# 🚀 Guía Completa: Del Desarrollo a Producción

## Menú Digital para Restaurante - Paso a Paso

**Fecha:** 2 de febrero de 2026
**Objetivo:** Poner el proyecto en internet con dominio propio y acceso público

---

## 📋 ÍNDICE



---

## 1️⃣ PREPARAR LA APLICACIÓN

### 1.1 Optimizar Frontend

```
# En terminal, carpeta frontend
cd d:\proyectos\restaurante-menu\frontend

# Instalar dependencias (ya debería estar hecho)
npm install

# Crear build de producción
npm run build

# Resultado: Carpeta 'dist/' lista para desplegar
```

**Qué hace:**

- Minimiza el código
- Optimiza imágenes
- Crea archivos estáticos listos para servir
- Tamaño final: ~100KB (muy rápido)

### 1.2 Backend Listo

```
# En terminal, carpeta backend
cd d:\proyectos\restaurante-menu\backend

# Instalar dependencias
npm install
```

**Archivos necesarios:**

Crear `backend/.env.production`:

```
PORT=3000
NODE_ENV=production
JWT_SECRET=tu_clave_secreta_segura_aqui
```

Crear `backend/.env.local`:

```
PORT=3000
NODE_ENV=development
JWT_SECRET=desarrollo_secret
```

### 1.3 Actualizar URLs del Frontend

**Archivo:** `frontend/src/services/api.js`

```
// ACTUALIZAR ESTA SECCIÓN:

// Desarrollo
const API_DEV = 'http://localhost:3000';

// Producción
const API_PROD = process.env.VITE_API_URL || 'https://api.tunombre.com';

export const SERVER_URL =
  process.env.NODE_ENV === 'production'
    ? API_PROD
    : API_DEV;
```

**Crear archivo:** `frontend/.env.production`

```
VITE_API_URL=https://api.tunombre.com
```

---

# 2 MIGRAR A BASE DE DATOS REAL

## 2.1 Elegir Base de Datos

**Opción A: MongoDB Atlas (RECOMENDADO - Más simple)**

- Gratis (hasta 512MB)
- Cloud (no necesita servidor)
- Mejor para empezar

**Opción B: MySQL (Tradicional)**

- Más universal
- Mejor si escalas mucho
- Requiere hosting adicional

## 2.2 Configurar MongoDB Atlas

**Paso 1: Crear Cuenta**

1. Ve a https://www.mongodb.com/cloud/atlas
2. Click "Try Free"
3. Crea cuenta con email
4. Confirma email

**Paso 2: Crear Cluster**

1. Haz click "Create a Deployment"
2. Selecciona "M0 Free" (gratis)
3. Cloud: AWS, Región: N. Virginia
4. Click "Create"
5. Espera 2-3 minutos

**Paso 3: Crear Usuario**

1. En la izquierda, click "Database Access"
2. Click "Add New Database User"
3. Username: `admin_restaurante`
4. Password: `GenererPasswordSegura123!`
5. Click "Add User"

**Paso 4: Obtener Connection String**

1. Vuelve a "Database Deployment"
2. Click "Connect"
3. Selecciona "Drivers"
4. Copia la URL (parece: `mongodb+srv://...`)
5. Reemplaza `<password>` con tu contraseña

**Paso 5: Crear Database**

1. Vuelve a Dashboard
2. Click "Browse Collections"
3. Click "Add My Own Data"

4. Database Name: `restaurante_menu`
5. Collection Name: `users`
6. Click "Create"

---

### 2.3 Instalar MongoDB Driver en Backend

```
cd d:\proyectos\restaurante-menu\backend

# Instalar dependencias MongoDB
npm install mongoose mongodb

# Instalar bcryptjs para contraseñas
npm install bcryptjs
```

---

### 2.4 Reescribir Backend para MongoDB

**Archivo:** `backend/db.js` (CREAR NUEVO)

```
import mongoose from 'mongoose';
import dotenv from 'dotenv';

dotenv.config();

const MONGODB_URL = process.env.MONGODB_URL || 'mongodb://localhost:27017/restaurante_menu';

export const connectDB = async () => {
  try {
    await mongoose.connect(MONGODB_URL, {
      useNewUrlParser: true,
      useUnifiedTopology: true,
    });
    console.log('✓ MongoDB conectado');
  } catch (error) {
    console.error('✗ Error conectando MongoDB:', error);
    process.exit(1);
  }
};

export default mongoose;
```

**Modelos MongoDB:**

**Archivo:** `backend/models/User.js`

```
import mongoose from 'mongoose';
import bcryptjs from 'bcryptjs';

const userSchema = new mongoose.Schema({
  usuario: { type: String, required: true, unique: true },
  contraseña: { type: String, required: true },
  role: { type: String, enum: ['admin', 'cliente'], default: 'cliente' },
  createdAt: { type: Date, default: Date.now }
});

userSchema.pre('save', async function(next) {
  if (!this.isModified('contraseña')) return next();
  this.contraseña = await bcryptjs.hash(this.contraseña, 10);
  next();
});

export default mongoose.model('User', userSchema);
```

**Archivo:** `backend/models/Category.js`

```
import mongoose from 'mongoose';

const subcategorySchema = new mongoose.Schema({
  id: String,
  name: String,
  icon: String
});

const categorySchema = new mongoose.Schema({
  name: { type: String, required: true },
  description: String,
  icon: String,
  image: String,
  subcategories: [subcategorySchema],
  order: { type: Number, default: 0 },
  createdAt: { type: Date, default: Date.now }
});

export default mongoose.model('Category', categorySchema);
```

**Archivo:** backend/models/Product.js

```
import mongoose from 'mongoose';

const productSchema = new mongoose.Schema({
  name: { type: String, required: true },
  description: String,
  price: { type: Number, required: true },
  categoryId: String,
  subcategoryId: String,
  image: String,
  available: { type: Boolean, default: true },
  createdAt: { type: Date, default: Date.now },
  updatedAt: { type: Date, default: Date.now }
});

export default mongoose.model('Product', productSchema);
```

---

### 2.5 Actualizar server.js

**Archivo:** backend/server.js

```
import express from 'express';
import cors from 'cors';
import dotenv from 'dotenv';
import { connectDB } from './db.js';
import authRoutes from './routes/auth.js';
import categoryRoutes from './routes/categories.js';
import productRoutes from './routes/products.js';

dotenv.config();

const app = express();
const PORT = process.env.PORT || 3000;

// Conectar a MongoDB
connectDB();

// Middlewares
app.use(cors({
  origin: [
    'http://localhost:5174',
    'http://localhost:3000',
    process.env.FRONTEND_URL || 'https://app.tunombre.com'
  ],
  credentials: true
}));

app.use(express.json());
app.use(express.urlencoded({ extended: true }));

// Servir archivos estáticos
app.use('/uploads', express.static('uploads'));
app.use('/images', express.static('images'));
app.use('/assets', express.static('assets'));

// Rutas
app.use('/api/auth', authRoutes);
app.use('/api/categories', categoryRoutes);
app.use('/api/products', productRoutes);

// Ruta de prueba
app.get('/', (req, res) => {
  res.json({ message: 'API Menú Digital - Restaurante' });
});

app.listen(PORT, () => {
  console.log(`✓ Servidor corriendo en puerto ${PORT}`);
});
```

---

### 2.6 Crear Variables de Entorno

**Archivo:** `backend/.env`

```
PORT=3000
NODE_ENV=development
MONGODB_URL=mongodb+srv://admin_restaurante:password@cluster.mongodb.net/restaurante_menu
JWT_SECRET=tu_clave_super_secreta_aqui_cambiar_en_produccion
FRONTEND_URL=http://localhost:5174
```

**Archivo:** `backend/.env.production`

```
PORT=3000
NODE_ENV=production
MONGODB_URL=mongodb+srv://admin_restaurante:password@cluster.mongodb.net/restaurante_menu
JWT_SECRET=GenerarConUnValorSeguroqui123!@#
FRONTEND_URL=https://app.tunombre.com
```

---

### 2.7 Importar Datos Existentes (JSON a MongoDB)

**Crear archivo:** `backend/scripts/importData.js`

```
import mongoose from 'mongoose';
import fs from 'fs';
import dotenv from 'dotenv';
import User from '../models/User.js';
import Category from '../models/Category.js';
import Product from '../models/Product.js';

dotenv.config();

const importData = async () => {
  try {
    await mongoose.connect(process.env.MONGODB_URL);

    console.log('Importando usuarios...');
    const usersData = JSON.parse(fs.readFileSync('data/users.json'));
    await User.insertMany(usersData);

    console.log('Importando categorías...');
    const categoriesData = JSON.parse(fs.readFileSync('data/categories.json'));
    await Category.insertMany(categoriesData);

    console.log('Importando productos...');
    const productsData = JSON.parse(fs.readFileSync('data/products.json'));
    await Product.insertMany(productsData);

    console.log('✓ Datos importados correctamente');
    process.exit(0);
  } catch (error) {
    console.error('✗ Error importando:', error);
    process.exit(1);
  }
};

importData();
```

**Ejecutar importación:**

```
cd backend
node scripts/importData.js
```

---

# 3 DESPLEGAR BACKEND

## 3.1 Crear Cuenta en Railway.app

1. Ve a https://railway.app
2. Click "Sign up with GitHub" (recomendado)
3. Autoriza Railway
4. ¡Listo!

## 3.2 Desplegar Backend en Railway

**Opción A: Desde GitHub (MEJOR)**

```
# Iniciarlizar git en el proyecto
cd d:\proyectos\restaurante-menu
git init
git add .
git commit -m "Initial commit"

# Subir a GitHub
# (Crear repositorio en github.com y pushear)
```

En Railway:

1. Click "New Project"
2. Click "Deploy from GitHub repo"
3. Selecciona tu repositorio

4. Selecciona la rama `main`
5. Click "Deploy"

**Opción B: Desde CLI (RÁPIDO)**

```
# Instalar Railway CLI
npm install -g @railway/cli

# Login
railway login

# En carpeta backend
cd d:\proyectos\restaurante-menu\backend

# Inicializar
railway init

# Agregar variables de entorno
railway service add

# Desplegar
railway up
```

### 3.3 Configurar Variables en Railway

En el dashboard de Railway:

1. Selecciona tu proyecto
2. Click "Variables"
3. Agrega:

```
MONGODB_URL=mongodb+srv://...
JWT_SECRET=tu_clave_segura_aqui
FRONTEND_URL=https://app.tunombre.com
```

### 3.4 Obtener URL del Backend

Railway te da una URL automática:

```
https://backend-prod-xxxxx.railway.app
```

**Copia esta URL, la necesitarás después**

---

## 4 DESPLEGAR FRONTEND

### 4.1 Crear Cuenta en Vercel

1. Ve a https://vercel.com
2. Click "Sign Up"
3. Elige "GitHub"
4. Autoriza Vercel

### 4.2 Desplegar Frontend

**En Vercel Dashboard:**

1. Click "New Project"
2. Selecciona tu repositorio de GitHub
3. Selecciona carpeta: `frontend`
4. Click "Deploy"

### 4.3 Configurar Variables en Vercel

En "Settings" → "Environment Variables":

```
VITE_API_URL=https://backend-prod-xxxxx.railway.app
```

Vercel automáticamente hace rebuild y redeploya.

### 4.4 Obtener URL del Frontend

Vercel te da una URL:

```
https://tu-app.vercel.app
```

**Copia esta URL, la necesitarás después**

---

# 5 CONFIGURAR DOMINIO

## 5.1 Comprar Dominio

**Opciones:**

- Namecheap: https://www.namecheap.com (~$10/año)
- Google Domains: https://domains.google (~$12/año)
- Godaddy: https://www.godaddy.com (~$15/año)

**Pasos:**

1. Busca tu nombre: `tunombre.com`
2. Click "Add to Cart"
3. Completa checkout
4. ¡Tendrás acceso al panel de DNS!

## 5.2 Conectar Dominio a Vercel (Frontend)

**En Vercel:**

1. Proyecto → Settings
2. "Domains"
3. Click "Add"
4. Ingresa: `tunombre.com`
5. Vercel te da instrucciones

**En tu registrador (Namecheap, etc):**

1. Ve a "Manage Domain"
2. Sección "Nameservers"
3. Cambia a los nameservers de Vercel
4. Espera 24-48 horas (puede ser inmediato)

## 5.3 Conectar Subdominio a Railway (Backend)

**En Railway:**

1. Proyecto → Settings
2. "Domains"
3. Click "Add Custom Domain"
4. Ingresa: `api.tunombre.com`

**En tu registrador:**

1. Agrega registro CNAME:
    - Name: `api`
    - Value: `[valor que te da Railway]`
2. Guarda

**O usar A Record (si es necesario):**

- Name: `api`
- Type: A
- Value: `[IP de Railway]`

## 5.4 Verificación

```
# Espera 24 horas, luego prueba:

# Frontend debería abrir
https://tunombre.com

# Backend debería responder
curl https://api.tunombre.com
# Respuesta: {"message":"API Menú Digital"}
```

---

## 6 VERIFICACIÓN FINAL

### 6.1 Checklist Pre-Launch

```
✓ Backend en Railway
  - Variables de entorno configuradas
  - MongoDB conectando correctamente
  - URL funciona: https://api.tunombre.com

✓ Frontend en Vercel
  - Build exitoso
  - VITE_API_URL apunta al backend correcto
  - URL funciona: https://tunombre.com

✓ Dominio
  - DNS configurado
  - HTTPS funcionando (🔒 candado verde)
  - Apunta correctamente

✓ Base de Datos
  - MongoDB Atlas conexión exitosa
  - Datos importados correctamente
  - Backups configurados (MongoDB Atlas automático)

✓ Funcionalidad
  - Puedo loguearme: admin/admin123
  - Ver menú: funciona
  - Admin panel: funciona
  - Crear producto: funciona
  - Los datos se guardan en BD
```

### 6.2 Pruebas Finales

```
# Test 1: Ver si Frontend carga
curl https://tunombre.com

# Test 2: Ver si Backend responde
curl https://api.tunombre.com/api/auth/me

# Test 3: Ver si puedo loguearme
curl -X POST https://api.tunombre.com/api/auth/login `
  -H "Content-Type: application/json" `
  -d '{"usuario":"admin","contraseña":"admin123"}'
```

### 6.3 Compartir con Clientes

```
¡Tu menú está online! 🎉

Acceso Cliente:
https://tunombre.com

Acceso Admin:
https://tunombre.com/login
Usuario: admin
Contraseña: admin123

(¡Cambiar contraseña después!)
```

---

## 7️⃣ MANTENIMIENTO

### 7.1 Backups Automáticos

**MongoDB Atlas:**

- Backups automáticos cada 12 horas
- Almacenados 7 días
- Ve a Atlas Dashboard → Backups

### 7.2 Monitoreo

**Railway:**

- Dashboard → Logs
- Revisa errores regularmente

**Vercel:**

- Analytics automático
- Puedes ver performance

### 7.3 Actualizar la Aplicación

**Cuando cambies código:**

```
# Backend (git push)
git add .
git commit -m "Descripción del cambio"
git push origin main
# Railway redeploya automáticamente

# Frontend (git push)
git add .
git commit -m "Descripción del cambio"
git push origin main
# Vercel redeploya automáticamente
```

### 7.4 Cambiar Credenciales

**En MongoDB:**

1. Ve a Database Access
2. Edit usuario
3. Genera nueva password
4. Actualiza .env en Railway

**En la App:**

1. Loguéate como admin
2. Cambia contraseña en BD

---

## 🎯 RESUMEN LÍNEA DE TIEMPO

```
DÍA 1:
☐ Crear cuenta MongoDB Atlas (5 min)
☐ Crear cuenta Railway.app (5 min)
☐ Crear cuenta Vercel.com (5 min)
☐ Comprar dominio (5 min)
☐ Importar datos a MongoDB (10 min)

DÍA 2:
☐ Desplegar Backend en Railway (5 min)
☐ Desplegar Frontend en Vercel (5 min)
☐ Configurar dominio (10 min)
☐ Pruebas finales (10 min)
☐ ¡LANZAMIENTO! 🚀

ESPERAR 24-48h por DNS
```

---

## ❓ PREGUNTAS FRECUENTES

**¿Cuesta dinero?**

- MongoDB: Gratis (512MB)
- Railway: $5/mes mínimo para backend
- Vercel: Gratis para frontend
- Dominio: ~$10/año
- **Total: ~$70/año para empezar**

**¿Qué pasa si necesito más datos?**

- MongoDB: Upgradea a tier pagado ($10/mes)
- Railway: Aumenta límites según uso

**¿Cómo agrego clientes nuevos?**

- Edita `backend/data/users.json` en MongoDB
- O crea API para registro de clientes

**¿Cómo actualizo el menú desde producción?**

- Loguéate en admin: https://tunombre.com/admin
- Crea/edita productos
- Se guardan automáticamente en MongoDB

**¿Se puede usar HTTPS?**

- Vercel: Automático ✓
- Railway: Automático ✓
- Ya está asegurado 🔒

---

## 📞 SOPORTE

Si algo no funciona:

1. **Revisa logs:**
    - Railway: Dashboard → Logs
    - Vercel: Deployments → Logs
2. **Verifica variables de entorno:**
    - ¿MONGODB_URL es correcta?
    - ¿JWT_SECRET configurado?
3. **Prueba en local primero:**
    - `npm run dev` en backend y frontend
    - Verifica que funcione antes de pusear

---

**¡Felicidades! Tu app está online y accesible para todos! 🎉**

Ahora puedes:

- Compartir el link con clientes
- Agregar más productos
- Recibir pedidos
- Crecer tu negocio

**¡Buena suerte! 🚀**