



DAMORUS PROJECTS

URG Technology V2 Reference document

Technical specifications for build, operation and deployment

Document version 2.0 · 19 June 2026

Public redacted edition · 15 September 2026

This edition omits credentials, service identifiers, contact details, admin routes, personal data and fees. The rest of the document is published in full.

damorus.com/docs/urg

Contents

URG TECHNOLOGY DIGITAL STORE · VERSION 2.0	5
About this document	5
Main objective (updated)	5
Architecture and Technologies (state as of v24.18.26)	5
Visual identity and Sharing Protocol (Favicon, OpenGraph, Twitter Cards)	7
Section: Strategic SEO and Security Hardening (current state)	8
CONTENTS (Table of Contents)	9
1. VISUAL IDENTITY AND PRICING LOGIC · URG TECHNOLOGY	11
1.1 Color Palette (Proportion Rule)	11
1.2 Price Architecture (Value Anchoring)	12
2. HOME PAGE	13
2.1 Header	13
2.2 Hero Section	14
2.3 CTA Bubbles	15
2.4 Deal of the Day (horizontal Listing #1)	15
2.5 Best Sellers (vertical Listing #1)	17
2.6 We Buy Your PC (Home promotional block)	18
2.7 Web Design Pro (promotional banner → damorus.com)	19
2.8 Our Store (catalog on the Home)	19
2.9 Dynamic SEO Directory	20
2.10 Master Ascent Button (Infinite Return CTA)	20
2.11 FOOTER	21
2.12 WHATSAPP BUTTON (floating FAB)	23
3. TABS / DEDICATED ROUTES	24
3.1 /catalogo · Search Ecosystem and Master Catalog	24
3.2 /productos/:slug · Listing 3 (Product Terminal)	26
3.3 /carrito	27
3.4 /checkout	27
3.5 /confirmacion/:idPedido	27
3.6 /preguntas-frecuentes (canonical) + /faq (alias)	27
3.7 /compramos-tu-pc	28
3.8 /rastrear-pedido	29
3.9 /privacidad	30
3.10 /compra-protegida	31
3.11 /diseno-web → redirect to damorus.com	31
3.12 Catalog shortcuts (/ofertas, /portatiles, /escritorio, /perifericos)	31
3.13 /mas-vendidos	31
4. LISTING SYSTEM	33
4.1 LISTING 1 · Catalog Standard	33
4.2 LISTING 2 · Advanced Search	34

4.3 LISTING 3 · Product Terminal (Final Conversion View)	34
4.4 LISTING ADAPTATION FOR PERIPHERALS	36
5. PAYMENTS SECTION	37
5.1 Billing Data Form	37
5.2 Digital Order Confirmation (Order PDF)	39
5.3 Naming and Traceability System (Order ID)	39
5.4 Atomic Stock Reservation Model (CRITICAL · server-side)	40
5.5 Cross Cancellation BRE-B ↔ MP	44
5.6 BRE-B traffic-light timer (draggable on mobile)	45
5.7 "Your time expired" screen (BRE-B)	45
5.8 Post-MP cart persistence (loss prevention)	45
6. ADMIN PANEL (/[panel])	47
BLOCK I · DAILY OPERATION	47
(position 1, 2, 3 and so on).	56
6 · Continues with BLOCK II (Customer Management)	62
6 · BLOCK II · CUSTOMER MANAGEMENT	63
6.5 URG Community	63
6.6 Messaging and Automation Center (Resend + Supabase)	64
6.7 Frequently Asked Questions Center (FAQ Hub)	69
6 · BLOCK III · INTELLIGENCE AND SECURITY	71
6.8 Statistics and Reports (Business Intelligence)	71
6.9 Intelligence and Surveillance Center (Black Box D1)	73
6 · BLOCK IV · CONFIGURATION AND STRUCTURE	76
6.10 Tax, Financial and Logistics Configuration	76
6.11 Master Panel (Technical Settings and Security)	77
6.12 Profile and Session	81
7. CROSS-CUTTING TECHNICAL LOGIC AND ERROR PREVENTION	82
7.1 Telecommunications Sanitizer (href="tel:")	82
7.2 Dynamic URL Encoder (WhatsApp)	82
7.3 Session and Inactivity Security Protocol	82
7.4 Action Confirmation Modals	82
7.5 Route Security and Middleware (Anti-Bypass)	83
7.6 Data Integrity Protocol (Soft Delete)	83
7.7 Media Optimization (Processing + Preview)	83
7.8 Audit Logic · Abandoned Cart	84
7.9 404 Page · Dark Terminal	84
7.10 Performance · Database Indexing (Supabase)	85
7.11 Real-Time Syntax Validation	85
7.12 Micro-interactions (Toast)	85
7.13 Dynamic Breadcrumb	85

7.14 Data Entry Protection (Anti-XSS)	85
7.15 Resilience Hierarchy (Kill Switch)	86
7.16 Catalog Query Optimization	86
7.17 Server Configuration (.htaccess)	87
7.18 External SDK Verification (Anti-AdBlock)	89
7.19 Timeout Management and Async Loading States	89
7.20 Resource Optimization and Storage Limits	89
7.21 Deployment Integrity and Time Logic	90
7.22 Payment Listening Protocol (Webhook)	90
7.23 Network Status Notification (Passive UX Resilience)	90
7.24 SPA History Synchronization (Native Feel)	90
7.25 Connection Efficiency (Anti-Error 508)	91
8. SERVER-SIDE BACKEND (NEW in V2.0)	92
8.1 Edge Functions (Deno runtime, Supabase Cloud)	92
8.2 pg_cron Trigger	93
8.3 MercadoPago Webhook · Atomic Detail	94
8.4 Cross Cancellation BRE-B ↔ MP	95
9. CI/CD PIPELINE (NEW in V2.0)	96
9.1 GitHub Actions Workflow	96
9.2 FTP Deploy with a Protected Clean Slate	97
9.3 Environment Variables and Secrets	97
9.4 Manual Rollback	97
9.5 HostGator Hosting · Configuration	97
X. COOKIE SYSTEM (URG)	99
1. Technical functionality	99
2. Legal notice interface	99
3. Supabase integration for analytics	99
4. Sales recovery (repurchase)	99
5. Segmentation for ads	99
6. Privacy and transparency	100
X. URG TECHNOLOGY LEGAL DATA	101
APPENDIX A · Summary Changelog (v1.0 → v24.18.26)	102
APPENDIX B · src/ Folder Structure	103
APPENDIX C · Summary Dictionary of Supabase Tables	104
CLOSING	105

URG TECHNOLOGY DIGITAL STORE • VERSION 2.0

Technical Specifications for Build, Operation and Deployment (V 2.0) Update date: 19 June 2026
Product status in production: v24.18.26 deployed at <https://damorus.space> (public mirror) and planned for urgtechnology.com.co

Public redacted version (September 2026). Credentials, service identifiers, internal contact details, admin routes, personal data and fees have been omitted. Every omitted item appears in square brackets, for example [store email].

About this document

This technical document is the major update of "URG TECHNOLOGY TIENDA DIGITAL VERSION 1.1" (27 April 2026). The site was built, debugged and hardened across more than 290 iterations logged in the internal task system; every section of this document reflects the **current, code-verified state** of the store rather than the original aspirational state.

The intent of this V2.0 is that an external team (another agency, an independent developer or a modern Artificial Intelligence) can **rebuild a functionally identical copy** (or at least 99% similar) of URG Technology working only from this document. The same blocks, the same numbering and the same didactic-technical tone of V1.1 are kept; what changes is the content: inconsistencies are rewritten to match reality, new decisions are documented and sections that did not exist are added (Sentinel, server-side architecture, CI/CD pipeline to HostGator, atomic stock reservation model, and so on).

Main objective (updated)

To operate a high-performance **Single Page Application**, built with React 19 + Vite 6 + Tailwind CSS v4 + Framer Motion, connected to a "BaaS" backend on **Supabase** (PostgreSQL + Deno Edge Functions + pg_cron + Realtime + Storage) and deployed as a static build on **HostGator (cPanel, Apache)** via **GitHub Actions with automated FTP** every time a `git push origin main` runs.

The final product is a Colombian B2C store for refurbished and new hardware (laptops, desktops and peripherals) with a proprietary admin panel (`/[panel]`) that governs catalog, payments (MercadoPago + BRE-B with proof of transfer), inventory, logistics, loyalty, transactional messaging (Resend), business intelligence, operational monitoring and security. It works **without any administrator having to keep a tab open**: the critical logic lives on the server (Edge Functions + cron) and the client is presentation only.

Architecture and Technologies (state as of v24.18.26)

Front-End stack

- **Core framework:** React 19.0.0 (functional components, hooks, Suspense, no Server Components: the site is served 100% static).
- **Bundler / dev server:** Vite 6.0.3 with `@vitejs/plugin-react` 4.3.4.
- **Router:** react-router-dom 7.0.2 (Routes with nested Layout, `Navigate`, 301-style redirects via `window.location.replace`).

- **Styles:** Tailwind CSS v4.0.0 with the official `@tailwindcss/vite` plugin. PostCSS 8 and Autoprefixer 10 as the pipeline; no SCSS, no CSS-in-JS, no StyleX.
- **Animations:** framer-motion 11.11.17 (animate presence, layout animations, useInView, useScroll).
- **Image metadata reading:** exifr 7.1.3 (used by `fileForensics.js` to strip EXIF from photos received from the customer in "We Buy Your PC").
- **Language:** pure ESM JavaScript (TypeScript is not used on the client; types live implicitly and are documented with JSDoc where needed).

Backend stack (Supabase Cloud)

- **Active project:** `[project-id]` (region **West US · Oregon**, Free plan, scalable to Pro).
- **Database:** PostgreSQL 15 with Row-Level Security (RLS) enabled on every operational table.
- **Realtime:** native Supabase WebSockets to refresh the admin panel (Pending, Community, Sentinel) without polling.
- **Storage buckets:**
 - `productos/` (product listing galleries, photos optimized to WebP $\leq 1920\text{px}$ and $\leq 350\text{ KB}$).
 - `compramos-pc/` (photos uploaded by the selling customer, purged automatically after 31 days).
 - `comprobantes-brebe/` (PDF/JPG of BRE-B payment proofs, retained until the sale is archived).
- **Edge Functions (Deno runtime):** 13 active functions listed in §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:** extensions enabled. There is a single job (`cleanup-stock-fantasmas-2min`) that invokes the cleanup Edge Function every 2 minutes; details are in §10.4.
- **Auth:** Supabase Auth with TOTP MFA, 15-minute magic links and RLS based on custom JWT claims (`role` and `bloques`).

Transactional messaging stack

- **Resend** (resend.com) for every automated email in the store flow (payment validation, shipments, reviews, reminders, Black Box alerts). Domain verified with SPF, DKIM and DMARC; the API key lives in an encrypted environment variable and is never exposed to the client.
- **Templates** written in HTML with inline CSS, compiled from React Email; persisted in the `plantillas_correo` table so the admin can edit copy and variables without touching code.

Payments stack

- **MercadoPago Checkout Pro** (preferences created by the `mp-create-preference` Edge Function, validated with `mp-validate-token` before any save, and confirmed exclusively by `mp-webhook` on its `payment.updated.approved` event).
- **BRE-B** (Banco de la República: Colombia's immediate interbank transfer). Manual payment: the customer receives a key and QR code, uploads the proof of transfer, and the admin validates it in the panel.
- **Cloudflare Turnstile**, invisible, on the admin login and the trade-in forms to stop bots.

Hosting and deployment

- **HostGator (shared)** with cPanel + Apache 2.4.

- Operational staging domain: **damorus.space** (CNAME pointing to the shared HostGator server). Future brand domain: `urgtechnology.com.co`. Both serve the same static build.
- **Pipeline:** private GitHub repository, `main` branch → GitHub Actions (workflow `Deploy a HostGator via FTP`) → `npm ci` + `npm run build` → FTP upload with `SamKirkland/FTP-Deploy-Action@v4.3.5`. Every `git push` triggers a full build and deploy (~2 min p99). Full coverage in §11.

Output structure and server

- **Final container:** all content lives in the `/dist/` folder (generated by `vite build`). Size stabilized after v24.18.26 at around 1.8 MB gzipped in total (excluding external images).
- **Route handling:** every asset uses relative paths (`./assets/...`). The `index.html` sits at the root of `/dist/` and references the hashes generated by Vite (`main-[hash].js`, `index-[hash].css`).
- **SPA protocol (Apache):** the package mandatorily includes an `.htaccess` with SPA fallback, HTTPS redirect, cache headers, security headers, CSP and blocking of `.env`, `.git*`, `package*.json`, `vite.config.js`. Exact detail in §7.17.
- **Universal encoding:** UTF-8 without BOM in every file. `AddDefaultCharset UTF-8` declared in `.htaccess`.

Optimization and resilience

- **Elite responsiveness:** mobile-first design; meta viewport with `width=device-width, initial-scale=1, viewport-fit=cover`. Standard Tailwind breakpoints (`sm` 640, `md` 768, `lg` 1024, `xl` 1280) plus a custom `xs` 480 breakpoint for small phones.
- **React Error Boundaries** wrapping every route (`src/components/system/ErrorBoundary.jsx`) so an error in an isolated widget does not break the site.
- **Animation throttling:** the `useIsInView` hook pauses expensive Framer Motion animations outside the viewport (introduced in v21: measurable CPU reduction on mobile).
- **Automatic cache busting:** Vite hashes every JS/CSS file by content and emits `Cache-Control: public, max-age=31536000, immutable` headers from `.htaccess`; `index.html` is always served with `no-store, no-cache, must-revalidate`.
- **Build cleanup:** Vite strips `console.log` in production, no source maps in the final bundle, no configuration files exposed.

Visual identity and Sharing Protocol (Favicon, OpenGraph, Twitter Cards)

No change in philosophy versus V1.1, but the asset inventory was consolidated in `public/assets/brand/`:

```

public/assets/brand/
├─ URG_FAVICON_2026.svg          (main favicon source)
├─ favicon.ico                  (32×32 generated from the SVG)
├─ icon-192.png                 (PWA Android/iOS · Apple Touch)
├─ icon-512.png                 (Large PWA)
├─ URG_HOMELOGO_2026.svg        (purple header logo)
├─ URG_FOOTER_2026.svg          (footer logo in matte white)
├─ 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          (vendor signature)

```

Meta tags injected into `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` = absolute path to `og-1200x630.jpg` (served over HTTPS).
- `og:url` = canonical URL of the current page, generated at runtime by `siteConfig.js`.
- `theme-color` = `#8A2BE2` (URG Purple): forces the mobile browser bar to the brand color, removing the "link spam" feel.
- `twitter:card` = `summary_large_image`.

Section: Strategic SEO and Security Hardening (current state)

1. High-impact SEO

- **Semantic hierarchy respected:** a single `<h1>` per page (product name on listing 3, section title on Home), `<h2>` for main benefits, `<h3>` for warranty/specs/support.
- **Dynamic meta tags:** generated with `useEffect` per page (each `pages/*.jsx` sets `document.title`). The product detail composes its title in the format `"{nombreMarketing} | URG Technology"`.
- **sitemap.xml** published at `/public/sitemap.xml` with static routes (`/`, `/catalogo`, `/preguntas-frecuentes`, `/compra-prottegida`, `/privacidad`, `/compramos-tu-pc`, `/rastrear-pedido`). Individual product details are NOT served in the sitemap because they live in Supabase and would rotate daily; instead the site relies on JSON-LD plus organic traffic from `/catalogo`.
- **JSON-LD Schema.org/Product** injected into every `ProductoDetalle.jsx` with name, image, description, brand, offers (price, priceCurrency: COP, availability). If the manual description is under 150 characters, the system automatically concatenates specs and warranty as a fallback block (`pdfBuilder.js` reuses that same logic for the order PDF).
- **alt attributes** generated dynamically: `alt="Computador {nombreMarketing} disponible en Bogotá - URG Technology"`.

2. Hardening of `/[panel]` (Security)

- `/public/robots.txt` declares `Disallow: /[panel]` and `Disallow: /[panel]/*`.
- `<meta name="robots" content="noindex, nofollow">` is injected dynamically into `AdminLogin.jsx` and `AdminLayout.jsx` through `useEffect`.

- **No internal links to the panel** from the public site: the only way to reach `/[panel]` is by typing the URL.
- **Invisible Cloudflare Turnstile** validating the login and the sensitive forms (`TurnstileWidget.jsx` component with the site key in the `VITE_TURNSTILE_SITE_KEY` environment variable and the server-side secret in the `turnstile-config` Edge Function).
- **Local rate limiter** (`src/lib/rateLimiter.js`) that applies a progressive delay of $1 \rightarrow 2 \rightarrow 4 \rightarrow 8$ seconds per failed login within the same session.

CONTENTS (Table of Contents)

- 1. VISUAL IDENTITY AND PRICING LOGIC · URG TECHNOLOGY
- 2. HOME PAGE
 - 2.1 Header
 - 2.2 Hero Section
 - 2.3 CTA Bubbles
 - 2.4 Deal of the Day (Listing #1 horizontal)
 - 2.5 Best Sellers (Listing #1 vertical, 4 categories × top 3)
 - 2.6 We Buy Your PC (9-step form + Zero Stress)
 - 2.7 Web Design Pro (moved to damorus.com · promotional banner)
 - 2.8 Our Store (catalog)
 - 2.9 Dynamic SEO Directory
 - 2.10 Master Ascent Button
 - 2.11 Footer
 - 2.12 WhatsApp Float
- 3. TABS / DEDICATED ROUTES
 - 3.1 /catalogo (Search ecosystem and master catalog)
 - 3.2 /productos/:slug (Listing 3)
 - 3.3 /carrito
 - 3.4 /checkout
 - 3.5 /confirmacion/:idPedido
 - 3.6 /preguntas-frecuentes (with redirect /faq → /preguntas-frecuentes)
 - 3.7 /compramos-tu-pc
 - 3.8 /rastrear-pedido
 - 3.9 /privacidad
 - 3.10 /compra-prottegida
 - 3.11 /diseno-web (301-style redirect to https://damorus.com)
 - 3.12 /ofertas /portatiles /escritorio /perifericos (catalog shortcuts)
 - 3.13 /mas-vendidos (full page with 12 listings)
- 4. LISTING SYSTEM
 - 4.1 Listing 1 (Catalog Standard)
 - 4.2 Listing 2 (Advanced Search)
 - 4.3 Listing 3 (Product Terminal)
 - 4.4 Adaptation for Peripherals
- 5. PAYMENTS SECTION
 - 5.1 Billing Form
 - 5.2 Digital Order Confirmation (Order PDF)
 - 5.3 Naming and Traceability (Order ID YYMM-###X)
 - 5.4 Atomic Stock Reservation Model (NEW · v24.18.15+)
 - 5.5 "Someone was faster!" modal (cross-method overselling)
- 6. ADMIN PANEL (/[panel])
 - BLOCK I · Daily Operation
 - 6.1 Login
 - 6.2 Immediate Control Center (Top bar + Sidebar + Dashboard)
 - 6.3 Listing Creator (8 blocks + Sandbox + 16 seed products)
 - 6.4 Operational Flow (Catalog, Pending, Sold, Returned, We Buy Equipment, Archived)
 - 6.X Sentinel (NEW · Operational Health)
 - BLOCK II · Customer Management
 - 6.5 URG Community
 - 6.6 Messaging Center (Resend)
 - 6.7 FAQ Hub
 - BLOCK III · Intelligence and Security
 - 6.8 Statistics and Reports (BI · 31 metrics)

- 6.8.2 Profit & Commission Manager
- 6.8.3 Report Builder (PDF + CSV)
- 6.9 Black Box (CIV)
- BLOCK IV · Configuration and Structure
 - 6.10 Tax, Financial and Logistics
 - 6.11 Master Panel (API keys, DNS, admins, AI chat)
- 7. CROSS-CUTTING TECHNICAL LOGIC AND ERROR PREVENTION
(7.1 to 7.25 · sanitizers, RLS, soft delete, performance, .htaccess, etc.)
- 8. SERVER-SIDE BACKEND (NEW)
 - 8.1 Complete Edge Functions (13)
 - 8.2 pg_cron cleanup trigger
 - 8.3 MercadoPago webhook · atomic consumption
 - 8.4 Cross cancellation BRE-B ↔ MP
- 9. CI/CD PIPELINE (NEW)
 - 9.1 GitHub Actions workflow
 - 9.2 FTP Deploy with protected clean slate
 - 9.3 Environment variables and secrets
 - 9.4 Manual rollback
- X. COOKIE SYSTEM (URG)
- X. URG TECHNOLOGY LEGAL DATA
- APPENDIX A · Public changelog (v1.0 → v24.18.26)
- APPENDIX B · src/ folder structure
- APPENDIX C · Supabase table dictionary

1. VISUAL IDENTITY AND PRICING LOGIC • URG TECHNOLOGY

1.1 Color Palette (Proportion Rule)

The palette is kept intact from V1.1 and is declared as CSS custom properties in `src/index.css` so that a change is global and components consume it as `var(--urg-purple)` and so on.

```
:root {
  /* 60% Neutral */
  --urg-bg: #FFFFFF; /* pure white background, public store */
  --urg-bg-soft: #F8F9FA; /* extra light gray for alternating tints */
  --urg-bg-admin: #0B0B14; /* admin panel background (dark) */

  /* 30% Primary */
  --urg-purple: #8A2BE2; /* URG purple • CTAs, active links, hero price */
  --urg-purple-soft: #B47BEE; /* hover/glow */
  --urg-purple-deep: #4F1880; /* deep gradients */

  /* 5% Black/Text */
  --urg-text: #0F0F14; /* main text */
  --urg-text-soft: #2D2D38; /* secondary text */

  /* 5% Psychological accents */
  --urg-green: #2ECC71; /* status badges, WhatsApp, free shipping */
  --urg-amber: #F5B400; /* rankings, medals */
  --urg-red: #E11D48; /* sold, errors, someone-was-faster modal */

  /* Prices */
  --urg-price-old: #95A5A6; /* struck-through anchor price */
  --urg-price-hero: var(--urg-purple);
  --urg-badge-bg: #F0E4FF; /* lavender background for the %OFF capsule */
  --urg-badge-text: #4F1880; /* dark purple capsule text */
}
```

Effective application across the site:

- **60% Neutral (Background):** pure white or extra light gray (`#F8F9FA`) alternated per section via the `urg-section-tint` utility class (defined in `index.css`). It keeps the interface clean and professional even after the massive Stripe-style redesign of the admin panel.
- **30% Primary (Brand):** URG purple `#8A2BE2` applied in:
 - Primary action buttons (CTA): the `urg-cta` class groups the shared styles.
 - Active links (`<NavLink>` with `aria-current="page"` receives `text-[var(--urg-purple)]`).
 - Hero price (final sale price): the `PriceDisplay.jsx` component applies it with a `0 0 12px rgba(138,43,226,0.18)` shadow for a subtle neon effect.
 - Glow on the Super Administrator avatar (gold in that case, see §6.2).
- **5% Black:** base text, column headers in admin tables, subtle `border-black/10` borders.
- **5% Accent (Psychology):**
 - Neon green `#2ECC71` → "Condition" badge (New / Like New / Excellent / Good), floating WhatsApp button, "Free shipping in Bogotá" badge.
 - Amber/gold `#F5B400` → gold/silver/bronze borders on the Top 3 best sellers podium (introduced in task #251), warranty medals on Listing 3.
 - Red `#E11D48` → SOLD status (centered stamp, not diagonal, introduced in v24.18.22), border on cards with a stock error, "Someone was faster!" modal.

1.2 Price Architecture (Value Anchoring)

The `PriceDisplay.jsx` component centralizes all price presentation logic. It receives the props `{ precio, precioOriginal, descuentoPct, tamano: "sm" | "md" | "lg" | "xl" }` and renders the correct hierarchy. The admin writes no HTML: they only enter the numeric values in Block 7 of the Creator.

1. "Anchor" price (original)

- **Color:** medium gray `#95A5A6` (variable `--urg-price-old`).
- **Style:** struck through (`line-through`), thin weight (`font-light`), 20% smaller than the hero price (`text-[0.80em]`).
- **Placement:** above the hero price on mobile and in the vertical Listing 1; to the left in Listing 2 and in Listing 3 on desktop.

2. "Hero" price (URG Purple)

- **Color:** vibrant purple `#8A2BE2`.
- **Font:** Inter (the site font) at weight 700 (Bold): Montserrat was dropped in V1.5 to avoid downloading additional webfonts; Inter covers the 100-900 range.
- **Lead size:** `clamp(1.8rem, 2.6vw, 2.6rem)` in Listing 3, `clamp(1.4rem, 2vw, 1.9rem)` in Listing 1.
- **Effect:** subtle shadow (light purple glow): `text-shadow: 0 0 14px rgba(138,43,226,0.20)`. In "Aura ON" mode the shadow intensifies on a 3 s pulse.

3. "Benefit" badge (% OFF)

- **Color:** lavender background `#F0E4FF`, dark purple text `#4F1880`.
- **Style:** small capsule (`rounded-full px-2 py-0.5`) next to the hero price.
- **Text:** "20% OFF" if `descuentoPct` is defined; when the discount is entered as a fixed amount, the system converts it to an implicit percentage and shows the percentage (decision taken in task #253: users scanned the % faster).

1.3 Additional rules learned in production

- **No prefixes in the data.** On every listing, values stand alone (`LA BESTIA DEL GAMING`, not `Name: LA BESTIA DEL GAMING`). This rule, originally from V1.1, was reinforced after incidents where admins wrote "Marketing Name:" as part of the name, polluting the `<h1>` tags.
- **Colombian period as thousands separator.** The `format.js → formatCOP(valor)` helper always emits `$ 1.234.567` (no cents, `.` separator) and is used exclusively: ad-hoc `Intl.NumberFormat` calls inside components are forbidden.
- **Inline VAT only when it applies.** If the product exceeds 50 UVT and is taxable, Listing 3 shows a gray sub-text reading `"IVA 19% incluido en el total"`. If it is excluded, it shows `"Excluido de IVA (Art. 424 E.T.)"`. The component that paints the sub-text reads the `producto.ivaModo` flag computed on the back end.
- **No shadow/glow on prices when the product is SOLD.** The red SOLD overlay disables price shadows (the centered stamp introduced in v24.18.22 already carries visual weight, and the price shadows would compete with it).

2. HOME PAGE

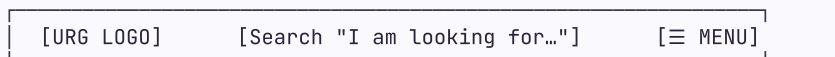
The Home page is made up, in scroll order, of: Header + compact Hero + CTA Bubbles + Deal of the Day + Best Sellers + We Buy Your PC (promotional block) + Web Design Pro (banner leading to damorus.com) + Our Store + Footer. The Dynamic SEO Directory was removed from the Home after task #73 (it saturated the page visually without contributing measurable conversion) and now lives only as an embedded block in the Footer and on some tab routes.

Every Home block has its component under `src/components/home/*.jsx` and is composed in `src/pages/Home.jsx`. Sections alternate their background tint through the `<div className="urg-section-tint">` wrapper, which applies a subtle `background: #F8F9FA` to create visual rhythm without breaking the store's white unity.

2.1 Header

Component: `src/components/Header.jsx`. Visible on ALL public routes, mounted inside `<Layout />`.

Desktop structure (≥ md)



- **URG logo:** `URG_HOMELOGO_2026.svg`, 36 px tall on desktop, 28 px on mobile. Clicking it goes to `/`. If the user is already on `/`, the click runs `scrollTo(0)` with `behavior: 'smooth'` instead of navigating (task #163). This avoids an unnecessary reflow and reinforces the feeling of "a single living page".
- **Search box "I am looking for...":** the `SearchAutocomplete.jsx` component, fluid width (max 640 px on desktop). It implements the real-time autocomplete (Primary Layer / Technical Layer / Invisible Tags Layer) described in §3.1; exhaustive behavior is covered in that section.
- **Hamburger menu:** a 320 px right-hand sidebar that slides in from the right with Framer Motion (`x: '100%' → 0` with a soft 240 ms spring). It contains:
 1. Home
 2. Deals (when there are products on offer; always visible after task #74)
 3. Catalog
 4. Best sellers
 5. We Buy Your PC
 6. Peripherals (auto-disabled until the first product with a peripheral category exists: task #16)
 7. Technical Service (leads to the [store WhatsApp])
 8. Track my order
 9. Web Design Pro (green "New" label; clicking it opens `https://damorus.com` in a new tab as of v24.18.26)
 10. **Divider + "Customer service" section** (task #159) with three extra entries: Privacy policy · Protected purchase · Frequently asked questions, each with a discreet emoji for fast visual parsing.

The mobile menu applies `position: fixed; inset: 0; background: rgba(11,11,20,0.92)` with a 12 px blur behind it (`backdrop-filter: blur(12px)`) so the Home content stays faintly visible underneath.

Mobile structure (< md)

The logo is compacted to 28 px and the search box moves below the header into a second thin bar (task #27: the original bar was very tall and stole vital viewport from the Hero). The new total mobile header height is 96 px (48 px logo + 48 px search) instead of the original 120 px.

2.2 Hero Section

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

The Hero is **compact and declarative**; it does not try to grab attention through size but through presence. That was a conscious decision by the founder: "the customer is in a luxury terminal that does not need big headlines". After multiple iterations (task #285 among others), the Hero ended up like this:

Visual composition

- **5-layer background** stacked with `position: absolute`:
 1. **Base gradient.** `linear-gradient(135deg, #2a0f5a 0%, #1a0840 35%, #0d0717 70%, #1a0f30 100%)`. Always visible even if the other layers fail.
 2. **Admin-uploaded blurred photo.** If `extra.fondo.imagen` is loaded into the `urg.heroExtra` bucket, it is injected as a `background-image` with `filter: blur(3px) saturate(1.1) brightness(0.8); transform: scale(1.05); opacity: 0.55; mix-blend-mode: overlay`. This is the "blurred gaming computer" from V1.1, now editable.
 3. **Floating neon orbs.** The `<NeonOrb />` component paints a diffuse radial circle in a configurable color. The Hero carries two: purple at `380 px` on `top: -10% / left: 62%` (opacity 0.40) and cyan at `260 px` on `top: 55% / left: 8%` (opacity 0.25).
 4. **Admin decorative tint.** When the admin picks a tint (Aura URG / Crazy URG / Cyber-Black / Orgullo Patrio / Alerta Oferta), it is injected as a gradient layer over the orbs.
 5. **Central darkener.** `radial-gradient(ellipse 70% 70% at 50% 50%, rgba(13,7,23,0.45) 0%, rgba(13,7,23,0.10) 60%, transparent 100%)` to guarantee contrast for the white text on top.
- **minHeight** of the section: `min(24vh, 240px)` (task #285; the previous 38vh/320px left an empty gap on mobile).
- **Vertical padding:** `py-6` on mobile (24 px) / `py-10` on desktop (40 px). This replaced the original `py-10 / py-14`, which produced dead visual gaps.

Text content



- **Headline.** White text, `font-weight: 800`, `font-size: clamp(1.7rem, 3.2vw, 2.4rem)`, `letter-spacing: -0.02em`. It animates with Framer Motion `initial={{opacity:0, y:18}}` `animate={{opacity:1, y:0}}` over 0.7 s with a cubic-bezier ease of `[0.22, 1, 0.36, 1]`.
- **"Free shipping in Bogotá" sub-badge.** A customizable component (task #16): the admin picks the icon (a small truck SVG by default), the background color (neon green by default), the text color (auto/white/dark, task #19) and the literal text. It appears wrapped in a fully rounded pill.

- **Horizontal anchoring.** The title + badge block uses `flex-col items-center justify-center` (refactored in v24.18.17): previously, with grid, the items stuck to the top instead of being centered, and the badge could throw the title off balance horizontally.

Both the headline and the badge are **editable from Master Panel > Block K (Home Banner)**.

2.3 CTA Bubbles

Component: `src/components/home/BurbujasCTA.jsx`. Horizontally scrollable **even on desktop** (task #23: previously on desktop they were statically centered and wasted whitespace).

Current order of the bubbles

1. **Deals:** percentage icon, orange-red gradient background, leads to `/ofertas`.
2. **Catalog:** grid icon, leads to `/catalogo`.
3. **Laptops:** laptop icon, leads to `/portatiles`.
4. **Desktops:** tower icon, leads to `/escritorio`.
5. **We Buy Your PC:** banknote + arrow icon, leads to `/compramos-tu-pc`.
6. **Web Design Pro:** globe icon, flagged `external: true` (introduced in a post-v24.18.26 task), opens `https://damorus.com` in a new tab.
7. **Peripherals:** mouse icon, visible only if there are products with an active peripheral category.

Technical specifications

- **Size:** 120 × 120 px on desktop, 96 × 96 px on mobile (slightly reduced so that 4 bubbles fit on an iPhone SE screen without scrolling).
- **Shape:** `border-radius: 50%` (perfect circle). 2 px border in `var(--urg-purple)` with `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%)`.
- **Active state:** hover ⇒ `transform: translateY(-4px) scale(1.04)`, `box-shadow: 0 8px 26px rgba(138,43,226,0.30)`.
- **Mobile horizontal scroll:** `overflow-x: auto; scroll-snap-type: x mandatory`. Each bubble has `scroll-snap-align: center` so they center themselves when swiped.

2.4 Deal of the Day (horizontal Listing #1)

Component: `src/components/home/OfertaDelDia.jsx`. It is the piece with the highest UX investment on the Home and has gone through 12 logged iterations (tasks #246, #251, #257, #273, #274, #275, #276 among others).

Visual identity and container

- **Background:** deep black card (`background: linear-gradient(135deg, #07070A 0%, #1a0f30 100%)`).
- **Border:** subtle pink-purple-cyan neon effect with a `urg-pulse` animation (defined in the `index.css` keyframes: opacity oscillates 0.6 ↔ 1 every 2.8 s).
- **Eyebrow:** giant pill with a `linear-gradient(90deg, #ff6b6b 0%, var(--urg-purple) 60%, #B47BEE 100%)` gradient, white text at `font-black uppercase letter-spacing 0.20em`, a 🔥 icon on the left, and a `0 4px 14px rgba(255,107,107,0.40)` shadow. Text: `OFERTA DEL DÍA`.

- **Countdown** in `HH:MM:SS` format, resetting at **8:00 AM Colombia time (UTC-5)**. The time-zone logic is implemented with a fixed offset inside the component (lines 53-78 of the file); `Intl.DateTimeFormat` was avoided so the final `Date` does not depend on the customer's browser.

Layout and responsive behavior

Desktop ($\geq$ md)

- **Two asymmetric columns** on a 12-column grid: the image takes columns 1-7 ($\approx 58\%$), the info panel takes columns 7-12 ($\approx 42\%$). The ratio was adjusted from 60/40 to 58/42 in task #257 when the cropper-based framing system was introduced.
- **Automatic gallery** with `setInterval(4000)`. If there is only 1 image, the interval is never mounted (early return in `useEffect`).
- **Loading bars** under the image: thin 3 px tall bars that fill with purple over 4 s (`@keyframes barFill`). When a bar reaches 100% the image changes to the next one with `AnimatePresence` (350 ms fade).
- **Main CTA** in the lower right corner of the panel: "BUY NOW!" with a sweeping glow effect (`@keyframes urg-glow-sweep`) every 3 seconds. Hover $\Rightarrow$ `scale: 1.05` and a more saturated purple. Clicking opens the product's Listing 3 directly at `/productos/{slug}` (the historical decision to jump straight to checkout was reversed in task #239: the customer must see the listing before buying in order to reduce returns).

Mobile ($<$ md)

- **Vertical stacking**, image on top at 16:9 (`aspect-ratio: 16/9`), info panel underneath.
- **Touch carousel** with `overflow-x: auto; scroll-snap-type: x mandatory` when there is more than 1 image. Indicator dots are 8 px on mobile (vs 4 px on desktop) for an accessible tap target.
- **Peek-a-boo**: the card does NOT cover the whole screen: it leaves about 12% of the following "Best Sellers" block peeking out underneath (an original V1.1 task, preserved).
- **"Buy Now" button**: 48 px tall (standard Apple HIG touch target), 100% panel width. Positioned below the price so the right thumb can reach it without a cradle gesture.

Listing components (in order of appearance)

1. **Marketing name**: bold, max 2 lines. If it overflows, it is truncated with `text-overflow: ellipsis` and a tooltip appears on desktop hover.
2. **Brand + model**: a single line, `text-zinc-400`, with no "Brand:" prefix label (rule §1.3).
3. **Original price** struck through in medium gray.
4. **Deal price** in URG purple at a large size.
5. **OFF badge** lavender capsule with `XX% OFF`.
6. **Shipping**: green chip "Free shipping in Bogotá" with a truck SVG icon.
7. **CTA "BUY NOW!"** described above.
8. **Click on the body** (not on the CTA): the whole container is a `<Link>` to Listing 3.

State logic and inventory control

SOLD state (psychological scarcity)

When `stockDisponible` reaches 0 (computed in real time by `lib/orders.js` $\rightarrow$ `getStockDisponibleAhora()`, see §5.4), the component mutates:

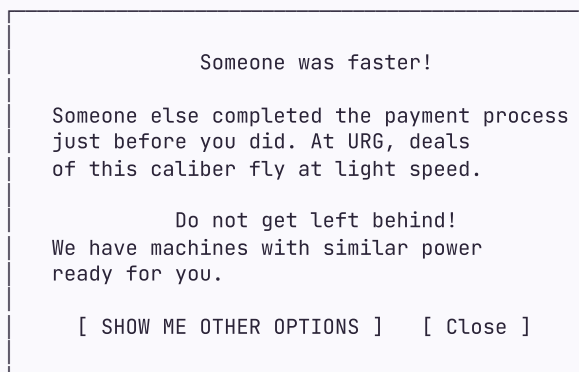
- **Visual:** the image does NOT turn black and white (decision reverted in task #243: the real photo sells more even when the item is sold); instead, a translucent red overlay `rgba(225,29,72,0.55)` is superimposed with a centered stamp.
- **Centered stamp** (not diagonal) reading `VENDIDO` (task v24.18.22): `font-size: clamp(1rem, 1.35rem, 2rem)`, weight 900, letter-spacing 0.10em, rounded white background, shadow `0 6px 18px rgba(0,0,0,0.55)`.
- **Central message under the stamp:** `MISIÓN CUMPLIDA: VENDIDO` with the sub-text `Regresa mañana para más ofertas de locura`.
- **The "BUY NOW!" CTA is replaced** by a "Notify me when it is back" button: a subscription system introduced in a post-v24.18.22 task, which stores the email plus the product ID in `lib/avisame.js`. When the product is republished with stock > 0, a Resend email is fired and a floating `<AvisameNotificacion />` component appears in the lower left corner of the site for subscribed users.
- **Blocking:** clicks on the listing are ignored (`pointer-events: none` except for the Notify-me button and the sale-saving WhatsApp).

Collision control (race condition)

The server-side atomic reservation system guarantees that two simultaneous purchases of the same unique product can never both complete. Full detail in §5.4: here it is enough to know that the Deal of the Day benefits from the same logic.

"Someone was faster!" modal

If the customer is paying by BRE-B and another user consumes the stock with MercadoPago at the same instant (or the other way around, when an admin approves a BRE-B while another customer is inside MP), the `<StockAgotadoModal />` modal defined in `src/pages/Confirmacion.jsx` is triggered:



The title and the main text are shown in **red** (`color: var(--urg-red)`, a founder decision to reinforce urgency). The CTA opens `/catalogo` and keeps the category filter of the lost product applied by default. The modal listens for the `FALLIDO_EXP` event that the webhook fires when it runs `cancelarHuerfanosPostConsumo()`: details in §5.5.

2.5 Best Sellers (vertical Listing #1)

Component: `src/components/home/MasVendidos.jsx`. It renders the TOP 1 of each of the 4 parent categories.

Categories and subtitles

1. **Gaming:** *Dominate the battlefield*, focused on FPS and graphics.

2. **Home / Study:** *Your unfailing companion for schoolwork*, focused on reliability, light weight and price.
3. **Creators:** *Unlimited power for your creativity*, focused on RAM and processing.
4. **Business:** *For building your future*, professional solutions.

Layout

- **Desktop:** `grid grid-cols-4 gap-6`. Each column is a vertical Listing #1 (image on top, info below, no visible button: the whole card is clickable, following the "click anywhere" psychology rule).
- **Mobile / tablet:** horizontal scroll with snap (`overflow-x-auto snap-x snap-mandatory`). The first column is fully visible and about 15% of the second one peeks in. An indicator with four 8 px dots sits underneath.
- **"See the full catalog →" link** underneath (task #283), visible only on mobile and tablet: it leads to `/mas-vendidos` (a dedicated page with all 12 positions: 4 categories × 3 positions = 12 products).

Hover effects (desktop)

- **Translation:** `translateY(-5px)`.
- **Shadow:** a soft `0 12px 24px rgba(0,0,0,0.08)` shadow appears.
- **Text highlight:** the PC name changes to the **category color**:
 - Gaming → URG purple.
 - Home/Study → cyan (`#00CCFF`).
 - Creators → neon green.
 - Business → white/silver (`#E5E7EB`).

Deal-of-the-Day-in-Best-Sellers badge

If the product that is currently the Deal of the Day belongs to one of the categories shown, its card in Best Sellers receives a fire badge instead of the golden #1 badge (task v24.18.7, fixed in v24.18.13). This prevents the customer from seeing the same product twice in a row without context and reinforces the urgency of the offer.

2.6 We Buy Your PC (Home promotional block)

Component: `src/components/home/CompramosTuPC.jsx`. This is only the **block that invites** the visitor to use the form; the real form lives in `src/pages/CompramosTuPC.jsx` (route `/compramos-tu-pc`).

Block content

WE BUY YOUR PC: INSTANT CASH OR AN UPGRADE.

We pay the best price for your machine. Get cash for whatever you need or use it as part of the payment. No complications!

- | | |
|---------------------|-----------------------|
| ⚡ Express Valuation | in under 5 minutes |
| 😌 Zero Stress | just upload photos |
| 🇪🇺 No Obligation | you choose the option |
| 🎁 Gift Bonus | 5% extra discount |

[SELL MY PC NOW →]

The CTA leads to `/compramos-tu-pc`, which is already the complete 9-step form. The dedicated page is documented in §3.7.

2.7 Web Design Pro (promotional banner → damorus.com)

Component: `src/components/home/DisenoWebPro.jsx`. In V1.1 this block was an extensive sales section with 3 plans and CTAs pointing to a `/diseno-web` tab. After the strategic decision of v24.18.26, **the entire Web Design Pro feature moved to the external site `damorus.com`**, and only this promotional banner stayed on URG.

Current content

Is your brand still not selling online?
We turn your business into a digital asset.

[VISIT DAMORUS.COM]

Professional web design · AI · Payments · BI

- The CTA is an `<a href="https://damorus.com" target="_blank" rel="noopener noreferrer">`. It has no ↗ arrow (removed in task #144: it caused confusion about whether the button left the site or not).
- The block has an embedded dark theme (`background: linear-gradient(135deg, #0B0B14, #1a0840)`) to set it apart visually from the rest of the white Home and reinforce the "jump" to another brand.
- The banner is controlled by a **toggle** in Master Panel > Block L: if the admin turns it off, the block disappears from the Home without a recompile.

Legacy `/diseno-web` route

So as not to break links shared before the migration, the `/diseno-web` route was left as a redirect:

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

The `ExternalRedirect` component runs `window.location.replace(to)` inside `useEffect`. This preserves the SEO value of old links.

2.8 Our Store (catalog on the Home)

Component: `src/components/home/NuestraTienda.jsx`. It shows a compact product grid with chip-style filters.

Layout

- **Parent category chips** right below the title:
 1. ⚡ All (default)
 2. 🎮 Gamer World
 3. 🎨 Creators
 4. 💼 Business
 5. 🏠 Home / Office
 6. 🔥 On sale
- Horizontal scroll on mobile, centered distribution on desktop.

- Tapping a chip instantly refilters the grid below with a Framer Motion `layout` animation plus a fade.

Grid

- **Desktop:** 4 columns (`grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6`).
- **Mobile:** 2 columns so the user sees more options with less scrolling. On tablet it is also 2 columns; on small mobile (xs) it drops to 1 full-width column.
- **Mobile cap:** 6 products maximum (task #22). If there are more, a "Load more machines" button is shown that replaces the initial load; on desktop it starts with 8 or 12 products.

Dynamic inventory counter

Under the expansion button there is a small line of text: *Showing 12 of 48 available machines*. The total number is computed with `getCatalogoActivo().filter(p => p.stock > 0).length`.

Clicking a product

Every card is a `<Link to="/productos/{slug}">`. It preserves the Home scroll position when the user comes back (thanks to `<ScrollRestoration>` from react-router v7).

2.9 Dynamic SEO Directory

Current decision: this block does NOT appear on the Home (removed in task #73 for being noisy). It is still active, however, in:

1. The site **Footer** (all pages), as a condensed horizontal block reading "EXPLORE OUR ECOSYSTEM".
2. The **dedicated tab routes** (`/ofertas`, `/portatiles`, `/escritorio`, `/perifericos`) where it appears just above the Footer to reinforce SEO relevance.

Component: `src/components/home/DirectorioSEO.jsx`. The generation logic is the same as in V1.1: it scans active categories and distinct brands in `adminProducts.published`, deduplicates them and renders them as clickable `[ Name ]` chips.

2.10 Master Ascent Button (Infinite Return CTA)

Component: `src/components/MasterAscent.jsx`. It is mounted in `Layout.jsx` between `<Outlet />` and `<Footer />`, which guarantees it appears at the bottom of EVERY public page (not only the Home).

Specifications

- **Width:** 100% (full width).
- **Height:** 48 px.
- **Background:** very dark gray `#16161f` with 1 px top and bottom borders in `rgba(138,43,226,0.18)`.
- **Hover:** the background transitions to `var(--urg-purple)` and the text turns matte black.
- **Text:** centered, uppercase, `font-weight: 700`, letter-spacing 0.10em, reading `INICIO DE PÁGINA`.
- **Action:** `window.scrollTo({ top: 0, behavior: 'smooth' })`. If the user is on `/`, it also runs `history.pushState({}, '', '/')` to clear the hash if there is one.
- **Conditional visibility:** it is hidden when the user is on `/[panel]`, `/[panel]/*`, or when the document scroll is ≤ 600 px (there is no point offering "go up" when you are already at the top).

2.11 FOOTER

Component: `src/components/Footer.jsx`. It has two conditionally rendered versions: **FooterMobile** (visible only `< md`), optimized in a thumb-friendly logical order, and **the classic 4-column grid** (visible `≥ md`).

2.11.1 Shared aesthetics

- **Visual atmosphere:** `linear-gradient(180deg, var(--color-urg-footer-top) 0%, #1a0840 25%, #0a0014 75%, var(--color-urg-footer-bottom) 100%)`. The top/bottom variables let the admin adjust the purple intensity from Block L.
- **Text coloring:** all text in very light lavender `#d8c2ff` with reduced opacity so the contrast against the background is not aggressive.
- **Finish:** minimalist yet immersive: the footer is a premium extension of the interface, not a block glued on at the end.

2.11.2 Desktop version (`≥ md`) · 4 columns

IDENTITY	TRUST CENTER	CONTACT	FIND US
[URG logo in pure matte white]	• Track your order • Payments & warranty • Privacy policy • Frequently asked quest.	General support 📍 [address] ✉ [email] 📱 [store WhatsApp] Specialists 🔧 Hardware: [redacted] 🔧 Software: [redacted] 🌐 Web design: Alejandro@damorus.com	[Google Maps iframe centered on the address] Tapping the map opens maps in a new tab ACCEPTED logos below (static 1×6 strip)
URG TECHNOLOGY SAS	★ Rate our service (see payment)		
Tagline...			
Hours			
M-F: 9:15-6			
Sat: 9:15-2			
Sun: Closed			
Store approved and supervised by: [DIAN logo grayscale] [CCB logo grayscale]			
© 2026 URG TECHNOLOGY SAS · Powered by Damorus.com [tiny white Damorus logo]			

Column 1 · Identity

- **URG logo in pure white.** A `filter: brightness(0) invert(1)` is applied to the original `URG_FOOTER_2026.svg` SVG to force matte white without needing an extra asset. Hover removes the filter and shows the original corporate colors (the same pattern used for DIAN/CCB).
- **Legal name:** `URG TECHNOLOGY SAS.`
- **Tagline:** *Colombian company leading in refurbished technology and web ecosystem development.*
- **Hours:**
 - Mon-Fri: 9:15 am to 6:00 pm
 - Sat: 9:15 am to 2:00 pm
 - Sun and holidays: Closed

Column 2 · Trust Center

- Vertical list with small purple bullets.
- "Rate our service" carries a **green** bullet (instead of purple), followed by a ★ star and an external icon, to stand out from the rest.
- Hyperlink to `[Google Maps link]` (editable from Master Panel § Contact Master).

Column 3 · Contact

- **General support** with address, email and main WhatsApp.
- **Specialists (Hardware / Software / Web Design)** with direct WhatsApp (the Web Design one changed to the email `Alejandro@damorus.com` after the v24.18.26 migration).
- Every phone number uses `href="tel:..."` with the number **sanitized** (no spaces or parentheses) by the `sanitizePhone()` helper in `lib/format.js`.
- Every WhatsApp link uses `https://wa.me/{telefono}?text=...` with `encodeURIComponent()` applied to the message.
- Editable from Master Panel § Contact Master. It allows adding non-primary rows for additional specialists.

Column 4 · Find Us

- **Google map** embedded as an iframe. The embed string is read from `contact.direccion?.iframeEmbed` (editable in Block L): by default `[Google Maps iframe of the office]`.
- Clicking the iframe is intercepted by the outer `<a>` wrapper, which opens `[Google Maps link]` in a new tab.
- **"We accept" strip** under the map: 1 × 6 static logos (Mercado Pago, BRE-B, Visa, Mastercard, American Express, Nequi). After multiple iterations (#29, #31, #36, #40, #42, #267, #268) the conclusion was: every logo has its scale adjusted through a `scale` prop to compensate for differences in internal whitespace, Mercado Pago `1.52×`, Mastercard `1.40×`, American Express `1.30×`, Visa/BRE-B/Nequi `1×`. The strip is **static** (no marquee, decision #40: the marquee was distracting and added little).

2.11.3 Trust Bar

A "semi-luxury" band right below the grid:

- **Background:** very dark gray with a subtle `noise.svg` texture layered on top.
- **Text:** *Store approved and supervised by:* in `font-medium` lavender.
- **Logos:** `DIAN_LOGO_2020.svg` and `CCB_LOGO_2026.svg`. Grayscale by default with opacity 0.75; on hover they regain their original color (a drop-shadow filter is added).

2.11.4 Closing

- **Text:** © 2026 URG TECHNOLOGY SAS: Powered by Damorus.com (the "Damorus.com" is an invisible hyperlink leading to `https://www.damorus.com`).
- **Damorus logo:** `DAMORUS_BLANCO.svg` below the text. Natural state at 70% opacity; hover raises it to 100% and applies `filter: drop-shadow(0 0 8px #ffffff)` to simulate the phoenix glowing.

2.11.5 Mobile version (FooterMobile)

Internal `FooterMobile` component (defined in the same `Footer.jsx` file). It reorganizes everything for the thumb:

A. Auto-marquee payment strip (the marquee is kept on mobile only) B. 2 large priority Trust buttons: **Track your order** + **Rate our service** C. Compact hours D. Find Us (small map + address) E. "Contact + Specialists" accordion, collapsed by default F. "More links" accordion (privacy, FAQ, etc.), collapsed G. Minimal branding with logo and copyright

The Trust Bar (DIAN/CCB) and the Damorus closing stay the same and appear below (shared).

The "We accept" section on mobile was **renamed and redesigned** in task #21: the strip now lives in block A (above everything) and the "We accept" text was removed because it was redundant given the Footer context.

2.12 WHATSAPP BUTTON (floating FAB)

Component: `src/components/WhatsappFloat.jsx`. Visible on ALL public routes except when the user is on `/checkout` (to avoid interfering with payment CTAs).

- **Position:** `position: fixed; right: 20px; bottom: 20px; z-index: 50`.
- **Button:** circular 56 × 56 px, WhatsApp green background `#25D366` with a `0 6px 16px rgba(37,211,102,0.45)` shadow.
- **Auto-appearing pop-up:** after 4 seconds on the page, a white bubble appears to the left of the button reading "Any questions? Get in touch." If the user dismisses it (X), the preference is stored in `localStorage` for 24 h so it is not shown again.
- **Click:** opens `https://wa.me/[store WhatsApp]` (main number, sanitized).

The WhatsApp Float is **NOT** shown on `/[panel]/*` (admin panel). The admin has their own internal channels.

With section 2 closed, the following pages document each dedicated tab of the site (routes that are not the Home), including the v24.18 decisions about `/compramos-tu-pc`, `/preguntas-frecuentes` (canonical) and the migration of `/diseno-web`.

3. TABS / DEDICATED ROUTES

Unlike V1.1, where the "bubbles" were conceived as overlays on top of the Home, in the final implementation each tab was resolved as its **own SPA route** inside the public Layout. The reason was pragmatic: accessibility, deep linking, the browser "back" button and SEO indexing all work better when each destination is a real URL instead of a temporary overlay. The "bubble without a reload" feel is preserved because all internal navigation is client-side (react-router) with no real document reloads.

Full list of active routes (declared in `src/App.jsx`):

```
/                → <Home />
/catalogo        → <Catalogo />
/productos/:slug → <ProductoDetalle />
/carrito         → <Carrito />
/checkout        → <Checkout />
/confirmacion/:idPedido → <Confirmacion />
/privacidad      → <Privacidad />
/compra-protégida → <CompraProtegida />
/compramos-tu-pc → <CompramosTuPC />
/rastrear-pedido → <RastrearPedido />
/preguntas-frecuentes → <FAQ /> (canonical)
/faq             → <Navigate to="/preguntas-frecuentes" replace />
/disenio-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 · Search Ecosystem and Master Catalog

Component: `src/pages/Catalogo.jsx`. It is the most complex module of the public front end; it replaces the original "bubble over the Home" idea with a complete page featuring a desktop filter sidebar and a mobile top bar, while keeping the SPA feel.

3.1.1 Multidimensional Indexing Architecture

Implemented in `src/lib/smartSearch.js`. The algorithm scans three layers in priority order:

1. **Primary layer (Title and Brand)**. Direct match on `nombreMarketing` and `marca`. Weight 3×.
2. **Technical layer (Attributes)**. Match on `procesador`, `ram`, `gpu`, `almacenamiento`. Weight 2×.
3. **Invisible tags layer**. The admin can assign hidden tags (the `tagsBusqueda: string[]` field on the listing). Weight 1×. Example: a product with `nombreMarketing="LA BESTIA DEL GAMING"` and `tagsBusqueda=["arquitectura","render","cad"]` shows up when searching for "Renderizado" even though the word is not in its name.

The algorithm is **accent-insensitive**

(`String.prototype.normalize("NFD").replace(/p{Diacritic}/gu,"")`) and **case-insensitive**. It handles light spelling mistakes through a Levenshtein distance ≤ 2 (the `fuzzyMatch` function in `smartSearch.js`).

3.1.2 Predictive Autocomplete

- **Placeholder**: "Estoy buscando..." (text in gray `#9ca3af`). It clears on focus (`onFocus={e ⇒ e.target.placeholder = ''}`) and is restored on blur).
- **Activation**: from the third character onward (`if (query.length < 3) return`).

- **Types of suggestions shown** in the floating menu (max 5):
 - Term suggestion: "Lenovo" (leads to `/catalogo?q=Lenovo`).
 - Specific product suggestion: "Lenovo Legion Slim 5" (leads to `/productos/{slug}`).
 - Category suggestion: "See all Lenovo laptops" (leads to `/catalogo?cat=portatil&marca=Lenovo`).
- **Local persistence:** the user's last 3 searches are stored in `localStorage['urg.searchHistory']` (module `src/lib/searchHistory.js`). They appear on focus with an empty field under the eyebrow "Recent searches".
- **Global top searches** (Weekly Trends): computed from the `bi_searches` table, filtering for the last 7 days and grouping by normalized term. They appear under the eyebrow "Trending". If the user types a letter, the system **prioritizes** the global terms starting with that letter (instead of showing the fixed global list).

3.1.3 Smart Search · Zero Results

If the search finds no match, showing an empty page is **strictly forbidden**. The interface shows:

```
We could not find an exact match for "macbook 2010",
but our specialists suggest these high-performance systems:

[Card 1]  [Card 2]  [Card 3]
(3 machines in the same price range or category, in stock)

Not what you were looking for?
[ TALK TO AN ADVISOR ON WHATSAPP ]
```

The three alternatives are selected as follows: same `categoria` → same `rango_precio` ($\pm 25\%$) → ordered by `stockReal` descending. If there is nothing in the same price range, the rule relaxes to the same category and the ranking order is kept.

3.1.4 Tactical Header (Sticky)

A horizontal sticky bar with `position: sticky; top: 96px` (below the header). It contains:

- **Dynamic breadcrumb:** *Home > Catalog > Laptops > Lenovo*.
- **Vital counter:** "Showing 1-12 of 48 configurations found for 'Lenovo'".
- **Sort selector** (dropdown):
 1. Relevance (Featured)
 2. Most recent (Default)
 3. Price: low to high
 4. Price: high to low
 5. Best sellers (Ranking #1, #2, #3)
 6. Excellence grade (New > Like New > Excellent)

3.1.5 Progressive Visual Navigation System

Desktop ≥ lg. Persistent left sidebar (320 px) with two layers:

- **Primary layer · Mirror bubbles (What is it?):** a vertical list of circles with icons.
 - [⚡ All]
 - [💻 Laptops]
 - [🖥️ Desktops]
 - [🖨️ Peripherals]

- Glassmorphism + neon border when active.
- Exclusive logic (selecting one deactivates the others).
- **Secondary layer · Personality chips (What am I going to use it for?):** appears after a primary option is selected.
 - [GAMER WORLD] [CREATORS] [BUSINESS] [HOME] [DEALS]
 - Cumulative rectangular pills (you can combine Laptops + Gamer World + Home).

Mobile / Tablet. The bubbles and chips move to the top in two compact rows. The user navigates with a horizontal thumb scroll. There is also a "FILTERS" button (ghost style with an animated arrow and an active state) that opens a modal with advanced filters: task #24.

3.1.6 Advanced filters (admin-generated sidebar)

Priority filters by default, all dynamic over the active inventory:

- **Brand** (auto-generated).
- **Processor** (Ultra tier i7/Ryzen 7+, Mid tier i5/Ryzen 5, Entry tier i3/Celeron).
- **RAM** (8GB, 16GB, 32GB+).
- **Storage** (128, 256, 512, 1TB, 2TB+).
- **Condition** (New, Like New, Excellent, Good).
- **GPU** (Dedicated graphics, Integrated graphics).
- **Size** (Small / Medium / Large / Extra Large).
- **Weight** (Featherweight / Light / Moderate / Heavy).

Each filter is rendered as a collapsible accordion. Active filter chips appear below the catalog header with an X to remove them individually. A "**Clear filters**" button (trash icon) sits in the upper right corner of the sidebar.

3.1.7 Grid and loading

- **Distribution:** desktop 6 columns at `xl`, 4 at `lg`, 3 at `md`, 2 on mobile.
- **Incremental loading:** it initially shows 18 products. On reaching the end of the scroll, an IntersectionObserver fires `loadMore()`, which adds another 16. There is also an explicit "LOAD MORE MACHINES" button for users who prefer manual control.
- **Position memory:** when coming back from Listing 3 through the "Back" button, the catalog restores the scroll position and the filters (via `sessionStorage['urg.catalogo.estado']`).

3.1.8 Search closing block

At the end of the list there is a retention block:

Do you need a custom configuration?
 [Frequently Asked Questions Hub] [Contact us on WhatsApp]

3.2 /productos/:slug · Listing 3 (Product Terminal)

Documented exhaustively in §4.3. Only the routing is covered here.

- The `slug` is generated by the back end when the listing is created (kebab-case, no accents, no special characters).
- If the slug already exists, a numeric suffix is appended (`macbook-pro-m2`, `macbook-pro-m2-1`).

- If the requested `slug` does not exist in `adminProducts.published`, the component briefly shows a skeleton and redirects to `/catalogo` with a floating notice reading "Product not found or sold out" (task v18.5).
- It implements dynamic `<meta>` tags (title, description, og:image, og:url, twitter:card, JSON-LD Product) injected with `useEffect` when the product loads.

3.3 /carrito

Component: `src/pages/Carrito.jsx`. It lists the added products using the `CartProvider` logic (context in `src/main.jsx`).

- **One line per product** with image (80×80), name, chosen color/size where applicable, unit price, quantity (-/+ input), subtotal and a remove button (X).
- **Real-time stock validation:** on cart load and every 30 s it recomputes `getStockDisponibleAhora(productoId)`, and if stock dropped to 0 it marks the line in red with the text "No longer available: remove it to continue". It blocks the "Pay" button until the customer removes the line.
- **Coupon:** a "Gift code" input plus an "Add" button. If valid, it appears as a negative line in the summary.
- **PagosStrip strip** (static 1×6) at the end, reminding the customer that MercadoPago + BRE-B + Visa + MC + Amex + Nequi are accepted.
- **"Continue to payment" CTA button** leads to `/checkout`.

3.4 /checkout

Component: `src/pages/Checkout.jsx`. This is the page that implements everything described in §5 (Payments Section). The only thing documented here is that it is the canonical payment URL. If the user arrives with no products in the cart, it redirects to `/catalogo` with a message.

3.5 /confirmacion/:idPedido

Component: `src/pages/Confirmacion.jsx`. This is where the customer lands:

- **After an approved MercadoPago payment:** through the automatic return with `?status=approved&payment_id=...`. The component confirms with the `mp-webhook` Edge Function that the transaction is approved and shows "Payment Confirmed!".
- **After uploading a BRE-B proof of transfer:** it shows "Proof received successfully!" with an estimated validation countdown (48 h).
- **After a failure / expiry:** it shows the `<StockAgotadoModal />` modal if the cause was overselling, or the "Payment not processed" screen with retry options.

It implements the listener for the custom `urg:fallido-exp` event that the webhook fires when it runs `cancelarHuerfanosPostConsumo()`, opening the "Someone was faster!" modal.

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

Component: `src/pages/FAQ.jsx`. V1.1 named the route `/faq` only; in v24.18.16 `/preguntas-frecuentes` was promoted to canonical (better SEO for LATAM) and `/faq` was left as a `<Navigate replace />`.

It renders the FAQ Hub categorized by the **"FAQ · URG Technology"** context (filtered by context in `faqHub.js → getFaqItems({ contexto: 'tienda' })`). Questions from the "Diseño Web Pro" context are NO LONGER shown here, since that part lives on damorus.com after the v24.18.26 migration.

UX features:

- **Category filter:** horizontal chips with scroll on mobile, a static row on desktop.
- **Expandable accordions** with a height animation (Framer Motion `layout`). Only one open at a time (controlled accordion).
- **Internal search box** that filters by title and description.
- **Auto-expand on landing:** if the user arrives with `?q=garantia`, the system finds the item, expands it automatically and scrolls to it.

3.7 /compramos-tu-pc

Component: `src/pages/CompramosTuPC.jsx`. It is one of the most sophisticated flows on the site. The complete 9-step form lives in `src/components/compramos/CompramosTuPCForm.jsx`.

3.7.1 Page header

After multiple iterations (#157, #158, #161, #196) the page ended up with:

- **Compact centered hero** with the title "COMPRAMOS TU PC" as a hyperlink to the physical store (Google Maps).
- **4 advantage bubbles** above the form: Express Valuation · Zero Stress · No Obligation · 5% Bonus.
- **Form block** opened at Step 01 by default (the user does not have to click "Start"; the first step is already visible).
- **Right sidebar** (desktop $\geq$ lg) with a vertical progress bar and a summary view.

3.7.2 Form steps

#	Question	Component
1	The Great Selector: Laptop vs Tower	<code>PasoTipoEquipo.jsx</code>
1.1	Payment preference: Cash vs URG Credit	<code>PasoPreferenciaPago.jsx</code>
2	Unit condition: New / Used / Damaged	<code>PasoEstadoUnidad.jsx</code>
3	Machine DNA: Brand and model + CPU + RAM	<code>PasoADN.jsx</code>
4	The Data Warehouse: SSD vs HDD + capacity	<code>PasoAlmacen.jsx</code>
5	Laptops only: Screen + battery + charger	<code>PasoSensores.jsx</code>
6	General condition (appearance): Grade A/B/C/D	<code>PasoEstadoFisico.jsx</code>
7	Briefing: Reason for selling + next use	<code>PasoBriefing.jsx</code>
8	Contact: Name + WhatsApp + email (optional)	<code>PasoContacto.jsx</code>
9	Final screen: Mission Accomplished	<code>PasoFinal.jsx</code>

Every step applies the didactic rules from V1.1: visual chips, personality labels, help tooltips ("Do not know the model? Look at the label under your PC, or press Win+R and type 'msinfo32'"), automatic advance after a selection, and so on.

3.7.3 Zero Stress (photo upload)

A floating "I would rather send photos" button is visible during steps 3, 4, 5 and 6. When it is pressed:

- It fades out the technical questions.
- It opens the Upload Panel with 3 guide frames (internals / front / ID card, depending on Laptop or Tower).
- **Client-side compression:** using the `Canvas API`, every uploaded image is resized to a maximum of 1920 px on its long side and converted to WebP at quality 0.85. The `src/lib/imageOptimizer.js` module also strips EXIF data (geolocation, camera model) for privacy: `fileForensics.js`.
- **Jump to Step 3 disabled, immediate jump to Step 7** (task #162: the "Zero Stress mode active" banner was removed because it created friction).

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

A redesigned closing screen with a "technological pipeline" animation: the customer's data "travels" through channels toward the URG processing center while a giant neon green check mark appears. Text:

Mission accomplished, Commander! Your data has been sent to the command center. We are analyzing your [Brand/Model] to give you the best offer.

List of next steps with icons:

1. **Analysis:** within 2 business days you will receive a WhatsApp message with your estimated price.
2. **Inspection:** if you like the price, we schedule a physical review to set the final value.
3. **Payment + Bonus:** you receive your money and we activate your 5% coupon.

Additional final CTA: [**ACCESS YOUR TRACKING PORTAL**] with a unique PIN generated when the form is submitted (the PIN system was introduced in v19, see §3.8).

3.7.5 Sticky progress bar

A thin top band (32 px tall) with a black background and a neon purple bottom border. Dynamic text:

- Initial state: "BONUS ACTIVE: +5% extra Credit when you sell your machine at URG."
- At 75%: "Almost there! Your 5% Bonus is waiting at the end."
- The progress bar is a neon green radar that fills as the user advances.

3.7.6 Persistence and resilience

- **Autosave:** every change is saved to `localStorage['urg.compramosPc.draft']`. If the user leaves and comes back another day, they resume exactly where they left off (the system asks them "Continue where you left off?").
- **Sync with Supabase** when the last screen closes: it writes the lead into the `buy_requests` table through `lib/buyRequests.js → addLead()`.
- **Admin notification:** it fires Email A.1 (Technical Notification) and A.2 (Received Confirmation to the customer) described in §6.6 Operations Matrix.

3.8 /rastrear-pedido

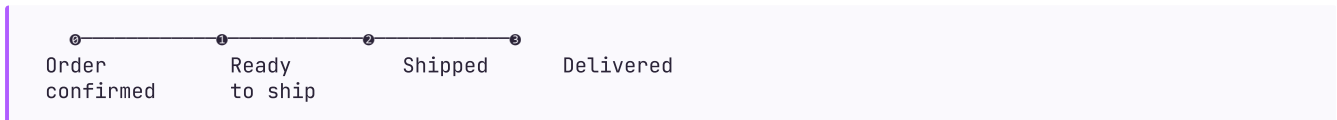
Component: `src/pages/RastrearPedido.jsx`. Design consolidated after v19 with the PIN:

- **Input:** "Enter your batch / order number (e.g. #2603-104A)" plus a second "PIN" input (4 digits).

- The PIN is generated when the order is confirmed (in `lib/orders.js` → `createOrder()`) and sent to the customer by email as well as shown on the confirmation page. The admin sees it in the Pending panel.
- **Auto-purge:** PINs are deleted from the `orden_pins` table 5 days after delivery so sensitive data does not pile up.

Timeline

Once the customer enters a valid code plus PIN, connected circular icons are shown:



- **Order confirmed** (Step 0: always active if the batch exists).
- **Ready to ship:** "We are preparing your machine for shipment."
- **Shipped:** "Your machine is already in the courier's hands!" It unlocks a block with the courier, the tracking number and a "TRACK WITH [NAME]" button.
- **Delivered:** "The machine has reached its destination. Thank you for trusting URG." A green "RATE OUR SERVICE" CTA appears, pointing to the configurable Google Maps link.

Delay banner

Below the timeline, a subtle banner with a dark gray background:

"Is your order taking longer than expected?" At URG we monitor every shipment, but if the courier is running late or you have not received your machine within the estimated time, we are ready to give you a solution.

[CONTACT SUPPORT]

The CTA opens WhatsApp with a preloaded message: *"Hi URG, my order with batch #[Number] is delayed, can you help me?"*.

3.9 /privacidad

Component: `src/pages/Privacidad.jsx` . Stable version of the Privacy Policy document under Law 1581 of 2012 (Habeas Data, Colombia).

- **Layout:** a clean page with a side menu (TOC) that lets the reader jump between collapsible sections.
- **Content editable from the Master Panel** (a CMS by section: each section has an editable icon/emoji, title and rich-text body).
- Default sections:
 1. What we know about you (personal data)
 2. How we improve your experience (cookies and analytics)
 3. Your secure payments (MercadoPago, BRE-B, encryption)
 4. Communications and support
 5. Your rights (Habeas Data): Access, Update, Rectify, Delete

The customer can request [Total Deletion] of their data by writing to `[store_email]` or, if a session exists, from a customer portal that points to the Privacy Module (§6.5.1: Right to Be Forgotten). The

deletion is a **surgical anonymization**, not a physical DELETE: amounts, dates and products remain in the financial history under the label "Cliente Anonimizado (Solicitud de Privacidad)".

3.10 /compra-prottegida

Component: `src/pages/CompraProtegida.jsx`. The page holding the support promise:

1. **Armored payments and total transparency** (MercadoPago + BRE-B + Law 1581 + we do not store cards).
2. **21-day delivery commitment or refund** (if the courier does not deliver within 21 calendar days, full refund).
3. **Technical fidelity** ("What you see is what you get"): cosmetic condition + specs + URG Certification (every port, key and pixel tested).
4. **Return policy, Plan 5/15:**
 - 5 business days after receipt → Right of withdrawal (Law 1480, Art. 47), the customer covers shipping.
 - 15 business days → Hardware technical failure: URG covers everything, picks up the unit and delivers a superior machine or refunds 100%.
5. **After-sales support:**
 - Hardware warranty of 3/6/12/24/48 months depending on the product.
 - Free technical support for 1 year.

Legal notes (fine print):

- Warranty exclusions (impacts, moisture, external voltage, broken seals).
- Software is the user's responsibility after delivery.
- Refunds through the same payment method within 8-15 business days.

A "card" design with modern line icons, a clean white background and purple accents. Floating WhatsApp button with the message "Questions about your warranty? Chat with an expert".

3.11 /diseno-web → redirect to damorus.com

As mentioned in §2.7. Component `<ExternalRedirect to="https://www.damorus.com" />`.

3.12 Catalog shortcuts (/ofertas, /portatiles, /escritorio, /perifericos)

Components in `src/pages/PortatilesEscritorio.jsx`. Each one is a wrapper that sets pre-applied filters over `<Catalogo />` and modifies the breadcrumb and the `<title>`. For example, `<Portatiles />` renders:

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

3.13 /mas-vendidos

Component: `src/pages/MasVendidosPage.jsx`. It shows all 12 positions of the ranking (4 categories × 3 positions = 12 products). Unlike the Home block, which shows only the TOP 1, this page shows the 3 positions of each category in vertically stacked sections. Every product carries its position badge (#1 with a glowing gold border, #2 silver, #3 bronze: task #251).

If a category has fewer than 3 ranked products, the empty slots are replaced by a gray placeholder reading "Position available: a new machine here soon". This reinforces the sense of editorial curation.

4. LISTING SYSTEM

Every product is rendered in one of 3 formats depending on the context. The component that governs the format is always **the same** (`ProductCard.jsx` with the prop `variante: "1" | "2" | "3"`), which guarantees that any brand or schema change applies to all three at once.

4.1 LISTING 1 • Catalog Standard

Component: `src/components/cards/Ficha1.jsx` . It is the basic unit of the **High-Density Grid**.

Usage

- Home: Hero (not applicable), Deal of the Day (horizontal Listing 1), Top 1 of each category in Best Sellers.
- Catalog: Listing 1 is NOT used in the general `/catalogo` (Listing 2 rules there). Listing 1 is used on the dedicated tabs that show few categories (`/ofertas` , `/mas-vendidos`).
- Peripherals: ALWAYS Listing 1 (decision §4.4).

Orientation

Mostly vertical. The **only exception** is the Deal of the Day on desktop, where it is rendered horizontally per a specific instruction from the founder (first image taking 58%, info panel 42%).

Components in order

1. **Main product image**: `aspect-ratio: 4/3` (a rollback to the original ratio after task #37, since the change to 1:1 lost context). Lazy loading with a skeleton.
2. **Condition badge** (upper right corner): a capsule with distinct colors for New/Like New/Excellent/Good (always visible, even in the placeholder, task #12).
3. **Customizable "Free shipping in Bogotá" badge** (upper left corner).
4. **Marketing name** in bold, max 2 lines (no descriptor: task #8 removed it because it saturated the card).
5. **Brand + model** on a single line, `text-zinc-500` (no "Brand:" prefix).
6. **Original price** struck through in faint gray.
7. **Deal price** in URG purple, large font.
8. **OFF badge** lavender capsule with `%OFF` .
9. **CTA "BUY NOW!"** with a sweeping glow effect every 3 s. Clicking goes to Listing 3 (not straight to checkout, decision in task #239).
0. **Clicking anywhere on the card** (not on the CTA): also goes to Listing 3.

SOLD state

When `stockDisponible === 0` :

- Translucent red overlay on the image.
- Centered "VENDIDO" stamp in white (not diagonal, task v24.18.22).
- CTA replaced by a "Notify me when it is back" button (the `avisame.js` system).

4.2 LISTING 2 · Advanced Search

Component: `src/components/cards/Ficha2.jsx`. A compact horizontal format that appears on `/catalogo`, `/portatiles` and `/escritorio`.

Layout

It is a vertical rectangle with:

- **Row 1 · Horizontal image** taking about 1/3 to 1/4 of the card height, with indicator dots in the style of Mercado Libre / Amazon (1 to 9 images).
- **Upper right corner · "Condition" badge** in eye-catching purple (New / Like New / Excellent / Good).
- **Row 2 · Marketing name** in bold.
- **Row 3 · Brand + model** on a single line, gray tone.
- **Row 4 · List of technical specs** with prefixed labels:
 - Memory: 32GB
 - Graphics: 30-Core GPU
 - Storage: 1TB SSD
 - Warranty: 12 months URG
- **Row 5 · Purple divider line.**
- **Row 6 · Price** (with strikethrough and discount where applicable) in the left corner plus a "Details" button on the right.

Category filter (top chips)

On the Listing 2 tabs there is a minimalist "Filter by category" dropdown menu at the top. It is documented in detail in §3.1 (Search Ecosystem filters).

4.3 LISTING 3 · Product Terminal (Final Conversion View)

Component: `src/pages/ProductoDetalle.jsx`. The most complete page of the public site.

4.3.1 Responsive layout

Mobile (< md)

A single column with an optimized order:

1. **Breadcrumbs + "Share" hyperlink** and "See more products from brand X" at the top.
2. **Wide image gallery** (aspect 4/3) with indicator dots. The Share button (a square with an arrow) is visible.
3. **Brand** as an inline link to the brand filter (task #32: it used to be a separate block and was compacted).
4. **Condition** highlighted in green plus the "#X in Y" ranking where applicable (same neon green).
5. **Marketing name** in bold.
6. **If it is the Deal of the Day**: a banner below the name.
7. **Price** (strikethrough + final + %OFF where applicable).
8. **Colors** selectable with a square chip per color.
9. **Size and weight** in chips (order Size → Weight after task #3).
0. **Technical specs** Memory / Graphics / Storage / Warranty.

1. **CTA buttons** "Add to cart" + "Buy now" (the "Buy now" button is bright green, the only element on the page with that shade).
2. **Warranty months medal** plus a green arrow reading "Protected purchase" (hyperlink to /compra-*protegida*).
3. **Dynamic delivery promise** (48 h algorithm) "FREE delivery on Tuesday 24 June in Bogotá".
4. **About this product** (manual description).
5. **Similar product at a competitive price** (1 card).
6. **Dynamic additional specifications** (Dimensions, Materials, Style, etc.: only the ones the admin filled in).
7. **1×6 payment logo strip**.
8. **"Other similar items" blocks** (Best sellers in the category, Same brand, Cheaper options).

Desktop ($\geq$ md)

Three columns on a 12-column grid:

- **Cols 1-7** (Multimedia): main gallery + thumbnail grid + share button.
- **Cols 7-10** (Identity): "See more products from brand X", condition/ranking, name, price, color, size/weight, brand, model, memory, graphics, storage, warranty. Below that, "About this product" and "Similar product at a competitive price".
- **Cols 10-12** (Purchase): price table + warranty + FREE 1 year of technical support (toggled in admin) + availability status + Add to cart / Buy now CTAs + warranty medal + Protected purchase + delivery promise.

4.3.2 Digital Identity and Routing System (Unique URLs)

- **Slug** generated automatically when the listing is created: kebab-case, no accents, no special characters.
- **Immutable** once published (so links shared on WhatsApp do not break). The admin can only change it manually under an explicit warning.
- **Automatic numeric suffix** on collision.
- **history.pushState** when Listing 3 is opened from the catalog (the URL changes without a reload).
- **popstate listener** to close Listing 3 when "Back" is pressed.

4.3.3 Smart Share Button

A square with an arrow (Material Design icon). On click:

- If the browser supports the Web Share API → `navigator.share({ title, text, url })`.
- Otherwise → it copies the URL to the clipboard with `navigator.clipboard.writeText()` and shows the toast "Link copied successfully! You can send it now".
- Meta tags are configured so that pasting the link into WhatsApp shows the product thumbnail, price and name.

4.3.4 Sticky CTA reminder (mobile)

When the user scrolls and the main purchase buttons leave the viewport, a thin fixed bar appears at the bottom with: machine name (truncated) + price + a single "BUY" button. It is visible only from the "About this product" section downward (task #33: it used to appear always and was distracting).

It avoids interfering with the floating WhatsApp button thanks to a calibrated `z-index` and a right margin (`pr-20` on mobile) that reserves space.

4.3.5 Premium "SpecTooltip"

In Listing 3 every technical spec (processor, RAM, and so on) has a small (i) icon that on hover opens a command-terminal-style tooltip (`background: #0a0a0f; font-family: 'JetBrains Mono'; border: 1px solid var(--urg-purple)`). It explains in plain language what that spec means ("Processor cores: the more there are, the better for heavy multitasking").

On mobile the tooltip is triggered by tap, not hover: task #269. A visual toggle sits in a small tab below the specs.

4.3.6 Dynamic delivery promise (Smart Delivery Timer)

Logic in `src/lib/format.js → calcularFechaEntrega()` :

- Adds 2 business days to today.
- Friday → Monday rule (skipping Sunday).
- Saturday/Sunday → Tuesday rule.
- Ignores Sundays in the count.
- Output: *"FREE delivery on Monday 24 June in Bogotá"*.

4.3.7 Advanced gallery

- **Shift-drag** to reorder images (visible only in admin: task #259).
- **Magnifier lens** on desktop hover (280×280 px, 500% zoom: task #239).
- **Mobile** horizontal swipe with snap.
- **Videos** are lazy loaded, muted by default but with a visible sound icon so as not to startle anyone in a public place.

4.4 LISTING ADAPTATION FOR PERIPHERALS

Logic in `lib/categoriasPerifericos.js`. When the admin marks a category as "Peripheral" (instead of "Computer"), the Listing Creator automatically hides Processor/RAM/GPU and instead enables 4 free-text fields called **"Technical Highlights"** (for example DPI, Switch Type, Connectivity, Battery Life).

- **On the Home and in the Catalog:** peripherals ALWAYS use Listing 1 (vertical, clean for fast scanning).
- **In the /perifericos bubble:** Listing 1 (unlike computers, which use Listing 2 in their bubbles: this was decided because peripherals do not have as many comparable specs as a PC).
- **In a peripheral's Listing 3:**
 - Column 2 shows 4 Technical Highlights instead of the computer specs block.
 - "Similar product at a competitive price" looks for **another peripheral in the same sub-category** (if the user is looking at a mouse, it suggests another mouse, never a keyboard).

The documentation of the **Payments Section** follows in section 5, with full coverage of the server-side atomic reservation model, which is the deepest change since V1.1.

5. PAYMENTS SECTION

The payments module went through a complete architectural redesign between v18.15 and v24.18.25. V1.1 described a client-centric flow where the website mutated stock locally when a purchase started. The reality after v24.18.25 is radically different: **the critical stock logic lives on the server** (Edge Functions + pg_cron + Postgres RPCs), and the client only writes intentions and reads results.

This section documents both the customer UX (identical on the surface to V1.1) and the data model plus the server-side atomic reservations that support it.

5.1 Billing Data Form

Screen: `src/pages/Checkout.jsx` step 1. *"We ask only for the information that is essential to complete and ship the purchase."*

5.1.1 Personal fields (always required)

1. **First names:** `maxLength: 60` (limit reduced by 30% after v24.18.18; previously there was no limit).
2. **Last names:** `maxLength: 60`.
3. **ID card or passport:** `pattern: ^[0-9A-Z]{6,12}$`, `maxLength: 15`.
4. **Phone / mobile:** `pattern: ^\+?[0-9\s()-]{7,15}$`, `maxLength: 18`.
5. **Email:** `type: email`, sanitized through a regex in `lib/validators.js` → `isEmail()`.

"Invoice to a company" option

A checkbox that expands an extra section:

- **Legal name** (text).
- **Company name** (text).
- **Tax ID** (`pattern: ^[0-9.-]{9,15}$`).
- **RUT** (optional PDF upload, max 5 MB).

If the customer changes their mind, they can collapse the section with "Do not add", and the data stays in memory in case they reactivate it.

5.1.2 Shipping information

3 mutually exclusive options (large radio buttons with icons):

Option A • Pick up at the physical store (FREE)

"I will pick up my purchase at [store address]". No extra fields.

Option B • Ship within Bogotá (FREE)

Expands:

- **Address** (main field).
- **Additional address** (apartment, office): optional, gray.
- **Postal code:** optional.
- **Name of the person receiving:** pre-filled with the name from the previous step, editable.

Option C · Ship to another city in Colombia (configurable fee)

Expands:

- **Department + city:** cascading selector using `src/data/coloDeptos.js` (the official catalog of Colombian departments and municipalities). Component `<DeptoCiudadSelector />`.
- **Address** (text).
- **Additional address** (gray, optional).
- **Postal code** (optional).
- **Name of the person receiving** (editable).

Additional delivery notes

Optional textarea with the placeholder *"Special instructions for the courier or comments about your order (for example, deliver after 2:00 PM or call on arrival)"*. Max 280 characters. It feeds the `{{notas_cliente}}` variable in the admin emails.

5.1.3 Error correction (validation)

Implemented with native HTML5 plus `lib/validators.js`:

- Neon red border on an invalid field.
- Message underneath: *"This field is required for the invoice"* or the specific error.
- The "Pay" button is disabled until validation passes.
- **Scroll-to-error:** if the customer tries to pay with empty fields, `window.scrollTo({ top: errorFieldOffset - 100, behavior: 'smooth' })` takes them to the first missing one.

5.1.4 Gift coupons

A "Gift code" field plus an "Add" button. It applies a percentage or fixed discount according to the configuration in Block 4 of the Tax module (§6.10). If the code is invalid, a red message reads *"This code does not exist or has already expired"*. Non-cumulative (only one coupon per transaction).

5.1.5 Order summary

A dynamic block that updates in real time:

```
Product: MacBook Pro M3 (with 10% OFF applied)
Subtotal:      $ 2.521.008
VAT (19%):     $ 478.992
Shipping:      (shipping fee)
Gift coupon:    -$ 150.000
-----
TOTAL TO PAY:  $ 2.850.000 + shipping
```

When the product is **excluded from VAT** (price ≤ 50 UVT), the breakdown is replaced by:

```
Subtotal:      $ 2.400.000
VAT (0%):      $ 0 (Excluded, Art. 424 E.T.)
Shipping:      $ 0
-----
TOTAL TO PAY:  $ 2.400.000
```

5.1.6 Payment method

Two main buttons with logos:

[MercadoPago logo] Mercado Pago Instant payment by card or account	[BRE-B logo] BRE-B Transfer with proof of transfer
---	---

We accept: [Visa] [MC] [Amex] [Nequi]

And below it the mandatory checkbox:

☐ I accept the Privacy Policy and the Processing of my
Personal Data by URG TECHNOLOGY SAS

- The box is NEVER checked by default.
- While it is unchecked, the MercadoPago and BRE-B buttons are grayed out and disabled.
- Clicking "Privacy Policy" opens </privacidad> in a new tab.

5.2 Digital Order Confirmation (Order PDF)

After a successful payment, an automatic PDF named "**ORDEN_URG_{ID_pedido}.pdf**" is generated (for example, [ORDEN_URG_2603-104A.pdf](#)). Generator: [src/lib/pdfBuilder.js](#) using [pdf-lib](#).

PDF structure (4 blocks)

- **Block A · Header:**
 - URG logo ([URG_FOOTER_2026.svg](#) converted to an embedded PNG).
 - Title: [ORDEN DE PEDIDO / COMPROBANTE DE COMPRA](#).
 - Order ID: [#2603-104A](#).
 - Company data: tax ID, address, email and WhatsApp of the store (values omitted in the public version).
- **Block B · Customer:** data from the form (name, ID/tax ID, address, phone).
- **Block C · Detail:** a table with description, condition (Excellent/Good), quantity and unit value.
- **Block D · Financial and notes:**
 - Subtotal, VAT, total.
 - Legal note: *"This document is a purchase order and does not constitute a sales invoice. Your legal electronic invoice will be delivered physically together with your product".*

The PDF is attached automatically to the confirmation email sent by Resend (Email B.2). A copy remains available for permanent download from the admin Sold module ([Download Center](#)).

5.3 Naming and Traceability System (Order ID)

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

- [YY](#) = last 2 digits of the year (26 for 2026).
- [MM](#) = month (03 for March).
- [-](#) separator.
- [###](#) = 3-digit sequential counter (000-999).
- [X](#) = cycle letter (A → B → C and so on).

Scalability rules

- **Monthly reset:** when a new month starts, the counter goes back to `000`.
- **Letter jump:** when 999 is exceeded within the same month, it jumps from `999A` to `000B`.
- **Persistence:** the ID is generated **at the microsecond when the customer decides to pay** (not on confirmation). If the session is interrupted, the admin still keeps a record of the "Sale Attempt".
- **An order ID is never deleted:** it only changes state (COMPLETED vs CANCELLED).

Generation is implemented as a Postgres function (RPC `generar_id_pedido()`):

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

Final resolution logic

"COMPLETED"

- Trigger: the MercadoPago `approved` webhook, or the admin pressing `[CONFIRM PAYMENT]` on a BRE-B order.
- Impact: stock is subtracted definitively, the state moves to `POR_ENVIAR`, and the metadata is locked.

"CANCELLED"

- Trigger 1 (expiry): an `INICIADO` order passes 48 h without a successful report → the cron cleans it up.
- Trigger 2 (manual rejection): the admin presses `[REJECT PAYMENT]`.
- Impact: state `RECHAZADO` or `FALLIDO_EXP`, automatic restock, and the ID stays in the history with a gray/red mark.

5.4 Atomic Stock Reservation Model (CRITICAL · server-side)

This is the most **important** section of the document. V1.1 assumed a simple client-centric architecture where the front end read `producto.stock` and subtracted from it. Production reality revealed the following problems:

1. **Double selling:** two customers could pay simultaneously for the same unique product.
2. **Ghost stock:** customers who started a payment and abandoned it left reservations that were never released.
3. **Dependence on an open tab:** the first implementation required an admin to have the panel open in order to process the webhooks → if the admin was asleep, the site did not update stock.

The v24.18.25 redesign moved **all the critical logic to the server:**

5.4.1 Data model

Table `productos` (extract):

```
CREATE TABLE productos (
  id          uuid PRIMARY KEY DEFAULT gen_random_uuid(),
  slug        text UNIQUE NOT NULL,
  nombre_marketing text NOT NULL,
  -- ... other fields ...
  stock       int NOT NULL DEFAULT 0,      -- real physical inventory
  stock_reservado int NOT NULL DEFAULT 0,  -- units temporarily set aside
  reservas    jsonb NOT NULL DEFAULT '[]', -- detail of each active reservation
  CONSTRAINT stock_no_negativo CHECK (stock - stock_reservado ≥ 0)
);
```

The `reservas` field is an array of objects:

```
[
  {
    "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 available for sale is computed as:

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

The front end calls this in real time from `src/lib/orders.js → getStockDisponibleAhora(productId)`, which runs `SELECT stock, stock_reservado FROM productos WHERE id = $1`.

5.4.2 RPC function `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
  -- Pessimistic lock on the row
  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;
```

Guarantee: only one transaction can hold the lock, so two concurrent reservations can never both pass the check.

5.4.3 RPC function `consumir_stock`

Called **only from the server** (`mp-webhook` when MP confirms `approved`, or the admin endpoint when a BRE-B payment is approved):

```
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-archive if stock reaches 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 RPC function `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 BRE-B flow (with proof of transfer)

1. The customer confirms their data and chooses BRE-B.
2. The customer uploads the proof of transfer (upload field: max 10 MB, JPG/PNG/PDF). Component `<ComprobanteUploader />`.

3. The proof triggers `subirComprobanteBreB(order)`, which internally calls `reservar_stock_atómico` with a TTL of 60 min and the state `PENDIENTE_COMPROBANTE`. If the reservation fails (stock sold out), the customer gets the "Someone was faster!" modal and is offered alternatives.
4. The order stays in the `RESERVADO_HV` state (Reserved Until Verification) in the `orders` table.
5. The admin receives email B.1 (Pending Payment Alert) and sees the order in the Pending module (with an orange badge).
6. The admin presses `[CONFIRM PAYMENT]` → a server-side endpoint runs `consumir_stock`. Stock is subtracted permanently. Email B.2 goes to the customer with the PDF.
7. If the admin presses `[REJECT PAYMENT]` → it runs `liberar_reserva`. The stock returns to the catalog. Email B.4 goes to the customer with the reason.

5.4.6 MercadoPago flow

1. The customer confirms their data and chooses MercadoPago.
2. The customer is redirected to the MP gateway (through the `mp-create-preference` Edge Function).
3. `redirectToCheckout()` **NO LONGER reserves stock locally**. That decision was reversed in v24.18.23: only the webhook may touch stock.
4. The customer pays in MP. MP fires a webhook to the `mp-webhook` Edge Function with `event: payment.updated`.
5. **The webhook runs two atomic steps:**
 - `reservar_stock_atómico(producto_id, order_id, 'MP', 0)`: TTL 0 because it is consumed immediately.
 - `consumir_stock(producto_id, order_id)`: if the previous step was OK.
6. The customer returns to `/confirmacion/{id}?status=approved` and sees "Payment Confirmed!".

5.4.7 Cleanup cron (NEW in v24.18.25)

There is a Postgres cron job that calls the `cleanup-stock-fantasmas` Edge Function every 2 minutes. That function:

1. Looks for every reservation with `expira_at < now()` and the state `PENDIENTE_COMPROBANTE` (BRE-B with no proof within 8 min).
2. For each one, it runs `liberar_reserva` and marks the associated order as `FALLIDO_EXP`.
3. It also looks for MP orders in the `PASARELA_ABIERTA` state that are more than 1 h old and releases them.
4. If a release frees the last available stock unit for a product, **it fires a Realtime event** to the connected clients: if someone is on `/checkout` or `/productos/X` looking at that product and another one sold, they are shown the "Someone was faster!" modal.

Schedule:

```

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

```

(The `app.service_role_key` secret is loaded through `ALTER DATABASE postgres SET app.service_role_key = '...'.`)

5.5 Cross Cancellation BRE-B ↔ MP

Before v24.18.23, it was possible for two customers to both end up buying the same unique product: one through BRE-B (with the proof already uploaded) and another through MP (with the webhook approved). Stock was 1 → both initially went through, creating operational debt.

The solution is the `cancelarHuerfanosPostConsumo()` function:

```

// In mp-webhook/index.ts after consumir_stock
async function cancelarHuerfanos(supabase, productoId, orderIdGanador) {
  // List every remaining active reservation for that product
  const { data: producto } = await supabase
    .from('productos').select('reservas, stock')
    .eq('id', productoId).single();

  if (producto.stock ≤ 0) {
    // Stock sold out: any pending reservation is an orphan
    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
        });
        // Mark the order as FALLIDO_EXP and notify the customer through Realtime
        await supabase.from('orders')
          .update({ estado: 'FALLIDO_EXP', motivo_cancelacion: 'OVERSOLD' })
          .eq('id_pedido', reserva.order_id);
        // Send an apology email through Resend
        await dispararEmailDisculpa(reserva.order_id);
      }
    }
  }
}

```

The losing customer (who was on `/checkout` or `/confirmacion` waiting) receives:

1. A Realtime event → the front end fires `<StockAgotadoModal />`.
2. An apology email through Resend with the CTA "Show me other options" → the catalog filtered by the lost category plus a compensatory 5% coupon.

Modal text:

Someone was faster!

Someone else completed the payment process just before you did. At URG, deals of this caliber fly at light speed.

Do not get left behind! We have machines with similar power ready for you.

[SHOW ME OTHER OPTIONS] [Close]

(Style: title and main text in red `var(--urg-red)` per an explicit instruction from the founder.)

5.6 BRE-B traffic-light timer (draggable on mobile)

For the BRE-B flow, the customer has 8 minutes to upload the proof of transfer. The traditional timer appears as a sticky top bar, but on mobile a **floating draggable version** was introduced (v24.18.18):

- A round 72 × 72 px bubble with the countdown in the center.
- It changes color like a traffic light:
 - Green when more than 5 min remain.
 - Yellow between 2 and 5 min.
 - Red under 2 min.
- The user can position it (touch-drag anywhere on the screen, the position persists in localStorage).
- Tapping it opens the full BRE-B screen if the user got distracted.

5.7 "Your time expired" screen (BRE-B)

If the counter reaches 00:00, the interface locks and shows:

```
Your time to confirm the payment has expired.  
  
[ RESTART BRE-B PAYMENT ]      (highlighted primary button)  
  
[ Pay with Mercado Pago ]    [ Talk to an Advisor ]  
                             (secondary buttons)  
  
Back to the Catalog (discreet link at the bottom)
```

- **Restart BRE-B payment** generates a new order ID (the previous one is left as `CANCELADO`) and a new 8-minute cycle.
- **Pay with MercadoPago** recovers the cart and redirects to the MP gateway.
- **Talk to an Advisor** opens WhatsApp with the message *"Hi, I ran out of time to pay for my order #[ID] and I need help completing my purchase"*.

5.8 Post-MP cart persistence (loss prevention)

An important decision: if the customer pays through MP and the payment fails (rejected), the cart **keeps the selected products** when they return to URG. This saves the customer from having to search for the machines again and raises the conversion rate on retries. The persistence is stored in `localStorage['urg.cart']` with a 24 h lifetime.

This closes the documentation of the Payments Section. Next, **section 6** documents the complete Admin Panel (with its 4 blocks: Daily Operation, Customer Management, Intelligence and Security, Configuration

and Structure) and every v24 addition, especially Sentinel (Operational Health) and the server-side architecture.

6. ADMIN PANEL (/[panel])

The administrative panel lives at `/[panel]` (login) and `/[panel]/*` (modules). It has its own Layout (`AdminLayout.jsx`), a permanently dark theme, and no Header, Footer or WhatsApp button from the public site.

Visual architecture: a fixed top bar (global status) + a left sidebar (App Launcher with 4 icons) + a central content area. Four thematic blocks: **I. Daily Operation · II. Customer Management · III. Intelligence and Security · IV. Configuration and Structure**. Access is role-based with granular View/Modify permissions per block (see §6.11).

BLOCK I · DAILY OPERATION

6.1 Login (/[panel])

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

Visual atmosphere

- **Dynamic background:** gradient from deep black (`#0B0B14`) to charcoal gray (`#1C1C28`). A moving particle system (a stardust effect generated with Canvas in `src/components/system/NeonAccent.jsx` → the `<ParticlesField />` component) drifts slowly, giving a sense of depth and advanced computing.
- **Login card:** glassmorphism effect with `background: rgba(20,20,30,0.55); backdrop-filter: blur(24px)`. A 1 px border in URG purple `#8A2BE2` with `box-shadow: 0 0 32px rgba(138,43,226,0.35)` that pulses every 4 s.
- **URG logo** at the top (the footer version in matte white) with a breathing animation (a subtle oscillating purple glow).

Input fields

- **Identity field** (email or username). Subtle "Profile" icon. On focus, the border transitions from gray to vibrant purple.
- **Smart password field:**
 - An "Eye" icon to toggle visibility.
 - A virtual LED indicator that lights up if Caps Lock is on (detected with `e.getModifierState('CapsLock')`).
- **CTA button:** "Access the System", solid purple background; on hover it emits a neon pulse toward the edges.
- **Recover password link** below the button (discreet style, light lavender color).

Invisible security shield

- **Cloudflare Turnstile** integrated (`<TurnstileWidget />`). Silent validation, no puzzles.
- **Progressive delay** (`lib/rateLimiter.js`):
 - 1st failure → 1 s wait.
 - 2nd failure → 2 s.
 - 3rd failure → 4 s.
 - 4th and beyond → 8 s (capped).
- **Honeypot:** decoy routes (the list is not published) registered as middleware → any access tags the IP as a "Potential Aggressor" in the Black Box.

"Ghost Mode" protocol (recovery)

- **Invisible security response:** after sending the email, the system always shows the same generic message: *"If the data matches a registered administrator, you will receive an access link in your inbox"*. It never confirms whether the email exists (anti-enumeration).
- **Magic link** through Supabase Auth with a **15 minute** expiry.
- **Forced reset:** when entering through recovery, it redirects to `/[panel]/resetear-clave` (Password Change): the middleware detects the `PASSWORD_RECOVERY` event and forces the flow.

MFA / TOTP protocol (post-login)

Implemented in `src/lib/mfa.js` using Supabase Auth MFA:

- **Activation:** from the admin's Profile (not mandatory on the first login, but mandatory after 7 days for a Super Admin).
- **TOTP sync:**
 1. The system generates a QR code with `qrcode.react`.
 2. The admin scans it with Google Authenticator / Authy.
 3. They enter a 6-digit code to validate.
- **Permanent effect:** every future login requires the 6-digit code through `supabase.auth.mfa.verify()`.

6.2 Immediate Control Center

The main screen after login. Component: `src/pages/admin/Dashboard.jsx`. Made up of the **Global Top Bar + Left Sidebar + Central Content**.

6.2.1 Global Status Top Bar

Component: `src/components/admin/AdminUI.jsx → TopBar`. Fixed at the top edge, present on ALL admin screens.

Left side (System Pulse)

- **Digital clock, Colombia time** in `HH:MM:SS` format (24 h). Updates every 1 s. Monospaced font `'JetBrains Mono'`.
- **Network telemetry (Ping):** a real-time gauge reading `Ping: 24ms`. LED indicator and color:
 - **Green:** online / latency under 100 ms.
 - **Yellow:** lag detected / high load (over 100 ms).
 - **Red:** offline / under attack (no response after 3 retries).

Implementation: every 5 s it sends a `HEAD` request to `/api/health` (an empty Edge Function endpoint). It measures `Date.now() - tStart`. There is a grace period of 3 retries before turning red.

Center (Notifications)

A bell icon with a dynamic red bubble. Clicking it opens a floating panel:

- "#X BRE-B payments to confirm" → clicking navigates to Pending filtered by the `RESERVADO_HV` state.
- "#X Orders ready for dispatch" → navigates to Pending filtered by `POR_ENVIAR`.
- "#X Leads for equipment purchase" → navigates to We Buy Equipment.
- "#X Critical Sentinel alerts" → navigates to Sentinel (NEW in v24.18.24).

Persistence: they only disappear when the object's state changes in the database. Realtime through Supabase channels: the red bubble updates without a refresh.

Right side (User + Kill Switch)

C.1 User bubble

A circular photo with a light ring (glow) per rank:

- **Gold:** Super Administrator (master user). Full access.
- **Purple:** Editor / Logistics (block I).
- **Gray:** Guest (read-only for blocks I and III).

Dropdown menu on click:

- Profile editing (name, photo, valid email, password change).
- Security: enabling/disabling TOTP MFA.
- Rank badge with the accessible blocks.
- Secure sign-out with cache clearing.

C.2 Kill Switch (panic and maintenance switch)

A protected toggle (it requires a lateral swipe, not a simple click). Exclusive to the Super Administrator and the master user.

Modes:

1. Scheduled maintenance:

- Configure a timer and a custom message.
- An elegant, non-intrusive banner across the whole site: "We will be back in [HH:MM:SS counter]".
- After the countdown, the site enters Offline Mode.

2. Panic "CLOSE NOW" (neon red button):

- Swipe confirmation.
- Immediate shutdown of the public site.
- The admin panel stays ONLINE for the authorized ranks.

Active maintenance phase:

- Active MP/BRE-B transactions are paused or cancelled.
- Access to the Home, catalog, bubbles and payments is blocked.
- The site enters Total Offline Mode with `<MaintenanceMode />` (a component with a black background + a breathing logo + a base message + a custom message + a direct WhatsApp button).

Back to ON:

- `MaintenanceBanner.jsx` runs a verification script every 10 s on the public client; when it detects the change to ON, it reloads the page automatically (the user does not need to press F5).
- The server sends `Cache-Control: no-store` while maintenance is active so the browser does not cache the offline screen.

6.2.2 Left Sidebar (App Launcher)

Component: `src/components/admin/AdminSidebar.jsx`. Fixed on the left side in ALL sections.

Collapsed state (default)

Only 4 neon icons plus a magnifier:

1. **Box icon:** Block I (Daily Operation).

2. **Users icon:** Block II (Customer Management).
3. **Chart icon:** Block III (Intelligence and Security).
4. **Gear icon:** Block IV (Configuration and Structure).
5. **Magnifier icon (Ctrl+K):** Quick Command Search at the base of the sidebar.

Neon notification dots appear over the icons when there are pending tasks in that block.

Expanded state (hover)

A glassmorphism panel slides out showing:

- The block name in the header.
- A list of "Apps" (sub-modules) with clear titles:
 - **Block I:** Dashboard · Listing Creator · Active Catalog · Pending · Sold · Returned · We Buy Equipment · Archived · Sentinel.
 - **Block II:** URG Community · Messaging Center · FAQ Hub.
 - **Block III:** BI / Statistics · Profit Manager · Report Builder · Black Box.
 - **Block IV:** Tax · Admins · Master Panel · Home & Footer.

Quick Search (Mac Spotlight modal) · task #261

The `Ctrl+K` shortcut (or `⌘K` on Mac) opens a centered Mac Spotlight-style modal:

- A large input at the top with the placeholder *"Where are you going, commander?"*.
- A list of results below, grouped by category (Products, Orders, Administrators, Actions).
- Fuzzy search in `lib/adminSearchIndex.js`, which indexes all accessible metadata.
- Results support keyboard navigation (↑/↓/Enter).

Breadcrumbs

A dynamic path line at the bottom of the global top bar (not of the sidebar): `Operation > Pending > Details`. Every level is a link for a fast return.

6.2.3 Main screen · "Immediate Control Center"

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

A. Activity and Traffic Monitor

- **Live traffic:** a large counter of active users in real time (a query to `bi_live_sessions` every 10 s).
- **Movement history (log audit):**
 - Shows the last 5 actions (for example: "Sale #102 approved", "Price updated on Listing #45").
 - The [Expand] button opens up to 30 movements with internal scrolling.
 - The full history is persisted in Block III (Black Box) for audits.

B. Critical Communication Module

It controls the main message shown to the customer. Two quick-edit cards:

1. Global banner (announcement bar): appears at the very top of the store (above the logo):

- Activation toggle (OFF by default).
- Inputs for text and a timer (visible countdown or not).
- Color picker for background and font.

2. Hero & Copy Editor:

- Fields for the main title and the shipping sub-text.
- A rescue system with the "Brand Standard Text" below.
- A field for the Hero banner image with a suggested size plus a preview simulator.
- An [Update Home] button for instant synchronization.

C. Dynamic Showcase Module

Component: `src/pages/admin/HomeFooter.jsx` . A vertical storyboard with block cards:

- Every card is a Home block (Hero, CTA Bubbles, Deal of the Day, Best Sellers, We Buy Your PC, Web Design Pro, Our Store, Peripherals, SEO Directory).
- **Aesthetics:** glassmorphism with a reactive border (neon purple ON / matte gray OFF).
- **Drag handle (6 dots):** reorders sections instantly with `react-dnd` or an equivalent.
- **[i] button:** explains what each section does and which listing format it uses.

Organization in two natures:

- **A. Fixed structure blocks:** Hero, CTA, Deal of the Day, Best Sellers, We Buy Your PC, Web Design Pro (banner), Our Store, Peripherals (can be disabled), SEO Directory.
- **B. Dynamic injection blocks:** categories created in Module 3 of the Creator (Mice, Gaming Chairs, and so on). By default they are a "Guest" (inside Peripherals); they can be promoted to an Independent Block.

CTA and data manager (side panel per block):

- Editable button text.
- Link destination (Internal Bubble / WhatsApp / external URL).
- Limit of items shown (1, 2, 3, 4, 8, 12).
- Listing style (Type 1 Vertical / Type 2 Compact).
- Highlight badge (NEW / DEAL).

Telemetry and inventory intelligence:

- Vital counter " 📦 14 items in stock".
- An empty alert with a yellow triangle if there are 0 products.
- An SEO health indicator based on completeness.

D. Aura and Neon Module

A visual identity control system with animated borders:

1. Quick actions:

- Master switch (Energy Mode vs Minimalist Mode, default OFF).
- Preset selector:
 - URG DNA (solid purple #8A2BE2).
 - CRAZY URG (dual purple/neon green).
 - Cyber-Black (cold white and gray).
 - National Pride (tricolor with blur).
 - Deal Alert (intense red pulse).

2. Advanced telemetry panel:

- Flow Speed slider (Static → Plasma).
- Glow Intensity slider.

- Toggles for the Lava Lamp effect and Scroll Acceleration.

3. Thickness configuration:

- Discreet Mode (3 px): the professional default.
- Impact Mode (6 px): for launches.

E. Live Preview Mode

A frame at the end of the page that simulates the Home in real time:

- It renders changes automatically when the Hero, Aura and so on are modified.
- A PC / Mobile switch to preview both formats.
- It stays in draft state until "Update Home" is pressed.

F. Master Publish

- A [SAVE CHANGES] button with a "Syncing..." animation and a success check mark.
- A confirmation modal with a green YES / red NO.
- A Pending Changes bar warns the admin when trying to leave without saving.

6.3 URG Listing Creator

Component: `src/pages/admin/CreadorFichas.jsx`. It is the most important tool in Block I. After the deep refactor (task #252) it has sticky tabs with a real preview (not a placeholder).

Module 1 · Listing Builder (8 logical blocks)

Block 1 · Inventory control (required):

- Available quantity (numeric field).
- If > 0 → the site shows a green "Available" plus the Add/Buy CTAs.
- If $= 0$ → a red "Sold out" plus a WhatsApp button and 3 similar alternatives.

Block 2 · Identity and condition (required):

- Marketing name (max 60 characters recommended).
- Machine condition: New / Like New / Excellent / Good (a visual selector with colored capsules).
- Category (Gaming / Home-Study / Creators / Business).
- Manual ranking #1, #2, #3 (a visual ranking selector: task #246, which supports N/A or 0 for the catalog background). If the admin does not assign a number, the product goes to the general catalog without competing for the Home podium.

Block 3 · Vital specifications (required):

- Brand, model (text).
- RAM (text, for example 16GB DDR5).
- Graphics (text, for example NVIDIA RTX 4060).
- Storage (text, for example 512GB SSD NVMe).
- Warranty (fixed select: 3, 6, 12, 24, 48 months).
- Technical Service promotion (ON/OFF toggle, default ON: "FREE for 1 year").

Block 4 · Visual and physical configuration (required):

- Color selector: name + hex code (it generates the swatch automatically).
- Size (Small / Medium / Large / Extra Large).

- Weight (Featherweight / Light / Moderate / Heavy).
- "About this product" description (free textarea).

Block 5 · Additional details management (optional, dynamic):

- The [+ Add New Descriptor] button: creates a new field (Liquid Cooling, Chassis Material, and so on).
- System memory: the descriptors created are saved and appear as suggestions in future listings.
- Preloaded suggestions: Dimensions, Specifications, Materials, Style, Extras.
- A numeric "Order of Appearance (TOP)" selector per descriptor.

Block 6 · Smart gallery:

- A drag & drop zone for up to 9 photos and/or videos.
- Reorderable by dragging (the first one is the main thumbnail).
- **Integrated cropper** with a 2-column layout (task #259):
 - Left panel: the original image with pan/zoom controls.
 - Right panel: a 2×5 thumbnail grid with a preview of each crop.
- **Shift-drag** to move the image in a natural pan.
- **Automatic optimization:** WebP, max 1920 px, EXIF sanitization (privacy), compression at quality 0.85.
- **Specific framing for Listing #1 (Deal of the Day):** a 16:9 system with a dedicated cropper (tasks #255 and #257).
- **Inline smart gallery** that shows the photo plus live previews of every listing format in which it will appear (task #254).

Block 7 · Price and VAT calculator:

- List price (numeric).
- Discount selector: toggle between Percentage (%) and Fixed amount (\$).
- A dynamic discount system (task #241) with a preview of the final price plus the savings.
- **Calculation logic (URG Golden Rule):**
 - If the final price is ≤ \$2,618,700 (50 UVT) → "Excluded from VAT (Art. 424)".
 - If the final price is > \$2,618,700 → "Taxed at 19%" (Subtotal = Total / 1.19, VAT = 19%).
- **UVT traffic light:** a green circle (excluded) or a blue one (taxed) next to the price.
- **DIAN connection** through an internal proxy (informational) to validate the current rate.

Block 8 · Publication and positioning:

- **Preview:** a modal with the exact view of Listings 1, 2 and 3, using the **real components** (task #252; it used to be a placeholder).
- **"Deal of the Day" turn system:**
 - Position selector (1, 2, 3 and so on) or N/A.
 - Unit counter.
 - The sold-out state auto-activates the "Domino Effect" (an automatic promotion when a position frees up).
- **Multi-category logic:** a product can belong to up to 2 categories; it can only be Top 1 in the main one.
- **Final action:**
 - The **Publish** button (enabled only once every required field is complete). Confirmation with a purple banner reading "Product successfully linked to the Digital Store".
 - The **Save as Draft** button.

- **Validation and progress:** the Publish button is disabled and highlights missing required fields in red. A progress bar shows how close the listing is (for example, 60%).

Module 2 · Draft Management

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

- **Autosave** every 30 s when there are changes.
- **Status indicator:** Manual vs Automatic.
- **Management table:**
 - Temporary identifier (TEMP-452).
 - Origin (Express vs Full Builder).
 - Smart progress bar with the tooltip "Missing: Gallery (3) and VAT Calculator".
 - Date of last edit.
 - Visual thumbnail (or a  icon if empty).
- **Actions:** Preview, Duplicate (useful for 5 PCs with similar specs), Edit (pencil icon), Discard (trash icon) → a 24 h recycle bin with an Undo snackbar.
- **Bulk actions:** checkbox + Bulk Delete / Batch Publish (only at 100%).
- **Filters:** by owner, by progress, by category/brand, by age.
- **Record locking system:** if two admins open the same draft, a visual padlock reads "Alejandro is editing this right now".
- **Pre-publication validation:** a final scan that verifies prices > 0, correct photo formats, and so on. If it fails, the progress bar turns red at the end.
- **UVT change alert:** drafts older than 30 days show a  icon reading "Review the VAT calculation before publishing".

Module 3 · New Categories and Web Ecosystem

A module integrated into the Creator.

- **Creating a new category:**
 - Name (Components, Mice, Gaming Chairs).
 - Visual identifier (an SVG/PNG icon or a solid color) for the Home bubbles.
 - Menu placement (toggle: appear in the hamburger menu or stay as a Peripheral).
- **Automatic synchronization:** menu, filters and the Creator selector.
- **Inherited attributes:** the admin decides which optional descriptors become standard for that category.
- **Global visibility:** a master Off button to hide the category without deleting data.

Module 4 · Test Environment (Sandbox)

A professional sandbox system (refactored in tasks #248 to #250).

4.1 Activation

- Master switch [TEST MODE: ON]. Yellow warning borders appear across the interface when it is activated.
- **Automatic seed of 16 listings** (task #250) that populate `KEY_PUBLISHED` so the admin sees a "full" site from the first second.
- ON/OFF governs both the admin and the store: when it is turned off, the store returns to its real catalog with no reset.

- **Factory reset** always available (the [Purge Test Data] button).

4.2 Express Listings (fast entry)

It allows creating products with only 6 Visual Pillars:

- Cover image.
- Marketing name.
- Condition.
- Brand and model.
- Price.
- Spec trio (processor + RAM + storage in a single input).

4.3 Dummy Population Generator

The [Inject Dummy Content] button creates 12 mixed listings (Laptops, Gaming, Peripherals) with filler photos and specs.

4.4 Test Mirror (Home Test Window)

A special view accessible only to the admin. It allows testing the Domino Effect, the Available ↔ Sold out switch, and so on.

4.5 Payment Gateway Simulator

- Ghost transactions with MP test credentials.
- Validation of notifications (Resend) without deducting real inventory.

4.6 Trace Cleanup

The [Purge Test Data] button with a critical red warning: *"Are you sure? This action will permanently delete every simulated product and sale. This operation cannot be undone"*. Confirmation with [YES].

4.7 "Test Only" Sandbox

A mode where sandbox listings are visible only in the admin and not in the public store (filtered in `lib/adminProducts.js → getCatalogoPublico()`). A warning modal appears when enabling or disabling it. Real items take priority over Sandbox items in every ranking (task #251).

6.4 Operational Flow

Divided according to the sale's life cycle.

6.4.1 Active Catalog

Component: `src/pages/admin/CatalogoActivo.jsx`. The engine behind what the customer sees in the store.

Positioning and Ranking Dashboard (global)

A. Free slot counter: A dynamic indicator reading "Gaming: 2/3 filled | Home: 1/3 | Creators: 3/3 | Deals: 1/1". When there is availability (1/3) → the number turns yellow/orange to flag the opportunity.

B. Visual Tops board (3×2 grid): task #260: 4 compact tables (one per parent category) plus 1 highlighted table for Deals.

Structure per category:

• (position 1, 2, 3 and so on).

- Thumbnail + marketing name.
- A hyperlink to the Builder (it lands on the exact Positioning block).
- A manual position selector plus drag & drop between rows AND between categories (task #265: dragging used to be intra-category only).
- A visibility indicator (Home Top vs Catalog Background).
- **Podium borders:** gold (#1), silver (#2), bronze (#3): task #251.
- **Real items take priority over sandbox** in the visual order (task #251).

C. Deal of the Day module:

- Master ON/OFF toggle.
- A table with the PC identification + ranking + UVT traffic light.
- Operating mode selector:
 - **Single mode (exclusive):** manages 1 deal per day with an automatic close when stock runs out. A cyclical system: automatic rotation at 8 AM Colombia time (task #273, `tryRotateOfertaDelDia()`).
 - **Multiple mode (carousel) · default.**
- **"Force rotation now" button** (task #274, `forceRotate API`).

Individual Management Dashboard

A. Navigation tools:

- A global predictive search box in real time.
- Quick filters (Category / Stock status / VAT status).
- A results counter.

B. Per-product actions:

-  Modify (opens the Builder with the data preloaded).
-  Unpublish (panic button): moves the item to Drafts for correction.
-  Delete (double confirmation with typed text).
-  View history.

C. Listing log (audit):

 Every action generates an Event Card with:

- Timestamp (day, month, year, time).
- Author identity (Modified by: [administrator]).
- Action performed (Price change / Stock update / and so on).
- **Before vs after with highlighting:** red for what was removed, green for what is new.

D. Smart modification logic:

- Automatic VAT recalculation when the price changes (the UVT traffic light updates).
- Image synchronization without re-uploading everything.

Additional post-v18 filters

- **Test/real toggle** in the header (task #247).
- **Unique/multiple filter** (products with color variants).

- **Expandable stock sum** by grouping.
- **Auto-archive at stock 0:** when it reaches 0 units, the item moves to the Graveyard automatically with the `[VENDIDO]` tag.

6.4.2 Pending Management (Logistics Workflow)

Component: `src/pages/admin/Pendientes.jsx`. The core of operational intelligence.

Notification bubble (master alert)

The "Pending" item in the sidebar shows a badge with the number of orders in `RESERVADO_HV` (Reserved Until Verification). The admin knows how many BRE-B payments are waiting for validation without opening the module.

Control table interface

Chronological (oldest first: this prevents delays). A multi-variable search box (customer name / email / phone / batch / ID number / city / transaction ID).

Top tabs (status filters):

- [ALL] General view.
- [RESERVED HV] Payments reserved until verification.
- [READY FOR PICKUP] The customer chose to collect in store.
- [TO SHIP] Approved payments ready for packing.
- [SHIPPED] In transit with the courier.
- [DELIVERED] Delivered to the customer.
- [RETURNED] Reverse logistics and warranties.

Gateway badges on each row: `[MP]` in blue or `[BRE-B]` in purple.

Payment verification block (BRE-B trigger)

Every `RESERVADO_HV` row has a decision panel:

- **Proof viewer:** a PDF/JPG thumbnail. Clicking it opens a Lightbox with zoom.
- **[CONFIRM PAYMENT]** (green): runs `consumir_stock` server-side. State → `POR_ENVIAR`. It fires Email B.2 with the order PDF.
- **[REJECT PAYMENT]** (red): a mandatory "Rejection reason" modal. It runs `liberar_reserva`. Stock returns to "Available". It fires Email B.4 with the reason to the customer.

Dispatch workflow

When the state changes to `ENVIADO`, a form opens:

If the customer chose shipping:

1. Courier selector (Servientrega / Interrapidísimo / Envía / DHL / Other).
2. Tracking number (text).
3. Tracking URL (a direct link to the portal).
4. The [CONFIRM DISPATCH] button fires Email C.1.
5. Post-sale clock: 5 days (Bogotá) or 7 days (national) before Email C.3 (Google review) fires automatically.

If the customer chose in-store pickup:

- The state changes automatically to [READY FOR PICKUP] and fires Email C.2 (Pickup Instructions).

- A 5-day post-sale clock.

Cycle closing

- **[DELIVERED]:** fires Email C.3 (Review).
- **48 h grace period:** the order stays visible in the DELIVERED tab so mistakes can be reverted.
- **Final transition:** after 48 h, the metadata is archived into SOLD and Pending is left clean.
- **[RETURNED]:** it goes back to Pending with an amber alert so the warranty can be handled.

Audit / timeline

Every row has a clock icon → a timeline:

- Creation: "Order generated through [MP/BRE-B] on [date/time]".
- Proof upload: "Proof uploaded by the customer".
- Human validation: "Payment approved by [Admin] on [date]".
- Rejections with a reason.
- Logistics jumps with the identity and the tracking number.

3 s visual cooldown (v18.6)

After a critical action (Confirm/Reject), the button is locked for 3 seconds with a visible animation to prevent a double tap. A green success toast appears on completion.

6.4.3 Sold (Historical and Accounting Memory)

Component: `src/pages/admin/Vendidos.jsx`. The definitive, perpetual archive of successful transactions.

Customer record (valuable data)

Every entry is a case file:

- Product identity (marketing name, batch #2603-104A, final specs).
- Customer identity (full name, ID/tax ID, email, phone, address).
- Economic metric (final price, MP or BRE-B, VAT breakdown if it applied).
- Timing (exact date and time of the transition to DELIVERED).

Features

- **Traceability search:** filters by name, batch, email or ID number.
- **Download Center:** permanent access to the order PDF (URG's "invoice").
- **Reversal trigger (safety):**
 - [Move to Pending]: to correct mistakes.
 - [Move to Returned]: for reverse logistics.

6.4.4 Returned (Contingency Management)

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

- **Inactive and grayscale** while there are no records.
- **Mandatory justification modal** when moving a product here. A "Reason for return" field.
- **Destination triptych:**
 - [Return to Catalog - AVAILABLE] (automatic Stock-In +1).
 - [Return to PENDING] (a new shipment or an adjustment).

- [Mark as SOLD] (a confirmed administrative error).
- **Traceability:** it keeps the link to the original purchase order.

6.4.5 We Buy Your Equipment

Component: `src/pages/admin/CompramosEquipo.jsx`. The Valuation Command Center.

Quick statistics bar (Operations Radar)

3 horizontal glassmorphism cards plus the average response time:

A. Pending valuation: number in yellow, blinking if over 5. **B. In negotiation:** number in cyan blue. **C. Closed today:** number in neon green with a . **D. Average response time** (for example "1h 4m 20s"). Red if over 60 min.

Real-time management table

Columns:

- Master checkbox + individual ones (bulk actions).
- Status (NEW / UNDER ANALYSIS / OFFER SENT / SCHEDULED FOR INSPECTION / PURCHASED / DISCARDED).
- Date/time (most recent first).
- Customer (name + icon).
- Machine (model + type).
- Origin (Full form / Zero Stress photos).
- A direct status dropdown.
- Quick actions (View detail / Preloaded WhatsApp / Change status).
- Internal notes (quick edit).

Filters + priority indicator (Grade A or a high-end CPU → gold border).

Technical Briefing (side detail panel)

It slides in from the right when a customer is clicked. 5 tactical blocks:

A. Mission header. Name, machine type (LAPTOP/TOWER icon), payment preference (Cash / URG Credit), and a floating [CONTACT ON WHATSAPP] button with a preloaded message.

B. Machine DNA (specifications). A command-terminal-style block:

```
> IDENTIFICATION: [Brand and model]
> PROCESSOR: [CPU] ([personality label: Beast Level])
> RAM: [Capacity]
> STORAGE: [Capacity] - [Rocket SSD / Turtle HDD]
```

C. Sensor scan. Dynamic:

- If LAPTOP: screen (size/condition), battery health, original charger.
- For both: unit condition (New/Used/Damaged), cosmetic grade (A/B/C/D with a description).

D. User briefing (business intelligence). "Why are they selling?" plus "What are they looking for next?".

E. Evidence gallery (Zero Stress). A grid of 3 thumbnails → tapping opens a Lightbox with zoom (280×280 px magnifier, 500% zoom: task #239). The viewer uses **createPortal** to escape the aside element (task #236).

Archive and security features

- **PURCHASED** moves the record to History. The photos are auto-purged after **31 days** (they take up a lot of space).
- Two-level editing (quick table / deep detail).
- Real-time push: a new customer appears with a blinking purple glow.
- Sync between admins (if one marks "Under Analysis", the others see it).

6.4.6 Archived (Logical Graveyard)

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

- **Dynamic table:** thumbnail + name, a gray `[VENDIDO]` or red `[ELIMINADO]` label, event date, associated sales.
- **Actions:** Restore (unarchive back to the live catalog) | Delete permanently (confirmation by typing "ELIMINAR" plus a warning about BI inconsistencies).
- **Bulk actions:** Select all + Empty recycle bin (2-level safety).

Product life-cycle logic:

1. **Active:** stock > 0, visible in the catalog.
2. **Transition through a sale:** stock = 0 → `hidden: true` plus the `[VENDIDO]` tag; it is automatically unlinked from the Deal of the Day and Best Sellers.
3. **Transition through discard:** the admin presses Delete → the `[ELIMINADO]` tag.
4. **Archived:** it consumes no front-end resources but still feeds BI.
5. **End of life:** physical deletion only by an express decision of the admin, ideally after the fiscal year closes.

6.X SENTINEL (NEW • Operational Health • v24.18.24)

Component: `src/pages/admin/Centinel.jsx`. A dedicated section introduced in v24.18.24 that does NOT exist in V1.1. It monitors the health of the stock + payments + sync system.

Operational health (alerts)

The main panel uses `<AlertasOperacionalesPanel />`, which groups failures into categories:

- **Orphan BRE-B reservations:** orders with `expira_at < now()` that were not released yet (a signal that the cron did not run or failed).
- **Unconfirmed MP webhooks:** orders in `PASARELA_ABIERTA` for more than 1 h.
- **Products with stock_reservado greater than real stock:** this should be impossible thanks to the CHECK constraint, but the monitor covers it as a defense.
- **Edge Functions with recent errors:** it consumes the logs of `mp-webhook` and `cleanup-stock-fantasmas`.
- **Inconsistencies between `orders.estado` and `productos.reservas`:** a cross-correlation.

Every category is an `<AlertaGrupo />` with `<AlertaItem />` cards (severity + description + a "Resolve" CTA).

Pending Cooldown section

A list of the last 50 orders with a timer showing when their reservation will expire → this makes cancellations easier to anticipate.

Sentinel reporting

The `reportarCentinela()` function in `mp-webhook` and `cleanup-stock-fantasmas` logs every critical movement into the `centinela_events` table. Useful for post-mortems.

6 • Continues with BLOCK II (Customer Management)

Because of its length, this section continues in **Part 5** of the document.

6 • BLOCK II • CUSTOMER MANAGEMENT

6.5 URG Community

Component: `src/pages/admin/Comunidad.jsx`. Back end in `src/lib/comunidad.js`.

An automatic registry table that is fed every time a sale is marked as **COMPLETED** (an MP payment confirmed, or a BRE-B approved by the admin). It centralizes the database of recurring customers and allows lightweight CRM management.

Registration logic

When the admin approves a sale in Pending, `comunidad.js → upsertCliente(cedula, datos)` checks whether the ID number already exists:

- If it does → it adds the new order to `historialCompras` and updates `inversionTotal`.
- If it does not → it creates a new row with the checkout data.

Table structure (6 visible columns)

1. **ID number:** the main search key.
2. **Customer:** first name + last name + a green WhatsApp button (a small icon that opens a direct chat).
3. **Purchase history:** an expandable cell with a "View history" button. Clicking it opens the list below:
 - `#2603-104A | MacBook Pro M2 | [amount] | 12/03/26`
 - `#2602-085A | 96W charger | [amount] | 10/02/26`
4. **Total value:** the accumulated `inversionTotal` sum (useful for identifying VIPs).
5. **Location data:** the city of the last shipment (format "Bogotá, Cundinamarca", task #210).
6. **Actions (...) three dots:** a menu with View last purchase order (PDF) | Start a new sale with this data | Total Deletion (Habeas Data).

Filtering and sorting system

- **Universal search:** filters by ID number or name in real time.
- **Column sorting:** clicking a header sorts (Total value, Customer alphabetically, Last purchase).
- **Time filter:** a "Customers who bought in: [March] [2026]" dropdown.
- **Inactivity filter:** a "Customers who have not bought in more than 6 months" button: useful for remarketing.

6.5.1 Privacy Module • Right to Be Forgotten

It implements the "Total Deletion" (Habeas Data) described in V1.1, but as a **surgical anonymization** (not a physical DELETE):

```
async function anonimizarCliente(cedulaId) {
  await supabase.rpc('anonimizar_cliente', { p_cedula: cedulaId });
  // The RPC clears: first name, last name, email, phone, WhatsApp, address, ID number
  // It preserves: sale value, date, product (marked "Cliente Anonimizado")
  await dispararEmailE4(); // Habeas Data confirmation
}
```

Result: the accounting balance is still real, but there is no trace of who paid.

Re-entry blocking: if the customer buys again after requesting deletion, the system treats them as a New Customer (total investment = \$0).

Orphan warranty ticket: on anonymization, the system generates a ticket tied only to the machine's serial number, so that if the customer comes back with the PC, that number works as the search key instead of their name.

6.5.2 Support guide ([i] button)

An information button next to the module title. Clicking it opens a modal with an explanation:

- What the Right to Be Forgotten is.
- Deletion inventory (what IS deleted: identity, contact, logistics, social history).
- Integrity guarantee (what is NOT affected: financial health, inventory continuity, hardware-based tracking).
- Security warning ("Once confirmed, the personal data cannot be recovered").

6.6 Messaging and Automation Center (Resend + Supabase)

Component: `src/pages/admin/CentroMensajeria.jsx`. It is the technical "mirror" of the `plantillas_correo` table in Supabase. Every change here updates the database that the Edge Functions read in real time before firing sends to the Resend API.

6.6.1 Server and sender configuration

- **Master sender email:** `[store_email]`. Editable from the panel.
- **Deliverability validation:** the system verifies that the domain is verified in Resend (SPF/DKIM). A green/red LED indicator sits next to the field.
- **Notification duality:** the same email address (or an Admin Mirror) receives alerts for every successful order ID.
- **Bulk send capability:** an API connection with URG Community segments. The admin can write manual announcements (deals, launches) and send them to opt-in contacts.
- **Text sanitizer:** it removes spaces and invisible characters from the email before saving (which prevents auth errors with the API).

6.6.2 Design engine: React Email

- React Email is used only as a **layout and compilation** tool.
- The final output is optimized inline-CSS HTML that is loaded into Supabase.
- The templates are **responsive** and tested on more than 50 email clients (Gmail, Outlook, Apple Mail, and so on).
- **Dark mode adapts** automatically.

6.6.3 Flexibility and template creation

- **Dynamic naming:** variables such as `{{nombre_cliente}}` and `{{id_pedido}}` can be **renamed** by the developer.
- **Modular scalability:** adding a new email only requires creating a template in the panel plus configuring the trigger event.
- **Replacement logic (fallbacks):** default values when a variable is empty (for example `{{nombre_cliente}}` → "Cliente URG").

- **Compilation independence:** the admin can change copy quickly in the panel; the developer handles the base structure from React Email.

6.6.4 Tab structure

4 dark-mode tabs with neon purple accents:

Tab 1 • Automations Dashboard

A 3-4 column grid with cards organized into 5 Critical Modules:

1. **Hardware acquisition** (We Buy Your PC).
2. **Transaction cycle and payment validation.**
3. **Logistics** (Dispatch, Delivery confirmation, Maps review).
4. **Growth and loyalty** (Abandoned cart, 6-month maintenance, 12-month replacement plan).
5. **Security and system** (Magic link, Black Box alert, Habeas Data).

Anatomy of each card:

- Left side border (4 px): **purple #8A2BE2** for a customer flow or **orange #FF8C00** for an admin flow.
- Dynamic icon (envelope, truck, lightning bolt, shield).
- Recipient badge [ADMIN] or [CUSTOMER].
- Status switch (neon ON/OFF toggle).
- Metadata "Last run: X hours ago".
- "Configure template" button.

Tab 2 • Content Editor (split view)

Left column (configuration):

- Locked trigger ID (`pago_aprobado`).
- Recipient (the dynamic variable `{{email_cliente}}` or a fixed email address).
- Subject with variable support (for example: "✅ Payment approved! Order `{{lote_pedido}}`").
- Body (a dark gray rich-text editor).
- Attachment management (a selector plus an Automatic Order PDF Generation switch).

Right column (tools):

- Variables panel (clickable `{{nombre_cliente}}` bubbles).
- Info buttons (i) with tooltips: Definition + Origin (table.column) + Example.
- Real-time preview (an iframe rendering both desktop and mobile).

Tab 3 • Global Configuration (Identity & Sender)

URG's visual DNA in every email:

Visual identity (header):

- Header logo (URG_FOOTER_2026.svg), uploadable and updatable.
- Injected first, at the top center of every email.
- Max 200 px wide for mobile.

Sender configuration:

- Editable sender email.
- Sender name alias ("URG TECHNOLOGY - Soporte Táctico").

Signature editor (footer):

- Purple `#8A2BE2` divider line.
- Background color of the legal block (charcoal gray suggested).
- Instant synchronization.
- Text hierarchy: highlighted help (white/purple), small legal text (light gray 11-12px).

Footer content:

```
----- (purple divider line)
A powerful greeting from,
The URG TECHNOLOGY team

Any questions? Write to us at [store email]
or reach us on WhatsApp: [store WhatsApp].
----- (purple divider line)
If you no longer want to receive our updates, unsubscribe here (link)
URG TECHNOLOGY Privacy Policy (link)
URG TECHNOLOGY S.A.S.
© 2026 URG TECHNOLOGY. All rights reserved.
Your tech future starts here.
```

Unsubscribe functionality:

- "Unsubscribe here" → an Edge Function that sets `marketing_opt_out = true` in Supabase. The customer stops receiving Module 4 (Growth) emails, but still receives the mandatory ones (Module 2: Payments).
- "Privacy Policy" → <https://urgtechnology.com/privacidad>.

Tab 4 • Security and Tools (CRUD)

A technical Outbound Log table (Resend logs):

1. Status (green LED Sent / yellow Queued / red Error).
2. Reference (the trigger in the source code).
3. Recipient.
4. Attachments (a clickable PDF icon → opens the file in Storage).
5. Timestamp.
6. Quick action (Resend button).

Global tools:

- Admin Only / Customer Only filters.
- Search by email or by `{{lote_pedido}}`.
- Panic button (Reset to Factory Version).
- Test mode (a real send with fictional data to [store email]).

Data inspection (JSON view): The "View event data" button opens a terminal-style modal showing the original JSON sent to Resend, with a "Copy to clipboard" button. Useful when a customer complains that their name does not appear: if the JSON says `"nombre_cliente": null`, the problem is at the source (the form), not in the template.

6.6.5 Dictionary of triggers and variables

3 accordions in the Editor's side panel:

-  **Process variables (trigger-specific):** `{{monto_total}}`, `{{metodo_pago}}`, `{{lote_pedido}}`.
-  **Customer variables (Habeas Data):** `{{nombre_cliente}}`, `{{email_cliente}}`, `{{telefono}}`.

-  **System variables:** `{{fecha_actual}}`, `{{url_seguimiento}}`.

Catalog of active triggers:

- `solicitud_Compra_PC` → when the trade-in form is submitted.
- `pago_manual_pendiente` → when a proof of transfer is uploaded.
- `envio_despachado` → when a tracking number is generated.
- `pago_aprobado` → when MP/BRE-B is confirmed.
- `pago_rechazado` → when the admin rejects a BRE-B payment.
- `cliente_anonimizado` → when Habeas Data is executed.
- `magic_link_solicitado` → when the admin requests a reset.
- `alerta_critica_blackbox` → when an MFA failure, brute force or geo anomaly is detected.

Auto-detection: if a new column is created in Supabase, it automatically appears as a clickable bubble.

6.6.6 Operations Matrix · Complete templates

Module 1 · Hardware acquisition

A.1 Technical notification (Admin)

Subject: WE FOUND A CUSTOMER WHO WANTS TO SELL US THEIR PC: `{{modelo_pc}}` - `{{nombre_cliente}}`

A body with all the technical data from the form (10 fields) plus "Action Required".

A.2 Received confirmation (Customer)

Subject: We are reviewing your `{{marca_modelo_PC}}` - URG TECHNOLOGY SAS

A didactic thank-you message plus response times (24-48 business hours).

Module 2 · Transaction cycle and payment validation

B.1 Pending BRE-B payment alert (Admin)

Subject: PAYMENT TO VALIDATE (Bre-B): Order `#{{id_pedido}}` - `{{nombre_cliente}}`

B.2 Payment confirmation + purchase order (Customer)

Subject: Payment approved! Your order `{{id_pedido}}` is confirmed

It attaches `ORDEN_URG_{{id_pedido}}.pdf`.

B.3 Successful sale! (Admin)

Subject: NEW SALE CONFIRMED: Start shipping - ID `#{{id_pedido}}`

A full summary with product, payment method, total, mode, recipient, address, and so on. If the customer requested a company invoice: legal name + tax ID.

B.4 Payment rejected (Customer)

Subject: Action required: Update about your order `#{{id_pedido}}`

It includes the `{{motivo}}` variable, editable by the admin at the moment of rejection. A [RETRY THE PURCHASE NOW] button takes the customer to the cart with the product preselected.

Module 3 · Logistics and tracking

C.1 Dispatch notification (Customer)

Subject: Your URG machine is on its way! - Tracking: {{numero_guia}}

C.2 In-store pickup confirmation (Customer)

Subject: 📦 All set! You can come pick up your machine at URG TECHNOLOGY.

C.3 Delivery confirmation + review (Customer)

- Trigger: an internal timer after dispatch.
- Bogotá / in-store pickup: 5 days.
- National: 7 days.

Subject: Your opinion matters: how was your URG Technology experience?

CTA: [\[RATE YOUR EXPERIENCE ON GOOGLE MAPS\]](#) → a link to Google Reviews.

Module 4 · Growth and loyalty

D.1 Maintenance reminder (Customer)

Subject: It is time for free maintenance on your machine!

Trigger: 6 months after the purchase.

D.2 Replacement plan offer (Customer)

Subject: Evolve your setup: it is time for an upgrade!

Trigger: 12 months after the purchase.

D.3 Active cart reminder (Customer)

- Trigger: a cron job 3 h after an ID moves to [CANCELADO](#).
- Safety filter: verify that there is no [POR_ENVIAR](#) order tied to the same email.

Subject: Your high-performance machine is still available (for now).

Module 5 · Security and system

E.1 Magic link / secure access (Admin)

Subject: 🔑 Secure Access Token - ID: #AUTH-{{token_id}}

An [\[ENTER THE ADMIN PANEL\]](#) button. The token expires in 15 min. The footer shows the IP, location and device.

E.2 Security change confirmation (Admin)

Subject: 🔒 URG Security: your password has been updated

An [\[EMERGENCY LOCKDOWN\]](#) button that freezes the account if it was someone else.

E.3 Critical Black Box alert (Admin)

Subject: [CRITICAL ALERT] Unauthorized access attempt - ID: #SEC-{{incident_id}}

A body with: incident summary, connectivity data (IP, location, device, fingerprint), incident report (15 technical data points), and Immediate Action Required with the buttons `[Ignore alert]` or `[BAN IP AND SHUT EVERYTHING DOWN]` (neon red).

E.4 Habeas Data deletion confirmation (Customer)

Subject: Confirmation: your privacy is our priority (Habeas Data)

A message explaining what was anonymized and what remains under an encrypted code for legal and accounting purposes.

6.7 Frequently Asked Questions Center (FAQ Hub)

Component: `src/pages/admin/FAQHub.jsx`. Back end in `src/lib/faqHub.js`.

"Zero-Code" philosophy (CRUD)

- **Create:** add new questions detected in support.
- **Publish/Hide:** a neon toggle per item.
- **Sort:** drag & drop to prioritize.
- **Edit:** modify texts, links and category.

Creation terminal

Every new question has required fields:

A. **Context selector:** toggles between:

- "Preguntas Frecuentes (FAQ)" → renders on `/preguntas-frecuentes` + the Footer + the general search.
- ~~"Todo lo que debes saber (Diseño Web)"~~: discarded in v24.18.26 because of the migration. The Web Design FAQs now live on damorus.com.

B. **Category selector:** a dropdown menu (Hardware, Payments, Methodology, Logistics, Warranty, and so on).

C. **Title field:** the question. D. **Description field:** the rich-text answer. E. **Visibility toggle:** Published / Draft.

Anatomy of an item

- **Category** (thematic filter).
- **Title** (the question in bold).
- **Description** (the answer supports text, links and lists).

Master category manager

- A [+ New Category] button with a name plus a representative icon.
- Technical synchronization with a foreign key to `faq_items` for data integrity. When a category is deleted, it asks: "Move to another one or delete?".
- Visual customization (accent color plus a Lucide/FontAwesome icon).

Default question list (store FAQ)

14 active questions spread across the Logistics, Transactions, Support, Technical Support and Reviews categories. The exact detail is in V1.1 §6.7.5: the base content was not modified, only the Web Design context was migrated.

5.6 Institutional CMS (Privacy + Protected Purchase)

An extension of the FAQ Hub. It allows editing the `/privacidad` and `/compra-prottegida` pages by **blocks**:

- Dynamic header (title + "Last updated" date).
- Block builder (add/edit/delete numbered blocks).
- Fields per block: icon/emoji + title + rich-text body.
- Independent rendering (the pages request the latest content from Supabase on load).
- **Server-side rendering** through an Edge Function that serves the HTML with the data already injected → zero flicker.
- **Local fallback** in cache if Supabase does not respond.

6 • BLOCK III • INTELLIGENCE AND SECURITY

6.8 Statistics and Reports (Business Intelligence)

Component: `src/pages/admin/BI.jsx`. Back end in `src/lib/biMetrics.js`. A command-center dashboard in dark mode with Neon Emerald and Purple accents.

Global control functions

- **Time-travel toggle:** a master selector at the top that switches the WHOLE panel between Daily, Weekly, Biweekly, Monthly, Half-yearly, Yearly and All time.
- **Custom dashboard:** a drag & drop / ranking system that lets the admin reorder blocks by priority.

Technical requirements

- **Interactivity:** everything is dynamic. Changing the range → animated charts (Framer Motion).
- **Connectivity:** the data comes from Supabase plus Resend email interactions.
- **Simplicity:** modals for deep detail; an impactful foreground.

Module 6.8.1 • Data Intelligence (31 metrics)

Every metric has an (i) icon with a tooltip explaining what it is, why it matters and how it is calculated.

I. Traffic and acquisition (6)

1. **Real-time visit counter:** the historical record of users.
2. **Traffic origin (referral):** `document.referrer` classifies Instagram/Google/WhatsApp/direct.
3. **Bounce rate:** sessions that leave in under 10 s with no interaction.
4. **New vs returning visitors:** a browser ID in localStorage.
5. **Popular landing pages:** the entry URL per session.
6. **Device segmentation:** User-Agent → pie chart (PC/Tablet/Mobile).

II. Interaction and search (5)

7. **Internal search CTR:** search volume vs clicks on suggestions.
8. **Top 10 most searched words:** the `bi_searches` table grouped (the `query` field was fixed in v18, bug 4).
9. **Click map by section:** events on category containers.
0. **Average session time.**
1. **Deal banner CTR.**

III. Sales and conversion (4)

2. **Conversion rate (CR):** approved payments / unique visitors.
3. **Cart abandonment rate:** A: cart without checkout / B: checkout without payment.
4. **Average order value (AOV):** gross revenue / successful orders.
5. **Sales by category.**
6. **Peak sales days and hours:** a heatmap (made responsive in v18.5).

IV. Finance and commissions (7)

7. **Sales generated:** the total history including VAT and shipping.

8. **Gross vs net revenue:** subtracts 19% VAT plus the automatic MP commission.
9. **Accumulated VAT collected.**
0. **Gateway investment:** calculated from the active MP plan (3.29% / 2.99% / 2.79%, editable per month).
1. **CAC:** advertising spend / new customers.
2. **ROAS:** sales revenue / advertising spend.
3. **RPC (Result Per Cost):** unique visits / advertising spend.

V. Trade-in module (1)

4. **Total trade-in requests received:** records in the `buy_requests` table.

VI. Security and support (2)

5. **Payment error rate:** `rejected/cancelled` webhooks.
6. **WhatsApp button usage.**

VII. Loyalty and advanced UX (5)

7. **Repeat purchase rate:** emails with more than 1 approved order.
8. **Deal of the Day effectiveness:** purchases that include the Deal of the Day plus the value of other products in the same cart (halo effect).
9. **Scroll depth:** checkpoints at 25/50/75/100% on listings.
0. **FAQ CTR.**
1. **Orders by geographic location:** a heatmap of Colombia.

NEW in v18 · 8-level conversion funnel

An additional `trackHomeScroll` metric (#192-#193). It records how far the user got on the Home (0 to 7):

- 0: entered without scrolling.
- 1: scrolled a little.
- 2: reached Best Sellers.
- 3: reached We Buy Your PC.
- 4: reached Web Design Pro.
- 5: reached Our Store.
- 6: reached the Directory.
- 7: reached the footer.

It detects abandonment with the `pagehide` event so it writes only ONE record per visit.

Module 6.8.2 · Profit & Commission Manager

Component: `src/pages/admin/ProfitManager.jsx`. Back end `src/lib/profitManager.js`.

An audit and commission calculation system for the Technical Administrator (Damorus, in this case).

Visually independent from 6.8.1: a clear separation.

Financial logic

- **Formula:** `Commission = Total_Customer_Sale × Current_Commission_%`.
- **Rationale:** the commission is applied to the gross total (MP gateway costs are the company's operational responsibility and are not deducted).

- **Historical persistence (snapshots):** the commission value is saved at the exact moment of the transaction. If the percentage changes, older records keep their value.
- **Logic suggestion:** the system stores which MP plan was active at the time of each sale. A mid-month change → sales on days 1-14 use the old plan, days 15-30 the new one.

UX/UI

- **Control center (cards):** Pending balance to collect / Next cut-off date / Total paid historically.
- **Chart module:** bars filterable by Daily/Weekly/Biweekly/Monthly/Half-yearly/Yearly.
- **Cut-off and payment management:**
 - A cut-off date selector (it sets the biweekly payment closing day).
 - An automatic biweekly table (every 14 days, with the total amount).
 - A permanent history (Pending/Paid states, never overwritten).

Tools

- Dynamic commission adjustment (configurable percentage).
- Gateway configurator (editable if MP changes its rates).
- Access and integrity: the data lives in Supabase and is visible to the admin and the owners with no discrepancies.

Module 6.8.3 · Report Builder (PDF + CSV)

Component: `src/pages/admin/ReportBuilder.jsx`. Back end `src/lib/reportBuilder.js`.

The URG Report Builder generates executive reports with:

- **Range selector:** a dynamic calendar (Today / Last week / Current month / Quarter / Custom range).
- **Metrics checklist:** the admin ticks which ones to include.
- **Formats:**
 - Executive PDF (a clean design with charts plus tables).
 - CSV/Excel for advanced accounting.
- **Synchronization:** the report remembers which MP plan was active so the calculations are exact.

The `exportarBI()` function in `lib/biMetrics.js` (fixed in v18, bug 5). Quarter support landed in the same update.

6.9 Intelligence and Surveillance Center (Black Box DI)

Component: `src/pages/admin/BlackBox.jsx`. Back end `src/lib/blackBox.js`.

The tactical monitoring core. It detects, tracks and neutralizes threats before they affect inventory or data.

1. Situation dashboard

- **Global heat map (Geo-Threat Map):** a dark-toned world map with URG Purple pulses (successes) and Neon Red ones (failures).
- **Authentication load monitor:** a real-time line chart of requests to `/[panel]`. Unusual spikes → a DDoS / brute force alert.
- **Active block counter:** a digital indicator of blacklisted IPs plus those repelled by Turnstile in the last 24 h.

2. Forensic audit log

A monospaced terminal table with the DNA of every login interaction:

- **Fingerprint:** it captures a unique device ID. If an attacker changes IP but uses the same PC, it recognizes them.
- **Intent analysis (status tag):**
 - `MFA_CHALLENGE_FAIL` : critical alert (they know the password but do not have the TOTP device).
 - `SUSPICIOUS_BOT_PATTERN` : preventive block for non-human behavior.
 - `SUCCESSFUL_ADMIN_ENTRY` : legitimate access.
- **Session tracking:** duration plus the actions performed (through Supabase RLS).

3. Active response module

3.1 Blacklist management (administrative firewall)

- **Blocked IP viewer:** IP + block date + origin (Automatic/Manual) + reason.
- **Unblock action (whitelist):** removes the IP from the blacklist in real time.
- **Proactive manual ban:** an input to block an IP without waiting for detection.
- **Global session shutdown (panic button):** instantly closes every active session.
- **Account isolation:** temporarily blocks an admin until manual verification.

3.2 Suspicion radar (pre-threat)

- **IPs under observation** (amber yellow): more than 3 failures that do not reach the automatic ban.
- **SOP_LEVEL:** LOW (erratic) / MEDIUM (minor config routes) / HIGH (multiple failures plus an unusual location).

4. Behavioral intelligence (local heuristics)

Credential robustness protocol:

- Minimum 9 characters.
- At least 1 number plus 1 special character.
- A neon indicator moving from red to green according to complexity.

Block escalation (rate limiting):

- Level 1: 5 failures / 2 min = 5 min ban.
- Level 2: 5 additional failures = 15 min ban.
- Level 3: 2 extra failures = permanent block until a manual Super Admin reset.

5. Honeypot trap system

Monitored decoy routes (the list is not published).

Automatic action:

- The IP is tagged as a "Potential Aggressor".
- A preventive 24 h ban when it interacts with more than 2 decoys.
- It captures timestamp + IP + country + User-Agent.

6. Critical notification protocol (Resend integration)

The Black Box fires Trigger E.3 when:

1. `MFA_CHALLENGE_FAIL` : more than 2 code failures in under 5 min.
2. `GEOGRAPHIC_IMPOSSIBILITY` : a successful access more than 500 km from the previous one in under 1 h.
3. `BRUTE_FORCE_LOCKOUT` : an automatic block for exceeding the allowed attempts.

Data synchronization: it sends `incident_id` plus `interaction_score` to the Resend API for an email with the complete forensic report.

There is a webhook so that the [BAN IP] button inside the email performs the manual ban without entering the panel.

6 • BLOCK IV • CONFIGURATION AND STRUCTURE

6.10 Tax, Financial and Logistics Configuration

Component: `src/pages/admin/Tributario.jsx`. Back end `src/lib/tributarioConfig.js`.

The system's Master Variables. Any change here propagates through a Global Variable to the Creator, Checkout, Resend and BI.

Block 1 • Global Tax Calculator (VAT/UVT engine)

- **Current UVT value:** \$52,374 (editable). The official 2026 figure.
- **Exclusion threshold (UVT):** 50 (editable, default).
- **Automatic calculation (read-only):** \$2,618,700 (the cap in pesos).
- **General VAT rate:** 19% (editable).

Tax Help Center ([i] button): a micro-manual explaining what a UVT is, the exclusion cap and the final-price rule.

Systemic impact: a UVT change from \$52k to \$55k → the threshold rises automatically → every product is reclassified as VAT-free / VAT-taxed instantly.

Block 2 • Logistics Management (Shipping Matrix)

Shipping rate table:

1. **In-store pickup** \$0 (not editable).
2. **Bogotá D.C.** \$0 (default, editable).
3. **National** (configurable rate).

A [+ Add City/Special Zone] button for exceptions (for example, a special rate for Medellín). If the customer picks Medellín, that rate takes priority over the National one.

A Global Bogotá Charge switch (ON/OFF toggle plus an editable cost field).

Block 3 • Profitability Calculator (Mercado Pago)

It does NOT affect the customer price: it is internal BI only.

Gateway rate configuration:

- Commission percentage (editable).
- Fixed transaction cost (editable).

Withdrawal plan selector:

- ☐ Instant: 3.29% + \$800 + VAT on the commission.
- ☐ 7 days: 2.99% + \$800 + VAT.
- ☐ 14 days: 2.79% + \$800 + VAT (the most profitable).

BI logic: the selected percentage is subtracted automatically from the Total Value in reports to produce the Real Net Profit.

Block 4 • Marketing Hub (Coupons)

A discount code system:

Promotion creator:

- Code (for example [coupon code]).
- Type (Percentage % vs Fixed value \$).
- Usage limit.
- Expiry (calendar).
- Golden rule: non-cumulative.

Active coupon viewer: Code | Usage progress (15/100) | Status | Edit/Delete.

Block 5 · VAT & UVT logic (cases)

A. NO VAT (excluded)

Price $\leq$ 50 UVT $\rightarrow$ VAT = 0.

B. WITH VAT (taxed)

Price > 50 UVT.

- Subtotal = Total price / 1.19
- VAT = Total price - Subtotal

Example, a PC at \$3,000,000:

- Subtotal \$2,521,008
- VAT \$478,992

C. EXEMPT (special cases)

Special cases under Art. 424, item 6 (international agreements, equipment for people with disabilities). It is marked as "Exempt" (not "Excluded") on the electronic invoice.

Explanatory special case

A PC at \$3,333,333 with a 10% discount (\$3,000,000) plus a 5% coupon plus national shipping (configurable rate):

```
Product: MacBook Pro (with 10% OFF applied)
Subtotal:      $ 2.521.008
VAT (19%):     $   478.992
Shipping:      (shipping fee)
Gift coupon:   -$  150.000
TOTAL TO PAY:  $ 2.850.000 + shipping
```

6.11 Master Panel (Technical Settings and Security)

Component: `src/pages/admin/PanelMaestro.jsx`. Back end `src/lib/siteConfig.js`, `src/lib/contactConfig.js`, `src/lib/bannerHero.js`.

1. Environment Variable Management (API keys & DNS)

A Connectivity Terminal. Every block has explanatory text plus a "Save and update the store" CTA button.

A. MercadoPago Connection Center

- **Access token** (production): a password field.

- **Webhook URL** generated by Supabase to receive confirmations. It has to be pasted into the MP Developers Panel.
- Edge Functions involved: `mp-update-token` and `mp-update-public-key` to refresh credentials without redeploying the bundle.
- Edge Function `mp-validate-token` to verify that the token is functional before saving.
- Edge Function `mp-diagnostics`, which runs a test sequence and reports the status.

B. BRE-B protocol

- **Verification key.**
- **QR manager:** a file selector (image/PDF of the account's QR code). It is reflected automatically in the customer's purchase flow.

C. Messaging engine (Resend)

- **Resend API key:** a high-security field. Pasted into the Supabase database and managed from the panel.
- Edge Function `resend-update-token` to save or update it.
- Edge Function `resend-validate-token` to verify the connection.
- Edge Function `resend-send-test` for a test send.
- **DNS records (DKIM, SPF, DMARC):** a table with Type / Name / Content / TTL / Prio.
 - SPF mx: reception.
 - SPF txt: authorization plus security (without it → SPAM).
 - DKIM: a digital signature that guarantees the message was not altered.
 - DMARC: an instruction to Gmail/Outlook about what to do if the check fails.

D. Server identity (nameservers)

- Primary NS: `[redacted]`
- Secondary NS: `[redacted]`
- A micro-manual about what they are and when to change them.

E. Return and security configuration (Site URL)

- **Official site URL:** `https://urgtechnology.com` (or `https://damorus.space` for the current staging).
- **Redirect routes:** `https://urgtechnology.com/[panel]`.
- Guidance: they guarantee a circular security flow.

F. Ignition protocol (first boot)

4 steps for the developer:

1. Open the tunnel (Management API): the Supabase Access Token.
2. Sync the compass (initial Site URL) under Auth > URL Configuration.
3. Link the postman (SMTP/Resend) under Auth Settings > SMTP.
4. Verify automatic redirection: `/auth/v1/callback` in the allowed URLs.

2. User and Hierarchy Administrator

Back end `src/lib/adminUsers.js`. A profile management system based on Supabase Auth's Bcrypt (zero knowledge).

Predefined levels

1. **Master user** · Hidden Master (password redacted). The profile is invisible to the others. Total control.
2. **Main company user** · Super Administrator (password redacted). Total control except for the Designer area.
3. **Guest user** · Read only (password redacted). Visual access to Blocks I and III. Edit buttons are grayed out.

Profile creator

- Username (public).
- Email (linked to Supabase Auth).
- Initial password (URG Standard: min 9 chars + 1 number + 1 special character; a red-to-green bar).
- Rank selector: Super (gold), Editor/Logistics (purple), Guest (gray).

Granular permission matrix

V (View) and M (Modify) checkboxes per section:

- **Block I (Daily Operation):** Login / Control Center / Listing Creator / Operational Flow.
- **Block II (Customer):** Community / Messaging / FAQ Hub.
- **Block III (Intelligence):** BI / Black Box.
- **Block IV (Configuration):** Tax / Master Panel.

Command directory (listing)

A Security Terminal-style table:

- **Invisibility filter:** the master user never appears (protection).
- Avatar (color glow) + user + rank + status + actions (Edit / Reset password / Delete).

Flow logic

1. **Password autonomy:** the admin sets a temporary one, but Supabase Auth forces the user to become the only real owner (zero knowledge).
2. **Interface synchronization:** if a [V] permission is removed while the user is logged in, the tab disappears on the next refresh.
3. **Self-deletion protection:** a Super Admin cannot delete themselves.

3. Global Contact Master and Footer

A sub-module that centralizes the contact identity. A Global Variable propagated to the site, the emails and the database.

A. Identity and operation

- **Editable tagline** ("Colombian company leading in refurbished technology...").
- **Opening hours** (multi-line textarea).
- **Web Design Pro switch** (ON/OFF toggle: it controls the visibility of the promo banner).
- **Corporate email:** [store_email] (it updates the mailto: in the footer and in emails).

B. Global support (master button)

- **Single WhatsApp field:** [store WhatsApp] (it updates ALL the CTAs marked as 'Global Support CTA').

- **Automatic sanitization:** a visual version (with spaces and symbols) plus an action version (stripped, for the `https://wa.me/[store WhatsApp]` link).

C. Specialist lines (Footer)

Editable row structure:

- Area (Hardware/Software/Web Design).
- Owner (name, optional).
- Contact (email or phone).

Footer behavior: adding a row here automatically generates an entry in Column 3.

D. Directory and geographic location (Dual-Map Control)

D.1 Visual control (iframe): a wide text area for the `<iframe ...>` code. Editable without touching HTML.

D.2 Action control (hyperlink): a URL input `[Google Maps link]`. It triggers the opening of Google Maps/Waze/Apple Maps on the phone.

4. AI Chat Module (Web Design Pro)

Component: `src/lib/aiChat.js`. After v24.18.26 this module changed purpose:

A. Master switch

An ON/OFF toggle for the Web Design Pro section. Turning it off automatically hides:

1. The mention in the hamburger menu.
2. The Home banner.
3. The mention in the Footer.

Rule: recalculate spacing so no gaps are left behind.

B. AI Control Center (Live Code Editor) · DEPRECATED in URG

This feature is still present in the code but **disabled**, because after the v24.18.26 migration all the AI/Typebot logic lives on `damorus.com`. The editor is kept as infrastructure in case it is reactivated in the future.

C. Backup and recovery (Phoenix button)

A neon red "Restore original value" button. A confirmation modal. A Protected Environment Variable in Supabase holds the Master Script.

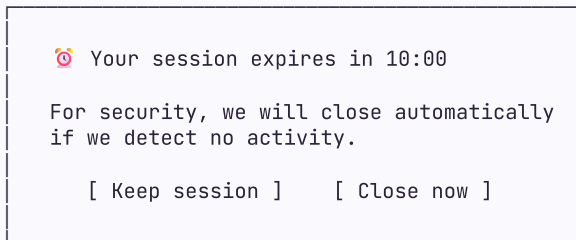
6.12 Profile and Session

Component: `src/pages/admin/Perfil.jsx` . Accessible to any authenticated admin (each one sees their own). It allows:

- Editing name, photo and a valid email.
- Password change with double validation (current + new + confirm).
- Remote session sign-out (checkbox).
- Enabling/disabling TOTP MFA with a QR code.

Inactivity session modal (v6 update)

After 50 minutes of inactivity a warning modal appears 10 minutes before the forced sign-out:



Any interaction (click, scroll, navigation) resets the timer. If it runs out, the user is redirected to `/[panel]` with the message "Session closed due to inactivity to protect your data".

Timer component: `src/hooks/useInactivityTimer.jsx` .

Part 6 continues with section 7 (cross-cutting technical logic), the complete server-side backend, the CI/CD pipeline, cookies and legal data.

7. CROSS-CUTTING TECHNICAL LOGIC AND ERROR PREVENTION

Automatic functions the system runs to guarantee database integrity and robust communications. This section consolidates the 25 points from V1.1 with the practical adjustments discovered in production.

7.1 Telecommunications Sanitizer (`href="tel:"`)

Implemented in `src/lib/format.js` → `sanitizePhone(input)`:

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

- **Admin input:** `"[store WhatsApp]"` or `"[store WhatsApp]"`.
- **Technical output:** `[store WhatsApp]` → `<a href="tel:[store WhatsApp]">`.
- **Visual output:** the original formatted version shown to the user.

7.2 Dynamic URL Encoder (WhatsApp)

Function `buildWhatsappLink(numero, mensaje)`:

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

- It applies `encodeURIComponent()` automatically to the message (accents, spaces, symbols).
- Result: `https://wa.me/[store WhatsApp]?text=Hola%2C%20necesito%20asesor%C3%ADa%20t%C3%A9cnica`.

7.3 Session and Inactivity Security Protocol

Hook `useInactivityTimer()`:

- A 60 min timer restarts with every action (click, scroll, navigation, keypress).
- When 60 min are reached:
 1. It removes the token from `localStorage` / `sessionStorage`.
 2. It redirects to `/[panel]` (login).
 3. It shows a toast: *"Session closed due to inactivity to protect your data"*.
- **Warning modal 10 min before** (task #131): it lets the admin extend the session without losing context.
- **Middleware:** attempts to go back after logout are rejected and the login is enforced.

7.4 Action Confirmation Modals

Component: `src/components/admin/ConfirmModal.jsx`. Every Save/Edit/Delete action goes through a modal:

"Are you sure you want to make these changes?"

[YES · Neon Green] [NO · Neon Red]

- **YES** performs the action and shows a green toast.
- **NO** closes the modal with no changes.

7.5 Route Security and Middleware (Anti-Bypass)

A. Token and role validation (middleware)

Component `<RequireBloque bloque="I">`. Before rendering:

- Is there an active session? If not → redirect to `/[panel]`.
- Does the user's role have permission for this route?
- If NOT → it aborts the load immediately.

B. Automatic ejection

- Render blocking (a momentary white screen).
- A forced redirect to `/[panel]/dashboard` (the allowed home page).
- A floating toast: *"Access denied: you do not have enough permissions to view this section."*

C. Obfuscation of critical routes

Critical routes use neutral names (not published) that the system maps internally: this makes them harder to guess.

7.6 Data Integrity Protocol (Soft Delete)

7.6.1 Replacing DELETE

Physical deletion in `productos` is forbidden from the main interface. The Delete/Finish actions run `UPDATE status='archived' / hidden=true`.

7.6.2 Origin differentiation

- **Origin A (manual)**: the admin sent the product to the recycle bin.
- **Origin B (automatic)**: stock = 0 after a confirmed sale.

7.6.3 Smart filtering

- **Front end (store)**: queries include `WHERE hidden = false AND status = 'active'`.
- **Back end (BI)**: calculations ignore the hidden state and process every record (so annual balances stay exact).

7.7 Media Optimization (Processing + Preview)

Module `src/lib/imageOptimizer.js`.

A. Layer 1 · Automatic processing

- Mandatory conversion to **WebP**.

- **Max-Pro resizing:** long side ≤ 1920 px (Full HD).
- **Background compression (lossless):** it removes EXIF metadata (privacy plus size).

B. Layer 2 • Manual adjustment interface

A Visual Comparison Console (a split screen, Original vs Processed). Sliders:

- **Resolution** (reduce below 1920 px).
- **Quality (0-100)** for WebP, default 85%.

Real-time weight monitor:

- Original weight (12.5 MB).
- Optimized weight (340 KB).
- A savings % indicator ("92% reduction").

C. Storage and delivery logic

- **Pro lazy loading:** Lazy Loading + a skeleton screen with the dominant color + a 0.3 s fade-in.

D. Editing safety

- A button to reset to the Layer 1 values.
- A 100% zoom preview of a specific area to verify legibility after compression.

7.8 Audit Logic • Abandoned Cart

Table `logs_pedidos` in Supabase. 4 states:

1. **Started** (cart): the customer filled in their data and clicked Continue.
2. **At the gateway** (pending): they jumped to MP or opened BRE-B.
3. **Failed / expired:** 8 min with no confirmation.
4. **Completed:** payment verified.

BI conversion funnel:

Payment sessions started this month:	100
Payments completed:	60
Abandonment rate:	40%

There is a distinction between MP sessions started and BRE-B sessions started so their effectiveness can be compared.

7.9 404 Page • Dark Terminal

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

Universal messaging:

- Title: `LINK NO DISPONIBLE`
- Subtitle: *"The address you requested does not exist or has been moved permanently. Check the URL or use the shortcuts below to continue."*

High-end UI:

- URG logo at 50% opacity, centered.

- `[STATUS CODE: 404]` in small purple text at the top.
- Thin rectangular control buttons:
 - `[ GO HOME ]` (black background, purple border).
 - `[ BROWSE THE CATALOG ]` (text only, underlined on hover).

7.10 Performance • Database Indexing (Supabase)

Target latency: under 50 ms regardless of volume.

```
-- Full-text search index (GIN)
CREATE INDEX idx_productos_search ON productos
  USING gin (to_tsvector('spanish', nombre || ' ' || descripcion));

-- Tree indexes (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;
```

This avoids a full table scan. The server responds instantly even with 1,000+ machines.

7.11 Real-Time Syntax Validation

Module `src/lib/validators.js`. Email through a regex:

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

If it is invalid → a neon red border plus a disabled submit button. The error disappears once it is corrected.

7.12 Micro-interactions (Toast)

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

- **Customer store:** "Add to cart" → a toast in the bottom center corner reading `[ ✓ Product added successfully ]`.
- **Admin:** after Create/Edit → `[ ✓ Changes saved ]` or `[ ✓ Category created ]`.
- **Behavior:** fade in, disappears automatically after 3 s, non-intrusive.

7.13 Dynamic Breadcrumb

Component: `src/components/Breadcrumb.jsx`. Light gray text, small size, separated by `>`.

Logic: `Home > Catalog > [Category or Brand]`. Every level is a link that lets the user step back without the browser "Back" button.

7.14 Data Entry Protection (Anti-XSS)

1. Native validation (front end)

Mandatory HTML5 (`required`, `type="email"`, `pattern`). The system does not allow submission if the format is wrong.

2. Sanitization (anti-hacking)

The `sanitizeXSS()` function in `lib/validators.js` removes or encodes `<`, `>`, `/`, `script` and `onerror`. Bug in v20: the sanitization broke base64 data URLs; fixed in task #207 with an explicit whitelist.

3. Concurrency control (anti-duplicates)

- On pressing Submit → `button.disabled = true` plus a "Loading..." state.
- It prevents duplicate records caused by anxious customers or slow connections.

4. System response

After a successful submission, the form is cleared and a clear confirmation message appears.

7.15 Resilience Hierarchy (Kill Switch)

1. Critical scripts

- Price/VAT/discount calculation.
- Database stock validation.
- Order ID generation plus the gateway signature.
- Loading of the privacy policy and terms.

2. Kill-switch protocol

If a critical function fails → disable every purchase button and payment access.

Banner: *"We are experiencing a technical interruption in the payments module. For your safety, transactions have been paused. Please refresh the page or contact an advisor".*

3. Resilience for non-critical items

If the carousel, dark mode or visual effects fail → a Plan B keeps the purchase flow working. The error is logged in the admin console for review.

4. Back-end integrity verification

An RPC function validates the final price using the server's master data. If the price sent by the user differs from the server's calculation → the transaction is aborted with a 'Data Integrity Failure'.

7.16 Catalog Query Optimization

1. PostgreSQL partial indexes

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

A fast lane covering only active products. It jumps straight to the relevant ones, ignoring the Graveyard.

2. Load separation

- **Public site:** queries mandatorily include `status = 'disponible'`.
- **Admin BI panel:** different indexes, optimized by date.

3. Efficient consumption

It only requests the fields needed for the preview (name, price, main photo). Heavy data (full listings, galleries) loads only on an individual view.

7.17 Server Configuration (.htaccess)

The complete file lives in `dist/.htaccess`:

```

# — Protection: forbid directory listing —
Options -Indexes
ServerSignature Off

# — Force 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]

    # Strip www (canonical without www for SEO)
    RewriteCond %{HTTP_HOST} ^www\.(.+)$ [NC]
    RewriteRule ^(.*)$ https://%1/$1 [L,R=301]

    # SPA fallback · if the file/folder does not exist, serve 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\.html">
        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>

    # Security headers
    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 (uncomment after a successful deployment)
    # Header set Strict-Transport-Security "max-age=15768000; includeSubDomains"
</IfModule>

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

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

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

# — Custom 404 —
ErrorDocument 404 /index.html

```

cPanel permissions (anti-403)

- Directories: **755** (R/X for everyone, W only for the owner).
- Files: **644** (R for everyone, W only for the owner).

- `.htaccess` protected (no directory listing).

7.18 External SDK Verification (Anti-AdBlock)

Function `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);
  });
}
```

Notice on failure: *"Note: your ad blocker may be interfering with the payment gateway. If you run into problems, please disable it temporarily so you can complete your purchase safely".*

7.19 Timeout Management and Async Loading States

Wrapper `lib/asyncTimeout.js`:

- **25 s timeout** for file uploads and payment registration.
- If it is exceeded → it aborts with a clear message: *"The connection is taking longer than expected. We could not confirm your submission. Please try again or check your internet connection".*
- It distinguishes a 500 server error from a timeout or a loss of connectivity.
- A smart retry button that does not reload the page.
- **Tab-close prevention** (`beforeunload`): *"Your submission is still in progress. If you leave now, your payment record may not be completed".*

7.20 Resource Optimization and Storage Limits

1. PHP / Edge Functions environment

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

2. "We Buy Your PC" validation

- Max 3 photos per request.
- Max 10 MB per incoming file.
- Formats: `.jpg`, `.jpeg`, `.png`.
- Dual-layer compression (client-side Canvas API → ≤1.5MB / server-side sanitization plus final compression).

3. Error handling

If a file is over 10 MB before compression: *"The file exceeds the 10MB limit. Please try a lighter version or take the photo directly from the form".*

A real progress bar during compression and upload (not a fake one).

7.21 Deployment Integrity and Time Logic

1. cPanel permissions · see §7.17.

2. Centralized time validation (anti-hacking)

Trusting the client clock is forbidden for reservation countdowns, cart expiry and coupon validity.

Server logic: compare against the server timestamp or Supabase `now()`.

JavaScript `new Date()` is only for visual effects, never as the basis for authorizing a transaction.

3. Anti-abuse

- **Atomic stock validation:** a double database check (the race condition is prevented by the `FOR UPDATE` pessimistic lock, see §5.4).
- **File privacy sanitization:** it strips EXIF data (geolocation, camera model).
- **Rate limiting:** a maximum of 3 attempts per minute on `/compramos-tu-pc` and on stock reservation (anti-DoS / anti-bot).

7.22 Payment Listening Protocol (Webhook)

Full detail in §8 (Server-Side Backend). Summary:

- The `mp-webhook` Edge Function receives MP notifications.
- It validates the origin (secret signature plus allowed IP).
- It runs the atomic `consumir_stock`.
- It returns 200 OK immediately to avoid MP retries.
- Queue handling: errors are logged in `webhook_errors` for a manual re-sync.

7.23 Network Status Notification (Passive UX Resilience)

Hook `useNetworkStatus.js`:

1. **Evolving feedback at 8 s:** if the database takes more than 10 s → "Syncing inventory in real time...".
2. **Immediate rescue channel:** WhatsApp works from second 1.
3. **Non-invasive loading:** blocking the screen with overlays is forbidden. The user can scroll and navigate while things load.
4. **Visual smoothness:** the loader stays active until at least 50% of the elements are ready.
5. **45-60 s fallback:** a "High traffic demand" notice plus a redirect to WhatsApp.

7.24 SPA History Synchronization (Native Feel)

1. **PushState:** opening the Catalog/Listing/Cart → `history.pushState({section: 'nombre'}, '', '/nombre')`.
2. **popstate capture:** the browser "Back" button closes the overlay instead of leaving the site.
3. **Preventive circular navigation:** closing with the X calls `history.back()` to clean up.

7.25 Connection Efficiency (Anti-Error 508)

To protect the Entry Processes on HostGator shared hosting:

1. **Load consolidation (batching):** several related requests → a single JSON POST.
2. **Local persistence before syncing:** BI tracking data → localStorage, synced only at key moments (opening the cart / starting checkout / closing the tab).
3. **Retry control with delay (exponential backoff):** 2 s, 4 s, 8 s before giving up.

8. SERVER-SIDE BACKEND (NEW in V2.0)

This section documents the complete back end. It is the largest architectural addition since V1.1 and represents the turn from "client-centric" to "server-centric" after v24.18.25.

8.1 Edge Functions (Deno runtime, Supabase Cloud)

13 active functions in `supabase/functions/`. Each one is deployed individually with `supabase functions deploy <name>`. They all share a single JSON response schema:

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

8.1.1 mp-webhook (CRITICAL)

Path: `supabase/functions/mp-webhook/index.ts` (about 600 lines).

Responsibility: receive Mercado Pago notifications and apply the atomic stock consumption plus the cancellation of orphans.

Flow:

1. It receives a POST with the body `{ type: 'payment', data: { id: '123456789' } }`.
2. It validates the origin with the MP secret (`MP_WEBHOOK_SECRET`).
3. It queries the MP API at `/v1/payments/{id}` with the access token to get the status.
4. If `status === 'approved'`:
 - It runs `aplicarConsumoStockYHuerfanos(order_id, producto_id)`.
 - That internal function calls the `consumir_stock` RPC plus the `cancelarHuerfanos` RPC.
 - It marks the order as `EFFECTUADA`.
 - It fires email B.2 (Payment Confirmation) to the customer plus B.3 (Successful Sale!) to the admin.
 - It calls `reportarCentinela(evento)` for the audit record.
5. If `status === 'rejected'`:
 - It marks the order as `RECHAZADO_MP`.
 - If there was a reservation, it runs `liberar_reserva`.
6. It **always responds 200 OK** to avoid endless MP retries.

Internal functions:

- `aplicarConsumoStockYHuerfanos()`: consumes stock and cancels orphans in a single transaction.
- `cancelarHuerfanos()`: walks the active reservations, releases those for the product that was already consumed, and fires apology emails.
- `liberarReservasOrden()`: when MP rejects a payment, it releases every reservation for that order.
- `reportarCentinela()`: INSERT into `centinela_events` with type + description + payload.

8.1.2 cleanup-stock-fantasmas (NEW in v24.18.25)

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

Responsibility: invoked by `pg_cron` every 2 minutes. It cleans up expired reservations.

Flow:

1. SELECT reservations with `expira_at < now()`, grouped by product.

2. For each expired reservation:
 - It runs the `liberar_reserva` RPC.
 - It updates the order to the `FALLIDO_EXP` state.
 - If the release frees the last stock unit and there are Realtime-connected customers viewing the product → it fires a broadcast event.
3. It cleans up MP orders sitting in `PASARELA_ABIERTA` for more than 1 h.
4. It reports to Sentinel.

8.1.3 admin-auth

An extra validation endpoint for the admin login. It verifies the Turnstile token against Cloudflare before allowing an attempt.

8.1.4 admin-config-write

A privileged endpoint that receives changes from the Master Panel (API keys, DNS and so on) and persists them encrypted. It is callable only by a Super Admin (it checks the JWT role).

8.1.5-8.1.10 mp-* (MercadoPago helpers)

- `mp-create-preference` : creates the payment preference when the MP checkout starts.
- `mp-validate-token` : validates the access token against the MP account.
- `mp-update-token` : updates the access token after an admin edit.
- `mp-update-public-key` : updates the public key (front-end SDK).
- `mp-diagnostics` : runs a battery of tests and reports the status.

8.1.11-8.1.13 resend-*

- `resend-validate-token` : verifies the Resend API key.
- `resend-update-token` : updates the credential.
- `resend-send-test` : sends a test email.

8.1.14 turnstile-config

It returns the Cloudflare Turnstile site key to the front end (without exposing the secret).

8.2 pg_cron Trigger

Configuration in `supabase/migrations/20260530_pg_cron_cleanup.sql` :

```

-- Enable extensions
CREATE EXTENSION IF NOT EXISTS pg_cron;
CREATE EXTENSION IF NOT EXISTS pg_net;

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

```

The `app.service_role_key` secret is loaded with:

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

Cron verification (a SQL query that is useful for the 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 MercadoPago Webhook · Atomic Detail

Pseudo-code of the flow:

```

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

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

  // 2. Query the MP API for the real status
  const payment = await consultarMercadoPago(body.data.id);

  // 3. If approved → consume
  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
    });

    // Cancel orphans of the same product
    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. Always 200 OK
return new Response('OK', { status: 200 });
});
```

8.4 Cross Cancellation BRE-B ↔ MP

Already documented in §5.5. Reinforcing the technical summary:

- When stock is consumed through one channel, the system scans **all** the pending reservations for the same product.
- If the final stock is ≤ 0 → it releases every orphan reservation, dispatches an apology email and emits a Realtime event for connected customers.

The `cancelarHuerfanos` function lives in `mp-webhook` and is replicated in the admin endpoint that approves BRE-B payments.

9. CI/CD PIPELINE (NEW in V2.0)

Complete documentation of the automatic deployment pipeline.

9.1 GitHub Actions Workflow

File: `.github/workflows/deploy.yml`. Trigger: every `git push origin main`, or a manual run from the Actions tab.

```
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 with a Protected Clean Slate

`dangerous-clean-slate: true` deletes ALL the files on the server before uploading the new ones. This guarantees that the folder ends up 1:1 with the current `dist/`.

Protection through `exclude` :

- `.well-known/**` : SSL certificates (Let's Encrypt / HostGator AutoSSL). If they were deleted, the next renewal would fail → the site would lose HTTPS.
- `cgi-bin/**` : HostGator CGI scripts (empty but present).
- `.ftpquota` : the internal FTP quota file.
- `.git*`, `node_modules/**`, `.DS_Store` : files that must NOT be uploaded (unnecessary in production).

9.3 Environment Variables and Secrets

GitHub Secrets configured under `Settings → Secrets and variables → Actions` :

Secret	Purpose
<code>VITE_SUPABASE_URL</code>	<code>https://[supabase-project]</code> (the URG project).
<code>VITE_SUPABASE_ANON_KEY</code>	Public anon key of the URG project (RLS-controlled reads).
<code>FTP_SERVER</code>	HostGator FTP server (for example <code>damorus.space</code> or an IP).
<code>FTP_USERNAME</code>	FTP user (<code>[ftp-user]</code>).
<code>FTP_PASSWORD</code>	FTP password.
<code>FTP_SERVER_DIR</code>	Target path (<code>/public_html/</code> for the current staging).

The `VITE_*` variables are embedded in the JS bundle during `vite build`. Since `.env` is in `.gitignore`, they are rebuilt from the secrets on every deploy.

9.4 Manual Rollback

If a deploy breaks the site:

1. **Revert the commit** on GitHub (`git revert HEAD` or from the UI).
2. **Push** → this triggers a new deploy with the previous version.
3. Approximate time: 2-3 minutes until the site is restored.

For critical emergencies, the admin can upload files manually through the cPanel File Manager (with access to `/public_html/`).

9.5 HostGator Hosting · Configuration

- **Plan**: Shared.
- **cPanel URL**: `<personal-hostgator-panel>`.
- **Active domain**: `damorus.space` (URG staging). It will point to `urgtechnology.com.co` in the future, once the definitive domain is acquired.
- **PHP**: 8.1+ configured in cPanel.
- **SSL**: HostGator AutoSSL (Let's Encrypt), renewed automatically as long as `.well-known/` stays intact.

- **FTP:** the `[ftp-user]` account with its root directory at `public_html` (not in a subfolder, an initial mistake corrected during setup).

X. COOKIE SYSTEM (URG)

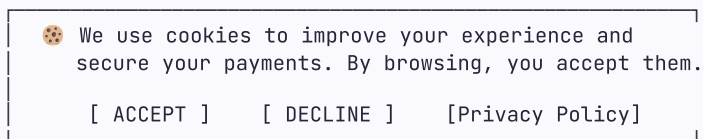
Implementation: `src/components/CookieBanner.jsx`. Vanilla JavaScript, no additional frameworks.

1. Technical functionality

- **Cart persistence:** `localStorage['urg.cart']` with a 24 h TTL. If the customer closes the tab or reloads, the cart comes back.
- API: `cartTracker.js` → `saveCart(items)` / `loadCart()`.

2. Legal notice interface

A minimalist banner at the bottom of the screen:



- It appears **only the first time**. After a click on [ACCEPT] or [DECLINE], the preference is stored in `localStorage['urg.cookieConsent']` and it does not come back.
- Text and design:
 - Inter typeface at 14 px.
 - Background `rgba(11,11,20,0.95)` plus `backdrop-filter: blur(8px)`.
 - Purple top border `1px solid var(--urg-purple)`.

3. Supabase integration for analytics

The `cartTracker.js` → `trackEvent(tipo, payload)` function sends data to the `eventos_rastreo` table:

- `product_view`: a listing was viewed.
- `add_to_cart`: the add button was pressed.
- `cart_abandoned`: items in `localStorage` with no completed payment after 24 h.

4. Sales recovery (repurchase)

The email captured at checkout (even before paying) → sent to the `Leads_checkout` table. If the purchase is not marked as `EFFECTUADA`, cron job D.3 fires the "Abandoned cart" email the next day with the pending product.

5. Segmentation for ads

Data is stored in an organized way (email, product of interest, category) so it can be exported to Facebook/Google Ads for retargeting with the models the user looked at.

6. Privacy and transparency

The banner states that accepting means consent to the processing of browsing data for statistical and advertising purposes, linking to the URG Data Processing Policy.

X. URG TECHNOLOGY LEGAL DATA

All of it editable from Master Panel § Contact Master (§6.11).

- **Legal name:** URG TECHNOLOGY S.A.S.
- **Main WhatsApp:** [store WhatsApp]
- **Web design (post-v24.18.26):** Alejandro@damorus.com
- **Tax responsibility:** Code 48 (VAT responsible).
- **Economic activity:** Code 4741 (Retail sale of computers, peripheral equipment, software and telecommunications equipment in specialized stores).

APPENDIX A • Summary Changelog (v1.0 → v24.18.26)

Versions follow a `vMajor.Minor.Patch` pattern. The complete history lives in the internal task system (293 tasks completed at the close of this documentation). The most important milestones:

- **v1.0 (April 2026)**: the first build based on the V1.1 document. A working store, MP + BRE-B integrated, FAQ Hub, basic BI, basic Black Box.
- **v1.1.x**: BI hardening plus the complete Resend messaging plus the Command Center.
- **v17**: dead code cleanup (1,080 lines removed), `_backups/` consolidated.
- **v18.x**: BI refactor (TopTerms, LandingPages, the 8-level scroll funnel, advanced metrics).
- **v19**: lead photo gallery with a magnifier viewer, order tracking PIN plus a 5-day auto-purge.
- **v20**: fix so sanitizeXSS no longer breaks base64, optimized magnifier viewer.
- **v21**: animations paused outside the viewport (performance).
- **v22-v23**: Typebot DAMORUS section + Coinbase + reactive Status (later migrated).
- **DAMORUS.com (parallel branch)**: a project cloning URG with a minimalist landing page plus a filtered admin.
- **v18.x (URG continuation)**: Active Catalog refactor, professional Sandbox, Cropper, FAB + Mac Spotlight modal, premium SpecTooltip.
- **v18.14**: critical BATCH for the session countdown, temporary BRE-B stock, the sale no longer disappears.
- **v18.15**: the server-side atomic reservation system (5 stages).
- **v18.16-v18.18**: form color fixes, step 4 validation, draggable traffic-light timer.
- **v18.20**: stock auto-fallback plus Pending cooldown.
- **v24.18.22**: SOLD seal without the diagonal, purchase blocking at stock = 0, Notify-me feature.
- **v24.18.23**: MP consumes only through the webhook, cross cancellation, overselling modal.
- **v24.18.24**: Sentinel Operational Health.
- **v24.18.25**: the server-side architectural fix (mp-webhook stock + BRE-B cron + orphans on the server).
- **v24.18.26**: migration of /diseno-web to damorus.com plus GitHub Actions FTP deploy to HostGator.

APPENDIX B • src/ Folder Structure

```
src/
├── App.jsx                # Route map, bootstrap
├── main.jsx               # Entry point + providers
├── index.css              # Variables, animations, utilities
├── components/
│   ├── system/           # ErrorBoundary, NeonAccent
│   ├── admin/            # AdminLayout, AdminSidebar, ToastProvider, Confirm, etc.
│   ├── compramos/        # CompramosTuPCForm + steps
│   ├── cards/            # Ficha1, Ficha2, reusable
│   ├── 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 web design pro, disabled)
│   └── 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    # Consolidated re-export
├── 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     # Official Colombia catalog
│   └── products.js        # Demo / fallback products
```

APPENDIX C • Summary Dictionary of Supabase Tables

Table	Purpose	RLS
<code>productos</code>	master catalog + stock + reservations (jsonb)	public SELECT, mutations admin only
<code>orders</code>	purchase orders (state, totals, customer, product, batch)	public INSERT, mutations admin only
<code>orden_pins</code>	order tracking PINs (4 digits, 5-day auto-purge)	SELECT with a customer join
<code>comunidad_clientes</code>	recurring customers (ID number, history)	admin only
<code>buy_requests</code>	leads from We Buy Your PC	admin only
<code>plantillas_correo</code>	editable Resend templates	admin only
<code>faq_items</code> + <code>faq_categories</code>	FAQ Hub CRUD	public SELECT
<code>bi_searches</code> , <code>bi_clicks</code> , <code>bi_sessions</code>	BI metrics	public INSERT, admin SELECT
<code>profit_snapshots</code>	commission snapshots per sale	admin only
<code>centinela_events</code>	log of critical events	admin only
<code>webhook_errors</code>	pending retries	admin only
<code>app_credentials</code>	encrypted API keys (MP, Resend, Turnstile)	master user only
<code>admin_users</code> (view)	registered admins (Supabase Auth)	Super Admin only
<code>audit_logs</code>	audit timeline (who did what)	admin only

Full migrations in [supabase/migrations/](#):

- `20260424021420_remote_baseline.sql`: base schema.
- `20260507000000_initial_schema.sql`: refinement.
- `20260507120000_app_credentials.sql`: encrypted credentials.
- `20260514000000_global_state_sync.sql`: global state sync.
- `20260514120000_phase24_operations_intelligence.sql`: BI + Operations.
- `20260515000000_rls_read_policies.sql`: RLS read policies.
- `20260515120000_profit_snapshots_pk.sql`: snapshots PK.
- `20260516000000_profit_cortes_cd_lado.sql`: cd-side cut-offs.
- `20260516120000_turnstile_site_key.sql`: Turnstile site key.
- `20260530_pg_cron_cleanup.sql`: pg_cron cleanup schedule.

CLOSING

This document consolidates the state of URG TECHNOLOGY as of **19 June 2026** (internal version v24.18.26). It covers 293 accumulated iterations since the V1.0 launch, with an emphasis on the architectural decisions that did not appear in V1.1:

- The client-centric to server-centric migration (Edge Functions + pg_cron + Postgres RPCs).
- The atomic reservation model with cross-method overselling prevention (MP ↔ BRE-B).
- Sentinel Operational Health as a dedicated monitoring module.
- The complete migration of `/diseno-web` to the external site `damorus.com`.
- The GitHub Actions → HostGator FTP CI/CD pipeline with `dangerous-clean-slate` and SSL exclusions.
- The deep refactor of the Listing Creator (sticky tabs + real previews + Sandbox + Cropper).
- Security hardening (Black Box + invisible Turnstile + rate limiter + honeypots).
- The mobile-first reorganization of the Header, Footer, Listing 3, Catalog and every critical flow.

To rebuild this site from scratch, it is enough to follow these steps:

1. Create a Vite + React 19 + Tailwind v4 project as described in §0 (Architecture).
2. Create a Supabase project with the migrations in Appendix C.
3. Deploy the 13 Edge Functions from §8.1.
4. Configure pg_cron according to §8.2.
5. Implement the components according to §2 (Home), §3 (Tabs), §4 (Listings), §5 (Payments), §6 (Admin).
6. Apply every rule from §7 (Cross-cutting logic).
7. Configure the domain + HostGator + GitHub Actions according to §9.
8. Upload the definitive `.htaccess` from §7.17.

The result is a high-performance digital store that runs 24/7 with no dependence on a human being present in the admin panel, with resilience against gateway failures and full traceability of every peso invoiced.

End of document V2.0.