



DAMORUS PROJECTS

URG Technology V2

Documento de referencia

Especificaciones técnicas de construcción, operación y despliegue

Versión 2.0 del documento · 19 de junio de 2026

Edición pública censurada · 15 de septiembre de 2026

Esta edición omite credenciales, identificadores de servicios, datos de contacto, rutas del panel, datos personales y tarifas. El resto del documento se publica completo.

damorus.com/docs/urg

Índice

URG TECHNOLOGY TIENDA DIGITAL · VERSIÓN 2.0	5
Sobre este documento	5
Objetivo principal (actualizado)	5
Arquitectura y Tecnologías (estado v24.18.26)	5
Identidad visual y Protocolo de Compartición (Favicon, OpenGraph, Twitter Cards)	7
Apartado: SEO Estratégico y Blindaje de Seguridad (estado actual)	8
CONTENIDO (Tabla de Contenidos)	9
1. IDENTIDAD VISUAL Y LÓGICA DE PRECIOS · URG TECHNOLOGY	11
1.1 Paleta de Colores (Regla de Proporción)	11
1.2 Arquitectura del Precio (Anclaje de Valor)	12
2. PÁGINA HOME	13
2.1 Cabecera (Header)	13
2.2 Hero Section	14
2.3 Burbujas CTA	15
2.4 Oferta del Día (Ficha #1 horizontal)	15
2.5 Más Vendidos (Ficha #1 vertical)	18
2.6 Compramos tu PC (bloque promocional del Home)	18
2.7 Diseño Web Pro (banner promocional → damorus.com)	19
2.8 Nuestra Tienda (catálogo en Home)	20
2.9 Directorio Dinámico de SEO	20
2.10 Botón Master Ascent (CTA de Retorno Infinito)	21
2.11 FOOTER	21
2.12 BOTÓN DE WHATSAPP (FAB flotante)	23
3. PESTAÑAS / RUTAS DEDICADAS	25
3.1 /catalogo · Ecosistema de Búsqueda y Catálogo Maestro	25
3.2 /productos/:slug · Ficha 3 (Terminal de Producto)	27
3.3 /carrito	28
3.4 /checkout	28
3.5 /confirmacion/:idPedido	28
3.6 /preguntas-frecuentes (canónica) + /faq (alias)	28
3.7 /compramos-tu-pc	29
3.8 /rastrear-pedido	30
3.9 /privacidad	31
3.10 /compra-protegida	32
3.11 /diseno-web → redirect a damorus.com	32
3.12 Atajos de catálogo (/ofertas, /portatiles, /escritorio, /perifericos)	32
3.13 /mas-vendidos	32
4. SISTEMA DE FICHAS	34
4.1 FICHA 1 · Estándar de Catálogo	34
4.2 FICHA 2 · Búsqueda Avanzada	35

4.3 FICHA 3 · Terminal de Producto (Vista de Conversión Final)	35
4.4 ADAPTACIÓN DE FICHAS PARA PERIFÉRICOS	37
5. APARTADO DE PAGOS	38
5.1 Formulario de Datos para la Facturación	38
5.2 Confirmación de Pedido Digital (PDF Orden)	40
5.3 Sistema de Nomenclatura y Trazabilidad (ID de Pedido)	40
5.4 Modelo de Reservas de Stock Atómico (CRÍTICO · server-side)	41
5.5 Cancelación Cruzada BRE-B ↔ MP	45
5.6 Cronómetro semáforo BRE-B (mobile draggable)	45
5.7 Pantalla "Tu tiempo expiró" (BRE-B)	46
5.8 Persistencia de carrito post-MP (anti-pérdida)	46
6. PANEL DE ADMINISTRADOR (/[panel])	47
BLOQUE I · OPERACIÓN DIARIA	47
(puesto 1, 2, 3...).	56
6 · Continúa con BLOQUE II (Gestión del Cliente)	61
6 · BLOQUE II · GESTIÓN DEL CLIENTE	62
6.5 Comunidad URG	62
6.6 Centro de Mensajería y Automatización (Resend + Supabase)	63
6.7 Centro de Preguntas Frecuentes (FAQ Hub)	68
6 · BLOQUE III · INTELIGENCIA Y SEGURIDAD	70
6.8 Estadísticas e Informes (Business Intelligence)	70
6.9 Centro de Inteligencia y Vigilancia (Black Box D1)	72
6 · BLOQUE IV · CONFIGURACIÓN Y ESTRUCTURA	75
6.10 Configuración Tributaria, Financiera y Logística	75
6.11 Panel Maestro (Ajustes Técnicos y Seguridad)	76
6.12 Perfil y Sesión	80
7. LÓGICA TÉCNICA TRANSVERSAL Y PREVENCIÓN DE ERRORES	81
7.1 Sanitizador de Telecomunicaciones (href="tel:")	81
7.2 Codificador URL Dinámico (WhatsApp)	81
7.3 Protocolo de Seguridad de Sesión e Inactividad	81
7.4 Modales de Confirmación de Acción	81
7.5 Seguridad de Rutas y Middleware (Anti-ByPass)	82
7.6 Protocolo de Integridad de Datos (Soft Delete)	82
7.7 Optimización de Medios (Procesamiento + Pre-visualización)	82
7.8 Lógica de Auditoría · Carrito Abandonado	83
7.9 Página 404 · Dark Terminal	83
7.10 Performance · Indexación BD (Supabase)	84
7.11 Validación de Sintaxis en Tiempo Real	84
7.12 Micro-interacciones (Toast)	84
7.13 Breadcrumb Dinámico	84

7.14 Protección Entrada de Datos (Anti-XSS)	84
7.15 Jerarquía Resiliencia (Kill-Switch)	85
7.16 Optimización Consultas Catálogo	85
7.17 Configuración Servidor (.htaccess)	86
7.18 Verificación SDKs Externos (Anti-AdBlock)	87
7.19 Gestión Timeouts y Estados Carga Asíncrona	87
7.20 Optimización Recursos y Límites Almacenamiento	87
7.21 Integridad Despliegue y Lógica Temporal	88
7.22 Protocolo Escucha de Pagos (Webhook)	88
7.23 Notificación Estado Red (Resiliencia UX Pasiva)	88
7.24 Sincronización Historial SPA (Native-Feel)	88
7.25 Eficiencia Conexiones (Anti-Error 508)	89
8. BACKEND SERVER-SIDE (NUEVO en V2.0)	90
8.1 Edge Functions (Deno runtime, Supabase Cloud)	90
8.2 Trigger pg_cron	91
8.3 Webhook MercadoPago · Detalle Atómico	92
8.4 Cancelación Cruzada BRE-B ↔ MP	93
9. PIPELINE CI/CD (NUEVO en V2.0)	94
9.1 GitHub Actions Workflow	94
9.2 FTP Deploy con Clean-Slate Protegido	95
9.3 Variables de Entorno y Secrets	95
9.4 Rollback Manual	95
9.5 Hosting HostGator · Configuración	95
X. SISTEMA DE COOKIES (URG)	97
1. Funcionalidad Técnica	97
2. Interfaz Aviso Legal	97
3. Integración Supabase para Analítica	97
4. Recuperación de Ventas (Recompra)	97
5. Segmentación para Anuncios	97
6. Privacidad y Transparencia	98
X. DATOS LEGALES URG TECHNOLOGY	99
APÉNDICE A · Changelog Resumido (v1.0 → v24.18.26)	100
APÉNDICE B · Estructura de Carpetas src/	101
APÉNDICE C · Diccionario Resumido de Tablas Supabase	102
CIERRE	103

URG TECHNOLOGY TIENDA DIGITAL · VERSIÓN 2.0

Especificaciones Técnicas de Construcción, Operación y Despliegue (V 2.0) Fecha de actualización: 19 de junio de 2026 Estado del producto en producción: v24.18.26 desplegado en <https://damorus.space> (espejo público) y previsto para urgtechnology.com.co

Versión pública censurada (septiembre de 2026). Se omitieron credenciales, identificadores de servicios, datos de contacto internos, rutas del panel, datos personales y tarifas. Cada dato omitido aparece entre corchetes, por ejemplo [correo de la tienda].

Sobre este documento

Este documento técnico es la actualización mayor de "URG TECHNOLOGY TIENDA DIGITAL VERSION 1.1" (27 de abril de 2026). El sitio fue construido, depurado y endurecido a lo largo de más de 290 iteraciones registradas en el sistema de tareas interno; cada apartado de este documento refleja el estado **actual y verificado en código** de la tienda en lugar del estado aspiracional original.

La intención de esta V2.0 es que un equipo externo (otra agencia, un desarrollador independiente o una Inteligencia Artificial moderna) pueda **reconstruir una copia funcionalmente idéntica** (o cuando menos un 99% similar) de URG Technology partiendo únicamente de este documento. Se mantienen los mismos bloques, la misma numeración y el mismo tono didáctico-técnico de la V1.1; lo que cambia es el contenido: incongruencias se reescriben con la realidad, decisiones nuevas se documentan y se añaden secciones que no existían (Centinela, arquitectura server-side, pipeline CI/CD a HostGator, modelo de reservas de stock atómico, etc.).

Objetivo principal (actualizado)

Operar una **Single Page Application** de alto rendimiento, construida con React 19 + Vite 6 + Tailwind CSS v4 + Framer Motion, conectada a un backend "BaaS" sobre **Supabase** (PostgreSQL + Edge Functions Deno + pg_cron + Realtime + Storage) y desplegada como build estático en **HostGator (cPanel, Apache)** vía **GitHub Actions con FTP automatizado** cada vez que se hace `git push origin main`.

El producto final es una tienda B2C colombiana de hardware reacondicionado y nuevo (portátiles, escritorios y periféricos) con un panel de administración propietario (`/[panel]`) que gobierna catálogo, pagos (MercadoPago + BRE-B con comprobante), inventario, logística, fidelización, mensajería transaccional (Resend), inteligencia de negocio, monitoreo operacional y seguridad. Funciona **sin que ningún administrador deba mantener una pestaña abierta**: la lógica crítica vive en el servidor (Edge Functions + cron) y el cliente es solo presentación.

Arquitectura y Tecnologías (estado v24.18.26)

Stack del Front-End

- **Framework Core:** React 19.0.0 (componentes funcionales, hooks, Suspense, sin Server Components: el sitio se sirve 100% estático).

- **Bundler / Dev Server:** Vite 6.0.3 con `@vitejs/plugin-react` 4.3.4.
- **Router:** react-router-dom 7.0.2 (Routes con Layout anidado, `Navigate`, redirects 301-style vía `window.location.replace`).
- **Estilos:** Tailwind CSS v4.0.0 con el plugin oficial `@tailwindcss/vite`. PostCSS 8 y Autoprefixer 10 como pipeline; sin SCSS, sin CSS-in-JS, sin StyleX.
- **Animaciones:** framer-motion 11.11.17 (animate presence, layout animations, useInView, useScroll).
- **Lectura de metadatos de imágenes:** exifr 7.1.3 (usada por `fileForensics.js` para borrar EXIF al recibir fotos del cliente en "Compramos tu PC").
- **Lenguaje:** JavaScript ESM puro (no se usa TypeScript en el cliente; los tipos viven implícitamente y se documentan vía JSDoc cuando hace falta).

Stack del Backend (Supabase Cloud)

- **Proyecto activo:** `[id-del-proyecto]` (región **West US · Oregon**, plan Free escalable a Pro).
- **Base de datos:** PostgreSQL 15 con Row-Level Security (RLS) activa en todas las tablas de operación.
- **Realtime:** WebSockets nativos de Supabase para refrescar el panel admin (Pendientes, Comunidad, Centinela) sin polling.
- **Storage buckets:**
 - `productos/` (galerías de fichas, fotos optimizadas a WebP $\leq 1920\text{px}$ y $\leq 350\text{ KB}$).
 - `compramos-pc/` (fotos subidas por el cliente vendedor, purgadas automáticamente a los 31 días).
 - `comprobantes-brebe/` (PDF/JPG de comprobantes BRE-B, retenidos hasta archivar la venta).
- **Edge Functions (Deno runtime):** 13 funciones activas listadas en §10 (mp-webhook, cleanup-stock-fantasmas, admin-auth, admin-config-write, mp-create-preference, mp-validate-token, mp-update-token, mp-update-public-key, resend-validate-token, resend-update-token, resend-send-test, turnstile-config, mp-diagnostics).
- **pg_cron + pg_net:** extensiones habilitadas. Hay un único job (`cleanup-stock-fantasmas-2min`) que invoca la Edge Function de limpieza cada 2 minutos; los detalles están en §10.4.
- **Auth:** Supabase Auth con MFA TOTP, magic links de 15 minutos y RLS basada en JWT custom claims (`role` y `bloques`).

Stack de mensajería transaccional

- **Resend** (resend.com) para todos los correos automáticos del flujo de tienda (validación de pagos, despachos, reseñas, recordatorios, alertas Black Box). Dominio verificado con SPF, DKIM y DMARC; la API Key vive en una variable de entorno cifrada y nunca se expone al cliente.
- **Plantillas** redactadas en HTML inline-CSS compiladas a partir de React Email; persistidas en la tabla `plantillas_correo` para que el admin pueda editar copy y variables sin tocar código.

Stack de pagos

- **MercadoPago Checkout Pro** (preferencias creadas por Edge Function `mp-create-preference`, validadas con `mp-validate-token` antes de cualquier guardado, y confirmadas exclusivamente por `mp-webhook` en su evento `payment.updated.approved`).
- **BRE-B** (Banco de la República: transferencia inmediata interbancaria de Colombia). Pago manual: el cliente recibe llave + QR, sube comprobante, el admin valida en el panel.
- **Cloudflare Turnstile** invisible en el login de admin y formularios de retoma para frenar bots.

Hosting y Deploy

- **HostGator (compartido)** con cPanel + Apache 2.4.
- Dominio operativo de staging: **damorus.space** (CNAME hacia el servidor compartido de HostGator). Dominio de marca futura: `urgtechnology.com.co`. Ambos sirven el mismo build estático.
- **Pipeline:** repositorio GitHub privado, rama `main` → GitHub Actions (workflow `Deploy a HostGator vía FTP`) → `npm ci` + `npm run build` → upload por FTP con `SamKirkland/FTP-Deploy-Action@v4.3.5`. Cada `git push` dispara un build y deploy completo (~2 min p99). Cobertura completa en §11.

Estructura de salida y servidor

- **Contenedor final:** todo el contenido reside en la carpeta `/dist/` (generada por `vite build`). Tamaño estabilizado tras v24.18.26 en ~1.8 MB gzip total (sin las imágenes externas).
- **Gestión de rutas:** todos los activos usan rutas relativas (`./assets/...`). El `index.html` está en la raíz de `/dist/` y referencia los hashes generados por Vite (`main-[hash].js`, `index-[hash].css`).
- **Protocolo SPA (Apache):** el paquete incluye obligatoriamente un `.htaccess` con SPA fallback, redirección HTTPS, cache headers, cabeceras de seguridad, CSP y bloqueo de `.env`, `.git*`, `package*.json`, `vite.config.js`. Detalle exacto en §7.17.
- **Codificación universal:** UTF-8 sin BOM en todos los archivos. `AddDefaultCharset UTF-8` declarado en `.htaccess`.

Optimización y resiliencia

- **Responsividad Elite:** diseño mobile-first; meta-viewport con `width=device-width, initial-scale=1, viewport-fit=cover`. Breakpoints Tailwind estándar (`sm` 640, `md` 768, `lg` 1024, `xl` 1280) más un breakpoint propio `xs` 480 para celulares pequeños.
- **Error Boundaries de React** envolviendo cada ruta (`src/components/system/ErrorBoundary.jsx`) para que un error en un widget aislado no rompa el sitio.
- **Animation throttling:** hook `useIsInView` pausa animaciones costosas de Framer Motion fuera del viewport (v21 introducido: reducción medible de CPU en mobile).
- **Cache-busting automático:** Vite hashea por contenido cada JS/CSS y emite headers `Cache-Control: public, max-age=31536000, immutable` desde `.htaccess`; el `index.html` se sirve siempre con `no-store, no-cache, must-revalidate`.
- **Limpieza de build:** Vite elimina `console.log` en producción, sin source maps en el bundle final, sin archivos de configuración expuestos.

Identidad visual y Protocolo de Compartición (Favicon, OpenGraph, Twitter Cards)

Sin cambios respecto a V1.1 en filosofía, pero el inventario de activos se consolidó en `public/assets/brand/`:

```

public/assets/brand/
├─ URG_FAVICON_2026.svg          (fuente principal del favicon)
├─ favicon.ico                  (32x32 generado a partir del SVG)
├─ icon-192.png                 (PWA Android/iOS · Apple Touch)
├─ icon-512.png                 (PWA grande)
├─ URG_HOMELOGO_2026.svg        (logo del header púrpura/morado)
├─ URG_FOOTER_2026.svg          (logo del footer en blanco mate)
├─ og-1200x630.jpg              (OpenGraph card)
├─ MERCADO-PAGO_LOGO_2026.svg
├─ BRE-B_LOGO_2026.svg
├─ VISA_LOGO_2026.svg
├─ MASTERCARD_LOGO_2026.svg
├─ AMERICAN-EXPRESS_LOGO_2026.svg
├─ NEQUI_LOGO_2026.svg
├─ DIAN_LOGO_2020.svg
├─ CCB_LOGO_2026.svg
└─ DAMORUS_BLANCO.svg           (firma del proveedor)

```

Meta-tags inyectados en `index.html`:

- `og:title` = "URG Technology | Especialistas en Computadores de Alto Rendimiento"
- `og:description` = "Venta de hardware premium, equipos gamer y estaciones de trabajo. Calidad técnica garantizada en Bogotá."
- `og:image` = ruta absoluta a `og-1200x630.jpg` (servida vía HTTPS).
- `og:url` = URL canónica de la página actual generada en runtime por `siteConfig.js`.
- `theme-color` = `#8A2BE2` (Púrpura URG): fuerza la barra del navegador móvil al color de marca, eliminando la sensación de "link spam".
- `twitter:card` = `summary_large_image`.

Apartado: SEO Estratégico y Blindaje de Seguridad (estado actual)

1. SEO de Alto Impacto

- **Jerarquía semántica respetada:** `<h1>` único por página (nombre del producto en ficha 3, título de sección en Home), `<h2>` para beneficios principales, `<h3>` para garantía/specs/respaldo.
- **Meta-etiquetas dinámicas:** generadas con `useEffect` por página (cada `pages/*.jsx` setea `document.title`). El detalle de producto compone título en formato `"{nombreMarketing} | URG Technology"`.
- **sitemap.xml** publicado en `/public/sitemap.xml` con rutas estáticas (`/`, `/catalogo`, `/preguntas-frecuentes`, `/compra-prottegida`, `/privacidad`, `/compramos-tu-pc`, `/rastrear-pedido`). El detalle de cada producto NO se sirve en sitemap porque vive en Supabase y rotaría diario; en su lugar se confía en JSON-LD + tráfico orgánico desde `/catalogo`.
- **JSON-LD Schema.org/Product** inyectado en cada `ProductoDetalle.jsx` con name, image, description, brand, offers (price, priceCurrency: COP, availability). Si la descripción manual es < 150 caracteres, el sistema concatena automáticamente specs + garantía como bloque de respaldo (`pdfBuilder.js` reutiliza esa misma lógica para el PDF de orden).
- **Atributos alt** generados dinámicamente: `alt="Computador {nombreMarketing} disponible en Bogotá - URG Technology"`.

2. Blindaje de `/[panel]` (Seguridad)

- `/public/robots.txt` declara `Disallow: /[panel]` y `Disallow: /[panel]/*`.

- `<meta name="robots" content="noindex, nofollow">` se inyecta dinámicamente en `AdminLogin.jsx` y `AdminLayout.jsx` mediante `useEffect`.
- **Sin enlaces internos al panel** desde el sitio público: la única manera de llegar a `/[panel]` es teclear la URL.
- **Cloudflare Turnstile invisible** validando el login y los formularios sensibles (componente `TurnstileWidget.jsx` con site key en variable de entorno `VITE_TURNSTILE_SITE_KEY` y secret server-side en la Edge Function `turnstile-config`).
- **Rate limiter local** (`src/lib/rateLimiter.js`) que aplica retardo progresivo $1 \rightarrow 2 \rightarrow 4 \rightarrow 8$ segundos por fallos de login en la misma sesión.

CONTENIDO (Tabla de Contenidos)

1. IDENTIDAD VISUAL Y LÓGICA DE PRECIOS · URG TECHNOLOGY
2. PÁGINA HOME
 - 2.1 Cabecera
 - 2.2 Hero Section
 - 2.3 Burbujas CTA
 - 2.4 Oferta del Día (Ficha #1 horizontal)
 - 2.5 Más Vendidos (Ficha #1 vertical, 4 categorías × top 3)
 - 2.6 Compramos tu PC (formulario 9 pasos + Cero Estrés)
 - 2.7 Diseño Web Pro (migrado a damorus.com · banner promocional)
 - 2.8 Nuestra Tienda (catálogo)
 - 2.9 Directorio Dinámico SEO
 - 2.10 Botón Master Ascent
 - 2.11 Footer
 - 2.12 WhatsApp Float
3. PESTAÑAS / RUTAS DEDICADAS
 - 3.1 /catalogo (Ecosistema de búsqueda y catálogo maestro)
 - 3.2 /productos/:slug (Ficha 3)
 - 3.3 /carrito
 - 3.4 /checkout
 - 3.5 /confirmacion/:idPedido
 - 3.6 /preguntas-frecuentes (con redirect /faq → /preguntas-frecuentes)
 - 3.7 /compramos-tu-pc
 - 3.8 /rastrear-pedido
 - 3.9 /privacidad
 - 3.10 /compra-prottegida
 - 3.11 /diseno-web (redirect 301-style a https://damorus.com)
 - 3.12 /ofertas /portatiles /escritorio /perifericos (atajos al catálogo)
 - 3.13 /mas-vendidos (página completa con 12 fichas)
4. SISTEMA DE FICHAS
 - 4.1 Ficha 1 (Estándar Catálogo)
 - 4.2 Ficha 2 (Búsqueda Avanzada)
 - 4.3 Ficha 3 (Terminal de Producto)
 - 4.4 Adaptación para Periféricos
5. APARTADO DE PAGOS
 - 5.1 Formulario de Facturación
 - 5.2 Confirmación de Pedido Digital (PDF Orden)
 - 5.3 Nomenclatura y Trazabilidad (ID de Pedido YYYY-###X)
 - 5.4 Modelo de Reservas de Stock Atómico (NUEVO · v24.18.15+)
 - 5.5 Modal "¡Alguien fue más veloz!" (overselling cross-method)
6. PANEL DE ADMINISTRADOR (`/[panel]`)

BLOQUE I · Operación Diaria

 - 6.1 Login
 - 6.2 Centro de Control Inmediato (Barra superior + Sidebar + Dashboard)
 - 6.3 Creador de Fichas (8 bloques + Sandbox + 16 productos seed)
 - 6.4 Flujo Operativo (Catálogo, Pendientes, Vendidos, Devueltos, Compramos Equipo, Archivados)
 - 6.X Centinela (NUEVO · Salud Operacional)

BLOQUE II · Gestión del Cliente

 - 6.5 Comunidad URG
 - 6.6 Centro de Mensajería (Resend)

- 6.7 FAQ Hub
- BLOQUE III · Inteligencia y Seguridad
 - 6.8 Estadísticas e Informes (BI · 31 métricas)
 - 6.8.2 Profit & Commission Manager
 - 6.8.3 Report Builder (PDF + CSV)
 - 6.9 Black Box (CIV)
- BLOQUE IV · Configuración y Estructura
 - 6.10 Tributario, Financiero y Logístico
 - 6.11 Panel Maestro (API Keys, DNS, Admins, IA chat)
- 7. LÓGICA TÉCNICA TRANSVERSAL Y PREVENCIÓN DE ERRORES
(7.1 a 7.25 · sanitizadores, RLS, soft delete, performance, .htaccess, etc.)
- 8. BACKEND SERVER-SIDE (NUEVO)
 - 8.1 Edge Functions completas (13)
 - 8.2 Trigger pg_cron de limpieza
 - 8.3 Webhook MercadoPago · consumo atómico
 - 8.4 Cancelación cruzada BRE-B ↔ MP
- 9. PIPELINE CI/CD (NUEVO)
 - 9.1 GitHub Actions workflow
 - 9.2 FTP Deploy con clean-slate protegido
 - 9.3 Variables de entorno y secrets
 - 9.4 Rollback manual
- X. SISTEMA DE COOKIES (URG)
- X. DATOS LEGALES URG TECHNOLOGY
- APÉNDICE A · Changelog público (v1.0 → v24.18.26)
- APÉNDICE B · Estructura de carpetas src/
- APÉNDICE C · Diccionario de tablas Supabase

1. IDENTIDAD VISUAL Y LÓGICA DE PRECIOS • URG TECHNOLOGY

1.1 Paleta de Colores (Regla de Proporción)

La paleta se mantiene íntegra respecto a V1.1 y está declarada como CSS custom properties en `src/index.css` para que el cambio sea global y los componentes la consuman como `var(--urg-purple)` etc.

```
:root {
  /* 60% Neutro */
  --urg-bg: #FFFFFF; /* fondo blanco puro tienda pública */
  --urg-bg-soft: #F8F9FA; /* gris extra claro para tints alternos */
  --urg-bg-admin: #0B0B14; /* fondo del panel admin (dark) */

  /* 30% Primario */
  --urg-purple: #8A2BE2; /* morado URG • CTAs, enlaces activos, precio héroe */
  --urg-purple-soft: #B47BEE; /* hover/glow */
  --urg-purple-deep: #4F1880; /* gradientes profundos */

  /* 5% Negro/Texto */
  --urg-text: #0F0F14; /* texto principal */
  --urg-text-soft: #2D2D38; /* texto secundario */

  /* 5% Acentos psicológicos */
  --urg-green: #2ECC71; /* badges estado, WhatsApp, envío gratis */
  --urg-amber: #F5B400; /* rankings, medallas */
  --urg-red: #E11D48; /* vendido, errores, modal alguien más veloz */

  /* Precios */
  --urg-price-old: #95A5A6; /* precio ancla tachado */
  --urg-price-hero: var(--urg-purple);
  --urg-badge-bg: #F0E4FF; /* lavanda fondo cápsula %0FF */
  --urg-badge-text: #4F1880; /* morado oscuro texto cápsula */
}
```

Aplicación efectiva en el sitio:

- **60% Neutro (Fondo):** Blanco puro o gris extra claro (`#F8F9FA`) alternado por sección via clase utilitaria `urg-section-tint` (definida en `index.css`). Mantiene la interfaz limpia y profesional incluso después del rediseño masivo Stripe-style del panel admin.
- **30% Primario (Marca):** Morado URG `#8A2BE2` aplicado en:
 - Botones de acción principal (CTA): la clase `urg-cta` agrupa estilos compartidos.
 - Enlaces activos (`<NavLink>` con `aria-current="page"` recibe `text-[var(--urg-purple)]`).
 - Precio Héroe (precio final de venta): el componente `PriceDisplay.jsx` lo aplica con sombra `0 0 12px rgba(138,43,226,0.18)` para efecto neón sutil.
 - Glow del avatar de Super Administrador (color dorado en su caso, ver §6.2).
- **5% Negro:** texto base, encabezados de columnas en tablas admin, bordes sutiles `border-black/10` .
- **5% Acento (Psicología):**
 - Verde Neón `#2ECC71` → Badge "Estado" (Nuevo / Como Nuevo / Excelente / Bueno), botón flotante WhatsApp, badge "Envío Gratis Bogotá".
 - Ámbar/Oro `#F5B400` → bordes oro/plata/bronce en el podio del Top 3 vendidos (introducido en task #251), medallas de garantía en Ficha 3.
 - Rojo `#E11D48` → estado VENDIDO (sello centrado, no diagonal, introducido en v24.18.22), borde de tarjetas con error de stock, modal "¡Alguien fue más veloz!".

1.2 Arquitectura del Precio (Anclaje de Valor)

El componente `PriceDisplay.jsx` centraliza toda la lógica de presentación del precio. Recibe props `{ precio, precioOriginal, descuentoPct, tamano: "sm" | "md" | "lg" | "xl" }` y renderiza la jerarquía correcta. El admin no escribe HTML: solo introduce los valores numéricos en el Bloque 7 del Creador.

1. Precio "Ancla" (Original)

- **Color:** Gris medio `#95A5A6` (variable `--urg-price-old`).
- **Estilo:** Tachado (`line-through`), fuente delgada (`font-light`), 20% más pequeño que el precio héroe (`text-[0.80em]`).
- **Ubicación:** Arriba del precio héroe en mobile y en Ficha 1 vertical; a la izquierda en Ficha 2 y en Ficha 3 desktop.

2. Precio "Héroe" (URG Purple)

- **Color:** Morado vibrante `#8A2BE2`.
- **Fuente:** Inter (la fuente del sitio) en peso 700 (Bold): Montserrat se descartó en V1.5 para reducir descarga de webfonts adicionales; Inter cubre el rango 100-900.
- **Tamaño protagonista:** `clamp(1.8rem, 2.6vw, 2.6rem)` en Ficha 3, `clamp(1.4rem, 2vw, 1.9rem)` en Ficha 1.
- **Efecto:** Sombra sutil (glow morado claro): `text-shadow: 0 0 14px rgba(138,43,226,0.20)`. En modo "Aura ON" la sombra se intensifica al pulso de 3 s.

3. Badge de "Beneficio" (% OFF)

- **Color:** Fondo lavanda `#F0E4FF`, texto morado oscuro `#4F1880`.
- **Estilo:** Cápsula pequeña (`rounded-full px-2 py-0.5`) junto al precio héroe.
- **Texto:** "20% OFF" si `descuentoPct` está definido; cuando el descuento se introduce como monto fijo, el sistema lo convierte a porcentaje implícito y muestra el porcentaje (decisión tomada en task #253: usuarios escaneaban más rápido el `%`).

1.3 Reglas adicionales aprendidas en producción

- **Sin prefijos en los datos.** En todas las fichas, los valores van solos (`LA BESTIA DEL GAMING`, no `Nombre: LA BESTIA DEL GAMING`). Esta regla, originalmente del V1.1, se reforzó tras incidentes donde admins escribían "Nombre Marketing:" como parte del nombre, contaminando los `<h1>`.
- **Punto colombiano para miles.** El helper `format.js → formatCOP(valor)` siempre emite `$ 1.234.567` (sin centavos, separador `.`) y se usa de forma exclusiva: está prohibido `Intl.NumberFormat` ad-hoc en componentes.
- **IVA inline solo cuando aplica.** Si el producto supera 50 UVT y queda gravado, en la Ficha 3 aparece un sub-texto en gris `"IVA 19% incluido en el total"`. Si está excluido, aparece `"Excluido de IVA (Art. 424 E.T.)"`. El componente que pinta el sub-texto consulta el flag `producto.ivaModo` calculado en el back.
- **Sin sombra/glow en precios cuando el producto está VENDIDO.** El overlay rojo del estado VENDIDO desactiva las sombras de precio (el sello centrado introducido en v24.18.22 ya carga visualmente, las sombras del precio competirían).

2. PÁGINA HOME

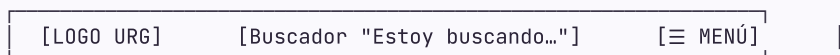
La página Home se compone, en orden de scroll, por: Cabecera (Header) + Hero compacto + Burbujas CTA + Oferta del Día + Más Vendidos + Compramos tu PC (block promocional) + Diseño Web Pro (banner que sale a damorus.com) + Nuestra Tienda + Footer. El Directorio Dinámico de SEO se eliminó del Home tras la task #73 (saturaba visualmente sin aportar conversión medible) y solo vive como bloque embebido en Footer y en algunas rutas de pestaña.

Cada bloque del Home tiene su componente bajo `src/components/home/*.jsx` y se compone en `src/pages/Home.jsx`. Las secciones alternan tinte de fondo mediante el wrapper `<div className="urg-section-tint">`, que aplica un sutil `background: #F8F9FA` para crear ritmo visual sin romper la unidad blanca de la tienda.

2.1 Cabecera (Header)

Componente: `src/components/Header.jsx`. Visible en TODAS las rutas públicas, montado dentro de `<Layout />`.

Estructura desktop (≥ md)



- **Logo URG:** `URG_HOMELOGO_2026.svg`, alto 36 px desktop, 28 px mobile. Click lleva a `/`. Si el usuario ya está en `/`, el click hace `scrollTo(0)` con `behavior: 'smooth'` en lugar de navegar (task #163). Esto evita el reflow innecesario y refuerza la sensación de "una sola página viva".
- **Buscador "Estoy buscando...":** componente `SearchAutocomplete.jsx`, ancho fluido (max 640 px desktop). Implementa autocompletado en tiempo real (Capa Primaria / Capa Técnica / Capa Etiquetas Invisibles) descrito en §3.1; comportamiento exhaustivo en esa sección.
- **Menú hamburguesa:** sidebar lateral derecho de 320 px que se desliza desde la derecha con Framer Motion (`x: '100%' → 0` con spring suave 240 ms). Contiene:
 1. Inicio
 2. Ofertas (cuando hay productos en oferta; visible siempre tras task #74)
 3. Catálogo
 4. Más vendidos
 5. Compramos tu PC
 6. Periféricos (auto-desactivado hasta que exista el primer producto con categoría periférica: task #16)
 7. Servicio Técnico (lleva al [WhatsApp de la tienda])
 8. Rastrear mi pedido
 9. Diseño Web Pro (etiqueta "Nuevo" verde; al hacer click abre `https://damorus.com` en pestaña nueva tras v24.18.26)
 10. **Divisor + sección "Servicio al cliente"** (task #159) con tres entradas adicionales: Política de privacidad · Compra protegida · Preguntas frecuentes, cada una con un emoji discreto para parsing visual rápido.

El menú móvil aplica `position: fixed; inset: 0; background: rgba(11,11,20,0.92)` con un blur de 12 px detrás (`backdrop-filter: blur(12px)`) para que el contenido del Home siga insinuándose por

debajo.

Estructura mobile (< md)

El logo queda compactado a 28 px y el buscador pasa debajo del header en una segunda barra delgada (task #27: la barra original era muy alta y robaba viewport vital del Hero). La nueva altura total del header móvil es 96 px (48 px logo + 48 px buscador) en vez de los 120 px originales.

2.2 Hero Section

Componente: `src/components/home/Hero.jsx`.

El Hero es **compacto y declarativo**, no busca llamar la atención por tamaño sino por presencia. Esa fue una decisión consciente del fundador: "el cliente está en un terminal de lujo que no necesita títulos grandes". Tras múltiples iteraciones (tasks #285 entre otras), el Hero quedó así:

Composición visual

- **Fondo en 5 capas** apiladas con `position: absolute`:
 1. **Gradiente base.** `linear-gradient(135deg, #2a0f5a 0%, #1a0840 35%, #0d0717 70%, #1a0f30 100%)`. Siempre visible aún si las demás capas fallan.
 2. **Foto blureada del admin.** Si `extra.fondo.imagen` está cargada en el bucket `urg.heroExtra`, se inyecta como `background-image` con `filter: blur(3px) saturate(1.1) brightness(0.8); transform: scale(1.05); opacity: 0.55; mix-blend-mode: overlay`. Es el "computador gaming blureado" de la V1.1, ahora editable.
 3. **Orbes neón flotantes.** Componente `<Neon0rb />` que pinta un círculo radial difuso de color configurable. El Hero lleva dos: morado `380 px` en `top: -10% / left: 62%` (opacidad 0.40) y cian `260 px` en `top: 55% / left: 8%` (opacidad 0.25).
 4. **Tinte decorativo del admin.** Cuando el admin elige un tinte (Aura URG / Crazy URG / Cyber-Black / Orgullo Patrio / Alerta Oferta), se inyecta como capa de gradiente sobre los orbes.
 5. **Oscurecedor central.** `radial-gradient(ellipse 70% 70% at 50% 50%, rgba(13,7,23,0.45) 0%, rgba(13,7,23,0.10) 60%, transparent 100%)` para garantizar contraste del texto blanco encima.
- **minHeight** del section: `min(24vh, 240px)` (task #285, antes 38vh/320px dejaba hueco vacío en mobile).
- **Padding vertical:** `py-6` en mobile (24 px) / `py-10` en desktop (40 px). Esto reemplazó los `py-10 / py-14` originales que generaban huecos visuales muertos.

Contenido textual

Tu futuro tecnológico comienza aquí.



Exclusivo para Bogotá:
Envío GRATIS y entrega 1-2 días.

- **Headline.** Texto blanco, `font-weight: 800`, `font-size: clamp(1.7rem, 3.2vw, 2.4rem)`, `letter-spacing: -0.02em`. Anima con Framer Motion `initial={{opacity:0, y:18}}` `animate={{opacity:1, y:0}}` con duración 0.7 s y ease cubic-bezier `[0.22, 1, 0.36, 1]`.
- **Sub-badge "Envío gratis Bogotá".** Componente personalizable (task #16): el admin escoge el icono (camioncito SVG por defecto), color de fondo (verde neón por defecto), color de texto

(auto/blanco/oscura, task #19) y texto literal. Aparece envuelto en una píldora con bordes redondeados completos.

- **Anclaje horizontal.** El bloque título + badge usa `flex-col items-center justify-center` (refactor en v24.18.17): antes con grid los items quedaban pegados arriba en lugar de centrados, y el badge podía desbalancear horizontalmente al título.

Tanto headline como badge son **modificables desde Panel Maestro > Bloque K (Banner del Home)**.

2.3 Burbujas CTA

Componente: `src/components/home/BurbujasCTA.jsx`. Scrollable horizontal **incluso en desktop** (task #23: antes en desktop se centraban estáticas y desperdiciaban whitespace).

Orden actual de las burbujas

1. **Ofertas:** icono porcentaje, fondo gradiente naranja-rojo, lleva a `/ofertas`.
2. **Catálogo:** icono grilla, lleva a `/catalogo`.
3. **Portátiles:** icono laptop, lleva a `/portatiles`.
4. **Escritorio:** icono torre, lleva a `/escritorio`.
5. **Compramos tu PC:** icono billete + flecha, lleva a `/compramos-tu-pc`.
6. **Diseño Web Pro:** icono globo, marca `external: true` (introducido en task post-v24.18.26), abre `https://damorus.com` en pestaña nueva.
7. **Periféricos:** icono mouse, visible solo si hay productos con categoría periférica activa.

Especificaciones técnicas

- **Tamaño:** 120 × 120 px en desktop, 96 × 96 px en mobile (reducido ligeramente para que entren 4 burbujas en pantalla de iPhone SE sin scroll).
- **Forma:** `border-radius: 50%` (círculo perfecto). Borde de 2 px en `var(--urg-purple)` con `box-shadow: 0 4px 18px rgba(138,43,226,0.16)`.
- **Glassmorphism:** `background: rgba(255,255,255,0.62); backdrop-filter: blur(8px) saturate(140%)`.
- **Estado activo:** hover ⇒ `transform: translateY(-4px) scale(1.04)`, `box-shadow: 0 8px 26px rgba(138,43,226,0.30)`.
- **Scroll horizontal mobile:** `overflow-x: auto; scroll-snap-type: x mandatory`. Cada burbuja con `scroll-snap-align: center` para que se centren al deslizar.

2.4 Oferta del Día (Ficha #1 horizontal)

Componente: `src/components/home/OfertaDelDia.jsx`. Es la pieza con mayor inversión de UX del Home y ha pasado por 12 iteraciones registradas (tasks #246, #251, #257, #273, #274, #275, #276 entre otras).

Identidad visual y contenedor

- **Fondo:** card profunda negra (`background: linear-gradient(135deg, #07070A 0%, #1a0f30 100%)`).
- **Borde:** efecto neón sutil rosa-morado-cian con animación de pulso `urg-pulse` (definida en `index.css` keyframes: opacity oscila 0.6 ↔ 1 cada 2.8 s).
- **Eyebrow:** píldora gigante en gradiente `linear-gradient(90deg, #ff6b6b 0%, var(--urg-purple) 60%, #B47BEE 100%)` con texto blanco `font-black uppercase letter-spacing 0.20em`, ícono 🔥 a la

izquierda, sombra `0 4px 14px rgba(255,107,107,0.40)` . Texto: `OFERTA DEL DÍA` .

- **Countdown** en formato `HH:MM:SS` que reinicia a las **8:00 AM Hora Colombia (UTC-5)**. La lógica de zona horaria está implementada con offset fijo dentro del componente (líneas 53-78 del archivo); se evitó `Intl.DateTimeFormat` para no depender del navegador del cliente cuando se construye el `Date` final.

Layout y comportamiento responsivo

Desktop ($\geq$ md)

- Diseño **2 columnas asimétricas** en grilla 12: imagen ocupa columnas 1-7 ($\approx 58\%$), panel de info ocupa columnas 7-12 ($\approx 42\%$). La ratio se ajustó desde 60/40 a 58/42 en task #257 cuando se introdujo el sistema de encuadre con cropper.
- **Galería automática** con `setInterval(4000)` . Si solo hay 1 imagen, el intervalo no se monta (early return en `useEffect`).
- **Barras de carga** (loading bars) debajo de la imagen: barras delgadas de 3 px de alto que se llenan de morado en 4 s (`@keyframes barFill`). Cuando una barra está al 100% la imagen muta a la siguiente con `AnimatePresence` (fade 350 ms).
- **CTA principal** en la esquina inferior derecha del panel: "¡COMPRAR AHORA!" con efecto glow recorrido (`@keyframes urg-glow-sweep`) cada 3 segundos. Hover $\Rightarrow$ `scale: 1.05` y morado más saturado. Click abre directamente Ficha 3 del producto en `/productos/{slug}` (la decisión histórica de saltar al checkout se invirtió en task #239: el cliente debe ver la ficha antes de comprar para reducir devoluciones).

Mobile ($<$ md)

- Diseño **apilamiento vertical**, imagen arriba ocupando 16:9 (`aspect-ratio: 16/9`), panel info debajo.
- **Carrusel táctil** con `overflow-x: auto; scroll-snap-type: x mandatory` cuando hay > 1 imagen. Indicadores (dots) de 8 px en mobile (vs 4 px en desktop) para tap target accesible.
- **Peek-a-boo**: la card NO cubre toda la pantalla: deja $\sim 12\%$ del bloque "Más Vendidos" siguiente asomando por debajo (task original V1.1, conservada).
- **Botón "Comprar Ahora"**: altura 48 px (touch target estándar Apple HIG), ancho 100% del panel. Posicionado debajo del precio para que el pulgar derecho lo alcance sin gesto en cuna.

Componentes de la ficha (en orden de aparición)

1. **Nombre Marketing**: negrita máx 2 líneas. Si excede, se trunca con `text-overflow: ellipsis` y aparece tooltip en hover desktop.
2. **Marca + Modelo**: una sola línea, color `text-zinc-400` , sin etiqueta "Marca:" prefija (regla §1.3).
3. **Precio Original** tachado en gris medio.
4. **Precio Oferta** en morado URG con tamaño grande.
5. **Badge OFF** cápsula lavanda con `XX% OFF` .
6. **Envío**: chip verde "Envío Gratis Bogotá" con icono camión SVG.
7. **CTA "¡COMPRAR AHORA!"** descrito arriba.
8. **Click en el body** (no en el CTA): todo el contenedor es un `<Link>` a la Ficha 3.

Lógica de Estados y Control de Inventario

Estado VENDIDO (escasez psicológica)

Cuando el `stockDisponible` llega a 0 (calculado en tiempo real por `lib/orders.js` → `getStockDisponibleAhora()`, ver §5.4), el componente muta:

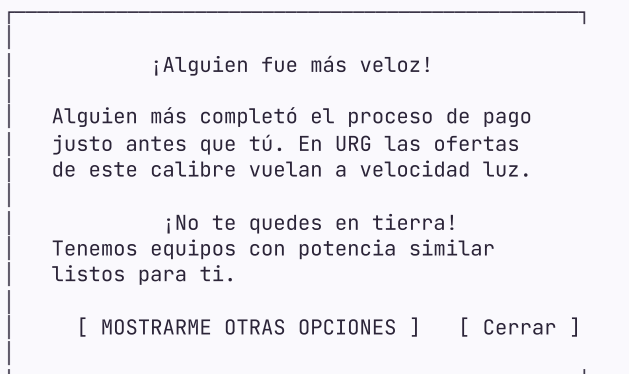
- **Visual:** imagen no se vuelve blanco-y-negro (decisión revertida en task #243: la imagen real vende más incluso vendida); en su lugar, se superpone overlay rojo translúcido `rgba(225,29,72,0.55)` con sello centrado.
- **Sello centrado** (no diagonal) `VENDIDO` (task v24.18.22): `font-size: clamp(1rem, 1.35rem, 2rem)`, peso 900, letter-spacing 0.10em, fondo blanco redondeado, sombra `0 6px 18px rgba(0,0,0,0.55)`.
- **Mensaje central debajo del sello:** `MISIÓN CUMPLIDA: VENDIDO` con sub-texto `Regresa mañana para más ofertas de locura`.
- **CTA "¡COMPRAR AHORA!" se reemplaza** por botón "Avísame cuando vuelva": sistema de suscripción introducido en task post-v24.18.22, almacena email + ID producto en `lib/avisame.js`. Al re-publicar el producto con stock > 0 se dispara un email Resend y aparece un componente flotante `<AvisameNotificacion />` en la esquina inferior izquierda del sitio para usuarios suscritos.
- **Bloqueo:** clicks en la ficha se ignoran (`pointer-events: none` excepto el botón Avísame y el WhatsApp salva-venta).

Control de Colisión (Race condition)

El sistema de reservas atómicas server-side garantiza que dos compras simultáneas del mismo producto único nunca puedan ambas finalizar. Detalle completo en §5.4: aquí basta saber que la Oferta del Día se beneficia de la misma lógica.

Modal "¡Alguien fue más veloz!"

Si el cliente está pagando por BRE-B y otro usuario consume el stock con MercadoPago en el mismo instante (o viceversa cuando admin aprueba un BRE-B mientras otro cliente está en MP), se dispara el modal `<StockAgotadoModal />` definido en `src/pages/Confirmacion.jsx`:



El título y texto principal van en **rojo** (`color: var(--urg-red)`, decisión del fundador para reforzar urgencia). El CTA abre `/catalogo` y mantiene el filtro de categoría que tenía el producto perdido aplicado por defecto. El modal se escucha vía evento `FALLIDO_EXP` que el webhook dispara cuando ejecuta `cancelarHuerfanosPostConsumo()`: detalles en §5.5.

2.5 Más Vendidos (Ficha #1 vertical)

Componente: `src/components/home/MasVendidos.jsx`. Renderiza el TOP 1 de cada una de las 4 categorías madre.

Categorías y subtítulos

1. **Gaming:** *Domina el campo de batalla*, enfoque en FPS y gráficos.
2. **Hogar / Estudio:** *Tu compañero de tareas infalible*, enfocado en fiabilidad, peso ligero y precio.
3. **Creadores:** *Potencia sin límites a tu creatividad*, enfoque en RAM y procesamiento.
4. **Empresarial:** *Para la construcción de tu futuro*, soluciones profesionales.

Layout

- **Desktop:** grid `grid-cols-4 gap-6`. Cada columna es una Ficha #1 vertical (imagen arriba, info debajo, sin botón visible: toda la card es área clickable, regla psicología "click en cualquier parte").
- **Mobile / tablet:** scroll horizontal con snap (`overflow-x-auto snap-x snap-mandatory`). Se ve la primera columna completa y ~15% de la segunda asomando. Indicador con 4 dots de 8 px debajo.
- **"Ver catálogo completo →" link** debajo (task #283) visible solo en mobile y tablet: lleva a `/mas-vendidos` (página dedicada con los 12 puestos completos: 4 cat × 3 puestos = 12 productos).

Efectos de hover (desktop)

- **Traducción:** `translateY(-5px)`.
- **Sombra:** aparece sombra suave `0 12px 24px rgba(0,0,0,0.08)`.
- **Resaltado de texto:** el nombre del PC cambia al **color de la categoría**:
 - Gaming → morado URG.
 - Hogar/Estudio → cian (`#00CCFF`).
 - Creadores → verde neón.
 - Empresarial → blanco/plata (`#E5E7EB`).

Badge ODD-en-MasVendidos

Si el producto que actualmente es Oferta del Día pertenece a una de las categorías mostradas, su ficha en MasVendidos recibe un badge fuego en lugar del badge dorado de #1 (task v24.18.7, fix en v24.18.13). Esto evita que el cliente vea el mismo producto dos veces seguidas sin contexto y refuerza la urgencia de la oferta.

2.6 Compramos tu PC (bloque promocional del Home)

Componente: `src/components/home/CompramosTuPC.jsx`. Es solo el **bloque que invita** a usar el formulario; el formulario real vive en `src/pages/CompramosTuPC.jsx` (ruta `/compramos-tu-pc`).

Contenido del bloque

COMPRAMOS TU PC: DINERO AL INSTANTE O RENOVACIÓN.

Pagamos el mejor precio por tu equipo. Recibe efectivo para lo que necesites o úsalo como parte de pago. ¡Sin complicaciones!

- | | | |
|---|-------------------|-------------------------|
| ⚡ | Valuación Express | en menos de 5 minutos |
| 😌 | Cero Estrés | solo sube fotos |
| 📁 | Sin Obligación | tú decides la modalidad |
| 📁 | Bono de Regalo | 5% de descuento extra |

[VENDER MI PC AHORA →]

El CTA lleva a `/compramos-tu-pc` que es ya el formulario completo en 9 pasos. La página dedicada se documenta en §3.7.

2.7 Diseño Web Pro (banner promocional → damorus.com)

Componente: `src/components/home/DisenoWebPro.jsx`. En la V1.1 este bloque era una sección de venta extensa con 3 planes y CTAs a una pestaña `/diseno-web`. Tras la decisión estratégica de v24.18.26, **todo el feature de Diseño Web Pro migró al sitio externo `damorus.com`**, y en URG quedó únicamente este banner promocional.

Contenido actual

¿Tu marca aún no vende en internet?
Convertimos tu negocio en un activo digital.

[VISITAR DAMORUS.COM]

Diseño web profesional · IA · Pagos · BI

- El CTA es un `<a href="https://damorus.com" target="_blank" rel="noopener noreferrer">`. No tiene flecha ↗ (eliminada en task #144: generaba confusión sobre si el botón salía del sitio o no).
- El bloque tiene tema oscuro embebido (`background: linear-gradient(135deg, #0B0B14, #1a0840)`) para diferenciarse visualmente del resto del Home blanco, refuerza el "salto" a otra marca.
- Banner controlado por **toggle** en Panel Maestro > Bloque L: si el admin lo desactiva, el bloque desaparece del Home sin recompilar.

Ruta `/diseno-web` legacy

Para no romper enlaces compartidos antes de la migración, la ruta `/diseno-web` quedó como redirect:

```
<Route path="diseno-web" element={<ExternalRedirect to="https://www.damorus.com" />} />
```

El componente `ExternalRedirect` ejecuta `window.location.replace(to)` en `useEffect`. Esto preserva el SEO de enlaces antiguos.

2.8 Nuestra Tienda (catálogo en Home)

Componente: `src/components/home/NuestraTienda.jsx`. Muestra una grilla compacta de productos con filtros tipo chip.

Layout

- **Chips de categoría madre** justo debajo del título:
 1. ⚡ Todos (*por defecto*)
 2. 🎮 Mundo Gamer
 3. 🎨 Creadores
 4. 📁 Empresarial
 5. 🏠 Hogar / Oficina
 6. 🔥 En oferta
- Scroll horizontal en mobile, distribución centrada en desktop.
- Al tocar un chip, la grilla de abajo se refiltra instantáneamente con animación Framer Motion `layout` + `fade`.

Grilla

- **Desktop:** 4 columnas (`grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6`).
- **Mobile:** 2 columnas para que el usuario vea más opciones con menos scroll. En tablet también 2 columnas, en mobile pequeño (xs) cae a 1 columna full-width.
- **Mobile cap:** 6 productos máximos (task #22). Si hay más, se muestra el botón "Cargar más equipos" que reemplaza la carga inicial; en desktop arranca con 8 o 12 productos.

Contador de inventario dinámico

Debajo del botón de expansión aparece el texto pequeño: *Mostrando 12 de 48 equipos disponibles*. El número total se calcula con `getCatalogoActivo().filter(p => p.stock > 0).length`.

Click en producto

Cada ficha es un `<Link to="/productos/{slug}">`. Mantiene la posición de scroll del Home cuando el usuario regresa (gracias a `<ScrollRestoration>` de react-router v7).

2.9 Directorio Dinámico de SEO

Decisión actual: este bloque NO aparece en el Home (eliminado en task #73 por considerarse ruidoso). Sigue activo, sin embargo, en:

1. El **Footer** del sitio (todas las páginas), como bloque condensado horizontal "EXPLORA NUESTRO ECOSISTEMA".
2. Las **rutas de pestaña dedicada** (`/ofertas`, `/portatiles`, `/escritorio`, `/perifericos`) donde aparece justo encima del Footer para reforzar relevancia SEO.

Componente: `src/components/home/DirectorioSEO.jsx`. La lógica de generación es la misma de V1.1: escanea categorías activas y marcas distintas en `adminProducts.published`, las dedupea y las renderiza como `[ Nombre ]` clickables.

2.10 Botón Master Ascent (CTA de Retorno Infinito)

Componente: `src/components/MasterAscent.jsx`. Se monta en `Layout.jsx` entre `<Outlet />` y `<Footer />`, asegurando que aparezca debajo de TODA página pública (no solo del Home).

Especificaciones

- **Ancho:** 100% (full-width).
- **Altura:** 48 px.
- **Fondo:** gris muy oscuro `#16161f` con borde superior y inferior de 1 px `rgba(138,43,226,0.18)`.
- **Hover:** background transiciona a `var(--urg-purple)` y el texto se vuelve negro mate.
- **Texto:** centrado, mayúsculas, `font-weight: 700`, letter-spacing 0.10em, `INICIO DE PÁGINA`.
- **Acción:** `window.scrollTo({ top: 0, behavior: 'smooth' })`. Si el usuario está en `/`, ejecuta también un `history.pushState({}, '', '/')` para limpiar el hash si lo hubiera.
- **Visibilidad condicional:** se oculta cuando el usuario está en `/[panel]`, `/[panel]/*` o cuando el scroll del documento es ≤ 600 px (no tiene sentido ofrecer "subir" cuando ya estás arriba).

2.11 FOOTER

Componente: `src/components/Footer.jsx`. Tiene dos versiones renderizadas condicionalmente:

FooterMobile (visible solo `< md`) optimizado en orden lógico para pulgar, y **el grid clásico de 4 columnas** (visible `≥ md`).

2.11.1 Estética compartida

- **Atmósfera visual:** `linear-gradient(180deg, var(--color-urg-footer-top) 0%, #1a0840 25%, #0a0014 75%, var(--color-urg-footer-bottom) 100%)`. Las variables top/bottom permiten al admin ajustar la intensidad del púrpura desde Bloque L.
- **Coloración de texto:** todos los textos en lavanda muy clara `#d8c2ff` con opacidad reducida para que el contraste con el fondo no sea agresivo.
- **Acabado:** minimalista pero inmersivo: el footer es una extensión premium de la interfaz, no un bloque pegado.

2.11.2 Versión Desktop (≥ md) · 4 columnas

IDENTIDAD	CENTRO DE CONFIANZA	CONTACTO	ENCUÉNTRANOS
[Logo URG en blanco puro]	• Rastrea tu pedido • Pagos y garantía • Política de privac. • Preguntas frecuentes	Soporte General • [dirección] • [correo] • [WhatsApp de la tienda] Especialistas • Hardware: [censurado] • Software: [censurado] • Diseño Web: Alejandro@damorus.com	[Mapa iframe Google Maps centrado en la dirección] Pulsar el mapa abre maps en pestaña nueva Logos de ACEPTAMOS abajo (strip 1×6 static)
URG TECHNOLOGY SAS	★ Califica nuestro servicio (ver pago)		
Tagline...			
Horarios L-V: 9:15-6 S: 9:15-2 D: Cerrado			
Tienda aprobada y vigilada por: [DIAN logo grayscale] [CCB logo grayscale]			
© 2026 URG TECHNOLOGY SAS · Potenciado por Damorus.com [Logo Damorus blanco minúsculo]			

Columna 1 · Identidad

- **Logo URG en blanco puro.** Se aplica `filter: brightness(0) invert(1)` al SVG original `URG_FOOTER_2026.svg` para forzar blanco mate sin necesidad de un asset adicional. Hover quita el filtro y muestra los colores corporativos originales (mismo patrón usado en DIAN/CCB).
- **Razón social:** `URG TECHNOLOGY SAS.`
- **Tagline:** *Empresa colombiana líder en tecnología reacondicionada y desarrollo de ecosistemas web.*
- **Horarios:**
 - L-V: 9:15 am a 6:00 pm
 - S: 9:15 am a 2:00 pm
 - D-Festivos: Cerrado

Columna 2 · Centro de Confianza

- Lista vertical con bullets morados pequeños.
- "Califica nuestro servicio" lleva el bullet **verde** (en lugar de morado), seguido de una estrella ★ y un icono externo, para resaltar sobre el resto.
- Hipervínculo a `[enlace de Google Maps]` (editable desde Panel Maestro § Master de Contacto).

Columna 3 · Contacto

- **Soporte General** con dirección, email y WhatsApp principal.
- **Especialistas (Hardware / Software / Diseño Web)** con WhatsApp directo (el de Diseño Web cambió a email `Alejandro@damorus.com` tras la migración v24.18.26).
- Cada teléfono usa `href="tel: ..."` con el número **sanitizado** (sin espacios ni paréntesis) por el helper `sanitizePhone()` de `lib/format.js`.
- Cada WhatsApp usa `https://wa.me/{telefono}?text=...` con `encodeURIComponent()` aplicado al mensaje.
- Editable desde Panel Maestro § Master de Contacto. Permite agregar filas no principales para más especialistas.

Columna 4 · Encuéntranos

- **Mapa Google** embebido como iframe. El embed string se lee de `contact.direccion?.iframeEmbed` (editable en Bloque L): por defecto `[iframe de Google Maps de la sede]`.
- Al hacer click en el iframe, el wrap `<a>` exterior lo intercepta y abre `[enlace de Google Maps]` en pestaña nueva.
- **Strip "Aceptamos"** debajo del mapa: 1 × 6 logos estáticos (Mercado Pago, BRE-B, Visa, Mastercard, American Express, Nequi). Tras múltiples iteraciones (#29, #31, #36, #40, #42, #267, #268) se determinó: cada logo tiene su escala ajustada por `scale` prop para compensar diferencias de whitespace interno, Mercado Pago `1.52×`, Mastercard `1.40×`, American Express `1.30×`, Visa/BRE-B/Nequi `1×`. El strip es **estático** (sin marquee, decisión #40: el marquee distraía y aportaba poco).

2.11.3 Trust Bar

Franja de "semi-lujo" justo debajo del grid:

- **Fondo:** gris muy oscuro con sutil textura `noise.svg` superpuesta.
- **Texto:** *Tienda aprobada y vigilada por:* en `font-medium` color lavanda.
- **Logos:** `DIAN_LOGO_2020.svg` y `CCB_LOGO_2026.svg`. Por defecto en escala de grises con opacity 0.75; en hover recobran color original (filter: drop-shadow añadida).

2.11.4 Cierre

- **Texto:** © 2026 URG TECHNOLOGY SAS: Potenciado por Damorus.com (el "Damorus.com" es un hipervínculo invisible que lleva a `https://www.damorus.com`).
- **Logo Damorus:** `DAMORUS_BLANCO.svg` debajo del texto. Estado natural opacity 70%; hover sube a 100% y aplica `filter: drop-shadow(0 0 8px #ffffff)` para simular que el fénix brilla.

2.11.5 Versión Mobile (FooterMobile)

Componente interno `FooterMobile` (definido en el mismo archivo `Footer.jsx`). Reorganiza para pulgar:

A. Strip de pagos auto-marquee (mantiene marquee solo en mobile) B. 2 botones grandes de Confianza prioritarios: **Rastrea tu pedido** + **Califica nuestro servicio** C. Horarios compactos D. Encuéntranos (mapa pequeño + dirección) E. Acordeón "Contacto + Especialistas" colapsado por defecto F. Acordeón "Más enlaces" (privacidad, FAQ, etc.) colapsado G. Branding mínimo con logo y copyright

El Trust Bar (DIAN/CCB) y el cierre con Damorus se mantienen iguales y aparecen debajo (compartidos).

La sección "Aceptamos" en mobile fue **renombrada y rediseñada** en task #21: ahora el strip vive en el bloque A (encima de todo) y el texto "Aceptamos" se eliminó porque era redundante con el contexto del Footer.

2.12 BOTÓN DE WHATSAPP (FAB flotante)

Componente: `src/components/WhatsappFloat.jsx`. Visible en TODAS las rutas públicas excepto cuando el usuario está en `/checkout` (evita interferir con CTAs de pago).

- **Posición:** `position: fixed; right: 20px; bottom: 20px; z-index: 50`.
- **Botón:** circular 56 × 56 px, fondo verde WhatsApp `#25D366` con sombra `0 6px 16px rgba(37, 211, 102, 0.45)`.
- **Pop-up auto-aparición:** después de 4 segundos en la página aparece globo blanco a la izquierda del botón "¿Tienes dudas? Contáctanos." Si el usuario lo descarta (X), se guarda en localStorage la

preferencia por 24h para no volver a mostrarlo.

- **Click:** abre [https://wa.me/\[WhatsApp de la tienda\]](https://wa.me/[WhatsApp de la tienda]) (número principal, sanitizado).

El WhatsApp Float **NO** se muestra en [/\[panel\]/*](#) (panel admin). El admin tiene sus propios canales internos.

Con la sección 2 cerrada, se documenta a continuación cada pestaña dedicada del sitio (rutas que no son el Home), incluyendo las decisiones de v24.18 sobre [/compramos-tu-pc](#) , [/preguntas-frecuentes](#) (canónica) y la migración de [/diseno-web](#) .

3. PESTAÑAS / RUTAS DEDICADAS

A diferencia de la V1.1 donde las "burbujas" se concibieron como overlays superpuestos al Home, en la implementación final cada pestaña se resolvió como **ruta SPA propia** dentro del Layout público. La razón fue pragmática: la accesibilidad, el deep-linking, el botón "atrás" del navegador y la indexación SEO funcionan mejor cuando cada destino es una URL real, no un overlay temporal. La sensación de "burbuja sin recarga" se preserva porque toda la navegación interna es client-side (react-router) sin recargas reales del documento.

Listado completo de rutas activas (declaradas en `src/App.jsx`):

```
/                → <Home />
/catalogo        → <Catalogo />
/productos/:slug → <ProductoDetalle />
/carrito         → <Carrito />
/checkout        → <Checkout />
/confirmacion/:idPedido → <Confirmacion />
/privacidad      → <Privacidad />
/compra-prottegida → <CompraProtegida />
/compramos-tu-pc  → <CompramosTuPC />
/rastrear-pedido  → <RastrearPedido />
/preguntas-frecuentes → <FAQ /> (canónica)
/faq             → <Navigate to="/preguntas-frecuentes" replace />
/diseno-web      → <ExternalRedirect to="https://www.damorus.com" />
/ofertas         → <Ofertas /> (alias /catalogo?filtro=ofertas)
/portatiles      → <Portatiles /> (alias /catalogo?cat=portatil)
/escritorio      → <Escritorio /> (alias /catalogo?cat=escritorio)
/perifericos     → <Perifericos /> (alias /catalogo?cat=periferico)
/mas-vendidos    → <MasVendidosPage />
/*              → <NotFound />
```

3.1 /catalogo · Ecosistema de Búsqueda y Catálogo Maestro

Componente: `src/pages/Catalogo.jsx`. Es el módulo más complejo del front público; reemplaza la idea original de "burbuja sobre el Home" por una página completa con sidebar de filtros desktop / barra superior mobile, manteniendo la sensación SPA.

3.1.1 Arquitectura de Indexación Multidimensional

Implementada en `src/lib/smartSearch.js`. El algoritmo escanea tres capas en orden de prioridad:

1. **Capa Primaria (Título y Marca).** Match directo sobre `nombreMarketing` y `marca`. Peso 3×.
2. **Capa Técnica (Atributos).** Match en `procesador`, `ram`, `gpu`, `almacenamiento`. Peso 2×.
3. **Capa de Etiquetas Invisibles.** El admin puede asignar tags ocultos (campo `tagsBusqueda: string[]` en la ficha). Peso 1×. Ejemplo: producto con `nombreMarketing="LA BESTIA DEL GAMING"` y `tagsBusqueda=["arquitectura", "render", "cad"]` aparece al buscar "Renderizado" aunque la palabra no esté en su nombre.

El algoritmo es **acent-insensitive**

(`String.prototype.normalize("NFD").replace(/p{Diacritic}/gu, "")`) y **case-insensitive**. Maneja errores ortográficos leves vía Levenshtein-distance ≤ 2 (función `fuzzyMatch` en `smartSearch.js`).

3.1.2 Autocomplete Predictivo

- **Placeholder:** "Estoy buscando..." (texto en gris `#9ca3af`). Al focus se borra (`onFocus={e ⇒ e.target.placeholder = ''}`) y restaurar en blur).

- **Activación:** desde el tercer carácter (`if (query.length < 3) return`).
- **Tipos de sugerencias mostradas** en menú flotante (max 5):
 - Sugerencia de Término: "Lenovo" (lleva a `/catalogo?q=Lenovo`).
 - Sugerencia de Producto Específico: "Lenovo Legion Slim 5" (lleva a `/productos/{slug}`).
 - Sugerencia de Categoría: "Ver todos los Portátiles Lenovo" (lleva a `/catalogo?cat=portatil&marca=Lenovo`).
- **Persistencia local:** las últimas 3 búsquedas del usuario se guardan en `localStorage['urg.searchHistory']` (módulo `src/lib/searchHistory.js`). Aparecen al hacer focus con campo vacío bajo el eyebrow "Últimas búsquedas".
- **Top Búsquedas Globales** (Weekly Trends): se calculan a partir de la tabla `bi_searches` filtrando por últimos 7 días y agrupando por término normalizado. Aparecen bajo el eyebrow "Tendencias". Si el usuario escribe una letra, el sistema **prioriza** las globales que empiezan por esa letra (en lugar de mostrar las globales fijas).

3.1.3 Smart Search · Zero Results

Si la búsqueda no encuentra coincidencia, **queda estrictamente prohibido** mostrar una página vacía. La interfaz muestra:

```
No encontramos una coincidencia exacta para "macbook 2010",
pero nuestros especialistas sugieren estos sistemas de alto rendimiento:

[Card 1]  [Card 2]  [Card 3]
(3 equipos del mismo rango de precio o categoría con stock)

¿No es lo que buscabas?
[ HABLAR CON UN ASESOR POR WHATSAPP ]
```

La selección de las 3 alternativas usa: misma `categoría` → mismo `rango_precio` ($\pm 25\%$) → orden por `stockReal` desc. Si no hay del mismo rango de precio, se relaja a misma categoría y se mantiene el orden por ranking.

3.1.4 Cabecera Táctica (Sticky)

Sticky horizontal con `position: sticky; top: 96px` (debajo del header). Contiene:

- **Breadcrumb dinámico:** *Inicio > Catálogo > Portátiles > Lenovo*.
- **Contador vital:** "Mostrando 1-12 de 48 configuraciones encontradas para 'Lenovo'".
- **Selector de orden** (dropdown):
 1. Relevancia (Destacados)
 2. Más recientes (Default)
 3. Precio: Menor a Mayor
 4. Precio: Mayor a Menor
 5. Más vendidos (Ranking #1, #2, #3)
 6. Estado de Excelencia (Nuevo > Como Nuevo > Excelente)

3.1.5 Sistema de Navegación Visual Progresiva

Desktop ≥ lg. Sidebar persistente izquierda (320 px) con dos capas:

- **Capa Primaria · Burbujas Espejo (¿Qué es?):** lista vertical de círculos con icono.
 - [⚡ Todos]

- [ Portátiles]
- [ Escritorio]
- [ Periféricos]
- Glassmorphism + borde neón cuando activa.
- Lógica excluyente (al seleccionar una, las demás se desactivan).
- **Capa Secundaria · Chips de Personalidad (¿Para qué lo voy a usar?):** aparece tras seleccionar una primaria.
 - [MUNDO GAMER] [CREADORES] [EMPRESARIAL] [HOGAR] [OFERTAS]
 - Píldoras rectangulares acumulativas (puedes combinar Portátiles + Mundo Gamer + Hogar).

Mobile / Tablet. Las burbujas y chips se trasladan a la parte superior en dos filas compactas. El usuario navega con scroll horizontal de pulgar. Existe además un botón "FILTROS" (estilo ghost con flecha animada y estado activo) que despliega un modal con filtros avanzados: task #24.

3.1.6 Filtros avanzados (sidebar admin-generado)

Filtros prioritarios por defecto, todos dinámicos sobre el inventario activo:

- **Marca** (auto-generada).
- **Procesador** (Gama Ultra i7/Ryzen 7+, Gama Media i5/Ryzen 5, Gama Entrada i3/Celeron).
- **RAM** (8GB, 16GB, 32GB+).
- **Almacenamiento** (128, 256, 512, 1TB, 2TB+).
- **Estado** (Nuevo, Como Nuevo, Excelente, Bueno).
- **GPU** (Gráficos Dedicados, Gráficos Integrados).
- **Tamaño** (Pequeño / Mediano / Grande / Muy Grande).
- **Peso** (Peso Pluma / Liviano / Moderado / Pesado).

Cada filtro se renderiza como acordeón colapsable. Chips de filtro activo aparecen debajo del header del catálogo con una X para eliminarlos individualmente. Botón "**Limpiar filtros**" (icono basura) en la esquina superior derecha del sidebar.

3.1.7 Grilla y carga

- **Distribución:** Desktop 6 columnas en `xl`, 4 en `lg`, 3 en `md`, 2 en mobile.
- **Carga incremental:** muestra inicialmente 18 productos. Al llegar al final del scroll, un `IntersectionObserver` dispara `loadMore()` que añade otros 16. También hay botón explícito "CARGAR MÁS EQUIPOS" para usuarios que prefieren control manual.
- **Memorización de posición:** al regresar de Ficha 3 vía botón "Atrás", el catálogo restablece scroll y filtros (vía `sessionStorage['urg.catalogo.estado']`).

3.1.8 Cierre de búsqueda

Al final de la lista hay un bloque de retención:

¿Necesitas una configuración a medida?
 [Centro de Preguntas Frecuentes] [Contáctanos por WhatsApp]

3.2 /productos/:slug · Ficha 3 (Terminal de Producto)

Documentada exhaustivamente en §4.3. Aquí solo se documenta el routing.

- El `slug` se genera por el back al crear la ficha (kebab-case, sin tildes, sin caracteres especiales).
- Si el slug ya existe, se añade sufijo numérico (`macbook-pro-m2` , `macbook-pro-m2-1`).
- Si el `slug` solicitado no existe en `adminProducts.published` , el componente muestra brevemente skeleton y redirige a `/catalogo` con aviso flotante "Producto no encontrado o agotado" (task v18.5).
- Implementa `<meta>` dinámicos (title, description, og:image, og:url, twitter:card, JSON-LD Product) inyectados con `useEffect` cuando el producto carga.

3.3 /carrito

Componente: `src/pages/Carrito.jsx` . Lista los productos agregados con la lógica del `CartProvider` (context en `src/main.jsx`).

- **Línea por producto** con imagen (80×80), nombre, color/tamaño elegido si aplica, precio unitario, cantidad (input -/+), subtotal, botón eliminar (X).
- **Validación de stock en tiempo real:** al cargar el carrito y cada 30 s, recalcula `getStockDisponibleAhora(productoId)` y si el stock cayó a 0 marca la línea en rojo con texto "Ya no disponible: eliminar para continuar". Bloquea el botón "Pagar" hasta que el cliente elimine la línea.
- **Cupón:** input "Código de regalo" + botón "Añadir". Si es válido, aparece como línea negativa en el resumen.
- **Strip PagosStrip** (1×6 estático) al final, recordando al cliente que MercadoPago + BRE-B + Visa + MC + Amex + Nequi son aceptados.
- **Botón CTA "Continuar al pago"** lleva a `/checkout` .

3.4 /checkout

Componente: `src/pages/Checkout.jsx` . Es la página que implementa todo lo descrito en §5 (Apartado de Pagos). Aquí solo se documenta que es la URL canónica de pago. Si el usuario llega sin productos en carrito, redirige a `/catalogo` con mensaje.

3.5 /confirmacion/:idPedido

Componente: `src/pages/Confirmacion.jsx` . Es donde aterriza el cliente:

- **Tras pago aprobado MercadoPago:** vía retorno automático con `?status=approved&payment_id=...` . El componente confirma con la Edge Function `mp-webhook` que la transacción está aprobada y muestra "¡Pago Confirmado!".
- **Tras subir comprobante BRE-B:** muestra "¡Comprobante recibido con éxito!" con countdown estimado de validación (48 h).
- **Tras fallo / expiración:** muestra el modal `<StockAgotadoModal />` si la causa fue overselling, o la pantalla de "Pago no procesado" con opciones de reintentar.

Implementa el listener del evento custom `urg:fallido-exp` que dispara el webhook cuando ejecuta `cancelarHuerfanosPostConsumo()` , abriendo el modal "¡Alguien fue más veloz!".

3.6 /preguntas-frecuentes (canónica) + /faq (alias)

Componente: `src/pages/FAQ.jsx` . La V1.1 nombraba la ruta `/faq` solamente; en v24.18.16 se promovió `/preguntas-frecuentes` como canónica (mejor SEO LATAM) y `/faq` quedó como `<Navigate replace />` .

Renderiza el FAQ Hub categorizado por contexto **"FAQ · URG Technology"** (filtrado por contexto en `faqHub.js → getFaqItems({ contexto: 'tienda' })`). Las preguntas del contexto "Diseño Web Pro" YA NO se muestran aquí, ya que esa parte vive en damorus.com tras la migración v24.18.26.

Características UX:

- **Filtro de categorías:** chips horizontales con scroll en mobile, fila estática en desktop.
- **Acordeones expandibles** con animación de altura (Framer Motion `layout`). Solo uno abierto a la vez (acordeón controlado).
- **Buscador interno** que filtra por título y descripción.
- **Auto-expand on landing:** si el usuario llega con `?q=garantia`, el sistema busca el ítem y lo expande automáticamente, scrolleando a él.

3.7 /compramos-tu-pc

Componente: `src/pages/CompramosTuPC.jsx`. Es uno de los flujos más sofisticados del sitio. El formulario completo de 9 pasos vive en `src/components/compramos/CompramosTuPCForm.jsx`.

3.7.1 Header de la página

Tras múltiples iteraciones (#157, #158, #161, #196) la página quedó con:

- **Hero compacto** centrado con título "COMPRAMOS TU PC" como hipervínculo a la sede física (Google Maps).
- **4 burbujas de ventaja** encima del formulario: Valuación Express · Cero Estrés · Sin Obligación · Bono 5%.
- **Bloque del formulario** abierto en Paso 01 por defecto (el usuario no tiene que clicar "Empezar"; el primer paso ya se ve).
- **Sidebar derecho** (desktop ≥ lg) con barra de progreso vertical y vista resumen.

3.7.2 Pasos del formulario

#	Pregunta	Componente
1	El Gran Selector: Laptop vs Torre	<code>PasoTipoEquipo.jsx</code>
1.1	Preferencia de pago: Efectivo vs Crédito URG	<code>PasoPreferenciaPago.jsx</code>
2	Estado de la Unidad: Nuevo / Usado / Dañado	<code>PasoEstadoUnidad.jsx</code>
3	ADN del Equipo: Marca y Modelo + CPU + RAM	<code>PasoADN.jsx</code>
4	El Almacén de Datos: SSD vs HDD + capacidad	<code>PasoAlmacen.jsx</code>
5	Sólo para Laptops: Pantalla + Batería + Cargador	<code>PasoSensores.jsx</code>
6	Estado General (Apariencia): Grado A/B/C/D	<code>PasoEstadoFisico.jsx</code>
7	Briefing: Motivo de venta + Próximo uso	<code>PasoBriefing.jsx</code>
8	Contacto: Nombre + WhatsApp + Email (opcional)	<code>PasoContacto.jsx</code>
9	Pantalla Final: Misión Cumplida	<code>PasoFinal.jsx</code>

Cada paso aplica las reglas didácticas del V1.1: chips visuales, etiquetas de personalidad, tooltips de ayuda ("¿No sabes el modelo? Mira la etiqueta debajo de tu PC o presiona Win+R y escribe 'msinfo32'"),

avance automático tras selección, etc.

3.7.3 Cero Estrés (Subida de fotos)

Botón flotante "Prefiero enviar fotos" visible durante los pasos 3, 4, 5 y 6. Al pulsarlo:

- Hace fade-out de las preguntas técnicas.
- Abre Panel de Carga con 3 recuadros guía (corazón / frontal / DNI según Laptop o Torre).
- **Compresión cliente:** usando `Canvas API`, cada imagen subida se redimensiona a max 1920 px lado largo, convertida a WebP con calidad 0.85. El módulo `src/lib/imageOptimizer.js` también elimina EXIF (geolocalización, modelo cámara) por privacidad: `fileForensics.js`.
- **Salto al Paso 3 desactivado, salto a Paso 7 inmediato** (task #162: el banner "Modo Cero Estrés activo" se quitó porque generaba fricción).

3.7.4 PasoFinal - "Tech Pipeline" (task #197)

Pantalla de cierre rediseñada con animación tipo "pipeline tecnológico": los datos del cliente "viajan" por canales hacia el centro de procesamiento URG mientras un check verde neón gigante aparece. Texto:

¡Misión cumplida, Comandante! Tus datos han sido enviados al centro de mando. Estamos analizando tu [Marca/Modelo] para darte la mejor oferta.

Lista de siguientes pasos con iconos:

1. **Análisis:** En menos de 2 días hábiles recibirás un mensaje por WhatsApp con tu precio estimado.
2. **Peritaje:** Si te gusta el precio, agendamos una revisión física para darte el valor final.
3. **Pago + Bono:** Recibes tu dinero y activamos tu cupón del 5%.

CTA final adicional: [**ACCEDER A TU PORTAL DE SEGUIMIENTO**] con un PIN único generado al enviar el formulario (sistema PIN introducido en v19, ver §3.8).

3.7.5 Barra de Progreso Sticky

Franja superior delgada (32 px de alto) con fondo negro y borde inferior morado neón. Texto dinámico:

- Estado inicial: "BONO ACTIVO: +5% de Crédito extra al vender tu equipo en URG."
- Al 75%: "¡Casi listo! Tu Bono del 5% te espera al final."
- La barra de progreso es un radar verde neón que se llena al avanzar.

3.7.6 Persistencia y resiliencia

- **Autosave** cada cambio se guarda en `localStorage['urg.compramosPc.draft']`. Si el usuario abandona y regresa en otro día, retoma exactamente donde quedó (el sistema le pregunta "¿Continuar donde lo dejaste?").
- **Sincronización con Supabase** al cerrar la última pantalla: graba lead en tabla `buy_requests` vía `lib/buyRequests.js → addLead()`.
- **Notificación al admin** dispara el Email A.1 (Notificación Técnica) y A.2 (Confirmación Recibido al cliente) descritos en §6.6 Matriz de Operaciones.

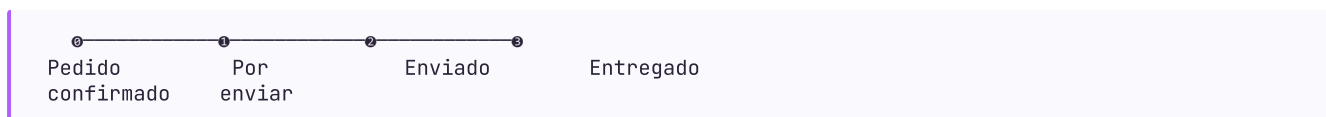
3.8 /rastrear-pedido

Componente: `src/pages/RastrearPedido.jsx`. Diseño consolidado tras v19 con PIN:

- **Input:** "Ingresa tu número de lote / orden de pedido (ej: #2603-104A)" + segundo input "PIN" (4 dígitos).
- El PIN se genera al confirmar el pedido (en `lib/orders.js → createOrder()`) y se envía al cliente por email + se muestra en la página de confirmación. El admin lo ve en el panel de Pendientes.
- **Auto-purge:** los PINs se borran de la tabla `orden_pins` a los 5 días post-entrega para no acumular datos sensibles.

Línea de Tiempo

Una vez el cliente ingresa código + PIN válidos, se muestran iconos circulares conectados:



- **Pedido confirmado** (Paso 0: siempre activo si el lote existe).
- **Por enviar:** "Estamos preparando tu equipo para envío."
- **Enviado:** "¡Tu equipo ya está en manos de la transportadora!" Desbloquea bloque con transportadora, número de guía y botón "RASTREAR EN [NOMBRE]".
- **Entregado:** "El equipo ha llegado a su destino. Gracias por confiar en URG." Aparece CTA verde "CALIFICA NUESTRO SERVICIO" hacia el link de Google Maps configurable.

Banner de retraso

Debajo de la línea de tiempo, banner sutil con fondo gris oscuro:

"¿Tu pedido tarda más de lo esperado?" En URG monitoreamos cada envío, pero si la transportadora presenta un retraso o no has recibido tu equipo en el tiempo estimado, estamos listos para brindarte una solución.

[CONTACTAR A SOPORTE]

El CTA abre WhatsApp con mensaje pre-cargado: *"Hola URG, mi pedido con lote #[Número] presenta un retraso, ¿me ayudan?"*.

3.9 /privacidad

Componente: `src/pages/Privacidad.jsx`. Versión estable del documento de Política de Privacidad bajo Ley 1581 de 2012 (Habeas Data, Colombia).

- **Layout:** página limpia con menú lateral (TOC) que permite saltar entre secciones colapsables.
- **Contenido bloqueable desde Panel Maestro** (CMS por secciones: cada sección tiene icono/emoji editable, título, cuerpo rich text).
- Secciones por defecto:
 1. Lo que sabemos de ti (datos personales)
 2. Cómo mejoramos tu experiencia (cookies y analítica)
 3. Tus pagos seguros (MercadoPago, BRE-B, cifrado)
 4. Comunicaciones y acompañamiento
 5. Tus derechos (Habeas Data): Conocer, Actualizar, Rectificar, Suprimir

El cliente puede solicitar el [Borrado Total] de sus datos escribiendo a `[correo de la tienda]` o, si existe sesión, desde un portal de cliente que apunta al Módulo de Privacidad (§6.5.1: Derecho al Olvido).

El borrado es **anonimización quirúrgica**, no DELETE físico: los montos, fechas y productos quedan en el histórico financiero bajo el rótulo "Cliente Anonimizado (Solicitud de Privacidad)".

3.10 /compra-protegida

Componente: `src/pages/CompraProtegida.jsx`. Página de la promesa de respaldo:

1. **Pagos Blindados y Transparencia Total** (MercadoPago + BRE-B + Ley 1581 + no almacenamos tarjetas).
2. **Compromiso de Entrega 21 Días o Reembolso** (si la transportadora no entrega en 21 días calendario, reembolso total).
3. **Fidelidad Técnica** ("Lo que ves es lo que recibes"): estado estético + specs + Certificación URG (cada puerto, tecla y píxel probado).
4. **Política de Devolución Plan 5/15:**
 - 5 días hábiles tras recepción → Derecho de Retracto (Ley 1480 Art. 47), cliente asume envío.
 - 15 días hábiles → Falla técnica de hardware: URG asume todo, recoge y entrega equipo superior o reembolsa 100%.
5. **Respaldo Post-Venta:**
 - Garantía de Hardware 3/6/12/24/48 meses según producto.
 - Soporte técnico gratis 1 año.

Notas Legales (Letra Menuda):

- Exclusiones de garantía (golpes, humedad, voltaje externo, sellos rotos).
- Software es responsabilidad del usuario tras la entrega.
- Reembolsos por mismo medio de pago en 8-15 días hábiles.

Diseño "cards" con iconos lineales modernos, fondo blanco limpio, acentos morados. Botón flotante WhatsApp con mensaje "¿Dudas sobre tu garantía? Chatea con un experto".

3.11 /diseno-web → redirect a damorus.com

Como se mencionó en §2.7. Componente `<ExternalRedirect to="https://www.damorus.com" />`.

3.12 Atajos de catálogo (/ofertas, /portatiles, /escritorio, /perifericos)

Componentes en `src/pages/PortatilesEscritorio.jsx`. Cada uno es un wrapper que setea filtros pre-aplicados sobre `<Catalogo />` y modifica el breadcrumb y `<title>`. Por ejemplo, `<Portatiles />` renderiza:

```
<Catalogo categoriaInicial="portatil" tituloOverride="Portátiles" breadcrumbCustom=
  [{"Inicio", "Portátiles"}] />
```

3.13 /mas-vendidos

Componente: `src/pages/MasVendidosPage.jsx`. Muestra los 12 puestos completos del ranking (4 categorías × 3 puestos = 12 productos). A diferencia del bloque Home que solo muestra el TOP 1, esta página muestra los 3 puestos de cada categoría en secciones apiladas verticalmente. Cada producto con su badge de posición (#1 con borde dorado glow, #2 plata, #3 bronce: task #251).

Si la categoría tiene menos de 3 productos rankeados, los slots vacíos se reemplazan por un placeholder gris con texto "Posición disponible: pronto un nuevo equipo aquí". Esto refuerza la sensación de cura editorial.

4. SISTEMA DE FICHAS

Cada producto se renderiza en uno de 3 formatos dependiendo del contexto. El componente que gobierna el formato es siempre **el mismo** (`ProductCard.jsx` con prop `variante: "1" | "2" | "3"`), garantizando que cualquier cambio de marca o de schema se aplique a las tres a la vez.

4.1 FICHA 1 • Estándar de Catálogo

Componente: `src/components/cards/Ficha1.jsx` . Es la unidad básica de la **Cuadrícula de Alta Densidad**.

Uso

- Home: Hero (no aplica), Oferta del día (Ficha 1 horizontal), Top 1 de cada categoría en Más Vendidos.
- Catálogo: NO se usa Ficha 1 en `/catalogo` general (allá manda Ficha 2). Se usa Ficha 1 en las pestañas dedicadas que muestran pocas categorías (`/ofertas` , `/mas-vendidos`).
- Periféricos: SIEMPRE Ficha 1 (decisión §4.4).

Orientación

Mayoritariamente vertical. La **única excepción** es la Oferta del Día en desktop, donde se renderiza horizontal por instrucción específica del fundador (1ª imagen ocupando 58%, panel info 42%).

Componentes en orden

1. **Imagen principal del producto:** `aspect-ratio: 4/3` (rollback al ratio original tras task #37, el cambio a 1:1 perdía contexto). Carga lazy con skeleton.
2. **Badge de estado** (esquina superior derecha): cápsula con colores diferenciados Nuevo/Como Nuevo/Excelente/Bueno (siempre visible incluso en placeholder, task #12).
3. **Badge de "Envío gratis Bogotá"** personalizable (esquina superior izquierda).
4. **Nombre Marketing** en negrita máx 2 líneas (sin descriptor: task #8 lo eliminó porque saturaba).
5. **Marca + Modelo** una sola línea, color `text-zinc-500` (sin prefijo "Marca:").
6. **Precio Original** tachado en gris tenue.
7. **Precio Oferta** color morado URG, fuente grande.
8. **Badge OFF** cápsula lavanda con `%OFF` .
9. **CTA "¡COMPRAR AHORA!"** con efecto glow recorrido cada 3 s. Click va a Ficha 3 (no al checkout directo, decisión task #239).
0. **Click en cualquier parte de la ficha** (no en el CTA): también va a Ficha 3.

Estado VENDIDO

Cuando `stockDisponible === 0` :

- Overlay rojo translúcido sobre la imagen.
- Sello centrado "VENDIDO" en blanco (no diagonal, task v24.18.22).
- CTA reemplazado por botón "Avísame cuando vuelva" (sistema `avisame.js`).

4.2 FICHA 2 · Búsqueda Avanzada

Componente: `src/components/cards/Ficha2.jsx`. Formato horizontal compacto que aparece en `/catalogo`, `/portatiles`, `/escritorio`.

Layout

Es un rectángulo vertical con:

- **Fila 1 · Imagen horizontal** ocupando ~1/3 - 1/4 del card height, con dots indicadores tipo Mercado Libre / Amazon (1 a 9 imágenes).
- **Esquina superior derecha · Badge "Estado"** morado llamativo (Nuevo / Como Nuevo / Excelente / Bueno).
- **Fila 2 · Nombre Marketing** en negrita.
- **Fila 3 · Marca + Modelo** una sola línea, tono gris.
- **Fila 4 · Lista de specs técnicas** con etiquetas prefijadas:
 - Memoria: 32GB
 - Gráficos: 30-Core GPU
 - Almacenamiento: 1TB SSD
 - Garantía: 12 Meses URG
- **Fila 5 · Línea divisora morada.**
- **Fila 6 · Precio** (con tachado y descuento si aplica) en la esquina izquierda + botón "Detalles" a la derecha.

Filtro de categoría (chips superiores)

En las pestañas de Ficha 2 hay un menú desplegable minimalista "Filtrar por categoría" en la parte superior. Se documenta detalladamente en §3.1 (filtros del Ecosistema de Búsqueda).

4.3 FICHA 3 · Terminal de Producto (Vista de Conversión Final)

Componente: `src/pages/ProductoDetalle.jsx`. La página más completa del sitio público.

4.3.1 Layout responsive

Mobile (< md)

Una sola columna con orden optimizado:

1. **Migas de pan + hipervínculo "Compartir"** y "Ver más productos marca X" en la parte superior.
2. **Galería de imágenes** ancha (aspect 4/3) con dots indicadores. El botón Compartir (cuadrado con flecha) es visible.
3. **Marca** como link inline al filtro de marca (task #32: antes era un bloque aparte, se compactó).
4. **Estado** resaltado en verde + ranking "#X en Y" si aplica (mismo color del verde-neón).
5. **Nombre Marketing** en negrita.
6. **Si es Oferta del Día**: banner debajo del nombre.
7. **Precio** (tachado + final + %OFF si aplica).
8. **Colores** seleccionables con chip cuadrado por color.
9. **Tamaño y Peso** en chips (orden Tamaño → Peso tras task #3).
0. **Specs técnicas** Memoria / Gráficos / Almacenamiento / Garantía.

1. **Botones CTA** "Agregar al carrito" + "Comprar ahora" (botón "Comprar ahora" verde brillante, único de la página con esa tonalidad).
2. **Medallita meses de garantía** + flechita verde "Compra protegida" (hipervínculo a /compra-protegida).
3. **Promesa de entrega dinámica** (algoritmo 48h) "Entrega GRATIS el Martes 24 de Junio en Bogotá".
4. **Sobre este producto** (descripción manual).
5. **Producto similar a precio competitivo** (1 card).
6. **Especificaciones adicionales dinámicas** (Medidas, Materiales, Estilo, etc.: solo los que el admin haya llenado).
7. **Strip de logos de pago 1x6**.
8. **Bloques "Otros similares"** (Más vendidos categoría, Misma marca, Más baratos).

Desktop (≥ md)

Tres columnas en grilla 12:

- **Col 1-7** (Multimedia): galería principal + thumbnails grid + botón compartir.
- **Col 7-10** (Identidad): "Ver más productos marca X", estado/ranking, nombre, precio, color, tamaño/peso, marca, modelo, memoria, gráficos, almacenamiento, garantía. Debajo "Sobre este producto" y "Producto similar a precio competitivo".
- **Col 10-12** (Compra): tabla precio + garantía + GRATIS 1 año servicio técnico (Toggle en admin) + estado disponibilidad + CTAs Agregar al carrito / Comprar ahora + medallita garantía + Compra protegida + Promesa de entrega.

4.3.2 Sistema de Identidad Digital y Enrutamiento (URLs Únicas)

- **Slug** generado automáticamente al crear la ficha: kebab-case, sin tildes, sin caracteres especiales.
- **Inmutable** una vez publicado (para no romper enlaces compartidos en WhatsApp). El admin solo puede cambiarlo manualmente bajo advertencia explícita.
- **Sufijo numérico automático** si hay colisión.
- **history.pushState** al abrir Ficha 3 desde el catálogo (URL cambia sin recargar).
- **popstate listener** para cerrar la Ficha 3 al presionar "Atrás".

4.3.3 Botón de Compartir Inteligente

Cuadrado con flecha (ícono Material Design). Al hacer click:

- Si el navegador soporta Web Share API → `navigator.share({ title, text, url })`.
- Sino → copia la URL al portapapeles con `navigator.clipboard.writeText()` y muestra toast "¡Link copiado con éxito! Ya puedes enviarlo".
- Meta-tags configurados para que al pegar el link en WhatsApp aparezca la miniatura del producto, precio y nombre.

4.3.4 Sticky CTA recordatorio (mobile)

Cuando el usuario hace scroll y los botones de compra principales salen del viewport, aparece una barra delgada fija en la parte inferior con: nombre del equipo (truncado) + precio + un solo botón "COMPRAR". Visible solo en la sección "Sobre este producto" hacia abajo (task #33: antes aparecía siempre, distraía).

Evita interferir con el botón flotante de WhatsApp gracias a un `z-index` calibrado y un `margin-right` (`pr-20` en mobile) que reserva espacio.

4.3.5 Tooltip Premium "SpecTooltip"

En Ficha 3 cada spec técnica (procesador, RAM, etc.) tiene un icono pequeño (i) que al hover abre un tooltip estilo terminal de comandos (`background: #0a0a0f; font-family: 'JetBrains Mono'; border: 1px solid var(--urg-purple)`). Explica en lenguaje sencillo qué significa esa spec ("Núcleos del procesador: cuantos más, mejor para multitarea pesada").

En mobile el tooltip se activa por tap (no por hover): task #269. Toggle visual en pequeña pestaña debajo de las specs.

4.3.6 Promesa de entrega dinámica (Smart Delivery Timer)

Lógica en `src/lib/format.js → calcularFechaEntrega()` :

- Suma 2 días hábiles a hoy.
- Regla viernes → lunes (saltando domingo).
- Regla sábado/domingo → martes.
- Ignora domingos para el conteo.
- Output: *"Entrega GRATIS el Lunes 24 de Junio en Bogotá"*.

4.3.7 Galería avanzada

- **Drag con shift** para reordenar imágenes (visible solo en admin: task #259).
- **Lupa magnifier** al hover desktop (280×280 px, zoom 500%: task #239).
- **Mobile** swipe horizontal con snap.
- **Videos** se cargan en lazy load, mute por defecto pero con icono sonido visible para no asustar en lugar público.

4.4 ADAPTACIÓN DE FICHAS PARA PERIFÉRICOS

Lógica en `lib/categoriasPerifericos.js` . Cuando el admin marca una categoría como "Periférico" (vs "Computador"), el Creador de Fichas oculta automáticamente Procesador/RAM/GPU y habilita en su lugar 4 campos de texto libre llamados **"Highlights Técnicos"** (ej: DPI, Tipo de Switch, Conectividad, Autonomía).

- **En Home y Catálogo:** los periféricos usan SIEMPRE Ficha 1 (vertical, limpia para escaneo rápido).
- **En burbuja /perifericos:** Ficha 1 (a diferencia de computadores que en sus burbujas usan Ficha 2: esto se decidió así porque los periféricos no tienen tantas specs comparables como un PC).
- **En Ficha 3 de periférico:**
 - Col 2 muestra 4 Highlights Técnicos en lugar del bloque de specs computador.
 - "Producto similar a precio competitivo" busca **otro periférico de la misma sub-categoría** (si el usuario está viendo un Mouse, sugiere otro Mouse, nunca un teclado).

Sigue la documentación del **Apartado de Pagos** en la sección 5, con cobertura completa del modelo de reservas atómicas server-side que es el cambio más profundo desde V1.1.

5. APARTADO DE PAGOS

El módulo de pagos vivió un rediseño arquitectónico completo entre v18.15 y v24.18.25. La V1.1 describía un flujo cliente-céntrico donde la web mutaba el stock localmente al iniciar una compra. La realidad post-v24.18.25 es radicalmente distinta: **la lógica crítica de stock vive en el servidor** (Edge Functions + pg_cron + RPCs Postgres), el cliente solo escribe intenciones y consulta resultados.

Esta sección documenta tanto la UX del cliente (idéntica en superficie al V1.1) como el modelo de datos y reservas atómicas server-side que la sostiene.

5.1 Formulario de Datos para la Facturación

Pantalla: `src/pages/Checkout.jsx` paso 1. *"Solicitamos únicamente la información esencial para la finalización y envío de la compra."*

5.1.1 Campos personales (siempre obligatorios)

1. **Nombres:** `maxLength: 60` (límite reducido +30% tras v24.18.18, anteriormente sin límite).
2. **Apellidos:** `maxLength: 60`.
3. **Cédula o Pasaporte:** `pattern: ^[0-9A-Z]{6,12}$`, `maxLength: 15`.
4. **Teléfono / Móvil:** `pattern: ^\+?[0-9\s()-]{7,15}$`, `maxLength: 18`.
5. **Correo electrónico:** `type: email`, sanitizado vía regex en `lib/validators.js` → `isEmail()`.

Opción "Facturar a nombre de empresa"

Checkbox que expande sección adicional:

- **Razón Social** (texto).
- **Nombre de la empresa** (texto).
- **NIT** (`pattern: ^[0-9.-]{9,15}$`).
- **RUT** (carga de archivo PDF opcional, máx 5 MB).

Si el cliente cambia de opinión, puede colapsar la sección con "No sumar" y los datos se mantienen en memoria por si reactiva.

5.1.2 Información para el envío

3 opciones excluyentes (radio buttons grandes con icono):

Opción A • Recoger en sede física (GRATIS)

"Recogeré mi compra en [dirección de la sede]". Sin campos adicionales.

Opción B • Enviar a Bogotá (GRATIS)

Expande:

- **Dirección** (campo principal).
- **Dirección Complementaria** (apto, oficina): opcional, gris.
- **Código postal:** opcional.
- **Nombre de la persona que recibe:** pre-rellenado con el nombre del paso anterior, editable.

Opción C · Enviar a otra ciudad de Colombia (tarifa configurable)

Expande:

- **Departamento + Ciudad:** selector cascada con `src/data/coloDeptos.js` (catálogo oficial de departamentos y municipios colombianos). Componente `<DeptoCiudadSelector />`.
- **Dirección** (texto).
- **Dirección Complementaria** (gris, opcional).
- **Código postal** (opcional).
- **Nombre de la persona que recibe** (editable).

Notas adicionales para la entrega

Textarea opcional con placeholder *"Instrucciones especiales para el mensajero o comentarios sobre tu pedido (ej. Entregar después de las 2:00 PM o llamar al llegar)"*. Máx 280 caracteres. Alimenta la variable `{{notas_cliente}}` en los emails admin.

5.1.3 Corrección de errores (validación)

Implementada con HTML5 nativo + `lib/validators.js`:

- Borde rojo neón en campo inválido.
- Mensaje debajo: *"Este campo es obligatorio para la factura"* o el error específico.
- Botón "Pagar" deshabilitado hasta validar.
- **Scroll-to-Error:** si el cliente intenta pagar con campos vacíos, `window.scrollTo({ top: errorFieldOffset - 100, behavior: 'smooth' })` lo lleva al primero faltante.

5.1.4 Cupones de regalo

Campo "Código de regalo" + botón "Añadir". Aplica descuento porcentual o fijo según configuración en Bloque 4 del Tributario (§6.10). Si el código es inválido, mensaje rojo *"Este código no existe o ya expiró"*. No-acumulable (solo un cupón por transacción).

5.1.5 Resumen del pedido

Bloque dinámico que se actualiza en tiempo real:

```
Producto: MacBook Pro M3 (con 10% OFF aplicado)
Subtotal:      $ 2.521.008
IVA (19%):     $ 478.992
Envío:         (tarifa de envío)
Cupón de Regalo: -$ 150.000
TOTAL A PAGAR: $ 2.850.000 + envío
```

Cuando el producto está **excluido de IVA** (precio $\leq$ 50 UVT), el desglose se reemplaza por:

```
Subtotal:      $ 2.400.000
IVA (0%):      $ 0 (Excluido Art. 424 E.T.)
Envío:         $ 0
TOTAL A PAGAR: $ 2.400.000
```

5.1.6 Método de pago

Dos botones principales con logos:

[Logo MercadoPago]
Mercado Pago
Pago inmediato
con tarjeta o cuenta

[Logo BRE-B]
BRE-B
Transferencia con
comprobante

Aceptamos: [Visa] [MC] [Amex] [Nequi]

Y debajo el checkbox obligatorio:

☐ Acepto la Política de Privacidad y el Tratamiento de mis
Datos Personales de URG TECHNOLOGY SAS

- Casilla NUNCA marcada por defecto.
- Mientras no esté marcada, los botones MercadoPago y BRE-B están en gris desactivado.
- Click en "Política de Privacidad" abre </privacidad> en pestaña nueva.

5.2 Confirmación de Pedido Digital (PDF Orden)

Tras pago exitoso, se genera un PDF automático "ORDEN_URG_{{ID_pedido}}.pdf" (ejemplo: [ORDEN_URG_2603-104A.pdf](#)). Generador: [src/lib/pdfBuilder.js](#) usando [pdf-lib](#).

Estructura del PDF (4 bloques)

- **Bloque A · Cabecera:**
 - Logo URG ([URG_FOOTER_2026.svg](#) convertido a PNG embebido).
 - Título: [ORDEN DE PEDIDO / COMPROBANTE DE COMPRA](#).
 - ID del Pedido: [#2603-104A](#).
 - Datos empresa: NIT, dirección, correo y WhatsApp de la tienda (valores omitidos en la versión pública).
- **Bloque B · Cliente:** datos del formulario (Nombre, Cédula/NIT, Dirección, Teléfono).
- **Bloque C · Detalle:** tabla con descripción, estado (Excelente/Bueno), cantidad, valor unitario.
- **Bloque D · Financiero y Notas:**
 - Subtotal, IVA, Total.
 - Nota legal: *"Este documento es una orden de pedido y no constituye una factura de venta. Su factura legal electrónica será entregada de forma física junto con su producto".*

El PDF se adjunta automáticamente al email de confirmación enviado por Resend (Email B.2). Una copia queda disponible para descarga permanente desde el módulo Vendidos del admin ([Centro de Descargas](#)).

5.3 Sistema de Nomenclatura y Trazabilidad (ID de Pedido)

Formato: [YYMM-###X](#)

- [YY](#) = últimos 2 dígitos del año (26 para 2026).
- [MM](#) = mes (03 para marzo).
- [-](#) separador.
- [###](#) = contador correlativo de 3 dígitos (000-999).
- [X](#) = letra de ciclo (A → B → C...).

Reglas de escalabilidad

- **Reinicio mensual:** al iniciar nuevo mes, el contador vuelve a `000`.
- **Salto de letra:** al superar 999 en un mismo mes, salta de `999A` a `000B`.
- **Persistencia:** el ID se genera **al microsegundo en que el cliente toma la decisión de pagar** (no al confirmar). Si la sesión se interrumpe, el admin mantiene registro del "Intento de Venta".
- **Nunca se elimina** un ID de pedido: solo cambia de estado (EFECTUADO vs CANCELADO).

Generación implementada como función Postgres (RPC `generar_id_pedido()`):

```
CREATE OR REPLACE FUNCTION generar_id_pedido() RETURNS text AS $$
DECLARE
    v_yymm text := to_char(now() AT TIME ZONE 'America/Bogota', 'YYMM');
    v_count int;
    v_letra text;
BEGIN
    -- Lock para evitar race condition entre dos creaciones simultáneas
    PERFORM pg_advisory_xact_lock(hashtext(v_yymm));
    SELECT COUNT(*) INTO v_count FROM orders
    WHERE id_pedido LIKE v_yymm || '-%';
    v_letra := chr(65 + v_count / 1000); -- A, B, C...
    RETURN v_yymm || '-' || lpad((v_count % 1000)::text, 3, '0') || v_letra;
END $$ LANGUAGE plpgsql;
```

Lógica de Resolución Final

"EFECTUADO"

- Trigger: webhook MercadoPago `approved` o admin presiona `[CONFIRMAR PAGO]` en BRE-B.
- Impacto: stock se resta definitivamente, estado pasa a `POR_ENVIAR`, metadata se bloquea.

"CANCELADO"

- Trigger 1 (expiración): orden `INICIADO` supera 48 h sin reporte exitoso → cron limpia.
- Trigger 2 (rechazo manual): admin presiona `[RECHAZAR PAGO]`.
- Impacto: estado `RECHAZADO` o `FALLIDO_EXP`, re-stock automático, ID queda en histórico con marca gris/roja.

5.4 Modelo de Reservas de Stock Atómico (CRÍTICO · server-side)

Esta es la sección más **importante** del documento. La V1.1 asumía una arquitectura simple cliente-céntrica donde el front-end consultaba `producto.stock` y restaba. La realidad de producción reveló los siguientes problemas:

1. **Doble venta:** dos clientes podían pagar simultáneamente por el mismo producto único.
2. **Stock fantasma:** clientes que iniciaban pago y abandonaban dejaban reservas que nunca se liberaban.
3. **Dependencia de pestaña abierta:** la primera implementación requería que un admin tuviera el panel abierto para procesar los webhooks → si el admin dormía, el sitio no actualizaba stock.

El rediseño v24.18.25 movió **toda la lógica crítica al servidor**:

5.4.1 Modelo de datos

Tabla `productos` (extracto):

```
CREATE TABLE productos (
  id          uuid PRIMARY KEY DEFAULT gen_random_uuid(),
  slug        text UNIQUE NOT NULL,
  nombre_marketing text NOT NULL,
  -- ... otros campos ...
  stock       int NOT NULL DEFAULT 0,      -- inventario físico real
  stock_reservado int NOT NULL DEFAULT 0,  -- unidades temporalmente apartadas
  reservas    jsonb NOT NULL DEFAULT '[]', -- detalle de cada reserva activa
  CONSTRAINT stock_no_negativo CHECK (stock - stock_reservado ≥ 0)
);
```

El campo `reservas` es un array de objetos:

```
[
  {
    "order_id": "2606-014A",
    "metodo": "BRE-B",
    "creada_at": "2026-06-19T14:30:00Z",
    "expira_at": "2026-06-19T14:38:00Z",
    "estado": "PENDIENTE_COMPROBANTE",
    "user_session": "anon-abc123"
  },
  {
    "order_id": "2606-015B",
    "metodo": "MP",
    "creada_at": "2026-06-19T14:32:00Z",
    "expira_at": "2026-06-19T15:32:00Z",
    "estado": "PASARELA_ABIERTA",
    "user_session": "anon-xyz789"
  }
]
```

Stock disponible para venta se calcula como:

```
stockDisponibleAhora(producto) = producto.stock - producto.stock_reservado
```

El front llama esto en tiempo real desde `src/lib/orders.js → getStockDisponibleAhora(productId)` que hace `SELECT stock, stock_reservado FROM productos WHERE id = $1`.

5.4.2 Función RPC `reservar_stock_atómico`

```
CREATE OR REPLACE FUNCTION reservar_stock_atómico(
  p_producto_id uuid,
  p_order_id text,
  p_metodo text,
  p_ttl_minutos int
) RETURNS jsonb AS $$
DECLARE
  v_disponible int;
BEGIN
  -- Lock pessimista sobre la fila
  SELECT stock - stock_reservado INTO v_disponible
    FROM productos WHERE id = p_producto_id FOR UPDATE;

  IF v_disponible ≤ 0 THEN
    RETURN jsonb_build_object('ok', false, 'motivo', 'STOCK_AGOTADO');
  END IF;

  UPDATE productos
    SET stock_reservado = stock_reservado + 1,
        reservas = reservas || jsonb_build_object(
          'order_id', p_order_id,
          'metodo', p_metodo,
          'creada_at', to_char(now(), 'YYYY-MM-DD"T"HH24:MI:SS"Z"),
          'expira_at', to_char(now() + (p_ttl_minutos || ' minutes')::interval, 'YYYY-MM-
```

```
DD"\"HH24:MI:SS\"Z\"'),
    'estado', CASE WHEN p_metodo = 'MP' THEN 'PASARELA_ABIERTA' ELSE 'PENDIENTE_COMPROBANTE' END
  )
  WHERE id = p_producto_id;

  RETURN jsonb_build_object('ok', true);
END $$ LANGUAGE plpgsql;
```

Garantía: solo una transacción puede sostener el lock, por lo que dos reservas concurrentes nunca pueden ambas pasar la verificación.

5.4.3 Función RPC `consumir_stock`

Llamada **únicamente desde el servidor** (`mp-webhook` cuando MP confirma `approved`, o admin endpoint cuando aprueba BRE-B):

```
CREATE OR REPLACE FUNCTION consumir_stock(
  p_producto_id uuid,
  p_order_id text
) RETURNS jsonb AS $$
BEGIN
  UPDATE productos
    SET stock = stock - 1,
        stock_reservado = GREATEST(stock_reservado - 1, 0),
        reservas = (
          SELECT COALESCE(jsonb_agg(r), '[]'::jsonb)
            FROM jsonb_array_elements(reservas) AS r
           WHERE r->>'order_id' ≠ p_order_id
        )
    WHERE id = p_producto_id;

  -- Auto-archivar si stock llega a 0
  IF (SELECT stock FROM productos WHERE id = p_producto_id) = 0 THEN
    UPDATE productos SET hidden = true WHERE id = p_producto_id;
  END IF;

  RETURN jsonb_build_object('ok', true);
END $$ LANGUAGE plpgsql;
```

5.4.4 Función RPC `liberar_reserva`

```
CREATE OR REPLACE FUNCTION liberar_reserva(
  p_producto_id uuid,
  p_order_id text
) RETURNS void AS $$
BEGIN
  UPDATE productos
    SET stock_reservado = GREATEST(stock_reservado - 1, 0),
        reservas = (
          SELECT COALESCE(jsonb_agg(r), '[]'::jsonb)
            FROM jsonb_array_elements(reservas) AS r
           WHERE r->>'order_id' ≠ p_order_id
        )
    WHERE id = p_producto_id;
END $$ LANGUAGE plpgsql;
```

5.4.5 Flujo BRE-B (con comprobante)

1. Cliente confirma datos y elige BRE-B.
2. Cliente sube comprobante (campo de carga: máx 10 MB, JPG/PNG/PDF). Componente `<ComprobanteUploader />`.

3. El comprobante dispara `subirComprobanteBreB(order)` que internamente llama `reservar_stock_atómico` con TTL 60 min y estado `PENDIENTE_COMPROBANTE`. Si la reserva falla (stock agotado), el cliente recibe modal "¡Alguien fue más veloz!" y se le ofrecen alternativas.
4. La orden queda en estado `RESERVADO_HV` (Reservado Hasta Verificación) en la tabla `orders`.
5. Admin recibe email B.1 (Alerta de Pago Pendiente) y ve la orden en el módulo Pendientes (con badge naranja).
6. Admin presiona `[CONFIRMAR PAGO]` → endpoint server-side ejecuta `consumir_stock`. Stock se resta permanentemente. Email B.2 al cliente con PDF.
7. Si admin presiona `[RECHAZAR PAGO]` → ejecuta `liberar_reserva`. Stock vuelve al catálogo. Email B.4 al cliente con motivo.

5.4.6 Flujo MercadoPago

1. Cliente confirma datos y elige MercadoPago.
2. Cliente es redirigido a la pasarela MP (a través de `mp-create-preference` Edge Function).
3. `redirectToCheckout()` ya NO reserva stock localmente. Esa decisión se invirtió en v24.18.23: solo el webhook puede tocar stock.
4. Cliente paga en MP. MP dispara webhook a `mp-webhook` Edge Function con `event: payment.updated`.
5. El webhook ejecuta dos pasos atómicos:
 - `reservar_stock_atómico(producto_id, order_id, 'MP', 0)`: TTL 0 porque se consume inmediato.
 - `consumir_stock(producto_id, order_id)`: si el paso anterior fue OK.
6. Cliente regresa a `/confirmacion/{id}?status=approved` y ve "¡Pago Confirmado!".

5.4.7 Cron de limpieza (NUEVO en v24.18.25)

Existe un cron de Postgres que cada 2 minutos llama a la Edge Function `cleanup-stock-fantasmas`. Esta función:

1. Busca todas las reservas con `expira_at < now()` y estado `PENDIENTE_COMPROBANTE` (BRE-B sin comprobante en 8 min).
2. Para cada una, ejecuta `liberar_reserva` y marca la orden asociada como `FALLIDO_EXP`.
3. Busca también órdenes MP en estado `PASARELA_ABIERTA` con más de 1h y las libera.
4. Si la liberación libera el último stock disponible para un producto, **dispara un evento Realtime** a los clientes conectados: si alguno está en `/checkout` o `/productos/X` viendo ese producto y otro se vendió, se le muestra modal "¡Alguien fue más veloz!".

Schedule:

```
SELECT cron.schedule(
  'cleanup-stock-fantasmas-2min',
  '* / 2 * * * *',
  $$
  SELECT net.http_post(
    url := 'https://[proyecto-supabase]/functions/v1/cleanup-stock-fantasmas',
    headers := jsonb_build_object(
      'Content-Type', 'application/json',
      'Authorization', 'Bearer ' || current_setting('app.service_role_key')
    )
  );
  $$
);
```

(El secret `app.service_role_key` está cargado vía `ALTER DATABASE postgres SET app.service_role_key = '...'`.)

5.5 Cancelación Cruzada BRE-B ↔ MP

Antes de v24.18.23, era posible que dos clientes terminaran ambos comprando el mismo producto único: uno por BRE-B (con comprobante ya subido) y otro por MP (con webhook aprobado). El stock era 1 → ambos pasaban inicialmente, generando deuda operacional.

La solución es la función `cancelarHuerfanosPostConsumo()`:

```
// En mp-webhook/index.ts después de consumir_stock
async function cancelarHuerfanos(supabase, productoId, orderIdGanador) {
  // Listar todas las reservas activas restantes para ese producto
  const { data: producto } = await supabase
    .from('productos').select('reservas, stock')
    .eq('id', productoId).single();

  if (producto.stock ≤ 0) {
    // Stock agotado: cualquier reserva pendiente está huérfana
    for (const reserva of producto.reservas) {
      if (reserva.order_id !== orderIdGanador) {
        await supabase.rpc('liberar_reserva', {
          p_producto_id: productoId,
          p_order_id: reserva.order_id
        });
        // Marcar orden como FALLIDO_EXP y notificar cliente vía Realtime
        await supabase.from('orders')
          .update({ estado: 'FALLIDO_EXP', motivo_cancelacion: 'OVERSOLD' })
          .eq('id_pedido', reserva.order_id);
        // Enviar email de disculpa via Resend
        await dispararEmailDisculpa(reserva.order_id);
      }
    }
  }
}
```

El cliente perdedor (que estaba en `/checkout` o `/confirmacion` esperando) recibe:

1. Evento Realtime → frontend dispara `<StockAgotadoModal />`.
2. Email de disculpa via Resend con CTA "Mostrarme otras opciones" → catálogo filtrado por la categoría perdida + cupón compensatorio del 5%.

Texto del modal:

¡Alguien fue más veloz!

Alguien más completó el proceso de pago justo antes que tú. En URG las ofertas de este calibre vuelan a velocidad luz.

¡No te quedes en tierra! Tenemos equipos con potencia similar listos para ti.

[MOSTRARME OTRAS OPCIONES] [Cerrar]

(Estilo: título y texto principal en rojo `var(--urg-red)` por instrucción explícita del fundador.)

5.6 Cronómetro semáforo BRE-B (mobile draggable)

Para el flujo BRE-B, el cliente tiene 8 minutos para subir el comprobante. El cronómetro tradicional aparece como una barra superior sticky, pero en mobile se introdujo (v24.18.18) una **versión flotante draggable**:

- Burbuja redonda 72 × 72 px con cuenta regresiva en el centro.
- Cambia color tipo semáforo:
 - Verde >5 min restantes.
 - Amarillo 2-5 min.
 - Rojo <2 min.
- Posicionable por el usuario (touch-drag por toda la pantalla, persiste posición en localStorage).
- Tap abre la pantalla completa BRE-B si el usuario se distrajo.

5.7 Pantalla "Tu tiempo expiró" (BRE-B)

Si el contador llega a 00:00, la interfaz se bloquea y muestra:

```
Tu tiempo para confirmar el pago ha expirado.

[ REINICIAR PAGO BRE-B ]      (botón primario destacado)

[ Pagar con Mercado Pago ]   [ Hablar con un Asesor ]
                             (botones secundarios)

Volver al Catálogo (link discreto inferior)
```

- **Reiniciar Pago BRE-B** genera un nuevo ID de pedido (el anterior queda como **CANCELADO**) y un nuevo ciclo de 8 minutos.
- **Pagar con MercadoPago** recupera el carrito y redirige a la pasarela MP.
- **Hablar con un Asesor** abre WhatsApp con mensaje *"Hola, se me acabó el tiempo para pagar mi pedido #[ID] y necesito ayuda para finalizar mi compra"*.

5.8 Persistencia de carrito post-MP (anti-pérdida)

Una decisión importante: si el cliente paga por MP y el pago falla (rejected), al regresar a URG **el carrito mantiene los productos seleccionados**. Esto evita que el cliente tenga que re-buscar los equipos y aumenta la tasa de conversión en reintentos. La persistencia se guarda en `localStorage['urg.cart']` con tiempo de vida de 24 h.

Esta sección cierra la documentación del Apartado de Pagos. A continuación, la **sección 6** documenta el Panel de Administrador completo (con sus 4 bloques: Operación Diaria, Gestión del Cliente, Inteligencia y Seguridad, Configuración y Estructura) y todas las novedades v24, especialmente Centinela (Salud Operacional) y la arquitectura server-side.

6. PANEL DE ADMINISTRADOR (/[panel])

El panel administrativo vive en `/[panel]` (login) y `/[panel]/*` (módulos). Tiene Layout propio (`AdminLayout.jsx`), tema oscuro permanente, sin Header ni Footer ni WhatsApp del sitio público.

Arquitectura visual: barra superior fija (estado global) + sidebar lateral izquierdo (App-Launcher con 4 iconos) + área central de contenido. Cuatro bloques temáticos: **I. Operación Diaria** · **II. Gestión del Cliente** · **III. Inteligencia y Seguridad** · **IV. Configuración y Estructura**. Acceso por rol con permisos granulares Ver/Modificar por bloque (ver §6.11).

BLOQUE I · OPERACIÓN DIARIA

6.1 Login (/[panel])

Componente: `src/pages/admin/Login.jsx`.

Atmósfera visual

- **Fondo dinámico:** degradado de negro profundo (`#0B0B14`) a gris carbón (`#1C1C28`). Sistema de partículas en movimiento (efecto stardust generado con Canvas en `src/components/system/NeonAccent.jsx` → componente `<ParticlesField />`) que se desplaza lentamente, dando sensación de profundidad y computación avanzada.
- **Login Card:** efecto glassmorphism con `background: rgba(20,20,30,0.55); backdrop-filter: blur(24px)`. Borde 1 px en morado URG `#8A2BE2` con `box-shadow: 0 0 32px rgba(138,43,226,0.35)` que pulsa cada 4 s.
- **Logo URG** en la parte superior (versión footer en blanco mate) con animación de respiración (brillo morado sutil oscilante).

Campos de entrada

- **Campo de Identidad** (email o username). Icono "Perfil" sutil. Al entrar en foco, el borde transiciona de gris a morado vibrante.
- **Campo de Contraseña Inteligente:**
 - Icono "Ojo" para alternar visibilidad.
 - Indicador LED virtual que se enciende si Caps Lock está activo (detección con `e.getModifierState('CapsLock')`).
- **Botón CTA:** "Acceder al Sistema", fondo sólido morado, al hover emite pulso neón hacia los bordes.
- **Link Recuperar Contraseña** debajo del botón (estilo discreto, color lavanda claro).

Escudo de Seguridad Invisible

- **Cloudflare Turnstile** integrado (`<TurnstileWidget />`). Validación silenciosa, sin puzzles.
- **Retardo Progresivo** (`lib/rateLimiter.js`):
 - 1er fallo → 1 s espera.
 - 2do fallo → 2 s.
 - 3er fallo → 4 s.
 - 4to+ → 8 s (capped).
- **Honeypot:** rutas señuelo (su lista no se publica) registradas como middleware → cualquier acceso etiqueta la IP como "Agresor Potencial" en Black Box.

Protocolo "Modo Fantasma" (Recuperación)

- **Respuesta de Seguridad Invisible:** tras enviar email, el sistema siempre muestra el mismo mensaje genérico: *"Si los datos coinciden con un administrador registrado, recibirás un enlace de acceso en tu bandeja de entrada"*. Nunca confirma si el correo existe (anti-enumeración).
- **Magic Link** vía Supabase Auth con caducidad **15 minutos**.
- **Reinicio Forzado:** al entrar por recuperación, redirige a `/[panel]/resetear-clave` (Cambio de Contraseña): middleware detecta el evento `PASSWORD_RECOVERY` y fuerza el flujo.

Protocolo MFA / TOTP (Post-Login)

Implementado en `src/lib/mfa.js` usando Supabase Auth MFA:

- **Activación:** desde Perfil del admin (no obligatorio en primer login, pero obligatorio tras 7 días si Super Admin).
- **Sincronización TOTP:**
 1. Sistema genera código QR con `qrcode.react`.
 2. Admin escanea con Google Authenticator / Authy.
 3. Introduce código de 6 dígitos para validar.
- **Efecto Permanente:** cada login futuro exige el código de 6 dígitos vía `supabase.auth.mfa.verify()`.

6.2 Centro de Control Inmediato

Pantalla principal post-login. Componente: `src/pages/admin/Dashboard.jsx`. Compuesta por **Barra Superior Global + Sidebar Izquierdo + Contenido Central**.

6.2.1 Barra Superior de Estatus Global

Componente: `src/components/admin/AdminUI.jsx` → `TopBar`. Fija en el borde superior, presente en TODAS las pantallas admin.

Lado Izquierdo (Pulso del Sistema)

- **Reloj Digital Hora Colombia** formato `HH:MM:SS` (24h). Actualiza cada 1 s. Fuente monoespaciada `'JetBrains Mono'`.
- **Telemetría de Red (Ping):** medidor en tiempo real `Ping: 24ms`. Indicador LED y color:
 - **Verde:** Online / Latencia <100ms.
 - **Amarillo:** Lag detectado / Alta carga (>100ms).
 - **Rojo:** Offline / Bajo ataque (no responde tras 3 reintentos).

Implementación: cada 5 s envía petición `HEAD` a `/api/health` (endpoint Edge Function vacío). Mide `Date.now() - tStart`. Grace period de 3 reintentos antes de marcar rojo.

Centro (Notificaciones)

Icono campana con burbuja roja dinámica. Click despliega panel flotante:

- "#X Pagos Bre-B por confirmar" → click navega a Pendientes filtrado por estado `RESERVADO_HV`.
- "#X Órdenes listas para despacho" → navega a Pendientes filtrado por `POR_ENVIAR`.
- "#X Leads para Compra de Equipos" → navega a Compramos Equipo.
- "#X Alertas críticas Centinela" → navega a Centinela (NUEVO v24.18.24).

Persistencia: solo desaparecen cuando el estado del objeto cambia en BD. Realtime via Supabase channels: la burbuja roja se actualiza sin refresh.

Lado Derecho (Usuario + Kill Switch)

C.1 Burbuja de Usuario

Foto circular con anillo de luz (Glow) por rango:

- **Dorado:** Super Administrador (usuario maestro). Acceso total.
- **Morado:** Editor / Logística (bloque I).
- **Gris:** Invitado (solo lectura bloques I y III).

Menú desplegable al click:

- Edición de Perfil (Nombre, Foto, Email válido, Cambio Contraseña).
- Seguridad: Activación/Desactivación MFA TOTP.
- Badge de Rango con bloques accesibles.
- Cerrar Sesión seguro con limpieza de caché.

C.2 Kill Switch (Interruptor de Pánico y Mantenimiento)

Toggle protegido (requiere deslizamiento lateral, no es simple click). Exclusivo para Super Administrador y el usuario maestro.

Modos:

1. Mantenimiento Programado:

- Configurar temporizador y mensaje personalizado.
- Banner elegante no intrusivo en toda la web: "Regresamos en [Contador HH:MM:SS]".
- Tras el countdown, sitio entra Modo Offline.

2. Pánico "CERRAR AHORA" (botón rojo neón):

- Confirmación por deslizamiento.
- Cierre inmediato del sitio público.
- Panel admin permanece ONLINE para los rangos autorizados.

Fase Activa de Mantenimiento:

- Transacciones MP/BRE-B activas pausadas o canceladas.
- Acceso a Home, catálogo, burbujas y pagos bloqueado.
- Web entra Modo Offline Total con `<MaintenanceMode />` (componente con fondo negro + logo respirando + mensaje base + mensaje custom + botón WhatsApp directo).

Vuelta a ON:

- `MaintenanceBanner.jsx` corre script de verificación cada 10 s en cliente público; al detectar cambio a ON, recarga automáticamente la página (sin necesidad de F5 del usuario).
- Servidor envía `Cache-Control: no-store` mientras está en mantenimiento para que el navegador no guarde la pantalla offline.

6.2.2 Sidebar Izquierdo (App-Launcher)

Componente: `src/components/admin/AdminSidebar.jsx`. Fijo en el lado izquierdo en TODAS las secciones.

Estado Contraído (default)

Solo 4 iconos neón + lupa:

1. **Icono Caja:** Bloque I (Operación Diaria).
2. **Icono Usuarios:** Bloque II (Gestión del Cliente).
3. **Icono Gráfica:** Bloque III (Inteligencia y Seguridad).
4. **Icono Engranaje:** Bloque IV (Configuración y Estructura).
5. **Icono Lupa (Ctrl+K):** Búsqueda Rápida de Comandos en la base del sidebar.

Puntos de notificación neón aparecen sobre los iconos cuando hay tareas pendientes en ese bloque.

Estado Expandido (Hover)

Se desliza panel glassmorphism que muestra:

- Nombre del Bloque en cabecera.
- Lista de "Apps" (sub-módulos) con títulos claros:
 - **Bloque I:** Dashboard · Creador de Fichas · Catálogo Activo · Pendientes · Vendidos · Devueltos · Compramos Equipo · Archivados · Centinela.
 - **Bloque II:** Comunidad URG · Centro de Mensajería · FAQ Hub.
 - **Bloque III:** BI / Estadísticas · Profit Manager · Report Builder · Black Box.
 - **Bloque IV:** Tributario · Admins · Panel Maestro · Home & Footer.

Búsqueda Rápida (Modal Mac Spotlight) · task #261

Atajo `Ctrl+K` (o `⌘K` en Mac) abre modal centrado tipo Mac Spotlight:

- Input grande en la parte superior con placeholder *"¿A dónde vas, comandante?"*.
- Lista de resultados debajo, agrupados por categoría (Productos, Pedidos, Administradores, Acciones).
- Búsqueda fuzzy en `lib/adminSearchIndex.js` que indexa toda la metadata accesible.
- Resultados con keyboard navigation (`↑/↓/Enter`).

Breadcrumbs

Línea de ruta dinámica en la parte inferior de la barra superior global (no del sidebar): `Operación > Pendientes > Detalles`. Cada nivel es link para retorno rápido.

6.2.3 Pantalla Principal · "Centro de Control Inmediato"

Componente: `src/pages/admin/Dashboard.jsx`.

A. Monitor de Actividad y Tráfico

- **Tráfico en Vivo:** contador grande de usuarios activos en tiempo real (consulta a `bi_live_sessions` cada 10 s).
- **Historial de Movimientos (Auditoría de logs):**
 - Muestra últimas 5 acciones (ej: "Venta #102 aprobada", "Precio actualizado en Ficha #45").
 - Botón [Expandir] despliega hasta 30 movimientos con scroll interno.
 - Histórico completo persiste en Bloque III (Black Box) para auditorías.

B. Módulo de Comunicación Crítica

Controla el mensaje principal del cliente. Dos tarjetas de edición rápida:

1. **Banner Global (barra de anuncios):** aparece en la parte más alta de la tienda (sobre el logo):

- Toggle activación (OFF por defecto).
- Inputs para texto y cronómetro (cuenta regresiva visible o no).

- Color Picker para fondo y fuente.

2. Hero & Copy Editor:

- Campos para Título Principal y Sub-texto de Envío.
- Sistema de rescate con "Texto Estándar de Marca" debajo.
- Campo para banner imagen del Hero con tamaño sugerido + simulador previo.
- Botón [Actualizar Home] para sincronización instantánea.

C. Módulo de Vitrina Dinámica

Componente: `src/pages/admin/HomeFooter.jsx` . Storyboard vertical con tarjetas de bloque:

- Cada tarjeta es un bloque del Home (Hero, Burbujas CTA, Oferta del Día, Más Vendidos, Compramos tu PC, Diseño Web Pro, Nuestra Tienda, Periféricos, Directorio SEO).
- **Estética:** glassmorphism con borde reactivo (Morado neón ON / Gris mate OFF).
- **Drag Handle (6 puntos):** reordena secciones instantáneamente con `react-dnd` o equivalente.
- **Botón [i]:** explica qué hace cada sección y qué ficha usa.

Organización en dos naturalezas:

- **A. Bloques de Estructura Fija:** Hero, CTA, Oferta del Día, Más Vendidos, Compramos PC, Diseño Web Pro (banner), Nuestra Tienda, Periféricos (desactivable), Directorio SEO.
- **B. Bloques de Inyección Dinámica:** categorías creadas en Módulo 3 del Creador (Mouses, Sillas Gamer, etc.). Por defecto "Huésped" (dentro de Periféricos); promovibles a Bloque Independiente.

Gestor de CTAs y Datos (panel lateral por bloque):

- Texto del botón editable.
- Destino del link (Burbuja Interna / WhatsApp / URL externa).
- Límite de items mostrados (1, 2, 3, 4, 8, 12).
- Estilo de Ficha (Tipo 1 Vertical / Tipo 2 Compacta).
- Badge de Resalte (NUEVO / OFERTA).

Telemetría e Inteligencia de Inventario:

- Contador Vital "  14 Items en Stock".
- Alerta de Vacío con triángulo amarillo si 0 productos.
- Indicador de Salud SEO basado en completitud.

D. Módulo de Aura y Neón

Sistema de control de identidad visual con bordes animados:

1. Quick-Actions:

- Master Switch (Modo Energía vs Modo Minimalista, default OFF).
- Selector de Presets:
 - ADN URG (Morado sólido #8A2BE2).
 - CRAZY URG (Dual Morado/Verde Neón).
 - Cyber-Black (Blanco frío y Gris).
 - Orgullo Patrio (Tricolor con desenfoque).
 - Alerta de Oferta (Pulso rojo intenso).

2. Panel Telemetría Avanzada:

- Slider Velocidad de Flujo (Estatismo → Plasma).
- Slider Intensidad de Glow.
- Toggles para Efecto Lava Lamp y Aceleración por Scroll.

3. Configuración de Grosor:

- Modo Discreto (3 px): profesional default.
- Modo Impacto (6 px): lanzamientos.

E. Modo Vista Previa (Live Preview)

Recuadro al final de la página que simula el Home en tiempo real:

- Renderiza cambios automáticamente al modificar Hero, Aura, etc.
- Interruptor PC / Móvil para previsualizar ambos formatos.
- Estado borrador hasta presionar "Actualizar Home".

F. Master Publish

- Botón [GUARDAR CAMBIOS] con animación "Sincronizando..." y check de éxito.
- Modal de Confirmación con SÍ verde / NO rojo.
- Barra de Cambios Pendientes alerta al intentar salir sin guardar.

6.3 Creador de Fichas URG

Componente: `src/pages/admin/CreadorFichas.jsx`. Es la herramienta más importante del Bloque I. Tras el refactor profundo (task #252) tiene tabs sticky con preview real (no placeholder).

Módulo 1 · Constructor de Fichas (8 bloques lógicos)

Bloque 1 · Control de Inventario (obligatorio):

- Cantidad disponible (campo numérico).
- Si >0 → web muestra "Disponible" verde + CTAs Agregar/Comprar.
- Si =0 → "Agotado" rojo + botón WhatsApp + 3 alternativas similares.

Bloque 2 · Identidad y Estado (obligatorio):

- Nombre Marketing (max 60 caracteres recomendado).
- Estado del Equipo: Nuevo / Como Nuevo / Excelente / Bueno (selector visual con cápsulas de color).
- Categoría (Gaming / Hogar-Estudio / Creadores / Empresarial).
- Ranking Manual #1, #2, #3 (selector visual de ranking: task #246, soporta N/A o 0 para fondo de catálogo). Si el admin no asigna número, el producto va al catálogo general sin competir por el podio del Home.

Bloque 3 · Especificaciones Vitales (obligatorio):

- Marca, Modelo (texto).
- Memoria RAM (texto, ej: 16GB DDR5).
- Gráficos (texto, ej: NVIDIA RTX 4060).
- Almacenamiento (texto, ej: 512GB SSD NVMe).
- Garantía (select fijo: 3, 6, 12, 24, 48 meses).
- Promoción Servicio Técnico (Toggle ON/OFF, default ON: "GRATIS 1 año").

Bloque 4 · Configuración Visual y Física (obligatorio):

- Selector de Color: nombre + código hex (genera cuadrito automático).
- Tamaño (Pequeño / Mediano / Grande / Muy Grande).
- Peso (Peso Pluma / Liviano / Moderado / Pesado).
- Descripción "Sobre este producto" (textarea libre).

Bloque 5 · Gestión de Detalles Adicionales (opcional, dinámico):

- Botón [+ Agregar Nuevo Descriptor]: crea campo nuevo (Refrigeración Líquida, Material del Chasis, etc.).
- Memoria del Sistema: los descriptores creados se guardan y aparecen como sugeridos en futuras fichas.
- Sugerencias pre-cargadas: Medidas, Especificaciones, Materiales, Estilo, Adicionales.
- Selector numérico "Orden de Aparición (TOP)" por descriptor.

Bloque 6 · Galería Inteligente:

- Zona Drag & Drop hasta 9 fotos y/o videos.
- Reordenable por arrastre (la primera es miniatura principal).
- **Cropper integrado** con layout 2-columns (task #259):
 - Panel izquierdo: imagen original con controles de pan/zoom.
 - Panel derecho: thumbnails grid 2×5 con preview de cada recorte.
- **Drag con shift** para arrastrar imagen en pan natural.
- **Optimización automática:** WebP, max 1920 px, sanitización EXIF (privacidad), compresión a calidad 0.85.
- **Encuadre Ficha #1 (Oferta del Día) específico:** sistema 16:9 con cropper dedicado (task #255 + #257).
- **Galería Inteligente inline** que muestra foto + previsualizaciones en vivo de cada formato de ficha donde aparecerá (task #254).

Bloque 7 · Calculadora de Precios e IVA:

- Precio de Lista (numérico).
- Selector de Descuento: Toggle Porcentual (%) o Monto Fijo (\$).
- Sistema de descuentos dinámico (task #241) con preview de precio final + ahorro.
- **Lógica de Cálculo (Regla de Oro URG):**
 - Si Precio Final $\leq$ \$2.618.700 (50 UVT) → "Excluido de IVA (Art. 424)".
 - Si Precio Final $>$ \$2.618.700 → "Gravado 19%" (Subtotal = Total / 1.19, IVA = 19%).
- **Semáforo UVT:** círculo verde (excluido) o azul (gravado) junto al precio.
- **Conexión DIAN** vía proxy interno (informativo) para validar tarifa vigente.

Bloque 8 · Publicación y Posicionamiento:

- **Visualización Previa:** modal con vista exacta de Ficha 1, 2 y 3: usando los **componentes reales** (task #252, antes era placeholder).
- **Sistema de Turnos "Oferta del Día":**
 - Selector de Posición (1, 2, 3...) o N/A.
 - Contador de Unidades.
 - Estado de Agotamiento auto-activa "Efecto Dominó" (promoción automática al desocuparse posición).

- **Lógica Multicategoría:** un producto puede pertenecer hasta 2 categorías; solo en la principal puede ser Top 1.
- **Acción Final:**
 - Botón **Publicar** (solo se habilita con todos los obligatorios completos). Confirmación con banner morado "Producto vinculado a la Tienda Digital con éxito".
 - Botón **Guardar como Borrador**.
- **Validación y Progreso:** botón Publicar deshabilitado y resalta en rojo campos obligatorios faltantes. Barra de progreso muestra qué tan cerca está la ficha (ej: 60%).

Módulo 2 • Gestión de Borradores

Componente: `src/components/admin/BorradoresTable.jsx`.

- **Autoguardado** cada 30 s si hay cambios.
- **Indicador de estado:** Manual vs Automático.
- **Tabla de Gestión:**
 - Identificador Temporal (TEMP-452).
 - Origen (Express vs Constructor Completo).
 - Barra de Progreso Inteligente con tooltip "Falta: Galería (3) y Calculadora IVA".
 - Fecha de Última Edición.
 - Miniatura visual (o icono  si vacía).
- **Acciones:** Vista previa, Duplicar (útil para 5 PCs con specs similares), Editar (icono lápiz), Descartar (icono basura) → papelera 24 h con snackbar Undo.
- **Acciones Masivas:** checkbox + Eliminación Masiva / Publicación en Lote (solo 100%).
- **Filtros:** por responsable, por progreso, por categoría/marca, por antigüedad.
- **Sistema de Bloqueo de Edición (Record Locking):** si dos admins abren el mismo borrador, candado visual "Alejandro está editando esto ahora".
- **Validación Pre-Publicación:** escaneo final que verifica precios > 0, formato de fotos OK, etc. Si falla, barra de progreso se torna roja al final.
- **Alerta de Cambio de UVT:** borradores con >30 días muestran icono  "Revisar cálculo de IVA antes de publicar".

Módulo 3 • Nuevas Categorías y Ecosistema Web

Componente integrado al Creador.

- **Creación de Nueva Categoría:**
 - Nombre (Componentes, Mouses, Sillas Gamer).
 - Identificador Visual (icono SVG/PNG o color sólido) para burbujas Home.
 - Ubicación en Menú (Toggle: aparecer en hamburguesa o quedar como Periférico).
- **Sincronización Automática:** menú, filtros, selector del Creador.
- **Atributos Heredados:** el admin decide qué descriptores opcionales se vuelven estándar para esa categoría.
- **Visibilidad Global:** botón maestro Apagar para ocultar la categoría sin borrar datos.

Módulo 4 • Entorno de Pruebas (Sandbox)

Sistema sandbox profesional (task #248-#250 refactor).

4.1 Activación

- Master Switch [MODO TEST: ON]. Bordes amarillos de advertencia en la interfaz al activarse.
- **Seed automático de 16 fichas** (task #250) que pueblan `KEY_PUBLISHED` para que el admin vea el sitio "lleno" desde el primer segundo.
- ON/OFF gobierna admin Y tienda: al apagar, la tienda vuelve a su catálogo real sin reset.
- **Reset a Fábrica** disponible siempre (botón [Purgar Datos de Prueba]).

4.2 Fichas Express (Carga Ágil)

Permite crear productos solo con 6 Pilares Visuales:

- Imagen de Portada.
- Nombre Marketing.
- Estado.
- Marca y Modelo.
- Precio.
- Trío de Specs (procesador + RAM + almacenamiento en un solo input).

4.3 Generador de Población Dummy

Botón [Inyectar Contenido Dummy] crea 12 fichas mezcladas (Portátiles, Gaming, Periféricos) con fotos y specs de relleno.

4.4 Espejo de Pruebas (Home Test Window)

Vista especial accesible solo para admin. Permite probar Efecto Dominó, cambio Disponible↔Agotado, etc.

4.5 Simulador de Pasarela de Pagos

- Transacciones Fantasma con credenciales de prueba MP.
- Validación de notificaciones (Resend) sin descontar inventario real.

4.6 Limpieza de Rastro

Botón [Purgar Datos de Prueba] con aviso crítico rojo: *"¿Estás seguro? Esta acción eliminará permanentemente todos los productos y ventas simuladas. Esta operación no se puede deshacer"*. Confirmación con [SÍ].

4.7 Sandbox "Solo Prueba"

Modo donde las fichas sandbox solo se ven en admin, no en la tienda pública (filtrado en `lib/adminProducts.js → getCatalogoPublico()`). Modal de warning al activar/desactivar. Real tiene prioridad sobre Sandbox en todos los rankings (task #251).

6.4 Flujo Operativo

Dividido según el ciclo de vida de la venta.

6.4.1 Catálogo Activo

Componente: `src/pages/admin/CatalogoActivo.jsx`. Motor de lo que el cliente ve en la tienda.

Dashboard de Posicionamiento y Ranking (Global)

A. Contador de Slots Libres: Indicador dinámico "Gaming: 2/3 ocupados | Hogar: 1/3 | Creadores: 3/3 | Ofertas: 1/1". Si hay disponibilidad (1/3) → número en amarillo/naranja para advertir oportunidad.

B. Tablero Visual de Tops (Grid 3×2): task #260: 4 tablas compactas (una por categoría madre) + 1 destacada para Ofertas.

Estructura por categoría:

• (puesto 1, 2, 3...).

- Miniatura + Nombre Marketing.
- Hipervínculo al Constructor (lleva al bloque exacto de Posicionamiento).
- Selector manual de posición + Drag & Drop entre filas Y entre categorías (task #265: antes drag intra-categoría solamente).
- Indicador de visibilidad (Top del Home vs Fondo de Catálogo).
- **Bordes por podio:** dorado (#1), plata (#2), bronce (#3): task #251.
- **Prioridad real > sandbox** en el orden visual (task #251).

C. Módulo Oferta del Día:

- Toggle ON/OFF maestro.
- Tabla con identificación PC + Ranking + Semáforo UVT.
- Selector Modo Operativo:
 - **Modo Único (Exclusivo):** gestión 1 oferta/día con cierre auto al agotarse stock. Sistema cíclico: rotación automática a las 8 AM Hora Colombia (task #273, `tryRotateOfertaDelDia()`).
 - **Modo Múltiple (Carrusel) · default.**
- Botón "Forzar rotación ahora" (task #274, `forceRotate API`).

Dashboard de Gestión Individual

A. Herramientas de Navegación:

- Buscador Global Predictivo en tiempo real.
- Filtros Rápidos (Categoría / Estado de Stock / Estado UVT).
- Contador de resultados.

B. Acciones por Producto:

-  Modificar (abre Constructor con datos precargados).
-  Despublicar (Botón de Pánico): mueve a Borradores para corregir.
-  Borrar (doble confirmación con texto).
-  Ver Historial.

C. Bitácora de Fichas (Auditoría):

 Cada acción genera Tarjeta de Evento con:

- Timestamp (Día, mes, año, hora).
- Identidad Autor (Modificado por: [administrador]).
- Acción Ejecutada (Cambio de Precio / Actualización Stock / etc.).
- **Antes vs Después con resaltado:** rojo lo eliminado, verde lo nuevo.

D. Lógica de Modificación Inteligente:

- Recálculo automático IVA al cambiar precio (Semáforo UVT actualiza).

- Sincronización de Imágenes sin resubir todo.

Filtros adicionales post-v18

- **Toggle prueba/real** en header (task #247).
- **Filtro únicas/múltiples** (productos con variantes de color).
- **Suma de stocks expandible** por agrupación.
- **Auto-archivar stock 0:** al llegar a 0 unidades, mueve al Cementerio automáticamente con tag `[VENDIDO]`.

6.4.2 Gestión de Pendientes (Workflow Logístico)

Componente: `src/pages/admin/Pendientes.jsx`. Núcleo de inteligencia operativa.

Globo de Notificación (Alerta Maestra)

Item "Pendientes" del sidebar muestra Badge con el número de órdenes en `RESERVADO_HV` (Reservado Hasta Verificación). Admin sabe sin entrar cuántos BRE-B esperan validación.

Interfaz Tabla de Control

Cronológica (más antiguo primero: evita retrasos). Buscador multivariable (Nombre cliente / Email / Teléfono / Lote / Cédula / Ciudad / ID transacción).

Pestañas superiores (filtros de estado):

- [TODOS] Vista general.
- [RESERVADO HV] Pagos reservados hasta verificación.
- [LISTO PARA ENTREGA] Cliente eligió Recoger en Local.
- [POR ENVIAR] Pagos aprobados listos para empaque.
- [ENVIADO] En ruta con transportadora.
- [ENTREGADO] Entregados al cliente.
- [DEVUELTOS] Logística inversa/garantías.

Badges de pasarela en cada fila: `[MP]` azul o `[BRE-B]` morado.

Bloque de Verificación de Pago (Trigger BRE-B)

Cada fila `RESERVADO_HV` tiene panel de decisión:

- **Visor de Comprobante:** miniatura PDF/JPG. Click abre Lightbox con zoom.
- **[CONFIRMAR PAGO]** (verde): ejecuta `consumir_stock` server-side. Estado → `POR_ENVIAR`. Dispara Email B.2 con PDF de orden.
- **[RECHAZAR PAGO]** (rojo): modal obligatorio "Motivo Rechazo". Ejecuta `liberar_reserva`. Stock vuelve a "Disponible". Dispara Email B.4 con motivo al cliente.

Workflow de Despacho

Al cambiar estado a `ENVIADO`, se despliega formulario:

Si cliente eligió Envío:

1. Selector Transportadora (Servientrega / Interrapidísimo / Envía / DHL / Otra).
2. Número de Guía (texto).
3. URL de Seguimiento (link directo al portal).
4. Botón [CONFIRMAR DESPACHO] dispara Email C.1.

5. Reloj Post-Venta: 5 días (Bogotá) o 7 días (Nacional) para auto-disparo Email C.3 (Reseña Google).

Si cliente eligió Recoger en Local:

- Estado cambia automáticamente a [LISTO PARA ENTREGA] + dispara Email C.2 (Instrucciones de Recogida).
- Reloj Post-Venta de 5 días.

Cierre de Ciclo

- **[ENTREGADO]:** dispara Email C.3 (Reseña).
- **Periodo de Gracia 48h:** pedido visible en pestaña ENTREGADO para revertir errores.
- **Transición Final:** tras 48h, metadata se archiva en VENDIDOS, Pendientes queda limpio.
- **[DEVUELTO]:** vuelve a Pendientes con alerta ámbar para gestionar garantía.

Auditoría / Timeline

Cada fila tiene icono reloj → línea de tiempo:

- Creación: "Pedido generado vía [MP/BRE-B] el [Fecha/Hora]".
- Carga de Soporte: "Comprobante subido por cliente".
- Validación Humana: "Pago aprobado por [Admin] el [Fecha]".
- Rechazos con motivo.
- Saltos logísticos con identidad y guía.

Cooldown visual 3s (v18.6)

Tras una acción crítica (Confirmar/Rechazar) el botón se bloquea por 3 segundos con animación visible para evitar doble-tap. Toast verde de éxito al completar.

6.4.3 Vendidos (Memoria Histórica y Contable)

Componente: `src/pages/admin/Vendidos.jsx`. Archivo definitivo perpetuo de transacciones exitosas.

Ficha del Cliente (Data de Valor)

Cada registro es un Expediente:

- Identidad Producto (Nombre Marketing, Lote #2603-104A, Specs finales).
- Identidad Cliente (Nombre completo, Cédula/NIT, Email, Teléfono, Dirección).
- Métrica Económica (Precio final, MP o BRE-B, desglose IVA si aplicó).
- Temporalidad (Fecha/hora exacta de transición a ENTREGADO).

Funcionalidades

- **Buscador de Trazabilidad:** filtra por Nombre, Lote, Email, Cédula.
- **Centro de Descargas:** acceso permanente al PDF de la Orden ("factura" de URG).
- **Trigger de Reversión (seguridad):**
 - [Mover a Pendientes]: corregir errores.
 - [Mover a Devueltos]: logística inversa.

6.4.4 Devueltos (Gestión de Contingencias)

Componente: `src/pages/admin/Devueltos.jsx`.

- **Inactivo y en escala de grises** mientras no haya registros.
- **Modal de Justificación obligatorio** al mover un producto aquí. Campo "Motivo de Devolución".

- **Tríptico de Destino:**
 - [Regresar a Catálogo - DISPONIBLE] (Stock-In +1 automático).
 - [Regresar a PENDIENTES] (nuevo envío/ajuste).
 - [Marcar como VENDIDO] (error administrativo confirmado).
- **Trazabilidad:** mantiene vínculo con la Orden de Pedido original.

6.4.5 Compramos tu Equipo

Componente: `src/pages/admin/CompramosEquipo.jsx`. Centro de Mando de Valoraciones.

Barra de Estadísticas Rápidas (Radar de Operaciones)

3 tarjetas glassmorphism horizontales + tiempo de respuesta promedio:

A. Pendientes de Valorar: número en amarillo, parpadea si >5. **B. En Negociación:** número en azul cian. **C. Cerrados Hoy:** número en verde neón con . **D. Tiempo Respuesta Promedio** (ej: "1h 4m 20s"). Rojo si >60min.

Tabla Gestión en Tiempo Real

Columnas:

- Checkbox Maestro + Individuales (acciones masivas).
- Estado (NUEVO / EN ANÁLISIS / OFERTA ENVIADA / AGENDADO PARA PERITAJE / COMPRADO / DESCARTADO).
- Fecha/Hora (Más reciente primero).
- Cliente (Nombre + icono).
- Equipo (Modelo + tipo).
- Origen (Formulario completo / Fotos Cero Estrés).
- Dropdown directo de estado.
- Acciones Rápidas (Ver Detalle / WhatsApp pre-cargado / Cambiar Estado).
- Notas Internas (quick edit).

Filtros + Indicador de Prioridad (Grado A o CPU alto → borde dorado).

Technical Briefing (Panel Detalle Lateral)

Se desliza desde la derecha al click en cliente. 5 bloques tácticos:

A. Cabecera de Misión. Nombre, tipo equipo (LAPTOP/TORRE icono), preferencia de pago (Efectivo / Crédito URG), botón flotante [CONTACTAR POR WHATSAPP] con mensaje pre-cargado.

B. ADN del Equipo (Especificaciones). Bloque estilo terminal de comandos:

```
> IDENTIFICACIÓN: [Marca y Modelo]
> PROCESADOR: [CPU] ([Etiqueta personalidad: Nivel Bestia])
> MEMORIA RAM: [Capacidad]
> ALMACENAMIENTO: [Capacidad] - [SSD Cohete / HDD Tortuga]
```

C. Escaneo de Sensores. Dinámico:

- Si LAPTOP: Pantalla (tamaño/estado), Salud Batería, Cargador Original.
- Para ambos: Estado de Unidad (Nuevo/Usado/Dañado), Grado Estético (A/B/C/D con descripción).

D. Briefing de Usuario (Inteligencia de Negocio). "¿Por qué vende?" + "¿Qué busca después?".

E. Galería de Evidencia (Cero Estrés). Cuadrícula 3 miniaturas → tap abre Lightbox con zoom (lupa 280×280 px, zoom 500%: task #239). Visor con **createPortal** que escapa del aside (task #236).

Funcionalidad Archivo y Seguridad

- **COMPRADO** mueve a Historial. Fotos se auto-purgan a los **31 días** (deben ocupar mucho espacio).
- Edición en Dos Niveles (Tabla rápida / Detalle profundo).
- Push Real-time: nuevo cliente aparece con resplandor morado parpadeante.
- Sincronización entre admins (si uno marca "En Análisis", otros lo ven).

6.4.6 Archivados (Cementerio Lógico)

Componente: `src/pages/admin/Archivados.jsx`.

- **Tabla dinámica:** miniatura + nombre, etiqueta [VENDIDO] gris o [ELIMINADO] rojo, fecha evento, ventas asociadas.
- **Acciones:** Restaurar (desarchivar al catálogo vivo) | Eliminar Definitivamente (confirmación con texto "ELIMINAR" + advertencia inconsistencias BI).
- **Acciones Masivas:** Seleccionar todos + Vaciar Papelera (seguridad de 2 niveles).

Lógica del ciclo de vida del producto:

1. **Activo:** Stock > 0, visible en catálogo.
2. **Transición por Venta:** stock = 0 → `hidden: true` + tag `[VENDIDO]`, desvincula de ODD/MasVendidos automáticamente.
3. **Transición por Descarte:** admin presiona Eliminar → tag `[ELIMINADO]`.
4. **Archivado:** no consume recursos front-end, pero alimenta BI.
5. **Fin de Vida:** borrado físico solo por decisión expresa del admin, idealmente tras cierre fiscal.

6.X CENTINELA (NUEVO · Salud Operacional · v24.18.24)

Componente: `src/pages/admin/Centinela.jsx`. Sección dedicada introducida en v24.18.24 que NO existe en V1.1. Monitorea la salud del sistema de stock + pagos + sync.

Salud Operacional (Alertas)

Panel principal con `<AlertasOperacionalesPanel />` que agrupa fallos en categorías:

- **Reservas huérfanas BRE-B:** órdenes con `expira_at < now()` que aún no se liberaron (señal: cron no corrió o falló).
- **Webhooks MP no confirmados:** órdenes en `PASARELA_ABIERTA` >1h.
- **Productos con stock_reservado > stock real:** debería ser imposible por el CHECK constraint, pero monitor lo cubre por defensa.
- **Edge Functions con errores recientes:** consume logs de `mp-webhook` y `cleanup-stock-fantasmas`.
- **Inconsistencias entre `orders.estado` y `productos.reservas`:** correlación cruzada.

Cada categoría es `<AlertaGrupo />` con cards `<AlertaItem />` (severidad + descripción + CTA "Resolver").

Sección Pendientes Cooldown

Lista de las últimas 50 órdenes con timer de cuándo expirará su reserva → permite anticipar cancelaciones.

Reportar Centinela

Función `reportarCentinela()` en `mp-webhook` y `cleanup-stock-fantasmas` que registra cada movimiento crítico en tabla `centinela_events`. Útil para post-mortem.

6 · Continúa con BLOQUE II (Gestión del Cliente)

Por longitud, esta sección continúa en la **Parte 5** del documento.

6 • BLOQUE II • GESTIÓN DEL CLIENTE

6.5 Comunidad URG

Componente: `src/pages/admin/Comunidad.jsx`. Backend en `src/lib/comunidad.js`.

Tabla de registro automático que se alimenta cada vez que una venta se marca **EFFECTUADA** (pago MP confirmado o BRE-B aprobado por admin). Centraliza la base de datos de clientes recurrentes y permite gestión CRM ligera.

Lógica de Alta

Cuando el admin aprueba una venta en Pendientes, `comunidad.js` → `upsertCliente(cedula, datos)` busca si la cédula ya existe:

- Si existe → añade el nuevo pedido al `historialCompras` y actualiza `inversionTotal`.
- Si no existe → crea fila nueva con datos del checkout.

Estructura de Tabla (6 columnas visibles)

1. **Cédula / ID:** llave principal de búsqueda.
2. **Cliente:** Nombre + Apellido + botón verde de WhatsApp (icono pequeño que abre chat directo).
3. **Historial de Compras:** celda desplegable con botón "Ver Historial". Al click despliega lista debajo:
 - `#2603-104A | MacBook Pro M2 | [valor] | 12/03/26`
 - `#2602-085A | Cargador 96W | [valor] | 10/02/26`
4. **Valor Total:** suma acumulada `inversionTotal` (útil para identificar VIP).
5. **Dato de Ubicación:** ciudad del último envío (formato "Bogotá, Cundinamarca", task #210).
6. **Acciones (...) tres puntos:** menú: Ver última Orden de Pedido (PDF) | Iniciar nueva venta con estos datos | Borrado Total (Habeas Data).

Sistema de Filtrado y Orden

- **Buscador Universal:** filtra por Cédula o Nombre en tiempo real.
- **Ordenamiento por Columnas:** click en cabecera ordena (Valor Total, Cliente alfabético, Última compra).
- **Filtro por Tiempo:** dropdown "Clientes que compraron en: [Marzo] [2026]".
- **Filtro de Inactividad:** botón "Clientes que no compran hace más de 6 meses": útil para remarketing.

6.5.1 Módulo de Privacidad • Derecho al Olvido

Implementa el "Borrado Total" (Habeas Data) descrito en el V1.1, pero como **Anonimización quirúrgica** (no DELETE físico):

```
async function anonimizarCliente(cedulaId) {
  await supabase.rpc('anonimizar_cliente', { p_cedula: cedulaId });
  // RPC limpia: nombre, apellido, email, telefono, whatsapp, direccion, cedula
  // Preserva: valor_venta, fecha, producto (con marca "Cliente Anonimizado")
  await dispararEmailE4(); // Confirmación Habeas Data
}
```

Resultado: el balance contable sigue siendo real, pero no hay rastro de quién pagó.

Bloqueo de Re-ingreso: si el cliente vuelve a comprar tras solicitar borrado, el sistema lo trata como Cliente Nuevo (inversión total = \$0).

Ticket de Garantía Huérfano: al anonimizar, el sistema genera ticket asociado solo al número de serie del equipo, para que si el cliente vuelve con el PC, este sirva como llave de búsqueda en lugar de su nombre.

6.5.2 Guía de Soporte (Botón [i])

Botón información junto al título del módulo. Al click abre modal con explicación:

- Qué es el Derecho al Olvido.
- Inventario de Eliminación (lo que SÍ se borra: identidad, contacto, logística, historial social).
- Garantía de Integridad (lo que NO se afecta: salud financiera, continuidad inventario, rastreo por hardware).
- Advertencia de Seguridad ("Una vez confirmada, los datos personales no pueden ser recuperados").

6.6 Centro de Mensajería y Automatización (Resend + Supabase)

Componente: `src/pages/admin/CentroMensajeria.jsx`. Es el "espejo" técnico de la tabla `plantillas_correo` en Supabase. Cada cambio aquí actualiza la BD que las Edge Functions consultan en tiempo real antes de disparar envíos a la API de Resend.

6.6.1 Configuración del Servidor y Remitente

- **Email Remitente Maestro:** `[correo de la tienda]`. Edita-ble desde el panel.
- **Validación de Entregabilidad:** el sistema valida que el dominio esté verificado en Resend (SPF/DKIM). Indicador LED verde/rojo junto al campo.
- **Dualidad de Notificación:** el mismo correo (o un Admin Mirror) recibe alertas en cada ID de Pedido exitoso.
- **Capacidad Envío Masivo:** conexión vía API con segmentos de Comunidad URG. Admin puede redactar anuncios manuales (ofertas/lanzamientos) y enviarlos a contactos opt-in.
- **Sanitizador de Texto:** elimina espacios/caracteres invisibles del email antes de guardar (previene errores de auth con la API).

6.6.2 Motor de Diseño: React Email

- Se usa React Email solo como herramienta de **maquetación y compilación**.
- El output final es HTML inline-CSS optimizado que se carga en Supabase.
- Las plantillas son **responsive** y probadas en >50 clientes de correo (Gmail, Outlook, Apple Mail, etc.).
- **Dark Mode adaptable** automáticamente.

6.6.3 Flexibilidad y Creación de Plantillas

- **Nomenclatura Dinámica:** variables como `{{nombre_cliente}}`, `{{id_pedido}}` son **renombrables** por el desarrollador.
- **Escalabilidad Modular:** agregar nuevo correo solo requiere crear plantilla en panel + configurar evento disparador.
- **Lógica de Reemplazo (Fallbacks):** valores por defecto cuando una variable está vacía (ej: `{{nombre_cliente}}` → "Cliente URG").

- **Independencia de Compilación:** admin puede cambiar copy rápido en panel; desarrollador maneja estructura base desde React Email.

6.6.4 Estructura por Pestañas

4 pestañas Dark Mode con acentos morado neón:

Pestaña 1 • Dashboard de Automatizaciones

Cuadrícula 3-4 columnas con cards organizadas en 5 Módulos Críticos:

1. **Captación de Hardware** (Compramos tu PC).
2. **Ciclo de Transacción y Validación de Pagos.**
3. **Logística** (Despacho, Confirmación de Entrega, Review Maps).
4. **Crecimiento y Fidelización** (Carrito Abandonado, Mantenimiento 6 meses, Plan Recambio 12 meses).
5. **Seguridad y Sistema** (Magic Link, Alerta Black Box, Habeas Data).

Anatomía de cada Card:

- Borde lateral izquierdo (4 px): **Morado #8A2BE2** flujo cliente o **Naranja #FF8C00** flujo admin.
- Icono dinámico (sobre, camión, rayo, escudo).
- Badge destinatario [ADMIN] o [CLIENTE].
- Switch de Estado (Toggle neón ON/OFF).
- Meta-data "Última ejecución: hace X horas".
- Botón "Configurar Plantilla".

Pestaña 2 • Editor de Contenido (Split View)

Columna Izquierda (Configuración):

- Trigger ID bloqueado (`pago_aprobado`).
- Destinatario (variable dinámica `{{email_cliente}}` o email fijo).
- Asunto con soporte de variables (ej: "  ¡Pago aprobado! Orden `{{lote_pedido}}`).
- Cuerpo (Editor Rich Text gris oscuro).
- Gestión de Adjuntos (selector + switch Generación Automática PDF Orden).

Columna Derecha (Herramientas):

- Panel de Variables (burbujas clicables `{{nombre_cliente}}`).
- Info-Botones (i) con tooltips: Definición + Origen (tabla.columna) + Ejemplo.
- Previsualización Real-Time (iframe con render escritorio + móvil).

Pestaña 3 • Configuración Global (Identidad & Remitente)

ADN visual de URG en todos los correos:

Identidad Visual (Header):

- Logotipo Cabecera (URG_FOOTER_2026.svg) cargable/actualizable.
- Inyectado de primero en parte superior central de todos los correos.
- Max 200 px ancho para móviles.

Configuración Remitente:

- Email Remitente editable.
- Nombre Remitente Alias ("URG TECHNOLOGY - Soporte Táctico").

Editor de Firma (Footer):

- Línea divisoria Morada `#8A2BE2`.
- Color de fondo del bloque legal (Gris carbón sugerido).
- Sincronización instantánea.
- Jerarquía de texto: ayuda destacada (blanco/morado), legal pequeño (gris claro 11-12px).

Contenido footer:

```
----- (línea divisoria morada)
Un saludo potente te desea,
Equipo URG TECHNOLOGY

¿Tienes alguna duda? Escribenos a [correo de la tienda]
o contáctanos al WhatsApp: [WhatsApp de la tienda].
----- (línea divisoria morada)
Si ya no deseas recibir nuestras actualizaciones, cancela aquí la suscripción (link)
Política de Privacidad de URG TECHNOLOGY (link)
URG TECHNOLOGY S.A.S.
© 2026 URG TECHNOLOGY. Todos los derechos reservados.
Tu futuro tecnológico comienza aquí.
```

Funcionalidad Cancelación:

- "Cancela aquí" → Edge Function que cambia `marketing_opt_out = true` en Supabase. No vuelve a recibir correos del Módulo 4 (Crecimiento), pero sí los obligatorios (Módulo 2: Pagos).
- "Política de Privacidad" → <https://urgtechnology.com/privacidad>.

Pestaña 4 · Seguridad y Herramientas (CRUD)

Tabla técnica Registro de Salida (Logs de Resend):

1. Estado (LED Verde Enviado / Amarillo En cola / Rojo Error).
2. Referencia (Trigger en fuente código).
3. Destinatario.
4. Adjuntos (icono PDF clicable → abre archivo en Storage).
5. Timestamp.
6. Acción Rápida (Botón Re-enviar).

Herramientas Globales:

- Filtros Solo Admin / Solo Cliente.
- Buscador por Email o por `{{lote_pedido}}`.
- Botón de Pánico (Reset a Versión Fábrica).
- Modo Prueba (envío real con datos ficticios a [correo de la tienda]).

Inspección de Datos (JSON View): Botón "Ver Datos del Evento" abre modal estilo terminal mostrando el JSON original enviado a Resend. Con botón "Copiar al Portapapeles". Útil cuando un cliente reclama que su nombre no aparece: si JSON dice `"nombre_cliente": null`, el problema es de origen (formulario), no de plantilla.

6.6.5 Diccionario de Triggers y Variables

3 acordeones en panel lateral del Editor:

-  **Variables del Proceso (Trigger-Specific):** `{{monto_total}}`, `{{metodo_pago}}`, `{{lote_pedido}}`.
-  **Variables del Cliente (Habeas Data):** `{{nombre_cliente}}`, `{{email_cliente}}`, `{{telefono}}`.

- **Variables de Sistema:** `{{fecha_actual}}`, `{{url_seguimiento}}`.

Catálogo Triggers activos:

- `solicitud_Compra_PC` → al enviar formulario retomas.
- `pago_manual_pendiente` → al subir comprobante.
- `envio_despachado` → al generar guía.
- `pago_aprobado` → al confirmar MP/BRE-B.
- `pago_rechazado` → al admin rechazar BRE-B.
- `cliente_anonimizado` → al ejecutar Habeas Data.
- `magic_link_solicitado` → al admin solicitar reset.
- `alerta_critica_blackbox` → al detectar MFA fail / fuerza bruta / anomalía geo.

Auto-detección: si se crea nueva columna en Supabase, aparece automáticamente como burbuja clicable.

6.6.6 Matriz de Operaciones · Plantillas Completas

Módulo 1 · Captación de Hardware

A.1 Notificación Técnica (Admin)

Asunto: ENCONTRAMOS UN CLIENTE QUE DESEA VENDER NOS SU PC: `{{modelo_pc}}` - `{{nombre_cliente}}`

Cuerpo con todos los datos técnicos del formulario (10 campos) y "Acción Requerida".

A.2 Confirmación Recibido (Cliente)

Asunto: Estamos revisando tu `{{marca_modelo_PC}}` - URG TECHNOLOGY SAS

Mensaje didáctico agradeciendo + tiempos de respuesta (24-48h hábiles).

Módulo 2 · Ciclo de Transacción y Validación de Pagos

B.1 Alerta Pago Pendiente Bre-B (Admin)

Asunto: PAGO POR VALIDAR (Bre-B): Pedido `{{id_pedido}}` - `{{nombre_cliente}}`

B.2 Confirmación de Pago + Orden de Pedido (Cliente)

Asunto: ¡Pago aprobado! Tu orden `{{id_pedido}}` está confirmada

Adjunta `ORDEN_URG_{{id_pedido}}.pdf`.

B.3 ¡Venta Exitosa! (Admin)

Asunto: NUEVA VENTA CONFIRMADA: Iniciar Envío - ID `{{id_pedido}}`

Resumen completo con producto, método pago, total, modalidad, destinatario, dirección, etc. Si cliente solicitó factura empresa: Razón Social + NIT.

B.4 Pago Rechazado (Cliente)

Asunto: Acción requerida: Actualización sobre tu pedido `{{id_pedido}}`

Incluye variable `{{motivo}}` editable por el admin al rechazar. Botón [REINTENTAR COMPRA AHORA] que lleva al carrito con producto pre-seleccionado.

Módulo 3 · Logística y Seguimiento

C.1 Notificación de Despacho (Cliente)

Asunto: ¡Tu equipo URG ya está en camino! - Guía: {{numero_guia}}

C.2 Confirmación Recogida Local (Cliente)

Asunto: 📦 ¡Todo listo! Ya puedes pasar por tu equipo a URG TECHNOLOGY.

C.3 Confirmación de Entrega + Reseña (Cliente)

- Trigger: cronómetro interno post-despacho.
- Bogotá/Recogida Local: 5 días.
- Nacional: 7 días.

Asunto: Tu opinión nos importa: ¿Cómo estuvo tu experiencia URG Technology?

CTA: [CALIFICAR EXPERIENCIA EN GOOGLE MAPS] → enlace a Google Reviews.

Módulo 4 · Crecimiento y Fidelización

D.1 Recordatorio Mantenimiento (Cliente)

Asunto: ¡Es hora de un mantenimiento gratuito para tu equipo!

Trigger: 6 meses post-compra.

D.2 Oferta Plan Recambio (Cliente)

Asunto: Evolucionar tu setup: ¡Es momento de una mejora!

Trigger: 12 meses post-compra.

D.3 Recordatorio Carrito Activo (Cliente)

- Trigger: cronjob 3h después de ID en CANCELADO.
- Filtro de seguridad: verificar que no exista orden POR_ENVIAR vinculada al mismo email.

Asunto: Tu equipo de alto rendimiento sigue disponible (por ahora).

Módulo 5 · Seguridad y Sistema

E.1 Magic Link / Acceso Seguro (Admin)

Asunto: 🗝️ Token de Acceso Seguro - ID: #AUTH-{{token_id}}

Botón [INGRESAR AL PANEL ADMINISTRATIVO]. Token caduca 15 min. Footer con IP, ubicación, dispositivo.

E.2 Confirmación Cambio de Seguridad (Admin)

Asunto: 🗝️ Seguridad URG: Tu contraseña ha sido actualizada

Botón [BLOQUEO DE EMERGENCIA] que congela la cuenta si fue alguien externo.

E.3 Alerta Crítica Black Box (Admin)

Asunto: [ALERTA CRÍTICA] Intento de Acceso No Autorizado - ID: #SEC-{{incident_id}}

Cuerpo con: Resumen del Incidente, Datos de Conectividad (IP, ubicación, dispositivo, fingerprint), Reporte de Incidente (15 datos técnicos), Acción Inmediata Requerida con botones [Ignorar Alerta] o

[BANEAR IP Y CERRAR TODO] (rojo neón).

E.4 Confirmación de Eliminación Habeas Data (Cliente)

Asunto: Confirmación: Tu privacidad es nuestra prioridad (Habeas Data)

Mensaje explicando qué se anonimizó y qué permanece bajo código cifrado para fines legales/contables.

6.7 Centro de Preguntas Frecuentes (FAQ Hub)

Componente: `src/pages/admin/FAQHub.jsx`. Backend en `src/lib/faqHub.js`.

Filosofía "Zero-Code" (CRUD)

- **Crear:** añadir nuevas dudas detectadas en soporte.
- **Publicar/Ocultar:** Toggle neón por ítem.
- **Ordenar:** Drag & Drop para priorizar.
- **Editar:** modificar textos, links, categoría.

Terminal de Creación

Cada nueva pregunta tiene campos obligatorios:

A. **Selector de Contexto:** Toggle entre:

- "Preguntas Frecuentes (FAQ)" → renderiza en `/preguntas-frecuentes` + Footer + Buscador General.
- ~~"Todo lo que debes saber (Diseño Web)":~~ descartado en v24.18.26 por migración. Las FAQs de Diseño Web ahora viven en damorus.com.

B. **Selector de Categoría:** menú desplegable (Hardware, Pagos, Metodología, Logística, Garantía, etc.).

C. **Campo Título:** pregunta. D. **Campo Descripción:** respuesta rich text. E. **Toggle Visibilidad:** Publicado / Borrador.

Anatomía del Ítem

- **Categoría** (filtro temático).
- **Título** (pregunta en negrita).
- **Descripción** (respuesta soporta texto, links, listas).

Gestor Maestro de Categorías

- Botón [+ Nueva Categoría] con nombre + icono representativo.
- Sincronización técnica con FK hacia `faq_items` para integridad de datos. Al borrar categoría, pregunta: "¿Mover a otra o eliminar?".
- Personalización Visual (color de acento + icono Lucide/FontAwesome).

Lista Preguntas Por Defecto (FAQ tienda)

14 preguntas activas distribuidas en categorías Logística, Transacciones, Respaldo, Soporte Técnico, Reseñas. Detalle exacto en el V1.1 §6.7.5: no se modificó el contenido base, solo el contexto de Diseño Web migró.

5.6 CMS Institucional (Privacidad + Compra Protegida)

Extensión del FAQ Hub. Permite editar las páginas `/privacidad` y `/compra-protegida` por **bloques**:

- Encabezado Dinámico (título + fecha "Última actualización").
- Constructor de Bloques (añadir/editar/eliminar bloques numerados).
- Campos por Bloque: Icono/Emoji + Título + Cuerpo Rich Text.
- Renderizado Independiente (las páginas piden el contenido más reciente a Supabase al cargar).
- **Server-Side Rendering** mediante Edge Function que serve la HTML con datos ya inyectados → cero parpadeo.
- **Respaldo Local** en caché si Supabase no responde.

6 • BLOQUE III • INTELIGENCIA Y SEGURIDAD

6.8 Estadísticas e Informes (Business Intelligence)

Componente: `src/pages/admin/BI.jsx`. Backend en `src/lib/biMetrics.js`. Dashboard de central de comando Dark Mode con acentos Neon Emerald & Purple.

Funciones de Control Global

- **Time-Travel Toggle:** selector maestro arriba que cambia TODO el panel entre Diario, Semanal, Bisemanal, Mensual, Semestral, Anual, Total.
- **Custom Dashboard:** sistema Drag & Drop / Ranking que permite al admin reordenar bloques según prioridad.

Requisitos Técnicos

- **Interactividad:** todo dinámico. Cambio de rango → gráficos animados (Framer Motion).
- **Conectividad:** datos provienen de Supabase + interacciones email de Resend.
- **Simplicidad:** modales para detalles profundos; primer plano impactante.

Módulo 6.8.1 • Inteligencia de Datos (31 métricas)

Cada métrica con icono (i) tooltip que explica: qué es + por qué importa + cómo se calcula.

I. Tráfico y Adquisición (6)

1. **Contador de visitas tiempo real:** registro histórico de usuarios.
2. **Origen de Tráfico (Referral):** `document.referrer` clasifica Instagram/Google/WhatsApp/directo.
3. **Tasa de Rebote:** sesiones que abandonan en <10s sin interacción.
4. **Visitantes Nuevos vs Recurrentes:** ID navegador en localStorage.
5. **Páginas de Destino Populares:** URL de entrada por sesión.
6. **Segmentación por Dispositivo:** User-Agent → Pie Chart (PC/Tablet/Móvil).

II. Interacción y Búsqueda (5)

7. **CTR del Buscador Interno:** volumen búsquedas vs clics en sugerencias.
8. **Top 10 Palabras más Buscadas:** tabla `bi_searches` agrupada (campo `query` corregido en v18 bug 4).
9. **Mapa de Clics por Secciones:** eventos en contenedores de categorías.
0. **Tiempo Promedio de Sesión.**
1. **CTR Banners de Oferta.**

III. Ventas y Conversión (4)

2. **Tasa de Conversión (CR):** pagos aprobados / visitantes únicos.
3. **Tasa de Abandono de Carrito:** A: carrito sin checkout / B: checkout sin pago.
4. **Ticket Promedio (AOV):** ingresos brutos / pedidos exitosos.
5. **Ventas por Categoría.**
6. **Días y Horas Pico de Ventas:** heatmap (responsive en v18.5).

IV. Finanzas y Comisiones (7)

7. **Ventas Generadas:** histórico total con IVA y envío.

8. **Ingresos Brutos vs Netos:** resta IVA 19% + comisión MP automática.

9. **IVA Recaudado Acumulado.**

0. **Inversión en Pasarela:** cálculo según plan MP activo (3.29% / 2.99% / 2.79% editable por mes).

1. **CAC:** gasto publicitario / clientes nuevos.

2. **ROAS:** ingresos ventas / gasto publicitario.

3. **RPC (Result Per Cost):** visitas únicas / gasto publicitario.

V. Módulo de Retomas (1)

4. **Total Solicitudes Retoma Recibidas:** registros tabla `buy_requests`.

VI. Seguridad y Soporte (2)

5. **Tasa de Errores en Pagos:** webhooks `rejected/cancelled`.

6. **Uso de Botones WhatsApp.**

VII. Fidelización y UX Avanzada (5)

7. **Tasa Repetición Compra:** email con >1 pedido aprobado.

8. **Eficacia Oferta del Día:** compras que incluyen ODD + valor de otros productos en mismo carrito (Efecto Halo).

9. **Profundidad de Scroll:** puntos control 25/50/75/100% en fichas.

0. **CTR Preguntas Frecuentes.**

1. **Pedidos por Ubicación Geográfica:** heatmap Colombia.

NUEVO en v18 · Embudo de Conversión 8 niveles

Métrica adicional `trackHomeScroll` (#192-#193). Registra hasta dónde llegó el usuario en el Home (0 a 7):

- 0: entró sin scroll.
- 1: scrolleó poco.
- 2: llegó a Más Vendidos.
- 3: llegó a Compramos tu PC.
- 4: llegó a Diseño Web Pro.
- 5: llegó a Nuestra Tienda.
- 6: llegó a Directorio.
- 7: llegó al footer.

Detecta abandono con `pagehide` event para volcar UN solo registro por visita.

Módulo 6.8.2 · Profit & Commission Manager

Componente: `src/pages/admin/ProfitManager.jsx`. Backend `src/lib/profitManager.js`.

Sistema de auditoría y cálculo de comisiones del Administrador Técnico (Damorus, en este caso).

Visualmente independiente del 6.8.1: separación clara.

Lógica Financiera

• **Fórmula:** `Comisión = Total_Venta_Cliente × %_Comisión_Vigente`.

• **Fundamento:** comisión sobre total bruto (los costos de pasarela MP son responsabilidad operativa de la empresa, no se descuentan).

- **Persistencia Histórica (Snapshots):** valor de comisión guardado en momento exacto de transacción. Si % cambia, registros antiguos mantienen su valor.
- **Sugerencia de Lógica:** sistema guarda qué plan MP estaba activo al momento de cada venta. Cambio mid-month → ventas días 1-14 con plan viejo, días 15-30 con nuevo.

Interfaz UX/UI

- **Centro de Control (Cards):** Saldo Pendiente por Cobrar / Próxima Fecha Corte / Total Histórico Pagado.
- **Módulo Gráfico:** barras filtrables por Diario/Semanal/Bisemanal/Mensual/Semestral/Anual.
- **Gestión Cortes y Pagos:**
 - Selector Fecha Corte (define día cierre pago bisemanal).
 - Tabla Bisemanal Automática (cada 14 días, monto total).
 - Histórico Permanente (estados Pendiente/Pagado, no se sobrescribe).

Herramientas

- Ajuste Dinámico de Comisión (porcentaje configurable).
- Configurador Pasarela (editable si MP cambia tarifas).
- Acceso e Integridad: datos en Supabase, visibles para admin y dueños sin discrepancias.

Módulo 6.8.3 · Report Builder (PDF + CSV)

Componente: `src/pages/admin/ReportBuilder.jsx`. Backend `src/lib/reportBuilder.js`.

URG Report Builder genera informes ejecutivos con:

- **Selector Rango:** calendario dinámico (Hoy / Última Semana / Mes Actual / Trimestre / Rango Personalizado).
- **Checklist de Métricas:** admin marca/desmarca cuáles incluir.
- **Formatos:**
 - PDF Ejecutivo (diseño limpio con gráficos + tablas).
 - CSV/Excel para contabilidad avanzada.
- **Sincronización:** informe recuerda qué plan MP estaba activo para que cálculos sean exactos.

Función `exportarBI()` en `lib/biMetrics.js` (corregida en v18 bug 5). Trimestre soportado en mismo update.

6.9 Centro de Inteligencia y Vigilancia (Black Box D1)

Componente: `src/pages/admin/BlackBox.jsx`. Backend `src/lib/blackBox.js`.

Núcleo de monitoreo táctico. Detecta, rastrea y neutraliza amenazas antes de que afecten inventario o datos.

1. Dashboard de Situación

- **Mapa de Calor Global (Geo-Threat Map):** mapa mundial tonos oscuros con pulsos Morado URG (éxitos) y Rojo Neón (fallos).
- **Monitor Carga de Autenticación:** gráfico líneas tiempo real de peticiones a `/[panel]`. Picos inusuales → alerta DDoS/Fuerza Bruta.
- **Contador Bloqueos Activos:** indicador digital de IPs en Lista Negra + repelidas por Turnstile en 24h.

2. Log de Auditoría Forense

Tabla terminal monospaciada con ADN de cada interacción de login:

- **Huella Digital (Fingerprint):** captura ID único dispositivo. Si hacker cambia IP pero usa misma PC, lo reconoce.
- **Análisis de Intención (Status Tag):**
 - `MFA_CHALLENGE_FAIL` : alerta crítica (conoce pass, no tiene dispositivo TOTP).
 - `SUSPICIOUS_BOT_PATTERN` : bloqueo preventivo comportamiento no humano.
 - `SUCCESSFUL_ADMIN_ENTRY` : acceso legítimo.
- **Rastreo de Sesión:** duración + acciones realizadas (vía RLS Supabase).

3. Módulo de Respuesta Activa

3.1 Gestión Lista Negra (Firewall Administrativo)

- **Visualizador IPs Bloqueadas:** IP + Fecha Bloqueo + Origen (Automático/Manual) + Razón.
- **Acción Desbloqueo (Whitelist):** elimina IP de lista negra en tiempo real.
- **Baneo Manual Proactivo:** input para bloquear IP sin esperar detección.
- **Cierre de Sesiones Global (Panic Button):** cierra todas las sesiones activas instantáneamente.
- **Aislamiento de Cuenta:** bloquea temp un admin hasta verificación manual.

3.2 Radar de Sospecha (Pre-Threat)

- **IPs en Observación** (amarillo ámbar): >3 fallos no llegan al baneo automático.
- **SOP_LEVEL:** LOW (errática) / MEDIUM (rutas config menores) / HIGH (múltiples fallos + ubicación inusual).

4. Inteligencia de Comportamiento (Heurística Local)

Protocolo Robustez Credenciales:

- Mínimo 9 caracteres.
- Al menos 1 número + 1 carácter especial.
- Indicador neón Rojo → Verde según complejidad.

Escalamiento Bloqueo (Rate Limiting):

- Nivel 1: 5 fallos / 2 min = Baneo 5 min.
- Nivel 2: 5 fallos adicionales = Baneo 15 min.
- Nivel 3: 2 fallos extra = Bloqueo Permanente hasta reset manual Super Admin.

5. Sistema de Trampas Honeytrap

Rutas señuelo monitoreadas (su lista no se publica).

Acción Automática:

- Etiquetado IP como "Agresor Potencial".
- Baneo preventivo 24h al interactuar con >2 señuelos.
- Captura Timestamp + IP + país + User-Agent.

6. Protocolo Notificaciones Críticas (Integración Resend)

Black Box dispara Trigger E.3 cuando:

1. `MFA_CHALLENGE_FAIL` : >2 fallos código en <5 min.
2. `GEOGRAPHIC_IMPOSSIBILITY` : acceso exitoso >500km del anterior en <1h.
3. `BRUTE_FORCE_LOCKOUT` : bloqueo automático por exceder intentos.

Sincronización Datos: envía `incident_id` + `interaction_score` a la API Resend para correo con informe forense completo.

Webhook para que botón [BANEAR IP] del correo ejecute el Baneo Manual sin entrar al panel.

6 · BLOQUE IV · CONFIGURACIÓN Y ESTRUCTURA

6.10 Configuración Tributaria, Financiera y Logística

Componente: `src/pages/admin/Tributario.jsx`. Backend `src/lib/tributarioConfig.js`.

Variables Maestras del sistema. Cualquier cambio aquí se propaga via Variable Global a Creador, Checkout, Resend, BI.

Bloque 1 · Calculadora Global de Impuestos (Motor IVA/UVT)

- **Valor UVT Actual:** \$52.374 (editable). 2026 oficial.
- **Umbral Exclusión (UVT):** 50 (editable, default).
- **Cálculo Automático (Read-Only):** \$2.618.700 (tope en pesos).
- **Tarifa General IVA:** 19% (editable).

Centro de Ayuda Tributaria (Botón [i]): micro-manual explica qué es UVT, tope exclusión, regla del precio final.

Impacto Sistémico: cambio UVT \$52k → \$55k → umbral sube automáticamente → todos productos reclasificados Sin IVA / Con IVA instantáneamente.

Bloque 2 · Gestión Logística (Matriz de Envíos)

Tabla tarifas envío:

1. **Recogida Local** \$0 (no editable).
2. **Bogotá D.C.** \$0 (default, editable).
3. **Nacional** (tarifa configurable).

Botón [+ Agregar Ciudad/Zona Especial] para excepciones (ej: una tarifa especial para Medellín). Si cliente elige Medellín, prioriza esta tarifa sobre Nacional.

Interruptor Cobro Global Bogotá (toggle ON/OFF + campo modificable de costo).

Bloque 3 · Calculadora Rentabilidad (Mercado Pago)

NO afecta precio cliente: solo BI interno.

Configuración Tarifas Pasarela:

- Porcentaje Comisión (editable).
- Costo Fijo Transacción (editable).

Selector Plan de Retiro:

- ☐ Al Instante: 3.29% + \$800 + IVA comisión.
- ☐ 7 Días: 2.99% + \$800 + IVA.
- ☐ 14 Días: 2.79% + \$800 + IVA (más rentable).

Lógica BI: % seleccionado se resta automáticamente del Valor Total en informes para Utilidad Neta Real.

Bloque 4 · Marketing Hub (Cupones)

Sistema códigos de descuento:

Creador Promociones:

- Código (ej: [código del cupón]).
- Tipo (Porcentaje % vs Valor Fijo \$).
- Límite Usos.
- Vencimiento (calendario).
- Regla Oro: No Acumutable.

Visor Cupones Activos: Código | Progreso uso (15/100) | Estado | Editar/Eliminar.

Bloque 5 · Lógica IVA & UVT (Casos)

A. SIN IVA (Excluido)

Precio $\leq$ 50 UVT $\rightarrow$ IVA = 0.

B. CON IVA (Gravado)

Precio $>$ 50 UVT.

- Subtotal = Precio Total / 1.19
- IVA = Precio Total - Subtotal

Ejemplo PC \$3.000.000:

- Subtotal \$2.521.008
- IVA \$478.992

C. EXENTO (Casos Especiales)

Casos especiales del Art. 424 numeral 6 (convenios internacionales, equipos para personas con discapacidad). Marca como "Exento" (no "Excluido") en factura electrónica.

Caso Especial Explicativo

PC \$3.333.333 con 10% descuento (\$3.000.000) + 5% cupón + envío nacional (tarifa configurable):

```
Producto: MacBook Pro (con 10% OFF aplicado)
Subtotal:      $ 2.521.008
IVA (19%):     $ 478.992
Envío:         (tarifa de envío)
Cupón de Regalo: -$ 150.000
TOTAL A PAGAR: $ 2.850.000 + envío
```

6.11 Panel Maestro (Ajustes Técnicos y Seguridad)

Componente: `src/pages/admin/PanelMaestro.jsx`. Backend `src/lib/siteConfig.js`, `src/lib/contactConfig.js`, `src/lib/bannerHero.js`.

1. Gestión de Variables de Entorno (API Keys & DNS)

Terminal de Conectividad. Cada bloque tiene texto explicativo + botón CTA "Guardar y actualizar tienda".

A. Centro de Conexión MercadoPago

- **Access Token** (producción): campo password.
- **Webhook URL** generada por Supabase para recibir confirmaciones. Debe pegarse en Panel Desarrolladores MP.

- Edge Functions involucradas: `mp-update-token` y `mp-update-public-key` para actualizar credenciales sin redeployar el bundle.
- Edge Function `mp-validate-token` para validar que el token es funcional antes de guardar.
- Edge Function `mp-diagnostics` que ejecuta secuencia de pruebas y reporta estado.

B. Protocolo BRE-B

- **Llave de Verificación.**
- **Gestor de QR:** selector archivos (imagen/PDF del código QR de la cuenta). Reflejado automáticamente en proceso de compra cliente.

C. Motor de Mensajería (Resend)

- **API Key Resend:** campo alta seguridad. Pegado en BD Supabase, gestionado desde panel.
- Edge Function `resend-update-token` para guardar/actualizar.
- Edge Function `resend-validate-token` para verificar conexión.
- Edge Function `resend-send-test` para envío de prueba.
- **DNS Records (DKIM, SPF, DMARC):** tabla con Type / Name / Content / TTL / Prio.
 - SPF mx: Recepción.
 - SPF txt: Autorización + seguridad (sin esto → SPAM).
 - DKIM: firma digital, garantiza no alteración.
 - DMARC: instrucción a Gmail/Outlook sobre qué hacer si falla.

D. Identidad del Servidor (Nameservers)

- Primary NS: `[censurado]`
- Secondary NS: `[censurado]`
- Micro-manual sobre qué son y cuándo modificar.

E. Configuración Retorno y Seguridad (Site URL)

- **URL Oficial Sitio:** `https://urgtechnology.com` (o `https://damorus.space` para staging actual).
- **Rutas Redirección:** `https://urgtechnology.com/[panel]`.
- Guía: garantizan flujo de seguridad circular.

F. Protocolo de Ignición (Primer Encendido)

4 pasos para desarrollador:

1. Apertura Túnel (Management API): Access Token Supabase.
2. Sincronización Brújula (Site URL Inicial) en Auth > URL Configuration.
3. Vinculación Cartero (SMTP/Resend) en Auth Settings > SMTP.
4. Verificación Redirección Automática: `/auth/v1/callback` en URLs permitidas.

2. Administrador de Usuarios y Jerarquía

Backend `src/lib/adminUsers.js`. Sistema gestión perfiles basado en Bcrypt de Supabase Auth (Zero-knowledge).

Niveles Predefinidos

1. **Usuario maestro** · Maestro Oculto (clave censurada). Perfil invisible para otros. Control total.
2. **Usuario principal de la empresa** · Super Administrador (clave censurada). Control total excepto Diseñador.

3. **Usuario invitado** · Solo Lectura (clave censurada). Acceso visual Bloques I y III. Botones edit en gris.

Creador de Perfiles

- Nombre de Usuario (público).
- Correo Electrónico (vinculado a Supabase Auth).
- Contraseña Inicial (URG Standard: min 9 chars + 1 número + 1 especial; barra Roja→Verde).
- Selector Rango: Super (Dorado), Editor/Logística (Morado), Invitado (Gris).

Matriz Permisos Granulares

Checkboxes V (Ver) y M (Modificar) por sección:

- **Bloque I (Operación Diaria):** Login / Centro Control / Creador Fichas / Flujo Operativo.
- **Bloque II (Cliente):** Comunidad / Mensajería / FAQ Hub.
- **Bloque III (Inteligencia):** BI / Black Box.
- **Bloque IV (Configuración):** Tributario / Panel Maestro.

Directorio de Mandos (Listado)

Tabla estética Terminal Seguridad:

- **Filtro Invisibilidad:** el usuario maestro nunca aparece (protección).
- Avatar (Glow color) + Usuario + Rango + Estatus + Acciones (Editar / Reset Clave / Eliminar).

Lógica Flujo

1. **Autonomía Contraseñas:** admin define temp pero Supabase Auth fuerza al usuario a ser único dueño real (Cero Conocimiento).
2. **Sincronización Interfaz:** si se quita permiso [V] mientras user logueado, pestaña desaparece en siguiente refresh.
3. **Protección Auto-Borrado:** Super Admin no puede eliminarse a sí mismo.

3. Master de Contacto Global y Footer

Submódulo que centraliza identidad de contacto. Variable Global propagada a Web, Emails, BD.

A. Identidad y Operación

- **Tagline editable** ("Empresa colombiana líder en tecnología reacondicionada...").
- **Horario de Atención** (textarea multilínea).
- **Interruptor Diseño Web Pro** (Toggle ON/OFF: controla la visibilidad del banner promo).
- **Email Corporativo:** [correo de la tienda] (actualiza mailto: en footer y emails).

B. Soporte Global (Botón Maestro)

- **Campo Único WhatsApp:** [WhatsApp de la tienda] (actualiza TODOS los CTAs marcados como 'Global Support CTA').
- **Sanitización Automática:** versión visual (con espacios/signos) + versión acción (sin nada, para link [https://wa.me/\[WhatsApp de la tienda\]](https://wa.me/[WhatsApp de la tienda])).

C. Líneas de Especialistas (Footer)

Estructura filas editables:

- Área (Hardware/Software/Diseño Web).

- Responsable (nombre, opcional).
- Contacto (email o teléfono).

Comportamiento Footer: agregar fila aquí genera entrada en Columna 3 automáticamente.

D. Directorio y Ubicación Geográfica (Dual-Map Control)

D.1 Control Visual (Iframe): área de texto amplia para código `<iframe ...>`. Modificable sin tocar HTML.

D.2 Control de Acción (Hipervínculo): Input URL `[enlace de Google Maps]`. Disparador apertura Google Maps/Waze/Apple Maps en el celular.

4. Módulo IA Chat (Diseño Web Pro)

Componente: `src/lib/aiChat.js`. Tras v24.18.26 este módulo cambió de propósito:

A. Interruptor Maestro

Toggle ON/OFF de la sección Diseño Web Pro. Apagar oculta automáticamente:

1. Mención en menú hamburguesa.
2. Banner en Home.
3. Mención en Footer.

Regla: re-calcular espacios para no dejar saltos.

B. Centro de Control IA (Live Code Editor) · DEPRECADO en URG

Esta funcionalidad sigue presente en código pero **deshabilitada** porque tras la migración v24.18.26 toda la lógica IA/Typebot vive en `damorus.com`. El editor se mantiene como infraestructura por si en futuro se quiere reactivar.

C. Respaldo y Recuperación (Botón Fénix)

Botón "Restablecer Valor Original" rojo neón. Modal confirmación. Variable de Entorno Protegida en Supabase contiene Script Maestro.

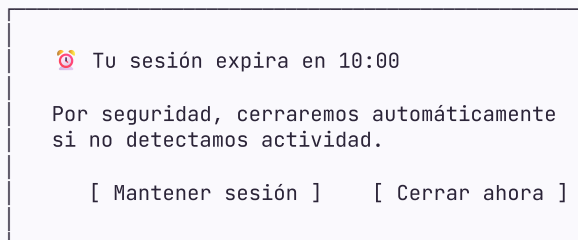
6.12 Perfil y Sesión

Componente: `src/pages/admin/Perfil.jsx`. Accesible a cualquier admin autenticado (cada uno ve el suyo). Permite:

- Editar Nombre, Foto, Email válido.
- Cambio de Contraseña con doble validación (actual + nueva + confirmar).
- Cierre Sesiones Remotas (Checkbox).
- Activar/Desactivar MFA TOTP con QR.

Modal de Sesión por Inactividad (v6 update)

A los 50 minutos de inactividad aparece modal Warning 10 minutos antes del cierre forzado:



Cualquier interacción (click, scroll, navegación) resetea el cronómetro. Si vence, redirige a `/[panel]` con mensaje "Sesión cerrada por inactividad para proteger tus datos".

Componente del cronómetro: `src/hooks/useInactivityTimer.jsx`.

Continúa en la **Parte 6** con la sección 7 (Lógica técnica transversal), backend server-side completo, pipeline CI/CD, cookies y datos legales.

7. LÓGICA TÉCNICA TRANSVERSAL Y PREVENCIÓN DE ERRORES

Funciones automáticas que el sistema ejecuta para garantizar integridad de BD y robustez de comunicaciones. Esta sección consolida los 25 puntos del V1.1 con los ajustes prácticos descubiertos en producción.

7.1 Sanitizador de Telecomunicaciones (`href="tel:"`)

Implementado en `src/lib/format.js` → `sanitizePhone(input)`:

```
export function sanitizePhone(input) {
  if (!input) return '';
  return String(input).replace(/[\d+]/g, '');
}
```

- **Input admin:** "[WhatsApp de la tienda]" o "[WhatsApp de la tienda]" .
- **Output técnico:** [WhatsApp de la tienda] → `<a href="tel:[WhatsApp de la tienda]">` .
- **Output visual:** formato estético original mostrado al usuario.

7.2 Codificador URL Dinámico (WhatsApp)

Función `buildWhatsappLink(numero, mensaje)`:

```
export function buildWhatsappLink(numero, mensaje) {
  const tel = sanitizePhone(numero);
  const msg = encodeURIComponent(String(mensaje ?? ''));
  return `https://wa.me/${tel}?text=${msg}`;
}
```

- Aplica `encodeURIComponent()` automáticamente sobre el mensaje (tildes, espacios, signos).
- Resultado: `https://wa.me/[WhatsApp de la tienda]?text=Hola%2C%20necesito%20asesor%C3%ADa%20t%C3%A9cnica` .

7.3 Protocolo de Seguridad de Sesión e Inactividad

Hook `useInactivityTimer()`:

- Cronómetro 60 min reinicia con cada acción (click, scroll, navegación, keypress).
- Al cumplir 60 min:
 1. Elimina token de `localStorage` / `sessionStorage` .
 2. Redirige a `/[panel]` (login).
 3. Muestra toast: *"Sesión cerrada por inactividad para proteger tus datos"*.
- **Modal Warning 10 min antes** (task #131): permite al admin extender sesión sin perder contexto.
- **Middleware:** intentos de volver atrás post-logout son rechazados, fuerza login.

7.4 Modales de Confirmación de Acción

Componente: `src/components/admin/ConfirmModal.jsx` . Toda acción Guardar/Editar/Eliminar pasa por modal:

"¿Estás seguro de que deseas realizar estos cambios?"

[SÍ · Verde Neón] [NO · Rojo Neón]

- **SÍ** ejecuta acción + muestra toast verde.
- **NO** cierra modal sin cambios.

7.5 Seguridad de Rutas y Middleware (Anti-ByPass)

A. Validación de Token y Rol (Middleware)

Componente `<RequireBloque bloque="I">`. Antes de renderizar:

- ¿Sesión activa? Sino → redirect `/[panel]`.
- ¿Rol del usuario tiene permiso para esta ruta?
- Si NO → aborta carga inmediatamente.

B. Acción de Expulsión Automática

- Bloqueo renderizado (pantalla blanco momentáneo).
- Redirección forzada a `/[panel]/dashboard` (página de inicio permitida).
- Toast flotante: *"Acceso Denegado: No tienes permisos suficientes para ver esta sección."*

C. Ofuscación de Rutas Críticas

Rutas críticas con nombres neutros (no se publican) que el sistema mapea internamente: dificulta adivinarlas.

7.6 Protocolo de Integridad de Datos (Soft Delete)

7.6.1 Sustitución del DELETE

Prohibido borrado físico en `productos` desde la interfaz principal. Acciones Eliminar/Finalizar ejecutan `UPDATE status='archived' / hidden=true`.

7.6.2 Diferenciación de Origen

- **Origen A (Manual):** producto a papelera por admin.
- **Origen B (Automático):** stock = 0 tras venta confirmada.

7.6.3 Filtrado Inteligente

- **Front-End (Tienda):** queries incluyen `WHERE hidden = false AND status = 'active'`.
- **Back-End (BI):** cálculos ignoran estado oculto, procesan todos los registros (balances anuales exactos).

7.7 Optimización de Medios (Procesamiento + Pre-visualización)

Módulo `src/lib/imageOptimizer.js`.

A. Capa 1 · Procesamiento Automático

- Conversión obligatoria a **WebP**.

- **Redimensión Max-Pro:** lado largo ≤ 1920 px (Full HD).
- **Compresión Fondo (Lossless):** elimina metadatos EXIF (privacidad + tamaño).

B. Capa 2 · Interfaz Ajuste Manual

Consola de Comparación Visual (pantalla dividida Original vs Procesada). Sliders:

- **Resolución** (reducir <1920 px).
- **Calidad (0-100)** WebP, default 85%.

Monitor Peso Tiempo Real:

- Peso Original (12.5 MB).
- Peso Optimizado (340 KB).
- Indicador % ahorro ("Reducción 92%").

C. Lógica de Almacenamiento y Entrega

- **Carga Perezosa Pro:** Lazy Loading + Skeleton Screen con color dominante + Fade-in 0.3 s.

D. Seguridad de Edición

- Botón Reset a valores Capa 1.
- Pre-visualización Zoom 100% en área específica para verificar legibilidad post-compresión.

7.8 Lógica de Auditoría · Carrito Abandonado

Tabla `logs_pedidos` en Supabase. 4 estados:

1. **Iniciado** (Carrito): cliente llenó datos + clic Continuar.
2. **En Pasarela** (Pendiente): saltó a MP o abrió BRE-B.
3. **Fallido / Expirado:** 8 min sin confirmación.
4. **Completado:** pago verificado.

Embudo Conversión BI:

Sesiones pago iniciadas mes:	100
Pagos completados:	60
Tasa de Abandono:	40%

Diferencia entre MP iniciadas vs BRE-B iniciadas para comparar efectividad.

7.9 Página 404 · Dark Terminal

Componente: `src/pages/NotFound.jsx`.

Mensajería Universal:

- Título: `LINK NO DISPONIBLE`
- Subtítulo: *"La dirección que has solicitado no existe o ha sido movida permanentemente. Verifica la URL o utiliza los accesos directos a continuación para continuar."*

UI Alta Gama:

- Logo URG opacidad 50% centrado.
- `[STATUS CODE: 404]` morado pequeño arriba.

- Botones controles rectangulares finos:
 - [IR AL INICIO] (fondo negro, borde morado).
 - [EXPLORAR CATÁLOGO] (solo texto subrayado hover).

7.10 Performance • Indexación BD (Supabase)

Latencia objetivo: <50 ms independientemente del volumen.

```
-- Índice Búsqueda Texto (GIN)
CREATE INDEX idx_productos_search ON productos
  USING gin (to_tsvector('spanish', nombre || ' ' || descripcion));

-- Índices Árbol (B-Tree)
CREATE INDEX idx_productos_marca ON productos (marca);
CREATE INDEX idx_productos_categoria ON productos (categoria);
CREATE INDEX idx_productos_activo ON productos (hidden) WHERE hidden = false;
```

Evita Full Table Scan. Respuesta servidor instantánea incluso con 1.000+ equipos.

7.11 Validación de Sintaxis en Tiempo Real

Módulo `src/lib/validators.js`. Email vía RegEx:

```
^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$
```

Si inválido → borde rojo neón + botón submit deshabilitado. Error desaparece al corregir.

7.12 Micro-interacciones (Toast)

Componente: `src/components/admin/ToastProvider.jsx`.

- **Tienda Cliente:** "Añadir al Carrito" → toast esquina inferior central [✓ Producto añadido con éxito].
- **Admin:** tras Crear/Editar → [✓ Cambios guardados] o [✓ Categoría creada].
- **Comportamiento:** fade-in, desaparece auto en 3 s, no intrusivo.

7.13 Breadcrumb Dinámico

Componente: `src/components/Breadcrumb.jsx`. Texto gris claro, tamaño pequeño, separado por >.

Lógica: Inicio > Catálogo > [Categoría o Marca]. Cada nivel es link que permite retroceder sin botón "Atrás" navegador.

7.14 Protección Entrada de Datos (Anti-XSS)

1. Validación Nativa (Front-End)

HTML5 obligatoria (`required`, `type="email"`, `pattern`). Sistema no permite envío si no cumple formato.

2. Sanitización (Anti-Hacking)

Función `sanitizeXSS()` en `lib/validators.js` elimina/codifica <, >, /, `script`, `onerror`. Bug v20: la sanitización rompía base64 dataURL, fix en task #207 con whitelist explícita.

3. Control de Concurrencia (Anti-Duplicados)

- Al presionar Enviar → `button.disabled = true` + estado "Cargando...".
- Evita registros duplicados por clientes ansiosos o conexiones lentas.

4. Respuesta del Sistema

Tras envío exitoso, formulario se limpia + mensaje confirmación claro.

7.15 Jerarquía Resiliencia (Kill-Switch)

1. Scripts Críticos

- Cálculo precios/IVA/descuentos.
- Validación stock BD.
- Generación ID pedido + firma pasarela.
- Carga políticas privacidad/términos.

2. Protocolo Kill-Switch

Si función crítica falla → deshabilitar todos los botones compra + acceso pago.

Banner: *"Estamos experimentando una intermitencia técnica en el módulo de pagos. Para tu seguridad, las transacciones han sido pausadas. Por favor, refresca la página o contacta a un asesor".*

3. Resiliencia No Críticos

Si fallan carrusel/modo oscuro/efectos visuales → Plan B permite flujo de compra. Error registrado en consola admin para revisión.

4. Verificación Integridad Back-end

Función RPC valida precio final basándose en datos maestros del servidor. Si precio enviado por usuario ≠ cálculo servidor → transacción abortada con 'Fallo de Integridad de Datos'.

7.16 Optimización Consultas Catálogo

1. Índices Parciales PostgreSQL

```
CREATE INDEX idx_productos_disponibles ON productos (id)
WHERE (hidden = false AND stock > 0);
```

Vía rápida solo de productos activos. Saltos directos a relevantes ignorando Cementerio.

2. Separación Carga

- **Web Pública:** queries obligatorias `status = 'disponible'`.
- **Panel Admin BI:** índices diferentes optimizados por fechas.

3. Consumo Eficiente

Solo solicita campos necesarios para vista previa (nombre, precio, foto principal). Datos pesados (fichas completas, galerías) solo en carga individual.

7.17 Configuración Servidor (.htaccess)

Archivo completo en `dist/.htaccess` :

```
# — Protección: prohibir listado de directorios —
Options -Indexes
ServerSignature Off

# — Forzar HTTPS —
<IfModule mod_rewrite.c>
    RewriteEngine On

    # http → https
    RewriteCond %{HTTPS} !=on [NC]
    RewriteCond %{HTTP:X-Forwarded-Proto} !=https [NC]
    RewriteRule ^(.*)$ https://%{HTTP_HOST}/$1 [R=301,L]

    # Quitar www (canonical sin www para SEO)
    RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]
    RewriteRule ^(.*)$ https://%1/$1 [L,R=301]

    # SPA fallback · si archivo/carpeta no existe, sirve index.html
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteCond %{REQUEST_URI} !^/api/
    RewriteRule ^ index.html [L]
</IfModule>

# — Cache-control + cache-busting —
<IfModule mod_expires.c>
    ExpiresActive On
    ExpiresByType application/javascript "access plus 1 year"
    ExpiresByType text/css "access plus 1 year"
    ExpiresByType image/svg+xml "access plus 1 year"
    ExpiresByType image/png "access plus 1 year"
    ExpiresByType image/jpeg "access plus 1 year"
    ExpiresByType image/webp "access plus 1 year"
    ExpiresByType font/woff2 "access plus 1 year"
    ExpiresByType image/x-icon "access plus 1 month"
    ExpiresByType text/html "access plus 0 seconds"
</IfModule>

<IfModule mod_headers.c>
    <FilesMatch "index\.$">
        Header set Cache-Control "no-store, no-cache, must-revalidate, max-age=0"
        Header set Pragma "no-cache"
    </FilesMatch>

    <FilesMatch "\.(js|css|woff2|svg|png|jpg|jpeg|webp|ico)$">
        Header set Cache-Control "public, max-age=31536000, immutable"
    </FilesMatch>

    # Cabeceras de seguridad
    Header set X-Content-Type-Options "nosniff"
    Header set X-Frame-Options "SAMEORIGIN"
    Header set Referrer-Policy "strict-origin-when-cross-origin"
    Header set Permissions-Policy "geolocation=(), microphone=(), camera=()"

    # Content-Security-Policy
    Header set Content-Security-Policy "default-src 'self'; \
    script-src 'self' 'unsafe-inline' https://sdk.mercadopago.com https://challenges.cloudflare.com https://www.google.com; \
    style-src 'self' 'unsafe-inline' https://fonts.googleapis.com; \
    img-src 'self' data: blob: https; \
    font-src 'self' https://fonts.gstatic.com; \
    frame-src 'self' https://www.google.com https://challenges.cloudflare.com; \
    connect-src 'self' https://*.supabase.co https://api.resend.com https://api.mercadopago.com wss://*.supabase.co; \
    object-src 'none'; \
    base-uri 'self';"

    # HSTS (descomentar tras despliegue exitoso)
    # Header set Strict-Transport-Security "max-age=15768000; includeSubDomains"
</IfModule>

# — Compresión gzip —
<IfModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/html text/css text/javascript application/javascript application/json image/svg+xml application/xml
</IfModule>

# — Bloquear archivos sensibles —
<FilesMatch "(^\.env|\.env\..*|\.git|\.gitignore|package(-lock)?\.json|vite\.config\.js|composer\.json|lock)$">
    Require all denied
</FilesMatch>

# — UTF-8 universal —
AddDefaultCharset UTF-8
AddCharset UTF-8 .html .css .js .xml .json .rss .atom .svg

# — 404 personalizada —
ErrorDocument 404 /index.html
```

Permisos cPanel (Anti-403)

- Directorios: **755** (R/X todos, W solo dueño).
- Archivos: **644** (R todos, W solo dueño).
- `.htaccess` protegido (no listado de directorios).

7.18 Verificación SDKs Externos (Anti-AdBlock)

Función `lib/sdkLoader.js`:

```
export async function ensureMercadoPagoLoaded() {
  return new Promise((resolve, reject) => {
    if (window.MercadoPago) return resolve(window.MercadoPago);
    let attempts = 0;
    const id = setInterval(() => {
      if (window.MercadoPago) { clearInterval(id); resolve(window.MercadoPago); }
      else if (++attempts > 50) {
        clearInterval(id);
        reject(new Error('AdBlock_or_Firewall_Blocking_MercadoPago'));
      }
    }, 100);
  });
}
```

Aviso si fallido: *"Nota: Tu bloqueador de anuncios podría estar interfiriendo con la pasarela de pago. Si tienes problemas, por favor desactívalo temporalmente para completar tu compra de forma segura".*

7.19 Gestión Timeouts y Estados Carga Asíncrona

Wrapper `lib/asyncTimeout.js`:

- **Timeout 25 s** para subida archivos / registro pago.
- Si supera → aborta + mensaje claro: *"La conexión está tardando más de lo esperado. No hemos podido confirmar tu envío. Por favor, intenta de nuevo o verifica tu conexión a internet".*
- Diferencia entre Error Servidor 500 vs Timeout/Sin internet.
- Botón Reintento Inteligente sin recargar página.
- **Prevención Cierre Pestaña** (`beforeunload`): *"Tu envío aún está en proceso. Si sales ahora, el registro de tu pago podría no completarse".*

7.20 Optimización Recursos y Límites Almacenamiento

1. Entorno PHP/Edge Functions

`memory_limit` ≥ 256M, `max_execution_time` ≥ 60s.

2. Validación "Compramos tu PC"

- Max 3 fotos/solicitud.
- Max 10 MB/archivo entrada.
- Formatos: `.jpg`, `.jpeg`, `.png`.
- Compresión Dual-Layer (cliente Canvas API → ≤1.5MB / servidor sanitización + compresión final).

3. Gestión Errores

Si archivo >10 MB pre-compresión: "El archivo supera el límite de 10MB. Por favor, intenta con una versión más ligera o captura la foto directamente desde el formulario".

Barra progreso real durante compresión/subida (no fake).

7.21 Integridad Despliegue y Lógica Temporal

1. Permisos cPanel · ver §7.17.

2. Validación Temporal Centralizada (Anti-Hackeo)

Prohibido confiar en reloj cliente para conteo regresivo de reservas, expiración carritos, vigencia cupones.

Lógica servidor: comparar contra timestamp servidor o `now()` Supabase.

`new Date()` JS solo para efectos visuales, nunca como base para autorizar transacción.

3. Anti-Abuso

- **Validación Stock Atómico:** doble verificación BD (race condition prevenido por `FOR UPDATE` lock pesimista, ver §5.4).
- **Sanitización Privacidad Archivos:** borra EXIF (geolocalización, modelo cámara).
- **Rate Limiting:** máximo 3 intentos/min en `/compramos-tu-pc` y reserva stock (anti-DoS / anti-bot).

7.22 Protocolo Escucha de Pagos (Webhook)

Detalle completo en §8 (Backend Server-Side). Resumen:

- Edge Function `mp-webhook` recibe notificaciones MP.
- Valida origen (firma secreta + IP allowed).
- Ejecuta `consumir_stock` atómico.
- Devuelve 200 OK inmediatamente para evitar reintentos MP.
- Manejo de Cola: errores se registran en `webhook_errors` para re-sync manual.

7.23 Notificación Estado Red (Resiliencia UX Pasiva)

Hook `useNetworkStatus.js`:

1. **Feedback Evolutivo 8s:** si BD >10s tarda → "Sincronizando inventario en tiempo real...".
2. **Canal Rescate Inmediato:** WhatsApp funcional desde segundo 1.
3. **Carga No Invasiva:** prohibido bloquear pantalla con overlays. Usuario puede scrollear/navegar mientras carga.
4. **Suavidad Visual:** cargador activo hasta ≥50% elementos listos.
5. **Fallback 45-60s:** aviso "Alta demanda de tráfico" + redirección a WhatsApp.

7.24 Sincronización Historial SPA (Native-Feel)

1. **PushState:** abrir Catálogo/Ficha/Carrito → `history.pushState({section: 'nombre'}, '', '/nombre')`.

2. **Captura popstate:** botón "Atrás" del navegador cierra overlay en lugar de abandonar sitio.
3. **Navegación Circular Preventiva:** cerrar con X → `history.back()` para limpiar.

7.25 Eficiencia Conexiones (Anti-Error 508)

Para proteger Entry Processes del hosting compartido HostGator:

1. **Consolidación Carga (Batching):** múltiples requests relacionados → un solo POST JSON.
2. **Persistencia Local antes Sincronización:** datos rastreo BI → localStorage, sync solo en momentos clave (abrir carrito / iniciar checkout / cerrar pestaña).
3. **Control Reintentos con Delay (Exponential Backoff):** 2s, 4s, 8s antes desistir.

8. BACKEND SERVER-SIDE (NUEVO en V2.0)

Esta sección documenta el backend completo. Es la mayor novedad arquitectónica desde V1.1 y representa el giro de "cliente-céntrico" a "server-céntrico" tras v24.18.25.

8.1 Edge Functions (Deno runtime, Supabase Cloud)

13 funciones activas en `supabase/functions/`. Cada una se despliega individualmente con `supabase functions deploy <nombre>`. Todas comparten un schema de respuesta JSON:

```
{ ok: boolean, data?: any, error?: string, motivo?: string }
```

8.1.1 mp-webhook (CRÍTICA)

Path: `supabase/functions/mp-webhook/index.ts` (~600 líneas).

Responsabilidad: recibir notificaciones de Mercado Pago y aplicar consumo atómico de stock + cancelación de huérfanos.

Flujo:

1. Recibe POST con body `{ type: 'payment', data: { id: '123456789' } }`.
2. Valida origen con secret MP (`MP_WEBHOOK_SECRET`).
3. Consulta API MP `/v1/payments/{id}` con access token para obtener estado.
4. Si `status === 'approved'`:
 - Ejecuta `aplicarConsumoStockYHuerfanos(order_id, producto_id)`.
 - Función interna llama RPC `consumir_stock` + RPC `cancelarHuerfanos`.
 - Marca orden `EFFECTUADA`.
 - Dispara email B.2 (Confirmación Pago) al cliente + B.3 (¡Venta Exitosa!) al admin.
 - Llama `reportarCentinela(evento)` para registro auditoría.
5. Si `status === 'rejected'`:
 - Marca orden `RECHAZADO_MP`.
 - Si había reserva, ejecuta `liberar_reserva`.
6. Responde **siempre 200 OK** para evitar reintentos infinitos de MP.

Funciones internas:

- `aplicarConsumoStockYHuerfanos()`: consume stock + cancela huérfanos en una transacción.
- `cancelarHuerfanos()`: recorre reservas activas, libera las del producto ya consumido, dispara emails de disculpa.
- `liberarReservasOrden()`: cuando MP rechaza, libera todas las reservas de esa orden.
- `reportarCentinela()`: INSERT en `centinela_events` con tipo + descripción + payload.

8.1.2 cleanup-stock-fantasmas (NUEVA en v24.18.25)

Path: `supabase/functions/cleanup-stock-fantasmas/index.ts`.

Responsabilidad: invocada por pg_cron cada 2 minutos. Limpia reservas expiradas.

Flujo:

1. SELECT reservas con `expira_at < now()` agrupadas por producto.

2. Para cada reserva expirada:

- Ejecuta RPC `liberar_reserva`.
- Actualiza orden a estado `FALLIDO_EXP`.
- Si la liberación libera el último stock y hay clientes conectados Realtime viendo el producto → dispara evento broadcast.

3. Limpia órdenes MP en `PASARELA_ABIERTA` >1h.

4. Reporta a Centinela.

8.1.3 admin-auth

Endpoint de validación adicional para login admin. Verifica token Turnstile contra Cloudflare antes de permitir intento.

8.1.4 admin-config-write

Endpoint privilegiado que recibe cambios del Panel Maestro (API keys, DNS, etc.) y los persiste cifrados. Solo callable por Super Admin (verifica JWT role).

8.1.5-8.1.10 mp-* (MercadoPago helpers)

- `mp-create-preference` : crea preferencia de pago al iniciar checkout MP.
- `mp-validate-token` : valida access token vs cuenta MP.
- `mp-update-token` : actualiza access token tras edición admin.
- `mp-update-public-key` : actualiza public key (SDK frontend).
- `mp-diagnostics` : ejecuta batería de pruebas y reporta estado.

8.1.11-8.1.13 resend-*

- `resend-validate-token` : verifica API Key Resend.
- `resend-update-token` : actualiza credencial.
- `resend-send-test` : envía email de prueba.

8.1.14 turnstile-config

Devuelve site key Cloudflare Turnstile al frontend (sin exponer secret).

8.2 Trigger pg_cron

Configuración en `supabase/migrations/20260530_pg_cron_cleanup.sql` :

```

-- Habilitar extensiones
CREATE EXTENSION IF NOT EXISTS pg_cron;
CREATE EXTENSION IF NOT EXISTS pg_net;

-- Job de limpieza cada 2 minutos
SELECT cron.schedule(
  'cleanup-stock-fantasmas-2min',
  '* / 2 * * * *',
  $$
SELECT net.http_post(
  url := 'https://[proyecto-supabase]/functions/v1/cleanup-stock-fantasmas',
  headers := jsonb_build_object(
    'Content-Type', 'application/json',
    'Authorization', 'Bearer ' || current_setting('app.service_role_key')
  ),
  body := '{}'::jsonb
);
$$
);

```

El secret `app.service_role_key` se carga con:

```

ALTER DATABASE postgres SET app.service_role_key = '<SERVICE_ROLE_JWT>';

```

Verificación del cron (query SQL útil para admin):

```

SELECT jobid, jobname, schedule, command, active
FROM cron.job
WHERE jobname = 'cleanup-stock-fantasmas-2min';

SELECT jrd.jobid, j.jobname, jrd.status, jrd.return_message, jrd.start_time
FROM cron.job_run_details jrd
JOIN cron.job j ON j.jobid = jrd.jobid
WHERE j.jobname = 'cleanup-stock-fantasmas-2min'
ORDER BY jrd.start_time DESC
LIMIT 10;

```

8.3 Webhook MercadoPago · Detalle Atómico

Pseudo-código del flujo:

```

// supabase/functions/mp-webhook/index.ts
Deno.serve(async (req) => {
  const body = await req.json();

  // 1. Validar firma
  if (!validarFirmaMercadoPago(req)) {
    return new Response('Forbidden', { status: 403 });
  }

  // 2. Consultar API MP por estado real
  const payment = await consultarMercadoPago(body.data.id);

  // 3. Si approved → consumir
  if (payment.status === 'approved') {
    const order_id = payment.external_reference;

    await supabase.rpc('consumir_stock', {
      p_producto_id: payment.metadata.producto_id,
      p_order_id: order_id
    });

    // Cancelar huérfanos del mismo producto
    await cancelarHuerfanos(supabase, payment.metadata.producto_id, order_id);
  }
});

```

```
// Dispatch emails
await dispararEmailB2(order_id);
await dispararEmailB3(order_id);

// Audit
await reportarCentinela('CONSUMO_OK', { order_id, producto_id });
}

// 4. Siempre 200 OK
return new Response('OK', { status: 200 });
});
```

8.4 Cancelación Cruzada BRE-B ↔ MP

Ya documentada en §5.5. Reforzando el resumen técnico:

- Al consumir stock por una vía, el sistema escanea **todas** las reservas pendientes del mismo producto.
- Si stock final $\leq 0 \rightarrow$ libera todas las reservas huérfanas + dispatcha email de disculpa + emite evento Realtime para clientes conectados.

Función `cancelarHuerfanos` en `mp-webhook` y replicada en endpoint admin de aprobación BRE-B.

9. PIPELINE CI/CD (NUEVO en V2.0)

Documentación completa del pipeline de despliegue automático.

9.1 GitHub Actions Workflow

Archivo: `.github/workflows/deploy.yml`. Trigger: cada `git push origin main` o ejecución manual desde tab Actions.

```
name: Deploy a HostGator vía FTP

on:
  push:
    branches: [main]
  workflow_dispatch:

jobs:
  build-and-deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout código
        uses: actions/checkout@v4

      - name: Setup Node.js 20
        uses: actions/setup-node@v4
        with:
          node-version: '20'
          cache: 'npm'

      - name: Install dependencies
        run: npm ci

      - name: Generar .env de producción
        run: |
          echo "VITE_SUPABASE_URL=${{ secrets.VITE_SUPABASE_URL }}" > .env
          echo "VITE_SUPABASE_ANON_KEY=${{ secrets.VITE_SUPABASE_ANON_KEY }}" >> .env

      - name: Build (vite)
        run: npm run build

      - name: FTP Deploy
        uses: SamKirkland/FTP-Deploy-Action@v4.3.5
        with:
          server: ${{{ secrets.FTP_SERVER }}}
          username: ${{{ secrets.FTP_USERNAME }}}
          password: ${{{ secrets.FTP_PASSWORD }}}
          local-dir: ./dist/
          server-dir: ${{{ secrets.FTP_SERVER_DIR }}}
          protocol: ftp
          port: 21
          dangerous-clean-slate: true
          log-level: standard
          exclude: |
            **/.git*
            **/.git*/**
            **/node_modules/**
            **/.DS_Store
            **/.well-known/**
            **/cgi-bin/**
            **/.ftpquota
```

9.2 FTP Deploy con Clean-Slate Protegido

`dangerous-clean-slate: true` borra TODOS los archivos del servidor antes de subir los nuevos. Garantiza que la carpeta queda 1:1 con el `dist/` actual.

Protección con `exclude` :

- `.well-known/**` : certificados SSL (Let's Encrypt / AutoSSL HostGator). Si se borrara, la próxima renovación fallaría → sitio sin HTTPS.
- `cgi-bin/**` : scripts CGI HostGator (vacío pero existe).
- `.ftpquota` : archivo interno cuota FTP.
- `.git*`, `node_modules/**`, `.DS_Store` : archivos que NO deben subirse (innecesarios en producción).

9.3 Variables de Entorno y Secrets

GitHub Secrets configurados en `Settings → Secrets and variables → Actions` :

Secret	Propósito
<code>VITE_SUPABASE_URL</code>	<code>https://[proyecto-supabase]</code> (proyecto URG).
<code>VITE_SUPABASE_ANON_KEY</code>	Anon key pública del proyecto URG (lectura RLS-controlada).
<code>FTP_SERVER</code>	Servidor FTP HostGator (ej: <code>damorus.space</code> o IP).
<code>FTP_USERNAME</code>	Usuario FTP (<code>[usuario-ftp]</code>).
<code>FTP_PASSWORD</code>	Contraseña FTP.
<code>FTP_SERVER_DIR</code>	Ruta destino (<code>/public_html/</code> para staging actual).

Las `VITE_*` se incrustan en el bundle JS durante `vite build` . Como `.env` está en `.gitignore` , se reconstruyen en cada deploy desde secrets.

9.4 Rollback Manual

Si un deploy rompe el sitio:

1. **Revertir el commit** en GitHub (`git revert HEAD` o desde UI).
2. **Push** → dispara nuevo deploy con la versión anterior.
3. Tiempo aproximado: 2-3 minutos hasta sitio restablecido.

Para emergencias críticas, el admin puede subir archivos manualmente vía cPanel File Manager (acceso a `/public_html/`).

9.5 Hosting HostGator · Configuración

- **Plan:** Shared (compartido).
- **cPanel URL:** `<panel-hostgator-personal>` .
- **Dominio activo:** `damorus.space` (staging URG). Apuntará a `urgtechnology.com.co` en el futuro al adquirir el dominio definitivo.
- **PHP:** 8.1+ configurado en cPanel.
- **SSL:** AutoSSL HostGator (Let's Encrypt) renovación automática mientras `.well-known/` esté intacto.

- **FTP:** cuenta `[usuario-ftp]` con directorio raíz en `public_html` (no en subcarpeta, error inicial corregido durante setup).

X. SISTEMA DE COOKIES (URG)

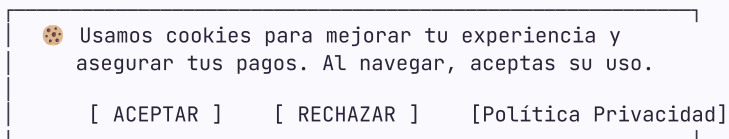
Implementación: `src/components/CookieBanner.jsx`. JavaScript Vanilla, sin frameworks adicionales.

1. Funcionalidad Técnica

- **Persistencia del carrito:** `localStorage['urg.cart']` con TTL 24 h. Si cliente cierra pestaña/recarga, el carrito reaparece.
- API: `cartTracker.js → saveCart(items)` / `loadCart()`.

2. Interfaz Aviso Legal

Banner minimalista parte inferior pantalla:



- Aparece **solo la primera vez**. Tras click [ACEPTAR] o [RECHAZAR], se guarda preferencia en `localStorage['urg.cookieConsent']` y no vuelve.
- Texto + diseño:
 - Tipografía Inter 14 px.
 - Fondo `rgba(11,11,20,0.95)` + `backdrop-filter: blur(8px)`.
 - Borde superior morado `1px solid var(--urg-purple)`.

3. Integración Supabase para Analítica

Función `cartTracker.js → trackEvent(tipo, payload)` envía a tabla `eventos_rastreo`:

- `product_view`: ficha vista.
- `add_to_cart`: botón agregar pulsado.
- `cart_abandoned`: items en localStorage + sin completar pago tras 24h.

4. Recuperación de Ventas (Recompra)

Email capturado en checkout (incluso antes de pagar) → enviado a tabla `leads_checkout`. Si compra no se marca como `EFFECTUADA`, cronjob D.3 dispara email "Carrito Abandonado" al día siguiente con producto pendiente.

5. Segmentación para Anuncios

Datos guardados de forma organizada (Email, Producto interés, Categoría) para exportar a Facebook/Google Ads y hacer retargeting de modelos que el usuario miró.

6. Privacidad y Transparencia

El banner especifica que aceptar = consentimiento al tratamiento de datos de navegación con fines estadísticos y publicitarios, vinculando a Política de Tratamiento de Datos URG.

X. DATOS LEGALES URG TECHNOLOGY

Todos editables desde Panel Maestro § Master de Contacto (§6.11).

- **Razón Social:** URG TECHNOLOGY S.A.S.
- **WhatsApp principal:** [WhatsApp de la tienda]
- **Diseño Web (post-v24.18.26):** Alejandro@damorus.com
- **Responsabilidad Tributaria:** Código 48 (Responsable IVA).
- **Actividad Económica:** Código 4741 (Comercio al por menor de computadores, equipos periféricos, programas de informática y equipos de telecomunicaciones en establecimientos especializados).

APÉNDICE A · Changelog Resumido (v1.0 → v24.18.26)

Las versiones siguen un patrón `vMajor.Minor.Patch`. El histórico completo vive en el sistema de tareas interno (293 tareas ejecutadas al cierre de esta documentación). Los hitos más importantes:

- **v1.0 (abril 2026)**: primer build basado en doc V1.1. Tienda funcional, MP + BRE-B integrados, FAQ Hub, BI básico, Black Box básico.
- **v1.1.x**: endurecimiento BI + mensajería Resend completa + Centro de Mando.
- **v17**: limpieza código muerto (1.080 líneas eliminadas), backup `_backups/` consolidado.
- **v18.x**: refactor BI (TopTerms, LandingPages, embudo scroll 8 niveles, métricas avanzadas).
- **v19**: galería fotos lead con visor lupa, PIN rastrear pedido + auto-purge 5 días.
- **v20**: fix sanitizeXSS no rompe base64, visor lupa optimizada.
- **v21**: pause animations fuera viewport (rendimiento).
- **v22-v23**: Typebot DAMORUS section + Coinbase + Status reactivo (luego migrado).
- **DAMORUS.com (rama paralela)**: proyecto clonig de URG con Landing minimalista + admin filtrado.
- **v18.x (URG continuación)**: refactor Catálogo Activo, Sandbox profesional, Cropper, FAB + Modal Mac Spotlight, SpecTooltip premium.
- **v18.14**: BATCH crítico sesión countdown, stock temporal BRE-B, venta no desaparece.
- **v18.15**: sistema de reservas atómicas server-side (5 etapas).
- **v18.16-v18.18**: fixes form colors, paso 4 validación, cronómetro semáforo draggable.
- **v18.20**: stock auto-fallback + Pendientes cooldown.
- **v24.18.22**: VENDIDO seal sin diagonal + bloqueo compra stock=0 + Avísame.
- **v24.18.23**: MP solo consume al webhook + cancelación cruzada + modal overselling.
- **v24.18.24**: Centinela Salud Operacional.
- **v24.18.25**: Fix arquitectónico server-side (mp-webhook stock + cron BRE-B + huérfanos en server).
- **v24.18.26**: Migrar /diseno-web a damorus.com + GitHub Actions FTP deploy a HostGator.

APÉNDICE B • Estructura de Carpetas **src/**

```
src/
├── App.jsx                # Mapa de rutas, bootstrap
├── main.jsx               # Entry point + providers
├── index.css              # Variables, animaciones, utilidades
├── components/
│   ├── system/           # ErrorBoundary, NeonAccent
│   ├── admin/            # AdminLayout, AdminSidebar, ToastProvider, Confirm, etc.
│   ├── compramos/        # CompramosTuPCForm + Pasos
│   ├── cards/            # Ficha1, Ficha2 reutilizables
│   ├── home/             # Hero, BurbujasCTA, OfertaDelDia, MasVendidos, etc.
│   ├── Header.jsx, Footer.jsx, Layout.jsx
│   ├── PriceDisplay.jsx, PagosStrip.jsx, WhatsappFloat.jsx
│   ├── MasterAscent.jsx, CookieBanner.jsx, MaintenanceMode.jsx
│   ├── SearchAutocomplete.jsx, DeptoCiudadSelector.jsx
│   ├── UrgLab.jsx (legacy diseño web pro, deshabilitado)
│   └── AvisameNotificacion.jsx
├── pages/
│   ├── Home.jsx, Catalogo.jsx, ProductoDetalle.jsx
│   ├── Carrito.jsx, Checkout.jsx, Confirmacion.jsx
│   ├── Privacidad.jsx, CompraProtegida.jsx
│   ├── CompramosTuPC.jsx, RastrearPedido.jsx
│   ├── FAQ.jsx, NotFound.jsx
│   ├── PortatilesEscritorio.jsx, MasVendidosPage.jsx
│   └── admin/
│       ├── Login.jsx, Dashboard.jsx
│       ├── CreadorFichas.jsx, CatalogoActivo.jsx, Pendientes.jsx
│       ├── Vendidos.jsx, Devueltos.jsx, Archivados.jsx
│       ├── CompramosEquipo.jsx, Comunidad.jsx
│       ├── CentroMensajeria.jsx, FAQHub.jsx
│       ├── BI.jsx, ProfitManager.jsx, ReportBuilder.jsx
│       ├── BlackBox.jsx, Centinela.jsx
│       ├── Tributario.jsx, Admins.jsx, PanelMaestro.jsx
│       ├── HomeFooter.jsx, Perfil.jsx, PlaceholderModule.jsx
│       └── modules.jsx    # Re-export consolidado
├── lib/
│   ├── adminProducts.js, orders.js, mercadopago.js
│   ├── buyRequests.js, comunidad.js, messaging.js
│   ├── biMetrics.js, profitManager.js, reportBuilder.js
│   ├── blackBox.js, mfa.js, rateLimiter.js, turnstile.js
│   ├── tributarioConfig.js, contactConfig.js, siteConfig.js
│   ├── bannerHero.js, aiChat.js, seoConfig.js
│   ├── faqHub.js, categoriasPerifericos.js
│   ├── adminSettings.js, adminAudit.js, adminUsers.js
│   ├── adminSearchIndex.js, searchHistory.js, smartSearch.js
│   ├── format.js, validators.js, sanitizeXSS, asyncTimeout.js
│   ├── imageOptimizer.js, fileForensics.js, sdkLoader.js
│   ├── cartTracker.js, avisame.js, pdfBuilder.js, resend.js
│   ├── globalStateSync.js, supabaseBridge.js
│   ├── profitManager.js
│   └── admin.jsx          # AdminProvider context
├── hooks/
│   ├── useInactivityTimer.jsx
│   └── useNetworkStatus.js
├── data/
│   ├── coloDeptos.js     # Catálogo oficial Colombia
│   └── products.js        # Productos demo / fallback
```

APÉNDICE C · Diccionario Resumido de Tablas Supabase

Tabla	Propósito	RLS
<code>productos</code>	catálogo maestro + stock + reservas (jsonb)	SELECT público, mutate solo admin
<code>orders</code>	órdenes de pedido (estado, totales, cliente, producto, lote)	INSERT público, mutate solo admin
<code>orden_pins</code>	PINs rastrear-pedido (4 dígitos, auto-purge 5d)	SELECT con join cliente
<code>comunidad_clientes</code>	clientes recurrentes (cedula, historial)	solo admin
<code>buy_requests</code>	leads de Compramos tu PC	solo admin
<code>plantillas_correo</code>	templates Resend editables	solo admin
<code>faq_items</code> + <code>faq_categories</code>	FAQ Hub CRUD	SELECT público
<code>bi_searches</code> , <code>bi_clicks</code> , <code>bi_sessions</code>	métricas BI	INSERT público, SELECT admin
<code>profit_snapshots</code>	snapshots de comisión por venta	solo admin
<code>centinela_events</code>	log de eventos críticos	solo admin
<code>webhook_errors</code>	reintentos pendientes	solo admin
<code>app_credentials</code>	API keys cifradas (MP, Resend, Turnstile)	solo el usuario maestro
<code>admin_users</code> (vista)	admins registrados (Supabase Auth)	solo Super Admin
<code>audit_logs</code>	timeline auditoría (quién hizo qué)	solo admin

Migraciones completas en `supabase/migrations/`:

- `20260424021420_remote_baseline.sql`: esquema base.
- `20260507000000_initial_schema.sql`: refinamiento.
- `20260507120000_app_credentials.sql`: credenciales cifradas.
- `20260514000000_global_state_sync.sql`: sync state global.
- `20260514120000_phase24_operations_intelligence.sql`: BI + Operaciones.
- `20260515000000_rls_read_policies.sql`: políticas RLS read.
- `20260515120000_profit_snapshots_pk.sql`: snapshots PK.
- `20260516000000_profit_cortes_cd_lado.sql`: cortes lado-cd.
- `20260516120000_turnstile_site_key.sql`: site key Turnstile.
- `20260530_pg_cron_cleanup.sql`: pg_cron schedule cleanup.

CIERRE

Este documento consolida el estado de URG TECHNOLOGY al **19 de junio de 2026** (versión interna v24.18.26). Cubre 293 iteraciones acumuladas desde el lanzamiento V1.0, con énfasis en las decisiones arquitectónicas que no aparecían en V1.1:

- Migración cliente-céntrica → server-céntrica (Edge Functions + pg_cron + RPCs Postgres).
- Modelo atómico de reservas con prevención de overselling cross-method (MP ↔ BRE-B).
- Centinela Salud Operacional como módulo dedicado de monitoreo.
- Migración completa de `/diseno-web` al sitio externo `damorus.com`.
- Pipeline CI/CD GitHub Actions → HostGator FTP con `dangerous-clean-slate` y exclusiones SSL.
- Refactor profundo del Creador de Fichas (sticky tabs + previews reales + Sandbox + Cropper).
- Endurecimiento de seguridad (Black Box + Turnstile invisible + rate limiter + honeypots).
- Reorganización mobile-first del Header, Footer, Ficha 3, Catálogo y todos los flujos críticos.

Para reconstruir este sitio desde cero, basta seguir:

1. Crear proyecto Vite + React 19 + Tailwind v4 conforme a §0 (Arquitectura).
2. Crear proyecto Supabase con las migraciones del Apéndice C.
3. Desplegar las 13 Edge Functions del §8.1.
4. Configurar pg_cron según §8.2.
5. Implementar componentes según §2 (Home), §3 (Pestañas), §4 (Fichas), §5 (Pagos), §6 (Admin).
6. Aplicar todas las reglas de §7 (Lógica transversal).
7. Configurar dominio + HostGator + GitHub Actions según §9.
8. Subir el `.htaccess` definitivo de §7.17.

El resultado es una tienda digital de alto rendimiento que opera 24/7 sin dependencia de presencia humana en el panel admin, con resiliencia ante fallos en la pasarela y trazabilidad completa de cada peso facturado.

Fin del documento V2.0.