

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN



Trabajo de Titulación

Tema: DESARROLLO DE UN PROTOTIPO DE PÁGINA WEB DE COMERCIO
ELECTRÓNICO PARA LA VENTA E INVENTARIO DE PRODUCTOS ARTESANALES

AUTOR:
LEONARDO JOAQUÍN VILLARRUEL DE LA TORRE

QUITO DM, JUNIO DE 2024

DEDICATORIA

Dirijo mi trabajo de titulación principalmente a Dios, a mi Madre y a mi hermana. A Dios, por estar presente en cada momento en que sentía que no podía más, acompañándome hasta altas horas de la noche y dándome la fortaleza necesaria para seguir adelante. A mi Madre, por velar siempre por mi bienestar y educación, asegurándose de que no me faltara nada en ningún momento, y dándome la confianza de que todo sería posible. Por último, pero no menos importante, a mi hermana, por acompañarme en este largo trayecto y por ayudarme en cada paso que daba, confiando plenamente en mí y en mi capacidad intelectual. Gracias a cada uno de ellos, pude avanzar y lograr cumplir mis sueños.

Con amor y agradecimientos.

Leonardo Villarruel

AGRADECIMIENTOS

La culminación de este trabajo de titulación merece el agradecimiento a las personas que participaron en todo mi proceso. En primer lugar, quiero expresar mi gratitud a mis compañeros de carrera por su colaboración en aspectos esenciales del trabajo. También quiero agradecer profundamente a mi familia y amigos por ser elemento esencial en el progreso de este proyecto.

Agradezco sinceramente a mis profesores por su contribución fundamental en la adquisición de conocimientos a lo largo de mi formación académica. En particular, deseo destacar el invaluable apoyo y orientación de mi director de trabajo de titulación, el Ing. Susana Masapanta, cuya experiencia y paciencia fueron cruciales durante todo este proceso.

Además, quiero dar mi reconocimiento a la Pontificia Universidad Católica del Ecuador que me proporcionaron los necesarios para la culminación y desarrollo de este trabajo. Sin su apoyo, esta labor no habría sido posible.

Expreso mi más profundo agradecimiento a cada individuo y entidad involucrados, por su valiosa contribución y respaldo en este significativo logro.

Con amor y agradecimientos.

Leonardo Villarruel

ÍNDICE GENERAL

ÍNDICE GENERAL	4
ÍNDICE DE TABLAS	8
ÍNDICE DE FIGURAS	8
1.CAPÍTULO I – INTRODUCCIÓN.....	1
1. JUSTIFICACIÓN	1
2. PLANTEAMIENTO DEL PROBLEMA	1
3. ALCANCE.....	2
4. OBJETIVOS	2
GENERAL.....	2
ESPECÍFICOS	2
5. ANTECEDENTES	2
2.CAPÍTULO II - MARCO TEÓRICO	6
2.1. El Comercio Electrónico.....	6
2.2 Metodología Ágil de Desarrollo de Software	6
Prototipos	7
1. Prototipos de baja fidelidad	7
2. Prototipos de alta fidelidad	7
3. Prototipos desechables	8
4. Prototipos evolutivos	8
Fases de la metodología de prototipos	8
1. Definición de objetivos:	8
2. Diseño del prototipo:.....	8
3. Implementación del prototipo:	8

4. Evaluación del prototipo:	8
5. Refinamiento del prototipo:	9
6. Repetición del ciclo:	9
2.3 Herramientas	9
2.3.1 Base de Datos.....	9
2.3.2 Backend.....	10
2.3.3 FrontEnd	11
3. CAPÍTULO III PLANIFICACIÓN	13
3.1. Requerimientos Funcionales para el comercio electrónico	13
Diagramas de Casos de Uso.....	13
Caso de Uso Gestionar Productos.....	15
3.2. Requerimientos No Funcionales de la Página Web	17
3.3. Diseño Funcionales para el comercio electrónico	18
Estándares de Colores	19
Diseño de la Base de Datos.....	21
4. CAPÍTULO IV - DESARROLLO.....	22
4.1 Desarrollo del primer avance del prototipo	22
4.1.1. Diseño primer avance	22
Crear usuario.....	22
Inicio de sesión	23
Compra de productos	24
Base de Datos.....	26
4.1.2. Codificación	26

Backend en Express	26
Principales funcionalidades	27
4.1.3 Retroalimentación	27
4.2 Desarrollo del segundo avance del prototipo.....	28
4.2.1. Diseño segundo avance.....	28
Visualizar productos específicos en la página web.....	29
Inventario de los artesanos.....	32
Visualización de las ventas de los artesanos.....	32
Base de datos.....	33
4.2.2. Codificación	34
Principales Funcionalidades.....	34
4.2.3 Retroalimentación	35
4.3 Desarrollo del avance final del prototipo.....	36
4.3.1. Diseño final.....	36
Visualizar tu usuario	36
Editar Usuario	37
Cambiar Contraseña.....	38
4.3.2. Codificación	40
Funcionalidad Backend.....	40
Principales Funcionalidades.....	40
4.4.3. Pruebas del sistema	40
Plan de Pruebas del Sistema	40
Pruebas de Humo	40

Pruebas de Funcionalidad	44
Resultados Pruebas de Funcionalidad.....	45
Pruebas de Usabilidad.....	47
Resultados Pruebas de Usabilidad	48
Encuesta de Satisfacción de las pruebas de Usabilidad	51
CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES	56
5.1. Conclusiones	56
5.2. Recomendaciones	57
BIBLIOGRAFÍA	58
Anexos	61
Anexo A Formulario Todos los Usuario con tabulación.....	61
Anexo B Encuesta usuario Artesano junto a tabulación.....	71
Anexo C Código BackEnd	80
Anexo D Insertar Usuario.....	81
Anexo E Manejo Carrito de Compras	83
Anexo F Producto Detallado	85
Anexo G Ingresar Comentario.....	89
Anexo H Ingresar y editar productos en el usuario artesano.....	91
Anexo I Inventario de un producto del artesano	93
Anexo J Envío de Correos electrónicos por el backend.....	96
Anexo K Tu Perfil de usuario con diferentes funcionalidades.....	97
Anexo L Cambio de contraseña	103
Anexo M Resultados respuestas abiertas encuesta satisfacción.....	108

ÍNDICE DE TABLAS

Tabla 1 Cuadro comparativo entre distintos tipos de páginas de venta de productos artesanales en Ecuador.....	4
Tabla 2 <i>Pruebas de humo</i>	41
Tabla 3 <i>Tabla de las pruebas de funcionalidad sobre el ingreso y registro de los usuario a la página web</i>	45
Tabla 4 <i>Tabla de las pruebas de funcionalidad sobre el perfil del usuario</i>	46
Tabla 5 <i>Tabla de las pruebas de usabilidad sobre la navegación de los productos</i>	46
Tabla 6 <i>Tabla de las pruebas de funcionalidad sobre el carrito de compras</i>	46
Tabla 7 <i>Tabla de las pruebas de funcionalidad sobre la administración de los productos</i>	47
Tabla 8 <i>Pruebas de usabilidad para usuarios compradores</i>	49
Tabla 9 <i>Prueba de usabilidad para usuarios artesanos</i>	50

ÍNDICE DE FIGURAS

Figura 1 <i>Caso de Uso General</i>	14
Figura 2 <i>Caso de uso: Comprar Productos</i>	14
Figura 3 <i>Caso de uso: Gestionar Productos</i>	15
Figura 4 <i>Caso de uso: Gestionar Usuario</i>	16
Figura 5 <i>Diseño disposición página web</i>	19
Figura 6 <i>Diseño base de datos</i>	21
Figura 7 <i>Crear Cuenta</i>	23
Figura 8 <i>Iniciar Sesión</i>	24
Figura 9 <i>Compra de Productos</i>	24
Figura 10 <i>Previsualización carrito de compras</i>	25
Figura 11 <i>Carrito de Compra</i>	25
Figura 12 <i>Base de datos primer avance</i>	26
Figura 13 <i>Producto Detallado</i>	29
Figura 14 <i>Calificación productos</i>	30

Figura 15 <i>Tus Productos</i>	30
Figura 16 <i>Insertar producto</i>	31
Figura 17 <i>Editar producto</i>	31
Figura 18 <i>Inventario del Producto</i>	32
Figura 19 <i>Tus Ventas recientes por calendario</i>	33
Figura 20 <i>Base de datos segundo avance</i>	34
Figura 21 <i>Tu información personal</i>	37
Figura 22 <i>Editar tu usuario</i>	37
Figura 23 <i>Recuperar contraseña olvidada</i>	38
Figura 24 <i>Recuperar contraseña código</i>	39
Figura 25 <i>Cambiar contraseña del usuario</i>	39
Figura 26 <i>Pregunta 1: ¿Cree que la página web de venta de productos artesanales le puede ser de utilidad?</i>	51
Figura 27 <i>Pregunta 2: ¿Qué tan intuitiva le pareció la interfaz?</i>	51
Figura 28 <i>Pregunta 3: ¿Qué tan adecuados le parecieron los colores escogidos para la aplicación?</i>	52
Figura 29 <i>Pregunta 4: ¿Qué tan difícil le resultó aprender el funcionamiento de la aplicación?</i>	52
Figura 30 <i>Pregunta 5: ¿Qué tan difícil le resultó completar las tareas?</i>	53
Figura 31 <i>Pregunta 6: ¿El sitio web ofreció información clara y útil?</i>	53
Figura 32 <i>Pregunta 7: ¿Qué tan satisfecho está con el contenido del sitio web? (texto, imágenes, videos, etc.)</i>	54
Figura 33 <i>Pregunta 8: ¿Fue fácil encontrar la información que buscaba?</i>	54
Figura 34 <i>Pregunta 9: ¿Encontró algún problema al navegar por el sitio web?</i>	55
Figura 35 <i>Pregunta 10: ¿Hubo alguna funcionalidad o característica que sintió que faltaba en el sitio web?</i>	55
Figura 36 <i>Pregunta 11: ¿Recomendaría este sitio web a un amigo o familiar?</i>	55

1.CAPÍTULO I – INTRODUCCIÓN

1. JUSTIFICACIÓN

A medida que la implementación de tecnología para la venta aumenta, pequeños productores comienzan a enfrentarse a retos para mantenerse en el mercado. Los artesanos no son la excepción, por eso aparece la necesidad de ingresar al comercio electrónico para llegar a una audiencia que abarque más lugares, y así diversificar sus ventas al poseer un canal de ventas eficiente donde no se necesitan tener tiendas físicas, haciendo que los artesanos puedan posicionarse como marcas más profesionales y confiables, lo que puede atraer a más clientes.

A través del comercio electrónico, los artesanos pueden exhibir sus productos de manera visualmente atractiva y proporcionar información detallada sobre cada artículo. Esto permite a los clientes potenciales conocer mejor los artículos y tomar decisiones de compra informadas. Además, los compradores pueden acceder a la tienda en línea en cualquier momento y desde cualquier lugar, lo que aumenta la conveniencia y la accesibilidad de los productos.

2. PLANTEAMIENTO DEL PROBLEMA

En los medios digitales actuales, la venta y promoción de los productos a partir del comercio electrónico son un factor importante. Sin embargo, el mercado de productos artesanales, que puede abarcar desde joyería única hasta productos de decoración hechos a mano, ha enfrentado desafíos significativos en su adaptación al entorno en línea.

Por lo tanto, el problema que se abordará en esta investigación es: ¿Cuál sería la forma correcta para diseñar, desarrollar y gestionar con eficacia una página web que esté enfocada en la venta de productos artesanales para promover y facilitar la venta de estos productos en línea?

Y las siguientes preguntas: ¿Cómo se puede mejorar la gestión y venta de productos artesanales a través del comercio electrónico?

¿Cuáles son las mejores prácticas y estrategias de diseño web y experiencia del usuario que pueden aplicarse para destacar y promover eficazmente productos artesanales en una plataforma de comercio electrónico?

¿Cómo se pueden implementar los productos artesanales dentro del comercio electrónico?

3. ALCANCE

La plataforma permitirá a los artesanos exhibir y vender sus productos, gestionar su inventario e interactuar con sus clientes. Las pruebas que se realizarán usabilidad, pruebas de humo para la detección de fallos, pruebas unitarias para ver el correcto funcionamiento. La página web tendrá una interfaz fácil de entender para artesanos y compradores, implementando las mejores prácticas de comercio electrónico.

4. OBJETIVOS

GENERAL:

Desarrollar una página web que permita la venta y gestión del inventario de productos artesanales, lo que incluye la venta y control del inventario.

ESPECÍFICOS:

1. Identificar las necesidades del usuario relacionadas con la venta de productos.
2. Diseñar una aplicación web que facilite las ventas de los productos y el control del inventario.
3. Desarrollar una página web que permita administrar y vender productos de manera amigable.
4. Probar las funcionalidades de la aplicación de acuerdo con los requisitos y especificaciones definidos.

5. ANTECEDENTES

El desarrollo de páginas web de comercio electrónico es un proceso fundamental que comienza con la recopilación de requisitos y continúa con el diseño de un sitio web funcional y

atractivo. Una vez completado este proceso inicial, los desarrolladores de comercio electrónico crean el código backend necesario para que el sitio web tenga la funcionalidad correcta y permita a los clientes comprar productos. Finalmente, se procede con la sección que interactúa directamente con el usuario. Es importante destacar que las herramientas más recientes impulsan un desarrollo de comercio electrónico que transforma la manera en la que se venden los productos, haciendo que en la actualidad no se necesite una gran inversión para esto mismo, pudiendo hacerlo directamente desde nuestros hogares (Castañeda, 2012).

Se ha explorado la literatura existente, que abarca estudios de diversos autores/as como Smith (2019), García (2020), y Chen (2021), quienes han abordado problemáticas relacionadas con el diseño de plataformas de comercio electrónico, la gestión de inventario y la promoción de productos artesanales en línea.

Smith examina la experiencia del usuario en plataformas de comercio electrónico para artículos. El autor resalta tres aspectos cruciales: la usabilidad, donde la plataforma debe ser fácil de usar y navegar; la estética, que implica un diseño atractivo y visualmente agradable; y la personalización, ofreciendo opciones para adaptar la experiencia de compra a las preferencias individuales (Smith, 2019).

García se enfoca en estrategias de gestión de inventario para pequeñas empresas que venden productos en línea. Propone un modelo flexible basado en un seguimiento constante del inventario, la diversificación de proveedores y el establecimiento de niveles óptimos de inventario para evitar problemas de desabastecimiento y exceso de inventario (García, 2020).




Por otro lado, Chen explora distintas maneras para impulsar los productos en línea. Enfatiza en la importancia del marketing, las redes sociales, y las búsquedas de contenido optimizado. Sugiere utilizar las redes sociales para construir una comunidad, colaborar con influencers relevantes y crear contenido para mejorar la visibilidad en línea de los productos artesanales (Chen, 2021).

A continuación, se realizará un cuadro comparativo entre las páginas web de venta de productos artesanales ecuatorianas:

Cuadro Comparativo.

Tabla 1

Cuadro comparativo entre distintos tipos de páginas de venta de productos artesanales en Ecuador.

Nombre	Pimkay	ElQuinde	Ecuador Store
Logo			
Como operan	<p>Los productos comprados en la tienda en línea se envían con rapidez. El tiempo de entrega es de 2 a 3 días. Para proceder al envío, el pago debe haberse realizado previamente. Además, tiene una tienda física en Quito.</p>	<p>Los artículos comprados en la tienda online se envían mediante solicitudes a la tienda física, la cual envía los productos seleccionados en el momento en que el cliente lo desee. Además, ofrecemos tours.</p>	<p>La tienda online se basa en mostrar productos de artesanos directos. Donde el objetivo es apoyarlos e incentivarlos a seguir creando nuevas artesanías. La tienda se basa en tres categorías de artesanías que son: Paja toquilla, Tagua y Alpaca. Se pueden comprar y enviar a distintos países, además de comprar en grandes cantidades, aunque el mayor problema es el tiempo de</p>

			respuesta que tiene la página web frente a las otras dos.
Tipo de productos	Principalmente, artesanías propias.	Variedad de productos, pero directamente desde la tienda presencial.	Artesanías directas del artesano.
Precios	Mayormente elevados.	Variados.	Mayorista y variados.

Nota. El cuadro comparativo de páginas ecuatorianas (2024), Elaboración propia

2.CAPÍTULO II - MARCO TEÓRICO

2.1. El comercio electrónico

El comercio electrónico es la actividad de comprar y vender productos o servicios a través de internet, en la cual se obtiene a partir de compras realizadas por este mismo medio. Esta actividad ha experimentado un crecimiento exponencial en los últimos años, debido a la comodidad y accesibilidad que ofrece a los consumidores. (OECD, 2019; UNCTAD, 2018)

Una plataforma de comercio electrónico es un sistema en línea en forma de sitio web o aplicación móvil que permite a los usuarios comprar y vender bienes y servicios en línea. Estas plataformas pueden ser generales, como Amazon o Mercado Libre, o especializadas. (Amazon, n.d.; Mercado Libre, n.d.)

La experiencia del usuario (UX) desempeña un papel crucial en el éxito de cualquier plataforma de comercio electrónico (Hassenzahl, 2003).

El diseño de la interfaz de usuario debe ser intuitivo, atractivo y reflejar la autenticidad y singularidad de los productos ofrecidos. Se tiene que prestar atención a la forma de navegación, la presentación de productos, las imágenes de alta calidad y la facilidad de compra para garantizar una experiencia de usuario satisfactoria que fomente la confianza y la fidelidad del cliente. (Norman, 2013).

2.2 Metodología ágil de desarrollo de software

Los métodos ágiles para desarrollar una aplicación de software (MAD) han revolucionado la industria del software en las últimas décadas. Las metodologías tienen un enfoque iterativo e incremental para la gestión y desarrollo de software que enfatiza la flexibilidad, la colaboración con el cliente y la respuesta rápida a los cambios, estas han evolucionado desde enfoques estructurados y pesados hasta prácticas más ligeras y flexibles. (Canós et al., 2003).

En lugar de esperar a completar el proyecto en su totalidad, las metodologías ágiles abogan por la entrega de software funcional en pequeñas partes a lo largo del proyecto. Esto permite obtener feedback del cliente de manera temprana y realizar ajustes en el curso del desarrollo (Beck et al., 2001).

Las metodologías ágiles reconocen que los requisitos del software pueden cambiar con el tiempo y que la incertidumbre es inherente a cualquier proyecto. En lugar de intentar predecir y controlar todos los aspectos del proyecto, se busca crear un entorno flexible donde se pueda adaptar el plan a medida que surgen nuevos cambios (Highsmith, 2009).

El Manifiesto Ágil, creado en 2001, establece los valores y principios fundamentales de este enfoque.

Para Beck et al. (2001), los principios esenciales que engloban las MAD son los siguientes:

- Valorar más a las personas y como interaccionan ante cualquier otro medio.
- Entregar software funcional, en lugar de dar una entrega final.
- Colaborar con el cliente y los stakeholders a lo largo del proyecto.
- Adaptarse a los cambios.

Prototipos

La metodología de Prototipos es una estrategia dentro del desarrollo ágil que emplea un enfoque iterativo. Su proceso se basa en la fabricación de prototipos sucesivos del sistema para obtener feedback del usuario y realizar mejoras incrementales, permitiendo visualizar una versión preliminar.

Existen diferentes tipos de prototipos, como prototipos de baja fidelidad, prototipos de alta fidelidad, prototipos desechables y prototipos evolutivos (Sommerville, 2016).

1. **Prototipos de baja fidelidad:** Están basados en cómo sirve el sistema y no en la apariencia final. Se utilizan para obtener una idea general del sistema y para probar la viabilidad de las funcionalidades. Estos prototipos suelen ser simples y se crean con instrumentos de bajo costo, como bocetos, diagramas de flujo o herramientas de prototipado rápido.
2. **Prototipos de alta fidelidad:** Se enfocan en la apariencia final del sistema y en la experiencia del usuario. Se utilizan para obtener feedback del usuario sobre la estética del sistema y la facilidad de uso. Estos prototipos suelen ser más complejos y se crean con

instrumentos de alto costo, como software de diseño gráfico o herramientas de prototipado interactivo.

3. **Prototipos desechables:** Se generan para obtener una idea general del sistema y no se reutilizan en las etapas siguientes del desarrollo. Estos prototipos suelen ser simples y se producen con herramientas de bajo costo, como bocetos o diagramas de flujo.
4. **Prototipos evolutivos:** Se van refinando a lo largo del proceso de desarrollo y pueden servir como base para la versión final del sistema. Estos prototipos suelen ser más complejos y se crean con herramientas de alto costo, como software de diseño gráfico o herramientas de prototipado interactivo.

Fases de la metodología de prototipos

Las fases de la metodología de prototipos son: definición de objetivos, diseño del prototipo, implementación del prototipo, evaluación del prototipo, refinamiento del prototipo y repetición del ciclo (Pressman, 2010).

1. Establecimiento de objetivos:

Durante este paso se delinear los objetivos del prototipo, así como las características que debe abarcar. Es crucial tener una comprensión clara del problema a resolver y las exigencias del usuario.

2. Diseño del prototipo:

En esta fase se diseña la interfaz de usuario y la funcionalidad del prototipo. Se debe tomar en cuenta la usabilidad, la accesibilidad y la estética del sistema.

3. Implementación del prototipo:

En esta fase se implementa el prototipo utilizando herramientas de desarrollo rápido. Se debe elegir una herramienta que permita crear un prototipo funcional con un bajo costo y esfuerzo.

4. Evaluación del prototipo:

En esta fase se presenta el prototipo al usuario para obtener feedback. Se debe observar cómo el usuario interactúa con el sistema y recopilar sus opiniones y sugerencias.

5. Refinamiento del prototipo:

En esta fase se realizan mejoras al prototipo con base en el feedback del usuario. Se pueden modificar las funcionalidades, la interfaz de usuario o la estética del sistema.

6. Repetición del ciclo:

Las fases 2 a 5 se repiten hasta que se obtenga un prototipo que satisfaga las necesidades del usuario. El número de iteraciones dependerá de la complejidad del sistema y de la precisión que se requiera en el prototipo.

2.3 Herramientas

2.3.1 Base de Datos

Una base de datos consiste en una agrupación organizada de datos o información estructurada, habitualmente almacenada de forma sistemática para su gestión y análisis. Un sistema de gestión de bases de datos (DBMS) administra la base de datos, facilitando el acceso, la gestión y la consulta de los datos (Rawat & Purnama, 2021).

Las bases de datos son fundamentales para operaciones como almacenar, recuperar, agregar y eliminar información, y suelen utilizar un lenguaje de consulta estructurada (SQL) o lenguaje no estructurado (NoSQL) (Rawat & Purnama, 2021).

Las bases de datos relacionales son estructuras que almacenan información, las cuales, como su nombre lo indica, tienen una estructura, la cual se encarga de estructurar los datos en tablas interrelacionadas mediante el uso de claves primarias y foráneas, además de que almacenan la información en tablas que poseen filas y columnas que permiten tener una mayor organización y acceso a la data. El lenguaje que se utiliza para acceder, modificar y administrar la información de la base es SQL. Las filas de la tabla representan registros individuales, y las columnas representan los atributos de esos registros (Cabello, 2010).

MySQL es un sistema de gestión de bases de datos relacionales (SGBD) de código abierto y gratuito. Haciéndolo uno de los más populares del mundo, utilizado por millones de empresas e individuos para almacenar y gestionar datos. MySQL utiliza un modelo cliente-servidor que

permite el poder manejar una gran cantidad de bases de datos relacionados de una manera flexible y confiable (MySQL, A. B., 2001).

2.3.2 Backend

El backend, también conocido como “lado del servidor”, es la parte de una aplicación web o sitio web que no es visible para el usuario final, es decir, es la columna vertebral que ayuda a soportar los componentes del usuario y recibir peticiones. Lo que nos hace ver que es lo primordial se encarga de la lógica de la aplicación, la gestión de datos, la seguridad y la comunicación con otros sistemas que se desarrolla para poder tener un trabajo en completas condiciones y tenga lo suficiente este se maneja mediante peticiones que se envían desde la parte frontal (Van Meer, 2017).

Node.js es una plataforma de código abierto para la ejecución de aplicaciones multiplataforma, desarrollada sobre la base del motor de Google Chrome (Node.js, s/f).

Permite ejecutar código JavaScript fuera del navegador, lo que lo hace ideal para el desarrollo de aplicaciones del lado del servidor y de red escalables; este es conocido por ser un modelo de entrada/salida no bloqueante y orientado a eventos, lo cual permite que sea eficiente y adecuado para aplicaciones en tiempo real (Flanagan & Snook, 2023).

Express.js se destaca como un framework de aplicaciones web en Node.js, siendo ampliamente adoptado por su enfoque minimalista y adaptable, lo cual simplifica el desarrollo de aplicaciones web sólidas y escalables; este proporciona una base para organizar el código, gestionar las solicitudes y respuestas HTTP y administrar las funcionalidades esenciales de las aplicaciones web (Express - Node.js web application framework, s.f.).

Un middleware es una herramienta de software que funciona como un mediador entre diferentes aplicaciones o componentes de software en una red distribuida (Bakken, 2001). Funciona como una capa de traducción que facilita la comunicación y el intercambio de datos entre diferentes sistemas, uno de estos puede ser bcrypt, que es una biblioteca que tiene Express la cual nos permite poder utilizar un método de encriptación que funciona para tener contraseñas cifradas para su uso (Bakken, 2001).

La biblioteca bcrypt realizada en Javascript ofrece funcionalidades de las más destacadas se incluyen el cifrado de contraseñas, que convierte las contraseñas en cadenas de caracteres ilegibles, la verificación de contraseñas al iniciar sesión, la protección contra diversos tipos de ataques y la incorporación de un valor aleatorio conocido para añadir una capa adicional de seguridad (Npm: Bcrypt, s/f).

REST API (Interfaz de programación de aplicaciones de transferencia de estado representacional) es una interfaz de programación que se ajusta a los principios de la arquitectura REST y permite la comunicación entre aplicaciones y sistemas distribuidos (A., & Ranga, V, 2019, p. 3).

2.3.3 FrontEnd

El frontend, comúnmente denominado como el "lado del cliente", constituye la interfaz visible y la parte con la que interactúa el usuario final en una aplicación web o sitio web. Esta área abarca desde el diseño visual hasta el contenido y la interactividad que se ofrece. Esencialmente, el frontend es la puerta de entrada para el usuario, proporcionándole una experiencia visual y funcional que facilita la interacción con la aplicación o el sitio web. En esencia, el frontend representa la interfaz gráfica de usuario (GUI) que permite la interacción con el sistema (Duckett, 2020).

JavaScript es un lenguaje de programación de alto nivel y dinámico, interpretado comúnmente empleado para el desarrollo web en el lado del cliente, es decir, aquel que se ejecuta en el usuario (Flanagan, 2011, p. 122). JavaScript también se puede usar para crear aplicaciones web del lado del servidor (Flanagan, 2011, p. 123).

React es una biblioteca de JavaScript con Node.js de código abierto creada por Facebook, actualmente conocida como Meta, para crear interfaces que permitan al usuario ser eficiente y poder interactuar correctamente (React, s/f).

Se utiliza principalmente en el desarrollo de aplicaciones web del lado del cliente, siendo una de las más conocidas Vite, un entorno que permite ejecutar de una manera más sencilla. Este entorno existe tanto para JavaScript como para TypeScript. Además, React tiene frameworks para

el apartado gráfico, como Bootstrap y TailwindCSS, ambos de código abierto, los cuales permiten la edición directamente desde las clases de los objetos sin la necesidad de entrar al CSS que es el apartado de edición (React, s/f).

3. CAPÍTULO III PLANIFICACIÓN

Las herramientas que se utilizó para desarrollar el sitio web son una base de datos MySQL, un backend creado con Express y un frontend implementado en React usando Vite como paquete. Esta selección se basa en criterios de rendimiento, confiabilidad y escalabilidad.

Se eligió MySQL por su capacidad en la gestión eficientemente de una cantidad amplia de datos y su compatibilidad con muchas plataformas diferentes.

Se eligió Express como plataforma de servidor debido a su flexibilidad, velocidad y facilidad para crear potentes aplicaciones web, además de que permite la creación de una API REST de una manera más sencilla y simple de utilizar.

Finalmente, se utilizó React con Vite en la capa de presentación debido a su rendimiento optimizado, diseño modular y habilidad para desarrollar interfaces de usuario dinámicas y altamente útiles.

Este conjunto de herramientas está perfectamente integrado y proporciona una solución tecnológica completa y confiable para el desarrollo de la página web. A partir de las encuestas realizadas se llegó a las siguientes conclusiones basándonos en la necesidad y requerimientos de los usuarios:

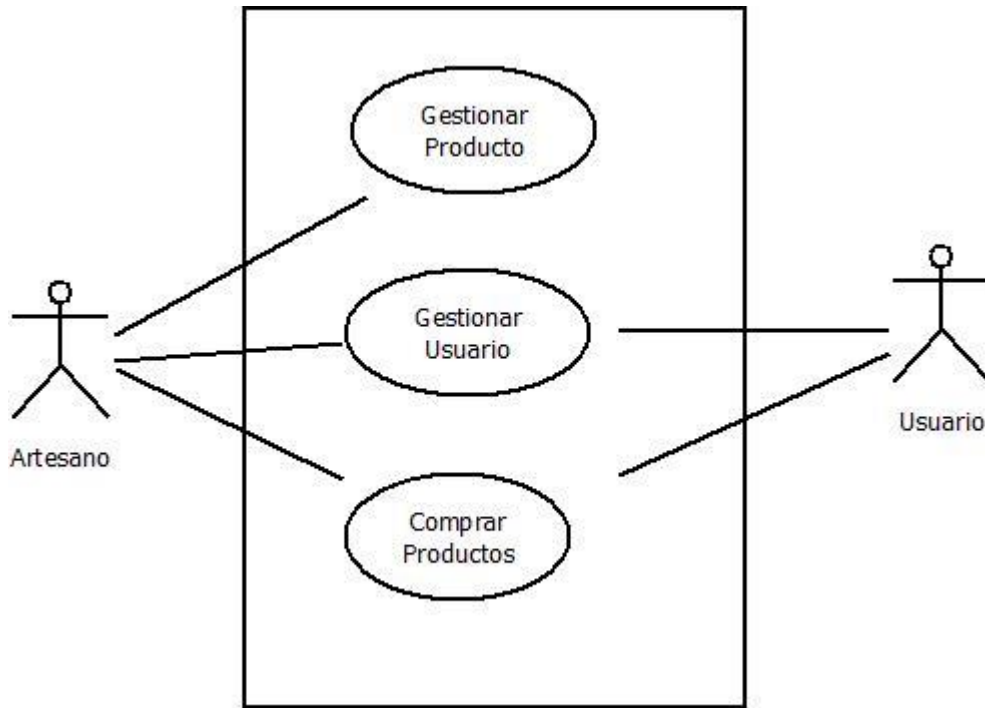
3.1. Requerimientos Funcionales para el comercio electrónico

Para el levantamiento de requerimientos funcionales para la página de venta de productos artesanales, se mantuvieron conversaciones extensas junto con un seguimiento diario para poder identificar las características esenciales que debería tener. Como resultado, se obtuvieron los siguientes casos de uso, los cuales representan de manera detallada las funcionalidades principales.

Diagramas de Casos de Uso

La **Figura 1** indica las principales funcionalidades del sistema junto con sus actores. En esta figura, se identifican los dos tipos de usuarios que utilizan el sistema y las funcionalidades principales que cada uno de ellos puede emplear.

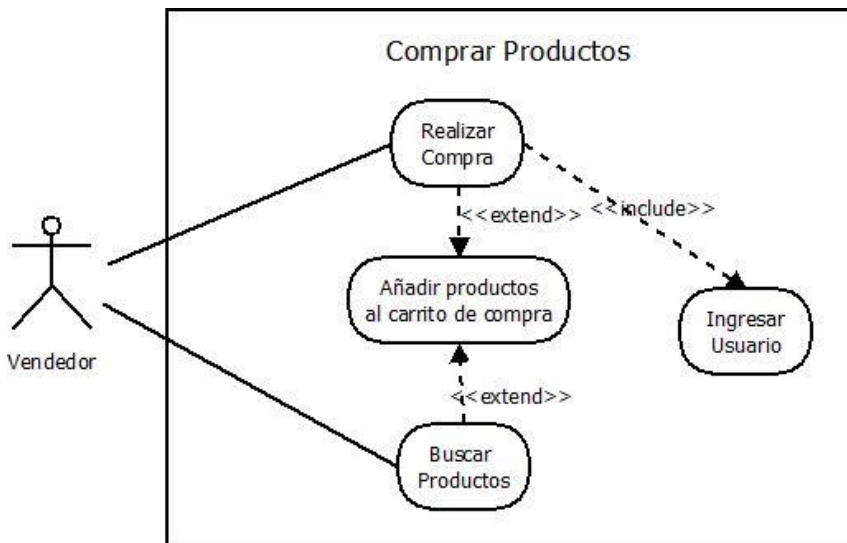
Figura 1
Caso de Uso General



Nota. El caso de uso general muestra una vista simplificada de las funcionalidades del sistema (2024), Elaboración propia

Caso de Uso Comprar Productos

Figura 2
Caso de uso: Comprar Productos



Nota. El caso de uso compra productos muestra una vista de las funcionalidades en el momento de realizar las compras (2024), Elaboración propia

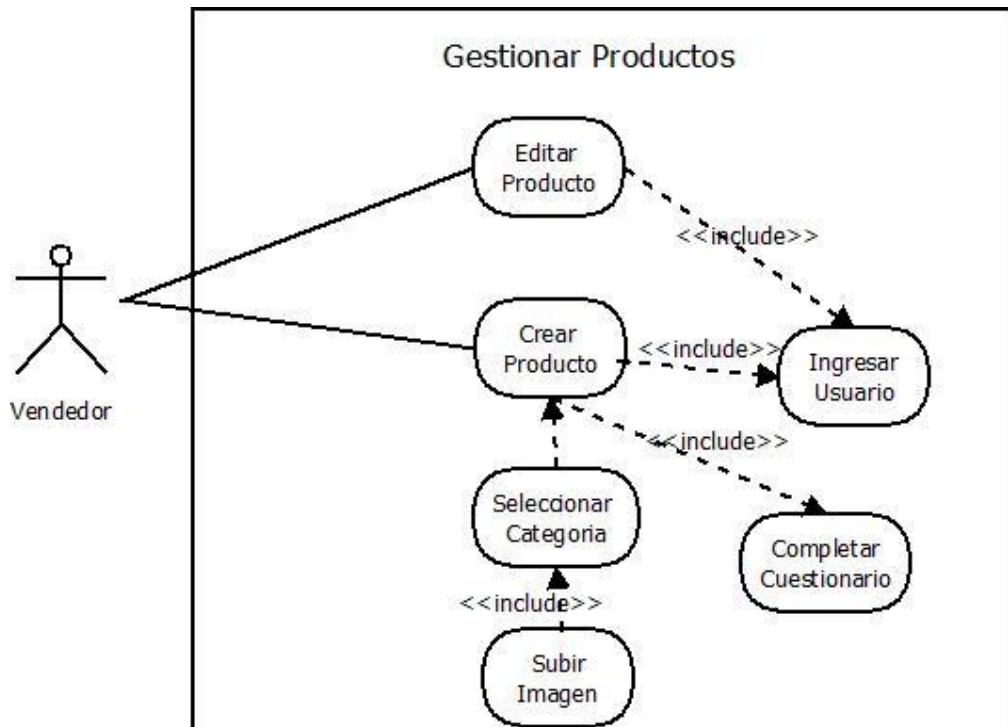
El Caso de Uso indicado en la **Figura 2** "Comprar Productos" tiene como objetivo principal permitir a los compradores poder seleccionar y añadir productos a su carrito de compras, facilitando así el proceso de compra. Como precondition, se requiere que el usuario haya iniciado sesión previamente. Una vez que los productos están en el carrito, el usuario puede visualizarlos y proceder a la compra. El proceso implica seleccionar un producto, añadirlo al carrito, visualizar los productos en el carrito, presionar el botón de compra, revisar la página final de la compra y confirmarla.

En caso de que el usuario no esté con una sesión activa, el sistema lo redirigirá a la página de inicio antes de comprar productos que estén dentro del carrito, como flujo alternativo. Asimismo, si un producto no está disponible en stock, el sistema notificará al usuario para informarle sobre la situación.

Caso de uso Gestionar Productos

Figura 3

Caso de uso: Gestionar Productos



Nota. El caso de uso gestionar productos muestra una vista de las funcionalidades en el momento de gestionar los productos para su compra (2024), Elaboración propia

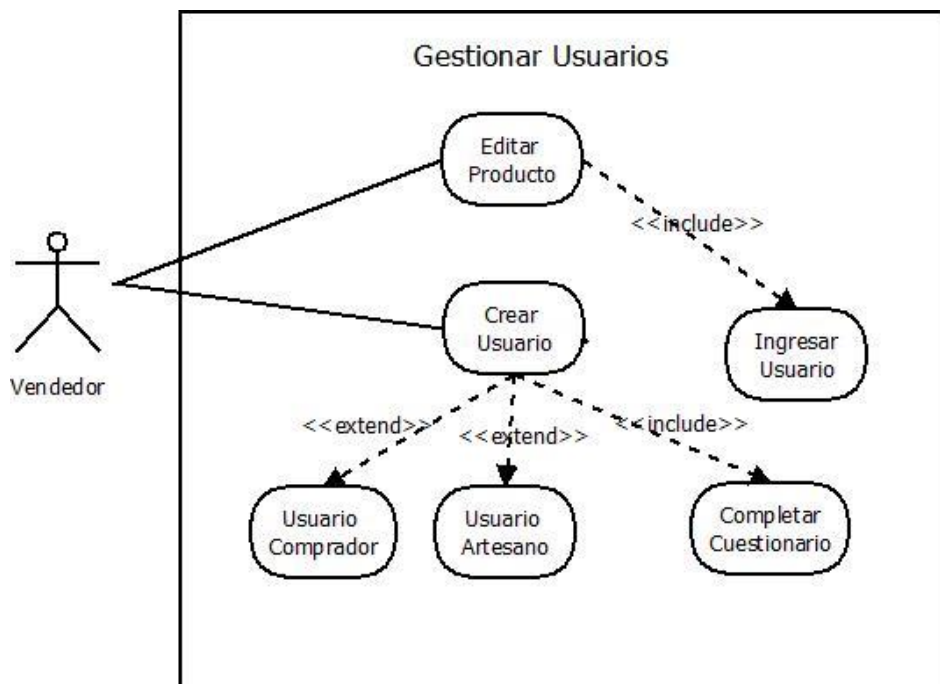
El caso de uso "Gestionar Producto" que se puede observar en la **Figura 3** tiene como objetivo habilitar a los usuarios registrados, específicamente artesanos, para que puedan insertar y administrar sus productos artesanales en la página web. Para utilizar esta funcionalidad, el usuario debe estar registrado como artesano y haber iniciado sesión en la página web. Además, necesita tener acceso a internet y un dispositivo compatible. La página web debe tener activada la opción para insertar productos artesanales. Una vez cumplidas estas condiciones, el usuario puede insertar un nuevo producto artesanal, editar la información de un producto existente o eliminar uno. Todos los productos artesanales se podrán ver para cualquier usuario adentro de la página web.

En el proceso, el artesano accede al formulario de inserción de productos artesanales mediante un enlace o botón específico en la página web. Luego, completa el formulario con información obligatoria como nombre, descripción, precio, categoría, fotos y stock del producto. Posteriormente, sube las fotos del producto y, tras revisar la información y las imágenes, publica el producto para que sea visible. Si llega a darse el caso de que el artesano intente subir fotos que no cumplen con los requisitos de formato, el sistema impedirá este paso.

Caso de uso Gestionar Usuario

Figura 4

Caso de uso: Gestionar Usuario



Nota. El caso de uso gestionar productos muestra una vista de las funcionalidades en el momento de gestionar tu usuario (2024), Elaboración propia

El caso de uso "Gestionar Usuario" que se puede observar en la **Figura 4** tiene como objetivo facilitar que nuevos usuarios se registren en la página web, lo que les permitirá comprar productos artesanales y gestionar su cuenta, incluida la opción de desactivarla si así lo desean. Para ello, el usuario necesita tener acceso a internet y un dispositivo compatible, mientras que la página web debe tener la opción de registro de usuarios activada. Una vez cumplidas estas condiciones, el nuevo usuario accede al formulario de registro a través de un enlace o botón específico en la página web. En este formulario, completa los campos obligatorios, como nombre completo, usuario, contraseña, correo electrónico válida, contraseña y confirmación de contraseña. Para que se pueda crear la cuenta, es necesario que la clave sea segura y su usuario no esté en existencia. Finalmente, al hacer clic en el botón "Crear cuenta", si no existen contratiempos, el proceso de registro se completa, y el usuario puede iniciar sesión con su nuevo usuario creado y la clave segura que fue colocada, lo que le permite acceder a las funcionalidades de la página web dependiendo del tipo de usuario y realizar compras de productos artesanales.

Perfil del Usuario:

- Artesano: Cada artesano registrado debe tener acceso a un apartado exclusivo mediante su usuario y contraseña, donde pueda ver y administrar sus productos, además de acceder a un apartado para visualizar el balance actual, este también posee las funcionalidades del Artesano.
- Compradores: Deben tener acceso a la página para añadir productos a su carrito, gestionar su perfil y dejar comentarios sobre los productos.
- Anónimo: Usuarios visitantes que no tendrán privilegios más allá de la visualización de la página.
- Servicio de Control del Artesano:
 - La página dispondrá de un apartado para agregar, editar y visualizar los productos.
 - Los artesanos podrán revisar el valor de sus ventas filtradas por fechas.

3.2. Requerimientos No Funcionales de la Página Web

- Apartado Gráfico: La página de comercio electrónico debe ofrecer una interfaz amigable que permita a los usuarios utilizarla de manera más sencilla.
- Usabilidad: La página web tendrá una interfaz lo más amigable posible para que el usuario comprenda todas las acciones que debe realizar de la manera más intuitiva posible.
Para lograrlo, se incluirán las siguientes características:
 - El idioma será en español.
 - El formato de cada página será apropiado.
 - Escalabilidad: La página web debe permitir acciones adicionales en el futuro sin perjudicar el funcionamiento actual.

3.3. Diseño Funcionales para el comercio electrónico

Una vez completado el proceso de levantamiento de requerimientos del sistema para los distintos usuarios y basándonos en las encuestas realizadas tanto a los usuarios artesanos como a otros usuarios, cuyos resultados se encuentran tanto en el Anexo A y Anexo B, es momento de definir la estructura que tendrá la página web.

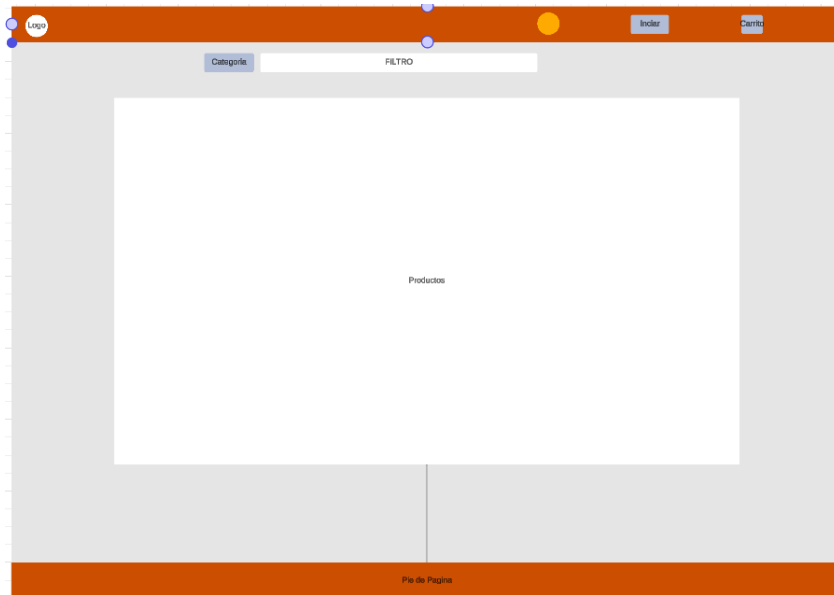
A partir de las encuestas realizadas tanto a usuarios como a artesanos, se ha identificado una estructura que proporcionará una idea clara del sitio en su fase final, destacando sus ventajas y facilitando la ejecución de las actividades correspondientes.

En la **Figura 5** se visualizará un pequeño diseño principal junto a la cromática que lleva cada elemento de la página web:

Página de Inicio:

Figura 5

Diseño disposición página web



Nota. Gráfico muestra una visual preliminar de la localización de los objetos de la página (2024),
Elaboración propia

Estándares de Colores

La página que conforma el programa estará diseñada para ser visualizada en cualquier navegador de internet, tanto en versión de escritorio como en dispositivos móviles.

Los colores elegidos para el diseño de la página web son:

Paleta de Colores:

Colores Primarios:

#ffffeb - Amarillo claro (Principal)

#f59e0b - Amarillo intenso (Destacado)

#451a03 - Marrón oscuro (Destacado)

Colores Secundarios:

#fef3c7 - Amarillo pálido (Secundario)

#fde68a - Amarillo brillante (Secundario)

#fcd34d - Amarillo medio (Secundario)

Colores Adicionales:

#F1F1F1 - Gris claro (Fondo)
#d97706 - Naranja (Acento)
#b45309 - Naranja oscuro (Acento)
#92400e - Marrón (Acento)
#78350f - Marrón medio (Acento)

A continuación, se proporciona una explicación exhaustiva de los usos de ciertos colores y sus respectivas funcionalidades:

Color Amarillo Claro (#ffffeb): Este tono se emplea estratégicamente para resaltar elementos de interacción críticos en la interfaz, tales como botones de acciones primarias incluyendo “Comprar” o “Agregar al carrito”. Su luminosidad sirve para enfatizar ofertas limitadas o promociones destacadas, facilitando así la identificación rápida por parte del usuario.

Color Amarillo Intenso (#f59e0b): La intensidad de este color se utiliza para atraer la atención del usuario hacia información de relevancia económica, como precios o descuentos significativos. Además, se aplica para resaltar productos de interés o aquellos que requieren una visibilidad prominente.

Color Marrón Oscuro (#451a03): Dada su capacidad para contrastar eficazmente con fondos de tonalidades claras, este color se destina para la tipografía de encabezados o títulos principales. Su uso asegura una lectura clara y directa, contribuyendo a la estructuración y comprensión del contenido.

Color Gris Claro (#F1F1F1): Este color se prefiere para fondos de página o secciones donde se busca que el contenido sobresalga sin competir visualmente. Proporciona una base neutra que favorece la concentración en los elementos de contenido.

Colores Secundarios y Adicionales: Estos colores complementan la paleta principal y se aplican con el propósito de enriquecer la experiencia visual. Se utilizan en detalles finos como bordes, iconografía y elementos de navegación, añadiendo así diversidad y cohesión al diseño global.

Secciones La página web contará con 3 secciones principales:

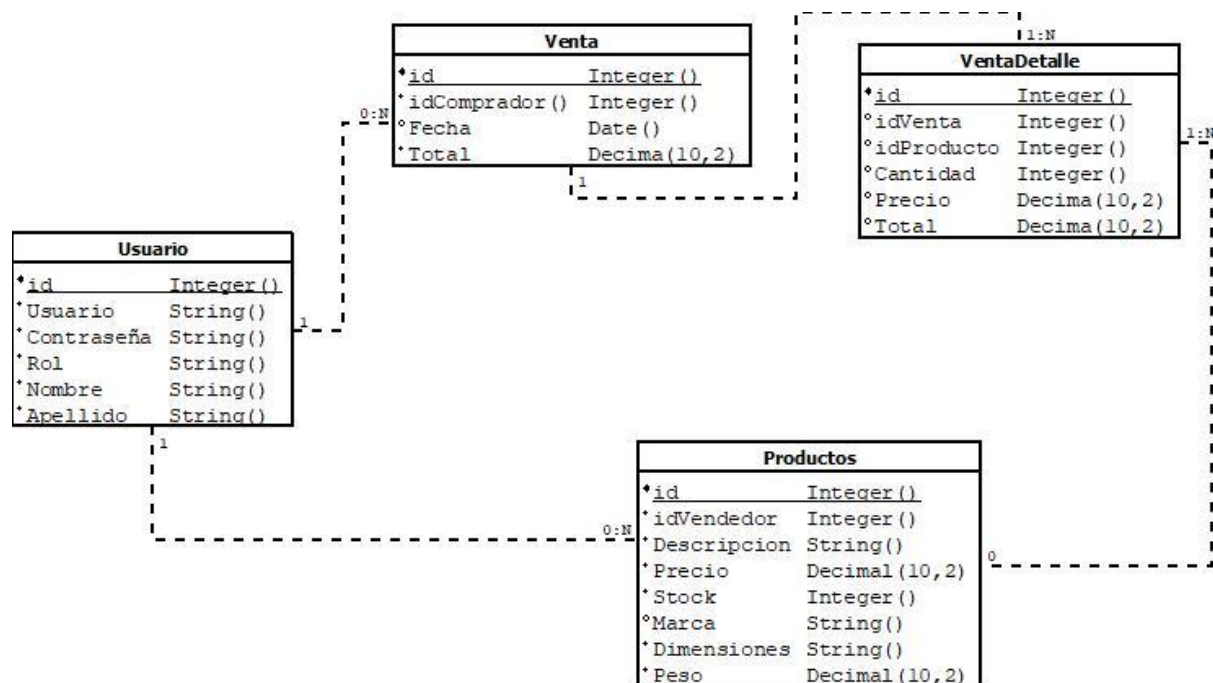
- Menú Principal: Aquí se encontrarán todos los módulos que componen la página web, como el logo, inicio de sesión, carrito de compras, y para usuarios artesanos, la sección 'Mis productos'.
- Sección Central: En esta parte se mostrará toda la información, desde formularios de inicio de sesión hasta productos tanto generales como del usuario.
- Pie de Página: En esta parte se mostrará toda la información adicional y utilitaria sin interrumpir el flujo del contenido principal

El diseño de la base de datos

Como se ilustra en la **Figura 6**, este diseño abarca la estructuración detallada de todas las entidades y relaciones necesarias para manejar productos, categorías, ventas, usuarios. Al crear una base de datos robusta y escalable, nuestro objetivo es no solo simplificar cómo se almacena y se recuperan los datos, sino también garantizar la protección y confidencialidad de la información.

Figura 6

Diseño base de datos



Nota. El grafico sobre la base de datos muestras las tablas principales para la página web (2024), Elaboración propia

4. CAPÍTULO IV - DESARROLLO

4.1 Desarrollo del primer avance del prototipo

4.1.1. *Diseño primer avance*

En el primer avance, lo que se realizó es la funcionalidad de venta de los productos, la cual es fundamental para el funcionamiento principal de la página. En esta etapa, se llegó tanto a la distinción de los usuarios como a la implementación de una secuencia lógica para la compra de los productos, las funcionalidades son las siguientes:

- Registro de usuario nuevo.
- Inicio de sesión.
- Compra de productos.
- Cierre de sesión.

Crear usuario

El inicio del primer avance se marcó con el desarrollo de una interfaz intuitiva, presentada en la **Figura 7** que simplifica el registro de nuevos usuarios al requerir información vital, como nombre, apellido, usuario, correo electrónico y una contraseña que cumpla con los estándares de seguridad, es decir, debe contener al menos 8 caracteres que incluyan números, letras y caracteres especiales. Esta función es crucial para incrementar la base de usuarios y garantizar una experiencia sin problemas desde el principio.

Figura 7
Crear Cuenta

TradArte

CREAR CUENTA

Nombre(s)

Apellido(s)

Usuario

Correo electrónico

Contraseña

Confirmar contraseña

Nota. La siguiente figura muestra una visualización sobre cómo se creará el usuario (2024),
Elaboración propia

Inicio de sesión

Tal como se puede apreciar en la **Figura 8**, se ha implementado una interfaz de inicio de usuario que posibilita el poder acceder utilizando sus credenciales previamente creadas. Este método es común y altamente confiable.

Figura 8
Iniciar Sesión

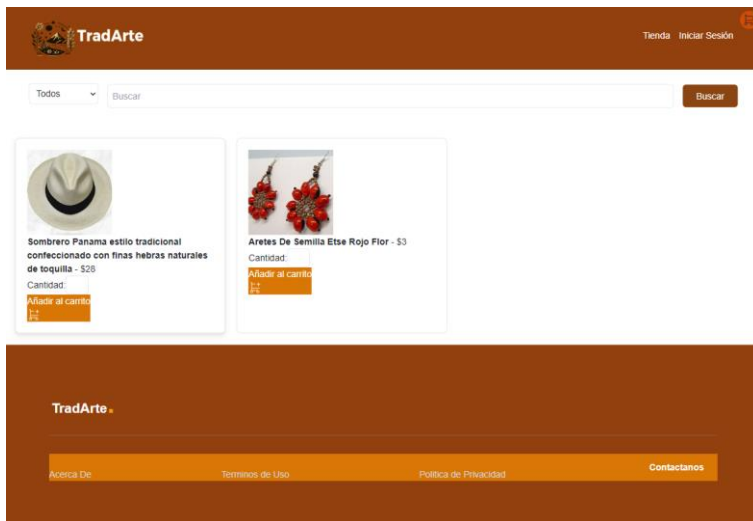


Nota. La figura muestra el iniciar sesión de la página (2024), Elaboración propia

Compra de productos

En cuanto a la funcionalidad de venta de productos, la **Figura 9**, indica como se diseñó una interfaz de usuario que capacita a los compradores para navegar en los productos, añadir productos artesanales dentro del carrito de compras y proceder con la muestra de la compra realizada. Además, se han implementado características adicionales, como filtros de búsqueda.

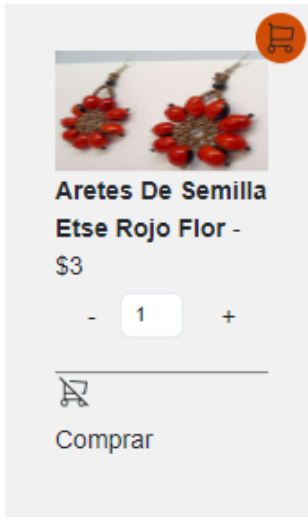
Figura 9
Compra de Productos



Nota. La siguiente figura muestra un preliminar de los productos (2024), Elaboración propia

La **Figura 10** muestra cómo se indica el producto en el apartado de previsualización antes del carrito de compras.

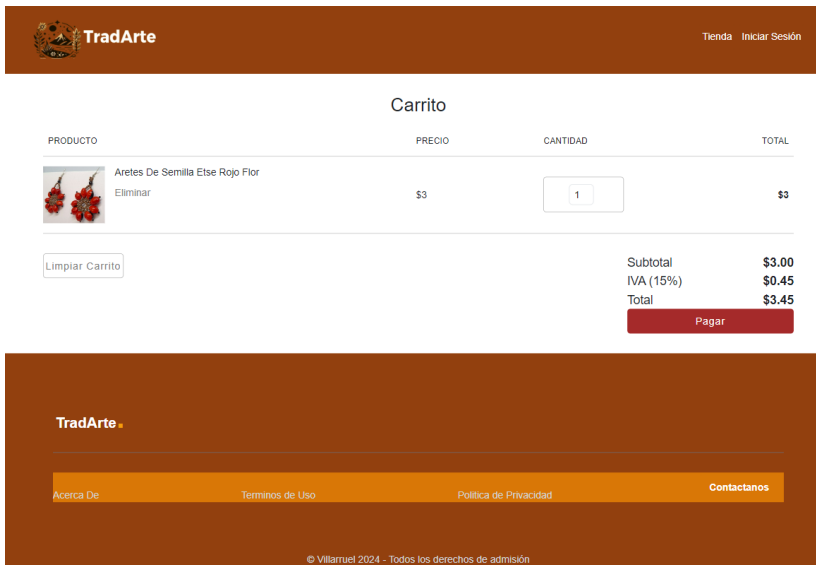
Figura 10
Previsualización carrito de compras



Nota. La figura muestra una previsualización para ver los productos añadidos (2024), Elaboración propia

La **Figura 11** muestra la visualización del carrito de compras para proceder a la compra de los productos.

Figura 11
Carrito de Compra



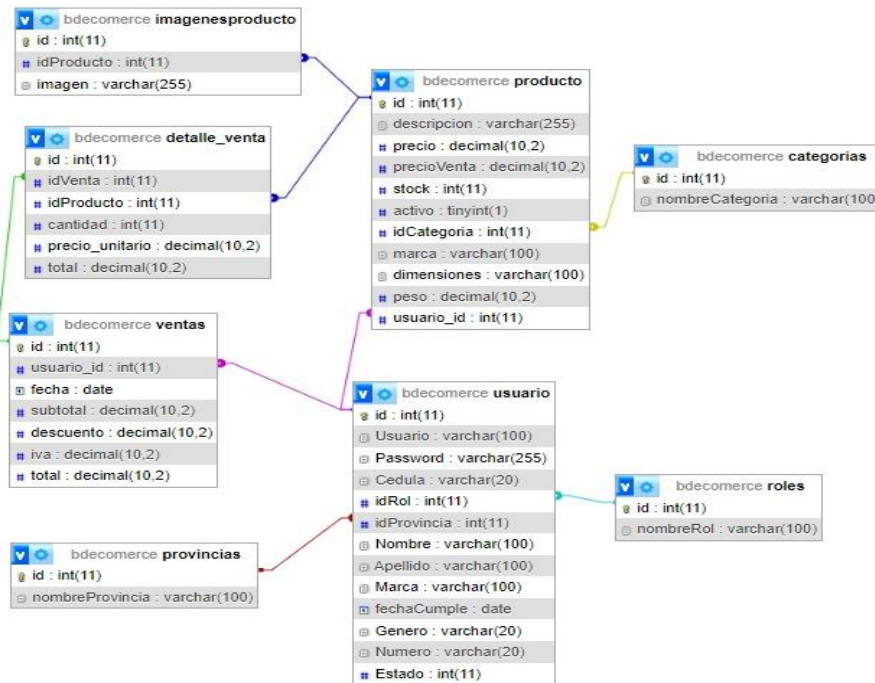
Nota. La figura muestra la visualización del carrito de compra (2024), Elaboración propia

Base de Datos

La **Figura 12** indica como está el primer modelo de la base de datos para el desarrollo de la página web.

Figura 12

Base de datos primer avance



Nota. La figura muestra la base de datos con sus tablas más específicas (2024), Elaboración propia

4.1.2. Codificación

Backend en Express

Para la implementación, se utilizó Node.js con Express, lo que permite una conexión más eficiente a la base de datos. Además, debido a la disponibilidad de múltiples bibliotecas, se pueden emplear diversas funcionalidades. Entre ellas se encuentran Morgan, que permite monitorear el envío de peticiones, y bcrypt, utilizado para encriptar las contraseñas. Estos detalles se pueden consultar en el Anexo C.

Principales funcionalidades

Se procede a indicar las principales funciones desarrolladas en el primer avance del prototipo, las cuales fueron desarrolladas en React con Vite, que es una herramienta de tercera generación para el desarrollo.

Se sugiere revisar el Anexo D para revisar de una manera más detallada la funcionalidad de insertar el usuario en el backend, incluyendo sus respectivos mensajes y validaciones.

La segunda funcionalidad que se desarrolló fue el carrito de compras, que se gestiona utilizando el almacenamiento local del navegador, conocido como LocalStorage. Este método de almacenamiento interno permite que los datos del carrito se mantengan disponibles mientras la sesión del usuario esté activa. Este aspecto se aborda en detalle en el Anexo E.

4.1.3 Retroalimentación

Durante el primer avance, se ha abordado la funcionalidad esencial de venta de productos, la cual constituye un principio importante para el funcionamiento de la página. En esta fase inicial, se ha logrado avanzar significativamente en el desarrollo de diversas características clave, incluyendo la distinción de usuarios y la implementación de una secuencia lógica para la compra de productos.

Las funcionalidades específicas que se han incorporado en esta etapa son las siguientes:

- **Registro de Usuario Nuevo:** Durante esta fase, se brinda la información necesaria para crear una cuenta personalizada, adaptada a cada tipo de usuario que utilizará la página web.
- **Inicio de Sesión:** Se ha desarrollado la funcionalidad que permite a los usuarios registrados iniciar sesión en sus cuentas.
- **Compra de Productos:** Se ha establecido una secuencia lógica y fluida que facilita a los usuarios la adquisición de productos disponibles en la plataforma. Esto incluye la navegación por el catálogo de productos y la selección de artículos deseados.
- **Cierre de Sesión:** Se ha implementado la opción para que los usuarios finalicen su sesión.

Durante la revisión del primer avance, se han identificado áreas clave para mejorar la implementación de la funcionalidad de venta de productos. A partir de los comentarios y sugerencias proporcionados, se proponen las siguientes mejoras:

- **Cantidad de Productos:** Se reconoce la importancia de implementar un sistema que permita gestionar adecuadamente la cantidad de productos que se pueda añadir a tu carrito de compra.
- **Pestaña "Pre-factura":** Se ha destacado la necesidad de incorporar una función de Pre-factura que brinde a los usuarios la capacidad de verificar de manera clara y concisa si la compra de un producto se ha realizado correctamente.
- **Cambio de Usuario:** Se reconoce la necesidad de que los usuarios compradores puedan convertirse en usuarios artesanos si desean vender sus productos conforme a sus necesidades.

La implementación de estas mejoras propuestas no solo permitirá abordar las áreas identificadas para optimización, sino que también contribuirá a fortalecer la calidad y la usabilidad de la plataforma de venta de productos.

4.2 Desarrollo del segundo avance del prototipo

4.2.1. Diseño segundo avance

En el segundo avance, lo que se realizó es la funcionalidad detallada del producto, además del control de los productos y ventas de los artesanos, lo cual le da un añadido a la página web.

En esta etapa, se realizó que los productos de cada uno de los artesanos se pudieran observar de una manera más ordenada con las siguientes funcionalidades:

- Visualizar productos detallados para comprar.
- Comentarios y conversación directa con el artesano.
- Insertar productos.

- Editar productos.
- Editar inventario.
- Visualizar ventas.

Visualizar productos específicos en la página web

Para dar inicio al segundo avance, se implementó una interfaz que permite al usuario comprador examinar los productos con mayor profundidad. En esta interfaz, se muestran detalles como dimensiones y peso, además de ofrecer la posibilidad de establecer contacto directo con el artesano para realizar compras más especializadas a través de WhatsApp. Este proceso se encuentra en la **Figura 13**.

Figura 13
Producto Detallado



Nota. La figura muestra una visualización más específica de los productos (2024), Elaboración propia

Visualizar comentarios de los productos

La **Figura 14** muestra los comentarios de los productos, acompañados de las calificaciones que han recibido.

Figura 14
Calificación productos



Nota. La figura muestra las calificaciones de los productos (2024), Elaboración propia

Productos Artesanos

La **Figura 15** indica cómo se ven los productos que tiene cada uno de los artesanos con una serie de colores dependiendo del stock que estos mismos tengan.

Figura 15
Tus Productos

Tus Productos

Añadir

Inventario	Descripción	Precio	Precio de Venta	Stock	Categoría	Marca	Dimensiones	Peso	Imagen
Inventario	Sombrero Panama estilo tradicional confeccionado con finas hebras naturales de toquilla	\$	\$28	1	Ropa	TodoArte	60cm-9cm-12cm	200gr	
Inventario	Aretes De Semilla Etse Rojo Flor	\$	\$3	4	Joyeria	TodoArte	2cm-1cm-5cm	10gr	
Inventario	Vidriera de colibrí	\$	\$45	9	Arte visual	TodoArte	5cm-4cm-10cm	100gr	

Nota. La figura muestra una visual de los productos de los artesanos (2024), Elaboración propia

La **Figura 16** presenta un diseño para el formulario de "Añadir tu producto".

Figura 16
Insertar producto

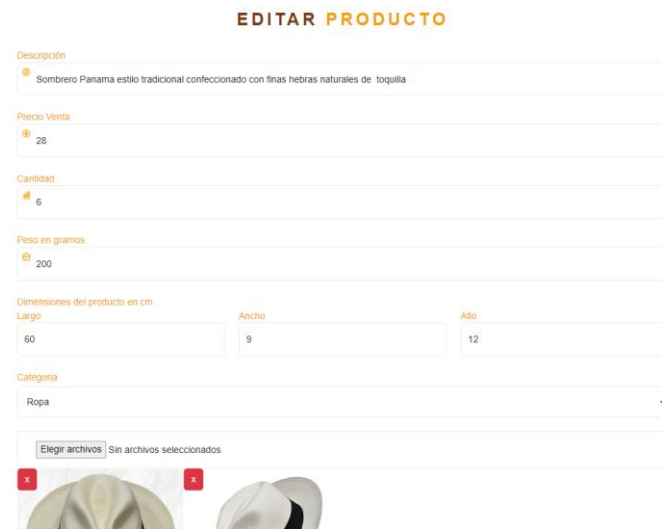


The screenshot shows the 'CREAR PRODUCTOS' (Create Products) form in the TradArte application. The form is set against a dark brown header with the TradArte logo. The form fields include: a text input for 'Descripcion'; a text input for 'Precio Creacion'; a text input for 'Precio de Venta'; a text input for 'Cantidad del producto'; a text input for 'Peso en gramos'; a section titled 'Dimensiones del producto' with three text inputs for 'Largo en cm', 'Ancho en cm', and 'Alto en cm'; a dropdown menu for 'Categoria' currently showing 'Joyeria'; and a file upload section with a button 'Elegir archivos' and the text 'Sin archivos seleccionados'. A thick orange horizontal line is positioned below the form.

Nota. La figura es una visualización de cómo se añadirá los productos para su venta (2024),
Elaboración propia

La **Figura 17** ilustra el proceso de edición de productos después de haber sido agregados. En esta pantalla, los usuarios tienen la capacidad de hacer cambios y actualizaciones en los detalles de los productos que han incorporado a la plataforma.

Figura 17
Editar producto



The screenshot shows the 'EDITAR PRODUCTO' (Edit Product) form in the TradArte application. The form is titled 'EDITAR PRODUCTO' in orange. The fields are pre-filled with the following data: 'Descripcion' contains 'Sombrero Panama estilo tradicional confeccionado con finas hebras naturales de toquilla'; 'Precio Venta' contains '28'; 'Cantidad' contains '6'; 'Peso en gramos' contains '200'; 'Dimensiones del producto en cm' has three inputs: 'Largo' with '60', 'Ancho' with '9', and 'Alto' with '12'; 'Categoria' dropdown is set to 'Ropa'; and the file upload section shows two image thumbnails of hats with red 'x' delete buttons. The 'Elegir archivos' button and 'Sin archivos seleccionados' text are also visible.

Nota. La figura muestra un producto que ya está ingresado en la página web (2024),
Elaboración propia

Inventario de los artesanos

La **Figura 18** presenta un modal del Inventario del producto, donde se facilita la capacidad de añadir o eliminar productos mediante una relación externa a la página web.

Figura 18
Inventario del Producto

Manejo Inventario Producto ×

Descripción:

Sombrero Panama estilo tradicional confeccionado con finas hebras naturales de toquilla



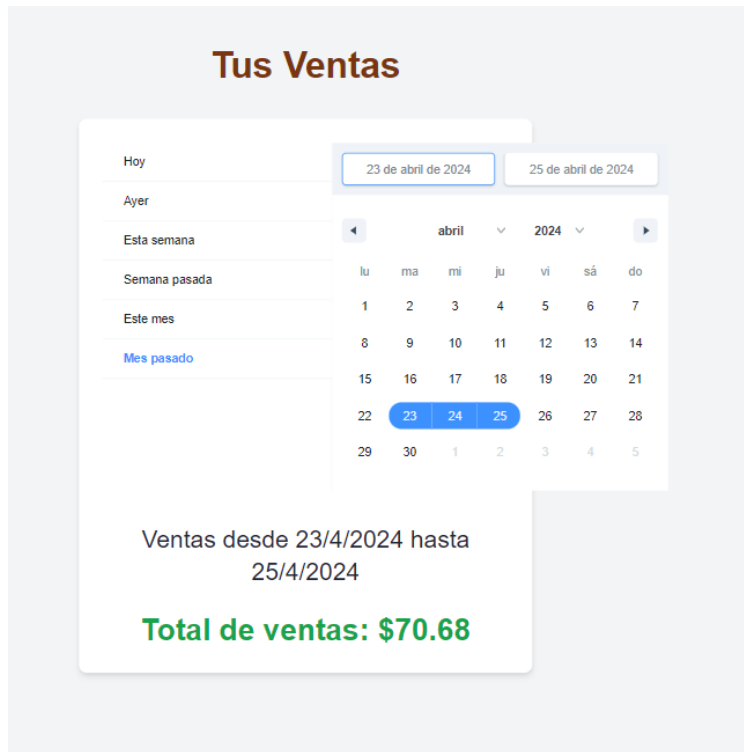
Cantidad

Tipo

Visualización de las ventas de los artesanos

La **Figura 19** ilustra cómo se muestran las ventas de los artesanos desde el día actual hasta la fecha seleccionada en el calendario. Esta pantalla permite visualizar las ventas según la fecha especificada, brindando una perspectiva de los ingresos en el período elegido.

Figura 19
Tus Ventas recientes por calendario



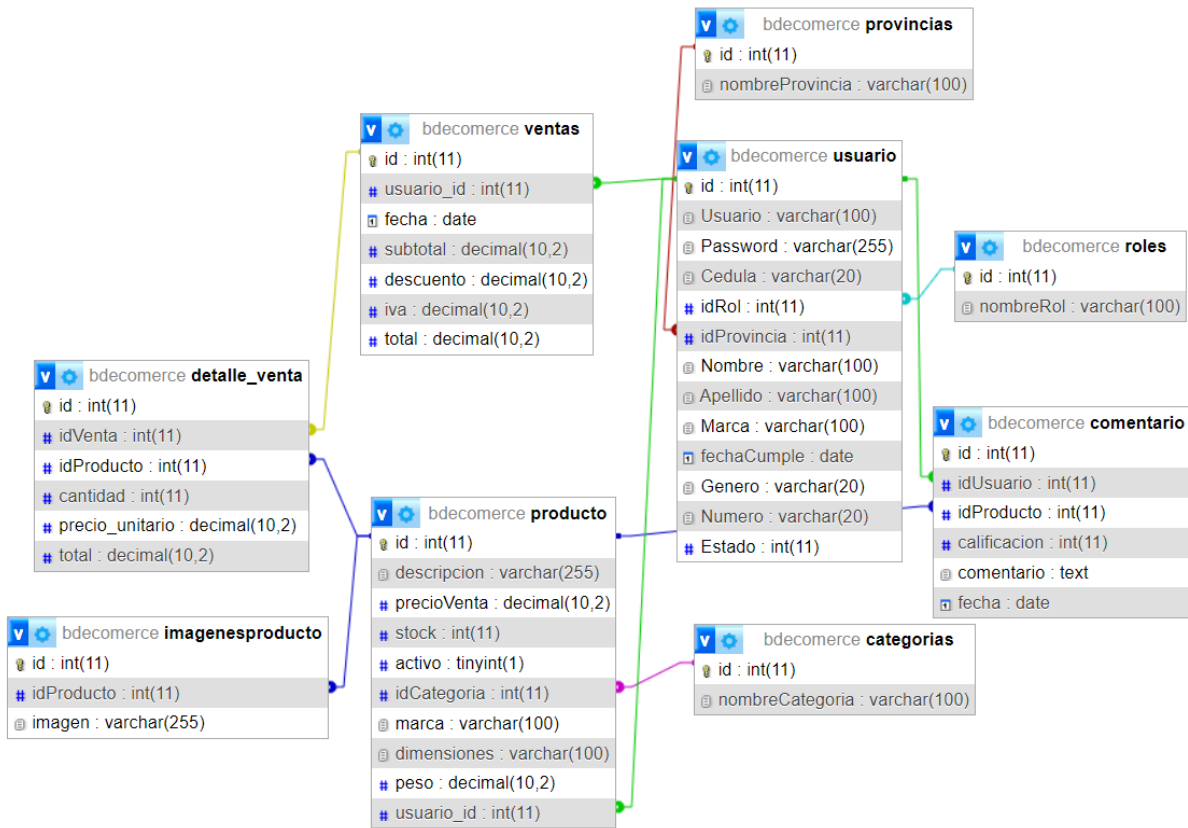
Nota. La figura muestra una interfaz de cómo se ven las ventas de los artesanos (2024),
Elaboración propia

Base de datos

La **Figura 20** presenta el segundo y último modelo de la base de datos, destacando cambios significativos que serán útiles para el desarrollo de nuevas funcionalidades, las cuales aportarán beneficios adicionales.

En esta base de datos para el segundo avance, las nuevas tablas han sido diseñadas específicamente para mejorar la gestión de la comunicación y la calificación de los productos. Estas mejoras permiten una interacción más efectiva entre los usuarios vendedores y compradores, así como una evaluación más precisa de los productos, lo que contribuye a una experiencia de usuario más enriquecedora y fiable.

Figura 20
Base de datos segundo avance



Nota. La figura indica la base de datos final (2024), Elaboración propia

4.2.2. Codificación

Principales Funcionalidades

Se procede a indicar las principales funcionalidades que tiene el apartado de Comentarios y panel del artesano que se desarrolló en la página web.

La primera funcionalidad que se implementó es la visualización detallada del producto, que además permitirá al usuario comunicarse directamente con el vendedor. Para ver el código correspondiente, consulte el Anexo F.

La segunda funcionalidad implementada fue la inserción de comentarios junto con la calificación de cada producto. Esta opción, disponible únicamente para los productos que el

usuario haya comprado, se puede observar en la pestaña de visualización detallada del producto. Los detalles completos se encuentran en el Anexo G.

En el Anexo H se presenta un fragmento de código que ilustra cómo insertar y editar productos artesanales utilizando la API de Imgbb, un servicio gratuito de alojamiento de imágenes. Este código permite a los usuarios subir imágenes de sus productos, gestionando tanto la inserción inicial como las ediciones posteriores (ImgBB, s/f).

La implementación de esta funcionalidad facilita una gestión más eficiente y visualmente atractiva de los productos en la plataforma.

Por último, en el Anexo I se describe un fragmento de código que muestra cómo gestionar el inventario para ventas exteriores o la elaboración de productos a partir de un modal. Este código permite a los usuarios registrar y actualizar información sobre su inventario.

4.2.3 Retroalimentación

Para el segundo avance, se han desarrollado e implementado tres funcionalidades específicas, además de incorporar las retroalimentaciones recibidas:

- **Visualización Detallada del Producto y Comunicación con el Vendedor:** La primera funcionalidad implementada permite una visualización detallada del producto y la posibilidad de comunicarse directamente con el vendedor.
- **Comentarios y Calificaciones:** La segunda funcionalidad permite la inserción de comentarios junto con la calificación de cada producto.
- **Inserción y Edición de Productos:** La tercera funcionalidad permite insertar y editar productos artesanales utilizando, además de poder utilizar la funcionalidad de inventario para venta del producto desde un apartado externo.

Durante la revisión del segundo avance, se han identificado áreas clave para mejorar la implementación de la funcionalidad de los productos. A partir de los comentarios y sugerencias proporcionados, se proponen las siguientes mejoras:

- Botón para deshabilitar producto: Se identificó la necesidad de permitir a los usuarios deshabilitar productos desde su lista de productos visibles.
- Notificaciones al usuario artesano: Se detectó la necesidad de informar al artesano, mediante una nueva pestaña, cuando sus productos han sido comprados recientemente.

La implementación de estas mejoras propuestas no solo permitirá abordar las áreas identificadas para optimización, sino que también contribuirá a fortalecer la calidad de la plataforma de venta de productos.

4.3 Desarrollo del avance final del prototipo

4.3.1. Diseño final

En el tercer y último avance del proyecto, se están introduciendo mejoras significativas en la plataforma, enfocadas en la gestión integral de las cuentas de usuario. Ahora, los usuarios tienen la capacidad de realizar acciones clave:

- Visualizar detalles relevantes de su cuenta.
- Editar su información personal.
- Cambiar su contraseña.

Estas funcionalidades se suman a las características previamente implementadas, creando un entorno más completo y personalizado para los usuarios. Con estas adiciones finales, se refuerza la experiencia del usuario al proporcionar un conjunto completo de herramientas para gestionar su cuenta de manera segura y realizar compras de forma eficiente, culminando así en una plataforma robusta y satisfactoria para todos los usuarios.

Visualizar tu usuario

La **Figura 21** presenta la interfaz visual que experimenta el usuario una vez ha iniciado sesión en la plataforma. En esta vista, se muestran claramente los botones que permiten editar información personal, cambiar la contraseña y eliminar la cuenta.

Figura 21
Tu información personal

Información del Usuario

Marca: TodoArte
Número: 0995364821
Usuario: GiGiVilla
Nombre: Gisela
Apellido: Garcia
Género: Femenino
Fecha de Cumpleaños: 7/12/1970
Provincia: Pichincha

[Desactivar Cuenta](#) [Editar](#) [Cambiar Contraseña](#)

Nota. La figura muestra el diseño central del usuario para ver su información (2024),
Elaboración propia

Editar Usuario

La Figura 22 presenta un modal que permite al usuario editar su perfil, ajustando los aspectos que considere necesarios para su actualización.

Figura 22
Editar tu usuario

Información del Usuario

Marca: VillaTienda
Número: 0961935197
Usuario: leovillarruel13
Nombre: Leonardo
Apellido: Villarruel
Género: Masculino
Fecha de Cumpleaños: 14/11/2002
Provincia: Pichincha

[Desactivar Cuenta](#) [Editar](#) [Cambiar Contraseña](#)

Nota. La figura muestra el diseño de un modal para que el usuario pueda editar su información (2024), Elaboración propia

Cambiar Contraseña

El proceso para cambiar la contraseña se inicia con la introducción del usuario y el correo electrónico que tiene la cuenta. Una vez ingresados estos datos, se envía un correo electrónico en el cual se encuentra un código de 4 dígitos que sirve para restablecer su contraseña. Al ingresar el código, el usuario es redirigido a una página donde puede establecer y confirmar una nueva contraseña. Una vez completado este paso, la contraseña se actualiza y el usuario puede iniciar sesión con sus nuevas credenciales.

A continuación, se mostrarán las figuras de cada uno de los diseños para este proceso.

La **Figura 23** muestra la primera pestaña que es para recuperar contraseña en la cual tenemos que ingresar el usuario y correo de la cuenta con la contraseña olvidada.

Figura 23

Recuperar contraseña olvidada



El formulario muestra el título "RECUPERAR CONTRASEÑA" en letras grandes y azules. Debajo hay dos campos de entrada: "Usuario" con un ícono de persona y "Correo electrónico" con un ícono de correo. Un botón azul con el texto "ENVIAR INSTRUCCIONES" está centrado. Al final, hay dos enlaces: "¿Ya tienes cuenta? Ingresar" y "¿No tienes cuenta? Regístrate".

Nota. La figura muestra el diseño para poder recuperar la contraseña de una cuenta en la cual se olvidó (2024), Elaboración propia

La **Figura 24** muestra la pantalla en la cual debes colocar el código de cuatro dígitos que se es enviado al correo del usuario olvidado.

Figura 24

Recuperar contraseña código



RECUPERAR CONTRASEÑA

Ingrese el código que aparece en su correo.

Four empty input boxes for a 4-digit code.

VERIFICAR CÓDIGO

Nota. La figura muestra el diseño para ingresar el código que se envió al correo (2024),
Elaboración propia

La **Figura 25** muestra la pestaña final para cambiar la contraseña, la cual pide la nueva contraseña que es la misma que sale cuando quieres cambiar el usuario sin la necesidad de todo el proceso.

Figura 25

Cambiar contraseña del usuario



CAMBIAR CONTRASEÑA

Contraseña

Confirmar contraseña

CAMBIAR CONTRASEÑA

Nota. La figura muestra el diseño para ingresar la nueva contraseña del usuario (2024),
Elaboración propia

4.3.2. Codificación

Funcionalidad Backend

En la codificación del segundo avance del prototipo para la venta de productos artesanales, se incluyó una nueva funcionalidad en el backend. Esta funcionalidad nos permite enviar correos electrónicos. Para lograr esto, se implementó la API de SendGrid, una plataforma que facilita el envío de correos. El código al respecto, de qué manera se utiliza, se puede consultar en el Anexo J.

Principales Funcionalidades

Se procede a indicar las principales funcionalidades que tiene el apartado de cambiar contraseña y perfil usuario.

La primera funcionalidad implementada consiste en visualizar tu perfil, donde se incluye un modal que te permite realizar ediciones. Además, ofrece opciones para desactivar la cuenta o acceder a la página de cambio de contraseña. Para obtener más detalles, consulta el Anexo K.

La siguiente funcionalidad implementada es la pantalla de cambio de contraseña. Consulte el Anexo L para obtener más información sobre cómo se realiza esta funcionalidad.

4.4.3. Pruebas del sistema

Plan de Pruebas del Sistema

El presente Plan de Pruebas presenta como objetivo garantizar la calidad y la estabilidad del software a través de un enfoque estructurado y sistemático. Para lograr este objetivo, hemos seleccionado tres tipos de pruebas fundamentales. Aunque inicialmente se incluyeron las pruebas de humo y las pruebas unitarias, se observó que ambas persiguen la misma finalidad. Por ello, se decidió priorizar las pruebas de funcionalidad, ya que permiten verificar que todas las funcionalidades del sistema estén en correcto funcionamiento. Por ende, las pruebas seleccionadas son: pruebas de humo, pruebas de funcionalidad y pruebas de usabilidad.

Pruebas de Humo

Realizar una prueba de humo en el sistema web ayudará en la verificación rápida del funcionamiento básico, detección temprana de errores críticos y la calidad del sistema.

Objetivo de la prueba:

Verificar la viabilidad y el correcto funcionamiento de las principales funcionalidades del sistema de ventas web mediante un listado de casos de prueba, para asegurar que no existan errores graves que impidan su operatividad básica.

Criterios de Éxito:

Las pruebas de humo se considerarán exitosas si:

Todos los casos de prueba son probados correctamente.

La página web es capaz de iniciarse correctamente sin errores críticos que impidan su funcionamiento básico.

Las funciones principales del sistema operan correctamente.

La revisión de registros y mensajes de error garantiza que no haya problemas que no se estén mostrando directamente a los usuarios.

Resultados de las pruebas:

Tabla 2

Pruebas de humo

Áreas de prueba	Caso Prueba	Funcionamiento		Observaciones
Registro de Usuarios	Caso de prueba 1: Verificar que un nuevo usuario pueda registrarse correctamente ingresando todos los campos requeridos.			Fue exitoso, ya que cumplió con las necesidades de la prueba y se pudo registrar un usuario correctamente.
	Caso de prueba 2: Validar que no se pueda registrar un usuario con información inválida o faltante.			Obtuvo problemas, puesto que la contraseña no es segura, lo cual podría permitir ingresar cualquier tipo de clave a los usuarios.
Proceso de Venta	Caso de prueba 3: Confirmar que un artículo se pueda ingresar correctamente al carrito de compras.			Presentó errores, porque no permitía al usuario editar la cantidad de producto que quisiera añadir al carrito de compra de primera instancia, y la edición de la

				cantidad era confusa.
	Caso de prueba 4: Asegurarse de que el cálculo del total de la compra sea exacto al agregar varios repuestos al carrito.			Cumplió con los requisitos, donde se pudo observar un buen cálculo en el cambio de las cantidades.
	Caso de prueba 5: Verificar que se genere una confirmación de compra después de realizar el pago.			Presentó errores, por el hecho de que la confirmación solo se realizaba mediante una notificación, nunca se sabía si realmente se había realizado la compra correctamente.
Iniciar sesión	Caso de prueba 6: Asegurarse de que los usuarios registrados puedan acceder al sistema utilizando sus credenciales correctas.			Se ejecutaron correctamente, ya que se arrojan correctamente los mensajes de errores cuando no se ingresa
	Caso de prueba 7: Comprobar que los usuarios no puedan acceder al sistema utilizando credenciales erróneas.			correctamente la información para el ingreso de sesión.
Registro y edición de productos	Caso de prueba 8: Verificar que un artesano pueda ingresar nuevos productos.			Fue exitoso, donde se pudo registrar un producto nuevo de los artesanos y se indicó este mismo en sus productos.
	Caso de prueba 9: Verificar que un artesano pueda editar sus productos.			Tuvo problemas, ya que la eliminación de las fotos no se realizaba correctamente, y estas se eliminaban de la base de datos antes de su actualización.
	Caso de prueba 10: Verificar que los artesanos			Presentó errores, ya que al pulsar uno de

Ventas e inventario de los Artesanos	puedan ver sus ventas en un período de tiempo.			los botones laterales al intentar colocar otra fecha, se producían errores.
	Caso de prueba 11: Verificar que el cálculo de la venta de artesanos esté correcto.			Fue correcta, ya que la suma de las ventas se realiza de manera correcta sin presentar fallas.
	Caso de prueba 12: Verificar que se pueda actualizar el inventario de los productos artesanales de forma correcta, actualizando su stock.			No presentó errores, ya que se puede observar que no permite eliminar más cantidad de productos de la disponible, y la suma al añadir se realiza de manera correcta.
	Caso de prueba 13: Verificar que alerte a los artesanos de un stock bajo en los productos.			Indicó correctamente si un producto no tenía la cantidad necesaria.
Comentarios	Caso de prueba 14: Verificar que solo los usuarios que compraron el producto puedan añadir un comentario en este.			Indicó correctamente que solo los usuarios que compraron el producto pueden acceder a la sección para añadir su comentario.
	Caso de prueba 15: Verificar que el cálculo de las calificaciones de los comentarios se realice correctamente.			Se realizó satisfactoriamente, ya que la cantidad de comentarios refleja adecuadamente la calificación que deben tener.
Edición y Desactivación de Cuenta	Caso de prueba 16: Verificar que un usuario pueda editar su información personal correctamente y que los cambios se reflejen en su cuenta.			Presentó fallos, ya que, al editar el usuario, no permitía visualizar instantáneamente

				los cambios al cerrar el modal.
	Caso de prueba 17: Verificar que, al desactivar una cuenta, el usuario ya no pueda acceder a la plataforma.			No presentó errores, porque el usuario fue desactivado correctamente y no se le permitió ingresar.
Recuperación de Contraseña	Caso de prueba 18: Asegurar que un usuario pueda restablecer su contraseña mediante el correo electrónico vinculado a su cuenta, garantizando al mismo tiempo la seguridad y eficacia del proceso.			No presentó errores, ya que el usuario pudo recuperar su contraseña de manera sencilla.
	Caso de prueba 19: Verificar que no se pueda ingresar cualquier otro código para ingresar a la pestaña de cambio de contraseña.			No presentó errores, ya que el código de validación siempre cambiaba y se validaba que la contraseña fuera segura al cambiarla.

Nota. La tabla muestra los distintos casos de prueba realizados en las pruebas del sistema (2024),
Elaboración propia

Pruebas de Funcionalidad

El objetivo de este plan es definir y documentar las pruebas de funcionalidad que se realizarán en el sistema de venta de productos artesanales. Estas pruebas asegurarán que todas las características y funciones del sistema funcionen conforme a las especificaciones.

Las pruebas de funcionalidad se enfocarán en los siguientes aspectos del sistema.

- Registro e inicio de sesión de usuarios.
- Gestión del perfil de usuario.
- Navegación y búsqueda de productos.
- Gestión del carrito de compras.

Administración de productos (para vendedores).

Criterios de Éxito:

Las pruebas de funcionalidad se considerarán exitosas si:

Todas las pruebas descritas en este plan se ejecutan y pasan sin errores críticos.

Las funciones principales del sistema operan conforme a las especificaciones.

No se encuentran errores que impidan la operación normal del sistema.

Resultados Pruebas de Funcionalidad

4.1 Inicio de sesión y registro de sesión de Usuarios

Tabla 3

Tabla de las pruebas de funcionalidad sobre el ingreso y registro de los usuarios a la página web

Número de Prueba	Descripción	Resultado Esperado	Resultado Obtenido	Estado
Prueba 1	Registro de Nuevo Usuario	El usuario puede iniciar sesión.	El usuario pudo registrarse e iniciar sesión correctamente.	Aprobado
Prueba 2	Registro de Usuario con Datos Inválidos	El sistema muestra mensajes de error adecuados para los campos inválidos.	El sistema mostró mensajes de error para datos inválidos.	Aprobado
Prueba 3	Inicio de Sesión con Credenciales Correctas	El usuario será dirigido automáticamente a la página inicial del sistema.	El usuario fue dirigido correctamente de forma automática a la página inicial.	Aprobado
Prueba 4	Ingreso al sistema con datos de acceso inválidos.	El ingreso al sistema indica un mensaje de error que indica que el método de entrada no es válido.	El ingreso mostró un error con su respectivo mensaje adecuado.	Aprobado
Prueba 5	Cambio de contraseña del usuario	El sistema permite al usuario ingresar con la nueva contraseña.	El sistema permitió el ingreso con la nueva contraseña.	Aprobado

Nota. La tabla muestra las pruebas de funcionalidad sobre el registro e inicio de sesión (2024),
Elaboración propia

4.2 Gestión del Perfil del Usuario

Tabla 4*Tabla de las pruebas de funcionalidad sobre el perfil del usuario*

Número de Prueba	Descripción	Resultado Esperado	Resultado Obtenido	Estado
Prueba 6	Actualización de Perfil	Los cambios se guardan correctamente y se muestran actualizados en el perfil del usuario.	Los cambios se guardaron y se mostraron correctamente.	Aprobado

Nota. La tabla muestra la prueba de la funcionalidad sobre el perfil de usuario (2024), Elaboración propia

4.3 Navegación y Búsqueda de Productos

Tabla 5*Tabla de las pruebas de usabilidad sobre la navegación de los productos*

Número de Prueba	Descripción	Resultado Esperado	Resultado Obtenido	Estado
Prueba 7	Navegación por Categorías de Productos	Se muestran los productos correspondientes a la categoría seleccionada.	Los productos se mostraron correctamente por categoría.	Aprobado
Prueba 8	Búsqueda de Productos	Se presentan los resultados de las búsquedas pertinentes realizadas a los productos.	Los resultados de búsqueda fueron relevantes y correctos.	Aprobado

Nota. La tabla muestra las pruebas de funcionalidad sobre la navegación (2024), Elaboración propia

4.4 Gestión del Carrito de Compras

Tabla 6*Tabla de las pruebas de funcionalidad sobre el carrito de compras*

Número de Prueba	Descripción	Resultado Esperado	Resultado Obtenido	Estado
-------------------------	--------------------	---------------------------	---------------------------	---------------

Prueba 9	Añadir Producto al Carrito	El producto se añade al carrito y el contador de artículos se actualiza.	El producto se añadió al carrito y el contador se actualizó.	Aprobado
Prueba 10	Ver y Editar Carrito	El carrito muestra los cambios realizados correctamente.	Los cambios se reflejaron correctamente en el carrito.	Aprobado

Nota. La tabla muestra las pruebas de funcionalidad sobre el carrito de compras (2024), Elaboración propia

4.5 Administración de los Productos (para artesanos)

Tabla 7

Tabla de las pruebas de funcionalidad sobre la administración de los productos

Número de Prueba	Descripción	Resultado Esperado	Resultado Obtenido	Estado
Prueba 11	Añadir Nuevo Producto	El producto se añade correctamente y aparece en la lista de productos disponibles.	El artículo que se añadió se indica sin problemas en los productos disponibles.	Aprobado
Prueba 12	Actualizar Producto Existente	Los cambios se guardan correctamente y se reflejan en el producto.	Los cambios se guardaron y reflejaron correctamente en el producto.	Aprobado
Prueba 13	Manejar inventario del Producto Existente	Los cambios en las cantidades de los productos se reflejan correctamente en el producto.	Los cambios se guardaron y reflejaron correctamente en el producto.	Aprobado

Nota. La tabla muestra las pruebas de funcionalidad sobre los productos de los artesanos (2024), Elaboración propia

Pruebas de Usabilidad

El objetivo de este plan es definir y documentar las pruebas de usabilidad que se llevaron en el sistema de venta de productos artesanales. Las evaluaciones de usabilidad se enfocarán en

analizar la facilidad de uso, la eficiencia y la satisfacción de los usuarios, abarcando tanto a los compradores como a los artesanos.

Objetivos

- Evaluar la facilidad de uso del sistema.
- Detectar inconvenientes relacionados con la usabilidad.
- Obtener retroalimentación de los usuarios sobre su experiencia.
- Ofrecer sugerencias para optimizar la usabilidad del sistema.

Metodología

Las pruebas de usabilidad se llevarán a cabo mediante observación directa y recopilación de retroalimentación de los usuarios mientras interactúan con el sistema. Los participantes realizarán tareas específicas y se registrarán sus acciones, comentarios y problemas encontrados.

Participantes

Para la realización de esta prueba serán 6 usuarios artesanos y 6 usuarios compradores para poder tener un análisis de ambas partes.

Tareas de Prueba

Situaciones usuarios compradores:

1. Deseas comprar un regalo para tu mejor amigo por su cumpleaños.
2. Cambiaste de provincia y quieres actualizar tu usuario.
3. Te olvidaste tu contraseña.

Situaciones usuarios artesanos:

1. Realizaste un nuevo producto y quieres que las personas puedan comprarlo.
2. Te compraron un producto desde tu tienda física y deseas cambiar el stock del producto.
3. Quieres saber si has tenido ventas recientemente y cuál fue el producto vendido.

Criterios de Éxito

1. Los usuarios deben poder completar cada tarea sin asistencia.
2. Los usuarios deben expresar una satisfacción general positiva.
3. Se deben identificar y documentar los problemas de usabilidad encontrados.

Resultados Pruebas de Usabilidad

Tabla 8*Pruebas de usabilidad para usuarios compradores*

Situación	Éxito	Fallo	Problema Encontrado	Gravedad
Comprar un regalo para tu mejor amigo por su cumpleaños.	4	2	El ingreso de la cantidad al comprar al usuario en el detalle del producto confundía al usuario.	Media
Cambiaste de provincia y quieres actualizar tu usuario	4	2	La pestaña de ver tu perfil no indica que se puede editar.	Media
Te olvidaste tu contraseña.	3	3	La pestaña "Ingrese su código" debería indicar que el código fue enviado por correo electrónico y que, si no lo encuentran, revisen la carpeta de spam.	Media

Nota. La tabla indica los resultados de las pruebas de usabilidad sobre usuarios compradores (2024), Elaboración propia

Correcciones realizadas:

- Se realizaron cambios en la página de detalle del producto para que sea más evidente qué productos ya están agregados al carrito de compra. Además, se incluyeron mensajes de confirmación para cualquier modificación en el carrito de compras, garantizando así una experiencia de compra más clara y segura. También se verificaron todos los botones dentro de la página web para asegurar su funcionamiento adecuado.
- Se añadió un botón claramente visible en la página "Ver tu perfil" con la etiqueta "Editar información", lo cual simplifica la tarea de actualización del perfil para el usuario. Además, se agregaron notificaciones que confirman de manera explícita cuando se realiza con éxito una edición en la información del usuario, brindando una retroalimentación inmediata y útil.
- Se añadió un mensaje claro en la pestaña "Ingrese su código" para informar al usuario que el código de verificación ha sido enviado por correo electrónico. Además, se indica explícitamente que, en caso de no encontrar el correo electrónico, se debe revisar la carpeta de spam para asegurar la recepción del código. Esta mejora garantiza que los usuarios estén al tanto del proceso de verificación y puedan completarlo sin contratiempos.

Tabla 9*Prueba de usabilidad para usuarios artesanos*

Situación	Éxito	Fallo	Problema Encontrado	Gravedad
Realizaste un nuevo producto y quieres que las personas puedan comprarlo.	2	4	El formulario de publicación de productos es confuso y la navegación hacia la pestaña de artesanos no es intuitiva ni entendible.	Alta
Te compraron un producto desde tu tienda física y deseas cambiar la cantidad del producto	4	2	El botón de inventario no indica claramente que sirve para añadir o disminuir la cantidad del producto.	Media
Quieres saber si has tenido ventas recientemente y cuál fue el producto vendido.	3	3	La interfaz para revisar las ventas es confusa y se pierde la información	Media

Nota. La tabla presenta los resultados de las pruebas de usabilidad realizadas con usuarios artesanos (2024), Elaboración propia.

Correcciones realizadas:

- Se colocó el nombre "Panel de Artesano" junto a un ícono distintivo para indicar claramente que se trata de la página destinada al artesano. Esto mejora la identificación visual y facilita la navegación del usuario.
- Se añadieron ejemplos y una explicación más detallada sobre la información que se debe ingresar en el formulario de productos. Esto incluye instrucciones específicas para cada campo, con el fin de guiar al usuario en el proceso de llenado y asegurar que todos los datos ingresados sean correctos y completos.
- En el botón de inventario, se añadió la palabra "Inventario" y se mejoró la descripción de su funcionalidad. Ahora, al hacer clic en este botón, el usuario puede ver claramente que se trata de la sección donde se gestiona el inventario, incluyendo la posibilidad de agregar, eliminar productos.
- Se separaron las notificaciones de las ventas para proporcionar una presentación más detallada y organizada de las últimas ventas realizadas. Esta mejora permite al artesano revisar sus ventas recientes de manera más clara y estructurada, facilitando la gestión de su negocio.

Encuesta de satisfacción de las pruebas realizadas sobre la usabilidad del sistema.

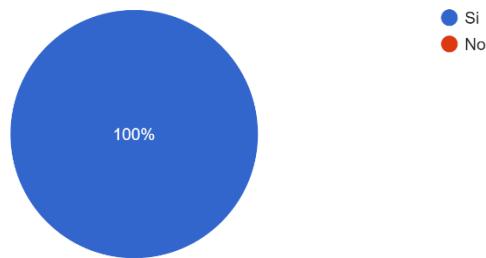
Para llevar a cabo las pruebas de usabilidad en nuestra página web, realizamos una encuesta de satisfacción para obtener una comprensión más detallada de la experiencia de los usuarios. La encuesta fue diseñada para recoger las impresiones y sugerencias de los usuarios, con el propósito de identificar áreas de mejora y fortalecer los aspectos positivos.

Los resultados de las preguntas son los siguientes:

Figura 26

Pregunta 1: ¿Cree que la página web de venta de productos artesanales le puede ser de utilidad?

¿Cree que la pagina web de venta de productos artesanales le puede ser de utilidad?
12 respuestas

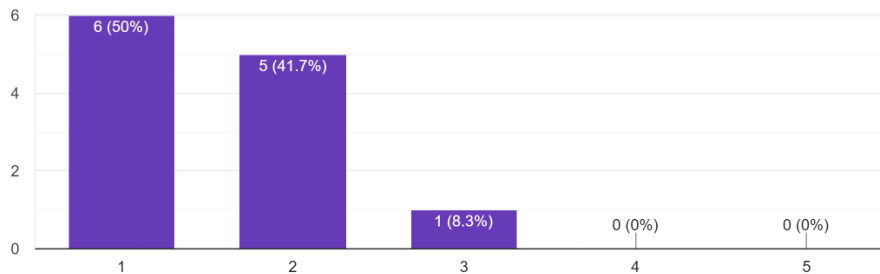


Nota. La figura muestra la tabulación de la pregunta uno en un gráfico de pastel (2024),
Elaboración propia

Figura 27

Pregunta 2: ¿Qué tan intuitiva le pareció la interfaz?

¿Qué tan intuitiva le pareció la interfaz?
12 respuestas

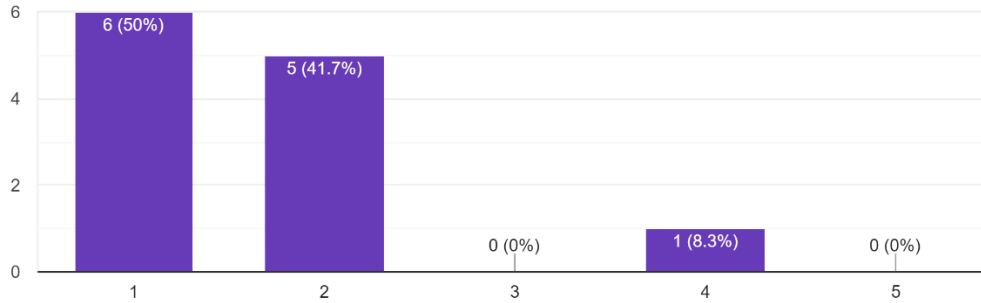


Nota. La figura muestra la tabulación de la pregunta dos en un diagrama de barras donde 1 significa Excelente 5 deficiente (2024), Elaboración propia

Figura 28

Pregunta 3: *¿Qué tan adecuados le parecieron los colores escogidos para la aplicación?*

¿Qué tan adecuados le parecieron los colores escogidos para la aplicación?
12 respuestas

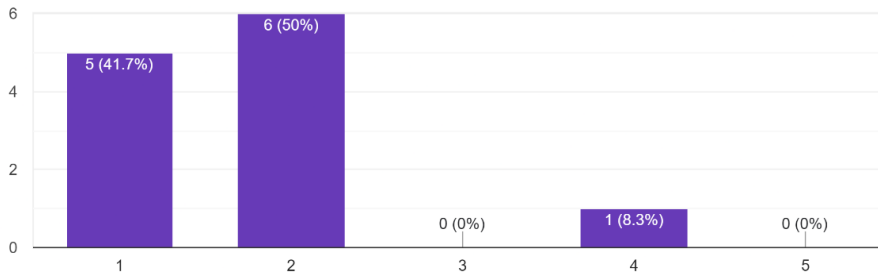


Nota. La figura muestra la tabulación de la pregunta tres en un diagrama de barras donde 1 significa Excelente 5 deficiente (2024), Elaboración propia

Figura 29

Pregunta 4: *¿Qué tan difícil le resultó aprender el funcionamiento de la aplicación?*

¿Qué tan difícil le resultó aprender el funcionamiento de la aplicación?
12 respuestas



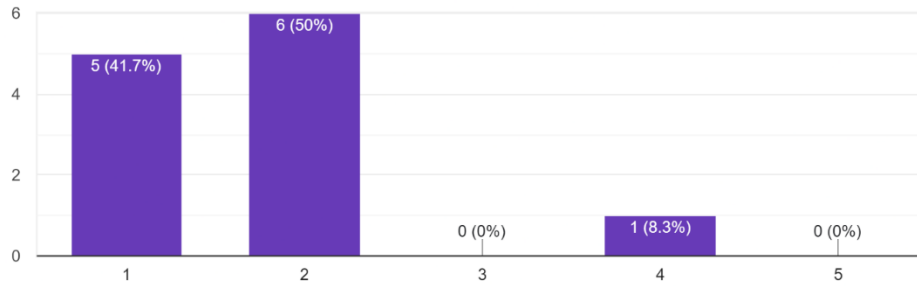
Nota. La figura muestra la tabulación de la pregunta cuatro en un diagrama de barras donde 1 significa muy fácil 5 prácticamente imposible (2024), Elaboración propia

Figura 30

Pregunta 5: ¿Qué tan difícil le resultó completar las tareas?

¿Qué tan difícil le resultó completar las tareas?

12 respuestas



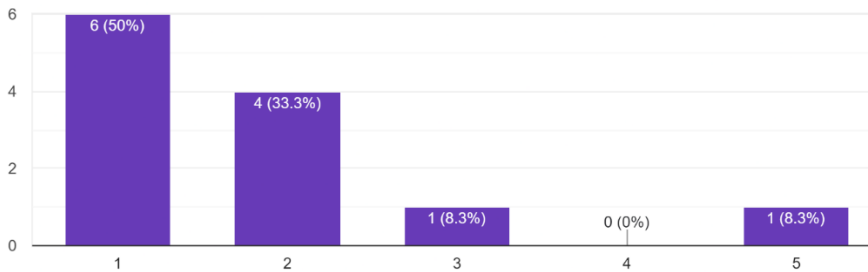
Nota. La figura muestra la tabulación de la pregunta cinco en un diagrama de barras donde 1 significa muy fácil 5 prácticamente imposible (2024), Elaboración propia

Figura 31

Pregunta 6: ¿El sitio web ofreció información clara y útil?

¿El sitio web ofreció información clara y útil?

12 respuestas



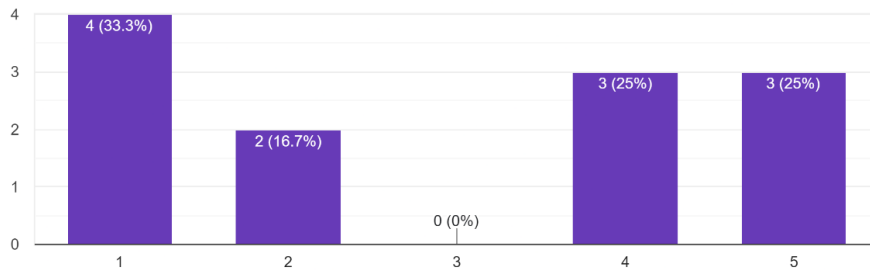
Nota. La figura muestra la tabulación de la pregunta seis en un diagrama de barras donde 1 significa muy clara 5 muy poco clara (2024), Elaboración propia

Figura 32

Pregunta 7: ¿Qué tan satisfecho está con el contenido del sitio web? (texto, imágenes, videos, etc.)

¿Qué tan satisfecho está con el contenido del sitio web? (texto, imágenes, videos, etc.)

12 respuestas



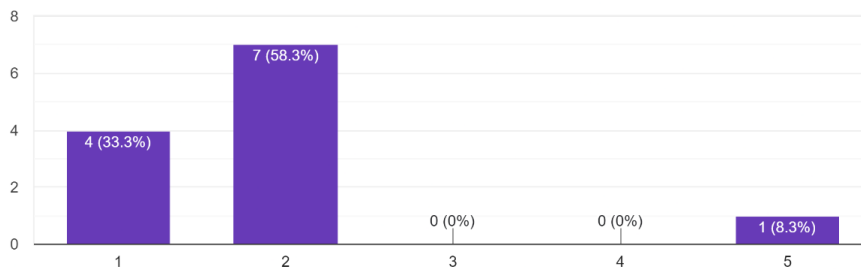
Nota. La figura muestra la tabulación de la pregunta siete en un diagrama de barras donde 1 significa muy satisfecho 5 muy insatisfecho (2024), Elaboración propia

Figura 33

Pregunta 8: ¿Fue fácil encontrar la información que buscaba?

¿Fue fácil encontrar la información que buscaba?

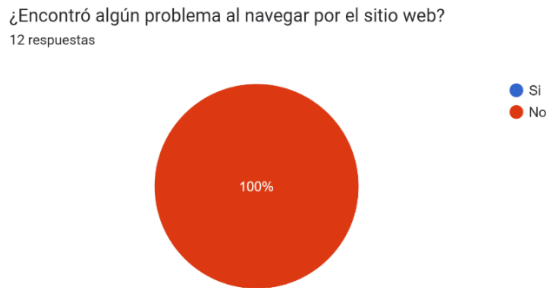
12 respuestas



Nota. La figura muestra la tabulación de la pregunta ocho en un diagrama de barras donde 1 significa muy fácil 5 muy difícil (2024), Elaboración propia

Figura 34

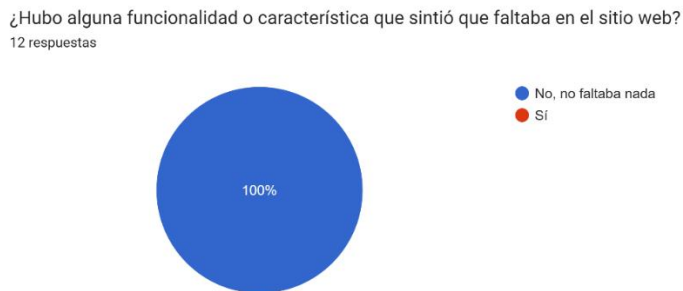
Pregunta 9: ¿Encontró algún problema al navegar por el sitio web?



Nota. La figura muestra la tabulación de la pregunta nueve en un gráfico de pastel (2024),
Elaboración propia

Figura 35

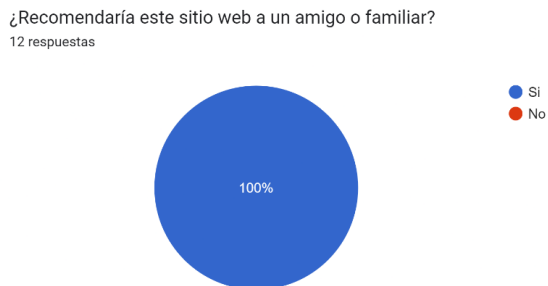
Pregunta 10: ¿Hubo alguna funcionalidad o característica que sintió que faltaba en el sitio web?



Nota. La figura muestra la tabulación de la pregunta diez en un gráfico de pastel (2024),
Elaboración propia

Figura 36

Pregunta 11: ¿Recomendaría este sitio web a un amigo o familiar?



Nota. La figura muestra la tabulación de la pregunta once en un gráfico de pastel (2024),
Elaboración propia

Las demás preguntas de la encuesta eran de respuesta abierta. Para conocer las respuestas detalladas, consulte el Anexo M.

A partir de estas respuestas se llegó a lo siguiente:

- Los usuarios consideraron que la página web cumplía con sus expectativas, lo que los llevaría a recomendarla a otras personas. Además, aunque algunos de ellos tuvieron problemas al completar las tareas, eso no fue motivo para que la aplicación no fuera de su agrado.
- Algunos usuarios señalaron la falta de imágenes y textos explicativos, sugiriendo que sería beneficioso mejorar en este aspecto. Implementar nuevos botones, iconos y páginas de explicación podría aumentar la claridad del sitio.
- Los usuarios de edad avanzada y con conocimientos limitados de tecnología prefieren secciones escritas claras y precisas, acompañadas de íconos. Esto les ayuda a entender mejor la información que, aunque pueda ser clara para algunos, no lo es tanto para ellos.

5. CAPÍTULO V - CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Al culminar el trabajo de titulación titulado “Desarrollo de un prototipo de página web de comercio electrónico para la venta e inventario de productos artesanales”, se puede concluir que se cumplido con los objetivos planteados. Por lo tanto, se pueden introducir las siguientes conclusiones:

- Se desarrolló una página web cuyo objetivo principal es permitir a los artesanos vender sus productos, incluyendo tanto la gestión de ventas como el control del inventario de cada uno de sus productos. Esto abarca tanto las ventas internas como las externas a la página web, facilitando así su utilización para estos propósitos.

- Una vez finalizado el desarrollo de la página web utilizando frameworks de Node.js como lo son React y Express, se pudo comprobar que estos frameworks, junto con sus bibliotecas especializadas, permiten el desarrollo de una aplicación que cumple con todas las funcionalidades de manera rápida y efectiva. Entre estas funcionalidades se incluyen el ingreso al perfil, la encriptación de contraseñas, el carrito de compra y la gestión del inventario de productos, lo que permite una página web más limpia.
- Aunque se siguió un proceso de desarrollo para crear una página web fácil de entender e intuitiva, las pruebas de usabilidad realizadas con usuarios finales revelaron oportunidades adicionales de mejora. Los resultados indicaron que se pueden mejorar los procesos y la recepción de los usuarios con pequeños cambios en el diseño, como lo pueden ser cambios de palabras, colores; estos cambios pueden generar una mejor experiencia de usuario.
- A partir de las pruebas de usabilidad del sistema, se llegó a la conclusión de que los usuarios prefieren una aplicación más visual, con una mayor cantidad de imágenes. Esto permitió desarrollar un diseño intuitivo que está más centrado en el usuario y mejorar su experiencia.
- Tras someter la página web a pruebas que permiten verificar todas sus funcionalidades, se llegó a la conclusión de que estas permitieron revelar errores críticos que afectan la funcionalidad básica del sistema, como la falta de validación de contraseñas seguras, problemas en los procesos de edición, gestión de cantidades y fallos en la confirmación de compra. Es crucial priorizar la corrección de estos errores para garantizar una experiencia de usuario fluida y evitar posibles repercusiones en la página web.

5.2. Recomendaciones

- Se recomienda que el desarrollador adquiriera un mayor conocimiento sobre el giro de negocio. Comprender los desafíos, necesidades y procesos es lo que puede ayudar a diseñar y adaptar la aplicación de manera más efectiva para satisfacer las demandas específicas de este sector.

- Se sugiere que se les proporcione a los usuarios de las páginas web una comprensión clara y completa sobre los beneficios tangibles que ofrece el mismo para sus negocios. Al educarlos sobre las ventajas de esta modalidad. Además, es esencial abordar directamente el sesgo arraigado en torno a los medios digitales, brindando información precisa y herramientas prácticas para identificar y evitar posibles riesgos.
- Es crucial recordar que cualquier sitio web debe someterse a un proceso de mejora continua debido a la tecnología emergente. Esta práctica garantiza que el sitio responda de manera efectiva a las necesidades de los usuarios. Por lo tanto, es crucial llevar a cabo un seguimiento periódico de la funcionalidad del sitio web para garantizar que se realicen las mejoras necesarias en futuras actualizaciones y así asegurar su correcto funcionamiento.

BIBLIOGRAFÍA

- Amazon. (n.d.). Recuperado el 5 de mayo de 2024, de <https://www.amazon.com/>.
- Bakken, D. (2001). Middleware. Encyclopedia of Distributed Computing, 11.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for Agile Software Development. Agile Alliance.
- Cabello, M. V. N. (2010). Introducción a las bases de datos relacionales. Vision Libros.
- Canós, J. H., Letelier, P., & Penadés, M. C. (2003). Metodologías ágiles en el desarrollo de software. Universidad Politécnica de Valencia, Valencia, 1-8.
- Castañeda, D. H. T., & Zavala, J. G. (2012). Comercio electrónico. Contribuciones a la Economía, 7, 1-12.

- Chen, Y. (2021). Digital marketing strategies for handcrafted products: A case study of Etsy. *Journal of Marketing Management*, 37(11-12), 1531-1554.
- Duckett, J. (2020). *JavaScript & jQuery: Interactive Front-End Web Development*. 5th Edition. Pearson Education.
- Express - Node.js web application framework. (s/f). Expressjs.com. Recuperado el 28 de mayo de 2024, de <https://expressjs.com/>
- Flanagan, D., & Snook, J. (2023). *JavaScript: The Definitive Guide* (7th ed.). O'Reilly Media.
- García, J. (2020). Inventory management strategies for small businesses selling handcrafted products online. *Journal of Small Business Management*, 58(2), 403-422.
- Highsmith, J. (2009). *Agile Project Management: Creating Innovative Products*. Addison-Wesley Professional.
- Hassenzahl, M. (2003). The thing and I: Understanding the relationship between user and product. In *Funology: From Usability to Enjoyment* (pp. 31-42). Springer.
- International Council of Societies of Industrial Design. (n.d.).
- Mercado libre. (n.d.). Recuperado el 5 de marzo de 2024, de <https://www.mercadolibre.com.ec/ayuda>
- MySQL (2023) *MySQL 8.0 Reference Manual*. Oracle Corporation.
- MySQL, A. B. (2001). *MySQL*.
- Norman, D. A. (2013). *The design of everyday things: Revised and expanded edition*. Basic Books.
- Npm: Bcrypt. (s/f). Npm. Recuperado el 28 de marzo de 2024, de <https://www.npmjs.com/package/bcrypt>
- Organisation for Economic Co-operation and Development (OECD). (2019). *Cross-border E-commerce - Issues and Trends*.
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach*. McGraw-Hill.

- Rawat, B., & Purnama, S. (2021). MySQL Database Management System (DBMS) On FTP Site LAPAN Bandung. *International Journal of Cyber and IT Service Management*, 1(2), 173-179.
- React. (s/f). React.dev. Recuperado el 25 de marzo de 2024, de <https://es.react.dev/>
- Run JavaScript everywhere. (s/f). Nodejs.org. Recuperado el 4 de junio de 2024, de <https://nodejs.org/en>
- Smith, A. (2019). User experience in e-commerce platforms for handcrafted products. *International Journal of Electronic Commerce*, 23(4), 547-572.
- Sommerville, I. (2016). *Software Engineering*. Addison-Wesley.
- Soni, A., & Ranga, V. (2019). API features individualizing of web services: REST and SOAP. *International Journal of Innovative Technology and Exploring Engineering*, 8(9), 664-671.
- Tokio. (2018, septiembre 5). ¿Qué es MySQL? ¿Cómo surgió? ¿Cuáles son sus características? Tokio School. <https://www.tokioschool.com/noticias/que-es-mysql/>
- United Nations Conference on Trade and Development (UNCTAD). (2018). *Report on E-commerce in 2018*.
- Van Meer, J. (2017). *Backend Development with Java: Building Scalable and Maintainable Web Applications*. Manning Publications.
- You are using an outdated browser. (s/f). ImgBB. Recuperado el 4 de junio de 2024, de <https://es.imgbb.com/>

Anexos

Anexo A

Formulario Todos los Usuario con tabulación

Formulario para diseño de venta de productos

Por favor si pueden responder este formulario para saber sus gustos acerca de una pagina web.

* Indica que la pregunta es obligatoria

1. 1.- ¿Alguna vez ha comprado productos en línea a partir de un comercio electrónico? *

Marca solo un óvalo.

SI

No

Si tu respuesta anterior fue positiva responde las siguientes preguntas

2. ¿Qué te gusto? de comprar en linea

Selecciona todas las opciones que correspondan.

- La amplia variedad de productos disponibles.
- La facilidad de comparar precios y características de los productos.
- La facilidad de búsqueda.
- La posibilidad de leer opiniones de otros compradores antes de realizar la compra

3. ¿Qué no te gusto?

4. ¿Qué tan difícil fue encontrar el producto que buscabas?

Marca solo un óvalo.

1 2 3 4 5

Dific Muy Facil

5. ¿Selecciona todas las opciones que encuentras pertinente al encontrar el producto que es lo que revisas?

Selecciona todas las opciones que correspondan.

- Precio del producto.
- Descripción detallada del producto.
- Opiniones y valoraciones de otros compradores.
- Imágenes del producto desde diferentes ángulos.
- Peso del producto.
- Dimensiones del producto.
- Material del producto.
- Marca del producto y reputación del fabricante.

6. ¿seleccione todas las opciones que le parezcan motivo para volver a comprar en una tienda en línea?

Selecciona todas las opciones que correspondan.

- Calidad de los productos ofrecidos.
- Experiencia de compra satisfactoria.
- Precios competitivos y ofertas especiales.
- búsqueda en el sitio web.
- Opiniones positivas de otros compradores.
- Innovación constante, con nuevos productos y funcionalidades

Si a la pregunta 1 respondió no responda las siguientes preguntas

7. ¿Cuál es su principal motivo para no comprar en línea?

8. 3.- ¿Prefieren revisar como es el producto antes de comprarlo?

Marca solo un óvalo.

Sí

No

9. ¿Les resulta complicado navegar por las plataformas de compras en línea? Explique el porque

Marca solo un óvalo.

Sí

No

Otros: _____

Preguntas para responder todos

Preguntas

10. ¿Cuáles de estos sitios web de comercio electrónico conoces?

Selecciona todas las opciones que correspondan.

- Amazon
- Mercado Libre
- Shein
- Frecuento
- Tipti

Preguntas sobre la pagina web

11. ¿Qué paleta de colores crees que se asocia más a una página de venta de productos artesanales?

Marca solo un óvalo.



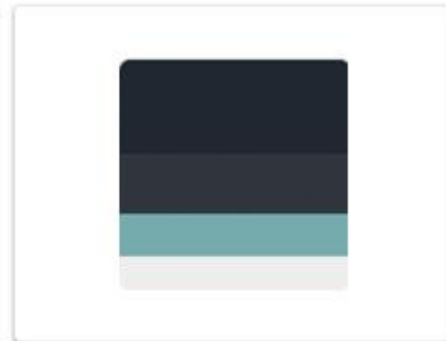
Opción 1



Opción 2



Opción 3



Opción 4

12. ¿Si tuvieras que escoger una página de comercio electrónico que el diseño mas te guste cual sería?

Marca solo un óvalo.



Opción 1



Opción 2



Opción 3



Opción 4

13. ¿Qué tan probable es que dejes tu opinión sobre un producto?

Marca solo un óvalo.

1 2 3 4 5

Muy Nada Probable

14. ¿Te gusta comunicarte con los vendedores de los productos?

Marca solo un óvalo.

1 2 3 4 5

Bast Poco

Formulario para diseño de venta de productos

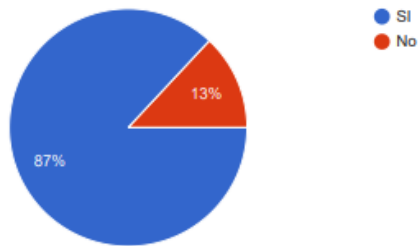
23 respuestas

[Publicar análisis](#)

1.- ¿Alguna vez ha comprado productos en línea a partir de un comercio electrónico?

[Copiar](#)

23 respuestas

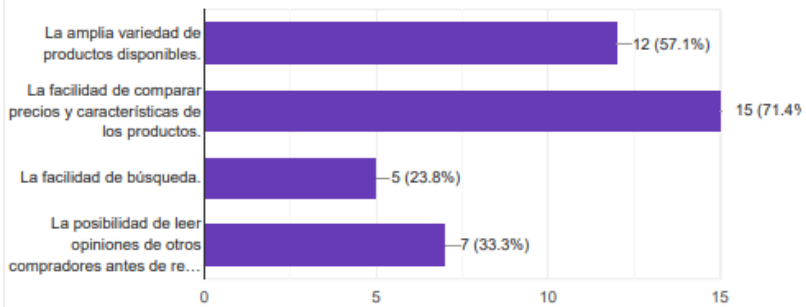


Si tu respuesta anterior fue positiva responde las siguientes preguntas

¿Qué te gusto? de comprar en linea

[Copiar](#)

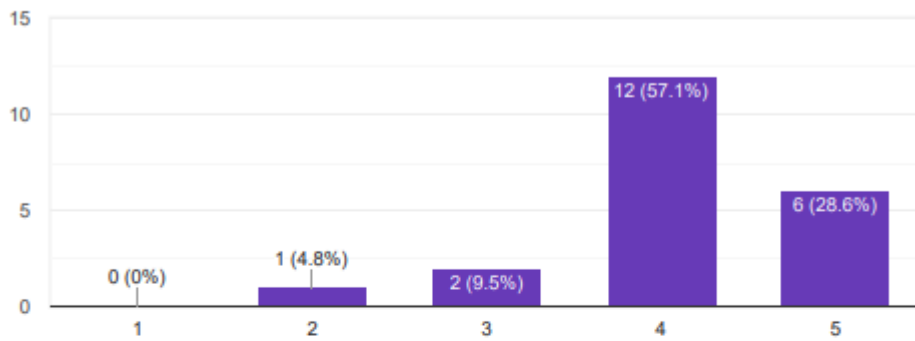
21 respuestas



¿Qué tan difícil fue encontrar el producto que buscabas?

 Copiar

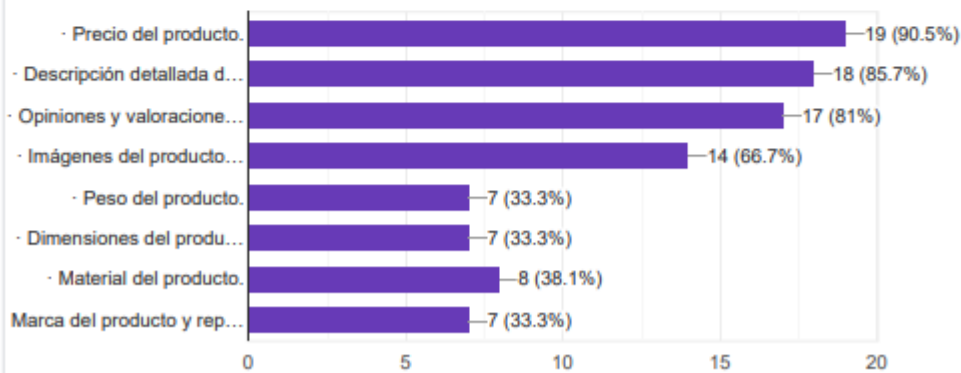
21 respuestas



¿Selecciona todas las opciones que encuentras pertinente al encontrar el producto que es lo que revisas?

 Copiar

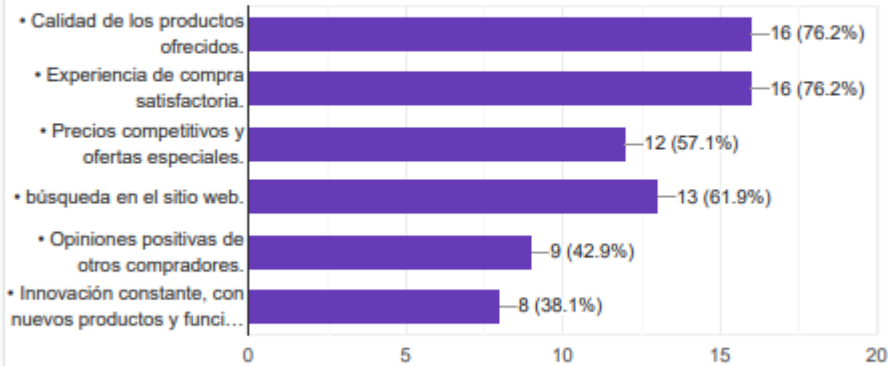
21 respuestas



¿seleccione todas las opciones que le parezcan motivo para volver a comprar en una tienda en línea?

[Copiar](#)

21 respuestas

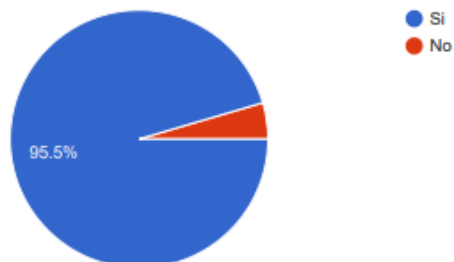


Si a la pregunta 1 respondió no responda las siguientes preguntas

3.- ¿Prefieren revisar como es el producto antes de comprarlo?

[Copiar](#)

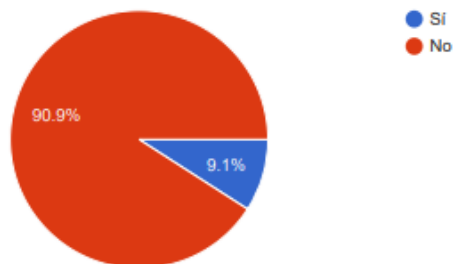
22 respuestas



¿Les resulta complicado navegar por las plataformas de compras en línea? Explique el porque

[Copiar](#)

22 respuestas

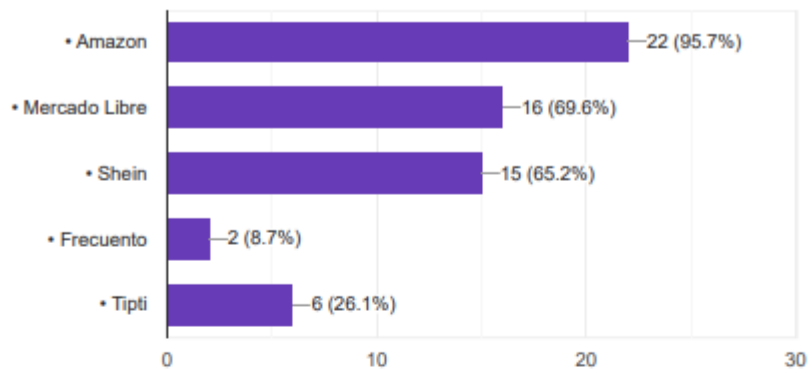


Preguntas para responder todos

¿Cuáles de estos sitios web de comercio electrónico conoces?

 Copiar

23 respuestas

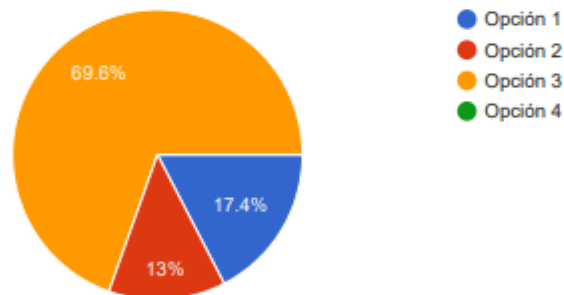


Preguntas sobre la pagina web

¿Qué paleta de colores crees que se asocia más a una página de venta de productos artesanales?

 Copiar

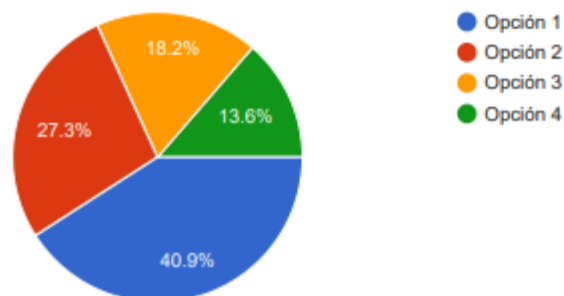
23 respuestas

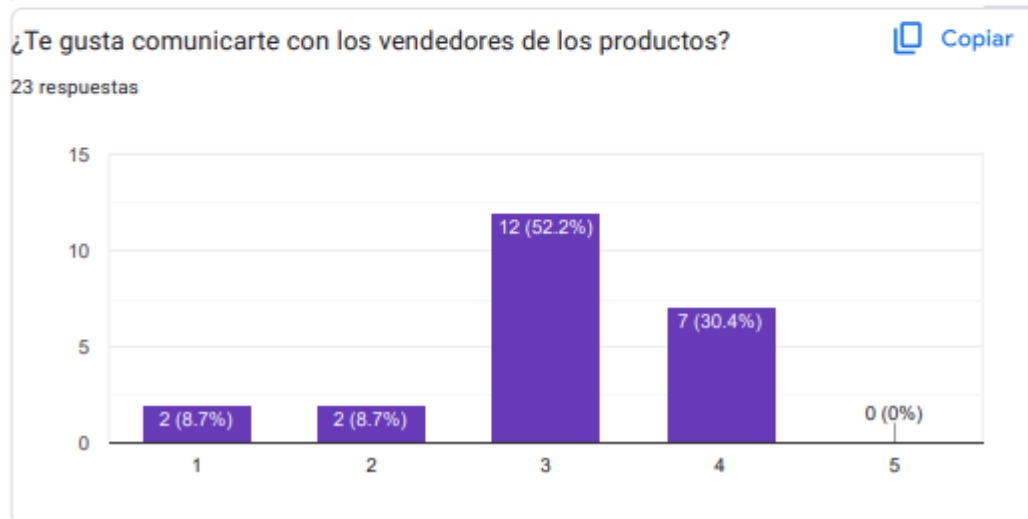
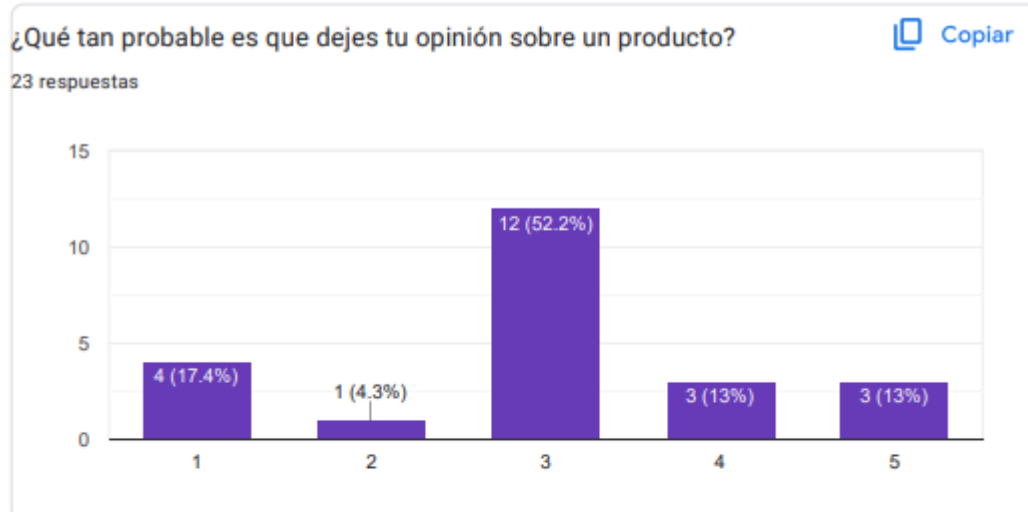


¿Si tuvieras que escoger una página de comercio electrónico que el diseño mas te guste cual sería?

 Copiar

22 respuestas





Anexo B
Encuesta usuario Artesano junto a tabulación

1. ¿Qué paleta de colores crees que se asocia más a los productos artesanales?

Marca solo un óvalo.



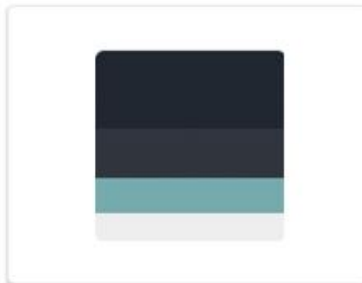
Opción 1



Opción 2



Opción 3



Opción 4

2. ¿Has utilizado un sitio de comercio electrónico para vender tus productos (No una red social)?

Marca solo un óvalo.

Sí

No

3. Si la segunda pregunta es un no salte a la siguiente pagina

3.- ¿Qué te gusto?

4. Si la segunda pregunta es un si responde las siguientes preguntas

3.- ¿Qué no te gusto?

5. ¿Por qué no has intentado vender productos por línea?

6. 4.- ¿Qué crees que te podría ayudar para poder utilizar un comercio electrónico?

Selecciona todas las opciones que correspondan.

- La sencillez de vender los productos
- El poder manejar inventario de los productos
- El saber de mejor manera como van la venta de los productos
- El conocer lo que opinan sobre tus productos

Preguntas generales para todos

7. ¿Selecciona las opciones sobre el tipo de información que le gustaría saber sobre * de sus clientes?

Selecciona todas las opciones que correspondan.

- Edad
- Sexo
- Lugar Geográfico
- Información de contacto
- dirección de Correo
- Nombre

8. 6. _ ¿En qué red social te resultaría más fácil comunicarte directamente con tus compradores?"

Marca solo un óvalo.

- Facebook
- Whatsapp
- Instagram
- Telegram

9. - ¿Seleccione las respuestas sobre que debería indicar la página web sobre tus productos?

Selecciona todas las opciones que correspondan.

- Imágen de alta calidad
- Precio
- Descripción del producto
- Medidas del Producto
- Materiales del producto
- Disponibilidad
- Información de contacto

10. ¿Cómo te gustaría que se evalúen tus productos por parte de los clientes?

Selecciona todas las opciones que correspondan.

- Estrellas (1 a 5 donde 1 es baja calificación y 5 la más alta calificación)
- Nivel de satisfacción (Bueno, Muy Bueno, Regular, Malo, muy Malo)
- Comentario
- Directamente

11. ¿Si tuvieras que escoger una página de comercio electrónico que el diseño mas te guste cual sería?

Marca solo un óvalo.



Opción 1



Opción 2



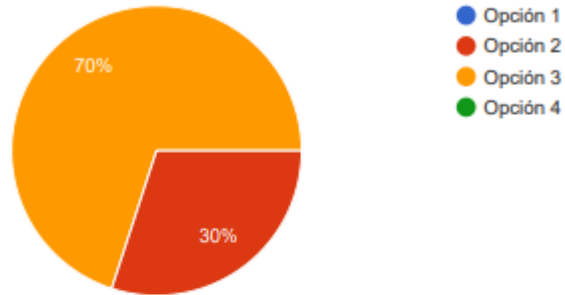
Opción 3



Opción 4

¿Qué paleta de colores crees que se asocia más a los productos artesanales?

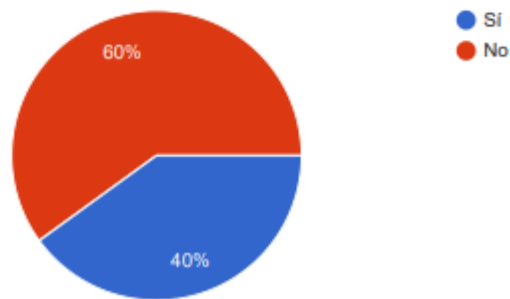
10 respuestas



Pregunta concedores

¿Has utilizado un sitio de comercio electrónico para vender tus productos (No una red social)?

10 respuestas



3.- ¿Qué te gusto?

3 respuestas

Me gustó la variedad de las cosas

Lo fácil que era vender un producto

La importancia que me tomaba la página

Si la segunda pregunta es un si responde las siguientes preguntas

3.- ¿Qué no te gusto?

5 respuestas

Opción 1

Lo difícil de crearse un usuario

Lo difícil que era vender un producto y la comunicación con el cliente

Si tu respuesta a la primera pregunta fue no responde las siguientes preguntas

¿Por qué no has intentado vender productos por línea?

5 respuestas

Se me ha complicado

Miedo

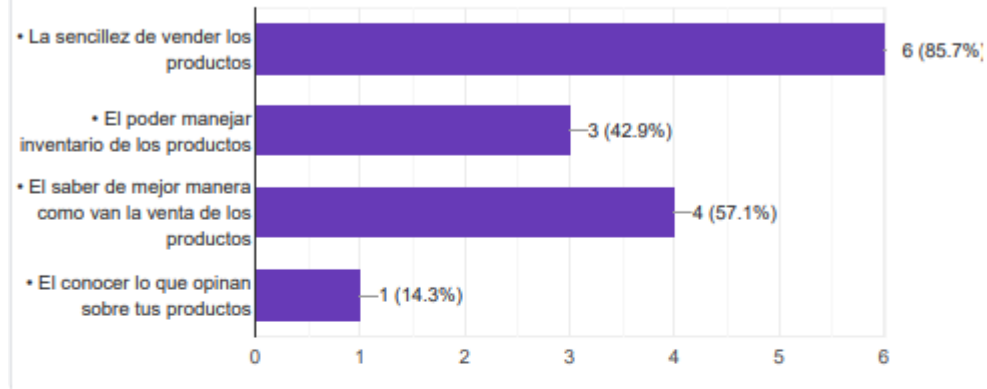
Porque no he tenido tiempo suficiente y lo he hecho de otra manera

No me he interesado por ese ámbito

Porque es muy difícil encontrar una página que sea fácil de usar

4.- ¿Qué crees que te podría ayudar para poder utilizar un comercio electrónico?

7 respuestas

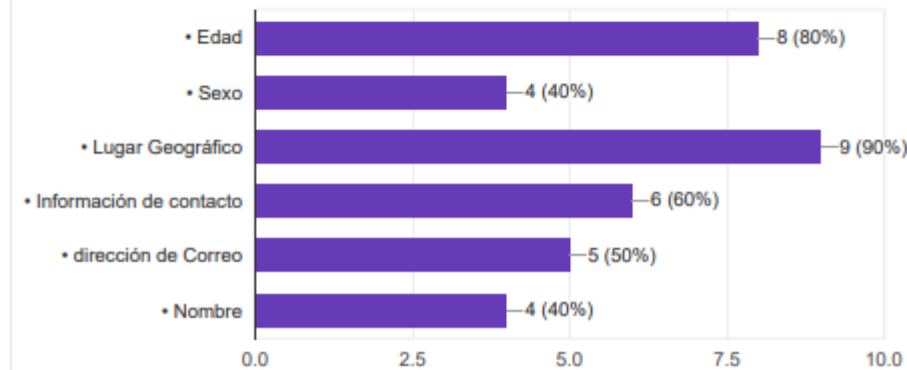


Preguntas generales para todos

¿Selecciona las opciones sobre el tipo de información que le gustaría saber sobre de sus clientes?

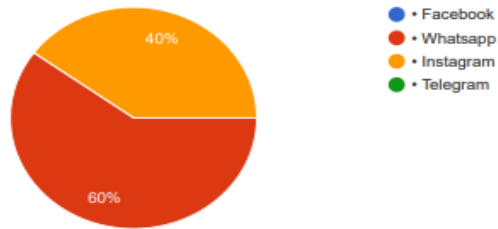
 Copi

10 respuestas



6. ¿En qué red social te resultaría más fácil comunicarte directamente con tus compradores?"

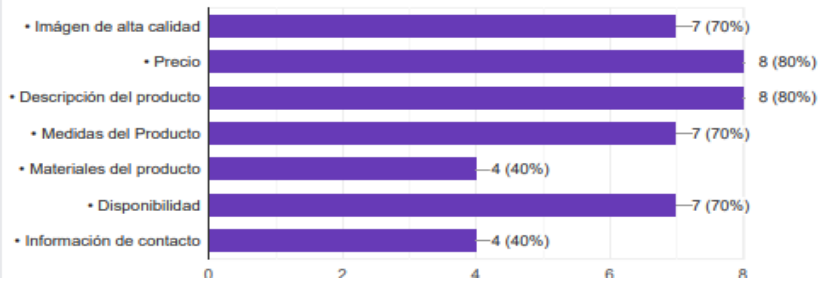
10 respuestas



- ¿Seleccione las respuestas sobre que debería indicar la página web sobre tus productos?

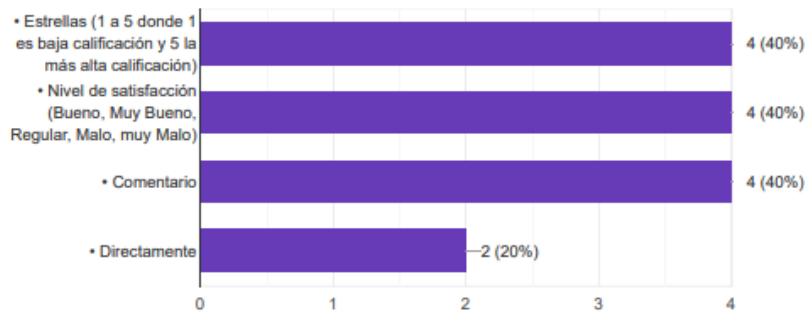
Cop

10 respuestas



¿Cómo te gustaría que se evalúen tus productos por parte de los clientes?

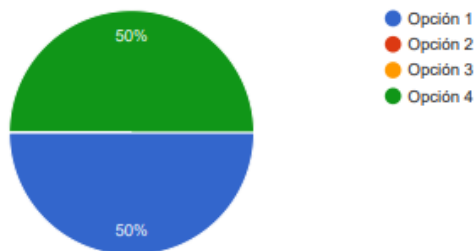
10 respuestas



Si tuvieras que escoger una página de comercio electrónico que el diseño mas te guste cual sería?

Cop

10 respuestas



Anexo C

Código BackEnd

```
const express = require('express');
const morgan = require('morgan')
const config =require('./config');
const usuarios =require('./modulos/usuarios/rutas');
const cors = require('cors');

const app = express();

app.use(morgan('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: true}));
// Configuracion
app.set('port',config.app.port);
app.use(cors());

//ruta
app.use('/api/auth', auth);
app.use('/api/provincias', provincias);
app.use(error);
module.exports = app;

const db = require('../..../DB/mysql');
const auth = require('../..../aut');
const bcrypt = require('bcrypt');
const TABLA = 'usuario'

async function login(usuario, password){
  const data = await db.query(TABLA, {Usuario: usuario});
  return bcrypt.compare(password, data.Password)
  .then(resultado =>{
    if(resultado === true){
      return data;
    }else{
      throw new Error('Información Invalida');
    }
  })
}

module.exports = {
  login,
}
```

```

async function agregar(data){
  console.log(data)
  const authData = {
    id: data.id,
  }
  if(data.Usuario){
    authData.Usuario = data.Usuario
  }

  if(data.Password){
    authData.Password = await bcrypt.hash(data.Password.toString(),5);
  }
  if(data.Numero){
    authData.Numero = data.Numero
  }
  if(data.Estado){
    authData.Estado = data.Estado
  }
  return db.agregar(TABLA, authData);
}

module.exports = {
  uno,
  agregar,
  validar,
  verificarUsuarioCorreo,
}

```

Anexo D

Insertar Usuario

```

const handleSubmit = async (e) => {
  e.preventDefault();
  const { Nombre, Apellido, Usuario, Correo, Password, ConfirmarPassword, Year, Mes, day, genero } = formData;
  if (!Nombre || !Apellido || !Usuario || !Correo || !Password || !ConfirmarPassword || !Year || !Mes || !day ) {
    setError("Por favor completa todos los campos");
    return;
  }
  if (formData.Password !== formData.ConfirmarPassword) {
    setError("Las contraseñas no coinciden");
    return;
  }
}

```

```

}
if (!isPasswordSecure(formData.Password)) {
  setError("La contraseña debe tener al menos 8 caracteres y contener al
  menos una letra minúscula, una letra mayúscula, un número y un carácter
  especial.");
  return;
}
if (!validarCorreo(formData.Correo)) {
  setError("El correo es invalido");
  return;
}
setError("");
// Hacer la solicitud a la API para validar si el usuario ya existe
try {
  const response = await
fetch(`http://localhost:4000/api/usuarios/validar/${formData.Usuario}`);
  const data1 = await response.json();
  const data=data1.body;
  if (data !== "") {
    // El usuario ya existe
    setError("El nombre de usuario ya está en uso");
    formData.Usuario = "";
    return;
  }
  // El usuario no existe, puedes registrar al usuario
  // Aquí puedes hacer la lógica para guardar los datos del usuario en tu
base de datos
  const fechaCumple = `${Year}-${Mes}-${day}`;
  const datosUsuario = {
    id: 0,
    Usuario,
    Password,
    idRol: 1,
    idProvincia: parseInt(formData.ProvinciaId),
    Nombre,
    Apellido,
    Correo,
    fechaCumple,
    Genero: genero.charAt(0).toUpperCase() + genero.slice(1), // Convertir la
primera letra del género a mayúscula
    Estado: 1, // Supongamos que el estado es fijo por ahora
  };
  try {
    // Enviar los datos al backend
    const response = await fetch("http://localhost:4000/api/usuarios", {

```

```

    method: "POST",
    headers: {
      "Content-Type": "application/json",
    },
    body: JSON.stringify(datosUsuario),
  });

  // Manejar la respuesta del backend
  if (response.ok) {
    // El usuario se registró exitosamente
    console.log("Usuario registrado correctamente");
  } else {
    // Hubo un error al registrar el usuario
    console.error("Error al registrar usuario:", response.statusText);
  }
} catch (error) {
  console.error("Error al enviar datos al backend:", error);
}
} catch (error) {
  console.error("Error al validar usuario:", error);
}
};

```

Anexo E

Manejo Carrito de Compras

```

export const cartInitialState = JSON.parse(window.localStorage.getItem('cart'))
|| []

export const CART_ACTION_TYPES = {
  SUBTRACT_FROM_CART: 'SUBTRACT_FROM_CART',
  ADD_TO_CART: 'ADD_TO_CART',
  REMOVE_FROM_CART: 'REMOVE_FROM_CART',
  CLEAR_CART: 'CLEAR_CART',
  UPDATE_QUANTITY: 'UPDATE_QUANTITY',
}

// update localStorage with state for cart
export const updateLocalStorage = state => {
  window.localStorage.setItem('cart', JSON.stringify(state))
}

const UPDATE_STATE_BY_ACTION = {
  [CART_ACTION_TYPES.SUBTRACT_FROM_CART]: (state, action) => {
    const { id } = action.payload;

```

```

const productInCartIndex = state.findIndex(item => item.id === id);

if (productInCartIndex >= 0) {
  let newState;
  if (state[productInCartIndex].cantidad > 1) {
    newState = [
      ...state.slice(0, productInCartIndex),
      { ...state[productInCartIndex], cantidad:
state[productInCartIndex].cantidad - 1 },
      ...state.slice(productInCartIndex + 1)
    ];
  } else {
    newState = [
      ...state.slice(0, productInCartIndex),
      ...state.slice(productInCartIndex + 1)
    ];
  }

  updateLocalStorage(newState);
  return newState;
}
// Si el producto no está en el carrito, no hacemos cambios
return state;
},
[CART_ACTION_TYPES.ADD_TO_CART]: (state, action) => {
  const { id, cantidad } = action.payload; // Obtener la cantidad del
payload de la acción
  const productInCartIndex = state.findIndex(item => item.id === id);

  if (productInCartIndex >= 0) {
    const newState = [
      ...state.slice(0, productInCartIndex),
      { ...state[productInCartIndex], cantidad:
state[productInCartIndex].cantidad + 1 }, // Sumar la cantidad
      ...state.slice(productInCartIndex + 1)
    ];
    updateLocalStorage(newState);
    return newState;
  }

  const newState = [
    ...state,
    {
      ...action.payload, // product
      cantidad // Establecer la cantidad inicial
    }
  ];
  updateLocalStorage(newState);
  return newState;
}
}

```

```

    }
  ];
  updateLocalStorage(newState);
  return newState;
},
[CART_ACTION_TYPES.UPDATE_QUANTITY]: (state, action) => {
  const { id, newQuantity } = action.payload;
  const productIndex = state.findIndex(item => item.id === id);

  if (productIndex >= 0) {
    // Creamos una copia del estado actual
    const newState = [...state];
    // Actualizamos la cantidad del producto especificado
    newState[productIndex].cantidad = newQuantity;

    // Actualizamos el almacenamiento local con el nuevo estado
    updateLocalStorage(newState);

    return newState;
  }
  // Si el producto no está en el carrito, no hacemos cambios
  return state;
},
[CART_ACTION_TYPES.REMOVE_FROM_CART]: (state, action) => {
  const { id } = action.payload
  const newState = state.filter(item => item.id !== id)
  updateLocalStorage(newState)
  return newState
},
[CART_ACTION_TYPES.CLEAR_CART]: () => {
  updateLocalStorage([])
  return []
}
}

export const cartReducer = (state, action) => {
  const { type: actionType } = action;
  const updateState = UPDATE_STATE_BY_ACTION[actionType];
  return updateState ? updateState(state, action) : state;
}

```

Anexo F

Producto Detallado

```

function ProductoDetalle() {
  const checkProductInCart = product => {
    return cart.some((item) => item.id === product.id);
  };

  const fetchRatings = async () => {
    try {
      const response = await
fetch(`http://localhost:4000/api/comentario/calificacion/${idProducto}`);
      if (!response.ok) {
        throw new Error('Failed to fetch ratings');
      }
      const data1 = await response.json();
      const data = data1.body;
      const ratingsArray = [];
      for (let i = 1; i <= 5; i++) {
        const calificacion = i;
        const cantidad = 0;
        ratingsArray.push({ calificacion, cantidad });
      }
      // Comparamos los datos del backend con el array existente para
actualizarlos si es necesario
      const updatedRatingsArray = ratingsArray.map(rating => {
        const existingRating = data.find(item => item.calificacion ===
rating.calificacion);
        if (existingRating) {
          return existingRating;
        } else {
          return rating;
        }
      });
      const handleQuantityChange = (productId, value) => {
        const quantity = value === '' || parseInt(value) >= 0 ? value :
quantities[productId] || '';

        setQuantities(prevQuantities => ({
          ...prevQuantities,
          [productId]: quantity
        }));
      });
    };

    const preventInvalidInput = e => {
      const charCode = e.which ? e.which : e.keyCode;
      if (charCode !== 8 && charCode !== 0 && (charCode < 48 || charCode > 57))
{

```

```

        e.preventDefault();
    }
};

const isProductInCart = checkProductInCart(producto);

return (
    <>
        <section className="text-gray-600 body-font overflow-hidden">
            <div className="container px-5 py-24 mx-auto">
                <div className="lg:w-4/5 mx-auto flex flex-wrap">
                    <div className="lg:w-1/2 w-full lg:h-auto h-auto object-cover
object-center rounded">
                        <ImageGallery items={images} />
                    </div>
                    <div className="lg:w-1/2 w-full lg:pl-10 lg:py-6 mt-6 lg:mt-
0">
                        <h2 className="text-sm title-font text-gray-500 tracking-
widest">{producto.marca}</h2>
                        <h1 className="text-gray-900 text-3xl title-font font-
medium mb-1">{producto.descripcion}</h1>
                        <div className="flex mb-4">
                            <span className="flex items-center">
                                {[...Array(5)].map((_, index) => (
                                    <svg key={index} className={`w-4 h-4 ${index <
Math.round(averageRating) ? 'text-yellow-300' : 'text-gray-300'} me-1 dark:text-
gray-500`} xmlns="http://www.w3.org/2000/svg" fill="currentColor" viewBox="0 0 22
20">
                                        )))}
                                <span className="text-gray-600 ml-
3">{totalRatings} Reseñas</span>
                            </span>
                        </div>
                        <p className="leading-relaxed">Dimensiones Largo-Ancho-
Alto : {producto.dimensiones}</p>
                        <p className="leading-relaxed">Peso: {producto.peso}
gr</p>
                        <label
htmlFor={`cantidad_${producto.id}`}>Cantidad:</label>
                        <input
                            type="text"
                            id={`cantidad_${producto.id}`}
                            value={quantities[producto.id] || ''}
                            onChange={(e) => handleQuantityChange(producto.id,
e.target.value)}
                            onKeyDown={preventInvalidInput}

```

```

                className="w-10 px-2 py-1 border border-gray-300
rounded mr-2"
            />
            <br></br>
            <br/>
            <a
href={`https://wa.me/${numeroVendedor}?text=Me%20interesa%20tu%20producto%20${pro
ducto.descripcion}`} target="_blank" rel="noopener noreferrer" className="flex
items-center ml-auto text-white bg-green-500 border-0 py-2 px-4 focus:outline-
none hover:bg-green-600 rounded">
                <FaWhatsapp className="mr-2" /> Contactate con el
vendedor
            </a>
            <div className="flex mt-6 items-center pb-5 border-b-2
border-gray-100 mb-5">
                </div>
                <div className="flex">
                    <span className="title-font font-medium text-2xl
text-gray-900"> ${producto.precioVenta || "0.00"}</span>
                    <button
onClick={() => addToCartHandler(producto,
quantities[producto.id])}
                    className={`flex ml-auto text-white border-0 py-2
px-6 focus:outline-none rounded ${isProductInCart ? 'bg-red-500 hover:bg-red-600'
: 'bg-tercero hover:bg-cuarto'}`>
                        data-product-id={idProducto}
                    >
                        {isProductInCart ? 'Eliminar del carrito de
compras' : 'Añadir al carrito de compras'}
                    </button>
                </div>
            </div>
        </div>
    </div>
    <div className="flex mt-6 items-center pb-5 border-b-2 border-gray-
100 mb-5">
        </div>

        <h4 className="text-2xl font-bold dark:text-white">Evaluaciones De
Clientes</h4>
    </section>
</>
);
}

```

```
export default ProductoDetalle;
```

Anexo G

Ingresar Comentario

```
export function Star() {
  const handleCommentChange = (event) => {
    setComment(event.target.value);
  };

  const handleCommentSubmit = async (event) => {
    event.preventDefault();

    // Validación
    if (!rating) {
      alert("Por favor selecciona una calificación.");
      return;
    }

    // Mostrar el alert para confirmar el envío
    const isConfirmed = window.confirm("¿Estás seguro de enviar este comentario?");
    if (!isConfirmed) {
      return;
    }

    // Otros datos
    const currentDate = moment().format('YYYY-MM-DD');
    const userDataString = localStorage.getItem('userData');
    let userId;
    console.log(currentDate);
    if (userDataString) {
      const userData = JSON.parse(userDataString);
      userId = userData.id;
    } else {
      alert("No se pudo obtener el ID del usuario.");
      return;
    }

    // Enviar la solicitud al backend
    try {
      const response = await fetch('http://localhost:4000/api/comentario/',
{
      method: 'POST',
```

```

        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({
            id:0,
            comentario: comment,
            calificacion: rating,
            fecha: currentDate,
            idProducto: idProducto,
            idUsuario: userId,
        })),
    });

} catch (error) {
    console.error('Error en la solicitud:', error);
}

// Reiniciar el estado del comentario
setComment("");
setRating(0);
};

return (
    <form onSubmit={handleCommentSubmit} className="flex flex-col items-center">
        <div className="flex items-center mr-auto">
            [...Array(5)].map((_, index) => {
                const currentRating = index + 1;
                return (
                    <label key={index} className="cursor-pointer">
                        <input
                            type="radio"
                            name="rating"
                            value={currentRating}
                            onClick={() => setRating(currentRating)}
                        />
                        <FaStar
                            className="star sm:w-8 sm:h-8 w-12 h-12"
                            color={currentRating <= (hover || rating) ?
"#ffc107" : "#e4e5e9"}
                            onMouseEnter={() => setHover(currentRating)}
                            onMouseLeave={() => setHover(null)}
                        />
                    </label>
                );
            });

```

```

    }}
  </div>
  <div className="w-full border border-gray-200 rounded-lg bg-gray-50
dark:bg-gray-700 dark:border-gray-600">
    <div className="px-4 py-2 bg-white rounded-t-lg dark:bg-gray-
800">
      <label htmlFor="comment" className="sr-only">Tu
Comentario</label>
      <textarea
        id="comment"
        rows="4"
        className="w-full px-0 text-sm text-gray-900 bg-white
border-0 dark:bg-gray-800 focus:ring-0 dark:text-white dark:placeholder-gray-400"
        placeholder="Escribe el comentario..."
        value={comment}
        onChange={handleCommentChange}
      ></textarea>
    </div>
    <div className="flex items-center justify-between px-3 py-2
border-t dark:border-gray-600">
      <button
        type="submit"
        className="flex mr-auto text-white bg-tercero border-0
py-2 px-6 focus:outline-none hover:bg-cuarto rounded"
      >
        Insertar Comentario
      </button>
    </div>
  </div>
</form>
)
}

export default Star;

```

Anexo H

Ingresar y editar productos en el usuario artesano.

```

const handleSubmit = async (e) => {
  const promises = [];

  for (let i = 0; i < images.length; i++) {
    const formDataImage = new FormData();
    formDataImage.append('image', images[i].file);

```

```

    const apiKey = 'tuApi'; // Reemplazar con tu propia clave de API de ImgBB
    const response = await fetch('https://api.imgbb.com/1/upload?key=' +
    apiKey, {
      method: 'POST',
      body: formDataImage
    });

    const data = await response.json();
    promises.push(data.data.display_url);
  }

  try {
    const imageUrls = await Promise.all(promises);
    console.log('Enlaces de imágenes subidas:', imageUrls);

    response = await fetch("http://localhost:4000/api/productos", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(datosProducto),
    });
    if (response.ok) {
      console.log("Producto registrado correctamente");
      const data = await response.json();
      console.log(data.body);
      // Haz aquí cualquier otra acción necesaria después de registrar el
      producto correctamente
      try {
        // Insertar los enlaces de las imágenes en la base de datos
        const promises = [];

        for (let i = 0; i < imageUrls.length; i++) {
          const imageUrl = imageUrls[i];
          const datosImagen = {
            id: 0,
            imagen: imageUrl,
            idProducto: data.body, // Suponiendo que `data.body` es el ID del
            producto al que pertenecen las imágenes
          };
          // Enviar los datos de la imagen al backend para guardar en la base
          de datos

```

```

        const responseImagen = await
fetch("http://localhost:4000/api/imagenes", {
    method: "POST",
    headers: {
        "Content-Type": "application/json",
    },
    body: JSON.stringify(datosImagen),
});

    promises.push(responseImagen);
}

// Esperar a que todas las solicitudes POST se completen
await Promise.all(promises);

    console.log("Imágenes registradas correctamente en la base de datos");
    navigate('/tusproductos');
} catch (error) {
    console.error("Error al subir imágenes o enviar datos al backend:",
error);
}
} else {
    console.error("Error al registrar producto:", response.statusText);
}
} catch (error) {
    console.error("Error al subir imágenes o enviar datos al backend:", error);
}
};

```

Anexo I

Inventario de un producto del artesano

```

const handleSubmit = async(e) => {
    e.preventDefault();
    if (inputCantidad < 0) {
        alert('La cantidad no puede ser negativa');
        return;
    }
    if(selectedOption=== "-"){
        const response = await
fetch(`http://localhost:4000/api/productos/cantidad/${selectedProductId}`);
        if (!response.ok) {

```



```

        try{
            const suma= cantidadDisponible+cantidad
            const datosProductosActu={
                id: selectedProductId,
                stock: suma,
            };
            const response = await fetch("http://localhost:4000/api/productos", {
                method: "POST",
                headers: {
                    "Content-Type": "application/json",
                },
                body: JSON.stringify(datosProductosActu),
            });
            if (response.ok) {
                console.log("Producto registrado correctamente");
            }
        }catch (error) {
            console.error("Error al registrar producto:", error);
        }
        closeModal();
    };
    Return (
    {showModal && (
        <div className="fixed inset-0 overflow-y-auto flex justify-center items-center z-50 bg-gray-500 bg-opacity-50">
            <div className="relative bg-white rounded-lg shadow-lg max-w-md w-full">
                <div className="flex justify-between items-center p-4 border-b">
                    <h3 className="text-lg font-semibold text-gray-900">
                        Manejo Inventario Producto
                    </h3>
                    <button onClick={closeModal} className="text-gray-400 hover:bg-gray-200 hover:text-gray-900 rounded-lg text-sm w-8 h-8 flex justify-center items-center">
                        <svg className="w-3 h-3" aria-hidden="true"
xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 14 14">
                            <path stroke="currentColor" strokeLinecap="round"
strokeLinejoin="round" strokeWidth="2" d="m1 1 6 6m0 0 6 6M7 7l6-6M7 7l-6 6" />
                        </svg>
                        <span className="sr-only">Cerrar modal</span>
                    </button>
                </div>
                <div className="p-4 flex flex-col items-center">
                    <label htmlFor="description" className="block mb-2 text-sm font-medium text-gray-900 dark:text-white">Descripción:</label>
                    <p>{selectedProduct && selectedProduct.descripcion}</p>
                </div>
            </div>
        </div>
    )
    )

```

```

        <img src={selectedProduct && selectedProduct.imagen}
alt={selectedProduct && selectedProduct.descripcion} className="h-40 w-40 object-
center" />
        <form onSubmit={handleSubmit} className="w-full mt-4">
          <div className="col-span-2 sm:col-span-1">
            <label htmlFor="cantidad" className="block mb-2 text-sm font-
medium text-gray-900 dark:text-white">Cantidad</label>
            <input type="number" name="cantidad" id="cantidad"
value={inputCantidad} onChange={(e) => handleQuantityChange(e.target.value)}
min="0" className="bg-gray-50 border border-gray-300 text-gray-900 text-sm
rounded-lg focus:ring-primary-600 focus:border-primary-600 block w-full p-2.5
dark:bg-gray-600 dark:border-gray-500 dark:placeholder-gray-400 dark:text-white
dark:focus:ring-primary-500 dark:focus:border-primary-500" placeholder="Cantidad"
required="" />
          </div>
          <div className="col-span-2 sm:col-span-1 mt-4">
            <label htmlFor="category" className="block mb-2 text-sm font-
medium text-gray-900 dark:text-white">Tipo</label>
            <select id="category" value={selectedOption} onChange={(e) =>
setSelectedOption(e.target.value)} className="bg-gray-50 border border-gray-300
text-gray-900 text-sm rounded-lg focus:ring-primary-500 focus:border-primary-500
block w-full p-2.5 dark:bg-gray-600 dark:border-gray-500 dark:placeholder-gray-
400 dark:text-white dark:focus:ring-primary-500 dark:focus:border-primary-500">
              <option value="+">Añadir(+)</option>
              <option value="-">Quitar(-)</option>
            </select>
          </div>
          <div className="flex justify-end mt-4">
            <button type="button" onClick={closeModal} className="bg-gray-300
text-gray-800 hover:bg-gray-400 px-4 py-2 rounded-md mr-2">Cancelar</button>
            <button type="submit" className="bg-primar hover:bg-tercero text-
white font-bold py-2 px-4 rounded">Enviar</button>
          </div>
        </form>
      </div>
    </div>
  </div>
);
export default ProductosTabla;

```

Anexo J

Envío de Correos electrónicos por el backend

```

const sgMail = require('@sendgrid/mail');

// Configura tu API Key de SendGrid
sgMail.setApiKey('tu_api');

const sendVerificationEmail = async (email, code) => {
  const msg = {
    to: email,
    from: 'ventaArtesanales2022@outlook.com', // Usa una dirección de correo
    // verificada en SendGrid
    subject: 'Código de verificación',
    text: `Tu código de verificación es: ${code}`,
  };

  try {
    await sgMail.send(msg);
    console.log('Correo enviado correctamente');
  } catch (error) {
    console.error('Error al enviar el correo:', error);
    throw new Error('Error al enviar el correo');
  }
};

module.exports = { sendVerificationEmail };

```

Anexo K

Tu Perfil de usuario con diferentes funcionalidades

```

const UserInfo = () => {

  if (!userData || loading) {
    return <div className="text-center mt-10">Cargando...</div>;
  }

  const provincia = provincias.find(prov => prov.id === userData.idProvincia);
  const handleDeactivate = async () => {
    const confirmed = window.confirm("¿Estás seguro de desactivar tu cuenta? Este
    proceso es irreversible.");
    if (confirmed) {
      // Lógica para desactivar la cuenta
      try {
        const usuario= {
          "id": userData.id,
          "Usuario": "UsuarioDesactivado",
          "Estado": 2

```

```

    } // Excluyendo la contraseña
    const response = await fetch("http://localhost:4000/api/usuarios", {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(usuario),
    });
    } else {
      console.error('Error actualizando el usuario:', response.statusText);
    }
  } catch (error) {
    console.error('Error actualizando el usuario:', error);
  }
  console.log('Cuenta desactivada');
}
};

const handleSaveChanges = async () => {
  try {
    const { Password, ...dataToSend } = userData; // Excluyendo la contraseña
    console.log(dataToSend);
    const response = await fetch("http://localhost:4000/api/usuarios", {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(dataToSend),
    });
    if (response.ok) {

    } else {
      console.error('Error actualizando el usuario:', response.statusText);
    }
  } catch (error) {
    console.error('Error actualizando el usuario:', error);
  }
};

const formatDate = (dateString) => {
  return (
    <div className="flex items-center justify-center min-h-screen bg-gray-100">
      <div className="max-w-2xl mx-auto p-6 bg-white shadow-lg rounded-lg">
        <h1 className="text-2xl font-bold mb-4 text-center">Información del
        Usuario</h1>
        <div className="space-y-4">
          {userData.idRol === 2 && (

```

```

        <>
        <div>
            <p className="text-gray-700"><strong>Marca:</strong>
{userData.Marca}</p>
        </div>
        <div>
            <p className="text-gray-700"><strong>Número:</strong>
{userData.Numero}</p>
        </div>
    </>
    )}
    <div>
        <p className="text-gray-700"><strong>Usuario:</strong>
{userData.Usuario}</p>
    </div>
    <div>
        <p className="text-gray-700"><strong>Nombre:</strong>
{userData.Nombre}</p>
    </div>
    <div>
        <p className="text-gray-700"><strong>Apellido:</strong>
{userData.Apellido}</p>
    </div>
    <div>
        <p className="text-gray-700"><strong>Género:</strong>
{userData.Genero}</p>
    </div>
    <div>
        <p className="text-gray-700"><strong>Fecha de Cumpleaños:</strong>
{new Date(userData.fechaCumple).toLocaleDateString()}</p>
    </div>
    <div>
        <p className="text-gray-700"><strong>Provincia:</strong> {provincia ?
provincia.nombreProvincia : 'No encontrada'}</p>
    </div>
</div>
<div className="mt-6 space-x-4 text-center">
    <button
        className="px-4 py-2 bg-red-600 text-white rounded hover:bg-red-700"
        onClick={handleDeactivate}
    >
        Desactivar Cuenta
    </button>
    <button

```

```

        className="bg-primar hover:bg-tercero text-white font-bold py-2 px-4
rounded"
        onClick={handleEdit}
    >
        Editar
    </button>
    <button
        className="px-4 py-2 bg-gray-600 text-white rounded hover:bg-gray-
700"
        onClick={handleChangePassword}
    >
        Cambiar Contraseña
    </button>
</div>
</div>

{/* Modal */}
{modalOpen && (
    <div className="fixed top-0 right-0 bottom-0 left-0 z-50 flex items-
center justify-center">
        <div className="absolute w-full h-full bg-gray-900 opacity-50"></div>
        <div className="relative w-full max-w-md bg-white rounded-lg shadow-lg
z-50">
            <div className="flex justify-between items-center px-4 py-3 bg-gray-
100 border-b">
                <h3 className="text-lg font-semibold">Editar Usuario</h3>
                <button onClick={closeModal}>
                    <svg className="w-6 h-6 text-gray-700" fill="none"
stroke="currentColor" viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg">
                        <path strokeLinecap="round" strokeLinejoin="round"
strokeWidth="2" d="M6 18L18 6M6 6l12 12"></path>
                    </svg>
                </button>
            </div>
            <form onSubmit={handleSubmit} className="p-4 space-y-4">
                {userData.idRol === 2 && (
                    <>
                        <div>
                            <label className="block text-sm font-medium text-gray-
700">Marca</label>
                            <input
                                type="text"
                                className="mt-1 block w-full border-gray-300 rounded-md
shadow-sm"
                                value={userData.Marca}

```

```

        onChange={(e) => setUserData({ ...userData, Marca:
e.target.value })}
        />
      </div>
      <div>
        <label className="block text-sm font-medium text-gray-
700">Número</label>
        <input
          type="text"
          className="mt-1 block w-full border-gray-300 rounded-md
shadow-sm"
          value={userData.Numero}
          onChange={(e) => setUserData({ ...userData, Numero:
e.target.value })}
          pattern="\d{10}"
          title="El número debe tener exactamente 10 dígitos."
        />
        {formErrors.Numero && (
          <p className="mt-2 text-sm text-red-
600">{formErrors.Numero}</p>
        )}
      </div>
    </>
  )}
  <div>
    <label className="block text-sm font-medium text-gray-
700">Usuario</label>
    <input
      type="text"
      className="mt-1 block w-full border-gray-300 rounded-md shadow-
sm"
      value={userData.Usuario}
      onChange={(e) => setUserData({ ...userData, Usuario:
e.target.value })}
    />
  </div>
  <div>
    <label className="block text-sm font-medium text-gray-
700">Nombre</label>
    <input
      type="text"
      className="mt-1 block w-full border-gray-300 rounded-md shadow-
sm"
      value={userData.Nombre}

```

```

        onChange={(e) => setUserData({ ...userData, Nombre:
e.target.value })}
      />
    </div>
    <div>
      <label className="block text-sm font-medium text-gray-
700">Apellido</label>
      <input
        type="text"
        className="mt-1 block w-full border-gray-300 rounded-md shadow-
sm"
        value={userData.Apellido}
        onChange={(e) => setUserData({ ...userData, Apellido:
e.target.value })}
      />
    </div>
    <div>
      <label className="block text-sm font-medium text-gray-
700">Género</label>
      <select
        className="mt-1 block w-full border-gray-300 rounded-md shadow-
sm"
        value={userData.Genero}
        onChange={(e) => setUserData({ ...userData, Genero:
e.target.value })}
      >
        <option value="Masculino">Masculino</option>
        <option value="Femenino">Femenino</option>
        <option value="Sin Preferencia">Sin Preferencia</option>
      </select>
    </div>
    <div>
      <label className="block text-sm font-medium text-gray-700">Fecha
de Cumpleaños</label>
      <input
        type="date"
        className="mt-1 block w-full border-gray-300 rounded-md shadow-
sm"
        value={formatDate(userData.fechaCumple)}
        onChange={(e) => setUserData({ ...userData, fechaCumple:
e.target.value })}
      />
    </div>
  </div>

```

```

        <label className="block text-sm font-medium text-gray-
700">Provincia</label>
        <select
            className="mt-1 block w-full border-gray-300 rounded-md shadow-
sm"
            value={userData.idProvincia}
            onChange={(e) => setUserData({ ...userData, idProvincia:
parseInt(e.target.value, 10) })}
            >
            {provincias.map((prov) => (
                <option key={prov.id} value={prov.id}>
                    {prov.nombreProvincia}
                </option>
            ))}
        </select>
    </div>
    <div className="flex justify-end mt-4">
        <button
            type="submit"
            className="bg-primar hover:bg-tercero text-white font-bold py-2
px-4 rounded"
            >
            Guardar Cambios
        </button>
    </div>
</form>
</div>
    )}
</div>
);
};
export default UserInfo;

```

Anexo L

Cambio de contraseña

```

import React, { useState } from "react";
import { useLocation,useNavigate } from "react-router-dom";
import { RiLockLine, RiEyeLine, RiEyeOffLine } from "react-icons/ri";
import { toast } from 'react-toastify';

const CambiarPassword = () => {
    const [formData, setFormData] = useState({
        Nombre: "",

```

```

    Apellido: "",
    Usuario: "",
    Correo: "",
    Password: "",
    ConfirmarPassword: "",
    Year: "",
    Mes: "",
    day: "",
    genero: "Masculino",
    Provincia: "",
    ProvinciaId: "",
  });
const [showPassword, setShowPassword] = useState(false);
const navigate = useNavigate();
const location = useLocation();
const DatosUsuario = location.state || {};

const handleChange = (e) => {
  const { name, value } = e.target;
  setFormData({ ...formData, [name]: value });
};

const isPasswordSecure = (password) => {
  const passwordRegex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&.-_])[A-Za-z\d@$!%*?&.-_]{8,}$/;
  return passwordRegex.test(password);
};

const handleSubmit = async (e) => {
  e.preventDefault();
  // Aquí puedes agregar la lógica para validar las contraseñas
  const { Password, ConfirmarPassword } = formData;
  if ( !Password || !ConfirmarPassword ) {
    setError("Por favor completa todos los campos");
    return;
  }
  // y enviar una solicitud para cambiar la contraseña
  if (formData.Password !== formData.ConfirmarPassword) {
    setError("Las contraseñas no coinciden");
    return;
  }
  if (!isPasswordSecure(formData.Password)) {
    setError("La contraseña debe tener al menos 8 caracteres y contener al menos una letra minúscula, una letra mayúscula, un número y un carácter especial.");
    return;
  }
}

```

```

// Después de cambiar la contraseña, navega a otra página
const datosUsuario = {
  id: DatosUsuario.id,
  Password
};
console.log(DatosUsuario);
try {
  // Enviar los datos al backend
  const response = await fetch("http://localhost:4000/api/usuarios", {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
    },
    body: JSON.stringify(datosUsuario),
  });

  // Manejar la respuesta del backend
  if (response.ok) {
    // El usuario se registró exitosamente
    toast.success('Contraseña cambiada exitosamente!', {
      position: "bottom-left",
      autoClose: 2000,
      hideProgressBar: true,
      closeOnClick: true,
      pauseOnHover: true,
      draggable: true,
      progress: undefined,
      theme: "light",
      style: { backgroundColor: '#10B981', color: '#FFFFFF' }
    });
    console.log("Usuario cambiado contraseña correctamente");
    // Redireccionar a la página de inicio de sesión
    navigate("/login");
  } else {
    // Hubo un error al registrar el usuario
    console.error("Error al registrar usuario:", response.statusText);
  }
} catch (error) {
  console.error("Error al enviar datos al backend:", error);
}
};

return (
  <div className="min-h-screen flex flex-col">

```

```

<header className="bg-background py-4 flex justify-center items-center">
  <a
    href="/"
    className="text-2xl font-bold relative p-1 bg-background text-white"
  >
    TradArte<span className="text-primar text-5xl">.</span>{" "}
  </a>
</header>
<div className="flex-1 flex items-center justify-center p-4">
  <div className="bg-secondary-100 p-8 rounded-xl shadow-2xl w-auto lg:w-[450px]">
    <h1 className="text-3xl text-center uppercase font-bold tracking-[5px] text-cuarto mb-8">
      Cambiar <span className="text-primar">Contraseña</span>
    </h1>
    <form className="mb-8" onSubmit={handleSubmit}>
      <div className="relative mb-4">
        <RiLockLine className="absolute top-1/2 -translate-y-1/2 left-2 text-primar " />
        <input
          type={showPassword ? "text" : "password"}
          name="Password"
          className="py-3 px-8 bg-secondary-900 w-full outline-none rounded-lg border border-gray-300 focus:ring-secondary focus:border-secondary"
          placeholder="Contraseña"
          value={formData.Password}
          onChange={handleChange}
        />
        {showPassword ? (
          <RiEyeOffLine
            onClick={() => setShowPassword(!showPassword)}
            className="absolute top-1/2 -translate-y-1/2 right-2 hover:cursor-pointer text-primar "
          />
        ) : (
          <RiEyeLine
            onClick={() => setShowPassword(!showPassword)}
            className="absolute top-1/2 -translate-y-1/2 right-2 hover:cursor-pointer text-primar"
          />
        )}
      </div>
      <div className="relative mb-8">
        <RiLockLine className="absolute top-1/2 -translate-y-1/2 left-2 text-primar" />

```

```

        <input
          type={showPassword ? "text" : "password"}
          name="ConfirmarPassword"
          className="py-3 px-8 bg-secondary-900 w-full outline-none
rounded-lg border border-gray-300 focus:ring-secondary focus:border-secondary"
          placeholder="Confirmar contraseña"
          value={formData.ConfirmarPassword}
          onChange={handleChange}
        />
        {showPassword ? (
          <RiEyeOffLine
            onClick={() => setShowPassword(!showPassword)}
            className="absolute top-1/2 -translate-y-1/2 right-2
hover:cursor-pointer text-primar"
          />
        ) : (
          <RiEyeLine
            onClick={() => setShowPassword(!showPassword)}
            className="absolute top-1/2 -translate-y-1/2 right-2
hover:cursor-pointer text-primar"
          />
        )}
      </div>
      <div>
        <button
          type="submit"
          className="bg-primar text-white uppercase font-bold text-sm w-
full py-3 px-4 rounded-lg"
        >
          Cambiar Contraseña
        </button>
      </div>
    </form>
  </div>
</div>
);
};

export default CambiarPassword;

```

Anexo M

Resultados respuestas abiertas encuesta satisfacción

¿Qué le gustó más del sitio web?

12 respuestas

Todo

La facilidad con la que podías comprar las cosas

Fácil de comprar las cosas

El detalle

La facilidad de adquirir productos

Su desempeño

Los productos

Los tipos de productos que ofrece

Su funcionalidad

La facilidad de poder comprar

La facilidad para comprar

¿Qué le gustaría mejorar en el sitio web?

12 respuestas

La página tiene todo lo necesario

Que exista más variedad

Nada

Más productos

Un catálogo más amplio

Nada

La tipografía

Nada

Más contenidos

Como tal, nada

La variedad

Que sea más didáctica

¿Tiene algún otro comentario o sugerencia?

12 respuestas

No

No

Ninguno

¿Cómo describiría su experiencia general con el sitio web en una frase?

11 respuestas

Excelente

Estuvo muy buena

Muy Buena

Bueno

Fácil de usar y entender

Agradable

Satisfecho

Divertido, con cosas alternativas

Genial

Muy buena