18-9-2025

EQUIPO 6

“SI FUNCIONA NO LO TOQUES”

**DOCUMENTACIÓN**

PaDuki

Contenido

[1. Declaración del problema 3](#_Toc209061854)

[2. Impacto y beneficios de la solución propuesta al negocio 4](#_Toc209061855)

[3. Descripción de la solución técnica 5](#_Toc209061856)

[3.1 Componentes del sistema 5](#_Toc209061857)

[4. Detalles de implementación técnica 6](#_Toc209061858)

[4.1 Requerimientos funcionales y no funcionales 6](#_Toc209061859)

[4.2 Justificación de tecnologías utilizadas 6](#_Toc209061860)

[4.3 Consideraciones de seguridad (hardening) 7](#_Toc209061861)

[5. Prototipado 7](#_Toc209061862)

[5.1 Prototipo inicial 7](#_Toc209061863)

[5.2 Prototipo funcional completo 8](#_Toc209061864)

[6. Código fuente y script de base de datos 9](#_Toc209061865)

[6.1 Backend (BE) 9](#_Toc209061866)

[6.2 Frontend (FE) 9](#_Toc209061867)

[6.3 Scripts de base de datos (BD) 9](#_Toc209061868)

[7. Documentación adicional 11](#_Toc209061869)

[7.1 Documentación técnica 11](#_Toc209061870)

[7.2 Manuales de usuario 18](#_Toc209061871)

[7.3 Diccionario de datos 23](#_Toc209061872)

[8. Conclusiones 24](#_Toc209061873)

Índice de Ilustraciones

[Ilustración 1 Initial prototyping design Fuente: Elaboración propia. 8](#_Toc209061723)

[Ilustración 2 complete functional prototype 9](#_Toc209061724)

[Ilustración 3 System Architecture Diagram 13](#_Toc209061725)

[Ilustración 4 backend flow 15](#_Toc209061726)

[Ilustración 5 main page 16](#_Toc209061727)

[Ilustración 6 Cart page 17](#_Toc209061728)

[Ilustración 7 admin interface flow 18](#_Toc209061729)

# Declaración del problema

Desarrollar una plataforma global de compraventa de productos artesanales que elimine las barreras de pago y comisiones entre países y facilite la experiencia de usuario. La solución permitirá adquirir productos artesanales con pocos clics y con una interfaz intuitiva.

# Impacto y beneficios de la solución propuesta al negocio

Beneficios para el negocio

* Ingreso recurrente y predecible por cada transacción (modelo de comisión).
* Ventaja tecnológica al implementar pagos interoperables con ILP y Open Payments, lo que pocas plataformas usan aún.
* Fidelización de usuarios gracias a una experiencia de compra fluida y justa.
* Imagen socialmente responsable al apoyar la inclusión financiera y la preservación cultural, lo cual fortalece la marca.

**Impacto en el negocio**

* Rentabilidad desde el primer año: con la comisión del 5% se genera flujo constante de ingresos, alcanzando sostenibilidad sin depender de subsidios.
* Escalabilidad: el modelo crece con el número de compradores y artesanos sin que los costos aumenten al mismo ritmo.
* Posicionamientocompetitivo: al ofrecer menores comisiones que Amazon Handmade o Etsy, la plataforma se vuelve más atractiva para artesanos y compradores.
* Acceso a un mercado global: abre nuevas oportunidades de negocio en distintos países, expandiendo el alcance cultural y económico.

# Descripción de la solución técnica

El sistema propuesto se centra en optimizar los pagos internacionales y reducir comisiones en la compraventa de productos artesanales mediante la integración del protocolo Interledger (ILP) y el estándar Open Payments.

# Componentes del sistema

1. Arquitectura de pagos interoperables

* Interledger Protocol (ILP): permite enrutar transacciones dividiéndolas en micro paquetes que viajan de nodo en nodo hasta el destino, garantizando seguridad, trazabilidad y entrega completa.
* Open Payments: provee un marco estandarizado para iniciar, validar y liquidar los pagos sin exponer datos bancarios de los usuarios.

1. Flujo de la transacción

* El comprador realiza el pago en su moneda local a través de su wallet digital.
* El Administrador (intermediario) recibe y valida la orden de compra, asegurando que el producto esté correctamente gestionado.
* Una vez confirmada la entrega, el pago es liberado hacia el wallet del artesano mediante ILP, con conversión automática de divisas y confirmación en tiempo real.

1. Seguridad y transparencia

* Todas las transacciones viajan cifradas a través del protocolo ILP.
* El sistema garantiza que los artesanos reciben el monto íntegro menos una comisión reducida (5%), lo que mejora su margen frente a plataformas tradicionales.
* El administrador solo actúa como validador, sin retener innecesariamente fondos, lo que fortalece la confianza en la plataforma.

1. Infraestructura técnica

* Frontend: desarrollado en HTML, CSS y JavaScript, con interfaz optimizada para la navegación y compra rápida.
* Backend: construido en Node.js, encargado de gestionar autenticación, inventario, transacciones y conexión con los protocolos de pago.
* Base de datos: MySQL/PostgreSQL para almacenamiento seguro de usuarios, productos y órdenes.

Intercambio de datos: JSON como formato ligero de comunicación entre frontend, backend y sistemas externos.

# Detalles de implementación técnica

# Requerimientos funcionales y no funcionales

**Funcionales:**

* Procesamiento de pagos internacionales mediante ILP + Open Payments.
* Consultasde ventas, ingresos y transacciones.

**No funcionales:**

* Seguridad: cifrado de transacciones y protección de datos personales.
* Escalabilidad: capacidad de soportar miles de transacciones simultáneas con mínima latencia.
* Compatibilidad: integración con múltiples wallets.
* Usabilidad: interfaz intuitiva y adaptable.

# Justificación de tecnologías utilizadas

* Frontend (HTML, CSS, JavaScript):

Se eligieron por su compatibilidad universal con navegadores, bajo costo de desarrollo y facilidad de mantenimiento. Además, permiten una interfaz ligera y adaptable, escalable a nuevas funcionalidades sin altos costos de licencias.

* Backend (Node.js):

Escogido por su alto rendimiento en aplicaciones de tiempo real y su ecosistema de librerías (NPM) que facilitan la integración con APIs externas, como las de pagos internacionales. Comparado con otros frameworks, Node.js ofrece mejor escalabilidad y menor consumo de recursos.

* Base de datos (MySQL/PostgreSQL):

Se seleccionaron por ser open source, con gran comunidad de soporte y capacidad para manejar grandes volúmenes de datos de manera estructurada. En comparación con bases de datos propietarias (ej. SQL Server u Oracle), representan una reducción considerable en costos sin sacrificar rendimiento ni seguridad.

* Intercambio de datos (JSON):

JSON fue elegido por su ligereza en la comunicación entre frontend y backend, y su compatibilidad nativa con JavaScript y APIs modernas. Esto mejora la eficiencia y la velocidad de respuesta frente a formatos más pesados como XML.

* Protocolos de pago (Interledger + Open Payments):

Seleccionados porque eliminan la dependencia de intermediarios financieros, reducen costos de transacción y habilitan la interoperabilidad global de pagos en tiempo real, lo que ninguna pasarela tradicional (PayPal, Western Union, bancos) garantiza en este nivel.

# Consideraciones de seguridad (hardening)

Para garantizar la protección de usuarios, datos y transacciones, se adoptan las siguientes medidas:

* Cifrado de datos sensibles (contraseñas, información de pagos) mediante algoritmos de hash seguros (ej. bcrypt, SHA-256).
* Protección contra ataques comunes (SQL Injection) mediante validación estricta de entradas y sanitización de datos.

# Prototipado

# Prototipo inicial

El primer prototipo corresponde a un wireframe que muestra la estructura general del catálogo de productos.

* Cada producto tiene su imagen, nombre, precio y un botón para agregar al carrito.
* El usuario puede acceder en cualquier momento a su carrito mediante el botón “View Cart”.
* Este diseño buscó validar la usabilidad y la disposición de elementos antes de implementar la versión final.

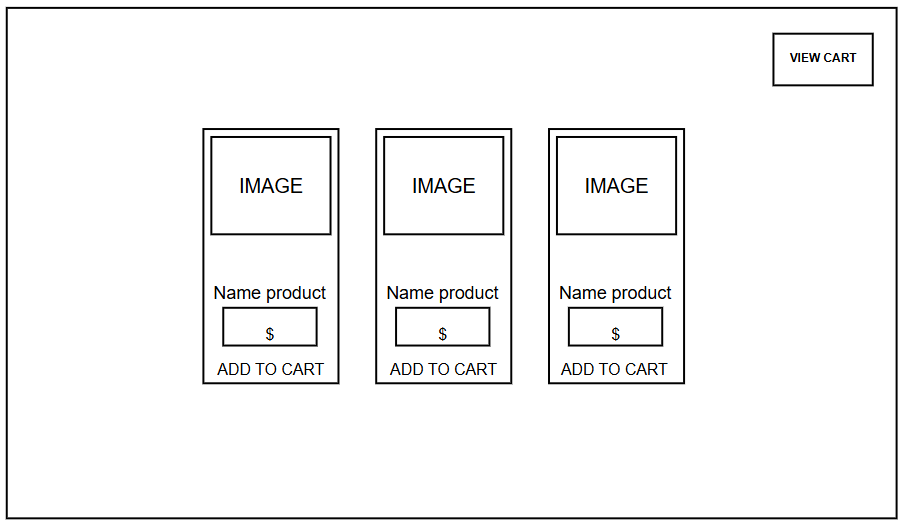


Ilustración Initial prototyping design  
  
Fuente: Elaboración propia.

# Prototipo funcional completo

El prototipo avanzado fue desarrollado con interfaz gráfica mejorada (colores, estilo visual) y con las funcionalidades implementadas, entre ellas:

* Agregar productos al carrito de compras.
* Visualizar el carrito con listado de artículos seleccionados.
* Aprobar la compra, confirmando cantidad y monto total.
* Confirmar el pago mediante Interledger Protocol (ILP), con validación en tiempo real.
* Flujo de transacción completo desde la selección del producto hasta la confirmación de pago.

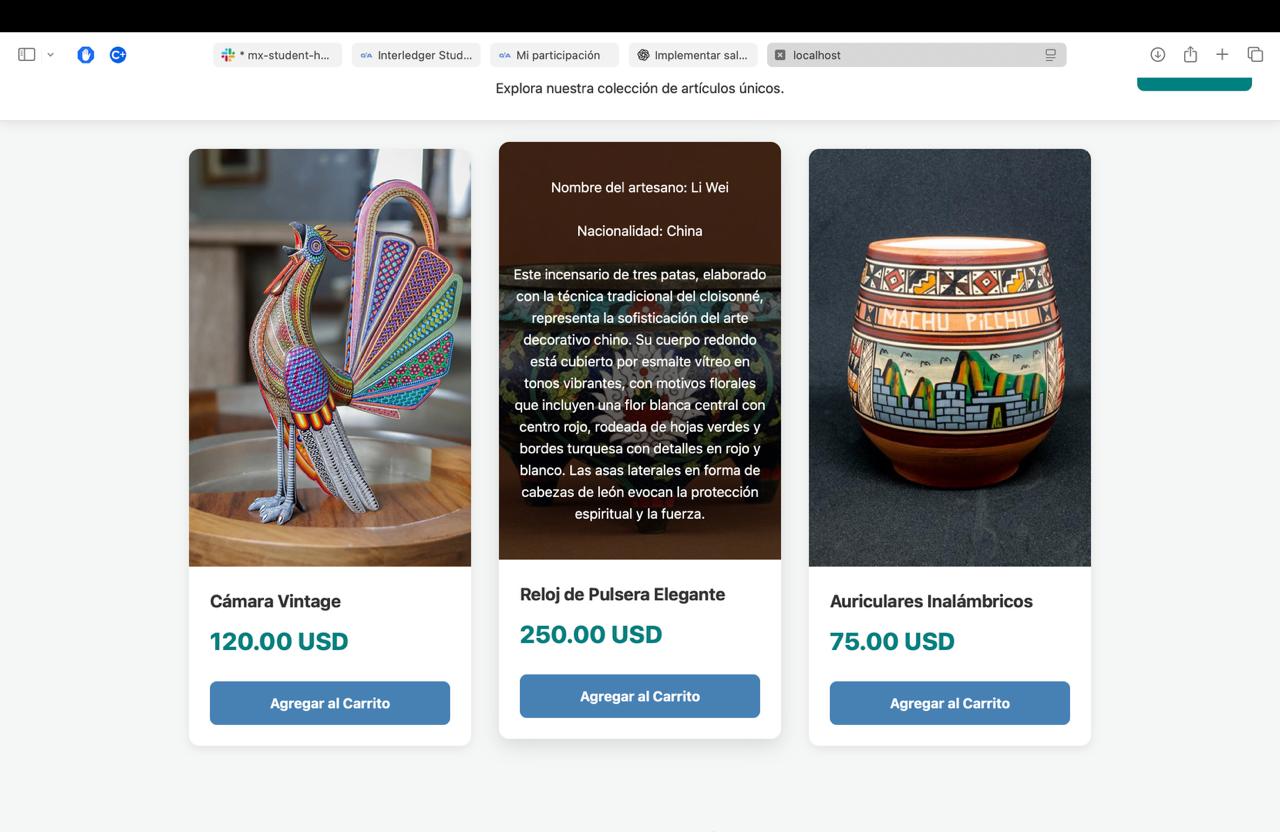
El prototipo funcional ya no solo representa la interfaz, sino que también integra la lógica de negocio, la comunicación con la base de datos y las pruebas iniciales de seguridad (hardening).

Ilustración complete functional prototype

Fuente: Elaboración propia.

# Código fuente y script de base de datos

# Backend (BE)

# Frontend (FE)

# Scripts de base de datos (BD)

El sistema utiliza una tabla central llamada pedido, en la que se registran todas las transacciones realizadas en la plataforma, desde la creación del pedido hasta la liquidación hacia el artesano.

Script de creación de tabla:

CREATE TABLE pedido (

id int(11) NOT NULL AUTO\_INCREMENT,

Transaccion varchar(50) NOT NULL,

Producto varchar(50) NOT NULL,

Cantidad int(11) NOT NULL,

Total decimal(10,2) NOT NULL,

wallet\_address\_artesano varchar(100) NOT NULL,

Status varchar(50) NOT NULL,

Divisa varchar(50) NOT NULL,

fecha\_creacion timestamp NOT NULL DEFAULT current\_timestamp(),

PRIMARY KEY (id)

) ENGINE=InnoDB AUTO\_INCREMENT=29

DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

**Justificación del diseño**

* La tabla concentra toda la información necesaria para procesar pedidos y transacciones en un solo registro.
* La inclusión de fecha\_creacion permite auditar y ordenar cronológicamente las operaciones.
* El campo Status facilita el seguimiento del ciclo de vida del pedido (pendiente → confirmado → liquidado).
* El uso de wallet\_address\_artesano elimina la dependencia de información bancaria tradicional, alineándose con la filosofía del sistema basado en ILP + Open Payments.

# Documentación adicional

# Documentación técnica

1. **Introducción**

PaDuki es un sistema de comercio electrónico diseñado para facilitar la venta de productos artesanales.  
Su principal innovación es que utiliza la API de Interledger, la cual aporta varios beneficios clave:

1. Pagos interoperables

2. Micro pagos.

3. Velocidad en transacciones

4. Reducción de costos

5. Seguridad y descentralización

1. **Objetivos del sistema.**

Objetivo General

Desarrollar un e-commerce inclusivo que facilite la venta de productos artesanales en mercados nacionales e internacionales, utilizando la API de Interledger para habilitar pagos rápidos, seguros y accesibles, con el fin de empoderar a los artesanos y preservar el valor cultural de sus creaciones en un mundo globalizado.

Objetivos Específicos:

* Facilitar la digitalización de artesanos  
  PADUKI brinda una plataforma amigable e intuitiva donde los artesanos publican, gestionan y venden sus productos sin necesidad de experiencia tecnológica avanzada.
* Habilitar pagos globales accesibles  
  Gracias a la integración con la API de Interledger, los artesanos reciben pagos internacionales al instante, con costos reducidos y sin depender de barreras bancarias tradicionales.
* Garantizar seguridad y confianza  
  El sistema cuenta con autenticación robusta, administración de permisos, registros lo que asegura que cada transacción sea transparente y confiable tanto para compradores como para vendedores.Conectar culturas a nivel global  
  A través de la plataforma, compradores de todo el mundo descubren y adquieren piezas artesanales únicas, fomentando el intercambio cultural y la apreciación del arte hecho a mano.
* Optimizar procesos de compra y venta  
  El sistema garantiza rapidez y eficiencia en cada transacción gracias a su arquitectura basada en caché de datos, sincronización de información y servicios básicos que mantienen la disponibilidad de productos en todo momento.
* Fomentar la sostenibilidad del comercio artesanal  
  PADUKI rescata el valor de lo hecho a mano frente a la producción industrializada, asegurando que los beneficios económicos lleguen directamente a los creadores y fortaleciendo así la economía local.

El presente documento técnico esta orientado a desarrolladores con experiencia en tecnologías web, en particular aquellos que dominan:

* Node.js: para la construcción de la lógica de negocio, servicios API y manejo del servidor.
* HTML y CSS: para el desarrollo y personalización de las interfaces de usuario (UI/UX).
* Integración de APIs: especialmente la API de Interledger, que permite la gestión de pagos internacionales seguros y de bajo costo.

1. **Arquitectura del sistema**

Diagramas de arquitectura del sistema

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.En la Figura 1 se muestra la arquitectura propuesta para el sistema de e-commerce PaDuki, estructurada en capas que permiten una mayor organización, escalabilidad y seguridad. Cada capa cumple un rol específico dentro del flujo de compra y pago.

Ilustración System Architecture Diagram

Fuente: Elaboración propia

1. **Tecnologías utilizadas:**

En la Tabla 1 se presentan las principales tecnologías utilizadas en el desarrollo de la plataforma de e-commerce PaDuki, junto con su respectiva función dentro del sistema.

|  |  |
| --- | --- |
| Tecnología | Descripción / Función |
| HTML | Estructura las páginas web del e-commerce, mostrando los productos y la información de los artesanos. |
| CSS | Define el estilo y la apariencia visual de la plataforma, haciendo que la interfaz sea atractiva y coherente con la identidad del proyecto. |
| JavaScript | Agrega interactividad al frontend, maneja eventos, valida formularios y permite la comunicación con el backend. |
| Node.js | Backend que gestiona usuarios, productos, autenticación, transacciones y lógica de negocio, ejecutando JavaScript en el servidor. |
| Visual Studio Code | Entorno de desarrollo donde se escribe, depura y organiza todo el código del proyecto (frontend y backend). |
| MySQL | Base de datos relacional que almacena información de productos, artesanos, usuarios y transacciones de forma segura y estructurada. |
| JSON | Formato de intercambio de datos entre frontend y backend, permitiendo enviar y recibir información de manera ligera y eficiente. |
| Interledger | Sistema de pagos internacionales que conecta distintas redes de dinero digital, permitiendo transacciones rápidas, seguras y de bajo costo para artesanos y clientes. |

**5. Módulos principales**

I. Backend (pagos.js)

El archivo pagos.js es el núcleo del sistema backend. Se encarga de levantar el servidor con Express, habilitar CORS, servir los archivos estáticos y definir las rutas (endpoints) principales de la API.  
En la Figura 1 se muestra el flujo del backend, donde se detalla cómo el servidor procesa las solicitudes del frontend, administra las rutas GET/POST y se comunica con la red de Open Payments para gestionar las transacciones.

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración 4 backend flow

Fuente: Elaboración propia

II. Frontend – Página Principal (client.js)

El script client.js controla la interactividad de la página index.html. Su principal responsabilidad es permitir al usuario agregar productos al carrito. Para ello utiliza un mapa interno (productWallets) que asocia productos con wallets, administra el carrito en el sessionStorage y genera notificaciones con la función showToast().

Diagrama

El contenido generado por IA puede ser incorrecto.  
En la Figura 2 se presenta el flujo de interacción en la página principal, desde que el usuario presiona el botón “Agregar al Carrito” hasta la actualización del carrito y la notificación al cliente.

Ilustración 5 main page

Fuente: Elaboración propia

III. Frontend – Página del Carrito (cart-client.js)

El archivo cart-client.js es el encargado de manejar la lógica de la página del carrito. Administra la representación visual del carrito (updateCartUI()), controla los eventos de los botones (+, -, eliminar), y gestiona el proceso de pago mediante los endpoints /iniciar-pago y /finalizar-pago.

Diagrama, Esquemático

El contenido generado por IA puede ser incorrecto.  
En la Figura 3 se muestra el flujo de la página del carrito, destacando el ciclo de actualización de la interfaz, el envío de datos al backend y la espera de confirmación del pago a través del polling.

Ilustración 6 Cart page

Fuente: Elaboración propia

IV. Interfaz del Administrador – Gestión de Transacciones

La interfaz del administrador centraliza la gestión de las transacciones. Desde este panel, el administrador puede listar, filtrar y buscar transacciones, además de ejecutar pagos hacia los proveedores. Una vez ejecutada una transacción, se envía el dinero a la wallet del proveedor y el estado cambia a “Completado”.

Diagrama

El contenido generado por IA puede ser incorrecto.  
En la Figura 5 se presenta el flujo de la interfaz del administrador, desde la visualización de las transacciones hasta la ejecución exitosa de una operación y la actualización del estado en la tabla.

Ilustración 7 admin interface flow

Fuente: Elaboración propia

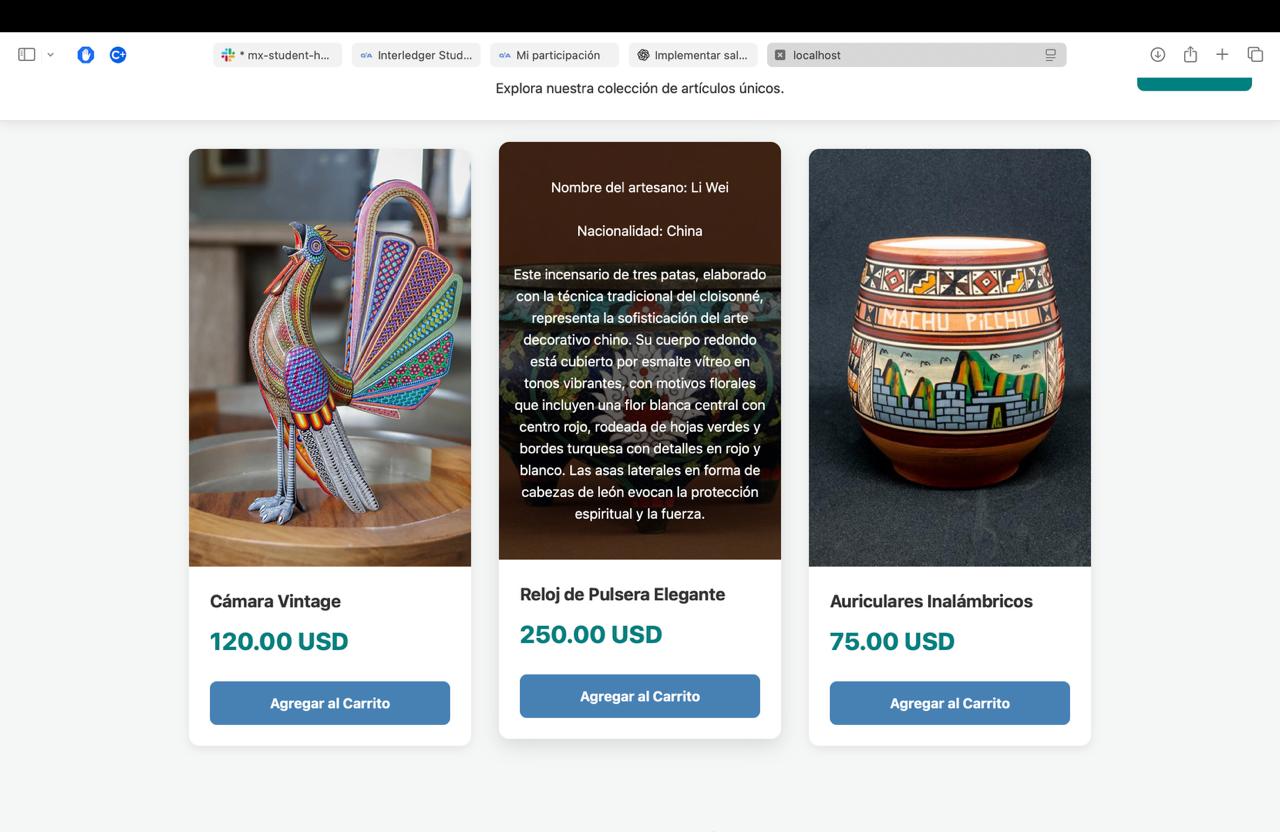
# Manuales de usuario

1. **Cliente**.

Este manual describe el proceso que debe seguir un usuario final (comprador) para realizar una compra dentro de la plataforma.

Pasos de uso:

1. Observar catálogo de productos
   * Al ingresar al sistema, el cliente puede visualizar el catálogo completo de productos artesanales disponibles.



1. Seleccionar producto y agregar al carrito
   * El cliente elige el producto deseado y presiona el botón “Agregar al Carrito”.

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

* + Es posible añadir uno o varios artículos antes de proceder al pago.

1. Acceder al carrito de compras
   * Haciendo clic en el icono del carrito de la esquina superior derecha, el cliente accede al listado de productos seleccionados.

Interfaz de usuario gráfica, Aplicación, Sitio web

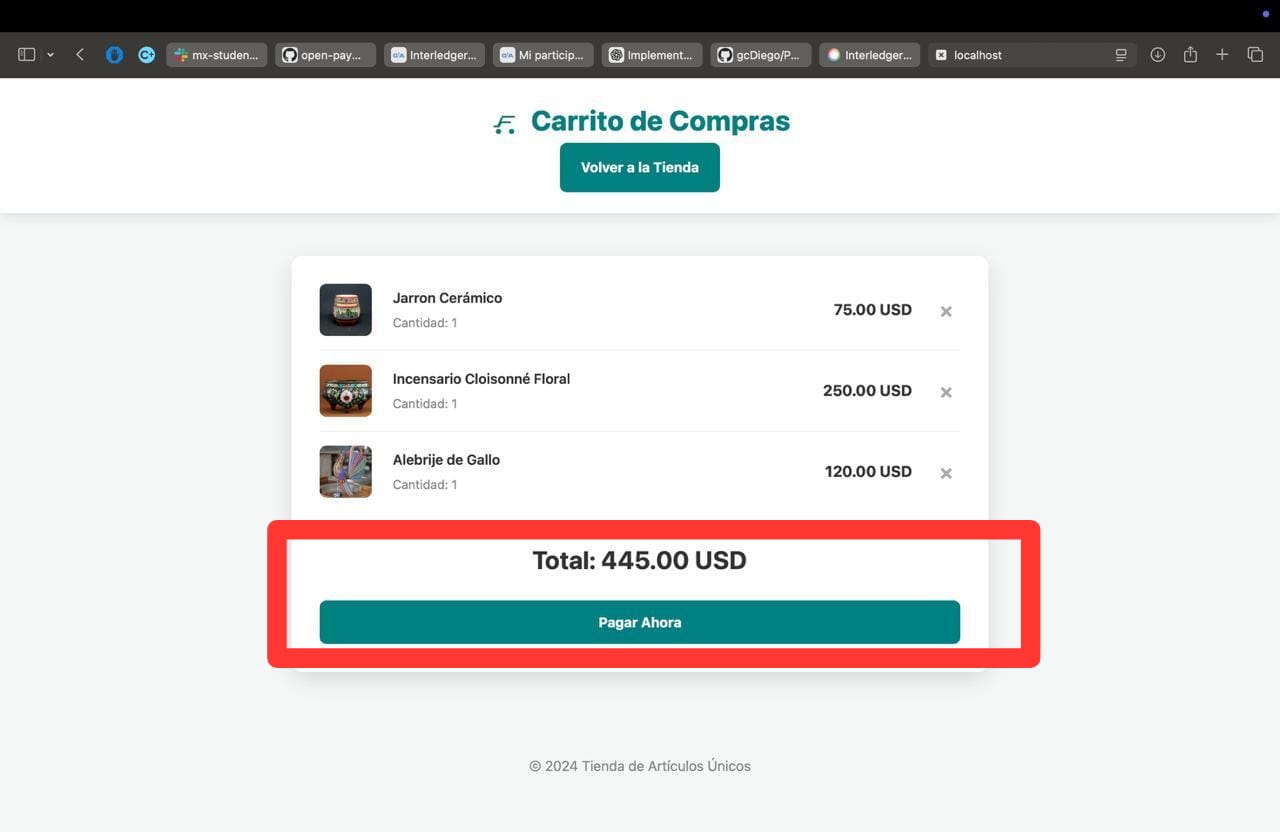
El contenido generado por IA puede ser incorrecto.

1. Confirmar productos y cantidad
   * El sistema muestra los artículos elegidos junto con la cantidad.
   * El cliente puede modificar el número de piezas o eliminar productos.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

1. Proceder al pago
   * Presionar el botón “Pagar ahora” para iniciar el proceso de compra.



1. Confirmación de pago mediante Interledger
   * Aparece una ventana de confirmación del protocolo Interledger (ILP).
   * El cliente debe seleccionar “Accept” para validar la transacción, o “Decline” en caso de cancelar querer cancelar la compra.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

1. Notificación de pago realizado

Interfaz de usuario gráfica, Texto, Sitio web

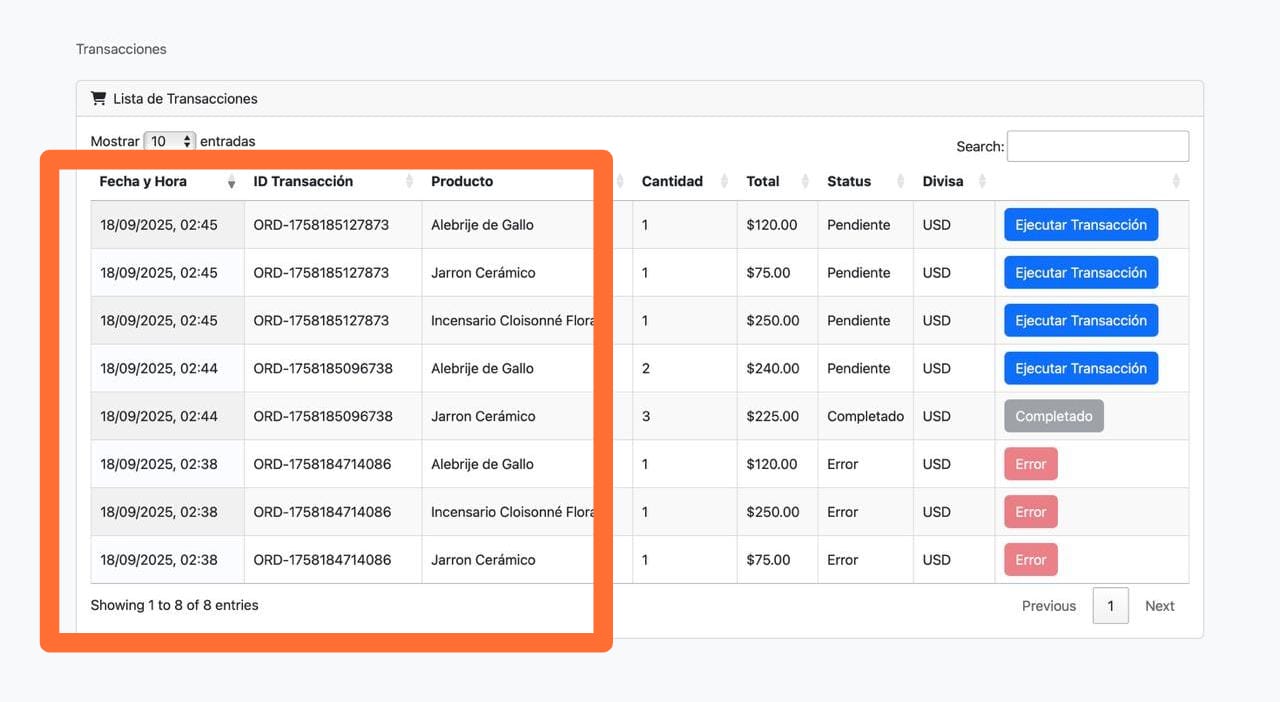
El contenido generado por IA puede ser incorrecto.

1. **Administrador**.

Este manual describe las funciones del administrador dentro del sistema, encargado de validar pagos y gestionar la liquidación hacia los artesanos.

Pasos de uso:

1. Recepción de pagos totales
   * El administrador visualiza los pagos recibidos, que corresponden a la suma de productos adquiridos por el cliente.
   * El sistema muestra la relación producto–artesano, indicando qué vendedor debe recibir cada parte del pago.



1. Validación de entrega del producto
   * Una vez confirmado que el cliente ha recibido el pedido, el administrador autoriza la ejecución de la transacción hacia los vendedores.

Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.

* + Esta validación asegura transparencia en la operación.

1. Confirmación de transacción hacia los artesanos
   * El administrador selecciona la opción “Accept”.
   * El sistema liquida automáticamente el monto correspondiente en la wallet del vendedor, descontando únicamente la comisión establecida (5%).

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

1. Notificación final
   * El artesano recibe confirmación en tiempo real de que los fondos ya están disponibles en su wallet.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

# Diccionario de datos

* 1. **Tabla Pedido**

|  |  |  |
| --- | --- | --- |
| Campo | Tipo de Dato | Descripcion |
| Transaccion | varchar (50) primary key | Identificador de la transacción que hizo el cliente hacia la wallet de administrador |
| Producto | varchar (50) | Nombre del Producto |
| Cantidad | int (100) | Cantidad del producto |
| Total | decimal (10,2) | Dinero total que se transferirá a la wallet del artesano |
| Wallet\_adress\_artesano | varchar(100) | Id o wallet adress del artesano |
| status | varchar(50) | Status de tranferencia a la wallet del artesano |
| Divisa | varchar(50) | Moneda en que realizo la transacción el cliente |

# Conclusiones

**Conclusiones generales**

* El sistema desarrollado demuestra que es posible crear una plataforma inclusiva, escalable y segura que permita a los artesanos participar en el comercio electrónico global con menores barreras de entrada.
* La integración de ILP (Interledger Protocol) y Open Payments ofrece una ventaja competitiva clara frente a plataformas tradicionales, al reducir costos de transacción y garantizar pagos interoperables en tiempo real.
* Se logró un equilibrio entre viabilidad técnica, económica y social, confirmando que la solución no solo es rentable, sino que también aporta valor cultural y de inclusión financiera.
* El prototipo funcional validó la usabilidad y experiencia de usuario mediante una interfaz sencilla, confirmando que los compradores pueden realizar una transacción en pocos clics y los artesanos reciben sus pagos de forma transparente.

**Conclusiones particulares**

* El modelo de comisiones del 5% resultó más justo que el de plataformas comerciales globales, generando incentivos tanto para artesanos como para compradores.
* El uso de una única tabla pedido permitió simplificar el desarrollo inicial y concentrar la información clave de las transacciones, facilitando auditoría y escalabilidad futura.
* El manual de usuario con capturas aporta valor agregado al facilitar la adopción del sistema tanto para clientes como para administradores.
* El flujo de validación a través del Administrador garantiza confianza en el proceso de compra y pago, fortaleciendo la transparencia del sistema.
* Se comprobó la viabilidad económica: con una base de 10,000 compradores anuales, el sistema puede generar ingresos sostenibles desde el primer año.
* A nivel cultural y social, la plataforma se convierte en un canal de visibilidad global para la artesanía, contribuyendo a la preservación y promoción de tradiciones locales.