



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE COMPUTACIÓN

SITIO WEB DINÁMICO PARA LA AGRUPACIÓN DE LOS SISTEMAS DE VENTA DE PRODUCTOS AGRÍCOLAS Y PROMOCIÓN DE SITIOS TURÍSTICOS.

Trabajo de titulación previo a la obtención del
Título de Ingenieros en Ciencias de la Computación

AUTORES: ROBBY STEBAN CORONEL FALCON

JUAN FERNANDO OBANDO CASTILLO

TUTOR: FRANKLIN EDMUNDO HURTADO LARREA

Quito – Ecuador

2023

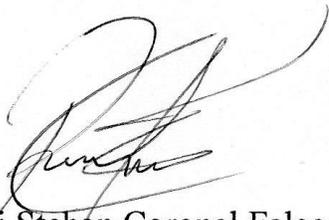
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Robbi Steban Coronel Falcon con documento de identificación N° 1726622614 y Juan Fernando Obando Castillo, y N° 1724494073; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 21 de agosto de 2023

Atentamente.



Robbi Steban Coronel Falcon

1726622614



Juan Fernando Obando Castillo

1724494073

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Robbí Steban Coronel Falcon con documento de identificación N° 1726622614 y Juan Fernando Obando Castillo, y N° 1724494073, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Sitio web dinámico para la agrupación de los sistemas de venta de productos agrícolas y promoción de sitios turísticos”, el cual ha sido desarrollado para optar por el título de: Ingenieros en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 21 de agosto de 2023

Atentamente.



Robbí Steban Coronel Falcon

1726622614



Juan Fernando Obando Castillo

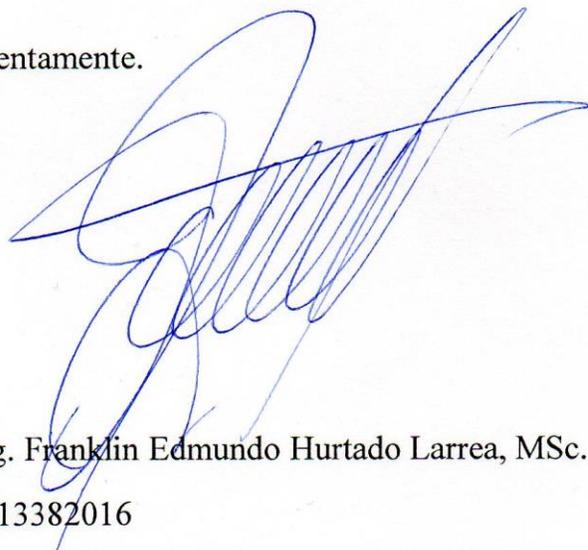
1724494073

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Franklin Edmundo Hurtado Larrea con documento de identificación N° 1713382016, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: SITIO WEB DINÁMICO PARA LA AGRUPACIÓN DE LOS SISTEMAS DE VENTA DE PRODUCTOS AGRÍCOLAS Y PROMOCIÓN DE SITIOS TURÍSTICOS, realizado por Robbi Steban Coronel Falcon con documento de identificación N° 1726622614 y Juan Fernando Obando Castillo, y N° 1724494073, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 21 de agosto de 2023

Atentamente.



Ing. Franklin Edmundo Hurtado Larrea, MSc.

1713382016

DEDICATORIA

Este proyecto de tesis está dedicado, a mis padres que a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento, han sabido formarme con buenos sentimientos, inculcarme hábitos y valores, los cuales me han ayudado a salir adelante buscando siempre el mejor camino.

A mi hermano y abuelita, ya que gracias a su esfuerzo y sacrificio diario me ayudaron a nunca rendirme en los momentos difíciles, puesto que con todo su amor y cariño supieron darme fuerzas para mantenerme constante en conseguir mis metas. Son la principal fuente de inspiración en mi vida personal.

Robbi Steban Coronel Falcon

| | | |
|----|--|----|
| 1. | Resumen | 10 |
| 2. | Abstract..... | 11 |
| 3. | Introducción..... | 1 |
| 4. | Problema..... | 3 |
| 5. | Objetivos generales | 4 |
| 6. | Objetivos específicos | 4 |
| 7. | Revisión de la literatura | 5 |
| | Sitio web dinámico..... | 5 |
| | Sitio web flexible | 5 |
| | Sitio web dinámico..... | 6 |
| | Sistemas web para mejorar el alcance de las organizaciones..... | 6 |
| | Patrones arquitectónicos | 7 |
| | Refactorización de Sistemas web..... | 7 |
| | Reingeniería de Sistemas web | 8 |
| | Proceso de reingeniería de software | 9 |
| | Reestructuración | 10 |
| | Ingeniería inversa..... | 10 |
| | Ingeniería directa | 11 |
| | Estandarización de código..... | 11 |
| | Base de datos no estructurada | 12 |
| | Proxy inverso..... | 12 |
| | Tecnologías..... | 13 |
| | Handlebars | 13 |
| | MongoDB | 13 |
| | Express | 14 |
| | Nginx | 14 |
| | Node.js | 15 |
| | Experiencia de usuario..... | 15 |
| | Usabilidad en sistemas web | 15 |
| | Chatbot en sistemas web..... | 16 |
| 8. | Marco Metodológico | 16 |
| | Modelo para Sistema web para venta de productos agrícolas y sistema web para la promoción de sitios turísticos. | 18 |

| | |
|---|----|
| Chatbot | 20 |
| Sitio web dinámico..... | 21 |
| Requisitos..... | 21 |
| Diseño y construcción | 22 |
| Fase de evaluación | 22 |
| 9. Resultados..... | 22 |
| Construcción de Sitio web dinámico..... | 23 |
| Requerimientos..... | 24 |
| Diseño | 36 |
| Construcción | 40 |
| Sistema web para venta de productos agrícolas..... | 49 |
| Análisis de Inventario | 49 |
| Reestructuración | 54 |
| Ingeniería inversa..... | 55 |
| Ingeniería Directa..... | 56 |
| Sistema web para la promoción de sitios turísticos..... | 56 |
| Análisis de Inventario | 56 |
| Reestructuración | 61 |
| Ingeniería inversa..... | 62 |
| Ingeniería Directa..... | 63 |
| Pruebas | 63 |
| Pruebas de usabilidad | 65 |
| Pruebas de funcionalidad | 68 |
| Pruebas de eficiencia | 73 |
| Perspectiva global del sitio web..... | 75 |
| Publicación del sitio web..... | 84 |
| 10. Conclusiones | 86 |
| 11. Recomendaciones..... | 87 |
| 12. Referencias bibliográficas | 89 |

ÍNDICE DE TABLAS

| | |
|---------------|----|
| Tabla 1..... | 24 |
| Tabla 2..... | 31 |
| Tabla 3..... | 49 |
| Tabla 4..... | 51 |
| Tabla 5..... | 57 |
| Tabla 6..... | 63 |
| Tabla 7..... | 64 |
| Tabla 8..... | 64 |
| Tabla 9..... | 65 |
| Tabla 10..... | 68 |
| Tabla 11..... | 69 |
| Tabla 12..... | 69 |
| Tabla 13..... | 70 |
| Tabla 14..... | 70 |
| Tabla 15..... | 71 |
| Tabla 16..... | 73 |
| Tabla 17..... | 84 |

ÍNDICE DE FIGURAS

| | |
|----------------|----|
| Figura 1..... | 9 |
| Figura 2..... | 17 |
| Figura 3..... | 17 |
| Figura 4..... | 29 |
| Figura 5..... | 30 |
| Figura 6..... | 34 |
| Figura 7..... | 35 |
| Figura 8..... | 38 |
| Figura 9..... | 40 |
| Figura 10..... | 41 |
| Figura 11..... | 43 |
| Figura 12..... | 44 |
| Figura 13..... | 45 |
| Figura 14..... | 46 |
| Figura 15..... | 47 |
| Figura 16..... | 48 |
| Figura 17..... | 52 |
| Figura 18..... | 55 |
| Figura 19..... | 59 |
| Figura 20..... | 62 |
| Figura 21..... | 65 |

| | |
|------------------------|----|
| Figura 22 | 66 |
| Figura 23 | 66 |
| Figura 24 | 67 |
| Figura 25 | 67 |
| Figura 26 | 72 |
| Figura 27 | 72 |
| Figura 28 | 73 |
| Figura 29 | 74 |
| Figura 30 | 74 |
| Figura 31 | 75 |
| Figura 32 | 75 |
| Figura 33 | 76 |
| Figura 34 | 77 |
| Figura 35 | 78 |
| Figura 36 | 79 |
| Figura 37 | 79 |
| Figura 38 | 80 |
| Figura 39 | 81 |
| Figura 40 | 82 |
| Figura 41 | 83 |
| Figura 42 | 83 |
| Figura 43 | 85 |

1. RESUMEN

El presente proyecto de titulación está enfocado en el desarrollo de un sitio web dinámico, en el cual se presentará información relevante que podrá ser totalmente administrable por los usuarios designados, con el principal objetivo de agrupar dos sistemas web heredados, un sistema web para la venta de productos agrícolas y un sistema web para la promoción de sitios turísticos.

El sitio web se creó con el propósito de brindar una experiencia interactiva y personalizada a los usuarios, presentando un sitio web informativo sobre la organización, integrando sistemas web ya existentes desarrollados de forma independiente.

Las metodologías que se utilizaron, por un lado, una basada en cascada y otra, con base en un modelo iterativo, las cuales en cada uno de los procesos se logró la unificación de las dos, dando una metodología con enfoque híbrido que permite el control del desarrollo del sistema en base a retroalimentaciones al finalizar cada etapa. El proceso base en cual se trabajo es de reingeniería para la mejora de los sistemas web existentes.

Los resultados que se obtuvieron en el sitio web dinámico fueron los deseados debido a que se presentó un sitio web completamente administrable con la capacidad de escalar a medida que la organización lo requiera. Se logró establecer una arquitectura solida que podrá ser replicada en otros sistemas web.

En conclusión, este proyecto de titulación logró la creación de un sitio web dinámico que agrupa sistemas web de venta de productos agrícolas y promoción de sitios turísticos, que el uso de tecnologías y procesos permitió la estandarización, la reestructuración y mejora de cada uno de los sistemas.

Palabras clave: *Sitio web dinámico, sistemas web heredados, metodología hibrida, reingeniería*

2. ABSTRACT

This titling project is focused on the development of a dynamic website oriented to an organization, where relevant information will be presented that can be fully administrable by designated users, with the main objective of grouping two inherited web systems, a web system for the sale of agricultural products and a web system for the promotion of tourist sites.

The website was created with the purpose of providing an interactive and personalized experience to users. Presenting an informative website about the organization, integrating already existing web systems developed independently.

The methodologies that were used are cascade methodology and iterative methodology, which in each of the processes the unification of the two was achieved, giving a hybrid methodology that allows control of the development of the system based on feedback at the end of each stage. The basic process in which work is reengineering for the improvement of existing web systems.

The results obtained were the desired ones because a fully manageable dynamic website was presented with the ability to scale as the organization requires it. It was possible to establish a solid architecture that can be replicated in other web systems.

In conclusion, this titling project achieved the creation of a dynamic website that groups web systems for the sale of agricultural products and promotion of tourist sites. Where the use of technologies and processes allowed the standardization, restructuring and improvement of each of the systems.

Keys words: Dynamic website, legacy web systems, hybrid methodology, reengineering

3. INTRODUCCIÓN

Varios sectores, como la agricultura y el turismo, han sido transformados por la tecnología y la digitalización en el mundo actual. Este proyecto de titulación se enfoca en la creación de un sitio web dinámico con el objetivo principal de agrupar sistemas web de venta de productos agrícolas y promoción de lugares turísticos específicos, dirigidos hacia una organización, este proyecto tiene como base tecnologías como Express, MongoDB y NodeJs, los proyectos que fueron agrupados en un enfoque de perspectiva del usuario han pasado por procesos de reingeniería para reestructurar todos los sistemas involucrados.

El brindar una experiencia interactiva y personalizada a los usuarios, es esencial al crear un sitio web dinámico. En el sector agrícola, se ha observado un aumento en la demanda de soluciones que permitan establecer conexiones entre los productores y los consumidores, con el objetivo de facilitar la venta directa de alimentos frescos y promover el desarrollo económico local. Por otro lado, en el ámbito turístico, es fundamental contar con una plataforma que resalte los atractivos y recursos de cada región, con el fin de fomentar el turismo responsable y sostenible.

En la ejecución del proyecto se han utilizado algunas tecnologías en el back-end para lograr estos objetivos. La implementación de un framework como Express, con un entorno de ejecución de aplicaciones web Node.js, los cuales han permitido la creación de una infraestructura robusta, eficiente y dinámica para el manejo de solicitudes y respuestas en el servidor. Para almacenar, recuperar datos y administrar los contenidos dentro del sitio web dinámico se implementó una base de datos no estructurada con MongoDB, la cual ayuda con la escalabilidad y flexibilidad que necesita el proyecto.

Además, en el front-end se ha utilizado handlebars, un motor de plantillas que permite crear páginas HTML basadas obtenidas de la base de datos de manera eficiente para crear una presentación visual atractiva y dinámica del sitio web. Esto ha facilitado la actualización de variables y personalización de la estructura del sitio web, de esta manera permite adaptar a las necesidades específicas del usuario.

También, se ha llevado a cabo un proceso de reingeniería en cada uno de los sistemas involucrados para mejorar su funcionalidad y estructura. Se han propuesto cambios y mejoras significativas que permitan optimizar su rendimiento y facilitar su integración en el sitio web dinámico, al identificar y analizar las debilidades y deficiencias de los sistemas existentes.

En conclusión, este proyecto de titulación se centró en crear un sitio web dinámico que combine sistemas web para la venta de productos agrícolas y la promoción de sitios turísticos para una organización. La reestructuración y mejora de cada uno de los sistemas ha sido posible gracias al uso de tecnologías que ayudan a que los sistemas tengan funcionalidades escalables que se adaptan al entorno en el que se trabaje, así como los procesos de reingeniería implementados. Esto ha proporcionado una plataforma útil y adecuada a la necesidad de gestionar contenidos de forma dinámica para promover la agricultura local y el turismo responsable.

4. PROBLEMA

Actualmente hay un sistema separado de venta de productos agrícolas y promoción de sitios turísticos, las cuales son dos sistemas web desarrollados por estudiantes de la Universidad Politécnica Salesiana. Para reducir los costos al lanzar en producción, se requiere la creación de un sitio web que agrupe desde un enfoque de perspectiva del usuario los dos sistemas mencionados. Al agrupar estos sistemas, se permitirá tener una sola infraestructura que soportará el sitio web dinámico que el usuario podrá interactuar con los dos sistemas.

La creación de un sitio web dinámico es importante para que los usuarios puedan interactuar entre los distintos sistemas web y que de esta forma conozcan sobre la provincia de Cotopaxi. Este sitio web dinámico permite la integración de los sistemas de venta de productos agrícolas y promoción de sitios turísticos. El sistema de producción de sitios turísticos se unió al sitio web dinámico con una nueva función llamada Chatbot. Este Chatbot podrá mantener conversaciones con los usuarios y proporcionar información sobre los sitios turísticos registrados en el sistema, ayudando a los usuarios a elegir su mejor opción.

Por lo tanto, se propone crear un sitio web dinámico agrupado en el que los usuarios puedan interactuar con los dos sistemas completamente distintos y a su vez enterarse lo que ofrecen, cuáles son sus noticias y proyectos actuales, para de esta manera lograr adaptarse con lo que se encuentra a la vanguardia en la actualidad y no estancarse de manera local con sus diferentes emprendimientos.

5. OBJETIVOS GENERALES

Proporcionar un sitio web dinámico para la agrupación de los sistemas de venta de productos agrícolas y promoción de sitios turísticos. En el cual se pueda informar, realizar procesos de compra/venta de productos agrícolas o promoción de servicios de turismo.

6. OBJETIVOS ESPECÍFICOS

- Implementar ajustes funcionales y de arquitectura para los sistemas de venta de productos agrícolas y promoción de sitios turísticos.
- Diseñar un patrón arquitectónico que ayuden a agrupar los sistemas de venta de productos agrícolas y promoción de sitios turísticos.
- Creación de Chatbot para facilitar el acceso a la información para los usuarios.

7. REVISIÓN DE LA LITERATURA

SITIO WEB DINÁMICO

Un sitio web es una colección de sistemas o páginas web relacionadas que están diseñadas para proporcionar información, presentar contenido o promover a una organización, empresa o individuo. Los sitios web suelen ser estáticos o con contenido actualizado periódicamente y su principal objetivo es brindar información o contenido como artículos o noticias, presentar productos o servicios entre otros, a los usuarios del sitio (Suh. W ,2005).

Sitio web flexible

En la actualidad se puede acceder a un sitio web desde cualquier dispositivo inteligente que tenga conexión a internet, ya no solo desde una computadora sino desde dispositivos móviles, televisores entre otros. Por ello se ve en la necesidad de desarrollar sitios web que puedan adaptarse y proporcionar una experiencia adecuada en diferentes dispositivos y resoluciones de pantalla. Este desarrollo se lleva a cabo para dar respuesta a las características del dispositivo que se utiliza para visualizarlo, y así evitando el desarrollo de varias versiones separadas para cada dispositivo (Doan. P, 2017).

Para la creación de un sitio web flexible existen técnicas de diseño y desarrollo que permiten que el contenido, la estructura y el diseño se ajusten y se presenten de manera óptima según las capacidades y restricciones de cada dispositivo. Esto permite que el sitio web se expanda y contraiga de manera fluida, manteniendo la legibilidad y la usabilidad del contenido. Dentro del enfoque de mejora progresiva en el diseño web se establece que de primera instancia se comienza con un sitio web sólido y funcional que funciona en dispositivos con capacidades limitadas y posteriormente agregar capas de mejoras y funcionalidades adicionales para dispositivos más

modernos y capaces. De esta manera se asegura una experiencia básica de calidad para todos los usuarios independientemente de su dispositivo (Mickley. Z, 2008).

Sitio web dinámico

Un sitio web es dinámico cuando tiene la capacidad de mostrar contenido interactivo y personalizado en tiempo real, adaptándose a las acciones y preferencias del usuario. A diferencia de un sitio web normal o estático que e presenta la información de manera predefinida, los sitios web dinámicos utilizan tecnologías y herramientas que permiten la generación de contenido en tiempo real y la interacción con el usuario (Keith J, 2005).

Para que un sitio web sea dinámico se necesitan tecnologías que permitan manipular y actualizar el contenido de un sitio web sin tener que actualizar o recargar por completo, por lo que brinda una experiencia más fluida. Dentro de las principales características de este tipo de sitios es la solución que brinda a los administrados del sitio al momento de actualizar, modificar o agregar sus contenidos ya que lo pueden hacer sin necesidad de modificar el código fuente ya que al tener la funcionalidad de ser dinámico pueden hacerlo en la misma interfaz de manera más accesible (Rajshkhar. A.P, 2008).

SISTEMAS WEB PARA MEJORAR EL ALCANCE DE LAS ORGANIZACIONES

En la actualidad los sistemas o páginas web son fundamentales para mejorar el alcance y la efectividad de las organizaciones en diferentes aspectos. Permiten a las organizaciones expandir su presencia, aumentar la accesibilidad a sus servicios y mejorar la eficiencia de sus operaciones. A través de los sistemas web se logra alcanzar a audiencias más amplias, sin restricciones geográficas y promover sus productos o servicios de manera efectiva. Los sistemas web ofrecen sus servicios la mayoría de tiempo desde cualquier lugar, debido que en cierto tiempo hay

actualizaciones, mantenimiento u otros factores que conlleva la detención del sistema (Kotler. P, 2017).

Los sistemas web pueden automatizar procesos internos, como la gestión de inventario, el seguimiento de pedidos y la comunicación interna, lo que ahorra tiempo y recursos. Esto mejora la productividad de la organización permitiendo una respuesta más rápida y eficiente a las demandas del mercado (Berisha-Shaqiri. A, 2015).

PATRONES ARQUITECTÓNICOS

Los patrones arquitectónicos son soluciones para problemas recurrentes en el diseño de software y sistemas. Estos patrones ayudan a los arquitectos de software a tomar decisiones informadas y a establecer una base sólida para el desarrollo de sistemas escalables. Uno de los patrones más comunes y reconocido dentro de la comunidad de desarrollo de software es Modelo-Vista-Controlador, en el cual separa los datos, la lógica del negocio y la presentación en tres componentes distintos, esto permite una mayor modularidad y flexibilidad en el diseño del software (Fowler. M, 2005).

Existen varios patrones arquitectónicos dentro del desarrollo de software que ayudan a organizar y estructurar las diferentes capas y componentes de una aplicación o sitio web, facilitando la mantenibilidad y escalabilidad del sistema, también ayudan en la comunicación y el descubrimiento dinámico entre servicios al igual que la capacidad de adaptación ante fallos (Newman. S, 2015).

REFACTORIZACIÓN DE SISTEMAS WEB

La refactorización dentro de los sistemas web es un proceso en el desarrollo de software que implica mejorar la estructura interna y la calidad del código de un sistema existente sin cambiar su

funcionalidad externa. A través de la refactorización se puede optimizar el rendimiento, aumentar la mantenibilidad y reducir la complejidad de los sistemas web. Existen técnicas de refactorización que propone patrones y enfoques para reorganizar el código, eliminar duplicaciones y aplicar buenas prácticas de diseño (Fowler. M, 2005).

La refactorización permite mejorar continuamente el diseño y la calidad del código y es recomendable realizarla de manera constante en lugar de esperar a que el código se deteriore, ya que al refactorizar de tal manera los equipos pueden mantener el código limpio y adaptable, lo que facilita la evolución del sistema web a medida que los requisitos cambian y surgen nuevas funcionalidades. Una de las técnicas que permiten refactorizar a un sistema es transformar gradualmente el código existente en pequeños pasos, mejorando su legibilidad y expresividad, esto permite la introducción de nuevas características y mejoras en el sistema web de manera más segura y controlada (Beck. K, 2012).

REINGENIERÍA DE SISTEMAS WEB

La reingeniería dentro de los sistemas web implica la transformación y mejora de sistemas existentes para adaptarlos a nuevas tecnologías, mejorar su rendimiento y mantenibilidad. La reingeniería de sistemas web no solo se enfoca en mejorar código ya existente, sino también en optimizar la arquitectura y la interfaz de usuario. Este enfoque permite actualizar los sistemas web para satisfacer las demandas cambiantes y mantenerse actualizadas con las nuevas tecnologías y tendencias (Pooley. R, 2000).

Los desafíos más comunes dentro de la reingeniería son la migración a versiones más recientes del lenguaje en el que se esté trabajando el sistema, la mejora de la seguridad y la optimización del rendimiento, además es importante tener una estrategia clara de refactorización para garantizar el éxito de la reingeniería. Dentro de este procedimiento existen técnicas que están asociadas a la

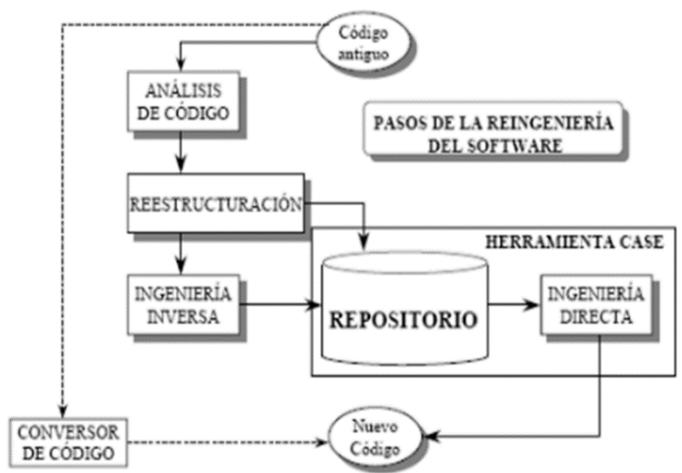
refactorización ya que una las modificaciones dentro de la reingeniería es la modificación gradual de código ya existente para desacoplar y aislar las partes problemáticas del sistema (Gane. C, 2000).

Proceso de reingeniería de software

El proceso que está contemplado dentro de la reingeniería de software es analizar, comprender y modificar un sistema existente con el objetivo de mejorar su rendimiento y mantenibilidad. Este proceso implica el rediseño y la reestructuración de los componentes del software en este caso de los sistemas web, para adaptarlos a los nuevos requisitos y estándares de calidad. Otro de los paradigmas de la reingeniería es la modernización de sistemas heredados, es decir es necesario utilizar este proceso cuando los sistemas existentes se vuelven obsoletos o ineficientes, se ofrecen enfoques y técnicas para abordar los desafíos asociados con la modernización. Las técnicas que se aplican después de identificar lo mencionado anteriormente, son refactorización, ingeniería inversa y la reestructuración del código para mejorar el sistema (Roger. P, 2010).

Figura 1

Proceso de reingeniería de software



Nota. Tomado de *Ingeniería del software. Un enfoque práctico. Séptima edición*, por R.S. Pressman, 2010.

Como se puede observar en la *Figura 1* este proceso es de tipo cíclico, es decir que en cada etapa las actividades que se realicen se pueden retroalimentar y regresar a completar las actividades siguientes o regresar sea el caso de que cambien los requisitos. El proceso no se mantiene de manera cíclica siempre, en cierto punto en una de las actividades el proceso puede finalizar si así el sistema lo permita.

Reestructuración

La reestructuración en el ámbito de mejora continua del software es el proceso que involucra realizar cambios en el diseño interno del código sin modificar su comportamiento externo. Por otro lado, la reestructuración ayuda a mejorar la legibilidad y facilita la adaptabilidad para futuras mejoras. Además, proporciona una serie de técnicas y patrones de reestructuración que los desarrolladores pueden aplicar para mejorar la calidad del código. Otro de los enfoques dentro de la reestructuración en el proceso de reingeniería de software es la mejora en la modularidad, reducir la complejidad y mejorar la mantenibilidad del sistema. Existen desafíos dentro de esta etapa como la detección de dependencias y la evaluación del impacto de los cambios. Es importante tener en cuenta que al momento de aplicar la reestructuración se debe realizar una planificación controlada, es decir utilizando técnicas de pruebas y asegurando que el sistema continúe funcionando correctamente después de los cambios. De igual forma conecta con los requisitos previamente establecidos (Martin. F, 2002).

Ingeniería inversa

La ingeniería inversa es un proceso que implica analizar un sistema existente para comprender su funcionamiento interno, su estructura y su diseño. A través de la ingeniería inversa se puede

obtener información valiosa sobre un sistema, incluso cuando la documentación o el código fuente original no están disponibles. Lograr obtener información de un sistema ya existente es fundamental para realizar cambios y mejoras efectivas en el sistema sin comprometer su funcionamiento. Mediante técnicas de ingeniería inversa se logra analizar la estructura y el diseño al momento de identificar áreas problemáticas y a comprender las relaciones entre cada uno de los componentes del sistema, esto permite abordar la complejidad del software y tomar decisiones informadas para mejorar su mantenibilidad y rendimiento (Raja. V, 2008).

Ingeniería directa

Uno de los pasos más importantes dentro del proceso de reingeniería es la aplicación de ingeniería directa, lo que significa es la etapa en la que se implementan las nuevas funcionalidades modernizando de esta manera el sistema, esto implica actualizar y mejorar tecnologías, arquitecturas y componentes obsoletos en el sistema existente. La implementación de estos cambios se debe realizar minuciosamente, asegurando que se integren todos los cambios propuestos con los componentes existentes. Como se mencionó anteriormente, esto puede incluir en la integración de nuevas funcionalidades, cambios visuales, cambio o eliminación de componentes obsoletos y la modificación de la estructura del sistema para mejorar su mantenibilidad. Existen técnicas de validación y verificación que son utilizadas para asegurar que el sistema funcione correctamente después de los cambios implementados (Roger. P, 2010).

ESTANDARIZACIÓN DE CÓDIGO

En el desarrollo de software, la estandarización del código es esencial porque promueve la legibilidad, la mantenibilidad y la colaboración eficiente. Dentro de este paradigma se resalta la necesidad de escribir el código de manera consistente y clara, siguiendo normas claras. El uso de nombres descriptivos, la estructura adecuada del código y evitar repeticiones innecesarias, facilitan

la comprensión y el mantenimiento del software. El proceso de desarrollo de software incluye la estandarización del código. Las convenciones de codificación consistentes reducen los errores y facilitan la colaboración en los equipos de desarrollo. Además, la estandarización del código facilita la identificación de patrones y problemas comunes. Esto ayuda a mantener una base de código de alta calidad y previene la aparición de errores al sistema. Es importante adaptar la estandarización del código a las necesidades y prácticas de cada proyecto o equipo de desarrollo, para ello se establece un conjunto claro de reglas y convenciones consensuadas para que se apliquen de manera consistente (Robert. M, 2009).

BASE DE DATOS NO ESTRUCTURADA

Las bases de datos no estructuradas son una forma de almacenamiento diferente a las bases de datos relacionales ya que no maneja las relaciones de forma tradicional, es decir la gestión de la información no es tabular y no tiene una estructura definida. Este tipo de bases de datos son adecuadas para manejar datos complejos, no relacionales y de formato variable. Las características principales de este tipo de base de datos es su flexibilidad y escalabilidad que le permiten al sistema adaptarse a cambios, existen diferentes categorías dentro de las bases de datos no estructuradas, como las de tipo documento, las de clave-valor y de grafos. Es importante tener en cuenta que al momento de utilizar una base de datos de tipo no estructurada se requiere un entendimiento claro de los requisitos y las necesidades del proyecto, ya que depende del alcance y el almacenamiento de los datos, ya que al ser tan flexibles se considera aspectos como la integridad de los datos y la seguridad (Prمود. S, 2013).

PROXY INVERSO

El proxy inverso es una técnica utilizada para mejorar la seguridad, escalabilidad y el rendimiento de los sistemas web. Un proxy inverso actúa como intermediario entre los clientes y los servidores

web, recibiendo las solicitudes de los clientes y redirigiéndose a los servidores correspondientes. De igual forma el proxy inverso ofrece beneficios como el equilibrio de carga, el almacenamiento en caché de contenido estático y la protección de los servidores contra ataques directos como ataque de denegación de servicios. Otra de las características importantes dentro del manejo de proxy inverso es la administración de sesiones. Para implementar un proxy inverso dentro de un servidor web es importante considerar aspectos como la seguridad, el monitoreo y la escalabilidad al implementar en un entorno de producción (Martin. F, 2018).

TECNOLOGÍAS

Handlebars

Handlebars es un motor de plantillas utilizado en el desarrollo web que permite la generación dinámica de contenido HTML. Este sistema de plantillas simplifica la creación de vistas en sistemas y sitios web al proporcionar una sintaxis clara y legible, junto con características poderosas. Esta tecnología tiene la capacidad de manejar la lógica condicional y los bucles, lo que permite una generación dinámica de contenido basada en los datos proporcionados por el sistema. La importancia de la legibilidad y la simplicidad de la sintaxis de Handlebars lo que facilita el mantenimiento y la colaboración en proyectos de desarrollo web. Utilizar un enfoque basado en plantillas de Handlebars para generar contenido dinámico lo que facilita la actualización y modificación de las vistas sin necesidad de modificar directamente el código ya existente (Hartl. M, 2013).

MongoDB

MongoDB es una base de datos de tipo NoSQL cuya principal ventaja es su enfoque flexible y escalable. Este sistema de gestión de base datos se basa en el modelo de documentos, lo que permite almacenar y manipular datos en forma de documentos JSON que pueden representar

estructuras complejas y anidadas. Esta flexibilidad permite almacenar y recuperar datos de manera eficiente, evitando la necesidad de unir varias tablas o hacer cambios de esquema complicados. Con MongoDB se puede manejar grandes volúmenes de datos y altas cargas de trabajo de manera eficiente, adaptándose a las necesidades cambiantes de una aplicación en crecimiento. Usar este tipo de base de datos ayuda cuando se desarrolla un sitio web con un enfoque dinámico o flexible ya que al tener este tipo de alcance se complementa de manera eficiente con las características que presenta MongoDB a comparación a bases de datos de tipo SQL que tienen un enfoque más estático y con otro tipo de estructura que limita el uso o crecimiento del sitio web final (Bradshaw. S, 2019).

Express

Express es un framework web para Node.js ampliamente utilizado en el desarrollo de aplicaciones web que proporciona características importantes que facilitan la creación de aplicaciones web rápidas y eficientes. Express permite crear aplicaciones web de manera rápida y sencilla mediante la definición de rutas y controladores. Además, es un framework minimalista, lo que significa que ofrece solo las funcionalidades esenciales para el desarrollo web, lo que permite personalizar y extender la aplicación según las necesidades específicas. Otra de las ventajas que ofrece Express es el enrutamiento sencillo y flexible, lo que permite definir rutas para diferentes URL y métodos HTTP, de tal manera que ayuda a procesar las solicitudes y respuestas en cada solicitud web, brindando un control preciso sobre el flujo de datos y la lógica de la aplicación (Mardan. A, 2014).

Nginx

Nginx es un servidor web de código abierto que puede ser configurado como un servidor de proxy inverso, esta tecnología es una de las más reconocidas debido a su rendimiento, escalabilidad y capacidad para manejar una gran cantidad de conexiones recurrentes. Una de las características

importantes de Nginx es su arquitectura modular, que permite un rendimiento óptimo y una configuración flexible. También resalta por su capacidad para manejar un alto volumen de solicitudes manteniendo una respuesta rápida del contenido. Nginx fue diseñado para resolver los problemas de escalabilidad y rendimiento encontrados en otros servidores web. Otra de las ventajas de utilizar este servidor es el manejo de varias conexiones simultáneas con recursos mínimos. Como se mencionó anteriormente, tiene la capacidad de actuar como servidor proxy inverso, dando una configuración sencilla y con un bajo consumo de recursos, y de esta manera mejorando la seguridad y el rendimiento de los sistemas web (Martin. F, 2018).

Node.js

Node.js es un entorno de ejecución de código abierto que permite la creación de sistemas web del lado del servidor con JavaScript. Las características principales de Node.js son su rendimiento y escalabilidad, lo cual se debe a su arquitectura basada en eventos y su capacidad para manejar múltiples conexiones simultáneamente. También tiene la capacidad de desarrollar aplicaciones en tiempo real y su integración con paquetes y módulos de terceros a través de npm (node package manager). Una de las ventajas más importantes de Node.js es su capacidad de ser un solo subproceso no bloqueante, lo que permite un manejo eficiente de solicitudes y una mayor capacidad de respuesta en sistemas web en tiempo real (Cantelon et al., 2014).

EXPERIENCIA DE USUARIO

Usabilidad en sistemas web

La usabilidad en sistemas web es un aspecto fundamental para garantizar una experiencia satisfactoria y efectiva para los usuarios. Los usuarios deben tener la capacidad de comprender y utilizar un sistema web de manera intuitiva, sin la necesidad de analizarlo demasiado. Además, es importante la simplicidad en el diseño ya que de este modo se evita elementos confusos o

innecesarios que puedan dificultar la navegación y el uso del sistema web. Es importante realizar pruebas de usabilidad para evaluar la facilidad de uso de un sistema web y descubrir posibles problemas. Existen normas que definen a la usabilidad como la medida en que un sistema puede ser utilizado por usuarios específicos para lograr objetivos específicos con satisfacción en un contexto específico de uso (NSAI, 2018).

Chatbot en sistemas web

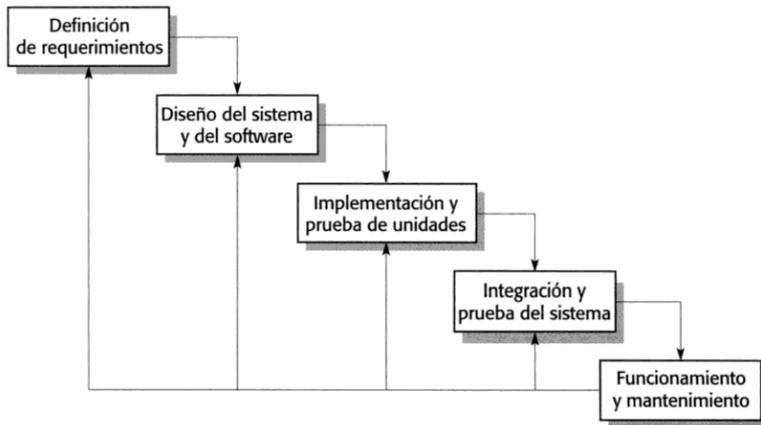
El uso de Chatbots en sistemas web ha logrado tener la capacidad para mejorar la interacción con los usuarios y brindar respuestas rápidas y automatizadas. Los Chatbots son programas que pueden proporcionar respuestas inmediatas a las consultas de los usuarios, sin la necesidad de esperar la atención de una persona. Además, pueden ayudar a los usuarios navegar por un sistema web, proporcionando asistencia en la búsqueda de información o realizando tareas específicas. Existen varios tipos de esta herramienta, unos más sencillos como consultas y respuestas rápidas, pero también existen otros más complejos que logran aprender de las interacciones con los usuarios y mejorar con el tiempo, brindando respuestas más precisas y personalizadas. El uso de cada uno depende del tipo de sistema web en el que se esté implementando esta herramienta ya que cada uno tiene cierto nivel de complejidad al momento de desarrollarlo (Muldowney. O, 2017).

8. MARCO METODOLÓGICO

Se decidió utilizar una metodología híbrida debido a la naturaleza del proyecto y la necesidad de tener un plan sólido para comenzar y garantizar que se cumplan los objetivos. La metodología híbrida seguirá la estructura de un modelo en cascada o lineal, con una línea de tiempo definida con etapas, y al finalizar cada una de estas etapas, se utilizará un modelo incremental para ayudar con la adaptabilidad o cambios en el proyecto. El objetivo de esta metodología es establecer tiempos, lo que permitirá que los procesos se desarrollen de manera flexible y adaptable. También, permitirá garantizar que los procesos funcionen correctamente antes de avanzar a la siguiente etapa en la línea de tiempo establecida.

Figura 2

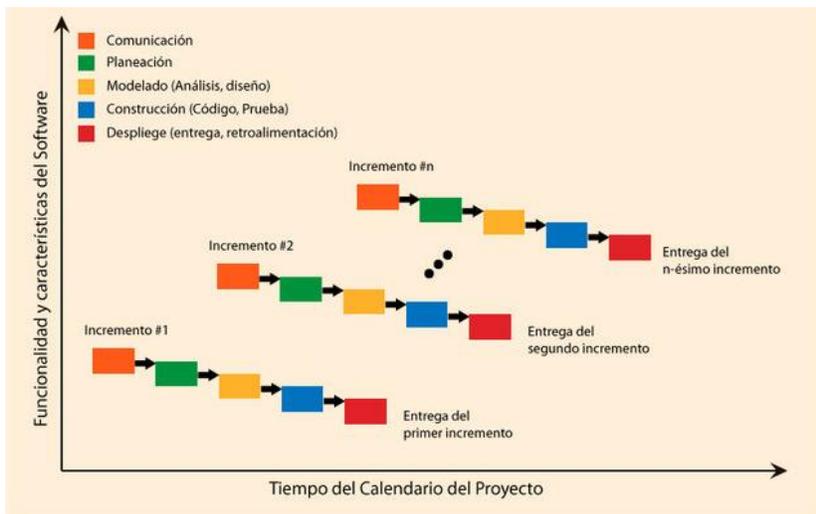
Metodología de cascada



Nota. Tomado de *Ingeniería del software Séptima edición*, por I. Sommerville, 20005.

Figura 3

Metodología iterativa



Nota. Tomado de *Desarrollo de una aplicación para el descubrimiento y supervisión de procesos de IDboxRT*, por E.B. Villegas, 2017.

La metodología híbrida, como se ve en las Figuras 2 y 3, incluye una serie de pasos que están marcados en líneas de tiempo ágiles y predecibles, así como la participación de roles específicos y el uso de herramientas clave para mejorar el desempeño y lograr resultados valiosos. La

metodología híbrida se beneficiará de los resultados obtenidos de la investigación previamente realizada. El siguiente proceso se aplicó a cada uno de los sistemas web, destacando los componentes clave de cada uno, tal como se describen a continuación:

MODELO PARA SISTEMA WEB PARA VENTA DE PRODUCTOS AGRÍCOLAS Y SISTEMA WEB PARA LA PROMOCIÓN DE SITIOS TURÍSTICOS.

El enfoque para el sistema web de venta de productos agrícolas y el sistema para la promoción de sitios turísticos es la reestructuración para la solución de problemas y mejora visual debido que se busca optimizar la funcionalidad y la experiencia del usuario, para ello la metodología que se utiliza en primera instancia es de tipo tradicional dando un proceso lineal y al final de la etapa dando una retroalimentación con un proceso incremental. Dado que los dos sistemas son heredados y tienen el mismo enfoque se utiliza los mismos procesos y metodologías para completar los objetivos planteados.

Para comenzar con el proceso que se llevó a cabo dentro del sistema está basado en el proceso de reingeniería ya que cada uno de los pasos está relacionado con los objetivos contemplados en el sistema. Por ello la primera etapa es el análisis y evaluación del sistema actual, este paso es importante para comprender las deficiencias y áreas de mejora del sistema web. Se analizó la arquitectura, la funcionalidad y la usabilidad del sistema, con lo que se logra examinar los procesos de registro, compra y búsqueda de los productos ingresados en el sistema, de igual forma se examinaron funcionalidades adicionales como la pasarela de pagos y el registro de usuarios. Además, se realizó un seguimiento de los comentarios y opiniones de los usuarios del sistema para identificar problemas específicos y funcionalidades que presentaban inconvenientes, debido a que son sistemas heredados es importante para entender el origen de los problemas y donde enfocarse para solucionarlos.

Una vez identificados los problemas, se procede al análisis si es necesario la reestructuración de la arquitectura del sistema, ya que en este paso incluye la optimización en la base de datos para mejorar el rendimiento, reorganización del código para una mayor escalabilidad, asegurando una comunicación fluida entre todos los componentes del sistema. La reestructuración es fundamental para abordar las deficiencias identificadas en el paso anterior, brindando una plataforma con una base más sólida y confiable.

La siguiente etapa es primordial para la interacción con el usuario ya que corresponde la mejora en la interfaz visual, debido a que tiene un impacto directo en la experiencia del usuario. En esta etapa, se lleva a cabo una mejora en la usabilidad y la interfaz de usuario. Por la naturaleza de los sistemas, al estar fuertemente relacionados es importante mantener una estructura y un estándar visual para todos los sistemas involucrados, logrando una integración adecuada en el sistema final. Además, el sistema contiene una interfaz legible y amigable para todos los usuarios que ingresen a cualquier sistema.

La siguiente etapa está relacionada con la optimización de la experiencia del usuario, la misma que se enfoca en simplificar y mejorar los procesos que los usuarios realizan en el sistema. Se analizan y mejoran los flujos de registro, compra y búsqueda de productos para que cada proceso sean más intuitivos y eficientes. El principal objetivo es la eliminación de formularios o pasos adicional y se proporciona una guía clara al usuario en cada etapa del proceso. La navegación se simplifica y se asegura que los usuarios puedan encontrar fácilmente la información que necesitan.

Como última etapa, es la implementación y pruebas, que se llevan a cabo la corrección de errores y problemas que fueron identificados durante el proceso. Se corrigen enlaces caídos, errores en las validaciones de datos, problemas que afecten a la funcionalidad y confiabilidad del sistema. Se llevan a cabo pruebas para asegurarse de que todas las funcionalidades estén operativas y sin

problemas. Esto implica realizar pruebas en base al enfoque del proyecto para asegurar que el sistema funcione correctamente en diferentes escenarios. Una vez completada las mejoras y correcciones, se realiza un seguimiento del rendimiento del sistema y se recopilan los comentarios de los usuarios, que validará la estabilidad y la satisfacción del usuario para detectar posibles problemas que no estaban contemplados, al igual que realizar ajustes adicionales en el caso de que sea necesario o sea un requisito indispensable para el funcionamiento del sistema.

CHATBOT

Como parte del proceso de reingeniería para el sistema de promoción de sitios turísticos se realizó el análisis para la implementación de una de las tecnologías más actuales que corresponde a la implementación de un programa de Chatbot. El enfoque que tiene es la búsqueda de lugares o servicios turísticos que ofrece el sistema en base a las necesidades del usuario presentando las actividades que puede realizar y los sitios que se adaptan mejor a las actividades seleccionadas. Para el desarrollo de este sistema se basó en una metodología en cascada es decir de manera lineal siguiendo el siguiente proceso:

Primero se analizaron los requisitos para este programa los cuales son las posibles preguntas y respuestas por parte del usuario y el sistema, como se mencionó el principal objetivo del Chatbot es presentar los lugares o servicios turísticos en base a las actividades que deseen hacer los usuarios. Para ello se diseñó un programa de Chatbot en el que su principal función es mostrar un mensaje por parte del sistema con actividades relacionadas en la base de datos y solicitando al usuario la elección entre 2 opciones para continuar o retroceder el proceso, una vez que las actividades si estén relacionadas con las necesidades el usuario se mostraran los sitios turísticos agregados al sistema con esas características, dando un acceso directo al lugar que deseen visitar. Una vez que el diseño sea solido se procede a la implementación realizando las conexiones a las

tablas correspondientes dentro de la base de datos, analizando las columnas necesarias que estén relacionadas con las actividades que se encuentran al momento de registrar un nuevo lugar o servicio turístico y de este modo cumplir con el objetivo del programa.

SITIO WEB DINÁMICO

El enfoque usado para el desarrollo del sitio web dinámico es distinto a los sistemas web sobre los cuales se realizaron los ajustes, debido a que se desarrolló por completo, para ello se utilizó un modelo iterativo y de igual forma se utilizó un modelo lineal ya que en cada iteración está contemplado el proceso de ciclo de vida del software y al final de cada iteración se contempla el avance y se mantiene una retroalimentación frecuente para resolver errores o aumentar requerimientos si el alcance lo permite, este proceso al ser lineal está basada en los siguientes pasos:

Requisitos

Es la etapa más crucial, en la que se realizan entrevistas, reuniones e intercambio de ideas para determinar los requisitos del proceso de desarrollo y el resultado final del sitio web. Los requisitos recopilados y documentados se analizan para posteriormente crear un diseño y decidir qué tareas se deben completar, resaltando prioridades para lograr el resultado final y se crea un plan de proyecto que incluye los entregables de cada tarea.

Para determinar los requisitos del proyecto y el resultado final deseado, se realizan entrevistas, reuniones e intercambio de ideas durante esta fase. Los requisitos del cliente se recopilan y documentan de manera sistemática durante estas interacciones.

Después de recopilar los requisitos, se realiza un análisis de los mismo. Esto implica examinar y comprender cada uno de los requisitos para asegurarse de que se cumplan todas las necesidades y expectativas de los usuarios.

Diseño y construcción

En esta etapa se incluye el proceso de implementación y la codificación. Durante la implementación es importante resaltar que implica comenzar a desarrollar el producto en función de los requisitos y el diseño. Durante la construcción se plantea buenas prácticas al momento de desarrollar el sitio web manteniendo una estructura sólida no solo en la arquitectura sino también en la estandarización en el código, dando un enfoque en las funcionalidades más importantes y relevantes en el sitio web. De esta manera se muestra una versión estable cumpliendo los requisitos previamente establecidos.

Fase de evaluación

Al final de la implementación del sitio web se coloca una fase de pruebas para el sistema al momento de comprobar la usabilidad y accesibilidad en el sitio web, de igual forma se evalúa la unión de los sistemas web de venta de productos agrícolas y promoción de sitios turísticos, en la página principal del sitio web dinámico. En esta etapa es fundamental la participación de los usuarios que proporcionaran una retroalimentación del sistema en base a la estructura que se mencionó anteriormente, y de esta forma lograr identificar componentes inestables o que afecten la usabilidad del sistema.

9. RESULTADOS

En esta parte del documento, se realizará el análisis y la ejecución de los diferentes procesos realizados tanto para los sistemas web existentes como para el sitio web dinámico.

Para los sistemas web existentes como es el caso del sitio web de turismo y el sitio web de productos agrícolas, se realizó un proceso de reingeniería, debido a que son sitios heredados, es decir que no fueron desarrollados desde cero. Se realizaron actividades como un análisis de los sistemas existentes, sus componentes e infraestructura, así como también sus debilidades, deficiencias y la visualización de cambios y mejoras permitiendo de esta manera la integración de estos sitios web.

Por otro lado, en el desarrollo del sitio web dinámico se realizaron procesos como el análisis de requisitos, el diseño e implementación para de esta manera garantizar un desarrollo del sitio web dinámico.

En esta etapa se analizaron y ejecutaron los distintos procesos que se llevaron a cabo tanto para cada uno de los sistemas web como para el sitio web dinámico. En cada uno de los procesos se detalla los pasos necesarios que se ejecutaron para alcanzar los resultados deseados. El proceso base que se llevó a cabo en los sistemas web de venta de productos agrícolas y promoción de sitios turísticos es el proceso de reingeniería, dado que los sistemas fueron heredados, es decir, no fueron desarrollados desde el inicio, y por otro para el desarrollo del sitio web dinámico se realizó en base al proceso SDLC, es decir el ciclo de vida del desarrollo de software, dado que este sitio web fue creado en su totalidad desde sus etapas iniciales. A continuación, se detallarán los procesos anteriormente mencionados:

CONSTRUCCIÓN DE SITIO WEB DINÁMICO

Para el desarrollo del sitio web principal se realizó en base a un enfoque flexible y dinámico, de tal modo que su principal ventaja la administración total de los contenidos sin necesidad de ingresar al código, ya que la configuración del sitio está diseñada para modificar o agregar

contenidos dentro de la interfaz. Los pasos que se llevaron a cabo para el desarrollo del sitio están detallados a continuación.

Requerimientos

Los sistemas web dinámicos son aquellos que presentan contenido y funcionalidades que cambian y se actualizan de forma constante, adaptándose a las necesidades y preferencias de los usuarios.

El proceso para levantar los requerimientos tiene como principal objetivo identificar y comprender las necesidades de los usuarios. Para este proceso se decidió llevar a cabo mediante la técnica del prototipado en la que se llevó a cabo con los siguientes pasos:

Definición de objetivos y requisitos.

Para empezar a desarrollar el sitio web dinámico, como primer punto es esencial comprender los objetivos que se planea llegar con el sitio web como: Promover a la organización, proporcionar información sobre la organización, dar acceso a los sistemas web de venta de productos agrícolas y promoción de sitios turísticos, sitio web totalmente administrable por personal de la organización. De esta manera se puede identificar el proceso guía que se llevara durante todo el proceso.

Una vez establecidos los objetivos es necesario identificar los requerimientos tanto como funcionales y no funcionales del sitio web, como se observa en la *Tabla 1* y *Tabla 2*.

Tabla 1

Requerimientos funcionales

| Requerimiento | Descripción |
|-------------------------|---------------------------------------|
| 1. Registro de usuarios | No permite registrar nuevos usuarios. |
| 2. Gestión de usuarios | Asignar permisos y restricciones. |

| | |
|-----------------------------------|---|
| 3. Administración de contenidos | Permitir la actualización y gestión de los contenidos. |
| 4. Personalización del contenido | Adaptar el contenido en función de las preferencias y comportamiento. |
| 5. Acceso a sistemas web | Permite el ingreso a los sistemas web. |
| 6. Personalización de pestañas | Incrementar varias pestañas dentro de la barra de navegación. |
| 7. Gestión de comentarios | Espacio para enviar comentarios sobre el sitio web. |
| 8. Integración con redes sociales | Enlaces que redirecciones a las redes sociales de la organización. |
| 9. Gestión de archivos multimedia | Carga y almacenamiento de imágenes. |

Como se puede observar en la *Tabla 1*, existen varios requisitos funcionales que logra alcanzar el sitio web. A continuación, se detallan la manera a través de la que se llevaron a cabo cada uno de los requerimientos.

Para el requerimiento 1 registro de usuarios, el sitio web está configurado de manera que no permita el registro de nuevos usuarios para administrar el sitio, a diferencia de los sitios web que tienes apartados para registrarse para realizar la acción que deseen dependiendo del sistema web en el que se encuentren, en este caso en el sitio web no está disponible esta opción ya que solo tendrán acceso los usuarios para poder administrar el sitio, y solo ellos tendrán el control de agregar más usuarios. De igual forma en este requisito está contemplado el lugar donde se encuentra el direccionamiento para la validación de las credenciales al momento de entrar al modo de administrador, el cual se encuentra fuera del

alcance inicial al usuario que visite el sitio, manteniendo el acceso oculto para el resto de los usuarios.

El requerimiento 2 gestión de usuarios, está relacionado con el requerimiento 1 ya que en este apartado el sitio web brinda el acceso a un único usuario que administre a todos los usuarios que puedan administrar el sitio web, y en el caso de que lo requieran podrán eliminarlos. El sitio web valida si es un usuario administrador o un super administrador, el cual como se menciona tiene el control de quien puede modificar el sitio. Existen solo dos roles, el administrador, tiene el control de modificar todos los campos configurados en el sitio web dinámico y por otro lado se encuentra el usuario super administrador el cual tiene acceso a los usuarios y a la modificación del sitio web.

El requerimiento 3 administración de contenido y el requerimiento 4 personalización del contenido están relacionados ya que el sitio web tiene la capacidad de gestionar los contenidos del sitio según las necesidades de los usuarios, el sitio web está configurado para que la actualización de los contenidos modificados se visualicen de manera instantánea por parte de los otros usuarios, al igual está contemplado el manejo de la creación de nuevos bloques de contenido adaptando el contenido que requiera en diferentes contextos. Dando el libre acceso a la edición y modificación del contenido a medida que el sitio web avance, es decir está diseñado de manera que pueda escalar a las necesidades de los usuarios.

El requerimiento 5 acceso a sistemas web, en este apartado está contemplado el redireccionamiento a cualquier sistema web que administren los usuarios, este requerimiento está solventado de manera que puedan redireccionar cualquier sitio que deseen ya que la dirección, nombre y descripción es administrable en base a las necesidades

de los usuarios. De este modo se mantiene una escalabilidad a todo el sitio ya que puede unir los sistemas web que se desarrollen posteriormente o redireccionar a sitios web que se desarrollaron de manera externa.

El requerimiento 6 gestión de pestañas, en este apartado se encuentra configurado para que los usuarios puedan agregar el número de pestañas que deseen, conectando de esta manera con los requerimientos anteriores, ya que pueden administrar su contenido de mejor manera. Lo que permite crear distintos entornos que ofrezcan los usuarios y administrar el contenido de cada una de las pestañas. No tiene un límite en cuanto a la creación de las pestañas, de este modo se da una libre modificación en el componente de la *barra de navegación*, la cual está configurada para adaptarse de manera dinámica a la creación de pestañas. De igual forma se pueden eliminar las pestañas creadas recientemente, debido a que las pestañas creadas por defecto en el sitio web dinámico no podrán ser eliminadas ya que están relacionadas con el cumplimiento de los otros requisitos y para mantener la estabilidad del sitio se optó por mantener deshabilitado la opción de eliminar, pero de igual forma como el resto de las pestañas, se pueden administrar los contenidos internos. Cabe recalcar que solo se pueden administrar los contenidos internos de cada componente ya que al momento de crear una nueva pestaña se creara con un diseño por defecto manteniendo una sola estructura en el sitio web.

El requerimiento 7 gestión de comentarios, para administrar los comentarios que generen los usuarios que visiten el sitio web se encuentra en una pestaña por defecto, la cual como se mencionó anteriormente es totalmente administrable en base a la información de la organización, de tal modo que se podrá modificar campos como dirección, número de teléfono, correo electrónico, página web personal, redes sociales y descripción. En este

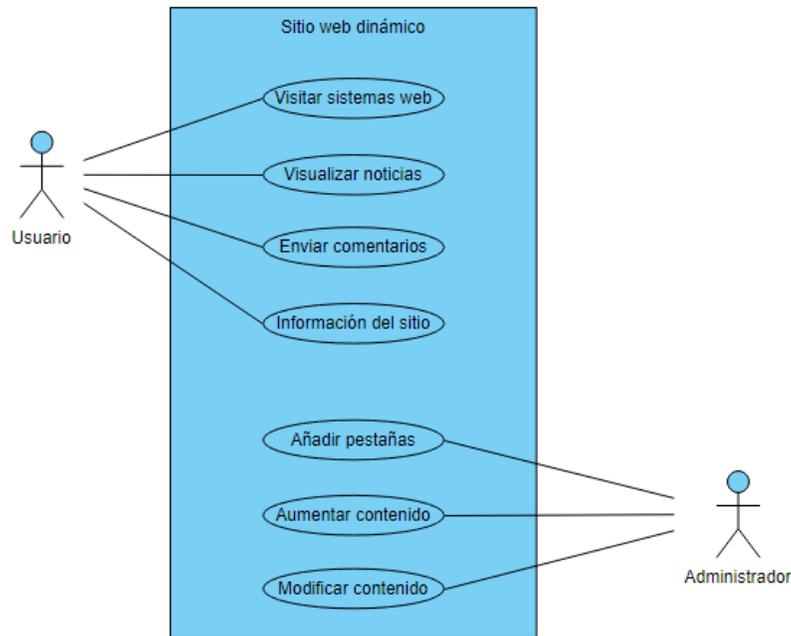
apartado también se encuentra un formulario que permite a los usuarios ingresar sus datos y enviar comentarios a la dirección de correo establecida. De este modo los usuarios podrán administrar el lugar donde recibirán todos los comentarios del sitio web.

El requerimiento 8 integración con redes sociales, está conectado con los componentes que corresponden al requisito 7 ya que la redirección a las redes sociales de la organización están conectadas con el campo de redes sociales en la pestaña que se menciona en el requisito 7, de este modo si el usuario decide modificar ese campo, la redirección a la red social modificada se podrá visualizar de manera inmediata por el resto de los otros usuario y no solo en la pestaña anteriormente mencionada, sino también en el componente del *Footer* el cual está conectado con el mismo componente para mantener al sitio web dinámico y escalable en el caso de que requieran aumentar redes sociales.

El último requerimiento gestión de archivos multimedia, está relacionado con todos los requerimientos previamente mencionados, ya que al modificar los contenidos no solo se puede modificar el título y la descripción, sino de igual forma se puede modificar la imagen que se mostrará, para ello la imagen que desee modificar se podrá obtener desde el explorador de archivos del usuario. El sitio web está configurado para gestionar todas las imágenes que se desee modificar y de este modo tener un mayor alcance con los servicios que se pretende brindar dando no solo información escrita sino también una información visual de lo que se trata cada uno de los contenidos.

Figura 4

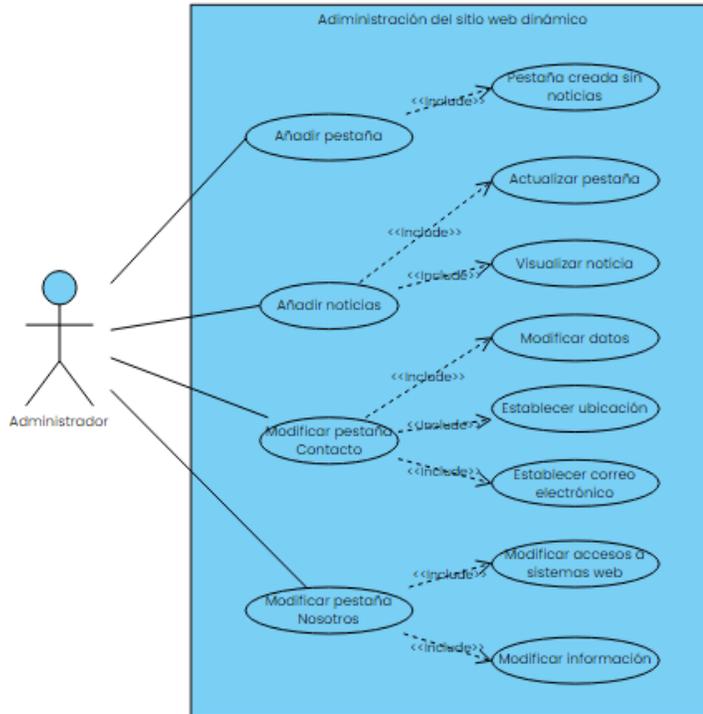
Diagrama caso de uso para requerimientos funcionales



Como se puede visualizar en la *Figura 4* es la secuencia de acciones que permite realizar el sitio web dinámico en base a los requerimientos funcionales previamente mencionados, donde se puede visualizar las distintas opciones que presentan tanto los usuarios como los administradores del sitio y de este modo tener una secuencia dinámica al momento de visualizar la información que se modifique en base a las necesidades de los usuarios o del entorno en el que se encuentren.

Figura 5

Diagrama caso de uso para administración de sitio web dinámico



Como se puede observar en la *Figura 5* es la secuencia de acciones que puede realizar el administrador del sitio en base a los requerimientos anteriormente mencionados, donde se puede visualizar un enfoque más profundo a las opciones que pueden realizar los usuarios si deciden modificar el sitio web ya que el principal objetivo es el control total del contenido que se pretende presentar y adaptarse si posteriormente decide cambiar algún componente y de este modo se actualice en base a las nuevas necesidades.

Tabla 2

Requerimientos no funcionales

| Requerimiento | Descripción |
|----------------------|--|
| 1. Rendimiento | El sitio web responde de manera rápida y eficiente. |
| 2. Seguridad | Protección de información sensible como claves de administradores encriptadas. |
| 3. Diseño responsivo | Sitio web adaptativo en distintos dispositivos. |
| 4. Escalabilidad | Diseño adaptativo a futuras necesidades y demandas. |
| 5. Usabilidad | Interfaz intuitiva y fácil de usar. |
| 6. Disponibilidad | Disponible de manera confiable. |
| 7. Mantenibilidad | Proporcionar herramientas y documentación adecuadas. |

Los requerimientos no funcionales que se pueden observar en la *Tabla 2* están basados en la norma ISO 9126 las cuales están adaptadas al sitio web dinámico. A continuación, se detallan como se ejecutaron cada uno de los requerimientos.

El requerimiento 1 rendimiento, el sitio web esta desarrollado de manera que, al realizar las peticiones al servidor, las respuestas sean rápidas y eficientes en términos de tiempo, de igual forma al momento de cargar cada uno de los componentes del sitio web no se tarda en el tiempo de respuesta. De igual forma para este requisito se contempla el manejo de un alto volumen de tráfico y conexiones simultáneas sin degradar el rendimiento. Una de las aplicaciones importantes en este requisito son técnicas de optimización, como el uso de caché, la comprensión de archivos y el manejo de la información, es decir una vez que los

contenidos sean obsoletos y se eliminen, se borrarán totalmente del servidor para así reducir la carga al momento de ingresar al sitio web.

El requerimiento 2 seguridad, corresponde al cumplimiento por parte del sitio web con estándares de seguridad y proteger la información confidencial. Este apartado corresponde al manejo de usuarios ya que al momento de que se registra un nuevo usuario administrador la contraseña se almacena de forma encriptada. De igual forma está contemplado el manejo al momento de autenticar y autorizar un usuario para que administre el sitio web, para ello como se mencionó anteriormente en los requerimientos funcionales, solo existe un solo usuario que se encarga de controlar a los usuarios administradores y las credenciales de ese usuario se entregaran exclusivamente a los usuarios y en el caso de que requieran otros acceso solo lo podrán accediendo de forma remota al servidor el cual tiene un acceso limitado para cumplir con este requerimiento.

El requerimiento 3 diseño responsivo, es uno de los más importantes ya que es la base para el resto de los requerimientos, debido a que el sitio web esta desarrollado de modo que se puede acceder de cualquier dispositivo inteligente, administrando todo el sitio de forma adaptativa al entorno en el que se encuentre.

El requerimiento 4 escalabilidad, está relacionado con el requerimiento 1 ya que, en parte corresponde a la capacidad que contiene el sitio web para manejar un aumento en el tráfico y la cantidad de usuarios sin afectar al rendimiento. Otro de los enfoques en la escalabilidad del sitio web dinámico es el alcance al momento de agregar contenidos a medida que el usuario lo requiera y de igual forma está en la capacidad de agregar recursos y capacidades adicionales según las necesidades y el entorno lo permita.

El requerimiento 5 usabilidad, corresponde a la interfaz de usuario del sitio web, ya que es una interfaz intuitiva y fácil de usar ya sea al momento de administrar el sitio o al momento de visitar el sitio. El sitio web ofrece un fácil acceso al momento de gestionar, actualizar y visualizar el contenido. Se proporciona una navegación clara y coherente, facilitando la búsqueda de información por parte de los usuarios visitantes y de igual forma se da una navegación de fácil entendimiento al momento de modificar los contenidos por parte de los usuarios administradores. Como se mencionó anteriormente está relacionado con el requerimiento 3 ya que una de las características principales para que el sitio web cumpla con el requerimiento de usabilidad, es el diseño adaptativo de la interfaz, logrando que se pueda acceder desde distintos dispositivos y resoluciones de pantalla.

El requerimiento 6 disponibilidad, corresponde a la confiabilidad al momento de acceder al sitio web, con un tiempo de actividad alto y un tiempo de inactividad mínimo. Al tener el acceso completo al servidor se puede controlar fallos mediante la detección de errores en tiempo real y manteniendo el sitio web disponible en todo momento.

El requerimiento 7 mantenibilidad, corresponde a la capacidad que ofrece el sitio web al momento de realizar mantenimiento, lo que implica que al momento de hacer cambios y actualizaciones se puedan realizar de manera eficiente, de modo que no cause interrupciones significativas en el sitio web. Para ellos se sigue un proceso de buenas prácticas de desarrollo de software, utilizando una estandarización en el código al igual que en la estructura. Una de las bases más importantes de la mantenibilidad es la entrega de documentación adecuadas para facilitar el mantenimiento y resolución de errores que ocurren posteriormente a la entrega hacia los usuarios.

Selección de tecnologías y frameworks

En base a la revisión de la literatura las tecnologías que mejor se adaptan a las necesidades del sitio web son Handlebars, como base de datos MongoDB y como framework Express. La principal razón para el uso de estas tecnologías es su alcance en el desarrollo de sistemas dinámicos y de igual forma cada una de las tecnologías son compatibles entre si dando una gran accesibilidad al momento de desarrollar el sitio web en base a las necesidades previamente requeridas.

Desarrollo de prototipo funcional

Una vez que los objetivos y requisitos están planteados, se procedió al desarrollo del prototipo funcional, resaltando las funcionalidades anteriormente mencionadas. Una vez listo el prototipo se presentó a los usuarios para la revisión del sitio y como en este caso para la implementación de nuevas funcionalidades y cambios en la interfaz de usuario. Dando un panorama más general para modificar lo solicitado y agregar funcionalidades requeridas exclusivamente para la organización.

Figura 6

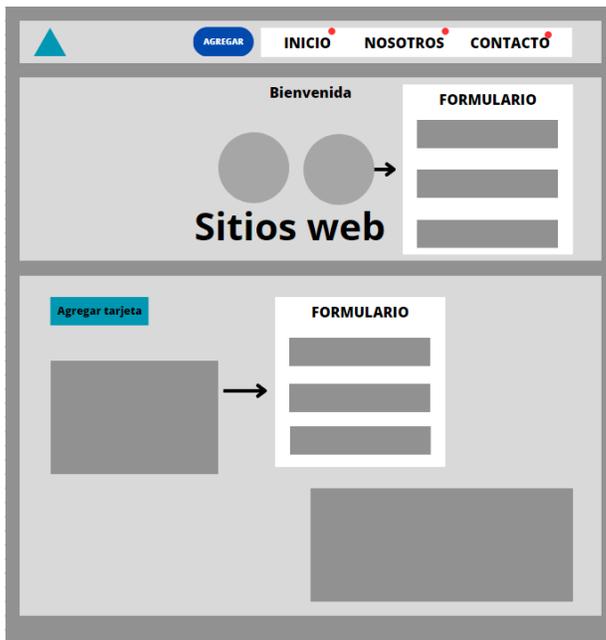
Prototipo funcional de vista de usuarios



Como se puede observar en la *Figura 6* el prototipo funcional que se realizó para la visualización por parte de los usuarios al ingresar al sitio web. Este prototipo tiene como principal objetivo la presentación de la estructura que se llevó a cabo para el desarrollo, dicha estructura está diseñada de modo que cumpla con los requisitos anteriormente mencionados tales que, exista un espacio para la visualización de los sistemas web y de igual manera exista un espacio para la visualización de la información en este caso se presenta en forma de noticias. Como se mencionó anteriormente este modelo es el que se presentará a los usuarios como primera vista y posteriormente si se decide crear más pestañas la estructura seguirá siendo la misma, omitiendo la presentación de los sistemas web ya que estos componentes son propios de la pestaña principal como se estableció en los requisitos.

Figura 7

Prototipo funcional de vista de administradores



Como se puede observar en la *Figura 7* el prototipo funcional que se realizó para la visualización por parte de los administradores al ingresar a la edición del sitio web. Este

prototipo tiene como principal objetivo la presentación de la estructura que se llevó a cabo para el desarrollo al momento de implementar herramientas dinámicas al sitio web, dicha estructura está diseñada de modo que cumpla con los requisitos anteriormente mencionados tales que, exista un espacio para la adición de nuevas pestañas en la *barra de navegación*, de igual forma existe un espacio donde le permite al administrador la visualización de los sistemas web y por ultimo existe un espacio para la edición de los contenidos de las información de la pestaña en la que se encuentre.

Diseño

Para realizar el diseño del sitio web dinámico se buscó comprender las necesidades del usuario, la funcionalidad del sitio y cómo ambos se entrelazan para proporcionar una excelente experiencia de usuario. Al ser dinámico este sitio web, el diseño se enfoca la interactividad y la capacidad de adaptación a las necesidades cambiantes de los usuarios y la visualización de la información en tiempo real al resto de usuarios. Para realizar el diseño del sitio web se llevó a cabo varios análisis en diferentes aspectos que se mencionan a continuación:

Análisis de las necesidades

Como se mencionó anteriormente primero se comprendido las necesidades y requerimientos que son necesarios en el sitio web, uno de los más importantes es que el sitio sea completamente administrable por parte de los administradores y que el resto de los usuarios puedan visualizar la información modificada al instante. Otra de las necesidades es el control para la administración de los contenidos, es decir no permite registrar usuarios y el acceso para editar la información no está accesible a primera vista cuando se ingresa al sitio web. Otra de las características requeridas es la creación de nuevas pestañas para presentar la información en distintos ámbitos, dentro de este apartado

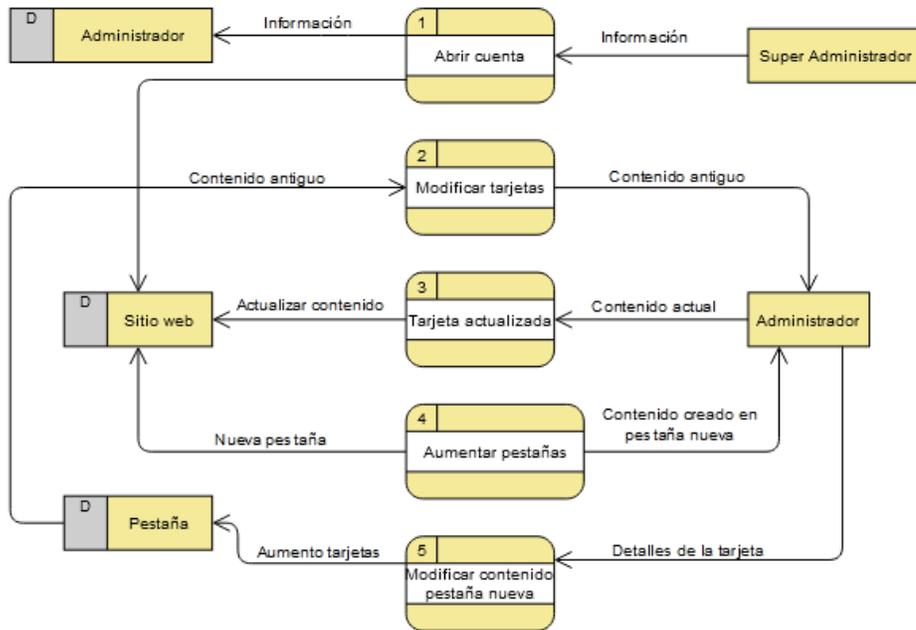
se vio la necesidad de mantener las pestañas de *Contacto y Nosotros* estáticas es decir que no podrán ser eliminadas pero los contenidos si son modificables. Estos aspectos anteriormente mencionados, son los más importantes para construir un diseño que pueda resolver cada una de las necesidades y requerimientos para todos los usuarios.

Diseño de la arquitectura de la información

Al ser un sitio web en el que se logra visualizar la información actualizada de forma inmediata requiere un diseño en la arquitectura para manejar la información, para ello como se mencionó anteriormente se utilizó una base de datos de tipo no estructural o de tipo NoSQL, para este caso se utilizó MongoDB, la principal razón para su uso es el dinamismo que ofrece al momento de gestión información en sitios web y de igual forma la escalabilidad que puede llegar a ofrecer dependiendo de las necesidades de los usuarios. Por ello la arquitectura de la información está desarrollada para que cada contenido este almacenado dentro de una base de datos que se logre adaptar a la modificación y actualización de forma inmediata sin perder tiempo de respuesta entre los clientes y el servidor. De igual forma las credenciales para la administración de los contenidos está en la misma arquitectura, dando una completa seguridad y confidencialidad al momento de administrar credenciales ya que son encriptadas al momento de ser registradas en la base de datos.

Figura 8

Diagrama de la arquitectura al modificar el sitio web dinámico



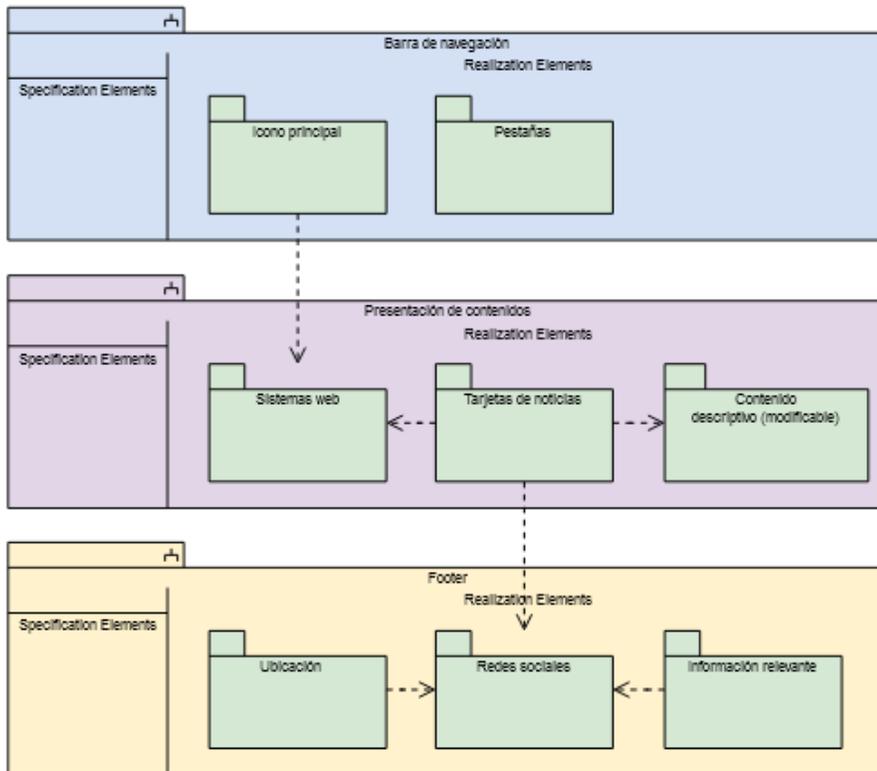
Como se puede observar en la *Figura 8* se muestra la estructura con la que se maneja la información al momento de modificar el sitio web dinámico, de este modo para que el contenido sea modificado tendrá que tener como primer paso la autenticación del usuario y validar si las credenciales ingresadas pertenecen a los administradores del sitio web, seguido de ello se podrá modificar el contenido que se desee, tales como el contenido de las tarjetas, el contenido de las pestañas y de igual forma el contenido de la redirección a los sistemas web. Dicha información actualizada se envía a la base de datos eliminando el contenido antiguo y reemplazando por el contenido actualizado y de igual forma la colección cambia a medida que se crea una nueva pestaña o se elimina una pestaña creada anteriormente.

Diseño visual y de interfaz de usuario

El diseño de la interfaz de usuario se desarrolló en base a las necesidades anteriormente mencionadas, ya que la interfaz debe ser accesible al momento de que se desee modificar los contenidos, para ello el diseño representa la gestión de la información por medio de etapas. Como primera etapa se encuentra la *barra de navegación* que puede ser modificada agregando o eliminando las pestañas que se requiera. Como segunda etapa se muestra el contenido principal de cada pestaña, por ejemplo, en la pestaña de *inicio* se muestra la bienvenida. La tercera etapa en la pestaña de *inicio* muestra la conexión con los sistemas web de venta de productos agrícolas y promoción de sitios turísticos y si en algún futuro aumenta otro sistema se le agregaría en la misma etapa. En la misma etapa a continuación se encuentra la gestión de la información en forma de tarjetas mostrando el contenido descriptivo de cada pestaña. Como última etapa se muestra el *Footer* presentando información relevante como la dirección y la promoción de las redes sociales de la organización.

Figura 9

Diagrama de etapas en la interfaz de usuario.



Como se puede observar en la *Figura 9*, se presenta el diagrama de paquetes que están incluidos en la interfaz ya que, como se explicó anteriormente se lleva a cabo en base a etapas donde se pueden visualizar los componentes que son completamente administrables por parte de los usuarios.

Construcción

La construcción de un sitio web dinámico implica el desarrollo de un sitio web interactivo que ofrece una experiencia fluida con todos los usuarios, permitiendo gestionar la información y visualizando de forma inmediata. Una vez establecido el diseño se procedió al desarrollo de una versión estable del sitio web cubriendo todos los aspectos relevantes que se mencionaron

anteriormente. Para realizar la construcción se llevó a cabo en base a diferentes aspectos que se mencionan a continuación:

Estandarización arquitectónica

Dentro de un sitio web dinámico, la estandarización arquitectónica es importante para garantizar la escalabilidad y la mantenibilidad del sitio web. Para ello es importante analizar la estructura y organización de los componentes que interactúan entre sí para ofrecer una experiencia personalizada por parte de los usuarios administradores y una experiencia interactiva por parte del resto de los usuarios. Debido a que el sitio web dinámico está desarrollado en base a otros sistemas web, se llevó a cabo un segundo análisis para comprender la arquitectura de los otros sistemas web y lograr adaptarlo a las necesidades que presenta este sitio. Para ello se estableció un patrón arquitectónico basado en la construcción de los otros sistemas web dando una arquitectura estándar tanto para el sitio web como para los sistemas web.

Figura 10

Estructura de carpetas sitio web dinámico

```
Sitio web dinámico
├─ src
│  ├─ models
│  ├─ passport
│  ├─ public
│  │  ├─ assets
│  │  │  └─ img
│  │  │     └─ about
│  │  │     └─ nuevas
│  │  │     └─ portfolio
│  │  └─ storage
│  └─ vendor
├─ css
├─ js
├─ routes
├─ views
│  └─ layouts
└─ partials
```

Como se puede observar en la *Figura 6*, la estructura se encuentra estandarizada y busca seguir una organización basada en las dos aplicaciones ya creadas. En la carpeta *models* se encontrará los archivos esquema que serán utilizados por la base de datos MongoDB para crear la estructura de las colecciones. La carpeta *passport* tendrá un archivo que contendrá las funciones de autenticación del usuario. En *public* existirán carpetas anidadas las cuales contendrán configuraciones de lenguaje Css, Java Script y carpetas donde se almacenarán las imágenes del proyecto. El contenido de la carpeta *routes* es un archivo donde se realizó la configuración y el manejo de las peticiones que recibe el servidor. En *views* se mostrarán los archivos Html que serán utilizados por un sistema de plantillas Handlebars el cual proporcionara la parte dinámica por medio de variables de forma directa en el Html.

Implementación de funcionalidades dinámicas

El sitio web está desarrollado para que sea completamente administrable, es decir que usuarios que tengan acceso a las credenciales de administración pueden realizar cambios como, modificación de la *barra de Navegación*, en este módulo el administrador podrá agregar opciones al menú para poder diferenciar o segmentar su información de manera más ordenada, los contenidos al seleccionarlos, permitirán actualizar información o inclusive agregar más información, esta funcionalidad dinámica se otorgó a todo el sitio el cual ayudará a que el administrador tenga el acceso total de todo lo que se muestra en el sitio web dinámico. Naturalmente, una de las opciones que se pueden realizar es agregar o eliminar, esta acción corresponde a las noticias como a las pestañas, dando una elección libre al momento de crear noticias, como al momento de crear una nueva pestaña. La pestaña será creada en base a la principal manteniendo la misma estructura para que los contenidos sean totalmente administrables cuando sean necesarios. Y de esta forma al

eliminar, una nueva pestaña en la *barra de navegación*, el sitio web dinámico borrará el archivo creado y la colección en la base de datos, para que no se guarde información innecesaria.

Se pueden apreciar las principales funciones de la parte dinámica, flexible y escalable del sitio web en las siguientes imágenes:

Figura 11

Primera parte de función para agregar tarjetas

```
router.post('/tasks/add1', async (req, res) => {  
  
  const path = req.body.path;  
  
  try {  
  
    const collectionName = path; // Obtén el nombre de la colección de la solicitud  
    // Obtén el modelo de la colección seleccionada  
    const existingModel = mongoose.models[collectionName];  
    const Task = existingModel || getCollectionModel(collectionName);  
  
    const imagenUrl = req.body.image; // Obtén la URL de la imagen desde req.body.image  
    const title = req.body.title;  
    const description = req.body.description;  
    const index = req.body.index;  
  }  
  
});
```

La primera parte de la función corresponde a la ruta específica que realiza una petición *Post* al momento de agregar una tarea. Seguido de ello se identifica la colección en la que se guardará la tarea. Se obtiene el modelo correspondiente, verificando si ya existe un modelo registrado para la colección especificada, en el caso que no se encuentre creada invoca a la función para obtener el modelo correspondiente. Esta lógica permite reutilizar modelos existentes y evitar crear duplicados. Al finalizar la primera parte se inicializa las variables a partir de los campos correspondientes los cuales se utilizarán para crear la nueva instancia de la tarea.

Figura 12

Segunda parte de función para agregar tarjetas

```
// Verificar que las variables sean cadenas de texto
if (typeof title !== 'string' || typeof description !== 'string' || typeof index !== 'string') {
  throw new Error('Los datos proporcionados no son válidos');
}

const task = new Task({
  title: title.toLowerCase(),
  description: description.toLowerCase(),
  index: index.toLowerCase(),
  image: {
    nombre: imagenUrl, // Utiliza la URL de la imagen
    ruta: '/assets/img/portfolio/cabin.png', // Ruta a la imagen por defecto
    tipo: 'image/jpeg' // Tipo de contenido por defecto para la imagen
  }
});
```

En la segunda parte se realiza una verificación para asegurarse de que las variables previamente inicializadas sean de tipo cadena de texto. En el caso que alguna de ellas no es de este tipo, envía un mensaje de error. Esta verificación garantiza que los datos cumplan con los requisitos esperados antes de crear la tarea. Una vez verificado se crea una nueva instancia, es decir una nueva tarea utilizando los valores obtenidos previamente, estos valores se asignan a las propiedades correspondientes de la tarea.

Figura 13

Tercera y última parte de función para agregar tarjetas

```
const savedTask = await task.save();

const taskIndex = savedTask.index;

res.json({
  taskIndex
});
} catch (error) {
  console.log(error);
  res.status(500).send('Ocurrió un error al crear la tarea');
}
```

La nueva tarea creada se guarda en la base de datos, una vez que valide que se guardó se obtiene el índice de la tarea guardada y se le asigna a la variable. Esto permite obtener una referencia al índice de la tarea para su posterior uso o respuesta. Seguido de ello se envía una respuesta JSON al cliente con el índice de la tarea guardada. Al estructurar la respuesta como un objeto, se facilita el acceso a ese valor específico en el lado del cliente. Por último, en el caso de que exista un error al momento de crear una tarea primero se envía un mensaje de error en la consola para facilitar la depuración y seguido se envía una respuesta de error al cliente, notificando que se produjo una falla en el servidor.

Agregar tarjetas permite que la información se mantenga organizada dentro de cada página, con esta funcionalidad buscamos la creación de tarjetas según el usuario las necesite.

Figura 14

Primera parte de función para agregar opciones al menú de navegación

```
$('.navbar-nav').prepend(botonAdministrador)

$('.navbar-nav').on('click', '#agregarBtn', function (event) {
  event.preventDefault();

  Swal.fire({
    title: 'Ingresa un nuevo título',
    input: 'text',
    showCancelButton: true,
    confirmButtonText: 'Guardar',
    cancelButtonText: 'Cancelar',
    allowOutsideClick: false,
    inputValidator: (value) => {
      if (!value) {
        return 'Debes ingresar un título';
      }
    }
  })
})
```

Primero se inicializa el contenedor de la *barra de navegación*, luego se asigna un controlador de eventos al botón de *agregar*, evitando el comportamiento predeterminado al dar clic. Después, se utiliza una biblioteca que permite mostrar un cuadro de diálogo interactivo, en el cual contiene campos de entrada de texto donde el usuario puede ingresar el nuevo título de la opción. De igual forma se muestran opciones de guardar o cancelar la acción. Al final se valida que el campo de título este completo para poder continuar y en el caso que no lo esté muestra un mensaje de aviso para que complete ese campo.

Figura 15

Segunda parte de función para agregar opciones al menú de navegación

```
}).then((result) => {  
  if (result.isConfirmed) {  
    var nuevoTitulo = result.value;  
    // Crea el archivo con el nombre del nuevo título y el contenido del index.hbs  
    var archivoNombre = nuevoTitulo + '.hbs';  
    var nombreBDD = nuevoTitulo + 'Noticias';  
    var archivoContenido = `Contenido del archivo ${nuevoTitulo}:  
    \n\n${archivoContenido}`;  
    imagen = '/assets/img/portfolio/cabin.png';  
    $.ajax({  
      url: '/api/crear',  
      type: 'POST',  
      data: { title: archivoNombre, description: archivoContenido, nombre: nombreBDD, index: "left", image  
      success: function (response) {  
        console.log('Colección creada exitosamente:', response);  
      }  
    });  
  }  
});
```

A continuación, se maneja la respuesta del cuadro de diálogo modal. Si el usuario confirma la acción, se asigna el valor del campo de entrada a la variable correspondiente, seguido se crea un nuevo archivo con el valor creado y concatenado la extensión del archivo, de este modo se establece el contenido del archivo con el nuevo título y una cadena adicional. Al final se crea la conexión para enviar los datos proporcionados validando si se creó correctamente con un mensaje de aviso junto con la respuesta recibida.

Figura 16

Tercera y última parte de función para agregar opciones al menú de navegación

```
$.ajax({
  url: '/crear-archivo',
  type: 'POST',
  data: { nombre: archivoNombre, contenido: archivoContenido },
  success: function (response) {

    console.log('Archivo creado exitosamente:', response);
    // Crea una nueva tarjeta con el título ingresado
    var nuevaTarjeta =
      '<div class="card">' +
      '<div class="card-body">' +
      '<h5 class="card-title">' +
      nuevoTitulo +
      '</h5>' +
      '</div>' +
      '</div>';
    // Agrega la nueva tarjeta al contenedor deseado
    // (por ejemplo, al final del cuerpo del documento)
    $('body').append(nuevaTarjeta);
    location.reload();
  },
  error: function (error) {
    console.log('Error al crear el archivo:', error);
  }
});
},
error: function (error) {
  console.log('Error al crear la colección:', error);
}
```

Por último, se realiza una solicitud *Post* a la ruta especificada, los datos que se envían en la solicitud incluyen en nombre del archivo y su contenido. En caso de que no exista ningún error se muestra un mensaje en la consola indicando que el archivo se creó exitosamente junto con la respuesta recibida. De igual forma, se crea una nueva tarjeta con el título ingresado y se agrega al contenedor seleccionado en el sitio web. Finalmente valida si existe errores mediante mensajes enviados a la consola.

Esta función es la petición que se envía del lado del cliente hacia el servidor, haciendo el transporte de la información necesaria para que se pueda crear otra opción en la *barra de navegación* y permitiendo la escalabilidad.

Estas nuevas opciones tendrán una estructura de tarjetas como la función para crear tarjetas, de esta forma se garantiza que el usuario elija la estructura final de la página web dinámica,

ya que por este medio permite la creación de archivos y colecciones independientes, de igual manera esta función tiene una opuesta, la cual permite eliminar las opciones creadas. Con el objetivo de optimizar los recursos y garantizar la fluidez de las aplicaciones, cuando se elimine la opción creada esta a su vez eliminara el archivo y la colección creada específicamente para esa opción.

SISTEMA WEB PARA VENTA DE PRODUCTOS AGRÍCOLAS

Al ser este un sistema web heredado se realizó el análisis en base a la revisión de la literatura sobre qué proceso base es el que mejor se acopla con el proceso que se realizó en dicho sistema. Como se muestra en la *Figura 1* es el proceso con el que se lleva a cabo una reingeniería de software en este caso a un sistema web, para llevar a cabo cada fase del procedimiento se ejecutaron los siguientes pasos:

Análisis de Inventario

Una vez que se recibió el sistema web como primer paso, es el análisis completo del sistema como: la información del sistema, análisis de los componentes fundamentales y los componentes que tenían problemas y no estaban funcionando. El sistema está orientado a emprendedores que requieran dar a conocer o vender sus productos agrícolas, los componentes principales y el resultado de la revisión se detalla de la siguiente manera:

Tabla 3

Análisis de inventario del sistema web.

| Componente | Resultados de la revisión |
|----------------------------|--|
| Interfaz de usuario | Interfaz poco amigable dando un cambio brusco al momento de cambiar entre aplicaciones |

| | |
|-------------------------------|---|
| Base de datos | Base de datos MySQL almacena en la nube dando problemas en varios componentes. |
| Agregar productos | Al momento de agregar productos el sistema dejaba de funcionar. |
| Ventana Login | Registro con interfaz poco amigable, un solo registro para usuario normal o administrador. |
| Registro nuevo usuario | Componente funcional parcialmente debido a conexión de base de datos. Campos del registro amigables con el usuario. |
| Panel de administrador | Panel no accesible desde la interfaz, ingreso únicamente si el usuario esta registrado como tal. |
| Pasarela de pagos | Componente sin funcionamiento en su totalidad. |
| Barra de navegación | Pestañas que no aportaban información dentro del sistema. |
| Carrito de compras | Componente funcional en su totalidad. |
| Visualizar ventas | Componente funcional parcialmente debido a que está indexado a la pasarela de pagos |

Como se puede observar en la *Tabla 3* los complementos que están desarrollados dentro del sistema web dando un panorama más general de donde enfocarse a mejorar o en algunos a reconstruir el componente. De igual forma se analizan los principales errores que afectan a toda la funcionalidad del sistema como la estructura de la base de datos. Dentro de dicho análisis se revisó la complejidad que conllevaba construir una base de datos de forma local en el que se tiene el control con cada componente del sistema. Como último paso en la revisión de los componentes se analizó la

eliminación de ciertos componentes que no aportaban con el sistema. Dentro de la revisión del sistema se contempló el análisis del código y se observó una correcta estructuración de los componentes y un patrón arquitectónico sencillo de comprenderlo y utilización de tecnologías como Handlebars y Express. Esto resulto ser importante para tener una base que servirá como enfoque a los otros sistemas incluyendo el sitio web principal.

Tabla 4

Tabla de componentes eliminados del sistema web de venta de productos agrícolas

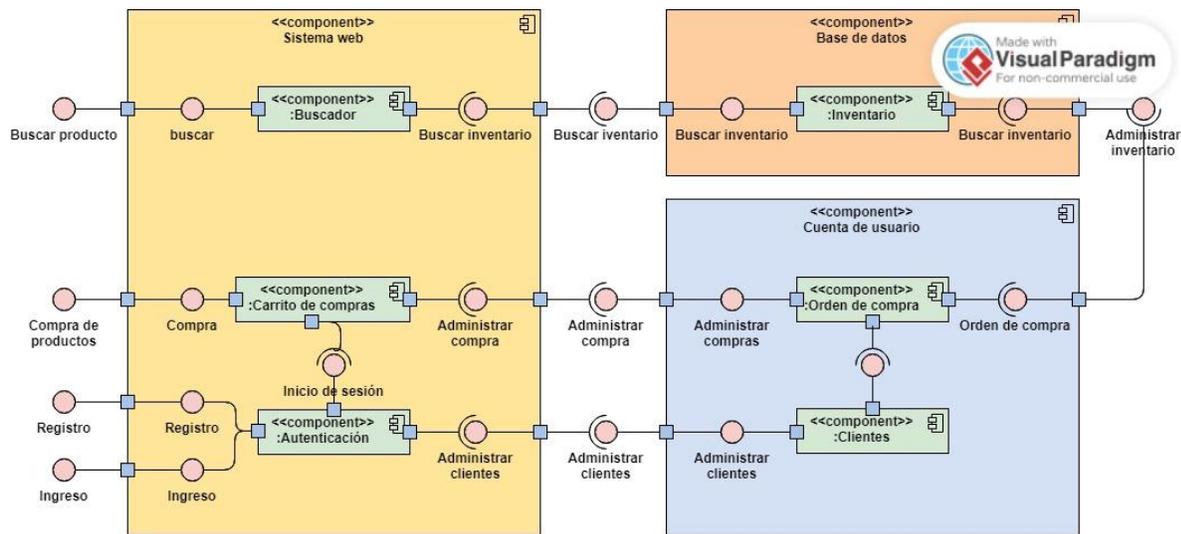
| Componente eliminado | Razón de eliminación |
|-----------------------------|--|
| Pestaña información | Contenido presentado en el sitio web principal |
| Pestaña registro usuario | Redundancia debido a que este componente estaba incluido en otra pestaña |
| Pestaña de búsqueda | Ubicación de componente donde no presentaba relevancia. |

Como se puede visualizar en la *Tabla 4* se evidencia el listado de componentes que no aportaban valor al sistema y por tales motivos se decidió removerlos, debido a que cada uno de los componentes presentaban información ya existente en otros componentes ya sea en el mismo sistema web o en el sitio web dinámico.

Para visualizar de mejor manera como interacciona cada uno de los componentes dentro del sistema web final se presenta el siguiente diagrama:

Figura 17

Diagrama de componentes de sistema web para venta de productos agrícolas.



En el diagrama se puede observar cómo está administrado cada uno de los componentes. Los componentes que se detallan en el diagrama son los componentes que presentaron problemas posteriormente por ello es importante destacar cada uno como influye en el sistema. Dentro del primer bloque que corresponde al sistema web se contempla los componentes que están incluidos en la interfaz de usuario es decir los componentes que tienen interacción directa con los usuarios del sistema web, dentro de estos componentes uno de los principales es el buscador el cual permite tener una mayor accesibilidad en el sistema al momento de hacer una búsqueda del producto que desee mostrando todos los resultados que encuentre, en el cual se podrá evidenciar la procedencia del producto y si se encuentra en stock. Otro de los componentes que está incluido en la interfaz de usuario es el *carrito de compras* y se podrá visualizar cuando el usuario agregue a su compra la cantidad de productos que desee, de igual forma en este componente se puede visualizar el precio subtotal y el total de cada compra que se llevará a cabo la orden de compra posteriormente para la comprobación del método de pago. Como último componente perteneciente a la interfaz de usuario es la autenticación de usuario, se logra validar si un usuario ya está registrado o en el

caso que no lo esté se redireccionará al usuario a una pestaña que le permita registrar mediante un formulario validando campos configurados inicialmente.

El segundo bloque corresponde a la base de datos en el que el principal componente es la administración del *inventario*, es decir cómo se administra la cantidad de producto que existe actualmente para la venta. Esta administración está a cargo de los usuarios que hayan registrado anteriormente productos al sistema, ya que una vez que el producto esté agregado podrán tener el control total en cuanto a la cantidad que presente, ya que el nombre, foto, descripción y lugar del producto se lo configura al momento de agregarlo y queda almacenado dentro de la base de datos que a medida que el usuario modifique la cantidad se verá reflejado para el resto de usuarios que ingresen al sistema en la pestaña de visualización de los productos. De igual forma al concluir una orden de compra se verá reflejado en el componente del inventario debido que a medida que la cantidad de cada uno de los productos sean vendidos, se modificará automáticamente en el inventario validando el usuario que agrego y la cantidad que fue vendido y por parte del usuario que administra los productos podrá visualizar la cantidad de cada uno de los productos. Debido a ello la administración de las cuentas de los usuarios está conectado a este componente, validando si el usuario está realizando una venta de producto o si está realizando una compra de productos, uniendo de esta manera a las órdenes de compra anteriormente mencionadas.

Como último bloque es la administración de cuentas de los usuarios, como se mencionó anteriormente este bloque se complementa con el bloque de la base de datos debido a que está relacionado con la administración de las órdenes de compra y con el registro de usuarios. Dentro de este bloque están incluidos los componentes principales para que la conexión con la base de datos sea correcta y administre los productos manteniendo el sistema de forma secuencial. Uno de los componentes es las órdenes de compra en el que se lleva a cabo la administración de las

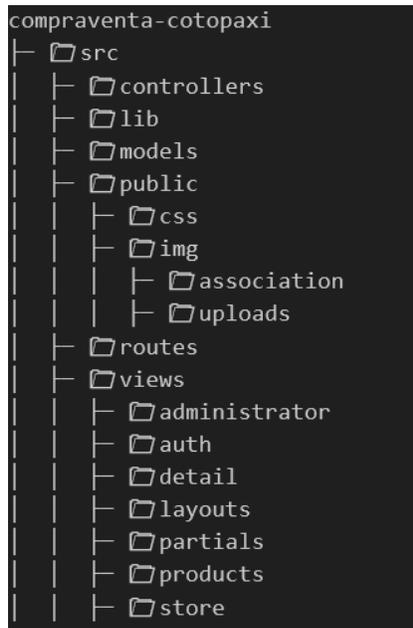
compras que realizan los usuarios, es decir una vez que se registra el usuario y decide comprar un producto se registra la compra de esa sesión y si decide realizar otro pedido un día diferente se crea una nueva orden de compra que posteriormente en cada sesión se validará si se realiza o no la compra y actualizando de esta manera la base de datos. De igual forma en el siguiente componente de clientes, es el encargado de administrar los clientes una vez que se haya autenticado, y a medida que desee realizar una actividad en el sistema web se conecta con la base de datos para validar el proceso que desee realizar.

Reestructuración

Como se mencionó anteriormente en este sistema la reestructuración del código no fue necesario ya que la estructura establecida es comprensible al momento de realizar modificaciones o aumentar componentes. En este caso se realizó una reestructuración de la interfaz de usuario, realizando un cambio en la *barra de navegación* específicamente en la barra de búsqueda de productos, se decidió colocarlo en el centro de la página para que sea relevante al momento de abrir el sistema web. Dentro del mismo componente se realizó un cambio en el orden de pestañas dando una estructura sólida para todos los sistemas. Por último, se eliminó componentes como la pestaña de información ya que ese componente está contemplado dentro del sitio web principal y la pestaña de registro de usuario ya que este apartado se encontraba en otro componente.

Figura 18

Estructura de carpetas sistema web de venta de productos agrícolas



Visualizando la *Figura 10* se puede evidenciar, que la estructura de este sitio web es muy parecido a la del sitio web dinámico, por lo tanto, no es necesario realizar una reestructuración ya que es muy parecido al estándar.

Ingeniería inversa

Para la construcción de la base de forma local se aplicó ingeniería inversa ya que, como se mencionó en el análisis de inventario el sistema web almacenaba toda su información en una base de datos en la nube dando acceso limitado para realizar cambios o configuraciones, de igual forma tener la base de datos de este modo dificultaba al momento de resolver los problemas que presentaba el sistema. Para la reconstrucción de la base de datos se partió del análisis de la documentación tomando en cuenta las tablas y relaciones que tenía cada una de ellas para configurarlas correctamente ya que uno de los problemas eran las relaciones que existían anteriormente. Una vez que se creó la base de datos se procedió a realizar la conexión. De esta

forma se logró comprobar que la razón principal de que los componentes no funcionaban era la configuración errónea dentro de la base de datos antigua, ya que al realizar la reconstrucción todos los componentes funcionaban correctamente y de esta forma se logró obtener el control de la pasarela de pagos.

Ingeniería Directa

Como último paso dentro de este sistema web se realizó el rediseño de la interfaz de usuario, utilizando fuentes y estilos que complementaban a los otros sistemas dando un cambio visual más suave al momento de cambiar entre sistemas. Dentro del rediseño también se optó por la modificación del *Footer* dando una mejor estructuración del espacio al igual que la *barra de navegación*. Dentro del cambio visual también se decidió colocar colores en base a los otros sistemas.

SISTEMA WEB PARA LA PROMOCIÓN DE SITIOS TURÍSTICOS

Como se mencionó anterior mente cada uno de los resultados que se obtuvieron de los sistemas web se desarrollaron en base al proceso de reingeniería, al igual que el anterior sistema de venta de productos agrícolas los pasos que se llevaron a cabo son los siguientes:

Análisis de Inventario

Una vez que se recibió el sistema web como primer paso, es el análisis completo del sistema como: la información del sistema, análisis de los componentes fundamentales y los componentes que tenían problemas y no estaban funcionando. El sistema está orientado a usuarios que permanecen a la organización que deseen difundir su lugar turístico o servicio ya sea restaurantes, iglesias hasta servicios de taxis. Los componentes más principales y sus características se detallan de la siguiente forma:

Tabla 5

Análisis de inventario del sistema web.

| Componente | Observaciones |
|---|---|
| Interfaz de usuario | Interfaz amigable y base para modificar los otros sistemas. |
| Base de datos | Base de datos MySQL almacenada de forma local. |
| Agregar sitios turísticos o servicios | Validaciones correctas en cada componente, pero no permite registrar el sitio. |
| Visualizar sitios turísticos O servicios | Componente funcional parcialmente ya que no permitía el acceso a la API de los mapas. |
| Ventanas de <i>Login</i> | Ventanas para un usuario normal y para un usuario administrador. |
| Registro de nuevo usuario | Componente funcional en su totalidad. Validación mediante correo electrónico. |
| Panel de administración | Panel accesible desde interfaz. |
| Barra de navegación | Pestañas que no aportaban información dentro del sistema. |
| Visualizar estadísticas | Componente funcional en su totalidad. |
| <i>Chatbot</i> | Desarrollo de nuevo componente. |

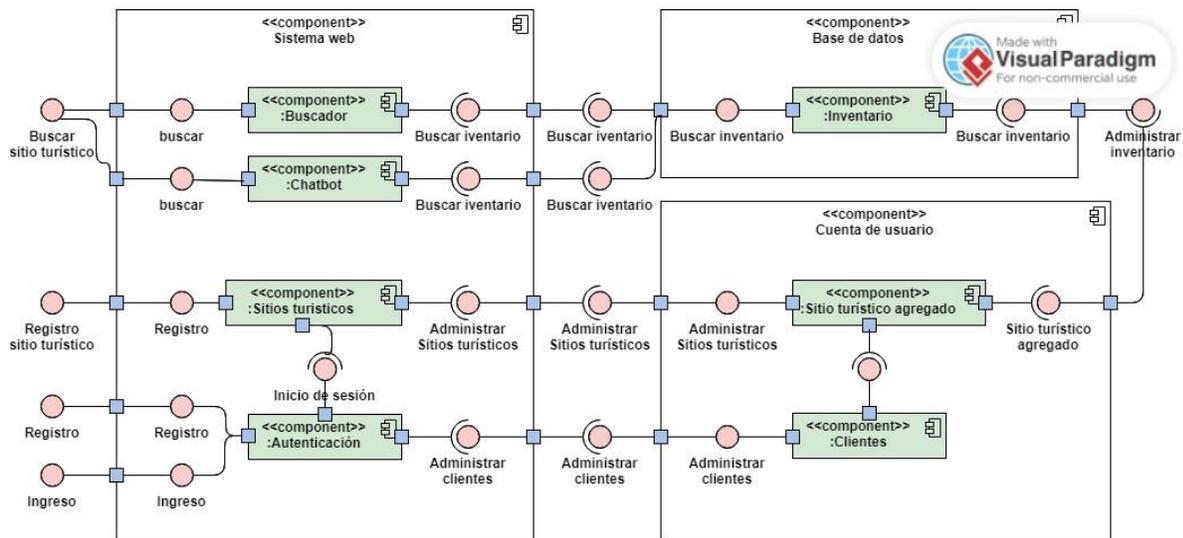
Como se puede observar en la *Tabla 5* los complementos que están desarrollados dentro del sistema web dando un panorama más general de donde enfocarse a mejorar, en algunos casos a reconstruir el componente y como en este caso a construir un nuevo componente. De igual forma se analizan

los principales errores que afectan a toda la funcionalidad del sistema como el registro de un nuevo sitio turístico o servicio y la visualización desde el mapa. Dentro de dicho análisis se revisó la complejidad de realizar una nueva función para crear el registro debido a que no existía una conexión alguna con el registro y la base de datos, de igual forma se analizó realizar una nueva conexión para visualizar los mapas tanto como en el registro como en la visualización desde el usuario. Como último paso en la revisión de los componentes se analizó la eliminación de componente que no aportaban con el sistema debido a que la información que presentaba estaba contemplada en otro componente. Dentro de la revisión del sistema se contempló el análisis del código, se observó un patrón arquitectónico similar al otro sistema, pero con una estructura no tan sencilla al momento de comprenderlo ya que se utilizaron tecnologías como JavaScript y Express. Esto resultó ser importante para comprender la lógica al momento de implementar las nuevas funcionalidades y arreglar las funcionalidades ya existentes.

Para visualizar de mejor manera como interacciona cada uno de los componentes dentro del sistema web se presenta el siguiente diagrama:

Figura 19

Diagrama de componentes de sistema web para promoción de sitios turísticos.



En el diagrama se puede observar cómo está administrado cada uno de los componentes. Los componentes que se detallan en el diagrama son los componentes que presentaron problemas posteriormente por ello es importante destacar cada uno como influye en el sistema. Dentro del primer bloque que corresponde al sistema web se contempla los componentes que están incluidos en la interfaz de usuario es decir los componentes que tienen interacción directa con los usuarios del sistema web, dentro de estos componentes uno de los principales es la implementación del programa de Chatbot el cual está relacionado con el *Buscador*, en el cual ambos componentes permiten tener una mayor accesibilidad en el sistema al momento de hacer una búsqueda del sitio turístico que desee el usuario mostrando todos los resultados que encuentre en base a las necesidades que el usuario requiera, en este apartado se podrá evidenciar las actividades que se pueden realizar, la ubicación y si existen ofertas actualmente. Otro de los componentes que está incluido en la interfaz de usuario es el registro de los sitios turísticos en el que cada usuario registrado podrá agregar el sitio turístico que desee, añadiendo las actividades que se adapten mejor y colocando campos importantes como nombre, ubicación y en el caso que lo requiera aumentar

ofertas para los turistas. Por último, el componente perteneciente a la interfaz de usuario es la autenticación de usuario, se logra validar si un usuario ya está registrado o en el caso que no lo esté se redireccionara al usuario a una pestaña que le permita registrar mediante un formulario validando campos configurados inicialmente.

El segundo bloque corresponde a la base de datos, el principal componente es la administración del inventario de los sitios turísticos, es decir cómo se administra los sitios turísticos agregados dentro de la base de datos. Esta administración está a cargo por los usuarios que hayan registrado anteriormente sitios turísticos al sistema, ya que una vez que el sitio turístico este agregado podrán tener el control total en cuanto a al nombre, foto, descripción y ubicación, a medida que el usuario modifique cualquiera de estos campos se verá reflejado para el resto de los usuarios que ingresen al sistema en la pestaña de visualización de los sitios turísticos. Debido a ello la administración de las cuentas de los usuarios está conectado a este componente, validando si el usuario está realizando un registro de un nuevo sitio turístico o si está realizando una búsqueda de un sitio turístico en particular, la administración de todos los sitios con la base de datos.

Como último bloque es la administración de cuentas de los usuarios, como se mencionó anteriormente este bloque se complementa con el bloque de la base de datos debido a que está relacionado con la administración del registro de los sitios turísticos y con el registro de usuarios. Dentro de este bloque están incluidos los componentes principales para que la conexión con la base de datos sea correcta y administre los sitios, manteniendo el sistema de forma secuencial. Uno de los componentes es la agregación de un nuevo sitio turístico, en el cual se lleva a cabo la conexión entre el usuario autenticado y el sitio turístico agregado por parte de ese usuario, es decir una vez que se registra el usuario y decide agregar un sitio turístico se registra la cuenta de esa sesión y si decide agregar otro sitio un día diferente se registra en al mismo usuario. De esta manera

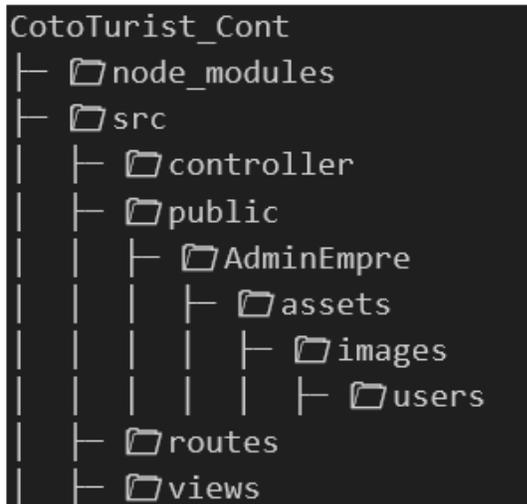
posteriormente podrá tener el control el usuario en la pestaña de administrador para observar las visitas que tuvo su lugar turístico o si tiene algún comentario al respecto. De igual forma en el siguiente componente de clientes, es el encargado de administrar los clientes una vez que se haya autenticado, y a medida que desee realizar una actividad en el sistema web se conecta con la base de datos para validar el proceso que desee realizar.

Reestructuración

Como se mencionó anteriormente en este sistema la reestructuración del código en ciertos componentes fue necesaria debido a funciones inexistentes y conexiones sin funcionamiento. Como primer punto se estableció una nueva conexión al momento al momento de consumir los servicios de la API para visualizar los mapas, para ello se analizó la antigua conexión, dando como resultado que el error se debía por una mala estructuración del código al momento de realizar la consulta, para ello la solución fue modificar la estructura de la consulta y de esta forma se logró visualizar los mapas. Dentro del componente de *barra de navegación* se realizó un cambio en el orden de pestañas dando una estructura sólida en base a los otros sistemas. Por último, se eliminó componentes como la pestaña de *nosotros* ya que ese componente está contemplado dentro del sitio web principal.

Figura 20

Estructura de carpetas sistema web de promoción de sitios turísticos



La estructura, esta estandarizada y busca seguir una organización basada en las otras arquitecturas.

Ingeniería inversa

Como proceso de ingeniería inversa fue el análisis para la comprensión del proceso al momento de realizar un registro ya sea un sitio turístico o un servicio. Como primer análisis, se observó que las validaciones funcionaban correctamente ya que todos los campos eran obligatorios. Como segundo análisis, se observó que el botón de registrar no funcionaba independientemente de si los campos estaban llenos o vacíos. Dentro de este análisis se observó que la función que permitía la conexión con el formulario de registro y la base de datos no existía dentro del código. Para resolver este problema se decidió reconstruir la funcionalidad que permite la conexión con la interfaz de usuario y la base de datos. El modelo final no resultó ser el que estaba contemplado en un inicio, pero ayuda a resolver la problemática al momento de agregar un sitio o servicio turístico. Debido a ello se optó por la utilización de funciones según las otras conexiones que tiene el sistema, de esta manera se mantiene una estandarización del código.

Ingeniería Directa

Como último paso dentro de este sistema web se realizó la implementación de una nueva funcionalidad como en este caso el uso de un Chatbot en la ventana principal. La lógica del Chatbot tiene un funcionamiento de fácil entendimiento ya que se desarrolló como sustituto de la barra de búsqueda que contiene el otro sistema. Dicha lógica está estructurada en base a preguntas por parte del sistema y respuesta por parte del usuario en base a dos opciones. Las preguntas están configuradas en base a los tipos de actividades que permite ingresar dentro del sistema y de esta forma la conversación fluirá dependiendo si el usuario desea buscar un lugar en base a las actividades mostradas o si desea buscar otro lugar con otras actividades. Una vez que las actividades mostradas sean las deseadas por el usuario, se mostrará los sitios turísticos que contemplen dichas actividades, independientemente si el sitio se registró como lugar turístico o como servicio. De esta forma damos mayor interacción al usuario con el sistema, brindando una solución de buscar el sitio turístico ideal para cada necesidad.

PRUEBAS

Tabla 6

Nivel de tipos de pruebas en cada sistema y sitio web

| Sitio/Sistema web | Pruebas funcionales | Pruebas de eficiencia | Pruebas de usabilidad |
|---|----------------------------|------------------------------|------------------------------|
| Sitio web dinámico | ↑ | ↑ | ↑ |
| Sistema web para venta de productos agrícolas | → | → | → |
| Sistema web para promoción de sitios turísticos | → | → | → |

Se planificó tener tres tipos de prueba, funcionales, eficiencia y de usabilidad, estas se presentarán con nivel alto o medio, en el cual alto corresponde a una flecha hacia arriba y media una flecha hacia la derecha, como se puede visualizar en la *Tabla 6*.

En el sitio web dinámico se trabajó un nivel alto en los tres tipos de pruebas, esto por el hecho de que fue realizado desde cero, con el objetivo de tener una retroalimentación de cómo está funcionando, cuáles son sus límites y en que podría mejorar.

Para los otros dos sistemas web se aplicó un nivel de pruebas medio ya que son sistemas en los cuales se realizó cambios en ciertas funcionalidades específicas, por lo cual el objetivo de estas pruebas fue que de igual forma se adaptaron estos cambios al sitio web inicial.

Tabla 7

Tipo de pruebas para los componentes del sitio web dinámico

| Componente | Tipo de prueba | Herramienta |
|-----------------------------|-----------------------|------------------------------|
| Navegación entre sitios web | Usabilidad | Usuarios prueba |
| | Funcionalidad | Caso prueba |
| | Eficiencia | Jmeter |
| Inicio de sesión | Funcionalidad | Caso prueba |
| Responsividad | Usabilidad | Prueba manual |
| Sitio web dinámico | Funcionalidad | Usuarios Prueba/ Caso prueba |
| | Usabilidad | Usuarios prueba |
| | Eficiencia | Jmeter |

Tabla 8

Tipo de pruebas para los componentes del sistema web de venta de productos agrícolas

| Componente | Tipo de prueba | Herramienta |
|---------------------------|-----------------------|--------------------|
| Navegación entre pestañas | Eficiencia | Jmeter |
| | Funcionalidad | Caso prueba |
| Interfaz de usuario | Usabilidad | Usuarios Prueba |

Tabla 9

Tipo de pruebas para los componentes del sistema web de promoción de sitios turísticos

| Componente | Tipo de prueba | Herramienta |
|------------|----------------|-----------------|
| Chatbot | Usabilidad | Usuarios Prueba |
| | Funcionalidad | Caso prueba |

Pruebas de usabilidad

Estas pruebas se las realizo con usuarios prueba, ya que al mencionar un sitio web dinámico, se debe verificar que tan amigable y fluido es para que así tenga una experiencia completa y sin ninguna interrupción. Para realizar estas pruebas se utilizó un formulario con el objetivo de obtener valores reales y promedios que ayuden a validar el trabajo que se realizó, se presentará algunos de los valores obtenidos.

Figura 21

Pregunta uno que ayudó con la prueba de usabilidad.

¿Que tan amigable es el sitio web?

13 respuestas

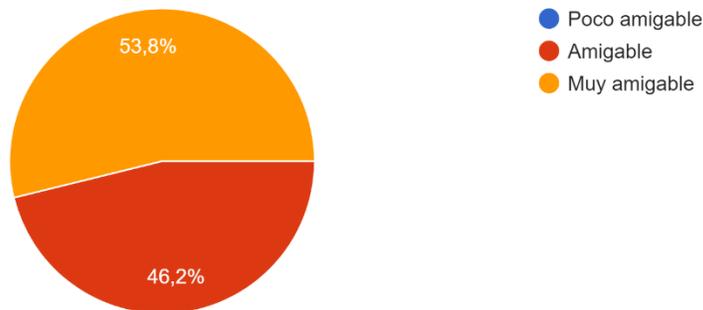


Gráfico de pastel que presenta un porcentaje de 53,8%, donde se menciona que el sitio web dinámico es muy amigable y el 46,2% es amigable.

Figura 22

Pregunta dos que ayudó con la prueba de usabilidad.

¿Que tan dinámico se observó el sitio web?
13 respuestas

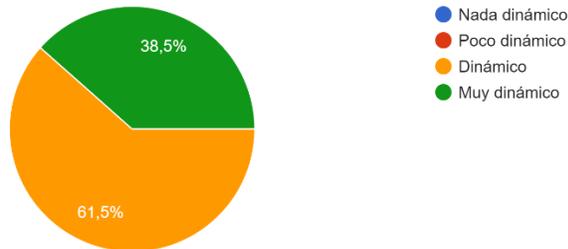
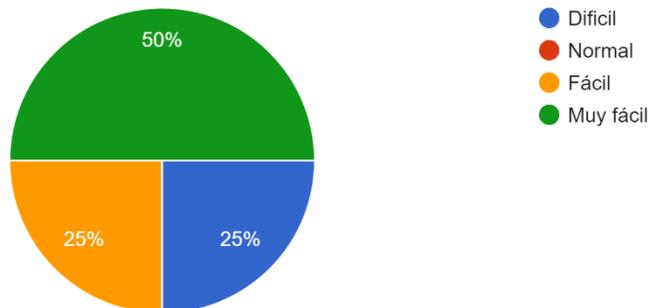


Gráfico de pastel que presenta un porcentaje de 61.5% mencionan que el sitio es dinámico y el 38.5% que es muy dinámico con respecto a las variables que se usan y están vinculadas en distintos puntos.

Figura 23

Pregunta tres que ayuda con la prueba de usabilidad

¿Que tan fácil fue encontrar el ingreso a la administración del sitio web?
4 respuestas



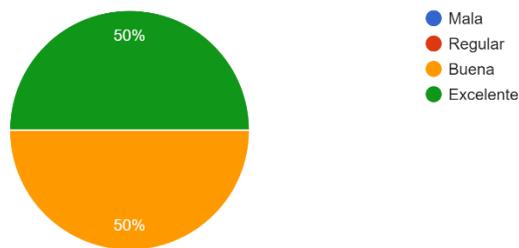
Aquí se muestra la dificultad al encontrar el botón para registro del usuario, tiene un valor de 50% con muy fácil de encontrar, 25% fácil y 25% difícil. Tomando en cuenta que el botón se encuentra en el pie de página del sitio web dinámico.

Figura 24

Pregunta cuatro que ayuda con la prueba de usabilidad

¿Cómo le parece la forma en que se editan, crean y borran la información del sitio web?

4 respuestas



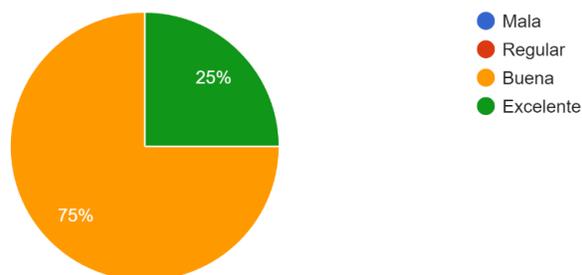
Aquí se muestra la facilidad que se brinda al administrador para realizar cambios en el contenido. Con porcentajes de 50% Excelente la forma en que se está manejando la parte dinámica, y 50% Bueno.

Figura 25

Pregunta cinco que ayuda con la prueba de usabilidad

¿Cómo le parece la interfaz de administración del sitio web?

4 respuestas



Los administradores mencionan que la interfaz es buena con un 75% y excelente con 25%, ya que el uso de la interfaz de administración es muy relacionado con la interfaz del usuario, por este motivo se esperaban esos porcentajes.

Al realizar estas pruebas se pudo observar que los sitios web, tienen una usabilidad aceptable en base a los usuarios prueba. La mayoría de los encuestados se encontraron satisfechos con la interacción de los sitios web.

Pruebas de funcionalidad

Están enfocadas al desarrollo correcto de solicitudes, rutas y características específicas en el sitio web, para esto se utilizó casos prueba y así obtener una retroalimentación sobre ciertas funcionalidades, también se realizó un formulario con preguntas específicas con usuarios prueba con el mismo objetivo.

Tabla 10

Caso de Prueba: Navegación entre sitios web

| Caso de Prueba: Navegación entre sitios web |
|---|
| Descripción |
| Verificar que se puedan navegar entre sitios web lo cuales se encuentran vinculados al sitio web dinámico. |
| Pasos de Prueba |
| Paso 1: Abrir el sitio web dinámico. |
| Paso 2: Verificar que se encuentre cargado correctamente el sitio web dinámico. |
| Paso 3: Elegir cualquiera de las tarjetas que se encuentra en la sección de servicios. |
| Paso 4: Hacer clic en el botón ir a página y verificar que se abra otra pestaña con la página web seleccionada. |
| Datos de Prueba |
| En este caso de prueba no se utilizan datos. |
| Resultados Esperados |
| Resultado 1: La nueva pestaña con el sitio web seleccionado se abre correctamente, mostrando la información de dicho sitio. |

Tabla 11*Caso de Prueba: Inicio de sesión*

| Caso de Prueba: Inicio de sesión |
|--|
| Descripción |
| Verificar que se pueda iniciar sesión de forma correcta a la administración |
| Pasos de Prueba |
| Paso 1: Abrir el sitio web dinámico. |
| Paso 2: Ir al <i>footer</i> y seleccionar el botón ingrese para editar. |
| Paso 3: Se redirigirá a la pestaña de iniciar sesión. |
| Paso 4: Ingresar los datos, email y contraseña de algún usuario registrado y dar clic en ingresar. |
| Paso 5: Se desplegará la pestaña de administración. |
| Datos de Prueba |
| Correo del usuario registrado: “jperez@gmail.com” |
| Contraseña del usuario: “1234usuario” |
| Resultados Esperados |
| Resultado 1: La pestaña que se desplego fue la de administración. |

Tabla 12*Caso de Prueba: Inicio de sesión*

| Caso de Prueba: Inicio de sesión |
|--|
| Descripción |
| Verificar que se pueda iniciar sesión de forma correcta a la super administración. |
| Pasos de Prueba |
| Paso 1: Abrir el sitio web dinámico. |
| Paso 2: Ir al <i>footer</i> y seleccionar el botón ingrese para editar. |
| Paso 3: Se redirigirá a la pestaña de iniciar sesión. |
| Paso 4: Ingresar los datos, email y contraseña de algún usuario registrado y dar clic en ingresar. |
| Paso 5: Se desplegará la pestaña de super usuario. |
| Datos de Prueba |
| Correo del usuario registrado: “superadmin@gmail.com” |
| Contraseña del usuario: “admin1234” |
| Resultados Esperados |
| Resultado 1: La pestaña que se desplego fue la de super administrador. |

Tabla 13

Caso de Prueba: Sitio web dinámico (Agregar opciones de navegación)

| Caso de Prueba: Sitio web dinámico (Agregar opciones de navegación) |
|---|
| Descripción |
| Verificar que se maneje de forma correcta las funcionalidades dinámicas del sitio web. |
| Pasos de Prueba |
| Paso 1: Abrir el sitio web dinámico. |
| Paso 2: Ir al <i>footer</i> y seleccionar el botón ingrese para editar. |
| Paso 3: Se redirigirá a la pestaña de iniciar sesión. |
| Paso 4: Ingresar los datos, email y contraseña de algún usuario registrado y dar clic en ingresar. |
| Paso 5: Se desplegará la pestaña de administrador |
| Paso 6: Se encontrará la barra de navegación la cual tendrá dos opciones adicionales. |
| Paso 7: Seleccionar la opción agregar |
| Paso 8: Se desplegará una ventana emergente donde se ingresará el nombre de la nueva opción de navegación que se desee tener. |
| Datos de Prueba |
| Correo del usuario registrado: “jperez@gmail.com” |
| Contraseña del usuario: “1234usuario” |
| Nombre de la nueva opción: “Capacitación” |
| Resultados Esperados |
| Resultado 1: En la barra de navegación del usuario y administrador se encuentra la nueva opción creada. |

Tabla 14

Caso de Prueba: Sitio web dinámico (Agregar tarjetas en pestañas)

| Caso de Prueba: Sitio web dinámico (Agregar tarjetas en pestañas) |
|---|
| Descripción |
| Verificar que se maneje de forma correcta las funcionalidades dinámicas del sitio web. |
| Pasos de Prueba |
| Paso 1: Abrir el sitio web dinámico. |
| Paso 2: Ir al <i>footer</i> y seleccionar el botón ingrese para editar. |
| Paso 3: Se redirigirá a la pestaña de iniciar sesión. |
| Paso 4: Ingresar los datos, email y contraseña de algún usuario registrado y dar clic en ingresar. |
| Paso 5: Se desplegará la pestaña de administrador |
| Paso 6: Bajo la barra de navegación se encontrará un botón con el nombre agregar noticia, seleccionar esta opción. |
| Paso 7: Se desplegará una tarjeta emergente donde se ingresará el título, descripción e imagen de la nueva tarjeta. |

| |
|--|
| Paso 8: Dar clic en enviar, y observar cómo se agregó a continuación de la última tarjeta creada con los datos ingresados. |
| Datos de Prueba |
| Correo del usuario registrado: “jperez@gmail.com” |
| Contraseña del usuario: “1234usuario” |
| Nombre de la nueva opción: “Capacitación” |
| Resultados Esperados |
| Resultado 1: En la barra de navegación del usuario y administrador se encuentra la nueva opción creada. |

Tabla 15

Caso de Prueba: Chatbot

| Caso de Prueba: Chatbot |
|--|
| Descripción |
| Verificar el Chatbot propuesto para la aplicación de sitios turísticos. |
| Pasos de Prueba |
| Paso 1: Abrir el sitio web dinámico. |
| Paso 2: Verificar que se encuentre cargado correctamente el sitio web dinámico. |
| Paso 3: Elegir la tarjeta que muestra el servicio de sitios turísticos. |
| Paso 4: Hacer clic en el botón ir a página y verificar que se abra otra pestaña con la página web seleccionada. |
| Paso 5: La página de sitios turísticos, mostrará en la parte inferior un botón con el nombre de Chatbot, al hacer clic en el mismo, se desplegará una ventana para poder interactuar con el sitio. |
| Paso 6: La primera pregunta será, el seleccionar la actividad |
| Datos de Prueba |
| Pregunta uno en Chatbot: “2” |
| Pregunta dos en Chatbot: “1” |
| Resultados Esperados |
| Resultado 1: Al seleccionar la opción uno en la segunda pregunta del Chatbot, permitirá el redireccionamiento a dicho lugar turístico dentro del sitio web |

De esta manera se espera lograr cubrir algunos escenarios que se presentan en los sitios web y así documentar cual es el funcionamiento que se espera tener y permitir su reproducibilidad en estos casos, así mismo se realiza un repaso de ciertas funcionalidades esperando disminuir los defectos para así mejorar los sitios web.

Figura 26

Pregunta uno al usuario para prueba de funcionalidad.

¿Cómo le parece el acceso a la pagina web?
13 respuestas

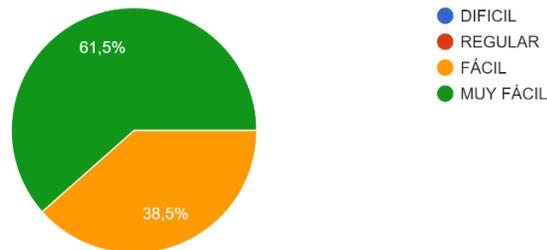


Gráfico de pastel que presenta un porcentaje de 61,5% de muy fácil el acceso a la página web, contra un 38,5% que muestran que esta tarea es fácil.

Figura 27

Pregunta dos al usuario para prueba de funcionalidad.

¿Tuvo algún problema para ingresar al sitio web?
13 respuestas

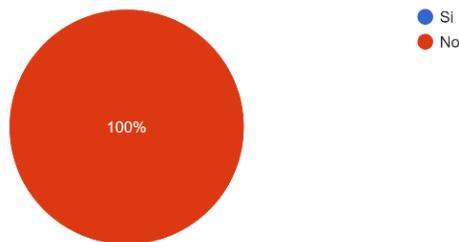


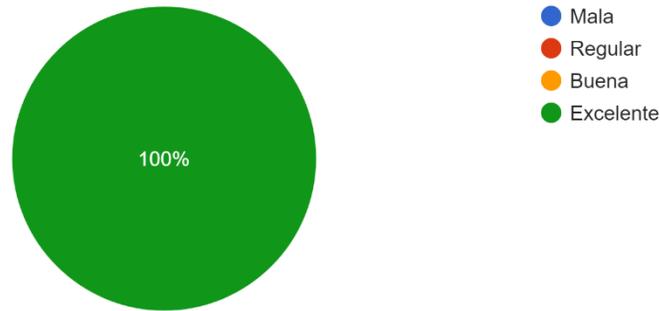
Gráfico de pastel que presenta un porcentaje de 100% en el que nadie tuvo problema al acceder al sitio web dinámico.

Figura 28

Pregunta dos que permite validar la funcionalidad

¿Cómo le parece que su contraseña sea encriptada al iniciar sesión?

4 respuestas



El gráfico de pastel muestra el 100% de usuarios encuestados a los que les parece una excelente idea que sus contraseñas sean manejadas mediante encriptación.

Pruebas de eficiencia

Esto permitirá evaluar el rendimiento y el comportamiento de los sitios web ante diferentes cargas que se lleguen a presentar. Al ejecutar estas pruebas buscamos conocer cuáles son los niveles de rendimiento en los que se encuentran los sitios web, de esta forma identificar problemas que puedan existir antes que llegue a los usuarios. Aquí se utilizó la herramienta Apache Jmeter con las siguientes pruebas donde el puerto 3000 nos ayudara a visualizar el sitio web dinámico, 4000 el sitio de turismo y 5000 para los productos agrícolas.

Tabla 16

Pruebas de eficiencia para el sitio web y los sistemas web

| Ruta | Configuración de hilos(usuarios) | Tiempo promedio de respuesta (segundos) | Observación |
|--------|----------------------------------|---|-------------|
| :3000/ | 1000 | 4.911 | Tiempo bajo |
| :3000/ | 2000 | 5.031 | Tiempo bajo |

| | | | |
|--------|-------|--------|---|
| :3000/ | 4000 | 9.871 | Tiempo alto |
| :4000/ | 1000 | 10.916 | Tiempo alto |
| :4000/ | 2000 | 10.451 | Tiempo alto |
| :4000/ | 4000 | 16.381 | Tiempo alto |
| :5000/ | 1000 | 8.051 | Tiempo alto |
| :5000/ | 2000 | 9.044 | Tiempo alto |
| :5000/ | 4000 | 14.332 | Tiempo alto |
| :3000/ | 75000 | 9.003 | Tiempo medio |
| :4000/ | 75000 | 25.154 | Tiempo alto, sé deja de enviar peticiones simultaneas a partir de 70000 |
| :5000/ | 75000 | 23.612 | Tiempo alto, sé deja de enviar peticiones simultaneas a partir de 70000 |

Se puede observar con los resultados de la *Tabla 16* que los tiempos de respuesta más bajos los maneja el sitio web dinámico, seguido por el sitio web de productos agrícolas y finalmente el sitio web de turismo. Cabe recalcar que las pruebas se ejecutaron para la mayoría de las peticiones, donde se analizó que los tiempos son similares a las rutas con las que inicia cada sitio web.

Se podrá observar en las siguientes imágenes la aplicación con algunos ejemplos de las cargas realizadas y el consumo de la infraestructura con la que se ejecutaron estas pruebas.

Figura 29

Peticiones realizadas al sitio web dinámico

| Etiqueta | # Muestras | Media | Min | Máx | Desv. Estándar | % Error |
|----------|------------|-------|-----|-------|----------------|---------|
| Listado | 4000 | 9920 | 0 | 21947 | 5437.23 | 41.05% |
| Total | 4000 | 9920 | 0 | 21947 | 5437.23 | 41.05% |

Figura 30

Peticiones realizadas al sitio web de productos agrícolas

| Etiqueta | # Muestras | Media | Min | Máx | Desv. Estándar | % Error |
|----------|------------|-------|-----|-------|----------------|---------|
| Listado | 4000 | 16301 | 0 | 24803 | 3834.94 | 4.67% |
| Total | 4000 | 16301 | 0 | 24803 | 3834.94 | 4.67% |

Figura 31

Peticiones realizadas al sitio web de turismo

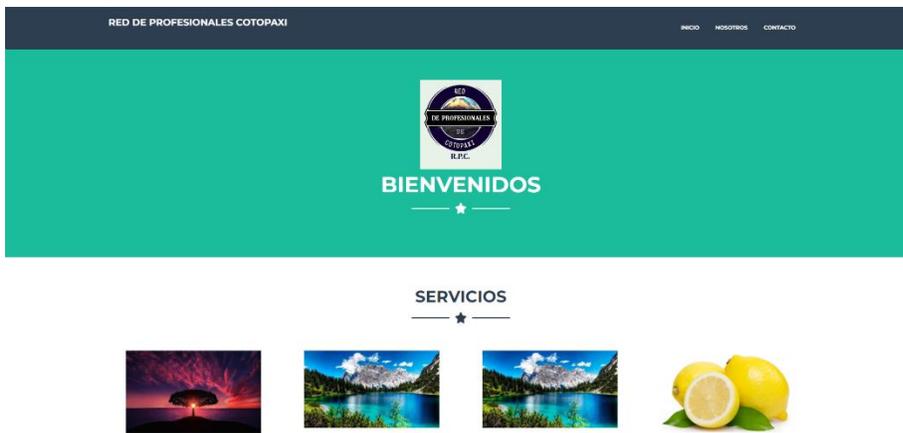
| Etiqueta | # Muestras | Media | Mín | Máx | Desv. Estándar | % Error |
|----------|------------|-------|-----|-------|----------------|---------|
| Listado | 4000 | 24038 | 0 | 36102 | 7958.03 | 0.00% |
| Total | 4000 | 24038 | 0 | 36102 | 7958.03 | 0.00% |

PERSPECTIVA GLOBAL DEL SITIO WEB

Permite detallar cual es la correcta navegación dentro del sitio web dinámico, mostrando sus funcionalidades principales y resaltando ciertas características.

Figura 32

Página inicial del sitio web dinámico.



Se muestra la página principal la que será visualizada por el usuario al momento de ingresar al sitio web donde se encuentra la *barra de navegación*, imagen de bienvenida y el *contenedor de tarjetas* donde permitirá redireccionar hacia otros sistemas web que se encuentren en línea configurados previamente por los administradores.

Figura 33

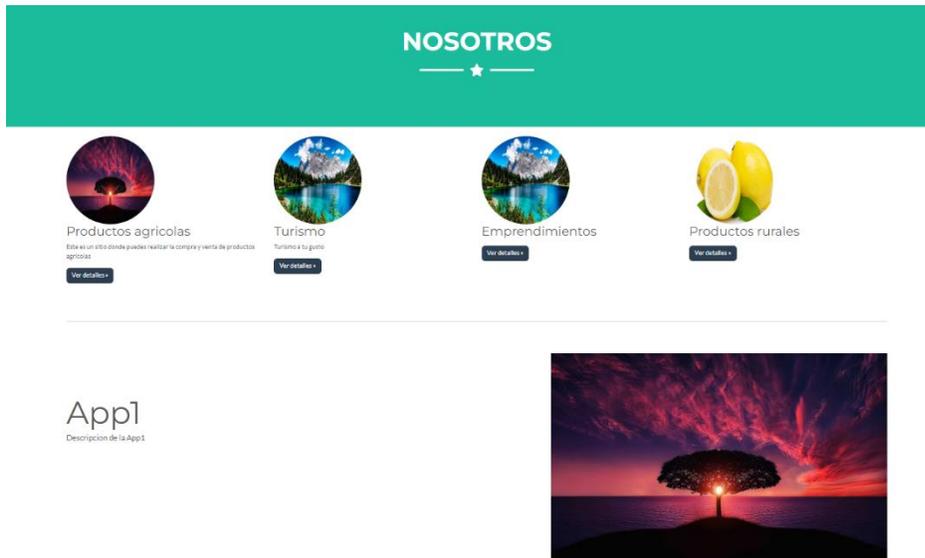
Tarjeta para redireccionar a otro sitio web.



Esta tarjeta permite la redirección entre sitios web tomando como lugar principal el sitio web dinámico, es preciso mencionar que los enlaces de las tarjetas son dinámicos, y se pueden redireccionar a cualquier sitio que se encuentre en línea. De igual manera es posible modificar el título, imagen y descripción de la tarjeta y se verá reflejado en la vista principal como se lo menciono anteriormente.

Figura 34

Página nosotros del sitio web dinámico.



La pestaña nosotros, permite ingresar información sobre los sistemas web que se están promocionando y redireccionando, hay una posibilidad de ingresar hasta contenido para cuatro sitios web diferentes. De igual forma en este apartado se podrá modificar contenido informativo como misión, visión o principales características, pero su principal funcionalidad es la administración y modificación de los sistemas web involucrados. Esta pestaña solo se encuentra en el sitio web debido a que se presentara toda la información relevante y mantener un estándar para poder visualizar y dar el acceso libre si se requiere modificar el contenido a medida que el entorno cambie.

Figura 35

Página contacto del sitio web dinámico.

Contáctanos

Envíanos un mensaje

Mensaje

Información

Esta es nuestra principal información que debes saber sobre nosotros

- 📍 Dirección: Cotopaxí
- ☎️ Teléfono: 0999999999
- ✉️ Correo electrónico: redprofesionalescotopaxi@gmail.com
- 🌐 Página Web: <http://redprofesionales.com>
- 📱 Redes sociales: [RedProfesionalesCotopaxi](#)
- 📄 Descripción: Esta es una descripción de la ubicación

DIRECCIÓN

Cotopaxí
Esta es una descripción de la ubicación

ENCUENTRANOS



RED DE PROFESIONALES COTOPAXI

Copyright © 2023 Red de Profesionales Cotopaxí

Se muestra información básica sobre la organización, la misma que estará anclada al pie de página. En este apartado permitirá a los usuarios enviar comentarios sobre el sitio web o los sistemas web anclados, de modo que se podrá mantener una interacción con los usuarios que visiten el sitio web y así lograr analizar en un futuro posibles cambios o mejoras tanto al sitio web dinámico como a los sistemas web ya que esta pestaña solo pertenece al sitio web se podrá visualizar todos los comentarios.

Figura 36

Página de registro de usuario administrador del sitio web dinámico.

The screenshot shows the registration page for the 'RED DE PROFESIONALES COTOPAXI' website. The header includes the site name and navigation links: INICIO, NOSOTROS, and CONTACTO. The main content area features an illustration of a hand pointing at a computer screen with a pen, symbolizing digital work or administration. To the right of the illustration is a registration form with the following fields and buttons:

- Correo electrónico: Ingrese el correo electrónico
- Contraseña: Ingrese la contraseña
- Botón: Iniciar Sesión

The footer contains three sections: DIRECCIÓN (Cotopaxi, Esta es una descripción de la ubicación), ENCUENTRANOS (with a Facebook icon and a 'Ingresar para editar' button), and RED DE PROFESIONALES COTOPAXI (Copyright © 2023 Red de Profesionales Cotopaxi).

Se ingresará el correo del usuario y la contraseña, para acceder a las opciones de administración, también se podrá ingresar desde este registro al super administrador el cual permitirá más opciones de edición, como la creación y eliminación de usuarios administradores que tienen acceso total para la modificación de los componentes establecidos anteriormente. Para visualizar esta página el ingreso se mantiene alejado de la vista principal dando mayor seguridad al tener el control total de los contenidos del sitio web.

Figura 37

Página que permite editar el inicio del sitio web dinámico.

The screenshot shows the editing page for the 'RED DE PROFESIONALES COTOPAXI' website. The header includes the site name and navigation links: AGREGAR, ELIMINAR, INICIO, NOSOTROS, CONTACTO, and CERRAR SESIÓN. Below the header is a 'Agregar Noticia' button. The main content area features a large light blue background with a vertical blue line and a circular marker. On the left side, there is a preview of a news item:

- Noticia N1
- (01) de junio de 2023
- Este es una pera
- Botón: Eliminar Noticia

La administración del inicio consta de la misma estructura, pero con la facilidad de edición, existen varias funcionalidades dentro de estas permite agregar, editar y eliminar las tarjetas. De igual manera permite crear o eliminar pestañas, en este apartado se logra visualizar todos los componentes configurados para que sea posible la modificación, colocando las funcionalidades a primera vista del usuario para no generar confusiones o perdida al momento de administrar el sitio web.

Figura 38

Página que permite editar la página nosotros del sitio web dinámico.

Administración Nosotros
Como administrador podrás editar todos los campos de la sección nosotros mediante este formulario.

Primera sección de código

| | | | |
|---|---|---|---|
| Cargar imagen app 1 Seleccionar Archivo Browse | Cargar imagen app 2 Seleccionar Archivo Browse | Cargar imagen app 3 Seleccionar Archivo Browse | Cargar imagen app 4 Seleccionar Archivo Browse |
| Cargar Título 1 Productos agrícolas | Cargar Título 2 Turismo | Cargar Título 3 Emprendimientos | Cargar Título 4 Productos rurales |
| Cargar descripción 1 Este es un sitio donde puedes realizar la compra y venta de productos agrícolas | Cargar descripción 2 Turismo a tu gusto | Cargar descripción 3 | Cargar descripción 4 |
| Link App1 redprofesionalescotopaxi.com | Link App2 redprofesionalescotopaxi.com | Link App3 redprofesionalescotopaxi.com | Link App4 redprofesionalescotopaxi.com |

Submit

Segunda sección del código

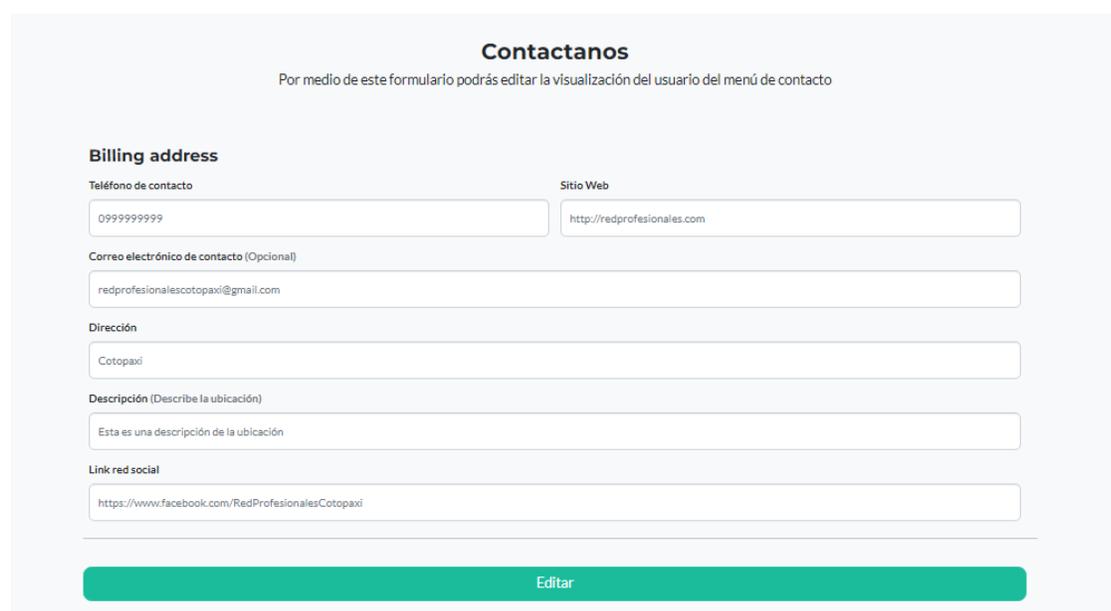
| | | |
|------------------|---|--|
| Título 1 App1 | Descripción 1 Descripcion de la App1 | Imagen 1 Seleccionar Archivo Browse |
| Título 2 App2 | Descripción 2 redprofesionalescotopaxi.com | Imagen 2 Seleccionar Archivo Browse |
| Título 3 App3 | Descripción 3 redprofesionalescotopaxi.com | Imagen 3 Seleccionar Archivo Browse |
| Título 4 aa | Descripción 4 redprofesionalescotopaxi.com | Imagen 4 Seleccionar Archivo Browse |

Submit

La administración de la página nosotros es la más compleja, como se mencionó anteriormente permite la edición de todo lo que observa el usuario, de esta manera garantiza la completa administración del contenido que se publique. Colocando las características que desee al momento de redireccionar a cualquier sistema o sitio web que se requiera publicar posteriormente. Como primer apartado se coloca las características que se visualizará en la pestaña principal, al momento de abrir la tarjeta del servicio seleccionado se podrá visualizar este contenido y en el siguiente apartado se puede colocar más características de los sistemas web o si se requiere añadir información más relevante para comunicar el sitio web con los usuarios.

Figura 39

Página que permite editar el contacto del sitio web dinámico.



El formulario, titulado "Contactanos", permite editar la visualización del usuario del menú de contacto. Incluye los siguientes campos:

- Billing address**
- Teléfono de contacto:** 0999999999
- Sitio Web:** http://redprofesionales.com
- Correo electrónico de contacto (Opcional):** redprofesionalescotopaxi@gmail.com
- Dirección:** Cotopaxi
- Descripción (Describe la ubicación):** Esta es una descripción de la ubicación
- Link red social:** https://www.facebook.com/RedProfesionalesCotopaxi

Un botón verde "Editar" está ubicado al final del formulario.

Esto pertenece al panel de edición de la pestaña *contacto*, el cual es dinámico y dependerá del administrador el cambio de esta información. Como se mencionó anteriormente la información que se modifica en este apartado es importante ya que dependen otros componentes de la información almacenada en esta pestaña, como la ubicación que se encuentra en el *Footer* y de

igual forma la dirección de correo electrónico es al cual se enviarán los comentarios que envíen sobre el sitio web o los sistemas web.

Figura 40

Página del super administrador del sitio web dinámico.

Lista de Usuarios

| Nombre | Correo Electrónico | Acciones |
|------------|----------------------|--------------------------|
| Perez | jperez@gmail.com | Eliminar |
| Superadmin | superadmin@gmail.com | Eliminar |

[Agregar Usuario](#) [Iniciar Sesión](#)

Nuevo Usuario

Usuario

Correo electrónico

Contraseña

[Añadir Usuario](#)

Esta funcionalidad es propia del super usuario, la cual permite agregar o eliminar usuarios administradores. Para agregar un usuario administrador se coloca un formulario en el que le permite ingresar las credenciales, y una vez agregado se podrá visualizar los datos creados en la tabla con su respectivo botón que le permitirá eliminar el usuario seleccionado, por seguridad no se coloca la contraseña en la tabla, pero al momento de almacenarla se envía con encriptación para que no se pueda visualizar por ningún usuario.

Figura 41

Página inicial del sitio web compra y venta de productos agrícolas al que fue redireccionado desde el sitio web dinámico.



El sitio web de productos agrícolas, el cual fue redireccionado desde una de las tarjetas que se encuentran en la página principal del sitio web dinámico. De este modo, se puede observar el cambio visual que se aplicó a este sistema web, manteniendo una simetría visual al momento de visitar el sitio web y los sistemas web.

Figura 42

Página inicial del sitio web de turismo al que fue redireccionado desde el sitio web dinámico



El sitio web de turismo, el cual fue redireccionado desde una de las tarjetas que se encuentran en la página principal del sitio web dinámico. En este sistema web se puede visualizar el nuevo programa de Chatbot la cual es la nueva funcionalidad implementada para buscar los sitios web de forma más interactiva con el usuario y el sistema.

PUBLICACIÓN DEL SITIO WEB

Para esto se realizó un análisis conforme a las pruebas ejecutadas por lo cual se decidió contratar una infraestructura VPS (Servidor virtual privado), la cual tiene los siguientes recursos y servicios pre-instalados:

Tabla 17

Características del servidor virtual privado.

| Procesador | Ram | Almacenamiento | CPU´s | Servicios |
|---|------|----------------|-------|--|
| Intel(R) Xeon(R) Gold 6330N CPU @ 2.20GHz | 4 gb | 500gb | 4 | *NodeJs version: 14.17.1 LTS *MySQL version: 8.0.25 *SSH *Servicio deU DNS *Mongodb version: 4.2.24 *Servicio ftp |

Cuando se concretó el servicio de hosting VPS, se adquirió unas credenciales y un enlace por donde se deberá acceder a las máquinas virtuales cliente y servidor para la posterior configuración de los sitios web, para eso se planifico la transferencia de datos la cual consiste en los siguientes pasos:

- Se subió las carpetas de los sitios web al OneDrive, el cual se descargó desde la máquina virtual VPS.
- Por medio del servicio FTP se realizó la transferencia de las carpetas de los sitios web desde el cliente hasta el servidor.

- Se colocó las carpetas, en la ruta especificada por parte del hosting, para que se pueda visualizar desde el Internet el sitio web dinámico.

Para vincular los sitios web de productos agrícolas y turismo, se instaló un servidor web Nginx con el propósito que actúe como intermediario entre los clientes y los servidores de back-end. Con el objetivo que el sitio web dinámico sea la ruta principal, desde donde se hará una redirección de solicitudes hacia las otras aplicaciones. Para esto se realizó la siguiente configuración en la ruta */etc/nginx/sites-available/default*.

Figura 43

Configuración ruta de puerto

```
location / {
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header Host $http_host;
proxy_set_header X-NginX-Proxy true;
proxy_pass "http://127.0.0.1:3000";
proxy_redirect off;
client_max_body_size 100M;
}
```

Se puede observar que en las configuraciones se envió la ruta con el puerto por donde se ejecuta el sitio web dinámico desde Express y se le asignó la ruta principal del servidor web Nginx para que desde este punto se muestre el sitio.

10. CONCLUSIONES

1. Se concluye que la agrupación de los sistemas web desde un enfoque de perspectiva de usuario es fundamental ya que de este modo se ofrece una experiencia unificada y coherente a los usuarios, brindando la posibilidad de informarse, comprar o vender productos agrícolas, así como promover y reservar servicios turísticos.
2. Durante el proceso se observó que el diseño de un patrón arquitectónico adecuado es importante si se decide agrupar dos sistemas web completamente distintos en cuanto a funcionalidades debido a que garantiza la estandarización de la estructura facilitando su mantenimiento y futuras expansiones.
3. Durante todo el proceso se observó que existe una evolución constante de las tecnologías referente a sistemas web, ya que al momento de adaptarse las demandas cambiantes por parte de los usuarios y las necesidades de las organizaciones. La introducción de nuevos conceptos, tecnologías y enfoques, como la integración de sistemas, la refactorización, la reingeniería y la estandarización del código, demuestra el dinamismo y la innovación en este campo.
4. Se concluyó que la usabilidad y la accesibilidad son aspectos cruciales en el desarrollo de sistemas web. La capacidad de ofrecer interfaces intuitivas, fáciles de usar y accesibles para una amplia cantidad de usuarios, es fundamental para garantizar una experiencia positiva y satisfactoria.
5. Se observó que el enfoque en el rendimiento y la escalabilidad son requerimientos primordiales en los sistemas web. La utilización de patrones arquitectónicos adecuados, el uso de tecnologías eficientes y la optimización del código son elementos esenciales para garantizar que los sistemas web puedan manejar grandes volúmenes de tráfico y ofrecer un rendimiento óptimo.

6. Se concluye que los sistemas web deben ser flexibles y adaptables a los cambios y requerimientos en constante evolución. La capacidad de integrarse con otros sistemas, la gestión de contenidos dinámicos y la posibilidad de personalización son características esenciales para satisfacer las necesidades cambiantes de las organizaciones y los usuarios

11. RECOMENDACIONES

1. Se recomienda que los sistemas web se administren bajo un plan de mantenimiento y una actualización constante para garantizar su rendimiento, seguridad y compatibilidad con las últimas tecnologías. La implementación de buenas prácticas de desarrollo, como la refactorización, reingeniería y la adopción de procesos de actualización y monitoreo adecuados son fundamentales para mantener los sistemas web eficientes y actualizados.
2. Al momento de comenzar a modificar el sitio web dinámico por primera vez se recomienda leer con detenimiento el manual de usuario, ya que se detallan todos los componentes necesarios para comprender el funcionamiento del sitio web, por ello es importante comprender como interaccionan todos los componentes y de esta forma manejar la información adecuadamente y con una buena estructura.
3. Al momento de trabajar con sistemas web heredados se recomienda realizar procesos de reingeniería o refactorización, ya que de este modo se podrá optimizar su rendimiento, escalabilidad o mantenibilidad. Esto permitirá mantener un sistema de alta calidad y adaptando a las necesidades cambiantes de la organización.
4. Si se desea adaptar el enfoque del sitio web dinámico en otros sistemas web se recomienda tomar en cuenta principalmente los requisitos es decir identificar la funcionalidad clave que se desea administrar como la gestión de contenidos y posteriormente crear un diseño flexible que permita

agregar, editar y eliminar secciones o elementos de forma sencilla, para ello se puede utilizar componentes reutilizables o plantillas para agilizar el proceso.

5. Al momento de realizar mantenimiento al sitio web dinámico se recomienda implementar un sistema de copias de seguridad automatizado para respaldar regularmente la información que se almacena en el sitio web, de este modo se garantiza que, en cualquier problema o pérdida de datos, sea posible restaurar a la última versión estable del sitio web.
6. Se recomienda mantener una gestión correcta de los permisos para los usuarios administradores, manteniendo los roles y permisos adecuados para la administración del sitio web. Limitando el acceso a las funcionalidades y áreas relevantes, evitando así modificaciones no deseadas o accidentales.
7. Para que el sitio web dinámico mantenga su estabilidad se recomienda actualizar regularmente el software es decir actualizar complementos, bibliotecas y cualquier otra tecnología utilizada, debido a que las actualizaciones incluyen mejoras de seguridad, corrección de errores y nuevas características que permiten al sitio web mantener su principal enfoque que es la adaptabilidad al entorno en el que se encuentre.

12. REFERENCIAS BIBLIOGRÁFICAS

- [1] Suh, W. (Ed.). (2005). Web engineering: principles and techniques. IGI Global.
- [2] Doan, P. (2017). Practical Responsive Web Design. Core.ac.uk.
- [3] Mickley, Z (2008). Flexible Web Design: Creating Liquid and Elastic Layouts with CSS. New Riders.
- [4] Keith, J (2005). DOM Scripting: Web Design with JavaScript and the Document Object Model. Friendsof.
- [5] Rajshkhar, A.P (2008). Building Dynamic Web 2.0 Websites with Ruby on Rails. Packt.
- [6] Kotler, P (2017). Marketing 4.0: Moving from Traditional to Digital. Wiley.
- [7] Berisha-Shaqiri, Aferdita & Namani, Mihane. (2015). Information Technology and the Digital Economy. Mediterranean Journal of Social Sciences. 6. 10.5901/mjss.2015.v6n6p78.
- [8] Fowler, M (2005). Patterns of Enterprise Application Architecture. Madim.
- [9] Newman, S (2015). Building Microservices: Designing Fine-Grained Systems. O'Reilly.
- [10] Fowler, M (2005). Refactoring: Improving the Design of Existing Code. Madim.
- [11] Beck, K (2012). Extreme Programming Explained: Embrace Change. Addison-Wesley.
- [12] Pooley, Rob. (2000). Software Engineering and Performance: A Roadmap. 189-199. 10.1145/336512.336553.
- [13] Raja, V, Fernandes, K (2008). Reverse Engineering: An Industrial Perspective. Springer.
- [14] Gane, Chris. (2000). Structured systems analysis: tools & techniques. Inlibrary.
- [15] Hartl, M (2013). Rails 3 in Action. Addison-Wesley.
- [16] Bradshaw, S (2019). MongoDB: The Definitive Guide. O'Reilly.
- [17] Mardan, A (2014). Express.js Deep API Reference. Apress.
- [18] Standard, I (2018). Ergonomics of human-system interaction - Part 11: Usability: Definitions and concepts (ISO 9241-11:2018). NSAI Standars.

- [19] Muldowney, O (2017). Chatbots: An Introduction and Easy Guide to Making Your Own. Curses & Magic.
- [20] Roger S. Pressman (2010). Ingeniería del software. Un enfoque práctico. Mac GrawHill.
- [21] Robert C. Martin (2009). Clean Code A Handbook of Agile Software Craftsmanship. Prentice Hall.
- [22] Fowler, M (2002). Refactoring: Improving the Design of Existing Code. Madim.
- [23] Pramod J. Sadalage (2013). NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley.
- [24] Martin Fjordvald (2018). Nginx HTTP Server - Third Edition. Packt.
- [25] Mike C, Marc H, T.J. Holowaychuck. Node.js in Action. Manning.