

Manual Técnico y de Despliegue

Sistema de Cotizaciones y Facturación (CotizaPro)

Generado: 24/04/2026

1. Tecnologías del Sistema

El sistema **CotizaPro** ha sido desarrollado utilizando un stack tecnológico moderno, robusto y escalable, basado en PHP y el ecosistema Laravel.

- **Backend Framework:** Laravel 11.x (PHP 8.3+)
- **Arquitectura:** MVC (Modelo-Vista-Controlador)
- **Motor de Base de Datos:** MySQL (recomendado para producción) / SQLite (soporte para desarrollo).
- **Frontend:** Blade Templates (Motor de plantillas de Laravel).
- **Estilos (CSS):** CSS nativo / variables CSS, optimizado sin dependencias pesadas de frameworks UI para mantenerlo ligero y personalizable.
- **JavaScript:** Vanilla JS y Chart.js para visualización de gráficos en el Dashboard.
- **Generación de PDFs:** `barryvdh/laravel-dompdf`.
- **Gestor de Dependencias:** Composer (PHP) y NPM (Frontend Assets / Vite).

2. Funcionalidades Principales

El sistema es un ERP simplificado centrado en la gestión ágil de cotizaciones comerciales.

Módulo	Descripción / Funcionalidad
Dashboard	Panel de control con métricas en tiempo real: total de cotizaciones emitidas/aprobadas, ingresos totales, ticket promedio, tasa de aprobación. Gráficos de ingresos mensuales y distribución por estado.
Clientes	CRUD completo de clientes. Historial de cotizaciones por cliente. Exportación de listados a Excel (CSV) y PDF.
Productos	Catálogo de productos o servicios. Precios y unidades configurables. Exportación a Excel y PDF.
Empresas	Gestión de empresas o entidades asociadas. Exportación a Excel y PDF.
Cotizaciones	Creación, edición y eliminación de cotizaciones. Cálculo automático de subtotales, descuentos e IGV. Conversión de estados (Borrador, Emitida, Aprobada, Rechazada). Duplicación rápida (Clonar). Generación y descarga de PDF con formato profesional. Envío directo por correo electrónico. Exportación de lista a Excel y PDF.

Módulo	Descripción / Funcionalidad
Reportes	Análisis detallado por año. KPIs anuales, desglose mensual y top de clientes. Exportación de resultados a Excel y PDF.
Configuración	Parámetros globales: Moneda por defecto (PEN, USD, EUR), IGV, logotipo de la empresa emisora, datos fiscales, términos y condiciones predeterminados, y credenciales SMTP para envíos de correo.
Mantenimiento	Generación de copias de seguridad (Backups SQL) y restauración de la base de datos completa directamente desde la interfaz web.

3. Instalación Local (Paso a Paso)

Esta guía asume que utilizarás un entorno de desarrollo local como **XAMPP**, **Laragon** o **Herd** en Windows.

Paso 1: Requisitos Previos

Asegúrate de tener instalados los siguientes programas:

- **PHP:** Versión 8.3 o superior.
- **Composer:** Gestor de dependencias de PHP (getcomposer.org).
- **Node.js y NPM:** Para compilar los assets del frontend (nodejs.org).
- **Servidor de Base de Datos:** MySQL (incluido en XAMPP/Laragon).
- **Git:** (Opcional, pero recomendado).

Paso 2: Obtener el Código y Dependencias

Abre tu terminal (Command Prompt, PowerShell o Git Bash) en la carpeta donde deseas alojar el proyecto (ej. C:\xampp\htdocs\ o C:\laragon\www\).

```
git clone [URL_DEL_REPOSITORIO] cotizacion
cd cotizacion
composer install
npm install
```

Paso 3: Configuración del Entorno (.env)

Copia el archivo de ejemplo y configura tu base de datos local:

```
cp .env.example .env
php artisan key:generate
```

Abre el archivo `.env` generado con un editor de texto y configura la conexión a la base de datos (asegúrate de haber creado una base de datos vacía llamada `cotizapro` en MySQL):

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=cotizapro
DB_USERNAME=root
DB_PASSWORD=
```

Paso 4: Migraciones y Compilación

Ejecuta las migraciones para crear las tablas en la base de datos y compila los assets CSS/JS:

```
php artisan migrate
npm run build
```

Paso 5: Iniciar el Servidor Local

Levanta el servidor de desarrollo de Laravel:

```
php artisan serve
```

El sistema estará disponible en tu navegador ingresando a: `http://127.0.0.1:8000`. Registra tu primer usuario administrador desde la opción "Register".

4. Hosting Web Recomendado

Para aplicaciones Laravel, es fundamental elegir un hosting que permita acceso SSH y soporte PHP 8.3+. A continuación se presentan las mejores opciones:

Recomendación Principal: Hostinger (Plan Premium o superior)

¿Por qué Hostinger?

- Excelente relación calidad-precio.
- Panel de control amigable (hPanel).
- Acceso SSH disponible (crucial para comandos de Laravel).
- Fácil cambio de versiones de PHP (soporta PHP 8.3).
- Bases de datos MySQL SSD.
- Certificados SSL gratuitos.

Otras Alternativas

- **DigitalOcean / Linode:** Servidores VPS no administrados (Requiere conocimientos avanzados de Linux, o usar Laravel Forge). Ideal para máxima escalabilidad.
- **SiteGround / A2 Hosting:** Excelentes opciones de hosting compartido con soporte robusto para Laravel y SSH, aunque ligeramente más costosos que Hostinger.

5. Guía de Despliegue (Hostinger)

Proceso paso a paso para subir CotizaPro a un hosting compartido estándar como Hostinger utilizando acceso SSH.

Paso 1: Preparación del Entorno en el Hosting

- Inicia sesión en tu hPanel de Hostinger.
- En la sección **Avanzado > Configuración de PHP**, asegúrate de seleccionar **PHP 8.3**.
- En **Avanzado > Acceso SSH**, activa el acceso y anota tus credenciales (IP, Puerto, Usuario, Contraseña).
- En **Bases de Datos > Bases de datos MySQL**, crea una nueva base de datos, un usuario y una contraseña. Anota estos datos.

Paso 2: Subir Archivos (Vía SSH/Git o Archivo ZIP)

Opción A (Recomendada vía SSH): Conéctate por SSH a tu servidor y clona el repositorio en la carpeta raíz (usualmente fuera de `public_html` por seguridad, pero para este ejemplo usaremos una ruta estándar).

```
ssh -p PORT USUARIO@IP_SERVIDOR
cd domains/tudominio.com
# Eliminar la carpeta public_html por defecto si vas a clonar directamente
rm -rf public_html
git clone [URL_REPOSITORIO] public_html
cd public_html
```

Opción B (Archivo ZIP):

1. En tu computadora local, ejecuta `npm run build`.
2. Comprime todo el proyecto en un archivo `proyecto.zip` (excluye `node_modules`, pero incluye `vendor` si no puedes correr `composer` en el server).
3. En Hostinger, ve a **Archivos > Administrador de Archivos**.
4. Sube el ZIP dentro de `public_html` (o en una carpeta superior por seguridad) y extráelo.

Paso 3: Configurar .env en Producción

En el servidor, copia `.env.example` a `.env`. Edita el archivo con los datos de producción:

```
APP_NAME=CotizaPro
APP_ENV=production
APP_KEY=base64:.... (asegúrate de correr php artisan key:generate si está vacío)
APP_DEBUG=false
APP_URL=https://tudominio.com

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=[Nombre_DB_Hostinger]
DB_USERNAME=[Usuario_DB_Hostinger]
DB_PASSWORD=[Clave_DB_Hostinger]
```

Paso 4: Comandos de Instalación (Vía SSH)

Conéctate por SSH y ejecuta los siguientes comandos dentro de la carpeta de tu proyecto:

```
composer install --optimize-autoloader --no-dev
php artisan migrate --force
php artisan config:cache
php artisan route:cache
php artisan view:cache
```

Paso 5: Enrutamiento Público (Importante en Hosting Compartido)

En Laravel, el directorio público es `/public`. En un hosting compartido, el dominio apunta a `/public_html`. Existen varias formas de resolver esto.

Método más seguro:

1. Sube los archivos del proyecto a una carpeta *fuera* de `public_html` (ej. `/domains/tudominio.com/cotizapro`).
2. Copia el contenido de la carpeta `cotizapro/public/` al interior de la carpeta `/domains/tudominio.com/public_html/`.
3. Edita el archivo `public_html/index.php`:

```
// Cambia estas líneas:  
require __DIR__.'/../cotizapro/vendor/autoload.php';  
$app = require_once __DIR__.'/../cotizapro/bootstrap/app.php';
```

Alternativa (Archivo .htaccess en la raíz): Si subiste todo dentro de `public_html`, crea un archivo `.htaccess` en `public_html` con el siguiente contenido para redirigir el tráfico a la carpeta `public`:

```
RewriteEngine On  
RewriteRule ^(.*)$ public/$1 [L]
```

Paso 6: Permisos de Carpetas

Asegúrate de que las carpetas `storage` y `bootstrap/cache` tengan permisos de escritura (generalmente 755 o 775).

```
chmod -R 775 storage  
chmod -R 775 bootstrap/cache
```