • Implementar las funcionalidades y elementos de UI/UX diseñados para mejorar la experiencia del usuario. • Realizar pruebas de funcionalidad y compatibilidad.
<p>Diseño de interfaces</p>	<ul style="list-style-type: none"> • Diseño de diagramas de flujo. • Crear diseños de interfaz de usuario centrados en la usabilidad y la estética. • Diseñar elementos visuales, como botones, iconos y esquemas de color, considerando la identidad del banco. • Desarrollar flujos de usuarios intuitivos y atractivos. • Colaborar con el equipo de desarrollo para asegurarse de que los diseños se implementen correctamente.

Evaluador	<ul style="list-style-type: none"> • Conocer detalladamente las características de eye-tracking. • Promover cambios en las interfaces para que las evaluaciones brinden resultados más reconocibles. • Generar la evaluación de interfaces • Representar resultados.
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Una vez establecidas las características del equipo, es relevante mencionar que los miembros de este (todos estudiantes de pregrado) propusieron desde el tema central del caso de estudio, así como los posibles enfoques de experiencia de uso, interfaces, mecanismos de evaluación, mientras eran asesorados en todo momento por el investigador encargado (con grado en ingeniería de sistemas) como se mencionó anteriormente.

El tema general del caso de estudio se eligió tomando como base varias consideraciones. La primera y más importante es que el caso de estudio tuviera la capacidad de desplegar las interfaces para generar una experiencia retadora y compleja, en la que las condiciones de las interfaces fueran lo suficientemente complejas para retar ligeramente el usuario, pero lo suficientemente intuitivas para que pudiera completar la tarea.

También se consideró atacar un área de oportunidad que, a vista del equipo, las aplicaciones actuales no cubrían de buena manera. Por último, también se tomó en cuenta la cantidad de tipos de usuarios que podrían usar el producto del caso de estudio.

Es por lo anterior que se seleccionó una aplicación web que respondiera a las principales funciones de una banca en línea, desarrollando interfaces visualmente atractivas, pero agradables y usables para el usuario. Ampliando lo anterior, también se generaron 4 perfiles de usuarios en los que se aplicaron conceptos de desarrollo centrado en el usuario (perfiles que serán descritos y abordados más detalladamente en secciones posteriores).

Resumiendo lo anterior, la banca en línea simulada denominada “Easy-Bank Vision” (nombre propuesto por el equipo de desarrollo que altamente inspirado en la facilidad de uso y la atención visual”) es un producto que busca mediante la implementación de técnicas de eye-tracking mejorar los procesos de análisis, diseño de interfaces, desarrollo y evaluación de un producto de software.

En la siguiente sección, se expresan detalladamente las etapas del proceso de desarrollo del caso de estudio Easy-Bank Vision, así como los procesos de evaluación que se elaboraron siguiendo los

fundamentos del marco de trabajo EVUSINET la evaluación más rigurosa de los elementos por parte de INUSINET.

4.2.2 Proceso de desarrollo de software.

Uno de los primeros aspectos que destaca del caso de estudio de Easy-Bank Vision comparándolo con el componente login es el tamaño de la experiencia general planteada. Si bien el equipo que sustenta le elaboración del segundo caso de estudio es mucho mayor al primero, la aplicación de las tecnologías de eye-tracking es una tarea minuciosa y que ocupa la productividad completa de varios de los miembros del equipo.

Además de lo anterior, el proceso de desarrollo debía de contemplar dos principales aspectos en todo momento: el diseño centrado en el usuario y, como se menciona en la idea anterior, el análisis de eye-tracking.

Para dar forma a este proyecto, se seleccionó el lenguaje de Angular para desarrollar las funcionalidades del proyecto. Por el lado del diseño de interfaces se utilizó la herramienta de diseño Figma, la cual permite generar un grupo de desarrollo en el cuál cada persona en el proyecto puede generar modificaciones.

A. *Objetivos*

Para establecer los objetivos se tomaron las consideraciones anteriormente expresadas en este capítulo. De manera general y priorizando el apartado funcional, los principales objetivos del software son los siguientes y están expuestos en la tabla 17:

Tabla 17: Objetivos del caso de estudio Easy-Bank Vision, (Elaboración conjunta con el equipo de desarrollo, 2023).

Objetivo general	
Desarrollar un sistema que además de proponer una solución a una necesidad del mercado, permita e implemente técnicas de eye-tracking para mejorar los procesos de análisis, diseño de interfaces, desarrollo y evaluación de software.	
Objetivos específicos	Prioridad.
Generar una aplicación en la que la experiencia de usuario sea compleja, pero eficiente y agradable para el usuario.	Obligatorio
Proporcionar interfaces con accesibilidad que permitan a los grupos de usuarios tener una experiencia correcta conforme sus necesidades.	Obligatorio
Buscar métodos para hacer más eficiente los procesos de la banca en línea.	Deseable

Diseñar interfaces que promuevan el análisis mediante eye-tracking.	Obligatorio
Contemplar distintos tipos de usuarios en el desarrollo del software.	Obligatorio

B. Requerimientos

En la tabla 18 se representan los requerimientos funcionales del caso de estudio de Easy-Bank Vision. Como se expresa en la tabla antes mencionada, el diseño de las interfaces de estudio son prioridades para generar un producto que sea fácilmente analizable. Posteriormente, en la tabla 19 se establecen los requisitos no funcionales.

Tabla 18: Tabla de requerimientos funcionales de Easy-Bank Vision, (Elaboración conjunta con equipo de desarrollo, 2023).

Clave.	Descripción del requerimiento funcional.	Fecha.
R1	Diseñar una interfaz para el login de usuario.	22/11/23
R2	Diseñar una interfaz alterna para el login de usuario.	Idem.
R3	Diseñar una interfaz Home.	“
R4	La interfaz Home debe tener una sección de monitoreo y pago de tarjetas de crédito y débito.	“
R5	La interfaz Home debe contener los íconos de las secciones de la plataforma de Easy-Bank Vision.	“
R6	Diseñar una interfaz Saldos y Movimientos.	“
R7	Diseñar una interfaz Transferencias.	“

Tabla 19: Tabla de requerimientos no funcionales de Easy-Bank Vision, (Elaboración conjunta con equipo de desarrollo, 2023).

Clave.	Descripción del requerimiento funcional.	Fecha.
NF1	Base de datos simulada.	22/11/23
NF2	Prioridad en ofrecer una experiencia apta para el análisis de eye-tracking.	Idem.
NF3	Debe contemplar diferentes experiencias para diferentes tipos de usuarios.	“

C. Tipos de usuarios

Los tipos de usuarios que se plantean para este uso están generados principalmente en rangos de edad, en los que se estima que, la mayoría de los usuarios de las bancas en línea son principalmente adultos y que, varios de los jóvenes no tendrán experiencia previa en este tipo de servicio y que también, por falta de experiencia en el uso de equipos de cómputo, los adultos mayores tendrían

dificultades en el uso de estas tecnologías. A continuación, se enlistan los tipos de usuarios específicos para el análisis de interfaces.

- **Usuario 18-25 años.**
- **Usuario 25-40 años.**
- **Usuario 40-60 años.**
- **Usuario 60+ años.**

4.2.3 Propiedades de las interfaces

Debido a que la etapa de diseño de interfaces se consideró una de las más importantes en cada etapa del sprint, varios de los diseños generados pudieron completarse mediante la herramienta de Figma antes de que comenzara el proceso de codificación como tal, sin embargo, la constante comunicación entre el investigador encargado, diseñadores y evaluadores permitieron generar diseños que rápidamente tuvieron el visto bueno del equipo para llevarse a cabo.

Una de las desventajas que tuvo diseñar las interfaces antes de que se estimaran los tiempos de codificación de cada uno de los diseños y, sobre todo, los esfuerzos y complejidad requerida para los módulos de back-end fue que varias de las propuestas de diseño tuvieron que ser removidas de las etapas de evaluación, ya que no era viable realizar pruebas que aumentaran radicalmente la complejidad del análisis de eye-tracking en el tiempo límite que se tenía (que era de poco menos de 6 meses). Mencionado lo anterior, se procede a exponer las interfaces que formaron parte de las evaluaciones.

Tomando como base la metodología de interfaces paralelas, pero sin llegar a implementar del todo el concepto, y también tomando inspiración en el componente login, para el caso de estudio de Easy-Bank Vision se propuso un botón que permitiera alternar entre dos propuestas de diseño de una misma interfaz, estas propuestas alternas de diseño están implementadas de manera más evidente en los diseños referentes al login del caso de estudio. Como es posible apreciar en las figuras 60 y 61 ambas interfaces cuentan con los mismos elementos, pero cuenta con diferencias entre la dimensión y distribución de estos.

La principal razón por la que se hace la aclaración de que, si bien las propuestas de la interfaz login se implementan algunos conceptos de la metodología de diseño de interfaces paralelas, las variaciones en las interfaces no están destinadas a encontrar directamente los elementos que peor

desempeño muestren en la interfaz, si no que están pensando para ofrecer herramientas específicas para diferentes tipos de usuarios que puedan presentar problemas al interactuar con algún u otro diseño. A pesar de lo anterior, sigue siendo posible evaluar las propuestas mediante las propuestas de INUSINET.

Listado de elementos de la interfaz Login Detallado:

- [1] Logos
- [2] Títulos
- [3] Etiqueta Numero de Cliente
- [4] Campo Ingreso Numero de Cliente
- [5] Etiqueta Contraseña
- [6] Campo Ingreso Contraseña
- [7] Botón Enviar Formulario
- [8] Switch Modo Detallado

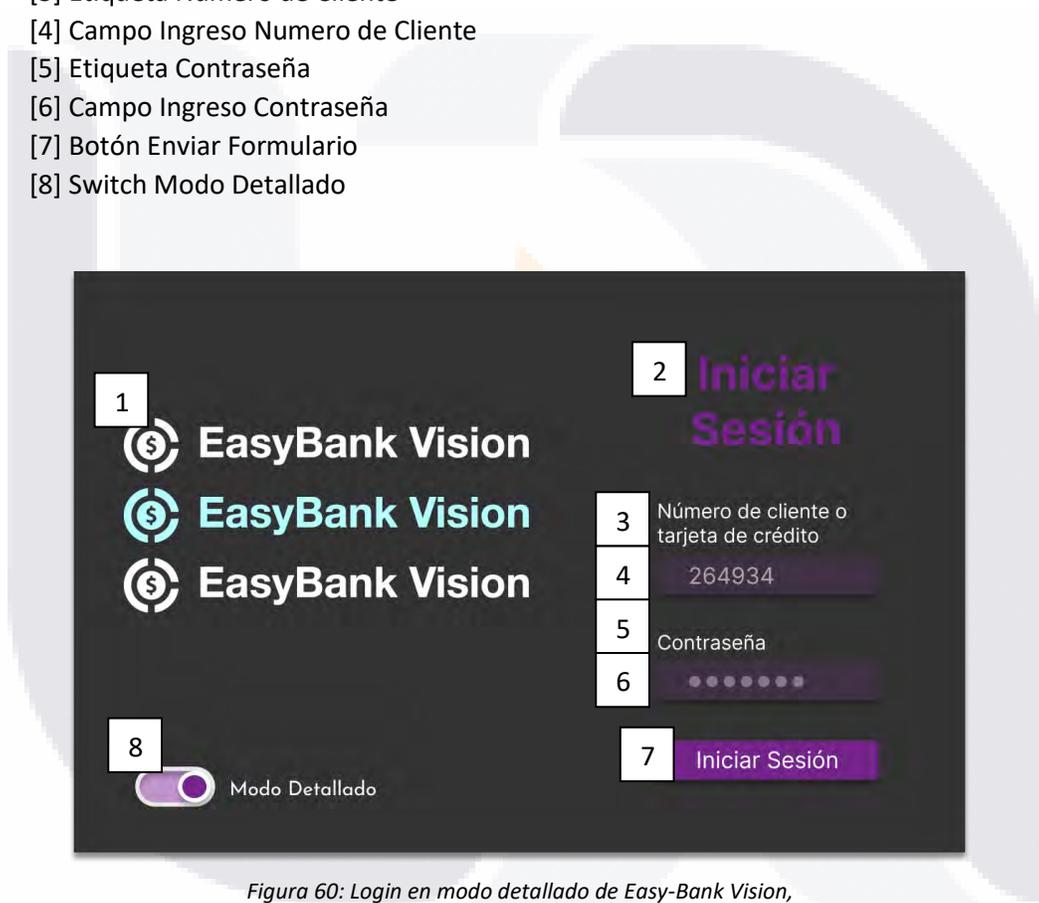


Figura 60: Login en modo detallado de Easy-Bank Vision, (Elaboración propia, 2024).

Listado de elementos de interfaz de Login Simple:

- [1] Logos
- [2] Títulos
- [3] Etiqueta Numero de Cliente
- [4] Campo Ingreso Numero de Cliente
- [5] Etiqueta Contraseña
- [6] Campo Ingreso Contraseña

- [7] Botón Enviar Formulario
- [8] Switch Modo Detallado



Figura 61: Login en modo simple del Easy-Bank Vision, (Elaboración propia, 2024).

Una vez completado el proceso inicio de sesión, que como se menciona en los requerimientos no funcionales, las credenciales y niveles de los usuarios en el sistema son generados a partir de una base de datos simulada. Esta base de datos cuenta con valores preestablecidos para el nombre de usuario, contraseña y los saldos y registros de tarjetas de crédito, débito, saldos y movimientos, etc. los cuáles son desplegados posteriormente en las demás interfaces de la interacción.

Después de la pestaña de login se despliega la interfaz de Home (representada en la figura 62), la cual permite acceder a los demás elementos de estudio en la interacción, los cuáles son la pestaña de transferencias (véase figura 63) y la ventana de saldos y movimientos (contenida en la figura 64). Además de lo anterior, se plantea otro caso de uso en el que el usuario puede acceder a la información de sus tarjetas y pagar algún adeudo hipotético.

Como se da a entender en el párrafo anterior, varios de los procesos de desarrollo de este producto se elaboraron de manera simultánea debido a que las interfaces y sus contenidos influían directamente en los diagramas que servirían al equipo de desarrollo para completar los objetivos del sistema en general, pero primordialmente de las interfaces. En el siguiente apartado se expresan los diagramas que fueron ampliamente influenciados tanto por el equipo de desarrollo como el

investigador encargado para, además de desarrollar un producto con buenas cualidades de uso, también permitiera la evaluación de interfaces mediante eye-tracking.

Listado de elementos de interfaz Home:

- [1] Logo
- [2] Salir
- [3] Botón Saldos y Movimientos
- [4] Botón Estado de Cuenta
- [5] Botón Inversiones
- [6] Botón Transferencias
- [7] Botón Pago Servicios
- [8] Ayuda
- [9] Tarjetas
- [10] Switch Modo Detallado

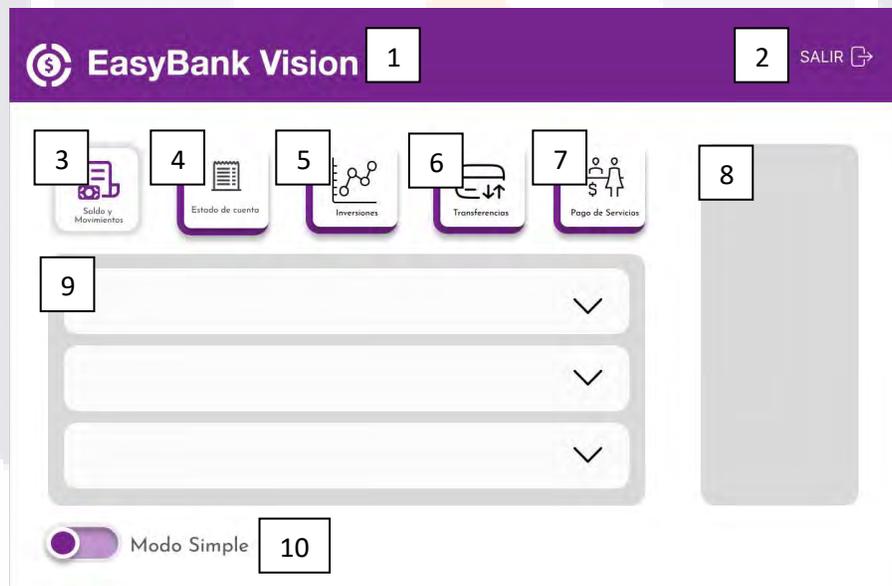


Figura 62: Interfaz Home de Easy-Bank Vision, (Elaboración propia, 2024).

Listado de elementos de interfaz Transferencias:

- [1] Logo
- [2] Salir
- [3] Datos Cuenta
- [4] Ayuda
- [5] Destino Tarjeta

- [6] Destino Nombre
- [7] Destino Cantidad
- [8] Botón Enviar Formulario
- [9] Switch Modo Detallado



Figura 63: Interfaz de Transferencias de Easy-Bank Vision, (Elaboración propia, 2024).

Listado de elementos de interfaz Saldos y Movimientos:

- [1] Logo
- [2] Salir
- [3] Botón Todos los registros
- [4] Botón Ingresos
- [5] Botón Egresos
- [6] Botón Ayuda
- [7] Tabla
- [8] Switch Modo detallado

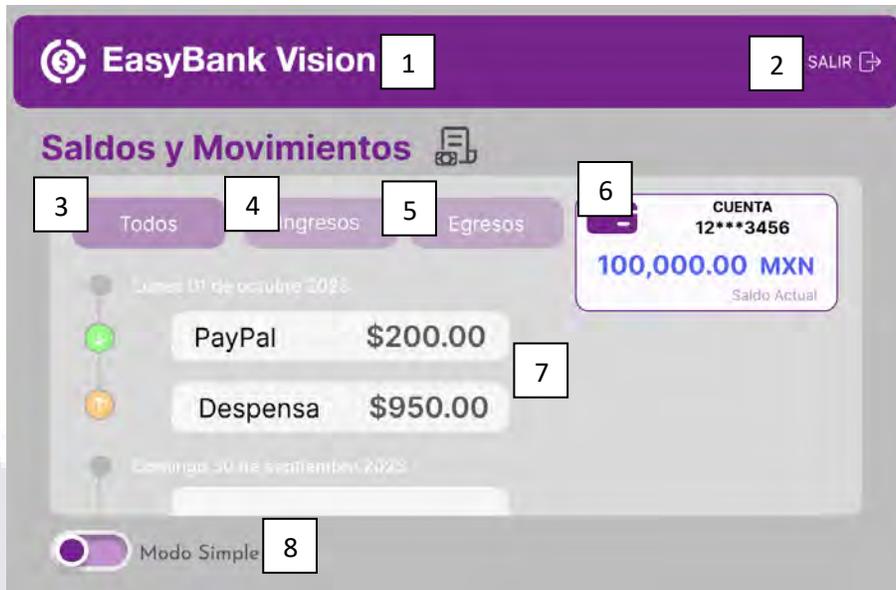


Figura 64: Interfaz Saludos y Movimientos de Easy-Bank Vision, (Elaboración propia, 2024).

Listado de elementos de interfaz Pagar Tarjeta:

- [1] Ventana Emergente
- [2] Datos de cuenta de Usuario
- [3] Campo Ingreso cantidad de pago
- [4] Botón enviar formulario

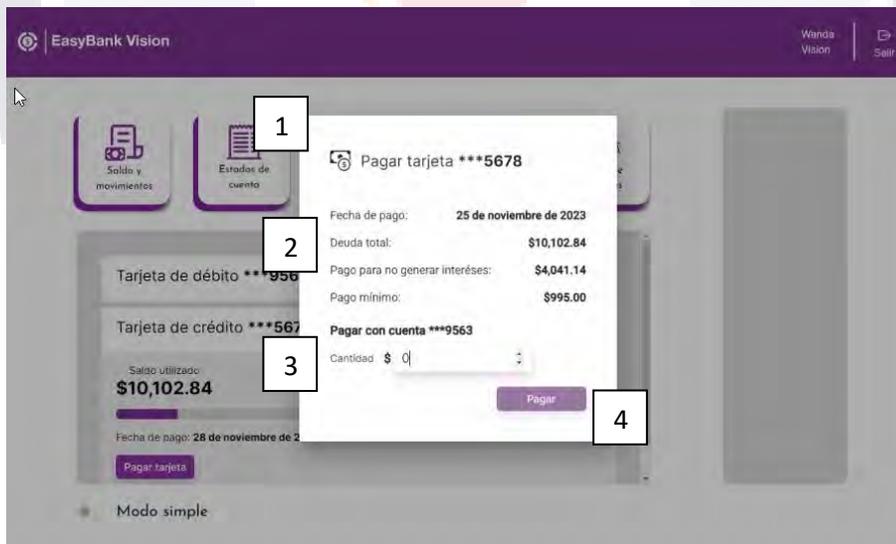


Figura 65: Interfaz Pago Tarjeta de Easy-Bank Vision, (Elaboración propia, 2024).

4.2.4 Diseño de la secuencia de interacción

Como se indica en la tabla 18 los requerimientos obligatorios están estrechamente ligados en el diseño de las interfaces que moldean el uso de la interfaz. Es preciso mencionar que, si bien el investigador encargado tenía influencia en las decisiones del equipo de diseño, este último fue responsable de la mayoría de las decisiones creativas y de delimitación de tiempos de cada Sprint del proceso de desarrollo de software.

Para que el equipo tuviera claros los objetivos de desarrollo y pudiera relacionar las propuestas de diseño interfaces de estudio con una evaluación mediante eye-tracking se diseñó un diagrama de casos de uso de Easy-Bank visión. Como se observa en la figura 64, existe un único usuario que interactúa con el sistema de experimentación. También se indica en el diagrama la secuencia esperada de uso. Dentro de los casos de uso existe un caso en particular denotado con la nota de “include”, esta notación hace referencia a que el caso “Home” incluye el apartado y las herramientas necesarias para realizar el caso de uso “Pago Tarjeta”. Caso contrario, son los elementos de “Transferencias” y “Saldos y movimientos” casos de uso que proporcionan sus propias interfaces y se accede a estas de manera secuencial desde el caso de uso “Home”.

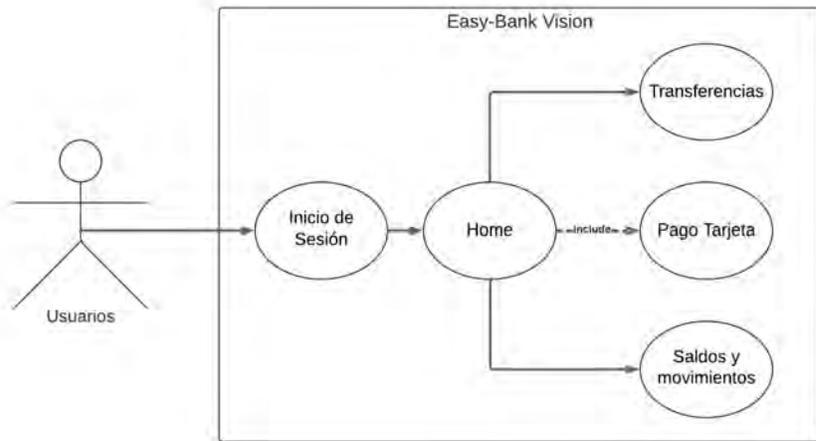


Figura 66: Diagrama de casos de uso para Easy-Bank Vision, (Elaboración propia, 2024).

El diagrama de casos de uso permite identificar fácilmente las interfaces o interacciones que se aplicaron en el proceso de experimentación y para respaldar los anterior, se diseñó un diagrama de

estructura de navegación en el cuál, tanto el equipo de desarrollo como de evaluación podían consultar para desarrollar los módulos funcionales a los primeros y el diseño experimental a los segundos.

Como se observa en la figura 67, es posible acceder a los distintos casos de estudio mediante la secuencia indicada en el diagrama de navegación del caso de estudio. Esta secuencia es la planteada para analizar mediante eye-tracking.

Como se ha mencionado en distintos puntos de este caso de estudio, el equipo de desarrollo ejecutaba diferentes fases del desarrollo del producto de manera síncrona, y uno de los aspectos que se consideró desde el momento cero del proceso de desarrollo era contemplar la evaluación mediante eye-tracking, por lo que desde etapas muy tempranas se propusieron distintos diseños experimentales, las tareas que el usuario tenía que ejecutar y la secuencia de procesos que mejorar la capacidad de evaluar objetivamente las interfaces.

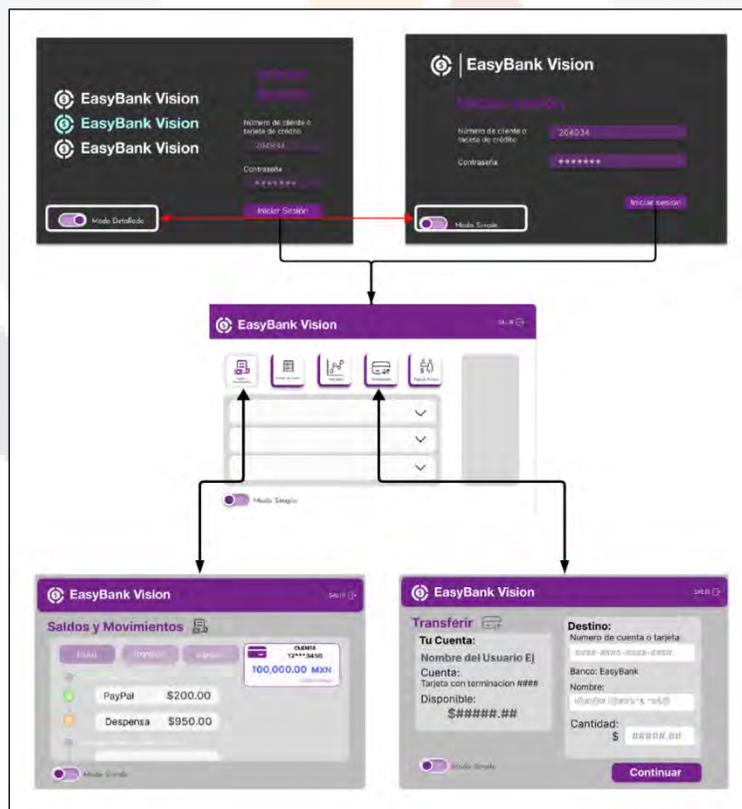


Figura 67: Diagrama de navegación de Easy-Bank Vision, (Elaboración propia, 2024).

Si se plantera una evaluación de interfaces convencional la secuencia de interacción propuesta no representaría a una complejidad muy elevada, sin embargo, al implementar las técnicas de eye-tracking se convierte en un proceso mucho más complejo, pero que brinda resultados mucho más precisos que el método convencional.

Para ampliar la idea anterior, se expresa en la figura 68 el diagrama del protocolo experimental que refleja cada uno de los procesos necesarios para que un producto como lo es Easy-Bank Vision pudiera implementar de la manera más precisa posible el marco de trabajo EVUSINET y posteriormente una evaluación de interfaces de INUSINET.

Abordando más detalladamente los procesos del protocolo experimental se encuentran cuatro elementos que representan cada uno de los tipos de usuarios establecidos para las pruebas de estudio. Esta especificación hace referencia que, si bien las tareas e instrucciones son las mismas para cada sujeto, se esperaba que algunos de los tipos de perfiles tendrían poca experiencia con el uso de equipos de cómputo, por lo que el equipo identificaba desde etapas tempranas las tareas que serán necesarias explicar para cada diferente sujeto.

Contrariamente a sesgar los resultados, la identificación temprana del tipo de usuario y las expectativas que se tenían sobre cada uno de estos, permite tanto los diseñadores de interfaces, desarrolladores y principalmente evaluadores sensibilizarse con las características de uso de cada tipo de perfil, generando diseños más cercanos a las expectativas de estos.

Otro elemento que difiere de este diagrama de protocolo experimental que se implementa como una fase particular de este protocolo de experimentación es el de "Asignar cuenta y tarjeta física". En este proceso se le asigna de manera arbitraria una cuenta y un objetivo físico que simula ser una tarjeta bancaria que cuenta con datos específicos que serán usados en la interacción de uso de experimentación.

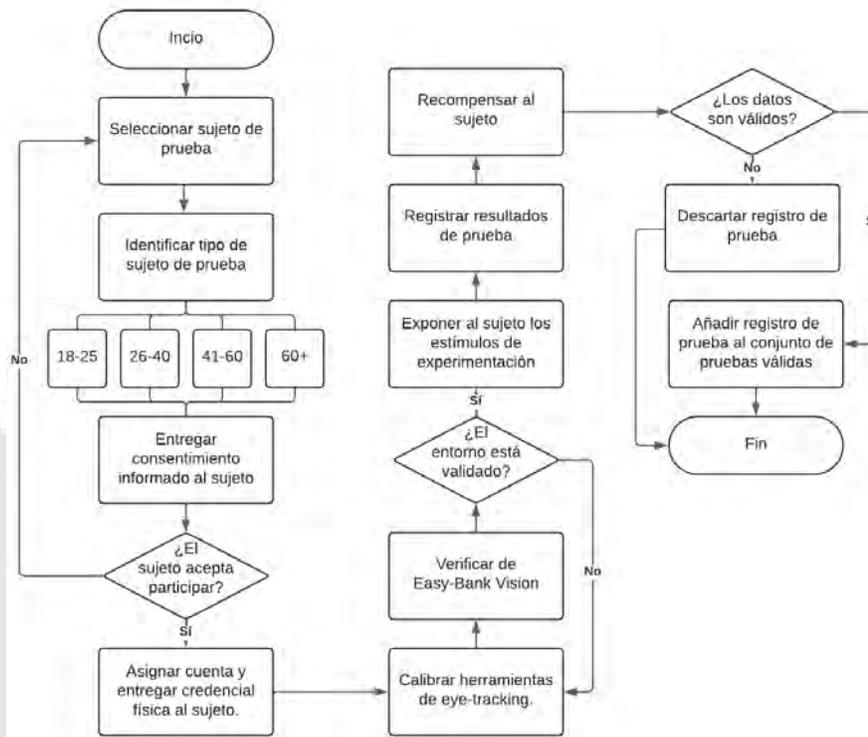


Figura 68: Diagrama de protocolo experimental de Easy-Bank Vision, (Elaboración propia, 2024).

4.2.5 Implementación de técnicas y conceptos de eye-tracking

En el caso de estudio del componente login las dimensiones de este proyecto y su inherente valuación del software respondió principalmente a probar la viabilidad de implementar las propuestas que relacionan el eye-tracking con la ingeniería de software. En este caso de estudio se propusieron dimensiones mucho mayores para probar que tan potente es el modelo y los índices mientras, además de que el tamaño de la población es mayor, también los tipos de usuarios son más variados.

Abordando más detalladamente sobre la cantidad de la población y los tipos de usuario, es preciso mencionar que se realizaron 22 pruebas con diferente cantidad de perfiles de usuario (expresados en la sección 4.2.2 apartado C) todas realizadas en las mismas condiciones controladas y siguiendo al pie de la letra el protocolo experimental de la figura 68.

En etapas iniciales del desarrollo se generaron más casos de uso para el caso de estudio del componente login, sin embargo, cuando el sistema alcanzó una etapa de madurez tal que permitió

generar una prueba piloto sobre el proceso de experimentación se descubrió que tener una experiencia tan larga en tiempo podría provocar la cancelación de la prueba por parte del usuario, o que sencillamente muchos de los sujetos sondeados para la prueba rechazaran realizarla al saber el prolongado tiempo que tendrían que invertir en la prueba.

Por lo anterior, algunos de los casos de estudio tuvieron que ser retirados de la experiencia para, principalmente, generar análisis más detallados y minuciosos de otras interfaces que se consideran mucho más relevantes en el proceso, tomando la frase de “menos es más” como dogma para la evaluación de esta experiencia más compleja y con más interfaces que el componente login. A continuación, se enlistan cada uno de los procesos elaborados mediante el software de TobiiProLab que permitieron unir ramas de la ingeniería de software con la teoría de atención visual e implementación de los conceptos y marcos de trabajo propuestos para su evaluación.

A. Diseño experimental de Easy-Bank Vision mediante EVUSINET

El diseño experimental de la evaluación del caso de estudio se representa en la figura 69. Como se puede observar, el total de la prueba consta de 18 procesos secuenciales e independientes uno del otro. Para generar los contenidos de los procesos la herramienta de diseño de TobiiProLab cuenta con diferentes elementos preestablecidos como lo son etapas de calibración, ventanas de texto y el despliegue de sitios WEB mediante la asignación de URL para exponer los estímulos al usuario y definir el proceso de experimentación.

En el caso de este diseño experimental, después de realizar la calibración, se procede a mostrar las instrucciones de la prueba, seguidas de una de las varias instrucciones previas a cada interfaz (denotadas con el prefijo “Ins.”). Estas instrucciones mencionan tareas específicas y objetivos a cumplir por parte del investigador, sin embargo, los usuarios no tenían límite de tiempo y eran libres de explorar el sistema libremente. Cuando el usuario notificaba al investigador que había completado la tarea, este terminaba el proceso de manera manual y automáticamente iniciaba el siguiente en la lista de procesos.

Para generar los procesos de interacción o estímulos se utilizó el elemento para desplegar URL como se mencionó anteriormente. El primero de estos URL fue el de la ventana login, el cual, una vez que el usuario ingresaba al sistema siempre se dirige a la ventana “Home” (como se observa en la figura 64 referente al mapa de estructura de navegación). La ventana “Home” es la más recurrente de

todas las interfaces de estudio debido a que desde aquí es posible dirigirse a otros casos de uso de la aplicación.

En la figura 70 se representa un ejemplo de los parámetros que tiene el elemento para desplegar una URL como un estímulo de evaluación mediante eye-tracking. Es a partir de los datos recolectados por este elemento en particular que se generan las métricas de atención visual sobre el estímulo que, en este caso, es un producto de software diseñado para ser evaluado precisamente de esta forma.

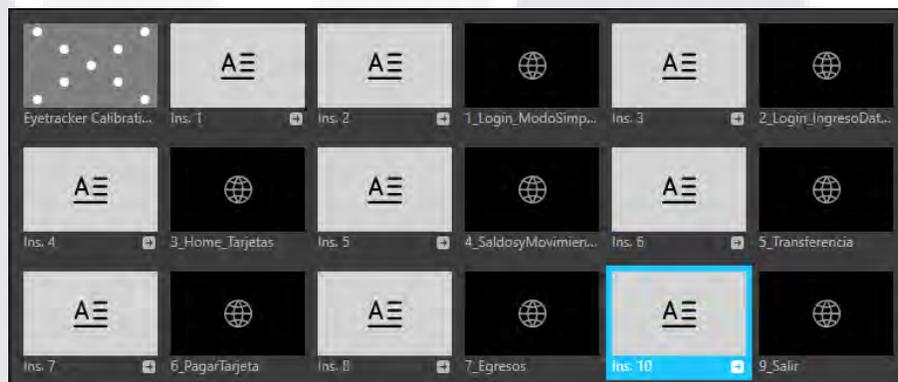


Figura 69: Diseño experimental para la evaluación de Easy-Bank Vision en la herramienta de Diseño de TobiiProLab, (Elaboración propia, 2024).

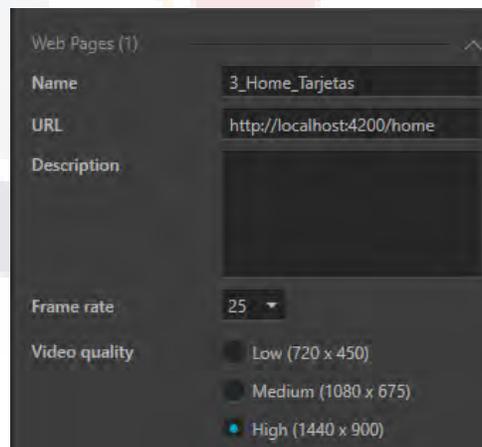


Figura 70: Uso de la herramienta para desplegar URL como un estímulo de TobiiProLab, (Elaboración propia, 2024).

B. Implementación de eye-tracking en el análisis de Easy-Bank Vision mediante EVUSINET

En este apartado se muestran las áreas de interés y eventos que se diseñaron para poder generar un análisis de cada una de las interfaces. Ahora bien, a diferencia del caso de estudio del componente login, no hay una secuencia fija de interacción con diferentes interfaces, es decir, después de presentar una primera interfaz no se despliega la segunda a continuación, si no que las interfaces contienen un botón la cuál cambia la distribución o dimensión de los elementos en pantalla u ofrece información extra mediante otro elemento que anteriormente no era visible.

Por lo anterior se estableció la regla en la que, por mucho que una sola interfaz complique la evaluación mediante eye tracking se evalúa cada una de ellas de manera separada. Como se observa en las figuras 71 y 72 cada una de las variantes de la interfaz genera sus propias áreas de interés que evaluar. Si bien los cambios no son mayores en el diseño, dan un primer acercamiento al estudio de la influencia de la disposición de los elementos en el uso del sistema.

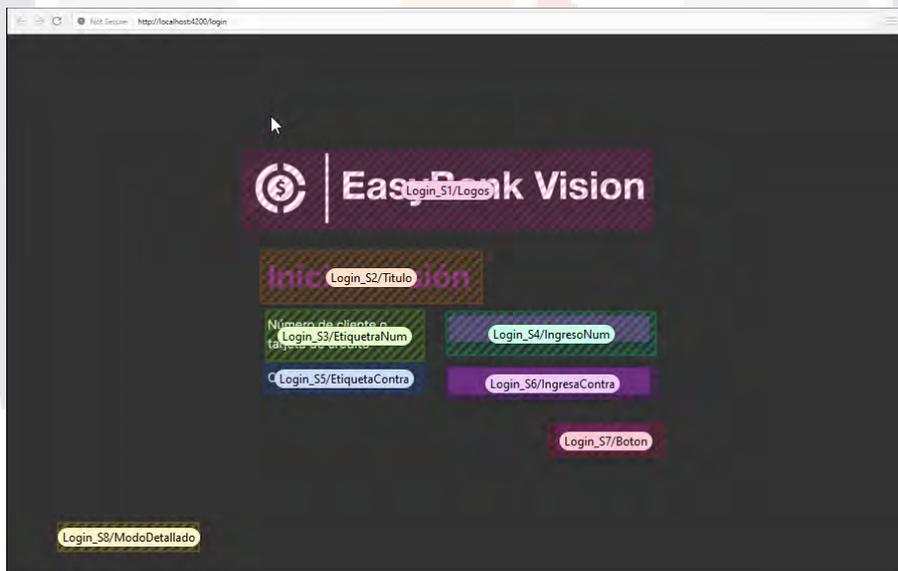


Figura 71: Áreas de Interés para la interfaz simple del Login de Easy-Bank Vision, (Elaboración, propia, 2024).

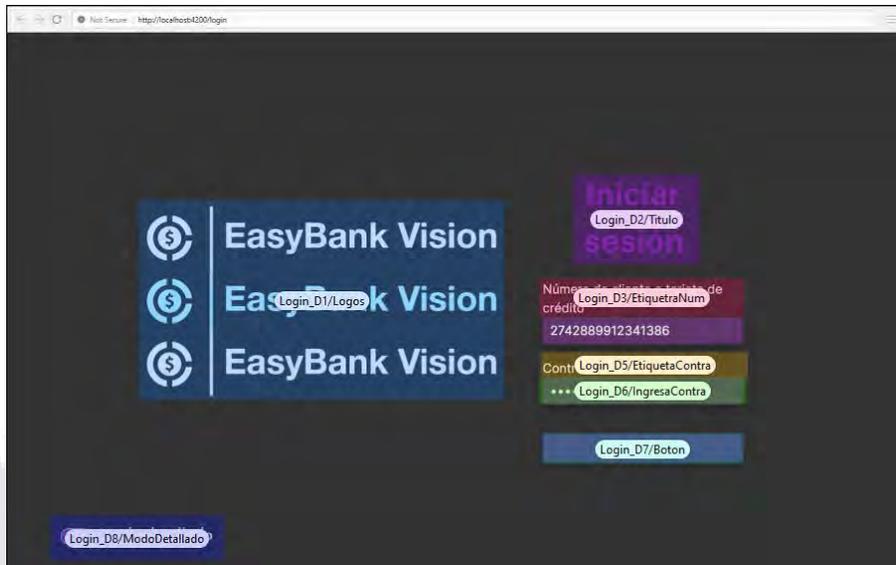


Figura 72: Áreas de Interés para la interfaz detallada del Login de Easy-Bank Vision, (Elaboración, propia, 2024).

En los demás casos, pulsar el botón de “Modo Detallado” hace visibles elementos que en la versión “Simple” de la interfaz no se muestran. Esta información, como se observa en las figuras 73, 74 y 75 correspondientes a los casos de uso de saldos y movimientos, pagar tarjeta y transferencias, se representa en un área de interés de “Ayuda” la cual cuenta con indicaciones y descripciones del uso de los elementos necesarios para completar la tarea más fácilmente o si por si el usuario está teniendo problemas para completarla.

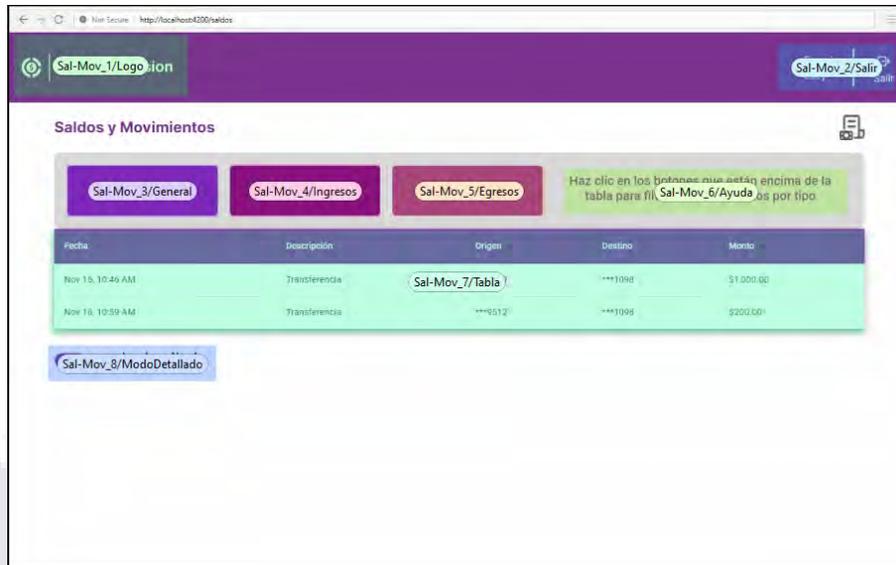


Figura 73: Áreas de Interés para la interfaz de saldos y movimientos de Easy-Bank Vision, (Elaboración, propia, 2024).

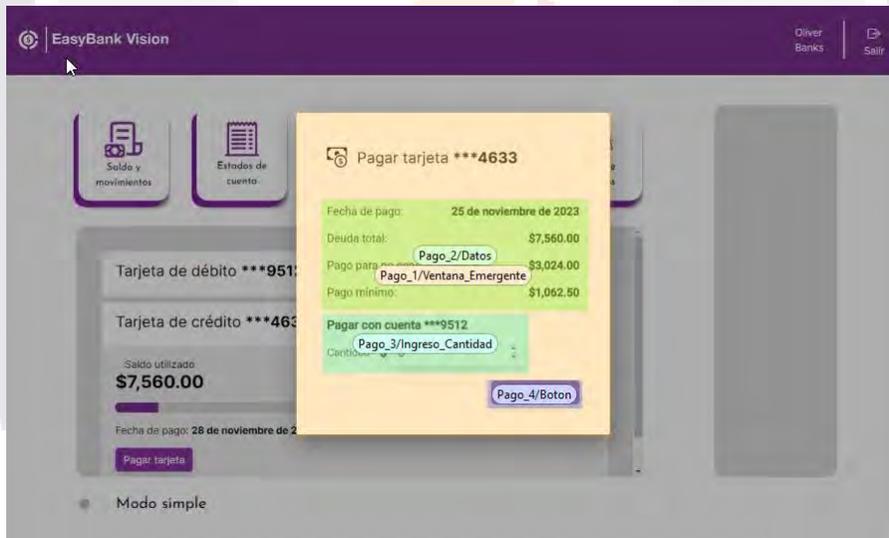


Figura 74: Áreas de Interés para la ventana emergente del caso de uso Pagar Tarjeta de Easy-Bank Vision, (Elaboración propia, 2024).

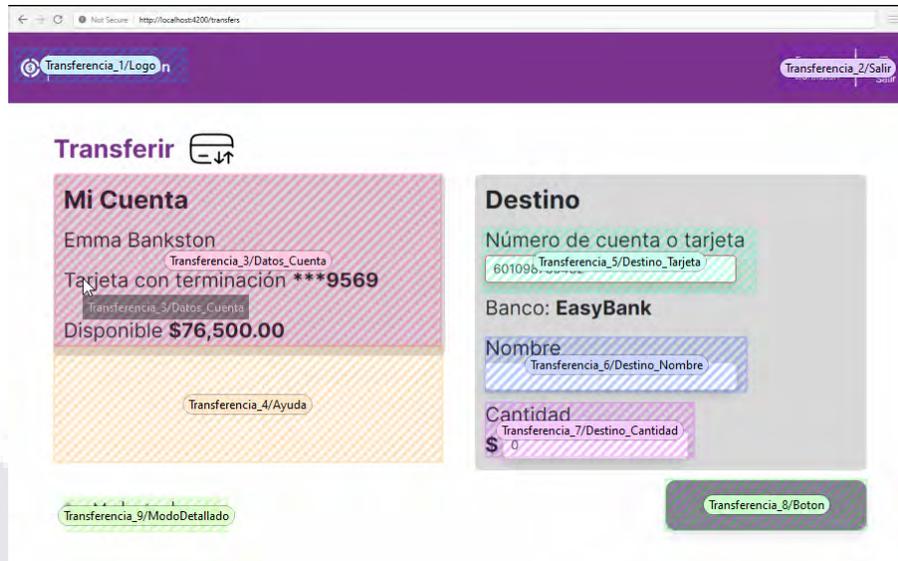


Figura 75: Áreas de Interés para la interfaz Transferir de Easy-Bank Vision, (Elaboración propia, 2024).

Dada la complejidad de este diseño experimental de Easy-Bank Vision utilizar el diagrama de recomendado de por el marco de trabajo EVUSINET para visualizar la secuencia de eventos no brinda de manera integral ni certera la información que se desea expresar, por lo que, y tomando como referencia la ventana para establecer periodos de *TobiiProLab*, la figura 76 representa de manera más fidedigna la secuencia de experimentación del usuario.

El diagrama de secuencia de eventos está inspirado en el diagrama de secuencia de UML y la herramienta de eventos de *TobiiProLab* (véase la figura 77), cada uno de los eventos representados en el diseño experimental eventualmente son expuestos al usuario, sin embargo, en algunos casos las métricas obtenidas no surgen de un solo evento, si no de un conjunto de estímulos presentados al usuario a lo largo de una prueba.

Para dar una lectura correcta al diagrama de eventos se sugiere comenzar por identificar las interfaces que intervienen en el uso del sistema, las cuáles se ubican en la parte izquierda del diagrama. Después de la calibración hay una entrada llamada "Instrucciones" la cual se representa varias veces a lo largo de toda la prueba. Esta característica indica que antes de interactuar con una interfaz previamente hay que exponer al usuario a un elemento de texto. Caso parecido al anterior es el estímulo "Home", el cual, como se mencionó anteriormente, contiene un caso de estudio particular (pagar tarjeta) y permite completar otras tareas como lo son transferencias y saldos y movimientos. Sobre estos últimos, existen dos periodos en los que interviene este estímulo, por lo

que, como se alcanza a distinguir en el diagrama, hay dos cuadros en la fila de “Saldo-Movimientos”.

Finalmente, cada uno de los cuadros en color verde representan la exposición de la interfaz al usuario en la cual se registrarán los datos de atención visual registrados por la herramienta de eye-tracking *TobiiProNano* y generarán los data-set correspondientes para implementar INUSINET en la evaluación de eye-tracking.

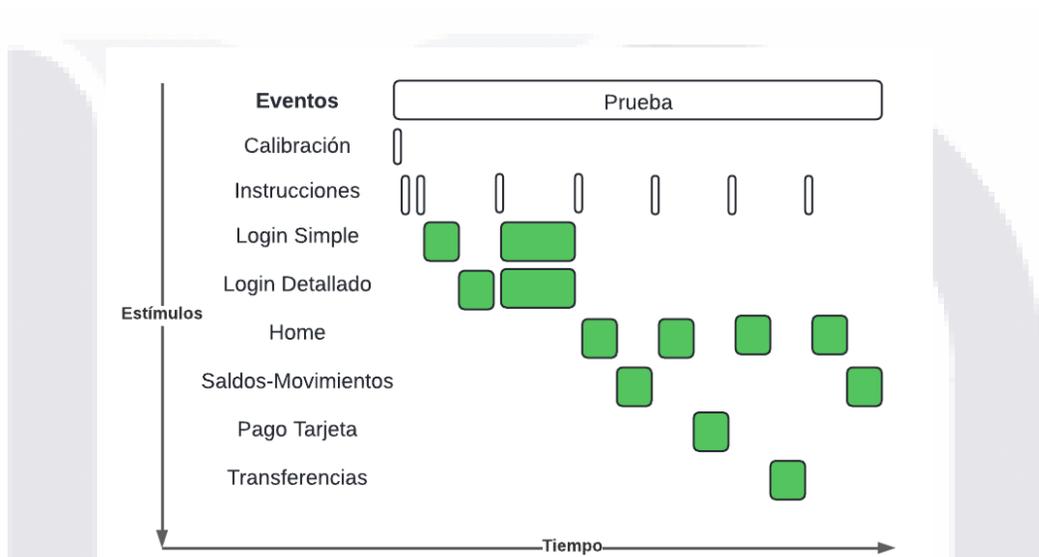


Figura 76: Diagrama de secuencia de eventos de las pruebas de Easy-Bank Vision, (Elaboración propia, 2024).

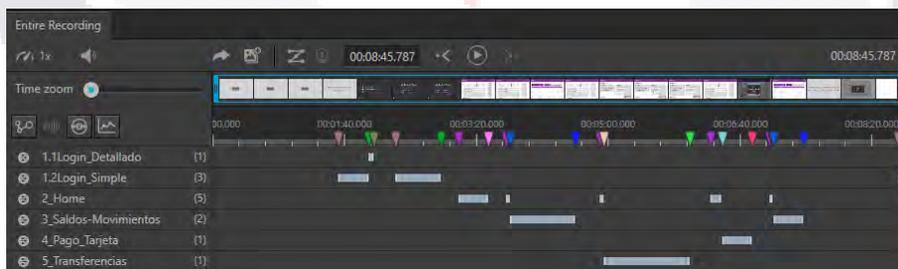


Figura 77: Eventos establecidos en una prueba de Easy-Bank Vision, (Elaboración propia, 2024).

C. Evaluación mediante eye-tracking de Easy-Bank Vision

Debido a que la experiencia que se busca evaluar contiene diversos casos de uso y diversas interfaces el procedimiento para generar los data-sets correspondientes a cada una de las etapas de las pruebas fue una tarea un tanto laboriosa. Una de las principales hipótesis que se tiene sobre la implementación de las técnicas de eye-tracking es que, si bien estas pueden consumir un tiempo importante en el proceso de desarrollo, la información que brinda sobre el uso de los participantes hace que el esfuerzo y tiempo sean redituables a medio y largo plazo.

Para generar las diferentes visualizaciones y evaluaciones particulares, y como se muestra en el diagrama de secuencia de eventos, existen estímulos que son expuestos al usuario varias veces. Para poder generar resultados generales de cada interfaz se generan las sumatorias de cada una de sus apariciones en el proceso experimental.

Esta agrupación de procesos de una interfaz específica permite obtener resultados del uso de cada una de ellas de manera independiente de las demás, pero más importante, durante toda la prueba.

A continuación, se exponen los mapas de calor, mapa de recorrido de abeja y mapa de fijaciones de cada interfaz de una prueba tomada aleatoriamente de la población con el fin de comprobar si existen patrones reconocibles para un individuo, y de ser así, generar resultados poblacionales.

a) Mapas de calor

Como ya se abordado varias veces en la investigación, los mapas de calor son una herramienta útil para conocer las áreas más visitadas de una interfaz específica. En la figura 78 son muy notorias las diferencias en el patrón de uso para cada interfaz. De entre las visualizaciones destacan los mapas de calor de la interfaz de login, las cuales representan resultados muy diferentes a pesar de que sus diseños tienen muchas similitudes, lo cual demuestra enormemente la influencia de la disposición de los elementos en los patrones de uso generados.

Uno de los casos de uso que no se había estudiado directamente mediante eye-tracking eran las ventanas emergentes, por lo que existía en los investigadores la duda si los resultados de este estímulo serían fácilmente identificables. Una vez realizada la visualización es evidente que las propiedades de cada interfaz o elemento son fácilmente identificables en una prueba de eye-tracking.

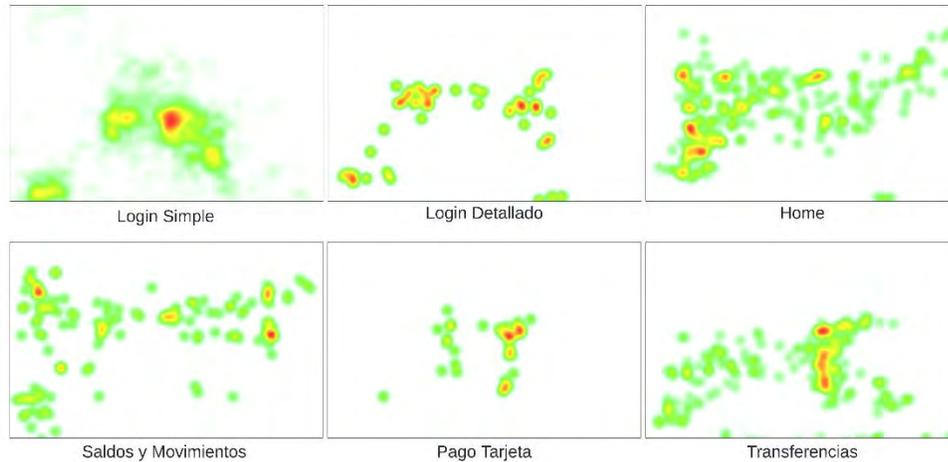


Figura 78: Mapas de calor de una sola prueba del conjunto de interfaces de estudio de Easy-Bank Vision, (Elaboración propia, 2024).

b) Mapa de recorrido de abeja

Ampliando los resultados anteriores, los patrones de mapa de abeja permiten reconocer las transiciones entre los puntos más destacados en el mapa de calor. En el caso de esta prueba aleatoria existen patrones bien definidos en cada recorrido y es preciso destacar que la ventana Home cuenta con transiciones que forman una figura parecida a la letra “F”, este patrón es comúnmente visto en páginas web en dónde la información está correctamente representada y es sencillo de leer para el usuario ya que, como se menciona en el apartado A de la sección 3.7.4 apartado A, los patrones de información se pueden leer de izquierda a derecha y de arriba abajo, lo que coincide con la experiencia previa por parte de este usuario (véase figura 79).

Otra interfaz para destacar es la de Login Simple ya que agrupa la mayoría de las fijaciones en un solo punto, lo que invita a pensar que posible los elementos no tengan la correcta dimensión y distribución en la pantalla, sin embargo, para aventurarnos a generar esta interpretación es necesario realizar una evaluación poblacional.

Los demás elementos siguen un patrón principalmente horizontal a lo ancho de toda la pantalla, como lo son el Login Detallado y Saldos y Movimiento. En este caso, es la pestaña transferencia la que representan un patrón relativamente diferente ya que son más notorias las áreas que cuentan con más fijaciones. En este caso, y reiterando que para proponer estas interpretaciones es necesario realizar pruebas con una muestra mayor o con la población entera, se podría establecer que los

elementos están distribuidos de tal manera que acceder a cada uno de ellos es complicado para el usuario.

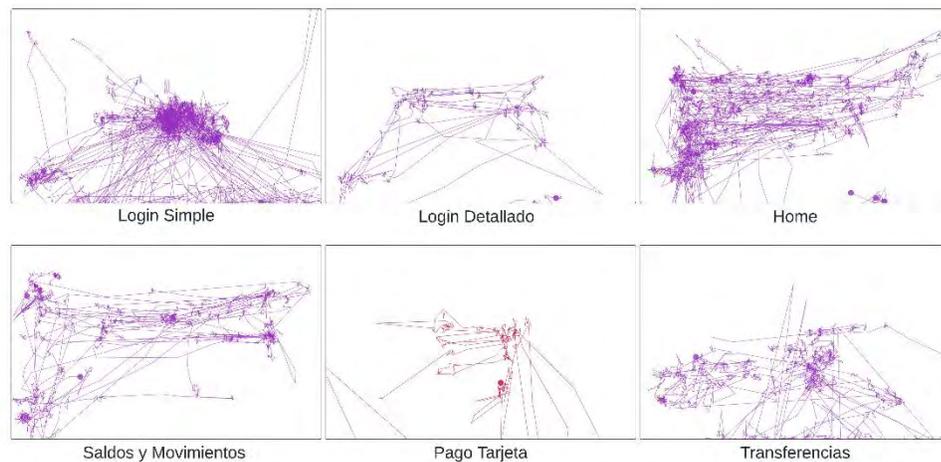


Figura 79: Mapa de recorrido de abeja de una sola prueba del conjunto de interfaces de estudio de Easy-Bank Vision, (Elaboración propia, 2024).

c) Mapa de fijaciones

Si bien los diagramas de recorrido de abeja y de recorrido de fijaciones son muy parecidos, es posible obtener información de estos últimos sobre los tipos de fijaciones a estimando “a ojo” estas.

Si se observa detenidamente la visualización de las fijaciones de la interfaz transferencia (contenida en la figura 80) es posible notar de lado izquierdo que existen muchas fijaciones pequeñas y dentro del área que forman las fijaciones pequeñas existen otras notablemente más grandes. Si bien las fijaciones pequeñas y con movimientos sacádicos cortos pueden indicar un esfuerzo de lectura, en el caso de las expectativas del diseñador de este elemento es proveer al usuario información únicamente de control, la cual no es una fuente de información necesaria para completar la tarea. Entonces, volviendo a las posibles interpretaciones que se le puede dar a la visualización, es posible que este elemento en particular esté molestando visualmente al sujeto.

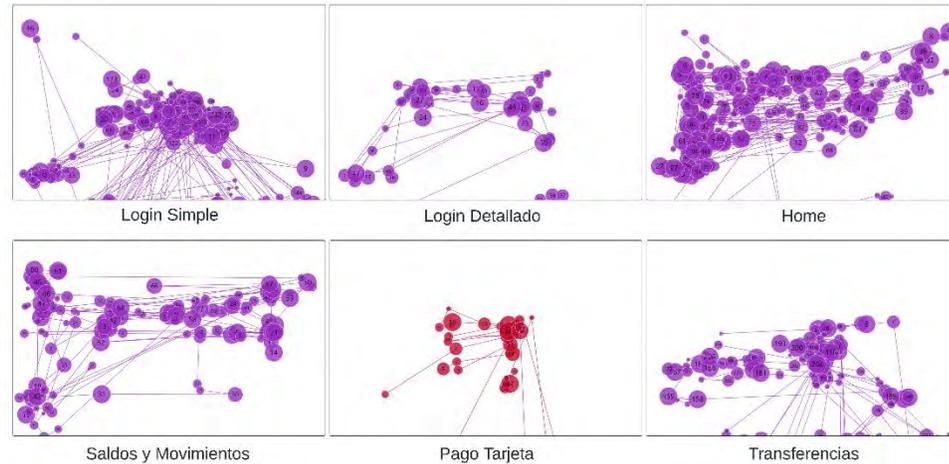


Figura 80: Mapas de recorrido de fijaciones de una sola prueba del conjunto de interfaces de estudio de Easy-Bank Vision, (Elaboración propia, 2024).

Como se ha mencionado en los apartados anteriores, para poder generar resultados más profundos se requiere realizar pruebas poblaciones o con una muestra mayor para comprobar las posibles interpretaciones de cada interfaz, y también estimar la cantidad de información que este tipo de evaluación arroja.

d) Data-set para el análisis de patrones de uso particulares

Para generar resultados poblaciones es necesario generar *exports* individuales para cada interfaz, esta tarea implica relacionar las áreas de interés de cada interfaz con los eventos generados manualmente para cada sujeto de prueba mediante un conjunto de botones seleccionables que habilitan la entrada de ese campo de información en el *export* (véase figura 82).

Cabe mencionar que si bien para un investigador experimentado esta tarea puede parecer rutinaria y sencilla, para un investigador con “menos horas de vuelo” en la herramienta de *TobiiProLab* puede resultar todo un rompecabezas. Para desarrollar esto, en la misma figura antes mencionada sobre los *exports* de *TobiiProLab* se observa que de lado izquierdo se presentan las posibles métricas que el data-set contendrá, al enfocar este análisis en INUSINET son las variables de este modelo las que se seleccionan en este apartado.

De lado derecho se encuentran los botones seleccionables antes mencionados en los que, de la parte superior de la pantalla se selecciona el estímulo (interfaz de estudio), en este caso Login

Detallado, posteriormente, en la parte inferior, se despliega el conjunto de pruebas realizadas las cuáles se dividen en cada uno de los eventos de la prueba, por lo que, para filtrar la información es necesario seleccionar manualmente cada usuario de cada evento seleccionando las áreas de interés del estímulo y dejando sin marcar las otras áreas de interés que no corresponden a la interfaz.

Lo anterior se debe a que siguiendo la filosofía de evaluación de interfaces mediante eye-tracking que dicta: “Cada pequeño cambio en la distribución, dimensión o diseños de fuente y estilo de un elemento se considera como otra interfaz, y cada interfaz debe ser analizada individualmente”, por lo que, al tener un elemento para cambiar las propiedades de la interfaz se debe generar dos evaluaciones en una misma interfaz para el caso de Login Simple y Login Detallado.

A su vez, la secuencia de experimentación para completar una tarea después de ingresar al sistema siempre comenzaba con la pestaña Home, por lo que, para evaluar Saldos y Movimientos, la cual cuenta con sus propias áreas de interés, antes de llegar a este punto se tiene que analizar también la pestaña Home junto con sus áreas de interés, es por esta razón que como se observa en la figura 81 el evento de experimentación que busca evaluar Saldos y movimientos, también cuenta con las áreas de interés de Home.

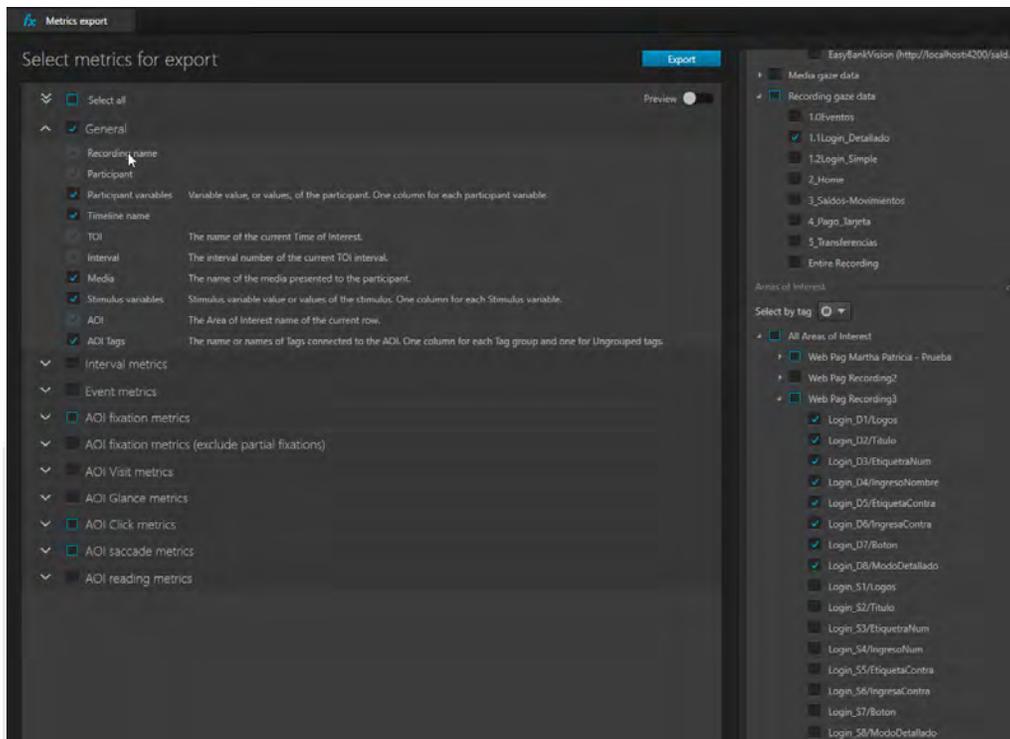


Figura 81: Ventana de generación de exports de la herramienta TobiiProLab para obtener resultados de la interfaz de estudio Login Detallado, (Elaboración propia, 2024).

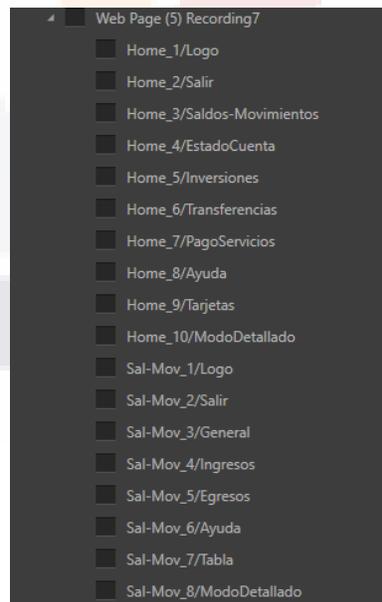


Figura 82: Áreas de Interés de Home y Saldos y Movimientos congregados para analizar ambas interfaces individualmente, (Elaboración propia, 2024).

A partir de este punto ya se tiene información poblacional de cada interfaz con sus respectivos registros de las métricas del modelo INUSINET. En la siguiente sección se exponen los resultados de la evaluación de esta primera versión de Easy-Bank Vision ya que, como se pudo observar, los patrones visuales de cada interfaz son reconocibles en un solo usuario y es conveniente continuar con una evaluación más amplia (con respecto a la muestra de sujetos de prueba) y precisa (al evaluar el uso mediante INUSINET).



5. Resultados y Discusiones

En este capítulo se abordan los resultados de aplicar las técnicas y conceptos de eye-tracking en el par de casos de estudio generados para probar y compartir los procedimientos y métodos desarrollados. En las secciones siguientes se expresan los resultados tanto de los patrones de uso generales como particulares para dar pie a la posterior discusión de los resultados obtenidos.

Es preciso mencionar que si bien la tecnología de eye-tracking entrega resultados que pueden llegar a ser muy precisos, las métricas como tal están fundamentadas principalmente en el movimiento ocular, mas no en las interpretaciones previas o las reacciones posteriores que el proceso cognitivo realiza a la hora de ver un estímulo. En otras palabras, obtener información precisa del uso de una interfaz no representa un enorme grado de dificultad, como sí lo tiene generar una interpretación válida y coherente del uso de la interfaz.

Y se podría preguntar: “¿qué es lo que hace tan complicado generar una interpretación de los resultados obtenidos?”. Si bien en esta misma investigación y más específicamente en la propuesta INUSINET se plantean un conjunto de interpretaciones que están respaldadas por otros trabajos de investigación referentes a las tecnologías de eye-tracking y atención visual, la unión con la ingeniería de software amplía el análisis por un grandísimo factor.

De manera convencional las pruebas de eye-tracking manejan estímulos con diseños y tiempos definidos, lo que facilita generar comparaciones muy precisas de un determinado objeto o figura que se muestra, pero en el caso del análisis de interfaces en la mayoría de los casos no existen periodos de tiempo plenamente definidos. A su vez, probar las capacidades de una interfaz solo por su diseño no es un análisis que, desde punto de vista del investigador de este trabajo, ya no es suficiente tomando en cuenta la cantidad y calidad de ciertos productos que están en el mercado.

Ampliando la idea anterior, en la sección 3.4 referente a la metodología de diseño de interfaces paralelas se aborda el concepto de “la supervivencia del más apto”. En el contexto de esa sección se usa el concepto como una forma de representar una pelea entre los elementos de cada propuesta de diseño para prevalecer en el diseño final. Sin embargo, el concepto también hace referencia a la necesidad de las propuestas de interfaces de ser mejores que sus rivales en el mercado.

Lo anterior es posible realizado de manera convencional, es decir, probando poco a poco las características de interacción que tendría el usuario hipotético, pero esto conlleva grandes

derroches en tiempo y dinero para el equipo de desarrollo. Es por esta razón que los resultados de los casos de estudio están enfocados en presentar información lo suficientemente general para que pueda ser implementado en muchos procesos de desarrollo e interfaces de distintas características, pero lo suficientemente específico para que la información permita al diseñador, desarrollador y evaluador aumentar la certidumbre del uso de sus diseños en el público y disminuir el tiempo empleado en encontrar de manera arbitraria los mejores elementos y su diseño para la experiencia que desea desarrollar.

5.1 Resultados de Caso de Estudio 1: Componente Login

En el caso de estudio de componente login se utilizó la metodología de diseño de interfaces paralelas, que busca confrontar directamente dos diseños para conocer los puntos más débiles de una interfaz para corregirlos, cambiarlos o suprimirlos. Lo anterior permite comparar además de las características de cada elemento, los efectos de la distribución, dimensión y, en menor medida, la efectividad de las fuentes y tamaños de texto.

5.1.1 Resultados de patrones de uso generales

En la sección de desarrollo del componente login se expresaron los principales métodos de visualización de las diferentes interfaces, sin embargo, igual que como se aborda en dicha sección, algunas de estas visualizaciones contienen más información y más precisa y útil para que el evaluador tenga mayores referencias sobre el uso de la población de los sujetos de pruebas que participaron en las pruebas.

Uno de los aspectos más notorios en las visualizaciones de las interfaces es la influencia de la distribución de los elementos. Anteriormente, en las figuras 53 y 54 se había podido visualizar mediante los mapas de calor que, en definitiva, el patrón de uso era diferente para cada interfaz, sin embargo, como se observa en las figuras 85 y 86 es muy relevante notar que la población de sujetos de prueba tuvo patrones de uso similares, incluso idénticos, debido a que estos nunca recibieron instrucciones específicas de uso y era la primera vez que se exponían a este estímulo.

Si se retoma el diseño de la interfaz 1 (representada en la figura 41) es posible relacionar las características de los elementos en general, desde el título, hasta los cuadros de texto para ingresar

el nombre de usuarios y contraseñas y el botón de enviar formulario con los recorridos de abeja de la figura 83. Incluso en esta visualización es posible determinar que existen ciertos puntos dónde es más recurrente la atención visual y, para confirmar esto, se generó el mapa de calor relacionado a la interfaz 1, si bien en la figura 55 existe cierta divergencia en los puntos más visitados por un único participante, la figura 83 concreta los puntos más visitados en la interfaz que se ubican en la parte derecha central de la pantalla. Igualmente, las figuras 85 y 86 corroboran los puntos donde más se presenta atención visual en la interfaz que representan.

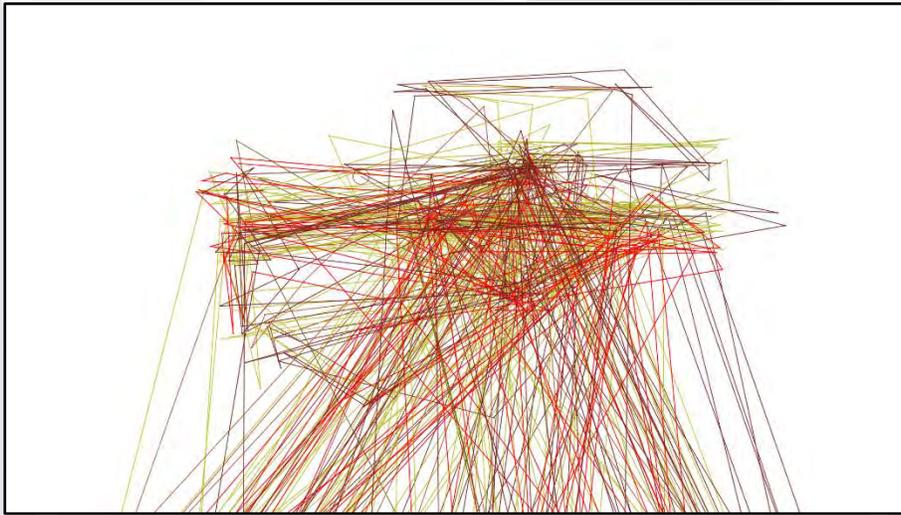


Figura 83: Visualización en recorrido de abeja del patrón de uso de la interfaz 1 de la población de sujetos de prueba, (Elaboración propia, 2024).



Figura 84: Visualización en recorrido de abeja del patrón de uso de la interfaz 2 de la población de sujetos de prueba, (Elaboración propia, 2024).



Figura 85: Mapa de calor de los resultados de la población de sujetos de prueba en la interfaz 1, (Elaboración propia, 2024).



Figura 86: Mapa de calor de los resultados de la población de sujetos de prueba en la interfaz 2, (Elaboración propia, 2024).

Finalmente, para no dejar de lado ninguna de las visualizaciones se representa en las figuras 87 y 88 los patrones de recorrido con información de las fijaciones. Estas fijaciones resultan más complejas de evaluar cuando se contraponen los datos de toda la población, sin embargo, también confirman los patrones antes encontrados en las demás visualizaciones.

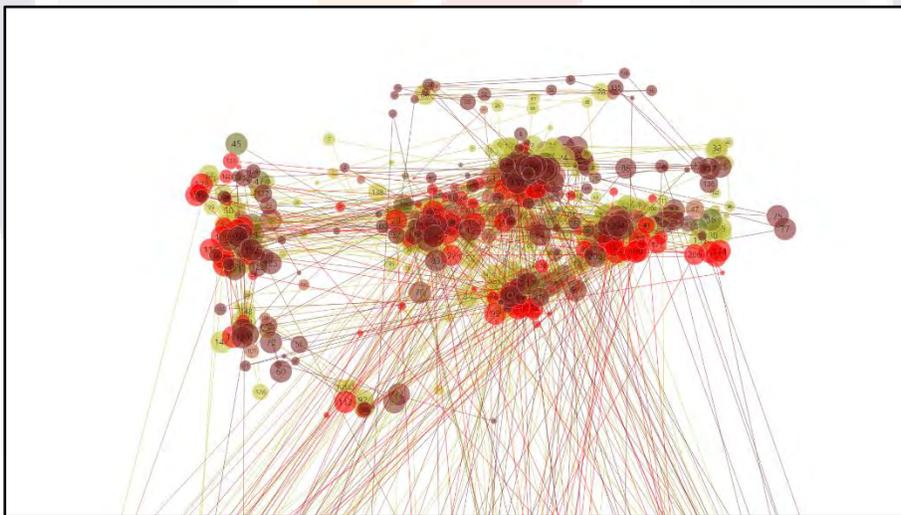


Figura 87: Visualización de ruta de recorrido con información de fijaciones de la interfaz 1 de la población de sujetos de prueba, (Elaboración propia, 2024).

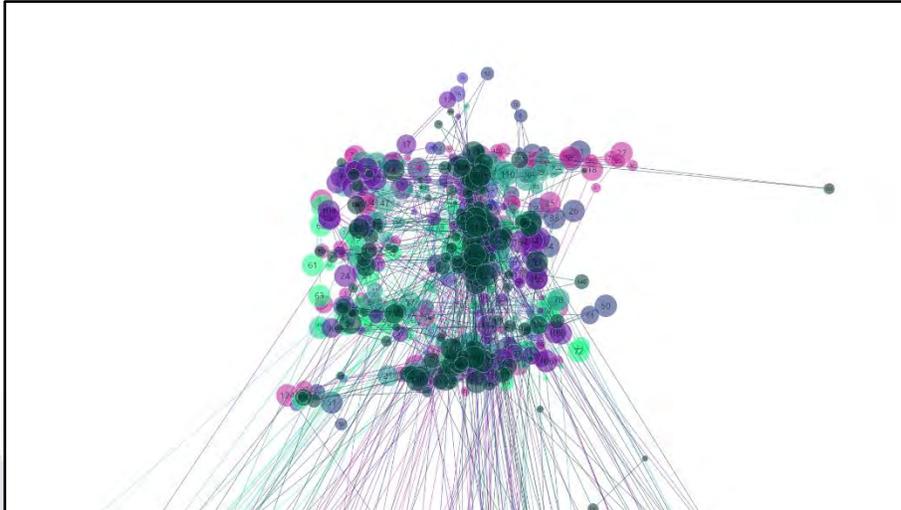


Figura 88: Visualización de ruta de recorrido con información de fijaciones de la interfaz 2 de la población de sujetos de prueba, (Elaboración propia, 2024).

5.1.2 Resultados de patrones de uso particulares.

Como se ha mencionado anteriormente, los patrones de uso particulares tienen la capacidad de indicar al evaluador los puntos más débiles de la interfaz, y al diseño le ofrecen un punto de inicio para que sus futuros diseños corrijan, mejoren o supriman los elementos que no están generando una buena experiencia.

El primer paso para generar una evaluación particular de cada elemento es establecer las variables que cada interfaz posee siguiendo las nomenclaturas de INUSINET. Para establecer esto, hay que identificar y nombrar las interfaces que se diseñaron para este caso de estudio.

Como se ha manejado en secciones anteriores, las interfaces siguen dos patrones de diseño diferentes, uno enfocado en información dinámica y volátil que permita tener menos texto en pantalla y el usuario pueda dar solución al formulario de manera intuitiva. A este interfaz se le denomina “Interfaz dinámica”. Por otro lado, la interfaz 2 responde al concepto más fundamental del formulario en donde cada campo de entrada es acompañado de una etiqueta, completando la tarea de forma secuencial y lineal, a esta interfaz se le denomina “interfaz de formulario convencional”, pero para fines de mejorar la legibilidad de las ecuaciones se le denomina “interfaz convencional”. Quedando las variables de la siguiente manera:

$I_{Dinámica} \rightarrow$ *Interfaz Dinamica*

$I_{Convencional} \rightarrow$ *Interfaz Convencional*

A. Evaluación de Interfaz dinámica

El primer paso para generar las evaluaciones es generar la nomenclatura de INUSINET y tomando como base la figura 41. Nótese que algunos de los nombres han sido reducidos para mejorar la legibilidad en las siguientes tablas de resultados.

Elementos de $I_{Dinámica}$:

$I_{Dinámica}(1) \rightarrow$ *Título Dinámico*

$I_{Dinámica}(2) \rightarrow$ *Campo de texto de nombre de usuario*

$I_{Dinámica}(3) \rightarrow$ *Campo de texto ingreso para Contraseña*

$I_{Dinámica}(4) \rightarrow$ *Campo de texto de Confirmación de contraseña*

$I_{Dinámica}(5) \rightarrow$ *Botón de Enviar Formulario*

$I_{Dinámica}(6) \rightarrow$ *Área de retroalimentación del estado del usuario.*

$I_{Dinámica}(7) \rightarrow$ *Botón Regresar*

Por lo que, la longitud de $I_{Dinámica}$:

$$I_{Dinámica} |N| = 7$$

El conjunto de elementos de $I_{Dinámica}$ son los siguientes:

$$I_{Dinámica}\{N\} = \left\{ \begin{array}{l} I_{Dinámica}(1) \\ I_{Dinámica}(2) \\ I_{Dinámica}(3) \\ I_{Dinámica}(4) \\ I_{Dinámica}(5) \\ I_{Dinámica}(6) \\ I_{Dinámica}(7) \end{array} \right\}$$

Una vez identificados los elementos de la interfaz se despliegan en la tabla 18 los resultados acumulados de la población de sujetos de prueba en la interfaz dinámica. Congregar la información

en este formato de tabla facilita obtener los resultados de cada elemento de manera sencilla y rápida.

Tabla 20: Resultados de métricas de eye-tracking de Interfaz Dinámica, (Elaboración propia, 2024).

$I_{Dinámica}(n)$	$Cf(n)$	$Df(n)$	$Int(n)$	$Cs(n)$	$Vs(n)$
(1) → Título Dinámico	18	2781	0	10	613.8
(2) → Campo nombre de usuario	109	40103	12	32	2118.2
(3) → Ingreso para Contraseña	113	31985	18	19	1489.2
(4) → Confirmación de contraseña	113	37457	20	37	1989.4
(5) → Botón de Enviar Formulario	78	23923	12	15	1891.5
(6) → Estado User	123	32392	5	62	2694.1
(7) → Botón Regresar	8	2082	0	0	787.9

Ya con los resultados congregados es que se puede generar los valores cuantitativos que permiten evaluar de manera objetiva los elementos de la propuesta de diseño mediante los índices de INUSINET. Como es posible comprobar en la tabla 19, existen elementos que no tuvieron interacciones ($I_{Dinámica}(1)$), sin embargo, aplicando el concepto de “caso cero” se logró resolver las indeterminaciones resultantes. Además de este caso, existe otro caso particular en el que, de manera inesperada, el elemento no contó con ninguna interacción ($I_{Dinámica}(7)$).

Para facilitar la identificación del elemento peor evaluado en cada índice, se asigna un código de colores en el que el azul se relaciona con el ICE, el anaranjado con el ICU y finalmente el amarillo para el ICR.

Tabla 21: Resultados de la aplicación de INUSINET en la interfaz dinámica, (Elaboración propia, 2024).

$I_{Dinámica}(n)$	ICE	ICU	ICR
(1) → Título Dinámico	0.006472	Caso cero	0.016292
(2) → Campo nombre de usuario	0.002718	9.083333	0.015107
(3) → Ingreso para Contraseña	0.003533	6.277778	0.012759
(4) → Confirmación de contraseña	0.003017	5.65	0.018599
(5) → Botón de Enviar Formulario	0.003260	6.5	0.007930
(6) → Estado User	0.003797	24.6	0.023013
(7) → Botón Regresar	0.003842	Elemento no usado	Elemento sin sacádicos

Si se toman en cuenta los valores mínimos de cada uno de los resultados de cada elemento es posible identificar los elementos más complicados de entender, usar y recorrer, por lo que se entiende que son los eslabones más débiles que conforman la interfaz dinámica y, por consiguiente, son elemento que el diseñador debe corregir, rediseñar o suprimir.

En el caso particular de la interfaz dinámica los elementos peor evaluados son:

Para el ICE: $I_{Dinámica}(2)$,

Para la ICU: $I_{Dinámica}(4)$ y

Para el ICR: $I_{Dinámica}(5)$

En la figura 87 es posible visualizar los elementos peormente diseñados para cada índice con el color asociado a este.

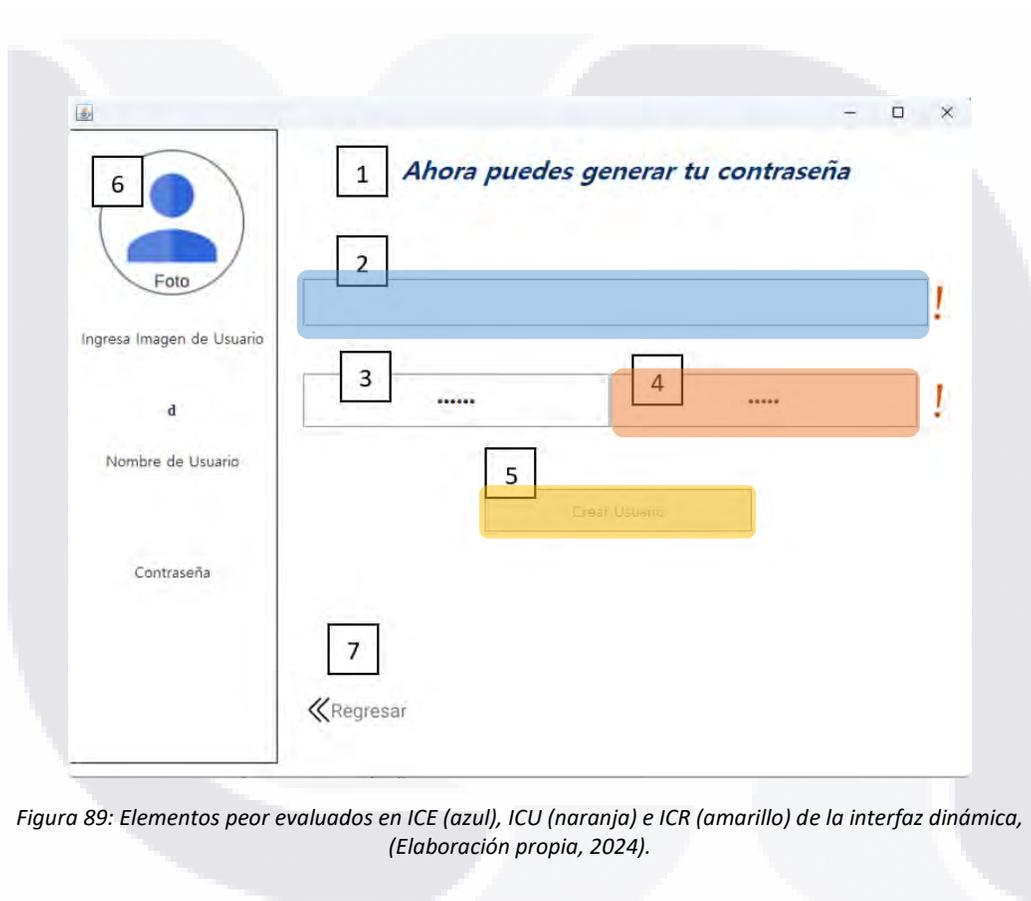


Figura 89: Elementos peor evaluados en ICE (azul), ICU (naranja) e ICR (amarillo) de la interfaz dinámica, (Elaboración propia, 2024).

B. Evaluación de interfaz de formulario convencional.

Gracias a las propiedades de la metodología de interfaces paralelas es posible, además de generar pruebas de manera mucho más eficiente, al llegar al punto de generación de resultados es mucho más sencillo conocer las diferencias y similitudes del uso de cada interfaz debido a que los elementos principales son idénticos.

Antes de proceder al análisis comparativo de cada propuesta, se expresan a continuación los resultados de la interfaz de formulario convencional mediante INUSINET.

Elementos de $I_{Convencional}$:

$I_{Convencional}(1) \rightarrow$ Título de interfaz.

$I_{Convencional}(2) \rightarrow$ Título para ingresar nombre de usuario.

$I_{Convencional}(3) \rightarrow$ Campo de ingreso de usuario.

$I_{Convencional}(4) \rightarrow$ Título para ingreso de contraseña

$I_{Convencional}(5) \rightarrow$ Campo de ingreso de contraseña

$I_{Convencional}(6) \rightarrow$ Título para confirmación de contraseña

$I_{Convencional}(7) \rightarrow$ Campo para confirmar contraseña

$I_{Convencional}(8) \rightarrow$ Botón de selección de imagen de perfil

$I_{Convencional}(9) \rightarrow$ Botón de creación de usuario

$I_{Convencional}(10) \rightarrow$ Etiqueta de mensaje de error

$I_{Convencional}(11) \rightarrow$ Botón Regresar

Por lo que, la longitud de $I_{Convencional}$:

$$I_{Convencional}|N| = 11$$

Y los elementos de $I_{Convencional}$ son los siguientes.

$$I_{Convencional}\{N\} = \left\{ \begin{array}{l} I_{Convencional}(1) \\ I_{Convencional}(2) \\ I_{Convencional}(3) \\ I_{Convencional}(4) \\ I_{Convencional}(5) \\ I_{Convencional}(6) \\ I_{Convencional}(7) \\ I_{Convencional}(8) \\ I_{Convencional}(9) \\ I_{Convencional}(10) \\ I_{Convencional}(11) \end{array} \right\}$$

En la tabla 20 se representan cada uno de los elementos de la interfaz convencional con sus respectivos resultados de las métricas después del análisis mediante eye-tracking. Los resultados en esta gráfica, al ser más elementos, complica la interpretación por parte de un evaluador que apenas se familiariza con las propuestas de INUSINET.

El primer aspecto que resalta a la vista es que se establecieron muchos más “caso cero” que en la interfaz anterior. Retomando el concepto antes mencionado (abordado más a profundidad en la sección 3.6.3 apartado C) este indica que el elemento en particular no está diseñado para que se realicen interacciones sobre él. Si se hace unión entre los casos cero y el tipo de elemento se distinguirá que todos responden a un elemento “título”, el cual está pensado para ofrecer información al usuario dentro de la interfaz.

En la variable relacionada con las interacciones de la interfaz convencional, en el elemento $I_{Convencional}(4)$ se aprecia un valor cuantitativo en un apartado con la condición de caso cero. En este tipo de caso es prudente generar un análisis más detallado la métrica que presenta una discrepancia en el data-set generado. Existen dos escenarios que pueden generar esta indeterminación: el primero, es que varios de los sujetos de prueba generaron interacciones en este elemento en particular con la expectativa de generar una acción dentro de la interfaz, la otra posibilidad puede ser explicada como una divergencia particular de un sujeto en la prueba realizada. Una vez que se analiza más minuciosamente la métrica de interacciones se descubre que un solo sujeto generó el conteo total de clicks en este elemento en particular, por lo que, de entre los casos anteriormente planteados, es el segundo caso el que determina el motivo de la discrepancia en los datos.

Parecido al elemento de título de contraseña, el elemento $I_{Convencional}(10)$ presenta una interacción en un elemento definido como caso cero, sin embargo, al ser una cantidad tan baja no es necesario rastrear a detalle el valor de la métrica en el data-set ya que puede ser un caso muy atípico de uso o incluso el error permitido del software de captura de datos mediante eye-tracking.

Tabla 22: Resultados de métricas de eye-tracking de Interfaz Convencional, (Elaboración propia, 2024).

$I_{Convencional}(n)$	$Cf(n)$	$Df(n)$	$Int(n)$	$Cs(n)$	$Vs(n)$
$I(1) \rightarrow$ Título de interfaz.	4	649	Caso cero	0	483.1
$I(2) \rightarrow$ Título nombre de usuario	37	8727	Caso cero	16	921.9
$I(3) \rightarrow$ Campo nombre de usuario	68	23690	11	22	912.6
$I(4) \rightarrow$ Título contraseña	30	10468	4 (Caso Cero)	1	824.9
$I(5) \rightarrow$ Campo contraseña	72	18145	15	8	1760.7
$I(6) \rightarrow$ Título confirmación de contraseña	25	5729	Caso cero	6	778.1
$I(7) \rightarrow$ Campo confirmar contraseña	62	24248	17	9	1288.1
$I(8) \rightarrow$ Botón imagen de perfil	86	19303	3	40	874.8
$I(9) \rightarrow$ Botón de Enviar Formulario	76	21882	23	26	1054.8
$I(10) \rightarrow$ Etiqueta de Estado Usuario	22	4939	1 (Caso Cero)	6	540.4

$I(11) \rightarrow$ Botón Regresar	3	882	0	0	795.8
------------------------------------	---	-----	---	---	-------

Replicando el procedimiento y código de colores de la interfaz dinámica, se expresan en la tabla 19 y figura 87 los elementos peor evaluados de la interfaz mediante INUSINET (código de colores: ICE-azul, ICU-naranja, ICR-amarillo).

En la interfaz de formulario convencional, los elementos peor evaluados fueron:

Para el ICE: $I_{Convencional}(7)$,

Para la ICU: $I_{Convencional}(9)$ y

Para el ICR: $I_{Convencional}(4)$.

Tabla 23: Resultados de la aplicación de INUSINET en la interfaz de formulario convencional.

$I_{Convencional}(n)$	ICE	ICU	ICR
$I(1) \rightarrow$ Título de interfaz.	0.006163	Caso Cero	Elemento sin sacádicos
$I(2) \rightarrow$ Título nombre de usuario	0.004240	Caso Cero	0.017355
$I(3) \rightarrow$ Campo nombre de usuario	0.002870	6.181818	0.024106
$I(4) \rightarrow$ Título contraseña	0.002866	Caso Cero	0.001212
$I(5) \rightarrow$ Campo contraseña	0.003968	4.8	0.004544
$I(6) \rightarrow$ Título confirmación de contraseña	0.004364	Caso Cero	0.007711
$I(7) \rightarrow$ Campo confirmar contraseña	0.002557	3.647059	0.006987
$I(8) \rightarrow$ Botón imagen de perfil	0.004455	28.666667	0.045726
$I(9) \rightarrow$ Botón de Enviar Formulario	0.003473	3.304348	0.024648
$I(10) \rightarrow$ Etiqueta de Estado Usuario	0.004454	22	0.011102
$I(11) \rightarrow$ Botón Regresar	0.003401	Elemento no usado	Elemento sin sacádicos

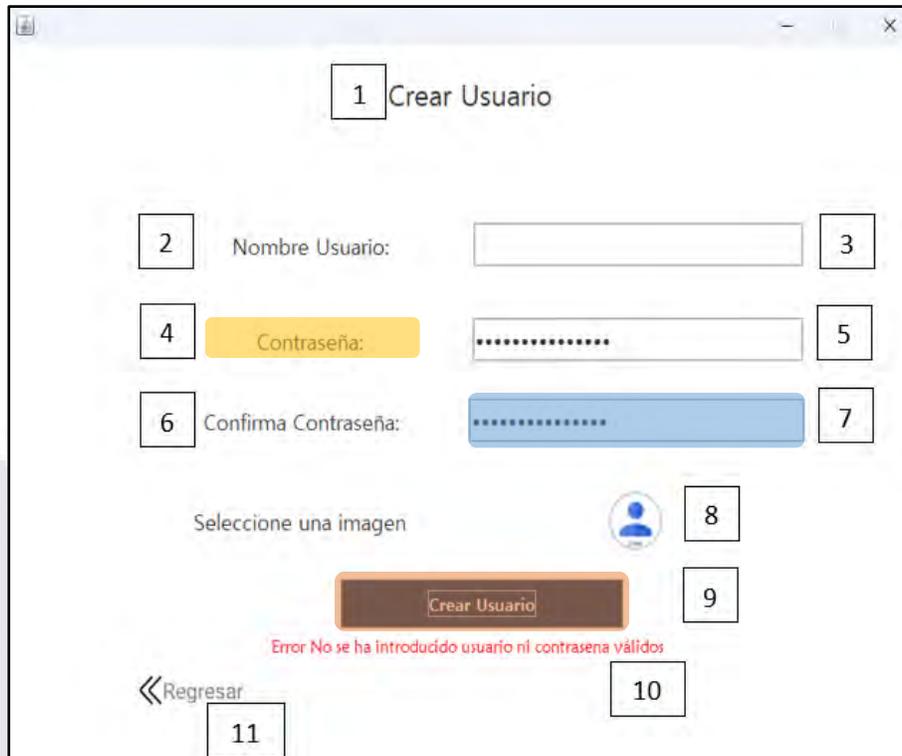
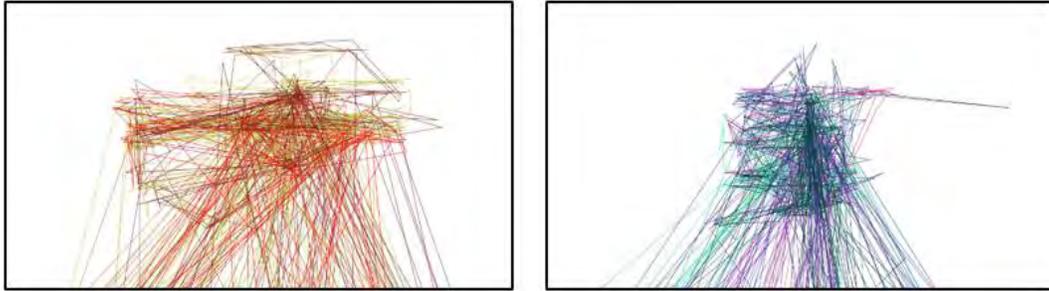


Figura 90: Elementos peor evaluados en ICE (azul), ICU (naranja) e ICR (amarillo) de la interfaz convencional, (Elaboración propia, 2024).

5.1.3 Análisis de resultados del componente login

En el análisis de ambas interfaces se hizo evidente la increíble influencia de las características de los elementos en la manera en la que los usuarios dan solución a una tarea a cumplir. Como se planteó en la sección 5.1.1, el componente login corresponde a una tarea muy específica de interacción, la cual, debe ser completada en de manera rápida y efectiva para evitar que el usuario desista de completarla, por lo que generar un patrón de uso que mejore las capacidades del usuario para completar su tarea son primordiales.

En este caso en el que el uso de las interfaces es tan distinto quedaría preguntarse: “como diseñador, ¿cómo elijo el mejor patrón de uso para que el usuario complete la tarea efectivamente, o eficientemente o ambas?”.



*Figura 91:Comparativa de las figuras 83 y 84 respectivamente,
(Elaboración propia, 2024)*

Para intentar dar solución al problema planteado en el párrafo anterior no basta con proponer cambios arbitrarios en la mayoría de los elementos de manera iterativa hasta “mágicamente” generar un diseño usable, efectiva y eficaz. Por lo contrario, promover un cambio menor implica un análisis profundo de las características de diseño del elemento que le están mermando la capacidad de ofrecer una solución dentro de la interfaz. Este análisis profundo no se limita al valor cuantitativo de las métricas que pueden observarse en las tablas 24 y 25 ya que, si bien, los índices de INUSINET están diseñados para resaltar los valores mínimos, no siempre estos elementos son los que están desempeñando una peor función en el “equipo” de elementos que conforman el diseño.

El ejemplo más claro de lo anterior es el elemento de “Botón Regresar” en el que en ambas propuestas fue un elemento que no tuvo las características de diseño que le permitieran ser una herramienta relevante en la interacción del usuario. Esto quiere decir, que independientemente de las características de diseños de los demás elementos que conforman ambas interfaces, este en específico no resulta ser entendible, usable o que facilite la lectura tomando en cuenta la falta de movimientos sacádicos, lo que motiva su corrección inmediata en las siguientes iteraciones.

Otro de los casos que se puede relacionar con elementos que, si bien no son marcados como valores mínimos, el contexto de uso puede indicar que alguno de los elementos cercanos a este es el que está otorgando información errónea en la secuencia de interacción esperada.

En las tablas 24 y 25 se remarcan en rojo los elementos que, si bien no son los peor evaluados en los índices, al comprobar las casillas colindantes y el contexto de uso que tienen en la interfaz, se identifican como puntos de mejora para la siguiente iteración.

En la tabla 24 se remarca el elemento 5 referente al botón de enviar formulario. Este elemento además de recibir la peor evaluación en el índice de ICR, también obtuvo el segundo peor valor en

el índice ICU. Ahora bien, si este elemento en particular tuviera otra función dentro de la interfaz en la que interactuar con él no sea tan relevante puede que el valor tan bajo del índice ICU no represente una alerta para el evaluador, sin embargo, entendiendo el contexto y la función tan importante que representa el elemento en la interacción esperada es preciso generar una nueva propuesta para este elemento en la siguiente versión de la interfaz.

Parecido al caso anterior, en la tabla 24 se remarca el elemento 5 debido a que, si se revisan las casillas colindantes de este elemento se encuentra que el elemento que contiene el título para el campo de la contraseña es el elemento peor evaluado para el índice ICR, tomado en cuenta que este índice indica principalmente condiciones de diseño pobres en el elemento en particular, se incita al diseñador encontrar métodos alternativos para expresar la misma información, pero con mejores mecanismos visuales. Ahora bien, volviendo al elemento 5, se puede interpretar que, además de que la información contenida en la etiqueta, también se demuestra que es el tercer peor elemento calificado mediante el índice ICU, lo que de nuevo invita a generar una revisión en las características de diseño.

Para visualizar más fácilmente los elementos remarcados como puntos débiles tomando en cuenta el contexto de uso esperado y las casillas colindantes de las tablas de resultados se generó la figura 90 en la que se presentan ambas propuestas de diseño indicando los elementos peor evaluados para los índices ICE, ICU e ICR y los puntos débiles antes mencionados.

Tabla 24: Resultados de la aplicación de INUSINET en la interfaz dinámica remarcando elementos que el evaluador detectó como puntos de mejora.

$I_{Dinámica}(n)$	ICE	ICU	ICR
(1) → Título Dinámico	0.006472	Caso cero	0.016292
(2) → Campo nombre de usuario	0.002718	9.083333	0.015107
(3) → Ingreso para Contraseña	0.003533	6.277778	0.012759
(4) → Confirmación de contraseña	0.003017	5.65	0.018599
(5) → Botón de Enviar Formulario	0.003260	6.5	0.007930
(6) → Estado User	0.003797	24.6	0.023013
(7) → Botón Regresar	0.003842	Elemento no usado	Elemento sin sacádicos

Tabla 25: Resultados de la aplicación de INUSINET en la interfaz convencional remarcando elementos que el evaluador detectó como puntos de mejora.

$I_{Convencional}(n)$	ICE	ICU	ICR
$I(1) \rightarrow$ Título de interfaz.	0.006163	Caso Cero	Elemento sin sacádicos
$I(2) \rightarrow$ Título nombre de usuario	0.004240	Caso Cero	0.017355
$I(3) \rightarrow$ Campo nombre de usuario	0.002870	6.181818	0.024106
$I(4) \rightarrow$ Título contraseña	0.002866	Caso Cero	0.001212
$I(5) \rightarrow$ Campo contraseña	0.003968	4.8	0.004544
$I(6) \rightarrow$ Título confirmación de contraseña	0.004364	Caso Cero	0.007711
$I(7) \rightarrow$ Campo confirmar contraseña	0.002557	3.647059	0.006987
$I(8) \rightarrow$ Botón imagen de perfil	0.004455	28.666667	0.045726
$I(9) \rightarrow$ Botón de Enviar Formulario	0.003473	3.304348	0.024648
$I(10) \rightarrow$ Etiqueta de Estado Usuario	0.004454	22	0.011102
$I(11) \rightarrow$ Botón Regresar	0.003401	Elemento no usado	Elemento sin sacádicos

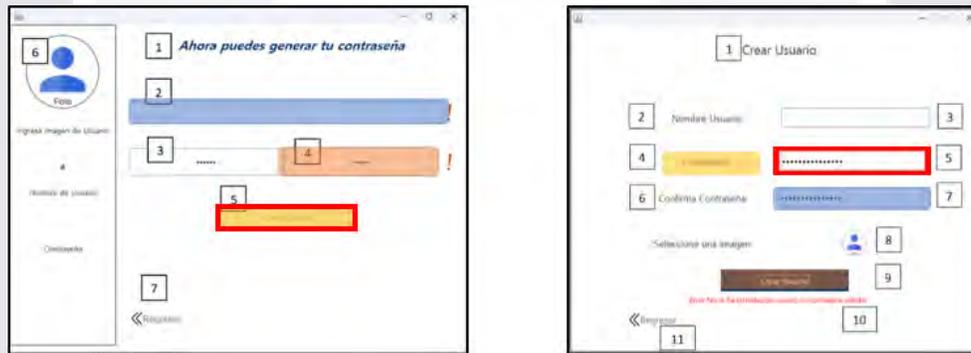


Figura 92: Comparativa de las figuras 63 y 64 con los elementos peor evaluados y, remarcado en rojo, elementos que son detectados por el evaluador como puntos débiles de uso, (Elaboración propia, 2024).

Es hasta este punto que la primera iteración del componente login llega a su fin ya que se obtuvo una buena información inicial sobre el uso de las interfaces. En secciones posteriores de discusión de resultados se abordará de manera general tanto las pruebas, como la representación de resultados y el análisis de estos mejoraron las capacidades del diseñador para conocer a detalle las características de uso y al evaluador si las herramientas de análisis mediante eye-tracking le permitieron evaluar objetivamente el uso de las interfaces.

5.2 Resultado de Caso de Estudio 2: Easy-Bank Vision

Las pruebas de uso para conocer las capacidades de interacción de Easy-Bank Vision constó 26 experimentos, de los cuáles varios tuvieron que ser descartados por motivos de la calibración o errores en la secuencia de interacciones por parte del usuario.

Para poder brindar más información sobre las pruebas de uso se menciona que había tres sub-roles dentro del equipo de evaluación, los cuáles constaban de un supervisor técnico del back-end de Easy-Bank Vision procurando que los datos en la base de datos fluyeran correctamente, así como el estado de ejecución del programa en el servidor dedicado. También estaba la figura del anfitrión, el cuál recibía al participante y lo orientaba a lo largo de toda la experimentación. Finalmente, pero no menos importante, el supervisor de eye-tracking mantenía las herramientas de TobiiProLab y TobiiProNano calibrados y preparados para cada prueba.

Es de destacar que, siguiendo este proceso, la implementación de la tecnología logró ser lo menos intrusiva posible para el sujeto de pruebas, permitiéndole abstraerse en la actividad e ignorar el entorno de experimentación de eye-tracking.

Después del filtrado de la calidad de los resultados de las pruebas se analizaron un total de 22 pruebas, conformadas por sujetos de prueba todos los tipos de perfiles establecidos en el diseño de la experiencia de usuario.

En las siguientes secciones referentes a los patrones de uso generales y particulares se aborda detalladamente el proceso de evaluación de cada una de las interfaces. Posteriormente, se analizan e interpretan los resultados para generar un resultado final de la evaluación de la experiencia completa de Easy-Bank Vision.

5.2.1 Resultados de Patrones de Uso Generales

Uno de los aspectos más notables para las pruebas del caso de estudio Easy-Bank Vision es la cantidad de pruebas que se lograron realizar a distintos tipos de usuarios. Ahora bien, esta misma cantidad de usuarios genera mapas de recorrido de abeja y de fijaciones muy poco claros para determinar ciertos patrones. Tomando en cuenta que la herramienta de Visualizaciones de *TobiiProLab* genera diferentes colores para cada recorrido, resulta complejo distinguir algunas rutas

en específico si los colores de las rutas de dos sujetos son las mismas, sesgando incluso las posibles interpretaciones que pueden generarse.

Para resolver esta problemática se optó por generar las visualizaciones con un tamaño de muestra en la que cada tipo de usuario se representara un par de veces, es decir, ocho muestras. Los sujetos fueron seleccionados de forma aleatoria y, debido a que se pretende evaluar si las características de uso del sistema tienen un uso promedio en los diferentes tipos de usuario, no se mencionarán las rutas específicas de cada tipo, en su lugar, se analizará la convergencia de los patrones.

La convergencia de los patrones de uso de una interfaz hace referencia al concepto del diseño centrado en el usuario en el que se busca que la mayor cantidad de usuarios pueda completar una tarea independientemente de sus características particulares. Es en este punto que tanto el diseñador como el evaluador deben comparar las expectativas del primero con los resultados del segundo.

A continuación, se expone los resultados de cada interfaz de la experiencia mencionando las principales características que se encontraron al analizar los patrones de uso de la muestra de sujetos de prueba.

A. Login Detallado

Como se observa en la figura 93 la interfaz de Login Detallado cuenta con una distribución de los patrones en forma de un trapecio, con movimientos sacádicos largas entre las agrupaciones de fijaciones. Una de las características importantes que se debe destacar principalmente en los tipos de recorridos es que estos no se interceptan en el centro de la pantalla.

Retomando la teoría de atención visual y el modelo INUSINET, un movimiento sacádico extendido se puede interpretar como una búsqueda poco rigurosa que intenta localizar elementos interesantes dentro de la pantalla, a su vez, las fijaciones recurrentes en un mismo punto se pueden reconocer como elementos que aportan buena información al usuario.

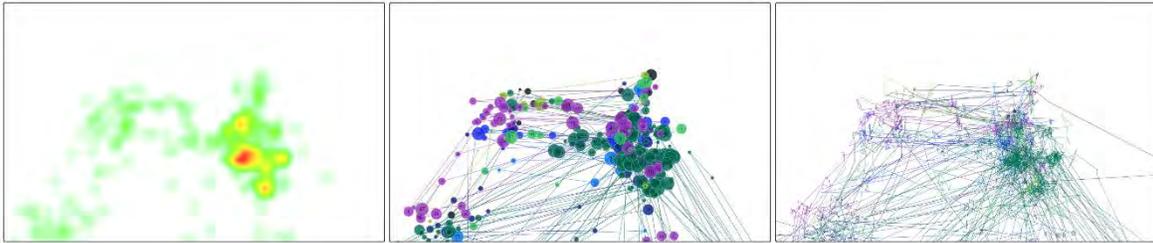


Figura 93: Visualizaciones de los patrones de uso generales de la interfaz Login Detallado (de izquierda a derecha mapa de calor, mapa de fijaciones y recorrido de abeja), (Elaboración propia, 2024).

B. Login Simple

A diferencia del Login Detallado, la interfaz de Login Simple presenta resultados de fijaciones mucho más congregados justo en el centro de la pantalla y a partir de ahí se despliega una gran cantidad de sacádicos hacia el exterior (figura 94).

Si se observa detalladamente la visualización del mapa de recorrido de fijaciones hay varias lecturas en la parte superior izquierda de la pantalla las cuales son muy pequeñas y dispersas. Este comportamiento se puede considerar una anomalía en el uso del sistema y puede relacionarse con la dimensión de los elementos, ya que al congregarlos en un espacio tan reducido el sujeto puede sentirse abrumado y decida realizar una nueva exploración en la pantalla buscando algún elemento que pueda reconocer más fácilmente.

Esta interfaz presentó a vista del evaluador algunas áreas de oportunidad muy relevantes, principalmente la acumulación de fijaciones en un área muy pequeña de la pantalla. A su vez, los movimientos sacádicos que se dirigen en muchas direcciones pueden indicar que identificar a los elementos más relevantes en distintas etapas de la interacción es un reto para los sujetos de la muestra.

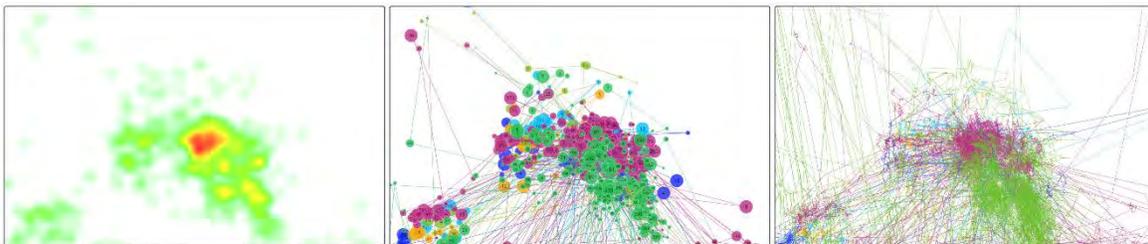


Figura 94: Visualizaciones de los patrones de uso generales de la interfaz Login Simple (de izquierda a derecha mapa de calor, mapa de fijaciones y recorrido de abeja), (Elaboración propia, 2024).

C. Home

Desde la prueba de un único sujeto aleatorio la interfaz Home parecía tener buenos resultados con respecto a sus patrones de uso. Es interesante observar que si bien hay algunos movimientos sacádicos que se cruzan perpendicularmente, la mayoría de estos van de izquierda a derecha y, por las congregaciones de fijaciones, se puede intuir que la información que se obtiene de estas áreas es relevante y permiten al usuario reconocer fácilmente los elementos y comprender más detalladamente las interacciones que puede realizar (resultados contenidos en la figura 95).

Ampliando las interacciones que puede realizar, es preciso mencionar que la parte más visitada por el usuario fue la parte media baja izquierda de la pantalla, lo interesante de este caso fue que los botones parecen ser poco visitados. Esta característica apoya la idea anterior de que los elementos son claros para el usuario y no necesita visitar estas áreas ya que ha reconocido que son efectivos para su interacción, por lo que solo vuelve a ellos cuándo tiene clara la siguiente acción a realizar, sin detenerse a profundidad en el contenido de estos.

El diseño de esta interfaz presentó buenos índices de convergencia entre la muestra de sujetos de prueba, y como se menciona en las ideas anteriores, cuenta con una buena relación entre las expectativas de diseño y los resultados de evaluación.

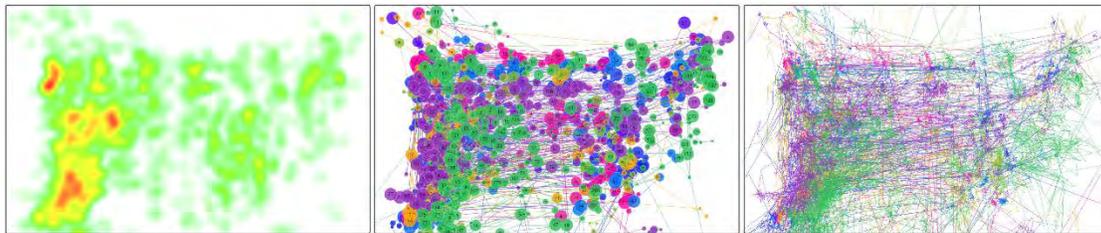


Figura 95: Visualizaciones de los patrones de uso generales de la interfaz Home (de izquierda a derecha mapa de calor, mapa de fijaciones y recorrido de abeja), (Elaboración propia, 2024).

D. Saldos y Movimientos

Los resultados de esta interfaz presentaron varios puntos a destacar para el diseñador y evaluador, el primero de ellos es que si bien en el mapa de calor parece existir una estructura definida que se corresponde a los elementos principales de la interfaz, los movimientos sacádicos presentan una

gran cantidad de cruces perpendiculares (consúltase figura 96). También las fijaciones parecen posarse en muchas áreas diferentes, incluso por toda la pantalla, además, estas son de un diámetro pequeño lo que se puede relacionar con elementos que en general brindan poca información relevante para la interacción

Si se observa el diagrama de recorrido de abeja se hace más evidente que: existe muy poca convergencia entre los patrones de uso de cada tipo de usuario, y al parecer el diseño de interacción planeado no parece corresponder a los resultados obtenidos.

Otro aspecto que es posible observar con el patrón de abeja es que se recorre la totalidad de la pantalla en algunos momentos de la interacción. Si comparamos los resultados de la interfaz Login Simple esta concentraba mucha información en un punto central, mientras aquí es precisamente la parte central la que es poco visitada. Esto puede empeorar el proceso de interacción del usuario ya que tiene que interactuar con elementos que no son parecidos en cuanto a la dimensión, disposición y finalidad de estos en la interacción.

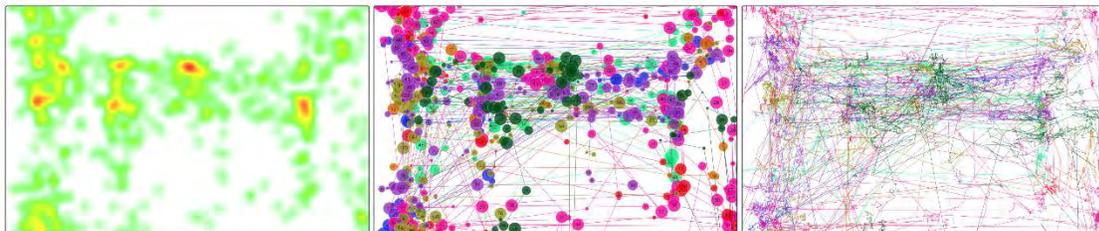


Figura 96: Visualizaciones de los patrones de uso generales de la interfaz Saldos y Movimientos (de izquierda a derecha mapa de calor, mapa de fijaciones y recorrido de abeja), (Elaboración propia, 2024).

E. Pago Tarjeta

Este caso de uso sobre una ventana emergente fue un caso de estudio relativamente experimental para conocer las capacidades de las técnicas de eye-tracking en eventos que duren periodos cortos. Para las expectativas de diseño se plateó que si el elemento tenía un buen diseño las fijaciones permanecerían dentro de este elemento durante la tarea.

Uno de los puntos positivos de este evento a criterio del evaluador es que el mapa de calor coincide directamente con la relevancia esperada de los elementos (figura 97), es decir, el punto con mayor

cantidad de fijaciones contiene el cuadro de texto en el que usuario tiene que ingresar un valor número que obtiene de los elementos aledaños, entonces, una vez mencionado lo anterior, y si se observa muy detalladamente existen muchos movimientos sacádicos entre dos grandes agrupaciones de fijaciones, una recargada a la izquierda y otra en la derecha, y debido a que los movimientos sacádicos son cortos, y muy paralelos entre los sujetos de prueba, se relaciona este comportamiento con un diseño de elementos que es efectivo en brindar información.



Figura 97: Visualizaciones de los patrones de uso generales de la interfaz Pago Tarjeta (de izquierda a derecha mapa de calor, mapa de fijaciones y recorrido de abeja), (Elaboración propia, 2024).

F. Transferencias

A pesar de que la interfaz de transferencia estaba planteada como una tarea poco complicada para el usuario es relevante observar lo complejo que resultaron los patrones de uso de la muestra de sujetos de prueba. Incluso el mapa de calor representa fijaciones en la mayor parte de la pantalla (véase figura 98).

Existen dos casos dentro de las visualizaciones que, como en las demás evaluaciones de interfaces, dependen directamente de las expectativas de uso. Si bien el área más relevante de la interfaz es el formulario central para desarrollar la tarea, existen muchos elementos distractores que hacen que la mirada se aleje del punto deseado por el diseñador, también existe poca convergencia en los patrones debido a que, además de que no es posible distinguir algún patrón reconocible, existen muchos cruces perpendiculares de movimientos sacádicos.

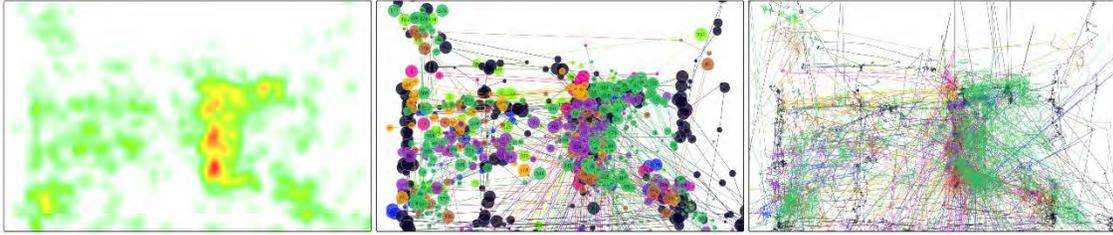


Figura 98: Visualizaciones de los patrones de uso generales de la interfaz Transferencias (de izquierda a derecha mapa de calor, mapa de fijaciones y recorrido de abeja), (Elaboración propia, 2024).

Una vez obtenidos los resultados de cada interfaz de estudio y se han generado varias interpretaciones sobre las características de uso de cada una de ellas, se procedió a generar un análisis más específico de cada elemento de la interfaz.

5.2.2 Resultados de patrones de uso particulares

Como se ha mencionado, los patrones de uso generales presentan información relevante del comportamiento de los movimientos oculares y de los comportamientos de la mirada al usar una interfaz, sin embargo, para conocer a detalle la capacidad de cada elemento para ser entendido, usado y recorrido se aplica una evaluación utilizando los conceptos y procedimientos propuestos por INUSINET.

Es relevante mencionar que, una vez más, las interpretaciones de los resultados dependen de las características deseadas de uso de cada elemento. Dado que el caso de estudio de Easy-Bank Vision representa una experiencia más compleja que el otro caso de estudio contenido en esta investigación, es preciso relacionar la interacción a través de las distintas interfaces que la conforman.

Mencionado lo anterior, y mediante las propuestas de INUSINET se presentan las evaluaciones de cada interfaz utilizando los data-sets generados para este fin. En este caso, los resultados son poblacionales debido a que se busca estimar si una gran cantidad de pruebas de uso mejora la precisión de las evaluaciones de eye-tracking.

A. Login Detallado

El análisis del login detallado consta de los elementos necesarios para que un usuario ingrese los valores específicos de su número de cuenta y contraseña, los cuáles eran asignados de manera aleatoria al inicio de la prueba, también es preciso recordar que para facilitarle esta información al usuario se le entregaba una tarjeta plástica que simulaba una tarjeta de débito o crédito.

A continuación, se expresan formalmente los elementos y las nomenclaturas asociadas a cada elemento de la interfaz de Login Detallado. Para facilitar la lectura, a esta interfaz se le asignó el identificador “LDetallado”.

Elementos de $I_{LDetallado}$:

$I_{LDetallado}(1) \rightarrow Logos$

$I_{LDetallado}(2) \rightarrow Títulos$

$I_{LDetallado}(3) \rightarrow Etiqueta Numero de Cliente$

$I_{LDetallado}(4) \rightarrow Campo Ingreso Numero de Cliente$

$I_{LDetallado}(5) \rightarrow Etiqueta Contraseña$

$I_{LDetallado}(6) \rightarrow Campo Ingreso Contraseña$

$I_{LDetallado}(7) \rightarrow Botón Enviar Formulario$

$I_{LDetallado}(8) \rightarrow Switch Modo Detallado$

Por lo que, la longitud de $I_{LDetallado}$:

$$I_{LDetallado}|N| = 8$$

Y los elementos de $I_{LDetallado}$ son los siguientes.

$$I_{LDetallado}\{N\} = \left\{ \begin{array}{l} I_{LDetallado}(1) \\ I_{LDetallado}(2) \\ I_{LDetallado}(3) \\ I_{LDetallado}(4) \\ I_{LDetallado}(5) \\ I_{LDetallado}(6) \\ I_{LDetallado}(7) \\ I_{LDetallado}(8) \end{array} \right\}$$

En la tabla 26 se representan los resultados de exponer la interfaz de login detallado a la población de sujetos de prueba. A primera vista, la cantidad de sujetos de prueba parece tener una relación parecida a la observada en los resultados del caso de estudio del componente login, sin embargo, es necesario realizar los cálculos pertinentes de los índices de entendimiento, uso y recorrido propuestos en INUSINET para conocer más a detalle el comportamiento de estos.

Como se observa en la tabla 27 los valores de los índices presentan valores dentro del rango del otro caso de estudio. Remarcando los elementos peor evaluados en la interfaz se tiene que los elementos: Botón de enviar formulario, campo de ingreso y switch detallado. En la sección de análisis se abordará más detenidamente la interpretación de estos resultados.

Tabla 26: Resultados de Interfaz Login Detallado de Easy-Bank Vision, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{L\text{Detallado}}(n)$	$Cf(n)$	$Df(n)$	$Int(n)$	$Cs(n)$	$Vs(n)$
$I(1) \rightarrow$ Logos	145	29596	Caso Cero	65	5357.02
$I(2) \rightarrow$ Títulos	34	6713	Caso Cero	7	2572.03
$I(3) \rightarrow$ Etiqueta Numero de Cliente	102	25829	Caso Cero	25	2942.96
$I(4) \rightarrow$ Campo Ingreso Numero de Cliente	49	13989	4	7	1325.335
$I(5) \rightarrow$ Etiqueta Contraseña	78	19728	Caso Cero	14	1097.67
$I(6) \rightarrow$ Campo Ingreso Contraseña	45	12208	12	3	1145.675
$I(7) \rightarrow$ Botón Enviar Formulario	39	10472	6	10	1582.405
$I(8) \rightarrow$ Switch Modo Detallado	33	6287	4	6	2959.8

Tabla 27: Resultados de INUSINET para la interfaz de Login Detallado, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{L\text{Detallado}}(n)$	ICE	ICU	ICR
$I(1) \rightarrow$ Logos	0.004899	Caso Cero	0.012134
$I(2) \rightarrow$ Títulos	0.005065	Caso Cero	0.002722
$I(3) \rightarrow$ Etiqueta Numero de Cliente	0.003949	Caso Cero	0.008495
$I(4) \rightarrow$ Campo Ingreso Numero de Cliente	0.003503	36.25	0.005282
$I(5) \rightarrow$ Etiqueta Contraseña	0.003954	Caso Cero	0.012754
$I(6) \rightarrow$ Campo Ingreso Contraseña	0.003686	12.08333	0.002619
$I(7) \rightarrow$ Botón Enviar Formulario	0.003724	24.16667	0.006319
$I(8) \rightarrow$ Switch Modo Detallado	0.005249	36.25	0.002027

B. Login Simple

Por las características del proceso experimental, esta interfaz fue más usada que la del modo detallado. También es de destacar que los elementos entre ambas interfaces de login tienen exactamente los mismos elementos, por lo que los resultados que se obtuvieron permiten comparar los efectos de la distribución y dimensión de los elementos de manera más precisa.

La formalización de esta interfaz para su análisis con INUSINET se representa a continuación y posteriormente se expresan en las tablas 28 y 29 el desglose de las métricas y resultados de índices obtenidos respectivamente.

Con el fin de mejorar la legibilidad de las variables se usará “LSimple” para denotar la interfaz de Login Simple.

Elementos de $I_{LSimple}$:

$I_{LSimple}(1) \rightarrow Logos$

$I_{LSimple}(2) \rightarrow Títulos$

$I_{LSimple}(3) \rightarrow Etiqueta\ Numero\ de\ Cliente$

$I_{LSimple}(4) \rightarrow Campo\ Ingreso\ Numero\ de\ Cliente$

$I_{LSimple}(5) \rightarrow Etiqueta\ Contraseña$

$I_{LSimple}(6) \rightarrow Campo\ Ingreso\ Contraseña$

$I_{LSimple}(7) \rightarrow Botón\ Enviar\ Formulario$

$I_{LSimple}(8) \rightarrow Switch\ Modo\ Detallado$

Por lo que, la longitud de $I_{LSimple}$:

$$I_{LSimple}|N| = 8$$

Y los elementos de $I_{LSimple}$ son los siguientes.

$$I_{LSimple}\{N\} = \left\{ \begin{array}{l} I_{LSimple}(1) \\ I_{LSimple}(2) \\ I_{LSimple}(3) \\ I_{LSimple}(4) \\ I_{LSimple}(5) \\ I_{LSimple}(6) \\ I_{LSimple}(7) \\ I_{LSimple}(8) \end{array} \right\}$$

Tabla 28: Resultados de Interfaz Login Simple de Easy-Bank Vision, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{LSimple}(n)$	$Cf(n)$	$Df(n)$	$Int(n)$	$Cs(n)$	$Vs(n)$
$I(1) \rightarrow Logos$	8739	46	Caso Cero	9	3499.97
$I(2) \rightarrow Títulos$	18475	95	Caso Cero	21	4021.655
$I(3) \rightarrow Etiqueta Numero de Cliente$	45657	184	Caso Cero	56	5492.86
$I(4) \rightarrow Campo Ingreso Numero de Cliente$	121026	402	20	110	5876.54
$I(5) \rightarrow Etiqueta Contraseña$	15919	85	Caso Cero	15	4292.405
$I(6) \rightarrow Campo Ingreso Contraseña$	55694	209	28	24	3848.55
$I(7) \rightarrow Botón Enviar Formulario$	31017	118	27	18	2503.47
$I(8) \rightarrow Switch Modo Detallado$	32994	79	14	22	5271.045

Tabla 29: Resultados de INUSINET para la interfaz de Login Simple, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{LSimple}(n)$	ICE	ICU	ICR
$I(1) \rightarrow Logos$	0.005264	Caso Cero	0.002571
$I(2) \rightarrow Títulos$	0.005142	Caso Cero	0.005222
$I(3) \rightarrow Etiqueta Numero de Cliente$	0.004030	Caso Cero	0.010195
$I(4) \rightarrow Campo Ingreso Numero de Cliente$	0.003322	20.1	0.018718
$I(5) \rightarrow Etiqueta Contraseña$	0.005340	Caso Cero	0.003495
$I(6) \rightarrow Campo Ingreso Contraseña$	0.003753	7.464285714	0.006236
$I(7) \rightarrow Botón Enviar Formulario$	0.003804	4.37037037	0.007190
$I(8) \rightarrow Switch Modo Detallado$	0.002394	5.642857143	0.004174

C. Home

La pestaña Home es un caso de uso muy particular en este caso de estudio ya que es la interfaz más usada a lo largo del proceso experimental. Estas características la convierten en un punto neurálgico de toda la experiencia, por lo que, garantizar que esta provea una buena capacidad para que los usuarios puedan completar las distintas tareas planeadas.

En la sección de patrones de uso generales se abordó la posibilidad que los patrones de uso observados podían reflejar un diseño que permitiera cubrir las necesidades de uso de la interacción esperada. Para poder refutar o comprobar esta posibilidad es necesario generar el proceso de evaluación de interfaces mediante el modelo INUSINET. Para lo anterior, se expresan las variables que permiten calcular los resultados de las tablas 30 y 31 que contienen los datos biométricos registrados para esta interfaz de la población de sujetos de prueba.

Elementos de I_{Home} :

$I_{Home}(1) \rightarrow Logo$

$I_{Home}(2) \rightarrow Salir$

$I_{Home}(3) \rightarrow Botón Saldos y Movimientos$

$I_{Home}(4) \rightarrow Botón estado de cuenta$

$I_{Home}(5) \rightarrow Botón Inversiones$

$I_{Home}(6) \rightarrow Botón Transferencias$

$I_{Home}(7) \rightarrow Botón Pago Servicios$

$I_{Home}(8) \rightarrow Ayuda$

$I_{Home}(9) \rightarrow Tarjetas$

$I_{Home}(10) \rightarrow Switch Modo Detallado$

Por lo que, la longitud de I_{Home} :

$$I_{Home}|N| = 10$$

Y los elementos de I_{Home} son los siguientes.

$$I_{Home}\{N\} = \left\{ \begin{array}{l} I_{Home}(1) \\ I_{Home}(2) \\ I_{Home}(3) \\ I_{Home}(4) \\ I_{Home}(5) \\ I_{Home}(6) \\ I_{Home}(7) \\ I_{Home}(8) \\ I_{Home}(9) \\ I_{Home}(10) \end{array} \right\}$$

Tabla 30: Resultados de Interfaz Home de Easy-Bank Vision,, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{Home}(n)$	$Cf(n)$	$Df(n)$	$Int(n)$	$Cs(n)$	$Vs(n)$
$I(1) \rightarrow Logo$	17	2474	2	1	1434.085
$I(2) \rightarrow Salir$	7	1882	0	1	1734.6
$I(3) \rightarrow Botón Saldo y Movimientos$	205	80010	26	46	11699.28
$I(4) \rightarrow Botón estado de cuenta$	120	28436	1	21	7819.035
$I(5) \rightarrow Botón Inversiones$	99	20402	0	17	8598.34
$I(6) \rightarrow Botón Transferencias$	127	42696	15	27	8950.54
$I(7) \rightarrow Botón Pago Servicios$	60	16091	1	8	5184.875
$I(8) \rightarrow Ayuda$	170	50018	Caso Cero	78	5509.035
$I(9) \rightarrow Tarjetas$	1560	399834	129	902	13051.72
$I(10) \rightarrow Switch Modo Detallado$	116	40232	20	33	4849.98

Tabla 31: Resultados de INUSINET para la interfaz Home, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{Home}(n)$	ICE	ICU	ICR
$I(1) \rightarrow Logo$	0.006871	8.5	0.000697
$I(2) \rightarrow Salir$	0.003719	Elemento no Usado	0.000577
$I(3) \rightarrow Botón Saldo y Movimientos$	0.002562	7.884615	0.003932
$I(4) \rightarrow Botón estado de cuenta$	0.00422	120	0.002686
$I(5) \rightarrow Botón Inversiones$	0.004852	Elemento no Usado	0.001977
$I(6) \rightarrow Botón Transferencias$	0.002975	8.466667	0.003017
$I(7) \rightarrow Botón Pago Servicios$	0.003729	60	0.001543
$I(8) \rightarrow Ayuda$	0.003399	Caso Cero	0.014159
$I(9) \rightarrow Tarjetas$	0.003902	12.09302	0.06911
$I(10) \rightarrow Switch Modo Detallado$	0.002883	5.8	0.006804

D. Saldos y Movimientos

Durante las pruebas piloto del proceso experimental de la evaluación de Easy-Bank Vision se encontró que los elementos superiores de la pantalla, como lo son el botón salir y el ícono del sistema, no eran usados a lo largo de la experiencia planteada hasta ese momento. Por lo tanto parte de las indicaciones para este evento en particular se añadió la directiva de salir a la interfaz Home, desde la interfaz de Saldos y Movimientos.

La instrucción motivaba al usuario a recorrer la interfaz de manera libre hasta que este, mediante su regreso a la interfaz home, indicaba al investigador que ya había completado su exploración en la pantalla.

Uno de los efectos que el equipo de evaluación notó sobre esta interfaz es que cuando a un usuario se le permite explorar libremente la pantalla, este tiende a realizar más clicks, de tal manera que se puede suponer que al no sentirse con la presión de resolver un problema o completar una tarea genera interacciones sin detenerse a pensar en una secuencia lógica o eficiente.

En las visualizaciones de los patrones de uso generales de la interfaz Saldos y Movimientos se generó una posible interpretación en la que el notorio desorden de las fijaciones y movimientos sacádicos indicaban un poco o casi nula convergencia, sin embargo, al analizar detalladamente el proceso de interacción y las instrucciones dadas, es más comprensible encontrar patrones tan diversos, ya que como se menciona en la idea anterior, el sujeto de prueba no reacciona o planea sus acciones de una forma rigurosa al no tener que completar una tarea, por lo que se da libertad de relajar sus procesos de búsqueda.

Para seguir desarrollando las posibles interpretaciones que se generaron al analizar esta interfaz se procede a generar las variables necesarias para la evaluación particular del uso de cada elemento. Para mejorar la exposición de los resultados se asignó a las variables el identificador "Sal-Mov" para hacer referencia a la interfaz de Saldos y Movimientos. En las tablas 32 y 33 se observan los resultados obtenidos para las variables mencionadas.

Elementos de $I_{Sal-Mov}$:

$I_{Sal-Mov}(1) \rightarrow Logo$

$I_{Sal-Mov}(2) \rightarrow Salir$

$I_{Sal-Mov}(3) \rightarrow Botón Todos los Registros$

$I_{Sal-Mov}(4) \rightarrow Botón Ingresos$

$I_{Sal-Mov}(5) \rightarrow$ Botón Egresos

$I_{Sal-Mov}(6) \rightarrow$ Ayuda

$I_{Sal-Mov}(7) \rightarrow$ Tabla de Registros

$I_{Sal-Mov}(8) \rightarrow$ Switch Modo Detallado

Por lo que, la longitud de $I_{Sal-Mov}$:

$$I_{Sal-Mov}|N| = 8$$

Y los elementos de $I_{Sal-mov}$ son los siguientes.

$$I_{Sal-Mov}\{N\} = \left\{ \begin{array}{l} I_{Sal-Mov}(1) \\ I_{Sal-Mov}(2) \\ I_{Sal-Mov}(3) \\ I_{Sal-Mov}(4) \\ I_{Sal-Mov}(5) \\ I_{Sal-Mov}(6) \\ I_{Sal-Mov}(7) \\ I_{Sal-Mov}(8) \end{array} \right\}$$

Tabla 32: Resultados de Interfaz Saldos y Movimientos de Easy-Bank Vision, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{Sal-Mov}(n)$	$Cf(n)$	$Df(n)$	$Int(n)$	$Cs(n)$	$Vs(n)$
$I(1) \rightarrow$ Logo	124	56622	25	44	4315.935
$I(2) \rightarrow$ Salir	31	9661	0	2	2198.005
$I(3) \rightarrow$ Botón Todos los Registros	79	18163	3	9	4700.42
$I(4) \rightarrow$ Botón Ingresos	77	18847	9	11	4392.84
$I(5) \rightarrow$ Botón Egresos	92	32439	27	9	4211.645
$I(6) \rightarrow$ Ayuda	81	18070	0	46	3492.845
$I(7) \rightarrow$ Tabla de Registros	657	167985	14	290	7200.955
$I(8) \rightarrow$ Switch Modo Detallado	35	13822	5	7	3439.08

Tabla 33: Resultados de INUSINET para la interfaz de Saltos y Movimientos,, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{Sal-Mov}(n)$	ICE	ICU	ICR
$I(1) \rightarrow$ Logo	0.00219	4.96	0.010195
$I(2) \rightarrow$ Salir	0.003209	Elemento no Usado	0.00091
$I(3) \rightarrow$ Botón Todos los Registros	0.00435	26.33333	0.001915

$I(4) \rightarrow \text{Botón Ingresos}$	0.004086	8.555556	0.002504
$I(5) \rightarrow \text{Botón Egresos}$	0.002836	3.407407	0.002137
$I(6) \rightarrow \text{Ayuda}$	0.004483	Caso Cero	0.01317
$I(7) \rightarrow \text{Tabla de Registros}$	0.003911	46.92857	0.040272
$I(8) \rightarrow \text{Switch Modo Detallado}$	0.002532	7	0.002035

E. Pagar Tarjeta

Como se menciona en el análisis de los patrones de uso generales, el caso de estudio de pagar tarjeta se despliega en una ventana emergente desde la pestaña de Home. Las características principales de esta interfaz son su tamaño reducido y que es posible completarlo en un breve periodo de tiempo.

Analizando muy detalladamente las visualizaciones referentes a esta interfaz se encontraron movimientos sacádicos que invitaban a pensar que los patrones de atención visual correspondían a una experiencia relativamente cómoda y que los elementos proveían información al usuario, sin embargo, realizar esta evaluación únicamente con las visualizaciones es una tarea casi imposible si se desea tener una precisión que influya positivamente en el proceso de diseño y evaluación.

Para lo anterior, se genera el análisis correspondiente de las variables de este diseño y en las tablas 34 y 35 se depositan los datos biométricos de eye-tracking de la población de sujetos de prueba. En esta ocasión, la variable referente a Pagar Tarjeta se menciona establece como “PagoTarjeta”.

Elementos de $I_{\text{PagoTarjeta}}$:

$I_{\text{PagoTarjeta}}(1) \rightarrow \text{Ventana Emergente}$

$I_{\text{PagoTarjeta}}(2) \rightarrow \text{Datos de Cuenta Usuario}$

$I_{\text{PagoTarjeta}}(3) \rightarrow \text{Campo Ingreso Cantidad de Pago}$

$I_{\text{PagoTarjeta}}(4) \rightarrow \text{Botón Enviar Formulario}$

Por lo que, la longitud de $I_{\text{PagoTarjeta}}$:

$$|I_{\text{PagoTarjeta}}| = 4$$

Y los elementos de $I_{\text{PagoTarjeta}}$ son los siguientes.

$$I_{PagoTarjeta}\{N\} = \begin{cases} I_{PagoTarjeta}(1) \\ I_{PagoTarjeta}(2) \\ I_{PagoTarjeta}(3) \\ I_{PagoTarjeta}(4) \end{cases}$$

Tabla 34: Resultados de Interfaz Pagar Tarjeta de Easy-Bank Vision, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{PagoTarjeta}(n)$	$Cf(n)$	$Df(n)$	$Int(n)$	$Cs(n)$	$Vs(n)$
$I(1) \rightarrow$ Ventana Emergente	604	158895	22	413	4290.83
$I(2) \rightarrow$ Datos de Cuenta Usuario	349	91982	Caso Cero	195	2618.2
$I(3) \rightarrow$ Campo Ingreso Cantidad de Pago	165	45067	7	62	2486.58
$I(4) \rightarrow$ Botón Enviar Formulario	24	7653	15	1	1291.47

Tabla 35: Resultados de INUSINET para la interfaz Pagar Tarjeta,, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{PagoTarjeta}(n)$	ICE	ICU	ICR
$I(1) \rightarrow$ Logo	0.003801	27.45455	0.096252
$I(2) \rightarrow$ Salir	0.003794	Caso Cero	0.074479
$I(3) \rightarrow$ Botón Todos los Registros	0.003661	23.57143	0.024934
$I(4) \rightarrow$ Botón Enviar Formulario	0.003136	1.6	0.000774

F. Transferencias

La pestaña de transferencias tiene características de diseño muy parecidas a las de un formulario, debido a que varios de sus elementos están pensados para ofrecer información al usuario, mientras otros son espacios para que el usuario ingrese algún dato o valor.

Para este caso se mantendrá el título de Transferencias para denotar las variables y, posteriormente, en las tablas 36 y 37 se muestran los resultados mediante INUSINET.

Elementos de $I_{Transferencias}$:

$I_{Transferencias}(1) \rightarrow$ Logo

$I_{Transferencias}(2) \rightarrow$ Salir

$I_{Transferencias}(3) \rightarrow$ Datos Cuenta

$I_{Transferencias}(4) \rightarrow$ Ayuda

$I_{Transferencias}(5) \rightarrow$ Destino Tarjeta

$I_{Transferencias}(6) \rightarrow Destino Nombre$

$I_{Transferencias}(7) \rightarrow Destino Cantidad$

$I_{Transferencias}(8) \rightarrow Botón Enviar Formulario$

$I_{Transferencias}(9) \rightarrow Switch Modo Detallado$

Por lo que, la longitud de $I_{Transferencias}$:

$$I_{Transferencias}|N| = 9$$

Y los elementos de $I_{Transferencias}$ son los siguientes.

$$I_{Transferencias}\{N\} = \left\{ \begin{array}{l} I_{Transferencias}(1) \\ I_{Transferencias}(2) \\ I_{Transferencias}(3) \\ I_{Transferencias}(4) \\ I_{Transferencias}(5) \\ I_{Transferencias}(6) \\ I_{Transferencias}(7) \\ I_{Transferencias}(8) \\ I_{Transferencias}(9) \end{array} \right\}$$

Tabla 36: Resultados de Interfaz Transferencias de Easy-Bank Vision, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{Transferencias}(n)$	$Cf(n)$	$Df(n)$	$Int(n)$	$Cs(n)$	$Vs(n)$
$I(1) \rightarrow Logo$	20	8955	2	4	1333.795
$I(2) \rightarrow Salir$	4	1065	0	0	808.47
$I(3) \rightarrow Datos Cuenta$	218	45726	Caso Cero	87	3896.67
$I(4) \rightarrow Ayuda$	250	49909	Caso Cero	118	2023.135
$I(5) \rightarrow Destino Tarjeta$	249	66762	21	101	3230.055
$I(6) \rightarrow Destino Nombre$	180	50347	20	26	1432.74
$I(7) \rightarrow Destino Cantidad$	201	58625	22	35	3227.585
$I(8) \rightarrow Botón Enviar Formulario$	74	20394	26	13	2693.505
$I(9) \rightarrow Switch Modo Detallado$	42	11342	0	7	1695.37

Tabla 37: Resultados de INUSINET para la interfaz de Transferencias, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{Transferencias}(n)$	ICE	ICU	ICR
$I(1) \rightarrow Logo$	0.002233	10	0.002999
$I(2) \rightarrow Salir$	0.003756	Elemento no Usado	Elemento sin sacádicos
$I(3) \rightarrow Datos Cuenta$	0.004768	Caso Cero	0.022327
$I(4) \rightarrow Ayuda$	0.005009	Caso Cero	0.058325
$I(5) \rightarrow Destino Tarjeta$	0.00373	11.85714	0.031269
$I(6) \rightarrow Destino Nombre$	0.003575	9	0.018147
$I(7) \rightarrow Destino Cantidad$	0.003429	9.136364	0.010844
$I(8) \rightarrow Botón Enviar Formulario$	0.003629	2.846154	0.004826
$I(9) \rightarrow Switch Modo Detallado$	0.003703	Elemento no Usado	0.004129

5.2.3 Análisis de resultados por interfaz de Easy-Bank Vision

Generar las evaluaciones correspondientes para una experiencia de uso tan grande como lo es Easy-Bank Vision resulta un trabajo bastante extenso y que para que los resultados no estén sesgados requiere una planificación y control de cada uno de los procesos que se tienen que llevar a cabo para obtener los data-sets.

Si bien el marco de trabajo EVUSINET permite implementar técnicas de eye-tracking de manera general para varios procesos de desarrollo de software, durante la realización de las pruebas de este caso de estudio surgieron numerosos contratiempos que no se habían estimado al inicio de las evaluaciones.

Sin embargo, a pesar de las complicaciones fue posible obtener resultados satisfactorios tanto de los patrones de uso generales como particulares que validan la implementación de las técnicas de eye-tracking en la evaluación de interfaces. Uno de los aspectos que es destacable en las características de esta prueba es que los resultados de los resultados de los índices no parecen ganar mucha precisión si el tamaño de la población es mucho mayor que en el caso de estudio del componente login.

Lo anterior es bastante relevante a la hora de generar los estudios de viabilidad para siguientes iteraciones o proyectos ya que para cada sujeto de prueba se tuvieron que añadir las áreas de interés de cada interfaz y diseñar los eventos para cada vez que estas se expusieron en la prueba.

Entonces, debido a que se estableció un tamaño de muestra de 5 sujetos mínimos por tipo de usuario, el caso de estudio Easy-Bank Vision buscó comprobar si analizando una mayor cantidad de pruebas se mejoraría notoriamente la precisión del modelo. Debido a que el rango de valores entre ambos casos de estudio es bastante similar, a primera vista, para los índices independientemente del tamaño de la población, por el momento se establece que realizar más de veinte pruebas no mejora relevantemente la precisión de la evaluación para elementos en particular.

Ampliando lo anterior, incluso en las visualizaciones de patrones de uso generales se optó por generar una muestra representativa de cada tipo de usuario para permitir encontrar información relevante en los diferentes mapas de cada interfaz.

Ya analizadas las características generales de la evaluación se procede a abordar cada una de las interfaces y sus respectivos resultados de evaluación.

A. Login Detallada y Simple

Debido a que las interfaces de Login Detallado y Login Simple tienen características parecidas a la metodología de diseño de interfaces paralelas, es posible comparar cada elemento e interpretar como la dimensión, disposición y tipos de fuente en cada una de las interfaces. Como se observa en la figura 99 los parámetros anteriores son los responsables de que, si bien la secuencia de interacción y el papel que cada elemento funge en la interfaz es el mismo, incluso las pequeñas variaciones de las ubicaciones de los elementos generan patrones de uso bastante diferentes.

En este caso, la evaluación entre ambas interfaces colocó al diseño de Login detallado con mejores prestaciones de uso que su similar Login Simple con respecto a los patrones de uso generales.

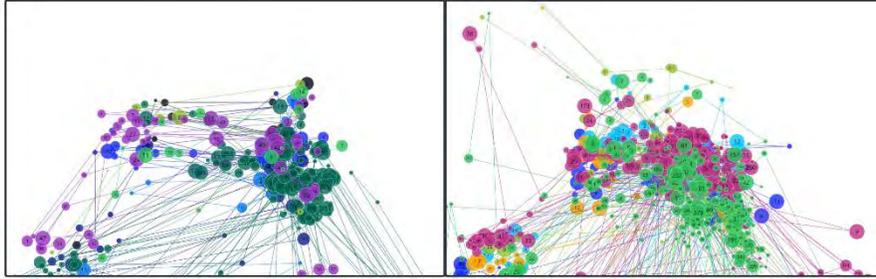


Figura 99: Comparación entre mapas de recorrido de fijaciones de las interfaces Login Detallado y Login Simple respectivamente, (Elaboración propia, 2024).

B. Home

La interfaz Home es la interfaz que más interés provocó para el equipo de evaluación desde las visualizaciones de una sola prueba debido a los interesantes patrones que demostraba. Al realizar las pruebas con la población se lograron confirmar algunas de las interpretaciones que se tenían sobre todo con los botones de la parte superior.

Como se observa en la tabla 38 el elemento con peor resultados en el índice de entendimiento es el primer botón de izquierda a derecha de la interfaz ($I(3)$). De manera individual se puede estimar que el elemento no cumple con las teorías de los resultados de los patrones de uso generales en donde se menciona que estos botones son poco visitados debido a que estos son lo suficientemente claros para el usuario, por lo que los visitan poco, pero al analizar el conjunto de elementos es relevante notar que los demás elementos no presentan malos valores para este índice.

Lo anterior, al contrario de negar la teoría parece reforzarla ya que, una vez que se observa el primer elemento de izquierda a derecha el usuario invierte una mayor cantidad de tiempo en las fijaciones para reconocerlo, pero una vez que lo ha hecho, relaciona todos los demás elementos similares para no tener que volverlos a analizar posteriormente.

Ampliando en la idea de que una vez catalogados los elementos ya no se visitan con frecuencia ni se les termina por leer del todo es que en los resultados cuantitativos de la interfaz los elementos con peor resultado del índice de recorrido son el logo y la etiqueta de salir.

Por otro lado, uno de los elementos mejor evaluados de las interfaces de estudio es el elemento $I(9)$ *Tarjetas*, el cual obtuvo dos segundas mejores evaluaciones en dos índices (ICE e ICU) y un primer lugar en una tercera (ICR), por lo que este elemento indica ser un punto fuerte de la interfaz.

Tabla 38: Resultados de la aplicación de INUSINET en la interfaz Home remarcando elementos que peores y mejores resultados obtuvieron, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{Home}(n)$	ICE	ICU	ICR
$I(1) \rightarrow Logo$	0.006871	8.5	0.000697
$I(2) \rightarrow Salir$	0.003719	Elemento no Usado	0.000577
$I(3) \rightarrow Botón Saldos y Movimientos$	0.002562	7.884615	0.003932
$I(4) \rightarrow Botón estado de cuenta$	0.00422	120	0.002686
$I(5) \rightarrow Botón Inversiones$	0.004852	Elemento no Usado	0.001977
$I(6) \rightarrow Botón Transferencias$	0.002975	8.466667	0.003017
$I(7) \rightarrow Botón Pago Servicios$	0.003729	60	0.001543
$I(8) \rightarrow Ayuda$	0.003399	Caso Cero	0.014159
$I(9) \rightarrow Tarjetas$	0.003902	12.09302	0.06911
$I(10) \rightarrow Switch Modo Detallado$	0.002883	5.8	0.006804

C. Saldos y Movimientos

Observando tanto los resultados de patrones de uso generales y particulares esta interfaz parece haber tenido un desempeño pobre para la población de sujetos de prueba a lo largo de la interacción. Como se menciona en la sección 5.2.1 apartado D los patrones de movimientos sacádicos indican que la mayoría de las visualizaciones fueron poco planeadas y eficientes, si bien las instrucciones de la prueba motivaban este comportamiento, el hecho de que no sea posible observar un solo patrón reconocible invita a los diseñadores a proponer nuevos métodos para representar esta información para futuras iteraciones.

Ampliando lo anterior, si se observan los resultados para el índice de recorrido de la tabla 39 se podrá notar que la mayoría de estos elementos tienen un valor muy bajo comparado con el único valor que el evaluador reconoció como un buen resultado.

Tabla 39: Resultados de la aplicación de INUSINET en la interfaz saldos y Movimientos remarcando el índice con peores resultados generale, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{Sal-Mov}(n)$	ICE	ICU	ICR
$I(1) \rightarrow Logo$	0.00219	4.96	0.010195

<i>I(2) → Salir</i>	0.003209	Elemento no Usado	0.00091
<i>I(3) → Botón Todos los Registros</i>	0.00435	26.33333	0.001915
<i>I(4) → Botón Ingresos</i>	0.004086	8.555556	0.002504
<i>I(5) → Botón Egresos</i>	0.002836	3.407407	0.002137
<i>I(6) → Ayuda</i>	0.004483	Caso Cero	0.01317
<i>I(7) → Tabla de Registros</i>	0.003911	46.92857	0.040272
<i>I(8) → Switch Modo Detallado</i>	0.002532	7	0.002035

D. Pagar Tarjeta

La ventana emergente del caso de uso de pagar tarjeta demostró tener muy buenas prestaciones con respecto a sus patrones de uso particulares ya que representa un elemento altamente identificable como el punto más débil de la interfaz, por lo que, para el diseñador fue sencillo determinar las posibles correcciones para este elemento en versiones posteriores (consúltese tabla 40).

Tabla 40: Resultados de la aplicación de INUSINET en la interfaz Pago Tarjeta remarcando elementos que peores resultados obtuvieron, (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{PagoTarjeta}(n)$	ICE	ICU	ICR
<i>I(1) → Logo</i>	0.003801	27.45455	0.096252
<i>I(2) → Salir</i>	0.003794	Caso Cero	0.074479
<i>I(3) → Botón Todos los Registros</i>	0.003661	23.57143	0.024934
<i>I(4) → Botón Enviar Formulario</i>	0.003136	1.6	0.000774

E. Transferencias

Contrastando con las interpretaciones que se mencionaron en la sección 5.2.2 apartado F, el formulario de Transferencias mostró mejores resultados de recorrido de lo esperado, ya que la mayoría de los valores obtenidos para este índice mostraron tener buenos resultados comparados con los promedios de otras interfaces (consúltese tabla 41).

Este detalle relevante en los datos mejora aún más la constante confrontación entre los patrones de uso generales y particulares que están comparando continuamente diferentes interpretaciones y que, para el diseñador y evaluador generan un ejercicio de retrospección que, a medio y largo plazo mejoran la objetividad de la evaluación de interfaces.

Tabla 41: Resultados de la aplicación de INUSINET en la interfaz Transferencias remarcando elementos que peores resultados obtuvieron, , (Elaboración en conjunto con los miembros del equipo de desarrollo, 2023).

$I_{Transferencias}(n)$	ICE	ICU	ICR
$I(1) \rightarrow Logo$	0.002233	10	0.002999
$I(2) \rightarrow Salir$	0.003756	Elemento no Usado	Elemento sin sacádicos
$I(3) \rightarrow Datos Cuenta$	0.004768	Caso Cero	0.022327
$I(4) \rightarrow Ayuda$	0.005009	Caso Cero	0.058325
$I(5) \rightarrow Destino Tarjeta$	0.00373	11.85714	0.031269
$I(6) \rightarrow Destino Nombre$	0.003575	9	0.018147
$I(7) \rightarrow Destino Cantidad$	0.003429	9.136364	0.010844
$I(8) \rightarrow Botón Enviar Formulario$	0.003629	2.846154	0.004826
$I(9) \rightarrow Switch Modo Detallado$	0.003703	Elemento no Usado	0.004129

5.2.4 Análisis de la experiencia general de Easy-Bank Vision

Uno de los descubrimientos más relevantes que se obtuvo al analizar a detalle los patrones de uso particulares es su capacidad para generar relaciones o discordancias entre los patrones de uso generales. Esto permite a los diseñadores y evaluadores tener discusiones más profundas sobre el uso real que se le da al sistema.

Otro aspecto altamente relevante es que al evaluar interfaces con propósitos y métodos diferentes se logra tener un análisis mucho más amplio de interfaces que anteriormente no habían sido analizadas con estas técnicas.

Ampliando lo anterior, la interfaz de Saldos y Movimientos representó un nuevo tipo de análisis de interfaces para tablas y registros de información, por lo que los debates del uso de esta interfaz interpretando los resultados fueron amplios y muy diversos.

Una de las grandes ventajas de proponer posibles interpretaciones de los resultados es que a la par surgen distintas variantes en las ideas de los diseñadores para “modelar” la experiencia mediante la modificación de las propiedades de los elementos.

Como se ha mencionado en el caso de uso del componente login, si bien los resultados de uso particulares buscan puntos débiles de la interacción, es el conjunto de todos los elementos en la interfaz los que modifican la efectividad de algunos en específico. Es recomendable virar la atención

a los elementos que conforman tareas muy específicas, y confrontarlo a los recorridos de los patrones de uso generales.

Por último, pero que resulta un gran avance en la profundización de las posibles interpretaciones que se realizan sobre los patrones de uso generales y particulares, se ha encontrado que es posible reconocer que los valores de un índice son mejores o peores que el promedio del resto de las interfaces de estudio. Si se comparan directamente las tablas de resultados de índices de las interfaces Transferencias (tabla 41) y de Saldos y Movimientos (tabla 39) es notorio que los valores del índice ICR son mucho mejores en la primera que en la segunda. Como se menciona en la idea anterior, esta característica no sería posible de no tener una buena variedad de tipos de interfaces de estudio.

Antes de generar una evaluación de la población se propuso que la cantidad de fijaciones no era un parámetro que alterara directamente los resultados obtenidos, sin embargo, el descubrimiento del comportamiento de los resultados de las interfaces Transferencias y Saldos y Movimientos en el valor del ICR invitan a pensar que sí existe una mejora en la precisión de los datos al aumentar la cantidad de pruebas. Teoría que se pretende comprobar en trabajos futuros como se aborda en la sección referente a este tópico.

5.3 Discusiones

A lo largo de los dos casos de estudios se fueron aplicando y analizando las diferentes propuestas que se proponen en este trabajo de investigación. Partiendo de lo general a lo particular, es notoria la calidad de información que brindan las técnicas de eye-tracking, sin embargo, y también como se insistió a lo largo y ancho de las secciones de desarrollo y resultados, una correcta evaluación depende de dos grandes factores: la calidad de los procesos para aplicar las técnicas de eye-tracking en la evaluación de interfaces y la sensibilidad del diseñador y evaluador del software para realizar comentarios y sugerencias atinadas de los resultados obtenidos con respecto a las expectativas previas que tenían en el diseño.

Una de las primeras preguntas que surgen a la hora de compartir los resultados es si estos son lo suficientemente amplios para que otros procesos de desarrollo los adopten, pero lo suficientemente detallados para que tengan resultados válidos en sus evaluaciones.

TESIS TESIS TESIS TESIS TESIS

Durante el caso de estudio 1, los métodos de evaluación son muy parecidos a los recomendados por EVUSINET, sin embargo, el caso de estudio 2 necesitó nuevas maneras de representar información de manera más específica para que pudiera ser más clara la evaluación.

Ampliando la comparación entre ambos casos de estudio, cabe destacar la diferencia entre la cantidad de la población de sujetos de prueba que, en lugar de disipar las dudas sobre la cantidad de las pruebas, aumentó la discrepancia entre las hipótesis del efecto de la cantidad de usuarios en los resultados de uso mediante eye-tracking. Si bien en un principio se plantearon ambos casos de estudio de manera similar, conforme el desarrollo fue avanzando y se contaba con más gente involucrada en el proyecto las características de la experiencia cambiaron, hablando particularmente del caso de estudio 2.

Ahora bien, si se analiza de manera independiente el desempeño de cada caso de estudio se considera un éxito bueno o muy bueno ya que las bases de la fundamentación de las propuestas de evaluación objetiva son mayoritariamente teóricas y, además, las técnicas de eye-tracking han sido muy poco estudiadas en el campo de la ingeniería de software.

Desarrollando lo anterior, la propuesta más fuerte que presenta este trabajo de investigación son los índices para la evaluación de interfaces mediante las tecnologías de eye-tracking, de nombre, INUSINET. Este modelo de evaluación representa un primer paso para lograr un entendimiento entre algunas características de la visión humana con el uso de interfaces, sin embargo, y como se aborda de mejor manera en la sección siguiente referente a conclusiones y trabajo futuro, es posible mejorar ampliamente el modelo para convertirlo en una herramienta más precisa y amplia para más casos de estudio y métricas obtenidas mediante esta novedosa tecnología.

Conclusiones y Trabajo Futuro

Para elaborar este trabajo tuvo que realizarse un análisis extensivo y exhaustivo de las técnicas de atención visual del ser humano, el funcionamiento de las tecnologías de eye-tracking y los conceptos de ingeniería de software que mejor se adaptasen a este tipo de análisis.

Es probable que en esta etapa del trabajo el lector pueda estimar que las propuestas son sencillas y que varias de las ecuaciones y modelos de evaluación de interfaces mediante eye-tracking son sencillas y tengan muchas oportunidades de mejora, sin embargo, este primer paso en el análisis cuantitativo de experiencias de uso propone una gran cantidad de visualizaciones y diagramas que si bien están basados en modelos de lenguajes como UML o los más convencionales entre las metodologías ágiles, permiten compartir de manera mucho más amplia las propuestas formales.

Repasando los objetivos de la investigación se estableció una manera de identificar las métricas de eye-tracking que de mejor manera permitieran la evaluación de la experiencia, pero para seleccionarlas fue necesario establecer todo un marco metodológico para fundamentar la elección de estas.

Ahora bien, el marco de trabajo EVUSINET es una herramienta que no solo contempla la implementación de eye-tracking en los procesos de desarrollo, también pretende mejorar las capacidades de evaluación de procesos que no implementen esta tecnología.

Como se menciona anteriormente, es posible mejorar cada una de las propuestas presentadas en este trabajo para poder alcanzar un siguiente nivel de evaluación y aprovechamiento de esta herramienta tan novedosa como lo es el eye-tracking.

De manera específica, uno de los aspectos más destacables que se encontraron en este trabajo es encontrar un modelo de patrones de uso convergentes, esto hace referencia a que una población de sujetos, en su mayoría, realizan un mismo patrón, sin embargo, y considerando la metodología de diseño de interfaces centrada en el usuario, es necesario considerar a los usuarios que utilizaron la interfaz de manera diferente.

El modelo de diseño de interfaces con patrones de uso divergente busca atacar problemas de usabilidad de personas con dificultades visuales o con condiciones cognitivas específicas como dislexia, analfabetismo, incluso TDAH, entre otras condiciones.

Por lado del marco de trabajo EVUSINET, es posible mejorarlo con una etapa de despliegue de resultados, en los cuáles se automaticen los resultados de la etapa de evaluación, y permitan exponer los resultados de manera más amplia y atractiva.

Para INUSINET se plantea una expansión en las métricas e interpretaciones estipuladas para generar mejores y más precisas evaluaciones, así como apoyarse de los nuevos métodos de EVUSINET para presentar resultados visualmente más claros e interpretables a simple vista.



Productividad Científica

Durante la elaboración de este trabajo de investigación se desprendieron varios proyectos relacionados a los temas y propuestas abordadas con el fin de compartir en foros nacionales e internacionales parte de los descubrimientos al público especializado.

1.- Artículos científicos

Aplicación de técnicas de eye-tracking para la mejora del Testing y procesos de desarrollo de software.
International Conference in Software Engineering Research and Innovation (CONISOFT) 11-2023 Artículo Co-autores: Eduardo Rodríguez López, Francisco Álvarez Rodríguez
Aplicación de técnicas de eye-tracking para la mejora del testing y procesos de desarrollo de software.pdf

2.-Memorias en congresos

Eye-tracking para el estudio de la carga cognitiva y su relación con el desempeño académico enfocado en el área de matemáticas.
Conf-IRM 10-2022 Memorias de congreso Co-autores: Eduardo Rodríguez López, Francisco Álvarez Rodríguez
Artículo Conf-IRM.pdf

3.-Ponencias

Aplicación de técnicas de eye-tracking para la mejora del Testing y procesos de desarrollo de software.

International Conference in Software Engineering Research and Innovation (CONISOFT) 11-2023 | Ponencia presencial, León, Gto. México

Modelo para la Aplicación del Análisis del Comportamiento de Mirada Mediante Eyetracking de la Interacción Usuario- Sistema para Mejorar la Calidad de la UX y Mejorar los Procesos de Desarrollo de Software.

Seminario Inter Institucional de Ingeniería de Software (UAA en conjunto con CENIDET) 10-2022 | Ponencia Presencial, Aguascalientes, Ags, México.

Implementación de métricas obtenidas mediante eye-tracking para la evaluación del uso de interfaces.

1er Congreso Internacional Multidisciplinario de la Investigación Científica (UAA)
10-2022 | Ponencia Presencial, Aguascalientes, Ags, México.

The certificate is issued by the Universidad Autónoma de Aguascalientes (UAA) through its Dirección General de Investigación y Posgrado (DGIP). It recognizes the participation of three individuals in a specific presentation at the 1st International Multidisciplinary Congress of Scientific Research (1er Congreso Internacional Multidisciplinario de la Investigación Científica). The presentation is titled 'Implementación de métricas obtenidas mediante eye-tracking para la evaluación del uso de interfaces' and is part of the 'CIENCIAS EXACTAS E INGENIERÍAS' table. The congress is held in Aguascalientes, Ags., on October 13, 2023, and is a hybrid modality. The certificate is signed by the Director General of Research and Postgraduate Studies, Dr. Francisco Javier Pedroza Cabrera, and the Rector, Dr. en Admón. Sandra Yesenia Pinzón Castro.

La Universidad Autónoma de Aguascalientes otorga la presente

CONSTANCIA

a

**ING. ÁNGEL VILLEGAS ORTÍZ;
M.C.C EDUARDO EMMANUEL RODRÍGUEZ LÓPEZ;
DR. FRANCISCO ÁLVAREZ RODRÍGUEZ**

por su participación con el **CARTEL:**

Implementación de métricas obtenidas mediante eye-tracking para la evaluación del uso de interfaces

en el marco del

1er Congreso Internacional Multidisciplinario de la Investigación Científica

Modalidad híbrida

en la mesa de: **CIENCIAS EXACTAS E INGENIERÍAS**

Se Lumen Proferre
Aguascalientes, Ags., a 13 de octubre de 2023

Dr. Francisco Javier Pedroza Cabrera
Director General de Investigación y Posgrado

Dr. en Admón. Sandra Yesenia Pinzón Castro
Rectora

Referencias

- Abbott, R. (1977). Review of "DECset-8000 System User's Guide, Vol #1, #2, by Vincent Manis, Peter van Den Bosch." *SIGDOC Asterisk J. Comput. Doc.*, 3(10).
<https://doi.org/10.1145/1110777.1110780>
- Albornoz, C. (2014). Diseño de Interfaz Gráfica de Usuario. *WICC 2014 XVI Workshop de Investigadores En Ciencias de La Computación*, 540–544.
- Andrade-Lotero, L. (2012). Teoría de la carga cognitiva, diseño multimedia y aprendizaje: un estado del arte. *Revista Internacional de Investigación En Educación*, 5(10), 75–92.
<http://magisinvestigacioneducacion.javeriana.edu.co/>
- Balladares-Burgos, J. A. (2018). Diseño pedagógico de la educación digital para la formación del profesorado. *Revista Latinoamericana de Tecnología Educativa*, 17(1).
<https://doi.org/10.17398/1695-288X.17.1.41>
- Barra, C. (1998). Software e Ingeniería de Software. *Revista de Marina*, 1–98.
- Bastien, C. (2010). Usability testing: some current practices and research questions. *International Journal of Medical Informatics*, 18–23.
- Blake, R., & Lee, S. H. (2005). The role of temporal structure in human vision. In *Behavioral and Cognitive Neuroscience Reviews* (Vol. 4, Issue 1, pp. 21–42).
<https://doi.org/10.1177/1534582305276839>
- Blancarte, O. (2016). *Introducción a los Patrones de Diseño* (Primera Edición, Vol. 1).
- Bødker, S. (2015). Third-Wave HCI, 10 Years Later-Participation and Sharing. *Interactions*, 22(5), 24–31. <https://doi.org/10.1145/2804405>
- Cairns, P., & Cox, A. L. (2008). *Research methods for human-computer interaction*. Cambridge University Press.
- Carr, C. T., McGinnity, T. M., & McDaid, L. J. (2004). Integration of UML and VHDL-AMS for analogue system modelling. *Formal Aspects of Computing*, 16(1), 80–94.
<https://doi.org/10.1007/s00165-003-0027-0>
- Cartwright, M., & Shepperd, M. (2000). An Empirical Investigation of an Object-Oriented Software System. *IEEE Transactions on Software Engineering*, 26(8), 786–796.
- Chaplin, P. K. N., & Bradford, G. E. (2011). A historical review of distance vision screening eye charts: what to toss, what to keep, and what to replace. *NASN School Nurse (Print)*, 26(4), 221–228. <https://doi.org/10.1177/1942602X11411094>
- Charland, A., & Leroux, B. (2011). Mobile Application Development: Web vs. Native. *Commun. ACM*, 49–53. <https://doi.org/10.1145/1941487.1941504>
- Clement, A., & Gotlieb, C. C. (1986). *Evolution of an Organizational Interface: The New Business Department at a Large Insurance Firm*.

- Coleman, G., & Connor, R. (2008). Investigating Software Process in Practice: A Grounded Theory Perspective. *Journal of Systems and Software*, 81(5), 772–784.
- Colmenero, J. M., Catena, A., & Fuentes, L. J. (2001). *Atención visual: Una revisión sobre las redes atencionales del cerebro*. 17(1), 45–67.
- Cortes-Camarillo, C. A., Alor-Hernández, G., Olivares-Zepahua, B. A., Rodríguez-Mazahua, L., & Gustavo Peláez-Camarena, S. (2016). Análisis comparativo de patrones de diseño de interfaz de usuario para el desarrollo de aplicaciones educativas Comparative Analysis of User Interface Design Patterns for Developing Educational Applications. In *Research in Computing Science* (Vol. 126).
- Darejeh, A., & Singh, D. (2013). A review on user interface design principles to increase software usability for users with less computer literacy. *Journal of Computer Science*, 9(11), 1443–1450. <https://doi.org/10.3844/jcssp.2013.1443.1450>
- Dehlinger, J., & Dixon, J. (2011). Mobile application software engineering: Challenges and research directions. *Workshop on Mobile Software Engineering*, 2, 29–32.
- Djamasbi, S. (2014). Eye Tracking and Web Experience. In *AIS Transactions on Human-Computer Interaction* (Vol. 2, Issue 6).
- Drusch, G., Bastien, C., & Dinet, J. (2011). From gaze plots to eye fixation patterns using a clustering method based on Hausdorff distances. *UFR Sciences Humaines et Arts, Université Paul Verlaine*, 169.
- Duchowski, A. (2017). *Eye Tracking Methodology* (3rd ed.). Springer Nature. <https://doi.org/10.1007/978-3-319-57883-5>
- Eastman, C. M. (1964). A technique for computer detection and correction of spelling errors. *IBM J. Res. Dev*, 7(3), 107–121.
- Engels, G., & Whittle, J. (2012). Ten years of software and systems modeling Reflecting on the past, looking to the future. *Softw Syst Model*, 11, 467–470. <https://doi.org/10.1007/s10270-012-0280-x>
- Evans, K. K., Horowitz, T. S., Howe, P., Pedersini, R., Reijnen, E., Pinto, Y., Kuzmova, Y., & Wolfe, J. M. (2011). Visual attention. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(5), 503–514. <https://doi.org/10.1002/wcs.127>
- Florián, B. E., Solarte, O., & Reyes, J. M. (2010). Propuesta para incorporar evaluación y pruebas de usabilidad dentro de un proceso de desarrollo de software. *Revista EIA*, 13, 123–141.
- Glinz, M. (2000). *Problems and Deficiencies of UML as a Requirements Specification Language*.
- Gómez, D., Jústiz, D., & Delgado, M. (2013). *Unit Tests of Software in a University Environment*. 17(1), 69–77.
- Gomolka, Z., Kordos, D., & Zeslowska, E. (2020). The application of flexible areas of interest to pilot mobile eye tracking. *Sensors (Switzerland)*, 20(4). <https://doi.org/10.3390/s20040986>

- Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Parkkonen, L., & Hämäläinen, M. S. (2014). MNE software for processing MEG and EEG data. *NeuroImage*, 86, 446–460. <https://doi.org/10.1016/j.neuroimage.2013.10.027>
- Gros, B., Escofet, A., & Marimónmartí, M. (2016). *Los patrones de diseño como herramientas para guiar la práctica del profesorado*. 15(3). <https://doi.org/10.17398/1695288X.15.3.11>
- Gutiérrez de Piñeres Botero, C. (2019). *Visión humana y movimientos oculares*. <https://commons.wikimedia.org/wiki/File:Blau->
- Hess, T. M., & Slaughter, S. J. (1990). Schematic Knowledge Influences on Memory for Scene Information in Young and Older Adults. *Developmental Psychology*, 26(5), 855–865.
- Johnson, J. (2010). *Designing with the Mind in Mind Simple Guide to Understanding User Interface Design Rules*.
- Landa-Durán, P., García-Pacheco, Á., & Serrano-Alvarado, K. (2017). Estudio de Usabilidad como estrategia de apoyo para evaluaciones de aula virtuales. In *Revista Electrónica de Psicología Iztacala* (Vol. 20, Issue 2). www.revistas.unam.mx/index.php/repwww.iztacala.unam.mx/carreras/psicologia/psiclin
- Lawson, H. B. (2018). The March into the Black Hole of Complexity. In *Communications of the ACM* (Vol. 61, Issue 5, pp. 43–45). Association for Computing Machinery. <https://doi.org/10.1145/3201606>
- Lukito Edi, E., Syarif Sihabudin Sahid, D., Insap Santosa, P., Ferdiana, R., & Edi, L. N. (2018). *Evaluation and measurement of Learning Management System based on user experience*. <https://doi.org/10.1109/INAES.2016.7821910>
- Luna-García, H., Mendoza-González, R., & Álvarez-Rodríguez, F. J. (2015). Design patterns to enhance accessibility and use of social applications for older adults. *Comunicar*, 23(45), 85–93. <https://doi.org/10.3916/C45-2015-09>
- Marques, L., Matsubara, P. G., Nakamura, W. T., Ferreira, B. M., Wiese, I. S., Gadelha, B. F., Zaina, L. M., Redmiles, D., & Conte, T. U. (2021). Understanding ux better: A new technique to go beyond emotion assessment. *Sensors*, 21(21). <https://doi.org/10.3390/s21217183>
- Martínez Barros, M., & Vásquez Muriel, N. (2021). *Evaluación del marketing sensorial en plataformas de e-commerce como influenciador en la intención de compra de productos para el cuidado de la piel*.
- Martínez Gutiérrez, F. (2008). Eyetracking y usabilidad: Claves de investigación en los procesos de lectura en línea. *Revista de Comunicación de La SEECI*, 0(16), 98. <https://doi.org/10.15198/seeci.2008.16.98-114>
- Martínez, V. (2017). Educación presencial versus educación a distancia. *La Cuestión Universitaria*, 9, 108–116. <http://webcast.berkeley.edu>

- Mondal, S., Pratim Das, P., & Bhattacharjee Rudra, T. (2022). *Measuring code comprehension effort using code reading pattern*. <https://doi.org/https://doi.org/10.1007/s12046-022-01876-5>
- Morejón Labrada, S. (2020). Principios del proceso de Diseño de Interfaz de Usuario. *Revista Cubana de Transformación Digital*, 1(3), 143–155. <http://orcid.org/0000-0003-0064-8758>
- Nagappan, M., & Shihab, E. (2016). Future trends in software engineering research for mobile apps. *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering, SANER 2016, 2016-January*, 21–32. <https://doi.org/10.1109/SANER.2016.88>
- Nathans, J. (1999). The Evolution and Physiology of Human Review Color Vision: Insights from Molecular Genetic Studies of Visual Pigments Present-day vertebrates vary enormously in the so-phistication of their color vision, the density and spatial distribution of cone classes, and the number and absorption maxima of their cone pigments (Figure 2; Lythgoe. *Neuron*, 24, 299–312.
- Navarro Cadavid, A., Fernández Martínez, J. D., & Morales Vélez, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva*, 11(2), 30–39.
- Nokleberg, C., & Hawkes, B. (2021). Application frameworks. *Communications of the ACM*, 64(7), 42–49. <https://doi.org/10.1145/3446796>
- Oppermann, R. (2002). User-interface design. *Handbook on Information Technologies for Education and Training*, 233–248.
- Parodi, G., & Julio, C. (2016). ¿Dónde se posan los ojos al leer textos multimedios disciplinarios? Procesamiento de palabras y gráficos en un estudio experimental con eye tracker. *Revista Signos*, 49, 149–183. <https://doi.org/10.4067/S0718-09342016000400008>
- Parra, M. A. (2004). Método para el estudio de los movimientos oculares. In *Colomb Med* (Vol. 35).
- Paucar Ati, E. (2017). *Desarrollo de pruebas unitarias automáticas para el sistema web escolar del instituto tecnológico superior stanford*.
- Priestley, M., & Rintjema, L. (1995). The VisualAge C++ for OS/2 User's Guide: a multi-writer, single-sourcing challenge. In *Proceedings of the 13th Annual International Conference on Systems Documentation: Emerging from Chaos: Solutions for the Growing Complexity of Our Jobs* (pp. 130–137). Association for Computing Machinery. <https://doi.org/10.1145/223984.224009>
- Prieto Álvarez, C. G., Figueroa Martínez, J., & Moreno, M. (2015). *Adaptación de las Metodologías Tradicionales Cascada y Espiral para la Inclusión de Evaluación Inicial de Usabilidad en el Desarrollo de Productos de Software en México*. Universidad Tecnológica de la Mixteca.
- Ramírez Coronel, A. (2018). Relación entre los movimientos sacádicos , lateralidad y proceso lector. *Espirales*.
- Reinholtz, K. (2000). *Java will be faster than C++*. 25–28. <https://doi.org/10.1145/345105.352548>
- Ribera Turró, M. (2005). Evolución y tendencias en la IHC. *El Profesional de La Información*, 15(6), 414–422.

- Ruiz, A., Arciniegas, J. L., & Giraldo, W. J. (2018). Characterization of user interface development frameworks for interactive systems based on video content distribution. In *Revista chilena de ingeniería* (Vol. 26, Issue 2).
- Ruiz-Contreras, A., & Cansino, S. (2005). Neurofisiología de la interacción entre la atención y la memoria episódica-revisión de estudios en modalidad visual. *REV NEUROL*, *41*, 733–743.
- Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, *35*(3), 8–13. <https://doi.org/10.1145/1764810.1764814>
- Sakamoto, Y., & Love, B. C. (2004). *Schematic Influences on Category Learning and Recognition Memory*. <https://doi.org/10.1037/0096-3445.133.4.534>
- Sammet, J., & Mahoney, M. (1995). *Software History*. 3. <http://www.sei.cmu.edu.CERThomepage>. <http://www.cert.org>.
- Sancho-Vinuesa, T., & Pérez-Navarro, A. (2009). *Problems Posed by Mathematical Notation in E-Learning: Transcription and Edition of Formulae*. <http://www.openmath.org/>
- Saquero Ros, C., & Berná Martínez, J. (2019). *Diseño y desarrollo de una metodología de evaluación de la UX*.
- Seckler, M., Heinz, S., Bargas-Avila, J. A., Opwis, K., & Tuch, A. N. (2014). *Designing Usable Web Forms-Empirical Evaluation of Web Form Improvement Guidelines*. <https://doi.org/10.1145/2556288.2557265>
- Sharafi, Z., Shaffer, T., Sharif, B., & Gueheneuc, Y. G. (2016). Eye-tracking metrics in software engineering. *Proceedings - Asia-Pacific Software Engineering Conference, APSEC, 2016-May*, 96–103. <https://doi.org/10.1109/APSEC.2015.53>
- Silvis-Cividjian, N., Went, M., Jansma, R., Bonev, V., & Apostolov, E. (2021). Good Bug Hunting: Inspiring and Motivating Software Testing Novices. *Germany Computer Science Education, Virtual Event*, *1*. <https://doi.org/10.1145/3430665.3456330>
- Sommerville, I. (2011). *Ingeniería de Software*. Pearson.
- Stephanidis, C., Salvendy, G., Antona, M., Chen, J. Y. C., Dong, J., Duffy, V. G., Fang, X., Fidopiastis, C., Fragomeni, G., Fu, L. P., Guo, Y., Harris, D., Ioannou, A., Jeong, K. ah (Kate), Konomi, S., Krömker, H., Kurosu, M., Lewis, J. R., Marcus, A., ... Zhou, J. (2019). Seven HCI Grand Challenges. In *International Journal of Human-Computer Interaction* (Vol. 35, Issue 14, pp. 1229–1269). Taylor and Francis Inc. <https://doi.org/10.1080/10447318.2019.1619259>
- Stroustrup, B. (1993). *A History of C++: 1979–1991*. <https://doi.org/10.1145/154766.155375>
- Sulikowski, P., Kucznerowicz, M., Bąk, I., Romanowski, A., & Zdziebko, T. (2022). Online Store Aesthetics Impact Efficacy of Product Recommendations and Highlighting. *Sensors*, *22*(23). <https://doi.org/10.3390/s22239186>
- Taniş, H., Akçay, A., Yılmaz, N., Yiğit, M. F., & Tüzün, H. (2022). How usable is Coursera? A usability analysis through eye-tracking and authentic tasks. *Participatory Educational Research*, *9*(4), 379–395. <https://doi.org/10.17275/per.22.96.9.4>

- Tarazona Evangelista, L., Aguilar Baca, Y., & Rivera Flores, V. (2018). *Uso de la teoría del color para mejorar la atención visual en estudiantes del segundo grado de secundaria de la I.E. San Martín de Socabaya, Arequipa 2017*. UNIVERSIDAD NACIONAL SAN AGUSTÍN DE AREQUIPA.
- Tarouco, L. (2006). *Alfabetización visual para reducir la sobrecarga cognitiva en un material educativo digital*.
- Tullis, T. S. (2019). Nigel Bevan: An Overview of His Contributions to Usability and UX. In *Journal of Usability Studies* (Vol. 14).
- Tupikovskaja-Omovie, Z., & Tyler, D. J. (2022). Experienced versus inexperienced mobile users: eye tracking fashion consumers' shopping behaviour on smartphones. *International Journal of Fashion Design, Technology and Education*, 15(2), 178–186.
<https://doi.org/10.1080/17543266.2021.1980614>
- Tyma, P. (1998). Why Are We Using Java Again? *Commun. ACM*, 41(6), 38–42.
<https://doi.org/10.1145/276609.276617>
- Ustárroz, J., & Grandi, F. (2016). Sobre la memoria de trabajo y la memoria declarativa: propuesta de una clarificación conceptual. *Cuadernos de Neuropsicología Panamerican Journal of Neuropsychology*, 10, 13–31. <https://doi.org/10.7714/CNPS/10.3.201>
- Varakin, D. A., Levin, D. T., Fidler, R., & Alexander, D. (2004). Unseen and Unaware: Implications of Recent Research on Failures of Visual Awareness for Human-Computer Interface Design. *HUMAN-COMPUTER INTERACTION*, 19, 389–422.
- Velasco Mamián, A., & Varón Torres, A. (2018). *Fundamentos básicos para crear una interfaz gráfica para un aplicativo web*.
- Velásquez Restrepo, S. M., Vahos-Montoya, J. D., Gómez-Adasme, M. E., Pino -Martínez, A. A., Restrepo-Zapata, E. J., & Londoño-Marín, S. (2019). Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software. *Medellín-Colombia Revista CINTEX*, 24(2), 13–23.
- Vite Ceballos, H., Dávila Cuesta, J., & Molina Montero, B. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espiraes Revista Multidisciplinaria de Investigación*. <https://doi.org/10.31876/re.v2i17.269>
- Wong, W., Bartels, M., & Chrobot, N. (2014). Practical eye tracking of the ecommerce website user experience. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8516 LNCS(PART 4), 109–118.
https://doi.org/10.1007/978-3-319-07509-9_11
- Zumba Gamboa, P. J., & León Arreaga, A. C. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. Evolution of the Methodologies and Models used in Software Development. *INNOVA Research Journal*, 3(10), 20–33.

Retribución social



Aguascalientes, Ags., a 13 de marzo de 2023.

CONSTANCIA DE ACTIVIDADES DE RETRIBUCIÓN SOCIAL:

ASISTENTE DE PROFESOR DE PREGRADO.

A quien corresponda:

Por medio del presente se hace constar que el alumno **Angel Eduardo Villegas Ortiz**, estudiante de la **Maestría en Ciencias de la Computación, Matemáticas Aplicadas**, fungió como asistente de profesor de pregrado con **4 horas semanales**, en la materia que impartí: **"Seminario de Investigación I"** en la carrera **Ingeniería en Computación Inteligente** de **9no semestre** en el periodo agosto – diciembre 2022, en consideración de las actividades de retribución social solicitadas como alumno de doctorado dentro de nuestra institución.

Sin más por el momento, quedo atento a cualquier comentario

A handwritten signature in black ink, appearing to read 'Francisco', written over a horizontal line.

Dr. Francisco Javier Álvarez Rodríguez

Profesor investigador.



Aguascalientes, Ags., a 13 de diciembre de 2023

CONSTANCIA DE ACTIVIDADES DE RETRIBUCIÓN SOCIAL:

ASISTENTE DE PROFESOR DE PREGRADO.

A quien corresponda:

Por medio del presente se hace constar que el alumno **Ángel Eduardo Villegas Ortiz**, estudiante de la **Maestría en Ciencias de la Computación, Matemáticas Aplicadas**, fungió como asistente de profesor de pregrado con **4 horas semanales**, en la materia que impartí: **"Seminario de Investigación I"** en la carrera **Ingeniería en Computación Inteligente** de **9no semestre** en el periodo agosto – diciembre 2023, en consideración de las actividades de retribución social solicitadas como alumno de maestría dentro de nuestra institución.

Sin más por el momento, quedo atento a cualquier comentario

A handwritten signature in black ink, appearing to read 'Francisco Javier Álvarez Rodríguez', written over a horizontal line.

Dr. Francisco Javier Álvarez Rodríguez

Profesor investigador.



Aguascalientes, Ags., a 13 de diciembre de 2023

CONSTANCIA DE ACTIVIDADES DE RETRIBUCIÓN SOCIAL:

ASISTENTE DE INVESTIGACIÓN.

A quien corresponda:

Por medio del presente se hace constar que el alumno **Angel Eduardo Villegas Ortiz**, estudiante de la **Maestría en Ciencias de la Computación, Matemáticas Aplicadas**, fungió como asistente de investigación con **4 horas semanales**, con estudiante de pregrado de la carrera **Ingeniería en Computación Inteligente** de **3er semestre** en el periodo agosto – diciembre 2023, en consideración de las actividades de retribución social solicitadas como alumno de maestría dentro de nuestra institución.

Sin más por el momento, quedó atento a cualquier comentario

A handwritten signature in black ink, appearing to read 'Francisco Javier Álvarez Rodríguez', written over a horizontal line.

Dr. Francisco Javier Álvarez Rodríguez

Profesor investigador.

Cursos





kaggle
CERTIFICATE OF COMPLETION

Angel Eduardo Villegas Ortíz
HAS SUCCESSFULLY COMPLETED THE COURSE

Pandas

ON FEBRUARY 8, 2023

Aleksey Bilogur
ALEKSEY BILOGUR, KAGGLE INSTRUCTOR

Alexis Cook
ALEXIS COOK, HEAD OF KAGGLE LEARN

This certificate is framed by a black border. To the right of the text, there are overlapping abstract shapes in yellow, green, and blue. The background of the certificate area is white.



kaggle
CERTIFICATE OF COMPLETION

Angel Eduardo Villegas Ortíz
HAS SUCCESSFULLY COMPLETED THE COURSE

Machine Learning Explainability

ON MARCH 24, 2023

Dan Becker
DAN BECKER, KAGGLE INSTRUCTOR

Alexis Cook
ALEXIS COOK, HEAD OF KAGGLE LEARN

This certificate is framed by a black border. To the right of the text, there are overlapping abstract shapes in yellow, green, and blue. The background of the certificate area is white.



