



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UNA APLICACIÓN WEB PARA LA EVALUACIÓN
DE CALIDAD DEL SOFTWARE BASADO EN LA NORMA ISO 25000

VILLARROEL ACARO RODRIGO ALEXANDER
INGENIERO DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UNA APLICACIÓN WEB PARA LA
EVALUACIÓN DE CALIDAD DEL SOFTWARE BASADO EN LA
NORMA ISO 25000

VILLARROEL ACARO RODRIGO ALEXANDER
INGENIERO DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

DESARROLLO DE UNA APLICACIÓN WEB PARA LA EVALUACIÓN DE
CALIDAD DEL SOFTWARE BASADO EN LA NORMA ISO 25000

VILLARROEL ACARO RODRIGO ALEXANDER
INGENIERO DE SISTEMAS

VALAREZO PARDO MILTON RAFAEL

MACHALA, 29 DE SEPTIEMBRE DE 2021

MACHALA
2021

DESARROLLO DE UNA APLICACIÓN WEB PARA LA EVALUACIÓN DE CALIDAD DEL SOFTWARE BASADO EN LA NORMA ISO 25000

INFORME DE ORIGINALIDAD

9%

INDICE DE SIMILITUD

7%

FUENTES DE INTERNET

2%

PUBLICACIONES

3%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	wicc.it.itba.edu.ar Fuente de Internet	1%
2	Submitted to UNILIBRE Trabajo del estudiante	<1%
3	docplayer.es Fuente de Internet	<1%
4	repositorio.ucp.edu.pe Fuente de Internet	<1%
5	www.ing.unlpam.edu.ar Fuente de Internet	<1%
6	Robin Valarezo, Teresa Guarda. "Comparative analysis of the laravel and codeigniter frameworks: For the implementation of the management system of merit and opposition competitions in the State University Península de Santa Elena", 2018 13th Iberian	<1%

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, VILLARROEL ACARO RODRIGO ALEXANDER, en calidad de autor del siguiente trabajo escrito titulado DESARROLLO DE UNA APLICACIÓN WEB PARA LA EVALUACIÓN DE CALIDAD DEL SOFTWARE BASADO EN LA NORMA ISO 25000, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 29 de septiembre de 2021



VILLARROEL ACARO RODRIGO ALEXANDER
0704870054

DEDICATORIA

Dedico este trabajo con todo mi esfuerzo y cariño a mi madre que siempre ha sido mi mayor motivación y pilar indispensable en mi formación académica y personal, a mi familia que siempre están apoyándome en cada uno de mis pasos para poder cumplir mis metas.

Dedico este trabajo a todas aquellas personas que, a pesar de mis tropiezos, me brindaron su confianza y apoyo incondicional para la culminación de este trabajo.

Sr. Rodrigo Alexander Villarroel Acaro

AGRADECIMIENTO

Un profundo agradecimiento a mi familia, en especial a mi madre por su comprensión y apoyo incondicional, a mis amigos y compañeros por compartir generosamente sus conocimientos y colaborar en el desarrollo del presente trabajo. A los docentes la carrera de Ingeniería de Sistemas por sus enseñanzas para desarrollarme profesionalmente

Sr. Rodrigo Alexander Villarroel Acaro

RESUMEN

En la actualidad la mayoría de los procesos cotidianos se están automatizando, y con ello existe una gran demanda de aplicaciones web que ayudan a los usuarios finales a realizar sus actividades, entonces surge la necesidad de implementar sistemas seguros, confiables y que cumplan con normativas de calidad de software muy importantes en el desarrollo; sin importar la escala del sistema o la cantidad de usuarios conectados en simultáneo, estos sistemas deben asegurar que todos los tipos de riesgos para los usuarios estén debidamente controlados. Es así que los sistemas de evaluación de aplicaciones web, tienen una gran acogida por parte de los usuarios interesados en obtener páginas accesibles, funcionales y seguras.

El presente trabajo de investigación propone desarrollar una plataforma web (Isosoft) que evalúe la calidad de software mediante el modelo de calidad ISO 25010, para lo cual primero se diseñaron las pantalla del sistema mediante la herramienta de diseño Adobe XD, en donde se pudo plasmar las ideas que se tenían en papel a diseños con características en específico más visibles y con mayor interacción, en el desarrollo de la programación se utilizó el visor de código Visual Studio Code ya que este visor ofrece muchas más funcionalidades que el resto, ayudando al programador a realizar su trabajo de forma más rápida y sencilla, en cuanto a las herramientas de programación se utilizó React en el FrontEnd para el diseño de las interfaces de usuario con las librerías Framer Motion, Material Ui, Api context entre otras; mientras que en el BackEnd se utilizó Laravel para los controladores del sistema hacia la base de datos con el gestor PostgreSQL ya que se implementó una estructura SQL debido a que está orientado a objetos, y se enfoca en las funcionalidades de escalabilidad y cumplimiento de estándares. Finalmente, para el manejo de seguridad de la información de la base de datos se optó por aplicar la encriptación de datos desde el gestor de base de datos mediante la herramienta Heroku que usa encriptación SSL, en donde se cifran los datos para el almacenamiento de información y cumple con los requisitos de seguridad.

Una vez que se desarrolló el sistema, se procedió a la evaluación del mismo en base a herramientas especializadas en la evaluación de calidad del software, como, por ejemplo, GTMetrix, Validator, Pingdom, que son los instrumentos más utilizados para este tipo de evaluaciones por su precisión y confianza en los resultados. Como

resultado de estas evaluaciones se obtuvo que la plataforma IsoSoft cumple con un 61% de rendimiento y un 96% de estructura según GTMetrix, también se reflejó que en usabilidad está en alrededor del 67% según MetricSpot y además en la herramienta Pingdom alcanzó un total de 85 puntos, concluyendo que después de haber realizado varias pruebas de evaluación del software, es un sistema que cumple con los parámetros de calidad de la ISO/IEC 25010, como funcionalidad, usabilidad, seguridad, portabilidad entre otros; obteniendo resultados confiables de páginas que se especializan en evaluar el funcionamiento con los estándares de calidad.

Palabras Clave: Aplicación web, calidad de software, ISO/IEC 25010, Heroku, React.

ABSTRACT

Currently, most of the daily processes are being automated, and with this there is a great demand for web applications that help end users to carry out their activities, then the need arises to implement safe, reliable systems that comply with regulations of software quality very important in development; Regardless of the scale of the system or the number of users connected simultaneously, these systems must ensure that all types of risks for users are properly controlled. Thus, web application evaluation systems are well received by users interested in obtaining accessible, functional and secure pages.

This research work proposes to develop a web platform (Isosoft) that evaluates software quality using the ISO 25010 quality model, for which the system screens were first designed using the Adobe XD design tool, where it was possible to capture the ideas that were had on paper to designs with specific characteristics more visible and with greater interaction, in the development of the programming the Visual Studio Code code viewer was used since this viewer offers many more functionalities than the rest, helping the programmer to carry out their work more quickly and easily, in terms of the programming tools, React was used in the FrontEnd for the design of the user interfaces with the Framer Motion, Material Ui, Api context libraries among others; While in the BackEnd, Laravel was used for the system controllers towards the database with the PostgreSQL manager, since an SQL structure was implemented because it is object-oriented, and focuses on scalability and standards compliance functionalities. Finally, for the security management of the database information, it was decided to apply data encryption from the database manager through the Heroku tool that uses SSL encryption, where the data is encrypted for information storage and meets the safety requirements.

Once the system was developed, it was evaluated based on specialized tools in the evaluation of software quality, such as, for example, GTMetrix, Validator, Pingdom, which are the most used instruments for this type of evaluation. for its precision and confidence in the results. As a result of these evaluations, it was obtained that the Isosoft platform complies with a 61% performance and a 96% structure according to GTMetrix, it was also reflected that in usability it is around 67% according to

MetricSpot and also in the Pingdom tool it reached a total of 85 points, concluding that after having carried out several evaluation tests of the software, it is a system that complies with the quality parameters of ISO / IEC 25010, such as functionality, usability, security, portability, among others; obtaining reliable results from pages that specialize in evaluating performance with quality standards.

Keywords: Web application, software quality, ISO / IEC 25010, Heroku, React.

ÍNDICE DE CONTENIDO

DEDICATORIA	1
AGRADECIMIENTO	2
RESUMEN	3
ABSTRACT	4
ÍNDICE DE FIGURAS	7
ÍNDICE DE TABLAS	8
INTRODUCCIÓN	9
1. CAPÍTULO I. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS	11
1.1. ÁMBITO DE APLICACIÓN: DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS	11
1.2. ESTABLECIMIENTO DE REQUERIMIENTOS	12
1.3. JUSTIFICACIÓN DEL REQUERIMIENTO A SATISFACER	12
2. CAPÍTULO II. DESARROLLO DE PROTOTIPO	14
2.1. DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO	14
2.2. Fundamentación teórica del prototipo	15
2.2.1. Desarrollo de Software	15
2.2.2. Desarrollo Ágil	17
2.2.3. Metodología Ágiles	18
2.2.4. Aplicaciones Web	21
2.2.5. Calidad del Software	22
2.2.6. ISO/IEC 25000	23
2.2.7. Lenguajes de Programación	24
2.2.8. Bases de datos SQL	28
2.2.9. Sistema de gestor de base de datos SQL	28
2.2.10. Seguridad de base de datos	29
2.2.11. Diseño de interfaces	29
2.2.12. IDE de Programación	30

2.3.	Objetivos del Prototipo	30
2.3.1.	Objetivo General	30
2.3.2.	Objetivos Específicos	31
2.4.	Diseño del Prototipo	31
2.4.1.	Tecnologías de desarrollo	31
2.4.2.	Prototipado de las interfaces	32
2.5.	Ejecución y/o ensamblaje del prototipo	35
2.5.1.	Pantalla principal	37
2.5.2.	Pantalla registro	38
2.5.3.	Pantalla Login del sistema	38
2.5.4.	Pantalla dashboard	39
2.5.5.	Pantalla sistemas	39
2.5.6.	Pantalla crear sistema	40
2.5.7.	Pantalla Evaluaciones	40
2.5.8.	Pantalla nueva evaluación	41
2.5.9.	Visualización de los Resultados	42
2.5.10.	Pantalla configuración de cuenta	43
2.5.11.	Configuración de Heroku	44
2.5.12.	Generación de Backups	44
3.	CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO	46
3.1.	Plan de evaluación	46
3.2.	Resultados de la evaluación	47
3.2.1.	GTmetrix	47
3.2.2.	Validator (HTML Y CSS)	47
3.3.	Conclusiones	48
3.4.	Recomendaciones	48
4.	BIBLIOGRAFÍA	50

ÍNDICE DE FIGURAS

Figura 1: Arquitectura del prototipo tecnológico	14
Figura 2: Diagrama de Proceso de Desarrollo de software de Software.	16
Figura 3: Ciclo de desarrollo de software	17
Figura 4: Proceso de un Desarrollo Ágil	18
Figura 5: Metodología Ágil y Metodología Tradicional	19
Figura 6: Procesos de la Metodología Scrum	20
Figura 7: Arquitectura de una aplicación web	22
Figura 8: Estructura de la calidad de software	23
Figura 9: Modelo de calidad del Software	24
Figura 10: Prototipo - Pantalla principal	32
Figura 11: Prototipo - Pantalla registro	32
Figura 12: Prototipo - Pantalla Login	33
Figura 13: Prototipo - Pantalla Evaluaciones	33
Figura 14: Prototipo - Pantalla Sistemas	34
Figura 15: Prototipo - Pantalla crear sistema	34
Figura 16: Pantalla principal del sistema	38
Figura 17: Pantalla Registro	38
Figura 18: Pantalla login	39
Figura 19: Pantalla Dashboard	39
Figura 20: Pantalla Sistemas	40
Figura 21: Pantalla crear sistema	40
Figura 22: Pantalla Evaluaciones	41
Figura 23: Pantalla nueva evaluación	41
Figura 24: Preguntas evaluación, norma ISO 27010	42
Figura 25. Puntuación general de la evaluación	42
Figura 26: Puntuación por característica	43
Figura 27: Pantalla configuración de cuenta	43
Figura 28: Configuración de Heroku	44
Figura 29: Generación de Backups	45
Figura 30: Encriptación de la base de datos	45
Figura 31: Anexo A: Herramienta GTmetrix – Evaluación Eficiencia	55

Figura 32: Herramienta Validator – Evaluación Funcionabilidad	55
Figura 33: Herramienta FAE – Evaluación Accesibilidad	56
Figura 34: Herramienta MetricSpot – Evaluación Usabilidad	56
Figura 35: Herramienta Grader – Evaluación Eficiencia-Seguridad-Portabilidad	57
Figura 36: Herramienta Grader – Evaluación Rendimiento	57
Figura 37: Herramienta Grader – Evaluación Portabilidad	58
Figura 38: Herramienta Grader – Evaluación Seguridad	58
Figura 39: Herramienta Pingdom – Evaluación Eficiencia	59

ÍNDICE DE TABLAS

Tabla 1: Tecnologías implementadas	31
Tabla 2: Métrica/preguntas utilizadas para la evaluación de calidad	35
Tabla 3: Escala de Likert para evaluación de calidad	37

INTRODUCCIÓN

Las empresas dedicadas a ofrecer aplicaciones webs, actualmente deben desarrollar aplicaciones con un grado de calidad muy superior, pues con la evolución de la tecnología, cada vez se están desarrollando aplicaciones que cumplan con normas que certifiquen la eficiencia y seguridad del software. Es por ello que se hace imprescindible medir la calidad de todas estas aplicaciones que manejan información crucial que circula por la internet, es aquí en donde esta labor resulta un poco difícil por lo que al momento de diseñar o programar sistemas automatizados se vuelve una labor compleja ya que ciertos profesionales lo pasan por alto, entonces se recomienda utilizar modelos de calidad que ayuden al desarrollo de aplicativos con alta calidad.

En el desarrollo de software los defectos o errores pueden ocurrir en cualquier momento, esto puede ocasionar que los requerimientos del cliente no se cumplan a cabalidad. Los defectos pueden ser de diseño, lógicos, procesos, tiempos, codificación, etc. Por consiguiente, el asegurar la calidad de software es un aspecto esencial en el desarrollo de sistemas. La calidad de software comienza con la evaluación de código hasta el resultado final del producto, mediante la aplicación de métricas o normas.[1]

Para confirmar que las funcionalidades de las aplicaciones web cumplen con todos los estándares de calidad, es necesario realizar una evaluación mediante las ISO de calidad del software; para efectuar este trabajo se utilizó la ISO 25010, que según [2] es un estándar de calidad que forma parte de la serie SQuaRE, y está compuesto por características (adecuación funcional, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad, portabilidad y eficiencia en el desempeño), que a su vez se subdividen en sub características relacionadas con el resultado de la aplicación a evaluar.

El presente proyecto se encuentra estructurado en tres capítulos principales, el capítulo I describe el contexto del tema, los requerimientos del prototipo, así como también la justificación del por qué se realizó este trabajo, mientras que en el capítulo II se define la estructura del prototipo con su debida fundamentación teórica respaldada en repositorios científicos, logrando cumplir con los objetivos planteados y desarrollar el producto final. En el último capítulo se desarrollan los resultados

obtenidos y también la evaluación del prototipo en base a las métricas propuestas según la ISO/IEC 25010.

1. CAPÍTULO I. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS

1.1. ÁMBITO DE APLICACIÓN: DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS

Hoy en día se está viviendo una era tecnológica que avanza a pasos gigantes, pues si bien es cierto que por factores epidemiológicos el uso de las herramientas virtuales se hizo imprescindible, esto ayudó a que todas las actividades cotidianas sean sistematizadas, por lo que se vio un aumento en el desarrollo de aplicaciones web. Según datos de [3], hasta principios del año 2021, los usuarios de internet aumentaron en un 1,5%, de los cuales la mayor parte son de usuarios que realizan “home office” desde páginas webs. Lo que nos lleva a preguntarnos si dichas páginas están siendo desarrolladas con las debidas normas de calidad de software.

La calidad de software es fundamental en el proceso de ciclo de vida del desarrollo de un sistema/producto. De modo que para el desarrollo y diseños de sitios web también es requerido asegurar la calidad del mismo. La calidad de un sitio web permite analizar qué tan eficiente es para los usuarios finales.

El hecho de que el desarrollo de aplicativos webs esté aumentando enormemente, nos hace surgir la necesidad de realizar un software que ayude a los desarrolladores a medir la calidad de su sistema, verificando si cumple o no con las características de la ISO/IEC 25010 la cual asegura la calidad de los sistemas.

Por lo tanto, la presente propuesta tecnológica tiene como propósito desarrollar una aplicación web que permita la evaluación de software basado en la norma de calidad ISO/IEC 25010, la cual permita evidenciar que el software desarrollado cumpla con los estándares de calidad y se adecue a los requerimientos de los usuarios finales, haciendo uso de la metodología de desarrollo ágil Scrum, utilizando el framework Laravel para el desarrollo del Backend, además de React.js para el diseño de interfaces y para el almacenamiento de información, el motor de base de datos PostgreSQL.

1.2. ESTABLECIMIENTO DE REQUERIMIENTOS

En la actualidad, los sitios web son algo común en nuestras vidas cotidianas. La gran mayoría de empresas tienen su sitio web, también los usuarios hacen uso de

sitios web para sus negocios o para realizar compras. Cada sitio web tiene sus propias características y requerimientos. De modo que los desarrolladores de estas aplicaciones deben asegurar que estos sitios sean los más eficientes, adaptables y usables para el usuario final. Además, de evitar que se presenten errores en la misma.

El sitio web para evaluar la calidad de software basado en la norma ISO 25010 ayudará a desarrolladores o personas interesadas, a evaluar de manera cuantitativa la calidad de los sitios web en base a una serie de preguntas que son obtenidas a partir de las métricas de la ISO en mención, de esta forma se puede asegurar que se desarrolla un software de calidad o mejorar de manera significativa en las características que obtengan bajos resultados

El usuario responsable de la evaluación de los sitios web es el encargado de responder las preguntas que pertenecen a las sub características de las ocho métricas de calidad que corresponde a norma ISO 25010 las cuales son: funcional, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad, portabilidad y eficiencia en el desempeño, se mostrarán los resultados obtenidos de la evaluación de forma gráfica para su análisis y toma de decisiones en cuanto al mejoramiento de la aplicación.

1.3. JUSTIFICACIÓN DEL REQUERIMIENTO A SATISFACER

El desarrollo de software en la actualidad es una parte común y cotidiana de nuestra vida, ya sea como desarrolladores de sistemas o como usuarios que hacemos uso de los mismos. Un software es un conjunto de instrucciones, programas, procedimientos, reglas, documentación que están relacionados con procesamientos de datos, que permiten llevar determinadas tareas a componentes físicos, también llamados hardware.[4][5]

La garantía de la calidad del software es una parte fundamental en el éxito del proceso de desarrollo del mismo. La calidad de software es un tema que surge desde muchos años atrás, en contraposición del desarrollo del software. Es una parte esencial que puede llegar a ser la clave del éxito del proceso de desarrollo de un sistema. Surge con la finalidad de que los sistemas estén libres de errores, en lo mayor posible, y sean eficientes. El asegurar la calidad de software debe llevarse

antes, durante y después del ciclo de vida de desarrollo de un software. Como lo hace notar [4], “la evaluación de la calidad del software es crucial en las primeras fases, ya que puede ayudarnos a reducir el esfuerzo, el tiempo y el dinero.”

La norma ISO 25010 es un modelo de calidad de software que nos ayuda en el proceso de evaluación de calidad de un sistema por medio de características propias de esta ISO como: funcional, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad, portabilidad y eficiencia en el desempeño. [6]

El principal objetivo de la presente propuesta tecnológica es desarrollar una plataforma web para la evaluación de calidad del software, mediante las características de la norma ISO 25010, la evaluación se realizará respondiendo cuantitativamente una serie de preguntas que corresponden a las métricas de la norma, además permitirá a las personas interesadas en asegurar la calidad del sitio web visualizar los resultados obtenidos por medio de tablas y gráficos estadísticos.

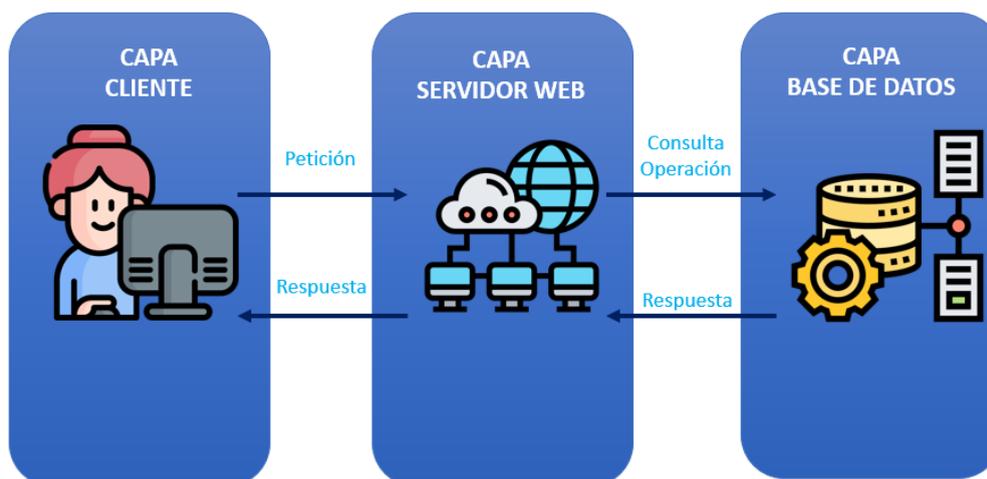
2. CAPÍTULO II. DESARROLLO DE PROTOTIPO

2.1. DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO

El modelo de la propuesta tecnológica está basado en la arquitectura cliente-servidor. La plataforma web consiste en la evaluación de calidad de software de los sitios web, para lo cual se emplea una serie de preguntas que serán contestadas en base a cada sitio web que se va a evaluar. Dichas preguntas son realizadas en base al modelo de calidad de software ISO 25010. Al final se presentará el resultado de la evaluación del sitio web evaluado.

La propuesta consta de tres capas que son cliente, servidor y datos como se puede observar en la Figura 1.

Figura 1: Arquitectura del prototipo tecnológico



Fuente: Elaboración propia

La arquitectura de la plataforma web para la evaluación de calidad de software se basa en tres capas, las que se mencionan a continuación:

- **Capa cliente:** En esta capa se presenta de manera visual la plataforma web para que el cliente pueda realizar la evaluación de calidad de software. Es decir, es la interfaz de usuario, en este caso del sitio web. En esta capa se utiliza el framework React, el cual permite la creación de interfaces de usuarios interactivas y sencillas.

- **Capa servidor web:** En esta capa se realizan las peticiones solicitadas por el usuario. Se utiliza el entorno de trabajo de Laravel que permite el desarrollo de aplicaciones bajo el lenguaje de programación PHP.
- **Capa de datos (base de datos):** Esta capa contiene la información de la plataforma web, en este caso el almacenamiento de las métricas de evaluación del modelo de calidad de software. En la propuesta se utiliza la base de datos PostgreSQL ya que permite una rápida respuesta y almacena grandes cantidades de información.

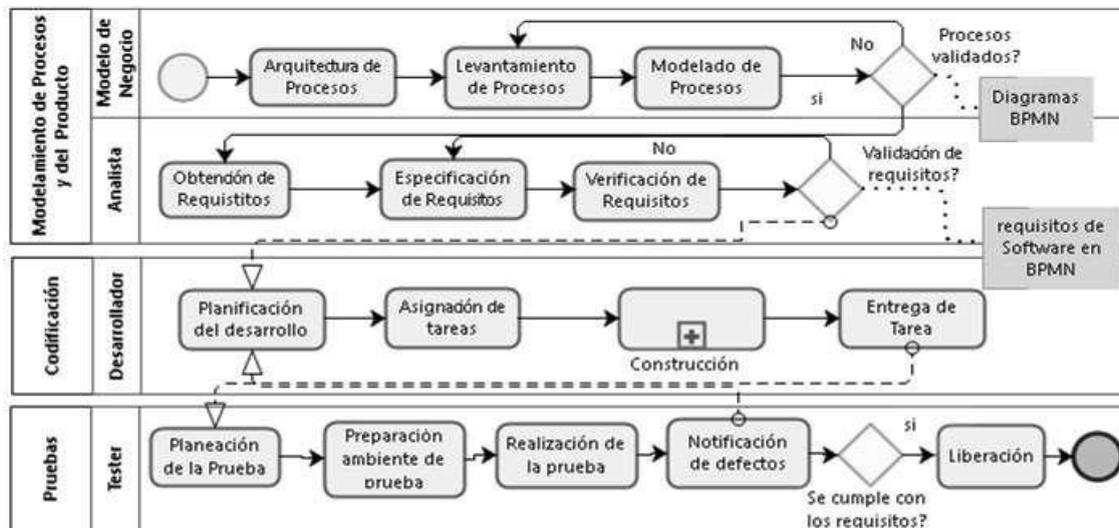
2.2. Fundamentación teórica del prototipo

2.2.1. Desarrollo de Software

El desarrollo de software es un proceso que se enfoca en el desarrollo de software de manera ordenada mediante la aplicación de etapas o continuas. De acuerdo con [7], el desarrollo de software implica aplicar ciertos modelos que ayuden a dirigir tareas, diseños, construcción, pruebas, mantenimiento, modificaciones, de los sistemas a desarrollar. Antes de la aparición de los métodos de desarrollo de sistemas, el desarrollo de nuevos sistemas o nuevos productos solía llevarse a cabo recurriendo a la experiencia e intuición del personal directivo y técnico. Sin embargo, la complejidad de los sistemas y productos informáticos modernos ya ha puesto de manifiesto la necesidad de un proceso de desarrollo ordenado.

En base a [8], “las organizaciones de desarrollo de software se han interesado en alcanzar niveles de capacidad en sus procesos para obtener la madurez organizacional y se presenta una relación positiva entre los modelos de procesos de certificación como CMMI e ISO 9000, y características de software de calidad tales como fiabilidad, prueba, usabilidad, eficiencia e integridad.”

Figura 2: Diagrama de Proceso de Desarrollo de software de Software.



Fuente:[8]

La programación informática es el elemento principal en el desarrollo de software, tal proceso lo lleva a cabo los programadores de software, esto incluye desde la investigación preliminar del sistema, diseño de procesos, diagramas, maquetación y programación. El desarrollo de un software puede ser un sistema independiente o parte de un sistema más grande. Además del desarrollo de software también se enfoca en el diseño del mismo y desarrollo de aplicaciones.

El desarrollo de software permite que las tecnologías digitales puedan ser programables. El software es considerado la infraestructura intangible, del cual se desarrollan recursos asociados a procesamiento, redes y datos, esto implica la gran adaptación de programas y aplicaciones de software para satisfacer las necesidades específicas de los usuarios. Por consiguiente, el software desempeña un papel fundamental que ayuda a que las empresas puedan automatizar procesos e implementar nuevas tecnologías para poder abarcar más mercado. Así mismo, los usuarios pueden utilizar tecnologías existentes que permiten expandirse en este mundo digital. [9]

Según [10] “El desarrollo de software seguro se lleva a cabo mediante la ejecución de un conjunto de actividades de ingeniería de seguridad junto con los procesos de desarrollo de software. Esto se hace siguiendo un modelo de ciclo de vida de desarrollo de seguridad, o una implementación de un modelo de madurez de seguridad.”

Figura 3: Ciclo de desarrollo de software



Fuente: [11]

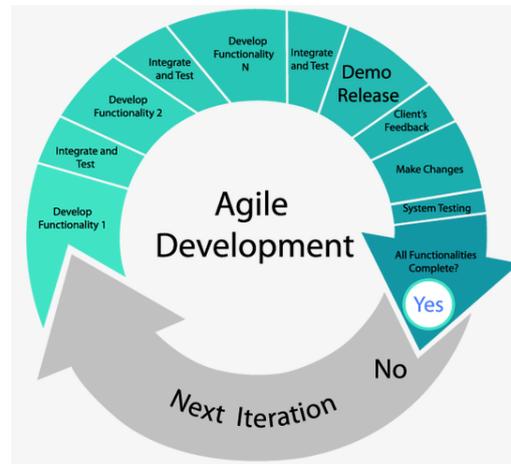
2.2.2. Desarrollo Ágil

El desarrollo ágil es un método el cual se enfoca en el desarrollo de software de manera flexible y con más rapidez. Además, de incorporar desarrollos iterativos para incrementar y garantizar una entrega infalible y acelerada. De acuerdo con [12], con la aceptación del desarrollo ágil surgen las metodologías ágiles que son “un marco de procedimientos y herramientas se conciben como una filosofía, una práctica orientada a las personas no a los procesos, en prospectiva se mueven hacia un ambiente de “Post-agilismo””. Con la facilidad que el desarrollo ágil permite el seguimiento hacia atrás de un proyecto, facilita a los equipos de desarrollo puedan componer características más amplias dentro de un tiempo definido. Por el contrario, los métodos tradicionales no permiten este seguimiento hacia atrás, como por ejemplo el método Cascada; el cual dominaba el sector de desarrollo de software, pero en la actualidad ya no tiene esa gran acogida que tenía en sus inicios. Este modelo de Cascada no permite agregar nuevas características por lo que empezó a decaer, además, su mantenimiento puede resultar complejo.

Según [13], en este tipo de desarrollo, es fundamental tener en cuenta la visión del cliente, poniendo énfasis en el satisfacer sus necesidades y requisitos. Un punto importante es la aplicación de pruebas de software ya que desempeñan un papel fundamental e involucran a varios del equipo de desarrollo incluyendo al propio

cliente. Las metodologías Scrum y XP son las más utilizadas dentro de esta comunidad.

Figura 4: Proceso de un Desarrollo Ágil



Fuente:[14]

2.2.3. Metodología Ágiles

Las metodologías ágiles son un método de desarrollo de software que se centra en los valores y principios en el Manifiesto Ágil para el Desarrollo de Software. Este tipo de metodología está diseñada para que los productos sean entregados de manera correcta, a través de pequeños equipos que están organizados para entregas de piezas pequeñas de funcionalidad con frecuencia y de manera incremental, lo que permite comentarios frecuentes de los clientes y correcciones según sea necesario.

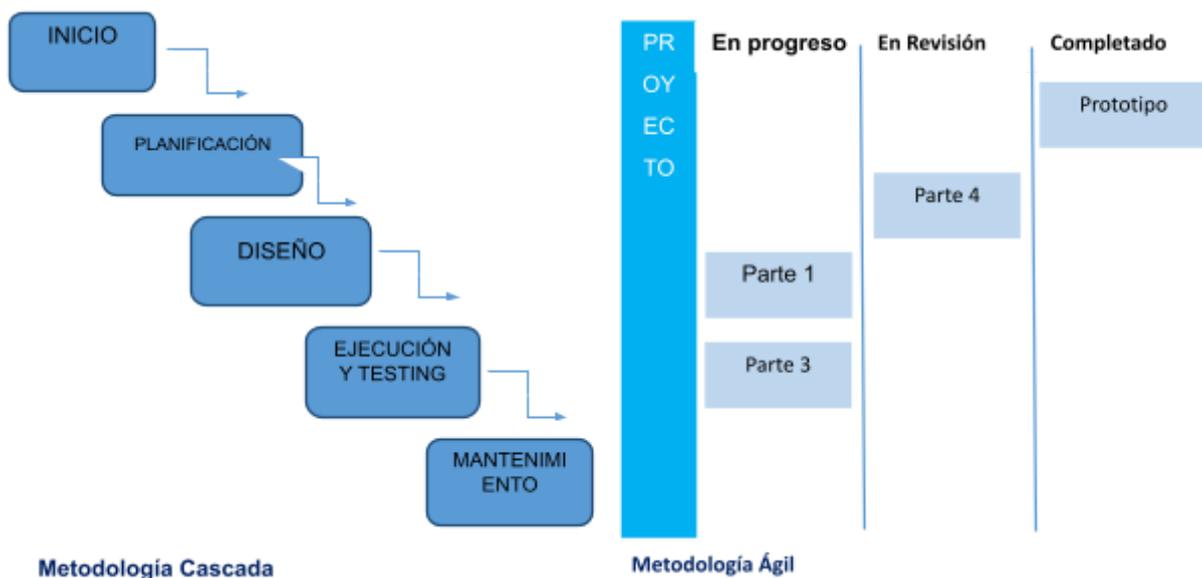
De acuerdo con [15], menciona que “las metodologías ágiles se crearon para mitigar las limitaciones de los métodos tradicionales de desarrollo de software, como el exceso de documentación o la ineficacia del time to market. Básicamente, surgieron para responder a la ineficacia de los métodos actuales de desarrollo de software.” La metodología ágil trata de responder al mercado y a los clientes enfocada en responder las necesidades y requisitos de los usuarios; además, de poder realizar cambios cuando la situación lo requiera. Procesos de planificación, gestión y técnicos son términos que se han incorporado a estas metodologías, los cuales son necesarios para gestionar proyectos de manera iterativa para el desarrollo del software.

En resumen, [16] las metodologías ágiles tienen ventajas sobre las metodologías tradicionales, ya que las primeras permiten una gestión más ágil y flexible cuando se

refiere a la implementación de cambios en el desarrollo del software. Además, permiten anteponer las tareas según sea necesario, se cuenta con la participación activa y directa con el usuario, con lo que se supone una mejor colaboración del proyecto. De modo que la retroalimentación de los resultados de las pequeñas entregas conlleva a un resultado esperado. En comparación con las metodologías tradicionales funcionan en cascada. Lo que implica una planificación a largo plazo, lo que puede generar que exista poca actividad y cuellos de botella entre equipos implicados en el desarrollo de software. Las metodologías ágiles permiten establecer pequeñas metas y poder cumplirlas en poco tiempo, esto al final permite una mejor retroalimentación antes que finalice el proyecto.

De acuerdo con [17], destaca que “Las metodologías ágiles son especialmente utilizadas en proyectos informáticos, es decir, con más enfoque en el desarrollo de software. Por su facilidad de ser adaptativo e implementar ciclos iterativos, la metodología ágil es una buena opción en el proceso de desarrollo de software.”

Figura 5: Metodología Ágil y Metodología Tradicional



Fuente: [16]

2.2.3.1. Metodología Scrum

La metodología Scrum es un marco de métodos ágiles de las más utilizadas en la actualidad. En este tipo de metodología los procesos se dividen en ciclos de desarrollo iterativos, los cuales son denominados Sprint. A inicios del sprint se da

prioridad a la recolección de los requisitos existentes y problemas en el desarrollo del producto. Las entregas en esta metodología se realizan con el afán de verificar los procesos de desarrollo de software y los posibles cambios que se presenten en el transcurso del mismo.[18]

Scrum es operado por un pequeño equipo de 7 a 9 personas, incluido un Scrum Master y un propietario de producto. Scrum es un marco de proceso para gestionar el desarrollo de productos y otros trabajos de conocimiento. Scrum es empírico porque proporciona una forma para que el equipo asuma cómo creen que funciona algo, lo prueba, reflexiona sobre la experiencia y realiza los ajustes necesarios. La estructura de Scrum permite al equipo incorporar prácticas de otros marcos que son significativos para el entorno del equipo.

Figura 6: Procesos de la Metodología Scrum



Fuente: [19]

Dentro de una metodología Scrum se aprecian dos factores importantes que son los actores y las acciones. Entre los actores tenemos: Product Owner o propietario, el cual se encarga de identificar los objetivos y establecer prioridades. Scrummaster o facilitador, se concentra en asegurar el funcionamiento correcto de la metodología. Scrum Team, comprende el equipo de desarrollo del producto. Encargados de cumplir lo establecido por el Product Owner. [16]

En esta metodología también se toman en cuenta las reuniones, las cuales mencionamos a continuación. Sprint planning o planificación, de este tipo de reunión se establecen las acciones a tomar en las posteriores; organización del trabajo.

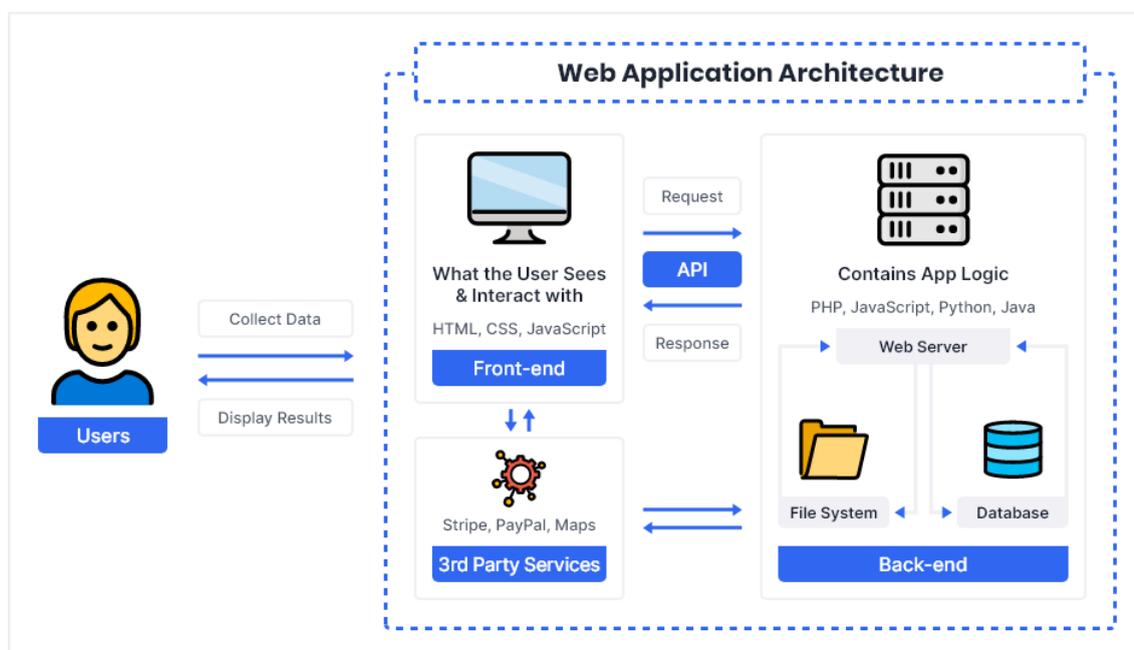
Daily Scrum o seguimiento diario de Sprint, como se destaca en su nombre, hace referencia al seguimiento diario del proceso de desarrollo. El equipo de trabajo informa que es lo que ha realizado en el tiempo establecido. Review/Retrospective o revisión del Sprint, se analiza todas las sugerencias y valoración del trabajo desempeñado. [16]

2.2.4. Aplicaciones Web

“Las aplicaciones web son cada vez más populares en los últimos años. En el año 2018, más de 3.900 millones (alrededor del 51%) de personas en el mundo tienen acceso a internet.” [20]

Las aplicaciones web, requieren de ciertas metodologías de desarrollo. En estas metodologías se recopilan los requerimientos y necesidades del cliente. Los sistemas web podemos destacar dos aspectos esenciales que conforman parte de una aplicación web las que denominamos como back-end y front-end. El back.end hace referencia a la parte lógica del sistema, es decir, la estructura interna del sitio. El front-end es la parte visual del sitio, es la que el usuario visualiza al navegar en un sitio. Los sitios web pueden ser dinámicas y estáticas.[21]

Figura 7: Arquitectura de una aplicación web



Fuente:[22]

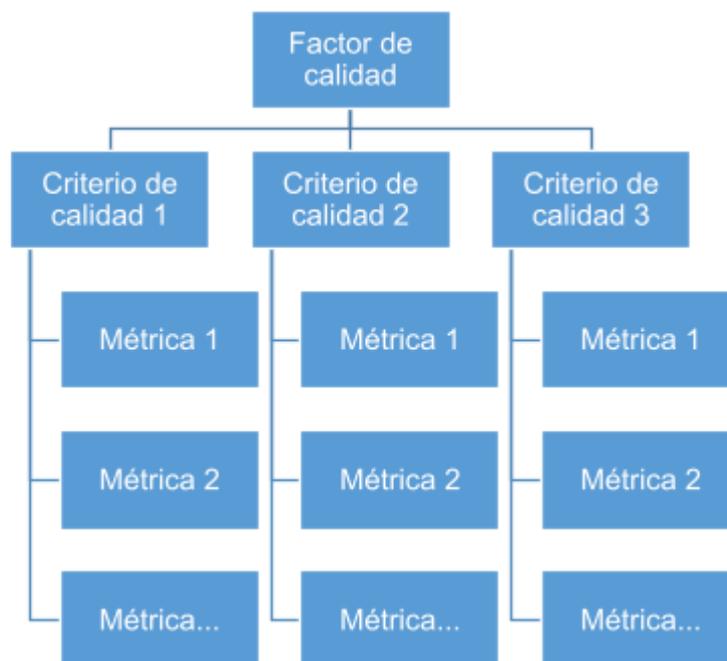
2.2.5. Calidad del Software

El desarrollo de software implica varios procesos y técnicas para su correcto funcionamiento. Por tanto, la calidad de software es esencial para garantizar que se cumplan ciertos requerimientos y llegue a satisfacer las necesidades del cliente o usuario. De acuerdo con [1], la calidad de software implica desde el análisis del código mediante el cumplimiento de ciertas métricas, reglas o normas a seguir.

De acuerdo con [23], “en el desarrollo de software, el modelo de calidad debe permitir evaluar el sistema, bien sea cualitativa o cuantitativamente, y de acuerdo con esta evaluación la organización podrá proponer e implementar estrategias que permitan la mejora del proceso dentro de las etapas de análisis, diseño, desarrollo y pruebas del software.”

Los productos de software, tienen que cumplir ciertos requisitos de calidad. La calidad de software puede llegar a tener varios puntos de vista de acuerdo a las perspectivas de los implicados en el desarrollo y uso de software. Para un usuario puede ser tener los menos defectos al utilizar un sistema. Para un desarrollador implica que la arquitectura sea la adecuada, código eficiente, entre otros.

Figura 8: Estructura de la calidad de software



Fuente: [23]

2.2.6. ISO/IEC 25000

De acuerdo con [24], “la familia ISO/IEC 25000, conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es un grupo de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.” Esta norma consta de cinco partes: gestión, modelo, medida, requisitos y evaluación de calidad.

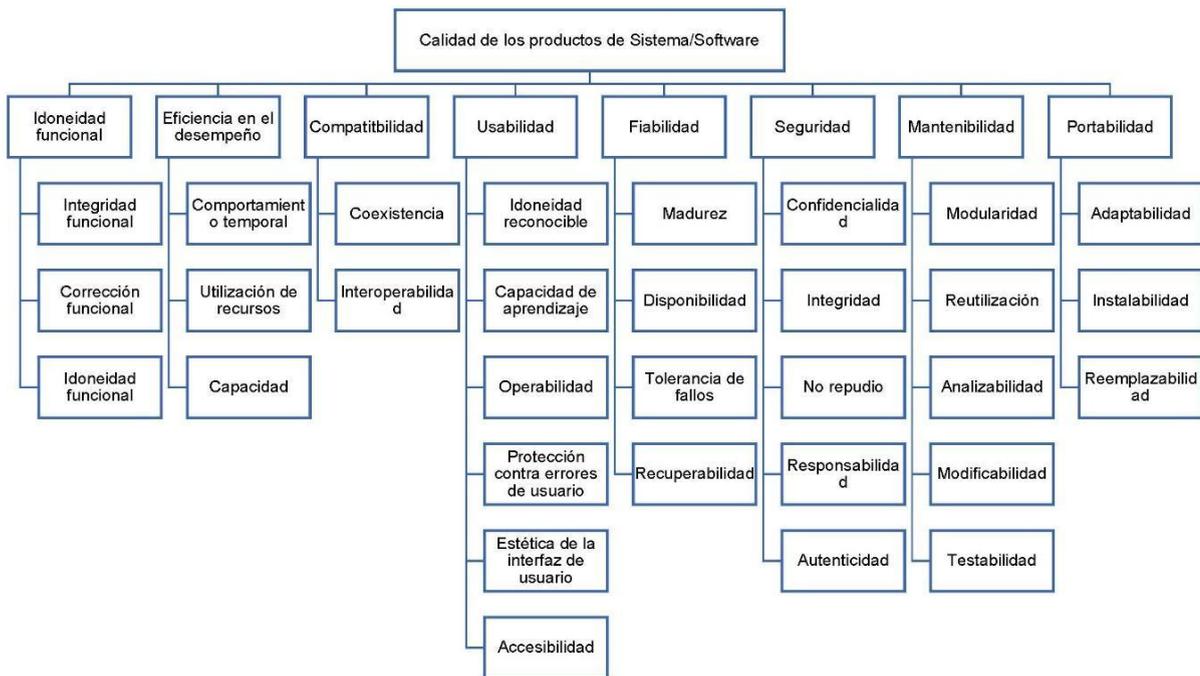
- ISO/IEC 2500n: División para la Gestión de la Calidad. Esta norma define modelos, términos principales de la ISO 25000. Además de apoyar orientación para la evaluación de calidad.[25]
- ISO/IEC 2501n: División para el Modelo de Calidad. Proporciona modelos más detallados sobre calidad de uso de software refiriéndose a la ISO 25010, calidad de datos y servicios. [25]
- ISO/IEC 2502n: División para la Medición de la Calidad. Comprende una división de modelos para la medición de calidad de software. [25]
- ISO/IEC 2503n: División para los Requisitos de Calidad. En esta división se especifican los requisitos de calidad que van servir de entrada para la evaluación de calidad. [25]
- ISO/IEC 2504n: División para la Evaluación de Calidad. “Proporcionan un proceso y requisitos para la evaluación de la calidad, junto con recomendaciones y directrices para desarrolladores, adquirentes y evaluadores independientes.” [25]

2.2.6.1. ISO/IEC 25010

La norma ISO/IEC 25010 es una norma internacional que se enfoca en la evaluación de calidad de software y sistemas. También conocida como Square (Systems and software Quality Requirements and Evaluation - Requisitos y evaluación de la calidad de los sistemas y el software).[26]

La ISO/IEC/25010 “es un conjunto de estándares de calidad para evaluar productos de software por medio de la evaluación de varias características: adecuación funcional, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad, portabilidad y eficiencia en el desempeño.”[6]

Figura 9: Modelo de calidad del Software



Fuente:[17]

2.2.7. Lenguajes de Programación

Los lenguajes de programación son lenguajes de computadoras, los cuales permiten a los desarrolladores de software comunicarse con las computadoras. Estos lenguajes permiten desarrollar sistemas, aplicaciones móviles y sitios web. Como lo menciona [27], “el objetivo esencial de la programación es la utilización de las computadoras con el fin de resolver problemas, a través de los recursos que brindan los diferentes lenguajes de programación.”

2.2.7.1. JavaScript

Es un lenguaje de secuencias de comandos, lo que significa que no es necesario compilarlo, pero mediante el uso de un intérprete (navegador) se ejecuta el código en tiempo real según sea necesario, lo que conlleva a la eliminación de software tradicional en dispositivos limitados a sistemas operativos específicos, de modo que se puede acceder a los navegadores web desde cualquier sistema operativo moderno [28].

JavaScript se vuelve cada vez más popular cada año. Según el informe "The State of the Octoverse" publicado anualmente por GitHub, JavaScript ha sido el lenguaje líder utilizado en proyectos de código abierto todos los años desde 2014. Hoy en

día, JavaScript cubre más plataformas de desarrollo diferentes que hace 10 años. Comenzando como un lenguaje de scripting de navegador del lado del cliente puro, con el lanzamiento de la plataforma Node.js en 2009, se convirtió en el estándar para el desarrollo del lado del servidor [29].

Esta combinación permite a los desarrolladores de JavaScript utilizar cualquiera de sus marcos web preferidos del lado del cliente (Angular, React, Vue, etc.) para desarrollar la interfaz de usuario de las aplicaciones de escritorio, así como utilizar las conocidas API y bibliotecas de la plataforma Node.js. Esto facilita el rápido desarrollo de aplicaciones de escritorio, que también pueden utilizar las mismas llamadas a la API del servidor que las aplicaciones web y móviles.

Javascript se ha convertido en un lenguaje universal de la web, por el cual se hace responsable de impulsar un gran número de aplicaciones interactivas para el usuario. La programación en Javascript tiene como prioridad la funcionalidad por eventos, dado que generalmente se ejecuta en un solo hilo, los cálculos de bloqueo o de ejecución prolongada a menudo hacen que la página o todo el navegador parezca que no responde. Por lo tanto, los programas JavaScript se escriben en un estilo controlado por eventos y los programas registran funciones de devolución de llamada en el bucle de eventos [30].

El código JavaScript se puede incrustar en una página web y manipular cambiando el Modelo de objetos de documento (DOM), que es la representación de árbol de la página web. La capacidad de cambiar el DOM (quizás en respuesta a la entrada del usuario) es muy útil para desarrollar aplicaciones web interactivas enriquecedoras [31].

2.2.7.2. Node.js

Node.js es una plataforma JavaScript de código abierto adecuada para aplicaciones web de eventos altamente competitivas que deben ser escalables. Utiliza la máquina virtual V8, la biblioteca libuv para su bucle de eventos e implementa la especificación CommonJS bajo la licencia MIT. Entre los módulos nativos de Node.js, existe HTTP que permite el desarrollo de servidores HTTP. Por lo tanto, al implementar sitios web y aplicaciones web desarrolladas con Node.js, es posible que no se utilicen servidores web como Nginx o Apache [32].

Específicamente, Node.js es un entorno de bajo nivel para ejecutar JavaScript en el lado del servidor. Node.js se utiliza especialmente como plataforma de servidor web; LinkedIn, Microsoft, Yahoo y PayPal lo están utilizando.

Cuando Ryan Dahl presentó Node.js en 2009, hubo un fuerte aumento en el interés en el uso de JavaScript en el lado del servidor en los campos empresarial y científico. Node.js se ejecuta en la implementación de JavaScript V8 de Google e introduce el concepto de un modelo de eventos de E/S sin bloqueo [33].

Funciona en un solo subproceso en lugar de utilizar varios subprocesos, como se conoce en el servidor en las redes tradicionales, este bucle de eventos de un solo subproceso puede consumir mejor los recursos de hardware y pueden aparecer más conexiones de cliente al mismo tiempo, lo que convierte a Node.js en una alternativa importante para el desarrollo de aplicaciones web del lado del servidor.

El uso de JavaScript para interfaces de programación de aplicaciones de alto nivel es muy común en la actualidad. Por lo tanto, su uso para el desarrollo de back-end simplifica enormemente todo el proceso de desarrollo de aplicaciones web. Además, Node.js viene con su propio administrador de paquetes "npm", que puede distribuir e instalar fácilmente bibliotecas de terceros y sus dependencias para ampliar los proyectos de desarrollo existentes [34].

Para establecer una conexión bidireccional entre el servidor Node.js y todos los clientes conectados, se requiere la biblioteca de JavaScript Socket.IO y se utiliza para la comunicación basada en eventos en tiempo real: un componente se ejecuta en el servidor Node.js y el otro se basa en la navegación web en la biblioteca cliente de JavaScript del lado del cliente que se ejecuta en el navegador [35].

2.2.7.3. ReactJS

ReactJS es un marco para crear interfaces de usuario desarrollado por Facebook. El marco admite el desarrollo de interfaces de usuario interactivas, que se dividen en componentes encapsulados que administran su propio estado, lo que da como resultado un rendimiento confiable, un código central simple y aplicaciones fáciles de ampliar. El enfoque modular de ReactJS también permite una fácil integración de bases de código utilizando Node Package Manager (npm) [36].

La biblioteca de Javascript de ReactJS se utiliza para mantener un estilo de flujo de datos receptivo, es decir, para abstraer el modelo de documento del código HTML y así habilitar un entorno de programación basado en componentes [37].

2.2.7.4. Laravel

Laravel fue creado en abril del año 2011 por Taylor Otwell, sigue un modelo vista controlador (MVC), el cual posee ventajas sobre la flexibilidad y simplicidad de diseño [38].

Para [39], es un marco de desarrollo PHP más utilizado, orientado a aplicaciones web y desarrollado bajo código abierto. Con el desarrollo de la red, todo el mundo quiere que su red sea más grande. Para ello, necesita conectarse con los clientes a través de su sitio web, que puede proporcionar información sobre su empresa y productos.

El marco PHP proporciona un código ya codificado; solo necesitamos enviar esta información al módulo de acuerdo con nuestros requisitos. Usar el marco de Laravel para el diseño web es muy fácil y simple, pero debe comprender los conceptos básicos de PHP para comprender la estructura del marco [40].

A través del framework de desarrollo PHP, se puede lograr una mayor productividad en el desarrollo de aplicaciones web, pues estas herramientas brindan una estructura definida que nos permite crear aplicaciones de manera más fácil y rápida, en un menor tiempo brindar la solución al problema [41].

2.2.8. Bases de datos SQL

Las bases de datos relacionales son atómicas, es decir, se pueden cambiar varias tablas al mismo tiempo, o no se puede cambiar ninguna de ellas. Esto no se aplica a las bases de datos no relacionales, porque si quieres cambiar la información de varios objetos, completará varias veces al objeto necesario [42].

Las bases de datos relacionales no están diseñadas para almacenar datos sin una estructura y tienen limitaciones para escalar en varios servidores, mientras que las bases de datos NoSQL pueden almacenar y procesar grandes cantidades de datos sin esquema sin utilizar un modelo relacional. [43].

2.2.9. Sistema de gestor de base de datos SQL

Desde que se empezó a utilizar el modelo de base de datos relacional en 1970, ha sufrido una serie de transformaciones y se ha convertido en el modelo de gestión de base de datos más utilizado en la actualidad. El modelo se basa básicamente en establecer relaciones o vínculos entre datos, concibiendo una tabla separada para cada relación existente, con sus propios registros y atributos [44]. En cambio, el modelo de datos de los DBMS NoSQL carece de un patrón fijo, por lo que los usuarios pueden cambiar fácilmente y de forma dinámica el modelo de datos de la aplicación. [45].

2.2.9.1. PostgreSQL

PostgreSQL es un sistema gestor de bases de datos relacionales que está orientado a objetos, que se enfoca a las funcionalidades de escalabilidad y cumplimiento de estándares. Trabaja bajo la licencia PostgreSQL de código abierto, semejante a la licencia MIT. PostgreSQL es creado por PostgreSQL Global Development Group, que está compuesto por voluntarios, personal por contrato y supervisores de empresas como Red Hat y EnterpriseDB [46].

Cuenta con más de quince años en el ámbito del desarrollo, ofreciendo avances dinámicos y diseños que han valido para demostrar confiabilidad, veracidad y corrección de información [47].

Una de las opciones que ofrece PostgreSQL para mejorar el rendimiento en estas situaciones es la partición de tablas, que puede proporcionar un mejor rendimiento al consultar estas tablas. La partición de tablas es una técnica que implica dividir una tabla enorme (principal) en un conjunto de tablas secundarias. Cuando la consulta se está ejecutando, esta técnica reduce el número de lecturas físicas en la base de datos [48].

2.2.10. Seguridad de base de datos

Actualmente las aplicaciones almacenan grandes cantidades de información que deben estar debidamente protegidas ya sea por firewalls, antivirus, IDS/IPS, entre otras herramientas y tecnologías que ayuden al manejo de datos sensibles. Muchas empresas optan por aplicar protecciones en sus servidores o desde la raíz aplicando controles a los usuarios finales, pero la verdad es que se requiere la implementación de sistemas seguros desde la programación. [49]

2.2.10.1. Heroku

Es una plataforma como servicio (PaaS) de computación en la nube, Heroku admite una gran variedad de lenguajes de programación, su desarrollo tiene base en el año 2007. Su conexión entre cliente y base de datos hace uso de encriptación SSL. Estas aplicaciones cliente cifran sus datos para el almacenamiento de información para cumplir con los requisitos de seguridad. Uno de los algoritmos de cifrado más populares y seguros es el Estándar de cifrado avanzado (AES). [50]

2.2.11. Diseño de interfaces

2.2.11.1. Adobe XD

El software Adobe XD permite el desarrollo de prototipos interactivos de interfaz de usuario de fidelidad media, que es muy adecuado para recopilar comentarios de los usuarios y diseño iterativo antes de desarrollar una interfaz con todas las funciones [51].

Adobe XD es un software que está enfocado para el diseño de experiencias de usuario. Estas interfaces pueden ser aplicaciones web, móviles, sistemas, y otros. Se puede crear prototipos para tus aplicaciones hasta ponerle interacciones para visualizar cómo se vería la interacción con el usuario.

Adobe XD “es una potente herramienta para trabajar de forma rápida y eficaz durante la fase de diseño y fase de diseño y prototipo del proceso de desarrollo de la web o la aplicación”[52]

2.2.12. IDE de Programación

Los entornos de desarrollo integrado son editores de código fuente basados en web, que combinan los sistemas de gestión para automatizar trabajos de construcción y la jerarquía de datos [53].

Un entorno de desarrollo integrado (IDE) es una gran herramienta para aumentar la productividad. Los desarrolladores de escritorio los han estado utilizando durante años. Quizás el ejemplo más común es el entorno Visual Studio de Microsoft. En el pasado, muchos proveedores de herramientas integradas han creado sus propios IDE patentados [54].

2.2.12.1. Visual Studio Code

Visual Studio Code IDE (VSCoDe) desarrollado y mantenido por Microsoft para Windows, Linux y macOS, es un editor de código fuente multiplataforma gratuito con un amplio ecosistema de complementos, se puede utilizar para depurar el módulo OPC Publisher y obtener información de registro [55].

Visual Studio Code brinda soporte, por ejemplo, para el resaltado de sintaxis específico de la plataforma o el empaquetado de código automatizado. Además, una plataforma puede proporcionar Bibliotecas de cliente para un conjunto de lenguajes de programación que envuelven las API de la plataforma como parte del Kit de desarrollo de software (SDK) de la plataforma para facilitar el proceso de desarrollo, por ejemplo, el uso de bibliotecas específicas de la plataforma para trabajar con datos de eventos escritos [56].

2.3. Objetivos del Prototipo

2.3.1. Objetivo General

Desarrollar una plataforma web que permita la evaluación de calidad de software mediante el uso de framework Laravel y React Js para la aplicación de métricas del modelo de calidad ISO/IEC 25010.

2.3.2. Objetivos Específicos

- Revisar las fuentes bibliográficas acerca del modelo de calidad ISO 25010 en las diferentes bases de datos científicas.
- Diseñar los prototipos de las interfaces que tendrá la plataforma web mediante la herramienta Adobe XD.
- Implementar la plataforma web para la evaluación de calidad de software de sitios web.
- Evaluar la calidad de software mediante la aplicación del modelo de calidad ISO 25010

2.4. Diseño del Prototipo

2.4.1. Tecnologías de desarrollo

Para la realización de la presente propuesta tecnológica se utilizaron diferentes tecnologías que se detallan a continuación.

Tabla 1: Tecnologías implementadas

TECNOLOGÍAS IMPLEMENTADAS	
Nombre	Descripción
Laravel	Framework de PHP para la implementación de Backend
React Js	Biblioteca de JavaScript para el desarrollo del Frontend
Material Ui	Framework de React para el desarrollo de interfaces
PostgreSql	Gestor de base de datos SQL
Visual Studio Code	Editor de código Fuente
Adobe XD	Herramienta para el diseño de prototipos r

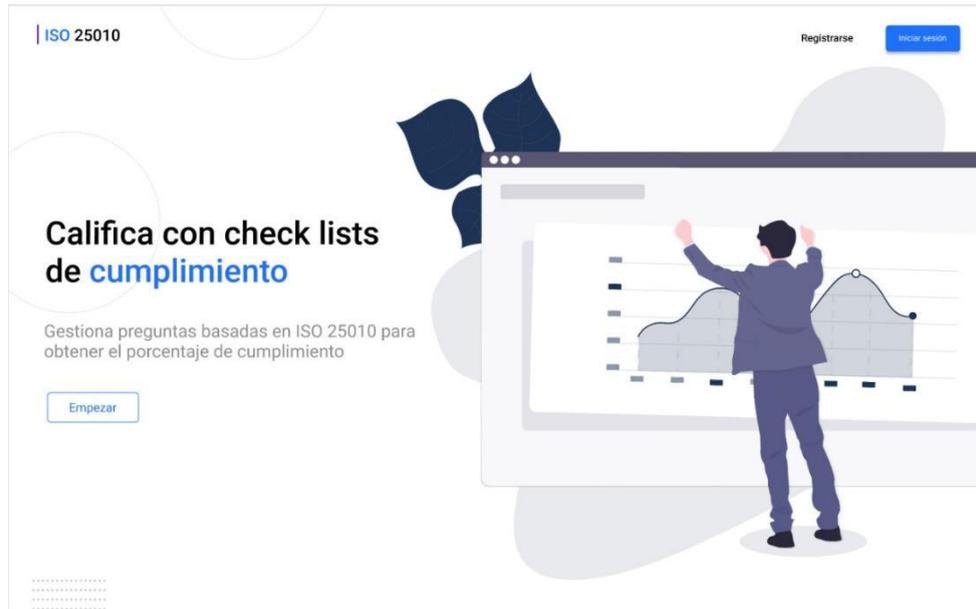
Fuente: Elaboración propia

2.4.2. Prototipado de las interfaces

Los prototipos de la plataforma web se realizaron en el programa Adobe XD, el cual se enfoca en el diseño de experiencias e interfaces de usuario. A continuación, se visualiza el diseño de las interfaces:

2.4.2.1. Pantalla principal

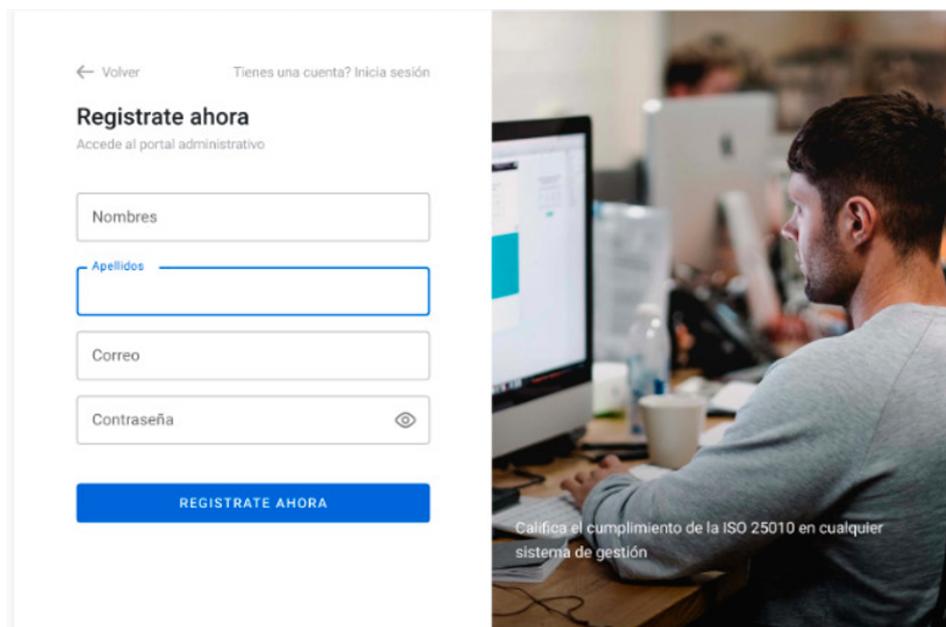
Figura 10: Prototipo - Pantalla principal



Fuente: Elaboración propia

2.4.2.2. Pantalla registro

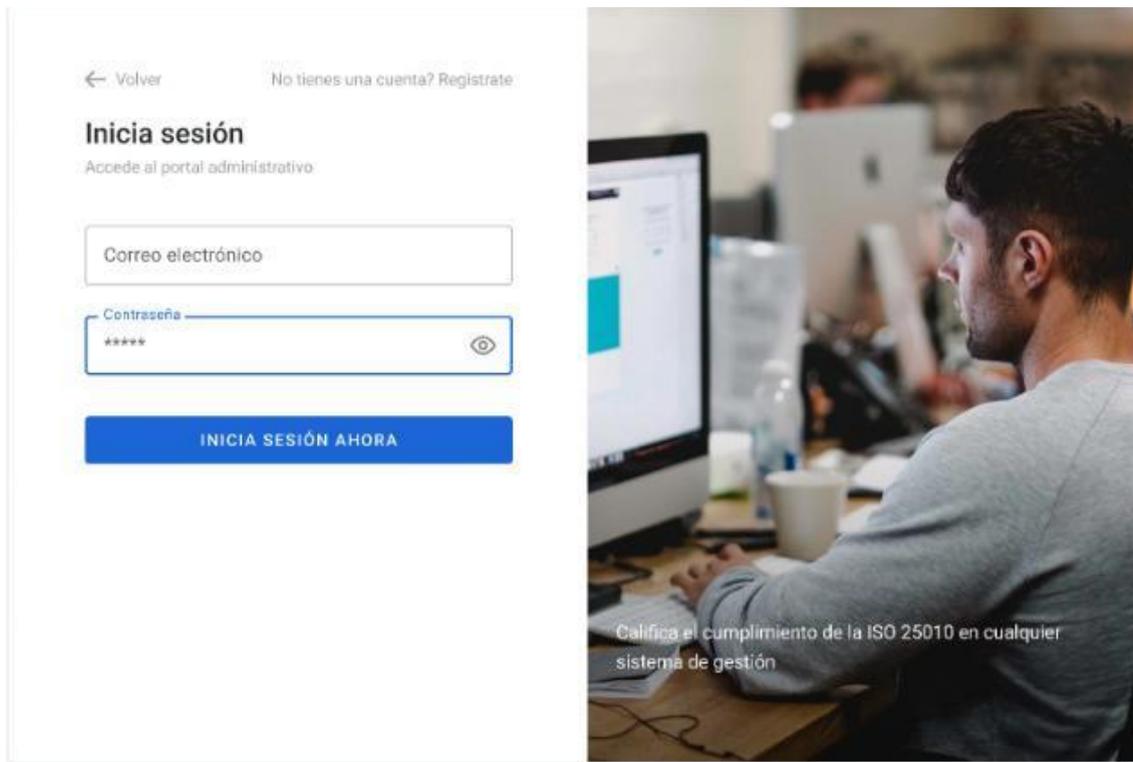
Figura 11: Prototipo - Pantalla registro



Fuente: Elaboración propia

2.4.2.3. Pantalla Login

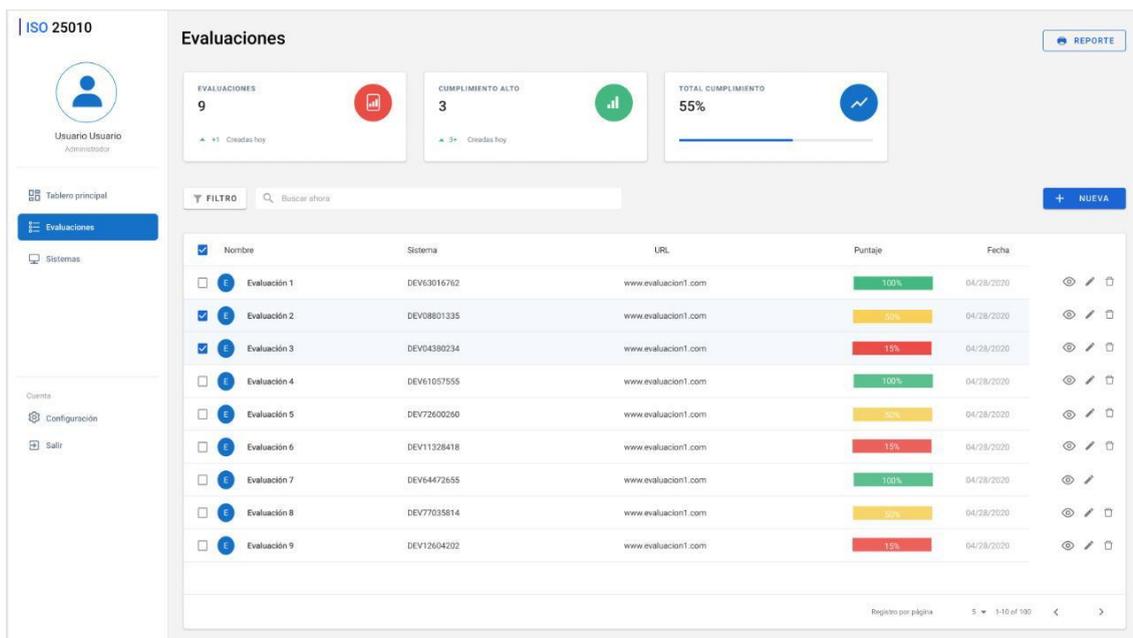
Figura 12: Prototipo - Pantalla Login



Fuente: Elaboración propia

2.4.2.4. Pantalla Evaluaciones

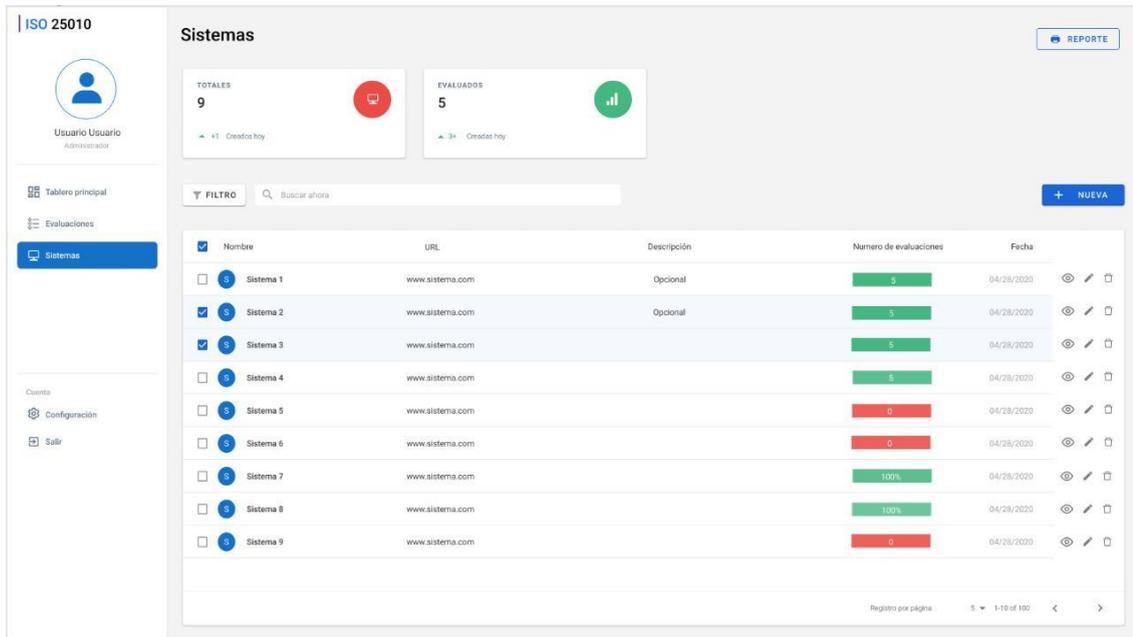
Figura 13: Prototipo - Pantalla Evaluaciones



Fuente: Elaboración propia

2.4.2.5. Pantalla Sistemas

Figura 14: Prototipo - Pantalla Sistemas



Fuente: Elaboración propia

2.4.2.6. Pantalla Nuevo Sistema

Figura 15: Prototipo - Pantalla crear sistema

The form is titled 'Crear nuevo sistema' and contains three input fields: 'Nombre', 'URL', and 'Descripción'. At the bottom, there are two buttons: a grey '← ATRÁS' button and a blue '+ NUEVA' button.

Fuente: Elaboración propia

2.5. Ejecución y/o ensamblaje del prototipo

En esta sección se detallan los procesos que se llevarán a cabo en la realización de la plataforma web. Para ello, se hace uso del modelo de calidad de software de la norma ISO 25010, como se puede observar en la Tabla 2, se describen las métricas, características y preguntas que formarán parte de la evaluación de los sitios web.

Tabla 2: Métrica/preguntas utilizadas para la evaluación de calidad

Característica	(Sub) Característica	Preguntas
Idoneidad funcional	Integridad funcional	¿Cumple con todas las tareas y objetivos especificados por el usuario?
	Corrección funcional	¿Proporciona resultados correctos con el grado de precisión necesario?
	Idoneidad funcional	¿Tiene el conjunto de funciones apropiadas para las tareas especificadas?
Eficiencia en el desempeño	Comportamiento temporal	¿Los tiempos de respuesta y procesamiento en ejecución de las funciones son adecuados?
	Utilización de recursos	¿Los recursos de información son utilizados de manera eficiente?
	Capacidad	Los parámetros de los requisitos del sistema se cumplen correctamente
Compatibilidad	Coexistencia	¿Puede realizar sus funciones correctamente mientras comparte recursos con otros sistemas?
	Interoperabilidad	¿Puede intercambiar y utilizar información con otros sistemas?
Usabilidad	Idoneidad reconocible	¿El sistema es adecuado para resolver las necesidades del usuario?
	Capacidad de aprendizaje	¿Puede ser utilizado con efectividad y eficiencia?
	Operabilidad	¿El sistema tiene funciones de manejo y control para el usuario?
	Protección contra errores de usuario	¿El sistema está validado contra errores de usuario?
	Estética de la interfaz de usuario	¿La interfaz de usuario es atractiva y agradable?

	Accesibilidad	¿Puede ser utilizado por los usuarios con la más amplia gama de características y capacidades?
Fiabilidad	Madurez	¿Con qué frecuencia presenta fallas por defectos o errores?
	Disponibilidad	¿La información del sistema se encuentra siempre disponible para todos los usuarios?
	Tolerancia a fallos	¿Si suceden fallas, como se comporta en cuanto a la performance especificada, es capaz de manejar errores?
	Recuperabilidad	¿Cuánto tiempo tarda el sistema en recuperarse cuando presenta fallas por defectos o errores?
Seguridad	Confidencialidad	¿Tiene mecanismos de seguridad para el acceso no autorizado?
	Integridad	¿Impide la modificación de sus recursos de información?
	No repudio	¿Se tiene un registro sobre todas las acciones que realiza un usuario sobre el sistema?
	Responsabilidad	¿El sistema rastrea las transacciones que realiza una entidad sobre el sistema?
	Autenticidad	¿El sistema tiene controles de verificación para los permisos y roles de usuario?
Mantenibilidad	Modularidad	¿Se puede realizar un cambio sin tener gran impacto sobre los demás módulos?
	Reutilización	¿El código construido puede ser utilizado en otro módulo o sistema diferente?
	Analizabilidad	¿El sistema tiene logs de errores para identificar las fallas que ocurren dentro del sistema?
	Modificabilidad	¿Es fácil de modificar y adaptar un nuevo módulo, sin afectar al sistema en producción?
	Testabilidad	¿Proporciona la facilidad de realizar pruebas sobre las acciones del sistema?

Portabilidad	Adaptabilidad	¿El sistema puede ser ejecutado en cualquier sistema operativo y visualizado en cualquier dispositivo?
	Instalabilidad	¿Puede ser instalado o configurado fácilmente en cualquier entorno?
	Reemplazabilidad	¿Puede ser reemplazado por otro sistema sin generar conflictos de compatibilidad?

Fuente: Elaboración propia

Para determinar la evaluación de cada pregunta se utiliza la escala de Likert, como se puede visualizar en la tabla 2. Esta evaluación se realiza a criterio del usuario, determinando para cada sitio web la puntuación que cree conveniente, para cada pregunta de evaluación. A continuación, se detalla la escala de Likert y sus respectivas opciones y valores.

Tabla 3: Escala de Likert para evaluación de calidad

Escala de Likert para evaluación de calidad	
Totalmente en desacuerdo	0%
En desacuerdo	25%
Ni de acuerdo, ni en desacuerdo	50%
De acuerdo	75%
Totalmente de acuerdo	100%

Fuente: Elaboración propia

Para la evaluación se utiliza la siguiente fórmula. La cual permite obtener un valor por sub-característica, se divide el 100% para la cantidad de preguntas.

$$VPS = \frac{100\%}{NP}$$

En donde:

VPS = Valor por sub-característica,

NP = Número de preguntas.

2.5.1. Pantalla principal

Esta pantalla será la principal del sistema, la cual se enfoca de manera visual en describir en qué consiste la plataforma, en este caso de la evaluación de calidad de software mediante la ISO 25010. Para ingresar se debe dar clic en el botón Registrarse.

Figura 16: Pantalla principal del sistema

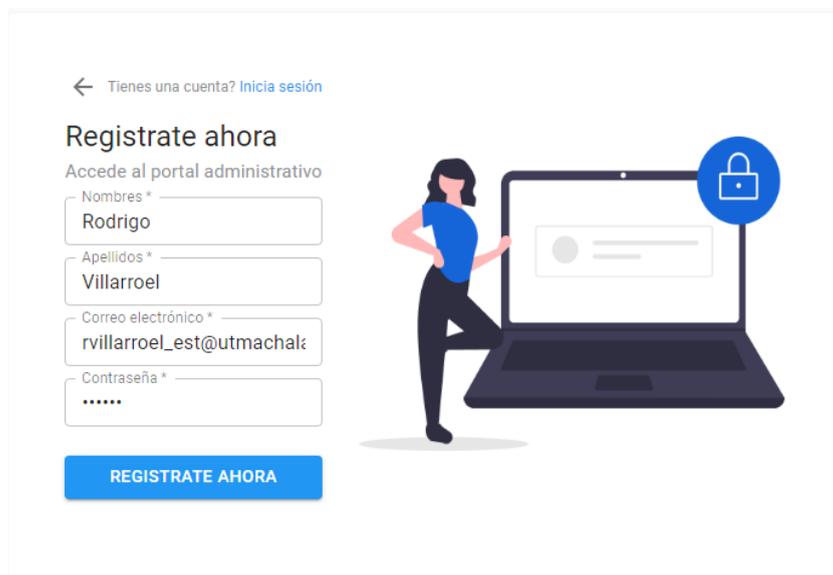


Fuente: Elaboración propia

2.5.2. Pantalla registro

En esta pantalla podremos registrarnos, es decir, crear una cuenta de usuario. Para lo cual, se debe llenar los campos nombres, apellidos, correo y contraseña. Por último, dar clic en el botón Registrar Ahora.

Figura 17: Pantalla Registro

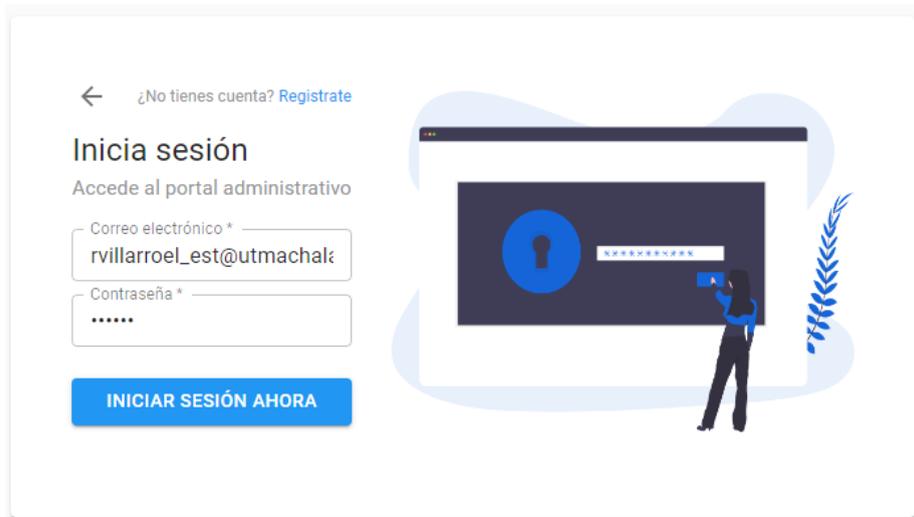


Fuente: Elaboración propia

2.5.3. Pantalla Login del sistema

La pantalla Login es el ingreso al sistema mediante una cuenta de usuario, la cual se crea, como se menciona anteriormente. En esta pantalla se ingresa el correo y contraseña del usuario, luego se da clic en el botón iniciar sesión ahora.

Figura 18: Pantalla login

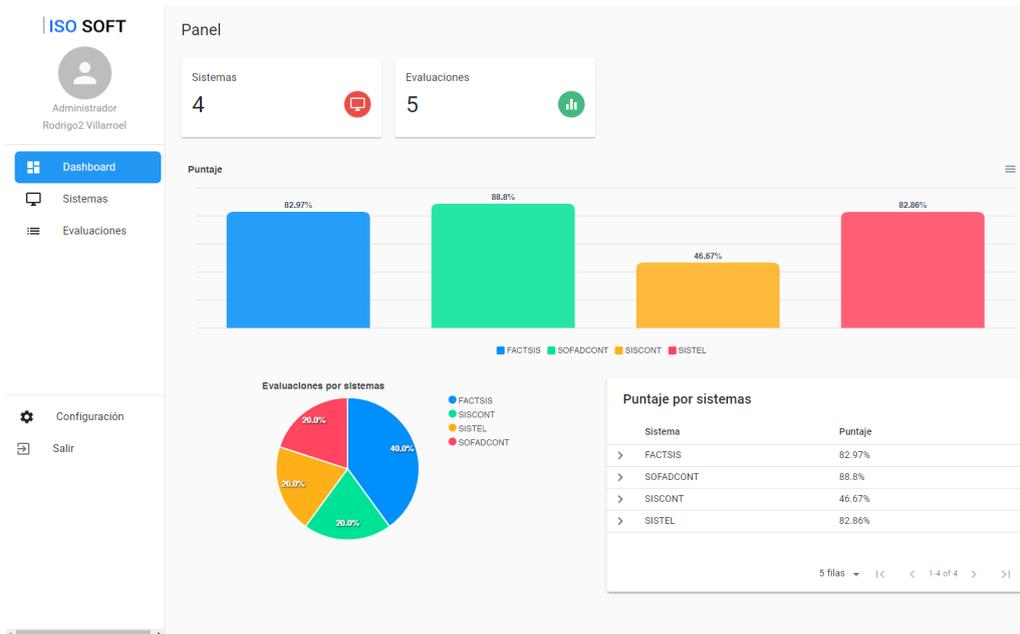


Fuente: Elaboración propia

2.5.4. Pantalla dashboard

En esta pantalla se puede visualizar los sistemas evaluados con sus respectivos porcentajes de cumplimiento. Es decir, un resumen de las estadísticas de los sistemas evaluados.

Figura 19: Pantalla Dashboard



Fuente: Elaboración propia

2.5.5. Pantalla sistemas

Esta pantalla visualiza todos los sistemas evaluados, donde se observan los campos de nombre, url, descripción, evaluaciones, fecha. Además, se puede editar o eliminar la evaluación.

Figura 20: Pantalla Sistemas

Nombre	Url	Descripción	Evaluaciones	Fecha
FACTSIS	www.factsis.com.ec	Facturación electrónica	2	4/9/2021 2:17:21
SOFADCONT	www.sofadcont.com.ec	Contabilidad de costos	1	4/9/2021 2:18:04
SISCONT	www.siscont.com.ec	Sistema de contabilidad	1	4/9/2021 2:18:30
SISTEL	www.sistel.net	Sistema de Telecomunicaciones	1	4/9/2021 2:19:24

Fuente: Elaboración propia

2.5.6. Pantalla crear sistema

La ventana crear sistemas, permite la creación de un nuevo sistema, para su posterior evaluación. En donde, se ingresan la información de: nombre, url, descripción.

Figura 21: Pantalla crear sistema

Sistemas

Formulario de creación de sistema

CANCELAR GUARDAR

Fuente: Elaboración propia

2.5.7. Pantalla Evaluaciones

La pantalla de evaluaciones visualiza todas las evaluaciones realizadas a los diferentes sitios web. En la cual, se observa la lista de evaluaciones, además de las columnas Nombre de la evaluación, nombre del sistema, URL del sitio evaluado, puntaje de la evaluación y fecha en la que se realizó la evaluación.

Figura 22: Pantalla Evaluaciones

Nombre	Sistema	Descripción	Puntaje	Fecha	
Evaluacion2	FACTSIS	Calidad	72.57%	4/9/2021 2:39:56	👁️ ✎️ 🗑️
Evaluacion sistel	SISTEL	Evaluación de calidad de software	82.88%	4/9/2021 2:35:07	👁️ ✎️ 🗑️
Evaluación siscont	SISCONT	Calidad de software	68.67%	4/9/2021 2:32:09	👁️ ✎️ 🗑️
Evaluacion	SOFADCONT	Calidad de Software	88.81%	4/9/2021 2:26:41	👁️ ✎️ 🗑️
Evaluación Factsis	FACTSIS	Evaluación calidad de software	90.38%	4/9/2021 2:26:09	👁️ ✎️ 🗑️

Fuente: Elaboración propia

2.5.8. Pantalla nueva evaluación

En esta sección nos permite crear una nueva evaluación para los sistemas previamente creados. Se ingresan los datos de: sistema, nombre y descripción.

Figura 23: Pantalla nueva evaluación

Fuente: Elaboración propia

Luego de ingresar los datos de la evaluación, se procede con la evaluación en base a la norma ISO 25 010, características y subcaracterísticas del modelo de calidad de software.

Figura 24: Preguntas evaluación, norma ISO 27010

Característica	Totalmente en desacuerdo	En Desacuerdo	Ni de acuerdo, ni en desacuerdo	De acuerdo	Totalmente de acuerdo
Integridad funcional ¿Cumple con todas las tareas y objetivos especificados por el usuario?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Corrección funcional ¿Proporciona resultados correctos con el grado de precisión necesario?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Idoneidad funcional ¿Tiene el conjunto de funciones apropiadas para las tareas especificadas?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Fuente: Elaboración propia

2.5.9. Visualización de los Resultados

Luego de contestar las preguntas de la evaluación se visualiza el resultado de la evaluación, existen dos modos de visualización, por puntuación general, Figura 26, o puntuación por característica, Figura 27

Figura 25. Puntuación general de la evaluación



Fuente: Elaboración propia

Figura 26: Puntuación por característica



Fuente: Elaboración propia

2.5.10. Pantalla configuración de cuenta

En esta pantalla se puede realizar la configuración de una cuenta de un usuario. Donde se puede editar la información de usuario.

Figura 27: Pantalla configuración de cuenta

The screenshot shows a user profile on the left with the name 'Administrador Rodrigo Villarroel'. The main content area is titled 'Configuración' and contains the following fields:

- Nombres *: Rodrigo
- Apellidos *: Villarroel
- Correo *: rvillarroel@gmail.com
- Contraseña *

A blue button labeled 'GUARDAR' is located at the bottom of the form.

Fuente: Elaboración propia

2.5.11. Configuración de Heroku

Las copias de seguridad de los datos son muy importantes para cualquier sistema informático, no solo para la recuperación de información en caso de fallos, sino también para ejecutar pruebas, migrar datos, etc.

Figura 28: Configuración de Heroku

The screenshot shows the Heroku configuration page for an application named 'shielded-anchorage-21966'. The page is divided into several sections:

- Free Dynos:** A section with a 'Change Dyno Type' button. The current configuration is 'web' with the command 'vendor/bin/heroku-php-apache2 public/' and a cost of '\$0.00'.
- Add-ons:** A section with a 'Find more add-ons' button. The current configuration is 'Heroku Postgres' attached as a database, with a 'Hobby Dev' plan and a 'Free' tier.
- Estimated Monthly Cost:** A section showing the total estimated monthly cost as '\$0.00'.

2.5.12. Generación de Backups

Heroku cuenta con dos métodos para generar copias de seguridad, a continuación, se muestra la generación de backups de forma manual.

```
Backup manual:  
heroku pg:backups:capture --app shielded-anchorage-21966
```

Además nos permite la restauración, eliminación y descarga de las copias de seguridad.

```
Restore:  
heroku pg:backups:restore b002 --app shielded-anchorage-21966  
  
Eliminación backup:  
heroku pg:backups:delete b001 --app shielded-anchorage-21966  
  
Descarga backup:  
heroku pg:backups:download
```

Además de las copias de seguridad que se realizan manualmente, también se pueden programar copias de seguridad automáticas periódicas, estas se ejecutan diariamente en la base de datos especificada.

```
Programar backup automático:  
heroku pg:backups:schedule --at '08:00 America/Guayaquil' --app  
shielded-anchorage-21966  
  
Quitar backup automático:  
heroku pg:backups:unschedule --app shielded-anchorage-21966
```

Figura 29: Generación de Backups

The screenshot shows the Heroku Datacenter interface for a PostgreSQL database. The main heading is "Continuous Protection & Postgres Rollbacks". A message states: "Continuous Protection and Postgres Rollbacks are **Not Available** on Hobby databases. Upgrade to a Standard or Premium database to enable continuous and automated backups of your data. With Heroku Postgres Rollbacks, you're able to safely roll a database back to a specific point in time when things go awry." Below this is a link: "Learn More: Heroku Postgres Rollbacks".

Under the heading "MANUAL BACKUPS & DATA EXPORTS", there is a table with the following data:

Name	Size	Database	Created	Finished	
b007	48.0 KB	DATABASE	a minute ago	a minute ago	Download Delete

At the top right of the table area is a "Create Manual Backup" button. At the bottom of the table area are "Download" and "Delete" buttons for the backup entry.

Footer information includes: heroku.com, Blogs, Careers, Documentation, Support, Terms of Service, Privacy, Cookies, © 2021 Salesforce.com

Fuente: Elaboración propia

3. CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO

3.1. Plan de evaluación

Para verificar el funcionamiento correcto de la aplicación web, se debe aplicar un plan de evaluación el cual nos permita identificar falencias del software, detectar futuros riesgos y mejorar los procesos que se vean afectados para cumplir con los objetivos propuestos.

Evaluación de calidad

La evaluación de calidad se enfoca en la valoración de herramientas web que tengan relación con los parámetros de calidad de la ISO/IEC 25010, evidenciando la funcionalidad del sistema.

Parámetros de calidad ISO/IEC 25010						
Herramientas	Funcionalidad	Eficiencia en el desempeño	Accesibilidad	Usabilidad	Seguridad	Portabilidad
GTmetrix		x				
Validator	x					
Functional Accessibility Evaluator			x			
MetricSpot				x		
Grader		x			x	x
Pingdom		x				

3.2. Resultados de la evaluación

3.2.1. GTmetrix

Con la herramienta GTmetrix se puede evaluar los parámetros de calidad de Funcionalidad y eficiencia en los sistemas web, de acuerdo a la evaluación realizada a la plataforma Isosoft se obtuvo resultados favorables en cuanto al rendimiento con un 61% y con respecto a la estructura un 96% (Anexo A).

3.2.2. Validator (HTML Y CSS)

Validator evalúa la funcionalidad de los sistemas, en donde al aplicar esta herramienta al prototipo se reflejó que no presentaba errores o advertencias, registrando buenos resultados. (Anexo B).

3.2.3. Functional Accessibility Evaluator

Según como se visualiza en el (Anexo C), con la herramienta FAE se obtuvieron resultados imparciales en cuanto a la accesibilidad de la plataforma Isosoft, registrándose 6 reglas aceptables y en contra 6 también.

3.2.4. MetricSpot

Con MetricSpot se evaluó la usabilidad de Isosoft la cual registró un total equivalente al 67%, lo que significa resultados positivos para el sistema (Anexo D), esta herramienta es considerada la más apta para la evaluación de usabilidad y el análisis de las páginas web.

3.2.5. Grader

Con la herramienta Grader se puede evaluar la seguridad, rendimiento y el seo de las plataformas web, según la evaluación registrada a la página isosoft se obtuvo un promedio del 95% en general, lo que significa que la plataforma cumple con los estándares de calidad evaluados (Anexo E).

3.2.6. Pingdom

Pingdom evalúa los estándares de calidad de las páginas web en específico la usabilidad en donde la plataforma Isosoft obtuvo resultados positivos con un total de 85 puntos. (Anexo F).

3.3. Conclusiones

- Finalmente se desarrolló una plataforma web que permite la evaluación de páginas en línea en base a la normativa ISO/IEC 25010, en donde se consideran los parámetros de calidad como funcionalidad, usabilidad, seguridad, portabilidad entre otros; obteniendo resultados confiables de páginas que cumplen con los estándares de calidad.
- Para lograr resultados verificados se revisaron fuentes bibliográficas que estén en revistas indexadas en bases de datos científicas, con ello asegurar que la información revisada sobre el modelo de calidad ISO 25010 sea exacta y confiable.
- Antes del desarrollo del prototipo se plasmó en papel la idea del software para luego diseñar los prototipos de las interfaces mediante la herramienta Adobe XD la cual permite realizar diseños con interacciones casi reales a las que se tendrán con el usuario.
- Teniendo la información clara sobre las métricas de calidad de la ISO 25010 y con todas las especificaciones del prototipo maquetadas, se procedió a la implementación de la plataforma web con las herramientas de programación React JS para el FrontEnd, Laravel para el BackEnd y para el almacenamiento de información se utilizó PostgreSQL como gestor de base de datos.
- Una vez que se realizó la implementación de la plataforma web, se aplicaron las pruebas necesarias de evaluación a las herramientas web que cumplieron con los criterios que se especifican en la normativa ISO 25010.

3.4. Recomendaciones

- En el desarrollo de software se recomienda determinar el alcance del proyecto, garantizar la calidad del producto. Obtener todos los requerimientos para la implementación del mismo.
- Consultar en fuentes viables la información necesaria para el desarrollo del proyecto, como bases de datos científicas, así obtener la información más actual y confiable.

- Utilizar herramientas de prototipado o diseño de interfaces de usuarios para tener una idea visual de cómo se verá la plataforma, además de entender cómo será el funcionamiento de la misma.
- Investigar las herramientas de desarrollo que se adapten más a nuestra propuesta. Para tener una mejor comprensión de los mismos. Además, identificar qué herramienta se utilizará en base a los requerimientos del sistema a desarrollar.
- Realizar todas las pruebas necesarias a la plataforma, para evitar posibles errores al momento de su implementación.

4. BIBLIOGRAFÍA

- [1] M. García-Valls, J. Escribano-Barreno, and J. García-Muñoz, “An extensible collaborative framework for monitoring software quality in critical systems,” *Inf. Softw. Technol.*, vol. 107, pp. 3–17, Mar. 2019, doi: 10.1016/J.INFSOF.2018.10.005.
- [2] A. Idri, M. Bachiri, J. L. Fernandez-Aleman, and A. Toval, “ISO/IEC 25010 Based Evaluation of Free Mobile Personal Health Records for Pregnancy Monitoring,” *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 1, pp. 262–267, 2017, doi: 10.1109/COMPSAC.2017.159.
- [3] M. Arboleda, “Estadísticas digitales Ecuador 2021 por Datareportal,” 2021.
<https://www.hablemosdemarcas.com/estadisticas-digitales-ecuador-2021/>
.
- [4] C. Yue, “A projection-based approach to software quality evaluation from the users’ perspectives,” *Int. J. Mach. Learn. Cybern. 2018 109*, vol. 10, no. 9, pp. 2341–2353, Sep. 2018, doi: 10.1007/S13042-018-0873-Y.
- [5] E. N. Desyatirikova, V. E. Belousov, V. N. Zolotarev, and O. Y. Lavlinskaia, “Design process of software quality management,” *Proc. 2017 Int. Conf. Quality Manag. Transp. Inf. Secur. Inf. Technol. IT QM IS 2017*, pp. 496–499, Oct. 2017, doi: 10.1109/ITMQIS.2017.8085870.
- [6] F. Salazar Fierro, C. A. Pineda Manosalvas, N. N. Cervantes Rodríguez, and P. Landeta, “Análisis de la eficiencia de desempeño en aplicaciones de Realidad Aumentada utilizando la normativa ISO/IEC/25010,” *risti*, pp. 256–267, 2019, Accessed: Jul. 18, 2021. [Online]. Available: https://www.researchgate.net/publication/338925171_Analisis_de_la_eficiencia_de_desempeno_en_aplicaciones_de_Realidad_Aumentada_utilizando_la_normativa_ISOIEC25010.
- [7] V. Esterkin and C. Pons, “Evaluación de calidad en el desarrollo de software dirigido por modelos,” *Ingeniare*, vol. 25, no. 3, pp. 449–463, 2017, doi: 10.4067/S0718-33052017000300449.
- [8] L. A. Blanquicett, M. C. Bonfante, and J. Acosta-Solano, “Testing practices from the software industry. the ASISTO platform as a case study,” *Inf. Technol.*, vol. 29, no. 1, pp. 11–18, Feb. 2018, doi: 10.4067/S0718-07642018000100011.
- [9] M. Andersson, A. Kusetogullari, and J. Wernberg, “Software development and innovation: Exploring the software shift in innovation in Swedish firms,” *Technological Forecasting and Social Change*, 2021.
<https://www.sciencedirect.com/science/article/pii/S004016252100127X>
(accessed Jul. 11, 2021).
- [10] K. Rindell, J. Ruohonen, J. Holvitie, S. Hyrynsalmi, and V. Leppänen, “Security in agile software development: A practitioner survey,” *Inf. Softw. Technol.*, vol. 131, pp. 1–13, 2021, Accessed: Jul. 11, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950584920302305?via%3Dihub>.

- [11] G. Méndez, "Proceso Software y Ciclo de Vida," 2008.
- [12] J. Zumba Gamboa, "Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software," *INNOVA Res. Journal, ISSN-e 2477-9024, Vol. 3, N°. 10, 2018, págs. 20-33*, vol. 3, no. 10, pp. 20–33, 2018, Accessed: Jul. 12, 2021. [Online]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=6777227&info=resumen&idioma=SPA>.
- [13] J. M. Loor Intriago, M. D. Delgado Dapena, and P. B. Fernández Oliva, "Priorización de casos de prueba en entornos de desarrollo ágil," *Ing. Ind.*, 2020, Accessed: Jul. 13, 2021. [Online]. Available: <http://scielo.sld.cu/pdf/rrii/v41n2/1815-5936-rrii-41-02-e4114.pdf>.
- [14] N. Gómez, "Los valores del manifiesto para el desarrollo ágil de software," *PLATZI*, 2019. <https://platzi.com/tutoriales/1750-scrum/2261-los-valores-del-manifiesto-para-el-desarrollo-agil-de-software/> (accessed Jul. 13, 2021).
- [15] A. Ribeiro and L. Domingues, "Acceptance of an agile methodology in the public sector," *Procedia Comput. Sci.*, vol. 138, pp. 621–629, Jan. 2018, doi: 10.1016/J.PROCS.2018.10.083.
- [16] S. M. Gómez, "Aplicación de las Metodologías Ágiles al proceso de enseñanzaaprendizaje universitario," *Rev. d'innovació docent Univ. RIDU, ISSN-e 2013-2298, N°. 12, 2020, págs. 62-73*, no. 12, pp. 62–73, 2020, Accessed: Jul. 13, 2021. [Online]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=7256650>.
- [17] S. Chantit and I. Essebaa, "Towards an automatic model-based Scrum Methodology," *Procedia Comput. Sci.*, vol. 184, pp. 797–802, Jan. 2021, doi: 10.1016/J.PROCS.2021.03.099.
- [18] M. Riesener, C. Doelle, S. Perau, P. Lossie, and G. Schuh, "Methodology for iterative system modeling in agile product development," *Procedia CIRP*, vol. 100, pp. 439–444, Jan. 2021, doi: 10.1016/J.PROCIR.2021.05.101.
- [19] A. Pawlicka, "➤ Agile Software Development Process – Everything You Need to Know," *SELLEO*, 2021. <https://selleo.com/blog/agile-software-development-process-everything-you-need-to-know> (accessed Jul. 13, 2021).
- [20] S. Sherin, M. Z. Iqbal, M. U. Khan, and A. A. Jilani, "Comparing coverage criteria for dynamic web application: An empirical evaluation," *Comput. Stand. Interfaces*, vol. 73, no. December 2019, p. 103467, 2021, doi: 10.1016/j.csi.2020.103467.
- [21] J. Molina-Ríos and N. Pedreira-Souto, "Comparison of development methodologies in web applications," *Inf. Softw. Technol.*, vol. 119, p. 106238, 2020, doi: 10.1016/j.infsof.2019.106238.
- [22] Bespoke IT Solutions, "Types of Web Application Architecture: A Concise Summary | Axisbits," *AXISBITS*, 2021. <https://axisbits.com/blog/Types-of-Web-Application-Architecture-A-Concis>

e-Summary (accessed Jul. 16, 2021).

- [23] M. Callejas Cuervo, A. Alarcón Aldana, and A. M. Álvarez Carreño, "Modelos de calidad del software, un estado del arte," *Entramado*, vol. 13, pp. 236–250, 2017, doi: 10.18041/entramado.2017v13n1.25125.
- [24] T. Vaca and A. Jácome, "(PDF) Calidad de software del módulo de talento humano del sistema informático de la Universidad Técnica del Norte bajo la norma ISO/IEC 25000," 2018, Accessed: Jul. 18, 2021. [Online]. Available: https://www.researchgate.net/publication/325022337_Calidad_de_softwar_e_del_modulo_de_talento_humano_del_sistema_informatico_de_la_Univ_ersidad_Tecnica_del_Norte_bajo_la_norma_ISOIEC_25000.
- [25] J. Verdugo and M. Rodríguez, "Assessing data cybersecurity using ISO/IEC 25012," *Softw. Qual. J.*, vol. 28, no. 3, pp. 965–985, 2020, doi: 10.1007/s11219-019-09494-x.
- [26] E. Peters and G. K. Aggrey, "An ISO 25010 based quality model for ERP systems," *Adv. Sci. Technol. Eng. Syst.*, vol. 5, no. 2, pp. 578–583, 2020, doi: 10.25046/aj050272.
- [27] K. I. Díaz Tejera, E. Fierro Martín, and M. A. Muñoz Pentón, "La enseñanza de la programación: una experiencia en la formación de profesores de informática," *Educación*, vol. 27, no. 53, pp. 73–91, 2018, doi: 10.18800/EDUCACION.201802.005.
- [28] L. O. Seman, L. A. Koehler, E. A. Bezerra, and R. Hausmann, "MPPTjs: A JavaScript Simulator for PV Panels Used in a PBL Application," *Energy Procedia*, vol. 107, no. September 2016, pp. 109–115, 2017, doi: 10.1016/j.egypro.2016.12.141.
- [29] K. Peguero and X. Cheng, "Electrolint and security of electron applications," *High-Confidence Comput.*, vol. 1, no. 2, p. 100032, 2021, doi: 10.1016/j.hcc.2021.100032.
- [30] E. Fritz and T. Zhao, "Typing and semantics of asynchronous arrows in." pp. 141–142, 2017.
- [31] M. Bugliesi, S. Calzavara, and R. Focardi, "Formal methods for web security," *J. Log. Algebr. Methods Program.*, vol. 87, pp. 110–126, 2017, doi: 10.1016/j.jlamp.2016.08.006.
- [32] H. Ben Hassen, N. Ayari, and B. Hamdi, "A home hospitalization system based on the Internet of things, Fog computing and cloud computing," *Informatics Med. Unlocked*, vol. 20, p. 100368, 2020, doi: 10.1016/j.imu.2020.100368.
- [33] NodeJS, "Node JS," 2021. <https://nodejs.org/en/>.
- [34] F. Kaimer and P. Brune, "Return of the JS: Towards a node.js-based software architecture for combined CMS/CRM applications," *Procedia Comput. Sci.*, vol. 141, pp. 454–459, 2018, doi: 10.1016/j.procs.2018.10.143.
- [35] Q. Zhang, "Medical data visual synchronization and information

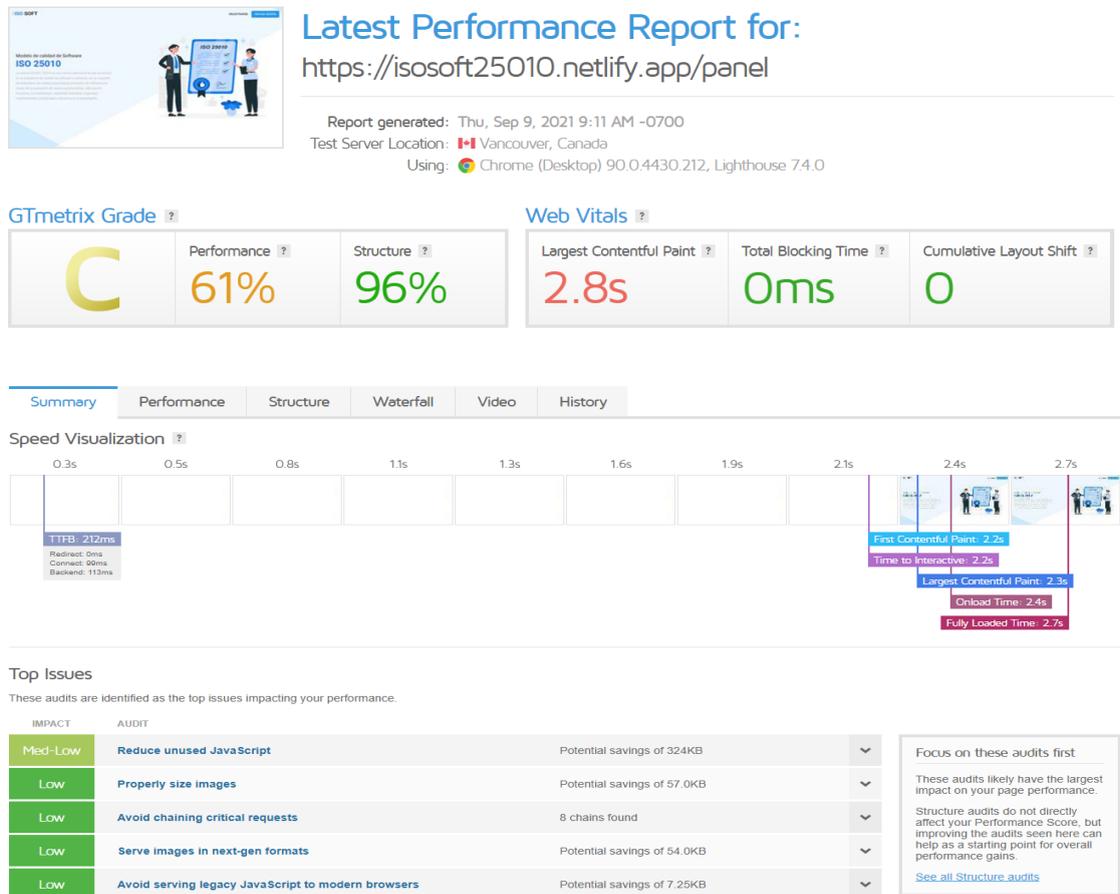
- interaction using Internet-based graphics rendering and message-oriented streaming,” *Informatics Med. Unlocked*, vol. 17, no. August, p. 100253, 2019, doi: 10.1016/j.imu.2019.100253.
- [36] T. Silva, M. Mota, C. Hernández, and J. F. De Abreu, “Automatic creation of informative TV videos to be delivered through iTV: A system architecture,” *Procedia Comput. Sci.*, vol. 121, pp. 584–591, 2017, doi: 10.1016/j.procs.2017.11.077.
- [37] A. Blanco-Míguez, G. Blanco, A. Gutierrez-Jácome, F. Fdez-Riverola, B. Sánchez, and A. Lourenço, “Computational prediction of the bioactivity potential of proteomes based on expert knowledge,” *J. Biomed. Inform.*, vol. 91, no. February, p. 103121, 2019, doi: 10.1016/j.jbi.2019.103121.
- [38] K. Nugroho, Sumardi, S. Murdowo, and Muljono, “Mobile Cloud Learning System Using Laravel Framework and Android Studio Web View,” *Proc. - 2019 Int. Semin. Appl. Technol. Inf. Commun. Ind. 4.0 Retrospect. Prospect. Challenges, iSemantic 2019*, pp. 141–144, 2019, doi: 10.1109/ISEMANTIC.2019.8884275.
- [39] A. Ramalingaiah and T. Sulthana, “Study of Blockchain with Bitcoin based Fund Raise Use case using Laravel Framework,” *Proc. 2018 3rd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. CSITSS 2018*, pp. 254–258, 2018, doi: 10.1109/CSITSS.2018.8768766.
- [40] N. Yadav, D. S. Rajpoot, and S. K. Dhakad, “LARAVEL: A PHP Framework for E-Commerce Website,” *Proc. IEEE Int. Conf. Image Inf. Process.*, vol. 2019-Novem, pp. 503–508, 2019, doi: 10.1109/ICIIP47207.2019.8985771.
- [41] R. Valarezo and T. Guarda, “Comparative analysis of the laravel and codeigniter frameworks: For the implementation of the management system of merit and opposition competitions in the State University Península de Santa Elena,” *Iber. Conf. Inf. Syst. Technol. Cist.*, vol. 2018-June, pp. 1–6, 2018, doi: 10.23919/CISTI.2018.8399242.
- [42] V. Valverde, N. Portalanza, and P. Mora, “Análisis descriptivo de base de datos relacional y no relacional,” *Cuad. Educ. y Desarro.*, no. 108, pp. 1–16, 2019.
- [43] D. Mahajan, C. Blakeney, and Z. Zong, “Improving the energy efficiency of relational and NoSQL databases via query optimizations,” *Sustain. Comput. Informatics Syst.*, vol. 22, pp. 120–133, 2019, doi: 10.1016/j.suscom.2019.01.017.
- [44] R. Marín, “Los gestores de bases de datos más usados en la actualidad.,” 2019.
<https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>.
- [45] 201 and S. Lee, “Forensic investigation framework for the document store NoSQL DBMS: MongoDB as a case study,” *Digit. Investig.*, vol. 17, pp. 53–65, 2016, doi: 10.1016/j.diin.2016.03.003.
- [46] J. Domínguez, *Cliente PSQL de PostgreSQL*, no. April. 2020.

- [47] S. Sultana and S. Dixit, "Indexes in PostgreSQL," *IEEE Int. Conf. Innov. Mech. Ind. Appl. ICIMIA 2017 - Proc.*, no. Icimia, pp. 512–515, 2017, doi: 10.1109/ICIMIA.2017.7975511.
- [48] A. Viloría, G. C. Acuña, D. J. A. Franco, H. Hernández-Palma, J. P. Fuentes, and E. P. Rambal, "Integration of data mining techniques to postgresQL database manager system," *Procedia Comput. Sci.*, vol. 155, no. 2018, pp. 575–580, 2019, doi: 10.1016/j.procs.2019.08.080.
- [49] J. Villalobos, A. Guevara, A. Reyes, E. G. De León, G. I. Cruz, and J. S. Borbón, "Hacktivismo y DDoS: Tendencias actuales de ataque," *.Seguridad, Cult. prevención TI*, vol. 12, 2012.
- [50] B. Lee, E. K. Dewi, and M. F. Wajdi, "Data security in cloud computing using AES under HEROKU cloud," *2018 27th Wirel. Opt. Commun. Conf.*, pp. 4–8, 2018, [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8372705>.
- [51] A. Francisco and J. E. Taylor, "Designing community-scale energy feedback," *Energy Procedia*, vol. 158, pp. 4178–4183, 2019, doi: 10.1016/j.egypro.2019.01.812.
- [52] B. Wood, *Adobe XD CC Classroom in a Book (2019 Release)*, Pearson Ed. 2019.
- [53] J. Janssen *et al.*, "pyiron: An integrated development environment for computational materials science," *Comput. Mater. Sci.*, vol. 163, no. July 2018, pp. 24–36, 2019, doi: 10.1016/j.commatsci.2018.07.043.
- [54] D. Abbott, "Eclipse integrated development environment," *Linux Embed. Real-Time Appl.*, pp. 67–86, 2018, doi: 10.1016/b978-0-12-811277-9.00005-5.
- [55] L. Beño, R. Pribiš, and R. Leskovský, "Processing data from OPC UA server by using Edge and Cloud computing," *IFAC-PapersOnLine*, vol. 52, no. 27, pp. 240–245, 2019, doi: 10.1016/j.ifacol.2019.12.645.
- [56] V. Yussupov, J. Soldani, U. Breitenbücher, A. Brogi, and F. Leymann, "FaaS: your decisions: A classification framework and technology review of function-as-a-Service platforms," *J. Syst. Softw.*, vol. 175, p. 110906, 2021, doi: 10.1016/j.jss.2021.110906.

5. ANEXOS

Anexo A: Herramienta GTmetrix – Evaluación Eficiencia

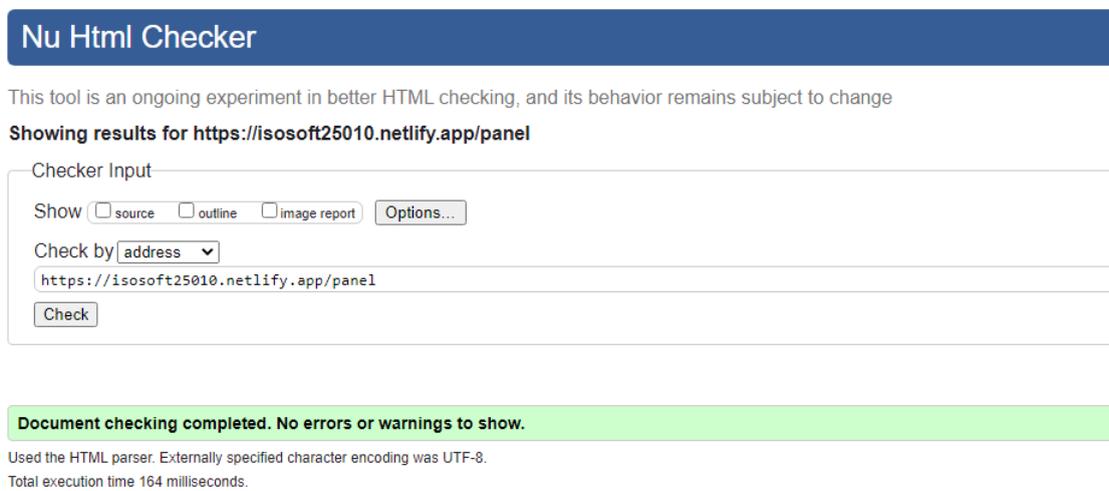
Figura 31: Anexo A: Herramienta GTmetrix – Evaluación Eficiencia



Fuente: Elaboración propia

Anexo B: Herramienta Validator – Evaluación Funcionabilidad

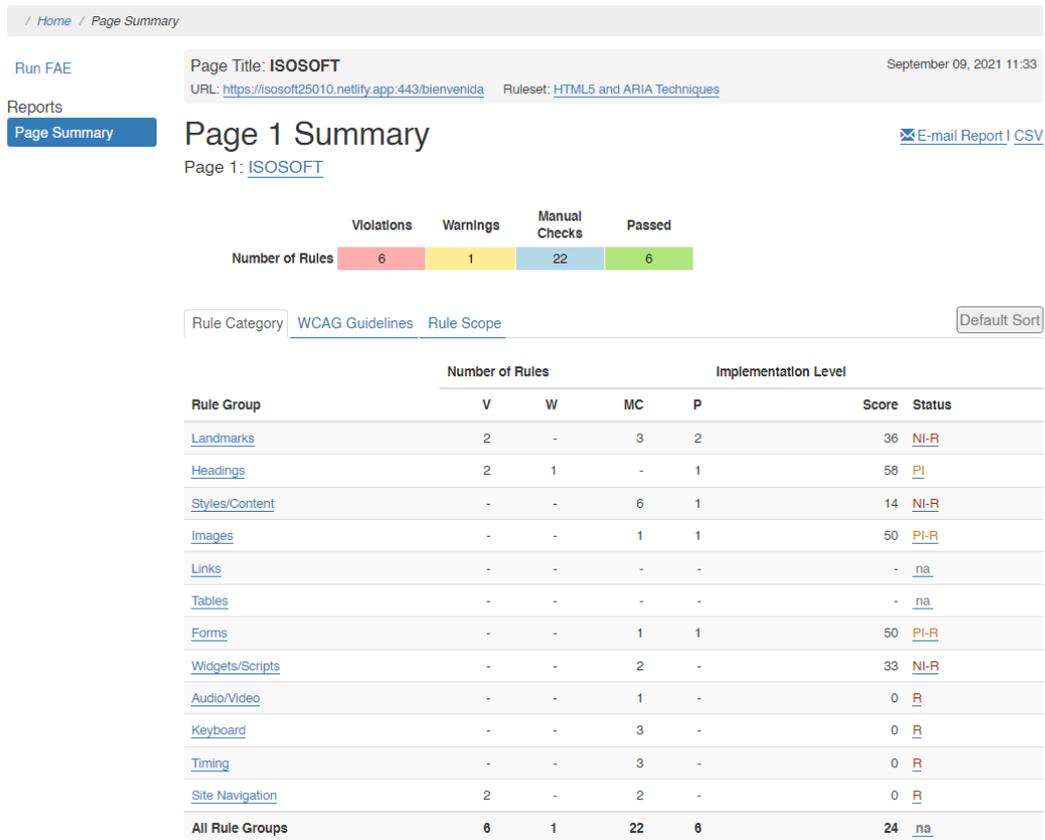
Figura 32: Herramienta Validator – Evaluación Funcionabilidad



Fuente: Elaboración propia

Anexo C: Herramienta FAE – Evaluación Accesibilidad

Figura 33: Herramienta FAE – Evaluación Accesibilidad



Fuente: Elaboración propia

Anexo D: Herramienta MetricSpot – Evaluación Usabilidad

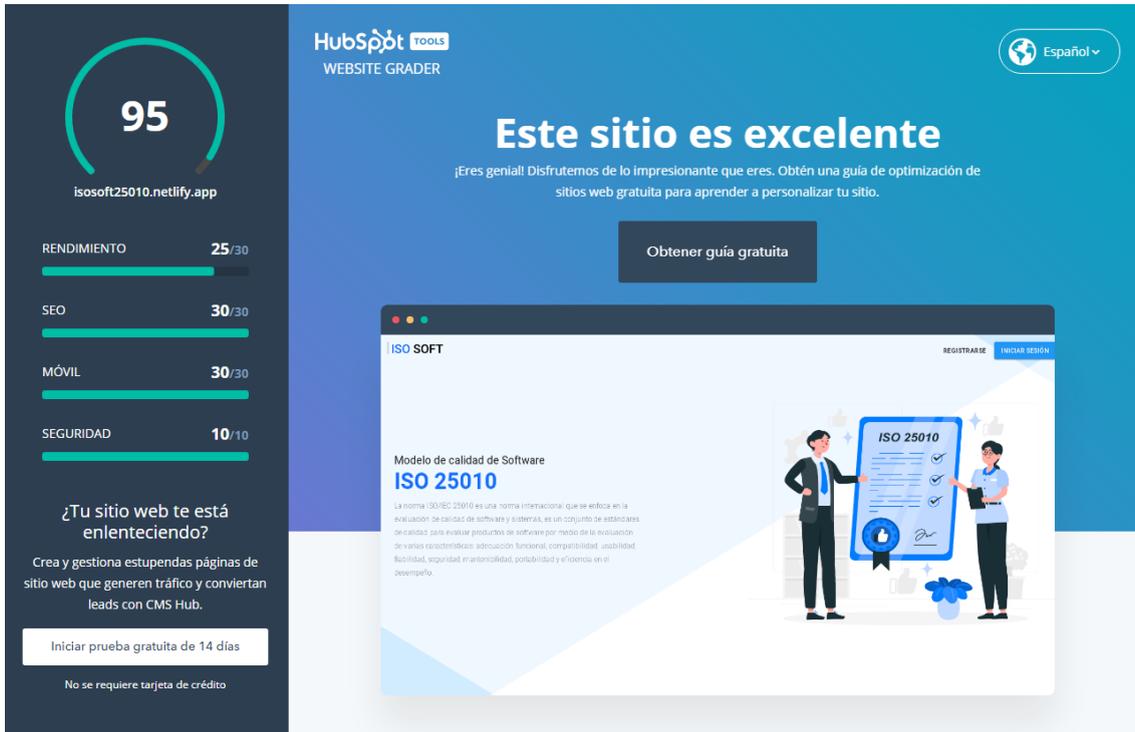
Figura 34: Herramienta MetricSpot – Evaluación Usabilidad



Fuente: Elaboración propia

Anexo E: Herramienta Grader – Evaluación Eficiencia, Seguridad, Portabilidad

Figura 35: Herramienta Grader – Evaluación Eficiencia-Seguridad-Portabilidad



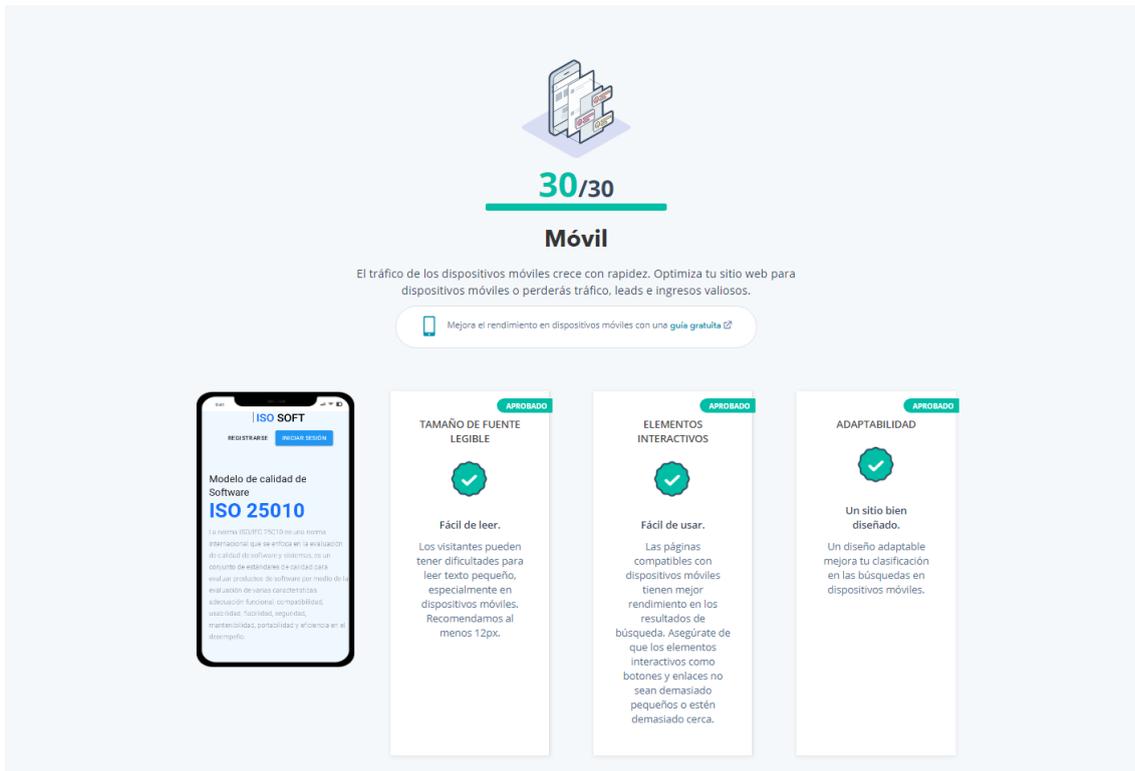
Fuente: Elaboración propia

Figura 36: Herramienta Grader – Evaluación Rendimiento



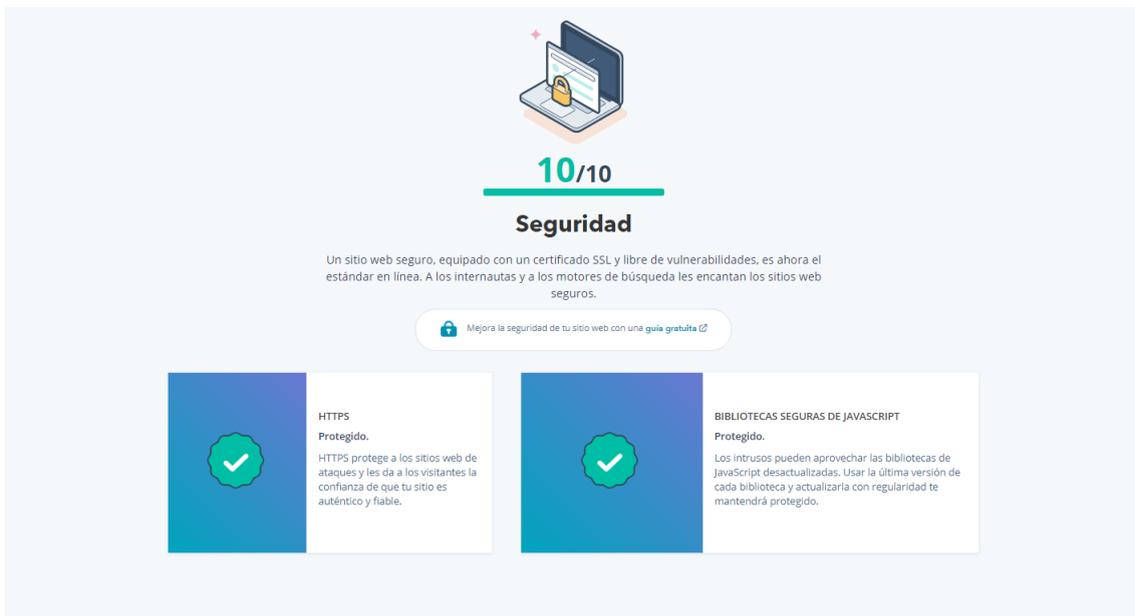
Fuente: Elaboración propia

Figura 37: Herramienta Grader – Evaluación Portabilidad



Fuente: Elaboración propia

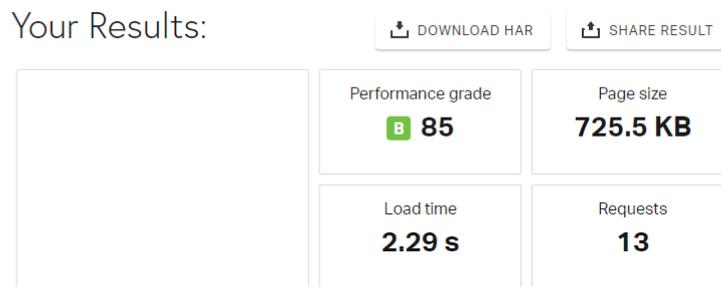
Figura 38: Herramienta Grader – Evaluación Seguridad



Fuente: Elaboración propia

Anexo F: Herramienta Pingdom – Evaluación Eficiencia

Figura 39: Herramienta Pingdom – Evaluación Eficiencia



Improve page performance

GRADE	SUGGESTION	
F 23	Add Expires headers	▼
E 56	Compress components with gzip	▼
A 96	Make fewer HTTP requests	▼
A 100	Avoid empty src or href	▼
A 100	Put JavaScript at bottom	▼
A 100	Reduce the number of DOM elements	▼
A 100	Make favicon small and cacheable	▼

Response codes

RESPONSE CODE	RESPONSES
200 OK	13

Fuente: Elaboración propia