NicoReparaciones - Documentación histórica y técnica

Plataforma Web de Tienda + Reparaciones (corte: 12 de diciembre de 2025)

Documento pensado para guardarse y retomar el proyecto semanas/meses después con el contexto completo.

Índice

1. Visión y objetivos

- **Qué es:** una plataforma web propia para un local de reparación de celulares y venta de accesorios, pensada para ciudad chica, con foco en simplicidad y control total (sin depender de SaaS cerrados).
- **Objetivos del dueño (Nico):** centralizar stock, pedidos y (próximamente) reparaciones; evitar pedidos falsos; medir costos y ganancias; mejorar visibilidad local.
- **Objetivos del cliente:** navegar productos desde el celular, armar carrito, hacer pedidos y consultar estado (pedidos y reparaciones).

2. Alcance actual (MVP funcional)

- Autenticación (login/registro/logout).
- Tienda (productos + categorías).
- Carrito por sesión (agregar/quitar/modificar cantidades con límite por stock).
- Checkout (finalizar pedido; requiere login).
- Pedidos (cliente: listado y detalle).
- Panel Admin de pedidos (listado, filtros por estado, detalle y cambio de estado).

3. Stack técnico

- Backend: Laravel 12.x, PHP 8.2, MySQL/MariaDB, patrón MVC.
- Frontend: Blade + HTML + CSS propio (mobile-first, minimalista).
- Entorno actual: XAMPP (local), Composer, Artisan, phpMyAdmin.
- Sesiones: driver en archivos (SESSION_DRIVER=file).

4. Arquitectura y organización

- **Capas:** rutas -> controladores -> modelos/consultas -> vistas Blade.
- **Autorización:** middleware auth para usuarios logueados y admin para el panel del dueño.
- **Carrito:** almacenado en sesión para permitir armar carrito sin cuenta; el checkout exige login.
- **Pedidos:** se guarda un snapshot de los items (nombre/precio del momento/cantidad/subtotal) para evitar inconsistencias si cambia el precio o el nombre del producto.

5. Modelo de datos (resumen)

- **users**: name, email, password, role (user|admin).
- **categories**: nombre (y campos de organización simples).
- **products**: nombre, precio, stock, category_id, descripción (y opcionalmente imagen).
- **orders**: user_id, total, status (pendiente/confirmado/preparando/listo_retirar/entregado/cancelado), payment_method, notes, timestamps.
- **order_items**: order_id, product_id (opcional), product_name, unit_price, qty, subtotal.

6. UX implementada (detalles que importan)

- Vista de stock para cliente: no se expone cantidad exacta, solo 'Hay stock' / 'Sin stock'.
- Al agregar al carrito: popup tipo MercadoLibre para evitar redirecciones y mantener fluidez ('Seguir comprando' / 'Ir al carrito').
- Mobile-first: navegación pensada primero para pantalla chica.

7. Rutas y accesos (referencia rápida)

- **Públicas:** / (home), /tienda.
- **Usuario (auth):** /carrito, /checkout, /mis-pedidos, /mis-pedidos/{id}.
- **Auth:** /login, /register, /logout.
- **Admin (auth+admin):** /admin, /admin/pedidos, /admin/pedidos/{id}, /admin/pedidos/{id}/estado.

8. Componentes clave (archivos habituales)

- Config: bootstrap/app.php (routing + middlewares en Laravel 12, sin Kernel clásico).
- Controladores: CartController, OrderController, AdminOrderController, AuthController.
- Vistas: layouts/app.blade.php, orders/*, admin/orders/*, carrito/*.

9. Historia de problemas resueltos (para recordar por qué quedó sólido)

- Laravel 12: ausencia de Kernel.php -> middlewares registrados en bootstrap/app.php.
- Aliases de middleware faltantes (ej. guest, Authenticate) -> se definieron/ajustaron correctamente.
- Rutas faltantes (ej. routes/api.php) -> se agregaron para evitar errores del bootstrap.
- Controladores o vistas no encontradas por estructura de carpetas (ej. admin/orders/index) -> se corrigió el path y convención de carpetas.

10. Checklist de pruebas recomendadas (QA rápido)

- **Auth:** registro, login, logout, acceso a rutas protegidas (redirige a login).
- **Roles:** usuario común recibe 403 al entrar a /admin/*; admin accede sin errores.
- **Carrito:** agregar, sumar cantidades, quitar, actualizar; no permitir exceder stock; carrito vacío bloquea checkout.
- **Checkout:** sin login pide login; con login crea order + items; total correcto.
- **Pedidos cliente:** ve solo sus pedidos; no puede ver pedidos de otro usuario (probar con ID ajeno).
- **Admin pedidos:** filtro por estado; ver detalle; cambia estado y persiste en DB.
- **Stock:** si se descuenta stock al confirmar/crear pedido, probar concurrencia básica (dos pestañas) y evitar negativos.

11. Próximo gran módulo: Reparaciones (roadmap técnico)

- **Objetivo:** gestionar reparaciones como negocio principal: recepción, diagnóstico, presupuesto, estados, costos internos, ganancia, garantía y vista cliente.
- **Tablas sugeridas:** repairs, repair_status_history (opcional), repair_payments (opcional), customers (si querés separar de users).
- **Campos mínimos en repairs:** código/folio, user_id o customer_name+phone, marca/modelo, problema reportado, diagnóstico, costo_repuestos, costo_mano_obra, precio_final, estado, garantía_días, fecha_ingreso, fecha_entrega, notas.
- **Estados típicos:** recibido -> diagnosticando -> esperando_aprobación -> en_reparación -> listo_retirar -> entregado -> cancelado.
- **Vistas:** admin (lista, filtros, alta/edición, detalle), cliente (consulta por código + teléfono o por login).
- **Reglas clave:** historial de cambios de estado con marca temporal; cálculo de ganancia = precio_final - costos; campos obligatorios mínimos para evitar 'reparaciones fantasmas'.

12. Recomendaciones para producción (cuando toque subirlo)

- Configurar .env para producción: APP_ENV=production, APP_DEBUG=false, APP_KEY generado, credenciales DB seguras.
- Definir almacenamiento de imágenes (public disk) y backups.
- Habilitar HTTPS, headers de seguridad básicos y rate limiting para login/registro.
- Agregar logs/monitoreo (Laravel logs + rotación).
- Versionado con Git + releases etiquetadas (ej. v0.1, v0.2).

13. Estado actual y siguiente paso

- Estado: estable y usable en entorno local; panel admin de pedidos funcional.
- Siguiente paso lógico: implementar módulo de reparaciones (MVP) y conectarlo con la experiencia del cliente (consulta de estado).