# Zellfy.ai — Blueprint Técnico Completo (Astro + Vue + Laravel)

Este documento es la **versión extendida** para construir la web completa de Zellfy.ai en **Warp**, con la arquitectura definida: **Frontend en Astro + Vue**, **Backend en Laravel**, y **Base de datos MySQL**.

## 🏗️ Arquitectura General del Proyecto

/zellfy-ai  
├── frontend/ # Sitio público (Astro + Vue)  
│ ├── src/  
│ │ ├── components/ # Componentes Vue y Astro reutilizables  
│ │ ├── layouts/ # Plantillas principales  
│ │ ├── pages/ # Cada página del sitio (rutas)  
│ │ │ ├── index.astro  
│ │ │ ├── como-funciona.astro  
│ │ │ ├── soluciones.astro  
│ │ │ ├── pricing.astro  
│ │ │ ├── casos.astro  
│ │ │ ├── contacto.astro  
│ │ │ └── blog.astro  
│ │ ├── assets/ # Imágenes, íconos, logos, estilos globales  
│ │ └── styles/ # Archivos CSS / Tailwind / variables  
│ ├── astro.config.mjs # Configuración Astro (build y rutas)  
│ ├── package.json # Dependencias frontend  
│ └── tsconfig.json  
│  
├── backend/ # API + Lógica de negocio en Laravel  
│ ├── app/  
│ │ ├── Http/Controllers/  
│ │ ├── Models/  
│ │ ├── Services/  
│ │ └── Routes/api.php  
│ ├── database/  
│ │ ├── migrations/  
│ │ ├── seeders/  
│ │ └── factories/  
│ ├── routes/web.php  
│ ├── composer.json  
│ └── .env.example  
│  
├── database/ # Configuración MySQL  
│ └── schema.sql  
│  
└── docs/ # Manuales, branding y documentación  
 ├── Zellfy\_BrandBook.pdf  
 └── Zellfy\_Website\_Blueprint.md

## ⚙️ Tecnologías Base

| Componente | Tecnología | Descripción |
| --- | --- | --- |
| Frontend | **Astro + Vue 3** | Combinación ideal para rendimiento y reactividad. |
| Backend | **Laravel 11** | Framework PHP robusto para lógica de negocio y APIs. |
| Base de datos | **MySQL 8** | Persistencia de datos de usuarios, leads y configuraciones. |
| Estilos | **TailwindCSS** | Sistema visual moderno y flexible. |
| Hosting | **Vercel** (frontend) + **DigitalOcean** (backend) | Separación de capas por eficiencia y seguridad. |
| Integraciones | Google, Meta, APIs Bancarias, Clave Única, Email SMTP | Conexiones preconfiguradas con endpoints REST. |

## 🎨 Paleta de Marca (Repetición Oficial)

* Azul eléctrico: #00BFFF
* Verde menta: #00C48C
* Violeta digital: #6C63FF
* Gris humo: #F5F6FA

**Tipografía:** Manrope (principal) / Poppins (alternativa)

**Slogan:** Conecta. Gestiona. Cierra.

## 📄 Páginas y Contenido (Astro)

### 🏠 index.astro — Inicio

* Hero con CTA “Empieza a fluir”.
* Bloques de beneficios (Conecta / Gestiona / Cierra).
* Sección Integraciones con logos.
* CTA final: Solicitar Demo.

### ⚙️ como-funciona.astro

* Timeline visual con 8 etapas del journey del cliente.
* Cards de ejemplos (Inmobiliario, Automotriz, Créditos, E-commerce).
* Animaciones Vue: entrada progresiva en scroll.

### 💡 soluciones.astro

* Dos columnas principales:
  + **Zellfy** (CRM de ventas)
  + **Zellflow** (automatización)
* Cuatro módulos secundarios: Integraciones, IA, Data, Escalabilidad.
* Frase central destacada.

### 💰 pricing.astro

* Tabla comparativa (Starter, Pro, Enterprise).
* Botones CTA (“Probar Gratis”, “Solicitar Demo”).
* Planes cargados dinámicamente desde API Laravel (/api/pricing).

### 🏆 casos.astro

* Grid con tarjetas de casos (Ecomac, Creces).
* CTA a detalle individual (/casos/:id).
* Componente <CaseCard /> reutilizable.

### 📬 contacto.astro

* Formulario con validación Vue + Captcha.
* Conexión directa a /api/contact en Laravel.
* Mensaje de confirmación visual.

### 📰 blog.astro

* Sistema de posts Markdown (/content/blog/\*.md).
* Renderizado estático con Astro MDX.

## 🔌 Conexión Front–Back (API Laravel)

### Endpoints Base (API)

| Método | Ruta | Función |
| --- | --- | --- |
| GET | /api/pricing | Devuelve lista de planes con precios dinámicos. |
| POST | /api/contact | Envía formulario de contacto a CRM Zellfy. |
| GET | /api/casos | Lista casos de éxito con descripción e imagen. |
| GET | /api/blog | Carga artículos publicados. |
| POST | /api/leads | Inserta nuevo lead capturado desde frontend. |

## 🧠 CRM Integrado (Zellfy)

El backend Laravel incluye módulo CRM con: - **Tabla leads** → Captura de formularios, fuente, estado. - **Tabla users** → Datos de usuarios y roles internos. - **Tabla campaigns** → Control de campañas activas. - **API Key system** para integraciones externas.

## 🔒 Seguridad y Autenticación

* Middleware Laravel: Auth, ThrottleRequests, Sanctum.
* Formularios protegidos con reCAPTCHA.
* CORS habilitado para frontend Astro.
* HTTPS obligatorio (redirección global).

## 🧰 Configuración de Integraciones

**Archivo .env (Laravel):**

APP\_NAME=Zellfy.ai  
APP\_ENV=production  
APP\_KEY=base64:xxx  
DB\_CONNECTION=mysql  
DB\_HOST=127.0.0.1  
DB\_PORT=3306  
DB\_DATABASE=zellfy  
DB\_USERNAME=root  
DB\_PASSWORD=secret  
MAIL\_MAILER=smtp  
MAIL\_HOST=smtp.gmail.com  
MAIL\_PORT=587  
MAIL\_USERNAME=contacto@zellfy.ai  
MAIL\_PASSWORD=\*\*\*\*\*\*\*\*  
API\_GOOGLE\_KEY=xxxxx  
API\_META\_TOKEN=xxxxx

## 🧱 Base de Datos (MySQL)

### Tablas Clave

CREATE TABLE leads (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 name VARCHAR(255),  
 email VARCHAR(255),  
 phone VARCHAR(50),  
 source VARCHAR(100),  
 status ENUM('nuevo','en\_proceso','cerrado','descartado') DEFAULT 'nuevo',  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE plans (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 name VARCHAR(100),  
 description TEXT,  
 price DECIMAL(10,2),  
 features JSON  
);

## 🚀 Despliegue y Hosting

### Frontend (Vercel)

* Conecta el repositorio GitHub con Warp.
* Define npm run build como comando de build Astro.

### Backend (DigitalOcean / Laravel Forge)

* Implementa con pipeline automático desde rama main.
* Define .env en el servidor.
* Certificados SSL mediante Let’s Encrypt.

## 📈 SEO & Analytics

* Sitemap dinámico generado con Astro.
* Meta tags automáticas por página.
* Integración con Google Analytics 4 + Meta Pixel.
* Schema.org JSON-LD para productos y casos de éxito.

## 💬 Copy Base de Marca

Zellfy.ai no es solo una plataforma de automatización comercial: es un ecosistema vivo que conecta, gestiona y acelera todo tu proceso de ventas.

Conecta. Gestiona. Cierra.

## ✅ Pasos para Implementar en Warp

1. Crea un nuevo proyecto Warp.
2. Sube este archivo .md en el asistente inicial.
3. Warp generará automáticamente las carpetas base y las rutas en Astro + Vue.
4. Importa los diseños desde Figma para vincular los componentes visuales.
5. Configura Laravel con MySQL (puede ser en Docker o DigitalOcean).
6. Sincroniza el backend con los endpoints del frontend.
7. Activa la API Zellfy interna para registro de leads y planes.

**Documento preparado por:** ChatGPT (GPT-5)  
**Proyecto:** Zellfy.ai — Plataforma Inteligente de Ventas  
**Fecha:** Octubre 2025