# 

# Retrospectiva del Proyecto – CondorShop (E‑commerce)

## 

## 1. Datos del Proyecto

* **Proyecto**: CondorShop (E‑commerce)
* **Organización**: Duoc UC – Escuela de Informática y Telecomunicaciones (Sección 004V)
* **Patrocinador**: Duoc UC
* **Docente**: Marcos Valenzuela
* **Equipo**: Benjamín Huenupe (Backend / DevOps) y Juan Rodríguez (Frontend)
* **Fechas**: Inicio 11 de agosto de 2025; término 24 de octubre de 2025
* **Tecnologías**: Node.js / Express, MySQL, HTML + Tailwind CSS + JS, integración con Transbank Webpay Plus; despliegue mediante pipeline CI/CD con staging y producción.

La siguiente retrospectiva global consolida las lecciones aprendidas, resultados y próximos pasos después de cinco sprints de desarrollo. Para mayor contexto, en el anexo se incluyen diagramas del modelo de datos (ERD), la arquitectura inicial, el mapa de actores y un mapa mental de la propuesta de valor.

## 2. Visión y Objetivos del Proyecto

La visión de CondorShop establece una plataforma de comercio electrónico **rápida**, **segura** y **confiable**, en la que cualquier persona pueda **descubrir productos**, **compararlos**, **añadir al carrito** y **pagar con Transbank Webpay Plus**. Se buscó un **flujo simple** que funcione en cualquier dispositivo, con gestión de stock en MySQL, un panel de administración básico y estrategias de SEO para crecer de forma orgánica.

El análisis de los **cuatro pilares** se resume así:

* **Clientes**: actuales y potenciales que esperan acceso conveniente (búsqueda, filtros, orden por precio/popularidad, ficha clara), confianza y seguridad (checkout simple, Webpay Plus, políticas visibles) y operación eficiente (stock consistente, órdenes fiables, reportes).
* **Necesidades**: experiencias de compra fluidas y agradables con catálogo rápido, carrito persistente, checkout sin fricciones (invitado o con cuenta), pago seguro y comunicación transparente (resumen/número de orden, correo de confirmación, estados de pedido).
* **Producto/Servicio**: plataforma web con catálogo y búsqueda de productos, filtros por categoría, ficha de producto, carrito de compras, checkout con Webpay Plus, autenticación básica y panel de administración (futuro).
* **Valor**: mejor experiencia (flujo simple, confianza en el pago), optimización operativa (stock/órdenes en tiempo real), crecimiento del negocio (SEO y mejor conversión en checkout) y seguridad/regulación (pagos regulados, protección de datos).



Mapa Mental

## 3. Resumen de los Sprints y Entregables

A continuación, se resume la evolución a lo largo de los sprints, indicando las épicas principales involucradas y los principales entregables. El desglose completo de historias y tareas se detalla en los documentos de sprint planning y backlog.

### Sprint 1 (11 – 22 de agosto de 2025)

Épicas abordadas: **E1 Catálogo y búsqueda**, **E2 Ficha de producto**, **E3 Carrito**, **E8 Autenticación**. Se implementaron los cimientos del MVP:

* Catálogo paginado y búsqueda por nombre, con paginación básica y orden predefinido.
* Ficha de producto con nombre, precio, stock, descripción e imagen principal; galería básica.
* Funcionalidad de **agregar al carrito** desde la ficha o el catálogo, con feedback visual y persistencia en la sesión.
* Desarrollo de **login y registro** básico (correo/contraseña) con sesiones cifradas y cookies.
* Preparación de IDs de orden para el checkout futuro.
* *Impedimentos: falta de dataset inicial y criterios de aceptación; se generaron datos de prueba y se establecieron criterios claros para cada historia*.

### Sprint 2 (25 de agosto – 5 de septiembre de 2025)

Épicas abordadas: **E3 Carrito**, **E4 Checkout & Webpay**, **E6 Seguridad**, **E8 Autenticación**. Se completó el flujo de compra:

* Formulario de checkout con captura de datos personales y dirección; resumen del pedido con IVA 19 % y aceptación de términos.
* Integración con **Transbank Webpay Plus** para crear y confirmar transacciones, incluyendo redirección a la pasarela y retorno con manejo de estados (aprobado/rechazado/abortado/timeout).
* Lógica de **descuento de stock** al aprobar el pago y carrito persistente con opción “recordarme”; reintento de pago si se rechaza.
* Implementación de validaciones de entrada, rate‑limit, políticas CORS y configuración de variables .env .
* *Impedimentos*: retraso de credenciales Webpay, complejidad en fusión de carritos y sesiones; se solucionó con pruebas sandbox, manejo de callbacks y mejora de la lógica de sesiones.

### Sprint 3 (8 – 26 de septiembre de 2025)

Épicas abordadas: **E3 Carrito & Envíos**, **E4 Checkout (confirmación)**, **E5 Administración**, **E8 Autenticación**. Se consolidó la operación post‑compra y se incorporó back‑office y cuentas:

* **Resumen del carrito** con totales e ítems visibles, controles ±/eliminar y actualización en tiempo real; se introdujo la regla de **envío gratis** (CLP 50 000).
* **Correo de confirmación** con plantilla personalizada y número de orden.
* Panel de administración MVP: CRUD de productos (con subida de imágenes) y gestión de pedidos con cambio de estados; exportación y vistas básicas.
* **Cuentas de usuario**: registro con feedback, recuperación de contraseña, historial de compras y opción de compra como invitado.
* *Impedimentos*: sincronización de stock y optimización del panel; se añadieron logs de stock, se optimizaron consultas y se reforzó la seguridad de la API.

### Sprint 4 (29 de septiembre – 10 de octubre de 2025)

Épicas abordadas: **E1 SEO**, **E6 Seguridad & Observabilidad**, **E7 Testing & CI/CD**. Se preparó el lanzamiento:

* Creación de sitemap.xml, robots.txt y meta tags básicas; URLs limpias y optimización para buscadores.
* Ajustes de **accesibilidad** (contraste, ARIA y navegación por teclado) y usabilidad.
* Configuración del **pipeline de despliegue** a staging y producción con plan de rollback y prueba de publicación.
* *Impedimentos*: coordinación de pruebas de rendimiento y ajustes de pipeline; se desarrollaron pruebas end‑to‑end y se documentó el plan de rollback.

### Sprint 5 (13 – 24 de octubre de 2025)

Épicas abordadas: **E6 Seguridad & Legal**, **E7 Testing & Entrega**. Se realizó el hardening final y la entrega:

* Simulación completa de compra (catálogo → ficha → carrito → checkout) con evidencias (capturas y vídeo).
* Ejecución del **checklist de publicación** y despliegue a producción con validación post‑deploy y rollback.
* Recolección de **métricas de rendimiento** (tiempos de carga, errores, logs) y generación de carpeta de entrega final.
* Redacción de la **política de privacidad**, implementación de banner/aviso de cookies y almacenamiento de preferencias de usuario.
* *Impedimentos*: pruebas de carga y latencias, ajuste de umbrales de alertas y afinación de textos legales; se mejoraron las pruebas de regresión y se adaptó la configuración de monitoreo.

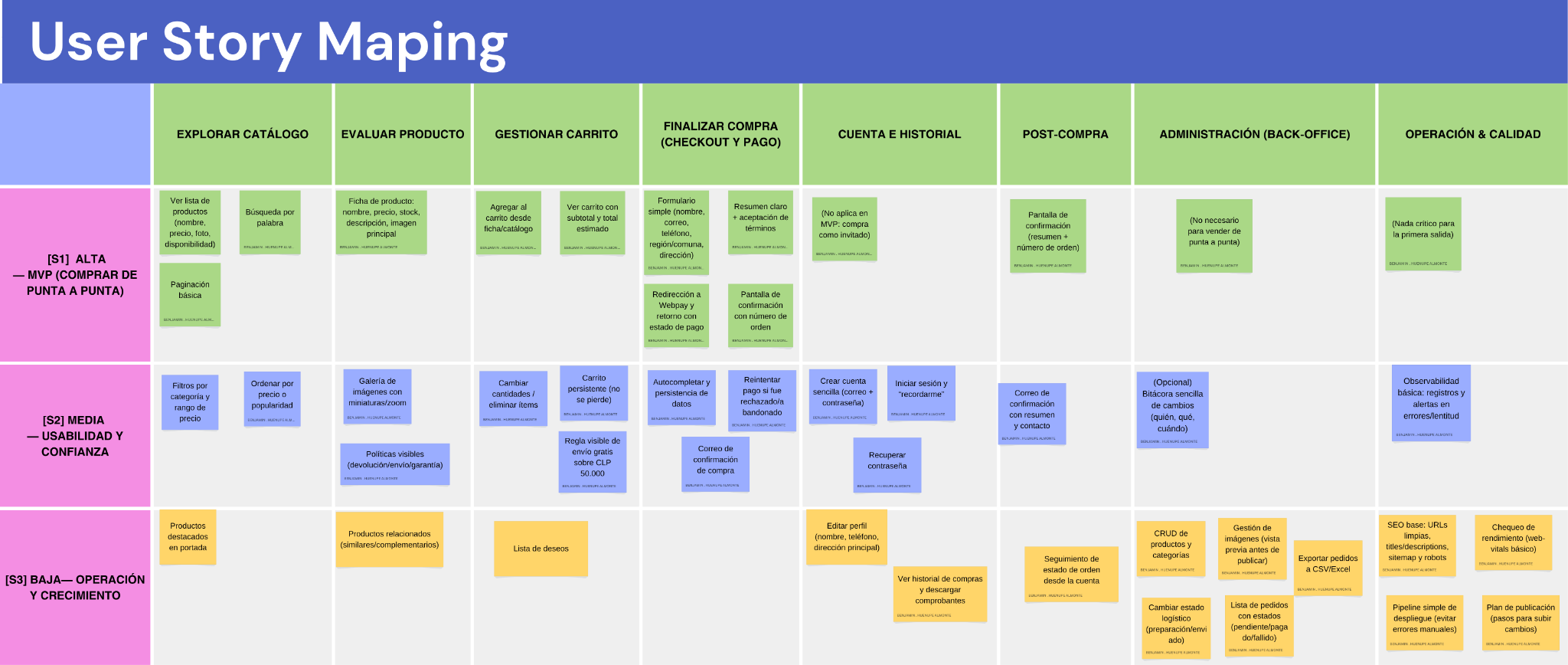
## 4. Mapeo de Historias de Usuario y Épicas

La **User Story Mapping** permite visualizar el producto desde la perspectiva del usuario y priorizar las funcionalidades. En CondorShop se organizaron las historias en ocho columnas que representan el recorrido completo:

1. **Explorar Catálogo** – ver lista de productos con nombre, precio, foto y disponibilidad; búsqueda básica por palabra y paginación.
2. **Evaluar Producto** – ficha de producto con detalle, stock, descripción, imagen principal y galería/zoom; políticas visibles de devolución y garantía.
3. **Gestionar Carrito** – agregar productos al carrito desde la ficha/catálogo, ver subtotal y total estimado, cambiar cantidades o eliminar ítems y contar con un carrito persistente.
4. **Finalizar Compra (Checkout y Pago)** – formulario simple con datos personales y dirección; resumen claro y aceptación de términos; integración con Webpay (redirigir, pagar y retornar); reintentar pago si se rechaza; regla visible de envío gratis.
5. **Cuenta e Historial** – creación de cuenta (correo + contraseña), login con opción “recordarme”, recuperación de contraseña, visualización de historial de compras y descarga de comprobantes.
6. **Post‑compra** – pantalla de confirmación con resumen y número de orden; bitácora de cambios y seguimiento de estado de órdenes.
7. **Administración (Back‑Office)** – CRUD de productos y categorías; gestión de imágenes con vista previa; exportar pedidos a CSV/Excel; cambiar estados logísticos; chequear rendimiento y Web Vitals; pipeline de despliegue y plan de publicación.
8. **Operación & Calidad** – observabilidad básica (logs y alertas), pruebas de rendimiento, QA, CI/CD y evidencias finales.

Cada fila de la historia map define el nivel de prioridad: - **S1 – Alta (MVP)**: Funcionalidades mínimas para comprar de punta a punta (listado, ficha, carrito, checkout básico, confirmación).  
- **S2 – Media (Usabilidad y confianza)**: Filtros, ordenamiento, galerías, carrito persistente, regla de envío gratis, cuenta de usuario, reintentos de pago y observabilidad básica.  
- **S3 – Baja (Operación y crecimiento)**: Destacados, productos relacionados, lista de deseos, perfil de usuario, seguimiento de pedidos, panel admin completo, SEO base, pipeline simple y plan de publicación.

El desarrollo de CondorShop cubrió completamente los ítems de **prioridad alta** y media, quedando las funcionalidades de crecimiento y operación avanzada como oportunidades para iteraciones futuras.



Mapa de historias de usuario y prioridades

Adicionalmente, se definieron **ocho épicas** que agrupan las historias en grandes bloques de valor:

| Épica (ID) | Descripción |
| --- | --- |
| **E1 Catálogo y búsqueda** | Listado y búsqueda de productos con SEO básico, filtros y ordenamiento. |
| **E2 Ficha de producto** | Detalle completo del producto con imágenes, metadatos y políticas visibles. |
| **E3 Carrito & Envíos** | Gestión de carrito persistente, actualización de cantidades, regla de envío gratis. |
| **E4 Checkout & Webpay** | Flujo de checkout con formulario, resumen, integración Webpay y manejo de estados. |
| **E5 Administración** | Panel para administrar productos, pedidos, roles y auditoría; exportaciones. |
| **E6 Seguridad, Observabilidad y Backups** | Protección (CORS, rate‑limit), logs, alertas, backups y manejo de secretos. |
| **E7 Testing, QA y Entrega** | Pruebas unitarias/E2E, métricas, CI/CD, evidencias y publicación del sitio. |
| **E8 Autenticación de Usuario** | Registro, login, recuperación de contraseña, perfil e historial. |

La priorización de épicas, calculada mediante la técnica de los 100 puntos, situó **E4** como la de mayor valor (25 puntos), seguida de **E1** (18 puntos) y **E3** (14 puntos). Las decisiones de desarrollo se alinearon a esta priorización, abordando primero las dependencias críticas para poder llegar a un MVP funcional.

## 5. Principales Resultados

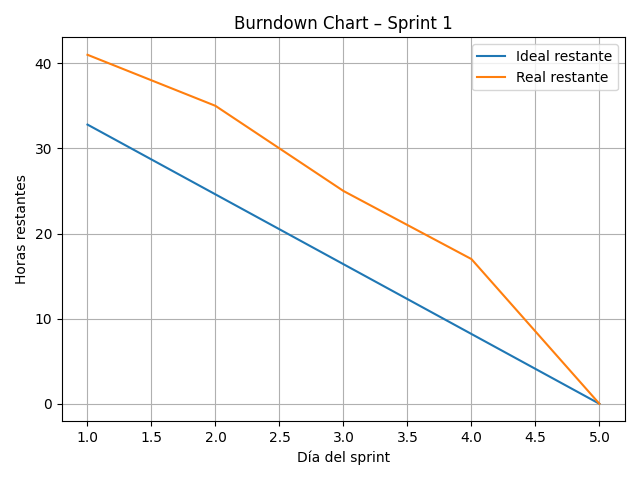
* **Catálogo y búsqueda funcionales** con paginación, filtros básicos y SEO que permiten explorar productos de forma ordenada y rápida.
* **Carrito de compras persistente** con actualización de cantidades y totales en tiempo real, integrado con la regla de envío gratis.
* **Pago seguro** mediante integración con Webpay Plus (sandbox), con manejo completo de estados de transacción y confirmación por correo electrónico.
* **Panel de administración MVP** para crear/editar/eliminar productos (con subida de imágenes), listar pedidos y cambiar su estado; exportar pedidos a CSV.
* **Cuentas de usuario** con registro, recuperación de contraseña, historial de compras y posibilidad de comprar como invitado.
* Mejoras de **SEO** (sitemap, robots, metas) y **accesibilidad** (contraste, ARIA, navegación por teclado), garantizando que el sitio sea indexable y usable.
* **Pipeline de despliegue** a staging y producción con plan de rollback, más evidencias de pruebas y métricas de rendimiento recopiladas.
* Cumplimiento **legal** básico mediante redacción de la política de privacidad y banner de cookies, y almacenamiento de preferencias.

## 6. Impedimentos y Cómo se Resolvieron

| Impedimento | Descripción | Solución |
| --- | --- | --- |
| **Datos de catálogo insuficientes** | Se inició el Sprint 1 sin dataset completo ni criterios de aceptación claros. | Se establecieron criterios de aceptación y Definition of Done, y se creó un dataset de productos de ejemplo para poder construir el catálogo y la búsqueda. |
| **Integración Webpay** | La recepción de credenciales sandbox se retrasó y surgieron dudas sobre el manejo de callbacks, estados abortados y timeouts. | Se solicitó formalmente al proveedor, se implementó un mock temporal y se realizaron logs de callbacks y pruebas negativas; se ajustó la lógica de la orden y los estados. |
| **Persistencia de carrito y sesiones** | Fusionar el carrito de invitado y el del usuario y mantener sesiones seguras fue más complejo de lo previsto. | Se desarrolló una lógica de merge robusta, se utilizaron cookies seguras y se añadieron opciones “recordarme”, validaciones y límites de tasa. |
| **Descuento de stock y concurrencia** | El descuento simultáneo de stock generaba riesgos de inconsistencia. | Se implementaron transacciones atómicas y se añadieron pruebas de concurrencia. |
| **Falta de assets finales y políticas de envío** | Las imágenes y algunos parámetros de negocio (como la regla de envío gratis) se definieron tarde. | Se emplearon imágenes temporales y se documentaron reglas de envío; estas se integraron en Sprints 3 y 5. |
| **Pruebas de rendimiento y regresión** | Hacia el final se detectaron latencias y tests inestables. | Se fortaleció la suite smoke/regresión, se configuró caching estático y CDN, y se ajustaron umbrales y alertas en el pipeline. |
| **Legalidad y privacidad** | La política de privacidad y el banner de cookies quedaron para el final. | En Sprint 5 se redactó la política, se implementó el banner y se almacenaron preferencias del usuario conforme a la normativa. |

## 7. Métricas y Desempeño

No se calcularon velocidades numéricas formales; sin embargo, los diagramas de **burndown** por sprint reflejan un patrón de trabajo efectivo: cada sprint comenzó con ciertos retrasos derivados de impedimentos pero se recuperó el ritmo a medida que se resolvían, completando las historias planificadas. Las horas estimadas versus realizadas se mantuvieron dentro de rangos aceptables. Se ejecutaron pruebas unitarias, de integración y de regresión, así como pruebas de usuario en staging; la integración con MySQL y Webpay no presentó fallas críticas. (Los burndown chart -Sprint 1- 5, estarán en el Anexo).



## 8. Aprendizajes y Recomendaciones

* **Definir criterios de aceptación y preparar datos de prueba desde el inicio** evita bloqueos y retrabajos.
* **Considerar todos los estados y errores en integraciones de pago**; no dejar la integración real para el último momento.
* **Asegurar la consistencia de stock y concurrencia** con transacciones y pruebas de carga.
* **Incluir SEO, accesibilidad y aspectos legales en etapas tempranas** del backlog, no al final.
* **Documentar y automatizar el pipeline de despliegue** con entornos de staging y rollback.
* **Mantener comunicación y transparencia** en las dailys y retros, actualizando el registro de impedimentos con claridad.
* **Apoyarse en el mapeo de historias y en las épicas** para priorizar y alinear expectativas con stakeholders.

## 9. Conclusiones y Próximos Pasos

El proyecto CondorShop logró entregar una aplicación web completa y coherente con los objetivos planteados: un MVP funcional que permite comprar de punta a punta. Se integraron las principales épicas (catálogo, ficha, carrito, checkout, administración básica y autenticación) y se estableció un camino claro para seguir iterando.

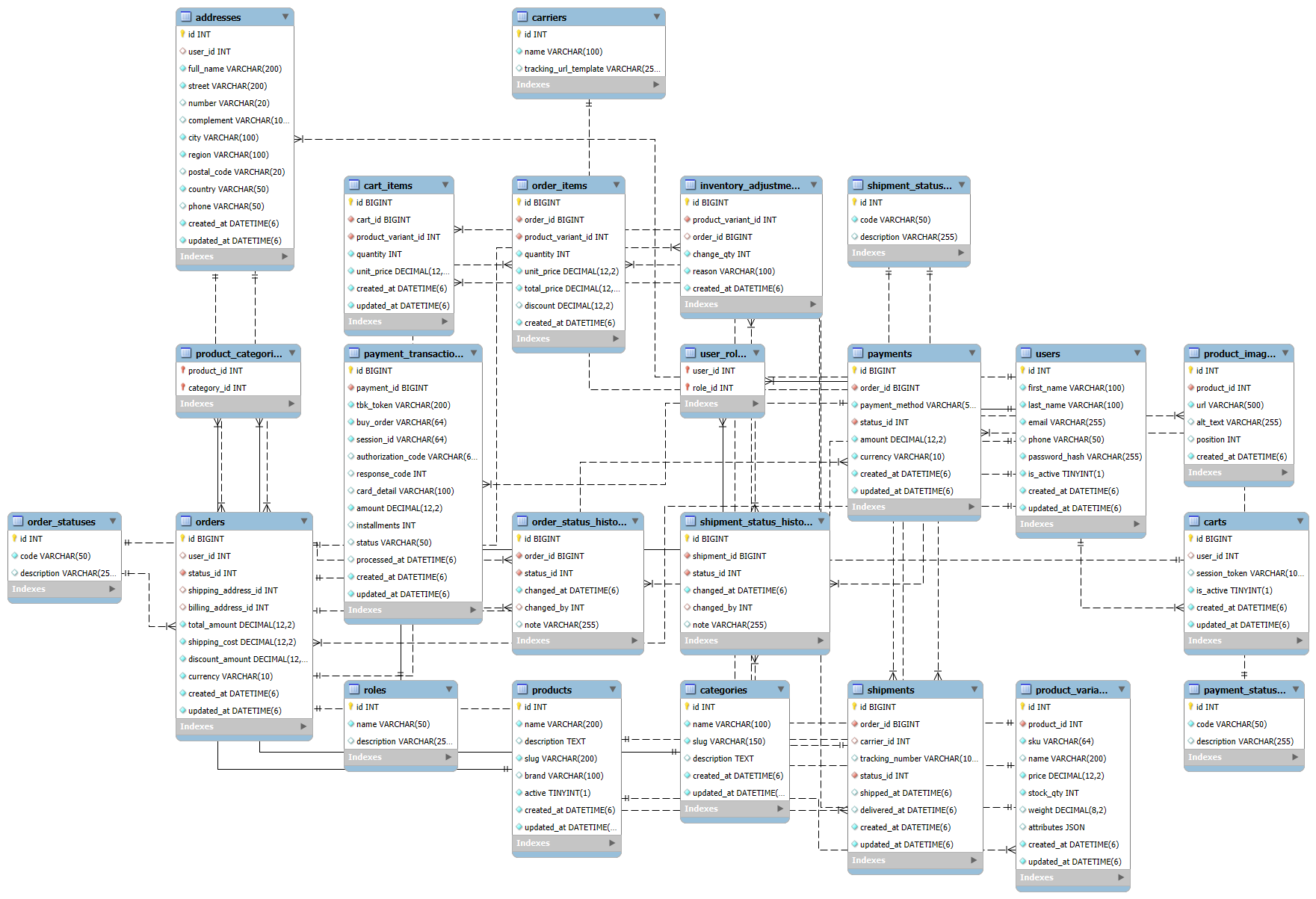
Como próximos pasos sugeridos:

1. **Activar Webpay en producción**, ajustando la configuración real y monitoreando transacciones.
2. **Optimizar el rendimiento** con caching avanzado, compresión y despliegues en contenedores escalables.
3. Ampliar el **panel de administración** con reportes de ventas, promociones, moderación de reseñas y gestión de campañas.
4. Implementar **analítica de comportamiento** y pruebas A/B para mejorar la conversión.
5. Publicar la **API y documentación** para futuras integraciones (mobile, socios).
6. Extender la **funcionalidad de cuenta** con perfil editable, lista de deseos y seguimiento de órdenes en tiempo real.
7. Evaluar la adopción de un **framework moderno** (por ejemplo React + REST/GraphQL) para mejorar la mantención y escalabilidad.

## 10. Anexos y Diagramas

### A. Modelo de Datos (ERD)

El diseño de base de datos soporta las funcionalidades core de CondorShop: usuarios, productos, variantes, categorías, carritos y líneas de pedido, órdenes, pagos, direcciones y bitácoras de estados.

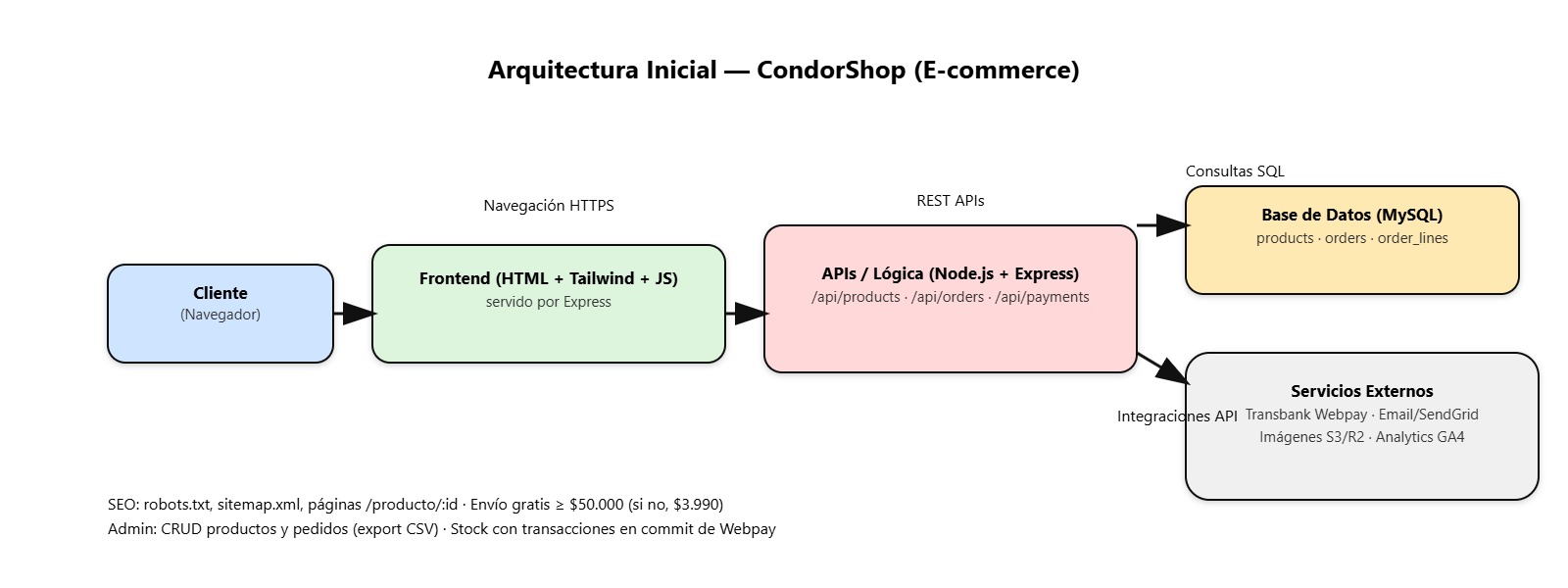


Modelo de datos (ERD)

### B. Arquitectura Inicial

La arquitectura de la solución sigue un modelo cliente–servidor clásico con integraciones externas:

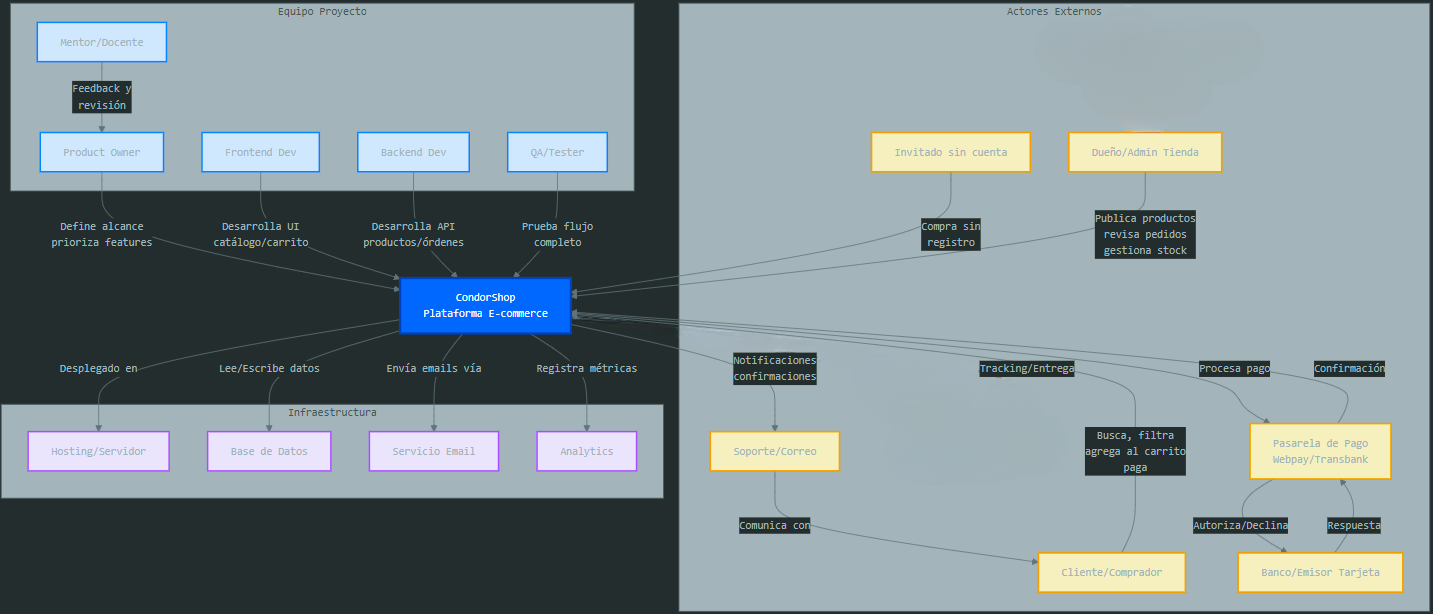
* **Cliente**: interfaz web responsiva consumida desde navegadores.
* **Frontend**: HTML + Tailwind + JS servido por Express, conectado a APIs REST.
* **APIs / Lógica**: Node.js/Express con endpoints /api/products, /api/orders, /api/payments, autenticación básica y seguridad.
* **Base de datos**: MySQL con tablas para productos, órdenes y líneas de pedido, usuarios, etc.
* **Servicios externos**: Transbank Webpay, servicio de correos (SendGrid), almacenamiento de imágenes (S3/R2) y analítica (GA4).



Arquitectura inicial

### C. Mapa de Actores y Responsabilidades

El siguiente mapa sintetiza los actores internos y externos que interactúan con la plataforma, así como sus responsabilidades: equipo de proyecto (mentor/docente, product owner, desarrolladores frontend y backend, QA) y actores externos (invitado, dueño/admin de tienda, plataforma de pago, servicio de correo, infraestructura).



Mapa de actores

### D. Mapa Mental de la Propuesta de Valor

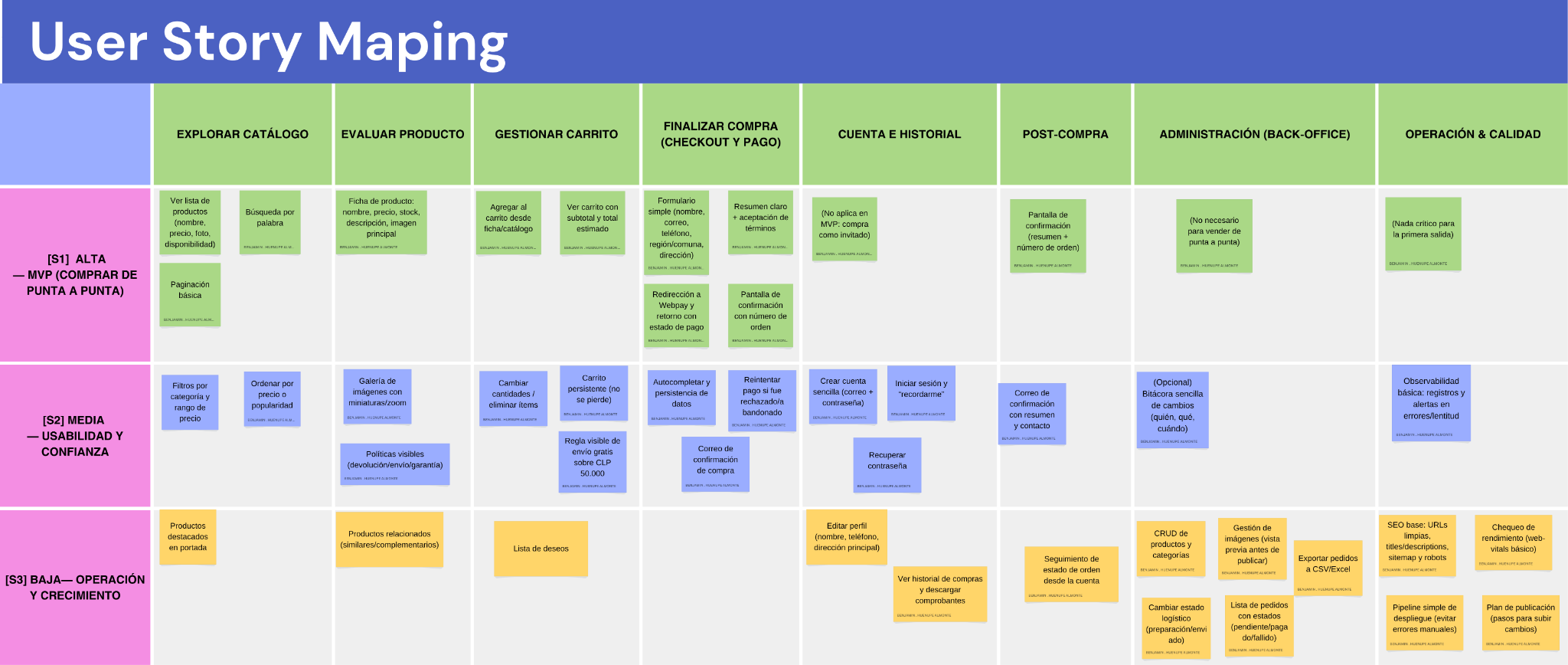
Para complementar la visión, se incluye un mapa mental que agrupa la propuesta de valor, los usuarios, la experiencia de compra, ideas y mejoras, así como los riesgos y planes de contingencia.



Mapa mental

### E. User Story Mapping

Como referencia adicional, se inserta el mapeo completo de historias con sus niveles de prioridad. Ver sección 4 para su explicación.



User Story Mapping

### F. Burndown Charts

Para ilustrar la evolución del trabajo y el cierre de tareas en cada iteración, pueden agregarse aquí los gráficos de burndown generados previamente.

