

**APLICACIÓN WEB PARA EL APOYO DE EVALUACIONES DE USABILIDAD DE
SISTEMAS INTERACTIVOS MEDIANTE MÚLTIPLES CONJUNTOS HEURÍSTICOS**

Carlos Andrés Riascos Pareja

**Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Tuluá - Valle
2017**

**APLICACIÓN WEB PARA EL APOYO DE EVALUACIONES DE USABILIDAD DE
SISTEMAS INTERACTIVOS MEDIANTE MULTIPLES CONJUNTOS HEURÍSTICOS**

Carlos Andrés Riascos Pareja
Código: 201356541
`carlos.a.riascos@correounivalle.edu.co`

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director
Royer David Estrada Esponda, Ms.c

Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Tuluá - Valle
2017

NOTA DE ACEPTACIÓN

Royer David Estrada Esponda
Director

Jurado 1

Jurado 2

Dedicatoria

Este trabajo de grado lo dedico a DIOS como principio y fin de cada paso que doy en mi vida, el gran baluarte que me socorre e ilumina mi mente en cada actuar, de igual manera dirijo este logro a mi gran aliciente, mi FAMILIA, especialmente a mis padres Carlos A. Riascos B. y Blanca N. Pareja M., quienes han unido sus fuerzas por mi bienestar y educación, convirtiéndose en la roca que me sustenta y acompaña en momentos de triunfos y flaquezas, simplemente por ser el reflejo del amor verdadero.

Carlos Andrés Riascos Pareja

Agradecimientos

A mi director de trabajo de grado, el Ingeniero Royer D. Estrada E., por la confianza depositada en mí y su empeño por culminar este trabajo con grandes logros.

A mis amigos y compañeros, con los que viví y compartí conocimientos y experiencias que alimentaron este proceso de formación profesional.

A los docentes y toda la familia Univalle que han impulsado y custodiado el desarrollo de mi formación profesional, uniendo esfuerzos para brindarme una educación de calidad e infraestructura idónea para el conocimiento.

Tabla de Contenido

Dedicatoria	II
Agradecimientos	III
Resumen	8
Abstract	9
1. Introducción	10
1.1. Descripción General	10
1.2. Problema	10
1.2.1. Descripción del problema	10
1.2.2. Formulación del problema	12
1.3. Objetivos	12
1.3.1. Objetivo general	12
1.3.2. Objetivos específicos	12
1.4. Alcance	13
1.5. Estructura del documento	13
2. Marco Referencial	14
2.1. Antecedentes	14
2.2. Marco Teórico	15
2.3. Marco Conceptual	19
2.3.1. Laravel	20
2.3.2. Cakephp	21
2.3.3. Yiiframework	21
2.3.4. Codeigniter	22
2.3.5. Simfony 2	22
2.4. Plataforma de Desarrollo	23
2.4.1. PHP (Hypertext Preprocessor)	23
2.4.2. MySQL	23
2.4.3. HTML	24
2.4.4. CSS (Cascading Style Sheets)	24
2.4.5. JavaScript	24
2.4.6. jQuery	24
2.4.7. GitHub	24

3. Desarrollo del proyecto	26
3.1. Ingeniería	27
3.1.1. Análisis	27
3.1.1.1. Diagrama de caso de uso (Negocio)	27
3.1.1.2. Historias de usuario	28
3.1.2. Diseño	29
3.1.2.1. Diagrama de casos de uso	29
3.1.2.2. Modelo entidad relación	30
3.2. Descripción del sistema	31
3.2.1. Interfaz gráfica de usuario	31
3.2.2. Módulo Heurísticas	32
3.2.2.1. Submódulo Principios	32
3.2.2.2. Submódulo Escala de valoración	32
3.2.2.3. Submódulo Autores	32
3.2.3. Módulo Evaluación	33
3.2.3.1. Submódulo Configuración Ambiente de Evaluación	33
3.2.3.2. Submódulo Solución de Evaluación	33
3.2.3.3. Submódulo Reporte de Evaluación	33
3.2.4. Módulo Reportes	33
3.2.5. Módulo Perfiles	33
3.2.6. Módulo Usuarios	33
3.3. Arquitectura del Sistema Implementado	33
3.4. Detalles de implementación	34
3.5. Pruebas	42
3.5.1. Desarrollo de pruebas unitarias	42
3.5.2. Desarrollo de pruebas de aceptación	45
4. Conclusiones y Trabajos Futuros	47
4.1. Conclusiones	47
4.2. Trabajos Futuros	48
Bibliografía	50
A. Historias de Usuario	51
B. Casos de Uso	54
C. Diagramas Entidad - Relación	55
D. Resultados evaluación heurística Sistema control docente - Manual	56
E. Resultados evaluación heurística página web Univalle Tuluá - HeuristicTool	57
F. Pruebas de Aceptación	58
G. Manuales de Usuario	61

Lista de Figuras

2.1. Composición de un SI	16
2.2. Arquitectura MVC aplicada a un framework web	19
3.1. Muro Kanban - HeuristicTools	27
3.2. Diagrama caso de uso (Negocio) Evaluación heurística	28
3.3. Diagrama caso de uso - Modulo evaluación.	30
3.4. Modelo Entidad Relación - Modulo Heurísticas.	31
3.5. Interfaz principal administrador de HeuristicTool.	32
3.6. Arquitectura HeuristicTool.	34
3.7. Selección de evaluadores.	36
3.8. Selección de principios.	36
3.9. Resolver evaluación.	37
3.10. Reporte de evaluación.	37
3.11. Reportes de la aplicación.	38
3.12. PHPUnit – Método visit.	42
3.13. PHPUnit – Método call.	43
3.14. PHPUnit – Método call Envío de datos.	43
3.15. PHPUnit – Base de datos.	43
3.16. PHPUnit – Ejecución de prueba 1.	44
3.17. PHPUnit – Ejecución de prueba 2.	45
B.1. CU - Modulo Heurísticas.	54
C.1. ER - Modulo Evaluación.	55

Lista de tablas

2.1. Cuadro comparativo entre herramientas	14
2.2. Cuadro comparativo entre frameworks PHP	23
3.1. Historia de Usuario, Crear Evaluación.	29
3.2. Resultados de evaluación manual – Sistema control docente.	40
3.3. Resultados evaluación HeuristicTool – Univalle - Tuluá.	41
3.4. Pruebas de aceptación, módulo Evaluación.	46
A.1. HU, Consultar una Evaluación.	51
A.2. HU, Seleccionar Evaluadores.	52
A.3. HU, Resolver Evaluación.	52
A.4. HU, Conclusión de la Evaluación.	53
A.5. HU, Reporte de la Evaluación.	53
F.1. PA, Consultar una Evaluación.	58
F.2. PA, Seleccionar Evaluadores.	59
F.3. PA, Resolver Evaluación.	59
F.4. PA, Conclusión de la Evaluación.	60
F.5. PA, Reporte de la Evaluación.	60

Resumen

La usabilidad de los sistemas interactivos es un atributo de calidad del software que posibilita a los usuarios realizar las tareas de forma fácil, intuitiva y cómoda, permitiendo alcanzar sus objetivos de manera rápida y satisfactoria. Paralelamente, la ausencia de herramientas para apoyar la evaluación de la usabilidad, específicamente evaluaciones heurísticas, se traduce en una desventaja para los evaluadores a la hora de llevar a cabo la verificación del grado de conformidad de un software en cuanto a un atributo importante como la usabilidad, así pues, la ausencia de dichas herramientas repercute en el tiempo empleado en la ejecución de la evaluación y a prescindir de aspectos importantes a la hora de analizar y realizar la respectiva conclusión.

Con base en lo anterior, el aplicativo web expuesto en el presente documento se desarrolló en el marco de una investigación aplicada, la cual posibilitó por medio de la metodología de desarrollo Kanban, ofrecer a la comunidad una aplicación web que sirve de apoyo en el proceso de evaluación heurística, permitiendo a los evaluadores facilitar el proceso de análisis de los resultados y contribuir con transmisión de conocimiento de manera oportuna.

Palabras Claves: Usabilidad, Evaluación heurística, Conjuntos heurísticos, Kanban, Sistemas de Información.

Abstract

The usability of interactive systems is an attribute of software quality that enables users to perform tasks easily, intuitively and conveniently, allowing them to achieve their goals quickly and satisfactorily. At the same time, the lack of tools to support usability assessment, specifically heuristic evaluations, results in a disadvantage for evaluators when carrying out verification of the degree of conformity of a software in terms of an important attribute such as Therefore, the absence of such tools has repercussions on the time spent in the execution of the evaluation and on the lack of important aspects when analyzing and making the respective conclusion.

Based on the above, the web application presented in this document was developed within the framework of an applied research, which made it possible through the Kanban development methodology to offer the community a web application that supports the process of heuristic evaluation, allowing the evaluators to facilitate the process of analysis of the results and contribute with transmission of knowledge in a timely manner.

Key Words: Usability, Heuristic evaluation, Heuristic sets, Kanban, Information systems.

Capítulo 1

Introducción

1.1. Descripción General

En la actualidad, el aumento y evolución del uso de las tecnologías de la información y la comunicación (TIC) [1], ha hecho necesario considerar un conjunto de criterios de calidad que logren satisfacer la interacción de los usuarios con sistemas interactivos de manera fácil, sencilla y cómoda, en consecuencia, la usabilidad es un atributo clave que debe ser perseguido en la Ingeniería de Software [2], la cual es considerada como un factor importante para lograr el éxito de una aplicación [3].

Paralelamente, para medir el nivel de usabilidad de un sistema interactivo, se plantean diversas técnicas y metodologías, entre ellas se pueden encontrar las evaluaciones heurísticas, las cuales hacen parte de un proceso evaluativo por inspección que es llevado a cabo por diferentes tipos de evaluadores, consiste en analizar y evaluar cada elemento de la interfaz de un sistema interactivo mediante diversos principios heurísticos establecidos previamente.

Dado lo anterior, el objetivo del trabajo de grado que aquí se presenta, estuvo orientado al desarrollo e implementación de una aplicación web, que facilite el proceso de evaluación de sistemas interactivos mediante múltiples conjuntos heurísticos, logrando de esta forma la obtención de resultados de una manera más rápida y confiable, lo que posteriormente facilitara el análisis de los mismos en el marco de un proceso de toma de decisiones. <https://es.sharelatex.com/project/59eca949c26a0a765618a8b8> Actualmente, gran parte de la gestión de estas evaluaciones, no cuenta con el apoyo de las herramientas que ofrecen las Tecnologías de la Información, lo que conlleva a maximizar el tiempo de elaboración y puesta en marcha de las mismas, dificultando el análisis y diagnóstico de resultados por parte de los evaluadores [4].

1.2. Problema

1.2.1. Descripción del problema

En la actualidad, los productos o servicios que ostentan certificados de calidad son preferidos por los compradores o usuarios, ya que transmiten seguridad y confianza, lo que constituye un atributo de valor para las estrategias de comercialización en una organización. En concordancia la Industria del Software no puede prescindir de este factor para el desarrollo de sistemas interactivos, por ello se han realizado avances en la definición, modelos, metodologías e implementación de estándares que contribuyan a la

evolución y evaluación de la calidad del software [5]. Así mismo, autores y organizaciones como Pressman, Garvin, McCall e ISO, categorizan la calidad del software en factores o atributos claves que deben ser perseguidos, entre ellos la usabilidad [2].

El término usabilidad ha sido adoptado por organizaciones y modelos como IPO (Interacción Persona-Ordenador), DCU (Diseño Centrado en el Usuario), MPIu+a (Modelo de proceso de la Ingeniería de la Usabilidad y de la Accesibilidad), UX (Experiencia de Usuario), entre otros. Sin embargo, desarrollar un sistema interactivo altamente usable no es una tarea fácil de conseguir, por lo que se hace necesario usar técnicas de evaluación que permitan medir el grado de usabilidad e indiquen los elementos a corregir. En relación, Ferré X asegura que: “Sin llevar a cabo algún tipo de evaluación es imposible saber si el sistema satisface las necesidades de los usuarios y si encaja adecuadamente en el contexto físico, social y organizacional en el que va a ser usado” [6].

Una de las técnicas de evaluación de usabilidad es la basada en la revisión de heurísticas, en el contexto de la valoración de los sistemas interactivos, es la que está más extendida. Este análisis tiene como objetivo encontrar problemas de usabilidad en el diseño de la interfaz de usuario para que estos puedan ser subsanados en el proceso de diseño iterativo [7]. El proceso de evaluación se divide en tres etapas que generan diferentes entregables: (I) Planificación, se centra en la selección de evaluadores que llevarán a cabo el proceso, y de principios heurísticos (PHs) que sean adecuados a la naturaleza (contexto) y tipo de interfaz a evaluar para la posterior creación de una lista de comprobación o “planilla en blanco” que se traduce en una serie de preguntas o sub-PHs condicionadas por una escala de valores para cada posible respuesta; (II) Puesta en marcha, los evaluadores inspeccionan la interfaz del sistema a evaluar de manera individual, esta actividad se hace a partir de la lista de comprobación generada en la etapa anterior y de ser necesario pueden agregar nuevos PHs a los ya existentes, después de concluidas las evaluaciones se comunican los hallazgos para establecer el grado de severidad de cada uno de los problemas encontrados; (III) Análisis de resultados, se lleva a cabo desde dos puntos de vista no excluyentes y complementarios, el cuantitativo que resume de manera clara y precisa la ponderación de cada una de las preguntas que cristalizan los PHs generales con respecto a la interfaz evaluada y el cualitativo que permite la justificación y recomendación de los evaluadores frente a los problemas de usabilidad hallados [8].

A pesar del “auge” de la usabilidad, Larman sostiene que “no existe probablemente ninguna técnica con mayor disparidad entre su importancia para el éxito del desarrollo de software y la falta de una atención y educación formal que la Ingeniería de la Usabilidad y el Diseño de las Interfaces de Usuario (UI)” [9]. Ante el avance tecnológico de la sociedad de la información, la realización de una evaluación heurística sigue siendo un proceso manual, donde a nivel general se realiza anotando las puntuaciones de severidad correspondientes a cada heurística en un papel o en una hoja de cálculo, lo que repercute en el factor tiempo tanto de la evaluación heurística como de los evaluadores, a prescindir de aspectos importantes para la elaboración de la lista de comprobación y a la anotación de resultados subjetivos [10].

Por lo tanto, se evidencia la necesidad de desarrollar una aplicación que apoye el proceso de la evaluación heurística, facilitando de esta manera tareas como la elección de principios heurísticos, creación y distribución del conjunto de criterios heurísticos con su debida puntuación y la extracción de resultados debidamente comparados o propuestas de mejora.

1.2.2. Formulación del problema

¿Cómo apoyar el proceso de evaluación heurística de sistemas interactivos mediante múltiples conjuntos heurísticos?

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar una aplicación web que apoye el proceso de evaluaciones heurísticas en sistemas interactivos.

1.3.2. Objetivos específicos

1. Implementar un módulo que permita configurar el ambiente de trabajo de una evaluación heurística.

Resultados Esperados:

- Categorización de principios heurísticos a partir del tipo de sistema interactivo a evaluar.
- Conjunto inicial de evaluación que incluye:
 - Selección de principios heurísticos para la realización del análisis de usabilidad a partir, del conjunto de heurísticas registradas.
 - Conjunto de usuarios de tipo evaluador seleccionados para dar respuesta a los principios heurísticos seleccionados.
- Permitir la gestión de nuevos principios heurísticos para ampliar los conjuntos heurísticos de la aplicación.

2. Implementar un módulo que permita desarrollar una evaluación heurística.

Resultados Esperados:

- Formulario interactivo con los principios heurísticos seleccionados que permitan realizar la evaluación.
- Permitir el intercambio e integración de las evaluaciones resueltas por el grupo de evaluadores.

3. Implementar un módulo que permita analizar los resultados de una evaluación heurística.

Resultados Esperados:

- Reporte sobre los resultados, problemas detectados y conclusiones finales de la evaluación.

4. Implementar un módulo de reportes generales de la aplicación, el cual ayudará al análisis y observar el uso la herramienta para evidenciar su comportamiento.

Resultados Esperados:

- Reporte sobre la cantidad de evaluaciones realizadas, clasificadas por el tipo de sistemas interactivos evaluados.
- Reporte sobre los evaluadores que más han realizado evaluaciones en la aplicación.
- Reporte sobre la cantidad de evaluaciones realizadas, filtradas por tiempo.

5. Realizar pruebas a la aplicación web desarrollada para la evaluación heurística de sistemas interactivos y comparar los resultados con evaluaciones heurísticas manuales, basados en los mismos criterios ambas evaluaciones.

Resultados Esperados:

- Hallazgos y conclusiones referentes a las evaluaciones heurísticas realizadas.
- Análisis comparativo de las evaluaciones realizadas con y sin la aplicación.

1.4. Alcance

Una vez concluido el proyecto, se obtuvo una aplicación web de apoyo para la realización de evaluaciones heurísticas, que tiene las características de ser responsiva para facilitar el uso de la misma en diferentes dispositivos móviles y de usabilidad para lograr una mayor satisfacción del usuario. La aplicación permite crear una evaluación eligiendo las heurísticas a utilizar, también se puede elegir entre tres y cinco evaluadores, a los cuales les llegará una notificación en su cuenta de la aplicación informando su participación en la evaluación heurística. Además la aplicación ofrece módulos para la generación de reportes que coayuden a la toma de decisiones o enriquecer los hallazgos una vez finalizadas las evaluaciones.

En lo referente al módulo de heurísticas de la aplicación, se tiene que los usuarios con los permisos necesarios, podrán registrar múltiples principios heurísticos con el fin de posibilitar evaluaciones de la naturaleza deseada.

El sistema no reemplazará de ninguna manera el criterio humano en lo referente a evaluaciones heurísticas, esto último, debido a que dichas valoraciones tienen componente subjetivo mayoritario, por ende la aplicación servirá como instrumental para apoyar operativamente la labor de evaluar heurísticamente sistemas interactivos. Además, la aplicación no realizará automáticamente la evaluación heurística, siendo necesario en la misma, el factor humano para dar respuesta a los principios heurísticos.

1.5. Estructura del documento

En el capítulo 2 se presentan el marco referencial, donde se encuentra los antecedentes de herramientas tecnológicas e informáticas orientadas al apoyo de las evaluaciones heurísticas y la teoría que fundamenta el proyecto.

En el capítulo 3 se presenta el proceso de ingeniería de software realizado para cumplir con los objetivos propuestos y adquirir los resultados esperados. Junto con esto, se modelan y ejecutan diferentes escenarios de pruebas para medir la efectividad y pertinencia de la aplicación HeuristicTool frente a los objetivos planteados.

Este trabajo finaliza con el Capítulo 4 donde se exponen las conclusiones y trabajos futuros.

Capítulo 2

Marco Referencial

2.1. Antecedentes

En la actualidad son pocas las herramientas que se han diseñado y desarrollado para facilitar o dar apoyo a los procesos de la evaluación heurística, entre estas se pueden describir cinco soluciones informáticas, donde algunas de las características que se pueden encontrar son (I) la elección de heurísticas a evaluar, (II) soporte durante la puntuación de cada uno de los criterios heurísticos y (III) proporcionar algún tipo de resultado tanto cualitativo como cuantitativo. En estas soluciones informáticas se pueden observar dos enfoques de la aplicación dependiendo el tipo de usuario, uno hacia el diseñador de la evaluación heurística y otro a los evaluadores según las funcionalidades que le sean otorgadas.

En la tabla 2.1, se muestra un resumen de características tenidas en cuenta para el desarrollo de la aplicación que se presenta en este trabajo de grado.

Aplicación	Gestión de Heurísticas	Gestión de Evaluadores	Resultados	Perfil Objetivo	Implementación	Lenguaje de Programación	Disponibilidad al Usuario
R-IDE	No	Si	No	Evaluador	Web	PHP/MySQL	No
UsabAipo-GestorHeurística	Si	Si	No	Evaluador	Escritorio	Microsoft Office Access	No
Accusa	No	Si	Informe con problemas detectados	Evaluador	Escritorio	Microsoft Office Access	No
Gestor Heurísticas	Si	Si	Informe cualitativo del nivel de usabilidad	Evaluador	Escritorio	.NET	No
Sírius	Si	Si	Informe con problemas detectados	Diseñador	Web	PHP	No

Tabla 2.1: Cuadro comparativo entre herramientas
Fuente: Elaboración propia

En base a la tabla 2.1, serán expuestas las cinco aplicaciones software con sus respectivas características. La primera herramienta web encontrada es R-IDE, publicada en el año 2006 por E. Kemp y D. SetungaMudalige, está daba soporte al proceso de evaluación heurística, donde se podría planificar las actividades de evaluación para el sistema interactivo, ofreciendo múltiples técnicas de evaluación como recorridos cognitivos, cuestionarios, observaciones y heurísticas [11]. Seguidamente se encuentran tres aplicaciones de escritorio (I) UsabAipo-GestorHeurística, basada en el diseño de un gestor de heurísticas, con la función de almacenar la información pertinente para la experimentación, los datos del sistema a evaluar, los criterios heurísticos a contestar por los evaluadores, como también las puntuaciones y observaciones acerca del sistema evaluado [10]; (II) Accusa, prototipo desarrollad por el grupo de investigación SQUaC

(Software Quality Usability and Certification) del Instituto Tecnológico de Informática de la Universidad Politécnica de Valencia, esta herramienta es capaz de gestionar los datos relacionados con todo el proceso de la evaluación heurística, permitiendo crear nuevas heurísticas, perfil de los evaluadores y brindar informes o listados con los problemas detectados de usabilidad [10]; (III) Gestor Heurísticas, esta aplicación se presentó como proyecto final de trabajo de grado de la carrera Ingeniería Técnica en Informática de Gestión en la Universitat de Lleida, donde aparte de dar soporte en el proceso de realización de evaluaciones heurísticas, se evidenciaban dos módulos que permitían administrar la creación de nuevos proyectos de evaluaciones heurísticas escogiendo los evaluadores, y el módulo evaluador, la cual llevaba el control de todos sus proyectos evaluados [10]. Finalmente se aprecia el aplicativo web Sirius presentado como tesis doctoral para aspirar al título de Doctor en Informática en la Universidad de Oviedo, como principales funcionalidades se encuentra el nivel de usabilidad brindado en forma cualitativa y cuantitativa al terminar la evaluación, criterios a mejorar en el sistema interactivo ordenados por la importancia en el arreglo y la visualización de comentarios asociados de los evaluadores ante un sistema evaluado [7].

A partir de los antecedentes expuestos y la tabla 2.1, entendiéndose por gestión al proceso de crear, modificar, consultar y eliminar, se puede concluir que hasta el momento los avances en facilitar el proceso de evaluaciones heurísticas por medio de herramientas informáticas, no cumplen completamente la integración de los resultados de todos los evaluadores, del mismo modo, no apoyan todo el proceso de la evaluación heurística teniendo en cuenta los diseñadores de la evaluación y los evaluadores. Por otro lado, los antecedentes expuestos no se encuentran disponibles para el uso de los usuarios, son herramientas las cuales se desarrollaron como objetivo de grado de diversas instituciones académicas y actualmente no prestan sus servicios a los evaluadores.

2.2. Marco Teórico

El estudio de las características comunes de los sistemas está basado en la “Teoría General de Sistemas (TGS)” de Bertalanffy [12], que se caracteriza por su perspectiva holística e integradora, donde priman las relaciones y los conjuntos. La idea más elemental de sistema lo define como un conjunto estructurado de partes interactuantes e interdependientes relacionadas entre sí que conforman un todo. El pensamiento sistémico con mayor interés para la Informática define al sistema como una composición de entradas, procesos y salidas; las entradas o insumos son los ingresos del sistema, pueden ser recursos materiales, humanos o información, constituyendo la fuerza de arranque que suministra al sistema sus necesidades operativas; el proceso transforma una entrada en salidas y las salidas son los resultados obtenidos de procesar las entrada o insumos [13].

En relación con lo anterior, los sistemas de información (SI) pueden ser considerados como segmentos de software que desempeñan tareas concretas, las cuales al interrelacionarse conforman un todo. En tal sentido, el concepto de SI se entiende como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de una organización; generalmente los sistemas de información deben cumplir con tres actividades que producen la información: (I) Entrada, captura o recolección de datos requeridos para procesar, (II) Procesamiento, transformación de datos fuente en información, (III) Salida, transferencia de información procesada al exterior; además, los SI requieren retroalimentación, que es la salida que es el resultado, que permite evaluar o corregir la etapa de entrada [14]. La figura 2.1 muestra la composición de un sistema de información.



Figura 2.1: Composición de un SI

Fuente: Basado de [13]

La creación de sistemas de información hace parte de un proceso ingenieril, en el cual, sobresalen las etapas de análisis, diseño, desarrollo y pruebas, siendo estas fases las más tradicionales dependiendo del tipo de metodología a utilizar. Como objetivo del desarrollo de software, se busca alcanzar un conjunto de características o propiedades, entre las cuales se encuentran la eficiencia, flexibilidad, confiabilidad, mantenibilidad, portabilidad, integridad, seguridad y usabilidad, que al cumplirlas se logra obtener un producto de calidad [2]. Actualmente, la Organización Internacional para la Estandarización propone dos definiciones relativas a la usabilidad del software, (I) ISO/IEC 9126 considera la usabilidad como la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso; (II) ISO/IEC 9241 aporta que la usabilidad es la efectividad, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico.

Por otro lado, interactivo se define como la acción o comunicación que se ejerce recíprocamente entre dos o más objetos, agentes, personas, o una combinación entre ellas. En el campo de las tecnologías de la información, es común ver la interacción entre personas con diversos dispositivos para llevar a cabo distintas tareas. Estos sistemas que interrelacionan y dependen de las acciones de un usuario para la realización de tareas es lo que denominaremos sistemas interactivos, es decir, todo sistema en el cual interactúan persona y máquina [15] de tal manera que se evidencie la comunicación y el intercambio de información entre ambas partes. Es importante considerar que al hablar de videojuegos, no se hablaría de usabilidad sino de jugabilidad.

En consecuencia, el termino usabilidad, la cual es una traducción directa del término anglosajón usability, es definido como la ciencia que estudia la forma de diseñar sistemas interactivos para que los usuarios puedan interactuar con ella de una manera fácil, intuitiva y cómoda [3]. Considerando lo anterior, la usabilidad ha ido evolucionando, surgiendo así especialidades como por ejemplo, la IPO (Interacción Persona-Ordenador) que se ocupa de estudiar el comportamiento de las personas al utilizar sistemas tecnológicos y todas las características presentes en el proceso de interacción entre el usuario y la máquina; el DCU (Diseño Centrado en el Usuario) conlleva a realizar un diseño pensado en y para el usuario, convirtiéndole como punto central del desarrollo; el MPIu+a (Modelo de proceso de la Ingeniería de la Usabilidad y de la Accesibilidad) basado en el DCU buscando añadir al desarrollo de software una serie de actividades estructuradas y explícitas a realizar [16]; por otro lado, se encuentra UX (Experiencia de Usuario) que pretende optimizar la facilidad de uso de los procesos, organizar la información de acuerdo al modelo mental del público objetivo, colaborar en el diseño de la interfaz de las aplicaciones, entre otros [17].

Teniendo en cuenta las definiciones anteriores, y la aplicabilidad de los mismos en las diferentes ciencias y ámbitos de la sociedad, tales como obras civiles, informática, medicina, educación, contabilidad,

entre otras, debemos considerar la usabilidad de los sistemas interactivos como un atributo clave para hablar de calidad [18]. Una heurística, en el campo de la usabilidad, es una guía, principio general o regla que da soporte en la toma de decisiones en el proceso de diseño de un sistema, con el objetivo de obtener un sistema interactivo usable [16].

Para evaluar el nivel de usabilidad podemos encontrar las evaluaciones heurísticas, como un método evaluativo por inspección, presentado como una de las técnicas más utilizadas en este tipo de valoraciones y realizadas por evaluadores a partir de principios establecidos por diferentes autores u organizaciones, tales como la IPO, J. Nielsen, R. Molich, Ben Schneiderman, D. Pierotti, L. Constantine, entre otros [8].

El proceso llevado a cabo en una evaluación heurística se divide en 3 fases o pasos [10]:

1. Planificación de la evaluación: en este paso se debe considerar la elección de los evaluadores (mínimo tres y máximo cuatro) para involucrarlos en el proceso de elección de las heurísticas y, en general, en todo el proceso de planificación del análisis de la usabilidad.

Otro aspecto importante en la planificación de la evaluación, es disponer de un listado de heurísticas, las cuales principalmente se encuentran establecidas en plantillas. La persona responsable de la evaluación, opcionalmente respaldado por los evaluadores escogidos, selecciona los criterios y conjuntos heurísticos más adecuados de acuerdo con el sistema que se va a analizar. El resultado de la primera fase se basa en la obtención de los evaluadores y las heurísticas a evaluar, que se deberán analizar en la siguiente fase.

2. Realización de la evaluación: una vez escogidos las heurísticas más idóneas para realizar la evaluación y se han seleccionado los evaluadores, se debe realizar el análisis de cada una de las heurísticas. El análisis se debe realizar de forma individual para que los resultados de un evaluador sean de su propia subjetividad. Los evaluadores van resolviendo cada una de las heurísticas establecidas en el paso anterior, cuyo resultado puede ser cuantitativo o cualitativo. También es recomendado ofrecer la posibilidad de añadir comentarios en lenguaje natural a cada uno de los criterios heurísticos propuestos.
3. Extracción de resultados de evaluación: una vez realizadas las evaluaciones heurísticas por todos los evaluadores, se pasa a realizar la confrontación de las respuestas de cada uno y su percepción acerca del sistema interactivo, consensuando las diferencias de puntuaciones entre las heurísticas. Seguido se extraen los problemas de usabilidad detectados y los aspectos a mejorar del sistema interactivo evaluado.

Dentro de los principios heurísticos más usados y reconocidos encontramos el elaborado por Jacob Nielsen y Rolf Molich [7], los cuales están basados en recomendaciones para verificar puntos críticos en interfaces de usuario. Estos principios heurísticos son los que se mencionan a continuación:

1. Visibilidad del estado del sistema. Se debe siempre mantener a los usuarios informados acerca de lo que ocurre en el sistema, es decir, una retroalimentación apropiada dentro de un tiempo razonable.
2. Similitud entre el sistema y el mundo real. Se debe mantener la comunicación con el usuario con un lenguaje natural, procurando que la información aparezca en un orden lógico y natural.
3. Control y libertad del usuario. Se debe ofrecer al usuario funciones para deshacer y rehacer las acciones que haya realizado.

4. Consistencia y estándares. Los usuarios no tienen que preguntarse si distintas palabras, situaciones o acciones significan lo mismo, por lo tanto se debe seguir con las convenciones y consistencia en toda la aplicación.
5. Prevención de errores. A parte de proporcionar mensajes de error, se debe procurar tener un diseño cuidadoso evitando que estos ocurran.
6. Preferencia al reconocimiento frente a la memorización. Los objetos, iconos o imágenes son buena ayuda para recordar información o significado en la aplicación.
7. Flexibilidad y eficiencia de uso. Contener atajos en la aplicación hacen más eficiente la aplicación para un usuario experto, por ende se recomienda hacer más rápida la interacción con ciertas acciones frecuentes.
8. Estética y diseño minimalista. Se debe reducir al máximo la cantidad de diálogos con información irrelevante o que de rara vez sea de utilidad, para sobresaltar la información relevante. Ayudar a los usuarios para reconocer, diagnosticar y recuperarse de los errores. Los mensajes y alertas de error deben estar expresados en lenguaje natural, entendible para el usuario indicando el problema y sugiriendo alguna solución.
9. Ayuda a los usuarios para reconocer, diagnosticar y recuperarse de los errores. Los mensajes de error deben estar expresados en lenguaje simple, indicando con precisión el problema y sugiriendo una solución.
10. Ayuda y documentación. Lo deseable de una aplicación es que pueda usarse sin documentación, sin embargo proporcionar ayuda y documentación puede ser beneficioso para todo tipo de usuarios.

En cuanto a los ítems evaluables, se dejan a la subjetividad del evaluador y el tipo de sistema interactivo a evaluar. La métrica asociada para este principio heurístico no proporciona ningún método para obtener un valor cuantitativo, el resultado arrojado es de tipo cualitativo dependiendo los criterios de cada evaluador.

Otra propuesta de principios heurísticos es la llamada UsabAIPO [4], la cual propone una medición de la usabilidad agrupando 4 características de un sistema interactivo, en las cuales se encuentran 25 preguntas en total, asignando un valor 0, 2 ó 4 dependiendo el criterio del evaluador. El peso de cada característica es el siguiente:

- Diseño (D): 28 %
- Navegación (N): 28 %
- Contenido (C): 20 %
- Búsqueda (B): 24 %

De esta manera, la fórmula mediante la cual estos autores obtienen el valor porcentual de usabilidad de un sistema interactivo es la siguiente:

$$UsabAipo = \frac{D}{0,28} + \frac{N}{0,28} + \frac{C}{0,2} + \frac{B}{0,24}$$

Fórmula: UsabAipo - Porcentaje de usabilidad.

2.3. Marco Conceptual

Las tecnologías informáticas recomiendan la utilización de frameworks de desarrollo, ya que estos ofrecen una estructura conceptual y tecnológica con un conjunto de bloques predefinidos de software los cuales se pueden extender o personalizar de acuerdo a las necesidades de un proyecto. En la ingeniería de software la palabra framework es usada para describir una diversidad de modelos y ambientes que tienen por común una característica mínima: establecen un marco de trabajo para desarrollar un producto de un género determinado; o también se define como la abstracción del esqueleto de una solución a una serie de problemas relacionados [19]. Un framework por tanto puede ser definido como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación específica.

Comúnmente los frameworks web están basados en el patrón de arquitectura MVC (Model View Controller). En la figura 2.2, se muestra como este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.

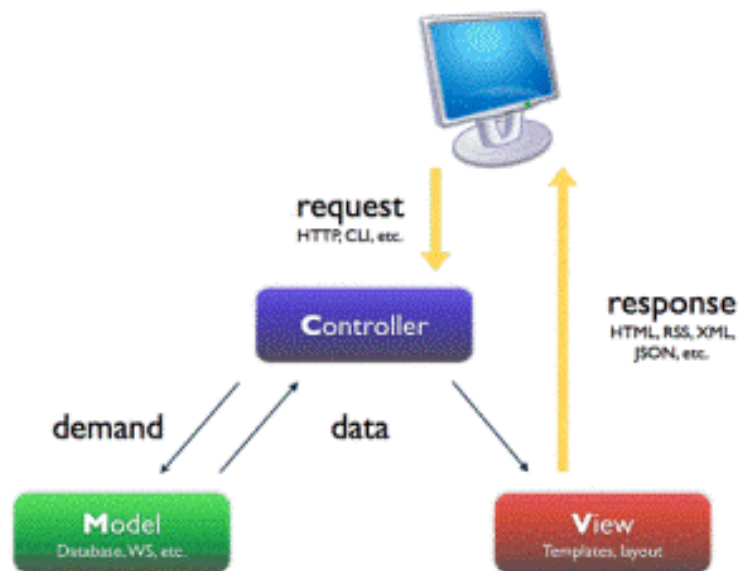


Figura 2.2: Arquitectura MVC aplicada a un framework web

Las siguientes son algunas características de los frameworks web.

- Utilizan un solo servlet que tiene la función de controlador, para toda la aplicación o gran parte de ella.
- Se configura el servlet controlador a través de propiedades, para indicarle a quien delegar la responsabilidad de atender peticiones entrantes. En algunas ocasiones esas propiedades están indicadas de acuerdo a los URLs y el URL entrante.
- Las vistas pueden tener nombres claves, sin necesidad que exista una relación con el nombre del archivo de la vista. El framework se encarga de realizar la conversión para obtener el nombre de la

vista que se debe cargar para ser desplegada.

En la actualidad existe una gran variedad de proyectos de frameworks web para los diferentes lenguajes de programación .NET, Ruby, Java, entre otros. Esto debido a la facilidad que ofrecen a la hora de crear sistemas desde lo más simple, hasta sistemas cada vez más robustos. Como una de las plataformas de desarrollo web más conocidas se encuentra PHP, por su flexibilidad, potencia, alto rendimiento, entre otras características. Por lo que este lenguaje posee un gran abanico de frameworks que son importantes de analizar, entre ellos se encuentran los siguientes [20]:

2.3.1. Laravel

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Laravel se basa en arquitectura MVC y tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla permitiendo multitud de funcionalidades.

Principales características

- Cuenta con un código modular y extensible por medio de un administrador de paquetes.
- Posee un sistema de enrutamiento rápido y eficiente, HTTP Routing, que permite relacionar las partes de la aplicación con las rutas que ingresa el usuario en el navegador.
- Ofrece el uso de Middleware para el análisis y filtro de llamadas HTTP en el servidor, con el fin de evitar problemas de tipo Cross-Site-Scripting (XSS) y otras medidas de seguridad.
- Provee un robusto sistema de caché configurable.
- Provee la autenticación de usuarios de forma nativa e incluye la opción recordar al usuario. Además permite incluir parámetros adicionales.
- Provee una API que permite la automatización de tareas con la tecnología Gulp, definiendo el uso pre-procesadores para comprimir CSS y JavaScript.
- Cuenta con la encriptación de datos, OpenSSL y cifrado AES-256-CBC.
- Facilita la definición, registro y escucha de eventos con la propiedad listen de EventServiceProvider, la cual contiene una lista de los eventos registrados en la aplicación.
- Incluye una capa para manejo de bases de datos que cuenta con un ORM (Object-Relational-Map) llamado Eloquent.
- Ofrece un framework transversal para realizar Unit Testing, PHPUnit.
- Cuenta con Micro-servicios y APIs para proyectos de menor envergadura.
- Se puede ejecutar procesos largos y complejos en segundo plano usando listas de tareas.
- Facilita la integración del desarrollo con el sistema de cobros Stripe.

2.3.2. Cakephp

Es un framework de código abierto para el desarrollo aplicaciones web escrito en PHP, creado sobre los conceptos de Ruby on Rails. Utiliza el patrón de diseño MVC y su objetivo principal es que el trabajo sea de forma estructurada y rápida, sin pérdida de flexibilidad.

Principales características

- Es compatible con PHP4 y PHP5.
- Posee licencia MIT.
- Provee un CRUD integrado para la interacción con la base de datos.
- Soporta scaffolding, una técnica para especificar el uso de la base de datos en la aplicación.
- Provee un despachador de peticiones que permite acceder a la aplicación a través de URLs amigables y configurables.
- Permite la generación de plantillas de manera rápida y flexible usando la sintaxis de PHP y asistentes o helpers.
- Incorpora asistentes de construcción de vistas para la automatización de la generación de código en AJAX (Asynchronous JavaScript and XML), JavaScript, formularios HTML, entre otros.
- Ofrece componentes de email, cookie, seguridad, sesión y manejo de solicitudes.
- Implementa Data Sanitization para determinar el ingreso y formato de los datos según las reglas de validación.

2.3.3. Yiiframework

Es un framework MVC de alto rendimiento basado en componentes, PHP y framework de aplicaciones web. Yii es un acrónimo para Yes, it is (en español: ¡Sí lo es!).

Principales características

- Incluye Database Access Objects (DAO), query builder, Active Record y migración de BD.
- Incorpora formularios de entrada, validación y soporte para AJAX.
- Provee Gii, una herramienta de generación de código para acelerar el desarrollo de aplicaciones.
- Permite la integración con otros frameworks, como Zend o PEAR.
- Ofrece variedad de plugins y widgets gratuitos de código abierto.
- Ofrece herramientas para realizar pruebas unitarias y funcionales, basadas en PHPUnit y Selenium.
- Soporta el esquema de caching por capas y permite el cambio del almacenamiento del cache.
- Permite la generación automática de WSDL, especificaciones y administración de peticiones Web Service.

2.3.4. Codeigniter

Es un framework pequeño de código abierto para la creación rápida de aplicaciones web. Contiene una serie de librerías estructuradas que sirven para el desarrollo de soluciones web, implementa el paradigma MVC.

Principales características

- Es compatible con PHP4 y PHP5.
- Compatible con un gran número de entornos o servidores, incluyendo sistemas de alojamiento compartido.
- Ofrece documentación fácil de seguir y asimilar, porque está escrita en modo tutorial.
- Tiene una curva de aprendizaje baja, lo que hace que sea menos rígido que otros frameworks.
- Permite la aplicación de módulos o clases de manera opcional, así su núcleo se vuelve bastante ligero, lo que permite que el servidor no se sobrecargue.

2.3.5. Simfony 2

Es un framework de código abierto diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón MVC. Symfony trabaja separando la lógica de negocio, la lógica de servidor y la presentación de la aplicación web.

Principales características

- Posee licencia MIT.
- Tiene independencia del sistema gestor de base de datos. Su capa de abstracción y el uso de ORM, permite cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.
- Permite la integración de bibliotecas de otros fabricantes.
- Permite la internacionalización para la traducción del texto de la interfaz, los datos y el contenido de localización.
- Ofrece el uso de templates y layouts que pueden ser contruidos por diseñadores de HTML que no posean conocimientos del framework.
- Posee un sistema de enrutamiento y URLs inteligentes.
- Soporta validación automática de formularios, lo cual asegura mejor calidad de los datos en las base de datos.

En la tabla 2.2, se muestra un resumen de características tenidas en cuenta para el desarrollo de la aplicación que se presenta en este trabajo de grado.

PHP Framework	ORM	Templates	Caching	Módulo Autenticación	Módulos	Pruebas
<i>Laravel</i>	✓	✓	✓	✓	✓	✓
<i>Cakephp</i>	✓		✓	✓	✓	✓
<i>Yii</i>	✓	✓	✓	✓	✓	✓
<i>Codeigniter</i>		✓	✓			✓
<i>Simfony 2</i>	✓	✓	✓	✓	✓	✓

Tabla 2.2: Cuadro comparativo entre frameworks PHP

Fuente: Elaborado a partir de [20]

En el ejercicio de conocer los diferentes framework PHP, se pudo concluir que Laravel es el apropiado para el desarrollo del software HeuristicTool, ya que cumple con los criterios descritos en la tabla 2.2 y conceptos cómo: desarrollo activo, comunidad activa y buena documentación.

2.4. Plataforma de Desarrollo

2.4.1. PHP (Hypertext Preprocessor)

PHP es un lenguaje de código abierto muy popular y especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Esta tecnología permite realizar páginas web dinámicas cuyo contenido puede ser completa o parcialmente generado en el momento de invocar la página, gracias a la información obtenida de un formulario o extraída de una base de datos. La característica que distingue a PHP de otros lenguajes de programación es su ejecución del lado del servidor, es decir, el código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página resultante para ser enviada al explorador [21].

2.4.2. MySQL

MySQL es un sistema administrador de bases de datos relacionales (RDBMS, Relational Data Base Management System) de código abierto, desarrollado en C, C++ e independiente del sistema operativo. Este gestor a su vez puede verse como un sistema cliente/servidor, ya que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, es decir, cada vez que se establece una conexión con el servidor este crea un subproceso para manejar la solicitud del cliente, controlando el acceso simultaneo de un gran número de usuarios a los datos y asegurando el acceso [21].

El conjunto de características de MySQL han logrado que se convierta en uno de los sistemas gestores de bases de datos más utilizados en la actualidad, no solo por pequeñas empresas sino también por algunas grandes corporaciones, como: Wikipedia, Google, Facebook, Twitter, Flickr y YouTube.

2.4.3. HTML

HTML es el lenguaje utilizado en Internet para definir las páginas delWorldWideWeb, básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que pueden componer una página web, como: imágenes, listas, videos, entre otros. Cabe destacar que HTML no es un lenguaje de programación ya que no cuenta con funciones aritméticas, variables o estructuras de control propias de dichos entornos, por lo que HTML genera únicamente páginas web estáticas, sin embargo, este se puede usar en conjunto con diversos lenguajes para la creación de sitios dinámicos.

2.4.4. CSS (Cascading Style Sheets)

CSS, en español Hojas de Estilo en Cascada, es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla, por impresora, por voz (cuando la información es pronunciada a través de un dispositivo de lectura) o en dispositivos táctiles basados en Braille.

CSS es una especificación desarrollada por el W3C (World Wide Web Consortium) para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra, entre otros, modificando así la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos.

2.4.5. JavaScript

Son interfaces web con CSS y JavaScript, que ayudan a adaptar la interfaz dependiendo del tamaño del dispositivo en el que se visualice. A la hora de crear interfaces con Bootstrap permite crear diseños simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El framework contiene plantillas de diseño basadas en HTML y CSS con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz así como extensiones opcionales de JavaScript.

2.4.6. jQuery

jQuery es una librería de JavaScript, rápida, pequeña y con muchas funcionalidades, que permite la manipulación de los objetos DOM, eventos, animaciones y la llamada a funciones AJAX, a través de un API, de una manera mucho más simple y sencilla.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Publica General de GNU v2, permitiendo su uso en proyectos libres y privados. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

2.4.7. GitHub

La colaboración en Ingeniería de Software se ha hecho más habitual ya que es orientada a brindar la posibilidad de desarrollar usando herramientas que trabajan en la Web y de implementar instrumentos de social media dentro de las mismas. En este auge social se ven involucrados los sistemas de control de versiones (SCV) con la bondad de funcionalidades claves para el desarrollo de proyectos como es el control de cambios en el código, la transformación de dichos cambios, y la posibilidad de colaborar en el

desarrollo del código.

Git es un SCV diseñado para la gestión eficiente de flujos de trabajo distribuido no lineales, este sistema gestiona a GitHub[22] un servicio comercial de alojamiento de repositorios que proporciona una interfaz web donde permite al usuario crear repositorios vacíos o por clonación de otro repositorio hospedado en GitHub (fork), enviar solicitudes de cambio entre repositorios hospedados (pull request) y gestionar dichas solicitudes.

Además del alojamiento, GitHub[23] proporciona a cada repositorio una wiki, un gestor de tareas (issues), un sistema de gestión de comentarios e incluso un cuadro de control con grafos sociales.

Capítulo 3

Desarrollo del proyecto

En el presente capítulo se presenta el proceso de desarrollo de software de la aplicación web con los entregables obtenidos, además se exponen las diferentes tecnologías usadas para la implementación del mismo y se evidencia el logro de los objetivos trazados.

La metodología de desarrollo de software seleccionada para llevar a cabo el proceso de elaboración del presente trabajo de grado fue Kanban. Esta está inspirada en el sistema de producción de Toyota, es una palabra japonesa que hace referencia a “tarjetas visuales”, y se utilizó para controlar el avance del trabajo en el contexto de línea de producción. Kanban tiene como objetivo principal gestionar de manera general como se van completando las actividades, para esto propone un marco visual para el control de las tareas, donde se muestra el qué hacer, qué se está haciendo y qué se ha hecho, estableciendo un límite de tareas en cada fase del proceso. Aunque este método no fue creado para la gestión de proyectos de software, David Anderson de la Unidad de Negocios XIT de Microsoft en 2004 la introdujo en esta área y en la actualidad se sigue haciendo uso de esta metodología [24].

Ordysięski (2013) indica que el marco de trabajo de Kanban para el desarrollo de software puede ser presentado en tres puntos principales [25]:

1. Visualizar el flujo de trabajo.
2. Determinar el límite de trabajo en curso (Work in progress).
3. Medir el tiempo en completar una tarea (Lead time).

Las fases o etapas del muro se deciden según el caso, la figura 3.1 muestra la adaptación de esta metodología para el desarrollo de software de la aplicación web HeuristicTools.



Figura 3.1: Muro Kanban - HeuristicTools

Fuente: Elaboración propia

Todo proceso de desarrollo de software además de comprender unas actividades a ser realizadas también implica el diseño de artefactos que permitan establecer la funcionalidad del sistema, para esto se adoptaron artefactos pertenecientes a las metodologías ágiles y UML (Unified Modeling Language), como fueron los diagramas de casos de uso del negocio disponible en la sección 3.1.1.1 y diagramas de caso de uso de la aplicación disponible en la sección 3.1.2.1, historias de usuario disponible en la sección 3.1.1.2, modelo entidad relación disponible en la sección 3.1.2.2 y el diagrama de componentes que describe en términos de componentes la arquitectura de la aplicación, disponible en la sección 3.3. Dado que Kanban es una metodología ágil y no propone artefactos, esta permite ajustarse a las características del proyecto mencionadas anteriormente y puede aumentar la flexibilidad, reducir los trabajos innecesarios, mejorar el flujo de entrega y minimizar el tiempo de ciclo de desarrollo.

3.1. Ingeniería

3.1.1. Análisis

Al iniciar esta fase se analizaron las tareas que debían ser realizadas en pro del cumplimiento satisfactorio de los objetivos planteados, para así planificar los módulos y sub módulos que harán parte de la aplicación e identificar los posibles riesgos que puedan afectar el flujo adecuado del proyecto. A partir de este análisis se estructuraron las tareas en el tablero Kanban y se realizaron las siguientes actividades: (I) Diagrama de casos de uso del negocio, describe el proceso de una evaluación heurística (sistema) independiente de la implementación y (II) Historias de usuario, definen las características básicas de la funcionalidad del sistema y la interacción con el usuario, permitiendo controlar las excepciones y las condiciones del correcto funcionamiento de lo implementado mediante un conjunto pruebas de aceptación que determinan si la aplicación desarrollada suple la funcionalidad requerida.

3.1.1.1. Diagrama de caso de uso (Negocio)

Para la comprensión de los procesos de una evaluación heurística en términos de actividades y actores, se representa cada actividad como un caso de uso y estos a su vez son relacionados con los actores influyentes.

Los actores son la abstracción de una persona o sistema que en el negocio realizan una o varias actividades e interactúan con otros actores. En la Figura 3.2 se muestra el diagrama de caso de uso obtenido para el proceso de una evaluación heurística.

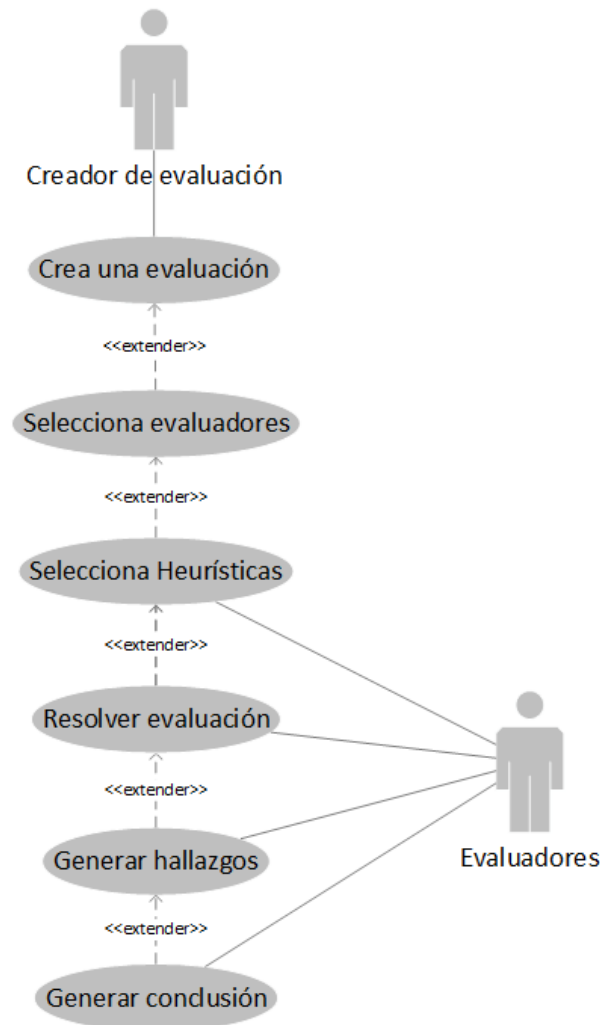


Figura 3.2: Muro Diagrama caso de uso (Negocio) Evaluación heurística

Fuente: Elaboración propia

3.1.1.2. Historias de usuario

Al realizar el análisis funcional del proceso de una evaluación heurística, se logra determinar las funcionalidades que debían hacer parte de la aplicación, esto a su vez implica la planeación de tareas a desarrollar, para ello se utilizó un valor de prioridad el cual fue establecido bajo los criterios de importancia y costo en el marco de desarrollo de la aplicación, como también, la dependencia de las demás historias de usuario sobre la misma. Se establece un criterio de Alta, siendo la más importante, media, de nivel intermedio y de baja siendo uno de las menos importantes. La tabla F.5 muestra la historia de usuario del módulo Evaluación para la creación de una evaluación, las demás especificaciones se encuentra en el apartado de anexos A.

Historia de Usuario	
Número: 6	Autor: Carlos Andrés Riascos Pareja
Nombre de Historia: Crear una evaluación	
Módulo: Evaluación	Iteración: 2
Prioridad: Media	Dependencia:
Responsable: Carlos Andrés Riascos Pareja	
Descripción: Como usuario con los permisos respectivos, podrá crear evaluaciones heurísticas, el formulario de ingreso tendrá la siguiente información: • Tipo de aplicación a evaluar: lista seleccionable – obligatorio. • Propósito: tipo alfabético - obligatorio. • Empresa (para quien se va a realizar la aplicación): tipo alfabético. • Fecha máxima de evaluación: tipo fecha – obligatorio. • Imágenes adjuntas: tipo imagen.	
Validación: Al crear una evaluación se podrá consultar y seguir los pasos de la misma, como seleccionar evaluadores y heurísticas.	
Creada: 14/08/2017	Modificada: 28/09/2017

Tabla 3.1: Historia de Usuario, Crear Evaluación.

Fuente: Elaboración propia

La historia de usuario se compone de una tabla con su número asociado, autor y responsable de la ejecución de la tarea, prioridad y dependencia entre tareas, cantidad de iteraciones en la misma, nombre de la historia y del módulo correspondiente, una breve descripción que narra en lenguaje no estructurado el proceso para llevar a cabo la tarea o el objetivo de la historia, junto con un conjunto de validaciones que permiten verificar los resultados obtenidos, y finalmente una fecha de creación y modificación que permite tener el control de versiones de cada historia.

3.1.2. Diseño

3.1.2.1. Diagrama de casos de uso

Describen los requisitos funcionales del sistema en términos de la secuencia de interacciones entre el sistema y uno o más actores, en respuesta a un estímulo inicial proveniente de un actor, es una descripción de un conjunto de escenarios, cada uno de ellos comenzado con un evento inicial desde un actor hacia el sistema. En la figura C.1 se presenta el diagrama de caso de uso para el módulo de evaluación, los diagramas referentes a los demás módulos se encuentran en el apartado de anexos B.

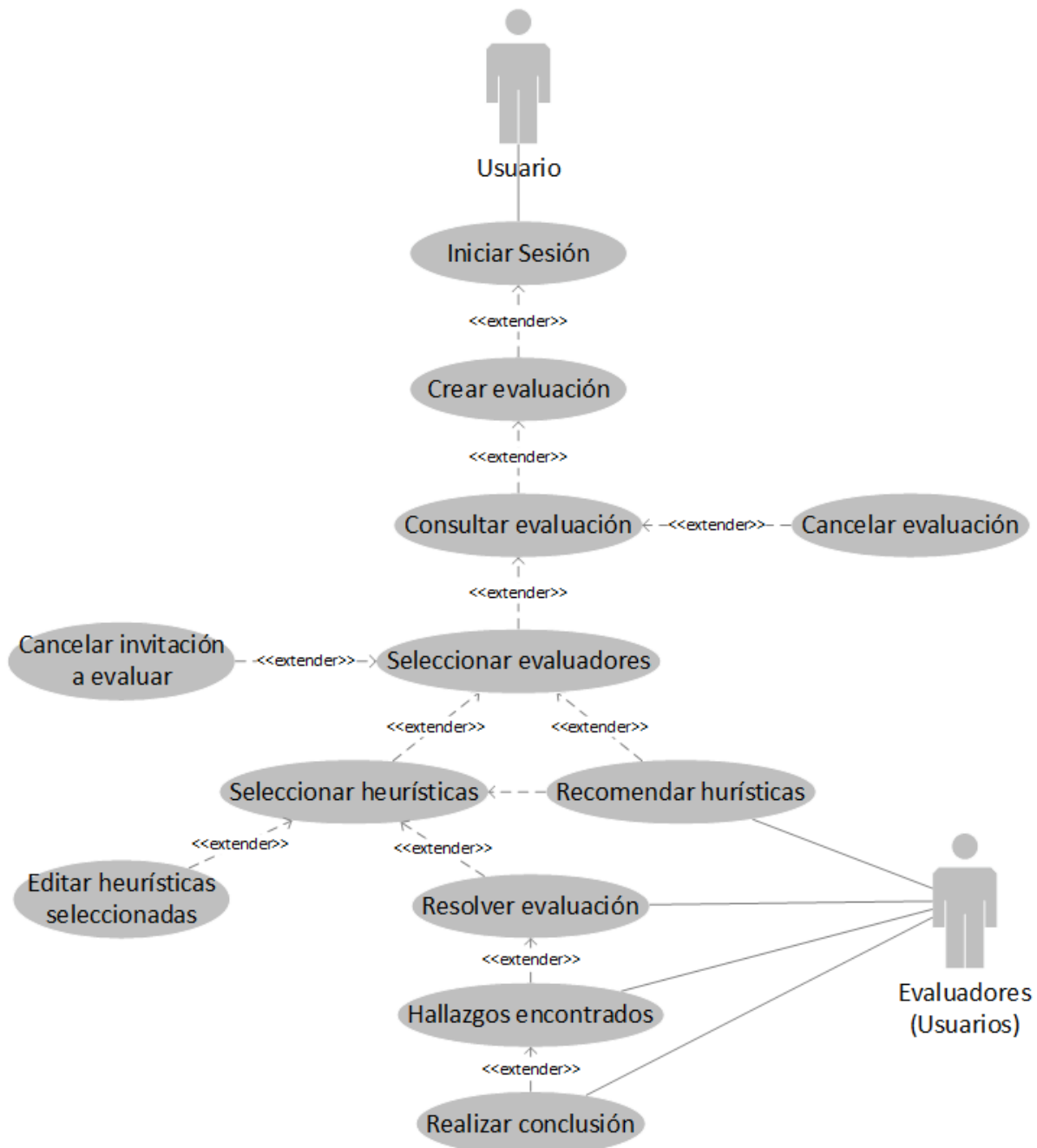


Figura 3.3: Diagrama caso de uso - Módulo evaluación.

Fuente: Elaboración propia

3.1.2.2. Modelo entidad relación

El modelo de datos de la aplicación HeuristicTools, se trabajó bajo el esquema de la base de datos relacional MySQL, que se presenta en el anexo C. En la figura 3.4 se muestra el segmento del modelo ER perteneciente al módulo de heurísticas, en esta se puede evidenciar la relación entre las diferentes entidades que intervienen en este proceso y los atributos de las mismas.

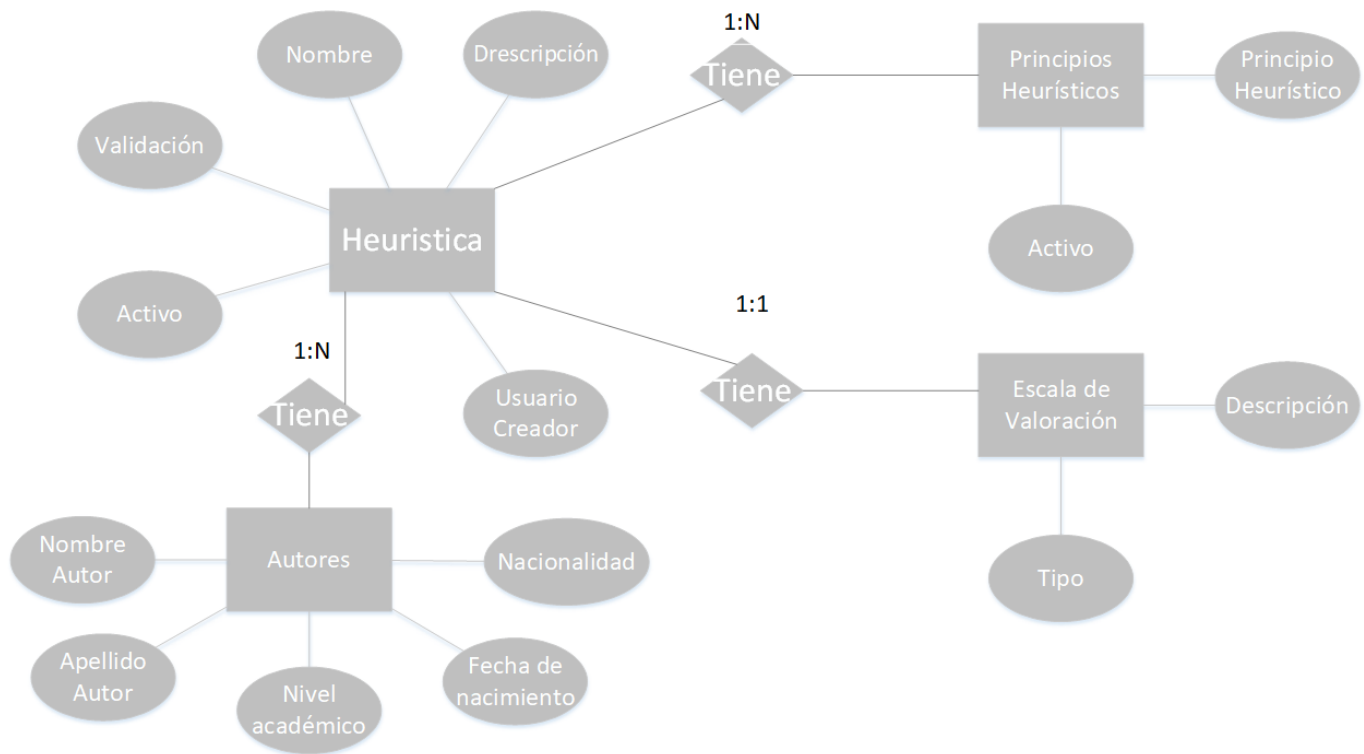


Figura 3.4: Modelo Entidad Relación - Módulo Heurísticas.

Fuente: Elaboración propia

3.2. Descripción del sistema

En esta sección se presenta de manera general la aplicación web HeuristicTool con tres módulos que reúnen las funcionalidades necesarias para la sistematización del proceso de evaluar mediante heurísticas un sistema interactivo. Dentro de la aplicación intervienen cuatro perfiles autor, desarrollador, evaluador y el administrador del aplicativo, estos comparten la misma interfaz principal compuesta de tres secciones: (I) Área de control, (II) Área de funcionalidades (módulos) y (III) Área contenedora de formularios y/o información. Actualmente la aplicación se encuentra alojada en un servidor web y contempla la facilidad de acceder desde diferentes dispositivos, al contar con interfaces de usuarios responsivas.

En los siguientes apartados se exponen las diferentes secciones de las que se componen las interfaces principales para cada uno de los perfiles anteriormente mencionados y se abordan los resultados de la aplicación en función del cumplimiento de los objetivos del proyecto.

3.2.1. Interfaz gráfica de usuario

La interfaz principal para los diferentes perfiles de la aplicación se basan en tres secciones importantes como fue mencionado anteriormente: (I) Área de control, comprende funciones de accesibilidad de las que puede hacer uso el usuario conforme a su necesidad, estas son cambiar el color y tamaño de la fuente, además, permite realizar acciones sobre la cuenta autenticada; (II) Área de funcionalidades, constituye los diferentes módulos y submódulos en los cuales el usuario puede realizar acciones sobre la aplicación, la disponibilidad de estos dependerán de los permisos concedidos al perfil adoptado por el usuario; y

(III) Área contenedora de formularios y/o información. La Figura 3.5 muestra la interfaz principal de HeuristicTool para el perfil del administrador de la aplicación.

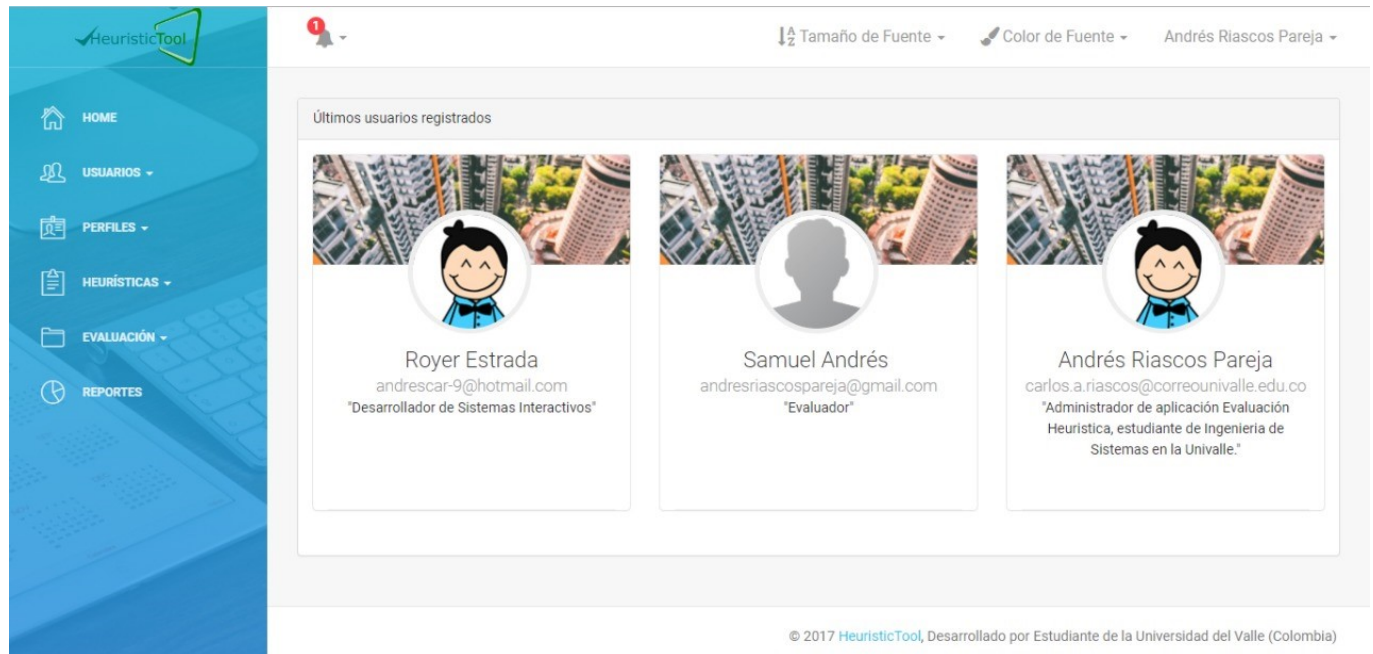


Figura 3.5: Interfaz principal administrador de HeuristicTool.

Fuente: Elaboración propia

3.2.2. Módulo Heurísticas

La gestión de las diferentes heurísticas que harán parte de la aplicación fue implementada en el módulo heurísticas; además permite la integración de los diferentes componentes de una heurística, por ejemplo los principios heurísticos y la escala de valoración de los mismos, a continuación se detallan brevemente las funcionalidades del mismo.

3.2.2.1. Submódulo Principios

Este submódulo comprende la gestión de los principios heurísticos que pueden ser aplicados en la evaluación de un sistema interactivo, estos son comprendidos como las preguntas que deben ser resueltas en base a la escala de valoración relacionada.

3.2.2.2. Submódulo Escala de valoración

Este submódulo registra los diferentes rangos cuantitativos o cualitativos junto a su descripción, esto facilita las posibles respuestas que pueda dar el evaluador frente a un principio heurístico.

3.2.2.3. Submódulo Autores

Este submódulo almacena la información del autor(es) de cada heurística registrada en la aplicación, esto con el fin de cumplir con la política de derechos de autor.

3.2.3. Módulo Evaluación

El módulo de evaluación comprende la gestión de las diferentes evaluaciones aplicadas a sistemas interactivos, relacionándose mediante diferentes mecanismos con el módulo de heurísticas y el módulo de usuarios, esto con el fin de permitir el desarrollo normal de una evaluación.

3.2.3.1. Submódulo Configuración Ambiente de Evaluación

Este submódulo permite la selección de: (I) Evaluadores, los cuales deben ser usuarios registrados en la aplicación, y (II) Heurística a ser aplicada en el proceso; además facilita la invitación a un nuevo usuario mediante un mensaje al correo electrónico para que cumpla el rol de evaluador de un determinado sistema. Por otro lado, los evaluadores pueden recomendar a su equipo de trabajo el uso de heurísticas que consideren pertinentes para el proceso de evaluación.

3.2.3.2. Submódulo Solución de Evaluación

Este submódulo permite poner en marcha la evaluación de un sistema interactivo, donde cada evaluador podrá dar respuesta a cada uno de los principios heurísticos relacionados al proceso y su percepción acerca del nivel de usabilidad del sistema evaluado.

3.2.3.3. Submódulo Reporte de Evaluación

En este submódulo se presenta el proceso más importante de HeuristicTool, ya que facilita el intercambio de información y fomenta la eficiencia en la comparación de resultados para lograr un análisis más detallado con propuestas de solución pertinentes. Lo anterior se logra gracias a la recopilación de la valoración que cada evaluador hizo sobre los principios heurísticos aplicados a un determinado sistema interactivo y al conocimiento de las opiniones individuales.

3.2.4. Módulo Reportes

El módulo de reportes permite el análisis de los datos y del uso de la aplicación mediante gráficos estadísticos, también facilita la exportación de los reportes en formato pdf y xls para el estudio adicional de los mismos.

3.2.5. Módulo Perfiles

Este módulo permite la gestión de perfiles que serán asignados a los usuarios para que hagan uso de las diferentes funcionalidades que la aplicación les permita en cada módulo y/o submódulo.

3.2.6. Módulo Usuarios

En este módulo se gestiona la información de los usuarios que hacen parte de la aplicación, a esta gestión se le añaden acciones como restablecer contraseña, cambio de contraseña, autenticación de usuarios y administración del perfil.

3.3. Arquitectura del Sistema Implementado

La aplicación Web HeuristicTools se basa en una arquitectura modelo-vista-controlador que se conoce como Arquitectura MVC. La Figura 3.6 muestra la arquitectura sobre la cual se soporta HeuristicTools mediante un diagrama de componentes.

La capa de presentación (vista) fue elaborada a partir de la plantilla Light Bootstrap Dashboard [26], en esta se usa la librería Bootstrap y diferentes plugins JQuery que permiten generar interfaces responsivas, siendo esto un punto clave para las tendencias actuales en el uso de dispositivos móviles o tablets en la Internet. La comunicación con el servicio Web se hace a través de HTTP junto con la tecnología AJAX. La capa lógica (controlador) la constituye los diferentes componentes de Evaluación, Evaluadores, Reportes, Heurísticas, Autores Heurísticas, Usuarios y Perfiles, que establecen las funcionalidades principales de la aplicación y conforman los diferentes modelos y controladores que a su vez se ven apoyados por el componente controlador de peticiones que tiene por función el gestionar las solicitudes provenientes de la capa de presentación y direccionarlas al siguiente componente de servicios o controladores, en el cual se procesan las peticiones y se determinan los servicios web que utilizan los Modelos o funcionalidades requeridas.

En relación a los modelos se encuentra el componente Model, en este se gestionan las consultas hacia el motor de base de datos y la carga dinámica de componentes e información a las interfaces de usuario de acuerdo a la petición que reciba la capa lógica. La capa de datos por su parte, tiene como responsabilidad interactuar con la capa lógica y mantener la información de la aplicación disponible.

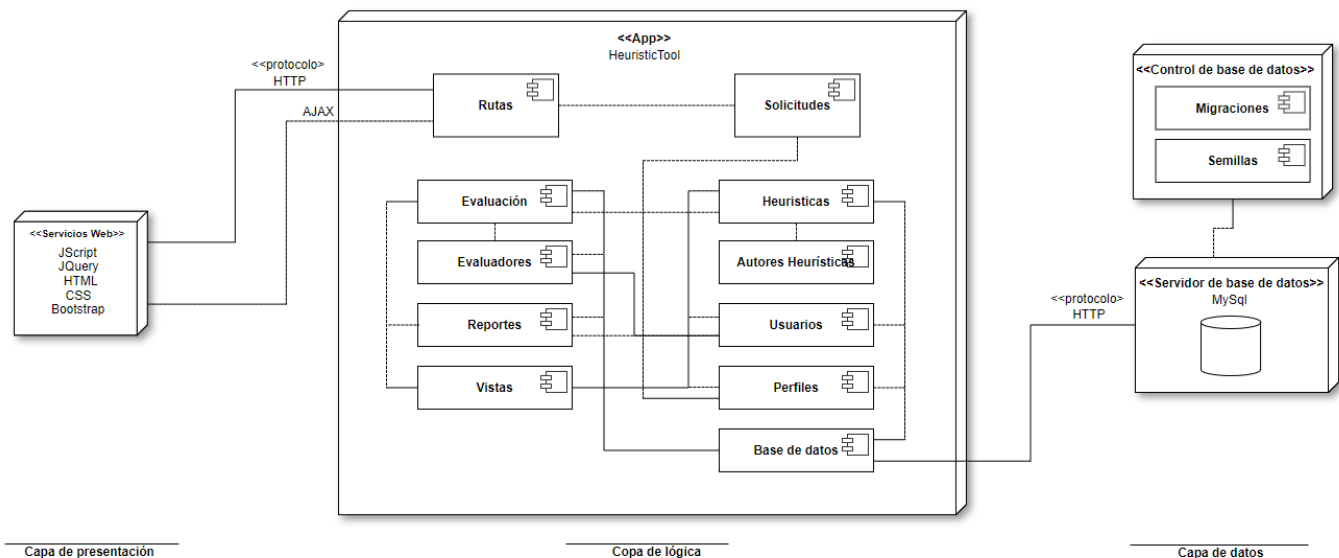


Figura 3.6: Arquitectura HeuristicTool.

Fuente: Elaboración propia

3.4. Detalles de implementación

Para la elaboración de la aplicación web HeuristicTool se usó el lenguaje de programación PHP5, integrado con JavaScript y JQuery, facilitando de esta manera el desarrollo web de contenido dinámico y la integración con servidores. Se dispuso de Hosting Latinoamérica como contenedor para probar y desplegar la aplicación. Como fue mencionado en la sección anterior, para generar la interfaz de comunicación entre el usuario y HeuristicTool se utilizó la plantilla Light Bootstrap Dashboard la cual unifica herramientas como HTML y CSS por medio de Bootstrap, además ofrece variados componentes y herramientas gráficas prediseñadas, que aceleran el diseño de la interfaz.

La administración de los datos almacenados en el sistema, se realiza a través del gestor de base de datos web phpMyAdmin en su versión 4.3, aprovechan la capacidad que tiene para ejecutar las características de MySQL (CRUD de datos, tablas, campos e índices, gestión de usuarios y privilegios, entre otros), su compatibilidad con múltiples servidores y la posibilidad de exportación de datos a diferentes formatos.

Es importante denotar que al inicio de este proceso fue necesario investigar sobre los beneficios y dificultades de usar los diferentes lenguajes de programación en la web, esto con el objetivo de determinar cuál sería el más beneficioso para el desarrollo de este proyecto, consecuentemente se debió hacer el mismo proceso en la elección del framework, y finalmente se realizó la curva de aprendizaje de los entornos seleccionados, en este caso PHP5 y Laravel.

Sobre la base de las consideraciones anteriores se dio inicio al desarrollo funcional del sistema, para ello se abarcó cada objetivo como módulo o submódulo independiente. El cumplimiento del primer objetivo *implementar un módulo que permita configurar el ambiente de trabajo de una evaluación heurística*; el cual dio inicio al desarrollo del módulo de evaluación que consta de dos submódulos, evaluadores y selección de heurísticas, para ello fueron programadas tareas en el tablero Kanban que apuntaran al cumplimiento de funcionalidades como creación, consulta y modificación del ambiente de trabajo.

Al alcanzar estabilidad en el desarrollo anterior se procedió al cumplimiento del segundo objetivo, *implementar un módulo que permita desarrollar una evaluación heurística*; en este se tuvo en consideración la necesidad de crear un módulo que contuviera las heurísticas a ser usadas en cada evaluación y la administración de las mismas, para esto se conceptuó el módulo heurísticas; simultáneamente se desarrolló el submódulo solución de evaluación, el cual está alineado al módulo evaluación, este permite el almacenamiento de los hallazgos (aspectos de mejora) encontrados en el sistema interactivo evaluado y las respuestas a cada principio heurístico utilizado.

Posteriormente se procedió a cumplir con el tercer objetivo que daría como finalizado el proceso de realizar una evaluación heurística, *implementar un módulo que permita analizar los resultados de una evaluación heurística*; en consideración a este se implementó el submódulo reporte de evaluación, el cual está alineado al módulo de evaluación, este fue pensando desde la necesidad que tiene toda evaluación de recibir una realimentación para poder efectuar un plan de mejoramiento, por eso se crearon funcionalidades que permitieran agrupar el resultado de todos los evaluadores involucrados en el proceso con el fin de facilitar el análisis de los mismos y generar una conclusión colaborativa que conllevara a la mejora continua del sistema interactivo evaluado.

En las figuras 3.7, 3.8, 3.9 y 3.10 se evidencia el desarrollo de los módulos y submódulos mencionados anteriormente para el cumplimiento de los tres primeros objetivos planteados en este trabajo de grado.

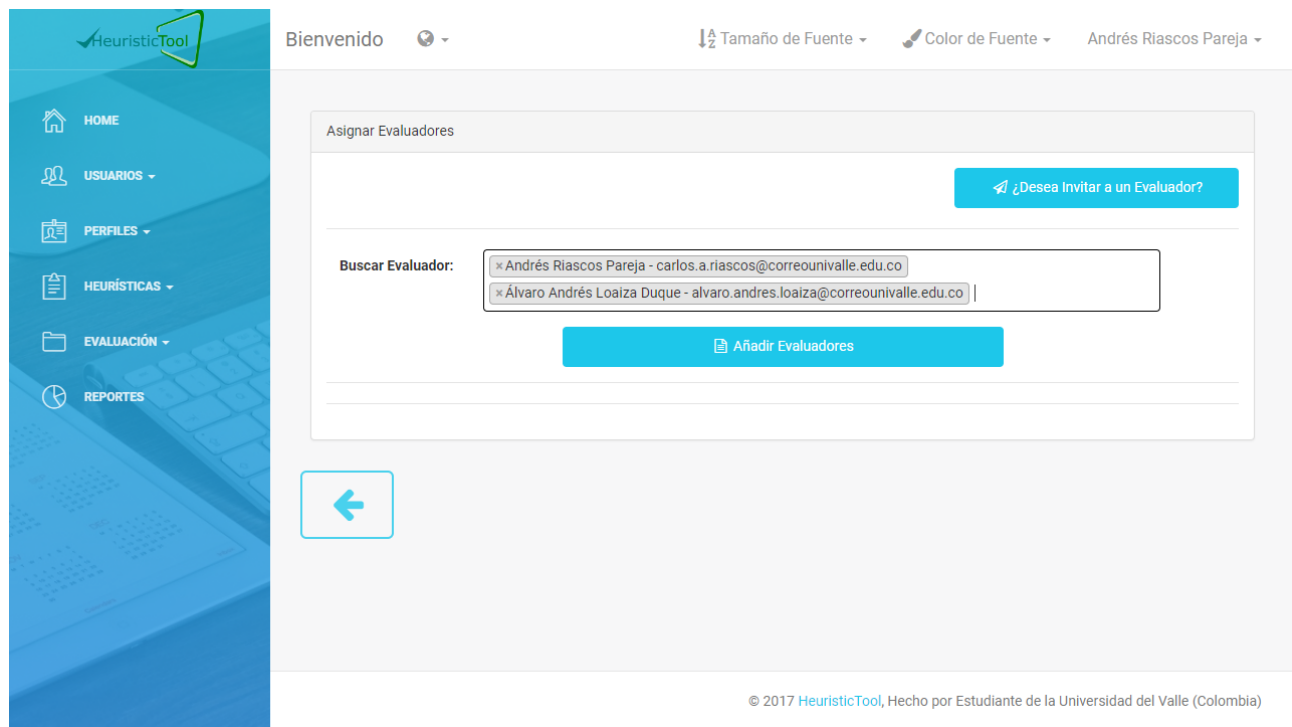


Figura 3.7: Selección de evaluadores
Fuente: Elaboración propia

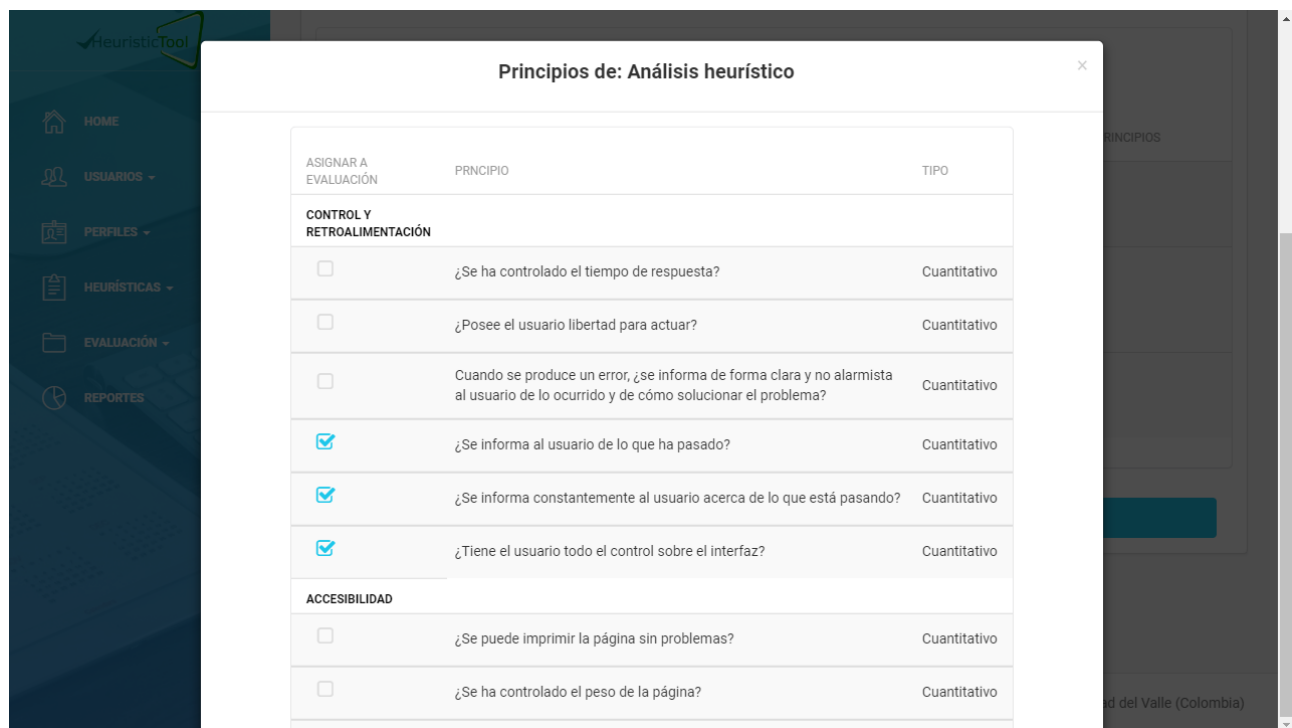


Figura 3.8: Selección de principios.
Fuente: Elaboración propia

HeuristicTool

HOME

USUARIOS

PERFILES

HEURÍSTICAS

EVALUACIÓN

REPORTES

Bienvenido

Tamaño de Fuente

Color de Fuente

Andrés Riascos Pareja

Análisis heurístico

Escala de Valoración

RANGO	DESCRIPCIÓN
1	Se da la mínima expresión del heurístico en las páginas evaluadas
2	Se da una expresión baja del heurístico en las páginas evaluadas
3	Se da una expresión media del heurístico en las páginas evaluadas
4	Se da una expresión alta del heurístico en las páginas evaluadas
5	Se da la máxima expresión del heurístico en las páginas evaluadas

Principios a responder

PRINCIPIO

¿Se informa al usuario de lo que ha pasado?

Seleccione su Valoración

1

2

3

4

5

4

¿Se informa constantemente al usuario acerca de lo que está pasando?

4

Guardar Respuestas

Figura 3.9: Resolver evaluación.
Fuente: Elaboración propia

HeuristicTool

HOME

USUARIOS

PERFILES

HEURÍSTICAS

EVALUACIÓN

REPORTES

Heurísticas respondidas

Respuestas de las heurísticas por cada evaluador

Respuestas de la heurística "Análisis heurístico"

NOMBRE PRINCIPIO	ÁLVARO ANDRÉS LOAIZA DUQUE	ANDRÉS RIASCOS PAREJA
¿Se informa al usuario de lo que ha pasado?	5 - Se da la máxima expresión del heurístico en las páginas evaluadas	4 - Se da una expresión alta del heurístico en las páginas evaluadas
¿Se informa constantemente al usuario acerca de lo que está pasando?	5 - Se da la máxima expresión del heurístico en las páginas evaluadas	4 - Se da una expresión alta del heurístico en las páginas evaluadas
¿Tiene el usuario todo el control sobre el interfaz?	4 - Se da una expresión alta del heurístico en las páginas evaluadas	5 - Se da la máxima expresión del heurístico en las páginas evaluadas
Si posee FAQs, ¿es correcta tanto la elección como la redacción de las preguntas? ¿y las respuestas?	5 - Se da la máxima expresión del heurístico en las páginas evaluadas	1 - Se da la mínima expresión del heurístico en las páginas evaluadas
¿Se ofrece ayuda contextual en tareas complejas?	5 - Se da la máxima expresión del heurístico en las páginas evaluadas	3 - Se da una expresión media del heurístico en las páginas evaluadas
El enlace a la sección de ayuda, ¿está colocado en una zona visible?	4 - Se da una expresión alta del heurístico en las páginas evaluadas	4 - Se da una expresión alta del heurístico en las páginas evaluadas
Si posee una sección de ayuda, ¿Es verdaderamente necesaria?	5 - Se da la máxima expresión del heurístico en las páginas evaluadas	4 - Se da una expresión alta del heurístico en las páginas evaluadas
¿Se ha evitado el uso de animaciones ciclicas?	5 - Se da la máxima expresión del heurístico en las páginas evaluadas	5 - Se da la máxima expresión del heurístico en las páginas evaluadas
¿El uso de imágenes o animaciones proporciona algún tipo de valor añadido?	5 - Se da la máxima expresión del heurístico en las páginas evaluadas	3 - Se da una expresión media del heurístico en las páginas evaluadas

Figura 3.10: Reporte de evaluación.
Fuente: Elaboración propia

Finalmente se cumplió el objetivo de *implementar un módulo de reportes generales de la aplicación, el cual ayudará al análisis y observar el uso la herramienta para evidenciar su comportamiento*; este fue el último desarrollo de la aplicación puesto que para su funcionamiento y cumplimiento era necesario tener el insumo de los objetivos anteriores, allí se pueden observar gráficos estadísticos que son generados bajo cuatro parámetros, número de evaluaciones por tipo, número de heurísticas en la aplicación, evaluaciones filtradas por tiempo y top 10 de evaluadores con más participación en evaluaciones. Todos los reportes anteriormente nombrados, permiten su exportación a formato png y xls. A continuación se evidencia la implementación de este objetivo en la figura 3.11.

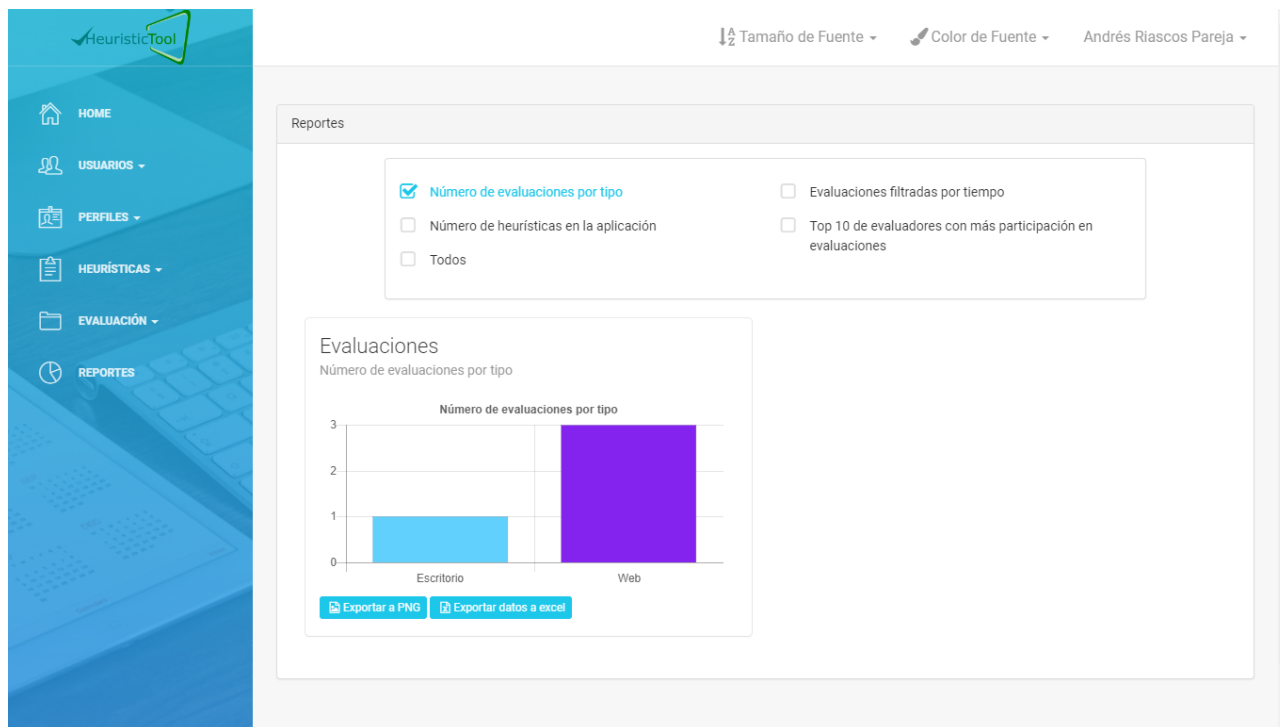


Figura 3.11: Reportes de la aplicación.

Fuente: Elaboración propia

De la misma manera, para evidenciar el cumplimiento del quinto objetivo de este trabajo de grado, *realizar pruebas a la aplicación web desarrollada para la evaluación heurística de sistemas interactivos y comparar los resultados con evaluaciones heurísticas manuales, basados en los mismos criterios ambas evaluaciones*. Para esto se usaron el Sistema de Control Docente y la página web de la Universidad del Valle Sede Tuluá como objetos (sistemas interactivos) a ser evaluados; las evaluaciones fueron desarrolladas bajo dos modelos basados en los mismo criterios, cantidad de evaluadores involucrados e iguales principios heurísticos, los modelos empleados fueron: (I) Clásico o Manual, donde la evaluación heurística hace uso de los principios heurísticos impresos en papel, y (II) Propuesto, consta del apoyo en la aplicación web HeuristicTool para la ejecución de la misma evaluación. En lo que respecta a la comparación de resultados y análisis de los mismos, se establecieron indicadores de desempeño y opiniones de los evaluadores para los modelos descritos anteriormente, esto se evidencia en las tablas 3.2 y 3.3 donde a su vez se exponen los resultados y observaciones dadas por cada evaluador en cada una de las evaluaciones.

Los evaluadores que hicieron parte de estas evaluaciones fueron:

- Carlos Andrés Riascos Pareja – Estudiante de ingeniería de sistemas X semestre de la Universidad

del Valle sede Tuluá.

- Royer David Estrada Esponda – Profesor tiempo completo de la universidad del valle sede Tuluá.
- Duvan Steven Naranjo Martínez – Estudiante de ingeniería de sistemas VIII semestre de la Universidad del Valle sede Tuluá.

Por otro lado, se eligió el principio heurístico llamado Iniciativa UsabAIPO (I) [7], debido a la naturaleza de los sistemas interactivos a evaluar, dado a que el principio heurístico fue desarrollado con la finalidad de evaluar sitios web latinoamericanos y por ende se estableció que eran los apropiados para realizar las evaluaciones heurísticas.

Indicadores	Carlos Andrés Riascos Pareja	Royer David Estrada Esponda	Duvan Steven Naranjo Martínez
<i>Tiempo de evaluación</i>	14'	10'	15'
<i>Oportunidad de consulta</i>	La oportunidad para consultar ya sean heurísticas o evaluaciones ya realizadas, se hace de manera fácil dado a que toda la información se encuentra en un solo lugar.	La consulta en medios físicos evidentemente es más compleja y menos flexible. Esto se traduce en una desventaja comparada con una solución.	La consulta es difícil debido a que no se brinda una forma sencilla porque se encuentran los datos agrupados sino que debemos dirigirnos a un navegador a consultar diferente información.
<i>Realización de informes</i>	Para la realización de informes, se requiere de utilizar tecnologías complementarias que faciliten el desarrollo colaborativo del mismo, lo cual indica luego que después se debe agrupar la información.	La realización de informes de forma manual se traduce en una tarea operativa que no permite concentrar a los evaluadores en temas especiales de la evaluación, su aplicación, resultados y evaluación.	Es un proceso complejo debido a que puede generar pérdida de tiempo porque los evaluadores deben intercambiar opiniones para realizar la conclusión.
<i>Facilidad para la difusión de la información</i>	Difundir la información requiere de trabajo operativo como el uso de correos electrónicos o imágenes para compartir los datos de los evaluadores.	En la dinámica manual es necesario concentrar a los usuarios en mismo sitio o por lo menos es necesario más que un escenario donde se utilice una solución web que complementada con otras tecnologías puede facilitar el análisis y conclusión correspondientes.	La difusión de la información en este caso, se puede realizar de dos maneras, (I) reunión de forma física o (II) utilizar diferentes tecnologías para el intercambio de información.
<i>Presencia y comunicación de los evaluadores</i>	Debido a que la comunicación de los evaluadores es importante para generar discusión acerca de los resultados obtenidos, en algunos momentos se torna difícil debido a los tiempos disponibles y ubicación geográfica de los mismos.	La solución web otorga cierta impronta de sofisticación. Sin embargo se reconoce que la interacción social presencial es inamovible de manera definitiva en proceso o fases de evaluación de evaluaciones heurísticas.	Aunque la interacción entre evaluadores es importante, en este caso se torna difícil ya que se requiere que los evaluadores se encuentren en un mismo lugar o utilizar alguna herramienta de apoyo como videoconferencias.

Tabla 3.2: Resultados de evaluación manual – Sistema control docente.

Fuente: Elaboración propia

Indicadores	Carlos Andrés Riascos Pareja	Royer David Estrada Esponda	Duvan Steven Naranjo Martínez
<i>Tiempo de evaluación</i>	15'	12'	18'
<i>Oportunidad de consulta</i>	La consulta de la información brinda a los usuarios cierto grado de ahorro en el tiempo, debido a que la información se puede encontrar en un solo lugar y en algunos casos sin la necesidad de recurrir a búsquedas en otras fuentes de información.	La aplicación facilita la consulta en la medida que ofrece ventanas para ello. Además solo por contar con una herramienta de dicha naturaleza se posibilita consulta en cualquier momento siempre y cuando se tenga una conexión a internet.	La consulta se realiza de manera sencilla, dado a que la aplicación ofrece una serie de heurísticas ya registradas disponibles para utilización de los usuarios.
<i>Realización de informes</i>	Aunque se debe mejorar en este aspecto la aplicación, cabe resaltar que permite el ahorro del tiempo dado a que la aplicación lo genera de manera automática después de que todos los evaluadores hayan registrado sus respuestas.	Los informes vienen preconfigurados en la aplicación lo cual facilita el análisis inmediato de los resultados. Sin embargo se destaca que añadir más reportes supone un desarrollo adicional en el proyecto web.	El informe se genera de manera automática después de que los evaluadores hayan realizado la evaluación, facilitando el análisis ya que no se debe preocupar por la difusión de los mismos con los demás evaluadores.
<i>Facilidad para la difusión de la información</i>	La difusión de la información lo realiza la aplicación mediante la creación de reportes donde se evidencia las respuestas de cada evaluador.	Los mecanismos incluidos en la aplicación, como realizar comentarios en línea y de manera concurrente facilitan la discusión incluso por medio de aplicaciones como HeuristicTool.	Todos los evaluadores una vez realizada la evaluación visualizaron los resultados de los demás, facilitando así el análisis y la comparación de las respuestas.
<i>Presencia y comunicación de los evaluadores</i>	Aunque en algunos casos se requiere de comunicación entre los evaluadores, la aplicación brinda los elementos básicos para realizar una evaluación sin presencia física de los mismos.	La herramienta no se traduce en una barrera para la comunicación entre los evaluadores. Por el contrario facilitaría el desarrollo de escenarios en el cual los evaluadores se ubiquen en distintos sitios geográficos.	La aplicación ofrece funcionalidades prácticas que reemplazan el uso de otras tecnologías, como por ejemplo la conclusión colaborativa. Lo anterior se traduce a que básicamente no es necesario la comunicación ni presencia de los evaluadores.

Tabla 3.3: Resultados evaluación HeuristicTool – Univalle - Tuluá

Fuente: Elaboración propia

Con respecto a los resultados obtenidos en las anteriores tablas, es posible concluir que el uso de una herramienta como apoyo al proceso de evaluación no reemplaza el factor humano en el desarrollo de la misma, por el contrario busca realizar ciertas tareas operativas y la difusión de la información para que los evaluadores puedan realizar el análisis de manera fácil y acertada. Por otro lado, la disponibilidad de la información es una ventaja dado a que se podrá acceder a ella desde cualquier lugar con acceso a internet y descargar el reporte de la evaluación cuantas veces lo requiera.

Como evidencia de la ejecución de las evaluaciones heurísticas descritas anteriormente, se incluyen los soportes de las mismas en los Anexos D y E.

Por último, la aplicación se encuentra disponible en el dominio: <https://heuristictool.com/> para el servicio de la comunidad que desee hacer uso de ella.

3.5. Pruebas

Este capítulo muestra el proceso de pruebas realizado para la aplicación HeuristicTool, las cuales se centran en la técnica de comprobación a nivel de módulos individuales para verificar el correcto funcionamiento por separado de los mismos donde se empleó el framework PHPUnit y las pruebas de aceptación, estos mecanismos estuvieron dirigidos a comprobar la integración de componentes, implementación correcta de requerimientos y corrección oportuna de defectos encontrados.

3.5.1. Desarrollo de pruebas unitarias

El framework Laravel ofrece el directorio test para albergar las pruebas de la aplicación HeuristicTool, basándose en la librería para testint PHPUnit, siendo este el más extendido en el desarrollo de aplicaciones PHP. Las pruebas unitarias se realizaron bajo las características que deben cumplir estos test para que se puedan considerar test unitarios: automatizables, completas, repetibles, independientes y profesionales. Estas se realizaron para la comprobación de las funcionalidades principales como enrutamiento, envío de datos, manejo de formularios y consultas en la base de datos, La ejecución de las pruebas con PHPUnit inicia con la verificación de datos retornados, para ello se utiliza el método visit, como se expone en la figura 3.12, la función de este método es probar que un controlador realice el proceso de manera correcta al momento de direccionar hacia alguna vista.

```
//test para ruta de Login
public function testRutaLogin(){
    $this->visit('/')
    ->see('auth/login');
}
```

Figura 3.12: PHPUnit – Método visit.

Fuente: Elaboración propia

Finalizadas las pruebas que verifiquen el retorno de las vistas, se continuó con la comprobación de los diferentes métodos de respuesta como Get o Post, el objetivo de esta prueba es demostrar que estos métodos se encuentren retornando la información pedida por la ruta, de igual manera se busca evidenciar que no exista pérdida de datos en el transcurso de la información. En la figura 3.13 se aprecia la implementación de esta prueba.

```
public function testRutaUsuarioIndex(){  
    $user = User::find(3); // Como se debe estar autenticado  
    $this->actingAs($user) //para poder acceder a esta ruta  
        ->call('GET', '/usuario');  
    $this->assertResponseOk();  
}
```

Figura 3.13: PHPUnit – Método call.

Fuente: Elaboración propia

Después de realizadas las pruebas para el retorno correcto de la información de las diferentes rutas, se realiza otro tipo de prueba para verificar que los datos enviados desde los diferentes formularios de la aplicación, lleguen perfectamente al controlador para su posterior manejo. En la figura 3.14 se evidencia la aplicación de esta prueba.

```
public function testEnvioDatos(){  
    $datos = array(  
        'evaluacion' => 6,  
        'users' => 3,  
        'aceptoEvaluar' => false,  
        'confirmado' => false  
    );  
    $this->call('POST', '/evaluadores/store', $datos);  
}
```

Figura 3.14: PHPUnit – Método call Envío de datos.

Fuente: Elaboración propia

Finalmente se realiza el último tipo de prueba que consiste en probar la conexión a la base de datos, la comunicación con ella y el efectivo intercambio de información entre la aplicación y la misma. En la figura 3.15 se observa la aplicación de esta prueba.

```
public function testDatabase() { // Hace el llamado desde la aplicación...  
    $this->seeInDatabase('users', ['email' => 'carlos.a.riascos@correounivalle.edu.co']);  
}
```

Figura 3.15: PHPUnit – Base de datos.

Fuente: Elaboración propia

Después de creadas las pruebas unitarias, se llevan a cabo por medio de la consola para conectar al entorno de PHPUnit y verificar los diferentes tipos de pruebas descritas anteriormente y su correcta ejecución. En la figura 3.16 se evidencia los resultados de la primera ejecución de las pruebas.

```
C:\xampp\htdocs\EvaluacionHeuristica\tests\ExampleTest.php:119
C:\xampp\php\pear\PHPUnit\TextUI\Command.php:176
C:\xampp\php\pear\PHPUnit\TextUI\Command.php:129

14) ExampleTest::testDatabase
PDOException: SQLSTATE[HY000] [2002] No se puede establecer una
ha conexi n.

C:\xampp\htdocs\EvaluacionHeuristica\vendor\laravel\framework\s
C:\xampp\htdocs\EvaluacionHeuristica\vendor\laravel\framework\s
C:\xampp\htdocs\EvaluacionHeuristica\vendor\laravel\framework\s
C:\xampp\htdocs\EvaluacionHeuristica\vendor\laravel\framework\s
C:\xampp\htdocs\EvaluacionHeuristica\vendor\laravel\framework\s
C:\xampp\htdocs\EvaluacionHeuristica\vendor\laravel\framework\s
C:\xampp\htdocs\EvaluacionHeuristica\vendor\laravel\framework\s
C:\xampp\htdocs\EvaluacionHeuristica\vendor\laravel\framework\s
C:\xampp\htdocs\EvaluacionHeuristica\vendor\laravel\framework\s
C:\xampp\htdocs\EvaluacionHeuristica\vendor\laravel\framework\s
C:\xampp\htdocs\EvaluacionHeuristica\vendor\laravel\framework\s
C:\xampp\htdocs\EvaluacionHeuristica\vendor\laravel\framework\s
se.php:21
C:\xampp\htdocs\EvaluacionHeuristica\tests\ExampleTest.php:133
C:\xampp\php\pear\PHPUnit\TextUI\Command.php:176
C:\xampp\php\pear\PHPUnit\TextUI\Command.php:129

FAILURES!
Tests: 17, Assertions: 3, Errors: 14.

C:\xampp\htdocs\EvaluacionHeuristica>
```

Figura 3.16: PHPUnit – Ejecuci n de prueba 1.

Fuente: Elaboraci n propia

En la figura anterior se puede observar la obtenci n de errores, por lo que fue necesario revisar los diferentes test que presentaron las dificultades para realizar las correcciones pertinentes al c digo fuente, finalizando el proceso de retroalimentaci n y correcci n se ejecuta nuevamente el entorno de PHPUnit obteniendo 0 errores como lo demuestra la figura 3.17.

```
Microsoft Windows [Versión 10.0.15063]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Andres>cd ..

C:\Users>cd..

C:\>cd xampp

C:\xampp>cd htdocs

C:\xampp\htdocs>cd EvaluacionHeuristica

C:\xampp\htdocs\EvaluacionHeuristica>phpunit
PHPUnit 4.8.36 by Sebastian Bergmann and contributors.

.....

Time: 6.47 seconds, Memory: 19.75MB

OK (17 tests, 17 assertions)
```

Figura 3.17: PHPUnit – Ejecución de prueba 2.

Fuente: Elaboración propia

3.5.2. Desarrollo de pruebas de aceptación

Las pruebas de aceptación se realizan con respecto a las necesidades del usuario, requerimientos y procesos de negocio, también determinan si un sistema satisface los criterios de un cliente tanto en la facilidad en el uso y en su correcto funcionamiento. En este caso por cada historia de usuario se realiza una prueba de aceptación, revisada y posteriormente aceptada. La tabla 3.4 muestra la prueba de aceptación del módulo Evaluación para la creación de una evaluación, las demás especificaciones se encuentran en el apartado de anexos F.

PRUEBA DE ACEPTACIÓN			
No. Caso de Prueba: 1		No. Historia de Usuario: 1	
Nombre caso de prueba: Creación de una Heurística			
Descripción: Se verificará la creación de heurísticas con su respectivo almacenamiento de datos en la aplicación y posterior uso de la misma.			
Condiciones de ejecución: • Usuario con permiso de crear heurísticas.			
Resultado esperado: La información de la heurística se almacenará correctamente en la aplicación.			
Estado de la prueba:			
Aprobada:		X	No aprobada:
Aprobado por:		Royer David Estrada Esponda	
Fecha de elaboración: 31/03/2017			

Tabla 3.4: Pruebas de aceptación, módulo Evaluación.

Fuente: Elaboración propia

Capítulo 4

Conclusiones y Trabajos Futuros

4.1. Conclusiones

Después de realizado el proceso de ingeniería del presente proyecto y con base en todo el trabajo desarrollado, se concluye:

1. Para la construcción de un sistema de información que sirva de apoyo a la gestión de diferentes procesos, en este caso particular una evaluación heurística, es fundamental la comprensión de conceptos propios en la ejecución de la misma y el entendimiento pleno del funcionamiento de cada paso, ya que esto facilita en la etapa de desarrollo la visión y asimilación de los objetivos que se busca alcanzar con el desarrollo de una herramienta tecnológica.
2. Se logró evidenciar que el uso de una herramienta de apoyo para el proceso de evaluación heurística, facilita la selección de los principios heurísticos bajo los cuales será evaluado un sistema interactivo determinado, y contribuye al acceso de la información con la que se encuentran trabajando los evaluadores de forma rápida y sencilla.
3. El uso de reportes de las diferentes evaluaciones aplicadas a los sistemas interactivos, permite condensar la información relevante del proceso, lo que a su vez facilita la interpretación de los resultados y contribuye con la creación de conocimiento en relación con la usabilidad del sistema interactivo evaluado.
4. El empleo de las evaluaciones heurísticas posee un gran delimitante en nuestra región, puesto que sólo puede ser realizada por personas con especialización en la disciplina de la usabilidad, así mismo el uso de una herramienta de apoyo al proceso de evaluación podría aumentar el interés de diferentes personas hacia la interacción humano computador, debido a la facilidad de acceder a la información y las ventajas que genera un sistema de información en el manejo de los datos.

4.2. Trabajos Futuros

En el desarrollo de la aplicación se identificaron líneas de trabajo que extienden o complementan HeuristicTool. Como trabajos futuros se consideran:

- Generar un mecanismo que permita compartir las diferentes evaluaciones realizadas en la aplicación, esto con el objetivo de compartir conocimiento y facilitar la adaptación de una ontología que brinde apoyo a los evaluadores en el proceso de la evaluación heurística.
- Crear un módulo que permita realizar otro tipo de evaluación de usabilidad, ya sea por indagación con el uso de cuestionarios o por test con la medida de las prestaciones.
- Implementar un módulo que permita la gestión de mensajes entre evaluadores, para facilitar la comunicación entre ellos y mejorar el proceso de análisis de la evaluación.
- Permitir a los evaluadores la modificación o eliminación de principios heurísticos una vez se haya puesto en marcha la evaluación heurística.

Bibliografía

- [1] UIT, (2015). La UIT publica datos mundiales anuales sobre las TIC y clasificaciones de los países según el Índice de Desarrollo de las TIC. [En línea] Disponible en: http://www.itu.int/net/pressoffice/press_releases/2015/57-es.aspx#.V6s7-ZjhDIX
- [2] R. S. Pressman, “Conceptos de Calidad,” in *Ingeniería del Software. Un enfoque práctico*, 7th ed. Ciudad de México, México: McGRAW-HILL, 2010, pp.339-345.
- [3] N. Bevan, J. Kirakowski and J. Maissel. What is Usability?. Proceedings of the 4th International Conference on HCI, Stuttgart, September 1991.
- [4] M. E. Alva, “Metodología de Medición y Evaluación de la Usabilidad de Sitios Web Educativos”, Tesis Doctoral, Dep. De Informática, Universidad de Oviedo, Oviedo, 2005.
- [5] A. M. Lopez, C. Cabrera, L. E. Valencia, “INTRODUCCIÓN A LA CALIDAD DEL SOFTWARE”, *DIALNET*, Vol. 1, Núm. 39, 2008, [En Línea]. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=4745899>
- [6] G. X. Ferré, “Marco de la integración de la usabilidad en el proceso de desarrollo de software”, Tesis Doctoral, Dep. De Lenguajes y Sistemas Informáticos e Ingeniería del Software, Universidad Politécnica de Madrid, Madrid, 2005.
- [7] M. C. Suarez, “SIRIUS: Sistema de Evaluación de la Usabilidad Web Orientado al Usuario y basado en la Determinación de Tareas Críticas”, Tesis Doctoral, Dpto. de Informática, Universidad de Oviedo, Oviedo, 2011.
- [8] M. P. González, A. Pascual y J. Lorés. (2001). Evaluación Heurística. *AIPO: Asociación persona ordenador*. [En línea] Disponible en: <http://w.aipo.es/libro/pdf/15-Evaluacion-Heuristica.pdf>
- [9] Larman, C. (2003). UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. 2nd ed. Madrid: Prentice-Hall, p.561.
- [10] L. M. Ardévol, “Análisis de viabilidad de soluciones para la Automatización de la evaluación heurística”, Tesis M.S., Universitat de Lleida, Lleida, España, 2010.
- [11] E. Kemp and D. Setungamudalige, “A resource support toolkit (R-IDE): supporting the DECIDE framework”, in *'CHINZ '06: Proceedings of the 7th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction*, New York, NY, USA , 2006.

- [12] O. Francisco, M.A. Cathalifaud, “Introducción a los Conceptos Básicos de la Teoría General de Sistemas”, *Cinta de Moebio*, 1998, [En Línea]. Disponible en: <http://www.redalyc.org/pdf/101/10100306.pdf>
- [13] A. M. Puentes, H. F. Castro, “Elementos onto-conceptuales presentes en la identidad del ingeniero de sistemas”, *ACOFI*, Vol. 11, Núm. 22, 2016, [En Línea]. Disponible en: <http://www.educacioneningenieria.org/index.php/edi/article/view/689/303>
- [14] C. Gastelum, A. C. Diaz, *Introducción a los Sistemas de Información*, Instituto Tecnológico de Sonora, [En Línea]. Disponible en: http://biblioteca.itson.mx/oa/dip_ago/introduccion_sistemas/p3.html
- [15] T. Granollers, J. Lorés and J. J. Cañas, “La Accesibilidad de los Sistemas Interactivos,” in *Diseño de sistemas interactivos centrados en el usuario*, 1ra ed. Catalunya, España: Editorial UOC, 2011.
- [16] Lorés, J., Granollers, T. and Lana, S. (2002). *Introducción a la Interacción Persona-Ordenador*. Universitat de Lleida.
- [17] Montero, H. and Fernández, Y. (2005). *La Experiencia del Usuario. No Solo Usabilidad*. [En Línea] Disponible en: http://www.nosolousabilidad.com/articulos/experiencia_del_usuario.htm
- [18] Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. 7th ed. México: McGRAW-HILL, pp.339-345.
- [19] Qué es un Framework. [En Línea] Disponible en: <http://www.bdigital.unal.edu.co/10588/1/71095012013.pdf>.
- [20] Comparativo del interés de framework a lo largo del tiempo. [En Línea] Disponible en: <https://www.google.es/trends/exploreq>
- [21] Oliver. HEURTEL. *Php y mysql*. Barcelona, Ed. Eri, 2014, pp. 255-272.
- [22] Lopez, F., Béjar R., Latre M., Nogueras J. and Zarazaga F. (2015). *GitHub como herramienta docente. XXI Jornadas de la Enseñanza*. [En Línea] Disponible en: <https://upcommons.upc.edu/bitstream/handle/2117/76761/JENUI2015.76-83.pdf?sequence=1&isAllowed=y>
- [23] GitHub, (2017). Seamless code review. [Online] Available at: <https://github.com/features>
- [24] M. Mader and B. Frei. (2016, Jun). Full Comparison: Agile vs Scrum vs Waterfall vs Kanban. [Online]. Available At: <https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>
- [25] Gamboa J. (2014). Aumento de la productividad en la gestión de proyectos, utilizando una metodología ágil aplicada en una fábrica de software en la ciudad de Guayaquil. *Revista Tecnológica ESPOL*, Vol. 27, Núm. 2 [En Línea] Disponible en: <http://learningobjects2006.espol.edu.ec/index.php/tecnologica/article/view/312/216>
- [26] Adquisición de template. [En Línea] Disponible en: <http://startbootstrap.com/template-overviews/sb-admin/>

Apéndice A

Historias de Usuario

En este Anexo se relaciona parte de las Historias de Usuario (HU) que fueron necesarias para el desarrollo de este proyecto.

Historia de Usuario	
Número: 7	Autor: Carlos Andrés Riascos Pareja
Nombre de Historia: Consultar una evaluación	
Módulo: Evaluación	Iteración: 2
Prioridad: Media	Dependencia: 6
Responsable: Carlos Andrés Riascos Pareja	
Descripción: El usuario creador de las evaluaciones o los evaluadores participantes de la misma, podrán consultar la evaluación y dependiendo del estado de la misma podrán realizar diferentes acciones.	
Validación: Solo el usuario creador de la evaluación podrá asignar evaluadores y heurísticas, los evaluadores podrán recomendar heurísticas, resolver la evaluación, dar sus respectivos hallazgos o hacer una conclusión colaborativa acerca de la evaluación con los demás evaluadores, por otro lado, también podrán ver los resultados de las evaluaciones y descargarlos en formato PDF.	
Creada: 15/08/2017	Modificada: 01/10/2017

Tabla A.1: HU, Consultar una Evaluación.

Fuente: Elaboración propia

Historia de Usuario	
Número: 8	Autor: Carlos Andrés Riascos Pareja
Nombre de Historia: Seleccionar evaluadores de una evaluación	
Módulo: Evaluación	Iteración: 2
Prioridad: Media	Dependencia: 7
Responsable: Carlos Andrés Riascos Pareja	
Descripción: El usuario creador de las evaluación podrá seleccionar los evaluadores que harán parte de la evaluación, se enviará una notificación a los evaluadores de su invitación a evaluar donde el podrá aceptar o rechazar la evaluación.	
Validación: Los evaluadores podrán ser usuarios registrados en la aplicación, o se podrán invitar ingresando su correo electrónico, al momento de registrarse e iniciar sesión observaran la invitación a la evaluación.	
Creada: 18/08/2017	Modificada: 08/10/2017

Tabla A.2: HU, Seleccionar Evaluadores.

Fuente: Elaboración propia

Historia de Usuario	
Número: 11	Autor: Carlos Andrés Riascos Pareja
Nombre de Historia: Resolver evaluación	
Módulo: Evaluación	Iteración: 2
Prioridad: Media	Dependencia: 7
Responsable: Carlos Andrés Riascos Pareja	
Descripción: Los evaluadores seleccionados en una evaluación podrán resolver la misma, basado en las heurísticas seleccionadas previamente.	
Validación: Un evaluador podrá seleccionar la respuesta de una heurística que anteriormente se seleccionó para la evaluación.	
Creada: 29/09/2017	Modificada: 12/10/2017

Tabla A.3: HU, Resolver Evaluación.

Fuente: Elaboración propia

Historia de Usuario	
Número: 13	Autor: Carlos Andrés Riascos Pareja
Nombre de Historia: Conclusión de la evaluación	
Módulo: Evaluación	Iteración: 2
Prioridad: Media	Dependencia: 7
Responsable: Carlos Andrés Riascos Pareja	
Descripción: Todos los evaluadores seleccionados en una evaluación podrán realizar una conclusión general de la misma, donde todos pueden entrar a editar al mismo tiempo y ver que escribe los demás evaluadores.	
Validación: Creación de una conclusión colaborativa por todos los evaluadores.	
Creada: 05/09/2017	Modificada: 29/09/2017

Tabla A.4: HU, Conclusión de la Evaluación.

Fuente: Elaboración propia

Historia de Usuario	
Número: 14	Autor: Carlos Andrés Riascos Pareja
Nombre de Historia: Reporte de la evaluación	
Módulo: Evaluación	Iteración: 2
Prioridad: Media	Dependencia: 7
Responsable: Carlos Andrés Riascos Pareja	
Descripción: Todos los evaluadores seleccionados y el usuario creador de la evaluación podrán visualizar un reporte el cual contiene las respuestas de todos los evaluadores a las heurísticas seleccionadas, sus hallazgos y la conclusión. Como también permitir exportar a formato PDF.	
Validación: Descargar el reporte de la evaluación en formato PDF.	
Creada: 08/09/2017	Modificada: 05/10/2017

Tabla A.5: HU, Reporte de la Evaluación.

Fuente: Elaboración propia

Apéndice B

Casos de Uso

En este Anexo se relaciona parte de los Casos de Uso (CU) que fueron necesarias para el desarrollo de este proyecto.

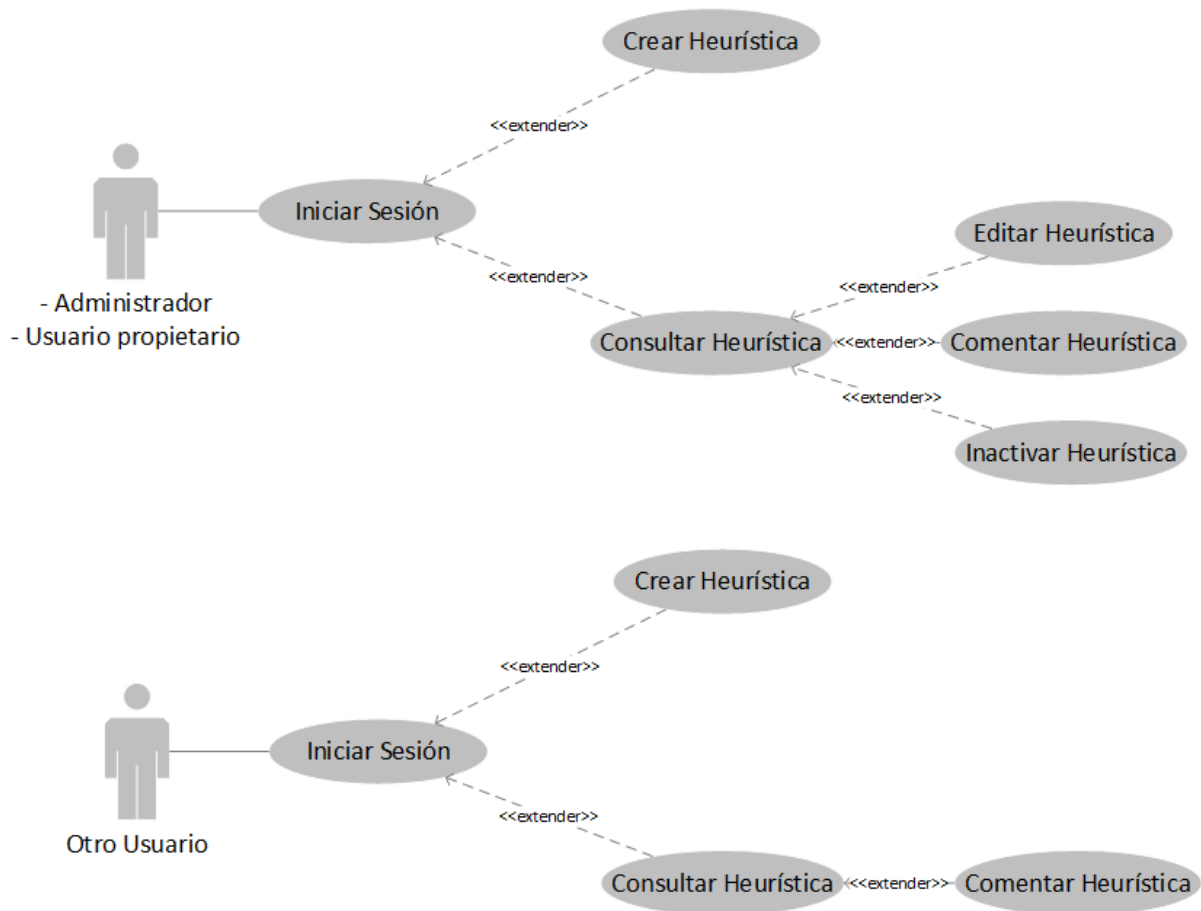


Figura B.1: CU - Módulo Heurísticas.

Fuente: Elaboración propia

Apéndice C

Diagramas Entidad - Relación

En este Anexo se relaciona parte de los diagramas Entidad - Relación (ER) que fueron necesarias para el desarrollo de este proyecto.

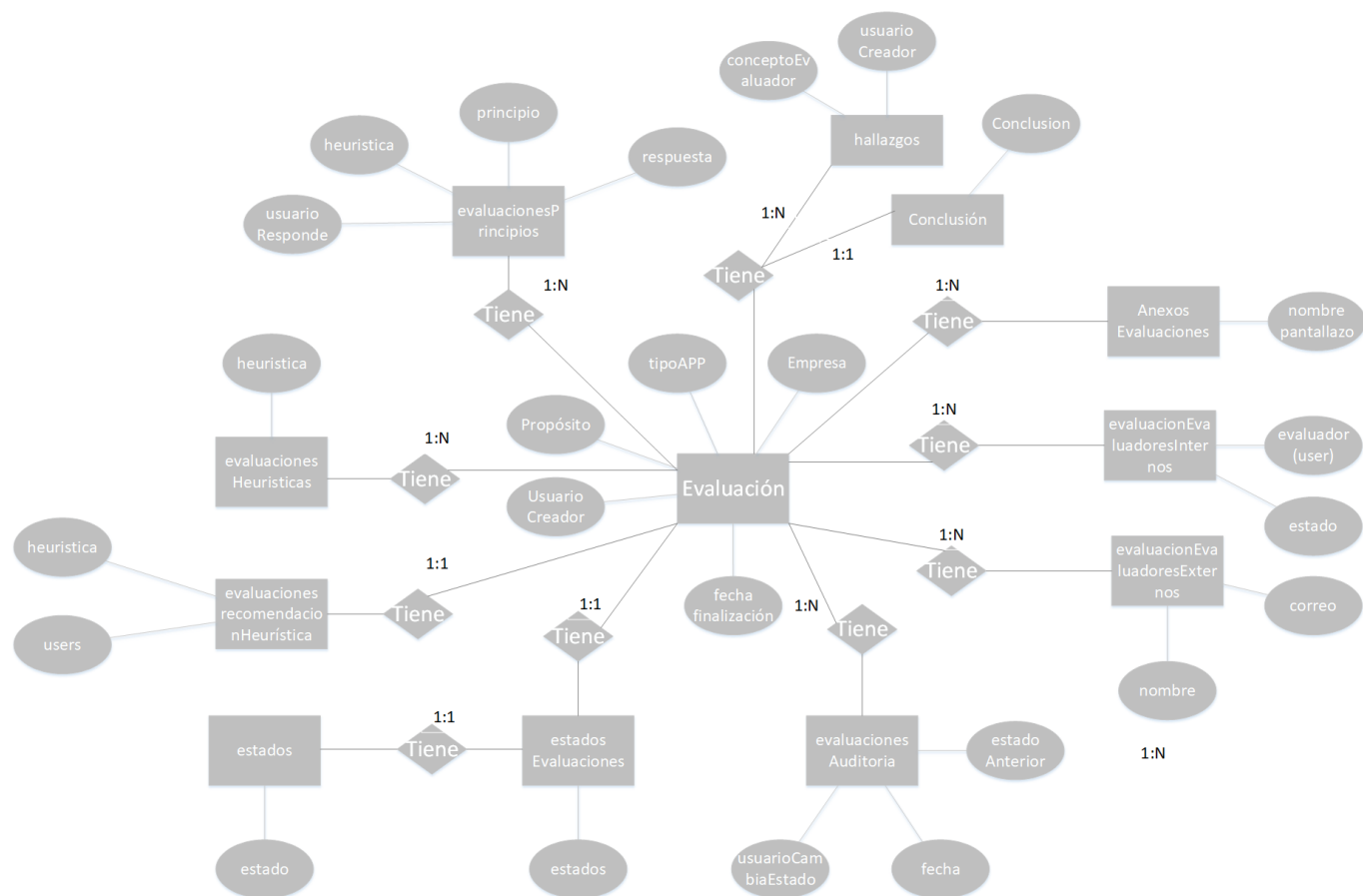


Figura C.1: ER - Módulo Evaluación.
Fuente: Elaboración propia

Apéndice D

Resultados evaluación heurística Sistema control docente - Manual

El documento se encuentra adjunto, llamado ResultadoEHManual-SistemaControlDocente.pdf

Apéndice E

Resultados evaluación heurística página web Univalle Tuluá - HeuristicTool

El documento se encuentra adjunto, llamado ResultadoEHHeuristicTool-UnivalleSedeTuluá.pdf

Apéndice F

Pruebas de Aceptación

En este Anexo se relaciona parte de las Pruebas de Aceptación (PA) que fueron necesarias para el desarrollo de este proyecto.

PRUEBA DE ACEPTACIÓN			
No. Caso de Prueba: 7		No. Historia de Usuario: 7	
Nombre caso de prueba: Consultar una evaluación.			
Descripción: Se verificará la consulta de las diferentes evaluaciones registradas en la aplicación. El usuario solo podrá acceder si es el creador de la evaluación o es evaluador de la misma.			
Condiciones de ejecución: • Usuario con permiso de consultar evaluaciones.			
Resultado esperado: Se mostrará la evaluación con sus respectivos datos y ofrecerá diferentes acciones dependiendo el estado de la evaluación.			
Estado de la prueba:			
Aprobada:		X	No aprobada:
Aprobado por:		Royer David Estrada Esponda	
Fecha de elaboración: 05/09/2017			

Tabla F.1: PA, Consultar una Evaluación.

Fuente: Elaboración propia

PRUEBA DE ACEPTACIÓN			
No. Caso de Prueba: 8		No. Historia de Usuario: 8	
Nombre caso de prueba: Selección de evaluadores para una evaluación.			
Descripción: El usuario creador de las evaluación podrá seleccionar los evaluadores que harán parte de la evaluación, se enviará una notificación a los evaluadores de su invitación a evaluar donde el podrá aceptar o rechazar la evaluación.			
Condiciones de ejecución: • Usuario creador de la evaluación.			
Resultado esperado: Se mostrará a los evaluadores invitados para participar en la evaluación.			
Estado de la prueba:			
Aprobada:		X	No aprobada:
Aprobado por:		Royer David Estrada Esponda	
Fecha de elaboración: 16/09/2017			

Tabla F.2: PA, Seleccionar Evaluadores.

Fuente: Elaboración propia

PRUEBA DE ACEPTACIÓN			
No. Caso de Prueba: 11		No. Historia de Usuario: 11	
Nombre caso de prueba: Resolver una evaluación.			
Descripción: Los evaluadores que hacen parte de la evaluación podrán resolver las diferentes heurísticas seleccionadas para la evaluación.			
Condiciones de ejecución: • Usuarios que hacen parte de la evaluación. • Heurísticas y evaluadores previamente asignados.			
Resultado esperado: Heurísticas resueltas por parte del evaluador.			
Estado de la prueba:			
Aprobada:		X	No aprobada:
Aprobado por:		Royer David Estrada Esponda	
Fecha de elaboración: 18/09/2017			

Tabla F.3: PA, Resolver Evaluación.

Fuente: Elaboración propia

PRUEBA DE ACEPTACIÓN			
No. Caso de Prueba: 13		No. Historia de Usuario: 13	
Nombre caso de prueba: Conclusión de la evaluación.			
Descripción: Los evaluadores que hacen parte de la evaluación podrán ingresar a realizar la conclusión general de la evaluación de una manera colaborativa.			
Condiciones de ejecución: • Usuarios que hacen parte de la evaluación. • Heurísticas respuestas y hallazgos registrados por todos los evaluadores.			
Resultado esperado: Se mostrará a los evaluadores un campo de texto donde podrán ingresar de manera conjunta su conclusión donde todos los evaluadores podrán editar al tiempo.			
Estado de la prueba:			
Aprobada:		X	No aprobada:
Aprobado por:		Royer David Estrada Esponda	
Fecha de elaboración: 20/09/2017			

Tabla F.4: PA, Conclusión de la Evaluación.

Fuente: Elaboración propia

PRUEBA DE ACEPTACIÓN			
No. Caso de Prueba: 14		No. Historia de Usuario: 14	
Nombre caso de prueba: Reporte de la evaluación.			
Descripción: Los evaluadores que hacen parte de la evaluación o el creador de la misma podrán visualizar las respuestas, hallazgos y conclusión de la evaluación, como también podrán descargarlo en formato PDF.			
Condiciones de ejecución: • Usuarios que hacen parte de la evaluación. • Heurísticas respuestas y hallazgos registrados por todos los evaluadores.			
Resultado esperado: Se mostrará las respuestas, hallazgos y conclusión de los evaluadores para facilitar el análisis de la evaluación.			
Estado de la prueba:			
Aprobada:		X	No aprobada:
Aprobado por:		Royer David Estrada Esponda	
Fecha de elaboración: 25/09/2017			

Tabla F.5: PA, Reporte de la Evaluación.

Fuente: Elaboración propia

Apéndice G

Manuales de Usuario

Los documentos se encuentran adjuntos, llamados:

1. ManualUsuario_HeuristicTool - Evaluación.pfd
2. ManualUsuario_HeuristicTool - Heurísticas.pdf
3. ManualUsuario_HeuristicTool - Reportes.pdf