

Pregrado

Carrera: Sistemas y Gestión de Data

Asignatura (UIC): Ejecución de Proyectos en TIC'S

Trabajo de titulación previo a la obtención del

Título en:

Tecnólogo Universitario en Sistemas y

Gestión de Data

Tema: Desarrollo de una plataforma Web para la

conexión de servicios y productos locales con enfoque

en la reactivación económica

Autor/s: Oscar Ivan Caisatoa Suntaxi

Tutor: Mg. Danny Santiago Páez Oscullo

Fecha: septiembre 2025





Autor: Caisatoa Suntaxi Oscar Ivan

Título a obtener: Tecnólogo Universitario en Sistemas y Gestión de Data.

Matriz: Sangolquí -Ecuador

Correo electrónico: oscar.caisatoa@ister.edu.ec



Dirigido por: Páez Oscullo Danny Santiago

Título: Ing. Computación Gráfica.

Mg. Lógica, Computación e Inteligencia Artificial.

Matriz: Sangolquí -Ecuador

Correo electrónico: danny.paez@ister.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

©2025 Tecnológico Universitario
Rumiñahui SANGOLQUÍ – ECUADOR
Caisatoa Suntaxi Oscar Ivan

CARTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-2-2.3

Sangolquí, (01) de (septiembre) del 2025

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

**INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE
UNIVERSITARIO**

Presente

Por medio de la presente, yo, Caisatoa Suntaxi Oscar Ivan declaro y acepto en forma expresa lo siguiente: Ser autor del trabajo de titulación denominado Desarrollo de una plataforma web para la conexión de servicios y productos locales con enfoque en la reactivación económica, de la Tecnología Superior Universitaria Sistemas y Gestión de Data; y a su vez manifiesto mi voluntad de ceder al Instituto Superior Tecnológico Rumiñahui con condición de Universitario los derechos de reproducción, distribución y publicación de dicho trabajo de titulación, en cualquier formato y medio, con fines académicos y de investigación.

Esta cesión se otorga de manera no exclusiva y por un periodo indeterminado. Sin embargo, conservo los derechos morales sobre mi obra.

En fe de lo cual, firmo la presente.

Atentamente,



OSCAR IVAN CAISATOA SUNTAXI
C.I.: 1727581231

FORMULARIO PARA ENTREGA DE PROYECTOS EN BIBLIOTECA INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

CT-ANX-2025-ISTER-3

CARRERA: TECNOLOGÍA UNIVERSITARIA EN SISTEMAS Y GESTIÓN DE DATA

AUTOR/ES: OSCAR IVAN CAISATOA SUNTAXI

TUTOR: DANNY SANTIAGO PAEZ OSCULLO

CONTACTO ESTUDIANTE: 0995548070

CORREO ELECTRÓNICO: oscar.caisatoa@ister.edu.ec

TEMA: Desarrollo de una plataforma web para la conexión de servicios y productos locales con enfoque en la reactivación económica

OPCIÓN DE TITULACIÓN: UNIDAD DE INTEGRACIÓN CURRICULAR.

Resumen

El objetivo central de este proyecto es crear una plataforma web que vincule a quienes ofrecen servicios o productos con los usuarios que los requieren dentro de su comunidad, impulsando así la reactivación económica mediante tecnologías modernas y accesibles. Surgida de la necesidad de brindar una solución digital asequible para emprendedores, trabajadores independientes y consumidores que buscan alternativas confiables en su entorno cercano y respaldada por los conocimientos adquiridos en la carrera de Sistemas y Gestión de Data del Instituto Tecnológico Universitario “Rumiñahui”, la propuesta, titulada “**Desarrollo de una Plataforma Web para la Conexión de Servicios y Productos Locales con Enfoque en la Reactivación Económica Comunitaria,**” integra los principios del Plan Operativo Anual (POA)

y la Economía Popular y Solidaria para dinamizar los mercados barriales. Construida con Laravel 8, PHP 8 y MySQL, la solución funciona como una MegaStore escalable de uno a múltiples proveedores y permite comercializar productos físicos, digitales o con licencia. Entre sus más de cien prestaciones destacan marketing de afiliados, planes de suscripción, logística con repartidores, punto de venta (POS), gestión de inventario, pagos por reembolso y paneles independientes para administradores, usuarios, proveedores y repartidores. Todo ello facilita que microempresas y cooperativas locales abran vitrinas digitales, generen ingresos recurrentes y amplíen su alcance sin grandes inversiones publicitarias. Desarrollada bajo la metodología ágil XP, con buenas prácticas de seguridad y un diseño responsive compatible con los principales navegadores, la plataforma demuestra la capacidad de las tecnologías web actuales para fortalecer las redes de comercialización solidarias y contribuir de manera tangible al desarrollo económico comunitario.

Palabras clave: E-commerce Laravel, Marketplace multiproveedor, Afiliados, POS, Repartidores, Inventario, Reactivación local.

Abstract:

The central objective of this project is to create a web platform that connects those who offer services or products with the users who need them within their community, thus driving economic recovery through modern and accessible technologies. Born from the need to provide an affordable digital solution for entrepreneurs, independent workers, and consumers seeking reliable alternatives in their immediate surroundings, and supported by the knowledge acquired in the Systems and Data Management program at the Rumiñahui University Technological

Institute, the proposal, entitled "Development of a Web Platform for Connecting Local Services and Products with a Focus on Community Economic Recovery," integrates the principles of the Annual Operating Plan (AOP) and the Popular and Solidarity Economy to boost neighborhood markets. Built with Laravel 8, PHP 8, and MySQL, the solution functions as a scalable MegaStore from one to multiple providers and allows for the sale of physical, digital, or licensed products. Its more than 100 features include affiliate marketing, subscription plans, logistics with delivery drivers, point of sale (POS), inventory management, cashback payments, and separate dashboards for administrators, users, suppliers, and delivery drivers. All of this makes it easier for micro-businesses and local cooperatives to open digital storefronts, generate recurring revenue, and expand their reach without significant advertising investments. Developed using the agile XP methodology, with best security practices and a responsive design compatible with major browsers, the platform demonstrates the ability of current web technologies to strengthen solidarity marketing networks and make a tangible contribution to community economic development.

Keywords: Laravel e-commerce, multi-vendor marketplace, Affiliates, POS, Delivery Drivers, Inventory, Local revitalization.



Firma del Estudiante (Autor)

OSCAR IVAN CAISATOA SUNTAXI

C.I.: 1727581231

SOLICITUD DE PUBLICACIÓN DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-4

Sangolquí, 01 de septiembre del 2025

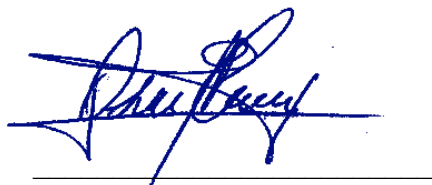
Sres.-

INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

Yo, OSCAR IVAN CAISATOA SUNTAXI, con C.I.: 1727581231 alumno de la Carrera TECNÓLOGO UNIVERSITARIO EN SISTEMAS Y GESTIÓN DE DATA, cedo al INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO, los derechos de publicaciones del presente trabajo de Titulación en el Repositorio Institucional para hacer uso de todos los contenidos con fines estrictamente académico o de investigación.

Atentamente,



Firma del Estudiante

Oscar Ivan Caisatoa Suntaxi

C.I.: 1727581231

SÓLO PARA USO DEL ISTER

Han sido revisadas las similitudes del trabajo en el software “TURNITING” y cuenta con un porcentaje de; motivo por el cual, el Proyecto Técnico de Titulación es publicable. (EL PORCENTAJE DE SIMILITUD DEBE SER MÁXIMO DE 15%)

MSc. Elizabeth Ordoñez

**DIRECTORA DE DOCENCIA
TITULACIÓN**

MSc. Mónica Loachamín

COORDINADORA

DE

Fecha del Informe / ____/

Contenido

Resumen.....	4
Abstract:	5
1. CAPITULO1	14
1.1. Introducción	14
1.2. El Problema de Investigación (Antecedentes).	16
1.3. Planteamiento del Problema. (Formulación problemática)	17
1.4. Justificación	19
1.4.1. Social.....	19
1.4.2. Técnica.....	20
1.5. Objetivos	21
1.5.1. General.....	21
1.5.2. Específicos	21
1.6. Alcance y limitaciones.....	21
1.7. Marco referencial	22
1.7.1. Gestión de Proyectos de Desarrollo de Software.....	23

1.7.2.	Tecnologías para el Desarrollo Web.....	24
1.7.3.	Adopción y Uso de Tecnologías en Contextos Locales.....	24
1.8.	Marcos de Referencia para la Gestión de Proyectos en Ecuador.....	24
1.8.1.	Principios para el Diseño y Uso de Tecnologías Emergentes.....	24
1.9.	Marco legales	25
1.9.1.	Reglamento para la adquisición de software en el sector público	25
1.9.2.	Política de Software Libre y Software Público	25
1.10.	Marcos políticos y sociales	26
2.	Marco Teórico.....	27
2.1.	Fundamentación Teórica del tema del proyecto	27
2.1.1.	Fundamentación Legal.....	28
2.1.2.	Fundamentación Teórica.....	29
2.2.	Fundamentación Teórica de las Herramientas de Desarrollo	31
3.	Marco Metodológico.....	36
3.1.	Enfoque de la investigación	36
3.2.	Tipo y diseño de investigación	37

3.3.	Ámbito, población y muestra	37
3.4.	Técnicas e instrumentos de recolección de información	38
3.4.1.	Encuestas (CSAT/NPS y hábitos de uso)	38
3.5.	Tipo y diseño de investigación	46
3.6.	Ámbito, población y muestra	46
3.7.	Técnicas e instrumentos de recolección.....	47
3.8.	Variables e indicadores	47
3.9.	Plan de análisis de datos	48
4.	Desarrollo y Resultados	49
4.1.	Descripción general del sistema.....	49
4.2.	Justificación de Laravel y Blade	53
4.3.	Impacto en el proyecto.....	53
4.4.	Modelo de datos (E/R)	53
4.5.	Diseño de la interfaz y UX.....	54
4.6.	Módulos funcionales (CRUD y flujo).....	54
4.6.1.	Historias de Usuario del Sistema Web Comunitario	55

4.7.	Planificación de las Iteraciones.....	60
4.7.1.	Planificación de las Iteraciones.....	61
4.7.2.	Ejecución de las Iteraciones.....	62
4.8.	Migraciones y controladores (criterios de implementación)	64
4.9.	Flujo de pago por WhatsApp (evidencia y control).....	91
5.	Referencias.....	92
ANEXOS		94
MANUAL DE USUARIO		95
Manual del Desarrollador — <i>Importadora Caisatoa</i> (Laravel 12).....		130

1. CAPITULO1

1.1.Introducción

En Ecuador, las dinámicas comunitarias y económicas han sufrido importantes transformaciones en los últimos años, especialmente a raíz de factores como la pandemia, el desempleo y el crecimiento del trabajo informal, muchas personas han optado por ofrecer servicios y productos de forma independiente, generando nuevas formas de comercio local, estas actividades suelen carecer de herramientas digitales adecuadas para facilitar la visibilidad, la contratación y la organización eficiente de la oferta y la demanda en su entorno cercano, trabajadores independientes, pequeños emprendedores y prestadores de servicios locales enfrentan dificultades para promocionar su trabajo, mantener un contacto directo con posibles clientes o gestionar sus servicios de manera ordenada. Esta situación se ve agravada por la ausencia de plataformas diseñadas específicamente para la conexión entre personas que ofrecen servicios y quienes los buscan dentro de su misma comunidad. Es aquí donde surge la necesidad

de una solución tecnológica que promueva la economía colaborativa, facilite la contratación segura y fomente la reactivación económica desde un enfoque local.

La implementación de una plataforma web funcional y accesible se convierte así en una herramienta clave para responder a esta necesidad, una solución que no solo permita crear publicaciones detalladas de servicios o productos, sino también ofrezca funciones de búsqueda con filtros por ubicación y categoría, contacto directo entre usuarios y paneles personalizados según el tipo de usuario (cliente o vendedor). Este tipo de sistema ayuda a optimizar el proceso de búsqueda, contratación y organización del trabajo local, mejorando la experiencia tanto para quienes ofrecen como para quienes demandan estos servicios.

El proyecto "Desarrollo de una Plataforma Web para la Conexión de Servicios y Productos Locales con Enfoque en la Reactivación Económica Comunitaria" ha sido concebido para atender esta problemática. Se sustenta en tecnologías modernas como Laravel, Vue.js, Tailwind CSS y MySQL, incorporando un sistema de autenticación mediante Laravel Breeze y diferenciación de roles, la arquitectura del sistema es escalable, lo que permite su evolución futura e integración con otras herramientas digitales si así lo requieren los usuarios o el contexto tecnológico.

Este documento se organiza de la siguiente manera: primero, se presenta el planteamiento del problema, donde se analizan las condiciones sociales y tecnológicas que motivaron este proyecto. Luego, se desarrolla el marco teórico, que respalda los conceptos y herramientas utilizadas. A continuación, se explica la metodología empleada, seguida de los resultados

alcanzados y se incluyen las conclusiones y recomendaciones que permiten valorar el impacto y las posibilidades de mejora de esta plataforma.

1.2.El Problema de Investigación (Antecedentes).

En el contexto actual, los pequeños emprendedores, trabajadores independientes y comerciantes locales enfrentan dificultades significativas para promocionar y comercializar sus productos y servicios de manera eficiente dentro de sus comunidades. La mayoría de estos actores económicos dependen de métodos tradicionales y poco integrados, como el uso de redes sociales personales o canales informales, que limitan su alcance y dificultan la generación de confianza con los clientes (Dialnet, 2023). Esta situación se agrava en un entorno donde las ventas en tiendas de barrio han disminuido, como se ha evidenciado en Ecuador, donde se reporta una caída del 10% en las ventas durante el año 2024, afectando directamente la economía local y la estabilidad de estos negocios (Dichter & Neira, 2024).

La falta de una plataforma digital accesible y adaptada a las necesidades locales representa un obstáculo para la reactivación económica comunitaria. Los sistemas tradicionales no ofrecen integración entre la gestión de productos, la comunicación directa con clientes ni herramientas para filtrar y buscar servicios según ubicación o categoría, lo que genera ineficiencias y limita el crecimiento de los pequeños negocios (Involves, 2024). Además, la baja adopción de tecnologías modernas y la escasa capacitación tecnológica en estos sectores dificultan la transición hacia modelos de comercio digital más competitivos y sostenibles (Dialnet, 2023).

Investigaciones recientes han demostrado que la implementación de plataformas digitales específicas para comunidades puede mejorar la visibilidad y el acceso a servicios y productos locales, fomentando la confianza y facilitando la interacción directa entre oferentes y demandantes. Por ejemplo, el estudio de plataformas digitales en el cantón Paján evidenció que, aunque los comerciantes utilizan redes sociales como Facebook e Instagram para vender, carecen de conocimientos para aprovechar estas herramientas de forma efectiva, lo que limita su potencial comercial (Dialnet, 2023). Esto resalta la necesidad de soluciones digitales integrales que incluyan funcionalidades como publicación de productos con imágenes, descripciones, ubicación y contacto directo, optimizando así la experiencia del usuario y la gestión comercial.

Por otro lado, la adopción de tecnologías modernas como Laravel, Vue.js, Tailwind CSS y bases de datos robustas como MySQL, junto con metodologías ágiles para el desarrollo, ha demostrado ser efectiva para crear plataformas web escalables, seguras y adaptadas a las necesidades de usuarios con distintos perfiles técnicos (Mendoza et al., 2021). Estos avances tecnológicos permiten superar las limitaciones de los sistemas tradicionales y ofrecen una oportunidad para impulsar la reactivación económica local mediante la digitalización.

1.3. Planteamiento del Problema. (Formulación problemática)

En Ecuador, los pequeños emprendedores, trabajadores independientes y comerciantes locales enfrentan serios desafíos en la gestión y promoción de sus productos y servicios dentro de sus comunidades. Estos actores económicos suelen depender de métodos tradicionales, poco integrados y manuales, para difundir sus ofertas, establecer contacto con clientes y gestionar sus actividades comerciales. Este enfoque fragmentado y poco eficiente genera una serie de

problemas que afectan negativamente tanto su rendimiento operativo como su competitividad en el mercado local.

Uno de los problemas más críticos es la limitada visibilidad y alcance de los productos y servicios ofrecidos, lo que dificulta la conexión entre oferentes y demandantes en un entorno cercano. Esta falta de una plataforma digital accesible y adaptada a las necesidades locales ralentiza el flujo comercial, reduce las oportunidades de negocio y genera frustración tanto en los vendedores como en los consumidores, quienes demandan rapidez, confianza y facilidad para encontrar opciones adecuadas a sus necesidades, la ausencia de herramientas tecnológicas integradas que permitan gestionar publicaciones con imágenes, descripciones, zonas geográficas y precios, así como la posibilidad de contactar directamente con los oferentes mediante canales como WhatsApp o llamadas telefónicas, limita la eficiencia y la experiencia del usuario. Esta carencia conduce a una dispersión de la información, pérdida de oportunidades comerciales y una baja fidelización de los clientes, la falta de capacitación tecnológica y recursos para implementar soluciones digitales modernas restringe la adopción de tecnologías que podrían optimizar estos procesos. La carencia de un sistema que integre funcionalidades como paneles personalizados para la gestión de actividades, filtros de búsqueda por ubicación o categoría, y contacto directo, impide que los pequeños negocios puedan adaptarse a las demandas del mercado digital actual, estas deficiencias tienen un impacto directo en la economía local, ocasionando pérdidas económicas por ventas no realizadas, baja visibilidad y falta de interacción efectiva entre oferentes y clientes. Además, limitan el crecimiento y desarrollo sostenible de los pequeños emprendedores, reduciendo su capacidad para contribuir a la reactivación económica comunitaria.

El presente proyecto se enfoca en analizar y proponer una solución tecnológica mediante el desarrollo de una plataforma web que conecte de manera eficiente a quienes ofrecen y demandan servicios y productos en sus comunidades. El objetivo es optimizar la difusión, gestión y contacto, mejorando la experiencia del usuario y fomentando la reactivación económica local a través del uso de tecnologías modernas y metodologías ágiles.

1.4. Justificación

1.4.1. Social

En Ecuador, alrededor del 70% a 85% de la población tiene acceso a internet, lo que representa una oportunidad significativa para el crecimiento del comercio electrónico y la digitalización de los negocios locales (CECE & UEES, 2025; Statista, 2025). Sin embargo, muchos pequeños emprendedores y trabajadores independientes aún enfrentan dificultades para acceder a plataformas digitales que les permitan promocionar y vender sus productos y servicios de manera eficiente dentro de sus comunidades (Dialnet, 2023). La falta de herramientas adecuadas limita la visibilidad de sus ofertas y reduce las oportunidades de crecimiento económico, afectando directamente la reactivación económica comunitaria.

El desarrollo de una plataforma web que conecte a oferentes y demandantes locales contribuirá a fortalecer el tejido económico de las comunidades, facilitando la generación de ingresos y fomentando la inclusión digital. Además, promoverá la confianza y la interacción directa entre usuarios, lo que puede mejorar la experiencia de compra y venta, y favorecer la sostenibilidad de los pequeños negocios en un contexto marcado por la transformación digital y los retos económicos post-pandemia (Primicias, 2024; Ekos, 2025). Así, esta iniciativa responde

a una necesidad social de impulsar el desarrollo económico local mediante el uso de tecnologías accesibles y adaptadas a la realidad de los usuarios.

1.4.2. Técnica

Desde el punto de vista técnico, la implementación de esta plataforma web se fundamenta en el uso de tecnologías modernas y probadas como Laravel, Vue.js, Tailwind CSS y MySQL, que permiten desarrollar sistemas robustos, escalables y con interfaces amigables para el usuario (Mendoza, Torres & Salazar, 2021). Estas herramientas facilitan la integración de funcionalidades clave como la publicación de productos y servicios con imágenes, descripciones, filtros por ubicación y categoría, y opciones de contacto directo, optimizando la experiencia y usabilidad del sistema.

Además, la adopción de metodologías ágiles en el desarrollo del proyecto garantiza una planificación ordenada y una implementación progresiva, permitiendo adaptarse a las necesidades cambiantes de los usuarios y mejorar continuamente el producto final. La plataforma propuesta no solo automatiza y centraliza la gestión de ofertas y demandas, sino que también contribuye a la digitalización de los procesos comerciales en comunidades que tradicionalmente han dependido de métodos manuales o poco eficientes, reduciendo errores y mejorando la eficiencia operativa (Dialnet, 2023).

En conclusión, este proyecto representa una solución técnica innovadora y accesible que aprovecha las ventajas del comercio electrónico para potenciar la reactivación económica local, facilitando la conexión efectiva entre quienes ofrecen y quienes demandan productos y servicios en sus comunidades.

1.5.Objetivos

1.5.1. General

- Desarrollar una plataforma web comunitaria con paneles por rol y verificación de pagos mediante envío de recibo/factura y comprobante por WhatsApp.con el objetivo de promover la reactivación económica comunitaria.

1.5.2. Específicos

- Analizar la problemática actual de trabajadores independientes, emprendedores y buscadores de servicios locales, con el fin de identificar las necesidades tecnológicas y sociales que orienten los requerimientos funcionales y no funcionales de la plataforma.
- Diseñar una solución digital integral que incluya módulos para el registro de usuarios con roles diferenciados (cliente y vendedor), publicación de servicios o productos, buscador con filtros y medios de contacto directo, asegurando una interfaz amigable y funcional.
- Implementar la plataforma utilizando tecnologías como Laravel, y MySQL, integrando autenticación segura con Laravel Breeze, control de acceso por roles y una arquitectura escalable que permita el crecimiento del sistema.
- Validar el funcionamiento de la plataforma mediante pruebas funcionales y de experiencia de usuario en un entorno controlado, con el propósito de detectar errores, medir su rendimiento y asegurar el cumplimiento de los objetivos antes de su implementación definitiva.

1.6. Alcance y limitaciones

El presente proyecto abarca el desarrollo de una plataforma web funcional orientada a facilitar la conexión entre personas que ofrecen servicios o productos locales y aquellos que los requieren, priorizando la facilidad de uso, la rapidez en la búsqueda y el contacto directo. La solución incluye un sistema de autenticación con roles diferenciados (cliente y vendedor), permitiendo a los vendedores publicar servicios o productos con información relevante como título, descripción, precio, zona y categoría.

El cliente, por su parte, podrá acceder a las publicaciones a través de un buscador inteligente con filtros dinámicos por tipo de servicio, ubicación y categoría. Cada publicación contará con opciones de contacto directo mediante WhatsApp o llamada telefónica desde dispositivos móviles, promoviendo una interacción ágil y eficaz entre las partes del sistema. Estará desarrollado utilizando tecnologías modernas como Laravel para el backend y el frontend dinámico, MySQL como sistema gestor de base de datos, se empleará Laravel Breeze para la autenticación y se incorporará un panel de usuario personalizado para cada rol, permitiendo a los vendedores gestionar sus publicaciones y visualizar estadísticas básicas, y a los clientes acceder a su historial de interacciones o servicios favoritos.

El alcance no contempla en esta etapa funcionalidades como pasarelas de pago integradas, control avanzado de disponibilidad de servicios, conexión con plataformas externas, el sitio web si se contemplara en su uso en internet y se dará a conocer en la web, el sistema será construido con una arquitectura escalable que permitirá incorporar otras características futuras según las necesidades de la comunidad usuaria.

1.7.Marco referencial

El desarrollo de plataformas web ha evolucionado significativamente con la incorporación de frameworks modernos que facilitan la creación de aplicaciones interactivas, seguras y escalables. Laravel, un framework PHP de código abierto, ha ganado popularidad debido a su estructura basada en el patrón MVC (Modelo-Vista-Controlador), su sintaxis clara y su robusto ecosistema, permitiendo el desarrollo de aplicaciones web complejas con rapidez y eficiencia. Entre sus características más destacadas se encuentran su sistema de autenticación, migraciones de base de datos, enrutamiento y compatibilidad con herramientas modernas como Laravel Breeze para autenticación sencilla el framework progresivo de JavaScript utilizado para construir interfaces de usuario dinámicas. Su integración con Laravel permite la creación de experiencias front-end reactivas sin comprometer la estructura del backend. facilita la manipulación del DOM de forma eficiente, la creación de componentes reutilizables y una comunicación fluida con las API desarrolladas en Laravel y base de datos MySQL es un sistema de gestión de bases de datos relacional ampliamente utilizado en entornos web, gracias a su estabilidad, rendimiento y compatibilidad con múltiples tecnologías. Su combinación con Laravel permite una gestión eficaz de datos mediante Eloquent ORM, lo que agiliza la interacción con la base de datos sin necesidad de consultas SQL complejas.

1.7.1. Gestión de Proyectos de Desarrollo de Software

Para la gestión del proyecto se toma como referencia el estándar PMBOK, que contempla cinco áreas de conocimiento fundamentales: gestión de la integración, alcance, tiempo, recursos humanos y comunicaciones, para asegurar una planificación adecuada, control de cambios y cumplimiento de objetivos (Chávez, 2021). Este marco ayuda a evitar problemas comunes como falta de planificación, retrasos y cambios de especificaciones durante el desarrollo.

1.7.2. Tecnologías para el Desarrollo Web

El proyecto utiliza tecnologías modernas de desarrollo web como Laravel (framework PHP), (framework JavaScript), Tailwind CSS (framework CSS) y MySQL (base de datos relacional), que permiten construir plataformas escalables, seguras y con interfaces amigables para el usuario. Estas tecnologías son ampliamente adoptadas en proyectos de software modernos y facilitan la integración de funcionalidades como publicación de productos, filtros de búsqueda y comunicación directa entre usuarios (Mendoza et al., 2021).

1.7.3. Adopción y Uso de Tecnologías en Contextos Locales

Diversos estudios sobre la incorporación de tecnologías de información y comunicación (TIC) en comunidades y pequeñas empresas muestran que la falta de capacitación, recursos y modelos adaptados a las necesidades locales limita la adopción efectiva de soluciones digitales (Cevallos, 2020). El marco referencial incluye enfoques que promueven la inclusión digital, la usabilidad y la adaptabilidad de las plataformas a contextos comunitarios.

1.8. Marcos de Referencia para la Gestión de Proyectos en Ecuador

La realidad ecuatoriana demanda marcos de referencia que consideren las particularidades de las PYMES y proyectos tecnológicos locales. Estudios como el de Villavicencio et al. (2007) proponen marcos basados en modelos internacionales (Moprosoft, CMMI, ISO/IEC 15504) adaptados a las condiciones del país para mejorar la gestión de proyectos y la competitividad (Villavicencio et al., 2007).

1.8.1. Principios para el Diseño y Uso de Tecnologías Emergentes

La adopción de tecnologías debe alinearse con principios de inclusión, transparencia, seguridad, innovación y sustentabilidad social, garantizando que las soluciones tecnológicas generen valor para los usuarios y el beneficio público (RedGEALC, 2023). Estos principios guían el diseño de la plataforma para asegurar que sea accesible, confiable y adaptable a las necesidades de la comunidad.

1.9.Marco legales

El desarrollo y la implementación de la plataforma web para la conexión de servicios y productos locales con enfoque en la reactivación económica comunitaria debe cumplir con un marco legal que garantice la legalidad, seguridad, transparencia y fomento de la innovación tecnológica en Ecuador.

1.9.1. Reglamento para la adquisición de software en el sector público

El Decreto Ejecutivo 1073, emitido el 12 de junio de 2020, regula la adquisición de software por parte de las entidades del sector público, priorizando el desarrollo y uso de software con valor agregado ecuatoriano, especialmente software de código abierto. Este reglamento busca incentivar la innovación local, facilitar la contratación pública y promover la soberanía tecnológica, estableciendo un orden de prelación para la adquisición de software que favorece los productos nacionales y de código abierto (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2020).

1.9.2. Política de Software Libre y Software Público

Ecuador ha adoptado el modelo de Software Libre como política tecnológica para promover la inclusión digital, la soberanía tecnológica y la innovación local. A través de

repositorios como MINKA, se facilita el acceso y desarrollo colaborativo de software público, lo que optimiza recursos estatales y fortalece el desarrollo tecnológico nacional (Gobierno Electrónico Ecuador, 2023). Esta política es fundamental para el desarrollo de plataformas digitales que buscan ser accesibles, seguras y sostenibles.

1.10. Marcos políticos y sociales

La transformación digital en Ecuador se ha consolidado como una prioridad nacional para impulsar el desarrollo económico, social y cultural, con un enfoque inclusivo que busca reducir brechas y promover la participación ciudadana en la era digital. La Política Pública para la Transformación Digital del Ecuador 2025-2030 establece un marco estratégico que articula infraestructura tecnológica, capacidades humanas y marcos normativos orientados a la innovación, eficiencia y democratización del acceso a servicios y datos públicos (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2025). Esta política se basa en principios fundamentales como la solidaridad, inclusión, libertad de elección, participación ciudadana en el espacio digital, seguridad y sostenibilidad. Busca fortalecer la infraestructura digital, fomentar la cultura e inclusión digital, impulsar la economía digital, promover tecnologías emergentes para el desarrollo sostenible, consolidar el gobierno digital, mejorar la interoperabilidad y tratamiento de datos, y reforzar la seguridad digital y la confianza (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2025).

Un ejemplo concreto de estas políticas es la inauguración de espacios como los Puntos Digitales Gratuitos, que ofrecen acceso a internet y capacitaciones en más de 400 temáticas relacionadas con tecnologías de la información y comunicación (TIC), incluyendo comercio

electrónico, programación y diseño gráfico. Estas iniciativas buscan empoderar a la población local, fortalecer emprendimientos y mejorar la calidad de vida mediante el uso de la tecnología, especialmente en zonas rurales y comunidades con menor acceso a recursos digitales (MINTEL, 2025), la inclusión digital en la educación es otro pilar fundamental, donde la Agenda Educativa Digital 2021-2025 promueve la equidad y la integración de las TIC en los procesos educativos para garantizar una formación adecuada en competencias digitales, contribuyendo a la reducción de brechas sociales y al desarrollo integral de los ciudadanos (Ministerio de Educación, 2022).

CAPITULO II

2. Marco Teórico

2.1.Fundamentación Teórica del tema del proyecto

El desarrollo de plataformas web para la conexión de servicios y productos locales representa una estrategia fundamental para fortalecer la economía comunitaria y promover la digitalización de los negocios en la comunidad. Este tipo de soluciones tecnológicas permiten que proveedores de servicios y productos, como emprendedores y pequeños comercios, puedan ofrecer sus servicios y llegar a clientes de manera eficiente a través de internet, facilitando la interacción, la promoción y la contratación directa (Montalvo et al., 2021).

La importancia de las plataformas digitales en el ámbito local se evidencia en su capacidad para democratizar el acceso al mercado, reducir las barreras de entrada para los emprendedores y contribuir a la formalización de actividades económicas que tradicionalmente operaban en la informalidad, estas herramientas propician el desarrollo de ecosistemas

colaborativos, donde los usuarios pueden compartir experiencias, calificar servicios y generar confianza dentro de la comunidad (Castillo & Sánchez, 2021).

Diversos estudios señalan que la implementación de plataformas digitales de servicios locales ha permitido incrementar la visibilidad de pequeños proveedores y optimizar los procesos de búsqueda y selección por parte de los usuarios. En el contexto ecuatoriano, el crecimiento de plataformas similares ha estado vinculado a la necesidad de reactivar la economía tras eventos disruptivos, como la pandemia de COVID-19, que aceleraron la adopción de soluciones tecnológicas en los negocios de barrio y en el sector de servicios (El Comercio, 2021).

Por otro lado, la automatización de procesos de registro, gestión de usuarios, publicación de servicios/productos y contacto directo a través de la web, contribuye a mejorar la eficiencia operativa y la transparencia en las transacciones (Torres, 2022). El uso de sistemas de calificación y opiniones refuerza la confianza entre oferentes y demandantes, mientras que la integración de canales de comunicación como WhatsApp o llamadas directas facilita la interacción y el cierre de acuerdos.

La experiencia internacional demuestra que estas plataformas no solo impactan en la competitividad de los negocios, sino que también inciden en el desarrollo social al fortalecer el sentido de comunidad y la economía circular, maximizando los recursos y capacidades locales (Gómez et al., 2020).

2.1.1. Fundamentación Legal

En Ecuador, la prestación de servicios y la venta de productos a través de plataformas digitales está regulada por varias normativas orientadas a garantizar la transparencia, la

protección del consumidor y el cumplimiento de obligaciones tributarias. La Ley de Comercio Electrónico, Firmas y Mensajes de Datos establece el marco legal para las transacciones electrónicas, asegurando la validez jurídica de los contratos y registros digitales (Asamblea Nacional, 2002).

La Ley Orgánica de Defensa del Consumidor protege los derechos de los usuarios en la adquisición de productos y servicios a través de plataformas digitales, estableciendo obligaciones en materia de información clara, calidad de servicio y mecanismos de reclamo (LODC, 2000). A su vez, el Código Orgánico de la Producción, Comercio e Inversiones (COPCI) y las resoluciones del Servicio de Rentas Internas (SRI) establecen lineamientos para la formalización de actividades comerciales y el cumplimiento tributario, incluyendo la facturación electrónica para operaciones realizadas en línea (SRI, 2023).

El Reglamento de Protección de Datos Personales, alineado con la Ley Orgánica de Protección de Datos Personales, regula el manejo y resguardo de la información de los usuarios en sistemas informáticos y plataformas web, obligando a los administradores a implementar medidas técnicas y organizativas adecuadas para salvaguardar la privacidad y los datos personales de los usuarios (LOPDP, 2021).

Implementar una plataforma digital bajo este marco legal permite ofrecer servicios con garantías de seguridad, transparencia y cumplimiento normativo, lo que resulta fundamental para el éxito y la sostenibilidad del proyecto en el contexto ecuatoriano. Además, contribuye a la confianza de los usuarios y promueve la inclusión digital y financiera de los actores locales.

2.1.2. Fundamentación Teórica

La gestión eficiente de servicios y productos a nivel local es un factor clave para impulsar el desarrollo económico y la competitividad de los emprendedores, técnicos y pequeños negocios. Un sistema automatizado que facilite la publicación, el registro, la búsqueda y la contratación de servicios o productos, permite conectar a oferentes y demandantes de manera ágil y segura, reduciendo la informalidad y promoviendo la formalización y profesionalización de los oficios (Martínez & Vaca, 2023).

La experiencia del usuario en estas plataformas se enriquece con funciones como la personalización de perfiles, la calificación de servicios, el contacto directo entre usuarios y la posibilidad de realizar búsquedas filtradas por zona, categoría o tipo de servicio. Al recopilar datos relevantes sobre los usuarios y sus preferencias, la plataforma puede ofrecer recomendaciones más acertadas y contribuir a la fidelización y satisfacción de los clientes (Díaz & Quishpe, 2022).

Estudios sobre la integración de soluciones tecnológicas en el ámbito empresarial muestran que la adopción de sistemas automatizados mejora la eficiencia operativa, la visibilidad del negocio y la capacidad de respuesta a las necesidades del mercado. En países de América Latina, la digitalización de servicios locales ha permitido a las comunidades acceder a una mayor variedad de opciones, optimizar el uso de recursos y fortalecer la economía circular (Castillo & Sánchez, 2021).

La gestión de usuarios, la automatización de publicaciones y el acceso a información en tiempo real no solo optimizan el trabajo de los proveedores, sino que también simplifican la experiencia para los clientes, quienes pueden tomar decisiones informadas y realizar

contrataciones de manera más rápida y segura. Esto es especialmente relevante en contextos donde la desconfianza o la falta de referencias han limitado históricamente el acceso a servicios profesionales (García, 2020).

2.2.Fundamentación Teórica de las Herramientas de Desarrollo

El desarrollo de plataformas web modernas requiere el uso de tecnologías robustas y ampliamente adoptadas por la comunidad de desarrollo. Entre las principales herramientas seleccionadas para este proyecto destacan PHP, Laravel, Vue.js, MySQL, HTML, CSS y JavaScript.

PHP es un lenguaje de programación de código abierto, ampliamente utilizado en el desarrollo web, que permite construir aplicaciones dinámicas e interactivas. Su integración con bases de datos y su facilidad para gestionar formularios, sesiones y usuarios lo convierten en la base de numerosos sistemas de gestión y plataformas web (PHP Documentation Group, 2024).

Laravel es uno de los frameworks PHP más populares. Destaca por su arquitectura moderna basada en el patrón MVC, su sistema de rutas, migraciones, controladores, y una sintaxis clara y expresiva. Laravel agiliza el desarrollo de aplicaciones seguras, escalables y mantenibles, integrando herramientas como Eloquent ORM para acceso a datos y Breeze/Jetstream para autenticación de usuarios (Laravel, 2024).

MySQL es un sistema de gestión de bases de datos relacional eficiente, seguro y de código abierto, utilizado para almacenar y organizar la información de usuarios, servicios, productos, categorías y transacciones en la plataforma (Londoño, 2023).

HTML (HyperText Markup Language) es el estándar para estructurar contenido en la web. Define la estructura básica de las páginas y permite integrar elementos multimedia, formularios, enlaces y tablas, facilitando la navegación y accesibilidad para el usuario (MDN Web Docs, 2024).

CSS (Cascading Style Sheets) es el lenguaje encargado de la presentación visual de los documentos HTML. Permite definir estilos, diseños responsivos y animaciones, contribuyendo a una experiencia visual atractiva y coherente en dispositivos de cualquier tamaño (MDN Web Docs, 2023b).

JavaScript es el lenguaje esencial para dotar de dinamismo e interactividad a las páginas web. Permite actualizar contenido en tiempo real, validar formularios, gestionar eventos del usuario y crear aplicaciones web avanzadas que funcionan sin recargar la página, integrándose perfectamente con HTML y CSS (MDN Web Docs, 2023).

El uso conjunto de estas tecnologías asegura el desarrollo de una plataforma robusta, segura y escalable, que responde a las necesidades tanto de los oferentes como de los usuarios de servicios y productos locales.

- **Plataformas de dos lados y mercados locales**

Las plataformas digitales que conectan oferta y demanda se estudian como mercados de dos lados. Su particularidad es el efecto de red cruzado: mientras más vendedores haya, más útil es para los compradores, y viceversa. Por ello el diseño del modelo de negocio (por ejemplo, reglas de acceso, comisiones, subsidios a un lado del mercado) incide directamente en la adopción y la salud del ecosistema. En entornos locales (barrios o parroquias) estos efectos se

manifiestan con rapidez: pequeños cambios en fricción de uso, visibilidad o costos pueden inclinar el equilibrio a favor o en contra de la participación.

Para el caso comunitario, el objetivo no es maximizar tarifas, sino maximizar transacciones confiables y reducir fricción. Por eso el proyecto prioriza: (i) bajos costos de entrada (onboarding guiado y plantillas), (ii) reglas claras (estados del pedido, tiempos objetivo, política de comisiones) y (iii) métricas de salud (conversión, pedidos por categoría, OTD). Estas decisiones responden a la lógica de plataformas: cuando las barreras de adopción bajan en un lado (vendedores), el otro lado (clientes) encuentra más variedad y mejores precios; y el círculo virtuoso se refuerza con reputación y repetición.

- **2.2. UX para confianza digital (perfiles, verificación y calificaciones)**

La confianza es un requisito para que el usuario arriesgue tiempo, dinero y datos. En plataformas locales, la credibilidad se construye con señales visibles y consistentes:

- **Perfiles públicos** de vendedor con nombre de tienda, zona, horarios, medios de contacto, políticas y evidencia de verificaciones básicas (KYC liviano) para elevar la percepción de legitimidad.
- **Información precisa** de producto/servicio (precio, stock/disponibilidad, fotos reales, descripción, condiciones) y divulgación honesta de costos y tiempos.
- **Calificaciones y reseñas** moderadas que permitan aprender del desempeño pasado y detectar patrones de calidad.

- **Retroalimentación en tiempo real:** cambios de estado, alertas, y comprobantes accesibles desde el detalle del pedido.
- **Diseño consistente y accesible** (móviles primero), con formularios claros, validación progresiva y mensajes de error accionables.

En el piloto, estas prácticas se materializan en: fichas con precio y stock visibles, estados del pedido con línea de tiempo, perfiles de tienda, y un tablero por rol que reduce la ambigüedad operativa.

- **2.3. Gestión de inventario/ofertas y trazabilidad en servicios**

A nivel operativo, la plataforma trata dos objetos: (a) artículos ofertados (bienes o servicios) y (b) órdenes (pedidos/solicitudes) con eventos registrados. La trazabilidad—capacidad de reconstruir la historia, uso o localización—se logra guardando hitos mínimos por orden: creación, confirmación, emisión de documento (recibo/factura), recepción y verificación del comprobante, preparación, despacho, entrega y calificación. Esta secuencia funciona tanto para productos físicos (inventario, stock, retiros) como para servicios (agenda, atención, cierre).

Buenas prácticas implementadas o previstas:

- **Catálogo normalizado:** categorías y subcategorías únicas, con códigos y visibilidad por barrio.
- **Control de disponibilidad:** para bienes, stock mínimo y alertas; para servicios, cupos o agenda por franja.

- **Identificadores de trazabilidad:** order_id, códigos/QR en el PDF de recibo/factura y vínculo al historial.
- **Bitácora inalterable:** registro de cambios de estado con usuario, fecha y motivo; adjuntos (comprobantes y evidencias de entrega) bajo política de retención.
- **Reportes:** edad de pendientes, entregas a tiempo (OTD), rechazos por observación de comprobante, ventas por categoría/barrio.

El principio es trazar menos, pero mejor: pocas variables obligatorias, fáciles de capturar desde móvil, y suficientes para auditar pedidos y mejorar el servicio.

- **2.4. Pasarelas de pago y flujo alternativo (WhatsApp + comprobante)**

Existen dos familias de pago: (1) pasarelas integradas (tarjetas, débito inmediato, wallets) y (2) flujos alternos usados en la práctica local, donde el cliente transfiere y envía el comprobante por un canal dominante (WhatsApp). El proyecto adopta inicialmente la vía alterna segura por su alta adopción y bajo costo:

1. El sistema genera recibo/factura en PDF con identificador/QR y datos de pago.
2. El cliente hace la transferencia y pulsa “Enviar por WhatsApp”: se abre el chat con mensaje prellenado (id de pedido, monto, referencia).
3. El cliente adjunta el comprobante (imagen/PDF).
4. En el panel del vendedor/administrador se registra Comprobante recibido Validación (aprobado/observado/rechazado). Al aprobar, se habilita preparación/entrega.

Para escalar, el flujo contempla integrar la API de WhatsApp Business (registro de medios y metadatos) o, en su defecto, carga manual del comprobante. La seguridad se refuerza con: HTTPS, almacenamiento cifrado de adjuntos, hash del archivo, y auditoría de quién/ cuándo validó cada pago.

CAPÍTULO III

3. Marco Metodológico

3.1. Enfoque de la investigación

La presente investigación es aplicada y adopta un enfoque mixto. Es aplicada porque desarrolla e introduce una plataforma web operativa que atiende una necesidad concreta de los sectores comunitarios: conectar oferta y demanda locales bajo reglas claras de trazabilidad y verificación de pago. El enfoque mixto combina dos perspectivas complementarias:

- **Componente cuantitativo.** Permite medir el desempeño del sistema a través de indicadores objetivos (por ejemplo, tasa de conversión a pedido, verificación de pagos, OTD—entregas a tiempo, edad de pedidos pendientes, SLA de respuesta por WhatsApp), así como analizar el comportamiento de uso mediante registros de base de datos y encuestas estandarizadas de satisfacción (CSAT) y recomendación (NPS).
- **Componente cualitativo.** Aporta comprensión profunda de la adopción y la experiencia a través de entrevistas semiestructuradas y observación de tareas (checklists de usabilidad por rol). Este insumo explica por qué se producen las fricciones, cómo perciben el valor los usuarios y qué ajustes priorizar para elevar la satisfacción y la confianza.

La plataforma fue construida con Laravel (Blade) y MySQL; el flujo de pago contempla la emisión de recibo/factura en PDF con identificador/QR y el envío del comprobante por WhatsApp para su validación (aprobado/observado/rechazado) antes de la preparación y entrega. Este esquema de “pago por evidencia” se alinea con hábitos locales de mensajería y reduce barreras de adopción sin sacrificar trazabilidad.

Soporte teórico breve. La investigación aplicada busca resolver problemas específicos y transferir resultados directamente utilizables (Hernández Sampieri et al., 2014). El enfoque mixto integra medición y comprensión contextual para explicar resultados y orientar decisiones (Creswell & Creswell, 2018). Para el desarrollo incremental se sigue XP (Beck & Andres, 2004), por su énfasis en entregas pequeñas, retroalimentación continua y mejora iterativa.

3.2. Tipo y diseño de investigación

- **Tipo de estudio:** no experimental, longitudinal de tipo pretest–posttest con un grupo piloto. Se observa el desempeño antes y después del despliegue, sin manipular variables externas.
- **Diseño de desarrollo:** incremental/iterativo (XP) en sprints de 2 semanas: planificación, diseño, codificación, pruebas, retroalimentación. Cada iteración libera capacidades utilizables (catálogo, pedidos con línea de tiempo, emisión de PDF, verificación de comprobantes, dashboards por rol) y recoge datos para retroalimentar la siguiente iteración.

3.3. Ámbito, población y muestra

- **Ámbito territorial:** parroquia Amaguaña y alrededores del Valle de los Chillos (Sangolquí, Uyumbicho, Tambillo) y Machachi.
- **Población objetivo:** vendedores/proveedores locales y compradores de los sectores mencionados.
- **Muestra piloto (criterio intencional):**
 - **Vendedores/proveedores:** 10–15 cuentas activas.
 - **Clientes:** 30–50 registros con al menos una interacción (pedido, mensaje o envío de comprobante).
 - **Repartidores:** 1–3 usuarios para validar asignación y tiempos de entrega.
- **Criterios de inclusión:** mayores de edad; residencia o actividad económica en los barrios definidos; aceptación del **consentimiento informado**.
- **Consideraciones éticas:** tratamiento de datos personales conforme a la LOPDP (consentimiento, finalidad, minimización y seguridad); anonimización de resultados de investigación; almacenamiento seguro de comprobantes y registros; atención a solicitudes ARCO (acceso, rectificación, etc.).

3.4. Técnicas e instrumentos de recolección de información

3.4.1. Encuestas (CSAT/NPS y hábitos de uso)

Para el desarrollo del presente proyecto, se emplearán técnicas mixtas de recolección de información que permitan obtener una visión integral sobre las necesidades, expectativas y

problemáticas que enfrentan actualmente los pequeños emprendedores, prestadores de servicios y consumidores en cuanto a la oferta y demanda de servicios locales. Las técnicas utilizadas serán:

- **Encuestas**

Se diseñarán encuestas estructuradas con preguntas cerradas y abiertas, orientadas a recolectar datos cuantitativos sobre los hábitos de búsqueda y contratación de servicios en comunidades locales, el nivel de digitalización de los trabajadores independientes y la percepción de utilidad de una plataforma web que permita visibilizar productos y servicios locales.

Estas encuestas serán distribuidas a una muestra representativa conformada por oferentes (vendedores de servicios y productos) y potenciales clientes, en sectores urbanos y periurbanos. La técnica de encuesta se seleccionó como herramienta clave para identificar, de forma medible, los factores que afectan la conexión directa entre quienes necesitan un servicio y quienes pueden proveerlo. El objetivo es obtener datos que permitan construir una plataforma funcional que responda a las expectativas reales de los usuarios y promueva una economía local más conectada y sostenible.

Testimonios destacados de la encuesta:

Luis Franklin Amaguaña, Ingeniero en Sistemas y colaborador en soporte digital comunitario en barrios del sur de Quito, trabaja con colectivos vecinales que promueven el autoempleo.

Sus aportes destacan:

- La necesidad de una plataforma sencilla que permita a los vecinos ofrecer sus servicios sin tener conocimientos técnicos avanzados.
- La importancia de incluir notificaciones automáticas vía WhatsApp para mantener la comunicación entre usuarios.
- Sugiere que el sistema tenga un diseño adaptado a móviles, ya que la mayoría de oferentes no utiliza computadoras.

Luis Horacio Arauz Demera, Gerente General de Arazoft, empresa especializada en soluciones web para negocios emergentes, comparte su visión estratégica sobre plataformas comunitarias.

Él destaca:

- La integración con redes sociales y WhatsApp como ventaja competitiva, pues muchas ventas comienzan en medios digitales informales.
- Sugiere que el sistema sea capaz de filtrar por ubicación y tipo de servicio para mejorar la experiencia del usuario.
- Resalta la importancia de que el sistema permita una gestión sencilla de publicaciones, sin requerir asesoría técnica.

Estefanía Rocío Chasi Toaza, Licenciada en Contabilidad y consultora en emprendimientos barriales, ha trabajado con ferias de economía popular y solidaria.

Desde su experiencia, menciona:

- La necesidad de un módulo de registro para clientes y vendedores, con opciones básicas de contacto (teléfono o WhatsApp).
- Recomienda que la plataforma permita a los usuarios ver historial de servicios contratados o favoritos para mejorar la fidelización.
- Sugiere que exista una opción para categorizar servicios (ej. salud, electricidad, construcción, etc.) y facilitar la búsqueda.

Rommel Andrés Suntaxi Nacimba, Ingeniero Ambiental y Analista de Datos en el INAMHI, colaborador en proyectos tecnológicos comunitarios para la toma de decisiones locales.

En su análisis, propone:

- Que el sistema incluya estadísticas básicas para los vendedores, como número de visitas por publicación o contactos generados.
- Señala que el uso de herramientas visuales puede ayudar a los oferentes a entender qué servicios tienen mayor demanda.
- Plantea la posibilidad futura de incluir un módulo analítico que permita identificar patrones de uso en ciertas zonas o temporadas.

• Entrevistas

Además de las encuestas, se realizarán entrevistas semi-estructuradas a un grupo selecto de participantes clave. Estas entrevistas permitirán obtener información cualitativa profunda sobre cómo las personas perciben la utilidad de la plataforma, cómo resuelven actualmente la contratación de servicios locales y qué elementos consideran indispensables en una herramienta digital comunitaria.

El enfoque de estas entrevistas será entender la experiencia del usuario en la vida real, más allá de lo que los datos estadísticos pueden reflejar. Este complemento cualitativo permitirá ajustar el diseño del sistema a las verdaderas necesidades de sus futuros usuarios.

Entrevista Oscar Caisatoa	
Entrevistado	Luis Horacio Arauz Demera, Gerente General de Arazoft
Fecha	30-MAY-2025
Lugar	Oficinas de Arazoft
Fase Inicial: La entrevista con el ingeniero Arauz abordó criterios clave en el diseño de sistemas web modernos, especialmente dirigidos a usuarios que buscan herramientas sencillas pero funcionales para promocionar y contratar servicios a nivel comunitario.	
Preguntas	Respuestas
¿Qué elementos debe priorizar una plataforma web para servicios locales?	Diseño modular, filtros por categoría o zona, y que funcione bien desde cualquier navegador.
¿Qué tan importante es que el buscador sea dinámico?	Es esencial. La búsqueda debe responder en tiempo real sin necesidad de recargar toda la página.
¿Qué funcionalidades pueden motivar a los usuarios a usarla?	Que sea fácil publicar, editar o eliminar servicios, y que se puedan contactar directamente desde la web.

¿Qué recomendaciones daría sobre la experiencia de usuario?	Menús simples, botones visibles, sin necesidad de instalar nada. También accesible desde teléfonos sin ser una app.
¿Qué medidas de seguridad serían fundamentales en la plataforma?	Manejo correcto de roles, protección de datos y una interfaz segura con autenticación validada.
Fase de Cierre: Luis destacó la necesidad de diseñar una interfaz web amigable y sin barreras técnicas. También subrayó la importancia de estructurar bien los roles (vendedor y cliente) y de incluir herramientas de búsqueda y publicación accesibles.	

Entrevista Oscar Caisatoa	
Entrevistado	Franklin Amaguaña, Ingeniero en Sistemas
Fecha	25-MAY-2025
Lugar	Oficinas de Tventas
Fase Inicial: La entrevista con Franklin Amaguaña se realizó con el objetivo de conocer los desafíos técnicos que enfrentan los trabajadores independientes y pequeños vendedores al ofrecer servicios o productos dentro de sus comunidades. Se presentó la idea de una plataforma web que sirva como un medio centralizado y accesible para promocionar sus servicios a nivel local.	
Preguntas	Respuestas
¿Qué medios usan actualmente los vecinos para ofrecer servicios?	Principalmente grupos de WhatsApp y redes sociales. No hay una plataforma organizada.
¿Qué dificultades enfrentan sin una plataforma digital estructurada?	Es difícil llegar a nuevos clientes y mantener la información organizada o actualizada.
¿Qué funcionalidades cree que serían útiles en una plataforma web comunitaria?	Publicar servicios con fotos, descripción, precios y contacto visible. También filtros por zona.
¿Qué tan importante es tener contacto directo desde la plataforma?	Muy importante. Que puedan llamar o escribir por WhatsApp desde la misma web sin instalar nada.

¿Cómo debería estar diseñada la interfaz para ser fácil de usar?	Debe ser clara, organizada y sencilla de usar desde cualquier navegador web, ya sea desde un celular o computadora.
Fase de Cierre: Franklin recalcó que la plataforma debe ser completamente funcional desde un navegador y no requerir instalación. Recomendó una interfaz web amigable, con opciones visibles para publicar o buscar servicios, y acceso rápido a los datos de contacto.	

Entrevista Oscar Caisatoa	
Entrevistado	Estefanía Rocío Chasi Toaza, Licenciada en Contabilidad, Mega Tienda
Fecha	1-JUN-2025
Lugar	Oficinas de Mega Tienda
Fase Inicial: Esta entrevista tuvo como objetivo conocer cómo una plataforma digital puede facilitar la gestión de ventas informales y qué elementos podrían apoyar a pequeños emprendedores a organizar su oferta de productos desde una interfaz web.	
Preguntas	Respuestas
¿Cómo impacta la digitalización en negocios pequeños?	Permite llevar un mejor control de lo que ofrecen y comunicarse fácilmente con los clientes.
¿Qué características cree que son importantes en la plataforma web?	Que permita crear publicaciones claras con fotos, precios y zona. Y que el historial sea fácil de revisar.
¿Qué funciones facilitarían el trabajo de los vendedores?	Poder editar sus publicaciones, ver cuántas veces han sido vistas, y tener su propio panel de control.
¿Cómo debería funcionar el acceso al contacto?	Que el cliente pueda escribir o llamar directamente desde un botón visible en la página.
¿Qué herramientas contables mínimas serían útiles?	Un panel que muestre estadísticas simples como número de vistas, clics o publicaciones activas.

Fase de Cierre:

Estefanía recalcó que la plataforma debe facilitar la organización del vendedor sin ser complicada. Un historial de publicaciones y contacto sería clave para quienes no tienen conocimientos técnicos, pero desean mantener sus servicios activos en línea.

Entrevistador		Oscar Caisatoa
Entrevistado	Rommel Andrés Suntaxi Nacimba, Ingeniero Ambiental y Analista de Datos, INAMHI	
Fecha	5-JUN-2025	
Lugar	Oficinas del INAMHI	
Fase Inicial: Con Rommel se discutió el valor de incorporar funciones de análisis básico en la plataforma web, para que los usuarios especialmente vendedores puedan conocer el comportamiento de sus publicaciones y tomar mejores decisiones.		
Preguntas		Respuestas
¿Qué métricas considera útiles para los vendedores en una plataforma comunitaria?		Número de visitas, publicaciones destacadas, clics en botón de contacto, y zonas con más búsqueda.
¿Cómo se pueden presentar estos datos de forma sencilla?		Con gráficos o tablas básicas dentro del panel del usuario, sin sobrecargar la interfaz.
¿Qué ventajas ofrece esta información?		Permite saber qué servicio es más solicitado, en qué zonas, y cuándo hay mayor actividad.
¿Qué retos hay con datos en tiempo real?		Que sean precisos, se almacenen de forma segura y no ralenticen el sitio web.
¿Qué medidas de seguridad deben aplicarse?		Roles definidos, protección de datos personales, y encriptación de toda la información sensible.
Fase de Cierre: Rommel sugirió implementar un módulo analítico básico para mejorar la toma de decisiones de los usuarios. También recomendó asegurar la privacidad y estabilidad del sistema sin sacrificar su agilidad.		

3.5. Tipo y diseño de investigación

- **Tipo:** estudio no experimental, longitudinal, con comparación antes y después del piloto.
- **Diseño de desarrollo:** trabajo iterativo (estilo XP). En ciclos cortos, diseñamos, construimos, probamos con usuarios y mejoramos. Para organizar tareas usamos un tablero simple con columnas “Pendiente”, “En curso” y “Hecho”.
- **Tabla 1. Enfoque y diseño**

Elemento	Descripción breve
Enfoque	Mixto: medimos con números y escuchamos a las personas.
Propósito	Resolver un problema real de conexión local con pagos verificados.
Tipo de estudio	No experimental, con comparación antes/después del piloto.
Diseño de desarrollo	Iterativo: construir → probar → ajustar (ciclos cortos).
Entregables clave	Catálogo y búsqueda; pedidos con estados; recibo PDF; verificación de comprobante por WhatsApp; tableros por rol.
Tecnología	Laravel (Blade) + MySQL.

3.6. Ámbito, población y muestra

- **Ámbito:** Amaguaña y alrededores del Valle de los Chillos (Sangolquí, Uyumbicho, Tambillo) y Machachi.
- **Población objetivo:** vendedores/proveedores locales y compradores de esas zonas.

Muestra piloto:

- **Vendedores/proveedores:** 10 a 15 cuentas activas.
- **Clientes:** 30 a 50 registros con alguna interacción real (pedido, mensaje o envío de comprobante).

- **Repartidores:** 1 a 3 usuarios para probar la entrega.
- **Inclusión y ética:** mayores de edad; actividad o residencia en los sectores; consentimiento informado. Se respetan principios de protección de datos (finalidad, minimización y seguridad). Los comprobantes se guardan de forma segura y sólo por el tiempo necesario.

3.7. Técnicas e instrumentos de recolección

- **Encuestas cortas:** una antes de usar y otra al terminar el piloto. Miden **satisfacción, facilidad** y si recomendarían la plataforma.
- **Entrevistas semiestructuradas:** 6 a 10 conversaciones (vendedores, clientes y 1–2 repartidores) para captar barreras, expectativas y mejoras.
- **Registros del sistema:** cada pedido deja una huella sencilla: se crea, se emite el PDF, se recibe el comprobante, se valida, se prepara, se entrega y se califica. Con eso medimos tiempos y resultados.
- **Checklist de usabilidad por rol:** observamos tareas clave (publicar, buscar/filtrar, pedir, enviar comprobante, validar pago, confirmar entrega) y anotamos si lo logran, cuánto tardan y qué errores aparecen.

3.8. Variables e indicadores

Variable independiente: uso de la plataforma (módulos activos y flujo de pago con comprobante).

- **Variable dependiente: reactivación económica local** (más contactos reales, más pedidos bien cerrados y primeros signos de recompra).

- **Tabla 2. Qué vamos a medir y cómo lo interpretaremos**

Indicador	¿Qué significa?	¿Cómo lo sacaré (en palabras)?	Meta del piloto
Conversión a pedido	De cada 100 personas que ven un producto/servicio, ¿cuántas llegan a pedirlo?	Contar vistas de fichas y pedidos que sí se crean.	Que al menos 5–8 de cada 100 se animen.
Verificación de pago	De los pagos con comprobante, ¿cuántos se aprueban?	Revisar cuántos comprobantes llegan y cuántos se validan.	Que 9 de cada 10 queden aprobados.
Respuesta por WhatsApp	¿Cuánto tarda el vendedor/admin en contestar?	Mirar el tiempo entre el mensaje del cliente y la primera respuesta.	30 minutos o menos en horario laboral.
Entregas a tiempo	¿Cuántos pedidos llegan cuando se prometió?	Comparar la hora prometida con la hora de entrega.	85 % o más a tiempo.
Edad de pendientes	¿Hace cuánto están “estancados” los pedidos sin avanzar?	Calcular la antigüedad de pedidos en estado “pendiente”.	Que en promedio sea menos de 24 h.
Clientes nuevos (30 días)	¿Cuántas personas nuevas se registran?	Contar los registros recientes.	—
Retención (30 días)	¿Cuántos vuelven a comprar pronto?	Ver cuántos nuevos repiten compra en 30 días.	Al menos 20 % repite.

3.9. Plan de análisis de datos

- **Cuantitativo:** promedios, porcentajes y comparaciones antes vs. después del piloto.

Nada de estadística complicada: si sube la conversión y bajan los pendientes, podemos decir que mejora.

- **Cualitativo:** agrupamos comentarios de entrevistas y preguntas abiertas en temas (confianza, facilidad, tiempos, problemas frecuentes). Luego cruzamos esas ideas con los números para priorizar mejoras.
- **Presentación:** tablas en formato APA y gráficos sencillos (por ejemplo, barras para “antes/después” y una línea para las semanas del piloto).

CAPITULO IV

4. Desarrollo y Resultados

4.1.Descripción general del sistema

La solución Conecta Comunidad es una aplicación web orientada a visibilizar la oferta de productos y oficios locales y a ordenar el proceso de compra con trazabilidad. El sistema permite que el cliente consulte un catálogo único por barrio/categoría, genere un pedido y envíe su comprobante de pago por WhatsApp, mientras que el proveedor valida el comprobante, prepara el pedido y, si corresponde, se coordina la entrega con un repartidor. Toda la operación queda auditada mediante registros de estados (línea de tiempo del pedido), recibos/facturas en PDF y almacenamiento seguro del comprobante.

La arquitectura adopta el patrón Modelo–Vista–Controlador (MVC), lo que separa responsabilidades y simplifica el mantenimiento. En términos prácticos, los Modelos (Eloquent + MySQL) representan entidades y relaciones; las Vistas (Blade) brindan una interfaz accesible y adaptable a móviles; y los Controladores (HTTP/REST) orquestan la lógica, validaciones,

autorizaciones y cambios de estado. Esta separación permitió iterar por módulos —catálogo, pedidos, validación de pagos, logística y paneles por rol— sin romper el resto del sistema.

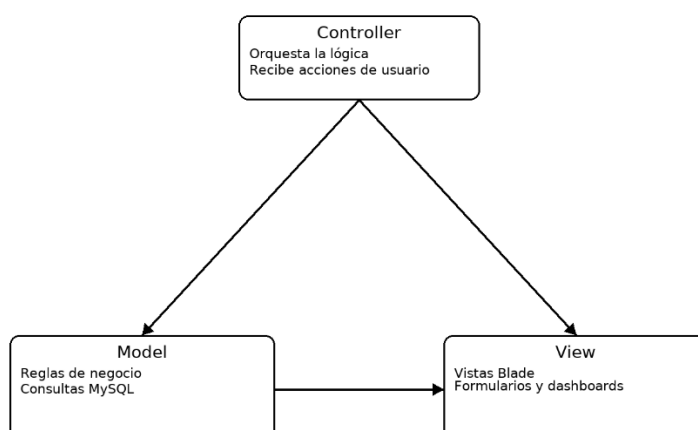


Ilustración 1. Patrón MVC de Conecta Comunidad.

4.1.1 Capa Modelo (datos y trazabilidad)

El núcleo de datos se implementó en MySQL, modelado para cubrir: usuarios y roles; proveedores/tiendas; categorías y productos; pedidos e ítems; recibos/facturas; comprobantes de pago; logs de estado; mensajería WhatsApp; y, de ser necesario, asignación de repartidores. Las tablas incorporan claves foráneas, índices por búsqueda (categoría, barrio/sector, estado) y soft deletes en entidades administrativas.

La trazabilidad se logra a través de `order_status_logs`, que almacena cada transición (creado, comprobante recibido, aprobado/observado/rechazado, en preparación, en ruta, entregado) con marca de tiempo y actor. Los recibos/facturas se generan en PDF y se versionan; los comprobantes se guardan con hash y metadatos para evitar duplicidades; y la relación con

whatsapp_messages permite medir SLA de respuesta y documentar el intercambio con el cliente.

El modelo de datos simplificado puede observarse en la Ilustración

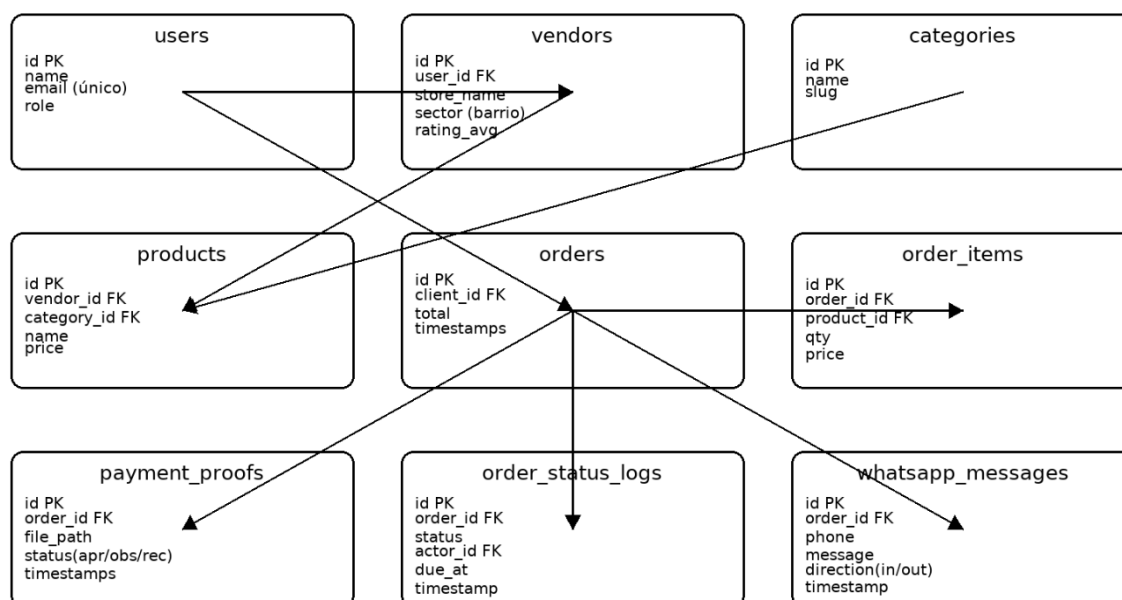


Ilustración 2. Diagrama entidad-relación (simplificado) de Conecta Comunidad.

4.1.2 Capa Vista (UI con Blade)

La interfaz se construyó con Laravel Blade y componentes reutilizables. La navegación responde a tres vistas maestras: cliente (búsqueda por barrio y categoría, tarjetas de producto/servicio, carrito/solicitud, botones de contacto/WhatsApp), proveedor (panel con KPIs, publicaciones recientes, validación de comprobantes y ventas), y administrador (gestión de usuarios/roles, catálogo, auditoría y reportes). Se cuidó la accesibilidad móvil con diseño mobile-first, formularios con validación en servidor y cliente, mensajes de error comprensibles y

ayudas contextuales. Los flujos críticos —generar recibo, subir comprobante, aprobar/observar— usan textos guiados para reducir errores y tiempos de aprendizaje.

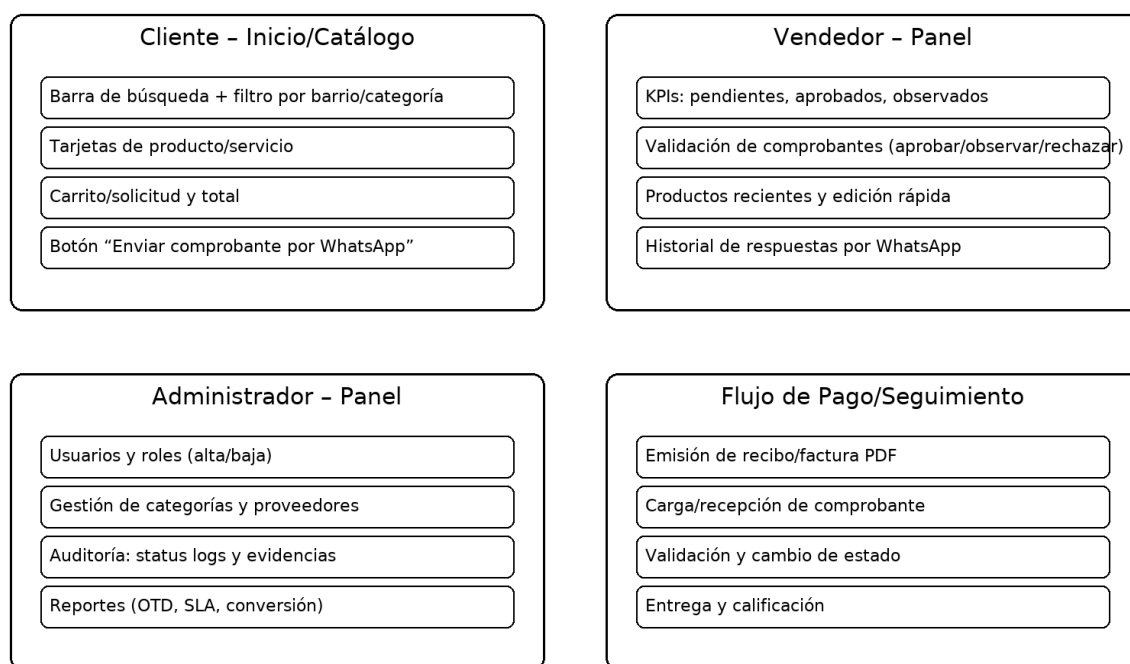


Ilustración 3. Wireframes funcionales por rol y flujo de pago.

4.1.3 Capa Controlador (lógica de aplicación)

Los controladores HTTP gestionan reglas de negocio y seguridad:

- Pedidos y línea de tiempo: crean la orden, agregan ítems, emiten el recibo/factura PDF, registran la llegada del comprobante y disparan la validación.
- Validación de pago: la decisión puede ser aprobado, observado (con motivo) o rechazado.

Cada acción registra evento, actor y hora; si hay observación, se notifica al cliente por WhatsApp.

- Logística y entrega (opcional en el piloto): asignación de repartidor y control de OTD (entrega en tiempo).
- Seguridad y privacidad: roles/ACL con políticas; cifrado de archivos sensibles; limpieza y retención de logs según la LOPDP; y minimización de datos en WhatsApp (solo lo imprescindible para validar el pago).

4.2. Justificación de Laravel y Blade

Laravel resulta idóneo por su ORM Eloquent (productividad y consistencia en relaciones), migraciones (versionado de esquema), sistema de plantillas Blade (UI clara sin dependencias complejas), validaciones robustas, políticas de autorización y ecosistema maduro para PDF, colas y almacenamiento. Elegir Blade —y no un framework SPA— priorizó simplicidad operativa y tiempos de entrega sin sacrificar experiencia en móvil. Esta combinación equilibró el alcance del proyecto piloto con mantenibilidad y seguridad.

4.3. Impacto en el proyecto

Organizar el desarrollo con MVC y migraciones versionadas facilitó liberar entregables incrementales y medir su efecto: más visibilidad de la oferta, menor fricción en el pago (porque el comprobante viaja por el canal que la comunidad ya usa: WhatsApp) y mayor control de lo que ocurre con cada pedido. Las evidencias de auditoría (PDF, hash, logs) incrementaron la confianza entre vecinos y redujeron reprocesos por información incompleta o tardía.

4.4. Modelo de datos (E/R)

El E/R refleja las relaciones esenciales: usuarios roles; proveedores, productos, categorías, órdenes, ítems; pagos y seguimiento (recibos, comprobantes, logs, WhatsApp). El diseño admite múltiples proveedores por barrio, múltiples productos por proveedor y múltiples ítems por pedido. El módulo de pagos y tracking agrupa entidades que garantizan la trazabilidad punta a punta y se integra con el flujo de mensajería para evidenciar respuesta y validación.

4.5.Diseño de la interfaz y UX

Las pantallas parten de tareas concretas por rol: el cliente encuentra rápido; el vendedor publica, valida y ve su desempeño; el administrador gobierna usuarios, catálogo y métricas. Las decisiones de diseño siguieron heurísticas de visibilidad del estado del sistema, control y libertad del usuario y prevención de errores. Ello incluye etiquetas claras, botones de acción visibles, confirmaciones no intrusivas y mensajes prellenados para el botón de WhatsApp con variables (número de pedido, monto y enlace al recibo).

4.6.Módulos funcionales (CRUD y flujo)

A nivel funcional, el sistema expone operaciones CRUD para usuarios/roles, proveedores/tiendas, categorías, productos, pedidos e ítems. Además, implementa flujos especializados: emisión de recibo/factura, carga del comprobante, validación con decisiones y seguimiento logístico. En el piloto, el rol repartidor es opcional; si no se activa, el pedido se cierra con “Retiro en punto” y registro de entrega.

Módulo	Create	Read	Update	Delete	Acciones extra
Usuarios	✓	✓	✓	✓	Activar/Desactivar, cambio de rol
Categorías / Subcategorías	✓	✓	✓	✓	Destacar (is_featured), reordenar

Productos	✓	✓	✓	✓	Cambiar estado, galería de imágenes, precios por talla/color
Galerías	✓	✓	✓	✓	Carga múltiple, miniaturas (thumbnails)
Sliders/Banners	✓	✓	✓	✓	Orden de aparición
Pedidos (Orders)	✓	✓	✓	✓	Estados del pedido, tracking

Resumen de módulos

Módulo	Alcance principal	Estado
Catálogo y búsqueda	Tarjetas por barrio/categoría, filtros y detalle	Implementado
Publicación de proveedor	Alta/edición de productos/servicios, imágenes y stock	Implementado
Pedidos e ítems	Carrito/solicitud, cálculo de totales, recibo PDF	Implementado
Comprobantes de pago	Carga/WhatsApp, <i>hash</i> , decisiones y motivos	Implementado
Logs de estado	Línea de tiempo con actor y hora	Implementado
Paneles por rol	KPIs de uso, actividades recientes, auditoría	Implementado
Logística (repartidor)	Asignación y OTD	Opcional en piloto

4.6.1. Historias de Usuario del Sistema Web Comunitario

Nro.	Historia Usuario	Usuario Principal	Prioridad	Descripción
1	Autenticación y Roles: Como usuario, necesito registrarme e iniciar sesión para acceder según mi rol (cliente o vendedor).	Todos	Alta	Permitir a los usuarios registrarse e iniciar sesión, verificando su email y asignando el rol correspondiente para mostrarles una interfaz adecuada a su función.
2	Publicar Servicios o Productos: Como vendedor, quiero publicar mis servicios o productos para ofrecerlos a personas cercanas.	Vendedor	Alta	Permitir a los vendedores cargar títulos, descripciones, imágenes, zonas, categorías y precios para promocionar sus productos o servicios de manera visual.
3	Buscar Servicios o Productos: Como	Cliente	Alta	Implementar un buscador con filtros por ubicación, oficio y

	cliente, quiero buscar servicios según categoría o ubicación para encontrar lo que necesito cerca.			categoría, con resultados dinámicos que no recarguen la página.
4	Contacto Directo: Como cliente, quiero contactar fácilmente al proveedor por WhatsApp o llamada para concretar la contratación.	Cliente	Alta	Incluir botones de contacto directo por WhatsApp o llamada desde la web, asociados a cada publicación.
5	Panel de Usuario: Como vendedor, quiero ver y administrar mis publicaciones, estadísticas y contactos recibidos.	Vendedor	Media	Desarrollar un panel donde los vendedores puedan revisar sus publicaciones, editarlas o eliminarlas y visualizar métricas simples como visitas o contactos.
6	Favoritos o Historial: Como cliente, quiero guardar mis servicios favoritos o ver a quiénes contacté anteriormente.	Cliente	Media	Permitir que los clientes marquen publicaciones como favoritas o revisen su historial de contactos para futuras contrataciones.
7	Gestión Administrativa: Como administrador, necesito revisar y moderar publicaciones para asegurar el buen uso de la plataforma.	Administrador	Media	Incorporar un módulo de revisión/moderación para que los administradores puedan aprobar, ocultar o eliminar contenido inapropiado.
8	Actualización de Perfil: Como usuario, quiero actualizar mis datos personales para mantener mi información al día.	Todos	Media	Permitir que los usuarios editen su nombre, contacto, dirección, contraseña, etc., desde su panel personal.

Gestión de Usuarios y Roles				
NRO.	USUARIO	PRIORIDAD	RIESGO	DESARROLLO
001	Administrador	Alta	Alta	Oscar Caisatoa
HISTORIA DE USUARIO				
Como administrador, necesito gestionar los usuarios de la plataforma asignando roles de "Cliente" o "Vendedor" para garantizar un acceso personalizado y seguro.				

Gestión de Publicaciones				
NRO.	USUARIO	PRIORIDAD	RIESGO	DESARROLLO
002	Vendedor	Alta	Media	Oscar Caisatoa
HISTORIA DE USUARIO				
Como vendedor, quiero crear, editar o eliminar publicaciones con mis servicios o productos para que otros usuarios puedan contactarme fácilmente.				
CRITERIOS DE ACEPTACIÓN				
Formulario de Publicación: Ingresar título, descripción, categoría, zona, precio e imágenes. Gestión de Publicaciones: Editar o eliminar publicaciones desde el panel personal. Visualización: Las publicaciones deben mostrarse en el sitio web con los datos completos, imagen destacada y botones de contacto.				
CRITERIOS DE ACEPTACIÓN				
Creación de Usuario: Registrar usuarios con nombre, correo electrónico, rol y contraseña temporal. Gestión de Roles: Permitir editar el rol del usuario entre "Cliente" y "Vendedor". Auditoría: Registrar todas las acciones realizadas en la gestión de usuarios (alta, baja y modificaciones).				

Buscador Inteligente y Filtros				
NRO.	USUARIO	PRIORIDAD	RIESGO	DESARROLLO
003	Cliente	Alta	Alta	Oscar Caisatoa
HISTORIA DE USUARIO				
Como cliente, quiero buscar productos o servicios por tipo, ubicación o categoría para encontrar lo que necesito de forma rápida.				

CRITERIOS DE ACEPTACIÓN
Buscador con filtros: Por palabra clave, categoría y zona. Carga dinámica: Mostrar resultados sin recargar la página. Mensajes de estado: Informar si no hay resultados para los filtros aplicados.

Panel de Usuario y Estadísticas				
NRO.	USUARIO	PRIORIDAD	RIESGO	DESARROLLO
004	Vendedor	Media	Media	Oscar Caisatoa
HISTORIA DE USUARIO				
Como vendedor, quiero acceder a un panel donde pueda ver mis publicaciones activas, editar información y revisar estadísticas básicas.				
CRITERIOS DE ACEPTACIÓN				
Listado Personalizado: Mostrar solo publicaciones del vendedor autenticado. Estadísticas: Número de vistas y contactos recibidos. Notificaciones: Alertas sobre publicaciones expiradas o productos sin fotos.				

Contacto Directo				
NRO.	USUARIO	PRIORIDAD	RIESGO	DESARROLLO
004	Cliente	Alta	Alta	Oscar Caisatoa
HISTORIA DE USUARIO				
Como cliente, quiero contactar al vendedor por WhatsApp o llamada directa desde el navegador para concretar el servicio o compra.				
CRITERIOS DE ACEPTACIÓN				
Botón de WhatsApp: Abre enlace con número del vendedor y mensaje predeterminado. Botón de llamada: Enlace tel: para abrir app telefónica en móviles o copiar número en escritorio. Privacidad: No mostrar el número completo hasta que el cliente haga clic para contactar.				

Historial de Contactos y Favoritos				
NRO.	USUARIO	PRIORIDAD	RIESGO	DESARROLLO 60
004	Cliente	Media	Media	Oscar Caisatoa
HISTORIA DE USUARIO				
Como cliente, quiero guardar publicaciones favoritas y ver con quiénes he interactuado anteriormente para repetir contrataciones fácilmente.				
CRITERIOS DE ACEPTACIÓN				
Marcado de Favoritos: Botón en cada publicación para guardar. Historial: Lista de publicaciones con las que el usuario hizo clic en contacto. Panel Cliente: Sección separada para visualizar favoritos y contactos.				

4.7. Planificación de las Iteraciones.

Nro.	Historia Usuario	Prioridad	Inicio	Finaliza	Descripción
1	Gestión de Usuarios Internos	Alta	25-Jun-2025	30-Jun-2025	Implementación de la funcionalidad para registrar y gestionar usuarios internos, incluyendo asignación de roles (Cliente/Vendedor) y control de accesos.
2	Gestión de Clientes	Alta	01-Jul-2025	07-Jul-2025	Registro y edición de información personal de los clientes, como dirección, contacto e historial de interacciones.
3	Gestión de Vendedores	Alta	08-Jul-2025	14-Jul-2025	Creación y gestión del perfil de vendedores, configuración de datos de contacto, zonas de atención y servicios ofrecidos.
4	Gestión de Publicaciones	Alta	15-Jul-2025	24-Jul-2025	Módulo para permitir a los vendedores publicar servicios/productos con título, descripción, precio, zona, categoría y fotos.
5	Buscador Inteligente	Alta	25-Jul-2025	31-Jul-2025	Implementación de búsqueda avanzada con filtros por ubicación, categoría y tipo de publicación, utilizando Vue para carga dinámica.

6	Contacto Directo	Alta	01-Agos-2025	07-Agos-2025	Integración de botones de contacto por WhatsApp y llamada desde cada publicación, accesibles desde la vista del cliente.
7	Panel de Usuario	Media	08-Agos-2025	15-Agos-2025	Desarrollo del panel para cada tipo de usuario: clientes (historial, favoritos) y vendedores (estadísticas, publicaciones activas).
8	Seguridad y Verificación	Alta	16-Agos-2025	20-Agos-2025	Activación de verificación de correo electrónico, políticas de contraseña segura y auditoría básica de accesos.
9	Pruebas e Integración Parcial	Alta	21-Agos-2025	25-Agos-2025	Pruebas de integración entre módulos y validación de flujos básicos de la plataforma.
10	Documentación Técnica Parcial	Media	26-Agos-2025	30-Agos-2025	Redacción de documentación base de funcionalidades implementadas y estructura de base de datos.

4.7.1. Planificación de las Iteraciones

#	Módulo	Duración	Objetivos	Actividades
1	Análisis y Diseño	25-Jun-2025 → 30-Jun-2025	- Definir requisitos y alcance del sistema. - Diseñar base de datos y arquitectura general.	- Reuniones y entrevistas con oferentes y usuarios. - Documentación de requerimientos funcionales y técnicos. - Diagramas de base de datos y estructura de carpetas en Laravel y Vue.
2	Gestión de Usuarios Internos	01-Jul-2025 → 05-Jul-2025	- Implementar login y registro de usuarios. - Definir roles: Cliente y Vendedor.	- Configuración de Laravel Breeze. - Backend para roles y permisos. - Pruebas de acceso y seguridad.
3	Gestión de Clientes	06-Jul-2025 → 10-Jul-2025	- Registro y actualización de clientes. - Historial de contacto y favoritos.	- Creación del modelo y controlador de clientes. - Diseño de vistas en Vue.js. - Pruebas funcionales.
4	Gestión de Vendedores	11-Jul-2025 → 15-Jul-2025	- Gestión de datos de vendedores y validación de perfiles.	- Desarrollo del módulo para editar datos de contacto, zonas de atención y servicios. - Integración con autenticación.

5	Gestión de Publicaciones	16- Jul - 2025 → 22- Jul - 2025	- Crear, editar y eliminar publicaciones de servicios/productos.	- Backend con Laravel para publicaciones. - Formulario en Vue con carga de imágenes, zona, categoría y precio. - Pruebas básicas.
6	Buscador Inteligente	23- Jul - 2025 → 28- Jul - 2025	- Implementar búsqueda con filtros dinámicos.	- Búsqueda SQL básica o con Laravel Scout. - Componente Vue para resultados dinámicos por zona, tipo o categoría.
7	Contacto Directo	29- Jul - 2025 → 01-Ago- 2025	- Facilitar contacto desde cada publicación.	- Botón de WhatsApp con número del vendedor. - Botón de llamada para móviles. - Validación de campos.
8	Panel de Usuario	02-Ago- 2025 → 08-Ago- 2025	- Crear vistas diferenciadas por tipo de usuario.	- Panel de Vendedor: publicaciones activas, estadísticas. - Panel de Cliente: historial y favoritos. - UI/UX en Vue.
9	Seguridad y Verificación	09-Ago- 2025 → 13-Ago- 2025	- Verificación de email y control de accesos.	- Activación de verificación de email en Laravel. - Middleware de protección por rol. - Pruebas de sesiones y logout.
10	Pruebas Finales e Integración	14-jun- 2025 → 18-Agos- 2025	- Validar integración entre módulos y comportamiento general del sistema.	- Pruebas de usuario (UAT). - Revisión cruzada de funcionalidades. - Ajustes menores y depuración.
11	Documentación Técnica	19-Agos- 2025 → 20-Agos- 2025	- Redactar documentación técnica y manuales de usuario.	- Documentar estructura de base de datos, rutas, controladores. - Crear manuales con capturas. - Exportar documentación en PDF.

4.7.2. Ejecución de las Iteraciones

Durante la ejecución de cada iteración, se llevó a cabo el desarrollo progresivo de las funcionalidades del sistema, siguiendo las historias de usuario previamente definidas y los

criterios de aceptación establecidos para cada módulo. La implementación se realizó de forma iterativa y controlada, permitiendo evaluar de manera continua el avance del proyecto.

Cada funcionalidad fue desarrollada utilizando Laravel para la lógica del backend y Vue.js para el manejo dinámico de las interfaces de usuario, integrando Bootstrap como framework de diseño visual. Además, se aplicaron pruebas funcionales para asegurar que cada módulo cumpliera con los requerimientos planteados y fuera usable tanto por clientes como por oferentes.

Las pruebas incluyeron:

Verificación de funcionalidades clave, como la autenticación de usuarios, la publicación de servicios/productos, los filtros de búsqueda inteligente, y los botones de contacto directo (WhatsApp o llamada).

Validación de roles y permisos, asegurando que cada usuario acceda solo a la funcionalidad correspondiente (cliente o vendedor).

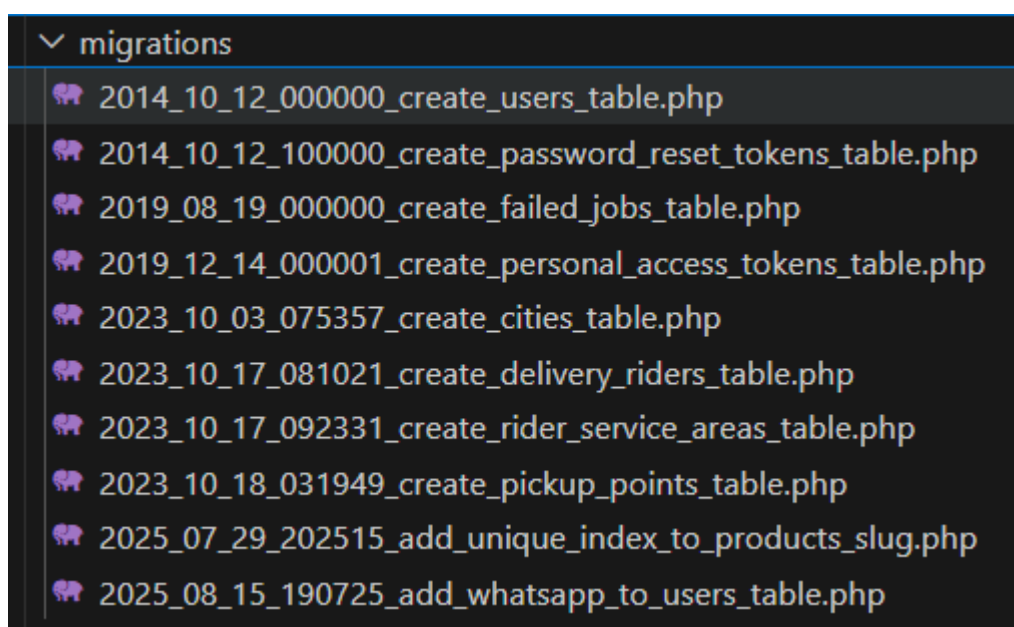
Pruebas de experiencia de usuario (UX) para revisar la usabilidad de la plataforma, la claridad de las vistas, y la simplicidad de los formularios.

Depuración de errores surgidos durante la navegación, el envío de formularios o la carga de archivos, garantizando la estabilidad del sistema antes de pasar a la siguiente iteración.

Gracias al enfoque ágil basado en Extreme Programming (XP), fue posible realizar ajustes rápidos en base a la retroalimentación recibida y garantizar una construcción modular, escalable y centrada en las necesidades reales de la comunidad usuaria de la plataforma.

4.8.Migraciones y controladores (criterios de implementación)

Las migraciones (carpeta project/database/migrations) definen la estructura de la base de datos y las relaciones entre entidades del e-commerce. A continuación se describen y documentan con capturas de código.



El proyecto también incorpora migraciones base de Laravel/Sanctum (usuarios, restablecimiento de contraseñas, tokens personales, etc.), además de otras adicionales; esto se refleja en el árbol de carpetas del proyecto.

A continuación, desde la **Ilustración 5** hasta la **Ilustración 12**, se presentan fragmentos de las migraciones principales que son fundamentales para el correcto almacenamiento de la información.

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('users');
    }
};
```

Ilustración 5. Código de migración para la tabla users.

Campos clave (ejemplo de tu código): id, name, lastname, email (único), email_verified_at, password, role (enum), deleted_at (soft deletes), rememberToken, timestamps.

Ilustración 6. Migraciones técnicas del framework.

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('password_reset_tokens', function (Blueprint $table) {
            $table->string('email')->primary();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('password_reset_tokens');
    }
};
```

- create_password_reset_tokens_table.php (tokens de restablecimiento de contraseña),

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('failed_jobs', function (Blueprint $table) {
            $table->id();
            $table->string('uuid')->unique();
            $table->text('connection');
            $table->text('queue');
            $table->longText('payload');
            $table->longText('exception');
            $table->timestamp('failed_at')->useCurrent();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('failed_jobs');
    }
};
```

- Create_failed_jobs_table.php (registro de trabajos fallidos en colas),

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('personal_access_tokens', function (Blueprint $table) {
            $table->id();
            $table->morphs('tokenable');
            $table->string('name');
            $table->string('token', 64)->unique();
            $table->text('abilities')->nullable();
            $table->timestamp('last_used_at')->nullable();
            $table->timestamp('expires_at')->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('personal_access_tokens');
    }
};

```

- create_personal_access_tokens_table.php (tokens personales para autenticación/API).

Propósito: soporte de seguridad, jobs y autenticación.

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('cities', function (Blueprint $table) {
            $table->id();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('cities');
    }
};
```

Ilustración 7. Código de migración para la tabla cities.

Propósito: catálogo de ciudades para segmentar operación/logística.

Campos típicos: id, name, (opcional: state_id/country_id), timestamps.

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('delivery_riders', function (Blueprint $table) {
            $table->id();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('delivery_riders');
    }
};
```

Ilustración 8. Código de migración para la tabla delivery_riders.

Propósito: registrar repartidores asociados al sistema.

Campos típicos: id, user_id (FK), datos de contacto (phone), vehicle_type, status, timestamps.

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('rider_service_areas', function (Blueprint $table) {
            $table->id();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('rider_service_areas');
    }
};
```

Ilustración 9. Código de migración para la tabla rider_service_areas.

Propósito: definir zonas/ciudades atendidas por cada repartidor.

Campos típicos: id, rider_id (FK → delivery_riders), city_id/zone_id (FK), timestamps.

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('pickup_points', function (Blueprint $table) {
            $table->id();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('pickup_points');
    }
};

```

Ilustración 10. Código de migración para la tabla pickup_points.

Propósito: puntos de recogida para pedidos.

Campos típicos: id, name, address, city_id (FK), (opcional: lat/long, contacto), timestamps.

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up()
    {
        Schema::table('products', function (Blueprint $table) {
            // Si tu columna slug ya existe como string:
            $table->unique('slug');
        });
    }

    public function down()
    {
        Schema::table('products', function (Blueprint $table) {
            $table->dropUnique(['slug']);
        });
    }
};
```

Ilustración 11. Migración add_unique_index_to_products_slug.

Propósito: agregar **índice único** al campo slug de la tabla products para evitar duplicados en URLs/permalinks.

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
use Illuminate\Support\Facades\DB;

return new class extends Migration {
    public function up(): void
    {
        // Crea la columna solo si no existe aún
        if (!Schema::hasColumn('users', 'whatsapp')) {
            Schema::table('users', function (Blueprint $table) {
                // Si tu tabla users tiene shop_number, puedes dejar el ->after('shop_number')
                // Si no estás 100% seguro, quita el ->after('shop_number')
                $table->string('whatsapp', 30)
                    ->nullable()
                    ->after('shop_number');
            });
        }
    }

    public function down(): void
    {
        // La borra solo si existe
        if (Schema::hasColumn('users', 'whatsapp')) {
            Schema::table('users', function (Blueprint $table) {
                $table->dropColumn('whatsapp');
            });
        }
    }
};

```

Ilustración 12. Migración add_whatsapp_to_users_table.

Propósito: añadir el campo whatsapp (y sus restricciones) a users para contacto directo.

Ilustración 13. Base de datos para la tabla **users**

Resumen de campos clave:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
☐	1 identificación	entero		NO FIRMADO	No	Ninguna		INCREMENTO AUTOMÁTICO	Cambiar Eliminar Más
☐	2 nombre	varchar(191)	utf8mb4_unicode_ci		No	Ninguna			Cambiar Eliminar Más
☐	3 foto	varchar(191)	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	4 cremallera	varchar(191)	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	5 ciudad	varchar(191)	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	6 pais	varchar(100)	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	7 id_de_estado	entero			Si	NULO			Cambiar Eliminar Más
☐	8 id de ciudad	entero			Si	NULO			Cambiar Eliminar Más
☐	9 DIRECCIÓN	varchar(191)	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	10 teléfono	varchar(191)	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	11 fax	varchar(191)	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	12 correo electrónico	varchar(191)	utf8mb4_unicode_ci		No	Ninguna			Cambiar Eliminar Más
☐	13 contraseña	varchar(191)	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	14 recordar_token	varchar(100)	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	15 creado_en	marca de tiempo			Si	NULO			Cambiar Eliminar Más
☐	16 actualizado_a las	marca de tiempo			Si	NULO			Cambiar Eliminar Más
☐	17 es_proveedor	diminuto			No	0			Cambiar Eliminar Más
☐	18 estado	diminuto			No	0			Cambiar Eliminar Más
☐	19 enlace de verificación	texto	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	20 correo electrónico verificado	enumeración('Si', 'No')	utf8mb4_unicode_ci		No	No			Cambiar Eliminar Más
☐	21 código de afiliado	texto	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	22 ingresos de afiliados	doble			No	0			Cambiar Eliminar Más
☐	23 nombre_de_la_tienda	texto	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	24 nombre_del_propietario	texto	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más

Ilustración 14. Base de datos para la tabla **categorías**

Resumen de campos clave:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
☐	1 identificación	entero			No	Ninguna		INCREMENTO AUTOMÁTICO	Cambiar Eliminar Más
☐	2 nombre	varchar(191)	utf8mb4_unicode_ci		No	Ninguna			Cambiar Eliminar Más
☐	3 babosa	varchar(191)	utf8mb4_unicode_ci		No	Ninguna			Cambiar Eliminar Más
☐	4 estado	tinyint(1)			No	1			Cambiar Eliminar Más
☐	5 foto	varchar(191)	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	6 imagen	varchar(191)	utf8mb4_unicode_ci		Si	NULO			Cambiar Eliminar Más
☐	7 es destacado	entero			Si	NULO			Cambiar Eliminar Más

Ilustración 15. Base de datos para la tabla **products**

Resumen de campos clave:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	identificación 	entero		NO FIRMADO	No	Ninguna		INCREMENTO AUTOMÁTICO	 Cambiar  Eliminar Más
2	SKU	varchar(255)	latin1_sueco_ci		Si	NULO			 Cambiar  Eliminar Más
3	tipo de producto	enum('normal', 'afiliado')	latin1_sueco_ci		No	normal			 Cambiar  Eliminar Más
4	enlace de afiliado	texto	latin1_sueco_ci		Si	NULO			 Cambiar  Eliminar Más
5	ID de usuario	entero			No	0			 Cambiar  Eliminar Más
6	id de categoría	entero		NO FIRMADO	No	Ninguna			 Cambiar  Eliminar Más
7	id de subcategoría	entero		NO FIRMADO	Si	NULO			 Cambiar  Eliminar Más
8	id de categoría infantil	entero		NO FIRMADO	Si	NULO			 Cambiar  Eliminar Más
9	atributos 	texto	utf8mb4_unicode_ci		Si	NULO			 Cambiar  Eliminar Más
10	nombre 	texto	utf8mb4_unicode_ci		No	Ninguna			 Cambiar  Eliminar Más
11	babosa	texto	utf8mb4_unicode_ci		Si	NULO			 Cambiar  Eliminar Más
12	foto	varchar(191)	utf8mb4_unicode_ci		No	Ninguna			 Cambiar  Eliminar Más
13	uña del pulgar	varchar(191)	utf8mb4_unicode_ci		Si	NULO			 Cambiar  Eliminar Más
14	archivo	varchar(191)	utf8mb4_unicode_ci		Si	NULO			 Cambiar  Eliminar Más
15	tamaño	varchar(191)	utf8mb4_unicode_ci		Si	NULO			 Cambiar  Eliminar Más
16	tamaño_cantidad	varchar(191)	utf8mb4_unicode_ci		Si	NULO			 Cambiar  Eliminar Más
17	talla_precio	varchar(191)	utf8mb4_unicode_ci		Si	NULO			 Cambiar  Eliminar Más
18	color	texto	utf8mb4_unicode_ci		Si	NULO			 Cambiar  Eliminar Más
19	precio	doble			No	Ninguna			 Cambiar  Eliminar Más
20	precio anterior	doble			Si	NULO			 Cambiar  Eliminar Más
21	detalles	texto	utf8mb4_unicode_ci		Si	NULO			 Cambiar  Eliminar Más

- **Ilustración 16. Tablas a crear**

En phpMyAdmin, una vista del listado de tablas del esquema (ej. categories, subcategories, childcategories, products, galleries, thumbnails, sliders, orders, order_items, users, currencies, generalsettings, etc.).

☐	complementos
☐	administradores
☐	idiomas de administración
☐	conversaciones_de_usuario_administrador
☐	mensajes de usuario administrador
☐	secciones de llegada
☐	atributos
☐	opciones_de_atributo
☐	pancartas
☐	blogs
☐	categorías del blog
☐	categorías
☐	categorías infantiles
☐	ciudades
☐	comentarios
☐	conversaciones
☐	contadores
☐	países
☐	cupones
☐	monedas
☐	repartidores
☐	depósitos

Componentes del Sistema (Vista General)

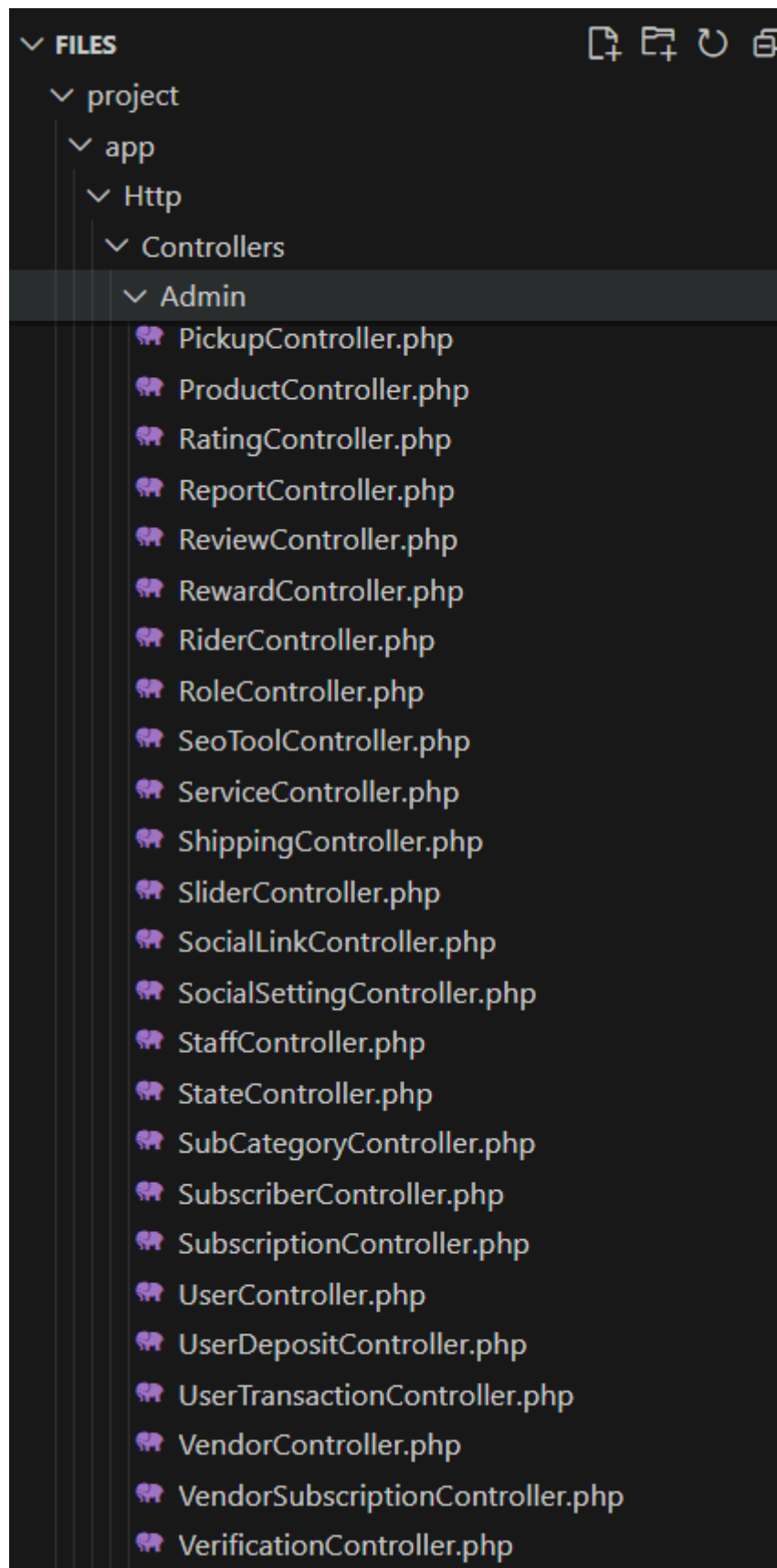
Esta sección resume los componentes principales del proyecto (controladores, modelos, middleware, helpers, resources y vistas) y dónde viven en el árbol de carpetas. Cada subsección incluye el rol del componente y la ruta exacta para ubicar el código.

2.1 Controladores (HTTP Controllers)

a) Panel Administrativo — app/Http/Controllers/Admin

Rol: CRUD y operaciones de negocio para catálogo, pedidos, envíos, contabilidad y configuración.

Ejemplos clave: ProductController.php, OrderController.php, PickupController.php, ShippingController.php, IncomeController.php, VendorController.php, UserController.php, CityController.php, StateController.php, CountryController.php, GeneralSettingController.php.



b) API Pública — app/Http/Controllers/Api/Front

Rol: endpoints JSON para catálogo, búsqueda, checkout y vista pública.

Ejemplos: ProductController.php, SearchController.php, FrontendController.php, CheckoutController.php, VendorController.php.

```
<?php

namespace App\Http\Controllers\Api\Front;

use App\{
    Models\Product,
    Models\Category,
    Models\Attribute,
    Models\Subcategory,
    Models\Childcategory,
    Models\AttributeOption
};

use App\{
    Http\Controllers\Controller,
    Http\Resources\CategoryResource,
    Http\Resources\AttributeResource,
    Http\Resources\ProductlistResource,
    Http\Resources\SubcategoryResource,
    Http\Resources\ChildcategoryResource,
    Http\Resources\AttributeOptionResource
};

use Illuminate\{
    Http\Request,
    Support\Collection
};

use Validator;

class SearchController extends Controller
{
    public function categories(Request $request) {
        if($request->type=="featured"){
            try{
                $cats = Category::where('status','=',1)->whereIsFeatured(1)->get();
            }
        }
    }
}
```

c) Autenticación API — app/Http/Controllers/Api/Auth

Rol: login/registro y emisión de tokens para apps/SPA.

Archivo: AuthController.php.

```

<?php

namespace App\Http\Controllers\Api\Auth;

use App\{
    Models\User,
    Models\GeneralSetting
};

use App\{
    Http\Controllers\Controller,
    Http\Resources\UserResource
};
use Illuminate\Support\Facades\Hash;
use Illuminate\Http\Request;
use Validator;
use Illuminate\Support\Str;
use JWTAuth;
use App\Classes\GeniusMailer;
use Auth;
class AuthController extends Controller
{
    /**
     * Create a new AuthController instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth:api', ['except' => ['login', 'register', 'logout', 'social_login', 'forgot']]);
        $this->middleware('setapi');
    }

    public function register(Request $request)
    {
        try {
            $rules = [

```

e) Front Web (Blade) — app/Http/Controllers/Front

Rol: flujo del carrito, catálogo, cupones, compare, checkout y vendors públicos.

Archivos: CartController.php, CatalogController.php, CheckoutController.php, VendorController.php, etc.

```

<?php

namespace App\Http\Controllers\Api\Payment;

use App\Http\Controllers\Front\FrontBaseController;
use App\Models\Currency;
use App\Models\Deposit;
use App\Models\Order;
use App\Models\PaymentGateway;
use DB;
use Illuminate\Http\Request;

class CheckoutController extends FrontBaseController
{
    public function loadpayment(Request $request, $slug1, $slug2)
    {
        if ($request->has('order_number')) {
            $order_number = $request->order_number;
            $order = Order::where('order_number', $order_number)->firstOrFail();
            $curr = Currency::where('sign', '=', $order->currency_sign)->firstOrFail();
            $payment = $slug1;
            $pay_id = $slug2;
            $gateway = '';
            if ($pay_id != 0) {
                $gateway = PaymentGateway::findOrFail($pay_id);
            }
            return view('payment.load.payment', compact('payment', 'pay_id', 'gateway', 'curr'));
        }
    }
}

```

f) Flujos de Autenticación (panel y roles) — app/Http/Controllers/Auth/*

Rol: login/registro/recuperación para Admin, User y Rider.

Rutas: Auth/Admin/*, Auth/User/*, Auth/Rider/*.

```

<?php

namespace App\Http\Controllers\Api\Auth;

use App\{
    Models\User,
    Models\Generalsetting
};

use App\{
    Http\Controllers\Controller,
    Http\Resources\UserResource
};
use Illuminate\Support\Facades\Hash;
use Illuminate\Http\Request;
use Validator;
use Illuminate\Support\Str;
use JWTAuth;
use App\Classes\GeniusMailer;
use Auth;
class AuthController extends Controller
{
    /**
     * Create a new AuthController instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth:api', ['except' => ['login', 'register', 'logout', 'social_login', 'forgot']
        $this->middleware('setapi');
    }
}

```

g) Paneles por Rol — app/Http/Controllers/User, .../Vendor, .../Rider

Rol: operaciones del usuario, vendedor y repartidor (pedidos, retiros, ingresos, notificaciones, verificación).

Archivos ejemplo: User/OrderController.php, Vendor/ProductController.php, Rider/RiderController.php.

```

<?php

namespace App\Http\Controllers\Vendor;

use App\Http\Controllers\Vendor\VendorBaseController;
use App\Helpers\PriceHelper;
use App\Models\Attribute;
use App\Models\AttributeOption;
use App\Models\Category;
use App\Models\Childcategory;
use App\Models\Currency;
use App\Models\Gallery;
use App\Models\Generalsetting;
use App\Models\Product;
use App\Models\Subcategory;
use Yajra\DataTables\Facades\DataTables;
use Illuminate\Support\Facades\DB;
use Illuminate\Http\Request;
use Illuminate\Support\Str;
use Intervention\Image\Facades\Image;
use Illuminate\Support\Facades\Validator;

class ProductController extends VendorBaseController
{
    /**** JSON Request
    public function datatables()
    {
        $user = $this->user;
        $datas = $user->products()
            ->where('product_type','normal')
            ->latest('id')
            ->get();

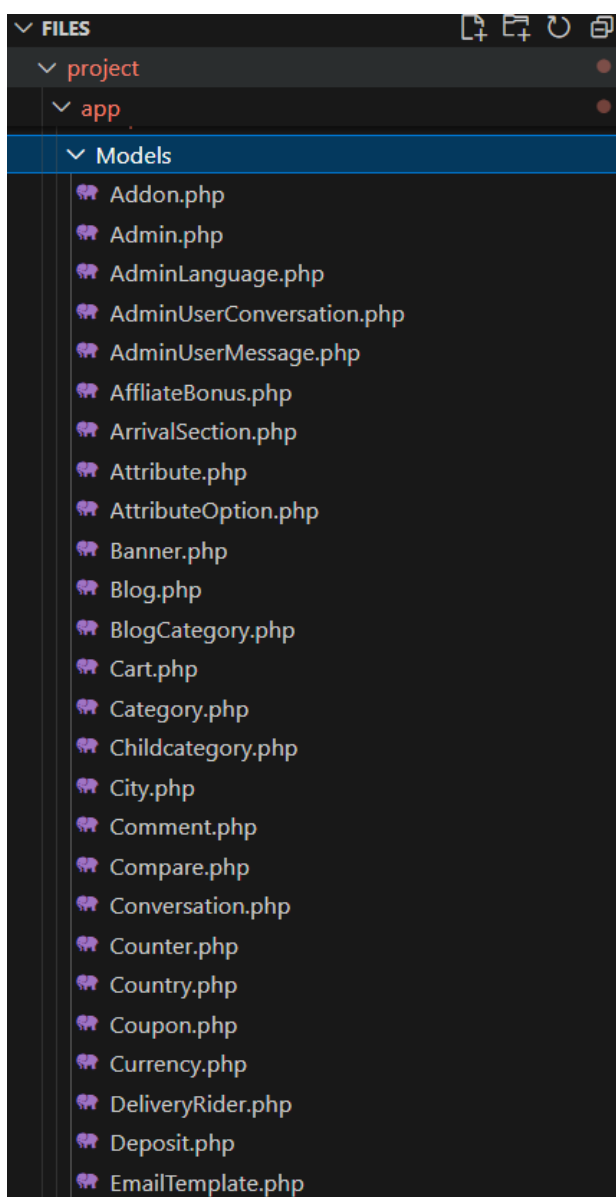
        return DataTables::of($datas)
            ->addIndexColumn()
    }
}

```

2.2 Modelos (Eloquent Models) — app/Models

Rol: representan tablas y relaciones (belongsTo/hasMany) del dominio.

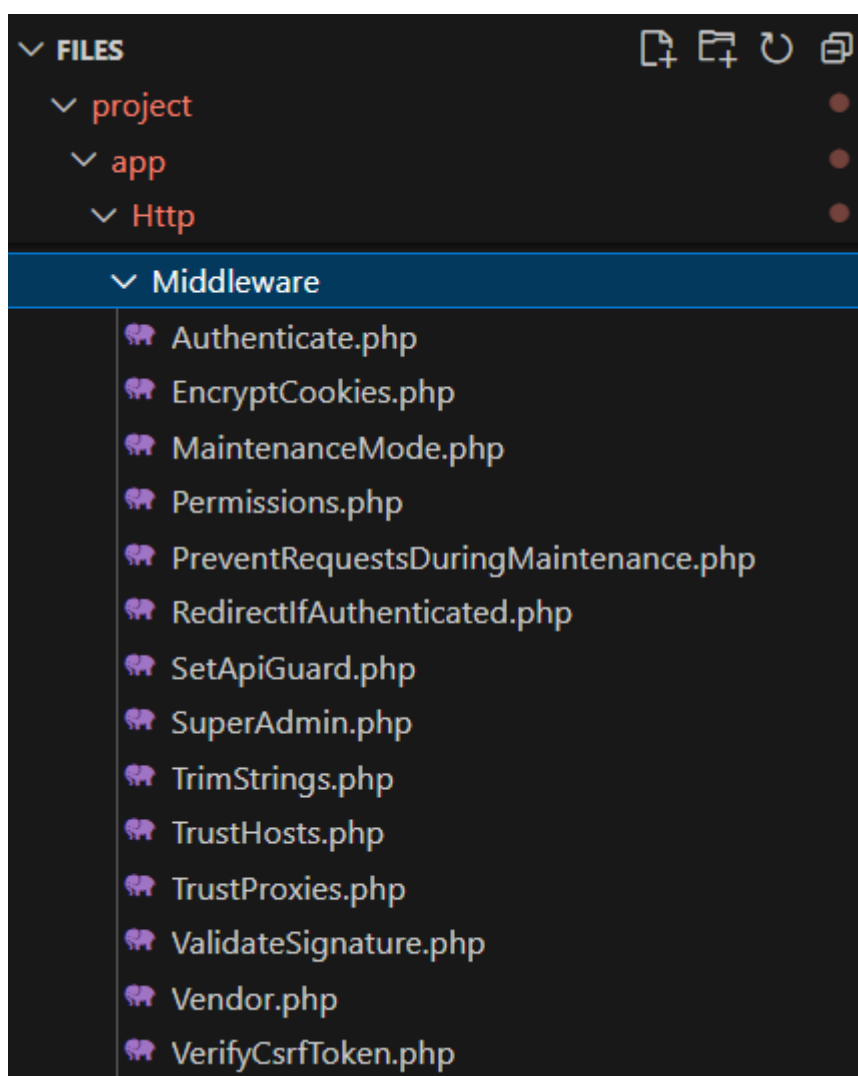
Modelos destacados: Product, Order, User, Vendor, PickupPoint, DeliveryRider, RiderServiceArea, Category, Subcategory, City, State, Country, Shipping, Income, Subscription, Review, Rating, Coupon.



2.3 Middleware — app/Http/Middleware

Rol: autorización y filtros para requests (csrf, auth, permisos, modo mantenimiento, guard API, roles Vendor/SuperAdmin).

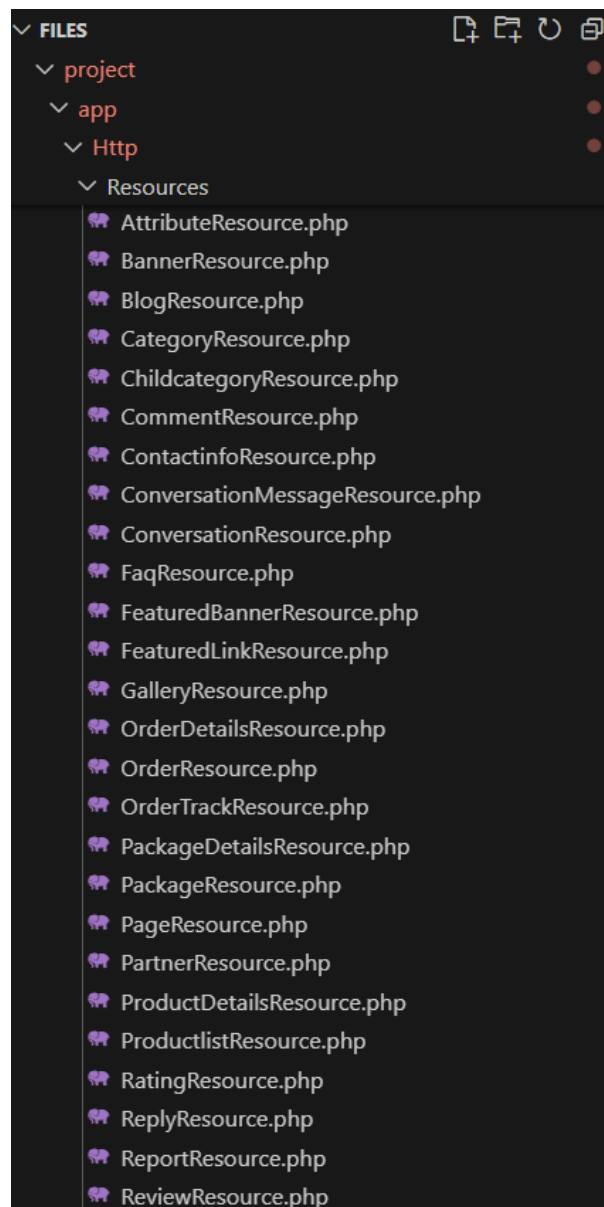
Archivos clave: Authenticate.php, Permissions.php, SuperAdmin.php, Vendor.php, SetApiGuard.php, VerifyCsrfToken.php.



2.4 API Resources — app/Http/Resources

Rol: serialización de respuestas JSON (transforman modelos a payload limpio para la API/Front).

Ejemplos: ProductResource.php, OrderResource.php, VendorResource.php, ShippingResource.php, ServiceResource.php, UserResource.php.



2.5 Helpers & Classes

Helpers — app/Helpers

Rol: utilitarios de negocio como precios, órdenes y cálculos.

Archivos: PriceHelper.php, OrderHelper.php.

```
<?php

namespace App\Helpers;

use App\{
    Models\Cart,
    App\Models\UserNotification
};

use App\Models\AffiliateBonus;
use Auth;
use Session;
use Illuminate\Support\Str;

class OrderHelper
{
    public static function auth_check($data)
    {
        try {
            $resdata = array();
            $users = User::where('email', '=', $data['personal_email'])->get();
            if (count($users) == 0) {
                if ($data['personal_pass'] == $data['personal_confirm']) {
                    $user = new User;
                    $user->name = $data['personal_name'];
                    $user->email = $data['personal_email'];
                    $user->password = bcrypt($data['personal_pass']);
                    $token = md5(time() . $data['personal_name'] . $data['personal_email']);
                    $user->verification_link = $token;
                    $user->affiliate_code = md5($data['personal_name'] . $data['personal_email']);
                    $user->email_verified = 'Yes';
                    $user->save();
                }
            }
        } catch (Exception $e) {
            // Handle exception
        }
    }
}
```

```
<?php

namespace App\Helpers;

use App\Models\Country;
use App\Models\Currency;
use App\Models\Package;
use App\Models\Shipping;
use App\Models\State;
use Session;
use DB;

class PriceHelper
{
    public static function showPrice($price)
    {
        $gs = cache()->remember('generalsettings', now()->addDay(), function () {
            return DB::table('generalsettings')->first();
        });
        if (is_numeric($price) && floor($price) != $price) {
            return number_format($price, 2, $gs->decimal_separator, $gs->thousand_separator);
        } else {
            return number_format($price, 0, $gs->decimal_separator, $gs->thousand_separator);
        }
    }

    public static function apishowPrice($price)
    {
        $gs = cache()->remember('generalsettings', now()->addDay(), function () {
            return DB::table('generalsettings')->first();
        });
        if (is_numeric($price) && floor($price) != $price) {
            return round($price, 2);
        } else {
            return round($price, 0);
        }
    }
}
```

Classes — app/Classes

Rol: servicios auxiliares (correo, certificados, SDKs).

Archivos: GeniusMailer.php, Instamojo.php, cacert.pem.

```
<?php

namespace App\Classes;

use App\{
    Models\Order,
    Models\EmailTemplate,
    Models\GeneralSetting
};
use Barryvdh\DomPDF\Facade\Pdf;
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
use Illuminate\Support\Str;

class GeniusMailer
{
    public $mail;
    public $gs;

    public function __construct()
    {
        $this->gs = GeneralSetting::findOrFail(1);

        $this->mail = new PHPMailer(true);

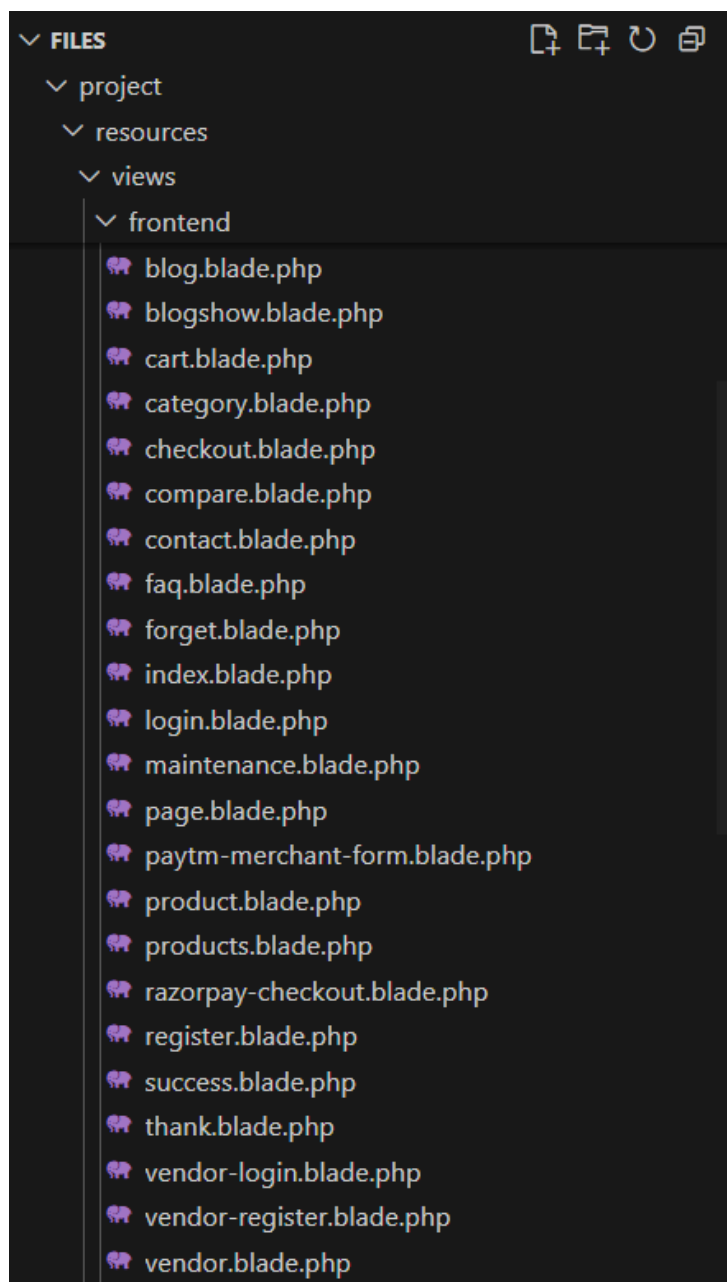
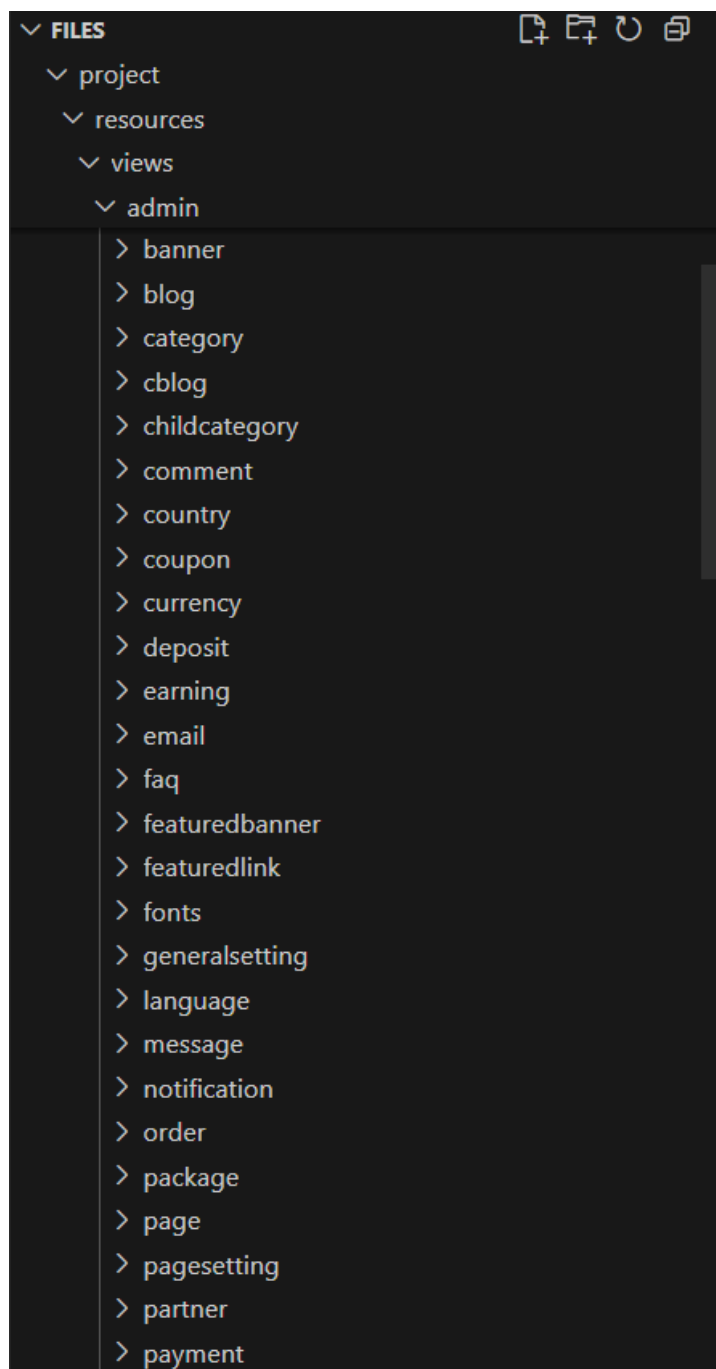
        if ($this->gs->is_smtp == 1) {
            $this->mail->isSMTP();
            $this->mail->Host = $this->gs->mail_host; // Set the SMTP server to send through
            $this->mail->SMTPAuth = true; // Enable SMTP authentication
            $this->mail->Username = $this->gs->mail_user; // SMTP username
            $this->mail->Password = $this->gs->mail_pass; // SMTP password
            $this->mail->SMTPSecure = $this->gs->mail_encryption; // Enable TLS encryption; "PHPMailer::ENCRYPTION_SMTPS" for SSL
            $this->mail->Port = $this->gs->mail_port;
        }
    }
}
```

2.6 Vistas (Blade) — resources/views

Admin: plantillas para productos, pedidos, envíos, idioma, SEO, reportes, suscripciones, usuarios y ajustes.

Front: páginas públicas (home, categoría, producto, carrito, checkout, vendor), parciales AJAX y paginación.

Incluye: secciones específicas como admin/product/*, admin/order/*, admin/shipping/*, frontend/*, y parciales en includes/*.

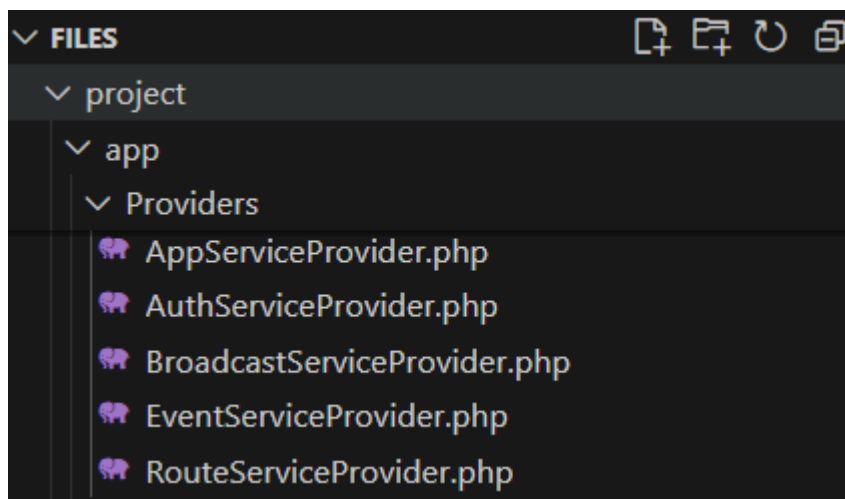


2.7 Proveedores & Configuración

Service Providers — app/Providers

Rol: bootstrap de servicios, eventos, rutas y políticas.

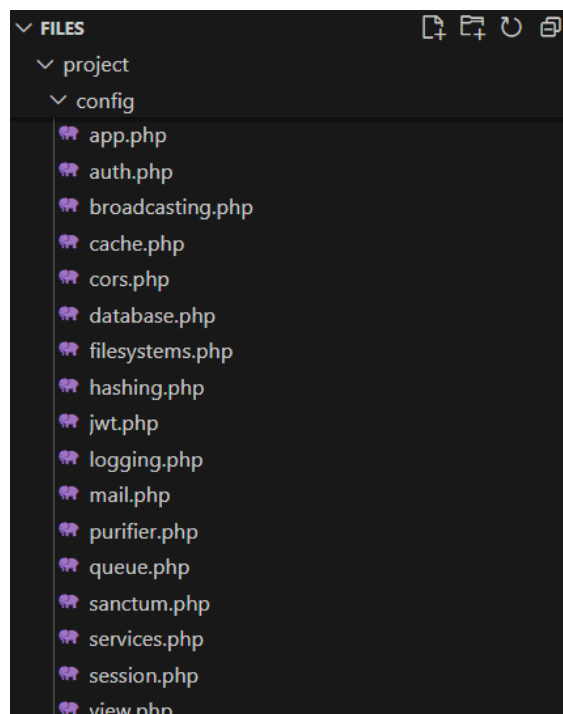
Archivos: AppServiceProvider.php, EventServiceProvider.php, AuthServiceProvider.php, RouteServiceProvider.php.



Configuración — config/*

Rol: parámetros de app, auth, cache, mail, queue, sanctum, services, etc.

Ejemplos: app.php, auth.php, database.php, mail.php, queue.php, sanctum.php, services.php.



4.9.Flujo de pago por WhatsApp (evidencia y control)

1. El cliente genera el pedido; el sistema emite recibo/factura PDF y muestra el botón “Enviar comprobante por WhatsApp” con mensaje prellenado (número de orden, total y enlace al recibo).
2. El cliente adjunta su comprobante; el sistema registra el archivo, sus metadatos y el mensaje en whatsapp_messages.
3. El proveedor revisa y decide: Aprobado, Observado (solicita corrección) o Rechazado (inconsistencia). La decisión se registra en payment_proofs y order_status_logs.
4. Si fue aprobado, se prepara el pedido; opcionalmente se asigna repartidor y se sigue el estado hasta “Entregado”.
5. El cierre del pedido solicita calificación, que alimenta los perfiles y la confianza barrial.

Este flujo reduce la fricción porque aprovecha el canal de comunicación más utilizado por la comunidad, sin dejar de ofrecer rastro verificable para auditoría y métricas.

5. Referencias

Dichter & Neira. (2024). El barómetro del Canal Tradicional y la percepción de los tenderos en Ecuador.

Dialnet. (2023). Las plataformas digitales y su impacto en las ventas de las pequeñas empresas.
Involves. (2024). Desafíos comunes en la gestión de tiendas tradicionales.

Mendoza, P., Torres, J., & Salazar, F. (2021). Desarrollo de una plataforma digital para el fomento del comercio local en comunidades rurales. *Revista de Tecnología y Sociedad*, 8(1), 22-35.

Chávez, S. (2021). Marco de referencia para la gestión de proyectos de desarrollo de software. *Revista Geeks*, Escuela Politécnica del Ejército, Quito, Ecuador.

CECE & UEES. (2025). Séptima medición del estudio de comercio electrónico en Ecuador. Cámara Ecuatoriana de Comercio Electrónico y Universidad Espíritu Santo.

Dialnet. (2023). Las plataformas digitales y su impacto en las ventas de las pequeñas empresas.
Ekos. (2025). El e-commerce ecuatoriano mantiene su crecimiento en 2025.

Mendoza, P., Torres, J., & Salazar, F. (2021). Desarrollo de una plataforma digital para el fomento del comercio local en comunidades rurales. *Revista de Tecnología y Sociedad*, 8(1), 22-35.

Primicias. (2024). El consumo a través de e-commerce creció 22% en 2024 en el país.

Statista. (2025). Proyecciones de acceso a internet y comercio electrónico en Ecuador.
Cevallos, A. (2020). Desarrollo de un marco referencial para la implantación y evaluación del uso de TIC en instituciones educativas. ESPOL.

Mendoza, P., Torres, J., & Salazar, F. (2021). Desarrollo de una plataforma digital para el fomento del comercio local en comunidades rurales. *Revista de Tecnología y Sociedad*, 8(1), 22-35.

Villavicencio, M. K., Cedeño, F., & Barba, P. (2007). Elaboración de marco referencial de gestión de proyectos aplicable a la realidad ecuatoriana basado en los modelos Moprosoft, CMMI y la normativa ISO/IEC 15504. ESPOL.

RedGEALC (2023). Marco de referencia para uso y desarrollo de tecnologías emergentes en América Latina y el Caribe.

Ministerio de Telecomunicaciones y de la Sociedad de la Información (2020). Decreto Ejecutivo 1073: Reglamento para la adquisición de software por parte de las entidades contratantes del sector público.

Ministerio de Telecomunicaciones y de la Sociedad de la Información (2017). Decreto Ejecutivo 1425: Reglamento para la adquisición de software.

Gobierno Electrónico Ecuador (2023). Software libre y Software Público.

Gobierno Electrónico Ecuador (2008). Estrategia para la implantación de Software Libre en la Administración Pública Central.

Servicio Nacional de Contratación Pública (2020). Resolución RE-SERCOP-2020-107: Adquisición de software y servicios relacionados.

Servicio de Rentas Internas (2023). Ley Orgánica para la Transformación Digital y Economía Audiovisual.

Asamblea Nacional (2002). Ley de Comercio Electrónico, Firmas y Mensajes de Datos. <https://www.asambleanacional.gob.ec/>

Castillo, J. & Sánchez, P. (2021). Plataformas digitales y economía colaborativa en América Latina. *Revista Tecnológica*, 18(2), 34-49.

Díaz, M. & Quishpe, C. (2022). Impacto de la digitalización en la economía local ecuatoriana. Universidad Andina Simón Bolívar.

El Comercio. (2021). Comercio electrónico y negocios de barrio en Ecuador. <https://www.elcomercio.com/>

García, F. (2020). Estrategias de digitalización para pequeños negocios. Quito: Ediciones UCE.

Gómez, R., Martínez, L. & Ruiz, F. (2020). La economía circular y las plataformas de servicios locales. *Innovar*, 30(77), 11-21.

LODC (2000). Ley Orgánica de Defensa del Consumidor. <https://www.defensoria.gob.ec/>

LOPDP (2021). Ley Orgánica de Protección de Datos Personales.
<https://www.asambleanacional.gob.ec/>

Montalvo, G., Ruiz, P. & Ordoñez, J. (2021). Digitalización y crecimiento de la economía local en Ecuador. Pontificia Universidad Católica del Ecuador.

SRI (2023). Resoluciones y normativa para la facturación electrónica. Servicio de Rentas Internas del Ecuador. <https://www.sri.gob.ec/>

Torres, D. (2022). Automatización de procesos en plataformas web de servicios. Revista TIC Ecuador, 7(1), 88-102.

astillo, J. & Sánchez, P. (2021). Plataformas digitales y economía colaborativa en América Latina. Revista Tecnológica, 18(2), 34-49.

García, F. (2020). Estrategias de digitalización para pequeños negocios. Quito: Ediciones UCE.

Gómez, R., Martínez, L. & Ruiz, F. (2020). La economía circular y las plataformas de servicios locales. Innovar, 30(77), 11-21.

Laravel (2024). Documentación oficial de Laravel. <https://laravel.com/docs>

Londoño, J. (2023). MySQL: gestión de bases de datos en aplicaciones web. Bogotá: Ediciones Uniandes.

Martínez, P. & Vaca, S. (2023). Digitalización de oficios y servicios: Experiencias en Ecuador. PUCE.

MDN Web Docs (2024). HTML: HyperText Markup Language. Mozilla Foundation.

Torres, D. & Ayala, C. (2022). Automatización de procesos en plataformas web de servicios. Revista TIC Ecuador, 7(1), 88-102.

Vásquez, C., Torres, M. & Salazar, J. (2022). La ingeniería de sistemas y soluciones ERP. Universidad Central del Ecuador.

Vue.js (2024). Documentación oficial. <https://vuejs.org/guide/introduction.html>

ANEXOS

MANUAL DE USUARIO

Primeros pasos

Requisitos del navegador

Para poder ingresar dentro de la red local al sistema web nos debemos dirigir a cualquier navegador (Firefox, Mozilla) se debe tomar en cuenta que el sistema solicita navegadores con al menos las dos últimas actualizaciones

Usa las **2 últimas versiones de Chrome, Firefox o Edge.**



1.2 URL de acceso

Escribe la dirección del sistema en la barra del navegador

<http://importadoracaisatoa.it.com/>

Verás la **página principal** con el menú superior (mi cuenta).



1.3 Estructura del encabezado (header)

En la barra superior encontrarás: **Menú, Buscador, Carrito, Mi cuenta.**

Desde **Mi cuenta** se despliegan accesos a iniciar sesión y registrarse.

Imagen aquí

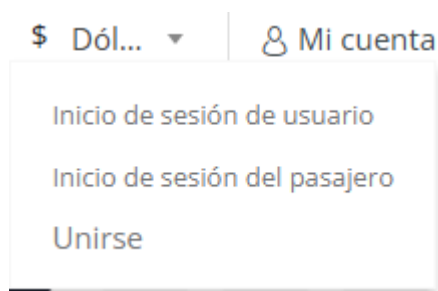


2. Crear una cuenta (Unirse)

2.1 Abrir menú “Mi cuenta”

Haz clic en **Mi cuenta** en el encabezado.

Se despliega el menú de opciones.



2.2 Seleccionar “Unirse”

En el menú, elige **Unirse** para crear una cuenta nueva.



2.3 Completar el formulario de registro

Nombre completo

Correo electrónico

Número de teléfono

Dirección

Contraseña y Confirmación

Presiona **Registro**.

Si el sistema requiere **verificación por correo**, revisa tu bandeja de entrada/spam.

Regístrate ahora

Registro

[¿Tienes una cuenta? Iniciar sesión](#)

2.4 Confirmación/Verificación (si aplica)

Mensaje indicando que se envió un correo de confirmación o que la cuenta fue creada.

Tratamiento...

Regístrate ahora

oscar caisatoa

oscarcaisatoa221@gmail.com

0995548070

AMAGUANA CUENDINA CALLE VENCEDORES Y 24 DE JULIO

....

....

Registro

3. Iniciar sesión / Recuperar acceso

3.1 Inicio de sesión de usuario (cliente)

Mi cuenta → Inicio de sesión de usuario.

Escribe **correo y contraseña** → **Entrar/Acceder.**

Editar perfil



Subir

Seleccionar país

▼

Seleccione Estado

▼

Seleccione Estado

▼

3.3 Recuperar contraseña

En la página de login, haz clic en **¿Olvidaste tu contraseña?**

Ingresa tu **correo** y confirma.

Revisa el **email** y sigue el enlace para **restablecer**.

Olvidé mi contraseña

Dirección de correo electrónico *

Introduzca su dirección de correo electrónico

Se enviará una contraseña a tu dirección de correo electrónico.

Entregar

4. Perfil y seguridad (usuarios de cualquier rol)

4.1 Ver el perfil

En la barra superior, clic en **Mi cuenta** → **Perfil**.

Verás **Nombre**, **Apellido**, **Email** (no editable).



PANEL

- Panel
- Panel de control de proveedores
- Artículos comprados
- Depósito
- Actas
- Recompensas
- Proceso de afiliación

Información de la cuenta

Óscar Iván Caisatoa Suntaxi
 Correo electrónico: oscar2@gmail.com
 Teléfono: 0995548070
 Fax: 0989971793
 Código postal: 171108
 Dirección: AMAGUAÑA/CUENDINA/CALLE VENCEDORES Y 24 DE JULIO

Mi billetera

Bono de afiliado:
0\$

Saldo de la billetera
0\$

[+ Agregar depósito](#)

4.2 Editar datos del perfil

Perfil → Editar.

Modifica **Nombre y Apellido**.

Presiona **Guardar**.

Nota: El **Email** no se edita porque es tu usuario de acceso.

Editar perfil

Panel > Editar perfil

Nombre de la tienda: importadoracaisatoa Verificado

Nombre del propietario * OSCARIN

Número de tienda * 67

WhatsApp (Opcional) 593989971793
Ingresa el número sin espacios ni signos. Ejemplo: 593099999999

Dirección de la tienda * Amaguaña alado de la panadería

Número de registro (Opcional) 10001

Detalles de la tienda *

B I U Tamaño de Familia de Formato de

4.3 Cambiar contraseña

Perfil → Seguridad/Contraseña.

Ingresa **Contraseña actual**, **Nueva** y **Confirmación**.

Guardar.

Editar perfil

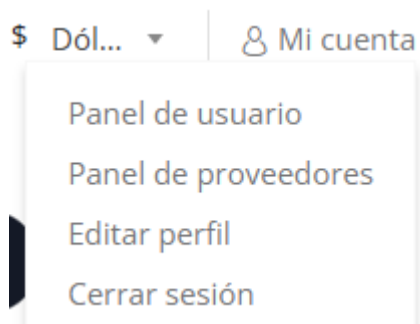


Subir

Oscar Ivan Caisatoa Sntaxi	oscar2@gmail.com
0995548070	0989971793
Seleccionar país ▼	▼
Sangolquí ▼	171108
AMAGUAÑA/CUENDINA/CALLE VENCEDORES Y 24 DE JULIO	

4.4 Cerrar sesión

Mi cuenta → **Cerrar sesión** para salir de forma segura.

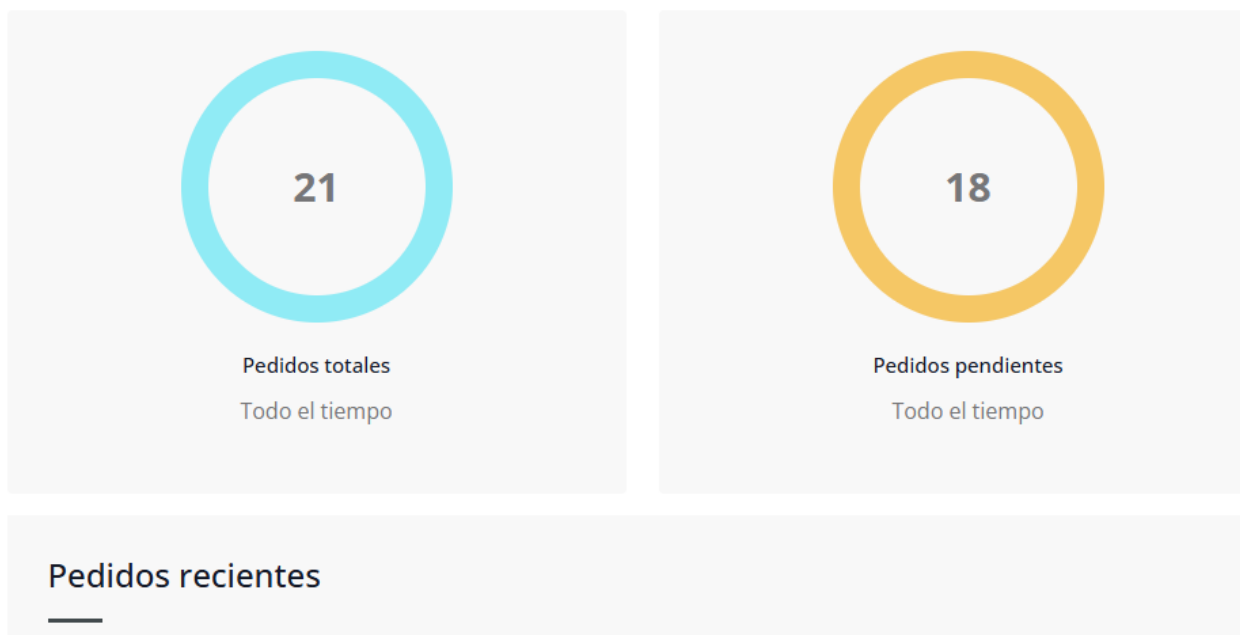


5. Panel del Usuario (Cliente)

5.1 Escritorio del cliente

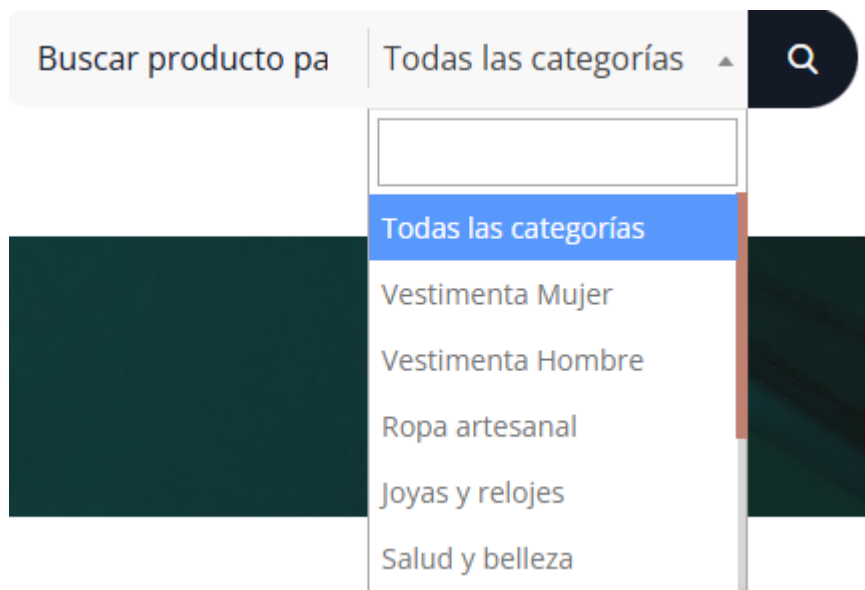
Resumen de **pedidos recientes**, accesos a **wallet**, **recompensas** (si aplica) y **transacciones**.

Imagen aquí



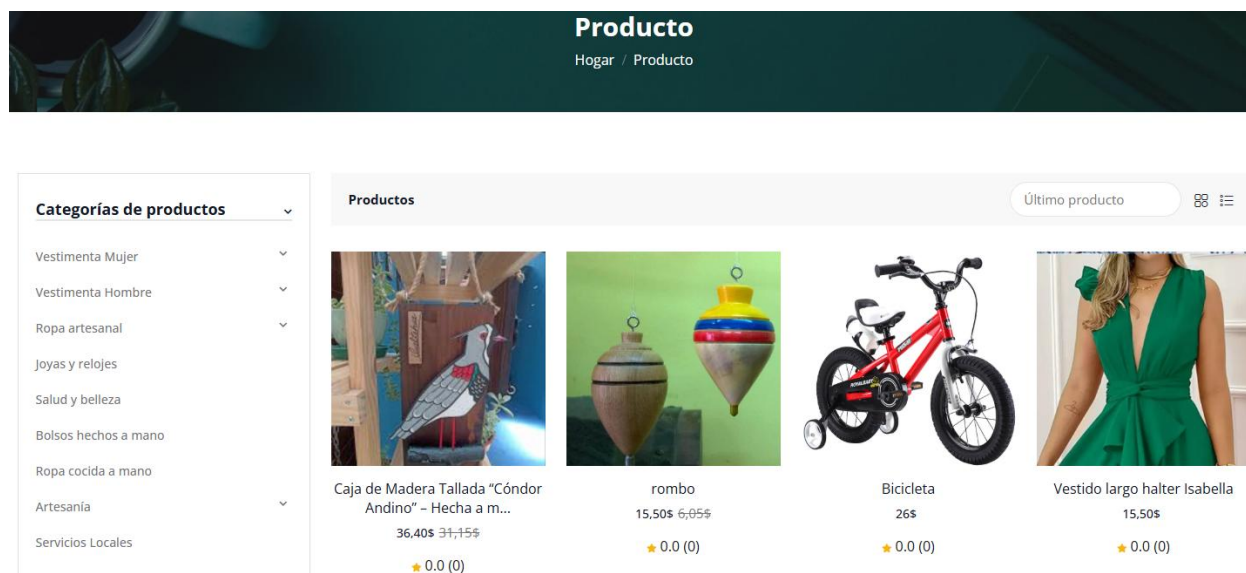
5.2 Catálogo y búsqueda

Usa el **buscador** y los **filtros** (categoría, precio, rating) para encontrar productos/servicios.



5.3 Detalle de producto/servicio

Verás **imágenes, descripción, precio, variantes** (si hay), **reseñas** y **botón de compra**.



5.4 Carrito y checkout

Añade al **Carrito**.

Revisa cantidades y aplica **cupones** si tienes.

Procede a **Checkout**: dirección, envío, pago.



Bicicleta

26\$

-

1

+

26\$ x

TOTALES DEL CARRITO

Total parcial	26\$
Descuento	0\$
Total	26\$

Pasar por la caja

1 DIRECCIÓN

2 Pedidos

3 Pago

Información personal:

Oscar Ivan Caisatoa Suintaxi

oscar2@gmail.com

Detalles de facturación

Dirección de envío

Oscar Ivan Caisatoa Suintaxi

oscar2@gmail.com

0995548070

AMAGUAÑA/CUENDINA/CALLE VENCEDORES Y 24 DE J

171108

Seleccionar país

☐ ¿Enviar a una dirección diferente?

DETALLES DE PRECIOS

MRP total	26\$
Impuesto	0.00 %
Costo de envío	\$0
Costo de embalaje	\$1
 ¿Tienes un código de promoción?	
Precio final:	27,00\$

5.5 Wallet (Billetera), depósitos y pagos

Puedes **depositar** saldo, **pagar con wallet** y consultar movimientos.

Información de pago

- ☒ **Pago contra reembolso**
Pagar en efectivo al momento de la entrega.
- ☐ **Billetera**
Paga desde tu billetera

5.6 Pedidos y seguimiento

Consulta el **historial**, estados (**pendiente**, **procesando**, **completado**, **rechazado**) y el **tracking**.

Todos los pedidos

Panel > Pedidos > Todos los pedidos

Espectáculo 10 entradas

Buscar:

Número de orden	Cantidad total	Costo total	Método de pago	Comportamiento	
ohKf1756425468	1	36,40\$	Pago contra reembolso	Detalles	Terminado
Kb5p1756422616	1	36,40\$	Pago contra reembolso	Detalles	Terminado
c57q1756411838	1	26\$	Pago contra reembolso	Detalles	Terminado
FjKk1756411828	1	26\$	Pago contra reembolso	Detalles	Tratamiento
dQvY1756231756	1	15,50\$	Pago contra reembolso	Detalles	Terminado
h4AB1756231067	1	26\$	Pago contra reembolso	Detalles	Pendiente

5.7 Favoritos y mensajes

Marca **vendedores favoritos** y **envía mensajes** para aclaraciones o soporte.

Vendedores favoritos

Panel / Vendedores favoritos

Vendedores favoritos

Nombre de la tienda	Nombre del propietario	DIRECCIÓN	Comportamiento
Talleres de prueba	Saxena	fgfg	 
Talleres de prueba	Saxena	fgfg	 

Mensajes

 Redactar mensaje

Espectáculo entradas

Buscar:

Nombre	Mensaje	Enviado	Comportamiento
No hay datos disponibles en la tabla			

Mostrando 0 a 0 de 0 entradas

Anterior

Próximo

5.8 Tickets y disputas

Abre **tickets** de ayuda o **disputas** por pedidos específicos (adjunta evidencia).

Disputar

✉ Agregar disputa

Espectáculo

10

entradas

Buscar:

Sujeto	Mensaje	Tiempo	Comportamiento
No hay datos disponibles en la tabla			

Mostrando 0 a 0 de 0 entradas

Anterior

Próximo

6. Panel del Proveedor/Vendedor

6.1 Escritorio del vendedor

KPIs: pedidos **pendientes/en proceso/completados**, total de **productos, artículos vendidos, ganancias** y **gráfico de 30 días**.



6.2 Crear producto

Menú **Productos** → **Crear**.

Completa **datos básicos** (título, precio, stock), sube **imágenes**.

Configura **SEO/etiquetas**, **variantes** (si aplica).

Guardar.

Producto físico [← Atrás](#)

Panel > Productos > Todos los productos > Agregar producto > Producto físico

Nombre del producto* (En cualquier idioma)

Introduzca el nombre del producto

Código del producto*

ghY1460BAN

Categoría*

Seleccionar categoría

Subcategoría*

Seleccionar subcategoría

Categoría infantil*

Seleccionar categoría infantil

☐ Permitir condición del producto

☐ Permitir pedidos anticipados de productos

Imagen destacada *

Subir imagen aquí

Imágenes de la galería de productos *

+ Galería de sets

Precio actual del producto* (En USD)

p. ej. 20

Precio de descuento del producto* (Opcional)

p. ej. 20

6.3 Configuración avanzada de producto

Atributos, inventario, envíos, imágenes, estado (activo/inactivo).

Productos

Panel > Productos > Todos los productos

Espectáculo 10 entradas Buscar: [+ Agregar nuevo producto](#)

Nombre	Tipo	Precio	Estado	Opciones
Caja de Madera Tallada "Cóndor Andino" – Hecha a m... ID Producto: 00000407	Físico	29,90\$	Activado ▾	Editar Eliminar
rombo ID del producto: 00000405	Físico	10\$	Activado ▾	Editar Eliminar
Bicicleta ID de producto: 00000404	Físico	20\$	Activado ▾	Editar Eliminar
Vestido Largo Halter Isabella ID del producto: 00000403	Físico	10\$	Activado ▾	Editar Eliminar
Televisor TCL ID de producto: 00000402	Físico	900\$	Activado ▾	Editar Eliminar
Aretes Gato Tiffany Entrelazos ID Producto: 00000401	Físico	15\$	Activado ▾	Editar Eliminar
Televisor Sony ID de producto: 00000398	Físico	10\$	Activado ▾	Editar Eliminar

6.4 Carga masiva y productos afiliados

Importa CSV/Excel para **carga masiva** o crea **producto afiliado** con URL externa.

Carga masiva de productos

Panel > Carga masiva

[Descargar muestra CSV](#)

Seleccione idioma*

Español ▾



Subir un archivo *

[Seleccionar archivo](#) Ningún archivo seleccionado

[Iniciar importación](#)

Producto afiliado

Atrás

Panel > Productos afiliados > Todos los productos > Agregar producto afiliado

Nombre del producto* (En cualquier idioma)

Introduzca el nombre del producto

Código del producto*

u9N1545Qb7

Enlace de afiliado del producto* (Enlace externo)

Ingresar el enlace del producto

Categoría*

Seleccionar categoría

Subcategoría*

Seleccionar subcategoría

Categoría infantil*

Seleccionar categoría infantil

Stock de producto* (Dejar vacío se mostrará siempre disponible)

Fuente de la imagen destacada*

Archivo

Imagen destacada *

Subir imagen aquí

Imágenes de la galería de productos *

+ Galería de sets

Precio actual del producto* (En USD)

p. ej. 20

6.5 Gestión de pedidos del vendedor

Lista y filtra pedidos; cambia estados; genera comprobantes.

Asignar entrega al Rider cuando el pedido lo requiera.

Imagen aquí

Número de orden	Cantidad total	Costo total	Método de pago	Comportamiento
ohKf1756425468	1	36,40\$	Pago contra reembolso	Detalles Terminado
Kb5p1756422616	1	36,40\$	Pago contra reembolso	Detalles Terminado

Terminado

Pendiente

Tratamiento

Terminado

Rechazado

Terminado

7. Panel de Administración

7.1 Escritorio (Dashboard)

KPIs de **pedidos** (pendientes, en proceso, completados, rechazados), totales de **productos**, **clientes**, **publicaciones**, **estadísticas** (30 días y acumuladas), **pedidos/clientes recientes**, **productos populares**, **gráfico de ventas**.

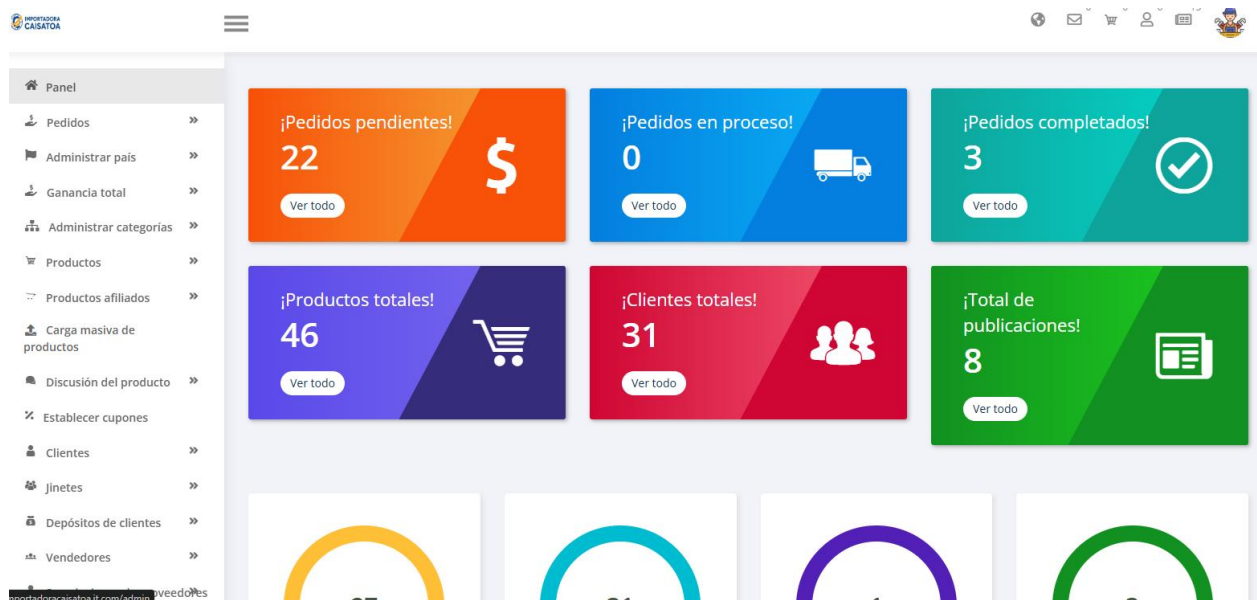


Ilustración 41. Dashboard del administrador: tarjetas KPI.



Ilustración 42. Gráfico de ventas de los últimos 30 días.

8.2 Pedidos (admin) y POS

Lista/filtra pedidos por estado; detalle e impresión; **crear pedido con POS.**

Todos los pedidos

Panel > Pedidos > Todos los pedidos

Espectáculo 10 entradas

Buscar:

[+ Agregar un pedido](#)

Correo electrónico del cliente	Número de orden	Cantidad total	Costo total	Opciones
oscar2@gmail.com	ohKf1756425468	4	37,40\$	Comportamiento ▼
oscar2@gmail.com	Kb5p1756422616	1	37,40\$	Comportamiento ▼
oscar2@gmail.com	c57q1756411838	1	29\$	Comportamiento ▼
oscar2@gmail.com	FjKk1756411828	1	29\$	Comportamiento ▼
kkkkkk@gmail.com	kv4O1756248256	2	35\$	Comportamiento ▼
oscar2@gmail.com	0IUv1756232047	1	329\$	Comportamiento ▼
oscar2@gmail.com	dQvY1756231756	1	22,50\$	Comportamiento ▼
oscar2@gmail.com	h4AB1756231067	1	33\$	Comportamiento ▼
oscar2@gmail.com	6Aux1756230949	3	186,40\$	Comportamiento ▼
oscar2@gmail.com	eVrc1756230124	2	43,40\$	Comportamiento ▼

Ilustración 43. Listado de pedidos (admin).

Detalles del pedido

[← Atrás](#)

Panel > Pedidos > Detalles del pedido

Detalles del pedido

No pagado

ID del pedido : ohKf1756425468
 Producto total : 4
 Costo de embalaje : 1\$
 Costo total : 37,40\$
 Fecha de pedido : 28 de agosto de 2025, 23:57:48
 Método de pago : Pago contra reembolso
 Estado del pago :

[Ver factura](#)

Ilustración 44. Detalle de pedido (admin).

Crear un pedido
Panel > Pedidos > Crear un pedido

car:

Producto



El título del producto de afiliado aparecerá aquí. VENDEDOR : Test
Stores.
Precio: 50 \$.



Cámara Digital Fujifilm Instax PAL Pistacho VENDEDOR: Tiendas de prueba
Precio: 139.9 \$

Opciones

Detalles del usuario

Seleccionar usuario

Nombre * Correo electrónico * Teléfono *

DIRECCIÓN *

País * Ciudad

Estado Código Postal

Ilustración 45. Creación de pedido mediante POS.

8.3 Envíos y embalaje

Configura **métodos de envío**, reglas por **zona** y **embalaje**.

Métodos de envío
Panel > Configuración general > Métodos de envío

Envío múltiple: Activado

Espectáculo entradas Buscar:

Título	Precio	Opciones
Envío gratis	0\$	Editar Eliminar
Envío exprés	10\$	Editar Eliminar
Servicios médicos de emergencia	26\$	Editar Eliminar
Básico	5\$	Editar Eliminar
Estándar	10\$	Editar Eliminar
Pro	15\$	Editar Eliminar
prueba	40\$	Editar Eliminar
Entrega en local	0\$	Editar Eliminar
Repartidor	0\$	Editar Eliminar

Ilustración 46. Configuración de envíos/embalaje.

8.4 ubicaciones

Lugares, ciudades, zonas y puntos de recogida (pickup);

Lugares de recogida

Panel > Configuración general > Lugares de recogida

Espectáculo entradas Buscar:

ubicación	Opciones
Universidad de las Fuerzas Armadas ESPE	✎ Editar 🗑
Parque Central de Sangolquí	✎ Editar 🗑
Plaza Comercial San Rafael	✎ Editar 🗑
Parque Central de Amaguaña	✎ Editar 🗑
Compras en Plaza del Valle	✎ Editar 🗑
Centro comercial River	✎ Editar 🗑
Compras en San Luis	✎ Editar 🗑
Quicentro Sur	✎ Editar 🗑
Terminal Terrestre Quitumbe	✎ Editar 🗑

Ilustración 49. Puntos de recogida (pickup points).

8.5 Catálogo y categorías

Categorías dinámicas, productos activos/inactivos, catálogo, reseñas/comentarios.

Categorías principales

Panel > Administrar categorías > Categorías principales

Espectáculo 10 entradas [+ Agregar nuevo](#)

Nombre	Babosa	Atributos	Presentado	Estado	Opciones
Educación y Oficios	Educación y Oficios	Crear	Activado	Desactivado	Editar Borrar
Salud y bienestar	Salud y Bienestar	Crear	Desactivado	Desactivado	Editar Borrar
Servicios Locales	Servicios Locales	Crear	Activado	Activado	Editar Borrar
Artesanía	Obras hechas a mano	Crear	Activado	Activado	Editar Borrar
Ropa cocida a mano	para-hombre-o-mujer	Crear	Activado	Activado	Editar Borrar
Aire libre, recreación	Recreación y ejercicio al aire libre	Crear Administrar	Desactivado	Desactivado	Editar Borrar
Bolsos hechos a mano	bolsas de mano	Crear	Desactivado	Desactivado	Editar Borrar
Bolsos hechos a mano	Decoración del hogar y electrodomésticos	Crear Administrar	Activado	Activado	Editar Borrar

Ilustración 50. Gestión de categorías del catálogo.

Productos

Panel > Productos > Todos los productos

Espectáculo 10 entradas [+ Agregar producto](#)

Nombre	Tipo	Existencias	Precio	Estado	Opciones
Caja de Madera Tallada "Cóndor Andino" – Hecha a m... ID: 00000407	Físico	Ilimitado	29,90\$	Activado	Editar Borrar
ID de rombo : 00000405	Físico	Ilimitado	10\$	Activado	Editar Borrar
Bicicleta ID: 00000404	Físico	Ilimitado	20\$	Activado	Editar Borrar
Vestido Largo Halter Isabella ID: 00000403	Físico	Ilimitado	10\$	Activado	Editar Borrar
Televisor TCL ID: 00000402	Físico	Ilimitado	900\$	Activado	Editar Borrar
Aretes Gato Tiffany Entrelazos ID: 00000401	Físico	10	15\$	Activado	Editar Borrar
Identificación del TCL QM8K : 00000400	Físico	Ilimitado	500\$	Activado	Editar Borrar

Ilustración 51. Lista de productos (admin).

8.6 Cupones y promociones

Crear, editar, vigencia, límites.

Cupones
Panel > Cupones

Espectáculo entradas Buscar: + Agregar nuevo

Código	Tipo	Cantidad	Usado	Estado	Opciones
procoo	Descuento por porcentaje	10%	0	Activado ▼	Editar Eliminar
proco	Descuento por importe	10\$	0	Activado ▼	Editar Eliminar

Ilustración 53. Gestión de cupones.

8.7 Clientes, transacciones y depósitos/retiros

Ver clientes, **transacciones**, **depósitos** y **retiros**; imagen predeterminada del cliente.

Clientes
Panel > Clientes

Espectáculo entradas Buscar: + Agregar cliente

Nombre	Correo electrónico	Opciones
Óscar Caisatoa	oscarcaisatoa221@gmail.com	\$ Administrar depósito Hacer proveedor Detalles Editar Enviar Desatascar ▼ Eliminar
Óscar Caisatoa	oscarcaisatoa22@gmail.com	\$ Administrar depósito Hacer proveedor Detalles Editar Enviar Desatascar ▼ Eliminar
Óscar Iván Caisatoa Suntaxi	jose.l.alvarez21@gmail.com	\$ Administrar depósito Proveedor Detalles Editar Enviar Desatascar ▼ Eliminar

Ilustración 54. Gestión de clientes.

Retiros						
Panel > Clientes > Retiros						
Espectáculo	10	entradas	Buscar:			
Correo electrónico	Teléfono	Cantidad	Método	Fecha de retiro	Estado	Comportamiento
No hay datos disponibles en la tabla						
Mostrando 0 a 0 de 0 entradas					Anterior	Próximo

Ilustración 55. Transacciones, depósitos y retiros.

8.8 Proveedores y verificaciones

Altas, verificaciones, suscripciones, retiros, comisiones.

Registro de proveedores: Activado

Espectáculo

10

entradas

Buscar:

Nombre de la tienda	Correo electrónico del proveedor	Número de tienda	Comisión pendiente	Estado	Opciones
ister.com	jose.l.alvarez21@gmail.com	113	0\$ Recolectar	Activado	Comportamiento
yosoyasi	oscarcaisatoa2@gmail.com	124	0\$ Recolectar	Activado	Comportamiento
importadoracaisatoaaaa	madeleine@gmail.com	1123	0\$ Recolectar	Desactivado	Comportamiento
importadoracaisatoa	oscar2@gmail.com	67	0\$ Recolectar	Activado	Comportamiento
Talleres de prueba	usuario@gmail.com	213213	112\$ Recolectar	Activado	Comportamiento

Ilustración 56. Gestión de proveedores.

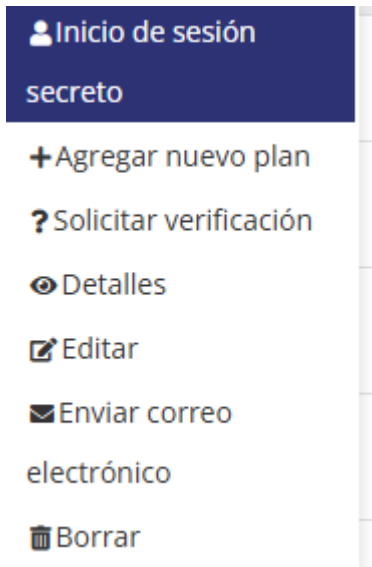


Ilustración 57. Verificaciones (pendientes y aprobadas).

8.9 Riders (dinámico) y áreas de servicio

Alta/baja de **riders**, retiros y zonas de reparto.

Imagen aquí

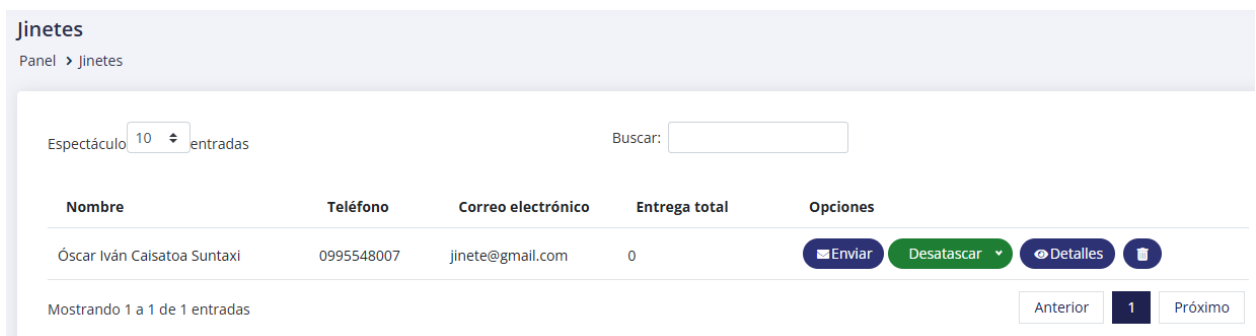


Imagen aquí

Retiros

Panel > Clientes > Retiros

Espectáculo 10  entradas

Buscar:

Correo electrónico	Teléfono	Cantidad	Método	Fecha de retiro	Estado	Comportamiento
--------------------	----------	----------	--------	-----------------	--------	----------------

No hay datos disponibles en la tabla

Mostrando 0 a 0 de 0 entradas

Anterior

Próximo

8.10 Contenido, apariencia y blog

CMS de páginas, **FAQ**, **contacto**; **logos**, **favicon**, **loader**; **temas**; **home** (slider, servicios, llegadas, oferta del día, socios); **blog** (categorías/publicaciones).

 Configuración general 

Logo

Favicon

Cargador

Métodos de envío

Embalajes

Lugares de recogida

Contenido del sitio web

Ilustración 61. Gestión de páginas y FAQ.

Personalización de la página de inicio

Panel > Configuración de la página de inicio > Personalización de la página de inicio

Control deslizante *	☒	Sección de Llegada *	☒
Productos destacados *	☒	Oferta del día *	☒
Mejor valorado, grandes ahorros y tendencia *	☒	Pareja *	☒
Producto más vendido *	☒	Blogs *	☒
Boletín informativo *	☒		

Entregar

Ilustración 63. Personalización de la página de inicio.

Configuración del blog

Panel > Blog > Configuración del blog

Disqus Activado ▾

Nombre corto del sitio web de Disqus *

importadora

Mostrar publicaciones por página *

6

Ahorrar

Ilustración 64. Blog: categorías y publicaciones.

8.12 Pagos

Pasarelas (Pago contra entrega.).

Panel > Configuración de pago > Pasarelas de pago

Espectáculo 10 entradas

Buscar:

+ Agregar nuevo pago

Nombre	Detalles	Verificar	Depósito	Suscripción	Opciones
SSLCommerz	Pagar a través de SSLCommerz.	No mostrado	Presentado	Presentado	Editar
Dinero móvil	Número de pago: 69234324233423	No mostrado	Presentado	Presentado	Editar Eliminar
Pago contra reembolso	Pagar en efectivo al momento de la entrega.	Presentado	No disponible	No disponible	Editar

Ilustración 67. Pago

Información de pago

Panel > Configuración de pago > Información de pago

Información básica

Información del proveedor

Pago como invitado

Activado

Formato de moneda *

Después del precio

Separador decimal *

Punto(.)

Separador de miles *

Coma(,)

Tarifa de retiro *

10

Cargo por retiro (%) *

5

Ilustración 68. Información pago

8.13 Social login, idioma y fuentes

Configura Facebook/Google Login, idiomas (sitio y panel) y fuentes.

Enlaces sociales

Panel > Configuración general > Enlaces sociales

Espectáculo 10 entradas Buscar: [+ Agregar nuevo enlace](#)

Enlace	Estado	Opciones
https://dribbble.com/	Activado	Editar Eliminar
https://www.google.com/	Activado	Editar Eliminar
https://linkedin.com/	Activado	Editar Eliminar
https://twitter.com/	Activado	Editar Eliminar
https://www.facebook.com/	Activado	Editar Eliminar

Mostrando 1 a 5 de 5 entradas [Anterior](#) **1** [Próximo](#)

Ilustración 69. Login con Facebook/Google.

Todas las fuentes

Panel > Fuentes de sitios web

Espectáculo 10 entradas Buscar: [+ Agregar nueva fuente](#)

Fuentes	Opciones
Open Sans	Editar Eliminar ✓ Por defecto
Guión de baile	Editar Eliminar Establecer predeterminado
Pangolín	Editar Eliminar Establecer predeterminado
Exhibición de Libre Caslon	Editar Eliminar Establecer predeterminado
Roboto Mono	Editar Eliminar Establecer predeterminado
Roboto	Editar Eliminar Establecer predeterminado

Mostrando 1 a 6 de 6 entradas [Anterior](#) **1** [Próximo](#)

Ilustración 71. Opciones de tipografía/fuentes.

8.15 Mantenimiento, perfil admin y recuperación

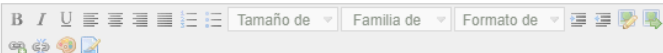
Modo mantenimiento, perfil del admin, cambio/recuperación de contraseña.

Mantenimiento de sitios web

Panel > Configuración general > Mantenimiento de sitios web

Mantenimiento de sitios web Desactivado ▾

Texto de mantenimiento *
(En cualquier idioma)



¡EN MANTENIMIENTO!

Ahorrar

Ilustración 75. Activar modo mantenimiento.

9.2 Compartir en redes (blog)

Comparte publicaciones de blog en redes sociales desde los botones del post.

Publicaciones

Panel > Blog > Publicaciones

Espectáculo 10 entradas [+ Añadir nueva publicación](#)

Imagen destacada	Título de la publicación	Vistas	Opciones
	¿Cómo diseñar artes efectivas?	10	Editar Eliminar
	¿Cómo diseñar artes efectivas?	82	Editar Eliminar
	¿Cómo diseñar artes efectivas?	11	Editar Eliminar

Ilustración 78. Compartir un post del blog.

9.3 Buscador y ordenamiento avanzado

Buscador global y **ordenar por**: relevancia, precio, fecha, rating.

Buscar producto pa

Todas las categorías
 Vestimenta Mujer
 Vestimenta Hombre
 Ropa artesanal
 Joyas y relojes
 Salud y belleza

Ilustración 79. Buscador y ordenamientos.

9.4 Notificaciones por correo

Correos automáticos por **pedidos, registro, reseteo, disputas.**

Espectáculo entradas Buscar:

Correo electrónico del cliente	Número de orden	Cantidad total	Costo total	Opciones
madeleine@gmail.com	JOoj1754700050	1	143\$	Comportamiento ▾ Ver detalles Enviar Seguimiento de pedidos Código
oscar2@gmail.com	z5c21754170368	2	143\$	
oscar2@gmail.com	CPQa1753822636	1	138\$	

Mostrando 1 a 3 de 3 entradas

Ilustración 80. Ejemplo de notificación por correo.

10. Buenas prácticas y soporte

Mantén tu **navegador** actualizado.

Usa **contraseñas seguras** y no las compartas.

Verifica tus **datos** (dirección, teléfono, método de pago).

Para incidencias, abre un **ticket** y adjunta captura/ID del pedido.



¿Tienes preguntas? ¡Llámanos 24/7!
0995548070

DIRECCIÓN :

Amaguaña

Correo electrónico :

oscarcaisatoa2@gmail.com

Ilustración 82. Centro de ayuda / soporte.

Manual del Desarrollador — *Importadora Caisatoa* (Laravel 12)

Propósito: Documentar instalación, configuración, estructura, base de datos, módulos e historias de usuario, ejecución local, despliegue mínimo y buenas prácticas para este proyecto.

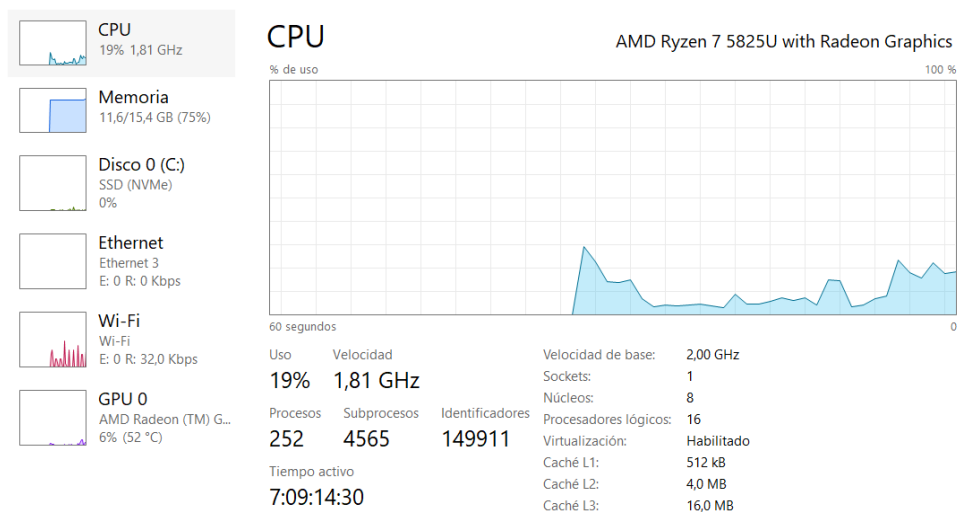
Requisitos del equipo

Hardware (Windows 10/11)

CPU: Intel i3 / Ryzen 3 o superior (recomendado 4 núcleos lógicos).

RAM: 4 GB mínimo (8 GB recomendado).

Almacenamiento: 10–20 GB libres (SSD recomendado).



Administrador de tareas → *Rendimiento* (CPU/RAM).

Software

PHP 8.2+ (CLI y extensiones comunes).

Laravel (12)

Composer 2.x (gestor de dependencias PHP).

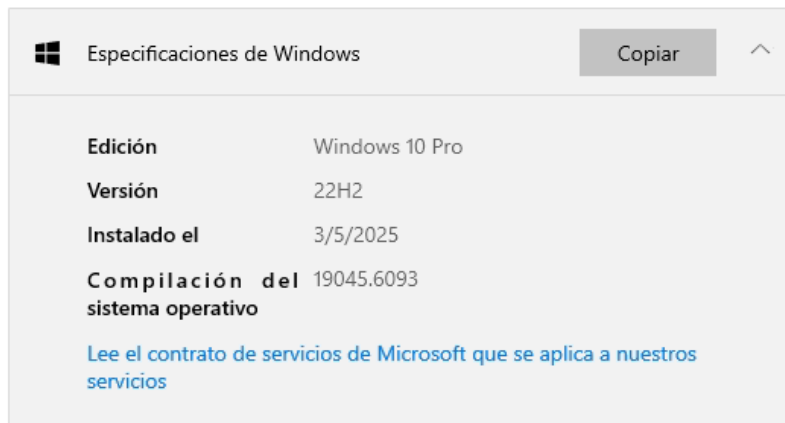
MySQL 8.4.x (servidor y cliente).

Git 2.x (opcional, si clonas desde repositorio).

Navegador moderno (Chrome/Edge/Firefox).

Editor: Visual Studio Code / PHPStorm.

Como se había mencionado en las especificaciones de software y hardware, debemos tener instalado un sistema operativo Windows 10 o superior, adicional memoria RAM de 4GB, en donde podamos usar mínimo 4 procesadores del sistema como se muestra a continuación:



Sistema > Información

 Almacenamiento

477 GB

360 GB de 477 GB usado

 Tarjeta gráfica

496 MB

AMD Radeon (TM) Graphics

 RAM instalada

16,0 GB

 Procesador

**AMD Ryzen 7 5825U
with Radeon Graphics**

2.00 GHz

Instalar PHP 8.2 (Windows)

Descarga PHP 8.2 “Thread Safe” para Windows desde la web oficial.

Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.

XAMPP for Windows 8.0.28, 8.1.17 & 8.2.4

Version		Checksum	Size
8.0.28 / PHP 8.0.28	What's Included?	md5 sha1 Download (64 bit)	144 Mb
8.1.17 / PHP 8.1.17	What's Included?	md5 sha1 Download (64 bit)	148 Mb
8.2.4 / PHP 8.2.4	What's Included?	md5 sha1 Download (64 bit)	149 Mb

[Requirements](#) [More Downloads »](#)

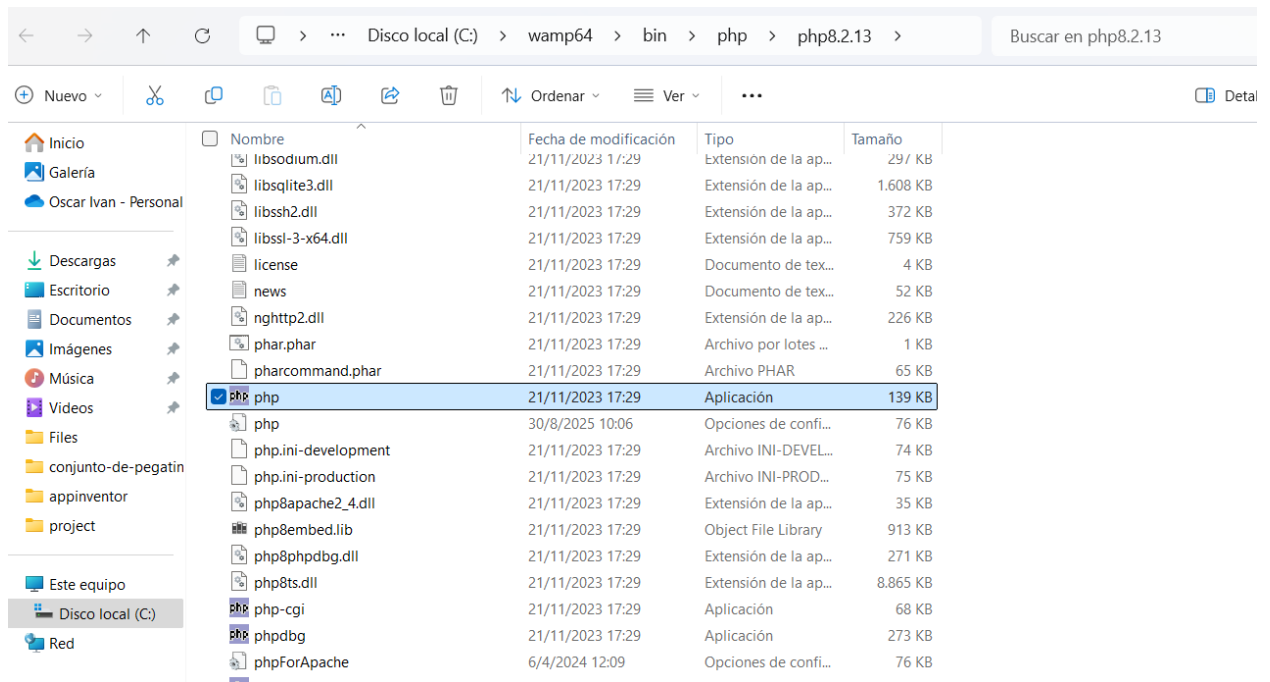
Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

Documentation/FAQs

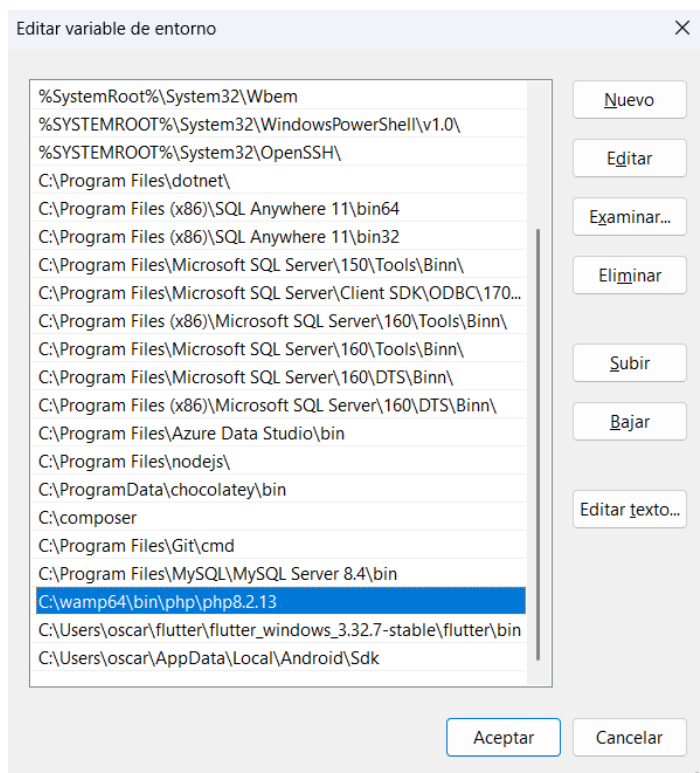
There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or Stack Overflow.

- [Linux FAQs](#)
- [Windows FAQs](#)
- [OS X FAQs](#)
- [OS X XAMPP-VM FAQs](#)

Descomprime en C:\php\8.2\.



Agrega C:\php\8.2\ al PATH del sistema.



Copia php.ini-development a php.ini y habilita extensiones recomendadas:

```
C:\Users\oscar>php -v
PHP 8.2.13 (cli) (built: Nov 21 2023 16:09:25) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.2.13, Copyright (c) Zend Technologies
    with Zend OPcache v8.2.13, Copyright (c), by Zend Technologies
    with Xdebug v3.2.2, Copyright (c) 2002-2023, by Derick Rethans
```

Instalar Composer

Descarga e instala Composer para Windows (instalador oficial).



[Hogar](#) | [Empezando](#) | [Descargar](#) | [Documentación](#) | [Explorar paquetes](#)

Descargar Composer Última versión: v2.8.11



PRIVATE PACKAGIST

Repositorio de paquetes seguro y confiable

Instalador de Windows

El instalador, que requiere que ya tenga PHP instalado, descargará Composer para usted y configurará su variable de entorno PATH para que pueda llamarlo simplemente `composer` desde cualquier directorio.

Descargue y ejecute [Composer-Setup.exe](#) : instalará la última versión de Composer cada vez que se ejecute.

[Haga clic aquí para descargar el instalador de Composer-Setup para Windows](#)

Instalación desde la línea de comandos

Para instalar rápidamente Composer en el directorio actual, ejecute el siguiente script en su terminal. Para automatizar la instalación, consulte [la guía sobre la instalación programática de Composer](#).

```
php -r "copiar('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "si (archivo_hash('sha384', 'composer-setup.php') === 'dac665fdc30fdd8ec78b38b980061b4150413ff2e3b6f885
php composer-setup.php
php -r "desvincular('composer-setup.php');"
```

```
C:\Users\oscar>composer -V
Composer version 2.7.6 2024-05-04 23:03:15
PHP version 8.2.13 (C:\wamp64\bin\php\php8.2.13\php.exe)
Run the "diagnose" command to get more detailed diagnostics output.
```

Instalar MySQL 8.4

Instala MySQL Server (elige contraseña del usuario root).

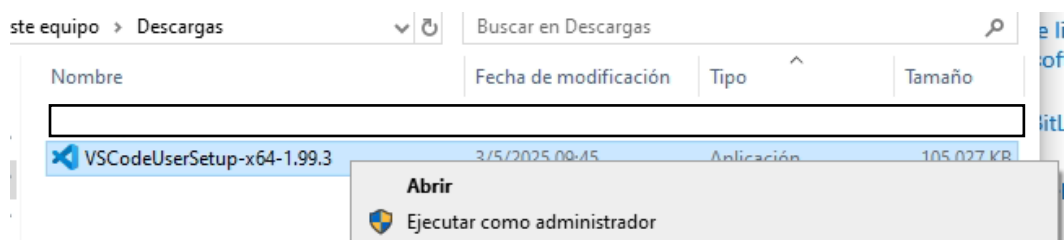
```
PS C:\Users\oscar> mysql -u root -p
```

Instalar dependencias PHP

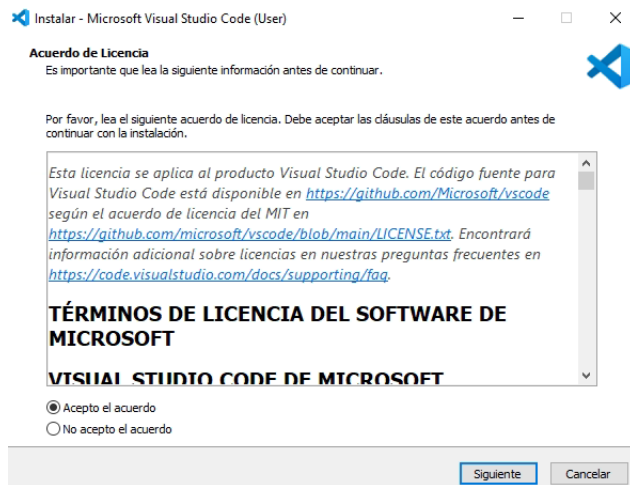
```
composer install
```

Instalación Visual Studio Code

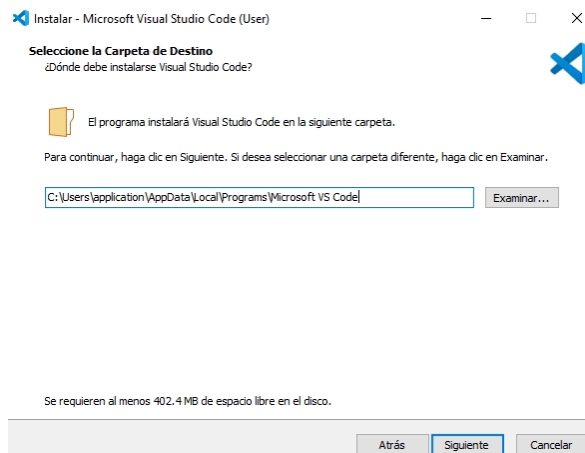
Visual Studio Code (VS Code) es un editor de código fuente ligero y potente, desarrollado por Microsoft, que funciona en múltiples plataformas (para el desarrollo de este sistema web usamos Windows), por lo cual realizamos los mismos pasos detallados en el apartado anterior, entonces procedemos a descarga *VS Code* desde el siguiente link <https://code.visualstudio.com/>, de la misma manera nos ubicamos en el directorio donde se descargó el programa y lo ejecutamos como administrador para que pueda realizar cambios en el sistema, adicional procedemos a seguir estos pasos que se detallan en las próximas ilustraciones:



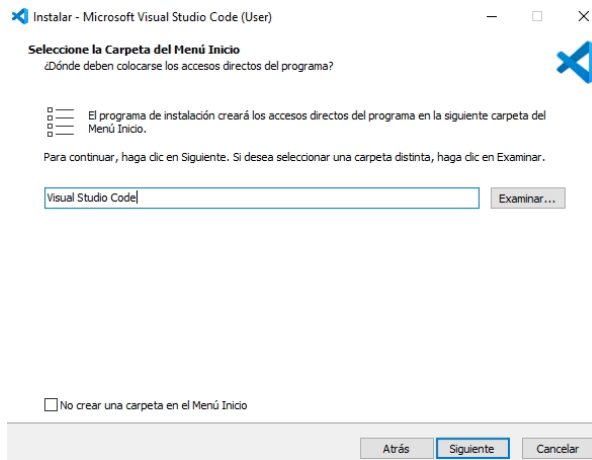
Directorio descargas.



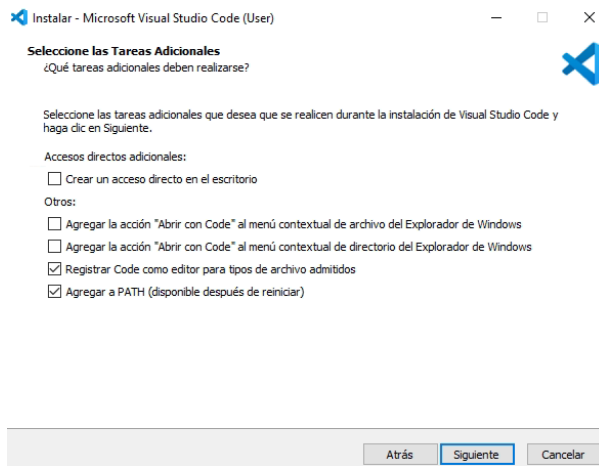
Aceptación de términos



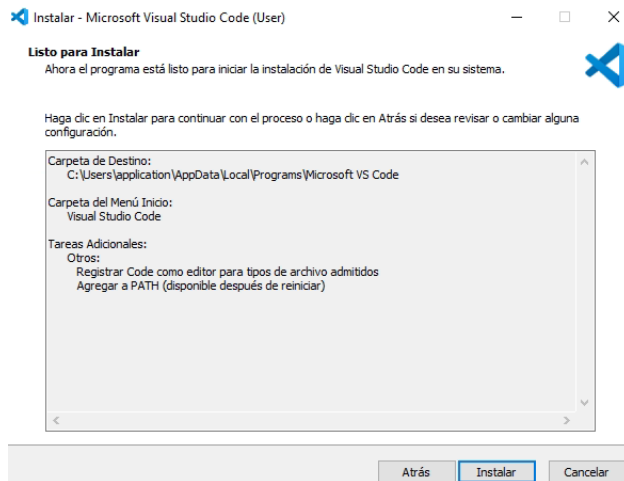
Directorio de instalación VSC.



Opciones predeterminadas.



. Opciones predeterminadas.



Inicio instalación VSC.

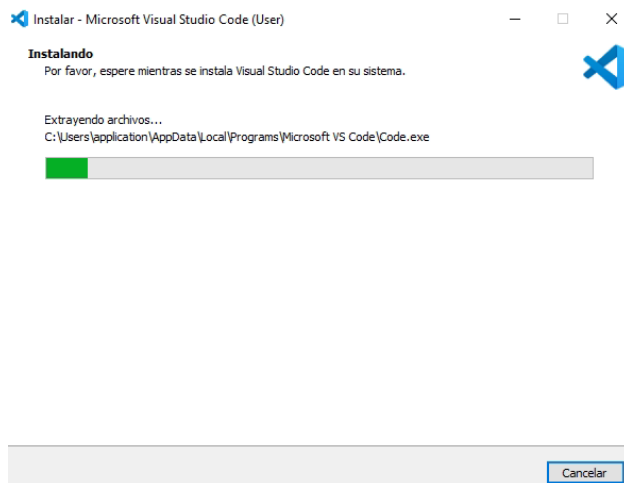
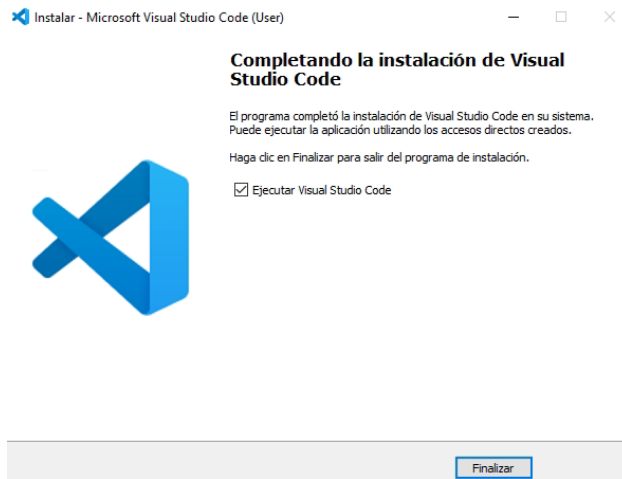


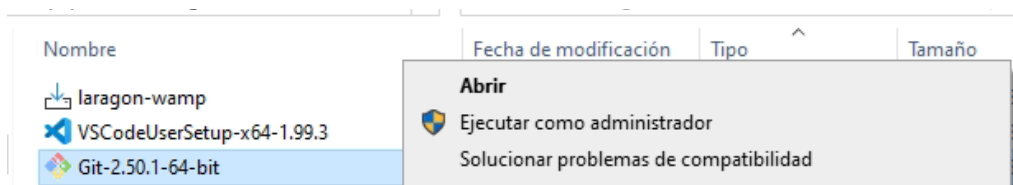
Ilustración 1. Progreso instalación VSC.



Instalación finalizada.

Instalación Git

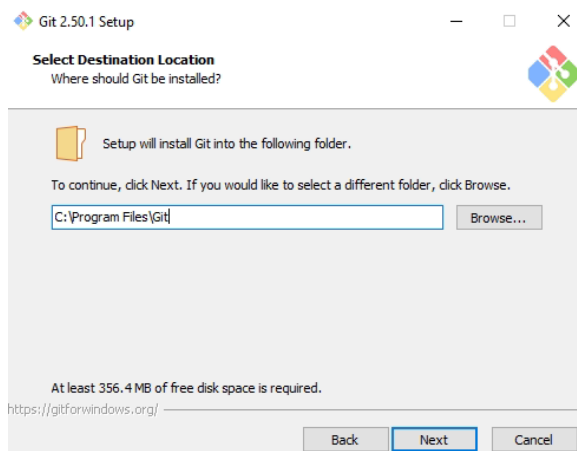
Git es un sistema de control de versiones distribuido, gratuito y de código abierto. Se utiliza para rastrear los cambios en los archivos de un proyecto y permite a los desarrolladores trabajar de forma colaborativa en el mismo proyecto de manera eficiente, la utilización de *Git* dentro del desarrollo de este sistema web se basó en el seguimiento y control de todas las versiones del mismo, en este apartado se instala *Git*, para poder descargar el proyecto que contiene todo el código fuente del sistema web para lo cual realizaremos la descarga del programa Git desde el siguiente link <https://git-scm.com/downloads/win> y posterior realizamos los siguientes pasos:



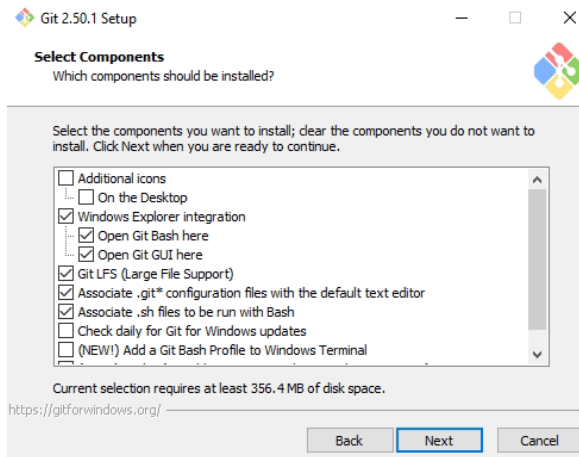
Directorio descargas.



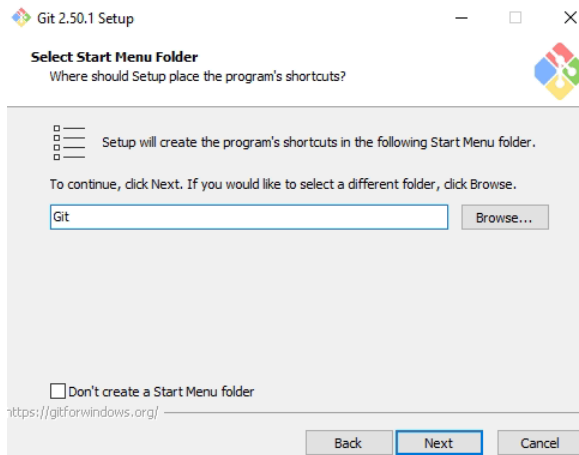
Opciones de instalación.



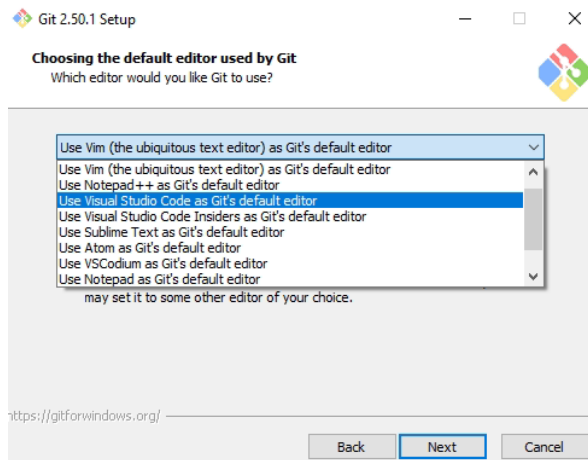
Directorio de instalación.



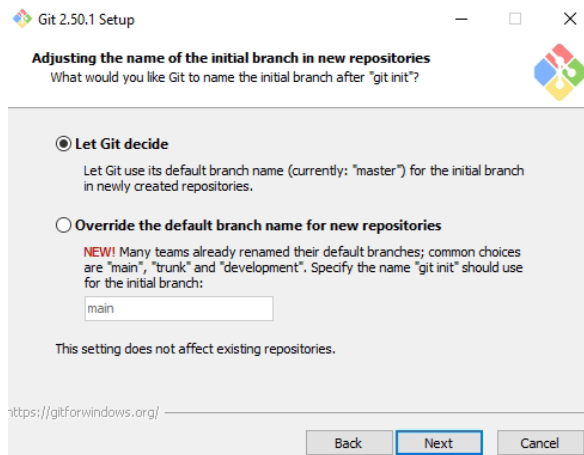
Opciones de instalación.



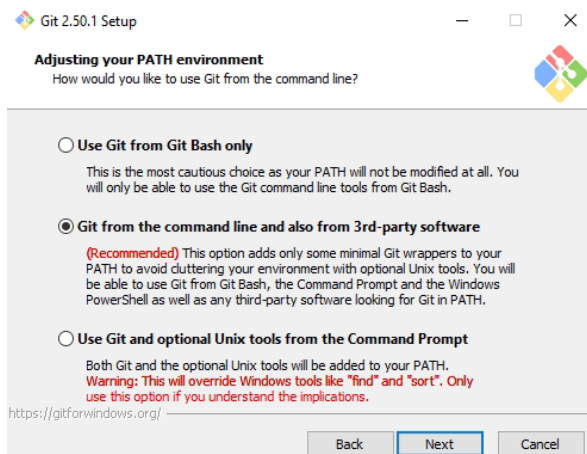
Opciones de instalación.



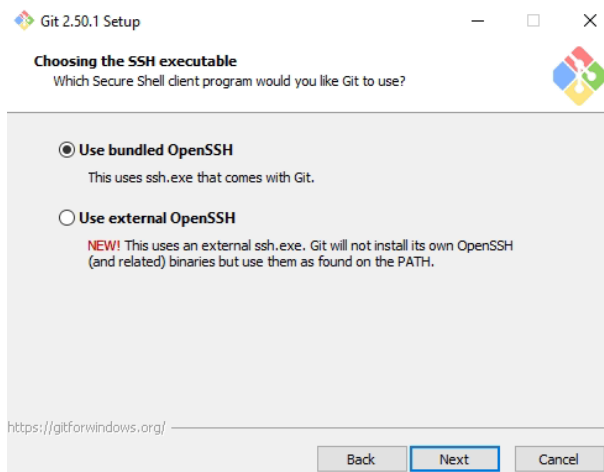
Selección de VSC como editor por defecto para Git.



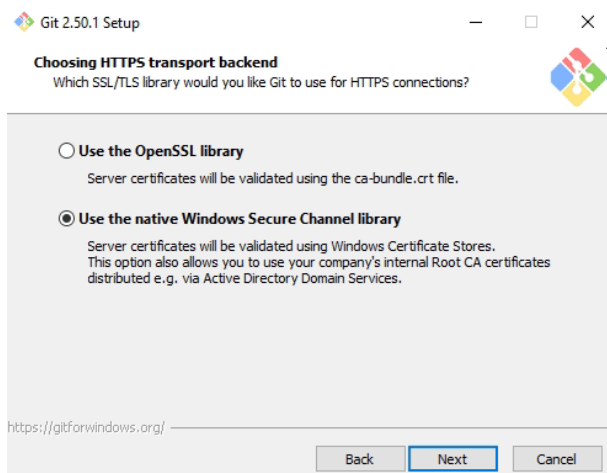
Opciones de instalación.



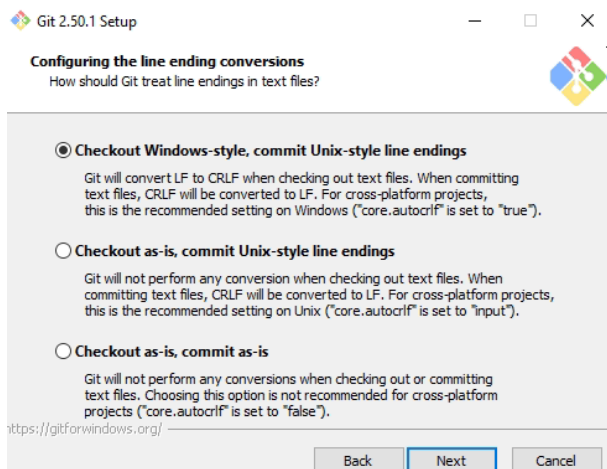
Opciones de instalación.



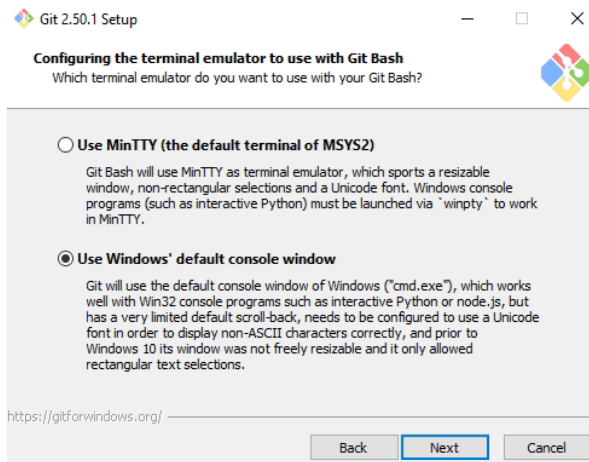
Opciones de instalación.



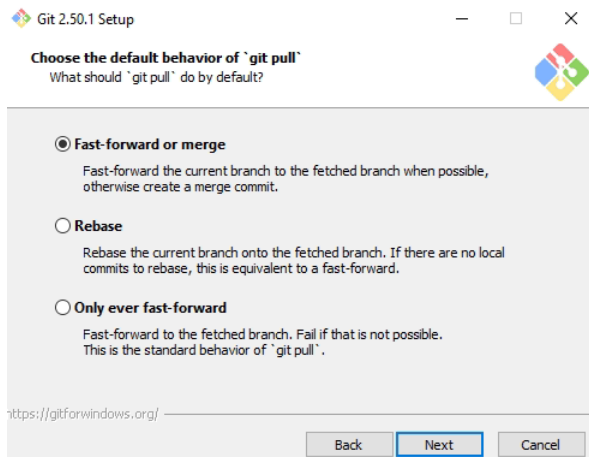
Opciones de instalación.



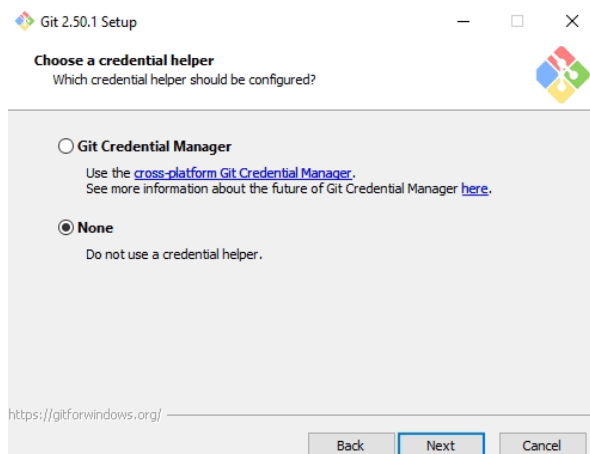
Opciones de instalación.



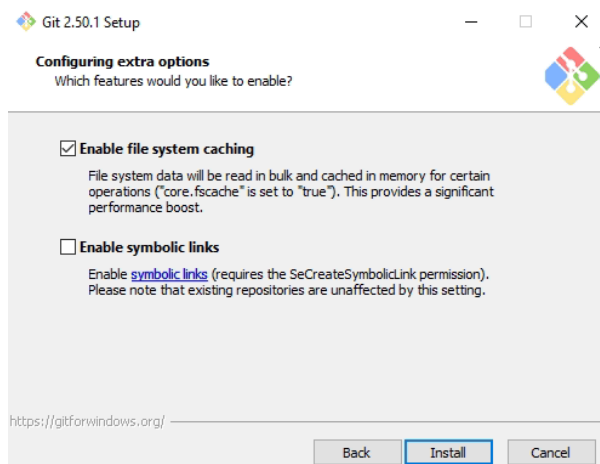
Símbolo del sistema como terminal por defecto.



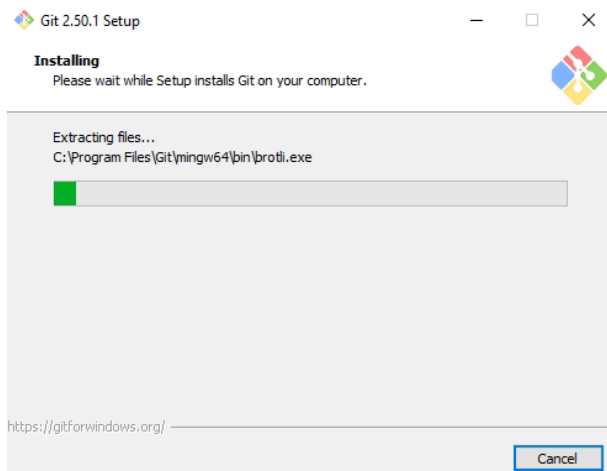
Opciones de instalación.



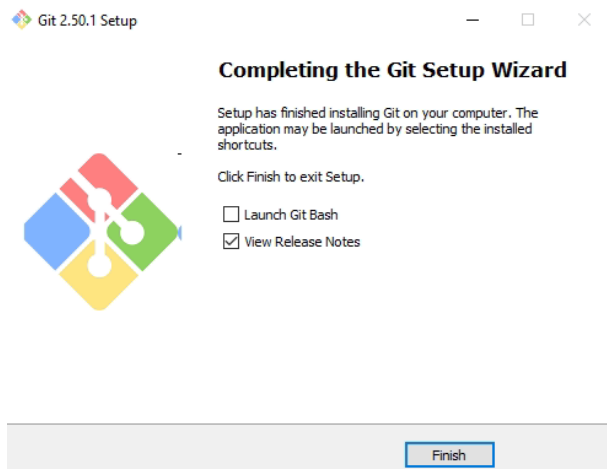
Opciones de instalación.



. Inicio instalación Git.



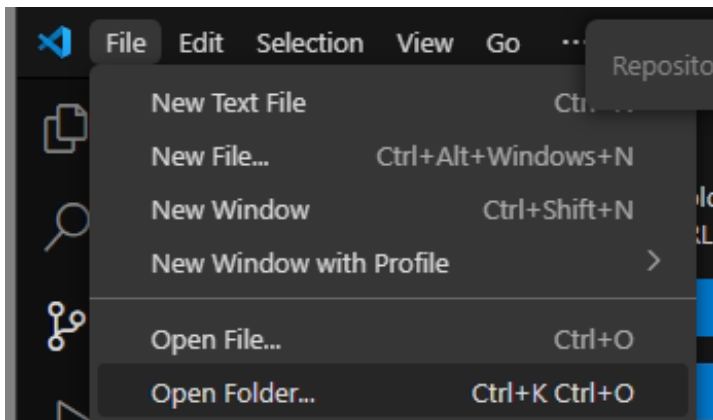
. Progreso instalación Git.



Finalización instalación Git.

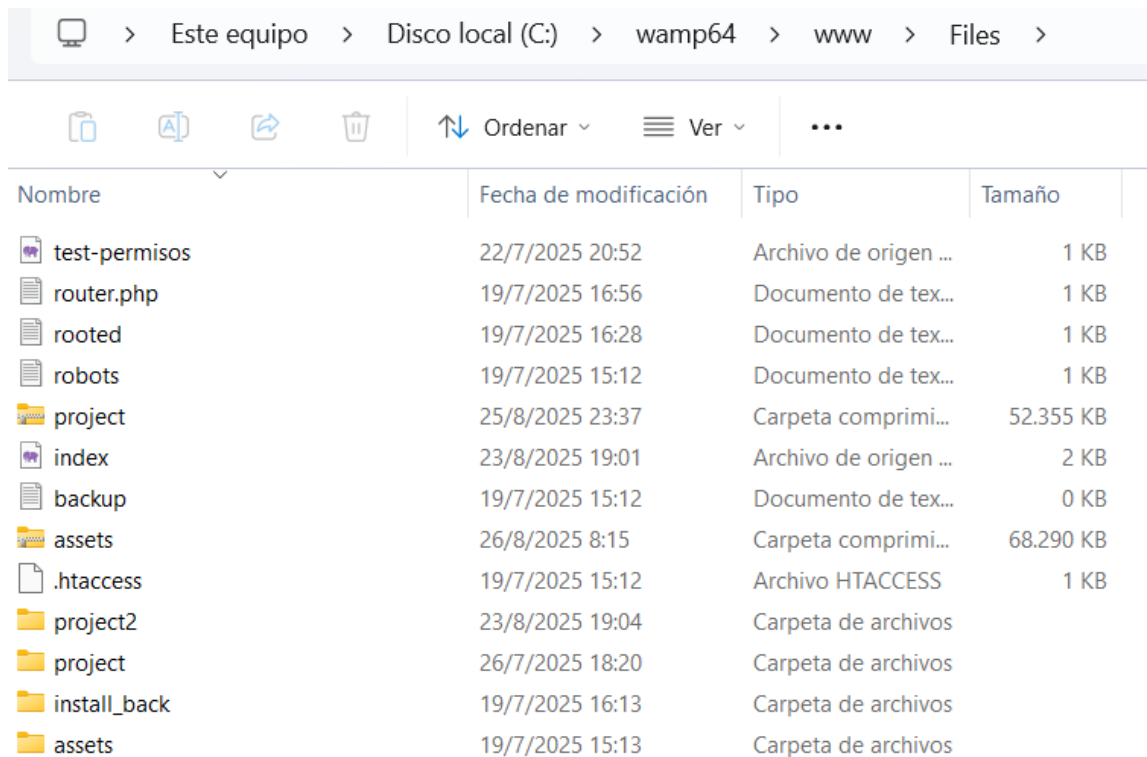
Abrimos VS Code

Nos dirigimos al menú File y seleccionamos Open Folder



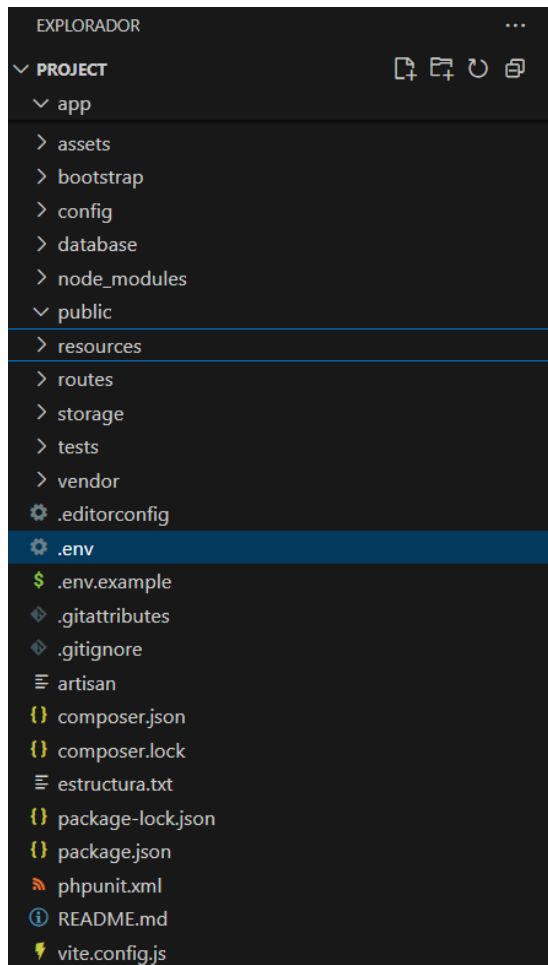
Abrir Proyecto.

Seleccionamos la carpeta donde se encuentra descargado el proyecto.



Seleccionar ubicación del proyecto.

Para saber que todo el proyecto se cargó correctamente en VSC debemos poder visualizar la estructura de archivos tal como se muestra en la siguiente ilustración:



Estructura de archivos del proyecto.

Creación del archivo de configuración

El archivo de configuración del proyecto es el que contiene todas las variables de entorno, este archivo es fundamental para que el sistema funcione correctamente, a continuación, se detallan los pasos para crear dicho archivo:

Crea el archivo.env en la raíz del proyecto

Ubícate en la raíz (nivel principal de tu carpeta de proyecto *muestrasmedicas-app*)

Haz clic derecho → Nuevo archivo.

Configurar .env

```
project > .env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:4Nc4BIaQAH38z2+sS7/56k0g3K0tZpc2p01JFFIN6EY=
4 APP_DEBUG=true
5 APP_URL=http://127.0.0.1:8000
6
7
8 LOG_CHANNEL=stack
9 LOG_DEPRECATIONS_CHANNEL=null
10 LOG_LEVEL=debug
11
12 DB_CONNECTION=mysql
13 DB_HOST=127.0.0.1
14 DB_PORT=3306
15 DB_DATABASE=importadoracaisatoa
16 DB_USERNAME=root
17 DB_PASSWORD=e2ndLaw
18
19
20 BROADCAST_DRIVER=log
21 CACHE_DRIVER=file
22 FILESYSTEM_DISK=local
23 QUEUE_CONNECTION=sync
24 SESSION_DRIVER=file
25 SESSION_LIFETIME=120
26
27 MEMCACHED_HOST=127.0.0.1
28
29 REDIS_HOST=127.0.0.1
30 REDIS_PASSWORD=null
31 REDIS_PORT=6379
32
33 MAIL_MAILER=smtp
34 MAIL_HOST=mailpit
35 MAIL_PORT=1025
36 MAIL_USERNAME=null
37 MAIL_PASSWORD=null
```

Genera la clave de la app:

```
php artisan key:generate
```

Crear base de datos vacía

En MySQL CLI:

```
CREATE DATABASE importadoracaisatoa CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

Migraciones

```
php artisan migrate  
php artisan db:seed
```

Ejecutar el servidor local

```
php artisan serve
```

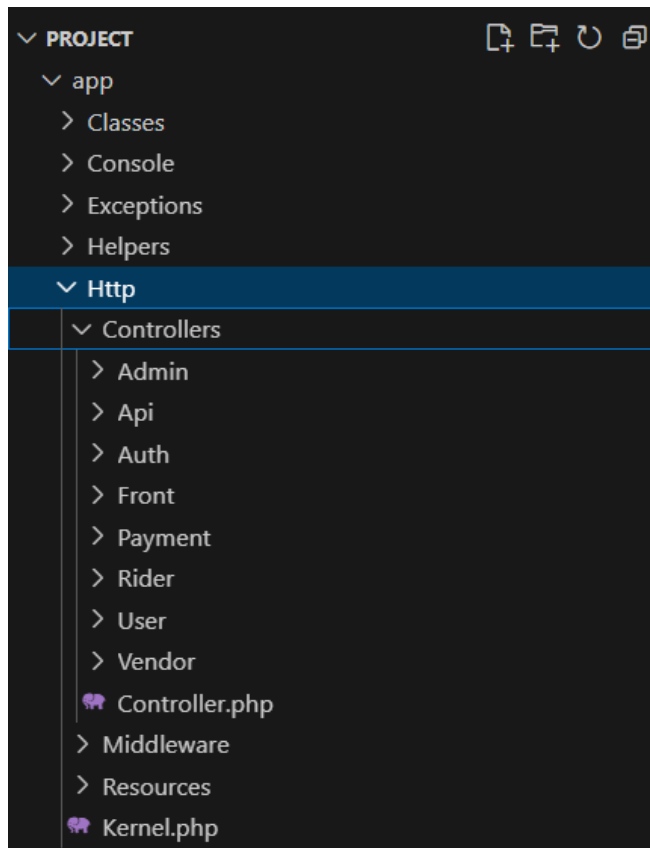
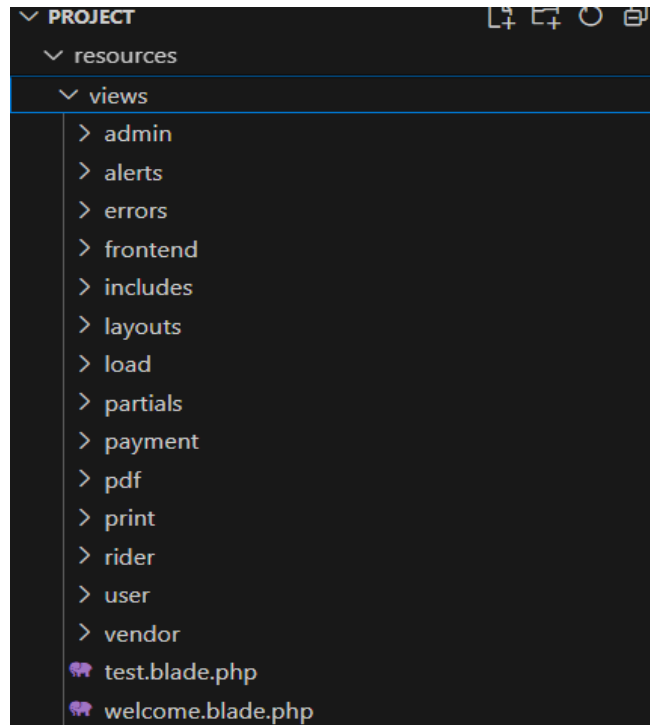
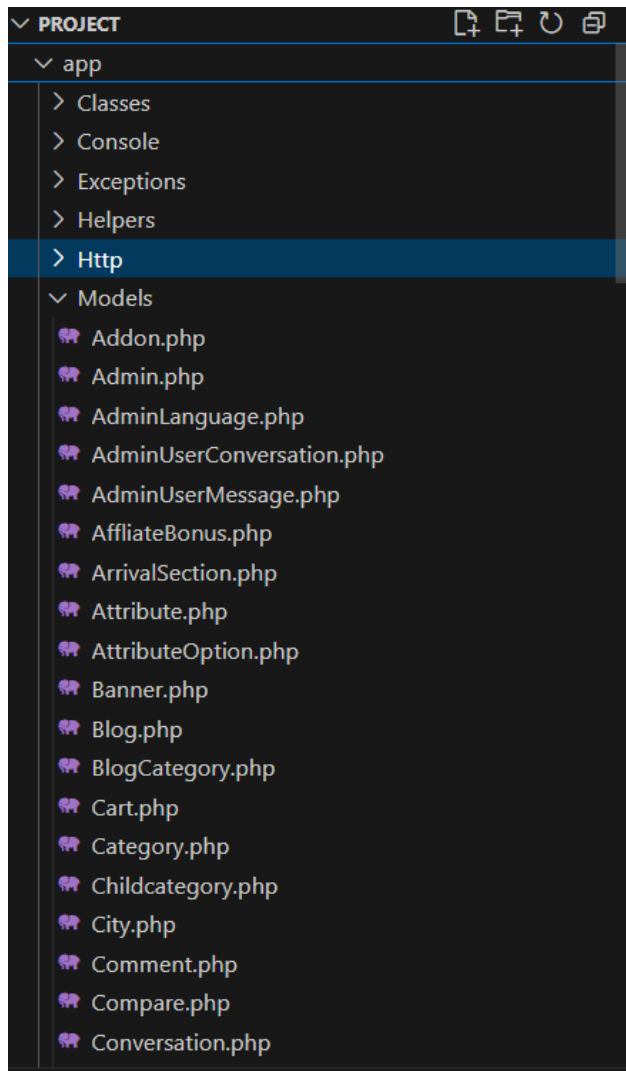
Abre <http://127.0.0.1:8000> en el navegador.

Estructura del proyecto (carpetas y roles)

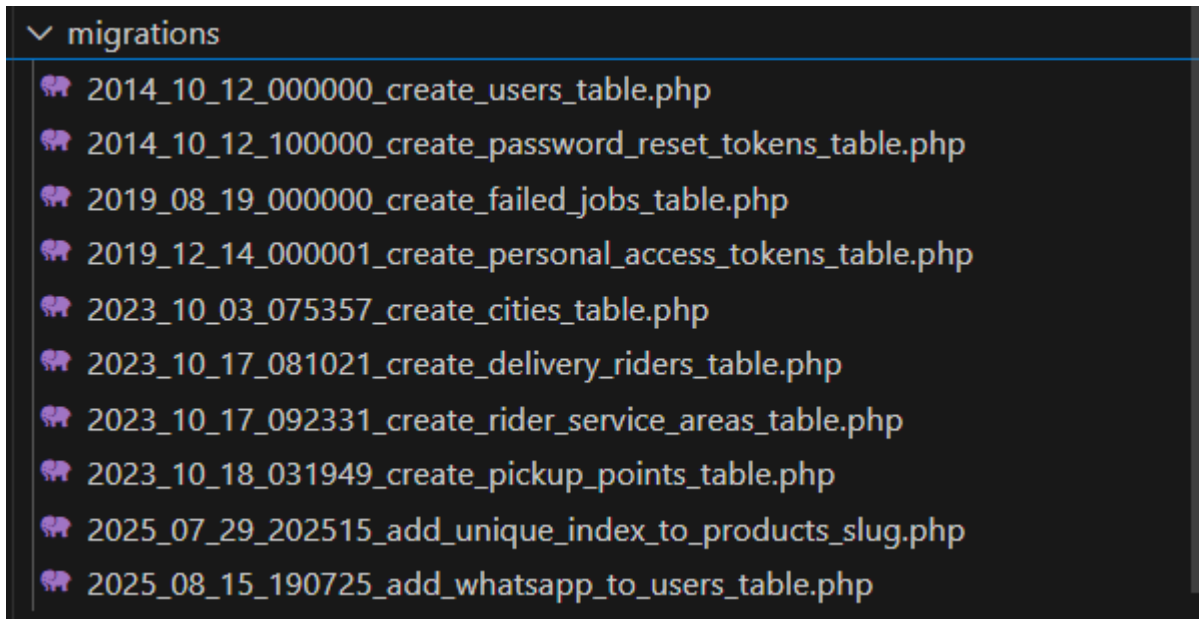
```

app/
  Http/
    Controllers/
      Admin/      # Backoffice (productos, pedidos, envíos, riders, etc.)
      Front/      # Flujo público (catálogo, carrito, checkout)
      Api/
        Front/    # Endpoints JSON de catálogo/búsqueda/checkout
        Auth/     # Autenticación por API (si la usas)
        Payment/  # Pasarelas (Stripe, PayPal, MercadoPago, Mollie, etc.)
    Middleware/   # Authenticate, Permissions, Vendor, SuperAdmin...
    Resources/    # API Resources (ProductResource, OrderResource, ...)
  Models/        # Product, Company, City, DeliveryRider, RiderServiceArea, PickupPoint...
  Helpers/       # PriceHelper, OrderHelper...
  Classes/       # GeniusMailer, Instamojo...
database/
  migrations/    # ver sección 5
  seeders/       # DatabaseSeeder y otros
resources/
  views/         # admin/*, frontend/*, dashboard/companies/* (módulo Proveedores)

```



Base de datos y migraciones

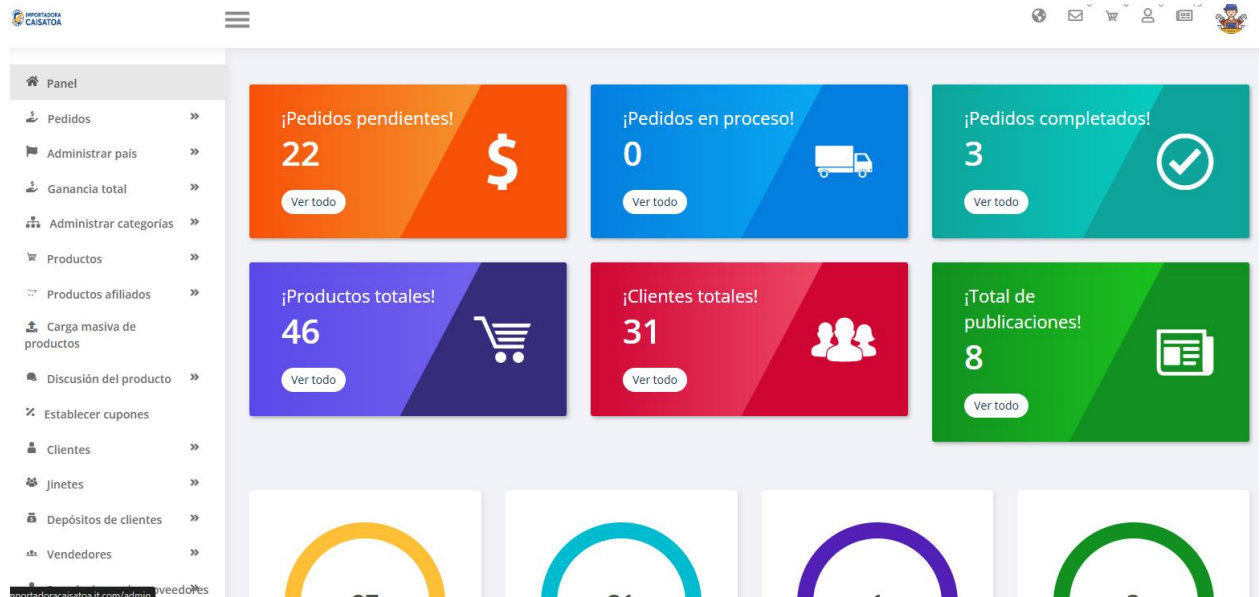


Si llegamos hasta este punto podemos deducir que la instalación del sistema web de control muestras médicas ha sido levantado correctamente.

Código fuente

Pantallas del panel de administración (vistas + controladores)

Dashboard (tarjetas KPI / gráfico 30 días / pedidos / clientes)



Vista: resources/views/admin/dashboard.blade.php

```

1  @extends('layouts.admin')
2
3  @section('contenido')
4  <div class="área-de-contenido">
5      @include('alertas.formulario-éxito')
6
7      @if($notificación_de_activación != "")
8          <div class="alerta alerta-peligro validación">
9              <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span
10                  aria-hidden="true">×</span></button>
11              <h3 class="text-center">{!!clean($activation_notify, array('Attr.EnableID' => true))
12              !!}</h3>
13

```

```

14 </div>
15 @endif
16
17 @if(Session::has('caché'))
18
19 <div class="alert alerta-éxito validación">
20     <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span
21         aria-hidden="true">×</span></button>
22     <h3 class="text-center">{{ Sesión::get("caché") }}</h3>
23 </div>
24
25 @endif
26
27 <div class="row row-cards-one">
28     <div class="col-md-12 col-lg-6 col-xl-4">
29         <div class="mycard bg1">
30             <div class="izquierda">
31                 <h5 class="title">{{ __('¡Pedidos pendientes!') }}</h5>
32                 <span class="number">{{ count($pending) }}</span>
33                 <a href="{{ route('admin-orders-all') }}"?status=pending" class="link">{{
34                 __('Ver todo') }}</a>
35             </div>
36             <div class="derecha d-flex align-self-center">
37                 <div class="icono">
38                     <i class="icofont-dollar"></i>
39                 </div>
40             </div>
41         </div>
42     </div>
43     <div class="col-md-12 col-lg-6 col-xl-4">
44         <div class="mycard bg2">
45             <div class="izquierda">
46                 <h5 class="title">{{ __('¡Pedidos en proceso!') }}</h5>
47                 <span class="number">{{ count($procesamiento) }}</span>
48                 <a href="{{ route('admin-orders-all') }}"?status=processing" class="link">{{
49                 __('Ver todo') }}</a>
50             </div>
51             <div class="derecha d-flex align-self-center">
52                 <div class="icono">
53                     <i class="icofont-truck-alt"></i>
54                 </div>
55             </div>
56         </div>
57     </div>

```

```

58     <div class="col-md-12 col-lg-6 col-xl-4">
59         <div class="mycard bg3">
60             <div class="izquierda">
61                 <h5 class="title">{{ __('¡Pedidos completados!') }}</h5>
62                 <span class="number">{{ count($completado) }}</span>
63                 <a href="{{ route('admin-orders-all') }}" ?status=completed" class="link">{{
64 __('Ver todo') }}</a>
65             </div>
66             <div class="derecha d-flex align-self-center">
67                 <div class="icono">
68                     <i class="icofont-check-circled"></i>
69                 </div>
70             </div>
71         </div>
72     </div>
73     <div class="col-md-12 col-lg-6 col-xl-4">
74         <div class="mycard bg4">
75             <div class="izquierda">
76                 <h5 class="title">{{ __('¡Total de productos!') }}</h5>
77                 <span class="number">{{ count($productos) }}</span>
78                 <a href="{{ route('admin-prod-index') }}" class="link">{{ __('Ver todo')
79 }}</a>
80             </div>
81             <div class="derecha d-flex align-self-center">
82                 <div class="icono">
83                     <i class="icofont-cart-alt"></i>
84                 </div>
85             </div>
86         </div>
87     </div>
88     <div class="col-md-12 col-lg-6 col-xl-4">
89         <div class="mycard bg5">
90             <div class="izquierda">
91                 <h5 class="title">{{ __('¡Total de clientes!') }}</h5>
92                 <span class="number">{{ count($usuarios) }}</span>
93                 <a href="{{ route('admin-user-index') }}" class="link">{{ __('Ver todo') }}</a>
94             </div>
95             <div class="derecha d-flex align-self-center">
96                 <div class="icono">
97                     <i class="icofont-users-alt-5"></i>
98                 </div>
99             </div>
100         </div>
101     </div>

```

```

102     <div class="col-md-12 col-lg-6 col-xl-4">
103         <div class="mycard bg6">
104             <div class="izquierda">
105                 <h5 class="title">{{ __('¡Total de publicaciones!') }}</h5>
106                 <span class="number">{{ count($blogs) }}</span>
107                 <a href="{{ route('admin-blog-index') }}" class="link">{{ __('Ver todo')
108 }}</a>
109             </div>
110             <div class="derecha d-flex align-self-center">
111                 <div class="icono">
112                     <i class="icofont-newspaper"></i>
113                 </div>
114             </div>
115         </div>
116     </div>
117
118 </div>
119
120 <div class="row row-cards-one">
121     <div class="col-md-6 col-xl-3">
122         <div class="área del cuadro de información de la tarjeta c">
123             <div class="c-info-box caja1">
124                 <p>{{ App\Models\User::where( 'created_at', ">", Carbon\Carbon::now()-
125 >subDays(30))->get()->count() }}</p>
126             </div>
127             <div class="c-info-box-content">
128                 <h6 class="title">{{ __('Nuevos clientes') }}</h6>
129                 <p class="text">{{ __('Últimos 30 días') }}</p>
130             </div>
131         </div>
132     </div>
133     <div class="col-md-6 col-xl-3">
134         <div class="área del cuadro de información de la tarjeta c">
135             <div class="c-info-box caja2">
136                 <p>{{ App\Models\Usuario::count() }}</p>
137             </div>
138             <div class="c-info-box-content">
139                 <h6 class="title">{{ __('Total de clientes') }}</h6>
140                 <p class="text">{{ __('Todo el tiempo') }}</p>
141             </div>
142         </div>
143     </div>
144     <div class="col-md-6 col-xl-3">
145         <div class="área del cuadro de información de la tarjeta c">

```

```

146         <div class="c-info-box box3">
147             <p>{{ App\Models\Order::where('estado','=', 'completado')->where(
148 'creado_en', ">", Carbon\Carbon::now()->subDays(30))->get()->count() }}</p>
149         </div>
150         <div class="c-info-box-content">
151             <h6 class="title">{{ __('Ventas totales') }}</h6>
152             <p class="text">{{ __('Últimos 30 días') }}</p>
153         </div>
154     </div>
155 </div>
156 <div class="col-md-6 col-xl-3">
157     <div class="área del cuadro de información de la tarjeta c">
158         <div class="c-info-box box4">
159             <p>{{ App\Models\Order::where('estado','=', 'completado')->obtener()-
160 >contar() }}</p>
161         </div>
162         <div class="c-info-box-content">
163             <h6 class="title">{{ __('Ventas totales') }}</h6>
164             <p class="text">{{ __('Todo el tiempo') }}</p>
165         </div>
166     </div>
167 </div>
168 </div>
169
170 <div class="row row-cards-one">
171
172     <div class="col-md-12 col-lg-6 col-sm-12 col-xl-6">
173         <div class="tarjeta">
174             <h5 class="card-header">{{ __('Pedido(s) reciente(s)') }}</h5>
175             <div class="cuerpo-de-la-tarjeta">
176
177                 <div class="table-responsive dashboard-home-table">
178                     <table id="popproducts" class="table table-hover dt-responsive"
179 cellspacing="0" width="100%">
180                         <cabeza>
181                             <tr>
182
183                                 <th>{{ __('Número de pedido') }}</th>
184                                 <th>{{ __('Fecha del pedido') }}</th>
185                             </tr>
186                             @foreach($rorders como $datos)
187                             <tr>
188                                 <td>{{ $data->número_de_pedido }}</td>
189                                 <td>{{ fecha('Ymd',strtotime($data->created_at)) }}</td>

```

```

190         <td>
191             <div class="lista-de-acciones"><a href="{{ route('admin-order-
192 show',$data->id) }}"><i
193                 class="fas fa-eye"></i> {{ __('Detalles') }}</a>
194             </div>
195         </td>
196     </tr>
197     @endforeach
198 </cabeza>
199 </tabla>
200 </div>
201
202 </div>
203 </div>
204
205 </div>
206
207 <div class="col-md-12 col-lg-6 col-sm-12 col-xl-6">
208     <div class="tarjeta">
209         <h5 class="card-header">{{ __('Clientes recientes') }}</h5>
210         <div class="cuerpo-de-la-tarjeta">
211
212             <div class="table-responsive dashboard-home-table">
213                 <table id="popproducts" class="table table-hover dt-responsive"
214 cellspacing="0" width="100%">
215                     <cabeza>
216                         <tr>
217                             <th>{{ __('Correo electrónico del cliente') }}</th>
218                             <th>{{ __('Unido') }}</th>
219                         </tr>
220                         @foreach($rusers como $datos)
221                         <tr>
222                             <td>{{ $data->correo electrónico }}</td>
223                             <td>{{ $data->creado_en }}</td>
224                             <td>
225                                 <div class="lista-de-acciones"><a href="{{ route('admin-
226 user-show',$data->id) }}"><i
227                                     class="fas fa-eye"></i> {{ __('Detalles') }}</a>
228                                 </div>
229                             </td>
230                         </tr>
231                         @endforeach
232                     </cabeza>
233                 </tabla>

```

```

234         </div>
235
236     </div>
237 </div>
238 </div>
239 </div>
240
241 <div class="row row-cards-one">
242
243     <div class="col-md-12 col-lg-12 col-sm-12 col-xl-12">
244         <div class="tarjeta">
245             <h5 class="card-header">{{ __('Producto(s) popular(es)) }}</h5>
246             <div class="cuerpo-de-la-tarjeta">
247
248                 <div class="table-responsive dashboard-home-table">
249                     <table id="popproducts" class="table table-hover dt-responsive"
250 cellspacing="0" width="100%">
251                         <cabeza>
252                             <tr>
253                                 <th>{{ __('Imagen destacada') }}</th>
254                                 <th>{{ __('Nombre') }}</th>
255                                 <th>{{ __('Categoría') }}</th>
256                                 <th>{{ __('Tipo') }}</th>
257                                 <th>{{ __('Precio') }}</th>
258                                 <th></th>
259
260                             </tr>
261                         </cabeza>
262                         <cuerpo>
263                             @foreach($popproducts como $datos)
264                                 <tr>
265                                     <td></td>
268                                     <td>{{ mb_strlen(strip_tags($datos->nombre),'UTF-8') > 50 ?
269 mb_substr(strip_tags($datos->nombre),0,50,'UTF-8').'...' : strip_tags($datos->nombre)
270 }}</td>
271                                     <td>{{ $data->categoría->nombre }}
272                                     @if(isset($datos->subcategoría))
273                                     <br>
274                                     {{ $data->subcategoría->nombre }}
275                                     @endif
276                                     @if(isset($datos->categoríahija))
277                                     <br>

```



```

322         </tr>
323     </cabeza>
324     <cuerpo>
325         @foreach($pproducts como $datos)
326             <tr>
327                 <td></td>
330                 <td>{{ mb_strlen(strip_tags($datos->nombre),'UTF-8') >
331 50 ? mb_substr(strip_tags($datos->nombre),0,50,'UTF-8').'...' : strip_tags($datos->nombre)
332 }} </td>
333                 <td>{{ $data->categoria->nombre }}
334                 @if(isset($datos->subcategoria))
335                     <br>
336                     {{ $data->subcategoria->nombre }}
337                 @endif
338                 @if(isset($datos->categoriahija))
339                     <br>
340                     {{ $data->categoriainfantil->nombre }}
341                 @endif
342             </td>
343             <td>{{ $datos->tipo }} </td>
344             <td>{{ $datos->mostrarPrecio() }} </td>
345             <td>
346                 <div class="lista-de-acciones"><a href="{{ {
347 route('admin-prod-edit',$data->id) }}"><i
348                     clase="fas fa-eye"></i> {{ __('Detalles')
349 }} </a>
350             </div>
351         </td>
352     </tr>
353 @endforeach
354 </tbody>
355 </tabla>
356 </div>
357
358 </div>
359 </div>
360
361 </div>
362
363 </div>
364
365 <div class="row row-cards-one">

```

```

366
367 <div class="col-md-12 col-lg-12 col-sm-12 col-xl-12">
368   <div class="tarjeta">
369     <h5 class="card-header">{{ __('Ventas totales en los últimos 30 días') }}</h5>
370     <div class="cuerpo-de-la-tarjeta">
371
372       <canvas id="gráfico de líneas"></canvas>
373
374     </div>
375   </div>
376
377 </div>
378
379 </div>
380
381
382
383
384 <div class="row row-cards-one">
385
386   <div class="col-md-12 col-sm-12 col-lg-6 col-xl-6">
387     <div class="tarjeta">
388       <h5 class="card-header">{{ __('Principales referencias') }}</h5>
389       <div class="cuerpo-de-la-tarjeta">
390         Tarjeta de corrección de altura de administrador
391         <div id="chartContainer-topReference"></div>
392       </div>
393
394     </div>
395   </div>
396
397 </div>
398
399 <div class="col-md-12 col-lg-6 col-sm-12 col-xl-6">
400   <div class="tarjeta">
401     <h5 class="card-header">{{ __('Sistema operativo más usado') }}</h5>
402     <div class="cuerpo-de-la-tarjeta">
403       Tarjeta de corrección de altura de administrador
404       <div id="chartContainer-os"></div>
405     </div>
406   </div>
407 </div>
408 </div>
409

```

```

410 </div>
411
412
413
414 </div>
415
416 @endsection
417
418 @section('scripts')
419
420 <script type="texto/javascript">
421
422     (función($) {
423
424
425         estricto";
426
427         mostrarLineChart();
428
429         función displayLineChart() {
430             var datos = {
431                 etiquetas: [
432                     {!!$días!!}
433                 ],
434                 conjuntos de datos: [{
435                     Etiqueta: "Prime y Fibonacci",
436                     color de relleno: "#3dbcff",
437                     color del trazo: "#0099ff",
438                     puntoColor: "rgba(220,220,220,1)",
439                     colorDeTrazoDePunto: "#fff",
440                     puntoRellenoResaltado: "#fff",
441                     puntoResaltadoTrazo: "rgba(220,220,220,1)",
442                     datos: [
443                         {!!$ventas!!}
444                     ]
445                 }]
446             };
447             var ctx = document.getElementById("lineChart").getContext("2d");
448             var opciones = {
449                 responsivo: verdadero
450             };
451             var lineChart = new Chart(ctx).Line(datos, opciones);
452         }
453

```

"uso

```

454 $('#popproducts').dataTable( {
455     "ordenando": falso,
456     'lengthChange': falso,
457     'buscando': falso,
458     'ordenar': falso,
459     'info' : falso,
460     'autoWidth' : falso,
461     'responsive': verdadero,
462     'paginación': falso
463 } );
464
465 $('#ppproducts').dataTable( {
466     "ordenando": falso,
467     'lengthChange': falso,
468     'buscando': falso,
469     'ordenar': falso,
470     'info' : falso,
471     'autoWidth' : falso,
472     'responsive': verdadero,
473     'paginación': falso
474 } );
475
476 var chart1 = nuevo CanvasJS.Chart("chartContainer-topReference",
477     {
478         exportEnabled: verdadero,
479         animaciónHabilitada: verdadero,
480
481         leyenda: {
482             cursor: "puntero",
483             horizontalAlign: "derecha",
484             verticalAlign: "centro",
485             Tamaño de fuente: 16,
486             relleno: {
487                 arriba: 20,
488                 abajo: 2,
489                 derecha: 20,
490             },
491         },
492         datos: [
493             {
494                 tipo: "pastel",
495                 showInLegend: verdadero,
496                 Texto de leyenda: "",
497                 toolTipContent: "{name}: <strong>{#percent%}</strong> (#percent%)</strong>",

```

```

498         Etiqueta de índice: "#porcentaje%",
499         indexLabelFontColor: "blanco",
500         indexLabelPlacement: "dentro",
501         Puntos de datos: [
502             @foreach($referencias como $navegador)
503                 {y:{{ $browser->total_count}}, nombre: "{{ $browser->referral}}"},
504             @endforeach
505         ]
506     }
507 ]
508 });
509 gráfico1.render();
510
511 var chart = nuevo CanvasJS.Chart("chartContainer-os",
512     {
513         exportEnabled: verdadero,
514         animaciónHabilitada: verdadero,
515         leyenda: {
516             cursor: "puntero",
517             horizontalAlign: "derecha",
518             verticalAlign: "centro",
519             Tamaño de fuente: 16,
520             relleno: {
521                 arriba: 20,
522                 abajo: 2,
523                 derecha: 20,
524             },
525         },
526     },
527     {
528         datos: [
529             {
530                 tipo: "pastel",
531                 showInLegend: verdadero,
532                 Texto de leyenda: "",
533                 toolTipContent: "{name}: <strong>{#percent%} ({#percent%})</strong>",
534                 Etiqueta de índice: "#porcentaje%",
535                 indexLabelFontColor: "blanco",
536                 indexLabelPlacement: "dentro",
537                 Puntos de datos: [
538                     @foreach($browsers como $browser)
539                         {y:{{ $browser->total_count}}, nombre: "{{ $browser->referral}}"},
540                     @endforeach
541                 ]
542             }
543         ]
544     }
545 ]

```

```

    });
    gráfico.render();

})(jQuery);

</script>

@endsection

```

Vista: resources/views/admin/dashboard.blade.php

```

1  <?php
2
3  espacio de nombres App\Http\Controllers\Admin;
4
5  utilizar App\Models\Blog;
6  utilizar App\Models\Counter;
7  utilizar App\Models\Order;
8  utilizar App\Models\Product;
9  utilizar App\Models\User;
10 utilizar Auth;
11 utilizar Illuminate\Http\Request;
12
13 utilice Illuminate\Support\Facades\Hash;
14 utilice InvalidArgumentException;
15 utilice Validator;
16
17 La clase DashboardController extiende AdminBaseController
18 {
19
20     índice de función pública ()
21     {
22
23         $data [ 'pendiente' ] = Pedido :: where ( 'estado' , '=' , 'pendiente' ) -> obtener ();
24         $data [ 'procesando' ] = Pedido :: where ( 'estado' , '=' , 'procesando' ) -> obtener ();
25         $data [ 'completado' ] = Pedido :: where ( 'estado' , '=' , 'completado' ) -> obtener ();
26         $data [ 'días' ] = "" ;
27         $data [ 'ventas' ] = "" ;

```

```

28     for ( $i = 0 ; $i < 30 ; $i ++ ) {
29         $data [ 'días' ] .= "" .date ( "d M" , strtotime ( '-' . $i.'días ' ) ) . ", " ;
30
31         $data [ 'ventas' ] .= "" .Orden:: donde ( ' estado ' , '=' , ' completado' ) ->
32 dondeFecha ( 'creado_en' , '=' , fecha ( "Ymd" , strtotime ( '-' . $ i.'días ' ))) -> contar () . "" ,
33 ;
34     }
35     $data [ 'usuarios' ] = Usuario :: todos ();
36     $data [ 'productos' ] = Producto :: todos ();
37     $data [ 'blogs' ] = Blog :: todos ();
38     $data [ 'pproductos' ] = Producto :: último ( 'id' ) -> tomar ( 5 ) -> obtener ();
39     $data [ 'rpedidos' ] = Pedido :: último ( 'id' ) -> tomar ( 5 ) -> obtener ();
40     $data [ 'poproductos' ] = Producto :: último ( 'vistas' ) -> tomar ( 5 ) -> obtener ();
41     $data [ 'usuarios' ] = Usuario :: último ( 'id' ) -> tomar ( 5 ) -> obtener ();
42     $data [ 'referencias' ] = Contador :: donde ( 'tipo' , 'referencia' ) -> último (
43 'conteo_total' ) -> tomar ( 5 ) -> obtener ();
44     $data [ 'navegadores' ] = Contador :: donde ( 'tipo' , 'navegador' ) -> último (
45 'recuento_total' ) -> tomar ( 5 ) -> obtener ();
46
47     $data [ 'activation_notify' ] = "" ;
48     if ( file_exists (public_path() . '/rooted.txt' ) ) {
49         $rooted = file_get_contents (public_path() . '/rooted.txt' );
50         if ( $rooted < date ( 'Ymd' , strtotime ( "+10 días" ) ) ) {
51             $data [ 'activation_notify' ] = "<i class='icofont-warning-alt icofont-
52 4x'></i><br>Por favor, active su sistema.<br>Si no activa su sistema ahora, estará inactivo
53 el " . $rooted . "!!<br> <a href='" . url( '/admin/activation' ) . "' class='btn btn-success'>
54 Activar ahora</a>" ;
55         }
56     }
57
58     devolver vista( 'admin.dashboard' , $data );
59 }
60
61 perfil de función pública ()
62 {
63     $data = Auth :: guard ( 'admin' ) -> usuario ();
64     return view( 'admin.profile' , compact ( 'data' ) );
65 }
66
67 función pública actualización de perfil (Solicitud $solicitud )
68 {
69     //--- Sección de validación
70
71     $reglas =

```

```

72     [
73     'foto' => 'mimes:jpeg,jpg,png,svg' ,
74     'correo electrónico' => 'único:admins,correo electrónico' . Aut :: guard ( 'admin' )
75 -> usuario () -> id .
76     ];
77
78     $validador = Validador :: make ( $solicitud -> all () ), $reglas );
79
80     si ( $validador -> falla () ) {
81         devolver respuesta() -> json ( matriz ( 'errores' => $validador -> getMessageBag
82 () -> toArray () ));
83     }
84     ///--- Finaliza la sección de validación
85     $input = $request -> all ();
86     $data = Auth :: guard ( 'admin' ) -> user ();
87     if ( $file = $request -> file ( 'photo' ) ) {
88         $name = \PriceHelper :: ImageCreateName ( $file );
89         $file -> move ( 'assets/images/admins/' , $name );
90         if ( $data -> photo != null ) {
91             if ( file_exists ( public_path() . '/assets/images/admins/' . $data -> photo ) ) {
92                 unlink ( public_path() . '/assets/images/admins/' . $data -> photo );
93             }
94         }
95         $input [ 'foto' ] = $nombre ;
96     }
97     $data -> update ( $input );
98     $msg = __( 'Su perfil se actualizó correctamente' );
99     return response() -> json ( $msg );
100 }
101
102 función pública passwordreset ()
103 {
104     $data = Auth :: guard ( 'admin' ) -> usuario ();
105     return view( 'admin.password' , compact ( 'data' ) );
106 }
107
108 función pública changepass (Solicitud $solicitud )
109 {
110     $admin = Auth :: guard ( 'admin' ) -> usuario ();
111     if ( $solicitud -> cpass ) {
112         if ( Hash :: check ( $solicitud -> cpass , $admin -> contraseña ) ) {
113             if ( $solicitud -> newpass == $solicitud -> renovarpass ) {
114                 $input [ 'contraseña' ] = Hash :: make ( $solicitud -> newpass );
115             } else {

```

```

116         return response() -> json ( array ( 'errores' => [ 0 => __( 'Confirmar
117 contraseña no coincide.' ) ] ) );
118     }
119 } else {
120     return response() -> json ( array ( 'errores' => [ 0 => __( 'La contraseña actual
121 no coincide.' ) ] ) );
122 }
123 }
124 $admin -> update ( $input );
125 $msg = __( 'Su contraseña se cambió correctamente' );
126 return response() -> json ( $msg );
127 }
128
129 función pública generate_bkup ()
130 {
131     $bkuplink = "";
132     $chk = file_get_contents ( 'copia de seguridad.txt' );
133     si ( $chk != "" ) {
134         $bkuplink = url( $chk );
135     }
136     devolver vista( 'admin.movetoserver' , compact ( 'bkuplink' , 'chk' ) );
137 }
138
139 función pública clear_bkup ()
140 {
141     $destino = ruta_pública() . '/instalar' ;
142     $bkuplink = "";
143     $chk = file_get_contents ( 'backup.txt' );
144     si ( $chk != "" ) {
145         desvincular ( ruta_pública( $chk ) );
146     }
147
148     si ( is_dir ( $destino ) ) {
149         $este -> deleteDir ( $destino );
150     }
151     $handle = fopen ( 'backup.txt' , 'w+' );
152     fwrite ( $handle , "" );
153     fclose ( $handle );
154     //devuelve "No se generó ningún archivo de respaldo.";
155     return redirect() -> back () -> with ( 'success' , '¡Archivo de respaldo eliminado
156 exitosamente!' );
157 }
158
159 activación de función pública ()

```

```

160 {
161     $datos_de_activación = "" ;
162     if ( file_exists (public_path() . '/project/license.txt' ) ) {
163         $license = file_get_contents (public_path() . '/project/license.txt' );
164         if ( $license != "" ) {
165             $datos_de_activación = "<i style='color:darkgreen;' class='icofont-check-
166 circled icofont-4x'></i><br><h3 style='color:darkgreen;'>¡Su sistema está
167 activado!</h3><br> Su clave de licencia: <b>" . $license . "</b>" ;
168         }
169     }
170     retorno vista( 'admin.activation' , compact ( 'activation_data' ));
171 }
172
173 función pública activación_enviar (Solicitud $solicitud )
174 {
175     //devolver config('services.genius.ocean');
176     $código_de_compra = $solicitud -> pcode ;
177     $mi_script = 'eCommerce Genius' ;
178     $mi_dominio = url( '/' );
179
180     $varUrl = str_replace ( ' ' , '%20' , config( 'services.genius.ocean' ) .
181 'purchase112662activate.php?code=' . $purchase_code . '&domain=' . $my_domain .
182 '&script=' . $my_script );
183
184     si ( ini_get ( 'allow_url_fopen' ) ) {
185         $contenido = archivo_obtener_contenido ( $varUrl );
186     } else {
187         $ch = curl_init ();
188         curl_setopt ( $ch , CURLOPT_URL , $varUrl );
189         curl_setopt ( $ch , CURLOPT_RETURNTRANSFER , 1 );
190         $contenido = curl_exec ( $ch );
191         curl_close ( $ch );
192     }
193
194     $chk = json_decode ( $contenido , verdadero );
195
196     si ( $chk [ 'estado' ] != "éxito" ) {
197
198         $msg = $chk [ 'mensaje' ] == null ? __( 'Código de compra inválido.' ) : $chk [
199 'mensaje' ];
200
201         devolver respuesta() -> json ( $msg );
202         //devolver redirección()->atrás()->con('sin éxito',$chk['mensaje']);
203

```

```

204 } de lo contrario {
205     $this -> setUp ( $chk [ 'p2' ], $chk [ 'lData' ] );
206
207     si ( archivo_existe ( ruta_pública() . '/rooted.txt' ) ) {
208         desvincular ( ruta_pública() . '/rooted.txt' );
209     }
210
211     $fpbt = fopen ( ruta_pública() . '/proyecto/licencia.txt' , 'w' );
212     fwrite ( $fpbt , $código_de_compra );
213     fclose ( $fpbt );
214
215     $msg = '¡Felicitaciones! Su sistema se activó correctamente.' ;
216     return response() -> json ( $msg );
217     //return redirect('admin/dashboard')->with('success','¡Felicitaciones! Su sistema se
218 activó correctamente. ');
219 }
220 //devuelve config('services.genius.ocean');
221 }
222
223 función pública setUp ( $mtFile , $goFileData )
224 {
225     $fpa = fopen ( public_path() . $mtFile , 'w' );
226     fwrite ( $fpa , $goFileData );
227     fclose ( $fpa );
228 }
229
230 función pública movescript ()
231 {
232     ini_set ( 'tiempo_máximo_de_ejecución' , 3000 );
233
234     $destino = ruta_pública() . '/instalar' ;
235     $chk = archivo_obtener_contenido ( 'copia_de_seguridad.txt' );
236     si ( $chk != "" ) {
237         desvincular ( ruta_pública( $chk ) );
238     }
239
240     si ( is_dir ( $destino ) ) {
241         $este -> deleteDir ( $destino );
242     }
243
244     $origen = ruta_base() . '/proveedor/actualizar' ;
245     $este -> recurse_copy ( $origen , $destino );
246     $archivos = ruta_pública();
247

```

```

248     $nombre_recuperación = 'Genius-tienda-Por-GeniusOcean-' . fecha ( 'Ymd' ) . '.zip'
249 ;
250
251     $zip = new \ZipArchive();
252     $zip -> abrir ( $bkupname , \ZipArchive :: CREAR | \ZipArchive :: SOBREScribir
253 );
254
255     $archivos = nuevo \RecursiveIteratorIterator(
256         nuevo \RecursiveDirectoryIterator(base_path()),
257         \IteradorRecursivoIterador :: SOLO HOJAS
258     );
259     foreach ( $archivos como $nombre => $archivo ) {
260         if ( ! $archivo -> isDir () ) {
261             $rutaDeArchivo = $archivo -> getRealPath ();
262             $rutaRelativa = substr ( $rutaDeArchivo , strlen (ruta_base()) + 1 );
263             $zip -> addFile ( $rutaDeArchivo , $rutaRelativa );
264         }
265     }
266
267     $zip -> cerrar ();
268
269     $identificador = fopen ( 'backup.txt' , 'w+' );
270     fwrite ( $identificador , $backupname );
271     fclose ( $identificador );
272
273     si ( is_dir ( $destino )) {
274         $este -> deleteDir ( $destino );
275     }
276     devolver respuesta() -> json ( [ 'estado' => 'éxito' , 'archivo de respaldo' => url(
277 $bkupname ) , 'nombre de archivo' => $bkupname ] , 200 );
278 }
279
280 función pública recurse_copy ( $src , $dst )
281 {
282     $dir = opendir ( $origen );
283     @ mkdir ( $dst );
284     mientras ( falso !== ( $archivo = readdir ( $dir ))) {
285         si (( $archivo != '.' ) && ( $archivo != '..' )) {
286             si ( es_dir ( $origen . '/' . $archivo )) {
287                 $esto -> recurse_copy ( $origen . '/' . $archivo , $dst . '/' . $archivo );
288             } else {
289                 copy ( $src . '/' . $archivo , $dst . '/' . $archivo );
290             }
291         }

```

```

292     }
        closedir ( $dir );
    }

función pública deleteDir ( $dirPath )
{
    if ( ! is_dir ( $dirPath ) ) {
        throw new InvalidArgumentException( " $dirPath debe ser un directorio" );
    }
    si ( substr ( $dirPath , strlen ( $dirPath ) - 1 , 1 ) != '/' ) {
        $dirPath .= '/';
    }
    $archivos = glob ( $dirPath . '*' , GLOB_MARK );
    foreach ( $archivos como $archivo ) {
        if ( is_dir ( $archivo ) ) {
            yo mismo :: deleteDir ( $archivo );
        } else {
            desvincular ( $archivo );
        }
    }
    rmdir ( $rutaDirección );
}
}

```

Pedidos (listado / detalle / impresión / POS)

Todos los pedidos

Panel > Pedidos > Todos los pedidos

Espectáculo 10  entradas

Buscar:

[+ Agregar un pedido](#)

Correo electrónico del cliente	Número de orden	Cantidad total	Costo total	Opciones
oscar2@gmail.com	ohKf1756425468	4	37,40\$	Comportamiento ▼
oscar2@gmail.com	Kb5p1756422616	1	37,40\$	Comportamiento ▼
oscar2@gmail.com	c57q1756411838	1	29\$	Comportamiento ▼
oscar2@gmail.com	FJkK1756411828	1	29\$	Comportamiento ▼
kkkkkk@gmail.com	kv4O1756248256	2	35\$	Comportamiento ▼
oscar2@gmail.com	0lUv1756232047	1	329\$	Comportamiento ▼
oscar2@gmail.com	dQvY1756231756	1	22,50\$	Comportamiento ▼
oscar2@gmail.com	h4AB1756231067	1	33\$	Comportamiento ▼
oscar2@gmail.com	6Aux1756230949	3	186,40\$	Comportamiento ▼
oscar2@gmail.com	eVrc1756230124	2	43,40\$	Comportamiento ▼

Vistas:

Listado: resources/views/admin/order/index.blade.php

Detalles del pedido [← Atrás](#)

Panel > Pedidos > Detalles del pedido

Detalles del pedido

No pagado

ID del pedido : ohKf1756425468

Producto total : 4

Costo de embalaje : 1\$

Costo total : 37,40\$

Fecha de pedido : 28 de agosto de 2025, 23:57:48

Método de pago : Pago contra reembolso

Estado del pago :

[Ver factura](#)

Detalle: resources/views/admin/order/details.blade.php

Controlador: app/Http/Controllers/Admin/OrderController.php.

Categorías principales

Panel > Administrar categorías > Categorías principales

Espectáculo
entradas

Buscar:

+ Agregar nuevo

Nombre	Babosa	Atributos	Presentado	Estado	Opciones
Educación y Oficios	Educación y Oficios	Crear	Activado	Desactivado	Editar Eliminar
Salud y bienestar	Salud y Bienestar	Crear	Desactivado	Desactivado	Editar Eliminar
Servicios Locales	Servicios Locales	Crear	Activado	Activado	Editar Eliminar
Artesanía	Obras hechas a mano	Crear	Activado	Activado	Editar Eliminar
Ropa cocida a mano	para-hombre-o-mujer	Crear	Activado	Activado	Editar Eliminar
Aire libre, recreación	Recreación y ejercicio al aire libre	Crear Administrar	Desactivado	Desactivado	Editar Eliminar
Bolsos hechos a mano	bolsas de mano	Crear	Desactivado	Desactivado	Editar Eliminar
Bolsos hechos a mano	Decoración del hogar y electrodomésticos	Crear Administrar	Activado	Activado	Editar Eliminar

Productos

Panel > Productos > Todos los productos

Espectáculo 10 entradas

Buscar:

+ Agregar producto

Nombre	Tipo	Existencias	Precio	Estado	Opciones
Caja de Madera Tallada "Cóndor Andino" – Hecha a m... ID: 00000407	Físico	Ilimitado	29,90\$	Activado	Editar Borrar
ID de rombo : 00000405	Físico	Ilimitado	10\$	Activado	Editar Borrar
Bicicleta ID: 00000404	Físico	Ilimitado	20\$	Activado	Editar Borrar
Vestido Largo Halter Isabella ID: 00000403	Físico	Ilimitado	10\$	Activado	Editar Borrar
Televisor TCL ID: 00000402	Físico	Ilimitado	900\$	Activado	Editar Borrar
Aretes Gato Tiffany Entrelazos ID: 00000401	Físico	10	15\$	Activado	Editar Borrar
Identificación del TCL QM8K : 00000400	Físico	Ilimitado	500\$	Activado	Editar Borrar

Vistas: resources/views/admin/category/*.blade.php y

resources/views/admin/product/*.blade.php

Controladores: app/Http/Controllers/Admin/CategoryController.php,

.../ProductController.php.

Cupones

Cupones

Panel > Cupones

Espectáculo 10 entradas

Buscar:

+ Agregar nuevo

Código	Tipo	Cantidad	Usado	Estado	Opciones
procoo	Descuento por porcentaje	10%	0	Activado	Editar Borrar
proco	Descuento por importe	10\$	0	Activado	Editar Borrar

Vistas: resources/views/admin/coupon/*.blade.php

Controlador: app/Http/Controllers/Admin/CouponController.php.

Cientes / Transacciones / Depósitos / Retiros

Cientes

Panel > Cientes

Espectáculo

10

 entradas

Buscar:

+ Agregar cliente

Nombre	Correo electrónico	Opciones
Óscar Calsatoa	oscarcalsatoa221@gmail.com	\$ Administrar depósito Hacer proveedor Detalles Editar Enviar Desatascar
Óscar Calsatoa	oscarcalsatoa22@gmail.com	\$ Administrar depósito Hacer proveedor Detalles Editar Enviar Desatascar
Óscar Iván Calsatoa Suntaxi	jose.l.alvarez21@gmail.com	\$ Administrar depósito Proveedor Detalles Editar Enviar Desatascar

Retiros

Panel > Cientes > Retiros

Espectáculo entradas

Buscar:

Correo electrónico	Teléfono	Cantidad	Método	Fecha de retiro	Estado	Comportamiento
No hay datos disponibles en la tabla						

Mostrando 0 a 0 de 0 entradas

Anterior

Próximo

Vistas: resources/views/admin/deposit/*.blade.php,
resources/views/admin/earning/*.blade.php

Controladores: app/Http/Controllers/Admin/UserController.php,
.../UserTransactionController.php, .../UserDepositController.php.

Riders y Áreas de servicio

Jinetes

Panel > Jinetes

Espectáculo

10

entradas

Buscar:

Nombre	Teléfono	Correo electrónico	Entrega total	Opciones
Óscar Iván Calsatoa Suntaxi	0995548007	jinete@gmail.com	0	Enviar Desatascar Detalles

Mostrando 1 a 1 de 1 entradas

Anterior
1
Próximo

Retiros

Panel > Clientes > Retiros

Espectáculo

10

entradas

Buscar:

Correo electrónico	Teléfono	Cantidad	Método	Fecha de retiro	Estado	Comportamiento
No hay datos disponibles en la tabla						

Mostrando 0 a 0 de 0 entradas

Anterior
Próximo

Controladores: `app/Http/Controllers/Admin/RiderController.php`,
`app/Http/Controllers/Admin/CityController.php` (gestión de zonas/ciudades).

Envíos / Embalaje

Metodos de envio

Panel > Configuración general > Métodos de envío

Envío múltiple: Activado

Espectáculo entradas

Buscar:

Título	Precio	Opciones
Envío gratis	0\$	Editar Eliminar
Envío exprés	10\$	Editar Eliminar
Servicios médicos de emergencia	26\$	Editar Eliminar
Básico	5\$	Editar Eliminar
Estándar	10\$	Editar Eliminar
Pro	15\$	Editar Eliminar
prueba	40\$	Editar Eliminar
Entrega en local	0\$	Editar Eliminar
Repartidor	0\$	Editar Eliminar

Vistas: resources/views/admin/shipping/*.blade.php

Controlador: app/Http/Controllers/Admin/ShippingController.php.

Pagos

Panel > Configuración de pago > Pasarelas de pago

Espectáculo entradas

Buscar:

[+ Agregar nuevo pago](#)

Nombre	Detalles	Verificar	Depósito	Suscripción	Opciones
SSLCommerz	Pagar a través de SSLCommerz.	No mostrado	Presentado	Presentado	Editar
Dinero móvil	Número de pago: 69234324233423	No mostrado	Presentado	Presentado	Editar Eliminar
Pago contra reembolso	Pagar en efectivo al momento de la entrega.	Presentado	No disponible	No disponible	Editar

Información de pago

Panel > Configuración de pago > Información de pago

Información básica	Información del proveedor
Pago como invitado	Activado ▼
Formato de moneda *	Después del precio ▼
Separador decimal *	Punto(.) ▼
Separador de miles *	Coma(,) ▼
Tarifa de retiro *	10
Cargo por retiro (%) *	5

Vista: resources/views/admin/payment/*.blade.php

Controladores de pasarelas: app/Http/Controllers/Payment

Apariencia / Home / Blog / Páginas

⚙ Configuración general ⌵

Logo

Favicon

Cargador

Métodos de envío

Embalajes

Lugares de recogida

Contenido del sitio web

Personalización de la página de inicio

Panel > Configuración de la página de inicio > Personalización de la página de inicio

Control deslizante *



Productos destacados *



Mejor valorado, grandes ahorros y tendencia *



Producto más vendido *



Boletín informativo *



Sección de Llegada *



Oferta del día *



Pareja *



Blogs *



Entregar

Configuración del blog

Panel > Blog > Configuración del blog

Disqus

Activado ▾

Nombre corto del sitio web de Disqus *

importadora

Mostrar publicaciones por página *

6

Ahorrar

Vistas: resources/views/admin/pagesetting/*.blade.php, resources/views/admin/blog/*,

resources/views/admin/page/*

Controladores: app/Http/Controllers/Admin/PageSettingController.php, .../BlogController.php, .../PageController.php.

Modo mantenimiento / Perfil admin

Mantenimiento de sitios web

Panel > Configuración general > Mantenimiento de sitios web

Mantenimiento de sitios web

Desactivado ▼

Texto de mantenimiento *
(En cualquier idioma)

B I U [iconos de alineación] Tamaño de [dropdown] Familia de [dropdown] Formato de [dropdown]

[iconos de formato]

¡EN MANTENIMIENTO!

Ahorrar

Vistas: resources/views/admin/generalsetting/*.blade.php,

resources/views/admin/profile.blade.php

Controladores: app/Http/Controllers/Admin/GeneralSettingController.php.

Acceso / Autenticación (lo de “Mi cuenta”, login, recuperar)

Editar perfil



Subir

Seleccionar país ▼

Seleccione Estado ▼

Seleccione Estado ▼

AMAGUANA CUENDINA CALLE VENCEDORES Y 24 DE JULIO

Tratamiento...

Regístrate ahora

Registro

Olvidé mi contraseña

Dirección de correo electrónico *

Introduzca su dirección de correo electrónico

Se enviará una contraseña a tu dirección de correo electrónico.

Entregar

- **Controladores de login/registro (usuario, admin, rider)**

app/Http/Controllers/Auth/Admin/LoginController.php

app/Http/Controllers/Auth/User/LoginController.php

app/Http/Controllers/Auth/Rider/LoginController.php (y sus Forgot/Register).

- **Vistas de login admin**

resources/views/admin/login.blade.php, resources/views/admin/forgot.blade.php.