

PA1710-7-RACUQuemes

Rediseño arquitectónico de la capa de presentación de la plataforma Quemes para incorporar atributos de calidad que mejoren la usabilidad

Genderson Barinas Guio

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
2017

PA1710-7-RACUQuemes

Rediseño arquitectónico de la capa de presentación de la plataforma Quemes para incorporar atributos de calidad que mejoren la usabilidad

Autor:

Genderson Barinas Guio

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO
DE LOS REQUISITOS PARA OPTAR AL TÍTULO DE
MAGÍSTER EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Director

Ing. Cesar Julio Bustacara Medina MSc.

Comité de Evaluación del Trabajo de Grado

Ing. María Mercedes Corral Strassman MSc.

Ing. Enrique González Guerrero PhD.

Página web del Trabajo de Grado

<http://pegasus.javeriana.edu.co/~PA1710-7-RACUQuemes>

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERIA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
Mayo, 2017

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

Rector Magnífico

Jorge Humberto Peláez, S.J.

Decano Facultad de Ingeniería

Ingeniero Jorge Luis Sánchez Téllez

Directora Maestría en Ingeniería de Sistemas y Computación

Ingeniera Angela Cristina Carrillo Ramos

Director Departamento de Ingeniería de Sistemas

Ingeniero Efraín Ortíz Pabón

Artículo 23 de la Resolución No. 1 de Junio de 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”

AGRADECIMIENTOS

Presento los más sinceros agradecimientos primeramente a Dios por siempre estar en cada momento de mi vida y darme la guía necesaria para culminar este trabajo. A mi esposa y familia, que con su apoyo hizo posible culminar la meta trazada. Al Ingeniero Cesar Julio Bustacara, quien guio mi trabajo de grado de la Maestría en Ingeniería de Software y Computación, y quien con sus conocimientos y experiencia influyó ampliamente en el éxito del presente proyecto y los excelentes resultados obtenidos.

A los profesionales e integrantes de la Comunidad Académica de la Universidad Javeriana, aquellos que infundieron en mí el fortalecimiento de saberes y competencias, y a todos aquellos que de una u otra manera han participado en este gran logro.

A todos, mil gracias.

Contenido

1.	INTRODUCCIÓN.....	13
2.	DESCRIPCIÓN GENERAL	14
2.1.	OPORTUNIDAD Y PROBLEMÁTICA.....	14
3.	DESCRIPCIÓN DEL PROYECTO.....	16
3.1.	OBJETIVO GENERAL	16
3.2.	OBJETIVOS ESPECÍFICOS.....	16
3.3.	METODOLOGÍA	16
3.4.	FASES DE DESARROLLO.....	17
3.4.1.	<i>Fase 1 - Diagnóstico de la plataforma Quemes.....</i>	<i>17</i>
3.4.2.	<i>Fase 2 - Formulación de la arquitectura de software</i>	<i>18</i>
3.4.3.	<i>Fase 3 - Construcción del prototipo</i>	<i>19</i>
4.	MARCO TEÓRICO / ESTADO DEL ARTE.....	21
4.1.	HEURÍSTICAS DE USABILIDAD	21
4.2.	INTERACCIÓN HUMANO COMPUTADOR.....	22
4.3.	ARQUITECTURA DE SOFTWARE	24
4.3.1.	<i>Estilos arquitectónicos.....</i>	<i>24</i>
4.3.2.	<i>Atributos de calidad.....</i>	<i>25</i>
4.3.3.	<i>Modelos de calidad.....</i>	<i>25</i>
4.4.	USABILIDAD.....	28
5.	TRABAJOS RELACIONADOS	30
6.	FASE I - DIAGNÓSTICO DE LA PLATAFORMA QUEMES	32
6.1.	EVALUACIÓN HEURÍSTICA.....	32
6.2.	EVALUACIÓN DE PRINCIPIOS DE DISEÑO ORIENTADOS A NIÑOS	37
6.3.	EVALUACIÓN DE USABILIDAD SIRIUS.....	39
7.	FASE 2 - FORMULACIÓN DE LA ARQUITECTURA	40
7.1.	DISEÑO DE LA ARQUITECTURA DE SOFTWARE	40
7.2.	EVALUACIÓN DE LA ARQUITECTURA DE SOFTWARE.....	42
7.3.	DISEÑO Y EVALUACIÓN DE LA INTERFAZ DE USUARIO.....	48

8.	FASE 3 - CONSTRUCCIÓN DEL PROTOTIPO	51
8.1.	PRESENTACIÓN DEL PROTOTIPO	51
9.	RESULTADOS DE LA VALIDACIÓN.....	56
9.1.	PERCEPCIÓN DE USABILIDAD A TRAVÉS DEL INSTRUMENTO SUS.....	57
9.2.	PERCEPCIÓN DE USABILIDAD A TRAVÉS DEL INSTRUMENTO PSSUQ.....	60
9.3.	VALIDACIÓN DEL PROTOTIPO APLICANDO LAS HEURÍSTICAS DE USABILIDAD....	62
9.4.	VALIDACIÓN DEL PROTOTIPO APLICANDO LOS PRINCIPIOS DE DISEÑO ORIENTADOS A NIÑOS	63
10.	IMPACTOS	64
11.	CONCLUSIONES.....	65
12.	TRABAJOS FUTUROS	67
13.	REFERENCIAS	68

Tabla de figuras

Figura 1. Representación general de la metodología	17
Figura 2. Esquema para la formulación de la arquitectura de software	19
Figura 3. Esquema formulación arquitectura de software	20
Figura 4. Interacción humano computador [23].....	23
Figura 5. Resultado evaluación heurística de usabilidad	36
Figura 6. Resultado evaluación principios de diseño.....	38
Figura 7. Vista lógica de la arquitectura	40
Figura 8. Vista de implementación de la arquitectura	41
Figura 9. Árbol de utilidad.....	43
Figura 10. Ingreso al sistema	52
Figura 11. Ayuda visual en campos	52
Figura 12. Ayuda visual campo obligatorio.....	53
Figura 13. Ayuda contextual.....	53
Figura 14. Estado normal de un botón	53
Figura 15. Representación visual de un botón cuando es presionado.....	54
Figura 16. Espacio de trabajo prototipo	55
Figura 17. Escala SUS [58].....	58
Figura 18. Resultado evaluación PSSUQ	61
Figura 19. Resultado validación heurística	62
Figura 20. Validación principios de diseño	63

Tabla de tablas

Tabla 1. Factores de calidad de McCall	25
Tabla 2. Atributos de calidad modelo FURBS	26
Tabla 3. Atributos de calidad modelo ISO 9126	27
Tabla 4. Definiciones de usabilidad	29
Tabla 5. Trabajos relacionados.....	30
Tabla 6. Principios heurísticos de usabilidad	33
Tabla 7. Características principios de usabilidad	34
Tabla 8. Resultado validación heurística.....	36
Tabla 9. Principios de diseño orientados a niños	37
Tabla 10. Resultado evaluación principios de diseño	38
Tabla 11. Evaluación de la arquitectura	44
Tabla 12. Resultado evaluador uno	48
Tabla 13. Resultado evaluador dos.....	48
Tabla 14. Resultado evaluación con el cliente	49
Tabla 15. Resultado evaluación principios de diseño	50
Tabla 16. Tecnologías empleadas	51
Tabla 17. Presentación del prototipo.....	51
Tabla 18. Elementos del espacio de trabajo del prototipo.....	54
Tabla 19. Escala alfa de cronbach.....	56
Tabla 20. Características demográficas de los usuarios de prueba.....	57
Tabla 21. Resultado evaluación SUS	58
Tabla 22. Resultado de las respuestas de la encuesta SUS.....	59
Tabla 23. Resultado cuatro dimensiones PSSUQ.....	60
Tabla 24. Resultado de las respuestas de la encuesta PSSUQ.....	62

ABSTRACT

In the present work, we address the problems identified in the Quemes platform in the software component associated to the system, that is not focused on the end user, we identify functional problems of usability and an opportunity to include new functionalities. To solve these problems and improve the user experience in its interaction with the system, a set of heuristic principles oriented to usability, guidelines and good practices for the design of user interfaces were implemented; As well as defined guidelines and principles in the field of human computer interaction, and the software quality attributes defined in ISO 9126.

RESUMEN

En el presente trabajo se aborda la problemática identificada en la plataforma Quemes en el componente de software asociado a que el sistema no está enfocado al usuario final, se identifican problemas funcionales de usabilidad y una oportunidad de incluir nuevas funcionalidades. Con el propósito de solucionar esta problemática y de mejorar la experiencia del usuario en su interacción con el sistema, se implementaron un conjunto de principios heurísticos orientados a la usabilidad, lineamientos y buenas prácticas para el diseño de interfaces de usuario; así como lineamientos y principios definidos en el campo de la interacción de humano computador, y los atributos de calidad de software definidos en la norma ISO 9126.

RESUMEN EJECUTIVO

El presente trabajo se elaboró en el marco del proyecto de investigación Quemes, realizado por los grupos SIDRE y TAKINA de la Pontificia Universidad Javeriana. En sus inicios Quemes se planteó como una propuesta de programación de robots que trabajaran de forma cooperativa y se aplicara a juegos tales como el fútbol [1]. Gracias a esta iniciativa se logró la participación en campeonatos mundiales de fútbol, y de dicha experiencia se recopiló una serie de oportunidades de mejora desde el punto de vista del diseño y de la programación cooperativa. En una segunda etapa del proyecto contando con el apoyo de Maloka y la Universidad de los Andes se proyectó a Quemes como una herramienta pedagógica para la sensibilización, apropiación y generación de conocimiento de estudiantes a través de la solución de problemas cotidianos. De esta forma se logra introducir a Quemes en el ámbito de colegios colombianos dirigido a estudiantes entre 9 y 13 años [2].

Dado que el enfoque inicial de Quemes no era pedagógico, no se realizó un diseño arquitectónico en la capa de presentación enfocado a niños entre los 9 y 13 años, lo cual limitó la extensibilidad del sistema a otras formas de interacción, por ejemplo, web, dispositivos móviles o cliente/servidor. Asimismo, no se realizó documentación arquitectónica, no siendo posible reutilizar sus componentes en la capa de presentación. En el diseño de la interfaz de usuario, el sistema no permitía generar mensajes de ayuda o guías que les permita a los usuarios recordar cómo realizar determinadas tareas. Las opciones de navegación no se plantearon consistentes, dado que en el sistema no se establecieron técnicas de interacción estándar y su diseño a nivel de presentación no era el deseable para el público objetivo [3]. Además, no se consideró al usuario y su entorno en el proceso de diseño. El lenguaje y las expresiones usadas no fueron orientados al lenguaje del usuario. Adicionalmente, cuando se presentan errores no se ofrece una alternativa para que el usuario continúe realizando la tarea o para que pueda recuperar lo último que estaba realizado.

La problemática descrita anteriormente se abordó a través de los siguientes aspectos: i) el diseño actual de interfaz de la plataforma Quemes no está orientado al usuario final, ii) existen problemas funcionales en la plataforma actual que no facilitan su operatividad por parte del usuario, iii) se identifica una oportunidad para incluir nuevas funcionalidades que mejoren la interacción del usuario con el sistema.

Por tales motivos y para que su comercialización tenga una mejor adopción, era importante para la Pontificia Universidad Javeriana, Maloka, profesores de tecnología de los centros pedagógicos donde se ha instalado la plataforma y para la misma iniciativa Quemes, proveer una arquitectura de software en la capa de presentación a la plataforma Quemes que resolviera los problemas de usabilidad hacia los estudiantes, y que mejorara la experiencia y satisfacción del usuario.

En consecuencia, para dar solución a la problemática descrita previamente, se propuso realizar un diseño arquitectónico orientado a la usabilidad, que mejorara la experiencia del usuario a través de su interacción con el sistema Quemes. Para lograrlo, se planteó la adopción de tecnologías que apoyaran la consecución de los requerimientos identificados, una metodología de diseño centrada en el usuario, la consideración de los atributos de calidad definidos en la norma ISO 9126, para realizar un diseño arquitectónico que cumpliera los requerimientos funcionales y no funcionales, la adopción de técnicas y principios dados por el campo de la interacción

humano computador (HCI, Human Computer Interaction), y finalmente la adopción de principios heurísticos orientados a la usabilidad y lineamientos de diseño de interfaz centrados en el usuario final.

Para el logro de los objetivos definidos en este proyecto, se propuso una metodología que se abordó en tres fases.

En la primera fase se realizó un diagnóstico de la capa de presentación de la plataforma Quemes. Esta fase, incluye el uso de la metodología de diseño centrada en el usuario, dicha metodología se siguió a lo largo de todo el proyecto. En esta fase, se realizó una validación de usabilidad de la plataforma actual de Quemes, a través de tres instrumentos, nombrados a continuación: Validación heurística de usabilidad, validación de principios de diseño orientados a niños entre los 9 y 13 años y validación de usabilidad a través del método SIRIUS (Sistema de Evaluación de la Usabilidad Web Orientado al Usuario y basado en la Determinación de Tareas Críticas). Los resultados obtenidos, permitieron identificar los requerimientos funcionales, no funcionales y las necesidades a nivel de diseño que se debían atender para mejorar la usabilidad del sistema. Así mismo, permitió generar una evaluación cuantitativa y cualitativa de la usabilidad del sistema Quemes.

A partir de los resultados obtenidos, en la segunda fase de este proyecto, se realizó el diseño arquitectónico. Para su validación, se adoptó el método Architecture Trade-Off Analysis Method (ATAM) considerando las etapas que se ajustaban a este escenario. Así mismo, se realizaron los mockups correspondientes a la propuesta de diseño orientada al público objetivo. Para la validación de la propuesta de diseño se contó con el apoyo de evaluadores expertos en diseño gráfico, se evaluó contrastando la propuesta contra los principios de diseño identificados en la primera fase y se realizaron validaciones recurrentes con el cliente, de esta forma se logró mantener alineado los objetivos de negocio con respecto al planteamiento realizado considerando las buenas prácticas de usabilidad identificadas en la primera fase del proyecto.

Con los resultados obtenidos en la primera y segunda fase, se realizó una fase final en la que se implementó un prototipo funcional aplicando los lineamientos, principios, técnicas y los diseños propuestos. Durante el proceso de implementación, se realizaron validaciones con el cliente para recibir retroalimentación y poder hacer ajustes al modelo propuesto. Se realizó una validación final con el cliente y con el usuario final. La validación con el usuario final, se realizó a través de dos instrumentos; el primero, la Escala de Usabilidad del Sistema (SUS, System Usability Scale) y el segundo, el Cuestionario de Usabilidad del Sistema Post Estudio (PSSUQ, Post-Study System Usability Questionnaire).

A partir de los resultados obtenidos de los instrumentos de validación, se obtuvo una excelente percepción de usabilidad del usuario en su interacción con el sistema. Con el instrumento SUS se obtuvo un valor de 89.38 que representa que el grado de aceptación de la plataforma por parte del usuario es altamente favorable. De igual forma, el resultado del instrumento PSSUQ para la dimensión de satisfacción general 6.94 indico un excelente nivel de satisfacción.

Por tanto, este trabajo tuvo como resultado la generación de un prototipo funcional, que considera la investigación realizada, desde las heurísticas de usabilidad identificadas, las buenas prácticas de diseño orientadas a niños, el planteamiento de una arquitectura orientada a la usabilidad y el diseño de la interfaz orientada al usuario, mejorando la percepción de usabilidad y la interacción con el usuario.

1. INTRODUCCIÓN

La usabilidad se plantea como uno de los múltiples principios a considerar para hacer de la interacción del usuario con el sistema una experiencia. Dicha experiencia tiene una relevancia muy significativa para el usuario ya que de ello dependerá si su percepción es positiva o negativa hacia el producto, lo que se traduce en sí el usuario volverá o no a usar el sistema y en si disfruta haciéndolo. La usabilidad es solo uno de los peldaños que se deben considerar para crear una experiencia para el usuario.

La usabilidad de un sistema de software es uno de los factores decisivos para que un usuario permanezca interesado en hacer uso de un determinado producto. De acuerdo a Shackel[4] la usabilidad es la capacidad en términos funcionales humanos para ser usado fácil y eficazmente por el rango especificado de usuarios, dada la capacitación especificada y el soporte al usuario, para cumplir con el rango de tareas especificadas dentro del rango especificado de escenarios. Una segunda definición que ayuda a entender el término de usabilidad es dada por la norma ISO 9241-11 la cual define como el grado en que un producto puede ser usado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso. Adicionalmente, la usabilidad también implica optimizar las interacciones que las personas tienen con el producto de tal forma que pueden llevar a cabo sus actividades en la vida cotidiana.

El aporte de este trabajo está orientado a mejorar la experiencia del usuario en su interacción con la plataforma Quemes mediante una nueva propuesta arquitectónica en la capa de presentación y la incorporación de principios de diseño de interfaz orientados al usuario final, principios heurísticos de usabilidad, tecnologías que faciliten la interacción del usuario con el sistema y una serie de técnicas, principios y buenas prácticas definidas a través de la interacción humano computador que mejoren la usabilidad del sistema.

Este documento se encuentra distribuido en diez secciones de la siguiente manera: 1. Descripción general, se describe la problemática y motivación para realizar este trabajo de grado; 2. Descripción del proyecto, donde se plasman los objetivos asociados a la problemática y la metodología que se planteó para alcanzar los objetivos propuestos; 3. Marco teórico, corresponde a la base teórica para el desarrollo del trabajo; 4. Trabajos relacionados, trabajos asociados en alguno de los ámbitos evaluados y que aportan a este trabajo desde el enfoque de la usabilidad; 5. Fase uno de la metodología propuesta, se aborda el diagnóstico y definición de requerimientos; 6. Fase dos de la metodología se formula una solución tecnológica a través de un diseño arquitectónico y un diseño de interfaz de usuario; 7. fase tres de la metodología, se realiza la construcción de un prototipo funcional y se registra en este trabajo el resultado de la validación con el cliente y con el usuario final de dicho prototipo; 8. Resultados de la validación, se describe los resultados obtenidos en la validación de la percepción de usabilidad del usuario en su interacción con el sistema; 9. Conclusiones; 10. Trabajos futuros. En las dos últimas secciones se registran conclusiones obtenidas como resultado de la realización de este trabajo y se listan los trabajos futuros propuestos que darían continuidad al trabajo aquí realizado.

2. DESCRIPCIÓN GENERAL

En esta sección se brinda una contextualización de las motivaciones que llevaron a realizar este trabajo. Se evidencia la situación actual del componente de software de la plataforma Quemes y aquellos puntos susceptibles de cambio o mejora para aumentar la satisfacción de uso por parte del usuario.

2.1. Oportunidad y problemática

La creación de la plataforma Quemes se inició en el año 2008 como una iniciativa de los grupos de investigación SIDRE y TAKINA de la Pontificia Universidad Javeriana, cuyo planteamiento inicial se enfocó en la programación de robots que trabajaran de forma cooperativa en juegos como el fútbol. Su enfoque principal fue desarrollar una plataforma robusta que brindara una solución a la cooperación, coordinación, comunicación, y colaboración entre sistemas multiagentes. Para la implementación del control de los robots se utilizó la plataforma BESA [5], la cual, fue desarrollada en el Departamento de Ingeniería de Sistemas de la Pontificia Universidad Javeriana para facilitar la implementación y ejecución de sistemas multiagentes.

A través de un proyecto de investigación desarrollado en conjunto entre Maloka, la Universidad de los Andes y la Pontificia Universidad Javeriana se buscó dar a Quemes un nuevo enfoque como una herramienta pedagógica para la sensibilización, apropiación y generación de conocimiento a través del trabajo colaborativo, mediante la programación de robots para dar solución a problemáticas cotidianas y de esta manera promover el aprendizaje colaborativo, el pensamiento creativo, sistemático, tecnológico, y el trabajo en equipo¹ en estudiantes de colegios entre los 9 a 13 años [1],[2].

En esta transición de llevar a Quemes a una plataforma con enfoque pedagógico no se realizó un diseño arquitectónico en la capa de presentación, lo cual ha limitado la extensibilidad del sistema a otras formas de interacción, por ejemplo, web, dispositivos móviles o cliente/servidor. No se tiene documentación arquitectónica, así como no es posible reutilizar sus componentes en la capa de presentación. En el diseño de la interfaz de usuario el sistema no cuenta con mensajes de ayuda para su uso o guías que les permita a los usuarios recordar cómo realizar determinadas tareas, las opciones de navegación son inconsistentes, no cuenta con técnicas de interacción estándar, su diseño a nivel de presentación es inconsistente para el público objetivo [3]. No se consideró al usuario y su entorno en el proceso de diseño, asimismo el lenguaje y las expresiones usadas no son orientadas al lenguaje del usuario, además, cuando se presentan errores no se ofrece al usuario una alternativa para que continúe realizando la tarea o para que pueda recuperar lo último que ha realizado.

La problemática descrita anteriormente se aborda en este trabajo desde tres enfoques: i) el diseño actual de interfaz de la plataforma Quemes no está orientado al usuario final, ii) existen problemas funcionales en la plataforma actual que no facilitan su operatividad por parte del

¹ P. U. Javeriana, "Educación en tecnología soportada en robots colaborativos." <http://www.javeriana.edu.co/investigacion/quemes>. [Última consulta: 15-Feb-2016].

usuario, iii) una oportunidad para incluir nuevas funcionalidades que mejoren la interacción del usuario con el sistema.

Por tales motivos y para que su comercialización tenga una mejor adopción, es importante para la Pontificia Universidad Javeriana, Maloka, profesores de tecnología de los centros pedagógicos donde se ha instalado la plataforma y para la misma iniciativa Quemes proveer de una arquitectura de software en la capa de presentación a la plataforma Quemes que resuelva los problemas de usabilidad hacia los estudiantes, que mejore la experiencia y satisfacción del usuario.

3. DESCRIPCIÓN DEL PROYECTO

En esta sección se presentan los objetivos del proyecto y la metodología empleada para el cumplimiento de los objetivos.

3.1. Objetivo general

Diseñar e implementar una arquitectura de software centrada en el usuario que permita mejorar los niveles de usabilidad de QUEMES.

3.2. Objetivos específicos

1. Diagnosticar el estado actual de la capa de presentación de la plataforma QUEMES, para definir los requerimientos de interacción de usuario.
2. Diseñar y validar una arquitectura de software orientada a la usabilidad que soporte diferentes medios de interacción.
3. Implementar y validar un prototipo funcional de la arquitectura de software.

3.3. Metodología

El desarrollo de la presente propuesta se abordó en tres fases que corresponden a:

1. Fase uno. Diagnóstico de la plataforma QUEMES.
2. Fase dos. Formulación de la arquitectura de software.
3. Fase tres. Construcción del prototipo.

En la fase uno (diagnostico) se aborda el primer objetivo de esta propuesta, el cual buscaba determinar el estado actual de la plataforma Quemes, definir los requerimientos del usuario a nivel de arquitectura[6] de software cuyo principal driver es la usabilidad y definir los requerimientos de interfaz de usuario. Se hace la recolección, análisis, especificación, validación y certificación de los requerimientos a nivel del sistema. A través de la metodología de Diseño Centrado en el Usuario (DCU) [7],[8] tomando como modelo de referencia la norma ISO 9126 [9]. Las etapas relacionadas con especificar el contexto y los requerimientos del usuario se abordaron dentro de las etapas de recolección, análisis y especificación de requerimientos de la metodología DoRCU, para lo cual se realizó una adaptación en estas etapas con el fin de contemplar los lineamientos de la metodología DCU en el análisis y especificación de requerimientos. Esta adaptación se realizó con miras a ser más efectivos en la interacción con el usuario y en optimizar el tiempo de las partes involucradas.

Una vez establecidos los requerimientos que se deben satisfacer, se continua con la fase dos (formulación) en la que se aborda el objetivo dos, en esta fase se realiza el diseño de la arquitectura de software siguiendo un enfoque iterativo de descomposición del sistema a través de la metodología de diseño dirigida por atributos (ADD, Attribute Driven Design) [10],[11],[12]. Para la evaluación de arquitectura se usa el método de evaluación de arquitecturas ATAM (Architecture Tradeoff Analysis Method) [13],[14] para lo cual, se realizó una adaptación a este

método de evaluación para enfocarlo hacia la usabilidad. En esta fase se generó el diseño de interfaz enfocado en el usuario final.

En la fase tres (construcción) se realiza la implementación del diseño arquitectónico y del diseño de interfaz de usuario, estos diseños se obtienen como resultado de las fases anteriores. También, se realiza la construcción de un prototipo funcional, para su implementación se sigue el modelo incremental de desarrollo de software. Finalmente, en esta fase se realiza una validación con el cliente y una validación con el usuario final a través de dos instrumentos; el primero, la Escala de Usabilidad del Sistema (SUS, System Usability Scale) y el segundo, el Cuestionario de Usabilidad del Sistema Post Estudio (PSSUQ, Post-Study System Usability Questionnaire).

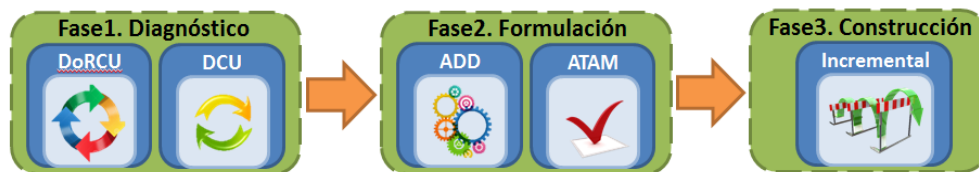


Figura 1. Representación general de la metodología

3.4. Fases de desarrollo

3.4.1. Fase 1 - Diagnóstico de la plataforma Quemes

En esta fase se realizó la recolección, análisis, especificación, validación y certificación de los requerimientos. Para ello se utilizaron tres instrumentos, una evaluación heurística de usabilidad[15], [16], una evaluación de principios de diseño orientados a niños entre 9 y 13 años y una evaluación de usabilidad a través del modelo SIRIUS. Adicionalmente, se realizó una recolección de requerimientos del sistema y de interfaz de usuario con el cliente.

En el desarrollo de esta fase se realizaron las siguientes actividades:

- Entendimiento de la plataforma Quemes
- Identificación de los principios, lineamientos y técnicas de usabilidad para niños
- Creación del instrumento de validación heurística a través de principios de usabilidad
- Creación del instrumento de validación de principios de diseño para niños
- Validación del sistema actual de Quemes a través del instrumento de validación de heurísticas
- Validación del sistema actual de Quemes a través del instrumento de validación de principios de diseño para niños
- Recolección y clasificación de requerimientos
- Validación de la claridad de los requerimientos

- Especificación de los requerimientos
- Validación y aprobación del documento de requerimientos

Al finalizar esta fase se obtuvo el documento de especificación de requerimientos. (Ver ANEXO II)

3.4.2. Fase 2 - Formulación de la arquitectura de software

Para la formulación de la arquitectura de software se siguió el esquema mostrado en la figura 2. A partir de los resultados obtenidos en la etapa de diagnóstico se especificaron los requerimientos arquitectónicos, los cuáles fueron la entrada para definir un marco de referencia arquitectónico [17] para satisfacer los principales requerimientos del sistema. En la etapa dos se identificaron los componentes del sistema, y para cada componente se identificó:

- a) cómo está conectado al marco de referencia
- b) cuáles son los servicios prestados
- c) cuáles son sus responsabilidades
- d) las dependencias que existen entre componentes

La representación de la arquitectura se realizó a través del modelo 4 + 1 propuesto por Philippe Kruchten [18].

Para la definición de la arquitectura se empleó la metodología ADD, y para la evaluación de la arquitectura se usó el método ATAM. Se realizó una adaptación de este método de evaluación para enfocarlo hacia la usabilidad.

En esta fase, se realizó el diseño de interfaz de usuario a través del método de mockups de usuario. La validación se realizó de tres maneras, la primera a través de un juicio de expertos, la segunda a través de una validación de principios de diseño para niños entre los 9 y 13 años y la tercera una validación con el cliente.

En el desarrollo de esta fase se realizaron las siguientes actividades:

- Definición de los requerimientos arquitectónicos.
- Identificación de los atributos de calidad que se van a satisfacer.
- Definición del estilo arquitectónico dominante de la arquitectura.
- Definición de los componentes de la arquitectura.
- Elaboración del diseño de la arquitectura a través de una herramienta.
- Adaptación del modelo de evaluación de la arquitectura.
- Evaluación de la arquitectura de software propuesta.
- Propuesta de interfaz de usuario
- Validación de la interfaz de usuario

Los resultados obtenidos de esta fase corresponden al documento de arquitectura de software y los mockups de la interfaz de usuario.

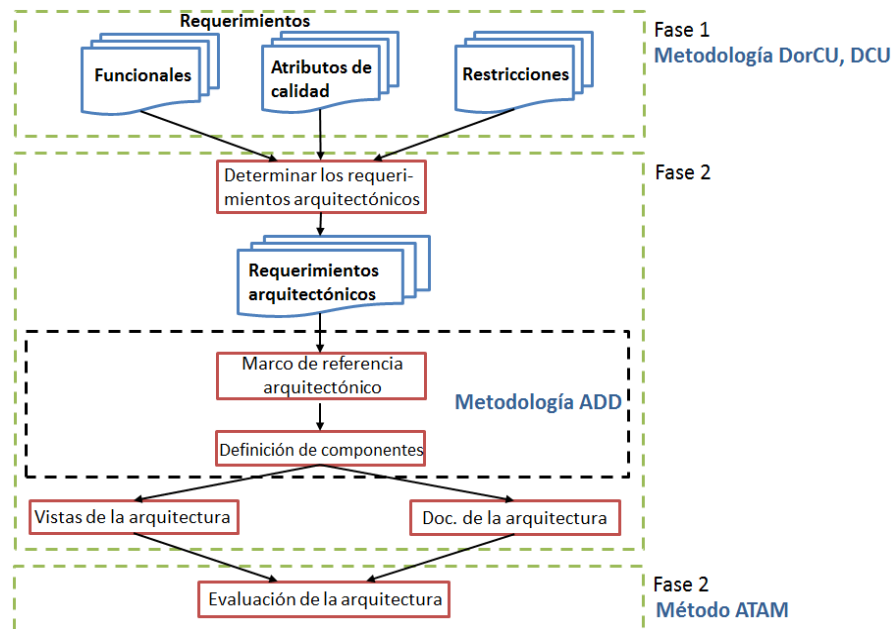


Figura 2. Esquema para la formulación de la arquitectura de software

3.4.3. Fase 3 - Construcción del prototipo

En esta fase se tuvo como entrada el diseño de la arquitectura y el diseño de la interfaz centrada en el usuario que fueron elaborados en las fases anteriores. En esta fase, la implementación se realizó a nivel de los componentes definidos en el diseño arquitectónico dando inicio por aquellos que representaban mayor valor para el usuario. Se usó el modelo incremental de desarrollo de software, en el que se abordaron las etapas de diseño, programación, y pruebas. El diagrama de la figura 3 muestra el esquema general que se aplicó en esta etapa:

En el desarrollo de esta fase se realizaron las siguientes actividades:

- Implementación de los requerimientos que aportaban mayor valor al usuario
- Pruebas con el cliente
- Refinamiento a partir de las validaciones realizadas con el cliente
- Validación del prototipo con el cliente
- Validación del prototipo con el usuario

El resultado obtenido de esta fase corresponde a una retroalimentación del cliente, la validación con el usuario a través de la Escala de Usabilidad del Sistema (SUS, System Usability Scale) y

el Cuestionario de Usabilidad del Sistema Post Estudio (PSSUQ, Post-Study System Usability Questionnaire). del modelo de aceptación de tecnología y la memoria de trabajo de grado.

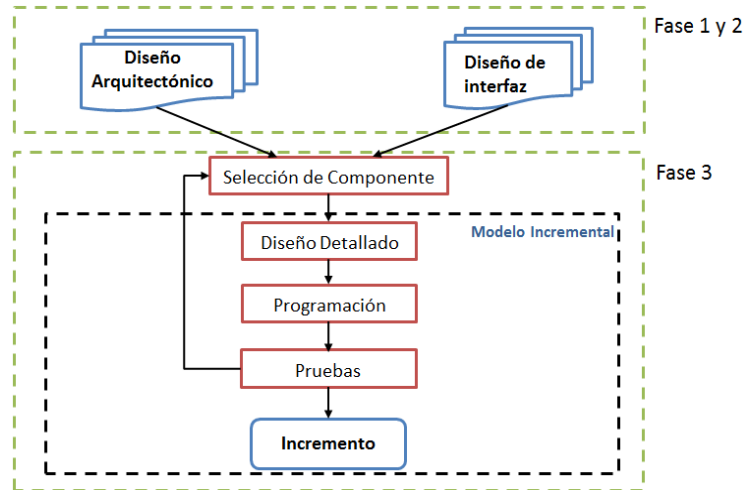


Figura 3. Esquema formulación arquitectura de software

4. MARCO TEÓRICO / ESTADO DEL ARTE

En esta sección se presentan los conceptos principales empleados para el desarrollo de este trabajo.

4.1. Heurísticas de usabilidad

A través de una heurística se definen los lineamientos que se emplearon como herramienta de evaluación [19]. Aplicada al contexto de este trabajo, se usó para evaluar la usabilidad de las interfaces.

Jacob Nielsen[20] uno de los principales ponentes en este campo propone diez principios heurísticos para evaluar la usabilidad. Estos principios son:

1. **Visibilidad del estado del sistema:** el sistema siempre debería mantener informados a los usuarios de lo que está ocurriendo, a través de retroalimentación apropiada dentro de un tiempo razonable.
2. **Relación entre el sistema y el mundo real:** el sistema debería hablar el lenguaje de los usuarios mediante palabras, frases y conceptos que sean familiares al usuario, más que con términos relacionados con el sistema. Seguir las convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico.
3. **Control y libertad del usuario:** hay ocasiones en que los usuarios elegirán las funciones del sistema por error y necesitarán una “salida de emergencia” para dejar el estado no deseado al que accedieron, sin tener que pasar por una serie de pasos.
4. **Consistencia y estándares:** los usuarios no deberían cuestionarse si acciones, situaciones o palabras diferentes significan en realidad la misma cosa; es importante seguir las convenciones establecidas.
5. **Prevención de errores:** mucho mejor que un buen diseño de mensajes de error es realizar un diseño cuidadoso que prevenga la ocurrencia de problemas.
6. **Reconocimiento antes que recuerdo:** se deben hacer visibles los objetos, acciones y opciones, el usuario no tendría que recordar la información que se le da en una parte del proceso, para seguir adelante. Las instrucciones para el uso del sistema deben estar a la vista o ser fácilmente recuperables cuando sea necesario.
7. **Flexibilidad y eficiencia de uso:** la presencia de aceleradores, que no son vistos por los usuarios novatos, puede ofrecer una interacción más rápida a los usuarios expertos que la que el sistema puede proveer a los usuarios de todo tipo.
8. **Estética y diseño minimalista:** los diálogos no deben contener información que es irrelevante o poco usada. Cada unidad extra de información en un diálogo, compite con las unidades de información relevante y disminuye su visibilidad relativa.
9. **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores:** los mensajes de error se deben entregar en un lenguaje claro y simple, indicando en forma precisa el problema y sugerir una solución constructiva al problema.
10. **Ayuda y documentación:** incluso en los casos en que el sistema pueda ser usado sin documentación, podría ser necesario ofrecer ayuda y documentación. Dicha información debería ser fácil de buscar, estar enfocada en las tareas del usuario, con una lista concreta de pasos a desarrollar y no ser demasiado extensa.

4.2. Interacción humano computador

El Grupo de Interés Especial sobre Interacción Humano – Computador (SIGCHI, Special Interest Group on Computer-Human Interaction) de la Asociación para la Maquinaria Informática (ACM, Association for Computing Machinery) define la interacción humano computador (HCI) como una disciplina relacionada con el diseño, la evaluación e implementación de sistemas informáticos interactivos para uso humano y con el estudio de los principales fenómenos que los rodean [21]. Una característica importante del HCI es que es altamente interdisciplinar y puede llegar a ser considerada como una combinación de ciencia, ingeniería, y diseño. Algunas de las áreas involucradas en HCI son: computación, psicología, sociología, antropología y el diseño industrial. A través de HCI se busca que las personas tengan una experiencia más productiva, eficiente y placentera con el uso del computador [22].

El SIGCHI ha considerado cinco aspectos de la interacción humano computador que están interrelacionados. Estos aspectos son:

- La naturaleza de la interacción humano computador (N)
- El uso y el contexto de los computadores (U)
- Características humanas (H)
- Sistema informático y arquitectura de la interfaz (C)
- Proceso de desarrollo (D)

En la figura 4 [23] se encuentran representados los aspectos enunciados previamente y su correspondiente correlación. El componente U1 representan los componentes informáticos que existen dentro de un entorno social, colaborativo y laboral, en dicho entorno existen áreas de aplicación en las cuales se pueden emplear sistemas de información (componente U2). Sin embargo, este proceso de incorporar tecnologías o componentes informáticos no suele ser un proceso sencillo, y se deben considerar aspectos humanos, técnicos y de trabajo en la adopción de estos componentes a través de estrategias de aprendizaje humano, adaptación de sistema u otras estrategias que se ajusten a cada caso particular (U3). Desde la perspectiva del ser humano es necesario considerar los mecanismos de procesamiento de información humana (H1), comunicación (H2) y las características físicas de los usuarios (H3). Desde la perspectiva del computador, han sido desarrolladas una variedad de técnicas para apoyar la interacción con humanos, estos mecanismos generalmente asociados a dispositivos de entrada y salida que conectan al humano con la máquina como interfaz de interacción (C1). De igual forma que han creado y usado técnicas para establecer un mecanismo de dialogo e interacción (C2). A través de estas técnicas se busca implementar el diseño de elementos como el uso de metáforas en la interfaz de usuario (C3). Para la interacción por medio de diálogos pueden ser usadas técnicas que involucren gráficos en el computador (C4). Finalmente, existen procesos de desarrollo que incorporan el diseño de diálogos para la interacción de humano computador (D1), así como una serie de técnicas y herramientas para incorporarlos (D2), técnicas para su evaluación (D3) y una serie de diseños para su estudio (D4). Todos estos componentes están relacionados y se articulan entre sí para generar un impacto en las áreas de aplicación definidas.

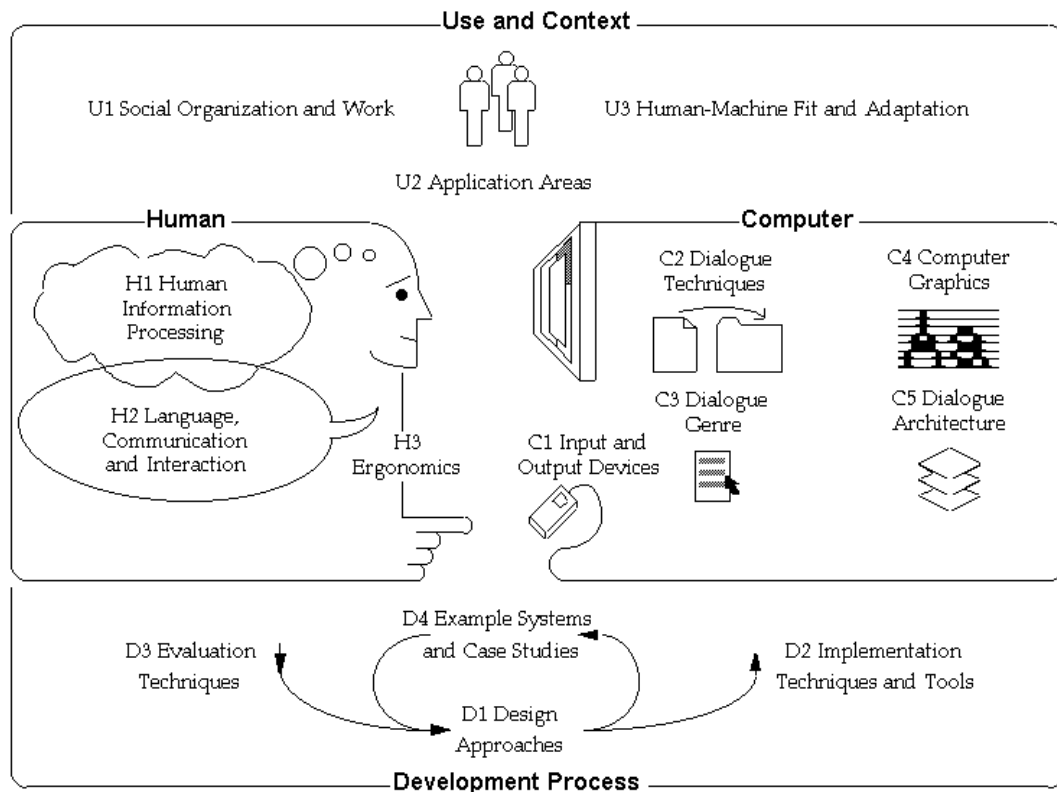


Figura 4. Interacción humano computador [23]

De acuerdo a Donald Norman, existen dos principios claves para una buena interacción humano computador. Estos principios son: visibilidad y provisión.

La visibilidad sugiere que los componentes de una interfaz deben cumplir los siguientes lineamientos:

- ser visibles, pero sin sobrecargar al usuario.
- tener una definición clara con respecto a los efectos que producen. Se considera la retroalimentación, y una relación entre los objetivos del usuario, las acciones que debe hacer y los resultados esperados.

La provisión propone que los componentes de una interfaz deben proveer de manera natural su funcionalidad.

Desde el punto del desarrollo de software el área más importante del HCI es el diseño de la interfaz de usuario. Esta área involucra el diseño centrado en el usuario (DCU), dimensiones de la variabilidad de la interfaz, herramientas y técnicas de diálogo, diseño de interfaz, implementación de interfaz, calidad de la interfaz y métodos de evaluación [21]. En el presente trabajo se consideraron los principios y técnicas orientadas al diseño de interfaz de usuario.

4.3. Arquitectura de software

Actualmente existen varias definiciones para el término de arquitectura de software y aunque cada autor ha dado su propia definición podemos identificar que no se contradicen entre ellas, sino que por el contrario se complementan. La IEEE 1471 [24] define la arquitectura como “la organización fundamental de un sistema incorporando sus componentes, sus relaciones entre ellos y con el entorno, así como los principios que guían su diseño y evolución”. Bass et al [25] definen la arquitectura de software como la estructura o estructuras del sistema, que comprenden elementos de software, las propiedades externamente visibles de esos elementos y las relaciones entre ellos. Hohmann[26] define la arquitectura de software como “la suma de los módulos, procesos y datos importantes del sistema, su estructura y las relaciones que existen entre sí”. Considerando las definiciones anteriores podemos ver la arquitectura de software en función de sus componentes, conectores y las relaciones que existen entre ellos para estructurar el sistema. Dicho sistema puede ser estructurado considerando los siguientes lineamientos:

- la organización de componentes y relaciones entre ellos
- los requerimientos que debe satisfacer el sistema y las restricciones a las que está sujeto
- las propiedades no funcionales del sistema y su impacto sobre la calidad del mismo
- las reglas y decisiones de diseño que gobiernan esta estructura
- los argumentos que justifican las decisiones tomadas

4.3.1. Estilos arquitectónicos

El estilo arquitectónico representa el patrón de más alto nivel del sistema y permite especificar la estructura fundamental de una aplicación. Bushmann et al[27] lo definen como “Los estilos arquitectónicos expresan esquemas fundamentales de organización estructural para sistemas de software. Proporcionan un conjunto de subsistemas predefinidos, especifican sus responsabilidades e incluyen reglas y pautas para organizar las relaciones entre ellos”.

Garlan y Shaw[28] definen un estilo arquitectónico como una familia de sistemas de software en términos de un patrón de organización estructural, que define un vocabulario de componentes, conectores y restricciones que se pueden combinar y aplicar a un determinado estilo.

Cada estilo arquitectónico describe una categoría de sistemas que incluye:

- Conjunto de componentes
- Conjunto de conectores
- Unas restricciones que definen la integración entre componentes
- Un modelo semántico que facilita entender las propiedades generales del sistema

4.3.2. Atributos de calidad

De acuerdo a Barbacci [29] la calidad del software se define como el grado en el cual el software posee una combinación deseada de atributos. Estos atributos son requerimientos no funcionales y hacen referencia a características que el sistema debe satisfacer. Por otro lado la norma ISO 9126 [30] ofrece la siguiente definición, la calidad es el conjunto de características de un producto o servicio que tienen la capacidad de satisfacer unas necesidades implícitas o explícitas. Esta definición de calidad, lleva a plantear el cuestionamiento de cuáles son esos atributos de calidad que el sistema debe satisfacer con el fin de diseñar una arquitectura de software que apoye la consecución de los objetivos propuestos en este trabajo.

Actualmente, se puede identificar varios atributos o cualidades que son de interés en la arquitectura de software, estos son importantes porque impactan el diseño y el desarrollo de muchas partes del software [24].

Existen varios modelos propuestos en los cuales se definen unas características o atributos de calidad que un sistema de software debe cumplir. El cumplimiento de dichos atributos puede variar entre sistemas, ya que de acuerdo a Bass et al [25] en la consecución de algún atributo de calidad se obtendrá como resultado una disminución de cumplimiento en otro atributo de calidad, es decir que dependiendo del sistema algunos atributos de calidad se cumplirán en mayor proporción que otros.

4.3.3. Modelos de calidad

A continuación, se presentan algunos de los modelos de calidad propuestos.

Modelo de McCall

En este modelo de calidad, se describen relaciones jerárquicas entre factores de calidad con base en unos criterios y métricas de calidad establecidos. Estos factores de calidad están enfocados en los siguientes aspectos: características operativas, capacidad de cambios y adaptabilidad a nuevos entornos [31]. En la tabla 1 se ilustra los factores de calidad y los criterios asociados a cada factor:

Tabla 1. Factores de calidad de McCall

Factor de calidad	Criterio
Correctitud	Rastreabilidad
	Compleitud
	Consistencia
Confiabilidad	Consistencia
	Exactitud
	Tolerancia a fallos
Eficiencia	Eficiencia de ejecución
	Eficiencia de almacenamiento

Integridad	Control de acceso
	Auditoria de acceso
Usabilidad	Operabilidad
	Entrenamiento
	Comunicación
Mantenibilidad	Simplicidad
	Concreción
Capacidad de prueba	Simplicidad
	Instrumentación
	Autodescriptivo
	Modularidad
Flexibilidad	Autodescriptivo
	Capacidad de expansión
	Generalidad
	Modularidad
Portabilidad	Autodescriptivo
	Independencia del sistema
	Independencia de máquina
Reusabilidad	Autodescriptivo
	Generalidad
	Modularidad
	Independencia del sistema
	Independencia de máquina
Interoperabilidad	Modularidad
	Similitud de comunicación
	Similitud de datos

Modelo FURBS

Es un modelo presentado por Robert Grady [32] en el cual se consideran los factores de calidad del software mostrados en la tabla 2:

Tabla 2. Atributos de calidad modelo FURBS

Factores de calidad	Atributos
Funcionalidad	Conjunto de características del sistema
	Seguridad
	Capacidades del sistema

Usabilidad	Factores humanos
	Consistencia en la interfaz de usuario
	Documentación de usuario
	Material de entrenamiento
	Factores estéticos
Confiabilidad	Frecuencia y severidad de las fallas
	Recuperabilidad
Rendimiento	Eficiencia
	Disponibilidad
	Tiempo de respuesta
	Tiempo de recuperación
	Recursos usados
Soporte	Capacidad de pruebas
	Extensibilidad
	Adaptabilidad
	Mantenibilidad
	Compatibilidad
	Configurabilidad
	Instalabilidad
	Internacionalización

Modelo ISO 9126

El modelo ISO 9126 [9] tiene sus bases en el modelo de McCall y en el de Boehm [31]. Este modelo define seis factores de calidad para el software. En tabla 3 se ilustra estos factores junto con sus subcaracterísticas:

Tabla 3. Atributos de calidad modelo ISO 9126

Característica	Subcaracterística
Funcionalidad	Adecuación
	Exactitud
	Interoperabilidad
	Seguridad
Confiabilidad	Madurez
	Tolerancia a fallos
	Recuperabilidad

Usabilidad	Entendibilidad
	Capacidad de aprendizaje
	Operabilidad
Eficiencia	Comportamiento en tiempo
	Comportamiento de recursos
Mantenibilidad	Analizabilidad
	Modificabilidad
	Estabilidad
	Capacidad de pruebas
Portabilidad	Adaptabilidad
	Instalabilidad
	Reemplazabilidad

4.4. Usabilidad

De acuerdo a Shackel[4] la usabilidad es la capacidad en términos funcionales humanos para ser usado fácil y eficazmente por el rango especificado de usuarios, dada la capacitación especificada y el soporte al usuario, para cumplir con el rango de tareas especificadas dentro del rango especificado de escenarios. Una segunda definición que ayuda a entender el término de usabilidad es dada por la norma ISO 9241-11 la cual define como el grado en que un producto puede ser usado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso. Adicionalmente, la usabilidad también implica optimizar las interacciones que las personas tienen con el producto de tal forma que pueden llevar a cabo sus actividades en la vida cotidiana. A continuación presentamos los objetivos en los cuales puede descomponerse la usabilidad [22]:

- Fácil de usar (eficacia)
- Eficiente de usar (eficiencia)
- Seguro de usar (seguridad)
- Útil (utilidad)
- Fácil de aprender (fácil de aprender)
- Fácil de recordar cómo usarlo (memorabilidad)

De forma muy similar a los atributos de calidad existen varios planteamientos en términos de atributos de usabilidad. En la tabla 4, se presenta el planteamiento de varios autores asociados al atributo de usabilidad:

Tabla 4. Definiciones de usabilidad

Definiciones de usabilidad						
	Shackel (1991)	ISO 9126 (1991)	Hix (1993)	Nielsen (1993)	Preece (1994)	Wilson (1997)
Rendimiento	Aprendizaje	Aprendizaje	Aprendizaje	Aprendizaje	Aprendizaje	Aprendizaje
	Efectividad	Operabilidad	Rendimiento a largo plazo	Eficiencia de uso	Rendimiento	Eficiencia
	Aprendizaje		Retenibilidad	Memorabilidad		Memorabilidad
	Efectividad	Operabilidad		Errores	Rendimiento	Taza de errores
Vista de usuario	Actitud	Atractividad	Satisfacción del usuario a largo plazo	Satisfacción	Actitud	Satisfacción

Las definiciones planteadas en la tabla 4 están basadas en la combinación de dos maneras diferentes de cómo debe ser medida la usabilidad. Estas definiciones son:

- Objetiva (rendimiento): La usabilidad es medida analizando como el usuario interactúa con el software.
- Subjetiva (vista de usuario): La usabilidad es medida analizando lo que el usuario piensa del producto.

A partir de la investigación realizada, se logró establecer las bases teóricas necesarias para guiar las siguientes fases de este trabajo. En esta sección se presentaron los conceptos principales empleados con el fin de detallar los elementos clave considerados en el desarrollo de este trabajo.

5. TRABAJOS RELACIONADOS

En esta sección se muestran algunos trabajos relacionados con plataformas de programación de robots enfocados a niños. Se consideraron los conceptos de interés para este trabajo de grado. En la tabla 5 se indica las fortalezas y debilidades de algunos de los trabajos relacionados que se revisaron.

Tabla 5. Trabajos relacionados

	[33]	[34]	[35]	[36]	[37]	[38]	[39]	[40]	[41]	[42]
Soporte web	+	-	-	-	-	-	-	-	-	+
Programación visual	+	+	+/-	+	+/-	-	+	+	+	+
Integración con robots	-	+	+	+	+	+	+	+	+	-
Diseño orientado al usuario	+	+/-	+/-	+	-	-	+	+	+	+
HCI	+/-	+/-	+/-	+	-	-	+	+/-	+/-	-

Se identificó una significativa tendencia en estas aplicaciones a manejar modelos de despliegue stane alone en lugar de web. Una de las posibles causas es la dependencia que se evidencia entre el hardware del computador con el hardware del robot para poder establecer la comunicación entre estos dos elementos, este planteamiento se apoya con el criterio de evaluación asociado a la integración con robots. Las plataformas que se ofrecen como web no cuentan con una interfaz de comunicación con dispositivos externos en este caso robots y, por el contrario, las plataformas que no son web si cuentan con dicho mecanismo de integración.

Pese a que las plataformas evaluadas en su mayoría cuentan con mecanismos de programación visual, ya sea por bloques, imágenes u otros, no llegan a ser lo suficientemente sencillos de entender y de utilizar por parte del usuario final considerado en este trabajo de grado. Por otro lado, se identificó la oportunidad y la necesidad de diseñar interfaces de usuario considerando principios de diseño orientados a niños y técnicas de HCI que mejoren la experiencia del usuario en su interacción con la plataforma. Siendo este uno de los puntos más importantes de aporte a través del presente trabajo.

En cuanto a plataformas de programación visual enfocadas a niños, se revisaron las siguientes propuestas:

Blockly: es una biblioteca que adiciona un editor de código visual a las aplicaciones web y Android. Este editor emplea bloques gráficos entrelazados para representar conceptos de código como variables, expresiones lógicas, bucles entre otros. Permite a los usuarios aplicar principios de programación sin tener que preocuparse por la sintaxis.

Algunas de las fortalezas de blockly son:

- Código exportable. Es posible exportar los programas creados a lenguajes de programación comunes.
- Fuente abierta. Es un producto de licencia abierta, lo cual permite extender sus funcionalidades y hacer uso libre de esta librería.
- Extensible. Permite agregar bloques personalizados y enriquecer el API, de igual forma es posible eliminar bloques predefinidos o modificar sus funcionalidades.
- Robusto. Puede llegar a implementarse tareas de programación complejas
- Internacionalización. Cuenta con soporte para más de 40 idiomas, incluyendo versiones de derecha a izquierda para árabe y hebreo.

Scratch: es proyecto desarrollado en colaboración entre Google y el Scratch Team del MIT (Instituto tecnológico de Massachussets), desarrollado sobre la tecnología Blockly de Google y con el apoyo de las habilidades del Scratch Team en el desarrollo de herramientas creativas para el aprendizaje de la gente joven. Scratch Blocks proporciona un entorno para construir bloques de programación con formatos tanto en vertical como en horizontal.

Droplet: es un editor de programación por bloques drag and drop como Scratch, que permite a los usuarios editar en los lenguajes de programación de textos convencionales. Su objetivo es cerrar la brecha de aprendizaje entre bloques y texto. Droplet realiza conversión bidireccional utilizando el texto del programa en el lenguaje de destino como modelo de datos principal.

Snap: es un lenguaje de programación gráfico basado en bloques, que permite crear animaciones interactivas, juegos, entre otros, enfocados en facilitar el aprendizaje de programación, matemáticas y conceptos de computación. Snap está inspirado en Scratch, sin embargo, no solo se centra en un público joven, sino que considera a adultos.

En la evaluación de las plataformas de programación visual se consideraron criterios como costo, tipo de licenciamiento, características funcionales, madurez, soporte y mantenimiento. A partir de estos criterios se realizó la selección de la plataforma usada para integrar en este trabajo.

Con respecto a los trabajos validados, se identificó una falta de aplicación de principios de diseño orientados al usuario final, no se consideran principios como el tamaño de los textos, colores, internacionalización, entre otros. Adicionalmente, se identificó una falencia en la adopción de principios heurísticos de usabilidad y de técnicas de HCI que enriquecieran la interacción del usuario con la plataforma. Finalmente, se identificó que la mayoría de propuestas similares a Quemes no están enfocadas a la web, esto abre una gran posibilidad de innovación al considerar en este trabajo un modelo de despliegue web.

6. FASE I - DIAGNÓSTICO DE LA PLATAFORMA QUEMES

Con el fin de identificar las limitantes y falencias a nivel de usabilidad presentes en la plataforma Quemes se emplearon tres métodos de evaluación que corresponde a, una evaluación heurística, una evaluación de principios de diseño orientados a niños y la aplicación de la evaluación de usabilidad SIRIUS. Se realizaron entrevistas con el cliente para recolectar nuevos requerimientos y finalmente, se identificaron los problemas funcionales de la plataforma actual detectados a partir de la interacción con el sistema.

6.1. Evaluación heurística

Se construyó una matriz a partir del planteamiento heurístico de varios autores para determinar los principios más aplicables al escenario. Una vez identificados los principios, se seleccionaron diez principios para ser aplicados en el modelo planteado. A partir de los principios seleccionados se realizó una evaluación de la plataforma actual para determinar su estado con respecto a dichos principios.

En la tabla 6 se ilustran los principios identificados a través de las heurísticas consideradas. Los principios seleccionados y aplicados a la evaluación fueron:

- Manejo de sesión
- Lenguaje de comunicación apropiado
- Retroalimentación
- Ayuda y documentación
- Entrenamiento – tutorial
- Interfaz de usuario coherente
- Navegación consistente
- Prevención y recuperación de errores
- Reducción de carga de memorización
- Consistencia y estándares

En la tabla 7 se ilustran las características asociadas a los principios empleados en la evaluación y su estado actual en Quemes. En la figura 5 se ilustra un diagrama de torta para reflejar el porcentaje en que los principios seleccionados están aplicados actualmente en Quemes. A través de esta evaluación se identificó que no se aplica ninguno de los conceptos seleccionados lo que corresponde a un 70% y parcialmente se cumplen tres, equivalente a un 30% lo que deja un 0% para los que se cumplen. Este hallazgo representó una motivación para incorporar estos conceptos con el fin de mejorar la usabilidad del sistema.

Tabla 6. Principios heurísticos de usabilidad

	Manejo de Sesión	Lenguaje de comunicación apropiado	Retroalimentación	Ayuda y documentación	Entrenamiento - tutorial	Interfaz de usuario coherente	Navegación consistente	Visibilidad estado sistema	Multimedia (sonidos)	Prevención y recuperación de errores	Reducción carga de memorización	Consistencia y estándares
D. Pinelle et al [43]	X	X	X	X	X							
H. Desurvire et al [19]	X		X	X	X	X	X		X			
H. Korhonen et al [44]	X	X	X		X	X	X		X	X	X	
Nielsen Norman [20]	X	X		X	X	X		X		X	X	X
A. Alsumait et al [15]		X	X	X		X	X	X		X		X
B. Schneiderman [20]		X	X							X	X	
M. Federoff [45]	X		X		X	X	X	X	X	X		
N. Wong et al [46]			X	X	X	X		X				
T. Reeves et al [47]		X	X	X		X	X	X		X		X
S. Chiasson et al [48]		X	X			X					X	
L. Kuparinen et al [49]				X		X	X			X		X
R. Miranda [50]	X	X		X		X		X		X	X	X
Propuesta	X	X	X	X	X	X	X			X	X	X

Tabla 7. Características principios de usabilidad

Concepto	Características	Estado actual en Quemes
Manejo de sesión	<ul style="list-style-type: none"> • Guardar el trabajo realizado • Recuperación del trabajo realizado • Preferencias de usuario • Gestión de sesión 	Parcialmente
Lenguaje de comunicación apropiado	<ul style="list-style-type: none"> • Presentación del contenido en forma clara y lógica • Consistencia en terminología (avisos, menús, pantallas de ayuda) • Terminología familiar para el usuario • Lenguaje 	Parcialmente
Retroalimentación	<ul style="list-style-type: none"> • Ayuda contextual en el uso del aplicativo • Proveer respuestas consistentes a las acciones del usuario • Mensajes de éxito, alerta, error 	No se cumple
Ayuda y documentación	<ul style="list-style-type: none"> • Guía fácilmente accesible para el usuario • El material de apoyo del usuario es comprensible y efectivo • Pasos concretos a seguir • Guía rápida enfocada en las tareas del usuario • Guía detallada 	No se cumple
Entrenamiento – Tutorial	<ul style="list-style-type: none"> • Entrenamiento para principiantes • Envolver al usuario a través de un tutorial interesante y absorbente 	No se cumple
Interfaz de usuario coherente	<ul style="list-style-type: none"> • Color, tipografía • Diseño de diálogos 	No se cumple

	<ul style="list-style-type: none">• Diseño eficiente y agradable, no intrusivo	
Navegación consistente	<ul style="list-style-type: none">•Consistente, lógica.•Objetos, acciones, y opciones visibles, de tal forma que el usuario no tenga que recordar información cuando se está navegando de una parte a otra.	Parcialmente
Prevención y recuperación de errores	<ul style="list-style-type: none">• Recuperación y salida a errores• Prevenir errores comunes	No se cumple
Reducción de carga de memorización	<ul style="list-style-type: none">• No debe ser necesario memorizar cosas innecesarias•Interfaces fuertemente visuales	No se cumple
Consistencia y estándares	<ul style="list-style-type: none">• Uso de estándares	No se cumple

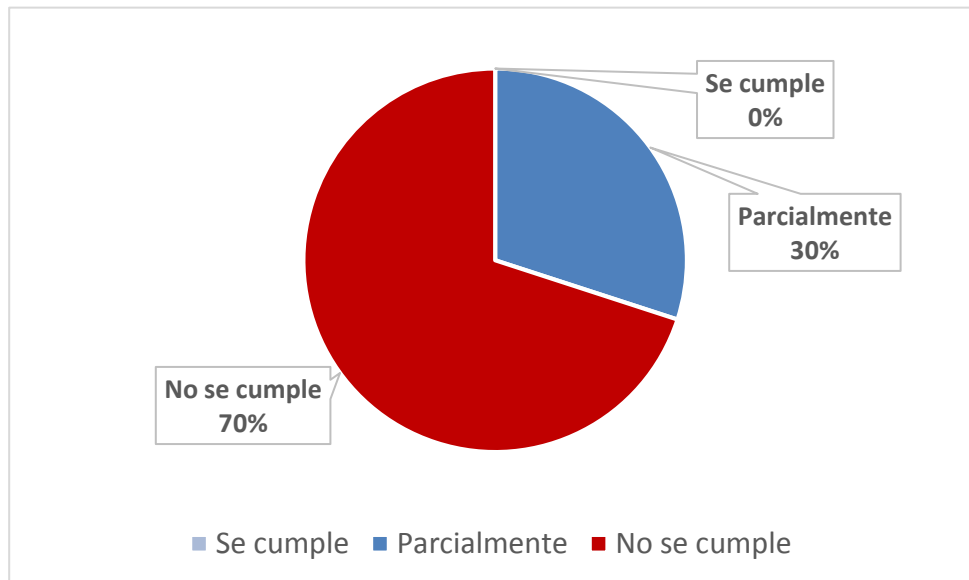


Figura 5. Resultado evaluación heurística de usabilidad

En la tabla 8 se ilustra la cantidad de principios empleados en la evaluación, adicionalmente se ilustra el porcentaje de cumplimiento para los valores de la escala definida.

Tabla 8. Resultado validación heurística

Indicador	Cantidad	Porcentaje
No se cumple	7	70%
Parcialmente	3	30%
Se cumple	0	0%
Total	10	100%

6.2. Evaluación de principios de diseño orientados a niños

Se realizó una matriz en la que se recopilieron lineamientos de diseño orientados a niños entre los 9 y 13 años que corresponden al usuario final de Quemes [51], [52]. En la tabla 9 se ilustran estos lineamientos² y el resultado de la validación en Quemes:

Tabla 9. Principios de diseño orientados a niños

Principios de diseño orientados a niños (9 - 13 años)		Estado actual
Interacción General	Evitar el uso de scroll	No se cumple
	Proveer Instrucciones explícitas	No se cumple
	Utilizar iconos y símbolos en formas familiares	Parcialmente
	Usar el puntero para recorrer la pantalla	Se cumple
Texto	Utilizar fuentes relativamente grandes (12, 14)	No se cumple
	No usar texto animado	Se cumple
	Usar texto fácilmente comprensible y conciso	No se cumple
	Proporcionar instrucciones que sean siempre accesibles	No se cumple
	Internacionalizar textos	No se cumple
	Utilizar lenguaje apropiado	Parcialmente
Navegación	Implementar un esquema de navegación consistente	No se cumple
	Categorías significativas	Parcialmente
	Denotar donde está ubicado el usuario	No se cumple
Interfaz	Hacer que los elementos que son cliqueables luzcan como cliqueables	No se cumple
	Hacer que las imágenes que son cliqueables luzcan como cliqueables	No se cumple
	Botones en lugar de textos como links	Se cumple
	Diseño enfocado en la edad	No se cumple
	Uso de colores vivos, brillos	No se cumple

² NN/g Nielsen Norman Group <https://www.nngroup.com/articles/childrens-websites-usability-issues/> [Ultima consulta: 20-Abr-2017].

Uso de iconos e imágenes	No se cumple
Uso de metáforas	No se cumple
Diseño llamativo	No se cumple
Uso estándar de interacción drag and drop	No se cumple
Crear interfaces de usuario que comprende varias ventanas	Se cumple

En la figura 6 y en la tabla 10 se ilustran los resultados obtenidos de aplicar esta evaluación de principios a la plataforma actual de Quemes. Se logró identificar que se cumplen algunos de los principios seleccionados, un 13%. Sin embargo, un 70 % de los principios empleados para la evaluación no son considerados en la plataforma y un 17% solo se considera parcialmente.

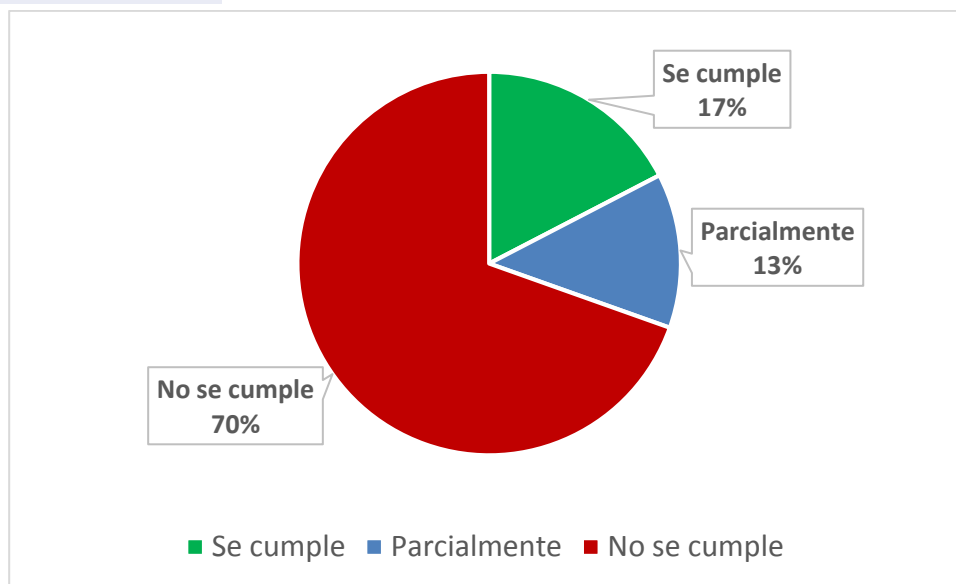


Figura 6. Resultado evaluación principios de diseño

Tabla 10. Resultado evaluación principios de diseño

Indicador	Cantidad	Porcentaje
No se cumple	16	70%
Parcialmente	3	13%
Se cumple	4	17%
Total	23	100%

6.3. Evaluación de usabilidad SIRIUS

SIRIUS es un sistema de evaluación de usabilidad orientado a la web que se fundamenta en la evaluación heurística. A través del uso del instrumento SIRIUS se obtiene una medida cuantitativa de la usabilidad del sistema evaluado [53].

Se consideraron los criterios que aplicaban al contexto de este trabajo. El resultado de usabilidad obtenido a través de este método fue del 46.26 %, este valor representa un bajo nivel de usabilidad del sistema, inferior al valor promedio. Se consideraron 42 criterios de los 83 actualmente disponibles. En el ANEXO I se encuentra la evaluación realizada a través de este instrumento.

El cálculo del porcentaje de usabilidad obedece a la siguiente formulación [53] :

$$PU = \frac{\sum_{i=1}^{i=nce} (fci * vci)}{\sum_{i=1}^{i=nce} (fci * 10)} * 100$$

Dónde:

nce: Número de criterios evaluados.

vc: Valor de evaluación de un criterio (entre 0 y 10).

fc: Factor de corrección aplicado al criterio evaluado.

Donde el factor de corrección está dado por:

$$fci = \frac{rci}{\sum_{j=1}^{j=nce} rcj}$$

Dónde:

rc: Valor de relevancia que corresponde a un criterio

En esta sección se presentó el resultado del diagnóstico de la plataforma Quemes. A partir de los tres instrumentos empleados para la evaluación, se logró identificar principios heurísticos orientados a la usabilidad y su estado actual en la plataforma Quemes. De igual forma, se identificaron principios de diseño orientados a niños y su estado actual en la plataforma Quemes. Los resultados de estas validaciones sentaron las bases para el diseño de la arquitectura y el diseño de la interfaz de usuario

7. FASE 2 - FORMULACIÓN DE LA ARQUITECTURA

En esta sección se describe como se abordó la fase 2 de formulación de la arquitectura y cuáles fueron las actividades y los resultados obtenidos.

7.1. Diseño de la arquitectura de software

En este apartado se ha incluido el diseño arquitectónico que se propuso para dar solución a los requerimientos funcionales y no funcionales definidos en la fase 1 de este trabajo. En el ANEXO II se encuentra el documento de especificación de requerimientos y en el ANEXO III se encuentra el documento de arquitectura de software, se registra la descripción detallada de las consideraciones, restricciones y representación de la arquitectura a través del modelo 4 + 1 propuesto por Philippe Kruchten [18]. A continuación, se presenta de manera resumida la arquitectura propuesta, para un mayor detalle dirigirse al ANEXO III.

En la figura 7 se ilustra la vista lógica del diseño arquitectónico propuesto, en esta vista se identifican los componentes gruesos de la arquitectura. Se definió una arquitectura cliente – servidor, y se seleccionó el estilo arquitectónico modelo – vista – controlador (MVC, Model View Controller) siendo este el estilo dominante, de igual forma se incluye los componentes definidos y los conectores existentes entre dichos componentes.

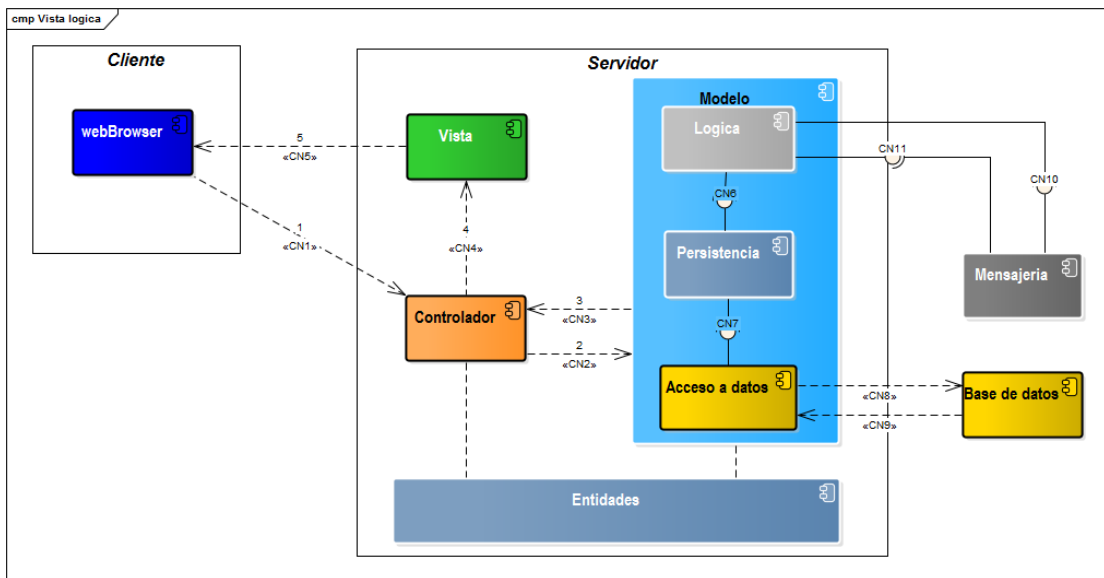


Figura 7. Vista lógica de la arquitectura

En la figura 8 se ilustra la vista de implementación que corresponde al detalle de la vista lógica. En esta vista, se identifican la estructura interna de los componentes principales ilustrados en la vista lógica y se adiciona un nivel de detalle mayor de los componentes y las conexiones que existen entre ellos.

A nivel de la lógica de negocio, se definió un componente de seguridad encargado de la autenticación y autorización en el sistema. Adicionalmente, se planteó un elemento de cifrado para la encriptación de la información sensible. Se planteó un componente de log de eventos para realizar la trazabilidad de errores en el sistema y también la captura de eventos que por su nivel de relevancia fuera necesario registrar la transaccionalidad que el usuario realizó.

Se planteó un componente asociado a la presentación, dicho componente tiene la responsabilidad de manejar la internacionalización de cara al usuario y adicionalmente, se encarga de los mensajes de ayuda al usuario.

Se definieron dos componentes asociados con la generación del proyecto en un archivo Java. El primer componente, se encarga de generar un archivo XML con toda la información del proyecto, es decir, relaciones, agentes, roles, variables y en general, toda la información necesaria para generar un archivo .jar con la especificación necesaria para ser gestionado por la plataforma de sistemas multiagentes BESA. El segundo componente (Generación proyecto JAR), parte de la especificación dada en el archivo XML generado en el componente Generación proyecto XML, y a partir de este, genera un archivo .jar con la especificación a nivel de clases Java del proyecto creado por el usuario a través de la plataforma Quemes.

Se defino un componente de persistencia de datos que en conjunto con el componente de acceso a datos proveen la parte transaccional de la plataforma Quemes. A este nivel se gestiona la creación, edición, consulta y eliminación de los registros en base de datos. De igual forma, se gestiona la conexión hacia el motor de base de datos.

En el componente de la vista, se representaron los frameworks empleados a nivel de presentación para la creación del espacio de trabajo, en el cual el usuario a través de los componentes visuales generados puede crear un proyecto a partir de la especificación de roles, agentes y relaciones entre ellos.

Con el fin de mantener el principio de alta cohesión bajo acoplamiento, se incluyeron interfaces y fachadas para desacoplar los componentes. De esta forma, se tiene menor dependencia entre componentes lo que mejora el mantenimiento, y de igual forma se define de forma concreta los propósitos de cada objeto dentro del sistema.

7.2. Evaluación de la arquitectura de software

Para la evaluación de la arquitectura propuesta, se realizó una adecuación del modelo ATAM considerando las etapas que aplicaran al contexto. En la figura 9 se presenta el árbol de utilidad construido para la evaluación. En él, se presenta la usabilidad como el atributo de calidad dominante seleccionado para el diseño. Asociado a este atributo, se identificaron las subcaracterísticas de este atributo y se definieron requisitos no funcionales que guiaron la definición de escenarios de prueba para evaluar la arquitectura.

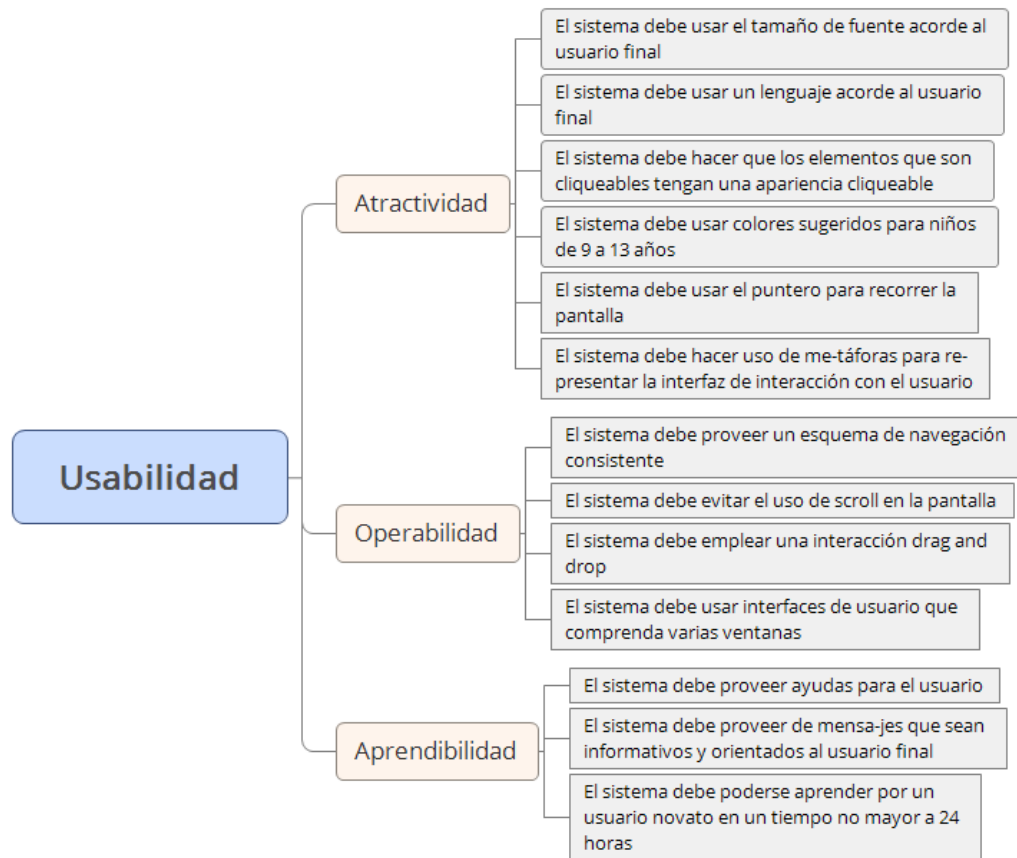


Figura 9. Árbol de utilidad

En la tabla 11 se ilustra la matriz que se elaboró a partir del árbol de utilidad. En esta matriz se realizó una priorización de los escenarios de prueba. Dicha priorización se realizó considerando las siguientes dimensiones:

- Por la importancia que cada escenario tiene para el éxito del sistema
- Por el grado de dificultad de lograr el escenario

La escala empleada para ambas dimensiones fue alto, medio y bajo.

Considerando la priorización de los escenarios se realizó un proceso iterativo con el fin de satisfacer a través de la arquitectura los escenarios de prueba propuestos. En la tabla 11, en la columna de Decisión, se registraron las decisiones arquitectónicas tomadas con el propósito de cumplir con el escenario de prueba definido.

Se realizaron cuatro iteraciones en el diseño de la arquitectura. En cada iteración se consideró los atributos de calidad de la norma ISO 9126 para alinear el modelo no solo al atributo de usabilidad sino a los demás atributos como lo son la funcionalidad, fiabilidad, eficiencia, mantenibilidad y la portabilidad, esto también con el fin de dar cumplimiento a los requerimientos no funcionales definidos en el ANEXO II.

Tabla 11. Evaluación de la arquitectura

Atributo de calidad	sub característica	Requisito	Necesidad de alcanzar el requisito	Dificultad de alcanzar el requisito	Descripción del escenario	Cumplimiento	Decisión
Usabilidad	Atractividad	El sistema debe usar el tamaño de fuente acorde al usuario final	alto	bajo	El sistema usa un tamaño de fuente apto para niños entre los 9 y 13 años	Se cumple	
	Atractividad	El sistema debe usar un lenguaje acorde al usuario final	alto	medio	El sistema debe usar lenguaje para niños entre los 9 a 13 años	Se cumple	
			alto	alto	Para facilitar la interacción con el sistema los mensajes y textos del sistema deben presentarse en el idioma del usuario	Se cumple	Este escenario nos condujo a considerar en la arquitectura un componente para tener la posibilidad de manejar multidioma
	Operabilidad	El sistema debe proveer un esquema de navegación consistente	medio	medio	La navegación del sistema es clara y fácil de recorrer. El usuario no debe hacer un mayor esfuerzo para recordar donde están las opciones	Se cumple	
Atractividad	El sistema debe hacer que los elementos que	alto	bajo	Los botones tienen una apariencia que emula la sensación de ser cliqueable	Se cumple		

		son clickeables tengan una apariencia clickeable					
Atractividad		El sistema debe usar colores sugeridos para niños de 9 a 13 años	alto	bajo	Los colores usados son orientados a niños entre los 9 y 13 años. Colores vivos, con un porcentaje alto de brillo	Se cumple	
Operabilidad		El sistema debe evitar el uso de scroll en la pantalla	alto	bajo	No se debe usar el scroll en más del 90% de las operaciones del sistema	No se cumple	Aun cuando no todas las opciones tienen la necesidad del uso del scroll. Se logró minimizar su uso. Sin embargo, no fue posible restringir su uso por completo. Se cumple actualmente en un 50%
Atractividad		El sistema debe usar el puntero para recorrer la pantalla	alto	bajo	Se usa el puntero del mouse como elemento para recorrer la pantalla	Se cumple	
Atractividad		El sistema debe hacer uso de metáforas para representar la	alto	medio	Se hace uso de metáforas que facilitan la asociación de conceptos con el mundo real	Se cumple	

		interfaz de interacción con el usuario					
	Operabilidad	El sistema debe emplear una interacción drag and drop	alto	alto	Se cuenta con una interacción drap and drop	Se cumple	La función de drap and drop se empleó para facilitar al usuario la ejecución de tareas en el espacio de trabajo.
	Operabilidad	El sistema debe usar interfaces de usuario que comprenda varias ventanas	alto	alto	Se usa un mecanismo de ventanas modal para evitar hacer transiciones entre pantallas que pueden generar pérdida de atención en el usuario final	Se cumple	Considerando que perfil de usuario, se implementó un sistema de ventanas modales cuando era requerido mostrar o ampliar la información. De esta manera se mitiga la posibilidad de que el usuario pierda la concentración o el interés
	Aprendibilidad	El sistema debe proveer ayudas para el usuario	alto	medio	Se cuenta con manual de usuario y/o ayudas durante la interacción del usuario con el sistema	Se cumple	A través de este escenario se apoyó la consideración de un componente para administrar los

							mensajes que se muestran al usuario
Aprendibilidad	El sistema debe proveer de mensajes que sean informativos y orientados al usuario final	alto	medio	El sistema usa tooltips para mostrar ayudas que guíen al usuario	Se cumple	Este escenario nos condujo a considerar en la arquitectura un componente para administrar los mensajes que se muestran al usuario	
		alto	medio	Los mensajes que se muestran al usuario aportar la suficiente información para guiar al usuario en las acciones que debe realizar para continuar con el uso normal del sistema	Se cumple	Este escenario nos condujo a considerar en la arquitectura un componente para administrar los mensajes que se muestran al usuario	
Aprendibilidad	El sistema debe poderse aprender por un usuario novato en un tiempo no mayor a 24 horas	medio	medio	Un usuario novato aprende a usar el sistema en menos de 24 horas			

7.3. Diseño y evaluación de la interfaz de usuario

El diseño de la interfaz de usuario fue elaborado en la herramienta Illustrator CC6 y fue abordado mediante mockups de la aplicación. A partir de los mockups se realizaron las propuestas de diseño de interfaz y la propuesta de navegación. Se empleó la herramienta inVision para la presentación del mockup y la propuesta de navegación.

Para la validación de esta propuesta se contó con el apoyo de dos expertos que aportaron desde el conocimiento de su campo de acción para guiar la disposición de los elementos, uso de imágenes, colores y en general el diseño de la interfaz. De igual forma se consideraron los principios de diseño orientados a niños identificados en la fase uno de este proyecto.

Finalmente, se presentó la propuesta de diseño al cliente, quien realizó una retroalimentación al diseño propuesto. Los comentarios aportados fueron considerados para realizar una nueva iteración y generar una nueva propuesta. En total se realizaron tres iteraciones.

En la tabla 12 y 13 se presenta el resultado de la evaluación realizada por dos expertos en diseño gráfico, tal como se observa se obtuvieron recomendaciones y directrices que se utilizaron para la toma de decisiones en el diseño de la interfaz de usuario.

Tabla 12. Resultado evaluador uno

Evaluador 1	
Profesión	Diseñador grafico
Comentarios	<p>El evaluador resalto la interfaz de login y destacó el uso de imágenes para captar la atención del usuario.</p> <p>El evaluador realizó las siguientes propuestas de mejora:</p> <ul style="list-style-type: none"> - No usar más de dos tipos de fuentes diferentes. - Usar una imagen en el background del espacio de trabajo del usuario que contextualice lo que está haciendo. Por ejemplo, una imagen para indicar que está atendiendo un desastre natural. - Usar una la paleta de colores para niños. Se sugirió emplear la de adobe. - Se sugirió emplear una tipografía sans serif de terminación redonda

Tabla 13. Resultado evaluador dos

Evaluador 1	
Profesión	Ingeniero de sistemas / Ingeniero de frontend

Comentarios

El evaluador destaco el planteamiento de la navegación, resalto que es clara y e intuitiva. De igual forma destaco el uso de imágenes para representar objetos del mundo real como una buena estrategia para ayudar a contextualizar al usuario.

- El evaluador sugirió utilizar colores con un porcentaje mayor de brillo.
- Recomendó manejar un fondo blanco o de un color tenue en el espacio de trabajo para ayudar a descansar la vista del usuario.
- Recomendó usar un tipo de fuente con terminaciones redondas.

En la tabla 14 se presenta parte de la retroalimentación resultate de la evaluación con el cliente. Se adaptaron aquellas recomendaciones que se alineaban con los principios de diseño identificados en la fase uno del proyecto. Asimismo, se presenta un resumen de los aportes realizados en la evaluación con el cliente, a partir de estos aportes se realizaron nuevas propuestas de diseño arquitectónico y de interfaz con el ánimo de considerar aquellas indicadas por el cliente que enriquecieran el modelo y que no fueran en contra de los requerimientos funcionales y no funcionales y de los lineamientos heurísticos y de diseño identificados y definidos en etapas anteriores.

Tabla 14. Resultado evaluación con el cliente

Propuestas del cliente
Emplear una cuadrícula en el backgorund del espacio de trabajo
Ubicar el logo de Quemes en la barra de menú
Ubicar los elementos que representan los agentes en la parte lateral izquierda o derecha
Ubicar los elementos que representan los roles en la parte superior
Ubicar el menú principal al costado lateral izquierdo
Ubicar la opción de salir en la parte superior derecha
En la pantalla de creación de variables dejar el botón eliminar al final de cada fila
Las opciones de los agentes dejarlas en la parte superior como menú principal
Tener un menú pequeño en la parte lateral izquierda con las opciones del proyecto

En la tabla 15 se ilustra el resultado de la evaluación de los principios de diseño realizada a la propuesta de la interfaz de usuario. Dicha evaluación, se realizó a partir de los principios de diseño orientados a niños, lo cual arrojó resultados bastante favorables para el diseño de la aplicación. Los lineamientos obtenidos enriquecieron el modelo y proporcionaron una guía en el diseño propuesto, así como permitieron mejorar la interacción del usuario con el sistema. En total se lograron aplicar satisfactoriamente en el diseño 22 de los 23 principios propuestos. El

principio asociado a evitar el uso de scroll se aplicó de forma parcial, debido a restricciones tecnológicas y por la necesidad de visualizar contenidos extensos en la aplicación.

Tabla 15. Resultado evaluación principios de diseño

Principios de diseño orientados a niños (9 - 13 años)		Estado actual
Interacción General	Evitar el uso de scroll	Parcialmente
	Proveer Instrucciones explicitas	Se cumple
	Utilizar iconos y símbolos en formas familiares	Se cumple
	Uso del puntero para recorrer la pantalla	Se cumple
Texto	Fuentes relativamente grandes (12, 14)	Se cumple
	No usar texto animado	Se cumple
	Uso de texto fácilmente comprensible y conciso	Se cumple
	Proporcionar instrucciones que sean siempre accesibles	Se cumple
	Internacionalización	Se cumple
	Lenguaje apropiado	Se cumple
Navegación	Esquema de navegación consistente	Se cumple
	Categorías significativas	Se cumple
	Denotar donde está ubicado el usuario	Se cumple
Interfaz	Hacer que los elementos que son cliqueables luzcan como cliqueables	Se cumple
	Hacer que las imágenes que son cliqueables luzcan como cliqueables	Se cumple
	Botones en lugar de textos como links	Se cumple
	Diseño enfocado en la edad	Se cumple
	Uso de colores vivos, brillos	Se cumple
	Uso de iconos e imágenes	Se cumple
	Uso de metáforas	Se cumple
	Diseño llamativo	Se cumple
	Uso estándar de interacción drag and drop	Se cumple
	Crear interfaces de usuario que comprende varias ventanas	Se cumple

En esta sección se presentaron las consideraciones para el diseño y validación de la interfaz de usuario, así como el resultado obtenido a partir de la validación con expertos en diseño gráfico. De igual forma, se presentó el diseño de la arquitectura propuesta para atender los requerimientos no funcionales del sistema. Finalmente, se presentó el resultado de la evaluación de la arquitectura a través del método ATAM y como esta guio la toma de decisiones arquitectónicas hasta llegar a la propuesta plasmada en este documento y en el ANEXO III.

8. FASE 3 - CONSTRUCCIÓN DEL PROTOTIPO

Como uno de los artefactos de validación de la arquitectura, del diseño de interfaz centrado en el usuario y en general de todo el proceso de investigación generado a través de este proyecto se realizó la implementación de un prototipo funcional. Este prototipo se construyó usando la tecnología Java Enterprise Edition y se implementó a través de Maven una herramienta para la gestión y construcción de proyectos.

A continuación, se presentan las herramientas y tecnologías seleccionadas:

Tabla 16. Tecnologías empleadas

Descripción	Item seleccionado
Entorno integrado de desarrollo	Netbeans 8.2
Lenguaje de programación	Java EE 7
Java Developer Kit	1.7
Gestor de proyecto	Maven
Servidor de aplicaciones	Glassfish 4.0
Motor de base de datos	PostgreSQL 9.6.2

8.1. Presentación del prototipo

A continuación, se realiza una descripción del prototipo construido. En la figura 10 se muestra el acceso al sistema que se construyó y en la tabla 17 las características de esta pantalla.

Tabla 17. Presentación del prototipo

Indicador	Elemento	Descripción
A	Ayuda	Haciendo clic presenta una ayuda inmediata para el usuario que le guía en el proceso de ingreso al sistema
B	Campo usuario	En este campo se digita el usuario para ingresar al sistema
C	Campo contraseña	En este campo se digita la contraseña del usuario para ingresar al sistema
D	Opción recuperación de contraseña	A través de esta opción se realiza un proceso de recuperación de la contraseña
E	Botón ingresar	Genera el evento para ingresar al sistema
F	Opción de registro	A través de esta opción se realiza el proceso de registro en el sistema



Figura 10. Ingreso al sistema

En la figura 13 se ilustra las ayudas que se implementaron para guiar al usuario en el proceso de ingreso al sistema. En el caso de la ayuda se despliega un componente con una breve descripción que guía al usuario. Para los campos de texto se emplearon colores para denotar estados de los elementos. Se emplearon las siguientes representaciones visuales, para el estado normal se usa la representación de la figura 11, cuando los campos son obligatorios se usa el color rojo y se muestra un texto debajo de cada campo indicando al usuario lo que debe hacer para continuar el proceso. En la figura 12 se muestra el estado del campo cuando se ha ingresado un valor.

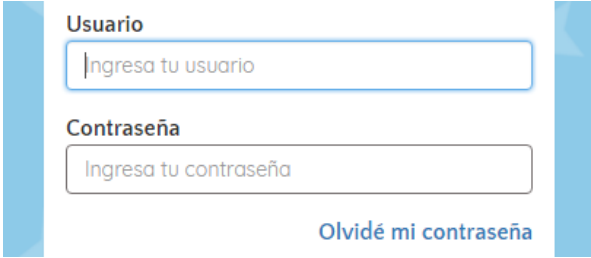
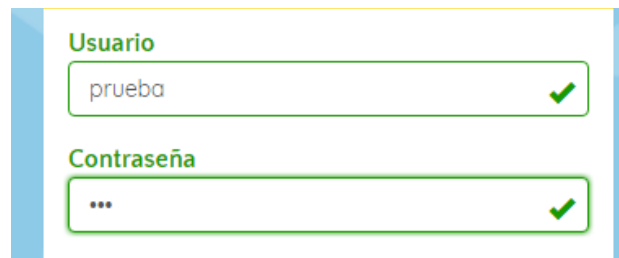


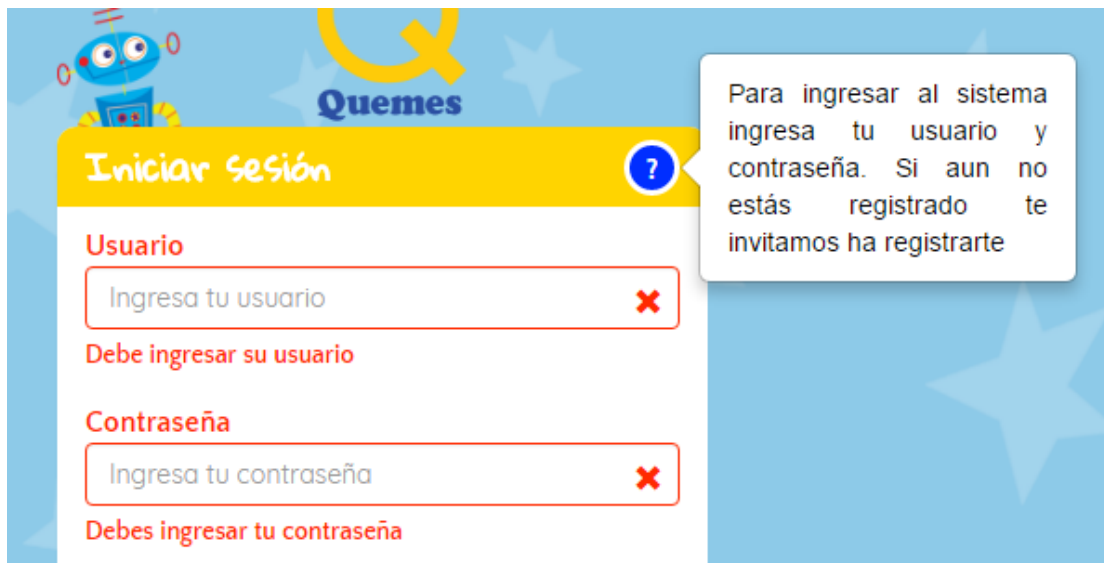
Figura 11. Ayuda visual en campos



The image shows a login form with two input fields. The first field is labeled 'Usuario' and contains the text 'prueba'. The second field is labeled 'Contraseña' and contains three dots. Both fields have a green checkmark icon on the right side, indicating that the fields are required and have been filled out.

Figura 12. Ayuda visual campo obligatorio

En la figura 13 se ilustra la ayuda implementada para guiar al usuario en el uso del sistema.



The image shows a login form titled 'Iniciar sesión' with a yellow header. The form has two input fields: 'Usuario' and 'Contraseña'. Both fields have a red border and a red 'X' icon, indicating they are required. Below each field is a red error message: 'Debe ingresar su usuario' and 'Debes ingresar tu contraseña'. A blue tooltip with a question mark icon is positioned over the 'Iniciar sesión' header, containing the text: 'Para ingresar al sistema ingresa tu usuario y contraseña. Si aun no estás registrado te invitamos ha registrarte'.

Figura 13. Ayuda contextual

Para los botones se utilizaron dos efectos visuales para ilustrar los estados de este elemento. En la figura 14 se indica el botón en estado normal, se generó un efecto de profundidad para dar al usuario la sensación de que puede pulsarse. En la figura 15 se ilustra el botón cuando es pulsado, se realiza un cambio de color y se remueve el efecto de profundidad para crear en el usuario la sensación de que ha sido pulsado.

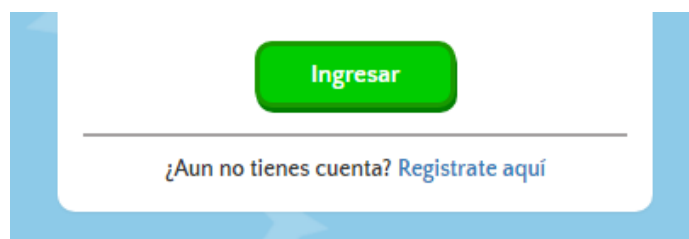


Figura 14. Estado normal de un botón



Figura 15. Representación visual de un botón cuando es presionado

En la figura 16 se ilustra el espacio de trabajo que se definió para la elaboración de un proyecto por parte de los usuarios. En la tabla 18 se listan los componentes y una breve descripción de cada uno de ellos.

Tabla 18. Elementos del espacio de trabajo del prototipo

Indicador	Elemento	Descripción
A	Menú	Se encuentran las opciones asociadas al proyecto, como nuevo, guardar y abrir.
B	Escenario	En esta opción el usuario puede cambiar el espacio de trabajo de programación. Las opciones posibles son roles o agentes.
C	Roles	Se listan los roles disponibles para ser agregados al espacio de trabajo
D	Estaciones	Se listan las estaciones disponibles para ser agregadas al espacio de trabajo
E	Tipo de conexiones	Se habilitan las opciones posibles para el cambio de conexión
F	Espacio de trabajo	En el espacio de trabajo se agregan los roles y se realizan las conexiones que existen entre los elementos.
G	Rol	Representa un robot en el mundo real. A través de este elemento se definen las variables a nivel de rol. Estas variables se asocian a los agentes que se crean del mismo tipo de rol.
H	Componente de eliminación	Al arrastrar los elementos del espacio de trabajo sobre la papelera se remueven del espacio de trabajo y por lo tanto del proyecto.
I	Conexión	Representa una conexión entre dos elementos del espacio de trabajo. Cuando se adiciona se abre la posibilidad de adicionar las variables asociadas a la conexión

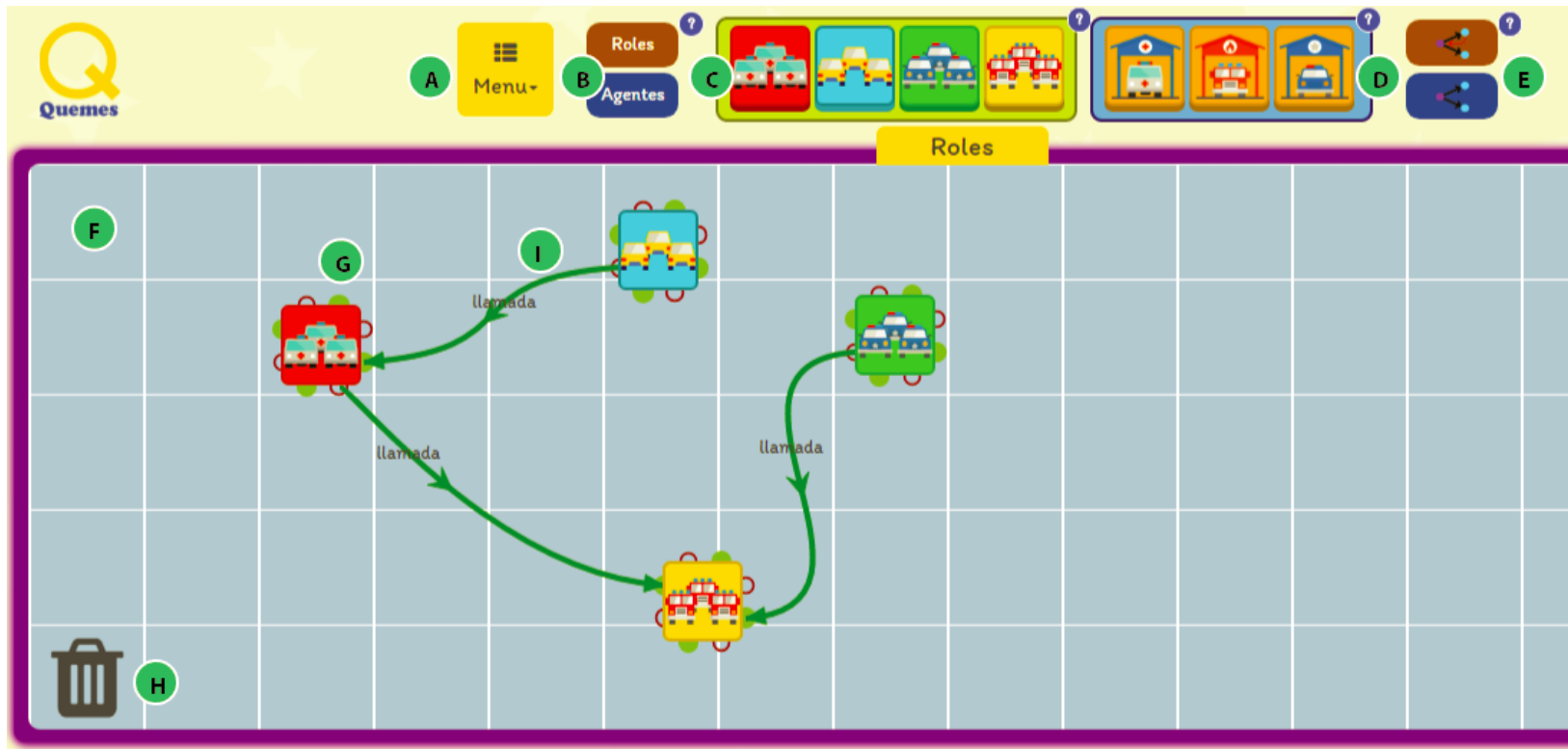


Figura 16. Espacio de trabajo prototipo

9. RESULTADOS DE LA VALIDACIÓN

En esta sección se presentan los resultados de la validación realizada al prototipo construido. A continuación, se detalla el manejo dado y la manera como se usaron los dos instrumentos aplicados para la validación.

Los instrumentos empleados, utilizan la escala Likert para conocer el grado de conformidad de los usuarios encuestados [54]. Adicionalmente, se utilizó el coeficiente de Cronbach (alfa de Cronbach) para medir la confiabilidad de los instrumentos empleados para la validación; es decir, en qué grado el instrumento mide aquello que se pretende medir. El alfa de Cronbach, arroja un valor entre 0 y 1, donde los valores más cercanos a 1 representan una mayor consistencia interna de los ítems analizados.

El alfa de Cronbach está dado por la siguiente formula:

$$\alpha = \frac{K}{K - 1} \left[1 - \frac{\sum S_i^2}{S_T^2} \right]$$

Dónde:

K : es el número de ítems considerados

$\sum S_i^2$: es la sumatoria de la varianza de cada ítem

S_T^2 : es la varianza de la suma de los ítems

La varianza está dada por la siguiente formula:

$$S^2 = \frac{\sum (X_i - \bar{X})^2}{n - 1}$$

Donde:

X_i : es un término del conjunto de datos

\bar{X} : es la media de la muestra

n : es la cantidad de puntos de datos

George y Mallory [55] proporcionan la escala mostrada en la tabla 19 para evaluar el coeficiente alfa de Cronbach:

Tabla 19. Escala alfa de cronbach

Alfa Cronbach	Resultado
Mayor a 0,9	Excelente

Entre 0,8 - 0,89	Bueno
Entre 0,7 - 0,79	Aceptable
Entre 0,6 - 0,69	Cuestionable
Entre 0,5 - 0,59	Pobre
Menor a 0,5	Inaceptable

En la tabla 20 se ilustra las características demográficas de los usuarios que participaron en el proceso de validación de la percepción de la usabilidad del usuario final.

Tabla 20. Características demográficas de los usuarios de prueba

Usuario	Edad	Género	Estrato	Tipo de colegio
Usuario 1	11	Femenino	3	Privado
Usuario 2	13	Femenino	3	Privado
Usuario 3	10	Femenino	3	Privado
Usuario 4	11	Masculino	3	Público
Usuario 5	9	Masculino	3	Público
Usuario 6	9	Masculino	3	Público
Usuario 7	12	Masculino	3	Privado
Usuario 8	9	Femenino	3	Privado

9.1. Percepción de usabilidad a través del instrumento SUS

El cuestionario Escala de Usabilidad del Sistema (SUS, System Usability Scale), es una escala Likert que genera un único valor, el cual representa el grado de conformidad de usabilidad global del sistema que ha sido sometido a estudio [56]. El resultado obtenido a través de este cuestionario es un valor entre 0 y 100. Está compuesto por diez ítems de evaluación cuya escala es del 1 al 5, donde el uno (1) representa totalmente en desacuerdo y el cinco (5) representa totalmente de acuerdo. Es importante resaltar que las puntuaciones independiente de cada ítem no son significativos por sí mismos [57].

Para calcular la puntuación del SUS, hay que sumar primero las contribuciones de cada punto. La contribución de cada punto valdrá entre 0 y 4. Para los puntos 1, 3, 5, 7 y 9, la contribución será la posición de la escala menos 1. Para los puntos 2, 4, 6, 8 y 10, la contribución será 5 menos la posición en la escala. En este sentido, se multiplica la suma de los resultados por 2.5 para obtener el valor global del SUS. El promedio de usabilidad es de 68, un valor mayor indica un nivel de usabilidad mejor que el promedio.

Los resultados obtenidos se ilustran en la tabla 21, para lo cual se empleó una muestra de ocho usuarios. Como se puede observar, este resultado representa la percepción de usabilidad del usuario en su interacción con el prototipo construido.

Tabla 21. Resultado evaluación SUS

	Pregunta	Resultado
1	Creo que me gustaría usar este sistema con frecuencia	30
2	El sistema me pareció complejo de usar	26
3	El sistema fue fácil de usar	28
4	Creo que necesitaría el apoyo de un experto para poder usar este sistema	26
5	He encontrado que las diversas funciones de este sistema eran claras y definidas	32
6	Encontré inconsistencias en este sistema	32
7	Pienso que la mayoría de la gente aprendería a usar este sistema rápidamente	27
8	El sistema me pareció muy engorroso de usar	28
9	Me sentí muy seguro usando el sistema	29
10	Necesite aprender muchas cosas antes de poder usar este sistema	28
Usabilidad global del sistema		89,38

Para interpretar el resultado se usó la escala propuesta por Bangor et al [58] mostrada en la figura 17. De acuerdo a esta escala y al resultado obtenido en nuestro caso de 89.38, de acuerdo al rango de aceptabilidad mostrado en la figura 17, la plataforma tiene un alto grado de aceptación por parte del usuario. Ubicado en un grado de escala B lo cual representa un rating Excelente respecto a la interfaz del sistema.

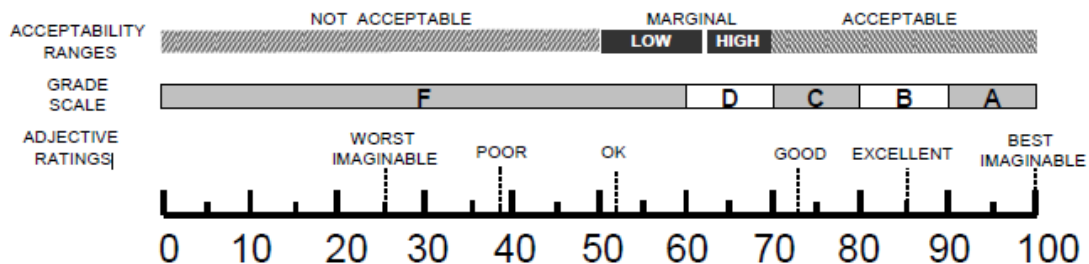


Figura 17. Escala SUS [58]

En el ANEXO IV se encuentra una descripción detallada de la formulación realizada para calcular el coeficiente alfa de Cronbach. En la tabla 22 se ilustra el resultado de las respuestas de la encuesta y el cálculo de la varianza total y por ítem. A partir de este resultado se calculó el alfa de Cronbach cuyo valor fue de 0,98.

A continuación, se presenta el cálculo del coeficiente del alfa de Cronbach:

$$\alpha = \frac{10}{10 - 1} \left[1 - \frac{6,32}{3,36} \right] = 0,98$$

De acuerdo a la escala de George y Mallory [55] la confiabilidad de la encuesta es Excelente, lo que significa que existe una alta consistencia interna del instrumento de medida y que sus mediciones son estables y consistentes. Por lo tanto, los resultados son fiables para medir la percepción de usabilidad del sistema.

Tabla 22. Resultado de las respuestas de la encuesta SUS

	Preguntas										
Usuario	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Total
Usuario1	3	1	5	3	5	1	3	1	5	1	28
Usuario2	5	1	5	3	5	1	4	1	5	2	32
Usuario3	5	5	5	1	5	1	5	1	4	1	33
Usuario4	5	1	3	1	5	1	5	1	4	3	29
Usuario5	5	1	5	1	5	1	5	1	5	2	31
Usuario6	5	2	4	2	5	1	4	4	5	1	33
Usuario7	5	1	5	2	5	1	4	1	5	1	30
Usuario8	5	2	4	1	5	1	5	2	4	1	30
suma ítems	38	14	36	14	40	8	35	12	37	12	246
Media muestra	4,75	1,75	4,50	1,75	5,00	1,00	4,38	1,50	4,63	1,50	30,75
Varianza	0,50	1,93	0,57	0,79	0,00	0,00	0,55	1,14	0,27	0,57	3,36

9.2. Percepción de usabilidad a través del instrumento PSSUQ

El cuestionario de usabilidad del sistema post estudio (PSSUQ, Post-Study System Usability Questionnaire) permite a los participantes proporcionar una evaluación general del sistema que se está usando. Comprende 19 puntos que permiten evaluar la satisfacción del usuario con la usabilidad del sistema. A partir de esta evaluación, se pueden medir cuatro dimensiones de la usabilidad: la satisfacción total del sistema, utilidad del sistema, calidad de la información y la calidad de la interfaz utilizando como escala uno para el valor mínimo y siete para el valor máximo posible.

En la tabla 23 se ilustra el resultado obtenido en las cuatro dimensiones de evaluación del instrumento PSSUQ. Según se observa, en las cuatro dimensiones se obtuvo un valor altamente favorable. En la satisfacción general el resultado obtenido fue de 6.89 lo cual indica una alta percepción de usabilidad por parte del usuario con la interacción del sistema. En cuanto Utilidad del sistema se obtuvo un resultado de 6.94. El valor más alto fue 6.96 en el ítem Calidad de la interfaz. Y el valor más bajo que fue de 6.84 que hace referencia a la calidad de la información. Todos estos valores en encuentran en un rango óptimo, ya que la dimensión máxima que mide la herramienta es de 7.0 lo cual representa un resultado altamente satisfactorio.

Tabla 23. Resultado cuatro dimensiones PSSUQ

Usuario	Satisfacción general	Utilidad del sistema	Calidad de la información	Calidad de la interfaz
Usuario1	6.95	6.88	7.00	7.00
Usuario2	6.84	6.88	6.71	7.00
Usuario3	6.84	7.00	6.57	7.00
Usuario4	6.95	7.00	7.00	7.00
Usuario5	7.00	7.00	7.00	7.00
Usuario6	6.84	6.88	6.71	7.00
Usuario7	6.89	7.00	6.86	7.00
Usuario8	6.84	6.88	6.86	6.67
Total	6.89	6.94	6.84	6.96

En la figura 18 se ilustra los resultados obtenidos en la evaluación de percepción de usabilidad del usuario en su interacción con el prototipo construido. En el gráfico de araña se identifica

un equilibrio en las cuatro dimensiones evaluadas a través del instrumento PSSUQ, los resultados obtenidos son muy cercanos al valor máximo posible de la escala, indicando una satisfacción general con la interacción del sistema.

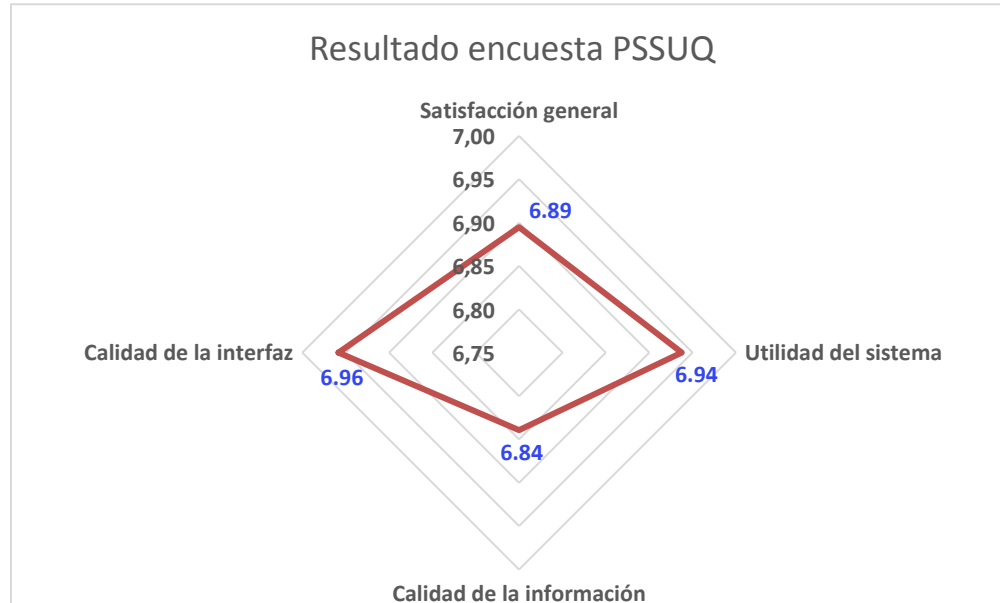


Figura 18. Resultado evaluación PSSUQ

En el ANEXO V se encuentra una descripción detallada de la formulación realizada para calcular el coeficiente alfa de Cronbach. Así mismo, en la tabla 23 se ilustra el resultado de las respuestas de la encuesta y el cálculo de la varianza total y por ítem evaluado.

La última columna de la tabla 24 corresponde a la suma de las valoraciones de cada usuario. Este valor permite apreciar la percepción general de cada usuario con respecto al sistema. El valor máximo posible a obtener es de 133 y el valor mínimo es 19. De acuerdo a los resultados obtenidos (media total de la muestra 131) se identifica una alta satisfacción por parte de los usuarios en su interacción con el sistema.

A continuación, se presenta el cálculo del coeficiente del alfa de Cronbach:

$$\alpha = \frac{19}{19 - 1} \left[\left| 1 - \frac{2,5}{1,43} \right| \right] = 0.8$$

A partir de este resultado se calculó el alfa de Cronbach total cuyo valor fue de 0.8. De acuerdo a la escala de George y Mallory [55] la confiabilidad de la encuesta es Buena, lo que significa que existe una buena consistencia interna del instrumento de medida y que sus mediciones son estables y consistentes. Por lo tanto, los resultados son fiables para medir la percepción de usabilidad del sistema.

Tabla 24. Resultado de las respuestas de la encuesta PSSUQ

Usuario	Pregunta																			Total	
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19		
Usuario1	7	7	7	7	7	7	7	6	7	7	7	7	7	7	7	7	7	7	7	7	132
Usuario2	7	7	7	7	7	6	7	7	7	6	7	6	7	7	7	7	7	7	7	7	130
Usuario3	7	7	7	7	7	7	7	7	7	7	7	7	7	4	7	7	7	7	7	7	130
Usuario4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	6	132
Usuario5	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	133
Usuario6	7	6	7	7	7	7	7	7	7	7	7	7	7	6	6	7	7	7	7	7	130
Usuario7	7	7	7	7	7	7	7	7	7	7	6	7	6	7	7	7	7	7	7	6	131
Usuario8	7	7	7	7	7	6	7	7	7	7	7	7	7	7	7	7	7	6	7	7	130
suma items	56	55	56	56	56	54	56	55	56	54	56	54	56	52	55	56	55	56	54	54	1048
Media	7,0	6,9	7,0	7,0	7,0	6,8	7,0	6,9	7,0	6,8	7,0	6,8	7,0	6,5	6,9	7,0	6,9	7,0	6,8	6,8	131
varianza	0,0	0,1	0,0	0,0	0,0	0,2	0,0	0,1	0,0	0,2	0,0	0,2	0,0	1,1	0,1	0,0	0,1	0,0	0,2	0,2	1,4

9.3. Validación del prototipo aplicando las heurísticas de usabilidad

En la figura 19 se ilustra el resultado final obtenido al aplicar las heurísticas de usabilidad definidas en la fase I de este proyecto. De acuerdo a los resultados obtenidos se identificó que se llegó a cumplir el 70% de las heurísticas planteadas y se cumplió de forma parcial con el 30%. Al contrastar con la validación inicial, se alcanzó un valor bastante significativo, se paso de no cumplir ninguno de los principios heurísticos a cumplir siete.

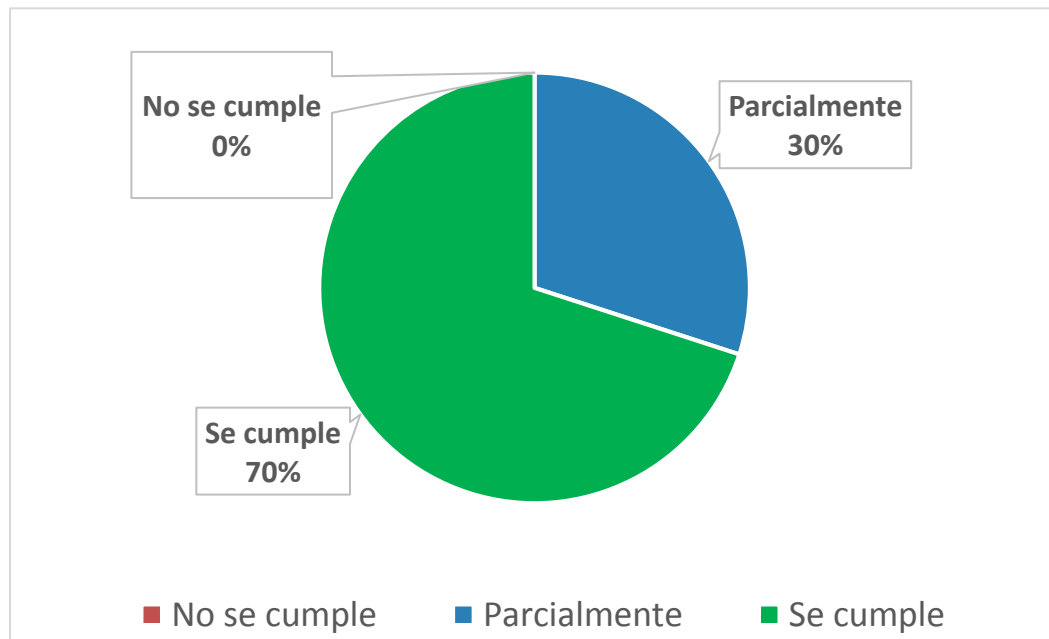


Figura 19. Resultado validación heurística

9.4. Validación del prototipo aplicando los principios de diseño orientados a niños

En la figura 20 se ilustra el resultado obtenido después de aplicar los principios de diseño enfocados en niños en el diseño de la interfaz de usuario. De acuerdo a los resultados obtenidos se identificó un aumento del 96% de cumplimiento de los principios seleccionados, se paso a cumplir de forma parcial el 4% de los principios y no se dejaron de cumplir principios de diseño. Al hacer el contraste con el estado inicial, se paso de cumplir 4 principios a 22, siendo un valor bastante representativo indicando un buen nivel de mejora. Adicionalmente, se paso de cumplir de forma parcial de 3 a 1 principio, indicando una reducción en la consideración parcial de los principios de diseño identificados. De forma general, se logró implementar en un gran porcentaje (96%) los principios de diseño seleccionados, logrando así, mejorar el diseño de la interfaz de interacción del sistema con el usuario.

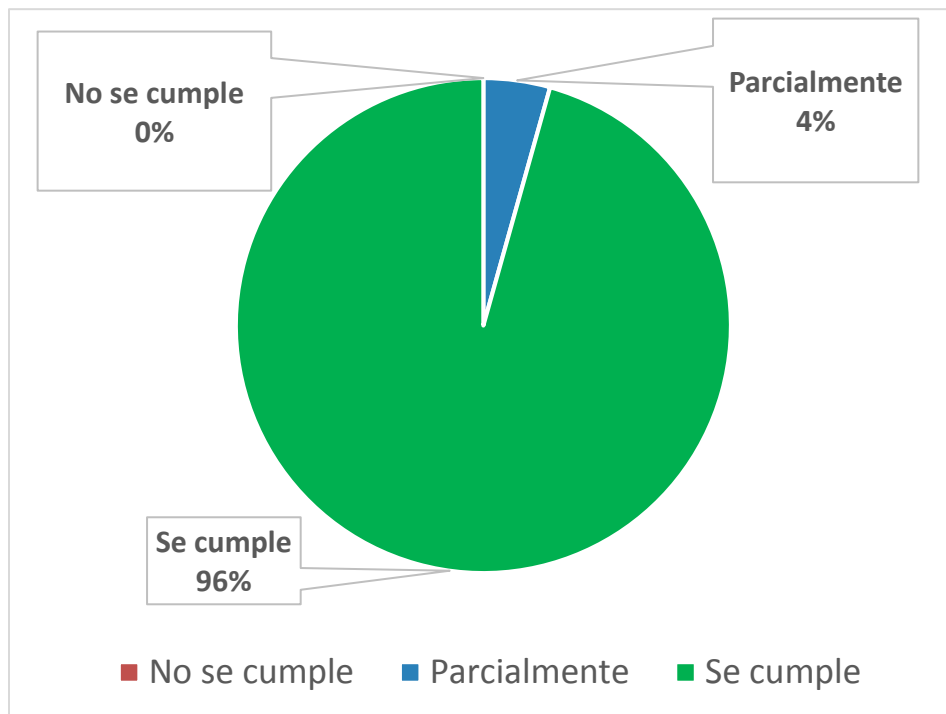


Figura 20. Validación principios de diseño

En esta sección se presentó el proceso de validación empleado para determinar la percepción de usabilidad por parte del usuario en su interacción con el prototipo elaborado. Se tabulo la información recogida y se obtuvo un resultado para cada uno de los instrumentos de validación aplicados. Adicionalmente, se realizó una verificación de la confiabilidad de los instrumentos empleados al calcular el coeficiente Alfa de Cronbach, el cual indico que se tenía un buen grado de confiabilidad de los instrumentos y de la información obtenida por medio de ellos. Finalmente, se identificó una mayor satisfacción en la interacción del usuario con el sistema.

10. IMPACTOS

Los resultados obtenidos en la ejecución del presente proyecto y en general el producto obtenido, le aportó considerablemente al trabajo de investigación que se ha venido desarrollando a través del proyecto Quemes, por lo que a continuación se presentan los siguientes impactos:

- Quemes se consolida y se afianza como una herramienta que apoya el aprendizaje en las aulas, facilitando el desarrollo de competencias sociales, el trabajo, en equipo y la colaboración.
- Los cambios realizados en la capa de presentación de la plataforma Quemes facilitan al usuario lograr sus objetivos en la interacción, ya que además de su mejora de usabilidad, el usuario cuenta con ayudas en línea.
- Con el esquema de despliegue definido se logra reducir costos asociados a la mantenibilidad del sistema y se facilita la realización de actualizaciones al ser estas efectuadas solo en el servidor donde esta publicado el software.
- El proyecto impacta positivamente en el proceso de formación de los estudiantes, dado que a través de las actividades realizadas en interacción con el sistema y ejercicios que este permite realizar pueden mejorar sus habilidades y fortalecer competencias cognitivas.
- Se entrega una plataforma web que facilita el trabajo cooperativo entre estudiantes ubicados geológicamente en diferentes lugares, de esta manera se potencializa el desarrollo de habilidades de colaboración y se inculca el uso de las tecnologías como un material de apoyo en su proceso de formación.

11. CONCLUSIONES

A partir de las validaciones planteadas en la fase de diagnóstico se logró identificar las falencias a nivel de usabilidad presentes en la plataforma Quemés. Esta identificación, permitió diseñar una estrategia aplicable para solventar las falencias identificadas o minimizarlas, con el objetivo de mejorar la experiencia del usuario en el uso de la plataforma Quemés. Lo cual dio como resultado, el logro de tener una evaluación cuantitativa en términos de usabilidad de la plataforma, y una métrica de evaluación inicial con la que se realizó la evaluación final; obteniendo una directriz palpable de los logros alcanzados en términos de usabilidad a través de esta propuesta.

Como metodología de desarrollo se empleó diseño centrado en el usuario, la cual se planteó de manera tal que se permitiera trabajar de forma cercana con el cliente y considerar al usuario a lo largo del proyecto. Esta metodología, facilitó la detección temprana de conformidades e inconformidades del cliente y una reacción rápida para atender sus requerimientos, guiando al cliente a través de un proceso de construcción por fases que permitiera la culminación de entregables y donde las mejoras o recomendaciones fueran abordadas en una fase posterior a la liberación de cada entregable. Lo cual permitió, asegurar la delimitación de los requerimientos del cliente y no llegó a convertirse en un ciclo interminable de iteraciones con el mismo.

Uno de los puntos en los cuales se invirtió gran parte de este trabajo fue el diseño de la arquitectura de software. La arquitectura cobra un valor significativo cuando se toma conciencia del impacto que tiene sobre la capacidad del sistema para satisfacer los atributos de calidad que sustentan los requerimientos no funcionales de una aplicación. En consecuencia, uno de los retos importantes de este trabajo representó alinear el atributo de usabilidad con la arquitectura, sin desarraigar los demás atributos de calidad definidos en la norma ISO 9126. Para lograrlo, se estudió a profundidad como interviene este atributo en la arquitectura, se consideraron los resultados de la evaluación heurística y de diseño centrado en niños, para plantear escenarios de prueba que a través de una adecuación del modelo de evaluación de arquitectura (ATAM) guiara la toma de decisiones arquitectónicas.

El diseño de la interfaz de usuario fue abordado mediante el modelo de construcción de mockups. Este modelo, facilitó la interacción con el cliente, obteniendo retroalimentación en una fase temprana del proyecto. Se logró ahorrar tiempo de implementación y no llegar al retrabajo.

A través del prototipo funcional que se implementó, se aplicaron los principios heurísticos de usabilidad y los lineamientos de diseño identificados en la fase uno de este trabajo. Adicionalmente, se implementaron exitosamente los componentes de la arquitectura necesarios para el prototipo que se construyó, así como para el diseño de interfaz de usuario definido. Con la armonización de estos elementos y como resultado el prototipo, se realizó una validación con el usuario en la que se corroboró que la percepción de usabilidad en la interacción con el sistema había mejorado a partir del trabajo aquí realizado.

Con este trabajo se logró alcanzar el objetivo general propuesto relacionado con mejorar la usabilidad del sistema Quemés. Dado que a partir de las pruebas realizadas con el usuario se evidenció una mejora representativa en la percepción de usabilidad, identificada a través de los

instrumentos de evaluación empleados. En la validación con el instrumento SUS se obtuvo un resultado de 89.38, lo cual posiciona la usabilidad del sistema en una calificación excelente de acuerdo a dicha escala. Adicionalmente, a través del instrumento PSSUQ, se obtuvo una satisfacción general del sistema de 6.89, el cual es un valor próximo al máximo posible, para nuestro caso siete (7). A partir de estos resultados, se confirma que esta propuesta cuenta con los conceptos, principios, lineamientos y diseños necesarios para mejorar la interacción del usuario con la plataforma Quemes y que, además, se aplicaron satisfactoriamente.

Una de las ganancias que tiene el proyecto Quemes con este trabajo, corresponde a una nueva propuesta de la interfaz de programación enfocada en el usuario final, lo cual hace que la interacción con el sistema sea fácil, cómoda, sencilla e intuitiva, y esto permite que Quemes se posicione en el ámbito comercial como un producto sólido, estable y amigable visualmente al usuario, agregando un valor diferenciador ante otros productos que pudieran ofrecer características similares. En este sentido Quemes se consolida como un sistema de fácil acceso, uso y que genera un impacto positivo en los usuarios.

El plantear esta propuesta como un modelo web, aporta a la plataforma quemes una posibilidad de llegar a más usuarios, facilitando así su comercialización, alcance, y mantenibilidad del sistema ya que no sería necesario procesos largos y dispendiosos de actualizar cada una de las maquinas donde se instala el producto, sino por el contrario, esta actualización se realizaría solamente donde esta publicado el producto. Adicionalmente, se logra la disminución de costos asociados a la actualización que se tiene que realizar por equipo, debido al esquema de despliegue del software actual.

12. TRABAJOS FUTUROS

Con el planteamiento de la plataforma a través de la web se abre la posibilidad de cambiar el sistema de comunicación de la plataforma con los robots. Es decir, cambiar el protocolo bluetooth que se tiene actualmente y establecer un nuevo mecanismo a través de http. Esto eliminaría la necesidad de tener que registrar los robots a la máquina y quedar asociado al sistema operativo; al final se lograría una mayor portabilidad de todo el sistema, permitiendo la ejecución de Quemes a través de dispositivos como tablets, teléfonos móviles, entre otros, y no solo computadores como ocurre actualmente.

Se abre la posibilidad de incluir un componente de simulación en la plataforma Quemes a partir del proyecto creado por el usuario. De esta forma el usuario tendría una retroalimentación temprana del resultado final de lo que está realizando, no sería necesario esperar hasta la finalización del proyecto para validar si lo que está plasmando funciona o tiene problemas. Considerando el tipo de usuario final, sería un mecanismo de generación de éxito de forma temprana.

Realizar un proceso de validación más riguroso con una muestra más representativa de usuarios, considerando diferentes características demográficas, esto con el fin de obtener un resultado más contundente y dicente de las mejoras obtenidas a través de la validación con el usuario final.

Continuar con el proceso de implementación del modelo propuesto en el presente trabajo. En el prototipo implementado se consideraron los componentes más importantes y representativos de acuerdo a la arquitectura y diseño de interfaz propuesto. Sin embargo, se requiere implementar el resto de componentes propuestos en el diseño de la arquitectura, así como las funcionalidades definidas en el documento de especificación de requerimientos y en el diseño de la interfaz de usuario.

13. REFERENCIAS

- [1] E. González Guerrero, J. Páez Rodríguez, and F. J. Roldan, "Uso de robots cooperativos para el desarrollo de habilidades de trabajo cooperativo en niños," *Rev. Investig. UNAD*, vol. 12, no. 2, pp. 43–56, 2013.
- [2] E. G. Guerrero, J. J. P. Rodríguez, and F. J. Roldán, "Robots cooperativos, Quemes para la educación," *Vínculos*, vol. 10, no. 2, pp. 47–62, 2013.
- [3] V. Nettet and A. Large, "Children in the information technology design process: A review of theories and their applications," *Libr. Inf. Sci. Res.*, vol. 26, no. 2, pp. 140–161, 2004.
- [4] E. Folmer, J. van Gorp, and J. Bosch, "Software Architecture Analysis of Usability," in *Engineering Human Computer Interaction and Interactive Systems*, 2004, pp. 38–58.
- [5] E. González, J. Avila, and C. Bustacara, "BESA : Behavior-oriented , Event-driven and Social-based Agent Framework," in *PDPTA'03. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, 2003, pp. 1033–1039.
- [6] M. Richards, *Software Architecture Patterns: Understanding Common Architecture Patterns and When to Use Them*, 1st ed. Sebastopol, CA: O'REILLY, 2015.
- [7] T. Lowdermilk, *User-Centered Design A Developer's Guide to Building User-Friendly Applications*, 1st ed. Sebastopol, CA: O'REILLY, 2013.
- [8] D. Wallach and S. C. Scholz, "User-Centered Design: Why and How to Put Users First in Software Development," *Springer*, vol. 6, pp. 11–39, 2012.
- [9] ISO/IEC, *INTERNATIONAL STANDARD ISO/IEC FDIS 9126*, vol. 1. 2005, p. 42.
- [10] W. G. Wood, "A Practical Example of Applying Attribute-Driven Design (ADD)," Software Engineering Institute, Hanscom AFB, MA, 2007.
- [11] R. Wojcik *et al.*, "Attribute-Driven Design (ADD)," Software Engineering Institute, Hanscom AFB, MA, 2006.
- [12] N. Ali and C. Solis, "Exploring How the Attribute Driven Design Method Is Perceived," in *Relating System Quality and Software Architecture*, 2014, pp. 23–40.
- [13] A. Delgado, A. Castro, and M. Germán, "Evaluación de Arquitecturas de Software con ATAM (Architecture Tradeoff Analysis Method): un caso de estudio," in *VI Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento-JIISIC'07*, 2007, pp. 151–159.
- [14] S. Angelov, J. Trienekens, and P. Grefen, "Extending and Adapting the Architecture Tradeoff Analysis Method for the Evaluation of Software Reference Architectures," *Beta Res. Sch. Oper. Manag. Logist.*, vol. 443, no. January, p. 30, 2014.

- [15] A. Alsumait and A. Al-Osaimi, "Usability heuristics evaluation for child e-learning applications," *J. Softw.*, vol. 5, no. 6, pp. 654–661, 2010.
- [16] B. Shneiderman, C. Plaisant, M. Cohen, and S. Jacobs, *Designing the User Interface*, 5th ed. Pearson, 2009.
- [17] R. S. Sangwan, *Software and Systems Architecture in Action*. Taylor & Francis Group, 2015.
- [18] P. Kruntchen, "Architectural blueprints—the" 4+ 1" view model of software architecture," *IEEE Softw.*, vol. 12, no. November, pp. 42–50, 1995.
- [19] H. Desurvire, M. Caplan, and J. Toth, "Using heuristics to evaluate the playability of games," in *CHI'04 extended abstracts on Human factors in computing systems*, 2004, pp. 1509–1512.
- [20] F. K. Mazumder and U. K. Das, "Usability Guidelines for usable user interface," in *IJRET: International Journal of Research in Engineering and Technology*, 2014, vol. 3, no. 9, pp. 2319–2322.
- [21] C. Rusu and V. Rusu, "Human-Computer Interaction from theory to practice," in *18th International Conference on Human-Computer Interaction, HCII*, 2016, no. 2241.
- [22] J. Preece, H. Sharp, and Y. Rogers, *Interaction Design: Beyond Human-Computer Interaction*, 3rd ed. John Wiley & Sons, 2011.
- [23] T. T. Hewett *et al.*, *ACM SIGCHI Curricula for Human-Computer Interaction*, 1 Ed. Baltimore, MD: Curriculum Development Group, 1992.
- [24] J. Garland and R. Anthony, *Large-Scale Software Architecture*, 1 Ed. John Wiley & Sons, 2003.
- [25] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 3rd ed. Addison-Wesley, 2012.
- [26] L. Hohmann, *Beyond Software Architecture: Creating and Sustaining Winning Solutions*, 1 Ed. Boston, MA: Addison Wesley, 2003.
- [27] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, "Pattern-Oriented Software Architecture Volume 1: A System of Patterns," *Wiley*, vol. Vol. 1, p. 476, 1996.
- [28] D. Garlan and M. Shaw, "An Introduction to Software Architecture," in *Advances in Software Engineering and Knowledge Engineering*, no. January, 1992, pp. 1–40.
- [29] M. Barbacci, M. H. Klein, T. A. Longstaff, and C. B. Weinstock, "Quality Attributes," Pittsburgh, Pennsylvania, 1995.
- [30] F. Losavio, L. Chirinos, N. Lévy, and A. Ramdane-Cherif, "Quality Characteristics for Software Architecture," in *Journal of Object Technology*, 2003, vol. 2, no. 2, pp. 133–150.

- [31] E. Folmer, *Software Architecture Analysis of Usability*, 1 Ed. PrintPartners Ipskamp, 2005.
- [32] D. Milicic, "Software Quality Models and Philosophies," *Softw. Qual. Attrib. Trade-Offs*, vol. 1, p. 17, 2005.
- [33] J. C. Olabe, M. A. Olabe, X. Basogain, I. Maiz, and C. Castaño, "Programming and Robotics with Scratch in Primary Education," in *Education in a technological world: communicating current and emerging research and technological efforts*, 2011, pp. 356–363.
- [34] G. Zaharija, S. Mladenović, and I. Boljat, "Introducing basic Programming Concepts to Elementary School Children," *4th Int. Conf. New Horizons Educ.*, vol. 106, pp. 1576–1584, 2013.
- [35] K. Ryokai and M. Lee, "Children's storytelling and programming with robotic characters," *C&C '09 Proc. seventh ACM Conf. Creat. Cogn.*, pp. 19–28, 2009.
- [36] M. U. Bers, "The TangibleK robotics program: Applied computational thinking for young children," *Early Child. Res. Pract.*, vol. 12, no. 2, pp. 1–20, 2010.
- [37] R. Goldman, "Using educational robotics to engage inner-city students with technology," *ICLS '04 Proc. 6th Int. Conf. Learn. Sci.*, vol. 2003, no. I, pp. 214–221, 2004.
- [38] E. B. B. Gyebi, M. Hanheide, and G. Cielniak, "Educational Robotics for Teaching Computer Science in Africa - Pilot Study," *Int. Work. Educ. Robot.*, 2015.
- [39] M. F. M. Ribeiro, C.R., Coutinho, C.P., Costa, "Robotics in Child Storytelling," *6th Int. Conf. Hands-on Sci.*, vol. 9, no. 2008, pp. 198–205, 2009.
- [40] M. Elkin, A. Sullivan, and M. U. Bers, "Implementing a Robotics Curriculum in an Early Childhood Montessori Classroom," *J. Inf. Technol. Educ. Innov. Pract.*, vol. 13, pp. 153–169, 2014.
- [41] M. F. M. Costa, C. Ribeiro, C. Coutinho, and M. Rocha, "A Study of educational robotics in elementary schools," *Sel. Pap. Hands-on Sci.*, vol. 1, pp. 580–595, 2008.
- [42] A. Wilson and D. C. Moffat, "Evaluating Scratch to introduce younger schoolchildren to programming," *Proc. 22nd Annu. Work. Psychol. Program. Interes. Gr.*, pp. 64–75, 2010.
- [43] D. Pinelle, N. Wong, T. Stach, and C. Gutwin, "Usability Heuristics for Networked Multiplayer Games," *Gr. '09 Proc. ACM 2009 Int. Conf. Support. Gr. Work*, pp. 169–178, 2009.
- [44] H. Korhonen and E. M. I. Koivisto, "Playability heuristics for mobile multi-player games," *2nd Int. Conf. Digit. Interact. media Entertain. arts*, pp. 28–35, 2007.
- [45] M. Federoff, "Heuristics and usability guidelines for the creation and evaluation of fun in video games," Indiana University, 2002.
- [46] D. Pinelle, N. Wong, and T. Stach, "Heuristic evaluation for games: usability principles for

- video game design,” *SIGCHI Conf. Hum. Factors Comput. Syst.*, pp. 1453–1462, 2008.
- [47] T. C. Reeves, K. Beaumie, K. Hyeonjin, and C. S. Loh, “Usability and Instructional Design Heuristics for E-Learning Evaluation,” *World Conf. Educ. Multimedia, Hypermedia Telecommun.*, pp. 1615–1621, 2002.
- [48] S. Chiasson and C. Gutwin, “Design Principles for Children’s Technology,” *Tech. Rep. HCI-TR-05-02*, p. 9, 2005.
- [49] L. Kuparinen, J. Silvennoinen, and H. Isomäki, “Introducing Usability Heuristics for Mobile Map Applications,” *26th Int. Cartogr. Conf. (ICC 2013)*, 2013.
- [50] R. M. Miranda, “Analysis of the Usability of Mobile device Applications based upon Heuristics,” *Universitat Paderborn*, 2014.
- [51] W. A. J. Wan Yahya and S. N. Abdul Salam, “Usability design strategies for children : Developing children learning and knowledge in decreasing children dental anxiety,” *Int. Conf. Prim. Educ. 2009*, pp. 1–15, 2009.
- [52] P. Barr, “User-Interface Metaphors in Theory and Practice,” *Victoria University*, 2003.
- [53] M. del C. Suárez Torrente, “Sistema de Evaluación de la Usabilidad Web Orientado al Usuario y basado en la Determinación de Tareas Críticas,” *Universidad de Oviedo*, 2011.
- [54] C. Tet Kun and N. Sahari, “Utilitarian or Experiential? An Analysis of Usability Questionnaires,” *Int. J. Comput. Theory Eng.*, vol. 7, no. 2, pp. 167–171, 2015.
- [55] D. George and P. Mallery, *SPSS for Windows step by step: A simple guide and reference. 11.0 update*, 4 ed. Boston, MA: Allyn & Bacon, 2003.
- [56] J. Brooke, “SUS : A Retrospective,” *J. Usability Stud.*, vol. 8, no. 2, pp. 29–40, 2013.
- [57] S. Angulo and D. Cebri, “Usabilidad y Satisfacción de la e-Rúbrica,” *REDU. Rev. Docencia Univ.*, vol. 12, no. 1, pp. 177–195, 2014.
- [58] A. Bangor, P. Kortum, and J. Miller, “Determining what individual SUS scores mean: Adding an adjective rating scale,” *J. usability Stud.*, vol. 4, no. 3, pp. 114–123, 2009.