

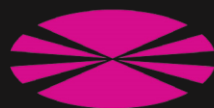
Tesis doctoral

**Modelo de evaluación de
metodologías de desarrollo de
software web.**



Jimmy Molina Ríos

A Coruña 2021



UNIVERSIDADE DA CORUÑA

Modelo de evaluación de metodologías de desarrollo de software web.

Autor/ a: Jimmy Molina Ríos

Tesis doctoral UDC / 2020

Directora y tutora de la tesis:

Nieves Pedreira Souto

Departamento de Computación e Tecnoloxías da Información

Universidade da Coruña

Codirectora de la tesis:

Anna De Liddo

KMi Knowledge Management Institute

Open University

Programa de doctorado en Tecnologías de la Información y las Comunicaciones.



Agradecimientos

Me gustaría empezar expresando mis más sinceros agradecimientos a mis directoras de tesis Nieves Pedreira Souto y Anna De Liddo, por su constante apoyo a lo largo de la realización de la presente tesis.

A todas las personas y compañeros que me brindaron su colaboración rápida y desinteresada durante las actividades realizadas durante todo el proceso. De igual manera, a las empresas y desarrolladores de software que participaron dentro de la investigación de campo y realización de encuesta, sin las cuales no hubiera sido posible la culminación de la investigación.

Finalmente, agradecer a Dios y a mis padres que siempre se han esforzado y apoyado a lo largo de mi vida profesional, y son quienes han hecho posible el haber llegado hasta aquí.

Resumo

O desenvolvemento de aplicacións web aumenta exponencialmente xunto coa dependencia dos usuarios en internet, de xeito que as técnicas e metodoloxías aplicadas tradicionalmente para o seu desenvolvemento evolucionaron cara a metodoloxías áxiles e híbridas, que proporcionan un proceso de desenvolvemento óptimo, incluída a avaliación final do software baseada en métricas e regulamentos de calidade que indican o proceso máis axeitado segundo os requisitos establecidos no proxecto. O obxectivo principal da nosa investigación reside no establecemento dun modelo de avaliación para a avaliación da eficiencia dos proxectos baseado nas fases e características proporcionadas pola metodoloxía híbrida SWIRL para a avaliación de proxectos a través dunha aplicación web. Levamos a cabo unha profusa revisión analítica bibliográfica sobre o estado da arte e unha investigación de campo que levou á definición do modelo. Como resultado, propónse un novo modelo de avaliación chamado "MEASS" (Application Evaluation Model Selon SWIRL), presentado a través dunha aplicación web desenvolvida baixo as fases SWIRL, que permite amosar os resultados porcentuais das fases. O modelo divídese en tres niveis de avaliación: básico, intermedio e rigoroso, cuxo obxectivo é avaliar o proceso que se seguiu para o desenvolvemento do software e ofrece indicadores específicos segundo cada nivel e, ademáis, as subcaracterísticas das fases.

Resumen

El desarrollo de aplicaciones web se incrementa exponencialmente conjuntamente con la dependencia de los usuarios hacia el internet, de manera que las técnicas y metodologías aplicadas tradicionalmente para el desarrollo de las mismas han evolucionado en metodologías ágiles e híbridas, las cuales brindan un proceso de desarrollo óptimo, incluyendo la evaluación final del software en base a métricas de calidad y normativas que indican el proceso más adecuado según los requerimientos establecidos en el proyecto. El objetivo principal de nuestra investigación recae en el establecimiento de un modelo de evaluación, para la valoración de la eficiencia de proyectos basado en las fases y características que proporciona la metodología híbrida SWIRL a través de una aplicación web. Realizamos tanto una profunda revisión analítica bibliográfica del estado del arte como una investigación de campo que condujo a la definición del modelo. Como resultado, se propone un nuevo modelo de evaluación denominado "MEASS" (Modelo de Evaluación de Aplicaciones Según SWIRL), presentado a través de una aplicación web desarrollada bajo las fases SWIRL, que permite mostrar los resultados porcentuales de las fases. El modelo se divide en tres niveles de valoración: básico, intermedio y riguroso, que tienen como objetivo evaluar el proceso que se ha seguido para el desarrollo del software, y proporciona indicadores específicos acorde a cada nivel y, además, las subcaracterísticas de las fases.

Abstract

The development of web applications increases exponentially together with the users' dependence on the internet, so that the techniques and methodologies traditionally applied to their development have evolved in agile and hybrid methodologies, which provide an optimal development process, including the final evaluation of the software based on quality metrics and regulations that indicate the most appropriate process according to the requirements established in the project. The main target of our research is to establish an evaluation model for the assessment of the efficiency of projects based on the phases and characteristics provided by the hybrid SWIRL methodology for the assessment of projects through a web application. We conducted both deep bibliographic analysis of the state of the art and field research which lead to the definition of the model. As a result, we propose a new evaluation model called "MEASS" (Application Evaluation Model According to SWIRL), presented through a web application developed under SWIRL phases, which enables to display percentage results of the phases. The model is divided into three levels of assessment: basic, intermediate and rigorous, which aim to evaluate the process that has been followed for the development of the software, and provides specific indicators according to each level, and sub-characteristics of the phases.

Índice de contenidos

Índice de contenidos	xi
Índice de tablas	xv
Índice de figuras	xvii
Índice de gráficos	xviii
Lista de abreviaturas	xix
1. INTRODUCCIÓN	1
1.1. Antecedentes.	1
1.2. Problemática.	2
1.3. Importancia del uso de una metodología de desarrollo.	3
1.4. Necesidad de una metodología adecuada.	4
2. ESTADO DE LA CUESTIÓN	7
2.1. Ingeniería web.....	7
2.2. Desarrollo de software web	8
2.2.1. Metodologías Ágiles.....	8
2.2.2. Metodologías Híbridas.....	8
2.3. Metodologías de desarrollo de aplicaciones web	9
2.3.1. Comparación de metodologías	10
2.3.2. Comparación por criterios de calidad según Norma ISO 25000	13
2.4. Carencias en las metodologías de desarrollo de aplicaciones web.	15
3. HIPÓTESIS DE TRABAJO Y OBJETIVOS	17
3.1. Hipótesis de trabajo.	17
3.2. Tesis.	17
3.2.1. Objetivo general.	17
3.2.2. Objetivos específicos.....	18
3.2.3. Preguntas de investigación.	18
3.3. Justificación.....	19
4. PROPUESTA DE UNA METODOLOGÍA DE DESARROLLO HÍBRIDA: SWIRL ...	
.....	21
4.1. Modelo y ciclo de vida.	21
4.2. Fases de la metodología.	22
4.2.1. Fase I: Análisis.	22

4.2.2.	Fase II: Planificación.....	24
4.2.3.	Fase III: Modelado.	25
4.2.4.	Fase IV: Implementación.	26
4.2.5.	Fase V: Revisión y pruebas.....	27
4.2.6.	Fase VI: Lanzamiento y marketing.	28
4.3.	Características de la metodología.	29
4.4.	Criterios de evaluación para SWIRL	30
5.	MODELO DE EVALUACIÓN DE METODOLOGÍAS	31
5.1.	Metodología de desarrollo del modelo de evaluación.....	31
5.1.1.	Uso del modelo de evaluación.....	36
5.2.	Consideraciones a tener en cuenta durante el desarrollo.	36
5.3.	Diseño del modelo de evaluación.....	40
6.	APLICACIÓN WEB PARA LA EVALUACIÓN DE PROYECTOS ORIENTADOS A LA WEB.....	55
6.1.	Metodología de desarrollo de la aplicación.....	55
6.2.	Fase 1: Análisis del sistema.	55
6.2.1.	Iteración 1: Definición del sistema	55
6.2.2.	Iteración 2: Modelado de negocio.	61
6.2.3.	Iteración 3: Identificación de interesados.....	63
6.2.4.	Iteración 4: Requerimientos.....	65
6.3.	Fase 2: Planificación.....	68
6.3.1.	Iteración 1: Historias de usuario.	68
6.4.	Fase 3: Modelado.....	69
6.4.1.	Iteración 1: Modelo Conceptual.	69
6.4.2.	Iteración 2: Modelo de navegación.	70
6.4.3.	Iteración 3: Arquitectura del sistema.....	71
6.4.4.	Iteración 4: Diseño de interfaz abstracta de usuario.	72
6.4.5.	Iteración 5: Diagrama de Gantt.....	75
6.4.6.	Iteración 6: Gestión de riesgos.	76
6.4.7.	Iteración 7: Gestión de comunicaciones.	77
6.4.8.	Iteración 8: Estructura de desglose de trabajo (EDT/WBS)	77
6.4.9.	Iteración 9: Gestión de calidad.	78
6.5.	Fase 4: Implementación.....	78
6.6.	Fase 5: Pruebas y revisión.	79

7. RESULTADOS.....	87
7.1. Modelo de Evaluación.....	87
7.1.1. Modelo de Evaluación 1 (Básico).....	87
7.1.2. Modelo de Evaluación 2 (Intermedio).....	89
7.1.3. Modelo de Evaluación 3 (Riguroso).....	92
7.2. Aplicación web en ejecución.....	96
7.3. Evaluación de proyectos mediante los modelos.....	99
8. DISCUSIÓN Y CONCLUSIONES.....	109
8.1. Discusión de los resultados.....	109
8.2. Conclusiones.....	111
9. FUTUROS DESARROLLOS.....	113
REFERENCIAS BIBLIOGRÁFICAS.....	115

Índice de tablas

Tabla 1 Artículos resultantes de la investigación de metodologías web.....	9
Tabla 2 Comparativa según la norma ISO 25000.....	14
Tabla 3 Comparativa de las características comunes en las metodologías de desarrollo	16
Tabla 4 Comparativa de SWIRL según la norma ISO 25000	32
Tabla 5 Indicadores del modelo de evaluación por fases	33
Tabla 6 Patrones de problemas en el desarrollo.	39
Tabla 7 Impacto de las etapas de la metodología en el proceso de desarrollo.....	41
Tabla 8 Indicadores por subcaracterísticas de la fase 1	43
Tabla 9 Indicadores por subcaracterísticas de la fase 2	44
Tabla 10 Indicadores por subcaracterísticas de la fase 3	45
Tabla 11 Indicadores por subcaracterísticas de la fase 4	46
Tabla 12 Indicadores por subcaracterísticas de la fase 5	47
Tabla 13 Indicadores por subcaracterísticas de la fase 6	48
Tabla 14 Evaluadores del modelo.	49
Tabla 15 Validación de modelos por fases.	50
Tabla 16 Valoración en validación del modelo.	51
Tabla 17 Resultados de la encuesta de nivel de utilidad del modelo	52
Tabla 18 Plantilla de usuarios: usuario común.....	56
Tabla 19 Plantilla de usuarios: usuario registrado	56
Tabla 20 Factibilidad técnica del proyecto	58
Tabla 21 Plantilla de riesgos: riesgo de interfaz	60
Tabla 22 Plantilla de riesgos: riesgo de valoración en resultados	60
Tabla 23 Plantilla de riesgos: asignación errónea de indicadores	60
Tabla 24 Plantilla de riesgos: riesgo de error de conexión con base de datos.....	61
Tabla 25 Análisis cuantitativo y cualitativo de riesgos	61
Tabla 26 Matriz de reglas de negocio	63
Tabla 27 Matriz general de interesados: fase de análisis	63
Tabla 28 Matriz general de interesados: fase de planificación.....	64
Tabla 29 Matriz general de interesados: fase de modelado	64
Tabla 30 Matriz general de interesados: fase de implementación.....	64
Tabla 31 Matriz general de interesados: fase de revisión y pruebas	64

Tabla 32 Matriz general de interesados: fase de revisión y pruebas	64
Tabla 33 Requerimientos funcionales de la aplicación.....	65
Tabla 34 Requerimientos no funcionales de la aplicación.....	66
Tabla 35 Requerimientos de usuario de la aplicación.....	66
Tabla 36 Matriz de priorización de requerimientos.....	67
Tabla 37 Historia de usuario: 1	68
Tabla 38 Historia de usuario: 2	68
Tabla 39 Historia de usuario: 3	69
Tabla 40 Historia de usuario: 4	69
Tabla 41 Plan de prevención de riesgos	76
Tabla 42 Plan de minimización de riesgos.....	77
Tabla 43 Revisión de pares: Página de inicio.....	80
Tabla 44 Revisión de pares: Registro e ingreso al sistema.....	81
Tabla 45 Revisión de pares: Perfil de usuario.....	81
Tabla 46 Revisión de pares: Creación y modificación de proyectos.....	82
Tabla 47 Revisión de pares: Creación y modificación de proyectos.....	83
Tabla 48 Prueba de Aceptación de la aplicación.....	84
Tabla 49 Valoración en prueba de aceptación.....	85
Tabla 50 Modelo de evaluación básico: fase 1	87
Tabla 51 Modelo de evaluación básico: fase 2	87
Tabla 52 Modelo de evaluación básico: fase 3	88
Tabla 53 Modelo de evaluación básico: fase 4	88
Tabla 54 Modelo de evaluación básico: fase 5	88
Tabla 55 Modelo de evaluación básico: fase 6	89
Tabla 56 Modelo de evaluación intermedio: fase 1	89
Tabla 57 Modelo de evaluación intermedio: fase 2	89
Tabla 58 Modelo de evaluación intermedio: fase 3	90
Tabla 59 Modelo de evaluación intermedio: fase 4	90
Tabla 60 Modelo de evaluación intermedio: fase 5	91
Tabla 61 Modelo de evaluación intermedio: fase 6	91
Tabla 62 Modelo de evaluación riguroso: fase 1	92
Tabla 63 Modelo de evaluación riguroso: fase 2	92
Tabla 64 Modelo de evaluación riguroso: fase 3	93
Tabla 65 Modelo de evaluación riguroso: fase 4	94

Tabla 66 Modelo de evaluación riguroso: fase 5	95
Tabla 67 Modelo de evaluación riguroso: fase 6	95
Tabla 68 Aplicaciones Web evaluadas.....	101
Tabla 69 Tabla de resultados de evaluación de aplicaciones web	102
Tabla 70 Valoración sobre porcentaje de cumplimiento	108

Índice de figuras

Figura 1 Página principal de la aplicación.	72
Figura 2 Página de registro de usuario	73
Figura 3 Página de inicio de sesión	73
Figura 4 Página de evaluación para usuarios registrados	74
Figura 5 Páginas de creación y edición de evaluación.....	74
Figura 6 Página de evaluación de procesos y subprocesos.	75
Figura 7 Página de evaluación de procesos y subprocesos (continuación).	75
Figura 8 Aplicación en ejecución: página principal.	96
Figura 9 Aplicación en ejecución: manual de usuario.....	97
Figura 10 Aplicación en ejecución: video informativo.	97
Figura 11 Aplicación en ejecución: registro e inicio de sesión.	98
Figura 12 Aplicación en ejecución: visualización del perfil.	98
Figura 13 Aplicación en ejecución: creación de proyectos.	98
Figura 14 Aplicación en ejecución: evaluación de proyectos.	99
Figura 15 Aplicación en ejecución: visualización de resultados.....	99
Figura 16 Resultado de la evaluación por fases.	103
Figura 17 Resultado de la evaluación en la fase de análisis.....	104
Figura 18 Resultado de la evaluación en la fase de planificación.	105
Figura 19 Resultado de la evaluación en la fase de modelado.....	105
Figura 20 Resultado de la evaluación en la fase de implementación.	106
Figura 21 Resultado de la evaluación en la fase de pruebas y revisión.....	107
Figura 22 Resultado de la evaluación en la fase de lanzamiento y marketing.	107

Índice de gráficos

Gráfico 1 Proceso de desarrollo de la metodología SWIRL.....	22
Gráfico 2 Modelo de negocio	62
Gráfico 3 Modelo conceptual.....	70
Gráfico 4 Modelo de navegación.	70
Gráfico 5 Arquitectura del sistema.	71
Gráfico 6 Planificación en diagrama de Gantt	76
Gráfico 7 EDT/WDS	77

Lista de abreviaturas

E-R	Entidad Relación
UML	Lenguaje de modelado unificado
WSDM	Web Semantic Design Method
SOHDM	Scenario-Based Object-Oriented Hypermedia Design Methodology
RMM	Relationship Managment Methodology
MEASS	Modelo de Evaluación de Aplicaciones Según SWIRL
OOHDM	Object-Oriented Hypermedia Design Model
HDM	Hypermedia Design Model
SWIRL	Software Web Iterativo Relacional Lógico
MDA	Object-Oriented Hypermedia Design Model
RG	Riesgo
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
SPIDR	Spikes, Path, Interfaces, Data, Rules

1. INTRODUCCIÓN

1.1. Antecedentes.

Desde sus inicios, el desarrollo de software ha sido establecido bajo el control de parámetros de revisión estrictos y metodológicos con la finalidad de la producción de programas de calidad, que logren satisfacer las necesidades de los usuarios finales y principalmente que resuelvan la problemática por la cuales fueron desarrollados. Para ello, se emplean metodologías de desarrollo, las cuales consisten en una serie de pasos metodológicos que guían al desarrollador a lo largo de todo el proceso, definiendo las actividades, herramientas e hitos que es necesario alcanzar para lograr una aplicación de calidad.

Sin embargo, con el paso del tiempo, y con la constante evolución de las TIC (tecnologías de la información y las comunicaciones), las aplicaciones web desarrolladas a base de metodologías tradicionales [1] se han encontrado con inconvenientes dentro del proceso de desarrollo. Los principales son los problemas de requerimientos dinámicos, sometidos a constantes cambios, y que deben ser integrados en las diferentes fases del ciclo de vida del software desarrollado. El crecimiento global de las empresas y su vinculación directa con las actuales tendencias tecnológicas, ha supuesto un reto insalvable a las metodologías tradicionales, logrando que no satisfagan las necesidades del usuario durante el proceso de desarrollo, debido a las características complejas que implica toda aplicación web que presente una mínima complejidad y calidad.

Como consecuencia de lo anteriormente planteado, se han desarrollado nuevas metodologías de desarrollo, metodologías ágiles [2], las cuales se enfocan específicamente en el ciclo de vida de las aplicaciones web, incorporando todas las características que conciernen a este tipo de aplicaciones, además de ser creadas con la finalidad de resolver los problemas habituales que los desarrolladores presentan durante el proceso de desarrollo, como son, sobre todo: constantes cambios de requerimientos por parte del cliente, carencia de seguridad y problemas de diseño e implementación de las componentes del sistema.

Desarrolladores independientes y empresas consagradas en el ámbito de desarrollo de software dedican habitualmente gran parte de su tiempo a la selección de la metodología más adecuada y acorde a los requerimientos que se plantean

inicialmente en el proyecto. Sin embargo, existen una diversidad de circunstancias adversas que emergen de forma inesperada en cualquiera de las fases de desarrollo, dificultando de esta manera definir la metodología óptima a base de indicadores de calidad en el proyecto.

Entre las técnicas para evaluar la metodología de desarrollo a utilizar, y conocer si es la más adecuada según el proyecto, se usan estrategias de evaluación de programas las cuales brindan una gran cantidad de beneficios para mejorar la competitividad que tiene la empresa desarrolladora dentro del mercado, desempeñándose de mejor manera en las áreas del proyecto. Las estrategias se encuentran establecidas por un conjunto de planificaciones con la finalidad de respaldar de manera adecuada las decisiones que toma la empresa durante el proceso de selección de la metodología, y estas se seleccionan de acuerdo a las características del sistema a desarrollarse [3].

1.2. Problemática.

Antes de iniciar el proceso de desarrollo de un programa basado en la web, tanto las empresas desarrolladoras como los programadores independientes, deben seleccionar la metodología más adecuada para el desarrollo de la aplicación, considerando características primordiales de aplicaciones basadas en la web, y los requerimientos que se establecen por parte del cliente en las etapas iniciales del proyecto, los cuales, de acuerdo con Agrawal [4], son uno de los factores más recurrentes en el fracaso de diversos proyectos de desarrollo de programas.

Sin embargo, este proceso tiende a fallar, o en ocasiones a presentar algunos problemas durante el ciclo de vida, debido a que no se consideran todas las características posibles [5], y los problemas que se pueden presentar; entre los que se encuentran los fallos de seguridad e integración de la información que las aplicaciones deben manipular; las características que los usuarios deben incorporar en etapas avanzadas; y, principalmente, la navegabilidad y verificación de las mismas dentro de la aplicación.

En consecuencia, a las adversidades que se presentan en el proceso de selección y uso de las metodologías por parte de los desarrolladores, se ha planteado como finalidad identificar las metodologías de desarrollo web que apliquen de manera efectiva para ambientes de desarrollo dinámico. Para ello, es requerida la

comparación entre las metodologías de desarrollo comúnmente usadas, además del uso de estrategias de evaluación que implican técnicas y características comunes de las metodologías, considerando primordialmente las métricas de calidad y las etapas de desarrollo dentro del ciclo de vida del programa.

1.3. Importancia del uso de una metodología de desarrollo.

Emplear una metodología en el momento de desarrollar un programa provee al desarrollador la capacidad de estructurar, planear y, sobre todo, gestionar el control de los procesos concernientes en el ciclo de vida del proyecto, de manera que las actividades cumplan un propósito para ayudar al desarrollo final del mismo.

En la actualidad, se han presentado una amplia gama de metodologías ágiles, las cuales se acoplan a las diversas condiciones que imponen los proyectos basados en la web; por ejemplo, McCaffery [6] señala que el enfoque primordial de la metodología XP se basa en el aspecto técnico del desarrollo, mientras que, por otro lado, la metodología SCRUM se enfoca más en la inclusión del usuario durante el ciclo de vida completo. De esta manera, se logra interpretar que cada una de las metodologías ofrecidas para el desarrollo de programas proporcionan enfoques diferentes y herramientas que cubren características específicas del programa para las cuales fueron diseñadas, entre estas se puede mencionar: la definición de requerimientos, las técnicas de análisis de requisitos, el diseño de interfaces y la implementación [7].

Los desarrolladores independientes, tienden a ejecutar proyectos de aplicaciones web sin las debidas directrices; es decir, no optan por el uso de una metodología para guiar el proceso de desarrollo, sino que emplean su experiencia en proyectos pasados para la realización de los nuevos. De esta manera, se desencadena un caos total en la gestión, tanto técnica como administrativa del proyecto; sin mencionar los constantes fallos de funcionamiento y de calidad que el producto puede llegar a presentar al ser finalizado o, incluso posteriormente durante su transferencia y mantenimiento.

Otros problemas que pueden llegar a presentarse por no emplear una adecuada estructura metodológica para el desarrollo de aplicaciones web, son: la insatisfacción del cliente con los resultados obtenidos [8], la disconformidad entre los integrantes del grupo de trabajo y, en los casos más graves, los fallos internos por omisión o inadecuada implementación y documentación de la aplicación [9].

Es por ello, que se asevera que el aporte que brindan las metodologías de desarrollo es radical en cuanto a las herramientas aplicadas específica y respectivamente a cada fase del ciclo de vida: estructuración, planeación y control del esquema de trabajo; además de los resultados y satisfacciones del cliente con la aplicación recibida en cuanto a métricas de calidad y funcionalidad.

1.4. Necesidad de una metodología adecuada.

La construcción de una aplicación web con resultados eficientes y que satisfagan las necesidades del cliente, es un gran desafío, tanto para las empresas desarrolladoras como para los programadores independientes. En ocasiones, se busca simplemente alcanzar los objetivos que se plantearon al iniciar el proyecto, sin considerar suficientemente, la planificación o características que el proyecto presente como resultado final. Es por ello, que se aplican las metodologías de desarrollo, debido a que su uso ayuda a simplificar y clarificar todos los procesos que deben ser llevados a cabo para obtener un producto con altos índices de calidad.

Según los trabajos de Bakhtouchi y Rahmouni [10], la selección de la metodología adecuada es un punto clave para el desarrollo del software, para lo cual se deben realizar estudios pertinentes sobre los patrocinadores, clientes/usuarios, requerimientos, estimación de esfuerzo, y el equipo de desarrollo con el cual se va a trabajar [11].

El beneficio del uso de una metodología idónea para desarrollar una aplicación web se correlaciona con aspectos tales como:

- Calidad del proyecto [12].
- Satisfacción del cliente [13].
- Reducción de riesgos [14].
- Entregables en corto tiempo [15].
- Eliminación de procesos redundantes [16].

En otras palabras, gestionar de manera correcta la selección de la metodología a emplear contribuye en el incremento de la calidad del proyecto, la satisfacción del cliente, la prevención de fallos o riesgos durante la gestión del proyecto, entre otros, beneficiando considerablemente el proceso que lo aplica. Sin embargo, al seleccionar

Capítulo 1. Introducción

erróneamente la metodología, se pueden llegar a presentar problemas relacionados con los cambios de requerimientos, planificación, y principalmente con la gestión del proyecto.

2. ESTADO DE LA CUESTIÓN

2.1. Ingeniería web.

La ingeniería de software se encuentra derivada en diversas ramas, que involucran el desarrollo y creación de un sistema de información, entre las cuales ha destacado últimamente la ingeniería web. Esta derivación de la ingeniería de software consiste en la unión de procesos respectivos para la planificación, diseño y desarrollo de proyectos basados en la web.

Es importante recalcar las características que diferencian a una aplicación web de un programa tradicional o de escritorio, para ello Torrecilla, Sedeño, Escalona & Mejía [17] en sus trabajos estipulan las siguientes:

- La navegación a través del sistema es más compleja, debido a que se posee una mayor cantidad de requerimientos. Para ello se crea un modelo de navegación, el cual permite a los desarrolladores identificar claramente el camino entre páginas que el usuario debe recorrer.
- El diseño de interfaz de usuario se encuentra estandarizada, debido a que el uso de los mismos es para usuarios no definidos.
- La seguridad es un aspecto primordial, esto debido a que las aplicaciones gestionan datos que se encuentran circulando por la red, y la degradación de la información puede llegar a ocurrir, así como la pérdida de la misma.
- Un aspecto básico de las aplicaciones web es la rapidez de respuesta que las funcionalidades deben presentar al usuario.

Las propuestas que se han presentado sobre ingeniería web, se encuentran basadas en modelos que abordan desde el diseño de las interfaces hasta el desarrollo de las aplicaciones, utilizando modelos específicos para cada uno de ellos [18], siendo los más básicos: el modelo conceptual, el modelo de navegación, y el modelo de presentación. Esta definición es compartida por Domínguez et al [19], quienes en sus trabajos establecen que la ingeniería web es un paradigma basada en modelos, la cual es construida en colaboración con desarrolladores y usuarios finales, de manera que se obtenga mayor flexibilidad y constante comunicación entre los interesados.

Los enfoques que busca conseguir la ingeniería web al emplear modelos son: "mejorar la capacidad de lidiar con la lógica de datos y negocios, y fortalecer la gestión de presentaciones" [20].

2.2. Desarrollo de software web

Las aplicaciones web se desarrollan siguiendo diversas directrices establecidas en las metodologías de desarrollo de software. A diferencia de aplicaciones móviles o de escritorio, estas se caracterizan por su ejecución dinámica sobre cualquier plataforma [21]. El desarrollo web presenta varios desafíos para los programadores y equipos de desarrollo, debido al flujo de información que debe procesar, los cambios constantes en los requerimientos o especificaciones del programa, y la seguridad, entre otros [22]. Esto ha dado origen a una gran variedad de metodologías que mantienen enfoques distintos sobre el desarrollo web. En la actualidad, las metodologías que son empleadas para el desarrollo de aplicaciones web se clasifican en: Metodologías Ágiles y Metodologías Híbridas.

2.2.1. Metodologías Ágiles

El enfoque ágil se caracteriza por la adaptabilidad a contextos cambiantes a lo largo del desarrollo y gestión del proyecto, la interacción con el cliente es constante y el tiempo de entrega de las soluciones parciales y final (entregables) es más corto [23].

Navarro [24], argumenta que las metodologías ágiles fragmentan el proyecto en partes más pequeñas que son tratadas en un corto lapso de tiempo y son desarrollados de forma individual. Esto fomenta a la participación y aplicación de las buenas prácticas presentes en el manifiesto ágil.

2.2.2. Metodologías Híbridas

Las metodologías híbridas surgen de la combinación de buenas prácticas y, o, enfoques que poseen otras metodologías. Como resultado, se obtiene un esquema completo y adaptable cuyo objetivo es favorecer al desarrollo del programa en un tiempo relativamente más corto [25]. Leiva y Villalobos [26], resaltan que este tipo de metodologías emergen de las debilidades de otros métodos u enfoques planteados en las metodologías ágiles y, o, tradicionales. El propósito de las metodologías híbridas es convertir aquellas debilidades en fortalezas desarrollando una nueva metodología capaz de ajustarse a las exigencias de otros proyectos de desarrollo.

2.3. Metodologías de desarrollo de aplicaciones web

Las metodologías orientadas al desarrollo de aplicaciones web están estructuradas de forma que puedan amoldarse a diferentes circunstancias relacionadas al cambio, como son: los requerimientos o especificaciones, las herramientas, el equipo de trabajo y el periodo de tiempo, entre otros. [27] Este tipo de metodologías hacen énfasis en la agilidad en sus procesos, por la limitación de tiempo que tiene el equipo de desarrollo.

Sin embargo, cada una de las metodologías propone un enfoque más determinado al cumplimiento de un requerimiento específico, obviando algunas necesidades esenciales de los proyectos web; lo cual genera la necesidad de desarrollar una comparativa entre las metodologías más empleadas en el ámbito profesional por las empresas desarrolladoras de software web.

Para ello, en la fase inicial de este trabajo, se han realizado investigaciones profundas sobre los enfoques de las metodologías web que comúnmente emplean los desarrolladores de aplicaciones web, y los puntos débiles o ausentes de las mismas, mediante un estudio comparativo. Estos resultados fueron expuestos y publicados respectivamente en revistas científicas como se detalla en la *Tabla 1*. A continuación, en los siguientes apartados de este capítulo, se expone una síntesis de los mismos. El análisis detallado se puede consultar en los artículos aquí referenciados.

Tabla 1 Artículos resultantes de la investigación de metodologías web.

Título	Revista	Datos relevantes.
Comparison of development methodologies in web applications.	Information and Software Technology	Autores: J. Molina, N. Pedreira. <i>Volume 119, March 2020.</i> JCR Impact Factor 2020: 2,730 Q2 Categoría: COMPUTER SCIENCE, SOFTWARE ENGINEERING Posición: 31/108 DOI: 10.1016/j.infsof.2019.106238
Approach of agile methodologies in the development of web-based software.	Information	Autores: J. Molina, N. Pedreira. <i>2019, Volume 10 (10).</i> SJR 2019: 0,353 Q3 En JCR desde 2020

		Categoría: COMPUTER SCIENCE, INFORMATION SYSTEMS, Q3 DOI: 10.3390/info10100314
--	--	---

2.3.1. Comparación de metodologías

Antes de realizar una comparación entre las metodologías es relevante conocer brevemente sus fases y enfoques para, posteriormente, evaluarlas según los criterios pertinentes.

2.3.1.1. WSDM

Web Site Design Method (WSDM) por sus siglas en inglés, es una metodología orientada al usuario. Tiene por objetivo resolver los problemas que se originan entre los usuarios por su ubicación geográfica. Por ende, el desarrollo de la aplicación web está dado por las especificaciones de grupos de usuarios.[28] Las fases que comprende esta metodología son cuatro:

- **Modelado de usuarios:** Dentro de esta fase se procede a definir y clasificar los usuarios que tendrán acceso al sistema final, identificando sus roles, permisos y funcionalidades dentro del mismo.
- **Diseño conceptual:** Se emplea principalmente el modelado de objetos, para definir formalmente los requerimientos, las funcionalidades, las interacciones a realizar entre usuarios y módulos, entre otros.
- **Diseño de implementación:** Centra su objetivo en el diseño de la navegación de la página (o conocido en esta metodología como diseño estructural), en el diseño de presentación (interfaz), y en el diseño lógico de datos.
- **Implementación:** Se encuentran inmersas la codificación de las funcionalidades, la puesta en marcha y finalmente la ejecución de pruebas.

2.3.1.2. SOHDM

La Metodología de Diseño de Escenarios Orientado a Objetos en Hipermedia está orientada a los escenarios para el desarrollo del programa y hace énfasis en el tratamiento de los requerimientos. Las fases de esta metodología guardan cierta relación con las fases de RMM, OOHDM Y EROM [29]. Las fases de SOHDM son seis:

- **Análisis:** Identificación de las necesidades y requerimientos de la aplicación incluyendo los usuarios, y escenarios representativos de las actividades a efectuar mediante la implementación del sistema.
- **Modelado de Objetos:** Se emplea como herramienta los CRC Cards (Class Responsibility Collaboration) para transformar los escenarios identificados en objetos.
- **Diseño de Vistas:** Cada vista representa la información que es presentada a un grupo de usuarios finales, existen tres tipos de vistas: base, de asociación, y de colaboración.
- **Diseño de navegación:** Define las relaciones y enlaces a seguir para efectuar cada funcionalidad dentro del sistema.
- **Diseño de la Implementación:** Dentro de este se definen los modelos para ser implementados en la siguiente fase, incluye el diseño de la interfaz de usuario, y la base de datos.
- **Construcción:** Se implementa la aplicación mediante la codificación de las funcionalidades y componentes.

2.3.1.3. OOHDM

La metodología OOHDM se caracteriza por emplear el estándar UML y otros elementos dentro del proceso de desarrollo y ciclo de vida del proyecto [30]. Vera [31], argumenta que esta metodología es una sucesora de HDM, la cual está orientada al modelado de sistemas de hipermedia. OOHDM, se compone de cuatro fases:

- **Diseño conceptual:** Se basa en la estructuración de la información con la finalidad de representar el contenido y funcionalidades de la página, sin considerar los usuarios ni sus funciones.
- **Diseño de navegación:** Se definen los enlaces de las navegaciones de las funcionalidades de la página de manera que se explique el proceso que debe seguir el usuario para explorar el contenido del sitio.
- **Diseño de interfaces abstractas:** Se diseñan los componentes a ser percibidos por los usuarios finales, incluyendo las interfaces de usuario, y sus componentes.

- **Implementación:** Codificación e implementación de las funcionalidades, los componentes y requerimientos adicionales que se han planificado para el sistema.

2.3.1.4. IWEB

La metodología IWEB o ingeniería web, se enfoca en el análisis de requerimientos y planificación para el desarrollo de la aplicación web a través de principios científicos y técnicas para la gestión del proyecto [32]. Las etapas de IWEB son seis:

- **Formulación:** Se emplea para la identificación del alcance, objetivos, motivación, y actores necesarios dentro de la aplicación.
- **Planificación:** Se definen los tiempos, costos, y se analizan también los posibles riesgos de ejecución dentro del proyecto.
- **Análisis:** Se analizan los requerimientos de usuario, de diseño y técnicos, mediante la identificación de su funcionalidad, contenido e iteración.
- **Ingeniería:** Se centra en la integración entre los diseños arquitectónico, de navegación, y de interfaz de usuario.
- **Generación de páginas y pruebas:** se codifican las páginas diseñadas, así como las funcionalidades de cada una para la implementación posterior de pruebas del sistema.

Evaluación del cliente: Busca obtener o conocer la satisfacción del cliente con cada fase o iteración, empleada para la corrección temprana de errores.

2.3.1.5. HDM

Hypermedia Design Model (HDM) por sus siglas en inglés, es una metodología centrada en la navegación, en el diseño a gran escala y pretende dar solución a problemas relacionados con las conexiones a bases de datos. De acuerdo con Rodríguez [33], esta metodología está dividida en dos secciones: el diseño a gran escala y el diseño a pequeña escala. Las fases del proceso de desarrollo con HDM son:

- **Identificación de identidades:** HDM considera como entidades a las estructuras de información abstraída del mundo real, de manera que identifican en primer lugar los objetos necesarios para el desarrollo de la aplicación.

- **Identificación de objetos:** Una vez identificados las entidades, se identifica la funcionalidad, actividades y como interactuará cada objeto a través de la aplicación.
- **Determinación de grupos de objetos:** Los objetos son reasignados por grupos, estructurando los diferentes componentes de manera jerárquica. El grupo de entidades se denomina componente.
- **Determinación del tipo de navegación:** En esta fase se determinan los enlaces que interconectan a las entidades y componentes, existen tres tipos de enlace: estructurales, de componente, y de aplicación.

2.3.2. Comparación por criterios de calidad según Norma ISO 25000

En la Norma ISO 25010 perteneciente a la familia ISO/IEC 25000, existen diversos aspectos que se emplean para evaluar la calidad del programa, se encuentran divididos en ocho grupos y cada uno evalúa una parte específica del programa. De acuerdo con [34], los criterios de evaluación se encuentran agrupados de la siguiente manera:

- **Adecuación funcional:** El software satisface las necesidades del usuario, y cumple los requerimientos por los que fue diseñado.
- **Eficiencia de desempeño:** Los recursos que emplea en condiciones específicas proporcionan un correcto desempeño de la aplicación.
- **Compatibilidad:** Facilita el intercambio de información con otros sistemas o componentes que tienen relación en su entorno.
- **Usabilidad:** Facilidad de uso, entendimiento, y aprendizaje por parte del usuario.
- **Fiabilidad:** El sistema ejecuta las funcionalidades de manera eficiente en condiciones y tiempo determinado.
- **Seguridad:** Protección de la información circulante dentro de la aplicación.
- **Mantenibilidad:** Facilidad del sistema para ser modificado o actualizado.
- **Portabilidad:** Facilidad de transferencia del sistema de entorno de hardware, software u otro.

En el cuadro que se presenta a continuación (*Tabla 2*), se realiza la comparación de resultados de cada metodología y se determina cuántas características de cada grupo que integra la evaluación de calidad con la Norma ISO 25000 cumple cada una de

ellas, determinando la ventaja que tiene la metodología SWIRL sobre las demás metodologías.

Tabla 2 Comparativa según la norma ISO 25000

Metodología	Normas de calidad								
	Adecuación Funcional	Eficiencia de desempeño	Compatibilidad	Usabilidad	Fiabilidad	Seguridad	Mantenibilidad	Portabilidad	Total
WSDM	1/3	2/3	2/2	2/6	1/4	2/5	3/5	3/3	16/31
SOHDM	2/3	1/3	2/2	4/6	2/4	1/5	2/5	3/3	17/31
OOHDM	2/3	2/3	2/2	3/6	3/4	2/5	4/5	3/3	21/31
IWEB	1/3	2/3	1/2	2/6	3/4	3/5	3/5	3/3	18/31
HDM	2/3	1/3	1/2	3/6	2/4	2/5	2/5	3/3	16/31

La *Tabla 2* presenta los campos que mayor importancia poseen los sistemas web en el momento de ser entregados y utilizados por el usuario final. Cada metodología de desarrollo posee procedimientos que cumplen (en grado mayor o menor) este requerimiento. Según la puntuación obtenida, se logra identificar que las metodologías presentan déficit en algunas características de calidad, dando cabida a problemas futuros para el equipo de desarrollo. Es claro que las metodologías web fueron creadas para satisfacer un requerimiento propio de los proyectos web, pero estos requerimientos pueden verse obviados en algunas metodologías, por lo que se justifica la creación de las metodologías híbridas.

Seleccionar una metodología híbrida consume una gran cantidad de tiempo, ya que es necesario comparar los procedimientos y cualidades que posee, con los requerimientos que el proyecto necesita. Es por ello que se realizó un estudio comparativo de las metodologías de desarrollo [35] y un estudio sobre los enfoques que poseen comúnmente las metodologías web [36], detallando las características importantes y las que son poco enfocadas por las metodologías.

En base a los resultados obtenidos, se concluye que las metodologías web no cumplen con todos los requerimientos que un proyecto web necesita, creando deficiencias que los “project manager” deben cubrir empleando otras metodologías o

métodos de desarrollo. Es por ello que se propone una nueva metodología de desarrollo híbrida denominada SWIRL, la cual satisface los requerimientos que todo proyecto o sistema web necesita.

2.4. Carencias en las metodologías de desarrollo de aplicaciones web.

La evaluación de una metodología de desarrollo de software está dirigida por el análisis de variables comunes que se destacan durante el proceso de construcción o desarrollo, y conllevan a mejorar los procedimientos. Según la investigación de Molina y Pedreira [36], las metodologías web presentan características ágiles comunes importantes para un correcto desarrollo de programas, el proceso de análisis y estudio primario de cada una de las características comunes según diversas metodologías de desarrollo se detalla en la artículo "Approach of Agile Methodologies in the Development of Web-Based Software" [36] :

- **Buena flexibilidad ante cambios:** La facilidad de agregar nuevos requerimientos, o especificar cambios en las funcionalidades del sistema a lo largo del ciclo de vida del software sin alterar riesgosamente el proceso de desarrollo.
- **Comunicación constante entre el equipo de desarrollo:** Los cambios, procesos y demás actividades efectuadas a lo largo del proyecto deben ser comunicados, ya sea por cualquier medio o con una buena documentación.
- **Inclusión del usuario final durante el proyecto:** La aceptación del usuario final a las funcionalidades creadas, permite mantener un margen reducido de riesgos, desarrollando los requerimientos de manera adecuada y con los cambios deseados por el cliente.
- **Documentación precisa y necesaria:** La documentación dentro del proceso es de relevancia en relación a los diagramas de componentes, navegación y funcionalidades del sistema, para que sean conocidos y entendidos por los interesados.
- **Implementación de artefactos UML:** Uso de diagramas UML para desarrollar la documentación y modelado del sistema.

Los aspectos mencionados brindan al equipo de desarrollo facilidades durante el proceso, con el objeto de que el proyecto final no se aleje en gran escala a la

aplicación deseada por el cliente. Además, facilitan el entendimiento dentro del mismo equipo gracias a la implementación de un lenguaje unificado fácil de entender como lo es UML. La **Tabla 3** presenta un análisis de las características comunes que cubre cada metodología de desarrollo presentada en el estudio actual, mediante lo cual se puede concluir la necesidad de una nueva metodología que englobe dichas características.

Tras la comparativa de las diferentes metodologías de desarrollo web, se concluye que las características que cubren estas metodologías no engloban en su totalidad las características que todo proyecto web debe contemplar. Un ejemplo de lo mencionado recae en las metodologías que centran su enfoque en la satisfacción del usuario, incluyéndolo en sus etapas, dejando como mínima preocupación las actividades del proceso de desarrollo, y en la mejora de las mismas. Es decir, buscan cumplir con un producto de calidad y requisitos básicos de un aplicativo web, sin intensificar la calidad en los procesos y actividades del desarrollo de proyectos orientados a la web.

Tabla 3 Comparativa de las características comunes en las metodologías de desarrollo

	WSDM	SOHDM	OOHDM	IWEB	HDM
Flexibilidad ante cambios	Alta	Alta	Media	Alta	Media
Comunicación constante entre el equipo de desarrollo.	Alta	Alta	Media	Alta	Media
Inclusión del usuario final durante el proyecto	Durante todo el proceso de desarrollo	Durante todo el proceso de desarrollo	Durante todo el proceso de desarrollo	Durante todo el proceso de desarrollo	En actividades específicas planteadas
Documentación precisa y necesaria	Necesaria	Abundante	Poca	Necesaria	Necesaria
Implementación de artefactos UML	Aplica	No aplica	Poco uso de UML	Aplica	Poco uso de UML

Ante esto, los directores de proyecto optan por el uso de procesos y características de diversas metodologías acoplándolas al proyecto en el cual se trabajan. Sin embargo, esto implica una gran aplicación de recursos, principalmente en tiempo, tal como destacan varios investigadores [37], [12]. Es por ello que se crea la necesidad

Capítulo 2. Estado de la cuestión

de diseñar una nueva metodología híbrida, la cual contemple las características sobresalientes de las otras metodologías, y cubra las características que se echan en falta de cada una de ellas. El objetivo será llevar a cabo un proceso óptimo y de calidad que sea empleado en los proyectos de software web, independientemente del entorno o recursos asignados.

3. HIPÓTESIS DE TRABAJO Y OBJETIVOS

3.1. Hipótesis de trabajo.

El surgimiento de nuevas metodologías de desarrollo web específicas para cubrir necesidades del proyecto, induce al director del proyecto a fusionar fases de diversas metodologías, estableciendo los procesos más adecuados acorde al proyecto creado.

H1: Las metodologías de desarrollo de aplicaciones web carecen de características específicas, principalmente en las fases de análisis y modelado.

En vista de las carencias que algunas metodologías de desarrollo presentan, es necesario proponer una nueva metodología que abarque todas las necesidades que los proyectos orientados a la web pueden llegar a presentar en diferentes ambientes, facilitando su desarrollo.

H2: Una nueva metodología de desarrollo híbrida brinda beneficios, tanto al desarrollador como al proyecto de desarrollo.

Las fases de la metodología propuesta precisan crear un modelo de evaluación mediante preguntas idóneas para evaluar los diferentes proyectos orientados a la web bajo cualquier ambiente de desarrollo.

H3: Los resultados de aplicar el modelo de evaluación en los proyectos orientados a la web, permiten mejorar su productividad y eficiencia.

3.2. Tesis.

Es posible crear una metodología, que denominaremos metodología híbrida SWIRL, que puede cubrir las carencias detectadas en las usadas habitualmente para desarrollos web.

Para demostrar la tesis propuesta es necesario alcanzar los siguientes objetivos.

3.2.1. Objetivo general.

Después de haber realizado los estudios citados en las hipótesis de trabajo del apartado anterior, se plantea la necesidad de diseñar un modelo de evaluación de metodologías de desarrollo de aplicaciones web para la determinación del grado

correcto de desarrollo mediante preguntas específicas durante las fases de análisis, planificación, modelado, implementación, revisión y pruebas, y lanzamiento que constan en la metodología híbrida SWIRL.

3.2.2. Objetivos específicos.

- Comparar las diferentes metodologías de desarrollo web mediante una amplia revisión sistemática de la literatura relacionada.
- Implementar la metodología propuesta cubriendo las carencias detectadas en el desarrollo de programas web con las metodologías actualmente usadas.
- Identificar las preguntas específicas de evaluación basadas en las etapas de desarrollo de la metodología propuesta mediante el establecimiento de actividades y requisitos.
- Implementar una aplicación web que emplee el modelo de evaluación propuesto para la evaluación de proyectos ya desarrollados orientados en la web y la identificación de mejoras durante el proceso de desarrollo para futuros proyectos.

3.2.3. Preguntas de investigación.

Con la finalidad de obtener un desarrollo del modelo de evaluación basado principalmente en las características y etapas planteadas dentro de la metodología propuesta, es sustancial considerar las siguientes preguntas de investigación. Ello permitirá lograr conocer las actividades esenciales dentro del proceso de desarrollo, las cuales brindan al desarrollador resultados destacados dentro de la aplicación web:

P1: ¿Qué impacto poseen las etapas del ciclo de vida de la metodología propuesta dentro de un proyecto web?

P2: ¿Cuáles son las actividades predominantes que el director del proyecto debe considerar durante el desarrollo de aplicaciones acordes a la metodología propuesta?

P3: ¿Cómo beneficiarán los indicadores el modelo en la evaluación de aplicaciones web?

Las preguntas planteadas poseen un enfoque característico. La primera inclina su enfoque al conocimiento de la importancia de cada una de las fases que posee la

metodología, planteando la relevancia y los beneficios dentro de los proyectos basados en la web.

La segunda centra sus esfuerzos en la catalogación de las actividades consideradas por los directores del proyecto durante las diferentes fases de desarrollo; mientras que la tercera pregunta, concluye la investigación valorando el modelo de evaluación desarrollado.

3.3. Justificación.

El desarrollo de aplicaciones web se ve inmerso en tendencias y normativas guiadas por metodologías de desarrollo. Cada una de ellas presenta fases en las que se ve inmerso el cliente y el equipo de desarrollo, predominando el uso en metodologías híbridas, las cuales permiten la inclusión de nuevos enfoques al proceso de desarrollo.

La necesidad de un modelo para evaluar la calidad y pertinencia de una aplicación web, es presenciada por parte de los directores del proyecto y desarrolladores. Cada aplicación es diseñada para un enfoque específico; por ejemplo, entre otros, para entretenimiento, para gestión contable, sistemas administrativos. Sin embargo, para diseñar un software se debe seguir un ciclo de procesos que permita obtener un software de calidad: ¿Durante el ciclo de vida del proyecto se permitiría su mejora?, o ¿Cómo identifica si las actividades realizadas son las más adecuadas? Estas interrogantes son constantes durante el ciclo de vida del proyecto. Para lograr responderlas es necesario poseer un modelo estándar que permita evaluar los resultados.

El modelo de evaluación basa su análisis en las características comunes, requerimientos e indicadores de calidad necesarios dentro de cada proyecto de software orientado a la web, los cuales se basan explícitamente en la metodología híbrida propuesta denominada SWIRL, que se describe en la siguiente sección. La evaluación basa su valoración según tres modelos de rigurosidad, los cuales poseen un enfoque de valoración desde muy estricto hasta poco estricto, sin dejar de lado las necesidades que todo proyecto orientado a la web debe poseer. La validez del modelo se fundamenta en la ejecución de pruebas de terceros analizando la correspondencia de cada una de las preguntas con las características, fases y sub-fases que la

Capítulo 3. Hipótesis de trabajo y objetivos

metodología SWIRL propone, con la finalidad de identificar las áreas y fases en las que el proceso de desarrollo de dicha aplicación puede mejorar.

El modelo propuesto contempla indicadores basados en una metodología híbrida, la cual es idónea para proyectos de diferentes tamaños, además de considerar enfoques comunes dentro del desarrollo web. Cada evaluador se encuentra basado en las actividades que contempla SWIRL, indicando además el impacto que puede llegar a presentar la ausencia o presencia del mismo.

4. PROPUESTA DE UNA METODOLOGÍA DE DESARROLLO HÍBRIDA: SWIRL

La metodología propuesta SWIRL, por sus siglas, (Software Web Iterativo Relacional Lógico), es una metodología de desarrollo enfocada en las aplicaciones basadas en la web, que combina el enfoque híbrido e iterativo [38], el cual según Brhel et. all [39] alcanza el refinamiento del producto mediante las divisiones de sus funcionalidades. Cada una de las operaciones dentro de sus fases brindan agilidad al proceso de desarrollo, además de reforzar la comunicación del equipo de desarrollo con el cliente, incluyéndolo en la mayoría de los procesos. Esto último beneficia tanto al producto como al proceso del proyecto. En cuanto al producto, al incluir al cliente se garantiza que la aplicación web al momento de ser lanzada cumpla con todos los requerimientos que ha establecido y además satisfaga las necesidades que presentan los proyectos orientados a la web.

Para el desarrollo de la metodología, se han analizado las necesidades que presentan los proyectos orientados a la web y las características que las metodologías consideran fundamentales, para así establecer las actividades de cada etapa acorde a las necesidades identificadas [38]. Para ello, tras el estudio de las metodologías en el capítulo 2, se llega a la propuesta de la metodología híbrida que cubra aspectos deficientes que se detectaron en otras metodologías.

La metodología centra su objetivo en la integración del cliente para las futuras modificaciones y mantenimiento, además de las técnicas SEO para el modelo de negocio. Es por ello que algunas de sus características con mayor relevancia son las actividades iterativas con validación del usuario, la reducción de los costos de recursos, y su uso en diferentes proyectos sean grandes o pequeños.

4.1. Modelo y ciclo de vida.

La metodología SWIRL se basa en el modelo de desarrollo iterativo, empleando dentro de cada iteración características como el costo, tiempo, calidad, alcance y comunicación. Dentro del ciclo de vida de la metodología se involucra cada parte interesada, el cliente y el “product manager” tienen participación activa dentro de las actividades. El proceso de desarrollo repetitivo se considera una iteración, incluyendo los entregables de cada fase y los entregables finales en la fase de clausura. El ciclo

de vida posee 6 fases como se muestra en el **Gráfico 1**. Las iteraciones cortas permiten verificar y corregir los errores a tiempo, antes de entregar el producto final al cliente.

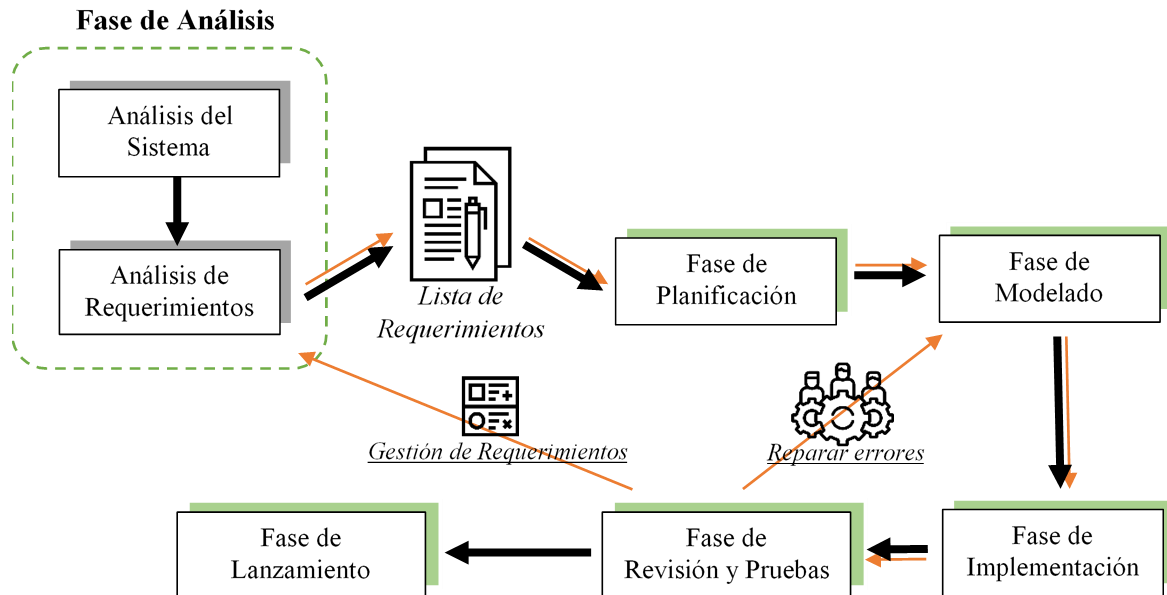


Gráfico 1 Proceso de desarrollo de la metodología SWIRL

4.2. Fases de la metodología.

Se basa en la ejecución de iteraciones dentro de sus fases, permitiendo la integración total del usuario durante el proceso, e incorporando la fase de modelo de negocios dentro de su ciclo de vida. Dichas iteraciones constantes simulan la forma de un remolino, de donde nace el nombre de esta metodología. Las fases de SWIRL son seis:

- Análisis
- Planificación
- Modelado
- Implementación
- Revisión y pruebas
- Lanzamiento y Marketing

4.2.1. Fase I: Análisis.

Dentro de esta fase se identifican los puntos clave para el desarrollo correcto del proyecto y las reglas, requerimientos y alcance bajo las cuales se va a desarrollar.

Esta fase es la principal y más común dentro de las metodologías de desarrollo, su principal descripción se debe a la necesidad de definir las funcionalidades y entrono base bajo el cual se va a desarrollar el proyecto. Elaborar de manera adecuada esta fase es esencial para que el proyecto tenga éxito, debido a que los resultados y entregables en esta fase son los pilares bajo los cuales se implementarán las funcionalidades generales del sistema.

Los objetivos de esta fase incluyen: la identificación de los procesos requeridos dentro del modelo de negocio, analizar la viabilidad del proyecto, definir los roles de las partes interesadas. Para el cumplimiento de los objetivos mencionados, se plantean diversas estrategias como: la identificación de la perspectiva del producto conforme a las necesidades del cliente, analizar los riesgos en base a cada una de las funcionalidades del sistema, entre otras. La fase se subdivide en cuatro subcaracterísticas, con sus respectivas actividades:

- **Definición del sistema:** se establecen los actores mediante: el uso de las plantillas de usuarios, el alcance general del proyecto, la factibilidad del sistema, la perspectiva del producto, el análisis de riesgos y una explicación detallada de los módulos del sistema:
 - **Reconocimiento del sistema:** se identifican las especificaciones del sistema, las características del sistema actual, y los requerimientos del proyecto a desarrollar.
 - **Factibilidad:** se determina si el desarrollo del proyecto es factible tanto económica, tecnológica y operativamente, para ello se analizan los beneficios, objetivos, y alcance del proyecto.
 - **Funcionalidad:** se describen las funcionalidades del sistema, generando una idea del producto final.
 - **Riesgos:** se determinan los posibles riesgos dentro del desarrollo siendo analizados de manera cuantitativa y cualitativa.
- **Modelado de negocio:** para ello se presenta el modelo de negocio a trabajar dentro del proyecto y la matriz detallada con las reglas de negocio:
 - **Descripción:** se identifican las actividades del proceso de negocio mediante una descripción textual y gráfica del proceso de negocio, además de incluir los roles del usuario final mediante la segmentación de usuarios y especificación de roles.

- **Evaluación:** se emplea el análisis FODA para conocer la situación en la cual se desarrollará el proyecto, las oportunidades, debilidades y posibles amenazas.
- **Innovación (Opcional):** centra su objetivo en identificar si el proceso actual de negocio necesita modificaciones para brindar mejores resultados.
- **Identificación de interesados:** mediante el uso de la matriz general de interesados se especifican los actores, sus roles y responsabilidades dentro de cada fase:
 - **Roles e interesados:** se establecen los interesados y sus roles respectivos dentro de cada fase.
 - **Actividades:** se describen las actividades para cada uno de los integrantes.
- **Requerimientos:** es primordial identificar los requerimientos funcionales, no funcionales, de usuario y de navegación; así como la priorización de los requerimientos mediante la matriz de priorización:
 - **Elicitación:** es la recolección, y descripción de los requisitos emitidos por el cliente, se centra en resolver las necesidades que el cliente o usuario final poseen. Es importante establecer los requerimientos funcionales, no funcionales, de usuario, y del sistema
 - **Análisis:** Se evalúan, refinan y priorizan los requerimientos elicitados para su posterior planificación de desarrollo.
 - **Validación:** proceso necesario antes de la codificación para asegurar que se aborden todas las problemáticas y necesidades.

4.2.2. Fase II: Planificación.

En esta fase se definen los recursos y actividades a ser ejecutadas por los interesados dentro del proceso de desarrollo, designando los tiempos, los recursos tecnológicos y humanos, la gestión de riesgo y las comunicaciones necesarias para el proyecto. Las actividades inmersas dentro de esta fase son importantes debido a que ayudan a llevar un mejor control de las actividades a realizar a lo largo del proyecto asegurando su correcto desarrollo y cumplimiento hasta el final del proceso, dichas actividades consideradas son:

- **Historias de usuarios:** para la definición de las historias de usuario se emplea el método SPIDR, mediante el cual se define la prioridad, el riesgo en desarrollo, la descripción de cada historia según los requerimientos funcionales detallados.
- **Definición de entregables:** se establecen, describen y planifican cuáles serán los entregables que se realizarán y a quienes irán dirigidos.
- **Gestión de cronograma:** Esta fase se realiza con la finalidad de conocer los tiempos y recursos que se emplean para desarrollar cada fase, mediante el uso del diagrama de Gantt.
- **Gestión de riesgos:** Los riesgos fueron identificados y establecidos cada uno según el plan de prevención de riesgos y el plan de minimización.
- **Gestión de comunicaciones:** Este punto consta solo de la definición y descripción del método y herramientas para establecer la comunicación entre los interesados.
- **Estructura de desglose de trabajo:** se grafican y detallan las actividades de forma jerárquica descomponiendo de esta manera el proyecto en actividades poco complejas.
- **Gestión de cambios y calidad:** un punto clave es detallar el plan de aseguramiento y control de la calidad; es decir, definir cada una de las actividades a realizar para garantizar que el proceso sea eficiente y el producto ofertado sea de calidad.

4.2.3. Fase III: Modelado.

Durante esta fase se diseñan los modelos necesarios para la estructuración del contenido, datos y el flujo de navegación a través de las páginas, lo cual es necesario para definir de manera entendible cómo será el diseño y funcionamiento del sistema a diseñar. Los modelos inmersos dentro del mismo son: modelo conceptual, modelo de navegación, prototipado de la interfaz de usuario y modelos UML. Las actividades a ejecutar dentro de esta fase son:

- **Diseño de modelo conceptual:** el modelado es fundamental para desarrollar las bases sobre las cuales se codificará la aplicación, además de tener una guía para posibles futuras modificaciones. El modelo conceptual se diseña

mediante el uso de herramientas CASE y se detalla la estructura de la base de datos a emplear dentro de la aplicación.

- **Diseño de modelo de navegación:** el modelo de navegación muestra la forma en que las páginas se encuentran interconectadas y los tipos de usuario que tendrán acceso a las mismas.
- **Diseño de interfaz abstracta de usuario:** es importante bosquejar el prototipo de la aplicación, estableciendo el diseño de las interfaces de usuario a codificar e la fase siguiente. Para ello, se hace uso de la herramienta “Balsamiq Mockups”.
- **Modelado de diagramas UML:** se realizan los diagramas UML respectivos para describir el proyecto y que sea entendible para los diferentes integrantes del equipo de desarrollo. Dentro de estos diagramas se incluyen, diagramas de actividades, de estados, de secuencia, de componentes, de estados.

4.2.4. Fase IV: Implementación.

Esta fase contempla los procedimientos necesarios para codificar cada una de las funcionalidades del sistema, así como el correcto del diseño de la interfaz de usuario. Sin esta fase el proceso de desarrollo de cada una de las funcionalidades del sistema no se encontraría documentado, lo cual generaría problemas a futuras modificaciones considerando que no se dispondría de una documentación entendible para cualquier nuevo integrante.

Dentro de esta fase existen aspectos importantes a considerar por el equipo de desarrollo como lo es la nomenclatura. Es necesario emplear una nomenclatura general independientemente del lenguaje de programación empleado, de manera que la comunicación y entendimiento por los demás miembros del equipo sea fácil. Las actividades dentro de esta fase son:

- **Separación de tareas entre los miembros de trabajo:** la separación de tareas consiste en la asignación de actividades inmersas en el área de la codificación, separando el diseño de la interfaz de la codificación de las funcionalidades de la aplicación.
- **Codificación:** de las funcionalidades y componentes, mediante el uso de herramientas de programación adecuadas.

- **Estándares de codificación:** las validaciones de las restricciones se realizan mediante el establecimiento de los parámetros de seguridad de la aplicación, además de implementar pruebas que aseguren las restricciones que se han establecido en el alcance y en ciertos casos como requerimientos.
- **Definición de módulos e implementación:** Se definen los componentes a incluir dentro del programa, y se detallan de manera general los módulos implementados dentro del sistema.
- **Diseño y codificación de páginas principales:** Se diseñan mediante la codificación las páginas principales y todas las interfaces de usuario, el lenguaje depende del seleccionado dentro del proyecto.
- **Interconexiones:** se realizan las conexiones y comunicación entre las páginas, además se realizan las conexiones con la base de datos, y enlaces externos en caso de ser necesarios.

4.2.5. Fase V: Revisión y pruebas.

Esta fase se basa en el proceso iterativo de evaluaciones y pruebas concurrentes de todas las funcionalidades implementadas. Esta fase se contempla en las diversas metodologías, y es de suma importancia para entregar un producto final de calidad y que cumpla con las expectativas del usuario final. Las actividades inmersas dentro de estas fases son:

- **Control integrado de cambios:** se gestionan los cambios realizados y se verifican el correcto cumplimiento y funcionamiento de cada uno de ellos.
- **Pruebas de integración:** las pruebas de integración se realizan por parte de los interesados, a diferencia de las pruebas del sistema que son desarrolladas por terceros. Las pruebas de integración se realizan a cada funcionalidad del sistema y específicas para cada requerimiento establecido
- **Detección y corrección de errores:** para la detección de errores se emplea la herramienta externa de W3C validator, mediante el cual se analizan los posibles errores o redundancias dentro del código.
- **Pruebas del sistema y de aceptación:** para las pruebas del sistema se realizan las revisiones por pares mediante la plantilla que se establece en la metodología. La revisión por pares se realiza al final del desarrollo de cada funcionalidad del programa; es decir, se realiza de modo incremental.

- **Control del alcance:** se verifica mediante la evaluación que el alcance establecido en la planificación sea cumplido correctamente.
- **Cierre:** se realiza el proceso respectivo para el cierre del proyecto, entrega del sistema, y entrega de la documentación y entregables establecidos.

4.2.6. Fase VI: Lanzamiento y marketing.

En este punto, se detallan actividades claves para el almacenamiento, lanzamiento y funcionamiento de la aplicación en la web. Pocas metodologías incluyen esta fase como obligatoria, sin embargo, su necesidad dentro de la metodología SWIRL se basa en garantizar el funcionamiento adecuado del sistema, y el conocimiento del mismo por parte de los diferentes usuarios finales. Dichas actividades son:

- **Selección y alojamiento del “hosting”:** Esta actividad es necesaria para el correcto almacenamiento de los archivos y códigos fuentes. Es importante que se consideren varios factores al momento de seleccionar el hosting adecuado, principalmente la capacidad que el programa necesitará, tanto para los archivos actuales como los datos que se almacenarán en la base de datos.
- **Preparación del dominio:** se debe realizar un proceso de búsqueda en el cual se establezca un dominio llamativo y que sea fácil de recordar por los usuarios, además de que se encuentre disponible. Esta actividad es necesaria antes de poner en funcionamiento la aplicación.
- **Configuración del certificado SSL:** Esta actividad es necesaria para asegurar la información que el usuario ingresará dentro de las funcionalidades que lo merezcan. Este paso es obligatorio si el programa recoge información sensible del usuario.
- **Campaña de marketing/ SEO:** Esta actividad no es empleada con mucha regularidad, sin embargo, es considerada dentro de esta metodología para dar a conocer a una mayor cantidad de usuarios el sistema, ya sea una aplicación informativa, o un proyecto en donde no exista un cliente definido.

4.3. Características de la metodología.

SWIRL, al ser una metodología híbrida, considera una gran amplitud de las necesidades que posee un proyecto web. Es por ello que, a continuación, se detallan las características que posee la metodología para resolver las especificaciones del proceso de desarrollo de una aplicación web:

Objetividad: La objetividad se relaciona a la habilidad de SWIRL para ajustarse a los objetivos del proyecto, gracias a la integración del cliente en todas las fases de desarrollo, lo que beneficia tanto al proceso de desarrollo como al producto final entregado. La objetividad en cuanto a la inclusión del cliente es una característica que se ha venido integrando en las metodologías ágiles actualmente, sin embargo, existen otras metodologías como la NDT, W2000, el modelo en cascada, entre otras que no se enfocan en la inclusión total del cliente en el ciclo de vida sino solo durante ciertas actividades. Durante la investigación llevada a cabo en [36], se realizó una comparativa de metodologías web, considerando la inclusión del usuario final como un indicador de estudio necesario para las metodologías de desarrollo web.

Ciclo de vida del software: En las metodologías de desarrollo de software resultan indispensables las etapas de: investigación, análisis de requisitos y diseño, sin embargo, muy pocas metodologías consideran el lanzamiento y marketing como una etapa, en algunas es conocida como despliegue del software, siendo necesaria para dar a conocer la aplicación y definir el hosting o alojamiento más adecuado para la aplicación. En el caso de SWIRL, sus fases satisfacen por completo el ciclo de desarrollo de los programas orientados a la web principalmente, ya que dentro de sus seis etapas comprende desde el análisis hasta el lanzamiento y marketing.

Comunicación: Los procesos iterativos de SWIRL estimulan la comunicación en el grupo de desarrollo del proyecto, en cada etapa los usuarios, directores de proyecto y clientes mantienen una comunicación puntual y puede ser evaluada durante todo el proceso de desarrollo del programa. Esta característica es la más común en las metodologías ágiles siendo un punto clave para un correcto desarrollo, sin embargo, cada metodología define un método de comunicación acorde a sus características o necesidades.

Adaptabilidad: Las características principales de SWIRL fueron planteadas para dar la flexibilidad de adaptarse a nuevos entornos de trabajo y al tamaño del proyecto a

desarrollar. Generalmente, el desarrollo de un programa web incluye cambios constantes en distintas áreas como los requerimientos, las especificaciones y el personal encargado del desarrollo, entre otras eventualidades. Esta característica presenta mayor relación con los enfoques de las metodologías ágiles.

Herramientas de modelado: Para el proceso de modelado del proyecto, SWIRL puede emplear cualquier herramienta de modelado que utilice el estándar UML. A diferencia de otras metodologías como WSDM, RMM, OOHDM, que emplean herramientas de soporte o notaciones propias para los diseños y modelados, lo cual dificulta o genera cierta incomodidad para el equipo de trabajo a la hora de realizar la esquematización del proyecto.

Técnicas SEO: Actualmente SWIRL es una de las pocas metodologías que integra las técnicas SEO en el proceso de desarrollo, lo cual es una clara ventaja que permite emplear procedimientos para que el programa tenga un mejor posicionamiento en los resultados de los motores de búsqueda como Google, Firefox, Opera, etc. Otra metodología de desarrollo que incluye las técnicas SEO durante su ciclo de vida es SNAIL, mientras que, para las otras metodologías, solo se emplea un modelo de posicionamiento que no se considera dentro de las fases o actividades del proceso de desarrollo.

Modelos de Negocios: A través de los modelos de negocio, SWIRL permite representar todos los actores y procesos que intervienen dentro del funcionamiento de una empresa. De esta manera, se logra determinar aspectos como el propósito, estructura y finalidad de la empresa. Todo ello representado de manera conceptual.

4.4. Criterios de evaluación para SWIRL

Los criterios empleados de la norma ISO 25000 para evaluar una metodología aseguran un proceso de desarrollo de calidad. SWIRL considera esos criterios para cubrir las diversas necesidades de las aplicaciones web. A continuación, se describen los criterios de evaluación de la ISO 25000 que intervienen en las diferentes fases de la metodología SWIRL.

Adecuación funcional: Un punto clave de SWIRL es la completitud y corrección funcional del sistema, establecido principalmente en la fase de análisis. La elicitación y análisis de los requerimientos brindan seguridad al proyecto, preparándolo adecuadamente para cambios repentinos dentro de cualquier etapa del proceso.

Eficiencia de desempeño: Los puntos clave para el desempeño del programa se evalúan constantemente en la fase de pruebas y revisión. De esa manera, se identifican los posibles errores y parámetros que reducen el rendimiento del programa. Sin embargo, los aseguramientos de los recursos no son estrictos bajo condiciones determinadas.

Compatibilidad: Característica asegurada en la fase de pruebas y revisión mediante las pruebas de integración.

Usabilidad: El aseguramiento de la usabilidad se genera mediante la integración del usuario en el proceso de desarrollo, lo que permite que el usuario final conozca perfectamente el funcionamiento del sistema. Las interfaces y la accesibilidad del programa juegan un gran rol dentro del proyecto, por lo que son consideradas en gran detalle en la fase de modelado.

Fiabilidad: Las especificaciones para la disponibilidad y madurez del programa se interpretan en la fase de planificación. Sin embargo, la capacidad de recuperación y tolerancia a fallos no son evaluadas estrictamente por SWIRL.

Seguridad: Dentro de este campo sobresalen la integridad y confidencialidad dentro de las fases de modelado e implementación.

Mantenibilidad: La mantenibilidad se encuentra enfocada principalmente por la rigurosa documentación del programa en cuanto a especificaciones importantes.

Portabilidad: El aseguramiento del “hosting” y su lanzamiento hacen que la metodología asegure la portabilidad del programa.

En la Tabla 4 se recogen las respuestas del análisis comparativo de las metodologías analizadas, junto con las de SWIRL, relativas a la norma ISO 25000.

Tabla 4 Comparativa de SWIRL según la norma ISO 25000

Metodología	Normas de calidad								
	Adecuación Funcional	Eficiencia de desempeño	Compatibilidad	Usabilidad	Fiabilidad	Seguridad	Mantenibilidad	Portabilidad	Total
WSDM	1/3	2/3	2/2	2/6	1/4	2/5	3/5	3/3	16/31
SOHDM	2/3	1/3	2/2	4/6	2/4	1/5	2/5	3/3	17/31
OOHDM	2/3	2/3	2/2	3/6	3/4	2/5	4/5	3/3	21/31
IWEB	1/3	2/3	1/2	2/6	3/4	3/5	3/5	3/3	18/31
HDM	2/3	1/3	1/2	3/6	2/4	2/5	2/5	3/3	16/31
SWIRL	3/3	2/3	2/2	5/6	3/4	3/5	4/5	3/3	25/31

5. MODELO DE EVALUACIÓN DE METODOLOGÍAS

En este apartado, se describe el proceso de diseño del modelo de evaluación de metodologías de desarrollo de software, el cual posteriormente será implementado a través de una aplicación web. Además, se detallan los posibles problemas presentados durante el diseño del modelo, así como las preguntas que el modelo contiene según su dificultad (Modelo básico, modelo intermedio y modelo riguroso).

El objetivo del modelo de evaluación es identificar el grado de cumplimiento de varios proyectos de desarrollo web de diferentes tamaños, con la finalidad de conocer las fases en las cuales el proyecto contempla un proceso de desarrollo completo y cuáles son las fases que necesitan ser mejoradas. El modelo se empleará para evaluar diferentes aplicaciones web ya desarrolladas, permitiendo analizar por completo todas las fases efectuadas en el proyecto y mejorar las actividades necesarias en proyectos futuros, e inclusive corregir ciertas actividades en proyectos actuales, siempre y cuando la fase no haya sido concluida, y su cambio no retrase la entrega del proyecto ni provoque algún riesgo durante el desarrollo.

Algunos de los aspectos que evalúa el modelo son: la buena comunicación con el cliente; la documentación necesaria; el diseño de los modelos necesarios para implementar y realizar futuras actualizaciones; la definición de historias de usuarios, y el análisis del sistema. Además, también se estudian otros aspectos de relevancia dentro de cada fase. El diseño del modelo de evaluación tiene como referencia diferentes aspectos de la norma ISO 25000 en las fases de análisis, planificación, modelado, implementación, revisión y pruebas y lanzamiento que constan en la metodología híbrida SWIRL.

5.1. Metodología de desarrollo del modelo de evaluación.

Para lograr diseñar un modelo de evaluación adecuado es necesario analizar la norma ISO 25000, las características comunes de los sistemas web, y cada una de las fases de la metodología SWIRL (análisis, planificación, modelado, implementación, revisión y pruebas, lanzamiento), ya que recopila de manera general las principales características de las metodologías de desarrollo web.

En base a lo mencionado, se plantea un proceso incremental para el desarrollo del modelo de evaluación que posee las siguientes etapas: (1) identificación del impacto de las fases; (2) clasificación de las actividades; (3) planteamiento de indicadores (preguntas y entregas específicas) en base a las subcaracterísticas; y (4) validación del modelo acorde a la relevancia de los indicadores.

Los pasos a desarrollar dentro de cada etapa son iterativos hasta obtener resultados favorables dentro del modelo:

1. Identificar las variables a ser evaluadas.
2. Proponer el modelo en base a las normativas de la metodología.
3. Evaluar el impacto que cada una posee acorde a las categorías propuestas.

Cada etapa de las establecidas previamente en el presente apartado (impacto, clasificación, planteamiento y validación), propone una visión diferente del modelo, con lo cual se logra abarcar características de diversa índole a ser evaluadas dentro de los proyectos web.

(1) Impacto de las fases de la metodología SWIRL.

Dentro de esta sección, se evalúa la eficiencia que cada una de las fases que conforman la metodología SWIRL aporta al desarrollo del proyecto. El impacto que un director del proyecto considera dentro de la aplicación implica en gran escala la actividad y la etapa de desarrollo en la cual se encuentren.

Es de gran importancia conocer la manera en la que influyen las fases de SWIRL dentro de un proyecto, debido a que, para obtener un modelo de evaluación adecuado para las aplicaciones web, deben considerarse las características propias y necesarias que ha de poseer un programa orientado a la web, de calidad y satisfactorio para los usuarios finales.

El impacto que proporciona cada una de las etapas se evalúa mediante un rango de porcentaje, los cuales son asignados en base a la literatura consultada referente a SWIRL, y las experiencias adquiridas durante el desarrollo de diversas aplicaciones web. No obstante, el impacto establecido es relativo al ambiente de desarrollo en el cual se ejecuta, por lo que se consideran dos ambientes dinámicos, el primero para equipos distribuidos, y el segundo para equipos no distribuidos.

Los indicadores establecidos para cada fase del modelo se pueden observar en la tabla 5. Considerar la importancia de cada una de las fases dentro de un modelo de evaluación permite asegurar la calidad durante el proceso de estimación de calidad y eficiencia de otros programas basados en la web. De manera que no se pierda realce durante la valoración del proceso y del producto desarrollado dentro de un proyecto web.

Tabla 5 Indicadores del modelo de evaluación por fases

Fases de desarrollo	Indicadores
Fase 1: Análisis	<ul style="list-style-type: none"> - Definición del sistema y modelado de negocio. - Identificación de interesados y requerimientos.
Fase 2: Planificación	<ul style="list-style-type: none"> - Historia de usuario. - Definición de entregables. - Gestión de cronograma. - Gestión de riesgos. - Gestión de comunicaciones. - Gestión de cambios y calidad.
Fase 3: Modelado	<ul style="list-style-type: none"> - Diseño de modelo conceptual. - Diseño de navegación. - Diseño de interfaz abstracta de usuario. - Diseño de diagramas UML.
Fase 4: Implementación	<ul style="list-style-type: none"> - Separación de tareas entre los miembros de trabajo. - Codificación de las funcionalidades y componentes. - Validación de las restricciones globales y generales. - Implementación de métricas de usabilidad y calidad.
Fase 5: Pruebas	<ul style="list-style-type: none"> - Pruebas de integración. - Detección y corrección de errores. - Pruebas del sistema. - Control integrado de cambios.
Fase 6: Lanzamiento	<ul style="list-style-type: none"> - Preparación del dominio. - Selección y alojamiento en el "hosting". - Configuración del protocolo SSL. - Presentación de entregables.

(2) Clasificación de las actividades por fase.

Dentro de esta sección se presentan las actividades consideradas en cada una de las fases de desarrollo de SWIRL. Listar las actividades ejecutadas ayuda al director del

proyecto a evaluar el proyecto de manera efectiva, incluyendo las características más relevantes del desarrollo web.

Identificar las actividades convenientes para realizar el proyecto es de gran relevancia durante el proceso de valoración de calidad del producto. Es por esa razón, que se estima el proceso de clasificación de las actividades como necesaria para la elaboración del modelo de evaluación de aplicaciones web. Esto incluye las necesidades que un director de proyecto y un desarrollador deben considerar vitales para elaborar un programa orientado a la web.

Después de identificar y clasificar adecuadamente las actividades para cada etapa, se desarrolla un modelo de valoración subjetivo, en base a la experiencia adquirida, de manera que, posteriormente, se seleccionen las actividades con mayor rigidez y beneficio para el proceso de desarrollo.

(3) Indicadores de evaluación según las subcaracterísticas.

Un modelo de evaluación de programas adecuado necesita ser guiado por un conjunto de indicadores, los cuales permitan evaluar al software durante cada uno de los subprocesos efectuados durante el desarrollo. Cada uno de los procesos del ciclo de vida posee etapas de mayor relevancia, enfocadas a un problema en concreto. Sin embargo, lograr evaluar las fases de manera globalizada implica un problema, tanto para el evaluador como para el director del proyecto, obteniendo como resultado una valoración superficial que tiende a ignorar la calidad completa del sistema de información.

Es importante recalcar que algunos de los subprocesos dentro de SWIRL suelen ser opcionales y su uso depende explícitamente del tipo de sistema que se está diseñando. Sin embargo, para evaluar los proyectos de forma globalizada se considera cada uno de los subprocesos definidos por la metodología, dando de esta forma una valoración distribuida de manera igualitaria.

Por lo mencionado anteriormente, surge la necesidad de plantear indicadores claves que orienten el modelo de evaluación a diseñar, para obtener como resultado un diseño de modelo eficiente y, sobre todo, que valore la calidad de los diferentes programas web desarrollados en ocasiones con otras metodologías de desarrollo.

Una vez que se encuentren identificados los indicadores de cada subproceso, se procede a detallar las interrogantes bajo las cuales se efectuará el modelo de evaluación. Las preguntas planteadas se enfocan en resolver y tasar la prioridad que demuestra SWIRL durante cada etapa de las fases del desarrollo. Cabe recalcar que dichas preguntas serán empleadas durante la evaluación efectuada por la aplicación web.

(4) Validación del modelo.

Ejecutando las tres etapas mencionadas en los apartados anteriores, se obtiene el prototipo del modelo de evaluación a emplear. Es importante que el arquetipo plasmado sea evaluado mediante procesos iterativos, logrando de esta manera perfeccionar el modelo con cada iteración.

Para ejecutar la evaluación del modelo se consideran tres grados de rigurosidad; es decir, se elaborarán tres modelos acordes a los grados (básico, intermedio y riguroso). Se evaluará cada modelo acorde a las siguientes pautas.

1. El modelo presenta preguntas adecuadas para el grado de rigurosidad establecido.
2. El modelo propone indicadores sólidos y claros para la valoración de programas.
3. El modelo considera todas las características propias de los sistemas web y su grado de rigurosidad para evaluar.

La evaluación consta de un proceso repetitivo. Dentro de cada iteración se deben efectuar los puntos establecidos anteriormente. Además, es importante mencionar que se ejecutará una prueba piloto con 3 aplicaciones, una aplicación para cada grado de dificultad del modelo de evaluación, y cada punto será evaluado según los indicadores respectivos.

Para verificar que cada una de las preguntas sea adecuada, se revisarán dos condiciones. La primera consiste en que la pregunta sea clara, precisa y fácil de entender; mientras que la segunda se basa en asegurar que la respuesta brinde información suficiente sobre el subproceso evaluado. En caso de que una de las condiciones no se cumpla, será necesario la modificación de la pregunta, mientras

que, si ambas condiciones no se cumplen, la pregunta debe ser omitida o reelaborada.

5.1.1. Uso del modelo de evaluación.

El modelo de evaluación elaborado será implementado dentro de una aplicación web desarrollada para lograr evaluar diferentes aplicaciones orientadas a la web, incluyendo la valoración según el grado de rigidez que se pretende elaborar.

Por otro lado, el modelo de evaluación diseñado se plantea en el capítulo siguiente, por consiguiente, puede ser empleado por directores de proyectos web o desarrolladores independientes, para valorar si los procesos considerados brindan un producto final de calidad. También ha de ser valorado por el analista del proyecto durante las etapas de verificación y revisión, para lograr refinar el sistema antes de ser entregado al usuario final.

5.2. Consideraciones a tener en cuenta durante el desarrollo.

Durante la creación del modelo de evaluación se presenta una serie de riesgos que pueden afectar indirectamente en la valoración de los resultados de las evaluaciones. Estos riesgos se basan en los posibles errores que se pueden cometer durante la creación de los modelos de evaluación y que pueden llegar a afectar la validez de los mismos. Los riesgos presentados a continuación son propuestos desde el punto de vista del investigador, considerando entre otras las dudas más habituales ¿es adecuado el modelo?, ¿las preguntas abordan de forma completa todas las características específicas de las aplicaciones web?, ...

- **No se consideran indicadores específicos para cada subcaracterística de la metodología SWIRL.**

En lo referente a esta preocupación se estima que cada subcaracterística derivada de los procesos de la metodología SWIRL debe ser evaluada conforme a criterios propios y que aseguren el cumplimiento de los estándares y expectativas propuestas para cada una de las actividades. Es decir, si las preguntas o indicadores que se establecen para evaluar una determinada actividad dentro de los procesos son erróneas, la valoración asignada a esa parte del proyecto afecta el resultado final, especulando con la calidad del producto o con la aplicación final que ha sido evaluada.

Sin embargo, no solo se toma en cuenta que las preguntas o indicadores posean relación con la actividad que se está evaluando, sino que logre responder satisfactoriamente a una verificación del proceso llevado a cabo durante el desarrollo del proyecto.

Es por ello que, ante los indicadores de cada subcaracterística, se establece el porcentaje de apropiación a la actividad. En otras palabras, se mide si la forma de evaluar responde correctamente y evidencia el cumplimiento o incumplimiento del ítem evaluado. De manera que la valoración final no sea superficial y se encuentre acorde a las actividades que presenta el proceso de desarrollo de la metodología SWIRL.

Un indicador es considerado como no específico cuando el impacto que provoca su respuesta no cubre las necesidades totales de las actividades realizadas, afectando no solo a la valoración de la actividad, sino también a la valoración del proyecto en general, indicando un bajo proceso de desarrollo.

- **Los indicadores no logran analizar en profundidad la calidad del proyecto implementado.**

Seleccionar los indicadores adecuados, como se ha explicado en el riesgo anterior, es de suma importancia; sin embargo, proponer una pregunta inadecuada no es la única preocupación. Una subcaracterística puede ser evaluada de diferentes maneras y bajo diferentes criterios, pero no todos logran obtener, en profundidad, la pertinencia de desarrollo durante la actividad de esa etapa.

Seleccionar los mejores indicadores no solo se basa en que se encuentren acordes a la actividad y cumplan con los requerimientos especificados del mismo; también es requerido un análisis y ponderación de los mismos, con lo cual se permite identificar las actividades de mayor peso y que enfocan a gran escala los resultados que se deben obtener o, en otras palabras, los resultados esperados.

Para ello, se emplea un análisis cuantitativo del impacto o peso generado por cada indicador ante la problemática que se pretende resolver dentro de un proceso.

- **El modelo no contempla las etapas básicas de las metodologías de desarrollo orientadas a la web.**

SWIRL es una metodología híbrida de desarrollo la cual implementa procesos de diversos modelos de desarrollo de programas. Sin embargo, una de las principales preocupaciones, aunque no la de mayor peso, es la inclusión de etapas básicas que los proyectos web deben seguir. De esta manera, se permite, no solo evaluar una aplicación web en base a parámetros estructurados por una metodología específica, sino también el análisis de calidad y permisión de la aplicación en base a modelos y estándares que los sistemas orientados a la web deben poseer, pese al modelo de desarrollo con el cual se han implementado.

Una estrategia para minimizar y, sobre todo, prevenir la aparición de las preocupaciones anteriormente mencionadas, es el previo conocimiento de los problemas comunes durante las etapas de desarrollo de proyectos basados en la web, los cuales son mencionados y analizados a fondo en la investigación de Molina y Pedreira [35]. Estos problemas son evaluados en la *Tabla 6*, de manera que se pone de relieve la importancia y el impacto, tanto positivo como negativo, que pueden ocasionar en el producto final.

Es necesario conocer los patrones de problemas comunes y las etapas en las cuales se suelen presentar para proponer un modelo que cubra dichas necesidades; es decir, se deben crear indicadores específicos dentro de los procesos que permitan ver si la aplicación a evaluar (la aplicación de terceros) puede presentar o presenta dicha problemática.

Tabla 6 Patrones de problemas en el desarrollo.

Etapa	Patrón del problema	Preocupación	Impacto
Descubrimiento	Elicitación y análisis de requerimientos.	Realizar una mala elicitación y un análisis superficial de requerimientos.	Alto. Afectando la calidad del producto final y el proceso de desarrollo del proyecto.
	Inclusión de los interesados.	Poseer poca intervención de los interesados a lo largo del proyecto.	Insatisfacción final de los usuarios durante las revisiones.
Refinamiento	Reuniones con usuarios finales	Presentar escasas reuniones con los usuarios finales o con el cliente.	No se logra conocer la opinión del usuario o del cliente en cuanto a la aplicación implementada.
	Inclusión de nuevas ideas.	La implementación de nuevas funciones del sistema es aceptada en etapas específicas (al inicio).	El usuario no puede agregar nuevas funcionalidades al sistema sin perjudicarlo.
	Aceptación de cambios continuos.	El proceso no es flexible, por lo que los cambios de requerimientos no son muy aceptados.	Un cambio puede retrasar la planificación o reestructurar las funcionalidades del proyecto.
Priorización	Medición del valor de los requerimientos.	Los requerimientos no son comprendidos por lo que no se asigna un correcto valor.	Mala selección de requerimientos y funcionalidades.
	Priorización de funciones.	Realizar una mala priorización de los requerimientos.	Las funcionalidades principales no poseen mayor importancia.
Revisiones	Revisiones con los usuarios finales o con el cliente.	La cantidad de reuniones para las revisiones son escasas.	No se conoce la opinión de los usuarios finales o clientes con respecto al sistema implementado.
	Verificación de las funcionalidades implementadas.	Falta implementar funcionalidades o el sistema no satisface todas las necesidades.	El sistema no cubre el alcance definido.

5.3. Diseño del modelo de evaluación.

Para el diseño del modelo de evaluación se han seguido las actividades propuestas en la *Sección 5.1*.

(1) Impacto de las fases de la metodología SWIRL.

El impacto que posee cada una de las fases de la metodología SWIRL se determina mediante la evaluación de las subcaracterísticas implícitas en cada una de ellas considerando los siguientes indicadores. Estos indicadores se enfocan en la importancia de las subcaracterísticas de cada fase para la calidad del proyecto final, basándose en las siguientes cuatro cualidades: útil, necesario, sencillo y beneficioso.

I1. Útil: ¿La realización de esta actividad beneficia la calidad y el resultado final del proyecto?

I2. Necesario: ¿Su omisión afecta o retrasa el proceso de desarrollo de la aplicación?

I3. Sencillo: ¿Su aplicación resulta fácil para el director del proyecto y el equipo de trabajo?

I4. Beneficioso: ¿Beneficia o facilita las actualizaciones futuras del sistema?

Para la valoración de cada uno de los indicadores planteados se estipula una escala del 0 al 3 en la cual:

- (3) Alto
- (2) Medio
- (1) Bajo
- (0) No aplica

En la tabla 7, se presenta el impacto de las diferentes etapas de la metodología en el proceso de desarrollo, establecido en tres categorías: bajo, medio y alto. Siendo alto si el promedio de los valores de los indicadores es desde 2.1 a 3; medio si el promedio es desde 1.1 a 2; bajo si el promedio es desde 0.1 a 1, y no aplica si el promedio es 0.

Tabla 7 Impacto de las etapas de la metodología en el proceso de desarrollo.

Fases	Subcaracterísticas	Indicadores				Promedio	Impacto
		I1	I2	I3	I4		
1. Análisis	Definición del sistema y modelado de negocio.	3	1	3	1	2	Medio
	Identificación de interesados y requerimientos.	3	3	3	3	3	Alto
	Análisis de Factibilidad.	1	1	3	0	1,25	Bajo
2. Planificación	Historia de usuario.	2	2	3	3	2,5	Alto
	Definición de entregables.	3	1	3	2	2,25	Alto
	Gestión de cronogramas.	2	3	2	0	1,75	Medio
	Gestión de riesgos.	3	2	3	1	2,25	Alto
	Gestión de comunicaciones.	2	1	2	0	1,25	Bajo
	Gestión de cambios y calidad.	3	3	2	3	2,75	Alto
3. Modelado	Diseño del modelo conceptual.	3	3	3	3	3	Alto
	Diseño de navegación.	3	3	3	3	3	Alto
	Diseño de interfaz abstracta de usuario.	3	3	3	3	3	Alto
	Diseño de diagramas UML.	2	3	2	3	2,5	Alto
4. Implementación	Separación de tareas entre los miembros de trabajo.	2	3	2	3	2,5	Alto
	Codificación de las funcionalidades y componentes.	3	3	2	3	2,75	Alto
	Validación de las restricciones globales y generales.	3	3	2	3	2,75	Alto
	Implementación de métricas de usabilidad y calidad.	3	3	2	3	2,75	Alto
5. Pruebas	Pruebas de integración.	3	3	2	2	2,5	Alto
	Detección y corrección de errores.	3	3	3	2	2,75	Alto
	Pruebas del sistema.	3	3	2	2	2,5	Alto
	Control integrado de cambios.	2	3	3	3	2,75	Alto
6. Lanzamiento	Preparación del dominio.	1	2	2	1	1,5	Medio
	Selección y alojamiento en el hosting.	3	2	2	2	2,25	Alto
	Configuración del protocolo SSL.	3	2	2	3	2,5	Alto
	Presentación de entregables.	2	1	2	2	1,75	Medio

(2) Clasificación de las actividades por fase.

Una vez que se ha evaluado el impacto que posee cada una de las fases de la metodología SWIRL, se procede a definir las actividades para el modelo de evaluación, creando una tarea para cada uno de los indicadores recogidos en la tabla 5.

(3) Indicadores de evaluación según subcaracterísticas.

La finalidad del proyecto es establecer un modelo de evaluación que sirva como marco para valorar la calidad y pertinencia de una aplicación web en base a las normativas y características de la metodología SWIRL, pero para ello es necesario especificar algunos indicadores o preguntas, que permitan evaluar al proyecto y conocer si sigue los preceptos se estipula en cada una de las fases de la metodología. En las siguientes tablas, se detallan una serie de preguntas relacionadas con las subcaracterísticas de cada una de las fases de la metodología SWIRL, cada modelo conlleva una serie de preguntas propias acorde a la rigurosidad, sea este:

- 1. Modelo Básico:** Este modelo contempla una complejidad de evaluación débil; es decir, se plantea una sola pregunta por cada subcaracterística. Por ende, no se abordan todas las especificaciones de cada una de las fases de la metodología SWIRL, sino solo la característica que presenta mayor importancia para un proyecto web.
- 2. Modelo Intermedio:** Este modelo evalúa el proyecto con mayor intensidad que el anterior, basando sus preguntas en los resultados finales de cada etapa y el proceso para obtenerlos. A diferencia del modelo anterior, el modelo intermedio no solo asegura un producto final de calidad, sino también permite conocer si el proceso de desarrollo del proyecto es eficiente para los interesados.
- 3. Modelo Riguroso:** La complejidad de este modelo es alta, implicando preguntas objetivas que permitan conocer la calidad del producto final, la eficiencia del proceso de desarrollo y el cumplimiento directo de los requerimientos del sistema. Este modelo de evaluación contempla, lógicamente, una cantidad mayor de preguntas en comparación a los modelos anteriores.

En la tabla 8 se presentan los indicadores presentes en las distintas actividades de la fase de análisis y el modelo a seguir para cada uno de ellos:

Tabla 8 Indicadores por subcaracterísticas de la fase 1

Fase 1: Análisis		
SC	Indicadores	Modelo
Definición del sistema	Se especifica la situación del sistema actual y futura.	1
	Se analiza la viabilidad del proyecto.	2
	Se plasman los riesgos, y beneficios de la implementación del nuevo sistema.	2
	Se realiza el análisis FODA en base a características actuales y futuras del sistema.	3
	Se identifica la perspectiva del producto.	3
	Se plasma en la documentación la empresa y las necesidades que posee.	3
	Se analiza el riesgo de las funcionalidades del sistema.	3
Modelado de negocio.	Se plasma el modelado del negocio en la documentación del sistema.	1
	Se identifican los procesos requeridos en el negocio.	2
	Se hace uso de diagrama de actividades para representar el proceso de negocio.	3
Identificación de interesados.	Se identifica los roles e interesados dentro del proyecto.	1
	Se definen los roles y actividades de los interesados	2
	Se realiza una matriz de trazabilidad con actores y sus funciones.	3
	Se segmentan los roles y tipos de usuarios finales.	3
Requerimientos	Se establecen los requerimientos dentro de la documentación.	1
	Se realizó la priorización detallada de requerimientos.	1
	Se encuentran delimitados los requerimientos relevantes mediante la priorización y refinamiento	2
	Se acordaron los requerimientos con el cliente.	2
	Se organizaron jerárquicamente según la funcionalidad.	3
	Se implementó una matriz de trazabilidad.	3

En la tabla 9 se presentan los indicadores presentes en las distintas actividades de la fase de planificación y el modelo a seguir para cada uno de ellos:

Tabla 9 Indicadores por subcaracterísticas de la fase 2

Fase 2: Planificación		
SC	Indicadores	Modelo
Historias de usuario.	Existen las historias de usuario en la documentación.	1
	Las historias se conversaron con el director del proyecto y el usuario	2
	Se establecen las interfaces de cada una de las historias de usuarios.	2
	Las historias de usuario fueron refinadas acorde a la funcionalidad principal.	3
	Se define la forma de realizar cada historia.	3
Definición de entregables.	Se define cada uno de los entregables.	1
	La documentación da conocimiento sobre las actividades realizadas.	2
	Se define la entrega de una planificación de actividades por iteraciones.	3
Gestión de cronograma.	Se estableció un plan de actividades en base al tiempo y recursos.	1
	Se establecen las fechas de las reuniones.	1
	Se acuerda una holgura acorde a la complejidad de la actividad.	2
	Se definen estratégicamente el tiempo y recursos.	3
Gestión de comunicaciones.	Se elabora un plan de comunicaciones.	1
	Se establece un plan de comunicaciones para el proyecto.	2
	Se entregó un plan de comunicaciones al cliente.	3
Gestión de riesgos	Se entregó un plan de prevención de riesgos.	1
	Se elabora un documento para analizar y responder a los riesgos.	2
	Se elabora un plan de minimización de riesgos.	3
Gestión de cambios y de la calidad.	Se definió el control de cambios.	1
	Se detalla cada cambio o modificación acorde a la matriz proporcionada.	2
	Se determinan las adquisiciones según la plantilla.	3

En la tabla 10 se presentan los indicadores presentes en las distintas actividades de la fase de modelado y el modelo a seguir para cada uno de ellos:

Tabla 10 Indicadores por subcaracterísticas de la fase 3

Fase 3: Modelado		
SC	Indicadores	Modelo
Diseño de modelo conceptual.	Se realizó el modelo conceptual de la base de datos.	1
	El modelo conceptual expresa el comportamiento de forma clara y precisa.	2
	Se diseña adecuadamente el modelo ER	3
	Se detalla un diccionario de datos propio para el cliente	3
	Se crea un modelo relacional	3
Diseño de navegación	El modelo de navegación es entendible por el cliente.	1
	Beneficia al equipo durante la implementación, verificación y pruebas.	2
	Se representó gráficamente el modelo de navegación	2
	El analista define la arquitectura del sistema	3
	Se usan actores para presentar el diseño de navegación	3
Diseño de interfaz abstracta de usuario.	Las interfaces generan un ambiente familiar al usuario	1
	Permite conocer los tipos de datos y campos a implementar en la base de datos.	2
	Posee un diseño sencillo sin especificaciones.	2
	Las interfaces se diseñaron mediante herramientas case de diseño.	3
	Contiene páginas de dependencia.	3
Diseño de diagramas UML.	Los diagramas se enfocan en el modelado del comportamiento del sistema.	1
	Brindan información sobre actividades y secuencias de las funciones del sistema.	2
	Se identifican los elementos que conforman el sistema	3
	Se representa el estado particular de un objeto contemplado dentro del sistema en una situación en específica por medio del diagrama de estado.	3

En la tabla 11 se presentan los indicadores presentes en las distintas actividades de la fase de implementación y el modelo a seguir para cada uno de ellos:

Tabla 11 Indicadores por subcaracterísticas de la fase 4

Fase 4: Implementación		
SC	Indicadores	Modelo
Separación de tareas entre los miembros de trabajo.	Se independizaron tareas acordes a la categoría de cada miembro del grupo.	1
	El director del proyecto interviene personalmente durante las reuniones de verificación de prototipos.	2
	Las actividades y responsabilidades se dividieron por módulos, funcionalidades e interfaces.	2
	Se separan las tareas acorde a las necesidades y requerimientos sobre la marcha.	3
	La separación de tareas se estableció mediante la comunicación con cada miembro del equipo.	3
Codificación de las funcionalidades y componentes	La codificación de los componentes ayuda a los desarrolladores a comprender el funcionamiento del sistema.	1
	Las funcionalidades y componentes de las interfaces se encuentran codificadas y en correcto funcionamiento.	2
	Se detallaron y definieron adecuadamente los componentes y módulos del sistema.	3
Validación de las restricciones globales y generales.	Se establecen restricciones para delimitar el proyecto.	1
	Se validaron las restricciones globales y generales.	2
	Se establecieron estándares de codificación para el conocimiento del equipo.	3
Implementación de métricas de usabilidad y calidad.	Se aplicaron las métricas que indica la normativa ISO/IEC 9126.	1
	Se hizo uso de una plantilla estándar para la evaluación del proyecto.	2
	Se realizaron las debidas pruebas de conexión con la base de datos.	3
	Se evaluó la usabilidad después de cada fase.	3

En la tabla 12 se presentan los indicadores presentes en las distintas actividades de la fase de pruebas y revisión, y el modelo a seguir para cada uno de ellos:

Tabla 12 Indicadores por subcaracterísticas de la fase 5

Fase 5: Pruebas y revisión		
SC	Indicadores	Modelo
Control integrados de cambios.	Se emplea la plantilla definida por la metodología para el control de cambios.	1
	En la documentación del control de cambios se logra identificar los cambios con la funcionalidad a la que afectan y la fecha en la que se realiza,	2
	La gestión de cambios se lleva a cabo en base la plantilla sugerida en la metodología aplicada.	3
Realización de pruebas de sistema.	Las pruebas del sistema fueron realizadas por un equipo externo y diferente al equipo de desarrollo.	1
	Se realizó la evaluación de calidad verificando que los indicadores asignados sean cumplidos.	2
	Se evaluó la tendencia de diseño.	2
	Se utilizó una plantilla para evaluar correctamente el sistema acorde a la calidad según los criterios de evaluación.	3
Realización de las pruebas de integración.	Cada uno de los componentes se evalúa adecuadamente.	1
	Se realizó la evaluación de la aplicación por componentes, elementos y la manera en que estos se interrelacionan entre sí.	2
	Existe manual de usuario y manual de programador.	3
Detección y corrección de errores.	Se emplearon herramientas externas como validador de W3C.	1
	La herramienta utilizada permite de una manera sencilla la detección y corrección de errores dentro del proyecto.	2
	Se hizo uso de dos herramientas de validación como mínimo.	3

En la tabla 13 se presentan los indicadores presentes en las distintas actividades de la fase de lanzamiento y el modelo a seguir para cada uno de ellos:

Tabla 13 Indicadores por subcaracterísticas de la fase 6

Fase 6: Lanzamiento y marketing		
SC	Indicadores	Modelo
Selección y preparación del dominio.	El proyecto tiene designado un dominio.	1
	El dominio es adecuado para la funcionalidad de la aplicación.	2
	Se seleccionó a un proveedor de confianza.	3
Selección y alojamiento de la aplicación en un hosting.	El proyecto tiene designado un hosting.	1
	El proceso de alojamiento contó con supervisión.	2
	El procedimiento para la publicación fue en base a normas.	2
	Se verificó la comunicación en el hosting a utilizar.	3
Configuración del certificación SSL.	El proyecto cuenta con certificados SSL.	1
	Cuenta con la versión mínima segura de certificados SSL.	2
	El certificado presta el cifrado adecuado.	3
Presentación de entregables.	Los entregables presentados fueron revisados y modificados bajo la supervisión de un experto.	1
	Los entregables se presentaron en el tiempo pertinente.	2
	Se presentaron todos los entregables acordados con el cliente.	3

(4) Validación del modelo por terceros.

Cada una de las preguntas planteadas dentro de las subcaracterísticas se basan en el proceso establecido dentro de las fases de la metodología SWIRL, las cuales son (1) análisis, (2) planificación, (3) modelado, (4) implementación, (5) revisión y pruebas, y (6) lanzamiento y marketing. Dentro de la metodología SWIRL, cada una de las fases posee sus actividades respectivas, y estas deben cumplir una serie de requisitos.

Dentro del proceso para la validación del modelo se emplea la técnica de evaluación por terceros, en la cual se contó con la participación de cinco expertos profesionales

ingenieros informáticos, que pertenecen a grupos de investigación relacionados con el desarrollo de software, o que han realizado proyectos de I+D+i sobre este aspecto. Los evaluadores seleccionados, así como la acreditación que los faculta para la tarea encomendada, son presentados en la *Tabla 14*.

Tabla 14 Evaluadores del modelo.

N°	NOMBRE	TÍTULO
1	Valarezo Pardo Milton Rafael.	- Ingeniero de Sistemas. - Magister en docencia y gerencia en educación superior. (Universidad de Guayaquil).
2	Rivas Asanza Wilmer Braulio	- Ingeniero de sistemas. - Magister en gestión estratégica de tecnología de la información. (Universidad de Cuenca). - Doctor en tecnologías de la información y las comunicaciones. (Universidade da Coruña)
3	Zea Ordoñez Mariuxi Paola	- Ingeniera en computación y ciencias de la informática. - Magister en sistemas de información gerencial. (Escuela Superior Politécnica del Litoral)
4	Hernández Rojas Dixys Leonardo	- Doctor en tecnologías de la información y las comunicaciones en redes móviles. (Universidade da Coruña)
5	Cartuche Calva Joffre Jeorwin	- Ingeniero en sistemas informáticos. - Magister en ingeniería de software. (Universidad de las Fuerzas Armadas de Ecuador)

Para garantizar que las preguntas satisfagan a la subcaracterística respectiva, se realiza una evaluación subjetiva del modelo según sus tres categorías. Los resultados obtenidos en la Tabla 15 se relacionan con la satisfacción e idoneidad de cada pregunta del modelo propuesto según las fase de la metodología SWIRL, con la finalidad de identificar si las preguntas planteadas cumplen con las características de una metodología de desarrollo adecuada para aplicaciones web y el modelo de evaluación es útil.

Tabla 15 Validación de modelos por fases.

PLANTILLA DE VALIDACIÓN DE MODELOS POR FASES.																		
Indicadores	Fase 1			Fase 2			Fase 3			Fase 4			Fase 5			Fase 6		
	Modelo Básico	Modelo Intermedio	Modelo Riguroso	Modelo Básico	Modelo Intermedio	Modelo Riguroso	Modelo Básico	Modelo Intermedio	Modelo Riguroso	Modelo Básico	Modelo Intermedio	Modelo Riguroso	Modelo Básico	Modelo Intermedio	Modelo Riguroso	Modelo Básico	Modelo Intermedio	Modelo Riguroso
Las preguntas son adecuadas para el grado de complejidad (Modelo).	5	4	5	5	5	5	5	4	5	5	4	4	5	4	5	5	3	4
Las preguntas son sólidas y claras para la valoración de aplicaciones web.	5	4	5	5	5	5	5	5	4	5	5	5	5	5	4	5	5	5
Las preguntas son fáciles de entender para cualquier tipo de persona.	5	4	4	5	3	3	5	4	3	3	4	3	4	3	3	5	4	3
Las preguntas cumplen con todas las actividades de la fase.	3	4	5	5	5	5	4	5	4	4	5	5	4	4	5	5	4	4
Se consideran todas las características propias de una aplicación web.	4	5	5	4	4	4	4	4	4	5	4	4	4	5	5	5	4	4
Las preguntas abordan las necesidades de cada fase.	4	4	5	4	4	5	4	5	5	4	5	5	5	5	5	5	4	5
TOTAL	4	4	5	5	4	5	5	5	4	4	5	4	5	4	5	5	4	4

Para calcular los resultados de la evaluación del modelo, cada evaluador asigna un valor del 1 al 5 según la escala de Likert presentada en la Tabla 166, indicando el grado de satisfacción de cada modelo en las seis fases de la metodología SWIRL.

Tabla 16 Valoración en validación del modelo.

ESCALA DE LIKERT	
1	Totalmente en desacuerdo.
2	En desacuerdo.
3	Ni en acuerdo ni en desacuerdo.
4	De acuerdo.
5	Totalmente de acuerdo.

Una vez que se evalúan los tres modelos se procede a modificar o mantener las preguntas dentro de cada fase y modelo, acorde al resultado obtenido. Según el promedio obtenido en cada una de las evaluaciones, se puede asegurar que las preguntas dentro de cada subcaracterística satisfacen las necesidades y cumplen con las actividades propuestas por la metodología SWIRL dentro de las seis fases.

Realizada la valoración por fases, se procede a evaluar cada una de las preguntas por subcaracterísticas, este proceso se realiza mediante la validación por expertos, empleando dos métodos de validez.

- El método del índice de congruencia ítem-objetivo de Rovinelli y Hambleton, en donde “cada juez valora si el contenido de cada ítem incluido en el cuestionario refleja los objetivos especificados” [40, p. 1012]. Sin embargo, en este caso se buscará validar que cada pregunta refleje el objetivo de cada subcaracterística, y se empleará la valoración con formato de respuesta tipo Likert, desde “totalmente relevante” hasta “nada relevante”.
- El método de Validez de Contenido de Hernández-Nieto [41], en donde se evalúa el grado de acuerdo de los expertos a cada pregunta mediante la aplicación de una escala de tipo Likert, iguales que el método anterior, para calcular el CVC (Coeficiente de Validez de Contenido) y según su valor modificar o mantener las preguntas.

Una vez realizadas los métodos de validez expresados, se obtuvieron preguntas las cuales no eran adecuadas a las subcaracterísticas, por lo que se procedieron a eliminarlas o modificarlas según el criterio obtenido en cada una. Como resultado de la validación se obtuvo el esquema final del modelo de evaluación en sus tres categorías (básico, intermedio, y riguroso) que se detalla en el apartado de resultados.

Es importante que la validación del modelo involucre no solo el aseguramiento de las preguntas sino también la funcionalidad y los beneficios del uso del modelo por parte de los desarrolladores y directores del proyecto. Es por ello que se propone una encuesta para evaluar el nivel de utilidad del modelo de evaluación para conocer el grado de beneficio de emplear la metodología y aplicar el modelo de evaluación en el proceso de desarrollo establecida en la *Tabla 17*.

Tabla 17 Resultados de la encuesta de nivel de utilidad del modelo.

	INDICADORES	Totalmente en desacuerdo (1)	En desacuerdo (2)	Ni en acuerdo ni en desacuerdo (3)	De acuerdo (4)	Totalmente de acuerdo (5)	Total
1	El modelo ayuda a conocer las carencias del proyecto para tenerlas en cuenta en futuros proyectos.	-	-	-	1	4	96%
2	Aplicar el modelo le ayudó a mejorar la forma en cómo desarrollar proyectos de desarrollo orientados a la web.	-	-	-	-	5	100%
3	Al aplicar el modelo logró adquirir nuevo conocimiento para mejorar el proceso de desarrollo de futuros proyectos.	-	-	-	1	4	96%
4	Aplicaría la metodología SWIRL en proyectos orientados a la web futuros.	-	-	-	2	3	92%
5	Aplicar la metodología SWIRL beneficia el proceso de desarrollo de los proyectos orientados a la web.	-	-	-	1	4	96%
6	Aplicar el modelo para evaluar el software beneficia a mejorar el proceso de desarrollo en cada fase.	-	-	1	2	2	84%
7	Cree que el modelo considera gran parte de las características de los proyectos web.	-	-	-	2	3	92%
8	Cree que el modelo garantiza el cumplimiento de la mayoría de las características de los proyectos web.	-	-	-	3	2	88%

	INDICADORES	Totalmente en desacuerdo (1)	En desacuerdo (2)	Ni en acuerdo ni en desacuerdo (3)	De acuerdo (4)	Totalmente de acuerdo (5)	Total
9	La metodología SWIRL garantiza el cumplimiento de la mayoría de las características de los proyectos web.	-	-	-	1	4	96%

En la *Tabla 17* se muestran los resultados obtenidos en la encuesta a nivel de utilidad del modelo, se evalúa cada indicador según la escala de Likert que va desde *totalmente en desacuerdo (1)* hasta *totalmente de acuerdo (5)*. Para cada indicador se realiza la sumatoria de las respuestas obtenidas por cada evaluador, es decir en el indicador 1, cuatro evaluadores la calificaron como *totalmente de acuerdo*, mientras que un evaluador lo calificó como *de acuerdo*.

Una vez realizada la sumatoria de las respuestas de cada evaluador, se realiza la ponderación total de cada indicador, para ello se considera la medición de cada opción según la escala de Likert, y se lo divide para 25 debido a que son en total 5 evaluadores. Es decir, en el primer indicador son 4 evaluadores que evaluaron con 5 (=20) y un evaluador que calificó con 4 (=4), lo que da un porcentaje del 96% ($24 \cdot 100 / 25$).

Según los resultados de la encuesta se logra concluir que el modelo de evaluación propuesto cumple con las expectativas de los evaluadores, siendo considerado útil dentro del proceso de desarrollo de aplicaciones.

6. APLICACIÓN WEB PARA LA EVALUACIÓN DE PROYECTOS ORIENTADOS A LA WEB

En este apartado se explica el proceso de desarrollo de la aplicación web, describiendo cada uno de los procesos inmersos en las fases de la metodología SWIRL, garantizando el desarrollo óptimo de la aplicación web.

6.1. Metodología de desarrollo de la aplicación.

El segundo objetivo propuesto dentro del trabajo aquí presentado se basa en la implementación de una aplicación web que permita evaluar proyectos basados en la web mediante el uso del modelo de evaluación planteado con anterioridad. Para la realización de este proyecto se ha llevado a cabo con la metodología de desarrollo SWIRL, en la cual también se basa el modelo de desarrollo.

Como ya se ha comentado anteriormente, la metodología SWIRL [38] se basa en el desarrollo de aplicaciones web mediante la fusión del enfoque híbrido e iterativo, efectuando iteraciones dentro de cada una de sus fases simulando un remolino. Se compone de seis fases, tal como se ha descrito en la sección anterior: análisis, planificación, modelado, implementación, revisiones y pruebas, lanzamiento y marketing.

Este tipo de metodología emplea la planificación adaptativa permitiendo respuestas rápidas y flexibles ante el cambio, además de proporcionar actividades adecuadas acorde al proyecto a elaborar, garantizando el desarrollo óptimo del proyecto al cumplir con los objetivos y requerimientos planteados.

6.2. Fase 1: Análisis del sistema.

Dentro de esta fase se describe el funcionamiento de la aplicación, los requerimientos y la identificación de interesados (individuos y organizaciones que participan activamente en el proyecto o cuyos intereses pueden verse afectados positiva o negativamente como resultado de la ejecución o de la finalización del mismo).

6.2.1. Iteración 1: Definición del sistema

La aplicación a realizar se basa en la implementación del modelo de evaluación para valorar otros proyectos basados en la web, mediante la selección de un grado de

rigurosidad y la evaluación del proyecto acorde las actividades que se efectúan durante el proceso de desarrollo.

El sistema se enfoca en mayor escala a los analistas del proyecto y a los desarrolladores independientes, con la finalidad de estimar si el producto final se encuentra correctamente desarrollado utilizando el modelo de evaluación de la metodología SWIRL.

Actores: los actores definidos para el producto son dos, el usuario común y el usuario registrado.

En la *Tabla 18* se detalla la plantilla para descripción del usuario común, el cual es considerado como los usuarios no registrados, que tiene acceso solamente a la página principal.

Tabla 18 Plantilla de usuarios: usuario común

Usuario:	US-común
Tipo de usuario:	Suscriptores
Nivel:	Segundo nivel
Tipo de interacción:	Indirecta parcial
Actividades:	Navegar a través de la página web, sin estar previamente registrado.
Contenido de interacción:	Página menú

En la tabla 19 se detalla la plantilla para descripción del usuario registrado, el cual es el que se ha registrado, posee un usuario y contraseña y efectuará las diferentes funcionalidades del sistema.

Tabla 19 Plantilla de usuarios: usuario registrado

Usuario:	US-registrado
Tipo de usuario:	Principal
Nivel:	Primer nivel
Tipo de interacción:	Directa temporal
Actividades:	Registrarse dentro del sistema. Seleccionar modelo de evaluación. Efectuar la evaluación de las aplicaciones.
Contenido de interacción:	Página menú, página de registro, página principal, página de evaluación.

Dentro de la definición de los usuarios del sistema es necesario clasificar el tipo de usuario, nivel al que pertenecen y el tipo de interacción que tendrán con el sistema. El tipo de usuario permite identificar las necesidades que poseen en el sistema, el tipo de usuario *Principals* significa que interactúa y tiene acceso a todas las funcionalidades del sistema, mientras que los *Suscriptores* son los que tienen acceso ocasionalmente al sistema y no administran ningún tipo de información.

El nivel indica el tipo de interacción que poseerán dentro del sistema, los usuarios de *Primer nivel* interactúan directamente con las funcionalidades del sistema; mientras que los usuarios de *Segundo nivel* interactúan indirectamente; es decir, acceden en ocasiones y no administran información.

El tipo de interacción es definido por dos parámetros: el primero asigna la interacción del usuario con el sistema <directa o indirecta>, mientras que el segundo especifica el tiempo de interacción <temporal o parcial>.

Alcance del proyecto:

La aplicación permite la inscripción de diferentes usuarios y la modificación posterior de sus datos. Un usuario que no se encuentra registrado solo será capaz de visualizar la página inicial, en la cual se describe la metodología SWIRL; mientras que un usuario al registrarse e ingresar al sistema puede agregar nuevas evaluaciones de proyecto.

Cada proyecto debe contar con su nombre, descripción, y selección de un modelo de evaluación regido por un grado específico de rigurosidad. La aplicación debe ser capaz de evaluar cada una de las fases de la metodología, permitiendo al usuario seleccionar el grado de cumplimiento de cada actividad o pregunta asignada por subprocesos. Sin embargo, el proyecto no puede ser modificado en cuanto a las preguntas evaluadas y al modelo de evaluación ya seleccionado.

Factibilidad del proyecto:

El análisis de factibilidad del sistema se realiza en dos secciones: la factibilidad operativa, identificando los recursos necesarios para el manejo del sistema por parte del usuario final; y la factibilidad técnica, considerando los recursos de equipos y programas empleados durante el transcurso del proceso de desarrollo.

En cuanto al análisis de factibilidad operativa, el único requerimiento solicitado para el usuario final registrado es el conocimiento base de conceptos de ingeniería de software, puesto que los indicadores del modelo de evaluación hacen referencia a las actividades llevadas a cabo durante el proceso de desarrollo.

Para la factibilidad técnica, se consideran como recursos de equipamiento hardware y software necesarios los siguientes, representados en la tabla 20:

Tabla 20 Factibilidad técnica del proyecto

Recursos de equipos (hardware)		Recursos de programas (software)	
Cant.	Descripción	Cant.	Descripción
1	Computador portátil.	1	Star UML
		1	Balsamiq Mockups
		1	Visual code

Perspectiva del producto:

La perspectiva principal de la aplicación se basa en valorar la funcionalidad de otros proyectos de desarrollo web creados, tomando como base el proceso de desarrollo de la metodología SWIRL, de manera que cada proyecto sea evaluado en funcionalidad del grado de rigurosidad que el usuario desea, los cuales oscilan entre : básico, intermedio, y riguroso.

Sin embargo, los modelos no pueden ser modificados una vez que se encuentra registrado el proyecto; es decir, el grado de rigurosidad de evaluación se define al inicio, sin posibilidades de cambio en el transcurso del proceso.

Funcionalidad del sistema:

La aplicación cuenta con dos módulos: el primero se basa en una sección informativa sobre la metodología, el programa y las etapas que esta propone; mientras que, el segundo, se basa en la evaluación de los proyectos mediante el modelo de evaluación de programas propuesto.

Módulo 1: consta de una página informativa en la cual se plantean las fases que componen la metodología SWIRL y la importancia de su uso. Esta sección es accesible para todos los tipos de usuario: el usuario común y el usuario registrado.

Módulo 2: este módulo solo es accesible para los usuarios que se han registrado o poseen una cuenta dentro de la aplicación. Se divide en dos: una sección para el perfil del usuario y otra para visualizar las evaluaciones realizadas anteriormente. Para iniciar una nueva evaluación, el usuario debe registrar el nombre, link (opcional), y descripción del proyecto, y seleccionar un modelo de evaluación el cual se encuentra clasificado debido al grado de rigurosidad con el que se desea valorar el sistema.

A continuación, el usuario procede a calificar al proyecto en base a las fases que cuenta la metodología SWIRL y a los indicadores que se establecieron en el modelo de evaluación seleccionado. En caso de que la evaluación se posponga, la calificación se almacena para luego continuar con las actividades que restan por cumplir.

Al finalizar la valoración de cada etapa de la metodología, se pueden visualizar mediante un gráfico de barras los resultados obtenidos dentro de la evaluación. Estos resultados son:

- Los obtenidos por fases.
- Los obtenidos por subprocesos.
- Los obtenidos por actividades.

Análisis de riesgos:

Los posibles riesgos tasados durante el desarrollo de la aplicación web se presentan en etapas tempranas, considerando como principales riesgos la interfaz poco comprensible y los errores de valoración dentro del código fuente. A continuación, se detallan los riesgos identificados.

La interfaz de usuario es un punto clave para evaluar la calidad del sistema final, ya que existen muchos proyectos que han fracasado por el hecho de poseer una interfaz poco amigable con el usuario final (difícil de entender o que cambia sustancialmente su forma habitual de trabajo). El diseño eficiente de las interfaces es de gran importancia dentro del proyecto; es por ello que los riesgos relacionados pueden ocurrir en dos etapas del proyecto: durante la fase de modelado, en donde la probabilidad de que se diseñen mal las interfaces es alta, dando como resultado un impacto negativo alto en el proyecto; y en la fase de pruebas y revisión, en donde la probabilidad de que las interfaces sean ineficientes para el usuario son menor, generando un leve impacto debido a que en esta fase pueden ser rediseñadas.

En la tabla 21 se describe el riesgo de diseño ineficiente de interfaz de usuario, detallando las etapas en las que puede suceder, la categoría, la actividad que lo detona, la probabilidad, y el impacto dentro del proyecto.

Tabla 21 Plantilla de riesgos: riesgo de interfaz

Riesgo No:	RG-1	Descripción:	Diseño ineficiente de interfaz de usuario	
Etapas	Categoría	Actividad	Probabilidad	Impacto
Modelado	Diseño	Diseñar prototipos de interfaces.	45%	0.40
Pruebas y revisión	Diseño	Falta de entendimiento de las funcionalidades por parte del usuario.	20%	0.27

Otro riesgo posible dentro del proyecto, es implementar inadecuadamente el modelo de evaluación dentro del código fuente. Esto se debe a errores dentro de la programación y la probabilidad de que exista un error dentro de esta etapa es del 50% (es decir muy probable), y el impacto que generaría dentro del proyecto sería significativo debido a que alteraría los resultados que el usuario final espera. Dicho análisis se representa en la tabla 22.

Tabla 22 Plantilla de riesgos: riesgo de valoración en resultados

Riesgo No:	RG-2	Descripción:	Valoración inadecuada de los modelos de evaluación.	
Etapas	Categoría	Actividad	Probabilidad	Impacto
Implementación	Desarrollo	Implementar dentro del código fuente una valoración de las fases de manera errónea.	50%	0.50

Un riesgo vinculado al anterior es la asignación incorrecta de indicadores del modelo de evaluación; es decir, presentar indicadores que no pertenecen a una subcaracterística. Dicho riesgo, se detalla en la plantilla de la tabla 23.

Tabla 23 Plantilla de riesgos: asignación errónea de indicadores

Riesgo No:	RG-3	Descripción:	Asignación incorrecta de indicadores para los modelos.	
Etapas	Categoría	Actividad	Probabilidad	Impacto
Implementación	Desarrollo	Presentar indicadores impropios del modelo.	50%	0.50

Finalmente, el riesgo que mayor impacto posee dentro del sistema es la gestión con la base de datos, el cual se detalla en la tabla 24. Es imprescindible que la conexión con la base de datos sea estable una vez lanzado el producto, ya que generaría un impacto negativo al proyecto. Sin embargo, la probabilidad que se presente este error es del 40% debido a que la conexión es verificada en cada sesión de pruebas y en cuando se agrega una nueva funcionalidad al sistema.

Tabla 24 Plantilla de riesgos: riesgo de error de conexión con base de datos

Riesgo No:	RG-4	Descripción:	Conexión fallida con la base de datos.	
Etapa	Categoría	Actividad	Probabilidad	Impacto
Implementación	Desarrollo	Que no se logren guardar los resultados dentro de la base de datos	40%	0.60

En las tablas anteriores se evalúa la probabilidad e impacto de cada uno de los riesgos de manera subjetiva, según el análisis realizado con referencia al proyecto. A continuación, se establece el análisis cuantitativo y cualitativo de los mismos en la tabla 25. Para la *Magnitud* se consideran los siguientes rangos según el *Impacto*: entre 0 a 0.24 (Muy Leve); 0.25 – 0.49 (Leve); 0.50 – 0.74 (Moderado); 0.75 – 1 (Severo), y para la *Prioridad* se consideran los siguientes rangos según la *Probabilidad*: entre 0 – 24% (Muy baja); 25% - 49% (Baja); 50% - 74% (Alta); 75% - 100% (Muy Alta).

Tabla 25 Análisis cuantitativo y cualitativo de riesgos

Riesgo No	Análisis Cuantitativo		Análisis Cualitativo	
	Probabilidad	Impacto	Magnitud	Prioridad
RG-1	45%	0.40	Leve	Baja
RG-2	50%	0.50	Moderado	Alta
RG-3	50%	0.50	Moderado	Alta
RG-4	40%	0.60	Moderado	Alta

6.2.2. Iteración 2: Modelado de negocio.

El modelo de negocio por el cual se rige la aplicación se basa en el proceso de evaluación que el usuario efectúa dentro del sistema, según se observa en el gráfico 2.

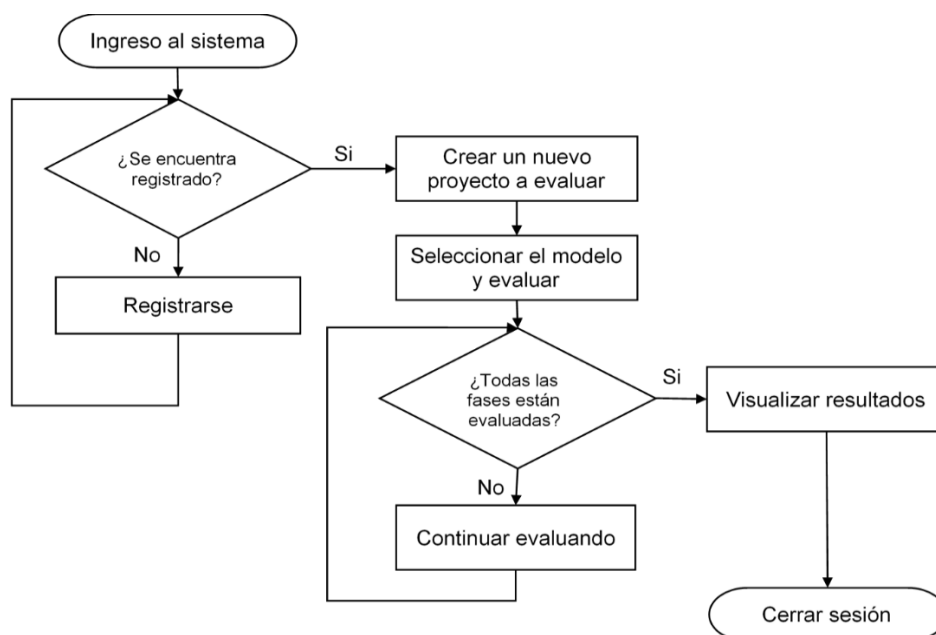


Gráfico 2 Modelo de negocio

El modelo de negocio consiste en el ingreso de usuarios registrados. En caso de no poseer una cuenta dentro del sistema el usuario debe registrarse y crear una. Una vez que se encuentra dentro del sistema debe proceder a crear un nuevo proyecto y seleccionar el grado de rigurosidad bajo el cual lo quiere evaluar. A continuación, evalúa cada una de las actividades por subprocesos de cada etapa que presenta la metodología SWIRL. En caso de que las fases se encuentren totalmente evaluadas se pueden visualizar los resultados finales. Caso contrario, se debe seguir evaluando. Sin embargo, los resultados se pueden visualizar sin necesidad de completar todas las fases, llevando un control del progreso de la evaluación y pudiendo continuar posteriormente.

En cuanto a las reglas de negocio consideradas para la implementación del sistema, se plantean en forma de matriz en la *Tabla 26*.

Tabla 26 Matriz de reglas de negocio

Regla	Condiciones	Acciones
Registro de un nuevo usuario	El usuario no debe estar registrado dentro de la base de datos	Completar el formulario inicial. Ingresar con usuario y contraseña
Creación de un nuevo proyecto de evaluación.	Se debe seleccionar un modelo. No se puede modificar el modelo seleccionado inicialmente.	Completar el formulario de datos del proyecto. Seleccionar un modelo de evaluación.
Evaluación de las actividades realizadas por subprocesos.	La valoración se encuentra dentro el rango establecido. Se deben evaluar todos los indicadores propuestos en el modelo.	Seleccionar la fase a evaluar. Seleccionar el subproceso Evaluar cada una de las actividades del proceso.
Visualización de resultados de la evaluación.	Se debe completar al menos una fase o subproceso	Evaluar un proceso o fase por completo. Visualizar los resultados en un gráfico de barras.

6.2.3. Iteración 3: Identificación de interesados.

Una parte importante para la ejecución del proyecto es la identificación de los interesados y los roles que cada uno de ellos cumple. SWIRL propone una identificación por fases; es decir, se establecen los roles y los involucrados en cada etapa del proceso de desarrollo, estableciendo las siguientes matrices representadas en las tablas 27 a 32.

Fase 1: Análisis

Tabla 27 Matriz general de interesados: fase de análisis

MATRIZ GENERAL DE INTERESADOS		
Nombre	Rol	Responsabilidades
Jimmy Molina Ríos	Analista de requerimientos	Identificar los requerimientos del sistema.
Nieves Pedreira Souto	Director del proyecto	Validar el alcance y los objetivos del proyecto.

Fase 2: Planificación

Tabla 28 Matriz general de interesados: fase de planificación

MATRIZ GENERAL DE INTERESADOS		
Nombre	Rol	Responsabilidades
Jimmy Molina Ríos	Responsable de calidad.	Realizar el cronograma. Identificar la gestión de riesgos y comunicaciones. Establecer la gestión de calidad.

Fase 3: Modelado

Tabla 29 Matriz general de interesados: fase de modelado

MATRIZ GENERAL DE INTERESADOS		
Nombre	Rol	Responsabilidades
Jimmy Molina Ríos	Diseñador	Realizar los modelos de interfaz gráficas. Diseñar los modelos conceptuales y de navegación del proyecto.

Fase 4: Implementación

Tabla 30 Matriz general de interesados: fase de implementación

MATRIZ GENERAL DE INTERESADOS		
Nombre	Rol	Responsabilidades
Jimmy Molina Ríos	Desarrollador	Implementar y codificar la aplicación web. Crear prototipos de la aplicación.

Fase 5: Revisión y pruebas.

Tabla 31 Matriz general de interesados: fase de revisión y pruebas

MATRIZ GENERAL DE INTERESADOS		
Nombre	Rol	Responsabilidades
Jimmy Molina Ríos	Tester	Realizar pruebas de los prototipos presentados.
Nieves Pedreira Souto	Director del proyecto	Revisar y validar los prototipos. Verificar el cumplimiento de los requerimientos.

Fase 6: Lanzamiento y marketing.

Tabla 32 Matriz general de interesados: fase de revisión y pruebas

MATRIZ GENERAL DE INTERESADOS		
Nombre	Rol	Responsabilidades
Jimmy Molina Ríos	Desarrollador	Almacenar la aplicación en un host. Detallar la documentación y presentar el producto.

6.2.4. Iteración 4: Requerimientos

Dentro de esta sección se detallan los requerimientos funcionales y no funcionales establecidos para el producto, empleando la plantilla general de desglose, tablas 33 y 34 respectivamente. Además, se crea la matriz evento-respuesta para plasmar los requisitos de usuario dentro del proyecto, que se expone en la tabla 35. Finalmente, de manera breve se listan los requerimientos de navegación, los cuales son un punto clave dentro de la metodología SWIRL.

Requerimientos funcionales.

Tabla 33 Requerimientos funcionales de la aplicación.

REQUERIMIENTOS FUNCIONALES	
Código	Descripción
RF-01	El usuario debe ser validado para ingresar al sistema, es decir, debe estar registrado (tener una cuenta).
RF-02	Se deben poder crear nuevos proyectos de evaluación y vincularlos a un modelo de evaluación.
RF-03	Se puede modificar la descripción y nombre del proyecto, sin embargo, el modelo y las valoraciones ya realizadas no pueden ser cambiadas.
RF-04	Los resultados se presentan mediante un gráfico de barras al finalizar cada fase y subprocesos respectivamente.
RF-05	La evaluación no posee un orden de seguimiento específico pueden valorar fases iniciales o intermedias como primer punto.
RF-06	Las evaluaciones no pueden ser eliminadas ni modificadas cuando su valoración ha sido completada.
RF-07	El usuario debe ser capaz de modificar sus datos personales.
RF-08	Se debe mostrar un botón de ayuda para guiar al usuario durante la evaluación de los proyectos.

Requerimientos no funcionales.

Tabla 34 Requerimientos no funcionales de la aplicación.

REQUERIMIENTOS NO FUNCIONALES	
Código	Descripción
RNF-01	Facilidad de uso: la interfaz debe ser intuitiva para la diversidad de usuarios finales.
RNF-02	Seguridad: la aplicación debe asegurar el respaldo de los datos y las evaluaciones realizadas a las diversas aplicaciones.
RNF-03	Compatibilidad: es capaz de ser ejecutada en los navegadores Google Chrome y Mozilla Firefox principalmente.

Requerimientos de usuario.

Tabla 35 Requerimientos de usuario de la aplicación.

Evento	Actividad	Respuesta
Registro	El usuario crea una nueva cuenta.	Retorno de página de creación de cuenta.
Crear nuevo proyecto.	El usuario crea un nuevo proyecto. Selecciona el modelo de evaluación que desea aplicar.	Retorno de página de creación y formulario con datos básicos.
Evaluar una fase.	El usuario selecciona el proyecto, y da clic en evaluar. Procede a seleccionar una fase, y comienza la evaluación de los subprocesos. Cada actividad considerada es valorada en la escala de Likert.	Retorno de página con actividades y calificación de cada una de ellas.
Modificar datos del perfil.	El usuario selecciona la opción perfil, con la cual visualiza los datos ingresados y es capaz de modificar los mismos.	Retorno de página con datos del perfil. Retorno de formulario para editar datos.
Ayuda	El usuario selecciona el botón de ayuda en la página principal.	Se visualiza el manual de usuario y un video explicativo para el uso de la aplicación.

Requerimientos de navegación.

- La página inicial se debe vincular mediante enlace al libro web de la metodología SWIRL.
- Las páginas de visualización de perfil y evaluaciones deben ser presenciadas en la barra lateral de la página principal.
- Los formularios de edición y registro de nueva información deben ser presentados de manera sobrepuesta a la página en la que se encuentra.
- Los comentarios de validación son presentados en la parte superior derecha de la página.
- La opción de visualizar debe permitir acceder a la página de resultados de los procesos del proyecto de evaluación establecidos conjuntamente con los datos del mismo.

Una vez que se han detallado los requerimientos, se procede a priorizarlos de manera jerárquica en la *Tabla 36*. Para ello, se consideran tres parámetros: *Alto* (se deben diseñar primero o son totalmente necesarios en el proyecto); *Medio* (su diseño durante el proceso de desarrollo y son dependientes de otros requerimientos) y *Bajo* (su diseño no afecta gravemente el desarrollo del proyecto ante el usuario final, ni su funcionalidad principal, y pueden, o no ser dependientes de otros requerimientos).

Tabla 36 Matriz de priorización de requerimientos.

Código	Descripción	Indicador de priorización
RF-01	Registro de usuario	Alto
RF-02	Creación de nuevos proyectos.	Alto.
RF-03	Modificación de proyecto.	Medio.
RF-04	Visualización de resultados.	Alto.
RF-05	Evaluación.	Alto
RF-06	Modificación o eliminación de proyectos.	Medio.
RF-07	Modificación del perfil del usuario.	Bajo.
RF-08	Ayuda para gestionar las funcionalidades del sistema.	Bajo

6.3. Fase 2: Planificación.

En esta fase se describe la planificación del proyecto, al igual que las historias de usuario, los riesgos y la estructura de desglose de trabajo.

6.3.1. Iteración 1: Historias de usuario.

Para la descomposición de las historias de usuario, se hace uso del método SPIDR [42], definiendo de esta manera las siguientes historias de usuario con mayor relevancia, tal como se aprecia en las tablas 37 a 40, para cuatro usuarios ejemplo (usuarios 1, 2, 3 y 4).

Tabla 37 Historia de usuario: 1

Historia de Usuario	
Número: 1	Usuario: Usuario común.
Nombre de historia: Registro de usuario.	
Prioridad: Medio	Riesgo en desarrollo: Bajo.
Requerimiento Funcional: RF-01	Iteración asignada: 2
Descripción: El usuario común debe tener la opción para poder registrarme y crear una cuenta en el sistema.	
Observaciones: Se debe emitir un mensaje de aceptación o denegación de la validación.	

Tabla 38 Historia de usuario: 2

Historia de Usuario	
Número: 2	Usuario: Usuario registrado
Nombre de historia: Creación de nuevo proyecto.	
Prioridad: Alto	Riesgo en desarrollo: Medio.
Requerimiento Funcional: RF-02	Iteración asignada: 1
Descripción: El usuario registrado puede crear un nuevo proyecto, en donde seleccione el modelo de evaluación con el cual se va a valorar.	
Observaciones: Solo se puede seleccionar un modelo por proyecto.	

Tabla 39 Historia de usuario: 3

Historia de Usuario	
Número: 3	Usuario: Usuario registrado
Nombre de historia: Visualizar resultados.	
Prioridad: Alto	Riesgo en desarrollo: Alto.
Requerimiento Funcional: RF-04	Iteración asignada: 1
Descripción: El usuario debe ser capaz de visualizar los resultados de las evaluaciones del proyecto en cada una de las etapas, mediante gráficos de barras.	
Observaciones: Ninguna.	

Tabla 40 Historia de usuario: 4

Historia de Usuario	
Número: 4	Usuario: Usuario registrado
Nombre de historia: Evaluación	
Prioridad: Medio	Riesgo en desarrollo: Alto.
Requerimiento Funcional: RF-05	Iteración asignada: 2
Descripción: El usuario puede evaluar cada fase del proyecto según desee, sin un orden específico.	
Observaciones: La evaluación puede ser reanudada después, sin embargo, las fases ya evaluadas no se pueden modificar.	

6.4. Fase 3: Modelado.

Dentro de la fase de modelado se define: el modelo conceptual, el modelo de navegación, la arquitectura del sistema y el diseño de la interfaz abstracta de usuario.

6.4.1. Iteración 1: Modelo Conceptual.

A continuación, se presenta el **Gráfico 3** del modelo conceptual, indicando los campos a emplear dentro de la base de datos con la finalidad de cubrir los requerimientos propuestos.

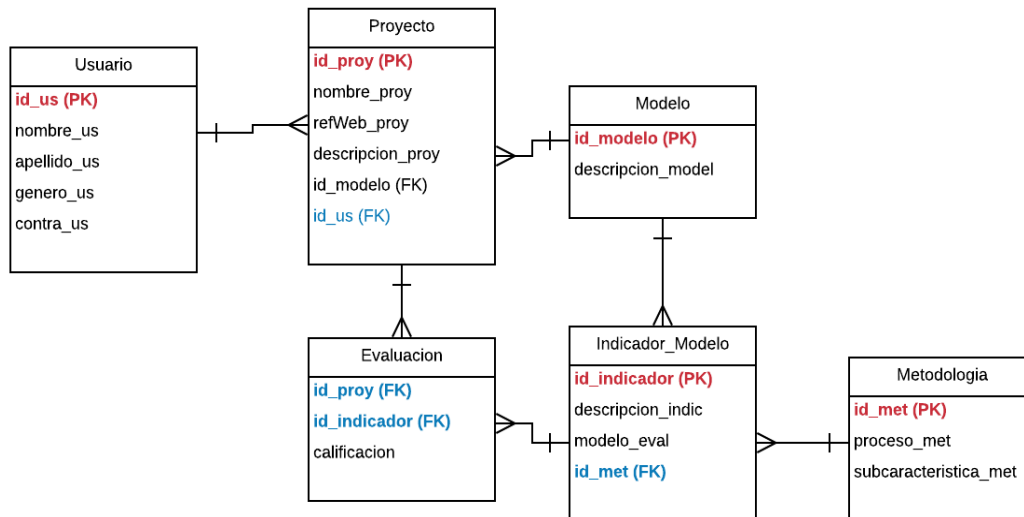


Gráfico 3 Modelo conceptual

6.4.2. Iteración 2: Modelo de navegación.

Es importante definir el modelo de navegación que posee el sistema, como se puede observar en el Gráfico 4, indicando las funcionalidades y módulos que tendrá, y que tipos de usuarios podrán acceder a los mismos.

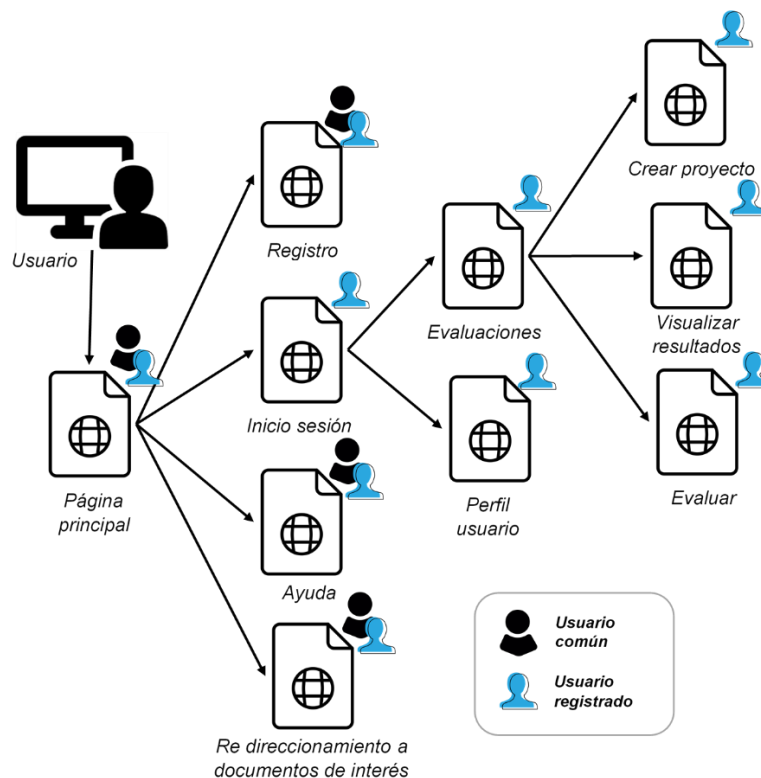


Gráfico 4 Modelo de navegación.

6.4.3. Iteración 3: Arquitectura del sistema.

La arquitectura general de la aplicación web se basa en cliente-servidor, compuesta por dos elementos, el servidor web (back-end), y el cliente (front-end) como se muestra en el Gráfico 5, favoreciendo la separación de responsabilidades para ambos elementos en las funcionalidades del sistema, y permitiendo el acceso a la página por parte de varios usuarios.

- **Servidor web:** Dentro de este se almacenan los servicios de la aplicación y el tratamiento de la información que es ofrecida al usuario mediante la interfaz. La estructura interna utilizada en el servidor es por capas, favoreciendo la separación de actividades designada a cada una de ellas. Por ejemplo, en la capa de acceso a datos se realiza la comunicación con la base de datos; mientras que en la capa de negocio se implementan las funcionalidades del sistema. Finalmente, en la capa de servicios se almacena la interfaz de usuario, gestionando las peticiones que realiza el cliente.
- **Cliente web:** Se basa en la presentación de la interfaz de usuario y la interacción del usuario con el sistema mediante peticiones al servidor.

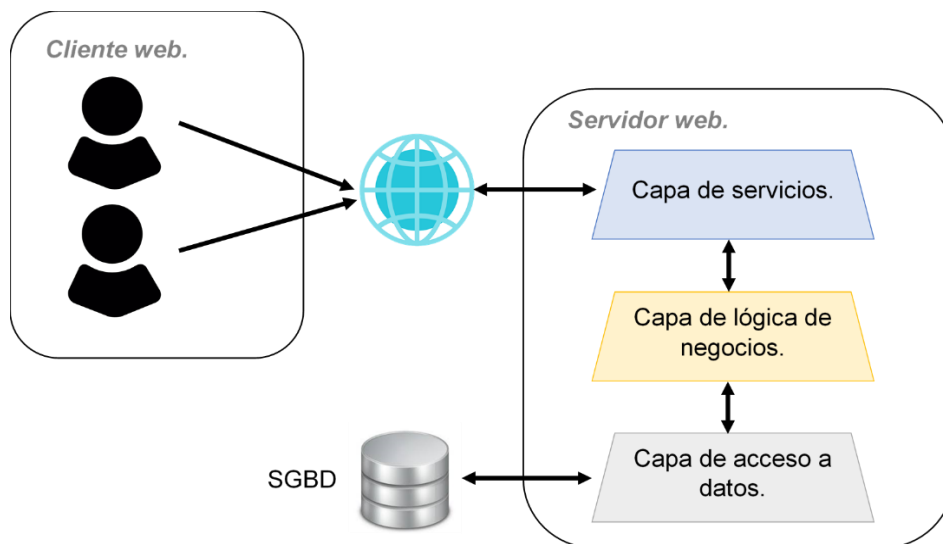


Gráfico 5 Arquitectura del sistema.

6.4.4. Iteración 4: Diseño de interfaz abstracta de usuario.

En esta sección se pretende detallar las funcionalidades y descripción que contiene cada una de las interfaces de usuario diseñadas para la aplicación web. Las páginas diseñadas son un prototipo de lo esperado en la aplicación web, y puede contener una mínima variación con la diseñada en el producto final; sin embargo, las funcionalidades son las mismas.

La pantalla principal de la aplicación se puede visualizar en la Figura 1, la cual es accesible a todo tipo de usuarios, y presenta información relevante sobre la metodología SWIRL y el modelo de evaluación propuesto. En una sección se especifican artículos de interés realizados por el equipo de trabajo relacionados con las metodologías de desarrollo web, y en otra, un enlace al libro electrónico con las fases de la metodología.

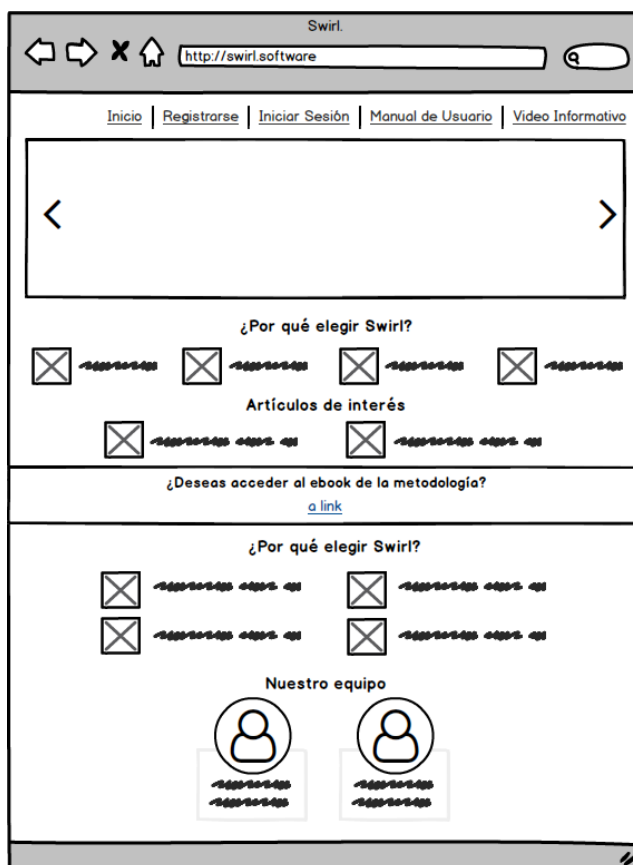


Figura 1 Página principal de la aplicación.

Los usuarios pueden realizar tres acciones dentro de esta página, la primera consiste en crear una nueva cuenta mediante la opción *Registrarse* (como se muestra en la Figura 2), y en caso de ser un usuario registrado puede *Iniciar Sesión* (como se

muestra en la Figura 3), finalmente, se cuenta con el botón de *Ayuda* que permite al usuario visualizar un manual de guía para evaluar los proyectos.

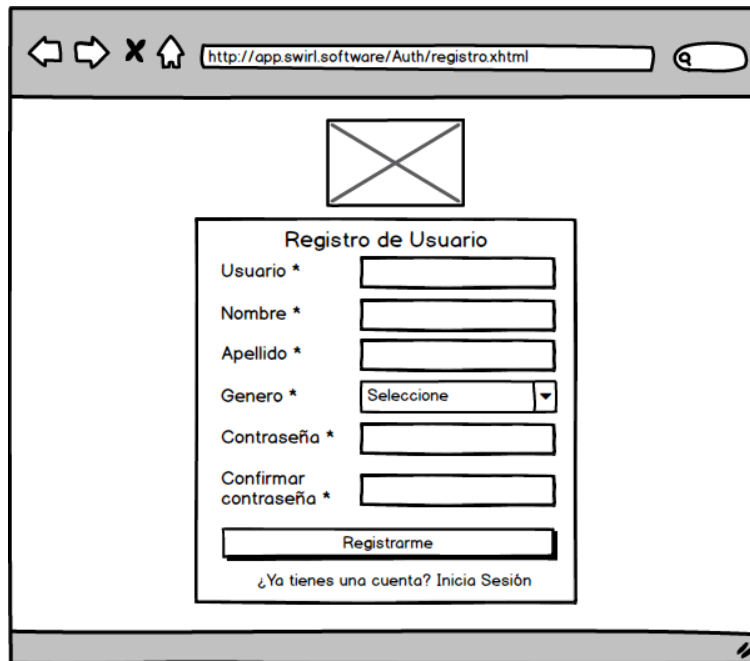


Figura 2 Página de registro de usuario

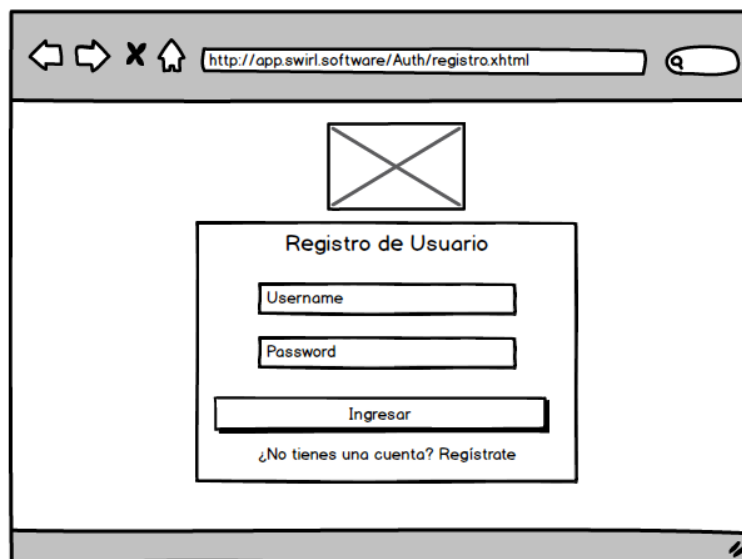


Figura 3 Página de inicio de sesión

Una vez que el usuario inicie sesión, se habilita la pantalla principal del sistema como se muestra en la Figura 4, en la cual se visualiza un listado de los proyectos creados, y permitiendo la creación de uno nuevo con los datos que se muestran en la Figura 5. El sistema permite al usuario realizar la evaluación del proyecto de manera progresiva y asíncrona; es decir, puede continuar las evaluaciones en otro momento. Para la evaluación de cada proyecto se aplica el botón contenido en acciones, desde el cual se puede: evaluar, ver resultados y modificar el proyecto.

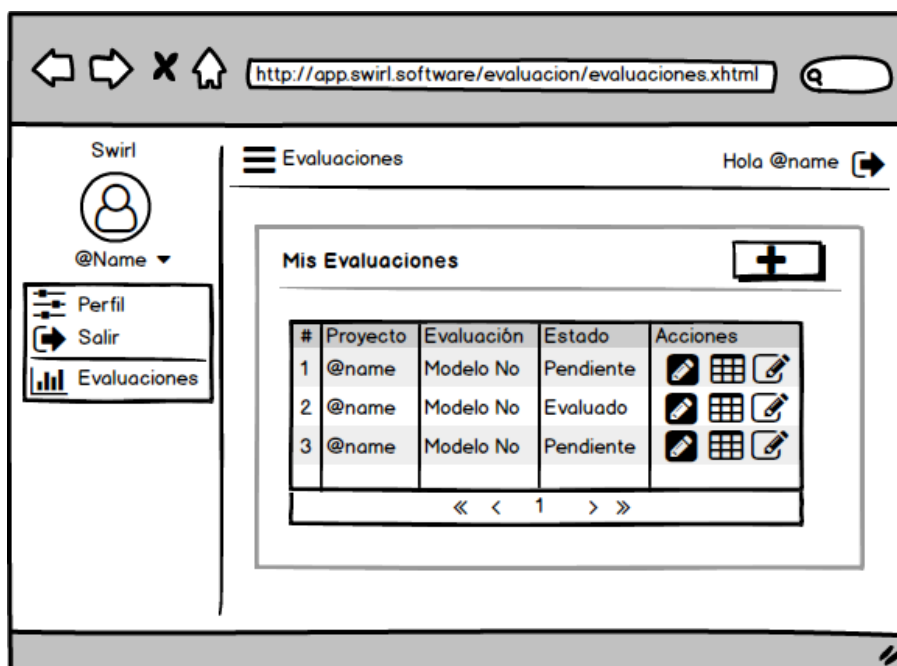


Figura 4 Página de evaluación para usuarios registrados

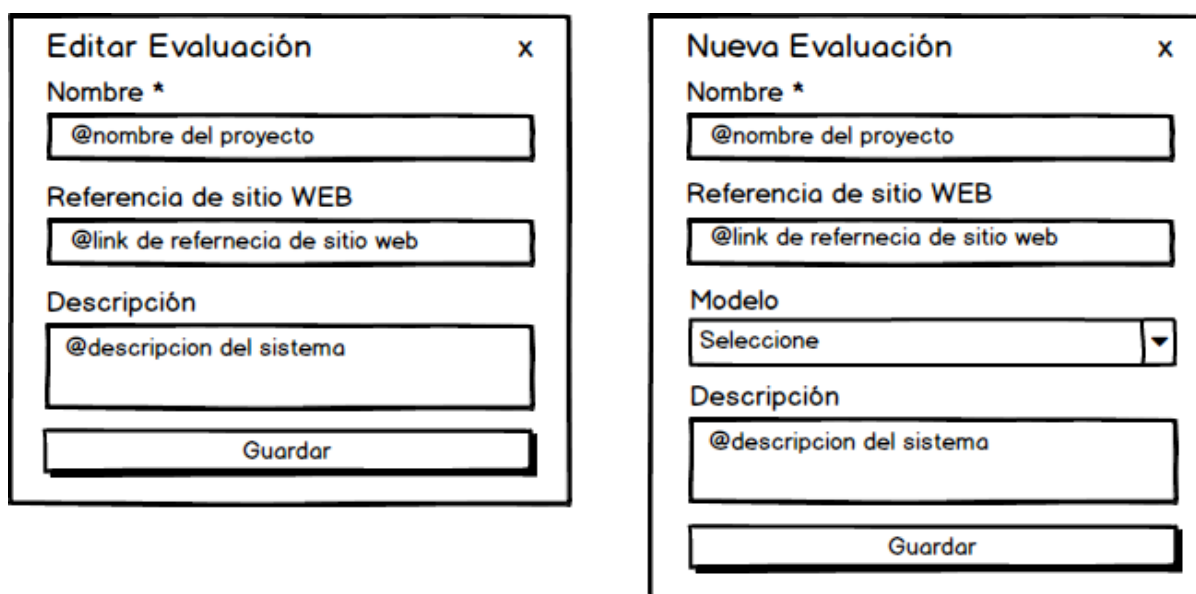


Figura 5 Páginas de creación y edición de evaluación

La opción de evaluar muestra en pantalla los procesos y subcaracterísticas para que el usuario seleccione y proceda con su debida evaluación. Para ello, es importante recalcar que se estipularon indicadores de valoración como se muestran en la Figura 6. Al seleccionar la opción “visualizar resultados”, se muestran los procesos y subcaracterísticas que ya han sido evaluadas con su respectiva gráfica. Los resultados son presentados en tres partes: los resultados generales del proyecto, los resultados por procesos y, finalmente, los resultados por subcaracterísticas como se muestra en la Figura 7.

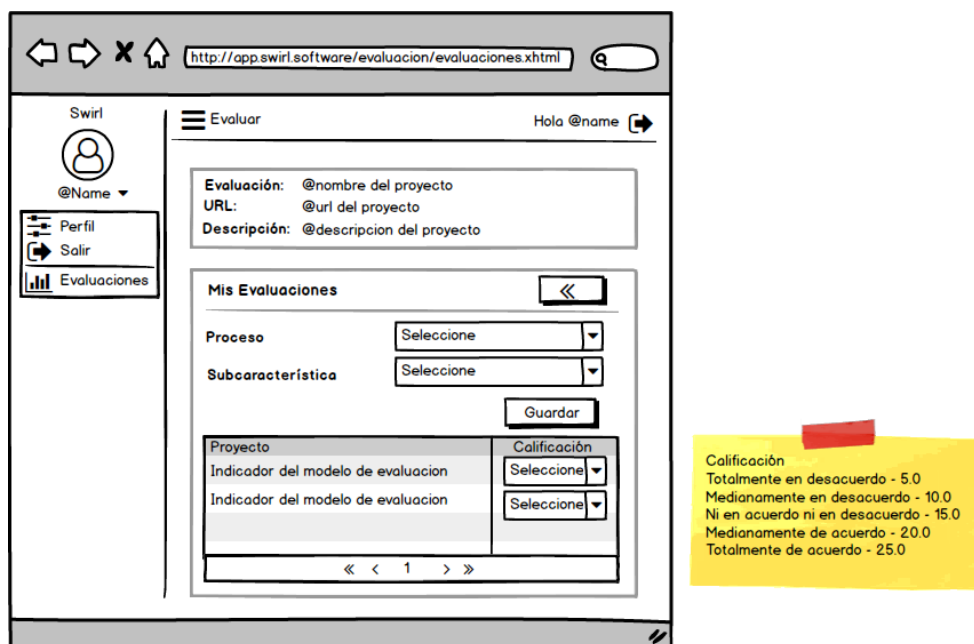


Figura 6 Página de evaluación de procesos y subprocesos.

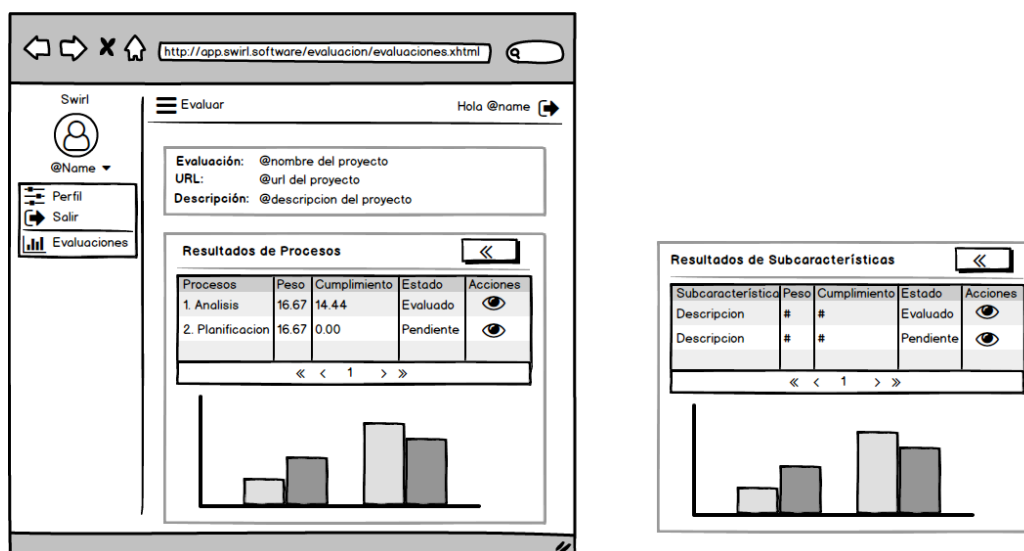


Figura 7 Página de evaluación de procesos y subprocesos (continuación).

6.4.5. Iteración 5: Diagrama de Gantt

Una vez realizada la fase preliminar para conocer el alcance y datos estratégicos del proyecto, se realizó una planificación en diagrama de Gantt, tal como se muestra en el *Gráfico 6*. El trabajo realizado se divide en dos etapas: la primera consiste en el diseño del modelo de evaluación, la cual duró aproximadamente 6 semanas, incluyendo las pruebas piloto y, la segunda, consiste en la planificación de cada una de las fases de la metodología SWIRL: fase de análisis (10 días), fase de planificación (2 semanas), fase de modelado (12 días), fase de implementación (20 días), fase de pruebas y revisión (7 días) y fase de lanzamiento (5 días).

Capítulo 6. Aplicación web para la evaluación de proyectos orientados a la web.

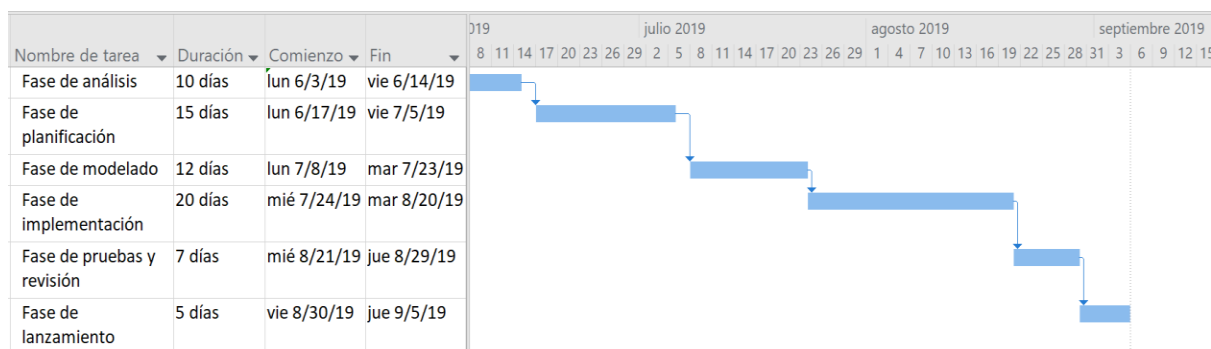


Gráfico 6 Planificación en diagrama de Gantt

6.4.6. Iteración 6: Gestión de riesgos.

Al ser la aplicación web un sistema para clientes no específicos; es decir, no posee un cliente definido, los riesgos que se pueden presentar durante el desarrollo del proyecto son diversos, y llegan a afectar directa e indirectamente al usuario. Sin embargo, se ha definido un plan de prevención de riesgo en el cual se consideran estrategias de prevención adecuadas a cada riesgo, tal como se puede observar en la tabla 41.

Tabla 41 Plan de prevención de riesgos

ID	Riesgo	Actividad detonante	Estrategia de prevención.
RG-1	Diseño ineficiente de interfaz de usuario.	Diseño de prototipos sin considerar todos los requerimientos.	Realizar pruebas de entendimiento y funcionalidad de los prototipos en usuarios aleatorios.
RG-2	Valoración inadecuada de los modelos de evaluación.	Asignar valores o procedimientos incorrectos dentro de las líneas de código.	Realizar un diagrama de flujo de la función de valoración de los modelos, mediante pruebas de escritorio adecuadas.
RG-3	Asignación incorrecta de indicadores para los modelos.	Implementar en el código una incorrecta asignación de las preguntas para cada modelo.	Revisión continua de la asignación de indicadores a los modelos conforme avanza el sistema.
RG-4	Conexión fallida con la base de datos.	Error de conexión con el host, fallo temporal en el gestor de base de datos.	Implementar excepciones y acciones en caso de conexión fallida que aseguren la disponibilidad.

Además, se incluye un plan de mitigación de riesgos, expuesto en la tabla 42, en el cual se proponen estrategias de minimización para los riesgos más potenciales.

Tabla 42 Plan de minimización de riesgos

ID	Riesgo	Estrategia de minimización
RG-1	Diseño ineficiente de interfaz de usuario.	Realizar revisiones y cambios necesarios al finalizar la implementación de una nueva funcionalidad en la aplicación.
RG-2	Valoración inadecuada de los modelos de evaluación.	Realizar pruebas de escritorio de los diferentes modelos de evaluación.
RG-3	Asignación incorrecta de indicadores para los modelos.	Realizar pruebas de escritorio de los diferentes indicadores.
RG-4	Conexión fallida con la base de datos.	Testear todas las funcionalidades que requieran conexión a la base de datos para verificar que sean correctos.

6.4.7. Iteración 7: Gestión de comunicaciones.

Durante la elaboración del proyecto, se realizó el proceso de comunicación con el tutor asignado mediante el correo electrónico debido a la ubicación geográfica. También se realizaron reuniones personales, pero en menor rango.

6.4.8. Iteración 8: Estructura de desglose de trabajo (EDT/WBS)

Para el desglose de trabajo se han empleado tres niveles: el nivel 1 cuenta con los módulos del sistema, mientras el nivel 2 y 3 contienen subprocesos relacionados a cada una de las funcionalidades, tal como puede observarse en el Gráfico 7.

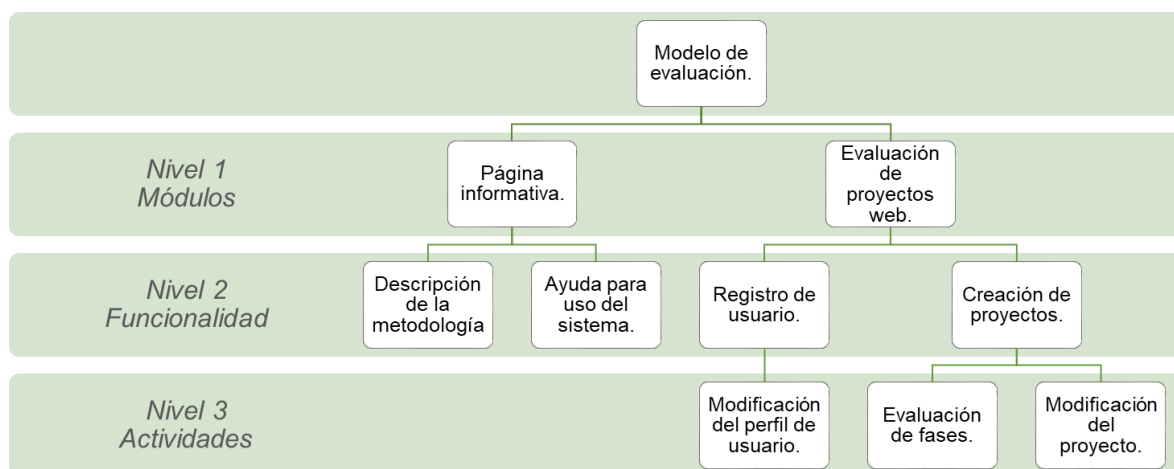


Gráfico 7 EDT/WDS

6.4.9. Iteración 9: Gestión de calidad.

Las actividades especificadas para el plan de aseguramiento y control de calidad son las siguientes:

- Identificar los requerimientos mediante la socialización e interacción con el equipo de desarrollo y el usuario.
- Definir modelos UML que permitan el fácil entendimiento de las funcionalidades y estructura del sistema.
- Realizar pruebas de validación de campos, y conectividad con la base de datos de manera constante, requerido al implementar una nueva funcionalidad.
- Testear los prototipos generados en cada iteración para identificar el cambio realizado.
- Realizar pruebas de ejecución a terceros.

6.5. Fase 4: Implementación.

En esta fase, solo se tiene una única iteración que se denomina diseño del sistema, en la cual se establece la arquitectura tecnológica empleada para implementar el sistema. A continuación, en los siguientes apartados de este punto, se especifican: las plataformas en las cuales se realizó la codificación, el sistema gestor de base de datos y el servidor donde se ejecuta.

Plataformas de programación.

La plataforma de programación bajo la cual se implementó el sistema es Java EE, conjuntamente con la biblioteca de Primefaces, las interfaces de Enterprise JavaBeans(EJB) y Java Persistence API (JPA).

- **Java EE:** es el conjunto de estándares y tecnologías aplicadas netamente al desarrollo de aplicaciones en lenguaje Java del lado del servidor. La plataforma consta de varios servicios, APIs y protocolos para desarrollar aplicaciones basadas en la web.

- **Primefaces:** es una librería de componentes visuales para JSF de código abierto que brinda soporte de Ajax específicamente para aplicaciones web y móviles.
- **Enterprise JavaBeans (EJB):** interfaces de programación de aplicaciones dentro del estándar de construcción JEE. Su uso se enfocó en los objetos que provee desde el lado del servidor.
- **Java Persistence API (JPA):** framework de Java para manipular datos relacionales en aplicaciones mediante Java SE y Java EE. Al mencionar la persistencia se mencionan 3 áreas fundamentales, el primero es la API en general; el segundo es el lenguaje de consulta JPQL y, como último, los metadatos del objeto relacional.

Sistema Gestor de Base de Datos.

La tecnología empleada para gestionar el almacenamiento de la información y los datos es PostgreSQL, un sistema gestor de bases de datos relacionales fácil de emplear y con una compatibilidad mayor a diferentes entornos.

Servidor.

Para el desarrollo del servidor se ha empleado Payara Server, el cual es el más adecuado para la producción de aplicaciones Java EE debido a su flexibilidad de “middleware” para admitir desarrollo continuo. Payara es un servidor de código abierto con soporte de alta calidad e innovador. La aplicación se encuentra puesta en producción dentro de un servidor Linux, con un sistema operativo Debian 8.

6.6. Fase 5: Pruebas y revisión.

Dentro de esta etapa se describe el proceso de revisión del programa empleando la revisión por pares y las pruebas de aceptación realizando las pruebas del modelo de evaluación acorde a los tres niveles: básico, intermedio y riguroso.

La revisión por pares se realiza al final del desarrollo de cada funcionalidad del programa; es decir, se realiza de modo incremental. Mediante esta técnica de revisión, se identifican los errores del programa a tiempo, permitiendo solucionarlos de manera rápida y evitar futuros errores cuando el programa se encuentre

funcionando y sea más complicado, o imposible a un coste asumible, su rectificación. A continuación, se indican los resultados finales de la prueba de pares acorde a cada funcionalidad, según se observa en la tabla 43.

Ingreso a opciones de la página de inicio: En este apartado ingresan todas las características que posee la página de inicio del programa. Se debe validar el diseño de la página de inicio, el redireccionamiento de los botones a enlaces externos, y el correcto funcionamiento de la opción de visualización del manual de usuario y video informativo.

Tabla 43 Revisión de pares: Página de inicio.

Revisión N°	1	RF abordados:	RF-08
Aspectos a considerar:	Diseño intuitivo y amigable en la página de inicio. Redireccionamiento a enlaces externos según botones respectivos. Botón de ayuda para el usuario (manual de usuario y video informativo).		
Proceso:	1. Ingresar a la página de inicio del programa. 2. Revisión del texto inmerso en la página de inicio. 3. Clic en los botones con redireccionamiento externo (artículos de interés, y acceso al libro electrónico de la metodología)		
Resultados esperados:	Acceso correcto a los artículos de interés, al libro electrónico de la metodología SWIRL, y al manual de usuario del programa. Además, de la visualización completa del video informativo.		
Resultados obtenidos:	La prueba se concluyó satisfactoriamente (no posee errores).		

Registro e ingreso al sistema: Se evaluará la validación de campos, el registro de un nuevo usuario e inicio de sesión. El registro se puede realizar mediante dos opciones, el primero es mediante la opción *Registrarse* que se encuentra en la página principal, y el segundo mediante la opción *Regístrate* que se encuentra en la parte inferior del inicio de sesión. De igual manera, se revisa el inicio de sesión de dos formas, la primera mediante la opción *Iniciar Sesión* que se encuentra en la página principal; y la segunda mediante la opción *Inicia Sesión* que se encuentra en la parte inferior del registro de usuario.

En la tabla 44 se describe la revisión por pares del registro e ingreso al sistema.

Tabla 44 Revisión de pares: Registro e ingreso al sistema.

Revisión N°	2	RF abordados:	RF-01
Aspectos a considerar:	Creación de nuevo usuario. Inicio de sesión con usuario registrado. Validación de campos.		
Proceso:	1. Ingresar a la página de inicio de la aplicación y seleccionar la opción <i>Registrarse</i> . 3. Completar los campos, y revisar si existen validaciones. 4. Finalizar el registro y proceder a la opción <i>Inicio de sesión</i> . 5. Iniciar sesión escribiendo el nombre de usuario y contraseña (Verificar que se valide al usuario). Método 2. Verificación de botones. 1. En la página de inicio de la aplicación dar clic en la opción <i>Iniciar Sesión</i> . 2. Seleccionar la opción <i>¿No tienes una cuenta? Regístrate</i> . 3. Seleccionar la opción <i>¿Ya tienes una cuenta? Inicia Sesión</i> .		
Resultados esperados:	Registrar un nuevo usuario exitosamente. Ingresar al sistema mediante un usuario y contraseña válidos. Vincular las funcionalidades de registro e inicio de sesión de manera adecuada.		
Resultados obtenidos:	La prueba se concluyó satisfactoriamente (no posee errores). Los botones de registro e inicio de sesión vinculan correctamente las funcionalidades en los diferentes modos.		

Visualización y modificación del perfil del usuario: Se evalúa que el botón perfil permita visualizar todos los datos personales del usuario, y que se modifiquen los campos a excepción de *Usuario*, tal como se observa en la evaluación de la tabla 45.

Tabla 45 Revisión de pares: Perfil de usuario.

Revisión N°	3	RF abordados:	RF-07
Aspectos a considerar:	Visualización correcta y entendible de los datos personales. Opción de modificación de los campos a excepción de <i>usuario</i> . Guardar los cambios modificados correctamente.		
Proceso:	1. Ingresar al sistema y en la barra lateral seleccionar la opción <i>Perfil</i> . 2. Revisión de todos los datos del usuario (que sean correctos). 3. Modificar el contenido de los campos que lo permitan. 4. Seleccionar el botón <i>Guardar</i> .		
Resultados esperados:	La información personal del usuario debe ser mostrada de manera amigable y de manera correcta según el campo asignado. Se modifican todos los datos personales del usuario, con excepción de <i>Usuario</i> . Al guardar los campos se presenta un mensaje que indica que se efectuó el proceso exitosamente.		

Resultados obtenidos:	<p>La prueba concluyó satisfactoriamente (no se presentaron errores). La información el usuario se presenta de forma entendible y se encuentra ubicada correctamente según el campo indicado. Al editar la información personal el campo <i>Usuario</i> se encuentra bloqueado, por lo cual no se permite su modificación; mientras que los otros campos si se pueden modificar. Al finalizar el proceso y guardar cambios, el mensaje de <i>modificación exitosa</i> aparece en la parte superior de la página.</p>
------------------------------	---

Creación y modificación de un proyecto: Dentro de este punto se evalúa la creación de un nuevo proyecto, y la modificación de algunos campos permitidos dentro de la misma, como se ve en la evaluación de la tabla 46.

Tabla 46 Revisión de pares: Creación y modificación de proyectos.

Revisión N°	4	RF abordados:	RF-2, RF-03, RF-06.
Aspectos a considerar:	<p>Creación de un nuevo proyecto. Modificación del proyecto creado.</p>		
Proceso:	<ol style="list-style-type: none"> 1. Ingresar al sistema y seleccionar la opción <i>Evaluaciones</i>. 2. Dar clic en <i>Nueva Evaluación</i>. 3. Completar los campos solicitados. 4. Seleccionar el modelo de evaluación y guardar. 5. Seleccionar la opción <i>Editar</i> de la evaluación creada, que se encuentra en la columna <i>Acciones</i> en la tabla de evaluación. 6. Modificar los campos permitidos. 7. Guardar los cambios realizados. 		
Resultados esperados:	<p>Se presenta un cuadro de dialogo para ingresar los campos necesarios con respecto al nuevo proyecto; mediante un cuadro de selección se presentan los modelos de evaluación. Cuando se guardan los cambios al evaluar un nuevo proyecto aparece un mensaje indicando que el proceso fue exitoso. En la tabla de evaluaciones se presenta la evaluación creada, en la columna estado aparece el termino <i>Pendiente</i> y en la columna <i>Acciones</i> se presenta la opción de <i>Editar, Eliminar y Evaluar</i>. Se permite modificar todos los datos del proyecto a excepción del modelo seleccionado. Al guardar los cambios aparece un mensaje indicando que el proceso fue exitoso.</p>		
Resultados obtenidos:	<p>La prueba concluyó satisfactoriamente (no se presentaron errores). Se creó un nuevo proyecto, mediante el modelo 1. Al guardar los cambios, el nuevo proyecto aparece en la tabla de evaluaciones y se presentan las acciones de editar, evaluar y eliminar. Posteriormente, se editó algunos campos de los mismos, con excepción del modelo ya seleccionado. Al finalizar aparece un cuadro indicando que el proceso fue exitoso.</p>		

Evaluación de un proyecto y visualización de resultados: En este punto se verifica que el usuario pueda completar la evaluación del proyecto creado dentro de todas las fases. Al finalizar la evaluación en todas las fases, se visualizan los resultados mediante un cuadro de barras, según la evaluación realizada acorde a la tabla 47.

Tabla 47 Revisión de pares: Creación y modificación de proyectos.

Revisión N°	5	RF abordados:	RF-04, RF-05.
Aspectos a considerar:	Evaluación del proyecto. Visualización de los resultados obtenidos al finalizar la evaluación. Verificar que se prohíba la opción de <i>Eliminar</i> al finalizar la evaluación. Cambio del estado de la evaluación al completarla.		
Proceso:	<ol style="list-style-type: none"> 1. En la tabla de evaluaciones seleccionar en la columna de acciones la opción <i>Evaluar</i>. 2. Seleccionar el proceso y la subcaracterística. 3. En la tabla de preguntas seleccionar la calificación a cada pregunta. 4. Seleccionar el botón <i>Guardar</i>. 5. Calificar todas las preguntas en todos las subcaracterísticas de todos los procesos. 6. Regresar a la página principal, y en el cuadro de las evaluaciones verificar que el estado cambie a <i>Completo</i>. 7. En la columna de <i>Acciones</i> seleccionar la opción <i>Visualizar</i> y ver los resultados obtenidos en cada proceso y subcaracterística. 		
Resultados esperados:	Al iniciar la evaluación de un proyecto se debe seleccionar el proceso y la subcaracterística que se desea evaluar. La subcaracterística debe aparecer según el proceso seleccionado. En calificación se deben presentar cinco opciones para seleccionar. Al evaluar todas los procesos en la página principal el estado del proyecto debe cambiar a <i>Completado</i> , y la opción de <i>Eliminar</i> se oculta.		
Resultados obtenidos:	La prueba concluyó satisfactoriamente (no se presentaron errores).		

La prueba de aceptación se realiza en base a la ejecución del programa por parte de tres usuarios finales escogidos arbitrariamente. Para esta prueba, se emplea una plantilla de evaluación de funcionalidades (*Tabla 48*), la cual contiene preguntas para evaluar el desarrollo y calidad de cada funcionalidad del programa. Para la valoración de estas preguntas se emplea la escala de Likert presentada en la *Tabla 49*. Estas pruebas se realizan después de la revisión por pares. Una vez analizados los resultados de estas pruebas, se realizan los cambios necesarios en las funcionalidades con baja puntuación.

Tabla 48 Prueba de Aceptación de la aplicación.

PLANTILLA DE PRUEBA DE ACEPTACIÓN					
Funcionalidad: Ingreso a opciones de la página de inicio.					
Pregunta	Valoración				
	1	2	3	4	5
El diseño de la página de inicio de la aplicación es fácil de entender.					
Los botones en la página de inicio cumplen con la función que indican.					
Las páginas externas se redireccionan de manera adecuada.					
El video informativo de la aplicación funciona correctamente.					
El manual de usuario es entendible.					
Funcionalidad: Registro e ingreso al sistema.					
Pregunta	Valoración				
	1	2	3	4	5
El proceso de creación de un nuevo usuario es sencillo.					
El diseño de la aplicación es intuitivo.					
Iniciar sesión es sencillo.					
Funcionalidad: Visualización y modificación del perfil del usuario.					
Pregunta	Valoración				
	1	2	3	4	5
La información personal del usuario se presenta de manera clara.					
El proceso de editar la información del usuario es sencillo.					
La información se modifica con éxito.					
Funcionalidad: Creación y modificación de un proyecto.					
Pregunta	Valoración				
	1	2	3	4	5
Es fácil crear un nuevo proyecto.					
Los modelos de evaluación se presentan con claridad.					
Los campos obligatorios al momento de crear y modificar son claros.					
Funcionalidad: Evaluación de un proyecto y visualización de resultados.					
Pregunta	Valoración				
	1	2	3	4	5
Es sencillo evaluar un proyecto ya creado.					
El proceso de evaluación es intuitivo.					
Las preguntas son claras y fáciles de entender.					
La aplicación indica correctamente el estado del proyecto (Pendiente o Completo)					
Los resultados se presentan en orden y de forma entendible.					

Tabla 49 Valoración en prueba de aceptación.

ESCALA DE LIKERT	
1	Totalmente en desacuerdo.
2	En desacuerdo.
3	Ni en acuerdo ni en desacuerdo.
4	De acuerdo.
5	Totalmente de acuerdo.

7. RESULTADOS.

En este apartado se presenta el modelo de evaluación diseñado según los tres niveles: Básico, Intermedio y Riguroso. Se describe la ejecución de la aplicación web que implementa el modelo de evaluación propuesto utilizando la metodología SWIRL (Software Web Iterativo Relacional Lógico). Además, se muestran los resultados de las evaluaciones realizadas mediante la aplicación web.

7.1. Modelo de Evaluación.

Con las fases y subcaracterísticas (SC) de la metodología analizados adecuadamente, se procede a plantear el modelo de evaluación propuesto en base a la metodología SWIRL para aplicaciones orientadas a la web.

7.1.1. Modelo de Evaluación 1 (Básico)

A continuación, se presenta el modelo de evaluación básico, indicando las preguntas de evaluación respectivas a las subcaracterísticas de cada una de las fases de la metodología SWIRL, representado en las tablas 50 a 55.

Tabla 50 Modelo de evaluación básico: fase 1

Fase 1: Análisis	
SC	Pregunta de evaluación
Definición del sistema.	La documentación del sistema cuenta con la definición del sistema actual y futuro.
Modelado de negocio.	En la documentación del sistema se puede observar el modelado del negocio.
Identificación de interesados	Existe una identificación de roles e interesados.
Requerimientos	La documentación del sistema brinda información de los requerimientos del proyecto.

Tabla 51 Modelo de evaluación básico: fase 2

Fase 2: Planificación	
SC	Pregunta de evaluación
Historias de usuario.	Se encuentra definidas las historias de usuario en la documentación del proyecto.
Definición de entregables.	Se encuentran definidos los entregables del proyecto.
Gestión de cronograma.	Hay establecido un plan de actividades en base al tiempo y recursos definidos, gestionando de manera adecuada el cronograma del proyecto.

Fase 2: Planificación	
SC	Pregunta de evaluación
	Se encuentra definido cómo se llevará la interacción y comunicación de todos los miembros e interesados dentro del proyecto.
Gestión de riesgos.	Se entregó un plan de prevención de riesgos al cliente.
Gestión de comunicaciones.	Se entregó un plan de comunicaciones al cliente.
Gestión de cambios y de la calidad.	Está definido el control de cambios y de calidad dentro de la documentación.

Tabla 52 Modelo de evaluación básico: fase 3

Fase 3: Modelado	
SC	Pregunta de evaluación
Diseño de modelo conceptual.	Se realizó el modelo conceptual de la base de datos.
Diseño de navegación.	El modelo de navegación es entendible por el cliente y por el equipo de desarrollo.
Diseño de interfaz abstracta de usuario.	Las interfaces gráficas de usuario generan un ambiente familiar para el cliente y guían de manera gráfica al usuario en las funcionalidades que presenta la aplicación web.
Diseño de diagramas UML.	Los diagramas UML se enfocan más en el modelado del comportamiento del sistema.

Tabla 53 Modelo de evaluación básico: fase 4

Fase 4: Implementación	
SC	Pregunta de evaluación
Separación de tareas entre los miembros de trabajo.	Se independizaron tareas acorde a la categorización que tenía cada miembro del grupo.
Codificación de las funcionalidades y componentes.	La codificación de las funcionalidades y componentes ayudan a que los desarrolladores tengan conocimiento concreto de cómo funcionarán los componentes del proyecto.
Validación de las restricciones globales y generales.	Existen restricciones establecidas para delimitar el proyecto.
Implementación de métricas de usabilidad y calidad.	Los integrantes del proyecto hicieron uso de la métrica ISO/IEC 9126.

Tabla 54 Modelo de evaluación básico: fase 5

Fase 5: Pruebas y revisión.	
SC	Pregunta de evaluación
Control integrados de cambios.	Se utilizó la plantilla definida en la metodología para restringir el control de cambios del sistema.
Realización de las pruebas de integración.	Se desarrollaron evaluaciones a los componentes de la aplicación.

Fase 5: Pruebas y revisión.	
SC	Pregunta de evaluación
Detección y corrección de errores.	Para esta prueba se usó herramientas externas como “validator de W3C”
Realización de pruebas de sistema.	Las pruebas del sistema fueron realizadas específicamente por un equipo externo y diferente al equipo de desarrollo.

Tabla 55 Modelo de evaluación básico: fase 6

Fase 6: Lanzamiento	
SC	Pregunta de evaluación
Selección y preparación del dominio.	El proyecto tiene designado un dominio para la aplicación.
Selección y alojamiento de la aplicación en un hosting.	El proyecto tiene designado un “hosting” para la aplicación.
Configuración del certificado SSL	El proyecto cuenta con certificados SSL para la aplicación.
Presentación de entregables.	Los entregables presentados al cliente fueron previamente revisados y modificados bajo la supervisión de un experto.

7.1.2. Modelo de Evaluación 2 (Intermedio)

A continuación, se presenta el modelo de evaluación intermedio, indicando las preguntas de evaluación respectivas a las subcaracterísticas de cada una de las fases de la metodología SWIRL.

Tabla 56 Modelo de evaluación intermedio: fase 1

Fase 1: Análisis.	
SC	Pregunta de evaluación
Definición del sistema	Existe un análisis de la viabilidad del proyecto a desarrollar en base a sus funcionalidades, restricciones y riesgos.
Modelado de negocio.	Están identificados los procesos requeridos dentro del modelo de negocio.
Identificación de interesados.	Definir los roles de las partes interesadas y sus actividades de interacción con el sistema.
Requerimientos.	Se encuentran delimitados los requerimientos relevantes dentro del sistema mediante su priorización y refinamiento.

Tabla 57 Modelo de evaluación intermedio: fase 2

Fase 2: Planificación.	
SC	Pregunta de evaluación
Historias de usuario	Se encuentran definidas las historias de usuario mediante el uso de plantillas las cuales sean entendibles por el usuario, director de proyecto, y equipo de trabajo.
	Se realizó una conversación entre el director del proyecto y el cliente sobre las historias definidas.
	Se aceptaron y validaron cada una de las historias definidas.

Fase 2: Planificación.	
SC	Pregunta de evaluación
Definición de entregables.	La documentación brinda al cliente y al director del proyecto conocimiento sobre las actividades que se realizaron, realizan y realizarán durante el desarrollo del proyecto.
Gestión de cronograma.	Se encuentra establecida la elaboración del cronograma.
	El tiempo estimado de holgura se establece acorde a la complejidad de las actividades y la distribución de la cantidad de personas involucradas.
	Se cumplieron los tiempos establecidos para la entrega del proyecto.
Gestión de riesgos.	Se elabora un documento donde se analiza y responde a los riesgos que puede tener el proyecto.
Gestión de comunicaciones.	Se establece un plan de comunicaciones para el proyecto.
Gestión de cambios y de la calidad.	Se lleva un registro detallado de cada cambio o modificación realizado siguiendo la matriz que proporciona la metodología.

Tabla 58 Modelo de evaluación intermedio: fase 3

Fase 3: Modelado.	
SC	Pregunta de evaluación
Diseño de modelo conceptual.	Es empleado el modelo conceptual como una ayuda para expresar el comportamiento del sistema de manera clara y fácil de entender.
Diseño de navegación.	El diseño de navegación logra brindar mayor beneficio al equipo de proyecto durante las etapas de implementación, verificación y prueba.
	Se implementaron diseños que representen de manera gráfica el comportamiento del sistema.
Diseño de interfaz abstracta de usuario.	El diseño de la interfaz permite a los programadores conocer que tipos de datos son requeridos en cada proceso y cómo presentar la información antes los usuarios.
	El diseño de interfaz logra brindar mayor beneficio al equipo de proyecto durante la etapa de implementación.
Diseño de diagramas UML.	Los modelos generados brindan al equipo conocimiento sobre los componentes que posee el sistema, y los actores que intervienen.
	Los modelos brindan al equipo conocimiento sobre las actividades y secuencia en la que estas se realizan, entre otros.

Tabla 59 Modelo de evaluación intermedio: fase 4

Fase 4: Implementación	
SC	Pregunta de evaluación
Separación de tareas entre los miembros de trabajos.	El director del proyecto interviene de manera personal con el cliente durante las reuniones de verificación del prototipo de la aplicación.
	El trabajo del proyecto fue dividido por módulos, funcionalidades e interfaces.

Fase 4: Implementación	
SC	Pregunta de evaluación
Codificación de las funcionalidades y componentes.	Se encuentran codificadas las funcionalidades y componentes de cada una de las interfaces.
Validación de las restricciones globales y generales.	Se realizó una validación de las restricciones globales y generales delimitadas por el cliente.
Implementación de métricas de usabilidad y calidad.	Se utilizó una plantilla de estándar para realizar la evaluación del proyecto.

Tabla 60 Modelo de evaluación intermedio: fase 5

Fase 5: Pruebas y revisión.	
SC	Pregunta de evaluación
Control integrado de cambios.	En la documentación del control de cambios se logra identificar los cambios realizados con la funcionalidad a la que afectan y la fecha en la que se realiza,
Realización de pruebas de sistema.	Se realizó la evaluación de calidad verificando que los indicadores asignados sean cumplidos.
	Se evaluó la tendencia de diseño de manera que la aplicación se logre ejecutar y cumpla con la tendencia que se especificó con anterioridad.
	Para realizar las pruebas de evaluación con herramientas SEO se utilizan herramientas externas.
Detección y corrección de errores.	La herramienta utilizada permite de una manera sencilla la detección y corrección de errores dentro del proyecto.
Realización de las pruebas de integración	Se realizó la evaluación de la aplicación por componentes, elementos y la manera en que estos se interrelacionan entre sí.
	Mediante la prueba se logra conocer si el proyecto se encuentra correctamente unificado, en función de los elementos, componentes y módulos.

Tabla 61 Modelo de evaluación intermedio: fase 6

Fase 6: Lanzamiento	
SC	Pregunta de evaluación
Selección y preparación del dominio.	El registro del dominio es adecuado y acorde a la aplicación.
Selección y alojamiento de la aplicación en un hosting.	Se realizó el procedimiento correspondiente para el alojamiento y publicación del "hosting".
Configuración del certificado SSL.	El proyecto cuenta con certificados SSL con la versión mínima segura para la aplicación.
Presentación de entregables.	Los entregables del proyecto fueron entregados en el tiempo pertinente al cliente.

7.1.3. Modelo de Evaluación 3 (Riguroso)

A continuación, se presenta el modelo de evaluación riguroso, indicando las preguntas de evaluación respectivas a las subcaracterísticas de cada una de las fases de la metodología SWIRL, según se aprecia en las tablas 62 a 67.

Tabla 62 Modelo de evaluación riguroso: fase 1

Fase 1: Análisis	
SC	Pregunta de evaluación
Definición del Sistema.	Existe un análisis de los riesgos en base a cada una de las funcionalidades que el sistema va a efectuar.
	Está considerado el análisis FODA en función de las características actuales del negocio y las futuras.
	Se encuentra identificada la perspectiva del producto conforme la necesidad que posee el cliente.
	En la documentación del sistema, está definida la empresa y las necesidades que este posee.
Modelado de negocio.	Se encuentran plasmadas las actividades del proceso de negocio mediante el uso de diagramas de actividades.
	Se realizaron cambios dentro del modelado de negocios para obtener alguna mejora.
Identificación de interesados.	Identificar todas las personas que intervengan en el proyecto.
	Conocer y comprender las necesidades y expectativas que cada uno posee.
	Se encuentra segmentados los roles y tipos de usuarios finales del sistema definiendo las posibles actividades a realizar.
Requerimientos.	Están organizados jerárquicamente los requerimientos en base a las funcionalidades del sistema.
	Se implementó una matriz de trazabilidad en función de los requerimientos y los casos de uso.
	Están definidas qué restricciones debe poseer la aplicación.

Tabla 63 Modelo de evaluación riguroso: fase 2

Fase 2: Planificación	
SC	Pregunta de evaluación
Historias de usuarios.	Se refinó las historias de usuario mediante el entendimiento de la funcionalidad principal del sistema ante la situación en específico.
	Se encuentran definidos los diversos caminos para ejecutar las historias grandes y realizados mediante diagramas de flujos para tener una mejor perspectiva de las soluciones.
	Las historias se encuentran divididas en función de diferentes interfaces que cubran la funcionalidad especificada.
	Al dividir cada una de las historias según la regla que efectúan, el equipo de desarrollo logra comprender en más detalle el comportamiento y funcionalidad del sistema.
Definición de entregables.	Existe la documentación de la priorización de los requerimientos del sistema.

Fase 2: Planificación	
SC	Pregunta de evaluación
	Se encuentra establecido entre la documentación un cronograma de actividades.
	Se encuentra definido la planificación de actividades por iteraciones.
	Existe dentro de la documentación un plan de contingencia de riesgos.
Gestión de cronograma.	Se encuentra establecida la elaboración del cronograma.
	Se encuentra establecido un tiempo de holgura en el proyecto.
	El documento de gestión de cronograma especifica la planificación de las iteraciones y sus actividades.
	Se encuentra definido de una manera estratégica el tiempo y recursos dentro del proyecto.
	Se logra identificar el propósito y enfoque de cada uno de los niveles del EDT.
	El EDT del proyecto logra descomponer el proyecto en tareas fáciles y con poco tiempo de implementación requerido.
	La descomposición de trabajos del EDT se hizo mediante la clasificación en niveles, en donde se representan las actividades generales en los niveles superiores y en los inferiores las tareas sencillas de realizar.
Gestión de riesgos	Se utiliza la plantilla definida en la metodología para registrar el plan preventivo de resto del sistema.
	Se elabora un plan de minimización de riesgos el proyecto.
Gestión de comunicaciones.	Se elabora un plan de comunicación para el proyecto y se le entrega al cliente.
Gestión de cambios y de la calidad.	Para determinar las adquisiciones que posee el proyecto se utiliza plantilla que brinda la metodología SWIRL.
	Se lleva un registro detallado de cada cambio o modificación realizado siguiendo la matriz que proporciona la metodología.
	El plan de aseguramiento de calidad implica las actividades que el director del proyecto especifica con la finalidad de que al finalizar el proceso de desarrollo el producto cumpla con dichas expectativas,

Tabla 64 Modelo de evaluación riguroso: fase 3

Fase 3: Modelado	
SC	Pregunta de evaluación
Diseño de modelo conceptual.	El modelo de entidad-relación enseña como las entidades, personas, objetos o conceptos, se relacionan entre si dentro de un sistema.
	El modelo relacional de la base de datos permite facilitar las tareas de diseño conceptual de la base de datos.
	El diccionario de datos es extenso e informativo.
	El diccionario de datos está compuesto por los datos y sus descripciones.
Diseño de navegación.	Para el diseño de navegación por usuarios se emplea el uso de actores
	El analista del sistema define la arquitectura del sistema.

Fase 3: Modelado	
SC	Pregunta de evaluación
	El diseño de navegación detalla narrativamente las navegaciones realizadas como respuesta a una funcionalidad del sistema.
Diseño de interfaz abstracta de usuario.	Las interfaces de usuario se desarrollan mediante el uso de herramientas case de diseño.
	Las interfaces comunes de usuario especifican los componentes de interfaz.
	Las interfaces de dependencia hacen referencia a las páginas web u otras aplicaciones que se encuentran inmersas o re-direccionadas dentro de la aplicación.
	El proyecto es implementado, codificado y probado mediante una herramienta de programación.
	El programa implementa una de las tendencias con mayor auge y uso dentro del desarrollo web.
Diseño de diagramas UML	Se puede observar la estructura física de la aplicación dentro de la documentación del programa.
	Se logra identificar los elementos que conforman un sistema mediante el diagrama de componentes.
	El diagrama de colaboración describe de una manera específica las interacciones que se realizan entre varios componentes dentro del sistema.
	Se representa el estado particular de un objeto contemplado dentro del sistema en una situación en específica por medio del diagrama de estado.
	El diagrama de secuencia indica la secuencia que seguirán las actividades internas del sistema.
	En el diagrama de actividades se representa el flujo de tareas realizadas por el sistema dentro de una función en específico.

Tabla 65 Modelo de evaluación riguroso: fase 4

Fase 4: Implementación	
SC	Pregunta de evaluación
Separación de tareas entre los miembros interesados.	El director del proyecto en esta etapa se comunica directamente con el cliente para recabar modificaciones.
	El director del proyecto interviene de manera personal con el cliente durante las reuniones de verificación del prototipo de la aplicación.
	La separación de las tareas del equipo fue establecida previamente y con buena comunicación con cada miembro del equipo.
Codificación de las funcionalidades y componentes	Se definieron los componentes y detalles generales de los módulos implementados dentro del sistema.
	Se diseñó y codificó las interfaces de usuario y paginas principales.
Validación de las restricciones globales y generales.	Se estableció estándares de codificación y de calidad empleados para el conocimiento global del equipo de desarrollo.

Fase 4: Implementación	
SC	Pregunta de evaluación
Implementación de métricas de usabilidad y calidad.	Se aplica una evaluación para saber si el sistema mantiene la información y la almacena correctamente en la base de datos.
	Se establecieron estándares de evaluación de calidad, verificando que el producto final cumpla con las especificaciones.

Tabla 66 Modelo de evaluación riguroso: fase 5

Fase 5: Pruebas y Revisión.	
SC	Pregunta de evaluación
Control integrado de cambios.	Existe un documento de control de cambios para el sistema.
	La gestión de cambios brinda al director del proyecto un mayor conocimiento sobre el cambio efectuado y las versiones anteriores del proyecto.
	La gestión de cambios se lleva a cabo en base la plantilla sugerida en la metodología aplicada.
Realización de pruebas de sistema.	La prueba de calidad se realiza mediante el uso de los indicadores que se especifican acorde a la métrica ISO/IEC 9126.
	En la evaluación de la calidad se permite conocer si la funcionalidad y comportamiento del sistema ante los usuarios finales.
	Se realizó una evaluación para la tendencia empleada.
	Se evalúa la tendencia en cada módulo del sistema.
Detección y corrección de errores.	Se utilizó una plantilla para evaluar correctamente el sistema acorde a la calidad según los criterios de evaluación.
	Después de la detección de errores se procedió a corregir cada una de ellos, y volver a realizar la evaluación con la misma herramienta.
Realización de las pruebas de integración.	Se hizo uso de dos herramientas de validación como mínimo.
	La prueba de integración brinda mayor certeza del componente en el cual se encuentra el fallo, y se pueda modificar de manera que se ahorra tiempo.
	Existe manual de usuario y manual de programador.

Tabla 67 Modelo de evaluación riguroso: fase 6

Fase 6: Lanzamiento	
SC	Pregunta de evaluación
Selección y preparación del dominio.	El dominio utilizado es corto para que el usuario logre aprenderlo con facilidad.
	Se seleccionó un proveedor de confianza para asegurar un buen servicio del dominio.
	El dominio utilizado no tiene similitud a otro dominio existente.
Selección y alojamiento de la	Se verificó que no haya fallo de comunicación con la aplicación en el "hosting" seleccionado.

Fase 6: Lanzamiento	
SC	Pregunta de evaluación
aplicación en un hosting.	
Configuración del certificado SSL.	El proyecto cuenta con certificados SSL que preste el cifrado adecuado para la aplicación.
Presentación de entregables.	Los entregables presentados al cliente fueron previamente revisados y modificados bajo la supervisión de un experto.

7.2. Aplicación web en ejecución.

En esta sección, se ilustrará el funcionamiento de la aplicación web desarrollada mediante la metodología SWIRL llevando a cabo las fases de análisis, planificación, modelado, implementación, revisión y pruebas. En la *Figura 8*, se observa la página principal para los usuarios no registrados y los registrados, conjuntamente con las imágenes posteriores que muestran las opciones que ofrece este menú como se muestra en las *Figuras 9 y 10*. A partir de la *Figura 11* (incluida) se muestra el funcionamiento del sistema para usuarios registrados, desde la creación de un nuevo proyecto, hasta la evaluación de cada actividad de las subcaracterísticas de las fases del ciclo de vida.

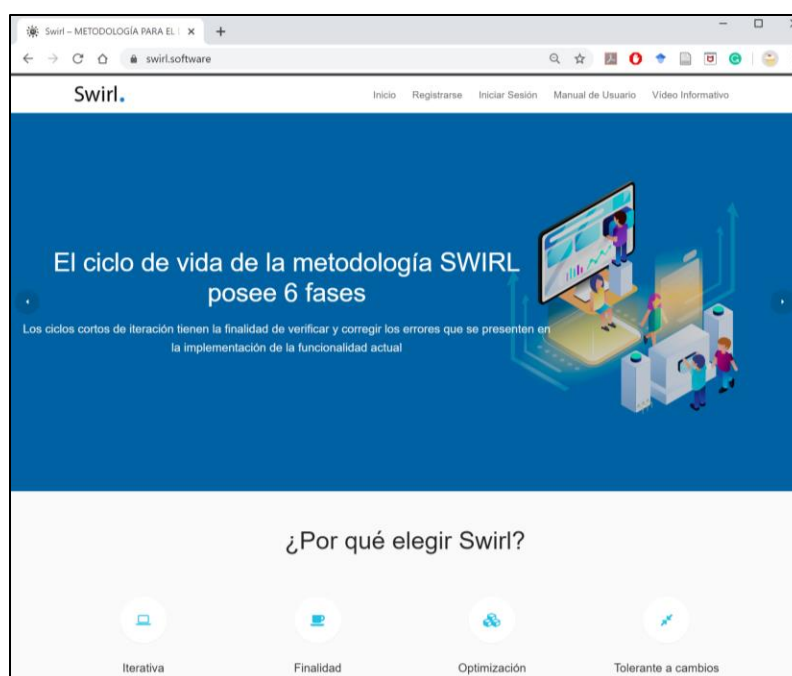


Figura 8 Aplicación en ejecución: página principal.

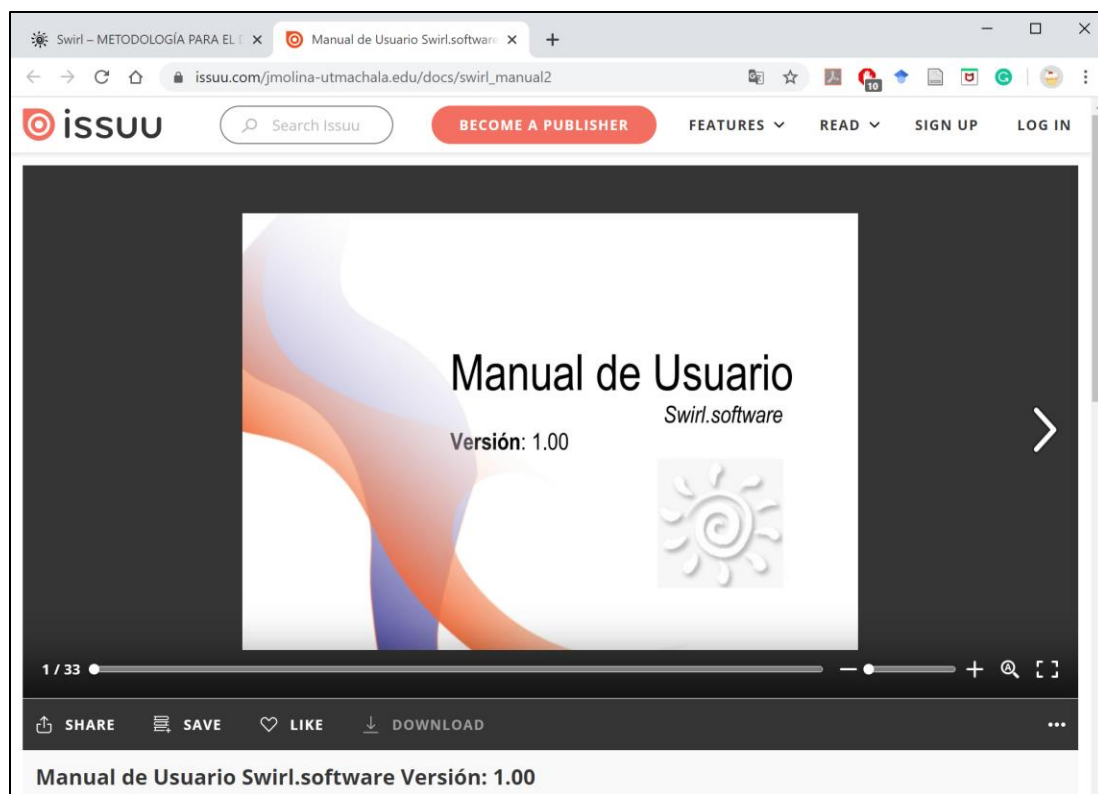


Figura 9 Aplicación en ejecución: manual de usuario.

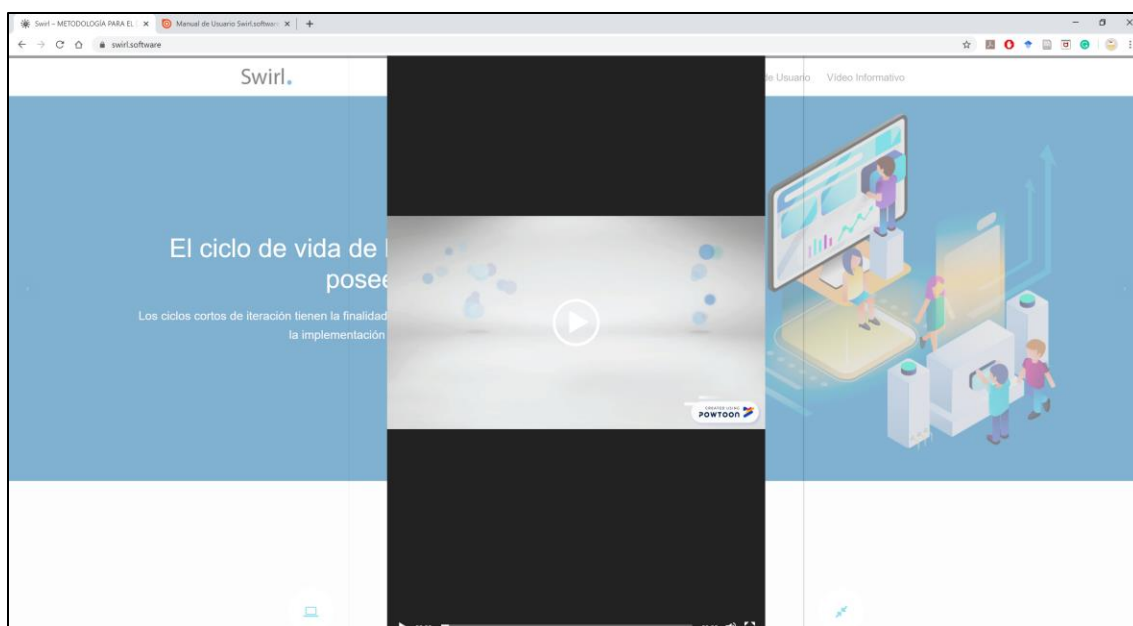


Figura 10 Aplicación en ejecución: video informativo.

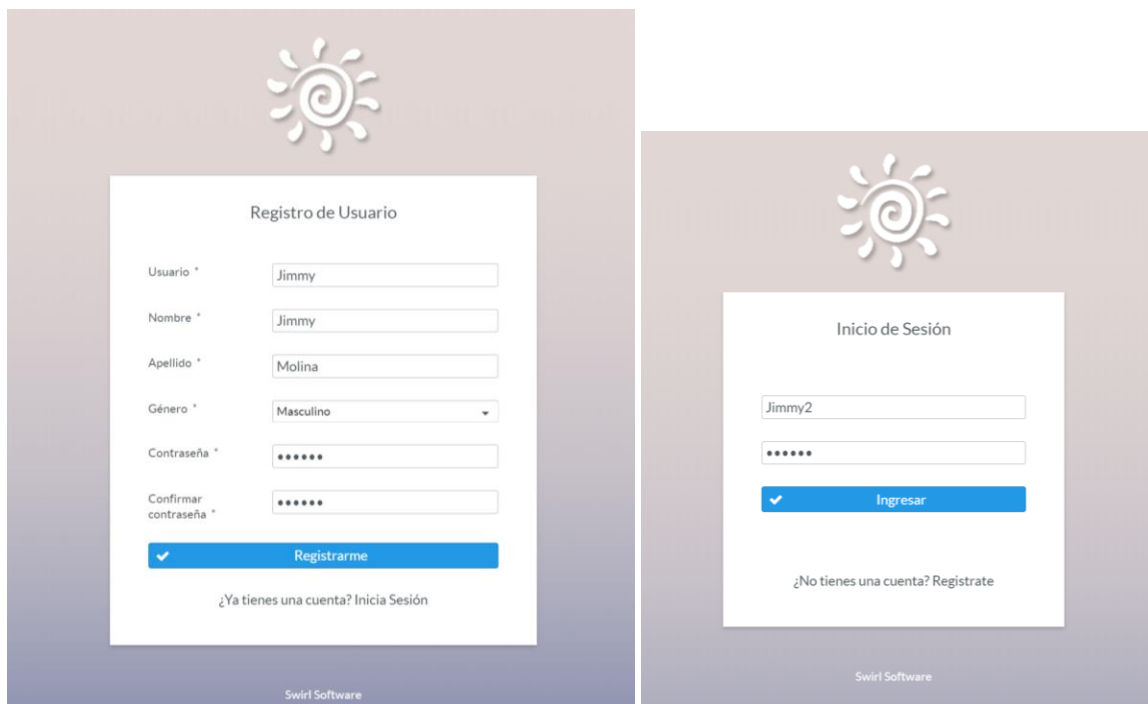


Figura 11 Aplicación en ejecución: registro e inicio de sesión.

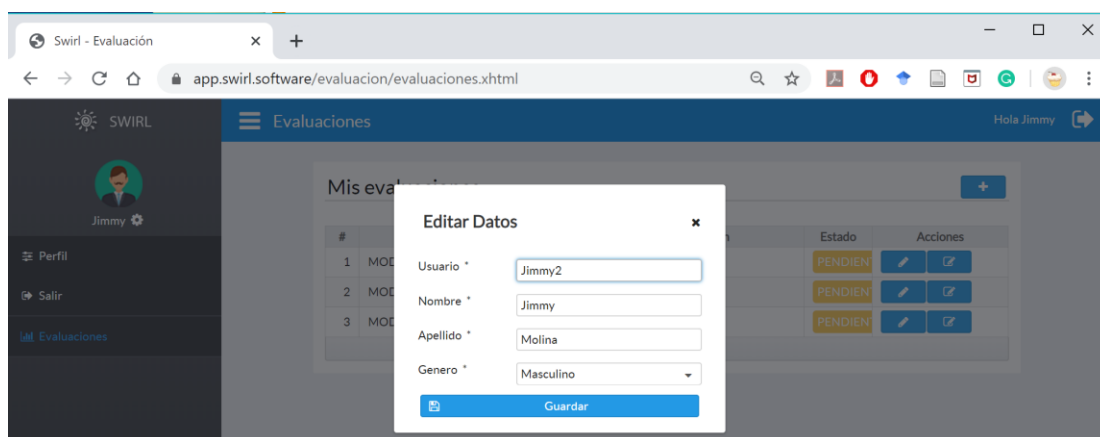


Figura 12 Aplicación en ejecución: visualización del perfil.

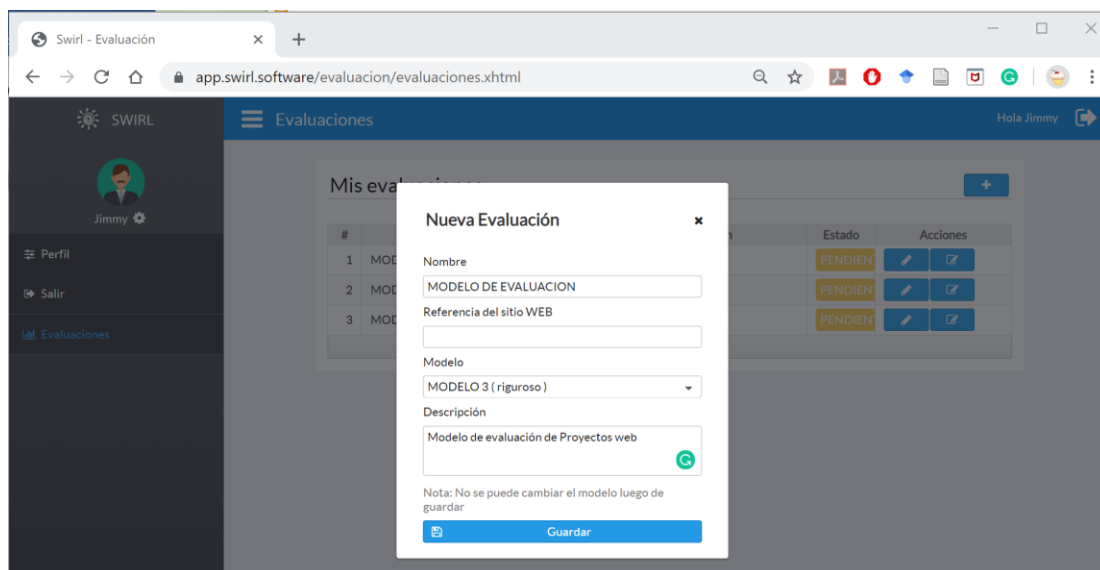


Figura 13 Aplicación en ejecución: creación de proyectos.

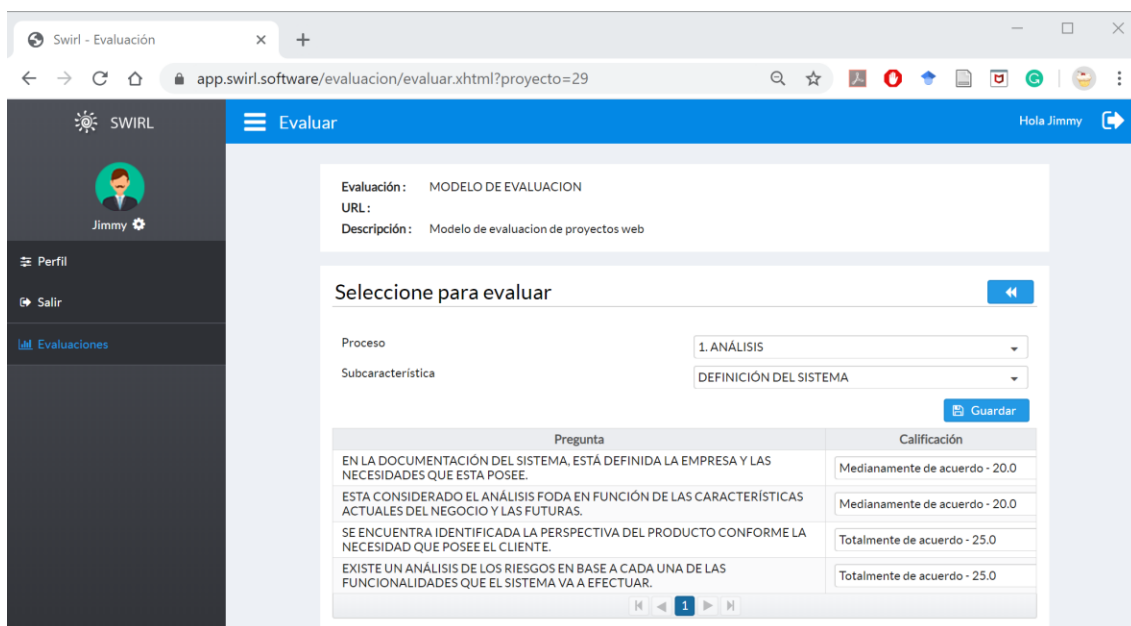


Figura 14 Aplicación en ejecución: evaluación de proyectos.

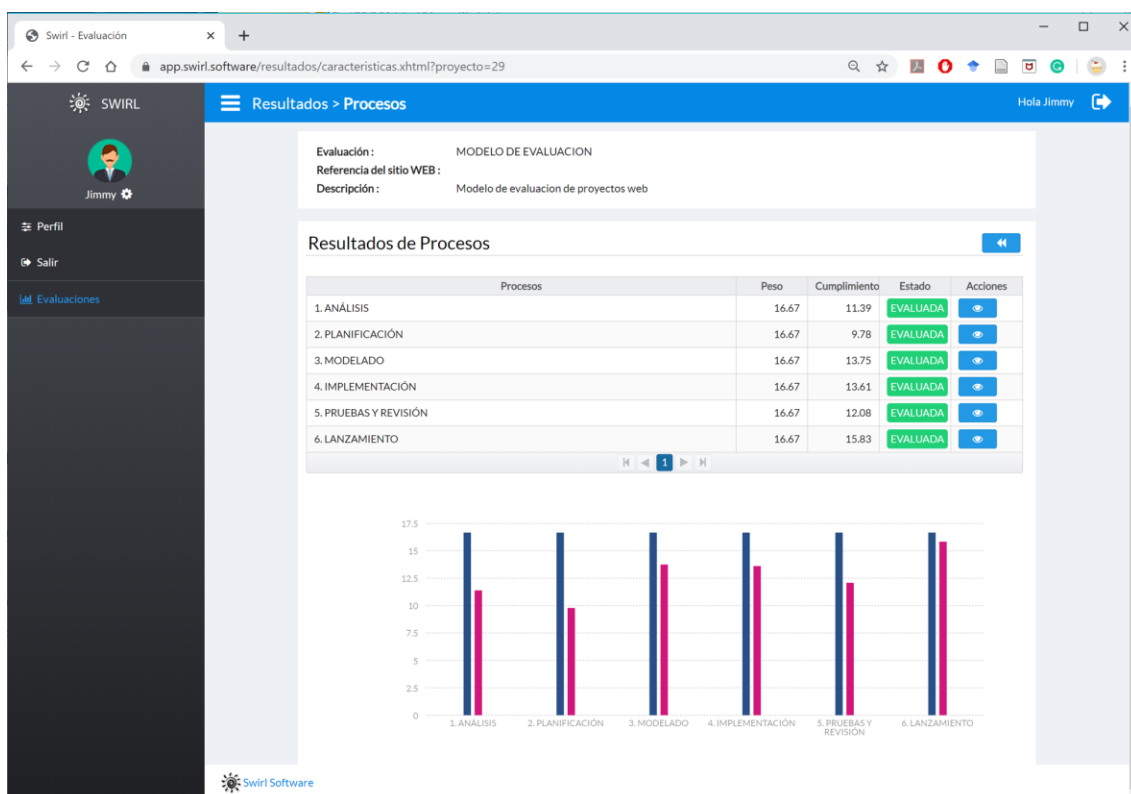


Figura 15 Aplicación en ejecución: visualización de resultados.

7.3. Evaluación de proyectos mediante los modelos.

Se han llevado a cabo diferentes evaluaciones con el Modelo Riguroso (Modelo 3) mediante 8 usuarios a 35 aplicaciones web de diferentes empresas, las cuales son presentadas en la *Tabla 68*. Se ha considerado solo el modelo 3 en las evaluaciones,

debido a que abarca en mayor medida los requerimientos de aplicaciones web según la metodología SWIRL en todas sus fases: análisis, planificación, modelado, implementación, revisión y pruebas, lanzamiento y marketing.

Para evaluar cada aplicación web se procede a crear una nueva evaluación, indicando los datos informativos de la misma y seleccionando el modelo deseado (en este caso el modelo 3). Una vez que se encuentra creada la nueva evaluación se selecciona la opción *Evaluar*, y se procede a seleccionar el proceso y la subcaracterística respectiva; luego se evalúan todas las preguntas que se encuentran dentro de cada subcaracterística, tal como se ha mostrado en la *Figura 14*. Una vez que se haya finalizado la evaluación en todos los procesos y subcaracterísticas se podrán visualizar los resultados obtenidos.

Los resultados de las evaluaciones dentro de la aplicación web se presentan mediante un gráfico de barras, tal como se ha expuesto en la *Figura 15*, en las cuales se indica el porcentaje de cumplimiento y el porcentaje obtenido en las preguntas, subcaracterísticas y fases respectivamente. De esta manera, se logra visualizar la fase en la que el proyecto evaluado presenta más debilidad.

A continuación, en la *Tabla 69*, se presentan los resultados obtenidos en cada proceso por las aplicaciones web evaluadas. Cada proceso tiene un porcentaje ideal de cumplimiento del 16,666667% respectivamente, que resulta al dividir el 100% entre los seis procesos de estudio. El valor asignado se redondea a dos decimales y el total se obtiene mediante la suma de todos los porcentajes obtenidos en cada proceso.

El resultado de la evaluación de aplicaciones web considerado en las empresas de la *Tabla 69* relacionada a las fases de aplicar la metodología SWIRL, establece el cumplimiento ideal identificando la fase de mayor cumplimiento por parte de las aplicaciones web.

Tabla 68 Aplicaciones Web evaluadas.

#	EMPRESA	ENLACE DE LA APLICACION.
1	AGUILERA SOFT	http://www.aguilerasoftware.com/
2	COOP. OCCIDENTALES.	http://occidentales.ec/
3	COLINEAL	https://colineal.com/
4	COLEGIO 9 DE OCTUBRE	http://nuevedeoctubre.edu.ec/
5	MOVISTAR	https://www.movistar.com.ec/
6	OBINTE	https://obinte.com/
7	FARMARED'S	http://www.farmareds.com.ec/
8	PINTO	https://pinto.com.ec/
9	CRÉDITOS ECONÓMICOS	https://www.creditoseconomicos.com/
10	SOFADCON	https://sofadcon.com/
11	SECAP	https://www.secap.gob.ec/
12	BIBIS	http://www.perfumeriabibis.com/
13	GMO	https://gmo.com.ec/
14	LEE	https://www.lee.es/es-es/new-collection.html
15	PAYLESS	https://ecuador.lovepayless.com/
16	PHARMACY'S	http://www.pharmacys.com.ec/
17	NATURISSIMO	http://naturissimo.com/
18	ALCALDÍA DE MACHALA	http://www.machala.gob.ec/
19	COMPUTRON	https://computron.com.ec/
20	CLINICA AGUILAR	http://clinicaaguilar.es/
21	GOBERNACIÓN EL ORO	https://gobnacioneloro.gob.ec/
22	CLARO	https://www.claro.com.ec/personas/
23	OFSERCONT S.A.	http://exa.ofsercont.com/
24	TRIONICA	http://trionica.com.ec/
25	IRFEYAL	http://web.irfeyal.org/
26	INFOTECSA	http://www.infotecsa.com.ec/
27	EXCELLENT SOFT	https://www.excellentsoft.net/
28	POSTGRADO UTMACH	https://postgrado.utmachala.edu.ec:8080/admision/programas/
29	COOPERATIVA JEP	https://www.coopjep.fin.ec/inicio#
30	PAGUPA SOFT	http://www.pagupasoft.com/
31	BANCO DE MACHALA	https://www.bancomachala.com/
32	CONSEJO NACIONAL ELECTORAL EL ORO	http://delegaciones.cne.gob.ec/el-oro
33	CUERPO DE BOMBEROS DE MUNICIPALIDAD MACHALA	https://www.bomberosmachala.gob.ec/
34	CORPORACIÓN DE TELECOMUNICACIONES CNT	https://www.cnt.gob.ec/
35	DIRECTV	https://www.directv.com.ec/

Tabla 69 Tabla de resultados de evaluación de aplicaciones web.

PROCESOS		ANÁLISIS (max. 16,6666667%)	PLANIFICACIÓN (max. 16,6666667%)	MODELADO (max. 16,6666667%)	IMPLEMENTACIÓN (max. 6,6666667%)	PRUEBAS Y REVISIÓN (max. 6,6666667%)	LANZAMIENTO Y MARKETING (max. 6,6666667%)	TOTAL (max. 100%)
1	AGUILERA SOFT	15.28	12.37	13.19	14.17	15.08	14.44	84.53
2	COOP. OCCIDENTALES.	14.72	11.31	14.31	13.61	10.42	11.39	75.76
3	COLINEAL	5.49	3.94	7.15	12.08	7.64	12.5	48.80
4	COLEGIO 9 DE OCTUBRE	5.21	3.55	4.86	4.72	3.33	4.17	25.84
5	MOVISTAR	12.22	10.3	8.89	14.17	10.83	13.33	69.74
6	OBINTE	15.49	13.03	13.75	15.83	13.75	16.11	87.96
7	FARMARED'S	6.46	5	7.22	5.83	8.33	14.17	47.01
8	PINTO	15.21	14.15	15	16.39	13.83	16.67	91.25
9	Créditos Económicos	9.79	9.7	11.04	12.92	10.78	12.5	66.73
10	SOFADCON	15.28	13.62	13.47	15	14.83	16.11	88.31
11	SECAP	15.56	14.7	16.39	15.97	15.08	16.67	94.37
12	BIBIS	9.86	9.13	8.33	12.08	9.28	12.5	61.18
13	GMO	14.86	13.68	14.51	13.61	12.72	15.83	85.21
14	LEE	10	8.56	5.14	10	7.83	10.83	52.36
15	PAYLESS	4.51	5.81	10.42	9.17	6.11	9.17	45.19
16	PHARMACY'S	15	11.53	12.85	15	14.08	16.39	84.85
17	NATURISSIMO	7.29	5.63	8.82	7.78	3.83	9.17	42.52
18	ALCALDÍA DE MACHALA	13.75	12.44	13.89	13.33	11.58	12.5	77.49
19	COMPUTRON	10.14	10.95	11.04	12.22	9	12.22	65.57
20	CLINICA AGUILAR	10.83	8.49	11.25	10.97	10.44	13.33	65.31
21	GOBERNACIÓN EL ORO	6.81	11.05	10.83	12.64	11.5	14.72	67.55
22	CLARO	10	10.66	9.58	11.67	11.39	10	63.30
23	OFSERCONT S.A.	12.92	10.38	11.67	13.33	11.14	15	74.44
24	TRIONICA	13.06	12.19	12.78	14.31	11.03	12.5	75.87
25	IRFEYAL	11.67	10.36	11.67	12.5	9.72	13.33	69.25
26	INFOTECSA	12.15	9.22	12.92	14.31	13.67	14.17	76.44
27	EXCELLENT SOFT	12.29	9.48	10.76	11.94	10.67	13.33	68.47
28	POSTGRADO UTMACH	11.04	10.4	11.25	12.08	11.03	15.83	71.63
29	COOPERATIVA JEO	11.67	10.64	10.83	11.94	11.17	15	71.25
30	PAGUPA SOFT	11.39	9.78	13.75	13.61	12.08	15.83	76.44
31	BANCO DE MACHALA	12.22	11.35	15.56	7.78	13.08	15.83	75.82
32	CONSEJO NACIONAL ELECTORAL EL ORO	9.44	9.31	14.51	15	11.42	15.83	75.51
33	CUERPO DE BOMBEROS DE MUNICIPALIDAD MACHALA	13.61	11.63	14.03	15.69	14.06	16.67	85.69
34	CORPORACIÓN DE TELECOMUNICACIONES CNT	15.14	12.41	14.86	13.19	13.97	16.67	86.24
35	DIRECTV	12.57	9.29	9.72	15.69	13.56	16.67	77.50

Los resultados obtenidos son evaluados según el porcentaje de cumplimiento, siendo el porcentaje ideal para cada fase del 16,6666667%. Según los datos presentados en la tabla, se puede obtener que el promedio en las fases son las siguientes: análisis (11.51%); planificación (10.17%); modelado (11.61%); implementación (12.59%); pruebas y revisión (11.09%); y lanzamiento y marketing (16.67%). Mediante los porcentajes se puede concluir que las fases de implementación, modelado y lanzamiento y marketing son las que poseen un mayor porcentaje de cumplimiento indicando mayor índice de preocupación y rigurosidad en los subprocesos. Esto se debe a que un fallo dentro de estas fases puede afectar el rendimiento final del programa. Además, las fases de análisis, planificación y pruebas poseen un porcentaje de cumplimiento intermedio, indicando que algunas de las actividades realizadas dentro de estas fases no son estrictamente consideradas por los directores de proyectos.

Para evaluar en profundidad los resultados obtenidos, se presentan a continuación, como ejemplo, los resultados de la evaluación obtenidos por la empresa PAGUPA SOFT, según el enlace de la aplicación. La *Figura 16* muestra los resultados obtenidos por la aplicación evaluada en todos los procesos, donde el color azul indica el porcentaje ideal a cumplir en cada fase (16,6666667%) y el rojo el porcentaje obtenido.

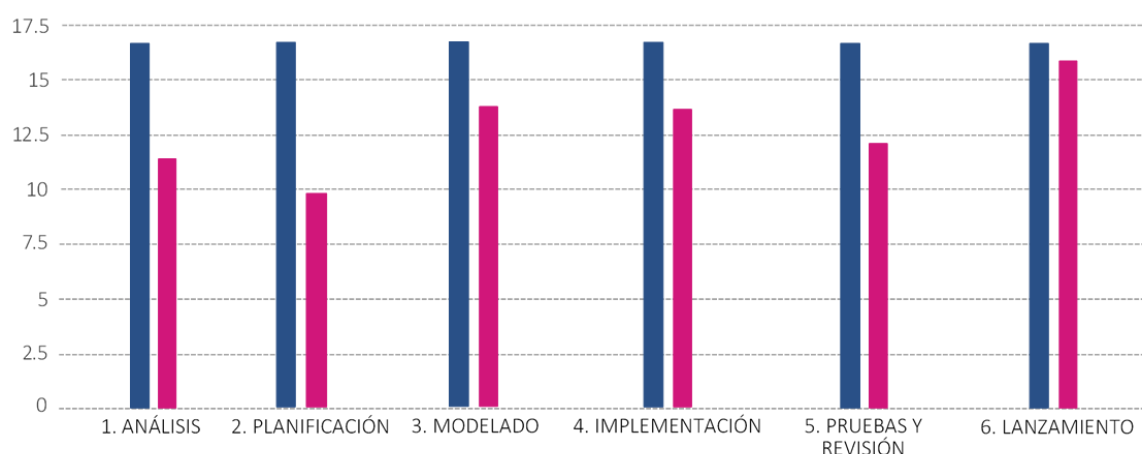


Figura 16 Resultado de la evaluación por fases.

Según la gráfica, se puede identificar que la aplicación web evaluada presenta un valor de 15.83% en la fase de lanzamiento, lo cual indica que existe un mayor grado de cumplimiento dentro de esta fase. Mientras que la fase de planificación es la que presenta un menor porcentaje de cumplimiento con 9.78%, interpretando que la

definición de actividades, entregables o gestiones no se realizaron de manera estrictamente rigurosa. Por otro lado, las demás fases presentan un grado admisible de cumplimiento sobrepasando el 10% de cumplimiento por cada fase (16,6666667% ideal).

Para conocer más a fondo el cumplimiento de cada fase por la aplicación web, se presentan los resultados obtenidos dentro de las subcaracterísticas de cada una de las fases. Dentro de cada fase el porcentaje de cumplimiento ideal varía según la cantidad de subcaracterísticas, si hay cuatro subcaracterísticas la barra azul indica 25% cada una; mientras que si hay seis subcaracterísticas indica un 16,6666667%.

PROCESO 1: ANÁLISIS

Los resultados mostrados en la *Figura 17* indican que la subcaracterística que posee mayor cumplimiento dentro de la fase de análisis es la Identificación de interesados con un 23.33%, mientras que las subcaracterísticas restantes (definición del sistema, modelado de negocio, y requerimientos) poseen un cumplimiento del 15%. De estos valores se puede concluir que es recomendable fortalecer las características dentro del proceso de análisis como la definición del sistema, modelado de negocio y requerimientos.

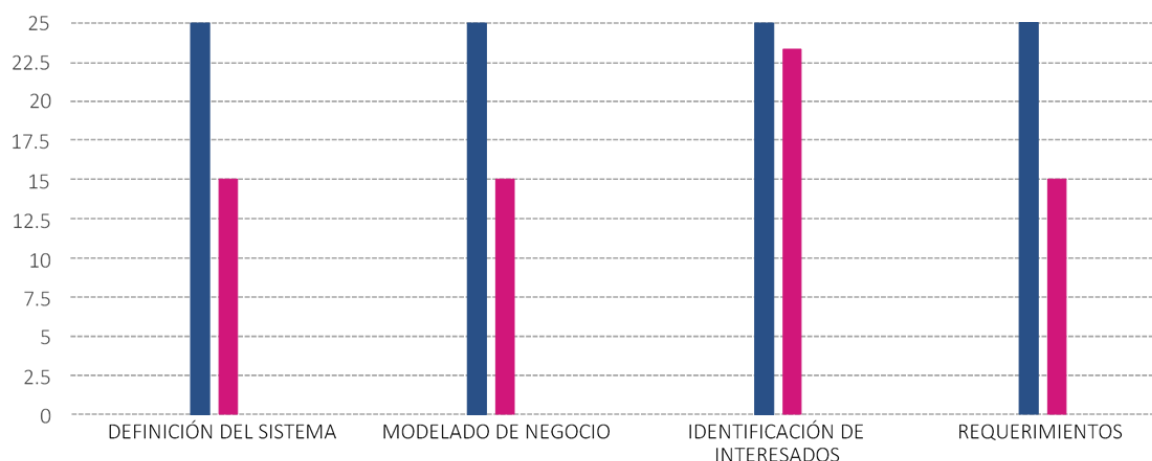


Figura 17 Resultado de la evaluación en la fase de análisis.

PROCESO 2: PLANIFICACIÓN.

Los resultados obtenidos en la fase de planificación y mostrados en la *Figura 18*, *Figura 19* indican que las subcaracterísticas con porcentaje de cumplimiento altos son Gestión de comunicaciones y Gestión de cronograma con 13.33% y 12.86% respectivamente; seguido de las Historias de Usuario con 10%. Por otro lado, las

subcaracterísticas con un grado inferior al 10% (es decir un porcentaje de cumplimiento bajo) son Gestión de riesgos (8.33), Definición de entregables (7.50%), y Gestión de cambios y de la calidad (6.67).

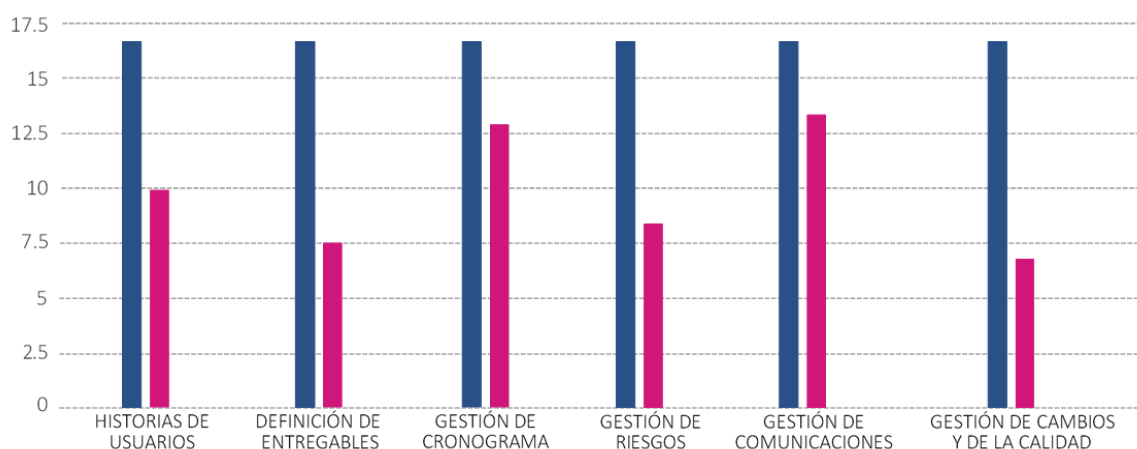


Figura 18 Resultado de la evaluación en la fase de planificación.

Mediante las tabulaciones de los resultados de la valoración es posible identificar las actividades en las cuales se ha presentado un bajo porcentaje de cumplimiento o en su defecto se han presentado mayor acontecimiento de riesgos, de manera que se logre identificar y generar estrategias preventivas para la planificación de proyectos futuros.

PROCESO 3: MODELADO.

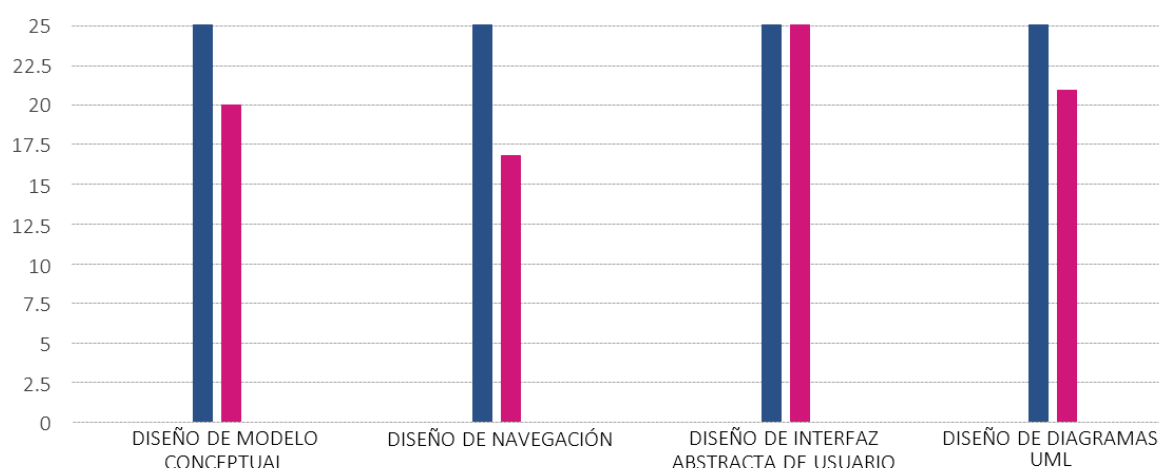


Figura 19 Resultado de la evaluación en la fase de modelado.

Dentro de la fase de modelado, tal como se aprecia en la figura 19, la subcaracterística con mayor porcentaje de cumplimiento es el Diseño de interfaz abstracta de usuario con un 25%, seguida del Diseño de diagramas UML con 20.83%,

Diseño de modelo conceptual 20% y, finalmente, el Diseño de navegación con 16.67%.

Cada indicador dentro de esta fase se centra en el diseño de modelos y diagramas necesarios para la documentación y entendimiento de las funcionalidades por el equipo de desarrollo y otros involucrados, es por ello que las actividades que presenten un bajo grado de cumplimiento deben ser mejoradas.

Este análisis no solo se realiza para proyectos futuros, también puede ser empleado si el proyecto no ha llegado a la fase de lanzamiento o cierre, ya que puede registrarse dentro de la gestión de cambios sin presentar problemas o retrasos en la entrega el producto final.

PROCESO 4: IMPLEMENTACIÓN.

El análisis del resultado obtenidos ayuda a mejorar los procesos de desarrollo y codificación del programa, vinculado relativamente al correcto funcionamiento del sistema final, y por ende su calidad.

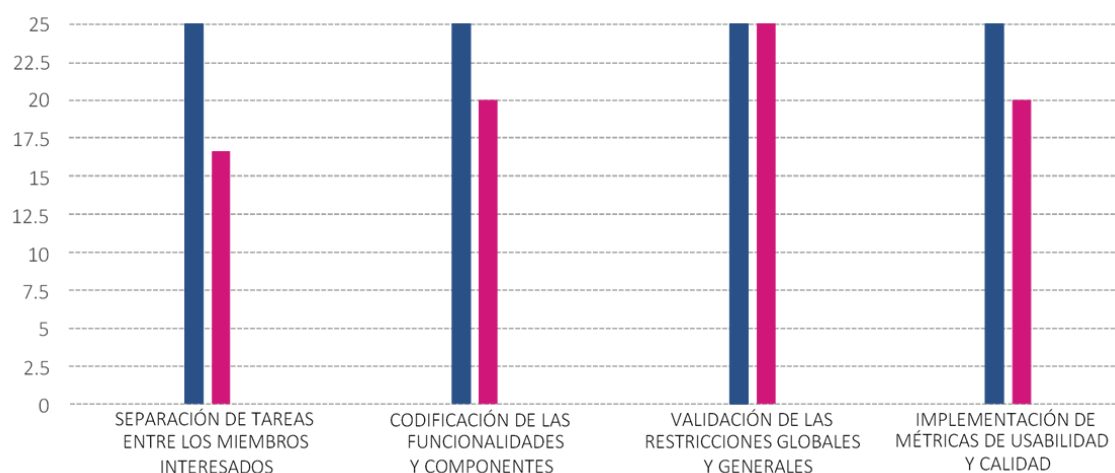


Figura 20 Resultado de la evaluación en la fase de implementación.

La *Figura 20* muestra los resultados obtenidos en la fase de implementación, los cuales presentan puntuaciones relativamente altas, iniciando con la Validación de las restricciones globales y generales con un 25%, seguido de la Codificación de las funcionalidades y componentes con 20%, Implementación de métricas de usabilidad y calidad con un 20% y, finalmente, la Separación de tareas entre los miembros interesados con un 16.67%.

PROCESO 5: PRUEBAS Y REVISIÓN.

Su análisis se enfoca mayormente en identificar si se ha llevado un proceso adecuado para la revisión del funcionamiento del sistema, y las actividades en las cuales se debe hacer más énfasis ya sea para proyectos futuros o para una nueva iteración dentro del mismo proyecto (siempre y cuando no haya llegado al cierre)

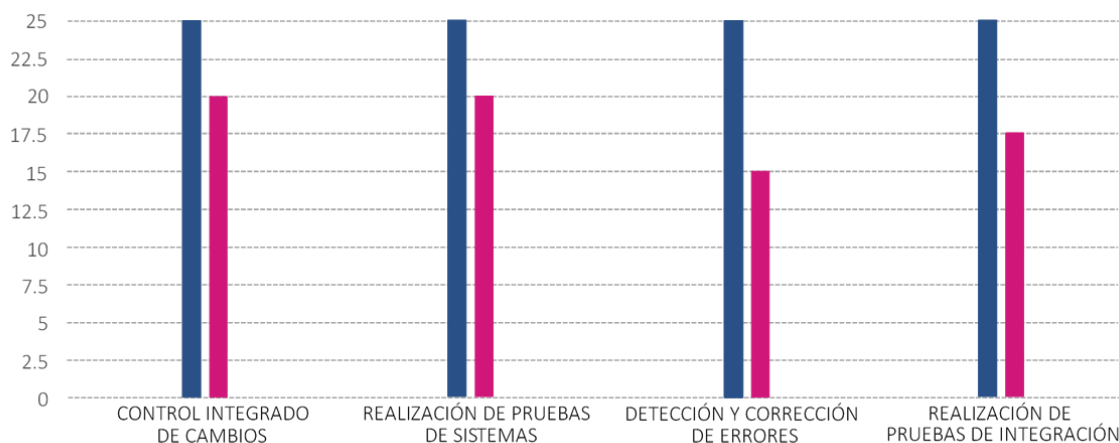


Figura 21 Resultado de la evaluación en la fase de pruebas y revisión.

Dentro de la fase de pruebas y revisión, las subcaracterísticas que presentan un resultado óptimo son, como puede observarse en la Figura 21, el Control integrado de cambios y la Realización de pruebas de Sistemas con un cumplimiento del 20%, seguidos de la Realización de pruebas de integración con 17.50% y, finalmente, la que presenta la puntuación más baja es la Detección y corrección de errores con un 15%.

PROCESO 6: LANZAMIENTO Y MARKETING.

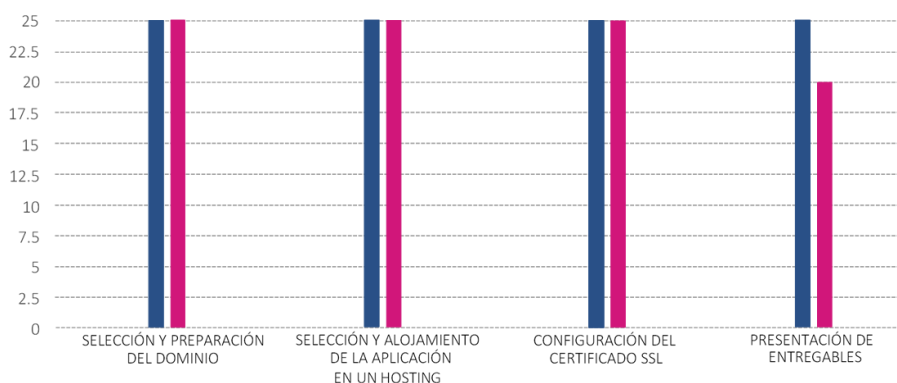


Figura 22 Resultado de la evaluación en la fase de lanzamiento y marketing.

La *Figura 22* muestra los resultados obtenidos por la aplicación web dentro de la fase de lanzamiento y marketing, la cual indica que el cumplimiento de la mayoría de subcaracterísticas es óptimo. Sin embargo, la Presentación de entregables contiene un porcentaje menor (20%).

Los resultados obtenidos dentro de esta fase ayudan al análisis de las actividades que deben mejorarse al llevar a cabo el lanzamiento y marketing de la aplicación. Estos resultados serán válidos solo para proyectos futuros, ya que al momento de obtener los resultados el proyecto actual ya finalizará.

En conclusión, de la evaluación desarrollada a la empresa PAGUPASOFT se obtiene que posee un resultado del 76%; es decir, un grado de cumplimiento completo según los valores establecidos en la Tabla 70. Esto indica que puede tener mejoras durante su proceso de desarrollo en algunas de sus fases, pero en general brinda un producto de calidad y procedimientos eficaces.

La Tabla 70 relaciona la valoración de los porcentajes con el grado de cumplimiento que tiene en relación al modelo riguroso efectuado en la evaluación.

Tabla 70 Valoración sobre porcentaje de cumplimiento

%	Descripción
0% - 50%	Cumplimiento bajo
51% - 75%	Cumplimiento intermedio
76% - 100%	Cumplimiento completo

8. DISCUSIÓN Y CONCLUSIONES

En este apartado, se presenta la discusión del trabajo en base a los modelos de evaluación y la aplicación web diseñada. La aplicación desarrollada presenta el modelo de evaluación propuesto en sus tres categorías (básico, intermedio y riguroso), permitiendo al usuario valorar diferentes proyectos web en base a la metodología SWIRL y visualizar cuán optimas son cada una de las etapas del proyecto según los indicadores cumplidos en cada subcaracterística. Además, se presentan las conclusiones obtenidas dentro de la investigación para cada objetivo específico establecido.

8.1. Discusión de los resultados.

Contrastar las diferentes técnicas y procedimientos que una metodología de desarrollo web implementa durante el ciclo de vida, requiere la consideración de aspectos internos y externos. Conocer los aspectos claves bajo los cuales se puede evaluar un proyecto para conocer su grado de usabilidad y calidad, en cuanto al producto y al proceso que se ha desarrollado, es fundamental para todos los directores de proyectos. El modelo de evaluación propuesto dentro de la investigación considera las fases de la metodología híbrida de desarrollo SWIRL, la cual no solo aborda las etapas de desarrollo de la aplicación web, sino también incluye actividades de gestión de proyecto.

El modelo contempla tres categorías. La primera “*básico*”, en la cual se valoran las actividades que se deben realizar en cada fase de manera “ligera”, permitiendo la accesibilidad de proyectos con otras metodologías ágiles. Una vulnerabilidad que presenta identificar las actividades de manera superficial, es que influye en la calidad final del proyecto. Por ello el modelo de evaluación básico presenta un grado de rigidez menor e incrementa la flexibilidad dentro del proyecto. Dentro de este modelo se debe incrementar la intensidad de las valoraciones en las etapas de análisis y planificación, debido a que, en base a las mismas, se desarrollarán las demás actividades del proyecto.

El segundo modelo denominado como “*intermedio*”, evalúa las características con un enfoque un poco más profundo, buscando el beneficio del producto y considerando las actividades del director dentro del proyecto. Un punto intermedio de valoración de

un proyecto web, es adecuado para conocer la calidad de las aplicaciones; sin embargo, se presentan defectos en cuanto a características internas del producto, las cuales son realizadas inmersamente en las fases del ciclo de vida. En este enfoque, se debe proponer mayor peso de valoración a las fases de modelado e implementación.

Finalmente, el tercer modelo propuesto es el “*riguroso*”, en el cual se consideran todas las actividades, procesos y artefactos que las etapas tienen inmersas, de manera que evalúa la actividad del director del proyecto para valorar la calidad final del producto. La rigurosidad con la que se establecen los indicadores dentro de este modelo es alta en todas las fases del ciclo de vida.

La aplicación desarrollada presenta el modelo de evaluación propuesto de manera ordenada, permitiendo al usuario valorar los diferentes proyectos web en base a la metodología SWIRL. En base a los resultados de las 35 evaluaciones realizadas a las diferentes aplicaciones web con el modelo “*riguroso*” se logra analizar las etapas que poseen mayor cumplimiento durante el proceso de desarrollo.

La fase de “Lanzamiento y marketing”, en promedio, presenta un porcentaje mayor de cumplimiento, indicando que durante el proceso de desarrollo se crea un enfoque mayor en considerar el “hosting” en el que se alojará la aplicación, los certificados SSL, entre otras subcaracterísticas que garantizan una ejecución segura de la aplicación web. A continuación de esta fase, se encuentra la fase de implementación, la cual busca asegurar un correcto desarrollo y diseño intuitivo de la interfaz de usuario. La fase de modelado, análisis, pruebas y revisión presentan un porcentaje intermedio dentro de los resultados obtenidos, permitiendo garantizar un correcto diseño de la base de datos y la interfaz final de usuario, así como el cumplimiento de requerimientos y la ejecución de la aplicación sin errores.

Finalmente, destacar que la fase que posee el menor porcentaje de cumplimiento es la fase de Planificación. Esto se debe a que en la mayoría de los casos los proyectos se enfocan en presentar un programa funcional y de calidad, el cual satisfaga las necesidades del usuario final, sin considerar en profundidad la planificación de las actividades durante el proceso de desarrollo, o la documentación para posibles actualizaciones futuras.

8.2. Conclusiones.

Al término del presente trabajo de investigación, se puede aseverar el cumplimiento de los objetivos (general y específicos) propuestos dentro del mismo, por lo cual se está en condiciones de validar la tesis inicialmente planteada:

- Se ha propuesto y diseñado un modelo de evaluación de metodologías de desarrollo basado en la metodología propuesta SWIRL, dividiendo su valoración en tres niveles según su grado de rigidez, que permiten valorar el grado de aceptación de los proyectos que han sido implementados bajo las diferentes metodologías ágiles de desarrollo, orientando su puntuación en las actividades generadas durante las fases de la metodología SWIRL.

El modelo de evaluación diseñado es validado mediante la técnica de revisión por terceros, los cuales han de poseer conocimientos de los procesos de desarrollo de software. Así, se realizan encuestas orientadas a la valoración de la afirmación de las preguntas efectuadas dentro del modelo, concluyendo, según los comentarios adheridos por los terceros, que el modelo propuesto establece mecanismos de evaluación que no son considerados comúnmente en otros modelos, permitiendo conocer características que no se aplican en modelos usuales de evaluación.

La validez del modelo se aplica no solo a aplicaciones desarrolladas, sino también en proyectos que se encuentran en etapas de revisión, ya que los resultados permiten garantizar el proceso de desarrollo con menores errores durante las diferentes fases de la metodología empleada para el desarrollo, siendo empleada en la evaluación de diversas aplicaciones para la demostración de su funcionamiento y verificación de los resultados.

- Mediante la comparativa de las diferentes metodologías de desarrollo, las cuales son analizados a profundidad en los estudios previos realizados y publicados por los autores en revistas científicas, se logra obtener las características comunes más relevantes de las metodologías orientadas a la web, y a su vez, definir los indicadores necesarios a valorar dentro del modelo de evaluación.
- La metodología propuesta SWIRL aborda las necesidades principales y específicas que un proyecto de desarrollo orientado a la web presenta, logrando incluir las actividades de otras metodologías que generen beneficios

a los objetivos de la misma, y sirvan para mejorar el proceso de desarrollo llevado a cabo. Para ello fue necesario el análisis de las características destacadas y también de las carencias que las otras metodologías de desarrollo poseen, mediante los estudios previamente realizados y ya publicados.

- Se definen los indicadores y preguntas pertinentes para cada modelo de evaluación, los cuales se clasifican en básico, intermedio y riguroso. Se incrementa así el nivel de impacto de los indicadores para cada nivel de manera ascendente. Dentro de cada modelo, se consideran de mayor interés algunas etapas específicas, es decir, existen etapas que poseen mayor rigidez que otras en los diversos modelos desarrollados.
- Se ha desarrollado una aplicación orientada a la web que permite evaluar de manera cuantitativa los proyectos. La aplicación posee restricciones para sus tipos de usuario, dando accesibilidad al sistema a los usuarios registrados, los cuales pueden crear y evaluar proyectos según el grado de rigurosidad deseado. Además, pueden visualizar los resultados de manera gráfica para cada una de las fases y subcaracterísticas que presenta la metodología híbrida, determinando, al finalizar, si el proyecto cumple con las expectativas de la metodología para un correcto desarrollo o no.

9. FUTUROS DESARROLLOS

Teniendo en cuenta los resultados obtenidos en el trabajo de I+D+i llevado a cabo, sería interesante ampliar el trabajo en lo referente al método de evaluación de la aplicación. Es importante contar con un modelo que evalúe el proceso de desarrollo del proyecto en general, considerando la documentación y desenvolvimiento del grupo y el usuario final. Un punto clave que se evalúa y suscita un gran interés en la actualidad, es la evaluación de las características de accesibilidad, seguridad y optimización de motores de búsqueda (SEO) que presenta el producto una vez que se encuentra lanzado.

Otros aspectos que despiertan gran interés son las facilidades de los productos para atender a los requerimientos éticos y normativos relacionados con la protección y seguridad de los datos, como pueden ser herramientas de anonimización, automatización de datos, etc.

Por lo mencionado anteriormente, se ha planteado una proyección a futuro de implementar un proceso adicional a la evaluación de proyectos, la cual simula el proceso de análisis de herramientas de evaluación de páginas web, las cuales analizan los metadatos ofrecidos por la página web ya publicada y proponen los resultados de seguridad y accesibilidad ante los usuarios. Los resultados serán incluidos dentro de la valoración final del proyecto conjuntamente con el porcentaje obtenido mediante el modelo de evaluación, emitiendo un breve análisis de cada una de las fases según el grado de cumplimiento de las actividades obtenidas en la evaluación.

Otra propuesta que se está empezando, es la realización de modelos de evaluación para proyectos enfocados en aplicaciones móviles, debido a que el desarrollo de apps ha tenido una gran acogida dentro del mercado laboral, y principalmente en el ámbito de entretenimiento e información. Existen actualmente muy pocas técnicas y herramientas que permiten evaluar y conocer el grado de calidad y pertinencia que posee el proyecto (o la “app”) desarrollado.

Respecto a este punto ya se han publicado dos artículos y un libro:

- Comparativa de Metodologías de Desarrollo de Aplicaciones Móviles.
3C Tecnología Glosas de innovación aplicadas a la pyme, 10(2), 73-93.
<https://doi.org/10.17993/3ctecno/2021.v10n2e38.73-93>;
- Estado del Arte: Metodologías de desarrollo de Aplicaciones Móviles.
3C Tecnología. Glosas de innovación aplicadas a la pyme, 10(2), 17-45.
<https://doi.org/10.17993/3ctecno/2021.v10n2e38.17-45>
- “MMS”, Metodología para el Diseño y Desarrollo de Aplicaciones Móviles”.
Colección de Ingeniería y Tecnología de 3ciencias. ISBN 978-84-123661-9-8.
DOI <https://doi.org/10.17993/IngyTec.2021.77>,

REFERENCIAS BIBLIOGRÁFICAS.

- [1] A. Tarhan y S. G. Yilmaz, «Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process», *Information and Software Technology*, vol. 56, n.º 5, pp. 477-494, may 2014, doi: 10.1016/j.infsof.2013.12.002.
- [2] F. K. Y. Chan y J. Y. L. Thong, «Acceptance of agile methodologies: A critical review and conceptual framework», *Decision Support Systems*, vol. 46, n.º 4, pp. 803-814, mar. 2009, doi: 10.1016/j.dss.2008.11.009.
- [3] Y. Wang y L. Zhao, «Eliciting user requirements for e-collaboration systems: a proposal for a multi-perspective modeling approach», *Requirements Eng*, vol. 24, n.º 2, pp. 205-229, jun. 2019, doi: 10.1007/s00766-017-0285-7.
- [4] K. Chari y M. Agrawal, «Impact of incorrect and new requirements on waterfall software project outcomes», *Empir Software Eng*, vol. 23, n.º 1, pp. 165-185, feb. 2018, doi: 10.1007/s10664-017-9506-4.
- [5] E. Robles Luna, G. Rossi, y I. Garrigós, «WebSpec: a visual language for specifying interaction and navigation requirements in web applications», *Requirements Eng*, vol. 16, n.º 4, p. 297, jun. 2011, doi: 10.1007/s00766-011-0124-1.
- [6] Ö. Özcan-Top y F. McCaffery, «To what extent the medical device software regulations can be achieved with agile software development methods? XP—DSDM—Scrum», *J Supercomput*, feb. 2019, doi: 10.1007/s11227-019-02793-x.
- [7] G. Beydoun y G. Low, «Centering ontologies in agent oriented software engineering processes», *Complex Intell. Syst.*, vol. 2, n.º 3, pp. 235-242, oct. 2016, doi: 10.1007/s40747-016-0025-5.
- [8] I. García, C. Pacheco, A. León, y J. A. Calvo-Manzano, «A serious game for teaching the fundamentals of ISO/IEC/IEEE 29148 systems and software engineering – Lifecycle processes – Requirements engineering at undergraduate level», *Computer Standards & Interfaces*, vol. 67, p. 103377, ene. 2020, doi: 10.1016/j.csi.2019.103377.
- [9] W. Wong, C. Lee, y K. Y. Tshai, «The Importance of a Software Development Methodology in IT Project Management: An Innovative Six Sigma Approach - A

- Case Study of a Malaysian SME Organization», sep. 2013. doi: 10.13140/2.1.5106.0807.
- [10] A. Bakhtouchi y R. Rahmouni, «A Tree Decision Based Approach for Selecting Software Development Methodology», en *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*, oct. 2018, pp. 211-216. doi: 10.1109/SaCoNeT.2018.8585699.
- [11] L. Chandi, C. Silva, T. Gualotuña, y D. Martinez, «Model for Selecting Software Development Methodology», en *Proceedings of the International Conference on Information Technology & Systems (ICITS 2018)*, Cham, 2018, pp. 62-73. doi: 10.1007/978-3-319-73450-7_7.
- [12] A. Suliman y N. Nazri, «A new hybrid model of software engineering and systems engineering for embedded system development methodology», en *Proceedings of the 6th International Conference on Information Technology and Multimedia*, nov. 2014, pp. 346-350. doi: 10.1109/ICIMU.2014.7066657.
- [13] M. Kohlbacher, E. Stelzmann, y S. Maierhofer, «Do agile software development practices increase customer satisfaction in Systems Engineering projects?», en *2011 IEEE International Systems Conference*, abr. 2011, pp. 168-172. doi: 10.1109/SYSCON.2011.5929091.
- [14] S. Beecham, T. Clear, R. Lal, y J. Noll, «Do scaling agile frameworks address global software development risks? An empirical study», *Journal of Systems and Software*, vol. 171, p. 110823, ene. 2021, doi: 10.1016/j.jss.2020.110823.
- [15] S. S. M. Monfared y A. Kamandi, «Agile techniques and frameworks based on the requirements for e-commerce applications», en *2016 Second International Conference on Web Research (ICWR)*, abr. 2016, pp. 131-138. doi: 10.1109/ICWR.2016.7498457.
- [16] E. Altameem, «Impact of Agile Methodology on Software Development», *CIS*, vol. 8, n.º 2, p. p9, abr. 2015, doi: 10.5539/cis.v8n2p9.
- [17] C. J. Torrecilla-Salinas, J. Sedeño, M. J. Escalona, y M. Mejías, «Agile, Web Engineering and Capability Maturity Model Integration: A systematic literature review.», *Information and Software Technology*, vol. 71, pp. 92-107, mar. 2016, doi: 10.1016/j.infsof.2015.11.002.
- [18] N. M. Vergara, J. M. T. Linero, y A. V. Moreno, «Model-driven component adaptation in the context of Web Engineering», *Eur J Inf Syst*, vol. 16, n.º 4, pp. 448-459, ago. 2007, doi: 10.1057/palgrave.ejjs.3000691.

- [19] F. J. Domínguez-Mayo, M. J. Escalona, M. Mejías, M. Ross, y G. Staples, «Quality evaluation for Model-Driven Web Engineering methodologies», *Information and Software Technology*, vol. 54, n.º 11, pp. 1265-1282, nov. 2012, doi: 10.1016/j.infsof.2012.06.007.
- [20] J. M. Conejero, R. Rodríguez-Echeverría, F. Sánchez-Figueroa, M. Linaje, J. C. Preciado, y P. J. Clemente, «Re-engineering legacy Web applications into RIAs by aligning modernization requirements, patterns and RIA features», *Journal of Systems and Software*, vol. 86, n.º 12, pp. 2981-2994, dic. 2013, doi: 10.1016/j.jss.2013.04.053.
- [21] «Análisis y Desarrollo Web - Jesús Hernández - Google Libros». https://books.google.com.ec/books?hl=es&lr=&id=nYDVBQAAQBAJ&oi=fnd&pg=PA4&dq=desarrollo+de+software+web+&ots=wQEA4Jeuwz&sig=GxLlrAwqFN DW_TzNx6gm7W7uGF0&redir_esc=y#v=onepage&q=desarrollo%20de%20software%20web&f=false (accedido sep. 20, 2019).
- [22] J. R. Molina Ríos, M. P. Zea Ordóñez, M. J. Contenido Segarra, y F. G. García Zerda, «COMPARACIÓN DE METODOLOGÍAS EN APLICACIONES WEB», *3C Tecnología*, vol. 7, n.º 1, pp. 1-19, mar. 2018, doi: 10.17993/3ctecno.2018.v7n1e25.1-19.
- [23] «Herramientas y buenas prácticas para el aseguramiento de calidad de software con metodologías ágiles - Dialnet». <https://dialnet.unirioja.es/servlet/articulo?codigo=6763102> (accedido sep. 20, 2019).
- [24] «Metodologías ágiles enfocadas al modelado de requerimientos | Informes Científicos Técnicos - UNPA». <http://journal.secyt.unpa.edu.ar/index.php/ICT-UNPA/article/view/494> (accedido sep. 20, 2019).
- [25] «Repositorio Digital de la UTMACH: Diseño de un modelo híbrido para la gestión de procesos de desarrollo de software web basados en rup, scrum, iconix». <http://repositorio.utmachala.edu.ec/handle/48000/10944> (accedido sep. 20, 2019).
- [26] «Método ágil híbrido para desarrollar software en dispositivos móviles». https://scielo.conicyt.cl/scielo.php?pid=S0718-33052015000300016&script=sci_arttext (accedido sep. 20, 2019).
- [27] O. Pinzon, «Ingeniería Web: Una Metodología para el Desarrollo de Aplicaciones Web Escalables y Sostenibles», p. 3.

- [28] «Marco inicial para el diseño y desarrollo de sitios web | SpringerLink». <https://link.springer.com/article/10.1007/s41870-017-0045-4> (accedido sep. 20, 2019).
- [29] «Estado del arte: Metodologías de desarrollo en aplicaciones web - Dialnet». <https://dialnet.unirioja.es/servlet/articulo?codigo=6143045> (accedido sep. 20, 2019).
- [30] «Repositorio Digital de la UTMACH: “Elaboración de una guía de aplicación de métricas de calidad para metodologías ágiles de desarrollo web en www.machalamovil.com”». <http://186.3.32.121/handle/48000/10499> (accedido sep. 20, 2019).
- [31] «Metodología de modelado de aplicaciones web móviles basada en componentes». <http://sedici.unlp.edu.ar/handle/10915/27215> (accedido sep. 20, 2019).
- [32] «Díaz-Pérez - Metodología estructurada para el desarrollo de apl.pdf». Accedido: sep. 20, 2019. [En línea]. Disponible en: <https://biblat.unam.mx/hevila/RevistaEthosvenezolana/2016/vol8/no2/7.pdf>
- [33] Ana Nieves del Valle Rodríguez, *METODOLOGÍAS DE DISEÑO USADAS EN INGENIERÍA WEB, SU VINCULACIÓN CON LAS NTICS*. UNIVERSIDAD NACIONAL DE LA PLATA, 2009. [En línea]. Disponible en: https://s3.amazonaws.com/academia.edu.documents/44488368/Rodriguez_Ana.pdf?response-content-disposition=inline%3B%20filename%3DINGENIERIAD_WEB.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20190921%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190921T184837Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=5126b5cafa234847ac405b7f0d6e2be0c0a78919a72b1dc98104ef2e67a098e9
- [34] «ISO 25010». <https://iso25000.com/index.php/normas-iso-25000/iso-25010?limit=3&start=6> (accedido sep. 21, 2019).
- [35] J. Molina-Ríos y N. Pedreira-Souto, «Comparison of development methodologies in web applications», *Information and Software Technology*, vol. 119, p. 106238, mar. 2020, doi: 10.1016/j.infsof.2019.106238.

- [36] J. Molina Ríos y N. Pedreira-Souto, «Approach of Agile Methodologies in the Development of Web-Based Software», *Information*, vol. 10, n.º 10, p. 314, oct. 2019, doi: 10.3390/info10100314.
- [37] K. Ordoñez, J. Hilera, y S. Cueva, «Model-driven development of accessible software: a systematic literature review», *Univ Access Inf Soc*, sep. 2020, doi: 10.1007/s10209-020-00751-6.
- [38] J. R. Molina Ríos y M. de las N. Pedreira-Souto, «SWIRL», *metodología para el diseño y desarrollo de aplicaciones web*, 1.ª ed. Editorial Científica 3Ciencias, 2019. doi: 10.17993/IngyTec.2019.55.
- [39] M. Brhel, H. Meth, A. Maedche, y K. Werder, «Exploring principles of user-centered agile software development: A literature review», *Information and Software Technology*, vol. 61, pp. 163-181, may 2015, doi: 10.1016/j.infsof.2015.01.004.
- [40] B. Gil y D. Pascual-Ezama, «La metodología Delphi como técnica de estudio de la validez de contenido», *analesps*, vol. 28, n.º 3, pp. 1011-1020, oct. 2012, doi: 10.6018/analesps.28.3.156211.
- [41] I. Pedrosa, J. Suárez-Álvarez, y E. García-Cueto, «Evidencias sobre la Validez de Contenido: Avances Teóricos y Métodos para su Estimación [Content Validity Evidences: Theoretical Advances and Estimation Methods]», *Acción psicol.*, vol. 10, n.º 2, p. 3, jun. 2014, doi: 10.5944/ap.10.2.11820.
- [42] M. Cohn, «SPIDR Poster Download», *Mountain Goat Software*. <https://www.mountangoatsoftware.com/exclusive/spidr-poster-download> (accedido may 10, 2020).

