Arturo Morgadanes - Fontanero en Vigo

Claude Code Agentic Prompt

Build a modern, SEO-optimized Spanish-language website for a solo plumber. The site must rank #1 locally, generate leads via contact form and WhatsApp, and require zero ongoing technical maintenance.

Context

Client: Arturo Morgadanes, solo plumber in Galicia, Spain
Current state: Paper notebook + phone calls
Goal: Professional web presence that captures leads 24/7
Constraint: Non-technical user—if it requires a CMS login, he won't use it

Critical insight: This is a brochure site, not a web app. Content changes maybe twice a year. Skip the CMS—hardcode the content and make it beautiful.

Tech Stack (Intentionally Simple)

```
Framework:    Next.js 14 (App Router, static export)
Styling:      Tailwind CSS 3.4
Deployment:   Vercel (free tier is sufficient)
Email:        Resend API (contact form notifications)
Scheduling:   Cal.com embed (optional, links to Google Calendar)
Analytics:    Google Analytics 4 + Search Console
Communication: WhatsApp Business floating button
```

Why no CMS? A solo plumber won't log into Strapi to update content. For the rare update (new phone number, add a service), he calls his developer or edits a simple JSON file. This eliminates an entire backend, reduces costs to €0/month hosting, and removes maintenance burden.

Directory Structure

```
fontanero-vigo/
├── app/
│   ├── layout.tsx          # Root layout with metadata
│   ├── page.tsx            # Homepage
│   ├── servicios/
│   │   ├── page.tsx           # Services overview
│   │   └── [slug]/page.tsx     # Individual service pages
│   ├── contacto/page.tsx       # Contact form
│   ├── sobre-mi/page.tsx       # About page
│   ├── zona-servicio/
│   │   └── [city]/page.tsx     # City landing pages for SEO
│   └── api/
│       └── contact/route.ts    # Form submission endpoint
├── components/
│   ├── layout/
│   │   ├── Header.tsx          # Sticky header with CTA
│   │   ├── Footer.tsx          # Contact info, links, hours
│   │   └── MobileNav.tsx       # Hamburger menu
│   ├── ui/
│   │   ├── ServiceCard.tsx     # Service preview card
│   │   ├── TestimonialCard.tsx # Customer review
│   │   ├── ContactForm.tsx     # Lead capture form
│   │   ├── WhatsAppButton.tsx  # Floating WhatsApp widget
│   │   ├── CallButton.tsx      # Mobile sticky call button
│   │   ├── HeroSection.tsx     # Above-the-fold hero
│   │   └── TrustBadges.tsx     # Years experience, reviews, etc.
│   └── seo/
│       ├── LocalBusinessSchema.tsx  # JSON-LD structured data
│       └── generateMetadata.ts      # SEO metadata helpers
├── content/
│   ├── services.ts            # All service data
│   ├── cities.ts              # City landing page content
│   ├── testimonials.ts        # Customer reviews
│   └── business.ts            # Business info (phone, hours, etc.)
├── lib/
│   ├── email.ts               # Resend integration
│   └── utils.ts               # Helper functions
├── public/
│   ├── images/
│   │   ├── hero/
│   │   ├── services/
│   │   └── gallery/
│   ├── icons/
│   └── og-image.jpg           # Social sharing image
├── .env.local
├── next.config.js
├── tailwind.config.js
└── package.json
```

Content Data Structure

Business Info

typescript

```typescript
// content/business.ts
export const business = {
  name: "Arturo Morgadanes",
  title: "Fontanero Profesional",
  tagline: "Fontanero de confianza en Vigo y alrededores",
  phone: "+34 XXX XXX XXX",
  whatsapp: "34XXXXXXXXX", // Without + for WhatsApp API
  email: "info@arturomorgadanes.es",
  address: "Vigo, Pontevedra, Galicia",
  hours: {
    weekdays: "08:00 - 20:00",
    saturday: "09:00 - 14:00",
    sunday: "Urgencias 24h",
  },
  experience: 15, // Years
  jobsCompleted: 2000, // Approximate
  googleReviewScore: 4.9,
  googleReviewCount: 127,
  certifications: ["Carnet de instalador autorizado", "Seguro de responsabilidad civil"],
  serviceRadius: "30 km desde Vigo",
};
```

Services

typescript

```
// content/services.ts
export interface Service {
  slug: string;
  name: string;
  shortDescription: string; // For cards (max 120 chars)
  description: string;      // Full page content (300+ words)
  priceRange: string;
  duration: string;
  icon: string;             // Lucide icon name
  image: string;
  gallery?: string[];
  isEmergency?: boolean;
  faqs: Array<{ question: string; answer: string }>;
}

export const services: Service[] = [
  {
    slug: "desatascos",
    name: "Desatascos",
    shortDescription: "Limpieza y desatasco de tuberías, desagües y arquetas con equipos profesionales.",
    description: `
      Servicio profesional de desatascos en Vigo y alrededores. Utilizamos equipos
      de alta presión y cámaras de inspección para localizar y eliminar cualquier
      obstrucción en sus tuberías.

      ## ¿Qué incluye el servicio?
      - Diagnóstico inicial con cámara de inspección
      - Desatasco con máquina de alta presión
      - Limpieza completa de la tubería
      - Comprobación final del correcto funcionamiento

      ## Tipos de desatascos que realizamos
      - Fregaderos y lavabos
      - Inodoros y bidés
      - Bañeras y platos de ducha
      - Bajantes y arquetas
      - Tuberías de saneamiento

      Disponemos de servicio de urgencias 24 horas para desatascos que no pueden esperar.
    `,
    priceRange: "60€ - 150€",
    duration: "1-2 horas",
    icon: "Droplets",
    image: "/images/services/desatascos.jpg",
    gallery: ["/images/gallery/desatasco-1.jpg", "/images/gallery/desatasco-2.jpg"],
    faqs: [
      {
        question: "¿Cuánto tarda un desatasco?",
        answer: "La mayoría de desatascos se resuelven en 1-2 horas. Los casos más complejos pueden requerir más tiempo."
      },
      {
        question: "¿Utilizáis productos químicos?",
        answer: "No. Utilizamos equipos mecánicos y de alta presión que son más efectivos y respetuosos con las tuberías."
      }
    ]
  },
  {
    slug: "reparacion-fugas",
    name: "Reparación de Fugas",
    shortDescription: "Detección y reparación de fugas de agua en tuberías, grifos y cisternas.",
    description: `...`, // Full content
    priceRange: "40€ - 200€",
    duration: "1-3 horas",
    icon: "Wrench",
    image: "/images/services/fugas.jpg",
    faqs: [...]
  },
  {
    slug: "instalacion-griferia",
    name: "Instalación de Grifería",
    shortDescription: "Montaje y sustitución de grifos, duchas y sistemas de agua en cocina y baño.",
    priceRange: "30€ - 80€",
    duration: "30-60 min",
    icon: "Gauge",
    image: "/images/services/griferia.jpg",
    faqs: [...]
  },
  {
    slug: "cambio-calentador",
    name: "Cambio de Calentador",
    shortDescription: "Instalación y sustitución de termos eléctricos y calentadores de gas.",
    priceRange: "80€ - 250€",
    duration: "2-4 horas",
    icon: "Flame",
    image: "/images/services/calentador.jpg",
    faqs: [...]
  },
  {
    slug: "reforma-banos",
    name: "Reforma de Baños",
    shortDescription: "Renovación completa o parcial de baños: fontanería, sanitarios y acabados.",
    priceRange: "Desde 1.500€",
    duration: "3-7 días",
    icon: "Bath",
    image: "/images/services/reforma-bano.jpg",
    gallery: [...],
    faqs: [...]
  },
  {
    slug: "urgencias-24h",
    name: "Urgencias 24h",
    shortDescription: "Servicio de fontanería urgente disponible las 24 horas, todos los días.",
    priceRange: "Desde 80€",
    duration: "Llegada en 30-60 min",
    icon: "Siren",
    image: "/images/services/urgencias.jpg",
    isEmergency: true,
    faqs: [...]
  },
];
```

City Landing Pages (for Local SEO)

```
typescript
```

```
// content/cities.ts
export interface City {
  slug: string;
  name: string;
  province: string;
  postalCodes: string[];
  responseTime: string;
  localContent: string; // 500+ words, unique per city
  nearbyAreas: string[];
}

export const cities: City[] = [
  {
    slug: "vigo",
    name: "Vigo",
    province: "Pontevedra",
    postalCodes: ["36201", "36202", "36203", "36204", "36205", "36206", "36207", "36208", "36209", "36210"],
    responseTime: "30 minutos",
    localContent: `
      ## Fontanero en Vigo - Servicio Profesional

      Soy Arturo Morgadanes, fontanero profesional con más de 15 años de experiencia
      trabajando en Vigo y su área metropolitana. Ofrezco servicios de fontanería
      a domicilio en todos los barrios de Vigo...

      ### Zonas de Vigo donde trabajo
      - Casco Vello
      - Bouzas
      - Navia
      - Teis
      - Lavadores
      - Coruxo
      - Sárdoma
      - Castrelos

      ### ¿Por qué elegirme como su fontanero en Vigo?
      ...
    `,
    nearbyAreas: ["Nigrán", "Gondomar", "Redondela", "Mos"]
  },
  {
    slug: "pontevedra",
    name: "Pontevedra",
    province: "Pontevedra",
    postalCodes: ["36001", "36002", "36003", "36004", "36005"],
    responseTime: "45 minutos",
    localContent: `...`,
    nearbyAreas: ["Poio", "Marín", "Sanxenxo"]
  },
  // Add 8-10 more cities for comprehensive local coverage
];
```

Testimonials

typescript

```
// content/testimonials.ts
export const testimonials = [
  {
    name: "María García",
    location: "Vigo",
    service: "Desatascos",
    rating: 5,
    text: "Llegó en menos de una hora y solucionó el atasco del baño rápidamente. Muy profesional y buen precio.",
    date: "2024-11"
  },
  {
    name: "Carlos Rodríguez",
    location: "Pontevedra",
    service: "Cambio de calentador",
    rating: 5,
    text: "Me instaló el termo nuevo el mismo día. Trabajo limpio y ordenado. Lo recomiendo totalmente.",
    date: "2024-10"
  },
  // Add 6-10 testimonials
];
```

Core Components

Header with Sticky CTA

tsx

```
// components/layout/Header.tsx
"use client";

import { useState } from "react";
import Link from "next/link";
import { Phone, Menu, X } from "lucide-react";
import { business } from "@/content/business";
import { MobileNav } from "./MobileNav";

export function Header() {
  const [mobileMenuOpen, setMobileMenuOpen] = useState(false);

  const navLinks = [
    { href: "/", label: "Inicio" },
    { href: "/servicios", label: "Servicios" },
    { href: "/sobre-mi", label: "Sobre Mí" },
    { href: "/contacto", label: "Contacto" },
  ];

  return (
    <header className="sticky top-0 z-50 bg-white shadow-sm">
      <div className="max-w-6xl mx-auto px-4">
        <div className="flex items-center justify-between h-16 md:h-20">
          {/* Logo */}
          <Link href="/" className="flex items-center gap-2">
            <span className="text-xl md:text-2xl font-bold text-blue-700">
              {business.name}
            </span>
          </Link>

          {/* Desktop Navigation */}
          <nav className="hidden md:flex items-center gap-8">
            {navLinks.map((link) => (
              <Link
                key={link.href}
                href={link.href}
                className="text-gray-700 hover:text-blue-700 font-medium transition-colors"
              >
                {link.label}
              </Link>
            ))}
          </nav>
          {/* CTA Buttons */}
          <div className="flex items-center gap-3">
            <a
              href={`tel:${business.phone}`}
              className="hidden sm:flex items-center gap-2 bg-blue-700 text-white px-4 py-2 rounded-lg font-semibold hover:bg
            >
              <Phone className="w-4 h-4" />
              <span className="hidden lg:inline">{business.phone}</span>
              <span className="lg:hidden">Llamar</span>
            </a>

            {/* Mobile Menu Button */}
            <button
              onClick={() => setMobileMenuOpen(!mobileMenuOpen)}
              className="md:hidden p-2"
              aria-label="Menú"
            >
              {mobileMenuOpen ? <X className="w-6 h-6" /> : <Menu className="w-6 h-6" />}
            </button>
          </div>
        </div>
      </div>

      {/* Mobile Navigation */}
      <MobileNav isOpen={mobileMenuOpen} links={navLinks} onClose={() => setMobileMenuOpen(false)} />
    </header>
  );
}
```

Hero Section

tsx

```
// components/ui/HeroSection.tsx
import Link from "next/link";
import { Phone, MessageCircle, Clock, Shield, Star } from "lucide-react";
import { business } from "@/content/business";

export function HeroSection() {
  return (
    <section className="relative bg-gradient-to-br from-blue-900 to-blue-700 text-white">
      {/* Background image overlay */}
      <div
        className="absolute inset-0 bg-cover bg-center opacity-20"
        style={{ backgroundImage: "url('/images/hero/plumber-hero.jpg')" }}
      />

      <div className="relative max-w-6xl mx-auto px-4 py-16 md:py-24">
        <div className="max-w-2xl">
          {/* Trust badge */}
          <div className="flex items-center gap-2 text-blue-200 mb-4">
            <Star className="w-5 h-5 fill-yellow-400 text-yellow-400" />
            <span>{business.googleReviewScore} · {business.googleReviewCount} reseñas en Google</span>
          </div>

          <h1 className="text-4xl md:text-5xl lg:text-6xl font-bold mb-4">
            Fontanero Profesional en Vigo
          </h1>

          <p className="text-xl md:text-2xl text-blue-100 mb-8">
            {business.tagline}. Más de {business.experience} años de experiencia.
            Servicio de urgencias 24 horas.
          </p>

          {/* Trust indicators */}
          <div className="flex flex-wrap gap-4 mb-8 text-sm">
            <div className="flex items-center gap-2">
              <Clock className="w-5 h-5 text-green-400" />
              <span>Respuesta en 30 min</span>
            </div>
            <div className="flex items-center gap-2">
              <Shield className="w-5 h-5 text-green-400" />
              <span>Presupuesto sin compromiso</span>
            </div>
          </div>
          {/* CTA Buttons */}
          <div className="flex flex-col sm:flex-row gap-4">
            <a
              href={`tel:${business.phone}`}
              className="flex items-center justify-center gap-2 bg-white text-blue-700 px-8 py-4 rounded-lg font-bold text-lg hov
            >
              <Phone className="w-5 h-5" />
              Llamar Ahora
            </a>
            <a
              href={`https://wa.me/${business.whatsapp}?text=Hola, necesito un fontanero para...`}
              target="_blank"
              rel="noopener noreferrer"
              className="flex items-center justify-center gap-2 bg-green-500 text-white px-8 py-4 rounded-lg font-bold text-lg ho
            >
              <MessageCircle className="w-5 h-5" />
              WhatsApp
            </a>
          </div>
        </div>
      </div>
    </section>
  );
}
```

Service Card

tsx

```
// components/ui/ServiceCard.tsx
import Link from "next/link";
import Image from "next/image";
import { ArrowRight } from "lucide-react";
import * as Icons from "lucide-react";
import type { Service } from "@/content/services";

interface ServiceCardProps {
  service: Service;
}

export function ServiceCard({ service }: ServiceCardProps) {
  // Dynamically get icon component
  const IconComponent = Icons[service.icon as keyof typeof Icons] as React.ComponentType<{ className?: string }>;

  return (
    <Link
      href={`/servicios/${service.slug}`}
      className="group block bg-white rounded-xl shadow-sm border border-gray-100 overflow-hidden hover:shadow-lg tran
    >
      {/* Image */}
      <div className="relative h-48 bg-gray-100">
        <Image
          src={service.image}
          alt={service.name}
          fill
          className="object-cover group-hover:scale-105 transition-transform duration-300"
        />
        {service.isEmergency && (
          <div className="absolute top-3 right-3 bg-red-500 text-white text-xs font-bold px-2 py-1 rounded">
            24H
          </div>
        )}
      </div>

      {/* Content */}
      <div className="p-5">
        <div className="flex items-center gap-3 mb-2">
          {IconComponent && <IconComponent className="w-6 h-6 text-blue-600" />}
          <h3 className="text-lg font-bold text-gray-900">{service.name}</h3>
        </div>

        <p className="text-gray-600 text-sm mb-4 line-clamp-2">
```

```
          {service.shortDescription}
        </p>

        <div className="flex items-center justify-between">
          <span className="text-blue-700 font-semibold">{service.priceRange}</span>
          <span className="flex items-center gap-1 text-blue-600 text-sm font-medium group-hover:gap-2 transition-all">
            Ver más <ArrowRight className="w-4 h-4" />
          </span>
        </div>
      </div>
    </Link>
  );
}
```

Contact Form

```
tsx
```

```tsx
// components/ui/ContactForm.tsx
"use client";

import { useState } from "react";
import { Send, Loader2, CheckCircle } from "lucide-react";
import { services } from "@/content/services";

interface FormData {
  nombre: string;
  telefono: string;
  email: string;
  servicio: string;
  mensaje: string;
  urgente: boolean;
}

export function ContactForm() {
  const [formData, setFormData] = useState<FormData>({
    nombre: "",
    telefono: "",
    email: "",
    servicio: "",
    mensaje: "",
    urgente: false,
  });
  const [status, setStatus] = useState<"idle" | "loading" | "success" | "error">("idle");

  const handleSubmit = async (e: React.FormEvent) => {
    e.preventDefault();
    setStatus("loading");

    try {
      const response = await fetch("/api/contact", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify(formData),
      });

      if (response.ok) {
        setStatus("success");
        setFormData({ nombre: "", telefono: "", email: "", servicio: "", mensaje: "", urgente: false });
      } else {
        setStatus("error");
      }
    } catch {
      setStatus("error");
    }
  };

  if (status === "success") {
    return (
      <div className="bg-green-50 border border-green-200 rounded-xl p-8 text-center">
        <CheckCircle className="w-16 h-16 text-green-500 mx-auto mb-4" />
        <h3 className="text-xl font-bold text-green-800 mb-2">¡Mensaje enviado!</h3>
        <p className="text-green-700">
          Gracias por contactar. Le responderé lo antes posible.
        </p>
      </div>
    );
  }

  return (
    <form onSubmit={handleSubmit} className="space-y-5">
      {/* Name */}
      <div>
        <label htmlFor="nombre" className="block text-sm font-medium text-gray-700 mb-1">
          Nombre *
        </label>
        <input
          type="text"
          id="nombre"
          required
          value={formData.nombre}
          onChange={(e) => setFormData({ ...formData, nombre: e.target.value })}
          className="w-full px-4 py-3 rounded-lg border border-gray-300 focus:ring-2 focus:ring-blue-500 focus:border-transpa
          placeholder="Su nombre"
        />
      </div>

      {/* Phone */}
      <div>
        <label htmlFor="telefono" className="block text-sm font-medium text-gray-700 mb-1">
          Teléfono *
        </label>
        <input
          type="tel"
          id="telefono"
    required
    value={formData.telefono}
    onChange={(e) => setFormData({ ...formData, telefono: e.target.value })}
    className="w-full px-4 py-3 rounded-lg border border-gray-300 focus:ring-2 focus:ring-blue-500 focus:border-transpa
    placeholder="600 000 000"
  />
</div>

{/* Email */}
<div>
  <label htmlFor="email" className="block text-sm font-medium text-gray-700 mb-1">
    Email (opcional)
  </label>
  <input
    type="email"
    id="email"
    value={formData.email}
    onChange={(e) => setFormData({ ...formData, email: e.target.value })}
    className="w-full px-4 py-3 rounded-lg border border-gray-300 focus:ring-2 focus:ring-blue-500 focus:border-transpa
    placeholder="su@email.com"
  />
</div>

{/* Service */}
<div>
  <label htmlFor="servicio" className="block text-sm font-medium text-gray-700 mb-1">
    Servicio necesitado
  </label>
  <select
    id="servicio"
    value={formData.servicio}
    onChange={(e) => setFormData({ ...formData, servicio: e.target.value })}
    className="w-full px-4 py-3 rounded-lg border border-gray-300 focus:ring-2 focus:ring-blue-500 focus:border-transpa
  >
    <option value="">Seleccione un servicio</option>
    {services.map((s) => (
      <option key={s.slug} value={s.name}>{s.name}</option>
    ))}
    <option value="otro">Otro</option>
  </select>
</div>

{/* Message */}
<div>
```

```
  <label htmlFor="mensaje" className="block text-sm font-medium text-gray-700 mb-1">
    Descripción del problema
  </label>
  <textarea
    id="mensaje"
    rows={4}
    value={formData.mensaje}
    onChange={(e) => setFormData({ ...formData, mensaje: e.target.value })}
    className="w-full px-4 py-3 rounded-lg border border-gray-300 focus:ring-2 focus:ring-blue-500 focus:border-transpa
    placeholder="Describa brevemente su problema o lo que necesita..."
  />
</div>

{/* Urgent checkbox */}
<div className="flex items-center gap-3">
  <input
    type="checkbox"
    id="urgente"
    checked={formData.urgente}
    onChange={(e) => setFormData({ ...formData, urgente: e.target.checked })}
    className="w-5 h-5 text-red-600 rounded focus:ring-red-500"
  />
  <label htmlFor="urgente" className="text-sm text-gray-700">
    <span className="font-medium text-red-600">Es urgente</span> - Necesito atención lo antes posible
  </label>
</div>

{/* Error message */}
{status === "error" && (
  <p className="text-red-600 text-sm">
    Error al enviar el mensaje. Por favor, intente de nuevo o llámenos directamente.
  </p>
)}

{/* Submit */}
<button
  type="submit"
  disabled={status === "loading"}
  className="w-full flex items-center justify-center gap-2 bg-blue-700 text-white px-6 py-4 rounded-lg font-bold text-lg
>
  {status === "loading" ? (
    <>
      <Loader2 className="w-5 h-5 animate-spin" />
      Enviando...
```

```
          </>
        ) : (
          <>
            <Send className="w-5 h-5" />
            Enviar Mensaje
          </>
        )}
      </button>
    </form>
  );
}
```

WhatsApp Floating Button

tsx

```tsx
// components/ui/WhatsAppButton.tsx
"use client";

import { MessageCircle } from "lucide-react";
import { business } from "@/content/business";

export function WhatsAppButton() {
  const message = encodeURIComponent("Hola, necesito un fontanero para...");
  const url = `https://wa.me/${business.whatsapp}?text=${message}`;

  return (
    <a
      href={url}
      target="_blank"
      rel="noopener noreferrer"
      className="fixed bottom-6 right-6 z-50 flex items-center gap-2 bg-green-500 text-white px-4 py-3 rounded-full shadow-
      aria-label="Contactar por WhatsApp"
    >
      <MessageCircle className="w-6 h-6" />
      <span className="hidden sm:inline font-medium">WhatsApp</span>
    </a>
  );
}
```

Mobile Sticky Call Button

tsx

```tsx
// components/ui/CallButton.tsx
"use client";

import { Phone } from "lucide-react";
import { business } from "@/content/business";

export function CallButton() {
  return (
    <a
      href={`tel:${business.phone}`}
      className="fixed bottom-6 left-6 z-50 sm:hidden flex items-center gap-2 bg-blue-700 text-white px-4 py-3 rounded-full
      aria-label="Llamar ahora"
    >
      <Phone className="w-6 h-6" />
    </a>
  );
}
```

Local Business Schema (SEO)

tsx

```
// components/seo/LocalBusinessSchema.tsx
import { business } from "@/content/business";

export function LocalBusinessSchema() {
  const schema = {
    "@context": "https://schema.org",
    "@type": "Plumber",
    "name": business.name,
    "description": business.tagline,
    "telephone": business.phone,
    "email": business.email,
    "address": {
      "@type": "PostalAddress",
      "addressLocality": "Vigo",
      "addressRegion": "Pontevedra",
      "addressCountry": "ES"
    },
    "geo": {
      "@type": "GeoCoordinates",
      "latitude": "42.2406", // Vigo coordinates
      "longitude": "-8.7207"
    },
    "areaServed": {
      "@type": "GeoCircle",
      "geoMidpoint": {
        "@type": "GeoCoordinates",
        "latitude": "42.2406",
        "longitude": "-8.7207"
      },
      "geoRadius": "30000" // 30km in meters
    },
    "openingHoursSpecification": [
      {
        "@type": "OpeningHoursSpecification",
        "dayOfWeek": ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday"],
        "opens": "08:00",
        "closes": "20:00"
      },
      {
        "@type": "OpeningHoursSpecification",
        "dayOfWeek": "Saturday",
        "opens": "09:00",
        "closes": "14:00"
      }
    ],
    "aggregateRating": {
      "@type": "AggregateRating",
      "ratingValue": business.googleReviewScore,
      "reviewCount": business.googleReviewCount
    },
    "priceRange": "€€"
  };

  return (
    <script
      type="application/ld+json"
      dangerouslySetInnerHTML={{ __html: JSON.stringify(schema) }}
    />
  );
}
```

API Route: Contact Form

```
typescript
```

```
// app/api/contact/route.ts
import { NextResponse } from "next/server";
import { Resend } from "resend";
import { business } from "@/content/business";

const resend = new Resend(process.env.RESEND_API_KEY);

// Simple in-memory rate limiting (use Redis in production for multi-instance)
const rateLimit = new Map<string, { count: number; timestamp: number }>();
const RATE_LIMIT_WINDOW = 60 * 60 * 1000; // 1 hour
const RATE_LIMIT_MAX = 5;

function isRateLimited(ip: string): boolean {
  const now = Date.now();
  const record = rateLimit.get(ip);

  if (!record || now - record.timestamp > RATE_LIMIT_WINDOW) {
    rateLimit.set(ip, { count: 1, timestamp: now });
    return false;
  }

  if (record.count >= RATE_LIMIT_MAX) {
    return true;
  }

  record.count++;
  return false;
}

export async function POST(request: Request) {
  // Get IP for rate limiting
  const ip = request.headers.get("x-forwarded-for") || "unknown";

  if (isRateLimited(ip)) {
    return NextResponse.json(
      { error: "Demasiadas solicitudes. Intente de nuevo más tarde." },
      { status: 429 }
    );
  }

  try {
    const body = await request.json();
    const { nombre, telefono, email, servicio, mensaje, urgente } = body;
    // Validation
    if (!nombre || !telefono) {
      return NextResponse.json(
        { error: "Nombre y teléfono son obligatorios" },
        { status: 400 }
      );
    }

    // Send email notification
    await resend.emails.send({
      from: "Web <noreply@arturomorgadanes.es>",
      to: business.email,
      subject: urgente
        ? `🚨 URGENTE: Nueva solicitud de ${nombre}`
        : `Nueva solicitud de ${nombre}`,
      html: `
        <h2>Nueva solicitud desde la web</h2>
        ${urgente ? '<p style="color: red; font-weight: bold;">⚠️ CLIENTE INDICA URGENCIA</p>' : ''}
        <p><strong>Nombre:</strong> ${nombre}</p>
        <p><strong>Teléfono:</strong> <a href="tel:${telefono}">${telefono}</a></p>
        ${email ? `<p><strong>Email:</strong> ${email}</p>` : ''}
        ${servicio ? `<p><strong>Servicio:</strong> ${servicio}</p>` : ''}
        ${mensaje ? `<p><strong>Mensaje:</strong></p><p>${mensaje}</p>` : ''}
        <hr>
        <p style="color: gray; font-size: 12px;">
          Enviado desde arturomorgadanes.es el ${new Date().toLocaleString('es-ES')}
        </p>
      `,
    });

    return NextResponse.json({ success: true });

  } catch (error) {
    console.error("Contact form error:", error);
    return NextResponse.json(
      { error: "Error al enviar el mensaje" },
      { status: 500 }
    );
  }
}
```

SEO Configuration

Root Layout with Metadata

tsx

```
// app/layout.tsx
import type { Metadata } from "next";
import { Inter } from "next/font/google";
import { Header } from "@/components/layout/Header";
import { Footer } from "@/components/layout/Footer";
import { WhatsAppButton } from "@/components/ui/WhatsAppButton";
import { CallButton } from "@/components/ui/CallButton";
import { LocalBusinessSchema } from "@/components/seo/LocalBusinessSchema";
import { business } from "@/content/business";
import "./globals.css";

const inter = Inter({ subsets: ["latin"] });

export const metadata: Metadata = {
  metadataBase: new URL("https://arturomorgadanes.es"),
  title: {
    default: `${business.name} - Fontanero en Vigo | Urgencias 24h`,
    template: `%s | ${business.name}`,
  },
  description: `${business.tagline}. Más de ${business.experience} años de experiencia. Desatascos, fugas, reformas de baños
  keywords: [
    "fontanero vigo",
    "fontanero urgente vigo",
    "desatascos vigo",
    "fontanería vigo",
    "reparación fugas vigo",
    "fontanero pontevedra",
    "fontanero 24 horas vigo",
  ],
  openGraph: {
    type: "website",
    locale: "es_ES",
    siteName: business.name,
    images: [{ url: "/og-image.jpg", width: 1200, height: 630 }],
  },
  robots: {
    index: true,
    follow: true,
  },
  verification: {
    google: "YOUR_GOOGLE_VERIFICATION_CODE",
  },
};
```

```
export default function RootLayout({ children }: { children: React.ReactNode }) {
  return (
    <html lang="es">
      <head>
        <LocalBusinessSchema />
      </head>
      <body className={inter.className}>
        <Header />
        <main>{children}</main>
        <Footer />
        <WhatsAppButton />
        <CallButton />
      </body>
    </html>
  );
}
```

Service Page with Dynamic Metadata

tsx

```tsx
// app/servicios/[slug]/page.tsx
import { Metadata } from "next";
import { notFound } from "next/navigation";
import Image from "next/image";
import { services } from "@/content/services";
import { business } from "@/content/business";
import { ContactForm } from "@/components/ui/ContactForm";

interface Props {
  params: { slug: string };
}

export async function generateStaticParams() {
  return services.map((service) => ({ slug: service.slug }));
}

export async function generateMetadata({ params }: Props): Promise<Metadata> {
  const service = services.find((s) => s.slug === params.slug);
  if (!service) return {};

  return {
    title: `${service.name} en Vigo - ${service.priceRange}`,
    description: `${service.shortDescription} Servicio profesional de ${service.name.toLowerCase()} en Vigo y alrededores. $
  };
}

export default function ServicePage({ params }: Props) {
  const service = services.find((s) => s.slug === params.slug);
  if (!service) notFound();

  return (
    <article className="max-w-4xl mx-auto px-4 py-12">
      {/* Hero */}
      <div className="relative h-64 md:h-96 rounded-xl overflow-hidden mb-8">
        <Image
          src={service.image}
          alt={service.name}
          fill
          className="object-cover"
          priority
        />
      </div>
      {/* Content */}
      <h1 className="text-3xl md:text-4xl font-bold mb-4">{service.name}</h1>

      <div className="flex flex-wrap gap-4 mb-8">
        <span className="bg-blue-100 text-blue-800 px-4 py-2 rounded-lg font-semibold">
          {service.priceRange}
        </span>
        <span className="bg-gray-100 text-gray-700 px-4 py-2 rounded-lg">
          ⏱️ {service.duration}
        </span>
      </div>

      <div
        className="prose prose-lg max-w-none mb-12"
        dangerouslySetInnerHTML={{ __html: service.description }}
      />

      {/* FAQs */}
      {service.faqs.length > 0 && (
        <section className="mb-12">
          <h2 className="text-2xl font-bold mb-6">Preguntas Frecuentes</h2>
          <div className="space-y-4">
            {service.faqs.map((faq, i) => (
              <details key={i} className="bg-gray-50 rounded-lg p-4 group">
                <summary className="font-semibold cursor-pointer list-none flex justify-between items-center">
                  {faq.question}
                  <span className="text-blue-600 group-open:rotate-180 transition-transform">▼</span>
                </summary>
                <p className="mt-3 text-gray-700">{faq.answer}</p>
              </details>
            ))}
          </div>
        </section>
      )}

      {/* Contact Form */}
      <section className="bg-gray-50 rounded-xl p-6 md:p-8">
        <h2 className="text-2xl font-bold mb-6">Solicitar Presupuesto</h2>
        <ContactForm />
      </section>
    </article>
  );
}
```

Environment Variables

env

```
# .env.local

# Email (Resend)
RESEND_API_KEY=re_XXXXXXXXXX

# Analytics
NEXT_PUBLIC_GA_ID=G-XXXXXXXXXX

# Optional: Cal.com for scheduling
NEXT_PUBLIC_CAL_LINK=https://cal.com/arturo-morgadanes
```

Deployment

Vercel Configuration

json

```
// vercel.json
{
  "regions": ["cdg1"], // Paris - closest to Spain
  "headers": [
    {
      "source": "/(.*)",
      "headers": [
        { "key": "X-Content-Type-Options", "value": "nosniff" },
        { "key": "X-Frame-Options", "value": "DENY" },
        { "key": "X-XSS-Protection", "value": "1; mode=block" }
      ]
    }
  ]
}
```

next.config.js

javascript

```
/** @type {import('next').NextConfig} */
const nextConfig = {
  output: 'export', // Static export for maximum performance
  images: {
    unoptimized: true, // Required for static export
  },
  trailingSlash: true,
};

module.exports = nextConfig;
```

Getting Started

bash

```bash
# Create project
npx create-next-app@latest fontanero-vigo --typescript --tailwind --app --src-dir=false

cd fontanero-vigo

# Install dependencies
npm install lucide-react resend

# Create directory structure
mkdir -p content components/{layout,ui,seo} public/images/{hero,services,gallery}

# Start development
npm run dev
```

Pre-Launch Checklist

Content

- □ All 6 service pages have 300+ word descriptions
- □ 8-10 testimonials with real names and dates
- □ Before/after gallery images (at least 6)
- □ Professional hero image
- □ Service images (one per service)
- □ Business hours verified

□ Phone number and WhatsApp number correct

SEO

□ Google My Business profile claimed and optimized
□ Google Search Console configured
□ Sitemap submitted
□ All pages have unique meta descriptions
□ Schema.org LocalBusiness markup validated
□ Mobile PageSpeed score > 90
□ All images have alt text in Spanish

Functionality

□ Contact form sends emails correctly
□ WhatsApp button opens correct chat
□ Phone links work on mobile
□ Site works without JavaScript (forms excluded)
□ 404 page exists and is helpful

Legal (GDPR)

□ Privacy policy page in Spanish
□ Cookie consent banner
□ Contact form includes data usage notice

Cost Summary

Item	Monthly Cost
Vercel Hosting	€0 (free tier)
Domain (.es)	~€1 (€10/year)
Resend Email	€0 (free tier: 3,000 emails/month)
Total	**~€1/month**

Optional additions:

- Cal.com Pro (scheduling): €12/month

- Google Workspace: €6/month

Maintenance

What the plumber needs to do: Nothing day-to-day.

What needs occasional updates (developer task, ~1 hour/year):

- Add new testimonials
- Update prices
- Add new service photos
- Update business hours

The site is designed to run maintenance-free. Content is hardcoded because it rarely changes. When it does, a simple code change and deploy takes 5 minutes.