

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA  
DEPARTAMENTO DE INFORMÁTICA  
SANTIAGO - CHILE



“DISEÑO E IMPLEMENTACIÓN DE UN SOFTWARE  
BASADO EN ANÁLISIS DE SENTIMIENTO PARA LOGRAR  
CONFORMIDAD EN CALIDAD Y MADUREZ DE UN  
SISTEMA”

NICOLÁS IGNACIO ARAYA URRUTIA

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Mauricio Olivares Faúndez  
Profesor Correferente: Jose Luis Martí Lara

Marzo - 2023

## **DEDICATORIA**

A mi yo de 17 años, nunca dejes de perseguir tu sueño.

## **AGRADECIMIENTOS**

A Dios, mi familia y novia, amigos y a todos quienes de alguna forma me ayudaron en esta larga pero gratificante carrera.

## RESUMEN

**Resumen—** En los proyectos de software actuales, uno de los problemas más frecuentes es la falta de cumplimiento de los estándares de calidad y madurez. Dicha problemática se traduce en funcionalidades poco útiles, ineficiencia en términos de costos debido al desperdicio de horas de desarrollo, además de una percepción negativa sobre la facilidad de uso, lo que conduce a una experiencia de usuario deficiente. Una de las causas principales de este problema radica en la escasez de herramientas para evaluar la usabilidad y comprender la opinión de los usuarios respecto al sistema. Con el objetivo de abordar esta problemática, se ha diseñado e implementado una plataforma automatizada que permite evaluar la usabilidad de un software. Esta plataforma emplea cuestionarios estandarizados y herramientas de minería de opinión para obtener métricas precisas sobre el sentimiento de los usuarios. Además, para validar esta solución, se realizaron tres iteraciones de mejora continua en un software de escritorio utilizando dicha herramienta. Esta metodología se basó en las métricas recopiladas sobre usabilidad y sentimiento de los usuarios, lo que permitió observar una evolución positiva en todos los aspectos evaluados.

**Palabras Clave—** Calidad de software; Usabilidad; Experiencia de Usuario (UX); Análisis de Sentimiento.

## ABSTRACT

**Abstract—** In current software projects, one of the most frequent problems is the lack of compliance with quality and maturity standards. This problem translates into unusable functionalities, inefficiency in terms of costs due to wasted development hours, as well as a negative perception about ease of use, which leads to a poor user experience. One of the main causes of this problem lies in the scarcity of tools to evaluate usability and understand users' opinions regarding the system. With the aim of addressing this problem, an automated platform has been designed and implemented that allows evaluating the usability of software. This platform uses standardized questionnaires and opinion mining tools to obtain precise metrics on user sentiment. Furthermore, to validate this solution, three continuous improvement iterations were carried out on desktop software using said tool. This methodology was based on the metrics collected on usability and user sentiment, which allowed us to observe a positive evolution in all aspects evaluated.

**Keywords—** Software Quality; Usability; User Experience (UX); Sentiment Analysis.



## GLOSARIO

API: *Application Programming Interfaces* (interfaz de programación de aplicaciones )  
BPI: *Business Process integration* (Integración de procesos de negocio )  
CMMi: *Capability Maturity Model integration* (Integración de modelos de madurez de capacidades )  
CSUQ: *Computer System Usability Questionnaire* (Cuestionario de usabilidad del sistema informático)  
DCU: Diseño Centrado en Usuario  
DI: Departamento de Informática.  
ER: *Expectation Ratings* (Calificaciones de expectativas)  
EUCS: *End User Computer Satisfaction* (Satisfacción informática del usuario final)  
HCI: *Human-Computer Interaction*(Interacción Persona-Ordenador)  
IEC: *International Electrotechnical Commission* (Comisión Electrotécnica Internacional )  
IEEE: *Institute of Electrical and Electronics Engineers* (Instituto de Ingenieros Eléctricos y Electrónicos)  
ISO: *International Organization for Standardization* (Organización Internacional de Normalización)  
ISTQB: *International Software Testing Qualifications Board* (Comité Internacional de Certificaciones de Pruebas de Software)  
NLG: *Natural Language Generation* (Generación de lenguaje natural)  
NLP: *Natural Language Processing* (Procesamiento de lenguaje natural)  
NLU: *Natural Language Understanding* (Comprensión del lenguaje natural)  
ORM: *Object Relational Mapping* (Mapeo Objeto-Relacional)  
PSSUC: *Post-Study System Usability Questionnaire* (Cuestionario de usabilidad del sistema posterior al estudio)  
SDLC: *Systems Development Life Cycle* (Ciclo de vida del desarrollo de un sistema)  
SEQ: *Single Ease Question* (Única pregunta sencilla)  
SMEQ: *Subjective Mental Effort Questionnaire* (Cuestionario de esfuerzo mental subjetivo)  
SUS: *System Usability Scale* (Escala de usabilidad del sistema)  
TAM: *Technology Acceptance Model* (Modelo de aceptación de Tecnología)  
URL: *Uniform Resource Locator* (Localizador de Recursos Uniforme)  
UTFSM: Universidad Técnica Federico Santa María.

# ÍNDICE DE CONTENIDOS

|                                                                          |     |
|--------------------------------------------------------------------------|-----|
| RESUMEN . . . . .                                                        | IV  |
| ABSTRACT . . . . .                                                       | IV  |
| GLOSARIO . . . . .                                                       | VI  |
| ÍNDICE DE FIGURAS . . . . .                                              | X   |
| ÍNDICE DE TABLAS . . . . .                                               | XII |
| ÍNDICE DE CÓDIGOS . . . . .                                              | XIV |
| INTRODUCCIÓN . . . . .                                                   | 1   |
| CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA . . . . .                            | 2   |
| 1.1 Objetivos . . . . .                                                  | 5   |
| 1.1.1 Objetivo general . . . . .                                         | 5   |
| 1.1.2 Objetivos específicos . . . . .                                    | 5   |
| CAPÍTULO 2: MARCO CONCEPTUAL . . . . .                                   | 6   |
| 2.1 Calidad de software . . . . .                                        | 6   |
| 2.1.1 Definición . . . . .                                               | 6   |
| 2.1.2 Modelo de calidad de software . . . . .                            | 6   |
| 2.2 Madurez de software . . . . .                                        | 8   |
| 2.2.1 Definición . . . . .                                               | 8   |
| 2.2.2 Integración de procesos de negocio y madurez de software . . . . . | 9   |
| 2.2.3 Modelo de madurez de software . . . . .                            | 9   |
| 2.2.4 CMMi . . . . .                                                     | 10  |
| 2.3 Pruebas de software . . . . .                                        | 11  |
| 2.3.1 Definición . . . . .                                               | 11  |
| 2.3.2 Tipos de pruebas de software . . . . .                             | 12  |
| 2.4 Usabilidad . . . . .                                                 | 13  |
| 2.4.1 Definición . . . . .                                               | 13  |
| 2.4.2 Atributos de usabilidad . . . . .                                  | 14  |
| 2.4.3 Pruebas de usabilidad . . . . .                                    | 15  |
| 2.4.4 Cuestionarios estandarizados . . . . .                             | 16  |
| 2.4.5 Cuestionario CSUQ . . . . .                                        | 17  |
| 2.5 Metodologías de desarrollo de software . . . . .                     | 19  |
| 2.5.1 Metodologías tradicionales . . . . .                               | 19  |
| 2.5.2 Metodologías ágiles . . . . .                                      | 19  |
| 2.5.3 SCRUM . . . . .                                                    | 20  |

|                                                 |                                                                |    |
|-------------------------------------------------|----------------------------------------------------------------|----|
| 2.6                                             | Análisis de sentimiento . . . . .                              | 22 |
| 2.6.1                                           | Procesamiento de lenguaje natural . . . . .                    | 22 |
| 2.6.2                                           | ¿Qué es el análisis de sentimiento? . . . . .                  | 23 |
| 2.6.3                                           | Técnicas para el análisis de sentimientos . . . . .            | 23 |
| 2.6.4                                           | Librerías para el análisis de sentimientos . . . . .           | 25 |
| 2.7                                             | Experiencia de Usuario . . . . .                               | 26 |
| 2.7.1                                           | Definición de Experiencia de Usuario . . . . .                 | 26 |
| 2.7.2                                           | Relación con el comportamiento emocional del usuario . . . . . | 27 |
| 2.7.3                                           | Diseño centrado en usuario . . . . .                           | 28 |
| 2.7.4                                           | Principios de diseño . . . . .                                 | 29 |
| CAPÍTULO 3: PROPUESTA DE SOLUCIÓN . . . . .     |                                                                | 31 |
| 3.1                                             | Metodología propuesta . . . . .                                | 31 |
| 3.2                                             | Diseño e implementación plataforma Feel UX . . . . .           | 32 |
| 3.2.1                                           | Descripción general . . . . .                                  | 32 |
| 3.2.2                                           | Diseño e implementación Base de datos . . . . .                | 33 |
| 3.2.3                                           | Diseño e implementación Back-end . . . . .                     | 35 |
| 3.2.4                                           | Diseño e implementación Front-end . . . . .                    | 39 |
| 3.2.5                                           | Despliegue de aplicación . . . . .                             | 43 |
| 3.3                                             | Funcionalidades de Feel UX . . . . .                           | 45 |
| 3.3.1                                           | Perfiles y registro de usuarios . . . . .                      | 45 |
| 3.3.2                                           | Creación de Estudios, Iteraciones y Tareas . . . . .           | 47 |
| 3.3.3                                           | Activación de Iteraciones . . . . .                            | 52 |
| 3.3.4                                           | Ingreso a Iteración y Tareas . . . . .                         | 53 |
| 3.3.5                                           | Cuestionario CSUQ . . . . .                                    | 54 |
| 3.3.6                                           | Opinión de usuarios . . . . .                                  | 55 |
| 3.3.7                                           | Visualización de resultados . . . . .                          | 57 |
| 3.4                                             | Definición previa para Testing de Usabilidad . . . . .         | 60 |
| 3.4.1                                           | Contextualización de software testeado . . . . .               | 60 |
| 3.4.2                                           | Definición de usuario . . . . .                                | 61 |
| 3.4.3                                           | Definición de tareas . . . . .                                 | 61 |
| 3.5                                             | Primera iteración de mejora continua . . . . .                 | 64 |
| 3.5.1                                           | Resumen demográfico de participantes . . . . .                 | 64 |
| 3.5.2                                           | Primer rediseño basado en feedback y observación . . . . .     | 65 |
| 3.6                                             | Segunda iteración de mejora continua . . . . .                 | 69 |
| 3.6.1                                           | Resumen demográfico de participantes . . . . .                 | 69 |
| 3.6.2                                           | Segundo rediseño basado en feedback y observación . . . . .    | 70 |
| 3.7                                             | Tercera iteración de mejora continua . . . . .                 | 75 |
| 3.7.1                                           | Resumen demográfico de participantes . . . . .                 | 75 |
| 3.7.2                                           | Feedback de usuarios y mejoras pendientes . . . . .            | 76 |
| CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN . . . . . |                                                                | 78 |
| 4.1                                             | Tiempo y éxito de tareas . . . . .                             | 78 |
| 4.2                                             | Cuestionario CSUQ . . . . .                                    | 82 |

|                                      |                                                                 |     |
|--------------------------------------|-----------------------------------------------------------------|-----|
| 4.3                                  | Análisis de sentimiento . . . . .                               | 87  |
| CAPÍTULO 5: CONCLUSIONES . . . . .   |                                                                 | 94  |
| 5.1                                  | Conclusiones Generales . . . . .                                | 94  |
| 5.2                                  | Trabajos Futuros . . . . .                                      | 98  |
| 5.2.1                                | Mejoras y nuevas funcionalidades a plataforma Feel UX . . . . . | 98  |
| 5.2.2                                | Trabajo en conjunto con laboratorio UX de la UTFSM . . . . .    | 99  |
| ANEXOS . . . . .                     |                                                                 | 100 |
| 5.1                                  | Anexo A . . . . .                                               | 100 |
| 5.2                                  | Anexo B . . . . .                                               | 102 |
| 5.3                                  | Anexo C . . . . .                                               | 103 |
| 5.4                                  | Anexo D . . . . .                                               | 106 |
| REFERENCIAS BIBLIOGRÁFICAS . . . . . |                                                                 | 108 |

# ÍNDICE DE FIGURAS

|    |                                                                                  |    |
|----|----------------------------------------------------------------------------------|----|
| 1  | Árbol del problema. . . . .                                                      | 4  |
| 2  | Modelo de calidad del producto de software ISO/IEC 25010. . . . .                | 8  |
| 3  | Ilustración de los Niveles de Madurez CMMi . . . . .                             | 11 |
| 4  | Marco de definición de usabilidad en la ISO/IEC 9241-11. . . . .                 | 14 |
| 5  | Escala de calificación curva de Sauro-Lewis. . . . .                             | 18 |
| 6  | Diagrama resumen de la metodología Scrum. . . . .                                | 22 |
| 7  | Técnicas para la clasificación de sentimientos. . . . .                          | 24 |
| 8  | Factores involucrados en la Experiencia del Usuario . . . . .                    | 27 |
| 9  | Diseño metodológico de la propuesta de solución. . . . .                         | 32 |
| 10 | Modelo relacional base de datos Feel UX. . . . .                                 | 34 |
| 11 | Maqueta realizada en Canva para vista del cuestionario CSUQ en Feel UX. . . . .  | 40 |
| 12 | Página inicial de sitio web Feel UX. . . . .                                     | 45 |
| 13 | Formulario de registro en sitio web Feel UX. . . . .                             | 46 |
| 14 | Página de inicio para perfil Tester en Feel UX . . . . .                         | 47 |
| 15 | Página de inicio para perfil Testing User en Feel UX . . . . .                   | 47 |
| 16 | Creación de estudio en opciones rápidas de Sidebar. . . . .                      | 49 |
| 17 | Creación de estudio dentro de sección «Mis Estudios». . . . .                    | 49 |
| 18 | Creación de iteración en opciones rápidas de Sidebar. . . . .                    | 50 |
| 19 | Creación de iteración dentro de un estudio ya creado. . . . .                    | 50 |
| 20 | Creación de tarea en opciones rápidas de Sidebar. . . . .                        | 51 |
| 21 | Creación de tarea dentro de una iteración ya creada. . . . .                     | 51 |
| 22 | Botón que permite activar una iteración. . . . .                                 | 52 |
| 23 | Código único de la iteración, presente en la parte superior de la vista. . . . . | 53 |

|    |                                                                                          |    |
|----|------------------------------------------------------------------------------------------|----|
| 24 | Resumen de la iteración, con botón dispuesto para iniciar estudio. . . . .               | 53 |
| 25 | Interfaz que presenta las tareas a realizar en el estudio. . . . .                       | 54 |
| 26 | Interfaz que permite responder cuestionario CSUQ dentro del estudio. . . . .             | 55 |
| 27 | Interfaz que permite recopilar la opinión del usuario. . . . .                           | 56 |
| 28 | Interfaz que permite al usuario seleccionar el sentimiento respecto al software. . . . . | 56 |
| 29 | Dashboard general para una iteración particular. . . . .                                 | 58 |
| 30 | Dashboard demográfico para una iteración particular. . . . .                             | 58 |
| 31 | Dashboard de Usabilidad para una iteración particular. . . . .                           | 59 |
| 32 | Dashboard de Sentimiento para una iteración particular. . . . .                          | 59 |
| 33 | Vista principal de la primera versión de Desk Habitum. . . . .                           | 65 |
| 34 | Vista principal de la segunda versión de Desk Habitum. . . . .                           | 66 |
| 35 | Vista de dashboard, con múltiples secciones en Desk Habitum v1. . . . .                  | 67 |
| 36 | Vista de dashboard, con secciones resumidas en Desk Habitum v2. . . . .                  | 67 |
| 37 | Múltiples alertas de confirmación e información del estado del sistema. . . . .          | 68 |
| 38 | Vista de inicio con error en pantalla completa de Desk Habitum . . . . .                 | 70 |
| 39 | Vista de inicio con elementos adaptados a pantalla completa. . . . .                     | 71 |
| 40 | Comparativa de versiones de gráfico acumulado de última sesión. . . . .                  | 72 |
| 41 | Comparativa de versiones de gráfico de tendencia histórica de manías. . . . .            | 72 |
| 42 | Configuración de notificaciones con opción de radio button vacía. . . . .                | 73 |
| 43 | Formulario de contacto profesional sin validación de campos. . . . .                     | 73 |
| 44 | Configuración de notificaciones con opción por defecto y limitación de campos. . . . .   | 74 |
| 45 | Formulario de contacto profesional con validación de campos. . . . .                     | 74 |
| 46 | Gráfico de Tiempos promedio por tarea para cada iteración . . . . .                      | 79 |
| 47 | Gráfico de porcentaje de éxito promedio por tarea para cada iteración . . . . .          | 81 |

|    |                                                                                |     |
|----|--------------------------------------------------------------------------------|-----|
| 48 | Gráfico de indicadores CSUQ promedio para cada iteración . . . . .             | 83  |
| 49 | Gráfico de indicadores CSUQ promedio por rango etario en cada iteración. . . . | 85  |
| 50 | Gráfico de indicadores CSUQ por conocimiento tecnológico en cada iteración. .  | 86  |
| 51 | Gráfico con tendencia de score promedio, con y sin falsos negativos. . . . .   | 88  |
| 52 | Gráfico con tendencia de score promedio por rango etario. . . . .              | 89  |
| 53 | Gráfico con tendencia de score promedio por nivel de conocimiento tecnológico. | 90  |
| 54 | Gráfico de porcentajes por tipo de opinión en cada iteración . . . . .         | 90  |
| 55 | Distribución de las opiniones positivas en cada iteración. . . . .             | 91  |
| 56 | Palabras más utilizadas en opiniones de usuarios en primera iteración. . . . . | 92  |
| 57 | Palabras más utilizadas en opiniones de usuarios en segunda iteración. . . . . | 93  |
| 58 | Palabras más utilizadas en opiniones de usuarios en tercera iteración. . . . . | 93  |
| 59 | Cuestionario CSUQ en su versión en Español. . . . .                            | 102 |
| 60 | Vista de configuraciones en primera versión de Desk Habitum. . . . .           | 103 |
| 61 | Vista de configuraciones en segunda versión de Desk Habitum. . . . .           | 103 |
| 62 | Vista de contacto profesional en primera versión de Desk Habitum. . . . .      | 104 |
| 63 | Vista de contacto profesional en segunda versión de Desk Habitum. . . . .      | 104 |
| 64 | Vista de respiración profunda en primera versión de Desk Habitum. . . . .      | 105 |
| 65 | Vista de respiración profunda en segunda versión de Desk Habitum. . . . .      | 105 |
| 66 | Tooltip en botones de técnicas de respiración profunda. . . . .                | 106 |
| 67 | Comparativa de menú rápido, con la opción de contacto profesional agregada.    | 106 |
| 68 | Implementación de color a estado de monitoreo. . . . .                         | 107 |

## ÍNDICE DE TABLAS

|   |                                                                                |   |
|---|--------------------------------------------------------------------------------|---|
| 1 | Resumen de las principales herramientas disponibles para evaluar usabilidad. . | 3 |
|---|--------------------------------------------------------------------------------|---|

|    |                                                                                 |    |
|----|---------------------------------------------------------------------------------|----|
| 2  | Comparativa entre metodologías tradiciones y metodologías Ágiles. . . . .       | 20 |
| 3  | Detalle de tareas a realizar para evaluación de usabilidad de Desk Habitum. . . | 63 |
| 4  | Detalle de dificultad de las tareas en las distintas iteraciones. . . . .       | 63 |
| 5  | Detalle de duración óptima de las tareas en las distintas iteraciones. . . . .  | 64 |
| 6  | Detalle de cantidad de usuarios por conocimiento tecnológico en Iteración 1. .  | 64 |
| 7  | Detalle de cantidad de usuarios por rango etario en Iteración 1. . . . .        | 65 |
| 8  | Detalle de cantidad de usuarios por conocimiento tecnológico en Iteración 2. .  | 69 |
| 9  | Detalle de cantidad de usuarios por rango etario en Iteración 2. . . . .        | 70 |
| 10 | Detalle de cantidad de usuarios por conocimiento tecnológico en Iteración 3. .  | 76 |
| 11 | Detalle de cantidad de usuarios por rango etario en Iteración 3. . . . .        | 76 |
| 12 | Tiempo promedio [seg] por tarea para cada iteración . . . . .                   | 78 |
| 13 | Tiempo promedio [seg] por rango etario para cada iteración . . . . .            | 79 |
| 14 | Tiempo promedio [seg] por conocimiento tecnológico para cada iteración . . .    | 80 |
| 15 | Porcentaje de éxito promedio por tarea para cada iteración . . . . .            | 80 |
| 16 | Porcentaje de éxito promedio por rango etario para cada iteración . . . . .     | 80 |
| 17 | Porcentaje de éxito promedio por conocimiento tecnológico para cada iteración   | 81 |
| 18 | Indicadores promedio de cuestionario CSUQ por iteración. . . . .                | 82 |
| 19 | Puntaje SUS obtenido en cada iteración, con su equivalencia en grados. . . .    | 82 |
| 20 | Indicadores promedio de cuestionario CSUQ por rango etario en Iteración 1. .    | 83 |
| 21 | Indicadores promedio de cuestionario CSUQ por rango etario en Iteración 2. .    | 84 |
| 22 | Indicadores promedio de cuestionario CSUQ por rango etario en Iteración 3. .    | 84 |
| 23 | Indicadores cuestionario CSUQ por conocimiento tecnológico en Iteración 1. .    | 85 |
| 24 | Indicadores cuestionario CSUQ por conocimiento tecnológico en Iteración 2. .    | 85 |
| 25 | Indicadores cuestionario CSUQ por conocimiento tecnológico en Iteración 3. .    | 86 |

|    |                                                                               |     |
|----|-------------------------------------------------------------------------------|-----|
| 26 | Score promedio por iteración, sin considerar falsos negativos. . . . .        | 87  |
| 27 | Score promedio por iteración, considerando falsos negativos. . . . .          | 87  |
| 28 | Score promedio por rango etario en cada iteración. . . . .                    | 88  |
| 29 | Score promedio por nivel de conocimiento tecnológico en cada iteración. . . . | 89  |
| 30 | Cantidad de opiniones por tipo de sentimiento en cada iteración. . . . .      | 91  |
| 31 | Cuestionarios <i>post-task</i> con sus ventajas y desventajas. . . . .        | 100 |
| 32 | Cuestionarios <i>post-task</i> con sus ventajas y desventajas. . . . .        | 101 |

## Índice de Códigos

|    |                                                                                                                                      |    |
|----|--------------------------------------------------------------------------------------------------------------------------------------|----|
| 1  | Definición del modelo InfoTask en Sequelize. . . . .                                                                                 | 35 |
| 2  | Definición de relación entre InfoTask y Tasks en Sequelize. . . . .                                                                  | 35 |
| 3  | Consulta de tipo <i>Select</i> a través del método <i>findAll</i> de Sequelize. . . . .                                              | 36 |
| 4  | Función de JavaScript para calcular el porcentaje completado de tareas. . . . .                                                      | 36 |
| 5  | Función de JavaScript que hace una llamada a la API de Eden AI para análisis de sentimiento a través del motor de Microsoft. . . . . | 38 |
| 6  | Ejemplo de objeto que retorna la llamada a la API de Eden AI. . . . .                                                                | 38 |
| 7  | Ejemplo de utilización modulo <i>NormalizerEs</i> en JavaScript. . . . .                                                             | 38 |
| 8  | Utilización de módulo <i>SentimentManager</i> de la librería Node-nlp para la identificación de <i>keywords</i> . . . . .            | 39 |
| 9  | Utilización de librería Axios para consultas a back-end en React. . . . .                                                            | 41 |
| 10 | Utilización de componentes predefinidos de React-bootstrap en el componente <i>LogoutModal</i> . . . . .                             | 41 |
| 11 | Utilización de componentes predefinidos de Tremor en el componente <i>DashboardDemogrIteration</i> . . . . .                         | 42 |
| 12 | Utilización de componentes predefinidos de React ApexCharts en el componente <i>BoxPlotChart</i> . . . . .                           | 43 |

## INTRODUCCIÓN

En el contexto actual, donde cada día se desarrollan nuevos sistemas y aplicaciones, la experiencia del usuario (UX) y la usabilidad son elementos fundamentales que influyen directamente en la percepción y satisfacción de los usuarios finales. Con el creciente énfasis en la interacción digital, la calidad de la experiencia del usuario se ha convertido en un diferenciador clave entre las distintas soluciones tecnológicas, en el que los usuarios buscan, no solo funcionalidad pura, sino también sistemas intuitivos, eficientes y que les generen una experiencia positiva en cada interacción.

En este sentido, el sentimiento de los usuarios respecto a los nuevos sistemas emerge como un aspecto crítico. Los avances en inteligencia artificial ofrecen herramientas como el análisis de sentimiento, también conocido como minería de opinión, que permiten capturar y cuantificar las percepciones de los usuarios. Al aplicar estas técnicas a sus opiniones sobre el software, se obtiene una comprensión más profunda y cuantitativa de cómo se sienten los usuarios en relación con la tecnología que están utilizando.

El presente trabajo tiene como objetivo diseñar e implementar una solución de software, basada en análisis de sentimiento, que permita automatizar la toma, almacenamiento, procesamiento y visualización de datos en un contexto de testing de usabilidad de un software. Para validar la solución propuesta, se llevan a cabo tres iteraciones de mejora continua a una aplicación de escritorio, donde la minería de opinión aplicada al feedback de los usuarios desempeña un papel fundamental en la conducción de las mejoras propuestas para la aplicación testeada.

## CAPÍTULO 1

### DEFINICIÓN DEL PROBLEMA

En la actualidad, gracias a los avances de la informática, el software se encuentra en casi todos los campos de la actividad humana, como la industria, el comercio, las finanzas, la salud, entre otros. Y en muchos de estos campos la dependencia de las organizaciones respecto a este recurso es crítica, por lo que existe una creciente preocupación por lograr que los productos de software cumplan con ciertos criterios de calidad [Estayno *et al.*, 2009].

Por su parte, la IEEE Std 610 indica que «la calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario» [Echeverry *et al.*, 2008]. Es decir, habla tanto de satisfacer los requerimientos funcionales especificados en las distintas fases del desarrollo de software en un contexto ágil, como también se refiere al grado de conformidad con las expectativas que tiene el cliente o usuario frente al sistema. Por lo tanto, un software carente de calidad no solo se define como tal debido a la insatisfacción de los requerimientos funcionales, sino también a un sentimiento de no conformidad del usuario respecto al sistema.

Es por ello que uno de los criterios de calidad que va acorde a la segunda parte de la definición antes mencionada es el de **Usabilidad**, y se define en el estándar ISO 9241 como «el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso» [Bevan *et al.*, 2015]. En este sentido, la usabilidad se suele descomponer en 5 atributos básicos, los cuales son: Facilidad de aprendizaje, eficiencia, recuerdo en el tiempo, tasa de errores y satisfacción [Nielsen, 1993].

Y aunque es un criterio de calidad un tanto abstracto, es uno de los criterios más importantes a la hora de evaluar la conformidad del usuario con respecto al software. Según Carmen Carvajal [Carvajal Jiménez, 2018], la usabilidad es un atributo de calidad del software crítico que debería ser incorporado en las distintas aplicaciones de igual forma que el resto de los atributos de calidad tradicionales. Existe incluso el concepto de Ingeniería de Usabilidad, en el cual el autor propone aplicarlo en todo el SDLC para obtener un sistema que hace al usuario más productivo, aumentando su eficiencia y satisfacción al utilizarlo [Grau, 2000].

Sin embargo, Carvajal indica que aunque la usabilidad debe ser parte de la cultura organizacional y estar presente en toda la experiencia del usuario, en la mayoría de los casos se ha integrado en el proceso de desarrollo del software como un elemento opcional a la funcionalidad de los sistemas [Carvajal Jiménez, 2018], por lo que se infiere que esta es una de las principales causas del problema que se plantea en esta memoria: **los sistemas no cumplen con los criterios de calidad y madurez de software relacionados a la experiencia de usabilidad del usuario.**

Otra principal causa identificada es la falta de herramientas que ayuden a comprender el gra-

do de conformidad respecto al **sentimiento de los usuarios** con el sistema. En la propuesta de Mascheroni de un software para automatizar la evaluación de la usabilidad en productos software [Mascheroni *et al.*, 2012a], se realiza una revisión de algunas herramientas disponibles en la actualidad para la evaluación de usabilidad de un software. En la Tabla 1 se encuentra un resumen de las herramientas con su funcionalidad principal.

| Herramienta                             | Descripción                                                                                                                                                                                                                                                                                   |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Datalogger y SUM Calculator             | Plantillas Excel que permiten recoger y analizar datos o atributos de usabilidad establecidos previamente, que se utilizan para evaluar una aplicación determinada.                                                                                                                           |
| UserPlus, FiveSecondTest y See-Volution | Aplicaciones web gratuitas que utilizan un screenshot de la aplicación o sitio web a evaluar y una muestra de usuarios, para determinar click heatmaps (zonas donde el usuario hace la mayor cantidad de clics), scroll heatmaps (zonas que el usuario visita más) y opiniones de los mismos. |
| Morae y Noldus Observer                 | Aplicaciones de pago que permiten grabar las interacciones con usuarios (como por ejemplo registro de los gestos del usuario) para después analizarlas.                                                                                                                                       |

Tabla 1: Resumen de las principales herramientas disponibles para evaluar usabilidad.  
Fuente: Elaboración Propia.

Si bien es cierto que algunas de las herramientas proponen funcionalidades interesantes para la evaluación de la usabilidad de un software, **ninguna es capaz de establecer** y evaluar el grado de conformidad con respecto al **sentimiento de los usuarios con el software testeado**. Es por esta razón que uno de los objetivos de la presente memoria es diseñar e implementar un sistema automatizado para el desarrollo de pruebas de usabilidad de un software, utilizando cuestionarios estandarizados y aplicando técnicas de **minería de opinión** al feedback de los usuarios. Esto permitiría tener un acercamiento bastante preciso, a través de técnicas de inteligencia artificial, con respecto al sentimiento del usuario y su conformidad con los requisitos de usabilidad del sistema.

Otras causas del problema, señaladas en el árbol del problema de la Figura 1, tienen que ver con la poca comunicación que existe entre los usuarios del sistema y el equipo desarrollador, además de la falta de claridad que los usuarios muchas veces cometen al expresar sus necesidades. A esto también se le suma la tendencia de los usuarios a acostumbrarse a un tipo de software o sistema, lo que conlleva a un sentimiento de disconformidad con respecto a la usabilidad de cualquier sistema que sea distinto al que está acostumbrado.

Al hablar de las consecuencias del problema planteado, [Grau, 2000] señala que «la usabilidad es un tema crítico para la aceptación de un sistema: Si el sistema no es percibido como una herramienta que ayuda al usuario a realizar sus tareas, se dificulta la aceptación del sistema. Puede ocurrir que el sistema no llegue a usarse en absoluto, o que se use con escasa eficiencia». Por lo tanto, el sentimiento de no conformidad del usuario respecto a la expe-

riencia de usabilidad puede llevar a que algunas funcionalidades del sistema (o todas) que hayan sido implementadas no sean útiles para ellos, y por ende, no lleguen a utilizarse.

Algunas otras consecuencias directas de la problemática planteada son el sentimiento de malestar, frustración y/o disconformidad del usuario con el acceso a las funciones del sistema. Esto es algo lógico, ya que si un usuario prueba un sistema que, por ejemplo, no le permite acceder a las funcionalidades básicas de forma fácil, o la interfaz de usuario le parece poco atractiva, efectivamente va a expresar un sentimiento de malestar o de frustración.

Por eso es importante tener en cuenta el grado de satisfacción o conformidad del usuario respecto a su experiencia de usabilidad, y de esta forma alcanzar de forma satisfactoria los criterios de calidad y madurez de software en el sistema.

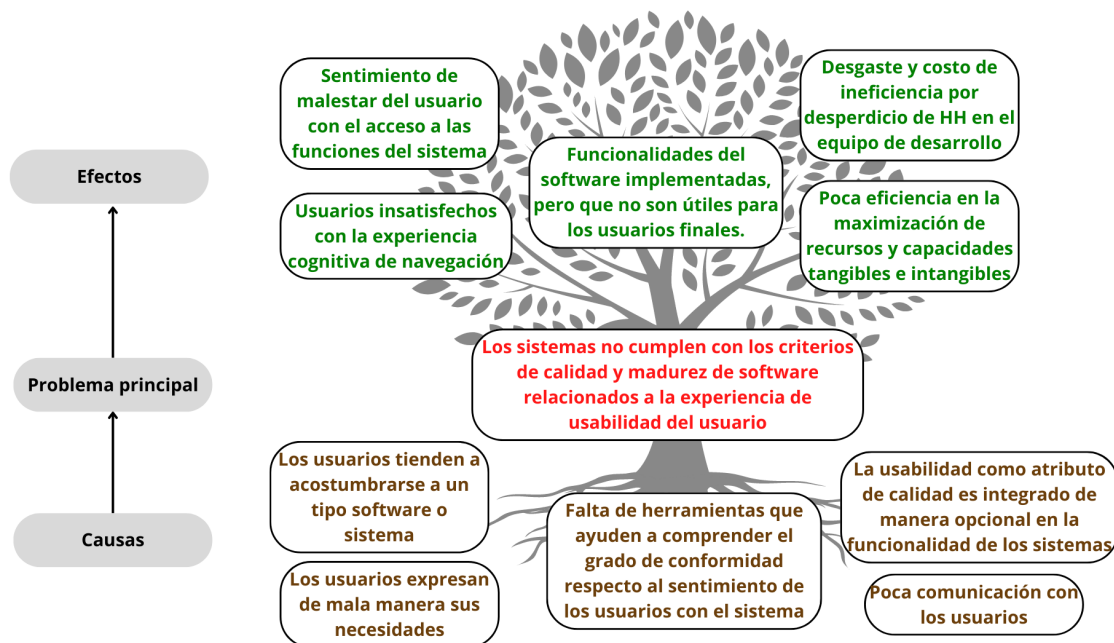


Figura 1: Árbol del problema.  
Fuente: Elaboración Propia.

## **1.1. Objetivos**

### **1.1.1. Objetivo general**

Diseñar e implementar una solución de software basada en técnicas de análisis de sentimiento para lograr conformidad en calidad y madurez de un producto final en entornos de desarrollo ágiles.

### **1.1.2. Objetivos específicos**

- Desarrollar una herramienta web automatizada para la aplicación de pruebas de usabilidad de un software, utilizando cuestionarios estandarizados y empleando técnicas de minería de opinión al feedback de los usuarios, con el fin de obtener información en forma de dashboard sobre la conformidad en calidad y madurez del sistema.
- Ejecutar ciclos de testing a una aplicación de escritorio (desarrollado en el contexto de la feria de software), y así poder medir la utilidad de la herramienta automatizada para este tipo de software.
- Proponer y realizar un rediseño continuo para la aplicación testeada, en base a las métricas obtenidas de la retroalimentación de los usuarios en el ciclo de iteraciones de testing.

## **CAPÍTULO 2**

### **MARCO CONCEPTUAL**

#### **2.1. Calidad de software**

##### **2.1.1. Definición**

El concepto de calidad, en el ámbito del desarrollo de software, puede ser considerado desde diversas aristas. Por un lado, se habla de calidad de software como la satisfacción de los requerimientos especificados en las etapas iniciales del desarrollo de la solución, pero también como el nivel de satisfacción de las expectativas del cliente. Por lo tanto, una de las definiciones que abarca ambos criterios es la establecida por la IEEE Std 610, que define calidad de software como «el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario» [Echeverry *et al.*, 2008].

Otra definición que aporta a lo señalado anteriormente es la del libro Ingeniería del Software de Pressman, en donde declara que calidad de software es la «concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario» [Pressman, 1993]. Es decir, si un producto de software es de calidad, no solamente debiese ser por su capacidad de satisfacer los requerimientos establecidos explícitamente, sino también porque el software cumple con lo que el usuario desea y espera de él, aún cuando el requerimiento no haya sido expresado de forma explícita por el cliente.

##### **2.1.2. Modelo de calidad de software**

Dada la definición de calidad de software en la sección anterior, la pregunta que puede surgir es la siguiente: ¿Cómo se puede medir la calidad de un producto de software?. Y para poder responder a esta pregunta, se recurrirá a la solución propuesta por el organismo internacional para la estandarización, la ISO.

El modelo de calidad del producto de software definido por la ISO/IEC es el 25010, y es fundamental para establecer el sistema de evaluación de la calidad del producto. En este se determinan las características de calidad relevantes para evaluar las propiedades de un software específico [ISO/IEC, 2010].

Actualmente el modelo cuenta con ocho características de calidad a ser evaluadas en un productos de software, las cuales son:

1. **Adecuación funcional:** Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas.
2. **Eficiencia de desempeño:** Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones.
3. **Compatibilidad:** Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software.
4. **Usabilidad:** Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones (se dedicará una sección completa para hablar sobre esta característica más adelante).
5. **Fiabilidad:** Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados.
6. **Seguridad:** Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.
7. **Mantenibilidad:** Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.
8. **Portabilidad:** Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro.

Cada una de estas categorías tiene a su vez subcategorías que la componen. No obstante, para no afectar la extensión de esta sección, y dado que no es el objetivo de esta investigación ahondar en el modelo de calidad, este documento se limita a presentar a modo de resumen dichas subcategorías en el diagrama de la figura 2.



Figura 2: Modelo de calidad del producto de software ISO/IEC 25010.

Fuente: [ISO/IEC, 2010].

## 2.2. Madurez de software

### 2.2.1. Definición

Aunque en la sección anterior se ha definido el concepto de calidad conforme al **producto** de software, es importante definir su relación con la calidad del **proceso** de software.

Esta relación se produce debido a que la calidad del producto va a estar en función de la calidad del proceso de desarrollo, por lo que los objetivos que se establezcan para la primera van a determinar los objetivos a establecer para la segunda. Es decir, sin un buen proceso de desarrollo de software es muy probable que el producto resultante no sea de calidad [Scalone, 2006].

En este sentido, la calidad de los procesos de desarrollo de software y la madurez de los mismos están directamente relacionados. [Chacón, 2004] indica que la madurez de un proceso de desarrollo de software es «el punto hasta el cual un determinado proceso es explícitamente definido, administrado, medido, controlado y efectivo. La madurez es el potencial de crecimiento de la capacidad del proceso, también nos indica tanto la riqueza del proceso de desarrollo de software de una organización, como la consistencia con la cual es aplicado en los proyectos a lo largo de la misma».

Por lo tanto, un proceso que presenta un alto grado de madurez implica una alta calidad en el proceso, dado que se encuentra definido, medido y controlado de forma correcta y completa, además de ser un proceso altamente efectivo y presentar gran consistencia, garantizando la calidad del proceso al aplicarse a distintos proyectos de software.

### **2.2.2. Integración de procesos de negocio y madurez de software**

La integración de procesos de negocio (BPI, *Business Process Integration*) es el proceso de conectar y coordinar los diferentes procesos de una organización para lograr una mayor eficiencia y eficacia en el logro de los objetivos comerciales. Esta integración puede involucrar la unificación de procesos de diferentes departamentos o áreas, o la integración de procesos de diferentes sistemas informáticos utilizados por la organización [Cflow, 2023].

En este sentido, la integración de procesos de negocio esta relacionada con la madurez del software en la medida en que los procesos de la organización estén automatizados y sean consistentes en su ejecución, lo cual se encuentra alineado con la definición de madurez de software establecida en la sección 2.2.1.

Por lo tanto, BPI puede contribuir a la madurez del software al permitir una mayor automatización de los procesos de desarrollo de software y al asegurar que los procesos sean consistentes en toda la organización [MuleSoft, 2020]. Por ejemplo, esta integración permitiría a la organización poder implementar herramientas útiles, como sistemas de control de versiones y herramientas de seguimiento de errores, mejorando la calidad y la eficiencia en el proceso de desarrollo de software.

Además, *Business Process Integration* ayuda a mejorar la madurez del software al permitir una mayor colaboración y comunicación entre los equipos de desarrollo de software y otros equipos de la organización [Henderson, 2020]. Al asegurar que los procesos estén estandarizados en toda la organización, la integración de procesos de negocio permite garantizar que los equipos de desarrollo de software estén trabajando en línea con los objetivos de la organización y que los cambios en los procesos sean comunicados y gestionados de manera efectiva.

### **2.2.3. Modelo de madurez de software**

En primer lugar, se debe comprender qué es un modelo de madurez. Según [Rosemann y vom Brocke, 2015], un modelo de madurez «es un mapa que guía a la organización en la implementación de buenas prácticas, ofreciendo un punto de partida». Además, según [Buhl *et al.*, 2011] «permite evaluar el estado de desarrollo de una organización o proceso de negocio, trazar claramente estrategias de mejoras para alcanzar los objetivos previstos e identificar las áreas donde la organización debe enfocarse para mejorar».

Entonces, un modelo de madurez de software es un esquema o una metodología utilizada para evaluar y mejorar la calidad y el desempeño de un software. Estos modelos, al igual que cualquier otro modelo de madurez, proporcionan una serie de criterios y niveles que se utilizan para determinar su madurez en los procesos que involucran el desarrollo del software.

Existen variados modelos de madurez de software, como Bootstrap, TickIt o CMMi. De este último se ahondará un poco más en la siguiente sección.

#### 2.2.4. CMMi

CMMi (por sus siglas en inglés, *Capability Maturity Model Integration*), es un modelo diseñado por el Instituto de Ingeniería de Software (SEI) para la mejora de procesos que proporciona a las organizaciones los elementos esenciales para procesos de desarrollo y mantenimiento de software [Echeverry *et al.*, 2008]. Este modelo de madurez es el estándar de calidad más utilizado a nivel internacional por las organizaciones desarrolladoras de software, aunque su uso no se limita solamente a este tipo de organizaciones [Pérez-Mergarejo *et al.*, 2014].

La estructura del CMMi se basa en cinco niveles de madurez, donde una organización puede evaluar su nivel de progreso y madurez en la producción de software en relación a los objetivos establecidos en cada nivel. Además, el modelo utiliza los niveles de madurez y capacidad para describir las rutas recomendadas que deben seguir las organizaciones para mejorar sus procesos.

Los cinco niveles de madurez que considera CMMi son [Echeverry *et al.*, 2008]:

1. **Inicial:** En este nivel, los procesos de la empresa se encuentran mal definidos y descontrolados, lo que lleva a una gran variación en la productividad y calidad de proyectos similares debido a la falta de estructura y control adecuado.
2. **Gestionado:** En este nivel, los procesos son intuitivos y los costos y calidad son altamente variables. Aun así, existe un cierto control en la planificación y se utilizan métodos o procedimientos informales que se realizan en el momento.
3. **Definido:** En este nivel, los procesos se han mejorado significativamente, se comienza a gestionar correctamente tanto los costos como la planificación dentro de unos límites razonables. Ahora se tiene un conocimiento detallado de la forma en que se construye el sistema, ya que se conocen las actividades intermedias y las entradas y salidas para cada una de ellas. Por lo tanto, es posible examinar y medir estas actividades, ya que los productos intermedios están bien definidos.
4. **Gestionado cuantitativamente:** A diferencia del nivel anterior, este cuarto nivel se caracteriza por ser cuantitativo. Ahora se tienen medidas detalladas del proceso del software y de la calidad del producto, lo que permite llevar a cabo un cierto control estadístico sobre la calidad del producto.

5. **Optimizado:** En este último nivel se busca la optimización continua, se utilizan datos cuantitativos para invertir continuamente en la automatización y mejora de procesos, con el fin de que la organización alcance el nivel más alto de madurez en sus procesos.

En la figura 3 se encuentra un resumen con los niveles de madurez del modelo CMMi mencionados anteriormente.

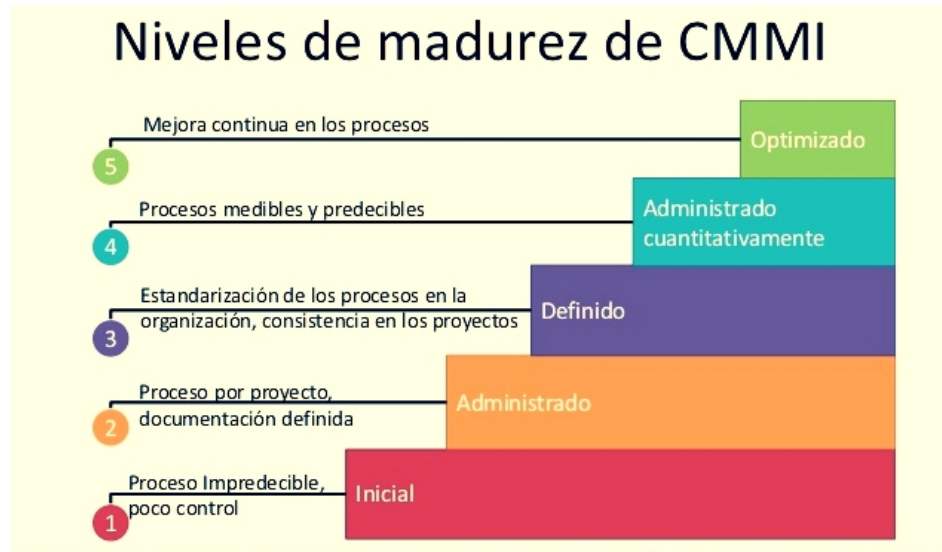


Figura 3: Ilustración de los Niveles de Madurez CMMi

Fuente: [Recuperado de <http://rupandcmmi.blogspot.com/p/niveles-de-madurez.html>].

## 2.3. Pruebas de software

### 2.3.1. Definición

El ISTQB (*International Software Testing Qualifications Board*) define las pruebas de software como «el proceso que consiste en todas las actividades del ciclo de vida, tanto estáticas como dinámicas, relacionadas con la planificación, preparación y evaluación de productos de software con el fin de determinar la satisfacción de los requisitos especificados, además de demostrar que son aptos para su propósito y también para detectar defectos» [Graham *et al.*, 2021].

Por su parte, Cem Kaner, profesor de Ingeniería de software en el instituto tecnológico de Florida, señala que «las pruebas de software son la investigación empírica y técnica realizada para facilitar a los interesados información sobre la calidad del producto o servicio bajo pruebas» [Kaner *et al.*, 1999].

Aunque ambas definiciones corresponden al mismo concepto, cada una se enfoca en distintos aspectos referentes al software y a la percepción que se tiene sobre él. En la definición del ISTQB, se especifican tres propósitos de las pruebas de software, que tienen que ver con la satisfacción de requisitos, demostración que el software es apto para su propósito y detección de defectos. Sin embargo, no se relaciona las pruebas de software con la calidad del mismo, como si lo hace la definición de Kaner, ya que identifica el proceso de pruebas de software como un aporte en la calidad del producto o servicio que esta siendo testado.

Además, cabe señalar que las pruebas de software son de vital importancia para cualquier proyecto de desarrollo, sobre todo en sistemas críticos donde una falla puede provocar peligro en la vida de las personas, ya que pueden identificar las fallas del sistema antes de estar en producción. Como señala Dijkstra: *«las pruebas de software pueden ser una manera muy eficaz de mostrar la presencia de errores, pero son totalmente inadecuadas para mostrar su ausencia»* [Dijkstra, 1972]

### 2.3.2. Tipos de pruebas de software

[Whittaker, 2000] propone utilizar la siguiente clasificación para las pruebas de software:

1. Pruebas Funcionales: se ocupa de verificar que el sistema y sus funcionalidades funcionen como se espera y para lo cual fueron desarrollados [Rodgers, 2022]. Algunos tipos de pruebas funcionales son:
  - Pruebas Unitarias.
  - Pruebas de Integración.
  - Pruebas de Regresión.
2. Pruebas No funcionales: se enfoca en factores transversales, basados en la operación de un software, no en la funcionalidad en sí [Rodgers, 2022]. Algunos tipos de pruebas no funcionales son:
  - Pruebas de Usabilidad.
  - Pruebas de Accesibilidad.
  - Pruebas de Seguridad.
  - Pruebas de Performance.
3. Pruebas Estructurales: Pueden realizarse en todos los niveles de prueba. Son las más idóneas, después de las técnicas basadas en la especificación, para ayudar a medir la exhaustividad de las pruebas mediante una evaluación de la cobertura de un tipo de estructura [Paz, 2016].

En la siguiente sección se profundizará sobre usabilidad, debido a que es uno de los principales tópicos a los cuales apunta la solución propuesta en esta memoria.

## 2.4. Usabilidad

### 2.4.1. Definición

El término usabilidad, que viene del inglés *Usability*, es un atributo cualitativo definido comúnmente como la facilidad de uso de un sistema informático, aunque en realidad puede ser aplicado a cualquier sistema que interactúe con un usuario. En el contexto de software, la ISO/IEC 9126 señala que «la usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso», haciendo énfasis en los atributos internos y externos del software. Es decir, la usabilidad depende tanto del sistema que está siendo probado como del usuario y el contexto en el que está siendo utilizado, por lo que un producto de software no puede ser intrínsecamente usable, sino que tendrá la capacidad de ser usado en un contexto particular y por usuarios particulares [Sanchez, 2015].

Otra definición que aporta a lo mencionado anteriormente es la establecida en la ISO/IEC 9241 [Bevan *et al.*, 2015], donde se señala que «Usabilidad es la eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico». Esta definición es aplicada al concepto de producto, y se centra en la calidad de uso del mismo, refiriéndose a la forma en que el usuario realiza tareas específicas en escenarios específicos de forma eficaz y efectiva [Sanchez, 2015]. La figura 4 resume el marco de definición de usabilidad analizado en este párrafo.

Al hablar de usabilidad aplicado al campo de la ingeniería de software, no se puede dejar de mencionar al padre de la usabilidad Jakob Nielsen, quien la define como «el atributo de calidad que mide lo fáciles de usar que son las interfaces Web» [Nielsen y Loranger, 2006]. Por lo tanto, según esta definición, un sitio web es considerado usable cuando los usuarios pueden interactuar con él de manera fácil, cómoda y segura. Además, es importante señalar que Jakob Nielsen se refiere a este atributo de calidad aplicado específicamente a las interfaces usuarias de los sistemas web, estableciendo así una conexión directa, pero no única, entre el nivel de usabilidad de un sistema y el diseño de la interfaz usuaria del mismo.

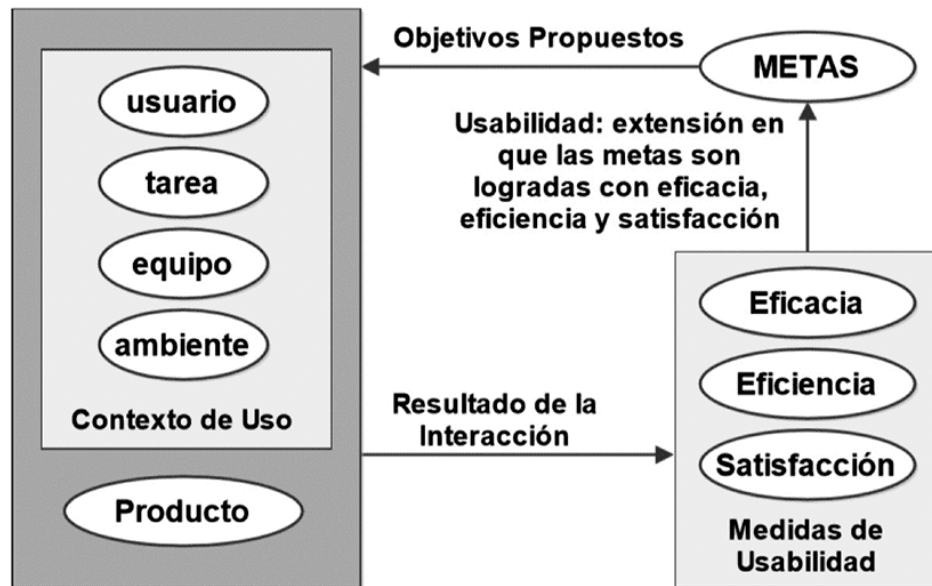


Figura 4: Marco de definición de usabilidad en la ISO/IEC 9241-11.  
Fuente: [ISO, 1998].

#### 2.4.2. Atributos de usabilidad

La usabilidad es una cualidad abstracta y no puede ser medida directamente. Por lo tanto, para poder medirla en un software, se utilizan cinco atributos básicos propuestos por Nielsen, los cuales son [Nielsen, 1993]:

1. **Facilidad de aprendizaje (*Learnability*):** Cuán fácil es aprender la funcionalidad básica del sistema, para que el usuario sea capaz de completar las tareas que desea realizar de forma satisfactoria. Es usualmente medido por el tiempo empleado por el usuario para completar ciertas tareas establecidas en un tiempo determinado (habitualmente, el tiempo empleado por un usuario experto).
2. **Eficiencia (*Efficiency*):** El número de transacciones por unidad de tiempo que el usuario puede realizar usando el sistema. Lo que se busca es la máxima velocidad de realización de tareas del usuario. Cuanto mayor es la usabilidad de un sistema, más rápido es el usuario al utilizarlo, y el trabajo se realiza con mayor rapidez. Se debe aclarar que eficiencia del software en cuanto a su velocidad de procesamiento no implica necesariamente eficiencia en la usabilidad.
3. **Recuerdo en el tiempo (*Memorability*):** Para usuarios que no utilizan el sistema regularmente es vital ser capaces de usar el sistema sin tener que aprender cómo funciona partiendo de cero cada vez. Entonces, este atributo refleja el recuerdo que mantiene el usuario sobre el funcionamiento del sistema cuando deja de utilizarlo después de un tiempo.

4. **Tasa de errores (*Errors*):** Se refiere al número de errores cometidos por el usuario mientras realiza una determinada tarea. Un buen nivel de usabilidad implica una tasa de errores baja, ya que los errores reducen la eficiencia y satisfacción del usuario, generando a su vez un sentimiento negativo de frustración en la persona por no saber como utilizar de forma correcta el software.
5. **Satisfacción (*Satisfaction*):** Éste es el atributo más subjetivo de todos. Muestra la impresión subjetiva que el usuario obtiene del sistema.

Es importante destacar que si un sistema tiene un alto nivel de usabilidad, no es debido a que cumple completamente todos estos atributos. Es más, algunos de ellos contribuyen a la usabilidad del sistema en direcciones contrarias, pudiendo ocurrir que el aumento de uno de ellos tenga como efecto la disminución de otro [Grau, 2000]. Por ejemplo, si se diseña un software de tal forma que su facilidad de aprendizaje sea la más alta posible, es probable que la eficiencia en las tareas se vea perjudicada, en pos de que el usuario sea capaz de aprender a utilizar el sistema de la manera más sencilla posible por sobre el tiempo que le demore realizar la tarea.

#### 2.4.3. Pruebas de usabilidad

Los atributos de usabilidad mencionados en la sección 2.4.2 pueden ser medidos a través de las denominadas **Pruebas de usabilidad**. Bajo un enfoque tradicional, las pruebas o test de usabilidad se aplican sobre el software para determinar si el mismo alcanza un nivel aceptable de usabilidad [Mascheroni *et al.*, 2012b]. Según [Grau, 2000], las pruebas de usabilidad son la práctica de usabilidad más extendida, y consisten en presentar al usuario una serie de tareas, y pedirle que las realice con el prototipo del sistema. Además, la motivación de llevar a cabo estas pruebas se basa en que no es posible conocer el nivel de usabilidad de un sistema si no se prueba con usuarios reales.

En general, existen variadas técnicas para realizar un test de usabilidad, por lo que a continuación se señalan las que se utilizan dependiendo del atributo de calidad que se quiera medir [Mascheroni *et al.*, 2012b]:

1. Facilidad de aprendizaje:
  - Sesiones guiadas con usuarios
  - Métodos de seguimientos
  - Protocolo de pensamiento manifestado (pensar en voz alta)
  - Inspecciones de expertos
2. Eficiencia:

- Inspecciones con expertos
  - Test exploratorios
  - Sesiones guiadas con usuarios
3. Recuerdo en el tiempo:
- Sesiones guiadas con usuarios (separadas en el tiempo)
  - Test de validación
4. Tasa/Manejo de errores:
- Inspecciones con expertos
  - Test de validación
  - Sesiones guiadas con usuarios
5. Satisfacción:
- Cuestionarios
  - Encuestas
  - Entrevistas

Tal como se puede observar en la lista anterior, las distintas técnicas se resumen en sesiones guiadas con usuarios, cuestionarios (ya sea estandarizados o no) y expertos en usabilidad que evalúen el sistema. Para la construcción del software propuesto en esta memoria, se implementarán las dos primeras técnicas mencionadas. La forma en la que se llevarán a cabo las sesiones de pruebas con los usuarios se describirán en capítulos posteriores, mientras que la definición y descripción de los cuestionarios estandarizados se describirán en la sección 2.4.4.

#### **2.4.4. Cuestionarios estandarizados**

Un cuestionario estandarizado es un cuestionario escrito y administrado de modo que a todos los participantes se les hagan exactamente las mismas preguntas en un formato idéntico, y las respuestas se registren y puntúen de manera específica y consistente [Boynton y Greenhalgh, 2004].

Los cuestionarios estandarizados son una de las herramientas mas utilizadas para evaluar aspectos de usabilidad en sistemas informáticos. [Sagar y Saha, 2017] identificaron que los cuestionarios estandarizados se utilizan con frecuencia por los equipos desarrolladores, y definieron la siguiente clasificación:

- **Cuestionarios post-estudio (*post-study*):** Se presentan y responden una vez al final del estudio.
- **Cuestionarios post-tarea (*post-task*):** Se presentan y responden inmediatamente después de una tarea o escenario.

Cabe mencionar que los cuestionarios posteriores a la tarea y posteriores al estudio no son excluyentes, ya que ambos pueden ser utilizados en un mismo estudio de usabilidad de ser necesario. [Hodrien *et al.*, 2021], en su revisión de cuestionarios subjetivos posteriores a la tarea y al estudio, realizan una comparativa entre todos los cuestionarios disponibles en la actualidad, indicando para cada uno de los cuestionarios las ventajas y desventajas, los aspectos de usabilidad cubiertos, y estableció la calidad psicométrica en base a parámetros como la fiabilidad, validez y sensibilidad.

En las tablas 31 y 32, presentes en el Anexo A (sección 5.1), se encuentra una selección de los mejores cuestionarios para cada tipo de clasificación, considerando su calidad en los distintos parámetros usados por el autor, además de tener una licencia *free use* y así poder ser utilizado en la propuesta de solución de esta memoria.

#### 2.4.5. Cuestionario CSUQ

El cuestionario CSUQ (*Computer System Usability Questionnaire*) es un cuestionario estandarizado y validado que se utiliza para evaluar la usabilidad de sistemas informáticos. Creado gracias a un proyecto interno de IBM, este cuestionario surgió a partir de otro denominado PSSUQ (*Post-Study System Usability Questionnaire*), con el fin de obtener la opinión de los usuarios acerca de la facilidad de uso y satisfacción con un sistema informático en particular [Hedlefs *et al.*, 2015].

El CSUQ cuenta con 16 preguntas (aunque existen otras versiones previas que cuentan con 19 preguntas), subdivididas en 4 dimensiones principales de usabilidad: satisfacción general, calidad de la información presentada, calidad de la interfaz de usuario y calidad del sistema en general. Cada una de estas preguntas están formuladas de manera que los encuestados puedan expresar su nivel de acuerdo o desacuerdo con afirmaciones específicas sobre la usabilidad del sistema que están evaluando, a través del uso de una escala Likert de 7 puntos, donde 1 indica «totalmente en desacuerdo» y 7 indica «totalmente de acuerdo».

Originalmente, el CSUQ es un cuestionario en idioma inglés. Sin embargo, para esta memoria se utilizará la versión en español incluida en el anexo 5.2.

La forma en la que se obtiene la puntuación del cuestionario es la siguiente [Lewis, 1995]:

- **General:** Promedio de los items 1 al 16.

- **Utilidad del sistema (*SysUse*):** Promedio de los ítems 1 al 6.
- **Calidad de la información (*InfoQual*):** Promedio de los ítems 7 al 12.
- **Calidad de la interfaz (*IntQual*):** Promedio de los ítems 13 al 15.

También es posible obtener una equivalencia del cuestionario CSUQ con la escala SUS. La Escala de Usabilidad del Sistema, o SUS, es una medida ampliamente utilizada para medir la usabilidad general de un software o sistema informático. Fue creado por John Brooke en el año 1986, y es una de las escalas más utilizadas en la investigación UX [Sagar y Saha, 2017]. La fórmula para calcular el puntaje SUS al cuestionario CSUQ viene dada por [Lewis, 2018]:

$$SUS = \left( \frac{\sum_{i=1}^{16} CSUQ_i}{16} - 1 \right) * \left( \frac{100}{6} \right)$$

La conversión del puntaje CSUQ a la escala SUS permite utilizar la escala de calificación curva propuesta por [Sauro y Lewis, 2016], la cual indica el rango de puntuación que corresponde a cada valoración y su respectivo percentil (ver figura 5). En este estudio, que abarcó más de 500 sitios web y aplicaciones, se concluyó que el puntaje promedio en el CSUQ es de 68, y que un resultado por debajo de este valor sugiere la necesidad de corregir varios aspectos en la aplicación testeada.

| SUS score range | Grade | Percentile range |
|-----------------|-------|------------------|
| 84.1–100        | A+    | 96–100           |
| 80.8–84.0       | A     | 90–95            |
| 78.9–80.7       | A–    | 85–89            |
| 77.2–78.8       | B+    | 80–84            |
| 74.1–77.1       | B     | 70–79            |
| 72.6–74.0       | B–    | 65–69            |
| 71.1–72.5       | C+    | 60–64            |
| 65.0–71.0       | C     | 41–59            |
| 62.7–64.9       | C–    | 35–40            |
| 51.7–62.6       | D     | 15–34            |
| 0.0–51.6        | F     | 0–14             |

Figura 5: Escala de calificación curva de Sauro-Lewis.  
Fuente: [Sauro y Lewis, 2016].

## 2.5. Metodologías de desarrollo de software

### 2.5.1. Metodologías tradicionales

Las metodologías tradicionales, denominadas en algunas ocasiones como metodologías pesadas, centran su atención en llevar una documentación exhaustiva de todo el proyecto, la planificación y control del mismo, en especificaciones precisas de requisitos y modelado, y en cumplir con un plan de trabajo, definido en la fase inicial del desarrollo del proyecto [Maida y Pacienza, 2015].

Este tipo de metodologías de desarrollo de software se caracterizan por imponer una disciplina rigurosa en el proceso de desarrollo, haciendo hincapié en la planificación completa de todo el trabajo a realizar antes de comenzar el ciclo de desarrollo del producto. Además, estas metodologías se enfocan en el control del proceso a través de la definición rigurosa de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada.

Algunos ejemplos de metodologías de desarrollo tradicionales son: Modelo en cascada, Modelo en espiral y Modelo en V. Sin embargo, el principal problema de las metodologías tradicionales es que no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno donde los requisitos no se pueden predecir o bien pueden variar. Es debido a estos que surgen las metodologías de desarrollo de software ágiles.

### 2.5.2. Metodologías ágiles

En la década de los noventa surgieron metodologías de desarrollo de software ligeras, más adelante nombradas como metodologías ágiles, que buscaban reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en los proyectos de desarrollo de software [Cadavid *et al.*, 2013].

Este tipo de metodologías se caracterizan por ser flexibles y pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto, siendo esta una de las principales diferencias con respecto a las metodologías tradicionales. Algunas de las características principales de la metodología de desarrollo de software ágil son:

- Los proyectos ágiles se subdividen en proyectos más pequeños mediante una lista ordenada de características, tratados de forma independiente desarrollándolos en periodos cortos de tiempo, de entre dos a seis semanas.
- La comunicación con el cliente del producto de software es fundamental y constante.
- Los proyectos son altamente colaborativos y se adaptan mejor a los cambios

- Entregas constantes al cliente y la retroalimentación por parte de él, para una mejora continua del producto.

En la tabla 2 se presenta una comparativa entre las características de las metodologías tradicionales y ágiles.

| Metodologías tradicionales                     | Metodologías ágiles                                         |
|------------------------------------------------|-------------------------------------------------------------|
| Predictivos                                    | Adaptativos                                                 |
| Orientados a procesos                          | Orientados a personas                                       |
| Proceso rígido                                 | Proceso flexible                                            |
| Se concibe como un proyecto                    | Un proyecto es subdividido en varios proyectos más pequeños |
| Poca comunicación con el cliente               | Comunicación constante con el cliente                       |
| Entrega de software al finalizar el desarrollo | Entregas constantes de software                             |
| Documentación extensa                          | Poca documentación                                          |

Tabla 2: Comparativa entre metodologías tradiciones y metodologías Ágiles.

Fuente: [Cadavid *et al.*, 2013]

Algunos ejemplos de metodologías de desarrollo ágiles son SCRUM, XP (Programación Extrema), Kanban y Lean. En la siguiente sección se profundizará sobre SCRUM, una de las metodologías ágiles más utilizadas en la actualidad para el desarrollo de software.

### 2.5.3. SCRUM

La metodología SCRUM es un marco de trabajo iterativo e incremental para el desarrollo de proyectos, productos y aplicaciones. Se basa en el trabajo en equipo con un enfoque ágil, y tiene como objetivo el control continuo sobre el estado actual del software, en el cual el cliente establece las prioridades y el equipo Scrum se auto-organiza para determinar la mejor forma de entregar resultados [Deemer *et al.*, 2009].

Se originó en la década de 1980, fuertemente influenciada por la publicación de Takeuchi y Nonaka en la *Harvard Business Review* del año 1986 [Takeuchi y Nonaka, 1986], y fue desarrollada en la década de 1990 por Jeff Sutherland y Ken Schwaber, quienes basaron la metodología en sus experiencias trabajando en proyectos de software [Sutherland y Schwaber, 2007].

Para que un equipo pueda desarrollar de manera correcta un proyecto bajo la metodología Scrum, debe considerar los siguientes roles [Schwaber y Sutherland, 2013]:

1. **Desarrolladores:** son las personas del equipo que se comprometen a crear cualquier aspecto de un incremento utilizable en cada Sprint.
2. **Scrum Master:** es responsable de hacer cumplir las directrices del marco de trabajo Scrum dentro del equipo. Lo hacen ayudando a todos a comprender la teoría y la práctica de Scrum, tanto dentro del equipo como en la organización.
3. **Product Owner:** es responsable de maximizar el valor del producto resultante del trabajo del equipo.

A su vez, Scrum define algunos **eventos** que el equipo deberá cumplir en el desarrollo del producto o software [Schwaber y Sutherland, 2013]:

1. **Sprint:** es el evento más importante en Scrum, ya que es donde las ideas se convierten en valor a través de la definición de tareas específicas y objetivos. Son eventos con duración de dos a cuatro semanas con el fin de crear consistencia en el trabajo grupal, y comienzan justo después de haber finalizado el anterior sprint. Además, contiene a los demás eventos definidos en Scrum.
2. **Sprint Planning:** este evento inicia el Sprint, y consiste en el diseño del trabajo que se realizará para el Sprint. Este plan resultante es creado por el trabajo colaborativo de todo el equipo Scrum.
3. **Daily Scrum:** es un evento diario de 15 minutos para los Desarrolladores del Equipo Scrum, y tiene como objetivo estar al tanto del progreso general del equipo.
4. **Sprint Review:** el propósito de este evento es inspeccionar el resultado del Sprint y determinar futuras adaptaciones. El equipo Scrum presenta los resultados de su trabajo a los *stakeholders* y se analiza el progreso hacia el objetivo del producto.
5. **Sprint Retrospective:** este es el último evento del sprint, en donde el equipo Scrum inspecciona cómo fue el último Sprint con respecto a las personas, las interacciones, los procesos, las herramientas y lo que cada uno considera como trabajo terminado, con el propósito de planificar formas de aumentar la calidad y la eficacia.

Por último, cada proyecto desarrollado bajo la metodología Scrum debe contar con los siguientes **artefactos** [Schwaber y Sutherland, 2013]:

1. **Product Backlog:** es una lista emergente y ordenada de las actividades que se necesitan realizar para lograr el objetivo del proyecto. Es la única fuente de trabajo y es realizada por todo el equipo Scrum.
2. **Sprint Backlog:** se compone del objetivo del Sprint (por qué), el conjunto de elementos del *Product Backlog* seleccionados para el Sprint (qué), así como un plan de implementación para entregar el avance del proyecto (cómo).

En la figura 6 se encuentra un resumen de los conceptos mencionados anteriormente.

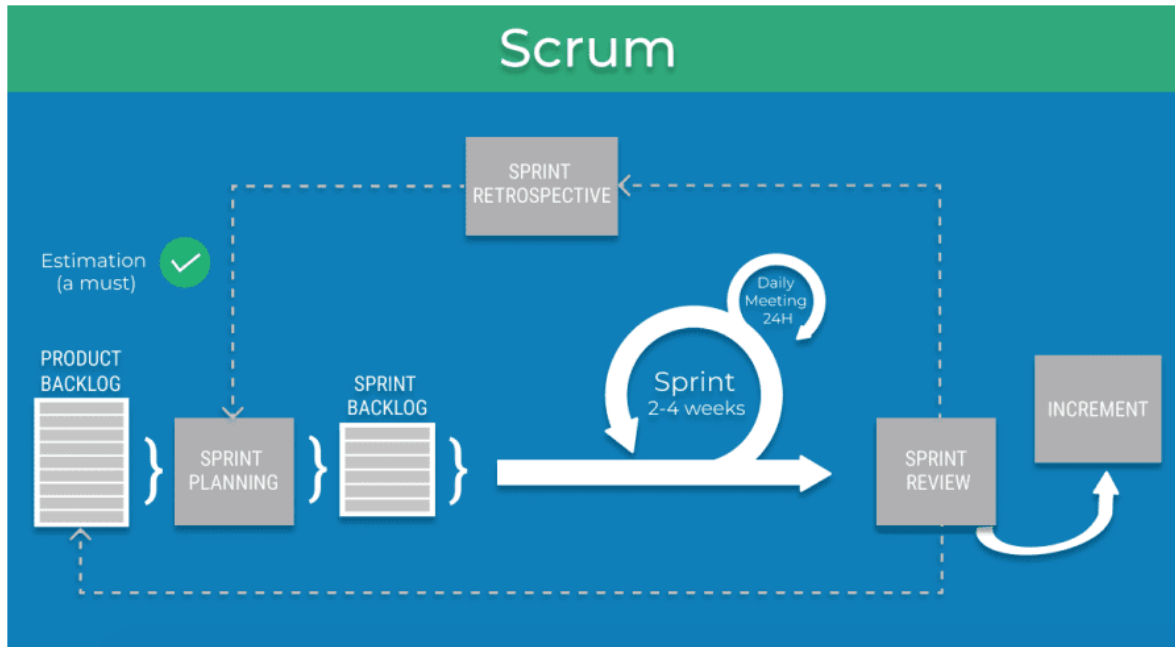


Figura 6: Diagrama resumen de la metodología Scrum.

Fuente: [Schwaber y Sutherland, 2011].

## 2.6. Análisis de sentimiento

### 2.6.1. Procesamiento de lenguaje natural

Para poder entender a que se refiere el análisis de sentimiento, primero hay que tener una idea sobre qué es el procesamiento de lenguaje natural en computación.

El NLP (por sus siglas en inglés, procesamiento de lenguaje natural) es una rama de la inteligencia artificial que a través de algoritmos de aprendizaje automático permite que una máquina analice e interprete el lenguaje humano [Zeroual y Lakhouaja, 2018]. Se puede subdividir en dos grandes grupos:

- **NLU** (Comprensión del lenguaje natural): su fin es interpretar la intención de la persona o usuario y comprender el significado de lo que desea expresar.
- **NLG** (Generación de lenguaje natural): tiene como objetivo modificar los datos estructurados que están almacenados en la máquina y transformarlos en lenguaje natural que sea conocido por la persona o usuario.

El procesamiento de lenguaje natural tiene diversas aplicaciones, y en la actualidad se utilizan para la traducción automática, recuperación y extracción de información, generación de resúmenes, resolución cooperativa de problemas, tutores inteligentes, reconocimiento de voz, entre otras. En esta memoria nos enfocaremos en la comprensión del lenguaje natural, a través de la técnica de análisis de sentimiento de textos.

### 2.6.2. ¿Qué es el análisis de sentimiento?

El análisis de sentimiento, también conocido como minería de opinión, hace referencia al uso de procesamiento lenguaje natural, lingüística computacional y análisis de texto (rama específica de la minería de datos) para identificar y extraer información subjetiva de los recursos [Feldman *et al.*, 2007]. Gracias al análisis de sentimientos es posible obtener un valor tangible, comparable y directo sobre el sentimiento del texto, y así poder determinar si tiene una connotación positiva o negativa.

Según describe [Mejova, 2009], el análisis de sentimiento involucra tres tareas fundamentales, las cuales son:

1. **Grado de subjetividad:** Esta tarea tiene que ver con detectar si el texto analizado se puede clasificar como objetivo o subjetivo.
2. **Polaridad:** Esta tarea consiste en clasificar el texto en dos extremos: positivo y negativo. Es decir, que tan positiva o que tan negativa es la opinión expresada en el texto. Usualmente se utiliza un rango continuo de -1 a 1 para expresar la polaridad.
3. **Objetivo de la opinión:** Es una tarea complementaria a la clasificación del sentimiento, y tiene que ver con descubrir sobre qué está hablando el texto. A menudo se utiliza la extracción de características, donde se identifican las palabras claves que determinan el objetivo o sentido del texto.

Actualmente las empresas están implementado análisis de sentimiento en comentarios a través de encuestas o redes sociales, con el fin de posicionar su marca y poder gestionar de mejor manera su reputación en plataformas públicas, además de realizar un marketing personalizado adaptándose al sentimiento que tienen los usuarios acerca de productos o servicios, o también para pronosticar ventas y poder crear una estrategia futura [Parra, 2019].

### 2.6.3. Técnicas para el análisis de sentimientos

[Medhat *et al.*, 2014] presenta en su estudio una categorización detallada de las técnicas de análisis de sentimiento. Esta sección se centrará en el enfoque de aprendizaje automático, en especial sobre el aprendizaje supervisado, ya que gran parte de las librerías actuales utilizan

alguna de estas técnicas de análisis de sentimiento para su implementación. Sin embargo, en la figura 7 se encuentra un resumen de todas las técnicas utilizadas para la clasificación de sentimiento.

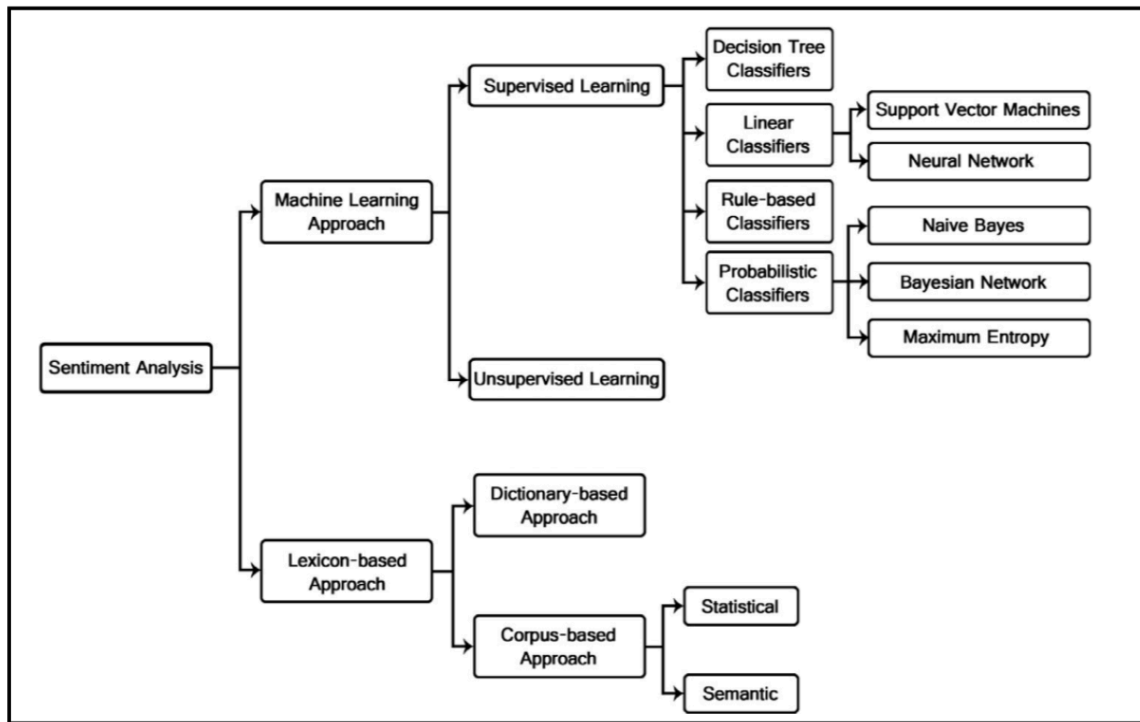


Figura 7: Técnicas para la clasificación de sentimientos.

Fuente: [Medhat *et al.*, 2014].

El enfoque de **machine learning** se subdivide en dos: técnicas supervisadas y no supervisadas. Ambas técnicas se basan principalmente en la selección y extracción de un conjunto apropiado de características utilizadas para detectar sentimientos, por lo que las técnicas de NLP son fundamentales para el éxito de esta tarea.

En cuanto a las técnicas de **aprendizaje supervisado**, estas utilizan algoritmos para encontrar una función que asigne etiquetas correctas a los datos de entrada con base en un historial de datos etiquetados, permitiendo así que el algoritmo haga predicciones sobre nuevos valores.

Algunos de los clasificadores de aprendizaje supervisado más utilizados para el análisis de sentimiento son:

- **Naive Bayes:** es un algoritmo de clasificación probabilístico basado en el teorema de Bayes. Se utiliza para clasificar objetos en una de varias categorías basándose en características específicas de esos objetos.
- **Modelo de máxima entropía:** es un método de clasificación discriminatorio donde los documentos del conjunto de datos son descritos a partir de una lista de atributos,

siendo cada uno, una restricción del modelo. Este método se basa en seleccionar la distribución de probabilidad que satisfaga todas las restricciones del modelo y maximice la entropía.

- **Arboles de decisión:** es un algoritmo de aprendizaje automático utilizado para clasificar y predecir la pertenencia a una clase o valor continuo. Cada rama del árbol representa una decisión basada en una característica específica, y cada hoja representa una clase o valor predicho.

En la siguiente sección se presentan algunas de las librerías más utilizadas para el análisis de sentimiento.

#### 2.6.4. Librerías para el análisis de sentimientos

Existen diversas librerías en distintos lenguajes de programación para poder realizar análisis de sentimiento a un texto.

A continuación se presentan algunas de las librerías más populares para lenguaje de programación **Python**:

- **Natural Language Toolkit (NLTK):** es una biblioteca de procesamiento del lenguaje natural que ofrece herramientas, como *Vader*, para el análisis de sentimiento. Posee aproximadamente 7520 elementos y calcula el sentimiento a partir de reglas creadas con anterioridad, por ejemplo, contar la cantidad de apariciones de una determinada palabra o analizar la estructura de la oración [Hutto y Gilbert, 2014].
- **TextBlob:** es una librería que utiliza un algoritmo de clasificación de sentimiento basado en una lista de palabras previamente etiquetadas como positivas o negativas. Este algoritmo se basa en la idea de que las palabras que aparecen con mayor frecuencia en el texto proporcionan una buena indicación del sentimiento general del texto. Cabe señalar que se encuentra disponible *NaiveBayesAnalyzer*, un módulo dentro de la biblioteca TextBlob que proporciona una interfaz para un análisis de sentimiento basado en el algoritmo Naive Bayes [Viteri, 2021].
- **BERT:** es un modelo de lenguaje pre-entrenado desarrollado por Google que se utiliza comúnmente para tareas de procesamiento de lenguaje natural, como el análisis de sentimiento. Para utilizar BERT como herramienta de análisis de sentimiento, se puede utilizar una versión de Keras de BERT (como Keras-BERT) o se puede utilizar el modelo pre-entrenado de BERT y adaptarlo a una tarea de análisis de sentimiento utilizando Keras [Collarte González, 2021].

También existen librerías para **Javascript**, uno de los lenguajes más utilizados para el desarrollo web:

- **Sentiment**: es un módulo de Node.js que utiliza la lista de palabras AFINN-165 y la clasificación de sentimiento emoji para realizar análisis de sentimiento en bloques arbitrarios de texto de entrada [Sliwinski, 2019].
- **Node-nlp**: esta librería está diseñada específicamente para Node.js. Ofrece soporte nativo para 41 idiomas y, con la ayuda de los embeddings de Bert, puede ser extendida a un total de 104 idiomas. Se utiliza principalmente para tareas como la construcción de bots, el análisis de sentimiento y la identificación automática del idioma, entre otras funciones de procesamiento de lenguaje natural [datos.gob.es, 2022].
- **Wink**: Wink es una familia de paquetes de código abierto para el análisis estadístico, el procesamiento del lenguaje natural y el aprendizaje automático en NodeJS. Los paquetes de Wink han sido optimizados para lograr un equilibrio entre rendimiento y precisión, lo que permite manejar grandes cantidades de texto en bruto a una alta velocidad [datos.gob.es, 2022].

Cada una de estas librerías posee ventajas y desventajas, y la elección de alguna de estas puede depender de varios factores, como el idioma, el contexto en el cual las palabras están presentes y los objetivos específicos del proyecto.

## 2.7. Experiencia de Usuario

### 2.7.1. Definición de Experiencia de Usuario

La Experiencia de Usuario (UX, *User eXperience*) es un concepto que está inmerso dentro del área de estudio dedicada a la Interacción Persona-Ordenador (HCI, *Human-Computer Interaction*), la cual se centra en el fenómeno de interacción entre usuarios y sistemas informáticos con el fin de proporcionar las bases teóricas, metodológicas y prácticas necesarias para el diseño y evaluación de productos interactivos que sean fáciles de usar, eficientes, seguros y satisfactorios.

[Arhippainen y Tähti, 2003] definen experiencia de usuario como «la experiencia que obtiene el usuario cuando interactúa con un producto en condiciones particulares». De forma más específica, [Hassan Montero y Martín Fernández, 2005] la definen como «la sensación, sentimiento, respuesta emocional, valoración y satisfacción del usuario respecto a un producto, resultado del fenómeno de interacción con el producto y la interacción con su proveedor».

El *User Experience*, que tiene su origen en Marketing vinculado al concepto de Experiencia de Marca, surge debido a las limitaciones de los enfoques tradicionales para el diseño de productos interactivos, que tienden a ser insuficientes debido a que no tienen en cuenta factores importantes como las emociones del usuario.

[Arhippainen y Tähti, 2003] presenta un modelo que descompone la Experiencia del Usuario en distintas variables que la afectan y condicionan. Estas variables se clasifican en cinco grandes grupos, los cuales son: factores propios del usuario, factores sociales, factores culturales, factores del contexto de uso y factores propios del producto. En la figura 8 se presenta el modelo con las distintas variables agrupadas en cada uno de los factores mencionados anteriormente.

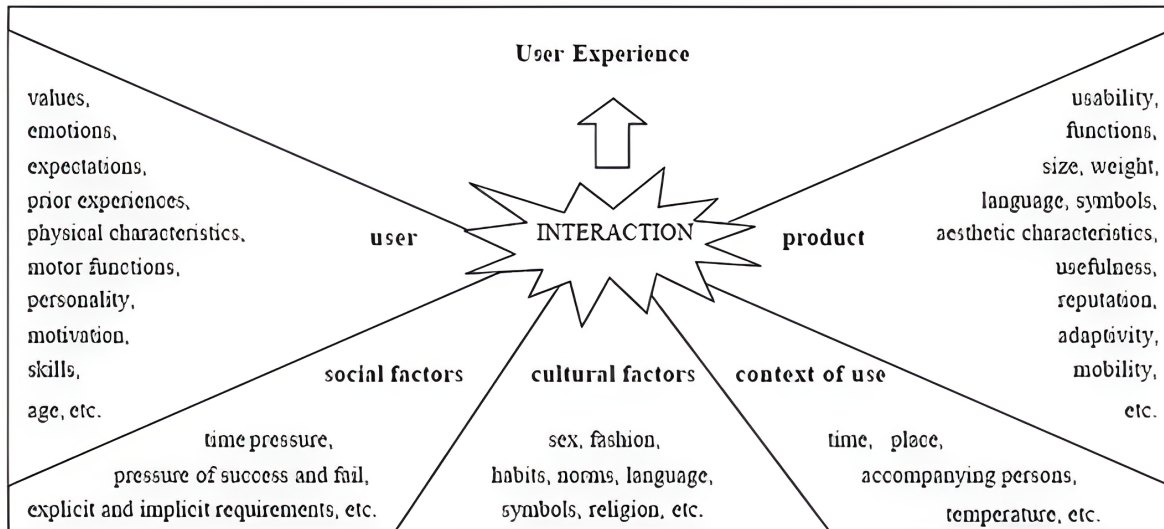


Figura 8: Factores involucrados en la Experiencia del Usuario  
Fuente: [Arhippainen y Tähti, 2003].

### 2.7.2. Relación con el comportamiento emocional del usuario

El estudio de la experiencia del usuario con sistemas informáticos está estrechamente relacionado con el comportamiento emocional del usuario, el cual es el resultado de tres distintos factores: las **emociones** provocadas por el producto durante la interacción, el estado de **humor** del usuario y los **sentimientos** previamente asociados por el usuario al producto.

Las emociones y el estado de humor son conceptos con distinto significado pero que a la vez se influyen mutuamente. La diferencia se encuentra en que las emociones son evocadas en la relación hacia un producto, mientras que el estado de humor es previo a la interacción con el sistema [Brave y Nass, 2007]. No obstante, la influencia mutua radica en que las emociones experimentadas influyen en el estado de humor del usuario, y por otro lado el humor del usuario condiciona la posibilidad de que un producto o sistema haga (o no) florecer en el usuario ciertas emociones.

Por otra parte, los sentimientos son propiedades de valor que el usuario asocia al producto, debido al resultado de experiencia previas con el mismo producto o similares. Por ejemplo, si una persona adquirió en el pasado un computador de una cierta marca, y el computador

le falló al poco tiempo de haberlo comprado, es probable que la persona adquiriera un sentimiento de disconformidad y desconfianza con el producto y con la marca.

Según [Hassan Montero y Martín Fernández, 2005], la interacción del usuario se ve fuertemente influenciada por los aspectos emocionales, debido a que los estados emocionales afectan los procesos cognitivos, afectando por ejemplo la capacidad de atención y memoria, el rendimiento del usuario y su percepción del producto. Por lo tanto, los estados afectivos del usuario influyen en la capacidad de resolver problemas racionales de mejor o peor manera.

Es por ello que es importante comprender las consecuencias de los estados emocionales del usuario durante la interacción con el sistema, y mayor aún, conocer cuales son sus causas, sobre todo las que dependen directamente del diseño del producto.

### 2.7.3. Diseño centrado en usuario

El diseño centrado en usuario (DCU) se refiere al diseño de productos y aplicaciones que sitúa al usuario en el centro de todo el proceso [Domingo y Pera, 2010]. Se entiende además como una filosofía de diseño en la que el proceso está conducido, en todas sus etapas, por la información proporcionada por la audiencia objetivo del producto.

Tal como la metodología de desarrollo SCRUM, el DCU presenta un proceso iterativo y cíclico, al contrario de otras metodologías lineales y secuenciales, con el fin de probar el diseño constantemente hasta alcanzar el nivel de calidad requerido.

Las etapas que componen el proceso de DCU son:

1. **Planificación/Investigación:** Se define el concepto del producto en base a la investigación de la audiencia objetiva y la competencia.
2. **Diseño/Prototipado:** Se toman las primeras decisiones de diseño y se construyen prototipos.
3. **Evaluación:** Las decisiones de diseño se ponen a prueba mediante métodos de evaluación que pueden involucrar a usuarios
4. **Implementación:** Si el diseño alcanza el nivel de calidad requerido, se procede a la implementación y puesta en producción.
5. **Monitorización:** Se estudia el uso que los usuarios dan al producto, con el fin de identificar oportunidades de mejora.

Para el diseño centrado en usuarios, es muy importante tener en consideración ciertos patrones o principios de diseño de software (ya sea web, móvil o *desktop*), los cuales ya han

sido probados por usuarios y estudiados por la comunidad, y que ayudaran al diseñador a poder realizar un mejor diseño de su producto.

#### 2.7.4. Principios de diseño

[Montero, 2015] presenta en su libro una selección con los principios de diseño más relevantes para aportar a una mejor experiencia de usuario en aplicaciones, entre los cuales se encuentran:

- **Clasificación:** La función principal de la clasificación del contenido en una aplicación es facilitar la recuperación de información, permitiendo al usuario satisfacer la necesidad de informarse. Se pueden considerar distintos tipos de clasificación, dependiendo del propósito, la naturaleza de los contenidos y el tipo de necesidad de información del usuario, como por ejemplo la clasificación orientada a la tarea, orientada a la audiencia o la clasificación temática.
- **Color:** El color es uno de los recursos con mayor impacto en la estética y atractivo de una interfaz de usuario, y funciona eficazmente tanto para destacar información y organizar elementos como para codificar información. Es importante tener en cuenta los aspectos de color tales como la cantidad de colores a utilizar, el uso que se le va a dar para codificar cierto tipo de información, la combinación de colores (por ejemplo, evitar la combinación de colores como el rojo, verde, azul o amarillo), y la saturación.
- **Eficiencia:** Cada segundo ahorrado al usuario mejora su experiencia de uso, y por ello el sistema o producto debe responder a cada acción del usuario en el menor tiempo posible, idealmente de forma instantánea. Un ejemplo donde el sistema debe ser eficiente es en el ingreso de datos a través de formularios, donde se aconseja minimizar el número de campos, establecer valores por defecto, buen tamaño de los campos de textos, cuidar la alineación de las etiquetas u organizar los campos en más de una columna.
- **Error Humano:** El diseño debe prevenir que el usuario cometa errores, y ofrecer vías de solución cuando ocurran, ya que son la mayor causa de frustración en el usuario y por tanto de experiencias de usuario negativas. Algunas de las medidas a tomar en cuenta en el diseño son limitar las posibles formas en las que el usuario puede introducir un dato o llevar a cabo una acción, ofrecer ayuda mediante breves explicaciones o *tooltips*, advertir y solicitar confirmación en acciones irreversibles, permitir deshacer cambios y mostrar mensajes de error comprensibles, idealmente ofreciendo solución al problema.
- **Jerarquía Visual:** El principio de jerarquía visual establece que debemos dar mayor peso visual a aquellos elementos que en cada momento de la interacción puedan resultar más relevantes para el usuario, guiando y facilitando de este modo su tarea

de búsqueda visual. Algunos atributos gráficos que se pueden utilizar para destacar visualmente un elemento son animaciones/movimientos, tamaño, color, orientación/rotación o forma.

- **Ordenación:** Los diferentes contenidos y opciones deben ordenarse de tal forma que se facilite al usuario la consecución de su tarea y la localización del contenido buscado en cada momento. Por ejemplo, se recomienda una ordenación alfabética de contenidos cuando el usuario conoce a priori el nombre de la opción que necesita en cada momento (selección de idiomas en una aplicación). Por otra parte, se recomienda una ordenación geográfica cuando el usuario busca por contenidos relativos a países o regiones conocidas. Por último, se recomienda una ordenación cronológica cuando el usuario conoce la fecha o los periodos en los que se enmarcan los contenidos deseados.

Cabe destacar que existen otros principios de diseño, como el mapeo natural, Ley de Fitts, inteligencia colectiva o selección de iconos. No obstante, los principios resumidos anteriormente son los que representan a la mayoría, y que sientan las bases a respetar por los diseñadores y desarrolladores para que los usuarios puedan tener la mejor experiencia en aplicaciones y productos de software.

## CAPÍTULO 3

### PROPUESTA DE SOLUCIÓN

#### 3.1. Metodología propuesta

La metodología propuesta en esta investigación se centra en la implementación de un rediseño continuo para soluciones de software, basándose en el análisis de sentimiento de las opiniones de los usuarios. Para ello, se incorpora el uso de la plataforma **Feel UX**, que automatiza la recopilación, procesamiento y visualización de datos, contribuyendo así al proceso de testing de usabilidad del software de manera efectiva.

A continuación, se detallan las cinco etapas que conforman la metodología, junto con el diseño metodológico ilustrado en la figura 9:

1. **Diseño e Implementación de Plataforma Automatizada Feel UX:** Esta etapa inicia con el diseño de los diversos componentes que integran una solución de software web. Continúa con la implementación del código correspondiente y culmina con el despliegue de la aplicación en servidores cloud, asegurando así su accesibilidad global a través de internet.
2. **Testing de Usabilidad mediante Feel UX:** Esta fase implica llevar a cabo una evaluación de facilidad de uso de algún software mediante la ejecución de tareas predefinidas. Todo este proceso se realiza a través de la plataforma Feel UX, que facilita la recopilación de datos del usuario que posteriormente serán procesados.
3. **Procesamiento de Datos y Minería de Opinión:** En esta etapa, se lleva a cabo el procesamiento de los datos (de forma automatizada) obtenidos durante el testing de la etapa 2. Esto implica la realización de diversas transformaciones y cálculos para extraer métricas significativas. Además, se efectúa un análisis de sentimiento de las opiniones de los usuarios con el objetivo de cuantificar el sentir general de los usuarios respecto al software testeado.
4. **Análisis y mejoramiento continuo:** En esta etapa, se lleva a cabo un análisis detallado de los datos a través de los diversos dashboards proporcionados por Feel UX, sin dejar de lado el sentir de los usuarios y las observaciones del encargado del testing. El objetivo es identificar oportunidades de mejora para el software evaluado e implementarlas de manera efectiva.
5. **Comparación de Resultados y Conclusiones:** Esta fase se lleva a cabo después de tres iteraciones de testing, procesamiento de datos, análisis y mejora continua. En este punto, se realiza una observación global del estudio comparando los resultados obtenidos en las diferentes iteraciones, y que culmina con la formulación de conclusiones

significativas que resumen los hallazgos y proporcionan una visión integral del desempeño y la evolución del software evaluado.

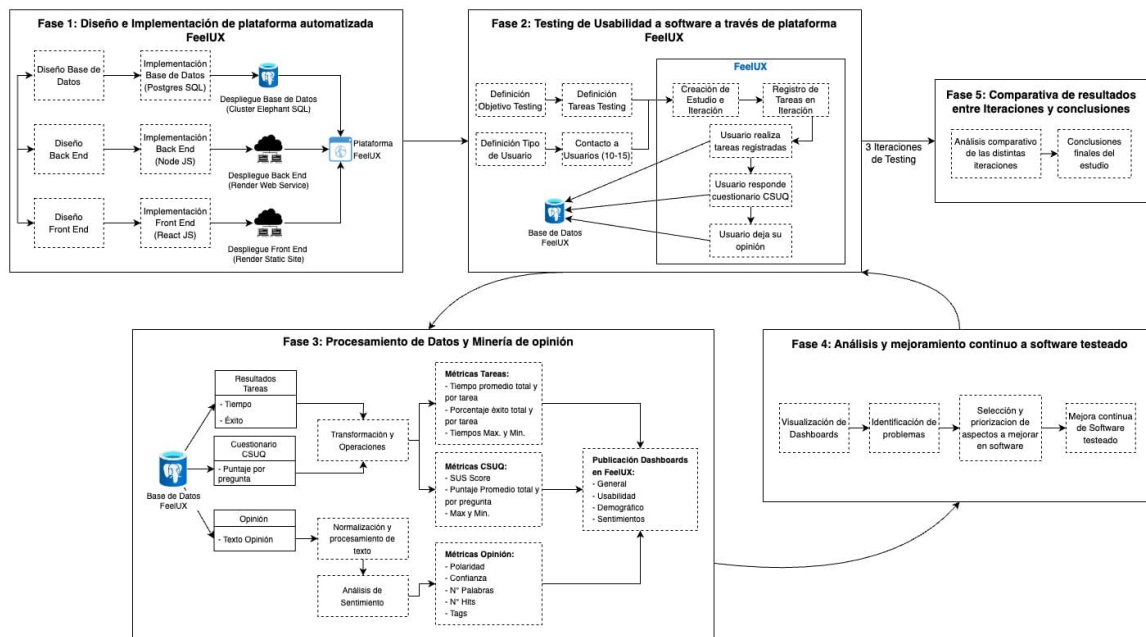


Figura 9: Diseño metodológico de la propuesta de solución.

Fuente: Elaboración propia.

## 3.2. Diseño e implementación plataforma Feel UX

### 3.2.1. Descripción general

Feel UX es una plataforma web que automatiza de manera integral el proceso de recopilación, procesamiento y visualización de datos en un contexto de testing de usabilidad para diversos tipos de software. Además, como componente esencial de su funcionalidad, la plataforma incorpora análisis de sentimiento basado en inteligencia artificial para evaluar las opiniones de los usuarios, contribuyendo así a asegurar la conformidad en calidad y madurez de los productos de software.

Dado que Feel UX es una plataforma que integra diversas funcionalidades, se dedicará una sección completa para detallar las diferentes características que ofrece, incluyendo imágenes de la interfaz de usuario que proporcionarán una visión más integral de su funcionamiento.

A continuación, se describen los detalles del diseño e implementación de los diversos componentes de la plataforma Feel UX.

### 3.2.2. Diseño e implementación Base de datos

En este proyecto de memoria, se ha optado por una base de datos de tipo relacional, representada en el modelo de la figura 10. Las tablas incluidas en dicho modelo y sus descripciones son las siguientes:

1. **Answers\_CSUQ:** Corresponde a las respuestas del cuestionario CSUQ de cada usuario, realizado al final de cada iteración. Contiene un total de 16 respuestas, junto con los promedios individuales de cada categoría y el puntaje final previamente calculado.
2. **Info\_task:** Almacena la duración y el éxito de las tareas realizada por cada usuario para una iteración determinada.
3. **Iteration:** Contiene la información de las iteraciones creadas para un estudio, como el estado (creada, activa o finalizada), el número de iteración, fecha de inicio y término de la iteración, cantidad de tareas, cantidad de usuarios activos y que completaron la iteración, además del objetivo de la iteración.
4. **Iteration\_state:** Almacena el estado de la iteración para cada usuario. Este puede estar en tres fases distintas, en la fase de realización de tareas, en la fase donde contesta el cuestionario CSUQ o en la fase donde deja su opinión. Además, guarda la última tarea completada por el usuario para evitar repeticiones de tareas previamente realizadas.
5. **Keywords:** Corresponde a todas las *keywords* o palabras clave de las opiniones de cada usuario.
6. **Sentiment\_analysis:** Contiene las opiniones de los usuarios para cada iteración, además de los datos correspondientes al análisis de sentimiento, como el *score* (valor decimal de -1 a 1), el voto (positivo, negativo, neutro), la cantidad de palabras y la cantidad de *hits* o palabras que aportan sentimiento a la opinión. Los campos *user\_sentiment* y *false\_positive* se utilizan para evaluar la confianza del análisis de sentimiento obtenido.
7. **Study:** Almacena la información de los estudios creados por los *Tester*, o encargados del testing, como el nombre del software a testear, el tipo de aplicación (móvil, web, escritorio u otros), la URL del sitio web del software, la cantidad de iteraciones del estudio, el número de la última iteración creada y la cantidad de iteraciones activas.
8. **Tasks:** Contiene los datos de las tareas de cada iteración, como el título, la descripción, su dificultad, y los minutos y segundos que el *tester* establezca como objetivo para las tareas.
9. **Testing\_user:** Corresponde al usuario que prueba la aplicación, almacenando su *username* o apodo, el correo electrónico, una contraseña para iniciar sesión, el nivel de conocimiento tecnológico (bajo, medio, alto), y su fecha de nacimiento.

10. **Tester:** Corresponde a la información del encargado o líder del *testing*, que es idéntica a la información que se almacena para el *Testing\_user*.

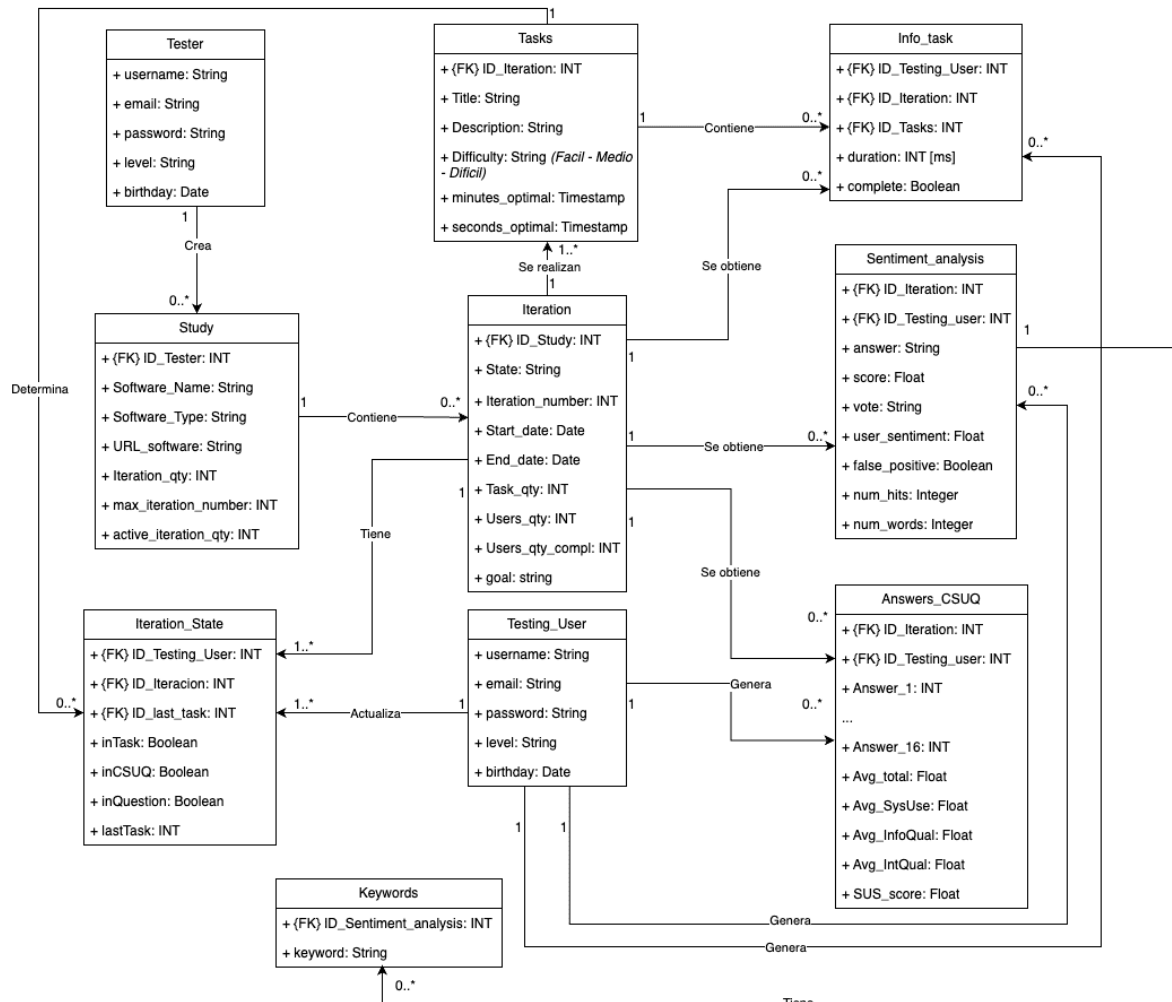


Figura 10: Modelo relacional base de datos Feel UX.

Fuente: Elaboración propia.

En cuanto a la implementación de la base de datos, se utilizó **PostgreSQL** como sistema de gestión de base de datos relacional, implementada como servicio en la nube a través de la plataforma **ElephantSQL**. Este tipo de servicios facilita el acceso a una base de datos sin tener que preocuparse por la administración de la infraestructura subyacente.

Finalmente, para la creación y manipulación de las tablas, se optó por emplear **Sequelize** como ORM. Esta elección permitió realizar consultas a la base de datos de manera eficiente, a través del lenguaje de programación **JavaScript**. En el extracto de código 1 se realiza la definición de la tabla *InfoTask*, mientras que en el código 2 se implementa la relación de 1 a muchos entre *Tasks* e *InfoTask*, todo a través de Sequelize.

```
1 module.exports = (sequelize, Sequelize) => {
2   const InfoTask = sequelize.define("infotasks", {
3     complete: {
4       type: Sequelize.BOOLEAN,
5     },
6     duration: {
7       type: Sequelize.INTEGER
8     }
9   });
10  return InfoTask;
11  };
```

---

Código 1: Definición del modelo InfoTask en Sequelize.

```
1 const db = {};
2 db.Sequelize = Sequelize;
3 db.sequelize = sequelize;
4 db.task = require("../models/task.model.js")(sequelize, Sequelize)
5 db.infotask = require("../models/infotask.model.js")(sequelize,
6   Sequelize)
7 db.task.hasMany(db.infotask, {
8   foreignKey: 'taskId'
9 });
10 db.infotask.belongsTo(db.task);
```

---

Código 2: Definición de relación entre InfoTask y Tasks en Sequelize.

### 3.2.3. Diseño e implementación Back-end

Para la implementación del back-end o servidor, se optó por el lenguaje JavaScript mediante el *framework* **Node.js**, siguiendo un patrón de diseño MVC (Modelo, Vista, Controlador).

Se introdujo además una pequeña variación a este patrón, a través del uso de *Middlewares*, para gestionar de manera eficiente las tareas intermedias, asegurando así una ejecución suave y ordenada de las solicitudes HTTP. Este enfoque permite una estructura organizada y modular, facilitando el mantenimiento y la escalabilidad del sistema.

Hay dos funcionalidades implementadas en el backend que se describirán con mayor detalle en esta sección. Estas funciones abarcan las consultas a la base de datos y el procesamiento de datos, así como la implementación del procesamiento del lenguaje natural para analizar el sentimiento de las opiniones de los usuarios.

En primer lugar, dado que uno de los objetivos principales de Feel UX consiste en presentar información de los estudios mediante dashboards, las consultas a la base de datos son una tarea recurrente. Estas consultas se ejecutan mediante Sequelize, y el procesamiento de los datos se lleva a cabo mediante funciones en JavaScript, como se evidencia en los extractos

de código 3 y 4 respectivamente.

```
1 const db = require("../models");
2 const Iteration = db.iteration;
3
4 exports.cards = async (req, res) => {
5   const idStudy = req.query.idStudy;
6   try {
7     const allIterations = await Iteration.findAll({
8       where: {
9         studyId: idStudy,
10      }
11    });
12    if (!allIterations) {
13      return res.status(404).json({ error: "Estudio no encontrado."
14        });
15    }
16  } catch (error) {
17    console.error(error);
18    res.status(500).json({ error: "Ha ocurrido un error al obtener
19      los datos" });
20  }
21 };
```

---

Código 3: Consulta de tipo *Select* a través del método *findAll* de Sequelize.

```
1 async function calculateCompletePercentage(taskIds) {
2   if (taskIds.length === 0) {
3     return 0;
4   }
5   // Obtener las InfoTasks correspondientes a los ids de las tareas
6   const allInfoTasks = await InfoTask.findAll({
7     where: {
8       taskId: {
9         [Op.in]: taskIds,
10      },
11    },
12  });
13  if (allInfoTasks.length === 0) {
14    return 0;
15  }
16  const total_task_qty = allInfoTasks.length
17  const completed_task_qty = allInfoTasks.filter(task =>
18    task.complete === true).length;
19  const percentageTaskComplete = completed_task_qty / total_task_qty
20  const porcentaje = (percentageTaskComplete * 100).toFixed(1) + "%";
21  return (porcentaje)
22 }
```

---

Código 4: Función de JavaScript para calcular el porcentaje completado de tareas.

Y en segundo lugar, otra de las funciones principales dentro de la aplicación es el procesamiento de lenguaje natural a través del análisis de sentimiento. Para ello se utilizaron tres librerías que cumplen distintas funciones dentro del código:

1. **Eden AI:** La plataforma Eden AI brinda la capacidad de emplear simultáneamente diversos motores de Análisis de Sentimientos, permitiendo seleccionar el motor más adecuado según el texto del cual se desee extraer el sentimiento. Esta funcionalidad se logra al consolidar múltiples proveedores de Análisis de Sentimientos en una única API. En el caso específico de la plataforma implementada en esta memoria, se optó por utilizar el motor de Microsoft, como se evidencia en la función del código 5. El objeto devuelto por la API (ver código 6) contiene tres propiedades: el texto analizado (*segment*), el sentimiento (positivo, negativo, neutral) y la tasa de sentimiento (número decimal entre 0 y 1) que corresponde a la intensidad del sentimiento del texto evaluado.
2. **nlpjs/lang-es:** Se trata de una biblioteca de procesamiento del lenguaje natural (NLP) de código abierto diseñada para la construcción de bots, que incluye características como extracción de entidades, análisis de sentimientos, identificación automática de idiomas, entre otras funcionalidades. En el contexto de esta memoria, se empleó el módulo *NormalizerEs* (ver código 7), el cual transforma un texto a minúsculas y elimina los caracteres decorativos de una oración, esto con el fin de preprocesar la opinión entregada por los usuarios y posteriormente ser analizada.
3. **node-nlp:** es una librería de procesamiento del lenguaje natural para Node.js que ofrece diversas funcionalidades. Entre sus características se incluyen el análisis de sentimientos para frases basado en el léxico AFINN, con soporte para negaciones. Para el software implementado en esta memoria, esta biblioteca fue empleada para identificar las palabras clave de cada opinión, seleccionando aquellas palabras que contribuían con un grado de sentimiento a la oración, como se evidencia en el código 8.

Es importante señalar que para la identificación de palabras clave en las opiniones de los usuarios, se podría haber optado por utilizar la API de Eden AI. No obstante, la capa gratuita de esta plataforma impone restricciones en la cantidad de consultas permitidas a sus servicios. Por consiguiente, al aprovechar node-nlp, una biblioteca de código abierto, se elude de cualquier limitación en la cantidad de consultas y así utilizar Eden AI para el análisis de sentimiento exclusivamente de las opiniones.

```
1 const sentimentAnalysisIA = (language, opinion) => {
2   const options = {
3     method: "POST",
4     url: "https://api.edenai.run/v2/text/sentiment_analysis",
5     headers: {
6       authorization: "Bearer " + apiKey,
7     },
8     data: {
9       providers: "microsoft",
10      text: opinion,
11      language: language,
12      fallback_providers: "",
13    },
14  };
15  return axios.request(options)
16 }
```

---

Código 5: Función de JavaScript que hace una llamada a la API de Eden AI para análisis de sentimiento a través del motor de Microsoft.

```
1 {
2   "segment": "En general, estoy satisfecho con el software.",
3   "sentiment": "Positive",
4   "sentiment_rate": 0.96
5 },
```

---

Código 6: Ejemplo de objeto que retorna la llamada a la API de Eden AI.

```
1 const { NormalizerEs } = require('@nlpjs/lang-es');
2 const normalizer = new NormalizerEs();
3 const input = 'Esto debería ser normalizado';
4 const result = normalizer.normalize(input);
5 console.log(result);
6 // output: esto deberia ser normalizado
```

---

Código 7: Ejemplo de utilización modulo *NormalizerEs* en JavaScript.

```
1 const { SentimentManager } = require('node-nlp');
2 async function analizarSentimiento(texto) {
3   const sentiment = new SentimentManager();
4   const result = await sentiment.process('es', texto);
5   return result;
6 }
7 const keywords_no_neutral = []
8 for (const keyword of keywords) {
9   const analisis_key = await analizarSentimiento(keyword);
10  if (analisis_key.vote !== "neutral") {
11    keywords_no_neutral.push(keyword)
12    await Keyword.create({
13      userId: req.body.idUser,
14      iterationId: req.body.idIteration,
15      keyword: keyword,
16    });
17  }
18 }
```

---

Código 8: Utilización de módulo *SentimentManager* de la librería Node-nlp para la identificación de *keywords*.

La siguiente sección proporcionará detalles sobre el diseño y la implementación del front-end o interfaz de usuario de la plataforma Feel UX.

#### 3.2.4. Diseño e implementación Front-end

Para la implementación del front-end, se seleccionó el lenguaje de programación JavaScript a través del *framework* **React**. Esta elección posibilita el diseño de interfaces web mediante la utilización de componentes, lo que proporciona flexibilidad y permite una estructuración modular y reutilizable del código.

El diseño de la interfaz de usuario se llevó a cabo de manera manual, comenzando con bocetos iniciales elaborados con lápiz y papel. Para evaluar colores y tamaños, se empleó la plataforma **Canva** para crear maquetas detalladas de las diversas vistas de Feel UX (ver figura 11).

La imagen muestra una maqueta de una interfaz de usuario para un cuestionario CSUQ. En la esquina superior izquierda hay un icono de casa y un enlace "Volver a Inicio". El título principal es "CUESTIONARIO CSUQ". Debajo del título, un texto explica que el cuestionario consta de 16 preguntas para evaluar la satisfacción del usuario, con una escala de 1 a 7 (1: "Muy en desacuerdo", 7: "Muy de acuerdo"). El formulario principal es un recuadro oscuro con el texto "1.pregunta 1" y una barra de escala horizontal con los números "1" y "7" en los extremos. En la parte inferior del formulario hay dos botones: "Atras" y "Siguiete". Debajo de los botones, se indica "1 de 4".

Figura 11: Maqueta realizada en Canva para vista del cuestionario CSUQ en Feel UX.  
Fuente: Elaboración propia.

Con el objetivo de implementar funcionalidad a la plataforma y un diseño visual acorde a los estándares actuales, se incorporaron diversas bibliotecas, entre las cuales se encuentran:

- **Axios:** Es una biblioteca que simplifica las solicitudes HTTP en aplicaciones React, facilitando la interacción con servidores web. Permite gestionar eficientemente la comunicación entre el front-end y el back-end. En el fragmento de código 9, se ilustra su uso mediante el método GET para obtener las iteraciones de un estudio específico.
- **React-bootstrap:** Esta biblioteca ofrece componentes React predefinidos que siguen el diseño y estilo de Bootstrap, facilitando la creación de interfaces de usuario modernas y receptivas. En el código 10 se encuentra la implementación del componente *Logout-Modal*, que a su vez utiliza los componentes *Button* y *Modal* de React-Bootstrap.
- **Tremor:** Esta biblioteca de React simplifica la creación de *dashboards* responsivos mediante componentes y clases diseñadas para mejorar la organización y visualización de gráficos e indicadores. En el extracto de código 11 se implementan los componentes *Grid* y *Column* dentro del componente *DashboardDemogrIteration*.
- **React-ApexCharts:** Es una biblioteca interactiva de gráficos diseñada para JavaScript, que posibilita la creación de una amplia gama de representaciones visuales, incluyendo gráficos de líneas, barras, áreas, donas y otros tipos. Esta fue la principal librería empleada para la creación de gráficos dentro de la aplicación, como se evidencia en el fragmento de código 12, donde se implementa un gráfico del tipo *Boxplot*.

```
1 import axios from "axios";
2 import authHeader from "../auth-header";
3 const API_URL = "http://localhost:8080/api/test/";
4 const getIterations = (idStudy) => {
5   return axios.get(API_URL + "iterations", { headers: authHeader(),
6     params: { idStudy: idStudy } });
7 }
```

---

Código 9: Utilización de librería Axios para consultas a back-end en React.

```
1 import { Modal, Button } from 'react-bootstrap';
2 const LogoutModal = ({ show, handleClose, handleLogout }) => {
3   return (
4     <Modal show={show} onHide={handleClose}>
5       <Modal.Header>
6         <Modal.Title style={{ color: '#344b60', fontFamily: "
7           Poppins, sans-serif", fontSize: '24px' }}>
8           Confirmación</Modal.Title>
9       </Modal.Header>
10      <Modal.Body>
11        <p
12          style={{ color: '#344b60', fontFamily: "Poppins,
13            sans-serif", fontSize: '16px' }}>
14          ¿Estás seguro que deseas Cerrar Sesión?</p>
15      </Modal.Body>
16      <Modal.Footer>
17        <Button variant="secondary" onClick={handleClose}>
18          Cancelar
19        </Button>
20        <Button variant="primary" onClick={handleLogout}
21          className="btn button-primary">
22          Cerrar Sesión
23        </Button>
24      </Modal.Footer>
25    </Modal>
26  );
27 };
28 export default LogoutModal;
```

---

Código 10: Utilización de componentes predefinidos de React-bootstrap en el componente *LogoutModal*.

```
1 import { Grid, Col } from "@tremor/react";
2 const DashboardDemogrIteration = (props) => {
3   return (
4     <div>
5       <Grid numItemsSm={1} numItemsLg={5} className="gap-6">
6         <Col numColSpan={1} numColSpanLg={2}>
7           <MetricCardList content={cardsContent.cantidad_usuarios
8             } color="amber" />
9           <div className="m-3"></div>
10          <PieChart title="Distribución Rango Etario" color="blue
11            " content={pieChartContent} legendPosition='right'
12            widthChart='120%' />
13        </Col>
14        <Col numColSpan={2} numColSpanLg={3}>
15          <BarChartGraphic
16            content={barChartContent.chartData}
17            valueFormatter={valueFormatter}
18            title="Usuarios por Rango Etario y Conocimiento
19              tecnológico"
20            categories={barChartContent.categories}
21            color={barChartContent.colors}
22            stack={true} />
23        </Col>
24      </Grid>
25    </div>
26  )
27 }
28 export default DashboardDemogrIteration
```

---

Código 11: Utilización de componentes predefinidos de Tremor en el componente *DashboardDemogrIteration*.

```
1 import React, { useState } from 'react';
2 import ReactApexChart from 'react-apexcharts';
3 import { Card, Title } from "@tremor/react";
4 const BoxPlotChart = (props) => {
5   const {content, title} = props
6   const [chartState] = useState({
7     series: content.series,
8     options: {
9       chart: {
10         type: 'boxPlot',
11         height: 350
12       },
13       plotOptions: {
14         boxPlot: {
15           colors: {
16             upper: '#5C4742',
17             lower: '#A5978B'
18           }
19         }
20       }
21     }
22   });
23   return (
24     <Card>
25       <Title>{title}</Title>
26       <div style={{margin: "2%"}}></div>
27       <ReactApexChart
28         options={chartState.options}
29         series={chartState.series}
30         type="boxPlot"
31         height={296} />
32     </Card>
33   );
34 };
35 export default BoxPlotChart;
```

---

Código 12: Utilización de componentes predefinidos de React ApexCharts en el componente *BoxPlotChart*.

En secciones posteriores, se expondrá con detalle las diversas funcionalidades de la plataforma Feel UX, presentando imágenes que ilustren visualmente los aspectos discutidos en esta sección.

### 3.2.5. Despliegue de aplicación

Para que la aplicación web pueda estar disponible a través de internet, es necesario hacer un despliegue en un servidor, ya sea físico o en la nube.

Para Feel UX, la elección fue desplegar tanto el front-end como el back-end en un servidor

en la nube utilizando la plataforma **Render**<sup>1</sup>. En su capa gratuita, esta plataforma facilita la carga de aplicaciones implementadas en diversas tecnologías, incluyendo Node.js y React JS, en 3 pasos:

1. **Seleccionar tipo de servicio:** Render da la posibilidad de desplegar distintos tipos de servicios, como sitios estáticos, *web services* o contenedores. En el caso del front-end de Feel UX, se desplegó como sitio estático, mientras que el back-end como *web service*.
2. **Vincular repositorio:** Para lograr un despliegue automático de la aplicación en respuesta a cambios, es necesario vincular un repositorio en Render. De esta manera, cada vez que se realice una actualización en el repositorio, Render detectará los cambios y llevará a cabo un re-despliegue automático de la aplicación sin tiempo de inactividad. Para Feel UX, se han establecido dos repositorios en GitHub, uno correspondiente al front-end y otro al back-end.
3. **Configurar despliegue:** Las configuraciones necesarias incluyen la selección de la rama del repositorio que Render monitoreará para el despliegue automático, la especificación del comando necesario para implementar la aplicación, y la configuración de las variables de entorno esenciales para el funcionamiento correcto de la aplicación. En el caso de Feel UX, se optó por utilizar la rama predeterminada *Main*, el comando *npm run build*, y las variables de entorno relacionadas con las claves secretas de la base de datos y de la API de Eden AI.

Tras completar estos pasos, es posible acceder a la plataforma implementada desde cualquier dispositivo mediante la dirección <https://feelux.onrender.com/> (ver figura 12).

Cabe destacar que, entre otras características, Render ofrece escalado automático basado en carga, que se refiere a la capacidad de ajustar automáticamente el número de recursos (como servidores) en función del nivel de demanda o tráfico que reciba la aplicación implementada. Por lo tanto, si Feel UX logra alcanzar a más usuarios, la plataforma estaría, en principio, preparada para poder dar servicio sin interrupciones.

Los enlaces a los repositorios públicos en GitHub, disponibles para visualización y colaboración, son los siguientes:

- **Back-end:** <https://github.com/nicolas772/backend-memoria>
- **Fron-end:** <https://github.com/nicolas772/frontend-memoria>

---

<sup>1</sup>Sitio Web: <https://render.com/>

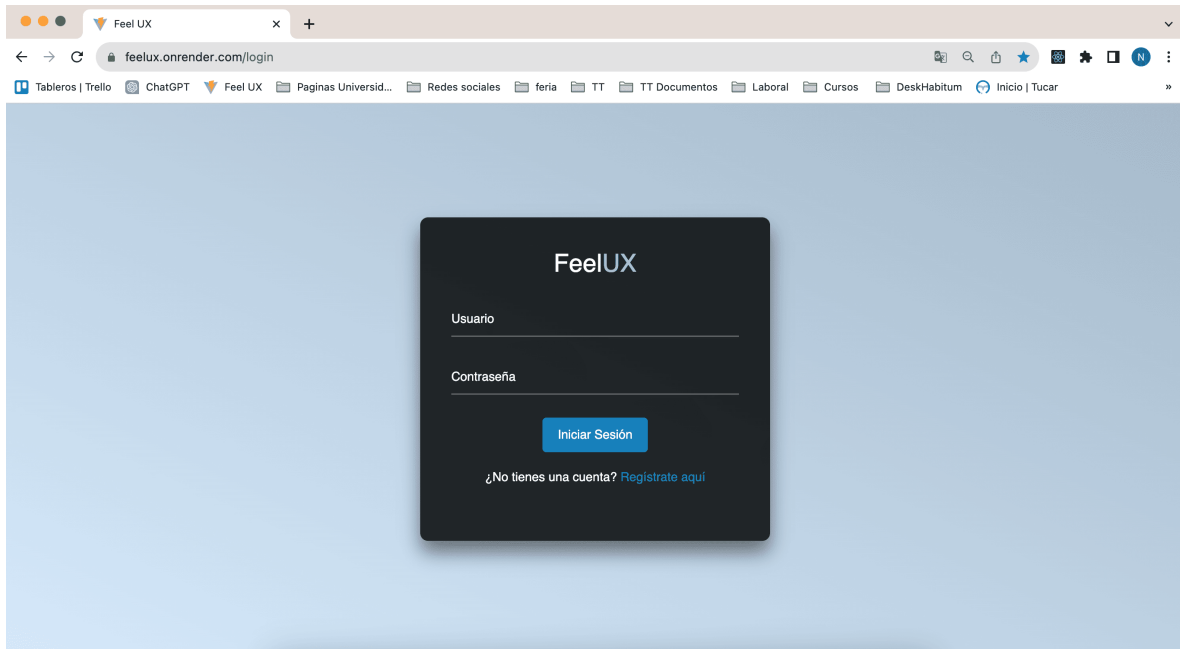


Figura 12: Página inicial de sitio web Feel UX.  
Fuente: Elaboración propia.

### 3.3. Funcionalidades de Feel UX

#### 3.3.1. Perfiles y registro de usuarios

Para comenzar, es necesario definir los tipos de perfiles presentes en la aplicación, que son:

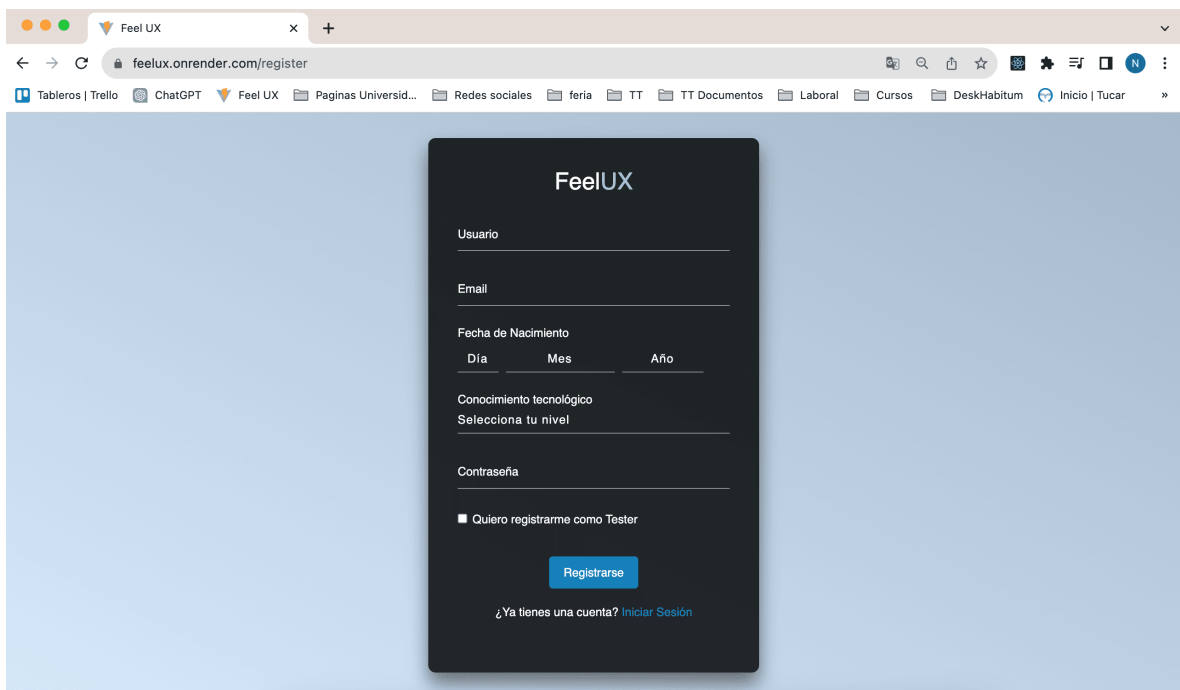
1. **Tester:** Este perfil está enfocado en el líder del testing, es decir, la persona interesada en evaluar su software. Este tipo de usuario tiene la capacidad de crear estudios, iteraciones y tareas. Además, podrá visualizar los resultados de sus estudios a través de los diversos dashboards disponibles en la aplicación.
2. **Testing User:** Este perfil corresponde a la persona que *testea* o evalúa el software. Dentro de la aplicación podrá participar en iteraciones creadas por el *Tester*, visualizar las tareas que debe realizar, completar el cuestionario CSUQ y dejar su opinión respecto al software testado.

El proceso de registro en la plataforma consiste en completar un formulario con los siguientes campos (ver figura 13):

- **Usuario:** Corresponde al nombre de usuario o *nickname*. Este campo es único para cada usuario.

- **Email:** Corresponde al correo electrónico del usuario. Este campo también es único para cada usuario.
- **Fecha de nacimiento:** Este campo se solicita al usuario para obtener métricas según el rango de edad, que es relevante evaluar para los estudios de usabilidad.
- **Conocimiento tecnológico:** Al igual que la fecha de nacimiento, este campo se solicita para obtener métricas relevantes según el nivel de conocimiento tecnológico del usuario, que puede elegir entre básico, medio y avanzado.
- **Contraseña:** Para poder ingresar a la sesión, el usuario debe registrar una contraseña segura. Esta contraseña se guarda en la base de datos de forma encriptada para garantizar la seguridad de los datos.
- **Quiero registrarme como Tester:** Este campo corresponde a un *checkbox*, que permite diferenciar el tipo de perfil de la persona que se está registrando. Si el *checkbox* se encuentra activado, el usuario tendrá perfil Tester; en cambio, si está desactivada, tendrá el perfil Testing User.

La figura 14 muestra la interfaz de la página de inicio del perfil Tester, mientras que la figura 15 representa la vista de inicio del perfil Testing User.



The image shows a web browser window with the URL 'feelux.onrender.com/register'. The browser's address bar and tabs are visible at the top. The main content is a dark-themed registration form titled 'FeelUX'. The form contains the following fields and elements: a 'Usuario' field, an 'Email' field, a 'Fecha de Nacimiento' section with 'Día', 'Mes', and 'Año' sub-fields, a 'Conocimiento tecnológico' section with the instruction 'Selecciona tu nivel', a 'Contraseña' field, a checkbox labeled 'Quiero registrarme como Tester', a blue 'Registrarse' button, and a link at the bottom that says '¿Ya tienes una cuenta? [Iniciar Sesión](#)'.

Figura 13: Formulario de registro en sitio web Feel UX.  
Fuente: Elaboración propia.



Figura 14: Página de inicio para perfil Tester en Feel UX  
Fuente: Elaboración propia.

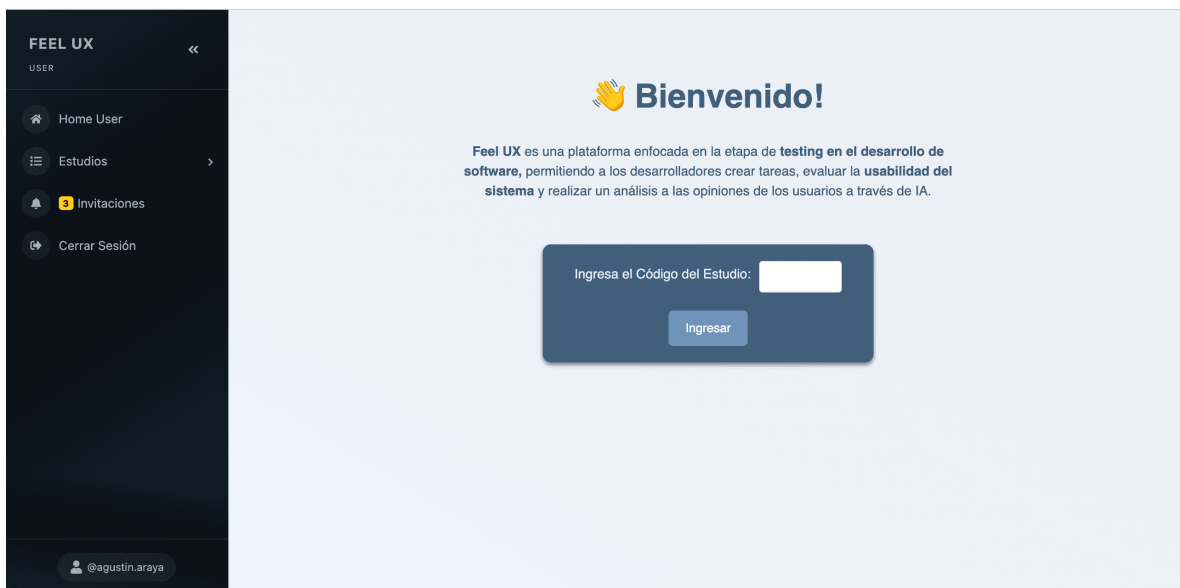


Figura 15: Página de inicio para perfil Testing User en Feel UX  
Fuente: Elaboración propia.

### 3.3.2. Creación de Estudios, Iteraciones y Tareas

La creación de estudios, iteraciones y tareas esta a cargo del **Tester**, y es fundamental para poder iniciar con todo el flujo de trabajo dentro de la plataforma Feel UX.

La relación entre estudios, iteraciones y tareas se ilustra en la figura 10. Esta relación implica que un estudio puede tener una o más iteraciones, y una iteración puede tener una o más tareas que el usuario debe realizar en el software que será evaluado.

El primer paso es **crear un estudio**. Para ello existen dos formas, una es a través de las opciones rápidas en la barra lateral o *Sidebar* (ver figura 16), y otra es a través del botón «Crear Estudio» dentro de la sección «Mis Estudios» (ver figura 17). En ambos casos, se debe completar un formulario con los siguientes campos: nombre del software, tipo de software y URL del software.

El segundo paso implica la **creación de una iteración** para un estudio. Al igual que con la creación de estudios, existen dos métodos para generar una iteración. El primero es mediante las opciones rápidas en el *Sidebar* (ver figura 18), mientras que el segundo es accediendo al estudio ya creado, donde se encuentra un botón dedicado para la creación de iteraciones (ver figura 19). En ambos casos, se debe completar un formulario con los siguientes campos: objetivo de la iteración, fecha de inicio y fecha de término de la iteración.

A diferencia de la creación de estudios, para crear una iteración mediante las opciones rápidas, es necesario seleccionar previamente el estudio al que estará asociada la iteración.

El tercer y último paso consiste en **crear tareas** para una iteración. En este caso también existen dos formas, una es a través de las opciones rápidas en el *Sidebar* (ver figura 20), mientras que la otra forma es accediendo a una iteración ya creada de un estudio en particular, donde se dispone de un botón para poder crear tareas (ver figura 21). En ambos casos se debe completar un formulario con los siguientes campos: título de la tarea, descripción de la tarea, dificultad (fácil, medio o difícil), y la duración óptima de la tarea.

De forma similar a la creación de una iteración, para crear una tarea a través de las opciones rápidas es necesario seleccionar previamente el estudio y la iteración al que estará vinculada la tarea.



Figura 16: Creación de estudio en opciones rápidas de Sidebar.  
Fuente: Elaboración propia.

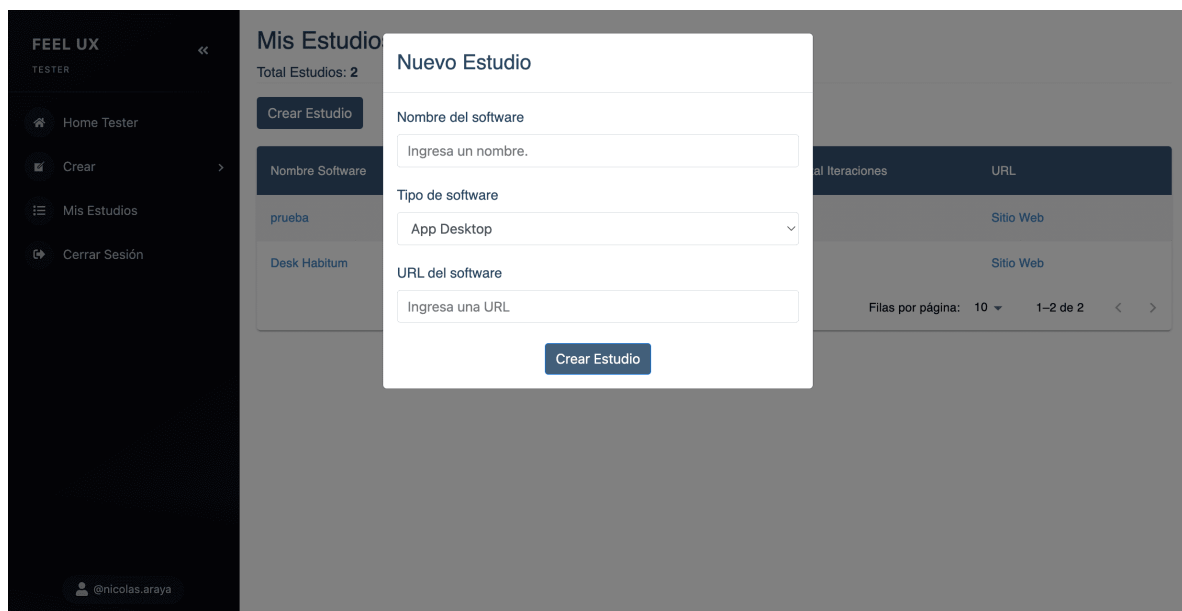


Figura 17: Creación de estudio dentro de sección «Mis Estudios».  
Fuente: Elaboración propia.



Figura 18: Creación de iteración en opciones rápidas de Sidebar.  
Fuente: Elaboración propia.

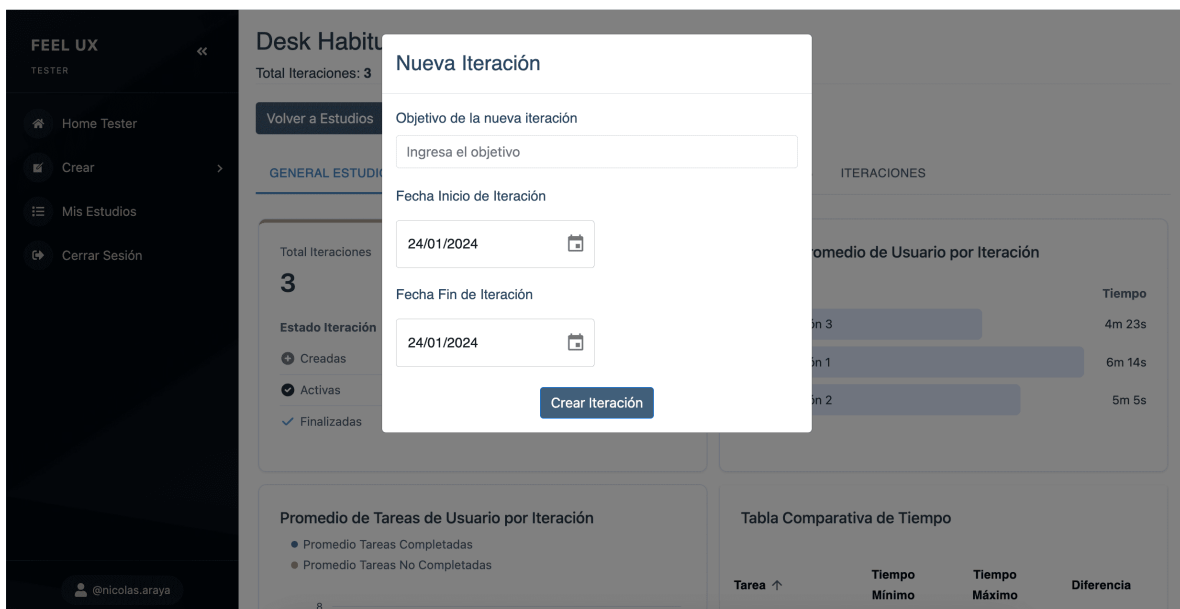


Figura 19: Creación de iteración dentro de un estudio ya creado.  
Fuente: Elaboración propia.

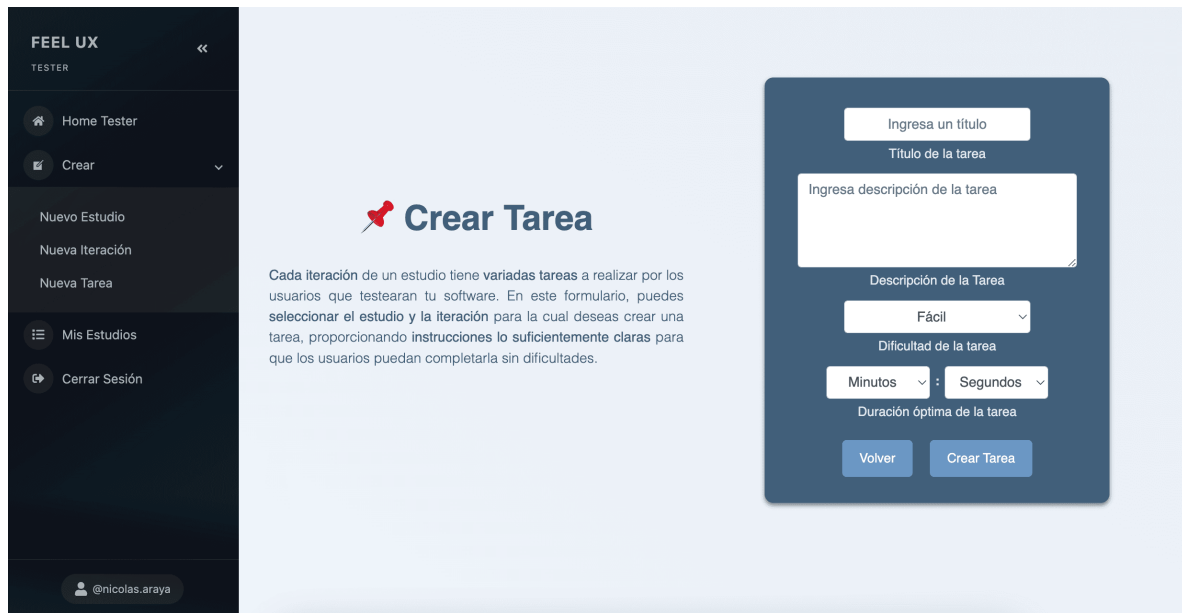


Figura 20: Creación de tarea en opciones rápidas de Sidebar.  
Fuente: Elaboración propia.

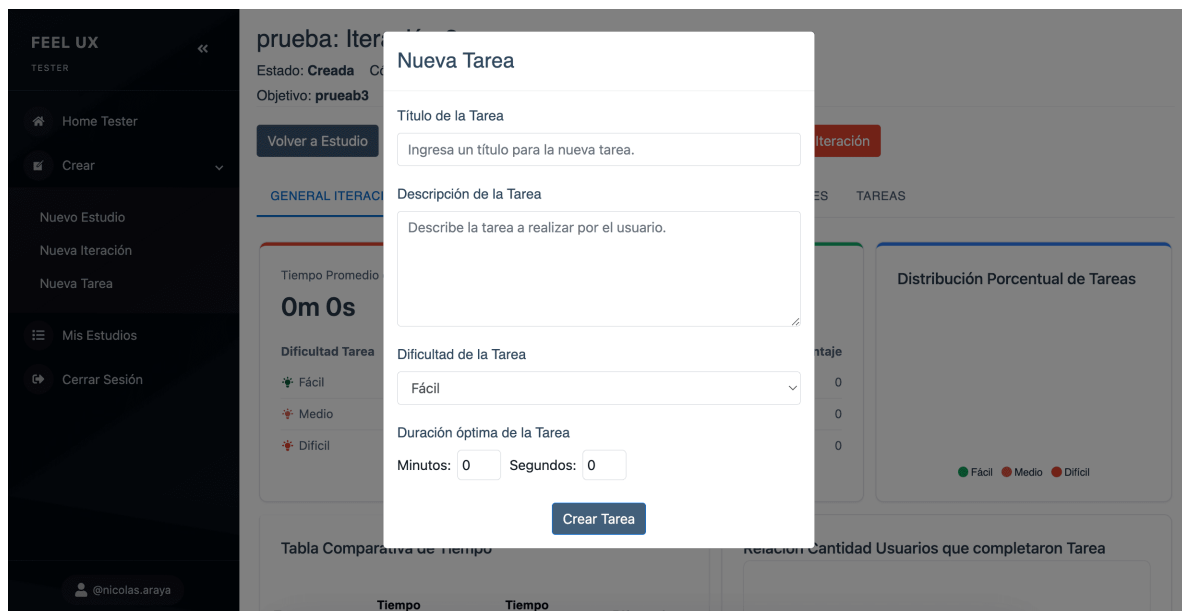


Figura 21: Creación de tarea dentro de una iteración ya creada.  
Fuente: Elaboración propia.

### 3.3.3. Activación de Iteraciones

Una vez creadas las tareas para una iteración de un estudio en particular, es necesario que el Tester realice la **activación de la iteración**. Esta acción es esencial para permitir que los Testing Users accedan a ella y lleven a cabo el estudio correspondiente.

Para activar una iteración, la plataforma cuenta con un botón ubicado en la parte superior de la vista de Iteración dentro de la sesión del Tester, como se muestra en la figura 22. Después de hacer clic en el botón, se solicita la confirmación del usuario para activar la iteración, ya que **hay acciones que no podrán llevarse a cabo** dentro de esta mientras se encuentre en estado «activa». Estas acciones incluyen:

- Eliminar la iteración.
- Crear nuevas tareas.
- Editar el título o descripción de las tareas.
- Eliminar tareas.

Después de confirmar la activación de la iteración, Feel UX generará un **código único de iteración**, que estará disponible en la parte superior de la vista de la Iteración, como se muestra en la figura 23. La idea principal es que este código sea compartido por el Tester con todos los Testing Users que participarán en el estudio.

Finalmente, la iteración está lista para poder ser completada por los Testing Users.



Figura 22: Botón que permite activar una iteración.  
Fuente: Elaboración propia.



Figura 23: Código único de la iteración, presente en la parte superior de la vista.  
Fuente: Elaboración propia.

### 3.3.4. Ingreso a Iteración y Tareas

Para que el Testing User acceda a la iteración y pueda comenzar el estudio, la plataforma dispone un cuadro en la vista principal de la sesión (ver figura 15), donde el usuario ingresará el código de la iteración proporcionado por el Tester.

Después de ingresar el código, el Testing User verá un resumen del estudio, que incluye detalles como el nombre del software, el tipo de software, el objetivo y la cantidad de tareas asociadas a la iteración. Cuando el usuario estime conveniente, podrá confirmar el inicio del estudio a través del botón dispuesto para ello, como se muestra en la figura 24.

Dentro del estudio, Feel UX presentará una nueva visual al usuario, con el título de la tarea, el enunciado de la tarea y dos botones: uno para comenzar y otro para finalizar la tarea (ver figura 25). Una vez concluida la tarea, se presentan dos botones adicionales para poder medir el éxito o fracaso de la tarea.



Figura 24: Resumen de la iteración, con botón dispuesto para iniciar estudio.  
Fuente: Elaboración propia.

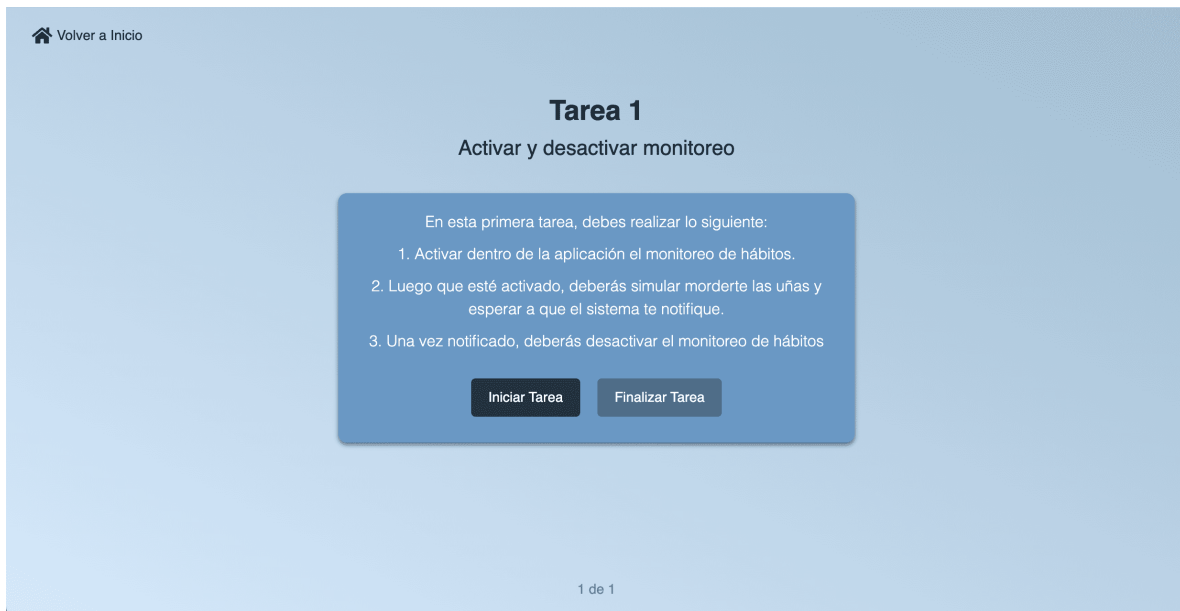


Figura 25: Interfaz que presenta las tareas a realizar en el estudio.  
Fuente: Elaboración propia.

Al completar todas las tareas de la iteración, el Testing User avanzará a la siguiente etapa del estudio, que consiste en responder al cuestionario CSUQ.

### 3.3.5. Cuestionario CSUQ

La siguiente etapa dentro del estudio corresponde al cuestionario CSUQ, en la que el Testing User debe responder 16 preguntas, asignando un puntaje de 1 a 7 a cada una de ellas (para ver detalle de preguntas, consultar figura 59).

Para ello, la plataforma presenta una interfaz similar a la etapa de tareas, que incluye una breve introducción al cuestionario CSUQ, el enunciado de cada pregunta y un selector horizontal para que el usuario asigne una respuesta, como se ilustra en la figura 26.

Es importante destacar que la resolución de este cuestionario es obligatoria, y el usuario no podrá finalizar el estudio hasta que responda todas las preguntas. Por lo tanto, el líder del estudio debe informar a los participantes que deben contemplar un tiempo adicional para completar este cuestionario.

Una vez respondidas todas las preguntas, el Testing User podrá avanzar a la última etapa, que consiste en expresar su opinión abierta respecto al software evaluado.

Volver a Inicio

### Cuestionario CSUQ

El siguiente cuestionario consta de 16 enunciados, los cuales fueron diseñados para evaluar tu satisfacción como usuario del software. Para cada uno de ellos debes responder de forma sincera, seleccionando la opción que mejor te represente.

1. En general, estoy satisfecho con lo facil que es utilizar este software.

●

Totalmente en desacuerdo   Bastante en desacuerdo   En desacuerdo   Indiferente   De acuerdo   Bastante de acuerdo   Totalmente de acuerdo

Anterior   Siguiente

1 de 16

Figura 26: Interfaz que permite responder cuestionario CSUQ dentro del estudio.  
Fuente: Elaboración propia.

### 3.3.6. Opinión de usuarios

La fase final del estudio consiste en la recopilación de la opinión del Testing User respecto al software evaluado. Para ello, se proporciona un cuadro donde puede expresar libremente su opinión, con un límite de 240 caracteres, tal como se muestra en la figura 27.

Esta funcionalidad es una de las más importantes de este trabajo, dado que la opinión proporcionada por el usuario en esta etapa es sometida a un análisis de sentimiento y guía el proceso continuo de rediseño de la aplicación evaluada. Por esta razón, la opinión del usuario es obligatoria, y no se le permite concluir el estudio hasta que haya completado esta etapa.

Como último paso, el usuario debe seleccionar el emoticono que mejor refleje su experiencia con el software evaluado, eligiendo entre 3 opciones que representan los sentimientos negativo, neutro y positivo (ver figura 28). Esta valoración se realiza con el objetivo de evaluar los posibles falsos positivos o falsos negativos que la biblioteca de Eden AI, en algunas ocasiones, no logra reconocer.

Es decir, si la API de análisis de sentimientos clasifica una opinión como «negativa», pero el usuario expresa un sentimiento positivo (selecciona el emoticono feliz), Feel UX considerará esta opinión como un falso negativo y no la incluirá en los resultados del estudio. De manera similar, sucede cuando una opinión es evaluada por la API como «positiva», pero el usuario manifiesta un sentimiento negativo (elige el emoticono triste).

The screenshot shows a light blue background. In the top left corner, there is a small house icon followed by the text 'Volver a Inicio'. In the center, the heading '¡Valoramos tu opinión!' is displayed in bold. Below it, a paragraph invites the user to share their general impressions of the software, stating that their input is essential for continuous improvement. A blue rounded rectangle contains the instruction '1. Expresa tu opinión general respecto al software.' followed by a large white text input field. At the bottom of this rectangle is a dark button labeled 'Siguiente'. At the bottom center of the page, the text '1 de 2' indicates the current step.

Figura 27: Interfaz que permite recopilar la opinión del usuario.  
Fuente: Elaboración propia.

The screenshot shows the same light blue background and top navigation as Figure 27. The heading '¡Valoramos tu opinión!' and the introductory paragraph are repeated. The blue rounded rectangle now contains the instruction '2. Elije el sentimiento que mejor te representa en relación al software testeado.' Below this, there are three circular icons: a sad face labeled 'Negativo', a neutral face labeled 'Neutro', and a happy face labeled 'Positivo'. At the bottom of the rectangle are two dark buttons: 'Atras' and 'Terminar Estudio'. At the bottom center of the page, the text '2 de 2' indicates the final step.

Figura 28: Interfaz que permite al usuario seleccionar el sentimiento respecto al software.  
Fuente: Elaboración propia.

### 3.3.7. Visualización de resultados

Para poder visualizar los resultados del estudio, Feel UX ofrece diversos *dashboards* con gráficos, tablas e indicadores relevantes que ayudaran al Tester a tomar decisiones respecto al diseño y usabilidad de su software. Estos dashboards pueden ser visualizados durante todo el desarrollo del estudio, proporcionando datos en tiempo real.

Los paneles de control o *dashboards* están divididos en 4 categorías, tanto para la visual general del estudio como para el detalle de la iteración, en función del tipo de información que se desea representar. Estas categorías son:

1. **General:** Presenta información relacionada con el tiempo y el éxito de las tareas, incluyendo tiempos promedio, máximos y mínimos, así como porcentajes de éxito y fracaso en las tareas (ver figura 29).
2. **Demográfico:** Describe a los usuarios que participaron en el estudio, incluyendo la cantidad total de usuarios y su distribución porcentual según el rango de edad y el nivel de conocimiento tecnológico (ver figura 30).
3. **Usabilidad:** Muestra los datos procesados del cuestionario CSUQ, tales como el promedio SUS Score, los promedios de las diversas categorías, así como indicadores específicos por pregunta, como máximos, mínimos y promedios, y la distribución de las respuestas mediante un gráfico tipo *Boxplot* (ver figura 31).
4. **Sentimiento y Opiniones:** Despliega información relacionada con el análisis de sentimiento aplicado a las opiniones de los usuarios, incluyendo el sentimiento general, el score promedio, la distribución porcentual de las opiniones según el sentimiento identificado, además de presentar una «nube de tags» para visualizar las palabras más utilizadas en las opiniones (ver figura 32).

Cabe destacar que estos 4 tipos de *dashboards* se encuentran disponibles tanto para la vista resumen del estudio, como para el detalle de cada iteración. En el primer caso, el objetivo es poder visualizar la perspectiva general del estudio y realizar una comparativa rápida entre las distintas iteraciones realizadas, mientras que el segundo caso es poder obtener un detalle específico para cada iteración en caso de necesitar un análisis más detallado.

DISEÑO E IMPLEMENTACIÓN DE UN SOFTWARE BASADO EN ANÁLISIS DE SENTIMIENTO PARA LOGRAR CONFORMIDAD EN CALIDAD Y MADUREZ DE UN SISTEMA

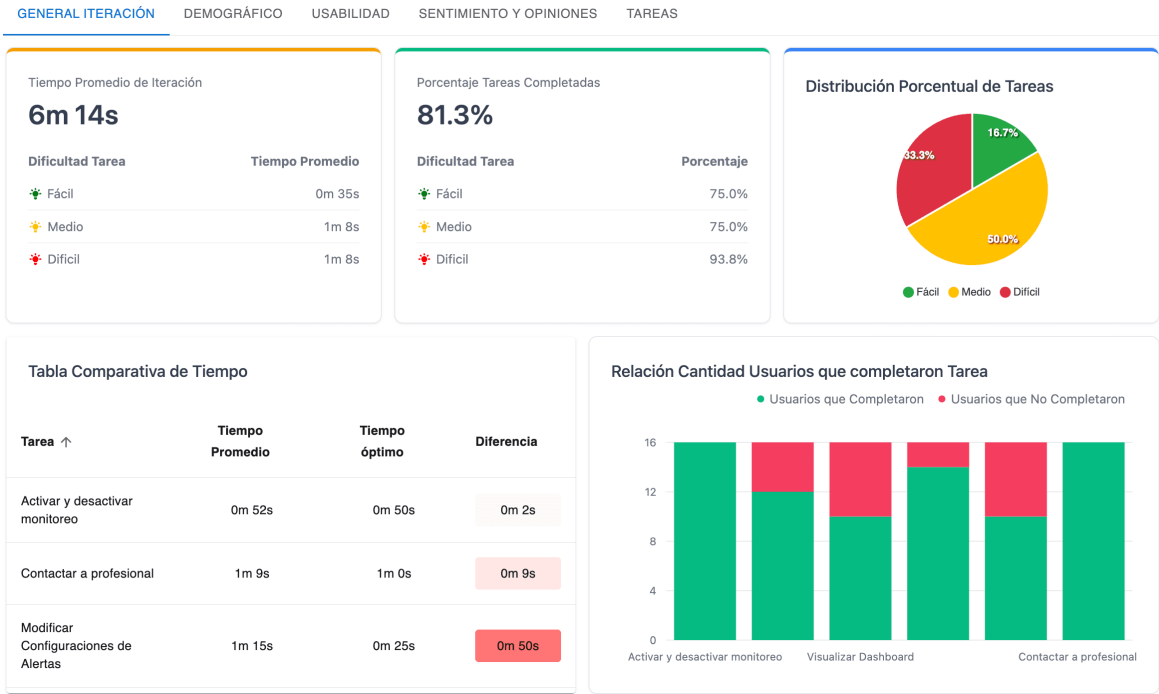


Figura 29: Dashboard general para una iteración particular.  
Fuente: Elaboración propia.

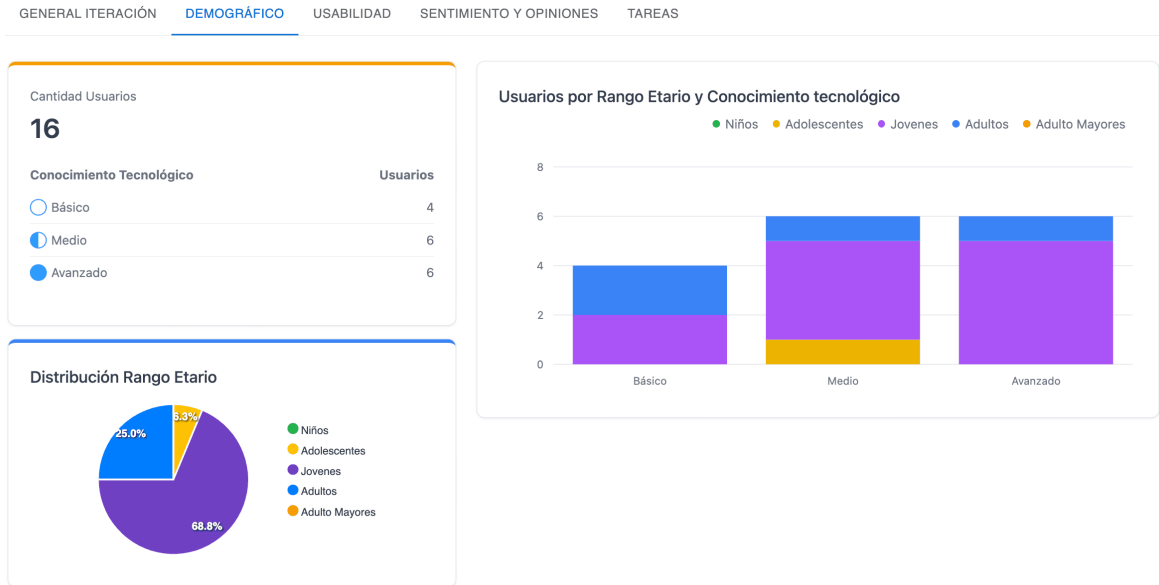


Figura 30: Dashboard demográfico para una iteración particular.  
Fuente: Elaboración propia.

# DISEÑO E IMPLEMENTACIÓN DE UN SOFTWARE BASADO EN ANÁLISIS DE SENTIMIENTO PARA LOGRAR CONFORMIDAD EN CALIDAD Y MADUREZ DE UN SISTEMA

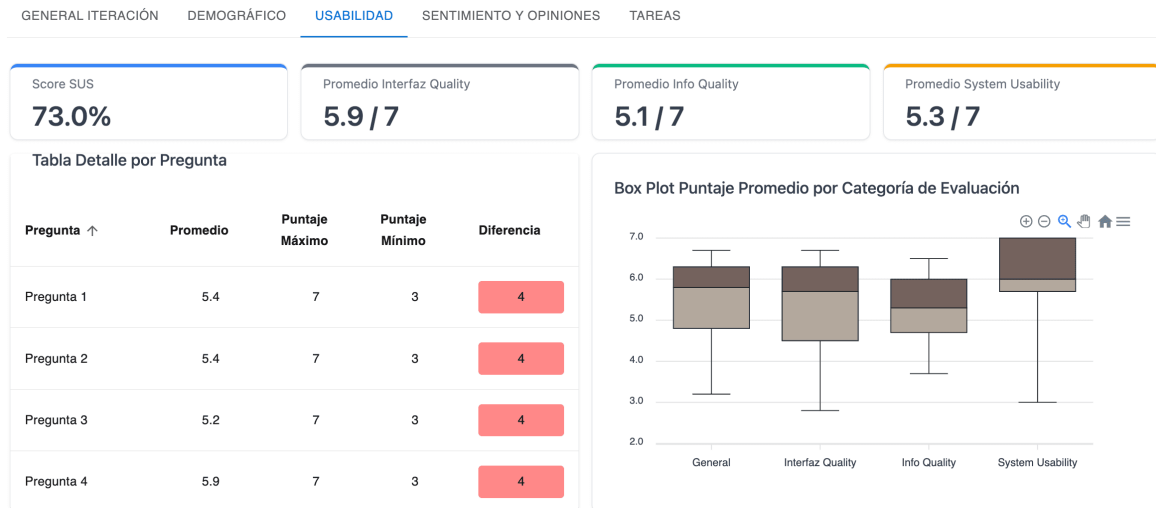


Figura 31: Dashboard de Usabilidad para una iteración particular.

Fuente: Elaboración propia.

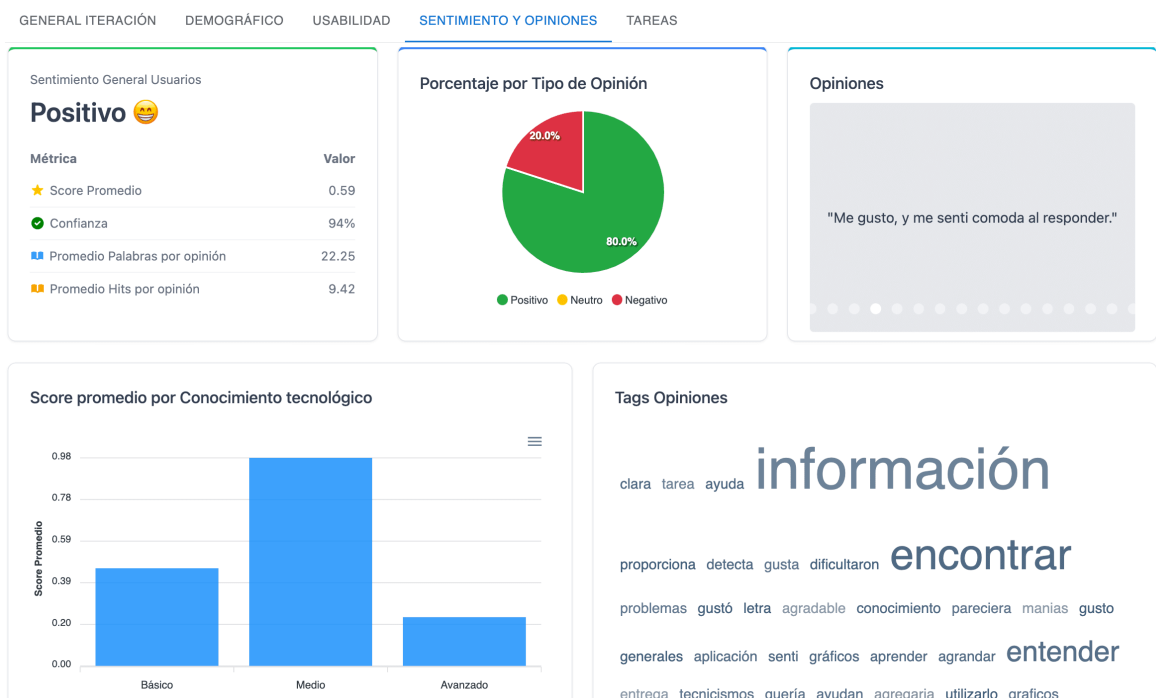


Figura 32: Dashboard de Sentimiento para una iteración particular.

Fuente: Elaboración propia.

### 3.4. Definición previa para Testing de Usabilidad

#### 3.4.1. Contextualización de software testeado

El software empleado para validar la metodología propuesta en esta memoria es **Desk Habitum**. Desarrollado en el contexto de la feria de software de la UTFSM en el año 2022, Desk Habitum es un aplicación de escritorio que busca ayudar a estudiantes y trabajadores a combatir manías compulsivas y malos hábitos que afectan a la salud en situaciones de estrés al momento de estudiar o trabajar frente al computador.

Esta aplicación utiliza la cámara web de la computadora para detectar, mediante modelos pre-entrenados de *machine learning*, diversas posturas y acciones que se consideran manías compulsivas o malos hábitos, entre los cuales se incluyen:

- **Onicofagia**, o morder compulsivamente las uñas.
- **Tricotilomanía**, o jalar el cabello hasta arrancarlo.
- **Dermatilomanía**, o pellizcar la piel de forma compulsiva.
- **Rinotilexomanía**, o hurgarse la nariz.
- **Malos hábitos**, como mala postura, morder objetos no comestibles o fatiga visual.

Entre las principales funcionalidades de la aplicación, se destacan:

- **Reconocimiento en tiempo real**: Desk Habitum utiliza las notificaciones nativas del sistema operativo para poder alertar al usuario cada vez que reconoce al usuario realizando algún mal hábito o manía compulsiva.
- **Visualización de gráficos**: Dada la gran cantidad de datos que generan los usuarios, la aplicación permite al usuario ver información relevante de su comportamiento frente al computador mediante *dashboards*, con el fin de evaluar su evolución y poder mejorar su comportamiento.
- **Ejercicios de respiración**: Cuando el usuario se encuentra muy estresado, Desk Habitum dispone de una sección de respiración profunda, donde podrá realizar diversos ejercicios para poder ayudar a calmar la ansiedad del usuario.
- **Personalización**: Es posible configurar el tiempo de notificación, los sonidos de la notificación, el tipo de manía o mal hábito que reconocerá la aplicación, entre otros.
- **Contacto con profesionales**: La aplicación cuenta con el contacto de profesionales de la salud, para ayudar a los usuarios cuando su problema requiere de atención médica profesional.

### 3.4.2. Definición de usuario

Para definir al usuario de Desk Habitum, se describirán tres áreas clave: su perfil demográfico, comportamiento y necesidades o deseos. Estas dimensiones nos ofrecen una visión completa de quiénes son los usuarios, cómo interactúan con la tecnología y qué esperan de la aplicación.

1. **Perfil demográfico:** Estudiante universitario o trabajador, con edades comprendidas entre los 18 y los 65 años, y posee conocimientos básicos en informática para utilizar aplicaciones de escritorio como software antivirus o de productividad. Dedicar varias horas al día al estudio o al trabajo utilizando el computador.
2. **Comportamiento:** Manifiesta su estrés mordiendo las uñas y se frota los ojos cuando está cansado. Durante el semestre, estudia intensamente, especialmente en las semanas de exámenes. Ha intentado dejar de morderse las uñas utilizando diversos métodos, pero ninguno ha sido efectivo.
3. **Necesidades o deseos:** Busca dejar de morderse las uñas debido a problemas de salud asociados. Además, aspira a mantener un buen rendimiento académico y laboral sin comprometer su salud mental.

En resumen, el usuario de la aplicación es un estudiante o trabajador que presenta algún mal hábito o manía compulsiva y pasa varias horas al día utilizando un ordenador para trabajar o estudiar.

### 3.4.3. Definición de tareas

Uno de los pasos más importantes previo a la evaluación de usabilidad, es la correcta definición de tareas que los usuarios deberán realizar en las distintas iteraciones de testing de usabilidad de la aplicación Desk Habitum.

Dado que el objetivo de las iteraciones de testing, en el marco de esta investigación, es poder mejorar de forma integral la experiencia de usabilidad del usuario, se han diseñado **6 tareas** que abarcan las **funcionalidades más importantes** del software evaluado, discutidas en la sección 3.4.1). Estas tareas se describen en detalle en la tabla 3, donde se especifica el orden de realización, el título y el criterio de éxito utilizado para determinar si el usuario completó o no la tarea.

Cabe destacar que a lo largo de las tres iteraciones, se realizaron ajustes menores en algunos criterios de éxito, sin perjudicar el objetivo de la tarea. Esto se hizo para evitar que el estudio se vea influenciado por las personas que participan en iteraciones consecutivas.

A continuación, se detallan las tareas que fueron modificadas en la iteración número 2:

- **Obtener resumen de última sesión:** La modificación en este criterio de éxito corresponde al cambio en los indicadores consultados. En este caso, el usuario debe señalar la cantidad de detecciones totales y las detecciones acumuladas a la mitad de la sesión en el dashboard resumen de última sesión.
- **Visualizar dashboard histórico:** Al igual que el anterior, el cambio en el criterio corresponde al cambio en los indicadores consultados al usuario. En este caso, consiste en encontrar los dashboards históricos y mencionar la principal manía compulsiva histórica del usuario y la tendencia del mal hábito de morder objetos.
- **Realizar técnica de respiración profunda:** El único cambio que sufrió esta tarea fue el tipo de técnica a emplear, modificando la técnica «Pranayama» por la técnica «Equal».

En el siguiente listado, se señalan las tareas y las modificaciones en sus criterios de éxito para la iteración número 3:

- **Obtener resumen de última sesión:** La modificación en este criterio de éxito corresponde al cambio en los indicadores consultados. En este caso, el usuario debe señalar la cantidad de detecciones de Onicofagia y las detecciones acumuladas a 3/4 del tiempo total de la sesión en el dashboard resumen de última sesión.
- **Visualizar dashboard histórico:** Al igual que el anterior, el cambio en el criterio corresponde al cambio en los indicadores consultados al usuario. Para esta tarea, consiste en encontrar los dashboards históricos y mencionar el tiempo promedio por sesión de la principal manía del usuario y la tendencia del mal hábito de fatiga visual.
- **Modificar configuraciones de alertas:** La modificación en este criterio de éxito implica ajustar el intervalo de notificaciones a cada minuto en lugar de cada cuatro reconocimientos.

Como complemento a la definición de las tareas, se ha establecido la dificultad y el tiempo óptimo a cada una de ellas (ver detalle en tabla 4 y 5 respectivamente). La dificultad de la tarea considera la cantidad de *clícs* necesarios para completarla, el nivel de información disponible y si es una tarea común en aplicaciones similares. Por otro lado, el criterio utilizado para establecer el tiempo óptimo es multiplicando por tres el tiempo que un experto en la aplicación demoraría en completarla.

Finalmente, con el software, los usuarios y las tareas definidas para la evaluación de usabilidad, se puede avanzar a la siguiente etapa correspondiente a las iteraciones de mejora continua.

| Nº Tarea | Título                                        | Criterio de Éxito                                                                                                                                                                                                 |
|----------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1        | Activar y desactivar el monitoreo de hábitos. | El usuario logra presionar botón para comenzar monitoreo, y al cabo de unos segundos, logra detener el monitoreo. Todo esto en la página principal de la aplicación                                               |
| 2        | Obtener resumen de última sesión              | Logra señalar la duración de la última sesión y menciona el porcentaje de tiempo de onicofagia de la última sesión                                                                                                |
| 3        | Visualizar Dashboard Histórico                | Logra navegar a la sección reportes, manías compulsivas, y en el apartado general logra señalar la cantidad total de detecciones y el detalle de porcentajes de tiempo en la sesión 2.                            |
| 4        | Realizar técnica de respiración profunda      | Logra navegar a la sección Herramientas, Respiración profunda, y selecciona la técnica Pranayama. Luego debe pulsar el botón play para activar la música y por último, empezar la técnica con el botón “empezar”. |
| 5        | Modificar Configuraciones de Alertas          | Logra navegar a la sección Configuraciones, y en el apartado “Notificaciones” desactiva la alerta sonora de manía y notificación cada 4 reconocimientos.                                                          |
| 6        | Contactar a profesional                       | Logra navegar a la sección “Contactar profesional”, visualiza al profesional “Joaquín Liras”, pulsa el botón contactar, rellena el formulario y envía.                                                            |

Tabla 3: Detalle de tareas a realizar para evaluación de usabilidad de Desk Habitum.  
Fuente: Elaboración propia.

| Nº Tarea | Tarea                                         | Iteración 1 | Iteración 2 | Iteración 3 |
|----------|-----------------------------------------------|-------------|-------------|-------------|
| 1        | Activar y desactivar el monitoreo de hábitos. | Medio       | Fácil       | Fácil       |
| 2        | Obtener resumen de última sesión              | Fácil       | Medio       | Medio       |
| 3        | Visualizar Dashboard Histórico                | Medio       | Medio       | Medio       |
| 4        | Realizar técnica de respiración profunda      | Difícil     | Difícil     | Difícil     |
| 5        | Modificar Configuraciones de Alertas          | Medio       | Medio       | Medio       |
| 6        | Contactar a profesional                       | Difícil     | Difícil     | Difícil     |

Tabla 4: Detalle de dificultad de las tareas en las distintas iteraciones.  
Fuente: Elaboración propia.

| Nº Tarea | Tarea                                         | Iteración 1   | Iteración 2   | Iteración 3   |
|----------|-----------------------------------------------|---------------|---------------|---------------|
| 1        | Activar y desactivar el monitoreo de hábitos. | 0 min 50 seg. | 0 min 18 seg. | 0 min 18 seg. |
| 2        | Obtener resumen de última sesión              | 0 min 20 seg. | 0 min 25 seg. | 0 min 25 seg. |
| 3        | Visualizar Dashboard Histórico                | 0 min 22 seg. | 0 min 32 seg. | 0 min 32 seg. |
| 4        | Realizar técnica de respiración profunda      | 0 min 40 seg. | 0 min 55 seg. | 0 min 55 seg. |
| 5        | Modificar Configuraciones de Alertas          | 0 min 25 seg. | 0 min 35 seg. | 0 min 35 seg. |
| 6        | Contactar a profesional                       | 1 min 0 seg.  | 1 min 0 seg.  | 1 min 0 seg.  |

Tabla 5: Detalle de duración óptima de las tareas en las distintas iteraciones.  
Fuente: Elaboración propia.

### 3.5. Primera iteración de mejora continua

#### 3.5.1. Resumen demográfico de participantes

En la primera iteración, participaron un total de 16 personas, distribuidas de manera casi equitativa en relación al nivel de conocimiento tecnológico, tal como se puede ver en el detalle de la tabla 6. El nivel básico fue el menos representado, mientras que el nivel medio y avanzado contaron con la misma cantidad de participantes.

Por el contrario, existe una amplia diferencia en cuanto al rango etario. El grupo joven fue el predominante en esta iteración, que abarca a personas entre 18 y 35 años, seguido por el grupo adulto, que comprende a personas de 35 a 65 años. Los detalles de los participantes por rango etario están disponibles en la tabla 7.

| Conocimiento Tecnológico | Cantidad |
|--------------------------|----------|
| Básico                   | 4        |
| Medio                    | 6        |
| Avanzado                 | 6        |

Tabla 6: Detalle de cantidad de usuarios por conocimiento tecnológico en Iteración 1.  
Fuente: Elaboración propia.

| Rango Etario          | Cantidad |
|-----------------------|----------|
| Joven (18 a 35 años)  | 12       |
| Adulto (35 a 65 años) | 4        |

Tabla 7: Detalle de cantidad de usuarios por rango etario en Iteración 1.  
Fuente: Elaboración propia.

### 3.5.2. Primer rediseño basado en feedback y observación

Tras observar el comportamiento de los usuarios frente a las tareas ejecutadas en la aplicación, además de analizar sus comentarios finales respecto a esta primera iteración, se identificaron diversos hallazgos vinculados a la usabilidad del software, que representan valiosas oportunidades de mejora.

El primer cambio se centró en el **rediseño de la pantalla principal** y la **eliminación de la barra lateral de navegación**. Durante las tareas realizadas, se observó que a los usuarios les resultaba difícil encontrar las distintas secciones debido a las múltiples opciones presentes en el *sidebar*, lo que generaba problemas de navegación (ver figura 33).

Por esta razón, se optó por una nueva versión de la pantalla principal, con secciones claramente definidas en la parte inferior, eliminando la navegación lateral para mejorar la accesibilidad a las funcionalidades de la aplicación (ver figura 34).

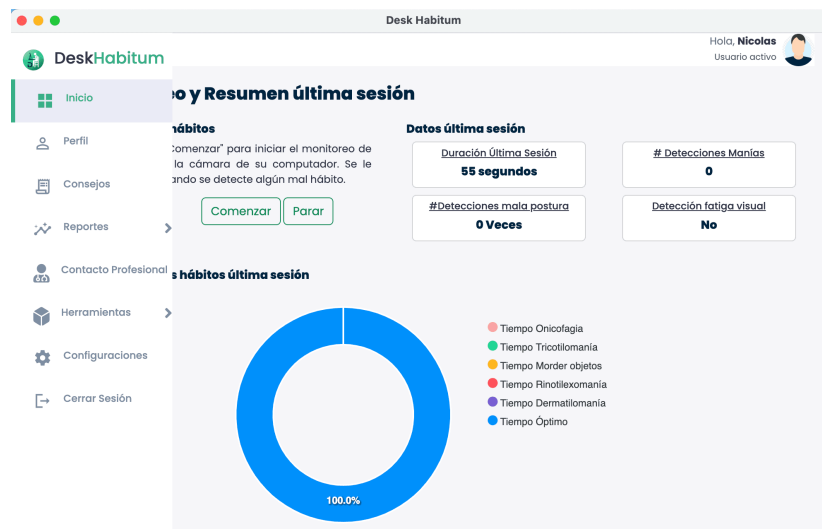


Figura 33: Vista principal de la primera versión de Desk Habitum.  
Fuente: Elaboración propia.

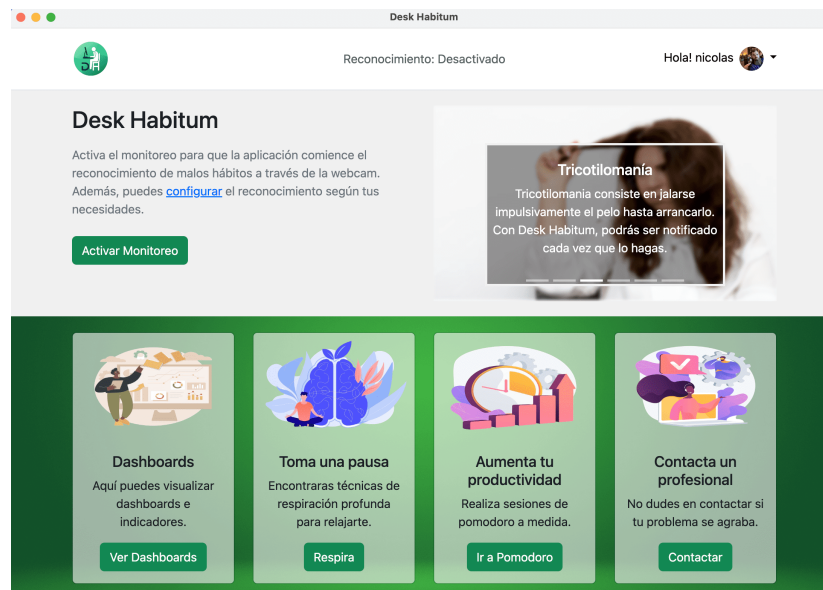


Figura 34: Vista principal de la segunda versión de Desk Habitum.  
Fuente: Elaboración propia.

El segundo cambio implicó un **nuevo ordenamiento de los dashboards** y la **creación de nuevos gráficos**. Durante las tareas de visualización de gráficos, se observó que los usuarios se sentían confundidos al encontrar los gráficos de la última sesión en la pantalla principal, mientras que otros estaban ubicados en una sección aparte. Además, algunos gráficos carecían de leyendas o títulos, lo que dificultaba la interpretación de los datos y generaba frustración (ver figura 35).

Para abordar estas dificultades, se decidió agrupar todos los dashboards en una única ventana con tres secciones: resumen de la última sesión, histórico de manías y histórico de malos hábitos. Además, se eliminaron los gráficos e indicadores que no proporcionaban información relevante, con el objetivo de reducir el exceso de información presente (ver figura 36).



Figura 35: Vista de dashboard, con múltiples secciones en Desk Habitum v1.  
Fuente: Elaboración propia.

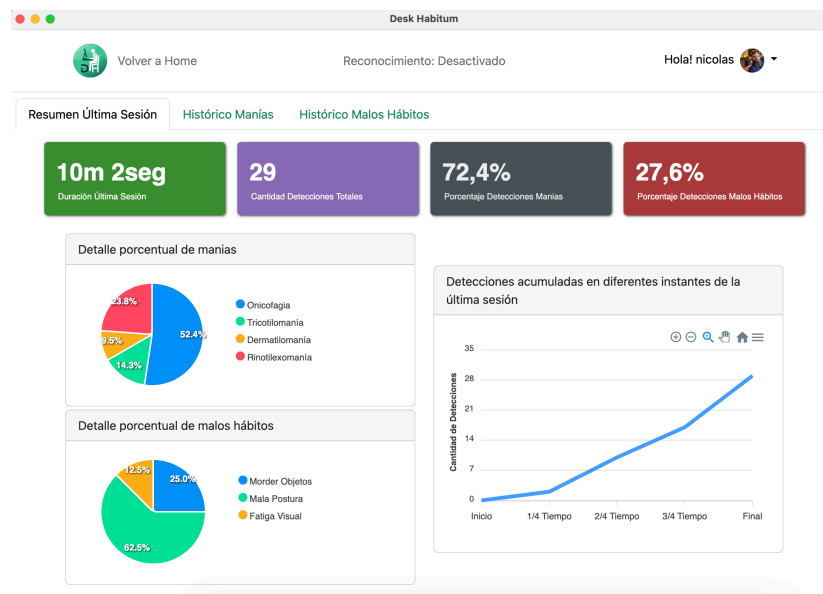


Figura 36: Vista de dashboard, con secciones resumidas en Desk Habitum v2.  
Fuente: Elaboración propia.

El tercer cambio corresponde a la **implementación de alertas e información en pantalla sobre el estado de la aplicación**. Los usuarios manifestaron confusión al presionar botones que parecían no responder debido al tiempo que la aplicación tardaba en ejecutar la acción, lo que llevaba a pensar que el botón no había sido pulsado.

Para solucionar este problema, se implementó una animación de carga en todos los botones cada vez que se pulsan, y se agregaron notificaciones para informar al usuario sobre el estado de la acción, ya sea de éxito o error. Además, se incluyó una confirmación para cerrar la aplicación y se utilizaron ventanas emergentes para confirmar acciones que implicaban un tiempo de espera prolongado, como se ilustra en la figura 37.

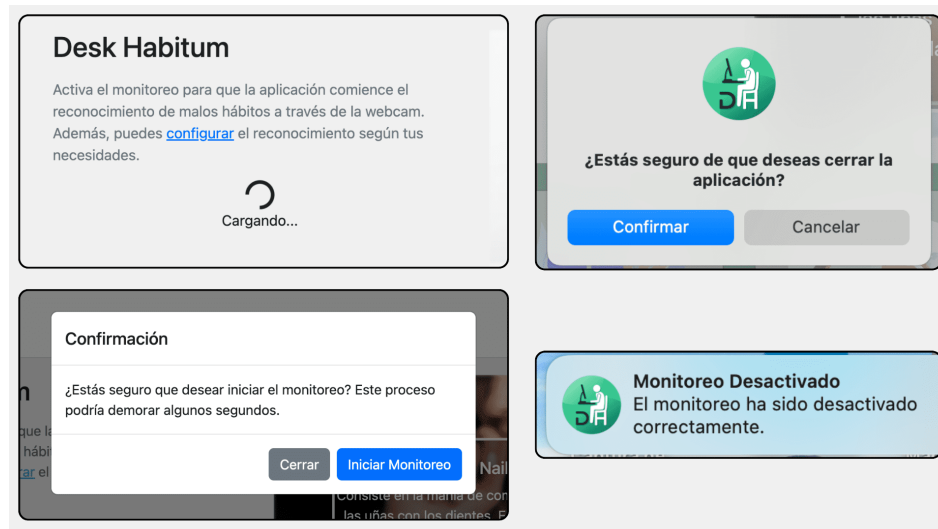


Figura 37: Múltiples alertas de confirmación e información del estado del sistema.

Fuente: Elaboración propia.

Otros ajustes adicionales fueron realizados en la aplicación para mantener la coherencia entre las diferentes secciones y para proporcionar información adicional sobre la aplicación y conceptos clave (para visualizar los cambios realizados, revisar anexo 5.3). Algunos de estos cambios incluyen:

- Rediseño de la pantalla de respiración profunda, modificando colores y animaciones para una mejor experiencia de relajación.
- Rediseño de la pantalla de contacto profesional, reordenando la información respecto a los profesionales disponibles para mejorar la búsqueda.
- Rediseño de la pantalla de configuraciones, agregando un tablero de opciones para mejorar la experiencia de navegación dentro de esta sección.
- Visualización de información relevante en pantalla principal, a través de un *carousel* de imágenes y texto sobre las manías compulsivas y malos hábitos.

A continuación, se presentan algunas opiniones de los usuarios que participaron en esta iteración y que contribuyeron a identificar las mejoras a implementar en la aplicación:

- «Me gusta lo que hace la aplicación pero me costó mucho encontrar lo que quería hacer y hay mucha información en pantalla.»
- «El software cumple su función, aunque hay algunas opciones o funciones del mismo que no son tan intuitivas.»
- «Un software innovador, fácil de entender y poder buscar lo que se necesita, lo que recomendaría es poner información sobre cada manías compulsivas.»
- «Tuve problemas con la visualización de los gráficos, y fue difícil poder encontrar los distintos elementos solicitados.»
- «En la primera tarea, el software pareciera que no responde. Se podría mejorar, pero es un buen software.»
- «Amigable con el usuario, le agregaría las notificaciones al presionar los botones pero en su esencia es sencilla de usar y de entender.»

### 3.6. Segunda iteración de mejora continua

#### 3.6.1. Resumen demográfico de participantes

En la segunda iteración, participaron 16 personas, y en comparación a la primera iteración, la distribución según su nivel de conocimiento tecnológico fue menos equitativa. El nivel avanzado fue el más representado, con la mitad de los participantes, seguido por el nivel básico, que tuvo un poco menos de la mitad de los participantes, y finalmente el nivel medio, con 3 participantes, como se detalla en la tabla 8.

En cuanto a la distribución por rango etario, los jóvenes predominaron en esta iteración, representando casi el 69 % del total de participantes. Le sigue el rango etario adulto, con un 31 % de la participación total (ver detalle de cantidad en la tabla 9). En su mayoría, los participantes son estudiantes universitarios, lo que explica la predominancia del rango joven.

| Conocimiento Tecnológico | Cantidad |
|--------------------------|----------|
| Básico                   | 5        |
| Medio                    | 3        |
| Avanzado                 | 8        |

Tabla 8: Detalle de cantidad de usuarios por conocimiento tecnológico en Iteración 2.

Fuente: Elaboración propia.

| Rango Etario          | Cantidad |
|-----------------------|----------|
| Joven (18 a 35 años)  | 11       |
| Adulto (35 a 65 años) | 5        |

Tabla 9: Detalle de cantidad de usuarios por rango etario en Iteración 2.

Fuente: Elaboración propia.

### 3.6.2. Segundo rediseño basado en feedback y observación

Después de llevar a cabo el segundo test de usabilidad, con las mejoras de la primera iteración ya implementadas, se identificaron ciertos aspectos importantes a resolver dentro de la aplicación.

En términos generales, la nueva versión de la aplicación fue recibida de forma positiva por los usuarios. Sin embargo, el primer problema de interfaz surgió cuando un usuario que suele utilizar aplicaciones de escritorio en pantalla completa notó que **los elementos de Desk Habitum no se ajustaban correctamente al cambiar el tamaño de la ventana**, como se muestra en la pantalla principal ilustrada en la figura 38.

En este sentido, la primera corrección se enfocó en ajustar todas las vistas de la aplicación para que se adapten correctamente a los cambios de tamaño de la ventana. De esta manera, los usuarios disfrutarán de una experiencia visual consistente, ya sea que utilicen sus aplicaciones de escritorio en pantalla completa o no (ver figura 39).

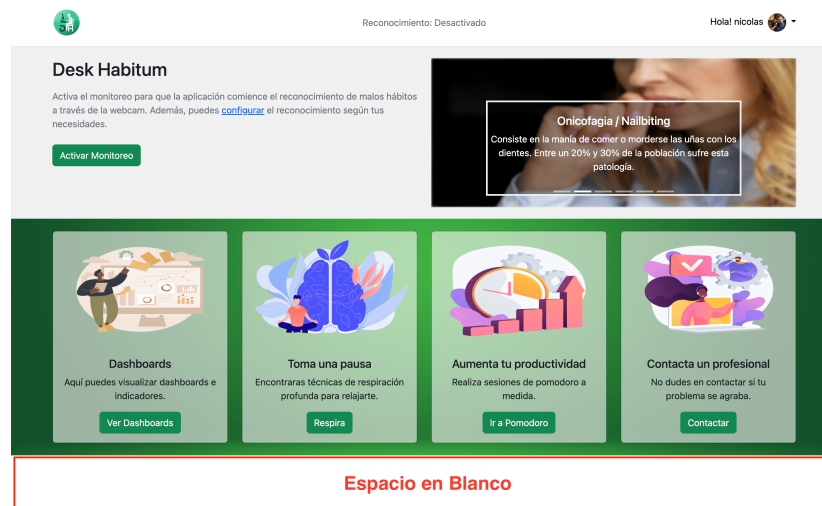


Figura 38: Vista de inicio con error en pantalla completa de Desk Habitum

Fuente: Elaboración propia.

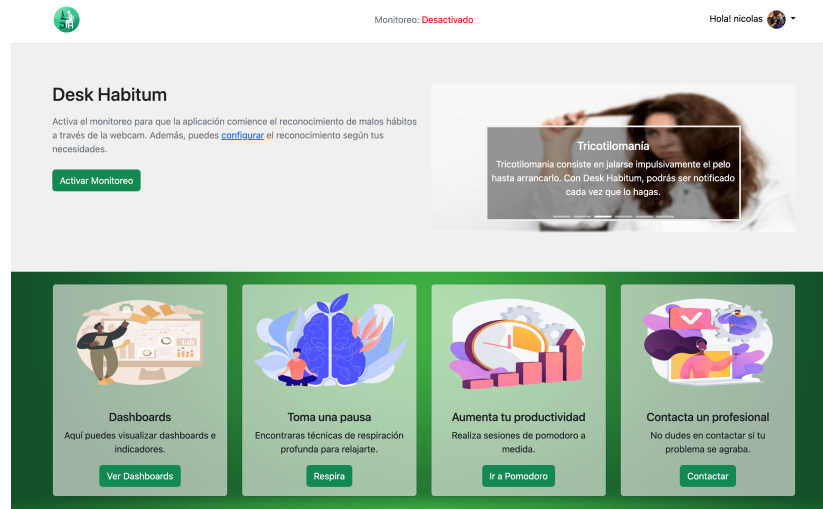


Figura 39: Vista de inicio con elementos adaptados a pantalla completa.  
Fuente: Elaboración propia.

En segundo lugar, se identificaron ciertos **obstáculos relacionados con los gráficos, tanto en el *dashboard* de resumen de la última sesión como en los históricos**. Durante la tarea de visualización de datos de la última sesión, algunos usuarios expresaron confusión con el gráfico de manías acumuladas a lo largo de la sesión debido a la falta de comprensión del eje X. Además, esperaban ver un gráfico de líneas en lugar del gráfico de barras apiladas en la información de tendencia histórica de las manías y malos hábitos.

Por lo tanto, se tomaron en cuenta estos comentarios de los usuarios y se realizaron ajustes. Para el gráfico de la última sesión, se añadió un título descriptivo al eje X y se separaron las líneas de tendencia tanto para manías como para malos hábitos (ver figura 40). En cuanto a los gráficos de tendencia histórica, se cambió el tipo de gráfico de barras apiladas por uno de líneas, como se muestra en la figura 41.

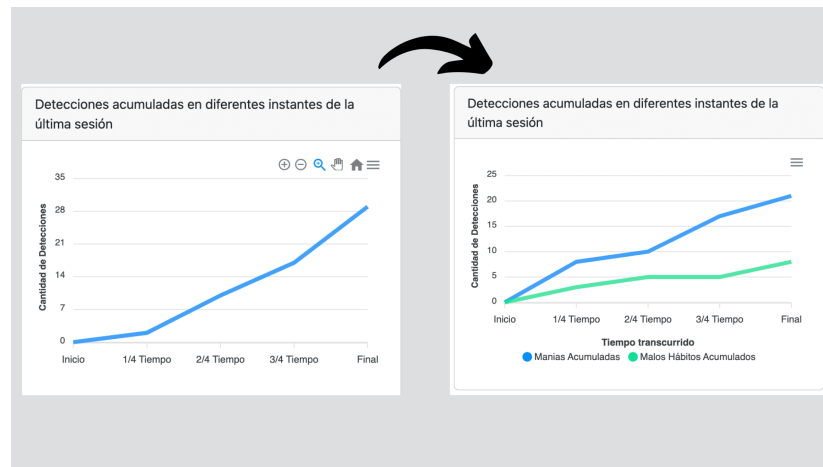


Figura 40: Comparativa de versiones de gráfico acumulado de última sesión.  
Fuente: Elaboración propia.

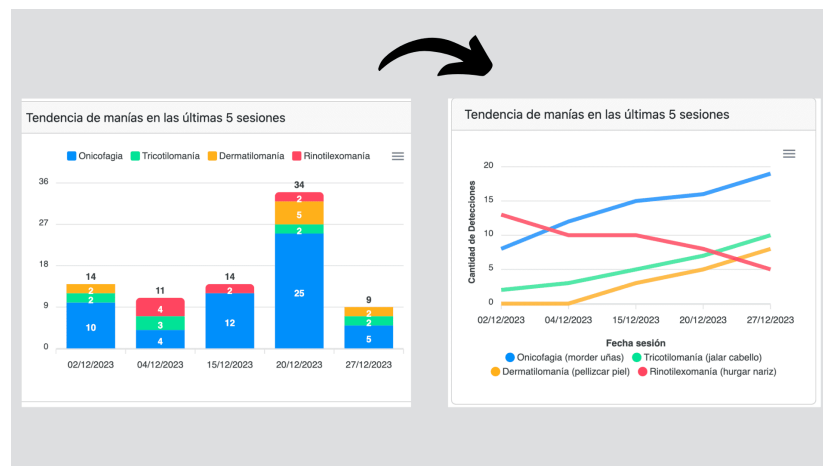


Figura 41: Comparativa de versiones de gráfico de tendencia histórica de manías.  
Fuente: Elaboración propia.

El tercer cambio implementado en la aplicación abordó la **validación de configuraciones y del formulario de contacto profesional**. Se detectó que los usuarios podían guardar cambios con opciones vacías (ver figura 42) y también podían enviar un mensaje al profesional sin completar los campos obligatorios del formulario (ver figura 43).

En respuesta, se tomaron medidas para solucionar estos problemas de cara a la tercera iteración. Se estableció una opción por defecto en las configuraciones y se bloquearon las demás opciones para que el usuario no pueda acceder a ellas a menos que las seleccione. Además, se añadió validación de datos al formulario de contacto profesional (ver figura 44 y 45 respectivamente).

Configuraciones

General

Notificaciones

Sonido

Pomodoro

Notificarme por

Aquí puedes elegir el intervalo con que se te notificará. Ajustalo según tus necesidades.

☐ Cantidad de reconocimientos Selecciona Intervalo

☐ Tiempo cada 10 minutos

Guardar Cambios

Figura 42: Configuración de notificaciones con opción de radio button vacía.  
Fuente: Elaboración propia.

volver a seleccionar profesional.' The form has four input fields: 'Nombre' (Ingresar tu nombre), 'Apellido' (Ingresar tu apellido), 'Correo electrónico' (Ingresar tu correo), and 'Tipo de consulta' (Elige una opción). Below these is a large text area for 'Motivo de la consulta'. At the bottom center is a circular loading spinner with the text 'Enviando...' below it."/>

Contacto Profesional

Formulario

Completa el siguiente formulario para contactar al profesional seleccionado. También puedes [volver a seleccionar profesional](#).

Nombre Ingresar tu nombre

Apellido Ingresar tu apellido

Correo electrónico Ingresar tu correo

Tipo de consulta Elige una opción

Motivo de la consulta

Enviando...

Figura 43: Formulario de contacto profesional sin validación de campos.  
Fuente: Elaboración propia.

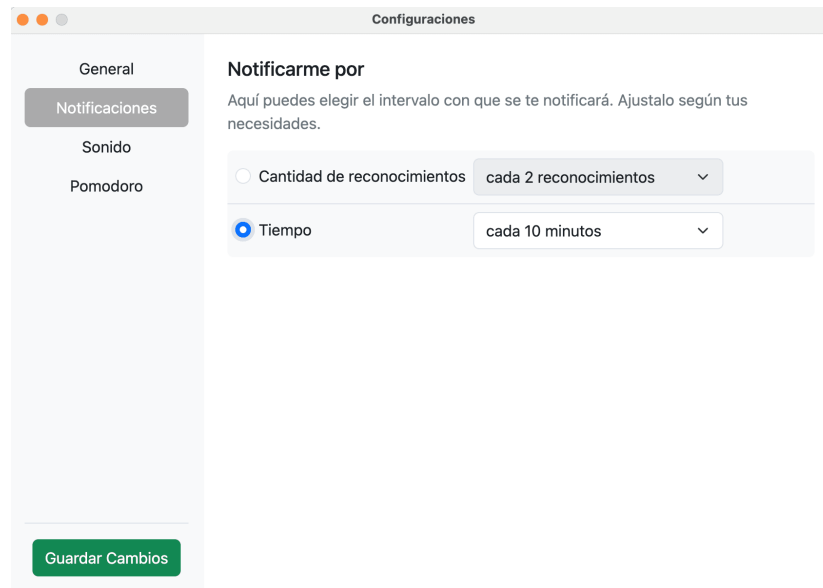


Figura 44: Configuración de notificaciones con opción por defecto y limitación de campos.  
Fuente: Elaboración propia.

Figura 45: Formulario de contacto profesional con validación de campos.  
Fuente: Elaboración propia.

Se realizaron otros cambios menores en la aplicación, en respuesta a observaciones realizadas durante la ejecución de las tareas y a los comentarios en voz alta de los usuarios. Estos cambios se detallan a continuación (para ver los cambios realizados, consultar el anexo 5.4):

- Se implementa *tooltip* a los botones de cada técnica de respiración, para que el usuario conozca en que consiste cada técnica previo a iniciar la sesión de respiración profunda.
- Se agrega opción de contactar profesional al menú rápido de opciones, para facilitar el acceso a esta funcionalidad desde cualquier parte de la aplicación.
- Se implementa cambio de color al texto del estado del monitoreo, siendo verde para estado activo y rojo para estado desactivado.

A continuación, se recopilan algunas opiniones de los usuarios que participaron en esta iteración, las cuales fueron fundamentales para identificar las mejoras a implementar en la aplicación:

- «Me gustó la idea y la buena implementación de ésta, sólo tuve una ligera confusión a la hora de entender los gráficos, pero en general está muy buena.»
- «Me gusto el software, encontré buena la idea que hay detrás, quizás faltó un poco mas de consistencia entre las diferentes pestañas de la aplicación, pero fue muy intuitiva en el uso.»
- «En general creo que hubo un buen crecimiento desde la versión anterior, se siente bien completo, Solo le agregaría las validaciones de correo y ese tipo de cosas.»
- «La ubicación de poder solicitar a un profesional en caso de ayuda esta muy escondida, y no es fácil de encontrar.»
- «Esta segunda versión de la app me pareció mucho mejor que la anterior. Una sugerencia podría ser hacer el desactivado y activado de colores, para que sea mas llamativo el estado. Todo lo demás excelente!»

### 3.7. Tercera iteración de mejora continua

#### 3.7.1. Resumen demográfico de participantes

En la tercera y última iteración, participaron 16 personas. Respecto al nivel de conocimiento tecnológico, el nivel avanzado volvió a ser predominante, con casi la mitad de los participantes, seguido del nivel medio y, por último, el nivel básico, como se detalla en la tabla 10. En comparación con las iteraciones anteriores, esta distribución fue menos equitativa que en la primera, pero más equilibrada que en la segunda.

En esta iteración, los jóvenes predominaron con un 62,5 % de la participación total, seguidos por el grupo etario adulto con un 37,5 % (ver detalle de cantidad en la tabla 11. A lo largo

de las tres iteraciones del estudio, el grupo etario predominante fue el joven y el nivel de conocimiento tecnológico predominante fue el avanzado.

| Conocimiento Tecnológico | Cantidad |
|--------------------------|----------|
| Básico                   | 4        |
| Medio                    | 5        |
| Avanzado                 | 7        |

Tabla 10: Detalle de cantidad de usuarios por conocimiento tecnológico en Iteración 3.  
Fuente: Elaboración propia.

| Rango Etario          | Cantidad |
|-----------------------|----------|
| Joven (18 a 35 años)  | 10       |
| Adulto (35 a 65 años) | 6        |

Tabla 11: Detalle de cantidad de usuarios por rango etario en Iteración 3.  
Fuente: Elaboración propia.

### 3.7.2. Feedback de usuarios y mejoras pendientes

A lo largo de estas tres iteraciones, el feedback de los usuarios ha sido positivo, lo cual se analizará con mayor detalle en el siguiente capítulo. Esta retroalimentación ha sido fundamental para mejorar tanto la calidad como la madurez de Desk Habitum.

A pesar de que la usabilidad del software ha sido testeada en varias ocasiones, siempre existe margen para mejorar. Basándonos en las opiniones recopiladas en esta última iteración y en la observación directa de los usuarios durante la realización de las tareas, se han identificado algunas oportunidades de mejora pendientes, entre las que se incluyen:

- **Opciones de accesibilidad:** Dado que nuestra aplicación atiende a usuarios de diferentes edades y con distintas necesidades, es crucial incorporar opciones que les permitan ajustar el tamaño del texto, activar un modo oscuro o daltónico, y hacer zoom en áreas específicas de la interfaz.
- **Dashboards dinámicos:** Aunque se ha dedicado considerable esfuerzo a la sección de gráficos y paneles de control en las iteraciones anteriores, sería beneficioso ampliar su funcionalidad. Esto podría incluir la implementación de filtros adicionales, la introducción de nuevos tipos de gráficos y la capacidad de personalizar los dashboards según las necesidades específicas de cada usuario.
- **Buscador de profesionales:** Actualmente, la aplicación muestra a los profesionales disponibles a través de tarjetas, pero esta presentación puede volverse poco escalable a

medida que aumenta el número de profesionales disponibles. Por ello, sería una buena opción implementar una visualización en forma de tabla o incluir un buscador por profesión o tipo de atención.

- **Animaciones y transiciones:** Para mejorar la experiencia del usuario en la aplicación, sería beneficioso incorporar animaciones modernas y transiciones fluidas entre las vistas. Sin embargo, es importante utilizarlas de manera moderada y equilibrada para evitar sobrecargar al usuario.

Por último, se presentan algunas de las opiniones recopiladas en la tercera y última iteración de mejora continua:

- «Esta tercera versión del software se nota mejor pulida, con cambios para bien respecto a los gráficos y a los mensajes de error. Muy buen software.»
- «En general me pareció una muy buena aplicación, me hubiera gustado en la sección de profesionales poder haber buscado la especialidad para mas rapidez.»
- «muy buena actualización de la aplicación, ya que le agregaron la validación de formulario como lo sugerí en el test anterior. Agregaría animaciones y transiciones para hacer la experiencia aun mas placentera, en general muy satisfecho!»
- «En esta ocasión fue mas fácil poder encontrar las cosas, y se nota mas consistente la aplicación, me gusto en general. En la parte de respiración profunda, agregaría mas posibilidades de personalización.»
- «Mantengo mi opinión respecto a la versión anterior, encuentro que la UX fue excelente, muy placentera. Quizá una fuente un poco mas moderna ayudaría aun mas, pero en general excelente.»

## CAPÍTULO 4

### VALIDACIÓN DE LA SOLUCIÓN

En este capítulo se presentarán los resultados obtenidos a lo largo de las tres iteraciones de mejora continua, realizando además un análisis comparativo de los datos que permitirán validar la solución propuesta en esta memoria.

#### 4.1. Tiempo y éxito de tareas

El primer aspecto a analizar corresponde al tiempo empleado por los usuarios para realizar las distintas tareas, y el porcentaje de éxito al completarlas.

La tabla 12 muestra los tiempos promedio por tarea en cada iteración. Se observa una disminución en todos los tiempos entre la primera y la tercera iteración, con reducciones de más del 50 % en algunos casos, como en la tarea «Activar y desactivar monitoreo», y en promedio una disminución de 18 segundos por tarea.

| Tarea                                | Iteración 1 | Iteración 2 | Iteración 3 |
|--------------------------------------|-------------|-------------|-------------|
| Activar y desactivar monitoreo       | 52          | 27          | 25          |
| Contactar Profesional                | 69          | 59          | 63          |
| Modificar Configuraciones de Alertas | 75          | 45          | 41          |
| Obtener Resumen Última Sesión        | 35          | 51          | 33          |
| Realizar Sesión Respiración Profunda | 68          | 62          | 58          |
| Visualizar Dashboard                 | 76          | 62          | 43          |

Tabla 12: Tiempo promedio [seg] por tarea para cada iteración  
Fuente: Elaboración propia.

Si comparamos entre la primera y segunda iteración, en casi todas las tareas se logró una disminución en el tiempo promedio. Destaca especialmente la reducción de 30 segundos en la tarea de «Modificar configuraciones de alertas». Sin embargo, también hubo una tarea, «Obtener resumen de última sesión», que registró un aumento en el tiempo promedio de 24 segundos, lo que se puede apreciar gráficamente en la figura 46.

Entre la segunda y tercera iteración, el tiempo promedio disminuyó en casi todas las tareas, a excepción de «Contactar a profesional», que aumentó su tiempo promedio en 4 segundos, aunque sigue siendo menor en comparación al de la primera iteración. En todas las demás tareas, se registró una disminución en el tiempo, donde destaca la tarea de «Visualizar Dashboard», con una reducción de 19 segundos en promedio.

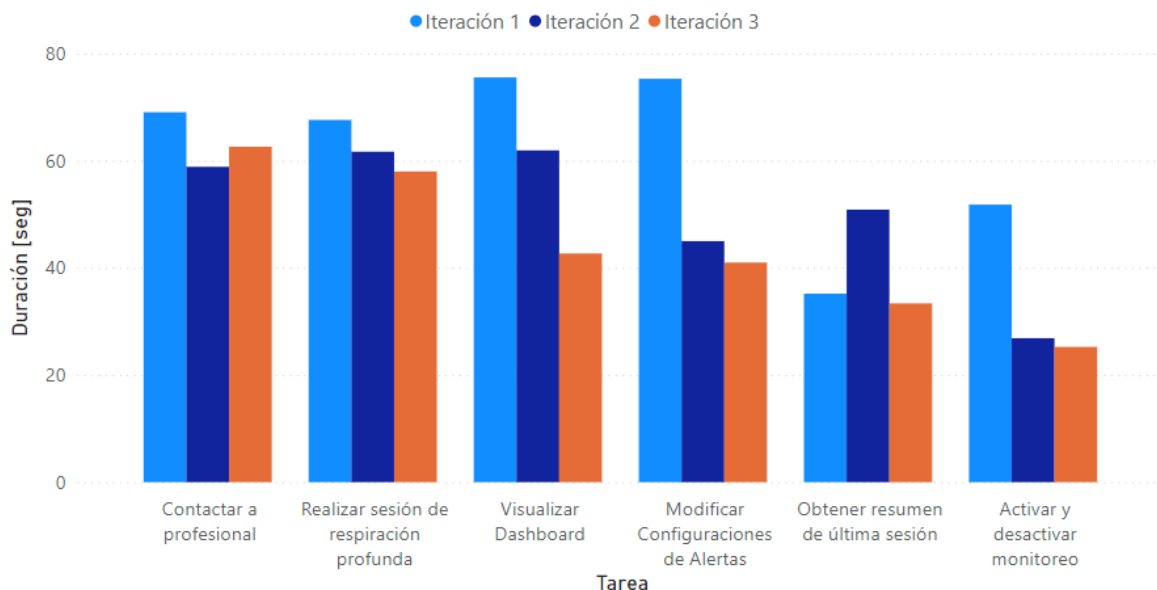


Figura 46: Gráfico de Tiempos promedio por tarea para cada iteración  
Fuente: Elaboración propia.

En cuanto al tiempo promedio por rango etario, se observó un decrecimiento sostenido en las tres iteraciones (ver tabla 13). Se destaca una diferencia significativa en el rango etario adulto, con una reducción de 45 segundos entre la primera y la tercera iteración. Además, es importante destacar la disminución de la brecha de tiempo entre el rango etario joven y adulto, pasando de 39 segundos en la primera iteración a solo 5 segundos en la tercera.

| Rango Etario | Iteración 1 | Iteración 2 | Iteración 3 |
|--------------|-------------|-------------|-------------|
| Adulto       | 92          | 67          | 47          |
| Joven        | 53          | 43          | 42          |

Tabla 13: Tiempo promedio [seg] por rango etario para cada iteración  
Fuente: Elaboración propia.

En relación al tiempo promedio por nivel de conocimiento tecnológico, se observa una disminución en todas las iteraciones (ver tabla 14). La diferencia entre la primera y la tercera iteración es de 39 segundos en el nivel básico, 10 segundos en el nivel medio y 14 segundos en el nivel avanzado. La mayor brecha inicial de tiempo promedio se encuentra entre el nivel básico y medio, con 46 segundos, reduciéndose a 17 segundos en la tercera iteración.

| Conocimiento Tecnológico | Iteración 1 | Iteración 2 | Iteración 3 |
|--------------------------|-------------|-------------|-------------|
| Básico                   | 94          | 68          | 55          |
| Medio                    | 48          | 42          | 38          |
| Avanzado                 | 56          | 44          | 42          |

Tabla 14: Tiempo promedio [seg] por conocimiento tecnológico para cada iteración  
Fuente: Elaboración propia.

La tabla 15 presenta el porcentaje de éxito promedio por tarea en cada una de las tres iteraciones, revelando un incremento en todas las tareas realizadas en el estudio. Destacan las tareas «Contactar Profesional» y «Realizar sesión de Respiración profunda», las cuales, al ser asignadas con una dificultad alta, alcanzaron un 100 % de éxito en 2 y 3 iteraciones respectivamente.

| Tarea                                | Iteración 1 | Iteración 2 | Iteración 3 |
|--------------------------------------|-------------|-------------|-------------|
| Activar y desactivar monitoreo       | 100 %       | 100 %       | 100 %       |
| Contactar Profesional                | 100 %       | 100 %       | 100 %       |
| Modificar Configuraciones de Alertas | 63 %        | 75 %        | 100 %       |
| Obtener Resumen Última Sesión        | 75 %        | 100 %       | 100 %       |
| Realizar Sesión Respiración Profunda | 88 %        | 100 %       | 100 %       |
| Visualizar Dashboard                 | 63 %        | 94 %        | 94 %        |

Tabla 15: Porcentaje de éxito promedio por tarea para cada iteración  
Fuente: Elaboración propia.

Asimismo, las tareas «Visualizar Dashboard» y «Modificar configuraciones de alertas» vieron un aumento en su porcentaje, pasando del 63 % en la primera iteración a un 100 % y 94 % respectivamente, representando los mayores incrementos en el éxito de las tareas. Estos cambios pueden ser visualizados gráficamente en la figura 47

En relación al porcentaje de éxito promedio por rango etario, se observó un aumento tanto para usuarios adultos como jóvenes en las tres iteraciones, como se detalla en la tabla 16. El rango etario adulto experimentó un incremento del 29 % entre la primera y la tercera iteración, mientras que el grupo joven aumentó en un 13 %. Es importante destacar que la brecha inicial de porcentaje de éxito entre los rangos etarios joven y adulto, de un 14 %, se redujo a un 2 % al concluir la tercera iteración.

| Rango Etario | Iteración 1 | Iteración 2 | Iteración 3 |
|--------------|-------------|-------------|-------------|
| Adulto       | 71 %        | 97 %        | 100 %       |
| Joven        | 85 %        | 94 %        | 98 %        |

Tabla 16: Porcentaje de éxito promedio por rango etario para cada iteración  
Fuente: Elaboración propia.

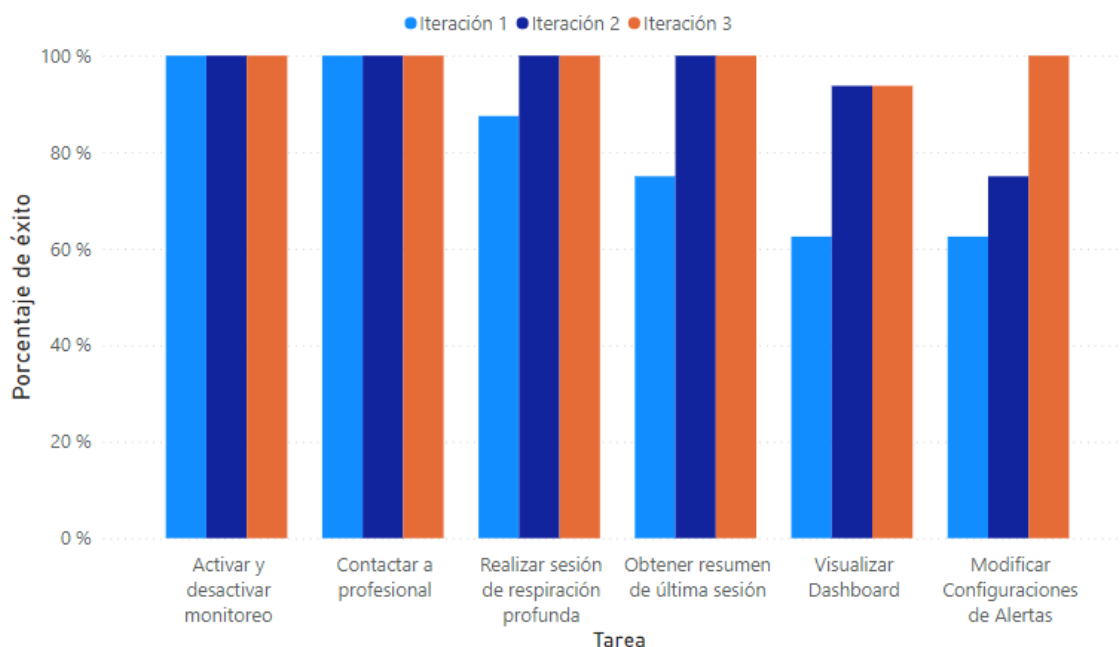


Figura 47: Gráfico de porcentaje de éxito promedio por tarea para cada iteración  
Fuente: Elaboración propia.

Por último, la tabla 17 muestra el porcentaje de éxito promedio según el nivel de conocimiento tecnológico de los usuarios. En las tres iteraciones, se observó un aumento en el éxito de las tareas, con diferencias significativas entre la primera y tercera iteración: un incremento del 29 % en el nivel básico, 17 % en el nivel medio y 12 % en el nivel avanzado. Es interesante analizar que, en la primera iteración, ningún nivel alcanzó más del 86 % de éxito; sin embargo, en la segunda iteración, todos los niveles superaron el 93 % de éxito promedio. Para la tercera iteración, dos de los tres niveles alcanzaron el máximo porcentaje de éxito, siendo el nivel avanzado el único que no lo logró, con un 98 % de éxito.

| Conocimiento Tecnológico | Iteración 1 | Iteración 2 | Iteración 3 |
|--------------------------|-------------|-------------|-------------|
| Básico                   | 71 %        | 93 %        | 100 %       |
| Medio                    | 83 %        | 94 %        | 100 %       |
| Avanzado                 | 86 %        | 96 %        | 98 %        |

Tabla 17: Porcentaje de éxito promedio por conocimiento tecnológico para cada iteración  
Fuente: Elaboración propia.

## 4.2. Cuestionario CSUQ

El segundo punto de análisis corresponde a los resultados obtenidos del cuestionario CSUQ aplicado a los usuarios durante las tres iteraciones de mejora continua.

La tabla 18 muestra los indicadores promedio del cuestionario CSUQ, revelando un aumento en todas las categorías a lo largo de las 3 iteraciones, y con una diferencia en el indicador CSUQ Total de 1,1 puntos. Esto se complementa con los puntajes SUS representados en la tabla 19, donde se alcanzó una calificación B+ en la primera iteración y la calificación máxima, A+, en la segunda y tercera iteración.

| Indicador Promedio        | Iteración 1 | Iteración 2 | Iteración 3 |
|---------------------------|-------------|-------------|-------------|
| CSUQ Total (máx. 7)       | 5,4         | 6,4         | 6,5         |
| Info Quality (máx. 7)     | 5,1         | 6,1         | 6,3         |
| Interfaz Quality (máx. 7) | 5,9         | 6,6         | 6,6         |
| System Usability (máx. 7) | 5,3         | 6,4         | 6,5         |

Tabla 18: Indicadores promedio de cuestionario CSUQ por iteración.  
Fuente: Elaboración propia.

| Indicador Promedio   | Iteración 1 | Iteración 2 | Iteración 3 |
|----------------------|-------------|-------------|-------------|
| Grade                | B+          | A+          | A+          |
| SUS Score (máx. 100) | 73,0        | 89,4        | 90,9        |

Tabla 19: Puntaje SUS obtenido en cada iteración, con su equivalencia en grados.  
Fuente: Elaboración propia.

Entre las diversas categorías incluidas en el cuestionario, cabe destacar el aumento en la percepción de la calidad de la información por parte de los usuarios, reflejado en una diferencia de 1,2 puntos entre la primera y tercera iteración. Asimismo, se observa un aumento en la percepción de la usabilidad general del sistema, pasando de una puntuación de 5,3 en la primera iteración a 6,5 en la tercera iteración.

El gráfico de la figura 48 ilustra tanto el aumento en los puntajes del cuestionario en las distintas categorías como la tendencia de estas categorías a equilibrar sus puntajes promedio a lo largo de las iteraciones. Se observa que, al inicio, existía una diferencia máxima de 0,8 puntos entre dos categorías, la cual disminuye a 0,3 puntos al final de la tercera iteración.

En relación a los indicadores promedio por rango etario de la primera iteración, se observa que el grupo joven percibió de manera más positiva los distintos aspectos de usabilidad de la aplicación en comparación con el grupo etario adulto (ver detalle en tabla 20). La calificación general muestra una diferencia de 0,4 puntos entre ambos grupos, aunque es importante señalar que el grupo adulto evaluó con nota menor a 5,0 tanto la calidad de la información

como la usabilidad general del sistema. Por otro lado, la calidad de la interfaz fue la mejor evaluada, con un puntaje de 5,8 en adultos y 5,9 en jóvenes.

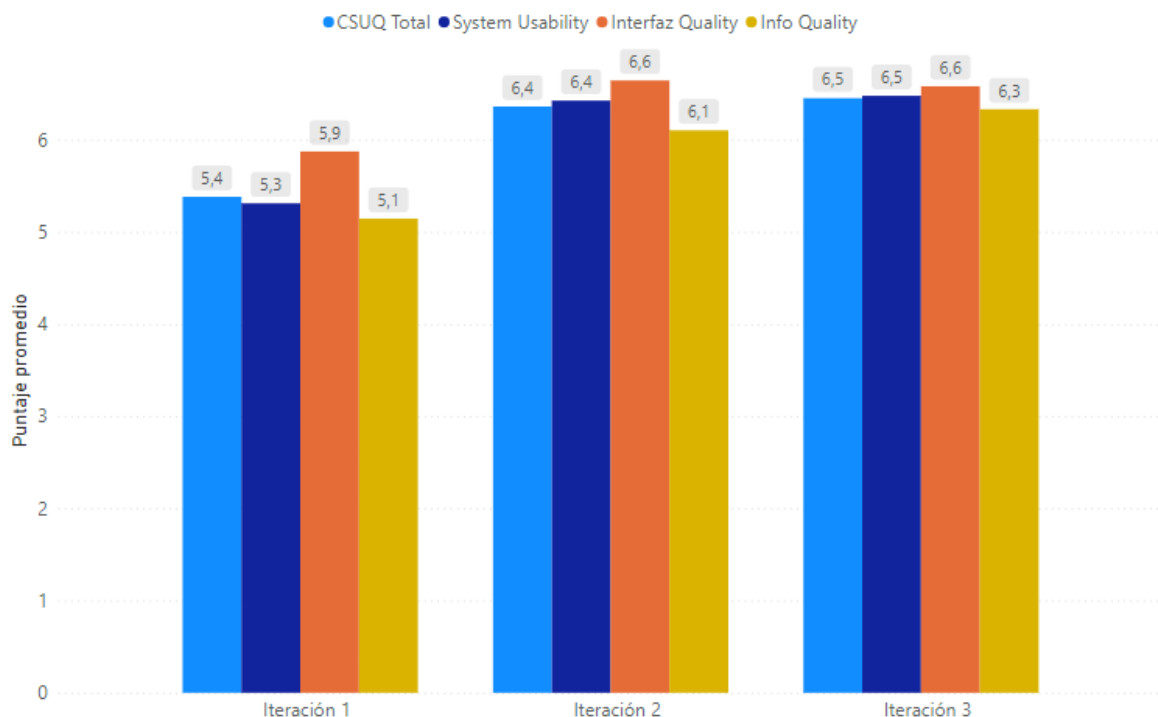


Figura 48: Gráfico de indicadores CSUQ promedio para cada iteración  
Fuente: Elaboración propia.

| Indicador Promedio Iteración 1 | Rango Adulto | Rango Joven |
|--------------------------------|--------------|-------------|
| CSUQ Total (máx. 7)            | 5,1          | 5,5         |
| Info Quality (máx. 7)          | 4,9          | 5,2         |
| Interfaz Quality (máx. 7)      | 5,8          | 5,9         |
| System Usability (máx. 7)      | 4,9          | 5,5         |

Tabla 20: Indicadores promedio de cuestionario CSUQ por rango etario en Iteración 1.  
Fuente: Elaboración propia.

En la segunda iteración, se mantiene una percepción general más positiva por parte del grupo etario joven en comparación con el grupo adulto, con una diferencia de 0,7 puntos en el promedio CSUQ Total, como se muestra en la tabla 21. Sin embargo, en comparación a la primera iteración, el grupo adulto tuvo una percepción más favorable de la aplicación en todos los aspectos evaluados en el cuestionario, con una calificación promedio mínima de 5,9 en comparación al 4,9 de la primera iteración. En cuanto al grupo joven, este mantuvo sus puntajes promedio por encima de 6,2 en todas las categorías, casi llegando a la máxima calificación en la categoría de calidad de interfaz.

| Indicador Promedio Iteración 2 | Rango Adulto | Rango Joven |
|--------------------------------|--------------|-------------|
| CSUQ Total (máx. 7)            | 5,9          | 6,6         |
| Info Quality (máx. 7)          | 5,9          | 6,2         |
| Interfaz Quality (máx. 7)      | 6,1          | 6,9         |
| System Usability (máx. 7)      | 5,9          | 6,7         |

Tabla 21: Indicadores promedio de cuestionario CSUQ por rango etario en Iteración 2.  
Fuente: Elaboración propia.

En la tercera iteración, el grupo etario adulto mantiene un puntaje promedio más bajo en comparación con el grupo joven, aunque con una diferencia menor respecto a la iteración anterior, de 0,5 puntos en el promedio CSUQ Total (ver detalle en tabla 22). Es destacable en esta iteración que los puntajes promedio en todas las categorías superan los 6,1 puntos, con el grupo adulto consiguiendo puntajes promedio similares, mientras que el grupo joven obtuvo una dispersión mayor entre sus categorías, como se ilustra en el gráfico de la figura 49.

| Indicador Promedio Iteración 3 | Rango Adulto | Rango Joven |
|--------------------------------|--------------|-------------|
| CSUQ Total (máx. 7)            | 6,1          | 6,6         |
| Info Quality (máx. 7)          | 6,1          | 6,5         |
| Interfaz Quality (máx. 7)      | 6,1          | 6,9         |
| System Usability (máx. 7)      | 6,2          | 6,7         |

Tabla 22: Indicadores promedio de cuestionario CSUQ por rango etario en Iteración 3.  
Fuente: Elaboración propia.

Con respecto a los indicadores promedio por nivel de conocimiento tecnológico en la primera iteración, los usuarios con nivel básico y avanzado obtuvieron el mismo puntaje promedio, mientras que el nivel medio alcanzó un puntaje superior, con una diferencia de 0,3 puntos respecto a los otros niveles (ver detalle en tabla 23). Para los tres niveles, la categoría de calidad de información fue la peor calificada, mientras que la calidad de la interfaz fue la mejor calificada, que además fue la única categoría que obtuvo un puntaje sobre 6,0 correspondiente al nivel medio; todas las demás se mantuvieron entre 5,0 y 5,9 puntos.

En la segunda iteración, todos los puntajes aumentaron en al menos 0,5 puntos en comparación a la iteración anterior, superando todos los 6,0 puntos de calificación como se visualiza en la tabla 24. Sin embargo, el orden en el puntaje promedio total entre los niveles experimentó cambios: el puntaje más alto fue para el nivel avanzado, seguido del nivel medio y, por último, el nivel básico. La categoría de calidad de la información sigue siendo la más baja para los niveles medio y avanzado, mientras que la usabilidad del sistema fue la más baja para el nivel básico. La calidad de la interfaz continuó siendo la mejor calificada en los tres niveles, al igual que en la iteración anterior.

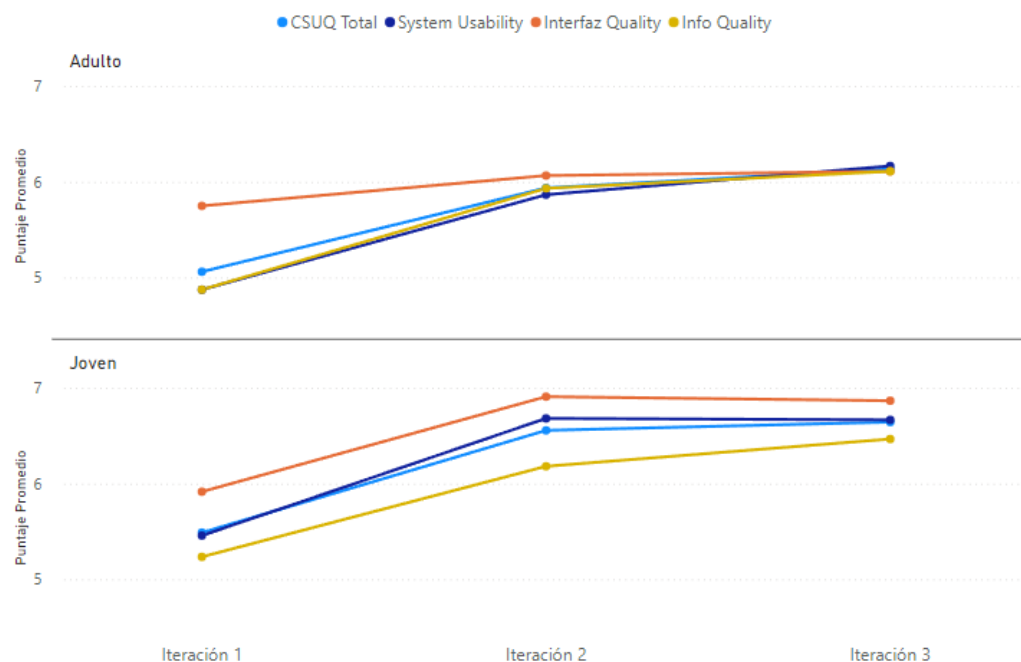


Figura 49: Gráfico de indicadores CSUQ promedio por rango etario en cada iteración.  
Fuente: Elaboración propia.

| Indicador Promedio Iteración 1 | Nivel Básico | Nivel Medio | Nivel Avanzado |
|--------------------------------|--------------|-------------|----------------|
| CSUQ Total (máx. 7)            | 5,3          | 5,6         | 5,3            |
| Info Quality (máx. 7)          | 5,0          | 5,3         | 5,1            |
| Interfaz Quality (máx. 7)      | 5,8          | 6,1         | 5,7            |
| System Usability (máx. 7)      | 5,2          | 5,6         | 5,1            |

Tabla 23: Indicadores cuestionario CSUQ por conocimiento tecnológico en Iteración 1.  
Fuente: Elaboración propia.

| Indicador Promedio Iteración 2 | Nivel Básico | Nivel Medio | Nivel Avanzado |
|--------------------------------|--------------|-------------|----------------|
| CSUQ Total (máx. 7)            | 6,1          | 6,3         | 6,5            |
| Info Quality (máx. 7)          | 6,1          | 6,1         | 6,1            |
| Interfaz Quality (máx. 7)      | 6,3          | 6,7         | 6,9            |
| System Usability (máx. 7)      | 6,0          | 6,3         | 6,7            |

Tabla 24: Indicadores cuestionario CSUQ por conocimiento tecnológico en Iteración 2.  
Fuente: Elaboración propia.

Entre la segunda y tercera iteración, no se observaron cambios significativos, aunque la categoría de calidad de la interfaz experimentó leves descensos en sus puntajes, de 0,1 y 0,2

puntos en los niveles básico y medio respectivamente. En las demás categorías, los puntajes se mantuvieron igual o aumentaron, con un mínimo de 6,2 y un máximo de 6,9 alcanzado en la categoría de calidad de la interfaz para el nivel avanzado (ver detalle en tabla 25).

| Indicador Promedio Iteración 2 | Nivel Básico | Nivel Medio | Nivel Avanzado |
|--------------------------------|--------------|-------------|----------------|
| CSUQ Total (máx. 7)            | 6,3          | 6,3         | 6,6            |
| Info Quality (máx. 7)          | 6,3          | 6,2         | 6,5            |
| Interfaz Quality (máx. 7)      | 6,2          | 6,5         | 6,9            |
| System Usability (máx. 7)      | 6,2          | 6,4         | 6,7            |

Tabla 25: Indicadores cuestionario CSUQ por conocimiento tecnológico en Iteración 3.  
Fuente: Elaboración propia.

Por último, en el gráfico de la figura 50, que muestra la tendencia de los resultados del cuestionario CSUQ por nivel de conocimiento tecnológico, se evidencia un incremento en los resultados de cada una de las categorías, especialmente entre la primera y la segunda iteración. Si bien entre la segunda y tercera iteración no se observa un aumento notable, es destacable la concentración de los resultados de la tercera iteración en comparación a las iteraciones anteriores en todos los niveles.

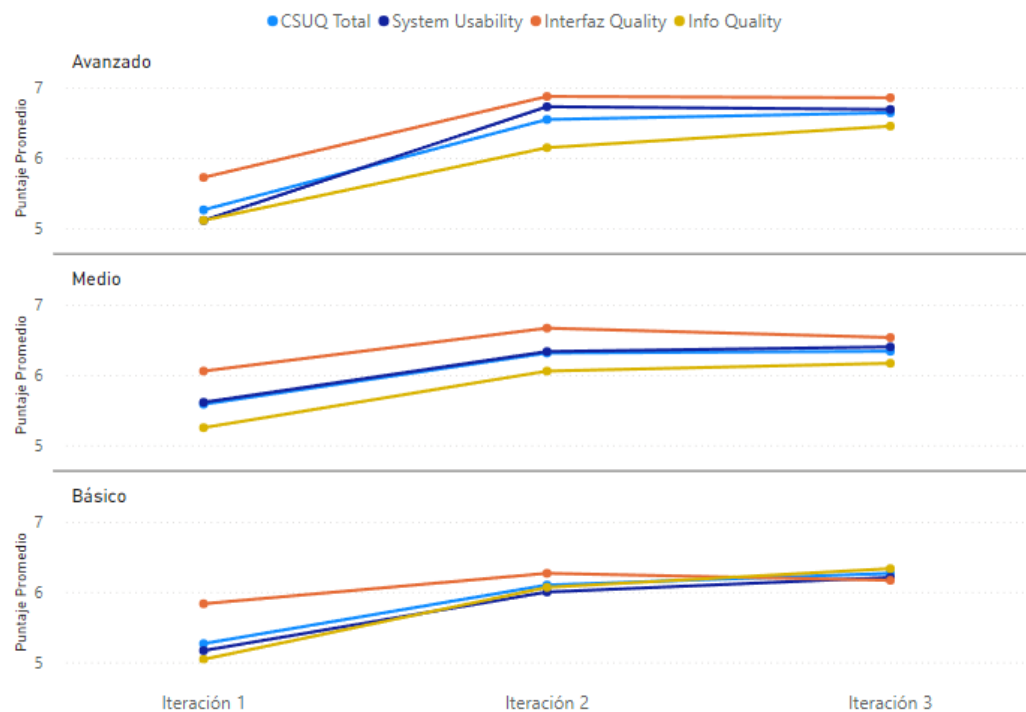


Figura 50: Gráfico de indicadores CSUQ por conocimiento tecnológico en cada iteración.  
Fuente: Elaboración propia.

### 4.3. Análisis de sentimiento

El tercer y último punto corresponde al análisis de sentimiento, aplicado a las opiniones de los usuarios al final de cada una de las sesiones de testing.

En la tabla 26 se presentan los resultados del score promedio para cada iteración. Este indicador, que puede variar entre -1 y 1, refleja el sentimiento de los usuarios hacia la aplicación, donde -1 indica un sentimiento muy negativo y 1 representa un sentimiento muy positivo. En general, en las **tres iteraciones**, se obtuvieron puntajes promedio mayores a 0, lo que indica que los usuarios **expresaron un sentimiento positivo** hacia la aplicación, aunque con diferentes intensidades.

| Iteración   | Promedio Score | Confianza |
|-------------|----------------|-----------|
| Iteración 1 | 0,59           | 100 %     |
| Iteración 2 | 0,97           | 100 %     |
| Iteración 3 | 0,96           | 100 %     |

Tabla 26: Score promedio por iteración, sin considerar falsos negativos.  
Fuente: Elaboración propia.

Entre la primera y la segunda iteración, se observó un aumento de 0,38 en el score promedio, alcanzando un puntaje cercano al máximo posible. Sin embargo, entre la segunda y la tercera iteración, se registró un ligero descenso de 0,01 puntos, aunque el score promedio sigue siendo cercano a 1. La confianza en estos resultados es del 100 %, ya que este análisis no incluye falsos negativos, lo que garantiza que todos los puntajes y sentimientos obtenidos reflejan fielmente las opiniones de los usuarios.

Si se consideran los falsos negativos, la tabla 27 muestra los resultados correspondientes al score promedio por iteración. En las tres iteraciones, también se obtuvieron resultados positivos, con una comparativa entre iteraciones similar a la que no considera falsos negativos, aunque con menor intensidad en el sentimiento. La diferencia principal se observa en la tercera iteración, donde se identificaron 2 falsos positivos, lo que reduce la confianza al 84 %, y el score promedio desciende más de 0,1 puntos en comparación a la segunda iteración. Además, como se puede apreciar en el gráfico de la figura 51, la discrepancia en el score promedio de la tercera iteración con y sin falsos positivos es mayor en comparación con las otras iteraciones, con una diferencia considerable de 0,22 puntos.

| Iteración   | Promedio Score | Confianza | Falsos Negativos |
|-------------|----------------|-----------|------------------|
| Iteración 1 | 0,51           | 94 %      | 1                |
| Iteración 2 | 0,87           | 94 %      | 1                |
| Iteración 3 | 0,74           | 88 %      | 2                |

Tabla 27: Score promedio por iteración, considerando falsos negativos.  
Fuente: Elaboración propia.

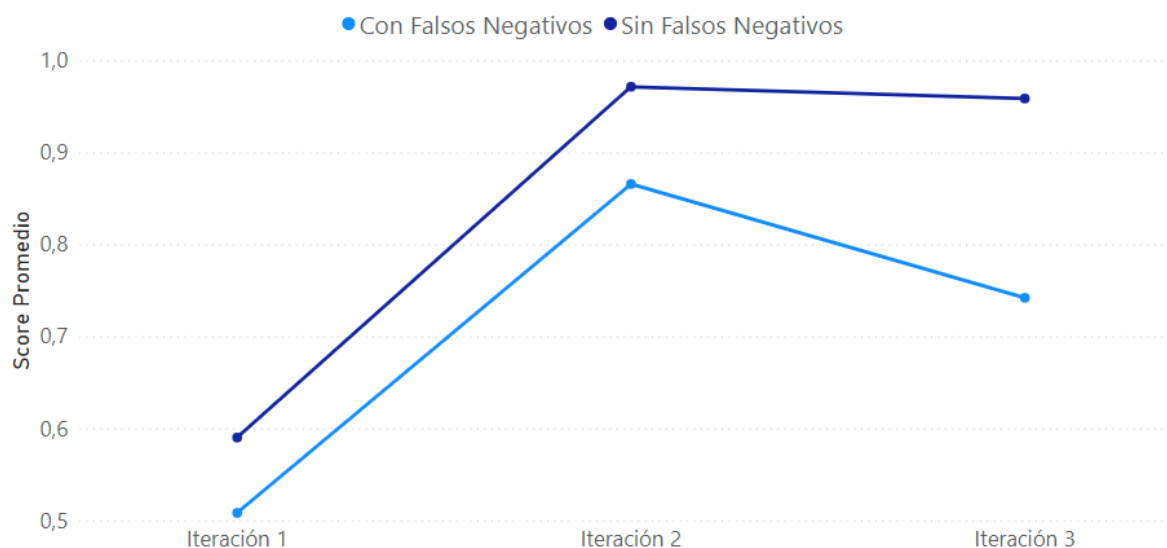


Figura 51: Gráfico con tendencia de score promedio, con y sin falsos negativos.

Fuente: Elaboración propia.

Antes de continuar con el análisis, cabe aclarar que los siguientes resultados de esta sección corresponden al análisis de sentimiento **sin considerar los falsos negativos**, ya que al ser una cantidad de opiniones reducida, el resultado erróneo de una de ellas afecta en algunos casos de forma considerable en el resultado promedio de la categoría analizada.

La tabla 28 muestra el score promedio por rango etario para las tres iteraciones. En ambos grupos, se observa un crecimiento entre la primera y la segunda iteración, con una diferencia positiva de 0,47 en adultos y 0,35 en jóvenes. Sin embargo, entre la segunda y la tercera iteración, se registró una ligera disminución de 0,02 en el rango etario adulto, mientras que en el grupo joven no se observó una variación en el score promedio.

| Rango Etario | Iteración 1 | Iteración 2 | Iteración 3 |
|--------------|-------------|-------------|-------------|
| Adulto       | 0,51        | 0,98        | 0,95        |
| Joven        | 0,62        | 0,97        | 0,97        |

Tabla 28: Score promedio por rango etario en cada iteración.

Fuente: Elaboración propia.

Es importante destacar que en la primera iteración, el rango etario adulto percibió el software ligeramente más negativo en comparación con el grupo joven. Sin embargo, para la segunda y tercera iteración, los valores de score promedio fueron similares entre ambos rangos etarios, con diferencias no mayores a 0,02 puntos entre ellos. Esta tendencia se puede apreciar gráficamente en la figura 52.

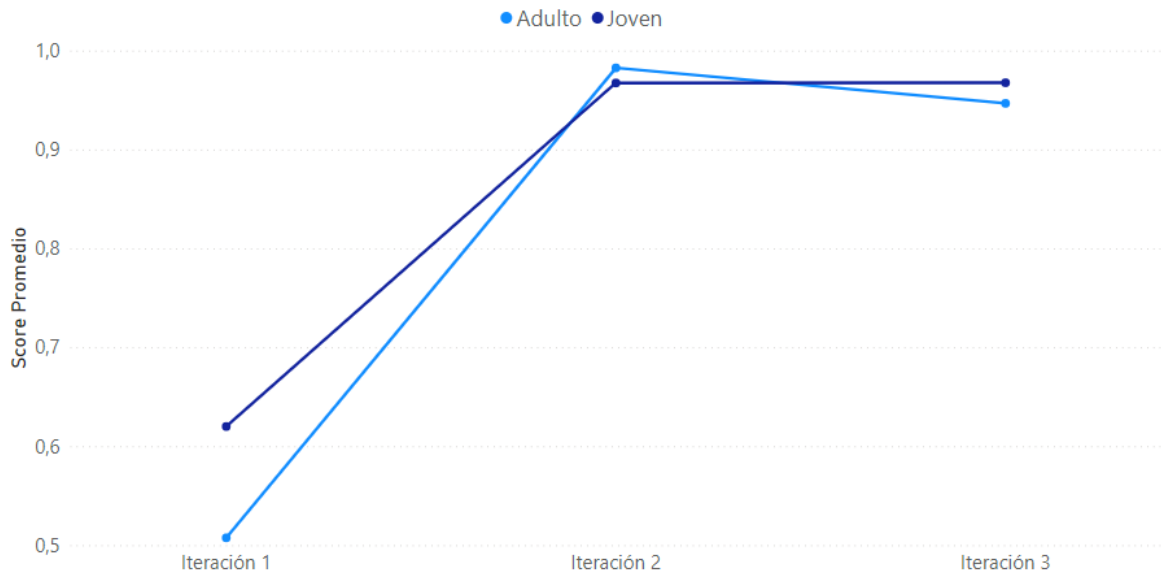


Figura 52: Gráfico con tendencia de score promedio por rango etario.  
Fuente: Elaboración propia.

Con respecto al score promedio por nivel de conocimiento tecnológico, tanto el nivel básico como en el avanzado obtuvieron un considerable incremento entre la primera y segunda iteración, con variaciones positivas de 0,52 y 0,77 respectivamente (ver detalle en tabla 29). No obstante, el nivel medio, contrario a la tendencia general, obtuvo un descenso de 0,1 puntos entre la primera y segunda iteración.

| Conocimiento tecnológico | Iteración 1 | Iteración 2 | Iteración 3 |
|--------------------------|-------------|-------------|-------------|
| Básico                   | 0,46        | 0,98        | 0,98        |
| Medio                    | 0,98        | 0,88        | 0,90        |
| Avanzado                 | 0,23        | 1,00        | 0,97        |

Tabla 29: Score promedio por nivel de conocimiento tecnológico en cada iteración.  
Fuente: Elaboración propia.

Ya hacia la tercera iteración, el nivel básico mantuvo su score promedio en comparación a la segunda iteración, mientras que el nivel avanzado disminuyó su valor en 0,03 puntos. En el caso del nivel medio, nuevamente contrario a la tendencia general, aumentó su score promedio en 0,02 puntos, aunque permanece por debajo del promedio obtenido en la primera iteración, tal como se puede observar en el gráfico de la figura 53.

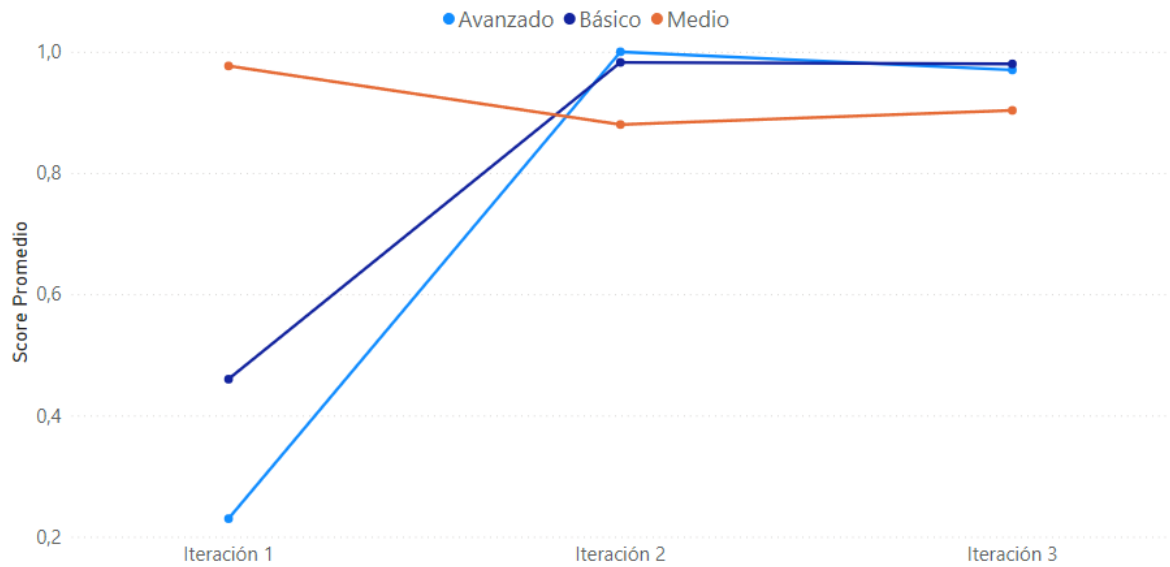


Figura 53: Gráfico con tendencia de score promedio por nivel de conocimiento tecnológico.  
Fuente: Elaboración propia.

Otro indicador relevante para analizar es la distribución de opiniones según su tipo de sentimiento, ya sea negativo, neutral o positivo. El gráfico de la figura 54 muestra un aumento progresivo en la cantidad de opiniones positivas a lo largo de las tres iteraciones. Desde un 75 % hasta un 81,25 % del total de las opiniones entre la primera y segunda iteración, alcanzando un 87,5 % en la tercera iteración, que corresponde al total de opiniones sin considerar los falsos negativos.

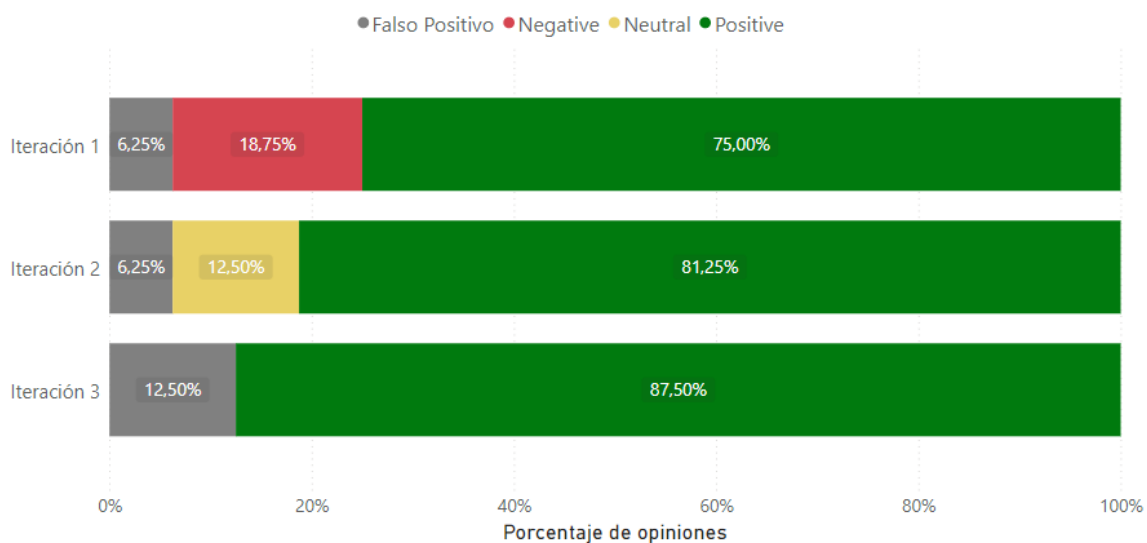


Figura 54: Gráfico de porcentajes por tipo de opinión en cada iteración  
Fuente: Elaboración propia.

En relación a las opiniones negativas, solo se registran 3 en todas las iteraciones, que corresponden al 18,75 % de las opiniones de la primera iteración. En la segunda iteración, no se registraron opiniones negativas, sino 2 opiniones neutrales, correspondientes al 12,5 % de las opiniones de la iteración 2. El detalle de la cantidad de opiniones por tipo de sentimiento se encuentra en la tabla 30

| Tipo de Opinión | Iteración 1 | Iteración 2 | Iteración 3 |
|-----------------|-------------|-------------|-------------|
| Negativo        | 3           | 0           | 0           |
| Neutral         | 0           | 2           | 0           |
| Positivo        | 12          | 13          | 14          |
| Falso Negativo  | 1           | 1           | 2           |

Tabla 30: Cantidad de opiniones por tipo de sentimiento en cada iteración.  
Fuente: Elaboración propia.

El análisis realizado hasta este punto da cuenta de una gran diferencia entre las opiniones positivas respecto a las neutrales y negativas. En este contexto, al enfocar el análisis exclusivamente en las respuestas positivas, adquiere especial relevancia el indicador que señala el porcentaje de respuestas con puntuaciones máximas, representado gráficamente en la figura 55.

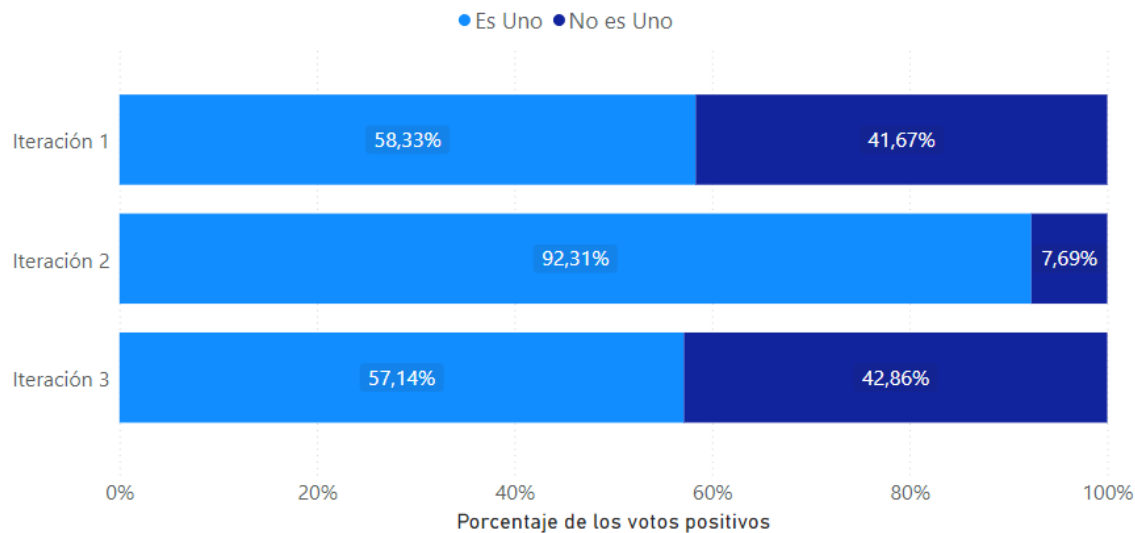


Figura 55: Distribución de las opiniones positivas en cada iteración.  
Fuente: Elaboración propia.

En la primera iteración, alrededor del 58 % de las opiniones positivas recibieron la puntuación máxima, lo que equivale a 7 de las 12 respuestas positivas identificadas. En la segunda iteración, este porcentaje aumentó significativamente a más del 92 %, con 12 de las 13 respuestas positivas obteniendo la puntuación máxima. Sin embargo, en la tercera iteración,

se observa un descenso en este indicador, con alrededor del 57 % de las opiniones positivas, equivalente a 8 de las 14 respuestas, y que equivale al porcentaje más bajo de las tres iteraciones.

El último análisis a realizar en esta sección se enfocará en las palabras clave más recurrentes en las opiniones de los usuarios. En la figura 56, se visualizan las palabras más utilizadas en la primera iteración. Destaca el término **«información»**, empleado en 5 ocasiones, seguido por **«fácil»**, utilizado 4 veces, y en tercer lugar, la palabra **«encontrar»**, presente en 3 oportunidades.

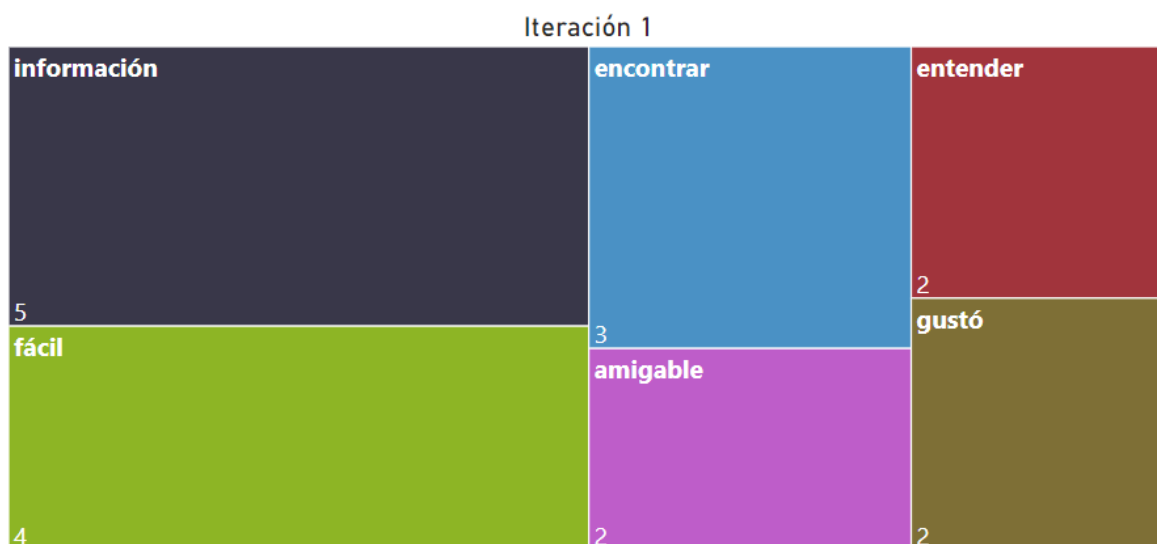


Figura 56: Palabras más utilizadas en opiniones de usuarios en primera iteración.

Fuente: Elaboración propia.

En la segunda iteración, el término **«fácil»** lidera el conteo, utilizado en 6 ocasiones, dos más que en la primera iteración. Le siguen las palabras **«clara»** y **«gustó»**, empleadas en 3 ocasiones cada una. En tercer lugar, varias palabras son mencionadas 2 veces, entre ellas destacan **«cambios»**, **«colores»** y **«excelente»**.

Por último, en la tercera iteración, la palabra más utilizada fue **«aplicación»**, mencionada por los usuarios en 5 ocasiones. Luego, varios términos fueron mencionados en 3 oportunidades cada uno, entre los que destacan **«cambios»**, **«encontrar»**, **«gustó»** (que repite el segundo lugar), y la palabra **«fácil»**, que disminuyó a la mitad sus apariciones en comparación con la segunda iteración.

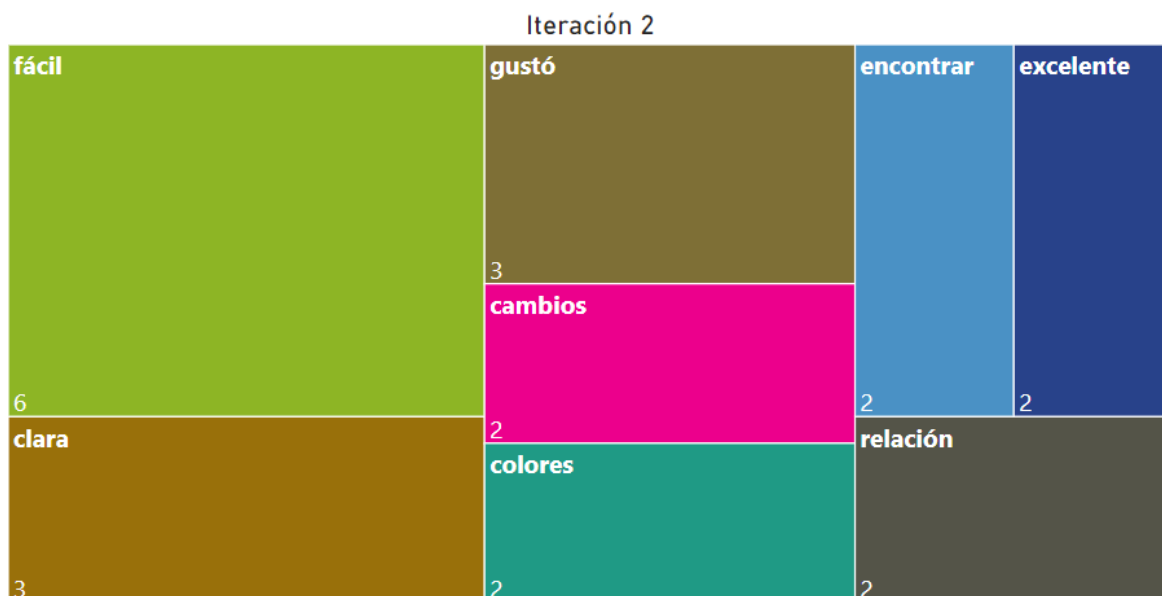


Figura 57: Palabras más utilizadas en opiniones de usuarios en segunda iteración.  
Fuente: Elaboración propia.



Figura 58: Palabras más utilizadas en opiniones de usuarios en tercera iteración.  
Fuente: Elaboración propia.

## CAPÍTULO 5

### CONCLUSIONES

#### 5.1. Conclusiones Generales

El trabajo presentado en esta memoria se centra en abordar un problema crítico en los sistemas actuales: la falta de cumplimiento de los estándares de calidad y madurez del software en cuanto a la experiencia de usabilidad del usuario. Entre las causas subyacentes a este problema, se identifica la escasa integración (o su carácter opcional) del testing de usabilidad en el ciclo de desarrollo de software, la falta de comunicación entre los desarrolladores y los usuarios finales, así como la tendencia de los usuarios a familiarizarse con un único tipo de software o sistema. Además, una de las causas más significativas tiene que ver con la carencia de herramientas que faciliten la comprensión del grado de satisfacción y el sentimiento del usuario con el sistema.

Es por ello que la plataforma diseñada e implementada en esta memoria, Feel UX, cobra especial relevancia en el contexto actual de desarrollo ágil de nuevos sistemas y aplicaciones, especialmente para aquellos que aspiran a alcanzar niveles óptimos de experiencia de usuario y usabilidad. Con las capacidades y funcionalidades implementadas en Feel UX, esta herramienta automatizada puede integrarse en cualquier estudio de usabilidad de un sistema, proporcionando una evaluación completa y detallada, desde aspectos generales como el tiempo y el éxito en la realización de tareas, hasta el análisis de los datos recolectados a través del cuestionario estandarizado CSUQ. Además, Feel UX se destaca por su innovador componente de análisis de sentimiento a las opiniones de los usuarios, impulsado por inteligencia artificial, el cual permite identificar el grado de satisfacción de los usuarios con el sistema en términos de sentimiento.

La capacidad de Feel UX para proporcionar visualizaciones en tiempo real de los resultados de cada estudio constituye otra de sus cualidades destacadas. Todos los datos recopilados durante las pruebas de usabilidad realizadas en la plataforma se someten a un proceso de limpieza y transformación, que permite a los responsables del testing acceder a información relevante mediante paneles de control o *dashboards*. Esta funcionalidad es crucial en la actualidad, donde los datos son un activo muy importante para las empresas, y por ende, las herramientas capaces de procesar y representar visualmente estos datos son altamente solicitadas y valoradas.

Por otro lado, la solución propuesta en este trabajo es una continuación de la investigación realizada por el exalumno de la UTFSM y actual Ingeniero Civil en Informática, Fabio Pazos. En su trabajo, titulado «Gestionar Madurez y Calidad de Producto de Software de Manera Ágil y Flexible por Medio de Técnicas de BI&A: Caso de Estudio Lyzerclass», Pazos propuso un modelo de madurez de calidad de software, que incluyó la ejecución de dos iteraciones de mejora continua en un software web, desarrollado en el contexto de la Feria de Software. Du-

rante estas iteraciones, se recopilaron opiniones de los usuarios, las cuales fueron sometidas a un análisis de sentimiento. Además, se emplearon herramientas de Business Intelligence & Analytics (BI&A) para visualizar y analizar los datos reunidos, lo que condujo a la propuesta de un rediseño para la aplicación web evaluada.

En este contexto, algunos de los desafíos futuros de su propuesta incluyen la extensión del modelo de madurez de calidad a otros tipos de software y la realización de un mayor número de iteraciones de testing. Es por ello que, para validar la solución implementada en la presente memoria, se optó por utilizar un software de escritorio en lugar de un software web, además de incrementar a tres la cantidad de iteraciones de mejora continua, en comparación a las dos realizadas en el trabajo previo.

La aplicación de escritorio seleccionada para llevar a cabo las iteraciones de mejora continua fue Desk Habitum. Esta elección resultó ser ideal para validar el estudio, dado que ofrece diversas funcionalidades, cada una lo suficientemente distinta como para proporcionar variedad en las tareas del estudio. A su vez, esta diversidad permitió identificar múltiples oportunidades de mejora, obteniendo como resultado un producto de software maduro y de calidad, lo que se ve evidenciado en los resultados presentados en el capítulo anterior.

En cuanto a la evolución del software testeado, es posible concluir que tuvo una evolución general positiva en todos los aspectos evaluados. En primer lugar, considerando los resultados obtenidos respecto al tiempo promedio de las tareas, en todas las tareas se logró reducir el tiempo promedio entre la primera y tercera iteración, análisis que se puede ampliar si separamos por rango etario y nivel de conocimiento, donde en todos los niveles ocurre una disminución en los tiempos promedio entre la versión inicial y final de la aplicación Desk Habitum. Esto obviamente provoca que los usuarios se sientan mas expertos en el software, y por ende, su percepción hacia el software sea más positiva, lo que se ve respaldado por los resultados del análisis de sentimiento.

Cabe destacar que en dos tareas particulares se obtuvo un aumento en el tiempo promedio entre iteraciones consecutivas, tal como se observa en la figura 46. En la tarea «Contactar a profesional», se observa un leve incremento entre la segunda y tercera iteración, atribuible a la validación de datos del formulario de contacto implementada en la segunda iteración, que requiere que el usuario complete el formulario. Por otro lado, en la tarea «Obtener resumen de última sesión», se registra un aumento en el tiempo promedio entre la primera y segunda iteración. Esto se debe a la reorganización de esta sección en la aplicación, ya que en la primera versión, el resumen estaba en la pantalla de inicio, mientras que en la segunda versión se trasladó al apartado de Dashboards. Sin embargo, estos casos particulares no representan la tendencia general, y responden a modificaciones que la aplicación necesitaba para favorecer otros aspectos de usabilidad.

El éxito promedio de las tareas mostró una evolución positiva constante, aumentando su porcentaje promedio a lo largo de las tres iteraciones. Lo más notable de este indicador es que, en la primera iteración, solo dos tareas lograron un 100 % de éxito, mientras que las demás oscilaban entre el 63 % y el 88 %. En comparación, en la tercera iteración, casi todas

las tareas alcanzaron un éxito del 100 %, lo que respalda la idea de que las modificaciones realizadas en el software, junto con la experiencia acumulada por los usuarios al llevar a cabo las tareas, contribuyeron a mejorar el nivel de usabilidad y madurez del software.

En relación a los resultados del cuestionario CSUQ, es factible concluir que hubo una tendencia general al alza, con un notable aumento en los niveles de usabilidad entre la primera y la segunda iteración, seguido de una estabilización de los indicadores promedio entre la segunda y la tercera iteración. Esto quiere decir que, entre la primera y segunda iteración, se registró el mayor alza en todos los aspectos de usabilidad, lo que es atribuible a que entre estas iteraciones se realizaron los cambios más importantes en la aplicación, y por ende, respalda la idea de que estos cambios fueron muy favorables para la percepción de usabilidad de los usuarios. Por otro lado, entre la segunda y la tercera iteración se evidenció una menor alza en los indicadores, pero tendiendo a igualar sus valores promedio, lo cual indica un equilibrio en los distintos aspectos de la aplicación, como la calidad de la información, la interfaz y la usabilidad general del sistema.

Otro indicador que respalda el alto nivel de usabilidad logrado por la aplicación testeada es el puntaje SUS. Este puntaje, que superó los 68/100 puntos en cada iteración, sugiere que la aplicación mantuvo un nivel satisfactorio de usabilidad en todas sus versiones. Es importante destacar que en la segunda y tercera iteración, alcanzó 89,4 y 90,9 puntos respectivamente sobre un máximo de 100, lo que corresponde a la calificación máxima, A+. En consecuencia, estos resultados sugieren que las modificaciones implementadas en las iteraciones sucesivas contribuyeron significativamente a la mejora general de la experiencia de usabilidad del usuario en la aplicación.

En la primera iteración, se destaca el bajo puntaje en el indicador de calidad de la información, que obtuvo una calificación general de 5,1 sobre 7. Este aspecto deficiente fue especialmente notorio en el rango etario adulto, donde alcanzó una puntuación de 4,9, y en el nivel de conocimiento tecnológico básico, con una calificación de 5,0. La baja puntuación en este indicador se vio respaldada por las opiniones de los usuarios, donde la palabra más recurrente fue precisamente "información" (consultar figura 56). Además, se observaron tiempos promedio elevados y un bajo porcentaje de éxito en las tareas relacionadas con la visualización de información en la aplicación. Esto permite confirmar la estrecha relación entre las tres dimensiones evaluadas para validar la solución: tiempo y éxito, cuestionario CSUQ y procesamiento de lenguaje natural, a través de la extracción de palabras claves a la opiniones de los usuarios.

Con respecto a los resultados del análisis de sentimiento, se observaron puntajes promedio superiores a 0 en todas las iteraciones, lo que indica que el sentimiento general de los usuarios hacia el sistema fue positivo. Además, se evidenció un incremento en el puntaje promedio entre la primera y la segunda iteración, seguido de un ligero descenso de apenas 0,01 puntos entre la segunda y la tercera iteración. Respecto al porcentaje de opiniones positivas, se registró un aumento constante a lo largo de las tres iteraciones, superando ampliamente a las opiniones negativas y neutrales en todas las iteraciones. Estos hallazgos reafirman la idea de que los usuarios experimentaron una evolución positiva respecto a su experiencia

de usabilidad en la aplicación.

Un caso particular surge al examinar la tendencia del score promedio según el nivel de conocimiento tecnológico. Mientras que los niveles básico y avanzado experimentan aumentos entre la primera y segunda iteración, especialmente el nivel avanzado que pasa de 0,23 a 1,00, el nivel medio muestra un ligero descenso de 0,98 a 0,88 en la segunda iteración, seguido de un pequeño aumento en la tercera, sin alcanzar el puntaje inicial. Esto sugiere que, hacia la tercera iteración, los usuarios con un nivel medio de conocimiento tecnológico muestran una percepción ligeramente más negativa en comparación con la primera iteración.

Esta situación podría deberse al hecho de que en la segunda y tercera iteración, las opiniones incluían sugerencias de mejora, lo que tiende a reducir el puntaje promedio del texto. Además, se debe considerar la cantidad de usuarios de nivel medio que participaron en la segunda iteración, que fue solo de 3, lo que significa que un puntaje bajo afecta significativamente al puntaje promedio. De todos modos, los puntajes promedio del nivel de conocimiento tecnológico medio superan los 0,88 puntos en todas las iteraciones, lo que sugiere que, a pesar de la ligera tendencia negativa, los usuarios continúan expresando un sentimiento bastante positivo hacia el software en sus opiniones.

En relación a las palabras más utilizadas en las opiniones de los usuarios, es destacable la presencia de la palabra «fácil» en todas las iteraciones, utilizada en 4 ocasiones en la primera iteración, 6 en la segunda y 3 en la tercera. En su mayoría, la expresión que contenía esta palabra era «fácil de usar», lo que indica claramente que la opinión de los usuarios refleja uno de los principales objetivos de estas iteraciones y de esta investigación, que es mejorar la facilidad de uso, o usabilidad, del software evaluado.

A modo de conclusión respecto a las modificaciones realizadas al modelo de madurez de calidad de software propuesto en la memoria previa de Fabio Pazos, el hecho de aplicarlo con éxito a distintos tipos de software, con resultados positivos, da cuenta de la validez y la versatilidad de este enfoque. La adaptabilidad del modelo a diferentes contextos y la capacidad de proporcionar resultados significativos demuestran su eficacia como una herramienta para evaluar y mejorar la calidad del software en diversos escenarios.

Además, al aumentar la cantidad a tres iteraciones, permite una observación más detallada de la evolución del software a lo largo del tiempo. Esto no solo proporciona una visión más completa de las mejoras realizadas, sino que también ofrece la oportunidad de identificar tendencias y patrones a largo plazo en la experiencia del usuario. De esta manera, se fortalece la capacidad de comprensión y análisis del proceso de desarrollo y mejora continua, lo que contribuye a una toma de decisiones más fundamentada en la gestión de la calidad y madurez del software.

Por último, en relación con los objetivos generales y específicos establecidos en este trabajo, se puede concluir que todos han sido cumplidos satisfactoriamente durante el transcurso de esta investigación. El objetivo general de «diseñar e implementar una solución de software basada en técnicas de análisis de sentimiento para alcanzar la conformidad en calidad y

madurez de un producto final en entornos de desarrollo ágiles» se logró mediante la creación e implementación de la plataforma web automatizada Feel UX, la cual emplea técnicas de minería de opiniones para analizar el feedback de los usuarios. Asimismo, los objetivos específicos también fueron alcanzados al llevar a iteraciones de testing en la aplicación de escritorio Desk Habitum, lo que permitió validar la solución propuesta mediante el rediseño continuo de la aplicación, con el propósito de mejorar tanto la calidad como la madurez del software.

## 5.2. Trabajos Futuros

### 5.2.1. Mejoras y nuevas funcionalidades a plataforma Feel UX

El siguiente listado presenta una serie de posibles mejoras a implementar en la plataforma automatizada. Estas mejoras no solo respaldarían la expansión del alcance del modelo de madurez que implementa, sino que también mejorarían la usabilidad del software en sí mismo, con el fin de aumentar su calidad y brindar un mayor apoyo a los desarrolladores de aplicaciones para que puedan llevar a cabo sus evaluaciones de usabilidad de manera más efectiva.

- **Añadir nuevas segmentaciones a los usuarios:** Actualmente, solo es posible segmentar por rango etario y nivel de conocimiento tecnológico, pero en ciertos estudios podría ser útil segmentar por otras categorías, como profesión, género, nivel de educación, entre otras.
- **Incluir diferentes cuestionarios estandarizados:** Aunque el cuestionario CSUQ es una herramienta útil para evaluar la usabilidad de un sistema, existen otros tipos de cuestionarios que podrían ser más adecuados según el contexto del problema y el tipo de software evaluado.
- **Permitir opiniones de usuarios en diferentes idiomas:** En la versión actual del sistema, los usuarios están limitados a expresar sus opiniones exclusivamente en español. Por lo tanto, sería una excelente mejora permitir la opinión del usuario en cualquier idioma, en donde el software detecte automáticamente el idioma de la opinión y pueda llevar a cabo el análisis de sentimiento correspondiente.
- **Aumentar funcionalidad de Dashboards:** Los paneles de control implementados en la versión actual del sistema carecen de opciones de filtrado o personalización visual. Para facilitar un análisis más completo de los datos y obtener información diversa, sería beneficioso introducir funcionalidades como segmentadores o filtros, además de la posibilidad de cambiar entre distintos tipos de gráficos. Se podría incluso considerar la integración de herramientas de inteligencia de negocios de terceros, como Power BI Embedded u otras alternativas similares.

### **5.2.2. Trabajo en conjunto con laboratorio UX de la UTFSM**

Desde el año 2021, la UTFSM cuenta con un laboratorio UX que ofrece una variedad de servicios, que van desde la realización de técnicas de investigación hasta la evaluación de los procesos de diseño centrados en las personas. Entre sus actividades se encuentran sesiones en grupo basadas en metodologías del design thinking y pruebas de usabilidad con usuarios finales<sup>2</sup>. Además, el laboratorio proporciona servicios de UX para empresas e instituciones públicas que deseen evaluar la experiencia que ofrecen sus tecnologías, productos o servicios.

En este sentido, el trabajo realizado en esta memoria tiene el potencial de establecer una colaboración estratégica con el laboratorio UX, fortaleciendo sus capacidades y ampliando su impacto mediante la integración de la plataforma Feel UX, que automatiza el proceso de testing de usabilidad e incorpora análisis de sentimiento a las opiniones de los usuarios. Como parte de los servicios ofrecidos, la plataforma puede proporcionar un valor agregado a los clientes externos de la universidad, brindándoles una herramienta automatizada para mejorar la comprensión de la experiencia de sus usuarios.

---

<sup>2</sup>Texto extraído desde su sitio web <https://labux.inf.utfsm.cl/laboratorio>

## ANEXOS

### 5.1. Anexo A

En el presente anexo se muestran las tablas de resumen para los cuestionarios posteriores a la tarea y posteriores al estudio.

| Cuestionario | Ventajas                                                                                                      | Desventajas                                                                                                |
|--------------|---------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| CSUQ         | De uso gratuito                                                                                               | No tiene                                                                                                   |
|              | Generalizable                                                                                                 |                                                                                                            |
|              | Algo flexible, con la opción de agregar elementos y se pueden eliminar tres elementos específicos si se desea |                                                                                                            |
|              | Datos normativos disponibles                                                                                  |                                                                                                            |
| EUCS         | De uso gratuito                                                                                               | No tiene                                                                                                   |
|              | Generalizable                                                                                                 |                                                                                                            |
|              | Rápido de completar                                                                                           |                                                                                                            |
| PSSUQ        | De uso gratuito                                                                                               | No tiene                                                                                                   |
|              | Generalizable                                                                                                 |                                                                                                            |
|              | Algo flexible, con la opción de agregar elementos y se pueden eliminar tres elementos específicos si se desea |                                                                                                            |
|              | Datos normativos disponibles                                                                                  |                                                                                                            |
| SUS          | De uso gratuito                                                                                               | Se centra en todo el sistema en lugar de aspectos específicos, lo que dificulta los juicios de diagnóstico |
|              | Rápido de completar                                                                                           |                                                                                                            |
|              | Los <i>items</i> se pueden alternar en tonos positivos y negativos                                            |                                                                                                            |
|              | Flexibilidad para cambiar la redacción del artículo                                                           |                                                                                                            |
|              | Particularmente confiable con tamaños de muestra pequeños                                                     |                                                                                                            |
|              | Generalizable en una amplia gama de tecnologías                                                               |                                                                                                            |
|              | Datos normativos disponibles                                                                                  |                                                                                                            |
| TAM          | De uso gratuito                                                                                               | No tiene                                                                                                   |
|              | Generalizable como prueba a través de diferentes sistemas, grupos de usuarios y entornos de investigación     |                                                                                                            |

Tabla 31: Cuestionarios *post-task* con sus ventajas y desventajas.

Fuente: Elaboración Propia.

| Cuestionario | Ventajas                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Desventajas                        |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|
| ER           | De uso gratuito                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | No tiene                           |
|              | El <i>rating before-and-after</i> permite trazar puntajes en un gráfico para mapear cuatro escenarios diferentes, los que incluye: 1) Se esperaba que la tarea fuera difícil pero no se experimentó como tal (oportunidad de promover tales características); 2) La tarea fue prevista y experimentada como difícil (oportunidad de mejorar); Ambas tareas consideradas como fáciles (considerar dejarlas solas); 4) Las tareas eran difíciles pero se esperaba que fueran fáciles (oportunidad de mejora enfocada para eliminar la insatisfacción) |                                    |
| SEQ          | De uso gratuito                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Muy limitado con una sola pregunta |
|              | Muy rápido de completar                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                    |
|              | Funciona bien a pesar de su simplicidad y se desempeña bien en comparación con medidas más complicadas, por ejemplo, SMEQ y UME                                                                                                                                                                                                                                                                                                                                                                                                                     |                                    |
|              | Datos normativos disponibles                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                    |
| SMEQ         | De uso gratuito                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Muy limitado con una sola pregunta |
|              | Muy rápido de completar                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                    |
|              | La ubicación de las etiquetas en la escala se basa en la calibración psicométrica con tareas                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                    |

Tabla 32: Cuestionarios *post-task* con sus ventajas y desventajas.  
Fuente: Elaboración Propia.

## 5.2. Anexo B

Cuestionario CSUQ (versión en español).

| CUESTIONARIO DE USABILIDAD EN SISTEMAS INFORMÁTICOS (CSUQ) |                                                                                                                     | Totalmente en desacuerdo |                       |                       |                       | Totalmente de acuerdo |                       |                       |
|------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|--------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|                                                            |                                                                                                                     | 1                        | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     |
| 1                                                          | En general, estoy satisfecho con lo fácil que es utilizar este sitio web.                                           | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 2                                                          | Fue simple usar este sitio web.                                                                                     | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 3                                                          | Soy capaz de completar mi trabajo rápidamente utilizando este sitio web.                                            | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 4                                                          | Me siento cómodo utilizando este sitio web.                                                                         | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 5                                                          | Fue fácil aprender a utilizar este sitio web.                                                                       | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 6                                                          | Creo que me volví experto rápidamente utilizando este sitio web.                                                    | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 7                                                          | El sitio web muestra mensajes de error que me dicen claramente cómo resolver los problemas.                         | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 8                                                          | Cada vez que cometo un error utilizando el sitio web, lo resuelvo fácil y rápidamente                               | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 9                                                          | La información (como ayuda en línea, mensajes en pantalla y otra documentación) que provee este sitio web es clara. | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 10                                                         | Es fácil encontrar en el sitio web la información que necesito.                                                     | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 11                                                         | La información que proporciona el sitio web fue efectiva ayudándome a completar las tareas.                         | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 12                                                         | La organización de la información del sitio web en la pantalla fue clara.                                           | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 13                                                         | La interfaz del sitio web fue placentera.                                                                           | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 14                                                         | Me gustó utilizar el sitio web.                                                                                     | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 15                                                         | El sitio web tuvo todas las herramientas que esperaba que tuviera.                                                  | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 16                                                         | En general, estuve satisfecho con el sitio web.                                                                     | <input type="radio"/>    | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Figura 59: Cuestionario CSUQ en su versión en Español.

Fuente: [Hedlefs et al., 2015].

### 5.3. Anexo C

Cambios adicionales realizados a la aplicación Desk Habitum en la primera iteración de mejora continua.

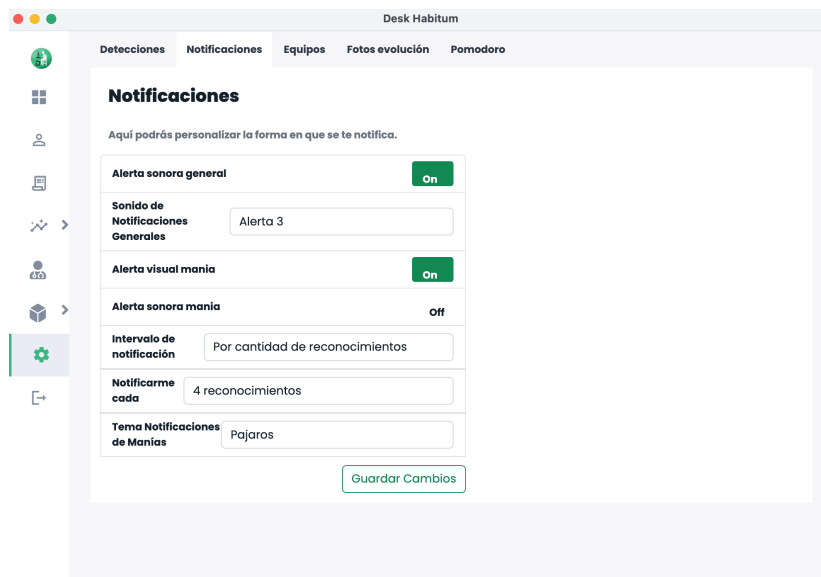


Figura 60: Vista de configuraciones en primera versión de Desk Habitum.  
Fuente: Elaboración propia.

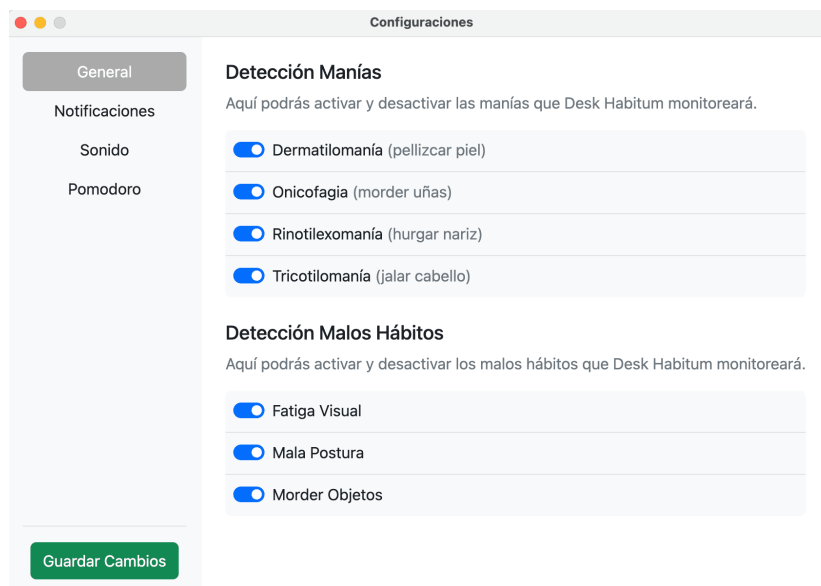


Figura 61: Vista de configuraciones en segunda versión de Desk Habitum.  
Fuente: Elaboración propia.

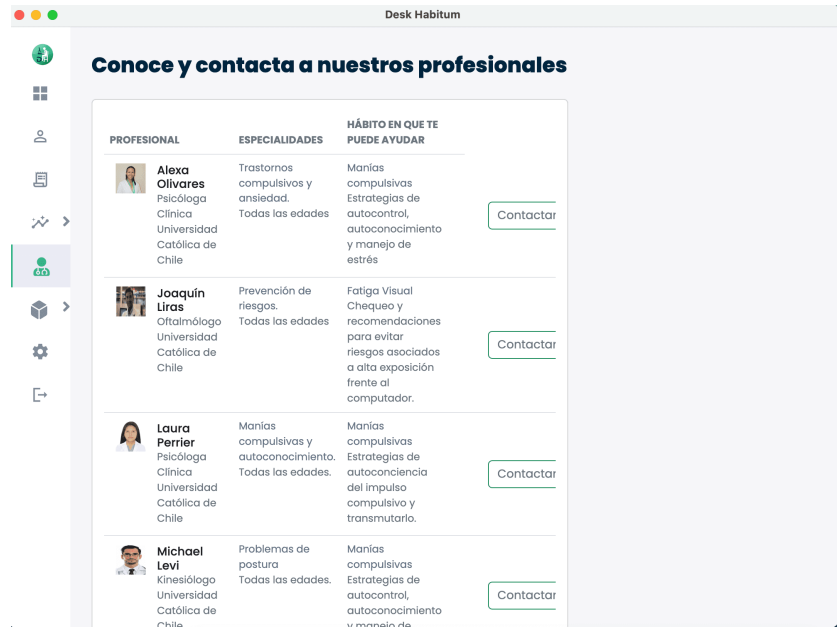


Figura 62: Vista de contacto profesional en primera versión de Desk Habitum.  
Fuente: Elaboración propia.

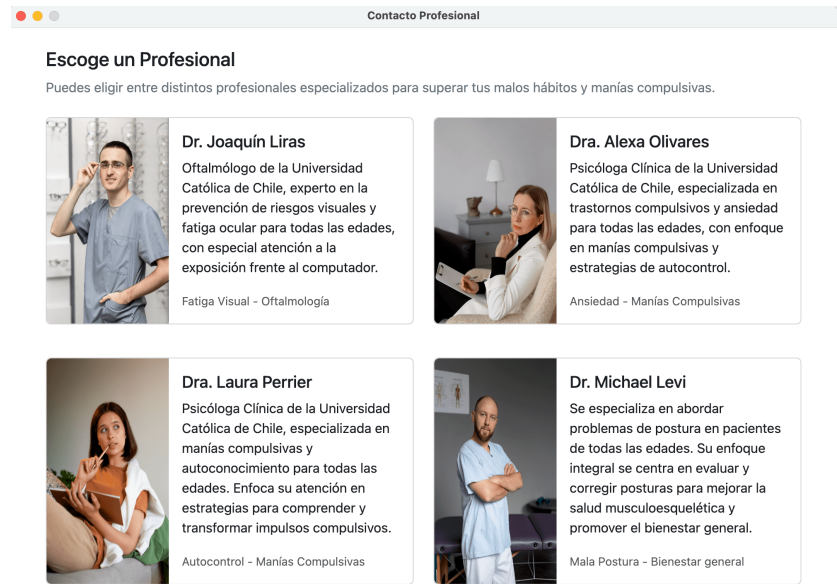


Figura 63: Vista de contacto profesional en segunda versión de Desk Habitum.  
Fuente: Elaboración propia.



Figura 64: Vista de respiración profunda en primera versión de Desk Habitum.  
Fuente: Elaboración propia.



Figura 65: Vista de respiración profunda en segunda versión de Desk Habitum.  
Fuente: Elaboración propia.

## 5.4. Anexo D

Cambios adicionales realizados a la aplicación Desk Habitum en la segunda iteración de mejora continua.



Figura 66: Tooltip en botones de técnicas de respiración profunda.  
Fuente: Elaboración propia.



Figura 67: Comparativa de menú rápido, con la opción de contacto profesional agregada.  
Fuente: Elaboración propia.

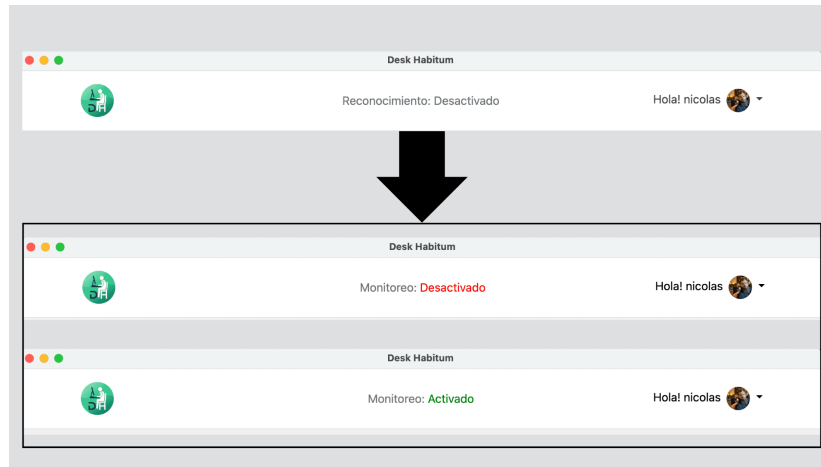


Figura 68: Implementación de color a estado de monitoreo.  
Fuente: Elaboración propia.

## REFERENCIAS BIBLIOGRÁFICAS

- [Arhippainen y Tähti, 2003] Arhippainen, L. y Tähti, M. (2003). Empirical evaluation of user experience in two adaptive mobile application prototypes. En *Mum 2003. proceedings of the 2nd international conference on mobile and ubiquitous multimedia*, número 011, pp. 27–34. Citeseer.
- [Bevan et al., 2015] Bevan, N., Carter, J., y Harker, S. (2015). Iso 9241-11 revised: What have we learnt about usability since 1998? En *International conference on human-computer interaction*, pp. 143–151. Springer.
- [Boynton y Greenhalgh, 2004] Boynton, P. M. y Greenhalgh, T. (2004). Selecting, designing, and developing your questionnaire. *Bmj*, 328(7451):1312–1315.
- [Brave y Nass, 2007] Brave, S. y Nass, C. (2007). Emotion in human-computer interaction. En *The human-computer interaction handbook*, pp. 103–118. CRC Press.
- [Buhl et al., 2011] Buhl, H. U., Röglinger, M., Stöckl, S., y Braunwarth, K. S. (2011). Value orientation in process management. *Business & Information Systems Engineering*, 3(3):163–172.
- [Cadavid et al., 2013] Cadavid, A. N., Martínez, J. D. F., y Vélez, J. M. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva*, 11(2):30–39.
- [Carvajal Jiménez, 2018] Carvajal Jiménez, C. L. (2018). *Patrones organizativos para la integración de usabilidad de software*. Tesis doctoral, ETSI\_Informatica.
- [Cflow, 2023] Cflow (2023). What is business process integration (bpi)?
- [Chacón, 2004] Chacón, W. (2004). Modelo de capacidad de madurez del software y su influencia en las mejoras de calidad del software. *Trabajo de graduación. Universidad de San Carlos de Guatemala*.
- [Collarte González, 2021] Collarte González, I. (2021). Procesamiento del lenguaje natural con bert: Análisis de sentimientos en tuits. B.S. thesis.
- [datos.gob.es, 2022] datos.gob.es (2022). 10 librerías populares de procesamiento del lenguaje natural.
- [Deemer et al., 2009] Deemer, P., Benefield, G., Larman, C., y Vodde, B. (2009). Información básica de scrum. *California: Scrum Training Institute*.
- [Dijkstra, 1972] Dijkstra, E. W. (1972). The humble programmer. *Communications of the ACM*, 15(10):859–866.
- [Domingo y Pera, 2010] Domingo, M. G. y Pera, E. M. (2010). Diseño centrado en el usuario. *Universitat Oberta de Catalunya*, pp. 9–12.

- [Echeverry et al., 2008] Echeverry, A. M. L., Cabrera, C., y Ayala, L. E. V. (2008). Introducción a la calidad de software. *Scientia et technica*, 2(39).
- [Estayno et al., 2009] Estayno, M. G., Dapozo, G. N., Cuenca Pletsch, L. R., y Greiner, C. L. (2009). Modelos y métricas para evaluar calidad de software. En *XI Workshop de Investigadores en Ciencias de la Computación*.
- [Feldman et al., 2007] Feldman, Ronen and Sanger, James and others (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press.
- [Graham et al., 2021] Graham, D., Black, R., y Van Veenendaal, E. (2021). *Foundations of software testing ISTQB Certification*. Cengage Learning.
- [Grau, 2000] Grau, X. F. (2000). Principios básicos de usabilidad para ingenieros software. En *JISBD*, pp. 39–46.
- [Hassan Montero y Martín Fernández, 2005] Hassan Montero, Y. y Martín Fernández, F. J. (2005). La experiencia del usuario. *No solo usabilidad*, (4).
- [Hedlefs et al., 2015] Hedlefs, A., de la Garza, G., Sánchez, M., y Garza, V. (2015). Adaptación al español del cuestionario de usabilidad de sistemas informáticos csuq. *RECI Revista Iberoamericana de las Ciencias Computacionales e Informática*, 4(8).
- [Henderson, 2020] Henderson, C. (2020). 4 pasos para la integracion eficiente de procesos.
- [Hodrien et al., 2021] Hodrien, Andrew and Fernando, TP and others (2021). A review of post-study and post-task subjective questionnaires to guide assessment of system usability. *Journal of Usability Studies*, 16(3):203–232.
- [Hutto y Gilbert, 2014] Hutto, C. y Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. En *Proceedings of the international AAAI conference on web and social media*, volumen 8, pp. 216–225.
- [ISO, 1998] ISO (1998). *ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs): Part 11: Guidance on usability*.
- [ISO/IEC, 2010] ISO/IEC (2010). *Iso/iec 25010 system and software quality models*. Technical report.
- [Kaner et al., 1999] Kaner, C., Falk, J., y Nguyen, H. Q. (1999). *Testing computer software*. John Wiley & Sons.
- [Lewis, 1995] Lewis, J. R. (1995). Computer system usability questionnaire. *International Journal of Human-Computer Interaction*.
- [Lewis, 2018] Lewis, J. R. (2018). Measuring perceived usability: The csuq, sus, and umux. *International Journal of Human-Computer Interaction*, 34(12):1148–1156.

- [Maida y Pacienza, 2015] Maida, E. G. y Pacienza, J. (2015). Metodologías de desarrollo de software.
- [Mascheroni et al., 2012a] Mascheroni, M. A., Greiner, C. L., Dapozo, G. N., y Estayno, M. G. (2012a). Herramienta para automatizar la evaluación de la usabilidad en productos software. En *XVIII Congreso Argentino de Ciencias de la Computación*.
- [Mascheroni et al., 2012b] Mascheroni, M. A., Greiner, C. L., Petris, R. H., Dapozo, G. N., y Estayno, M. G. (2012b). Calidad de software e ingeniería de usabilidad. En *XIV Workshop de Investigadores en Ciencias de la Computación*.
- [Medhat et al., 2014] Medhat, W., Hassan, A., y Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.
- [Mejova, 2009] Mejova, Y. (2009). Sentiment analysis: An overview. *University of Iowa, Computer Science Department*.
- [Montero, 2015] Montero, Y. H. (2015). *Experiencia de usuario: principios y métodos*. Editor no identificado.
- [MuleSoft, 2020] MuleSoft (2020). Business process integration: Bpi.
- [Nielsen, 1993] Nielsen, J. (1993). Usability engineering (ap professional, new york).
- [Nielsen y Loranger, 2006] Nielsen, J. y Loranger, H. (2006). *Usabilidad: prioridad en el diseño web (Prioritizing Web Usability)*. Anaya.
- [Parra, 2019] Parra, A. (2019). Análisis de sentimiento. qué es y cómo realizarlo.
- [Paz, 2016] Paz, J. A. M. (2016). Análisis del proceso de pruebas de calidad de software. *Ingeniería solidaria*, 12(20):163–176.
- [Pérez-Mergarejo et al., 2014] Pérez-Mergarejo, E., Pérez-Vergara, I., y Rodríguez-Ruiz, Y. (2014). Modelos de madurez y su idoneidad para aplicar en pequeñas y medianas empresas. *Ingeniería Industrial*, 35(2):184–198.
- [Pressman, 1993] Pressman, R. S. (1993). Ingeniería de software, 3ra. Ed., *McCraw Hill*.
- [Rodgers, 2022] Rodgers, N. (2022). Diferencias entre testing funcional y no funcional: Abstracta chile.
- [Rosemann y vom Brocke, 2015] Rosemann, M. y vom Brocke, J. (2015). The six core elements of business process management. En *Handbook on business process management 1*, pp. 105–122. Springer.
- [Sagar y Saha, 2017] Sagar, K. y Saha, A. (2017). A systematic review of software usability studies. *International Journal of Information Technology*, pp. 1–24.
- [Sanchez, 2015] Sanchez, Walter Ovidio, o. (2015). La usabilidad en ingeniería de software: definición y características.

- [Sauro y Lewis, 2016] Sauro, J. y Lewis, J. R. (2016). *Quantifying the user experience: Practical statistics for user research*. Morgan Kaufmann.
- [Scalone, 2006] Scalone, F. (2006). Estudio comparativo de los modelos y estándares de calidad del software. *Trabajo de maestría*. Universidad Tecnológica Nacional, Facultad Regional Buenos Aires.
- [Schwaber y Sutherland, 2011] Schwaber, K. y Sutherland, J. (2011). The scrum guide. *Scrum Alliance*, 21(19):1.
- [Schwaber y Sutherland, 2013] Schwaber, K. y Sutherland, J. (2013). La guía de scrum. *Scrumguides. Org*, 1:21.
- [Sliwinski, 2019] Sliwinski, A. (2019). Sentiment: Docs, community, tutorials, reviews.
- [Sutherland y Schwaber, 2007] Sutherland, J. y Schwaber, K. (2007). The scrum papers. *Nuts, Bolts and Origins of an Agile Process*.
- [Takeuchi y Nonaka, 1986] Takeuchi, H. y Nonaka, I. (1986). The new new product development game. *Harvard business review*, 64(1):137–146.
- [Viteri, 2021] Viteri, S. B. A. (2021). Análisis de sentimientos para twitter con vader y textblob. *Revista Odigos*, 2(3):9–25.
- [Whittaker, 2000] Whittaker, J. A. (2000). What is software testing? and why is it so hard? *IEEE software*, 17(1):70–79.
- [Zeroual y Lakhouaja, 2018] Zeroual, I. y Lakhouaja, A. (2018). Data science in light of natural language processing: An overview. *Procedia Computer Science*, 127:82–91.