

Documentación Técnica y de Usuario

Sistema Integral de Gestión para Panadería y Pastelería

1. Tecnologías de Desarrollo

El sistema ha sido desarrollado utilizando un stack tecnológico moderno, robusto y escalable, pensado para garantizar alto rendimiento y seguridad:

- **Backend:** PHP 8.2 con el framework **Laravel 12.0**, el estándar de la industria para aplicaciones web robustas.
 - **Frontend:** HTML5, CSS3, y JavaScript moderno. Se utiliza **Tailwind CSS 4.0** para un diseño responsivo y moderno, junto con **Alpine.js 3.4** para la interactividad de la interfaz.
 - **Base de Datos:** Configurable (por defecto SQLite/MySQL), gestionada a través del ORM Eloquent de Laravel para asegurar integridad de datos.
 - **Empaquetado de Assets:** **Vite 7.0** para la compilación ultrarrápida de recursos estáticos (CSS y JS).
-

2. Funcionalidades del Sistema

El sistema cuenta con múltiples módulos integrados, diseñados para cubrir todas las necesidades operativas de la panadería y pastelería:

- **Dashboard (Panel de Control):** Vista principal con métricas clave (KPIs), gráficas de ventas, ingresos y productos con bajo stock, ofreciendo un resumen gerencial al instante.
 - **Módulo de Punto de Venta (POS):** Interfaz táctil intuitiva y ágil para el registro de ventas, integración con caja registradora, búsqueda rápida de productos y generación de tickets de venta.
 - **Gestión de Inventario y Almacenes:** Control de existencias, ingresos y salidas, múltiples almacenes, alertas de stock mínimo y gestión de ajustes de inventario.
 - **Gestión de Producción y Recetas:** Creación de recetas maestras (insumos requeridos, tiempo de horneado, rendimiento), control de mermas y órdenes de producción diarias.
 - **Catálogo de Productos y Categorías:** Registro de productos finales (panes, pasteles, bebidas), control de variantes (entero, porción) y asignación de imágenes descriptivas.
 - **Compras y Proveedores:** Registro de múltiples proveedores, gestión de órdenes de compra para insumos y recepción de mercancía.
 - **Gestión de Clientes (CRM):** Base de datos de clientes, historial de compras, gestión de clientes frecuentes para la emisión de facturas o boletas y aplicación de descuentos.
 - **Gestión de Caja:** Apertura y cierre de caja, registro de movimientos manuales de efectivo (ingresos y egresos) y cuadro diario.
 - **Reportes y Exportaciones:** Generación de reportes detallados de ventas, producción y existencias con exportación a formatos CSV y PDF para análisis.
 - **Usuarios y Permisos:** Control de acceso basado en roles (Administrador, Cajero, Panadero) con permisos granulares para mantener la seguridad de la información.
-

3. Requisitos, Instalación y Configuración Local

Para que la aplicación funcione en un nuevo computador de desarrollo o servidor local, siga estos pasos:

Requisitos Previos:

- Instalar **PHP 8.2** o superior.
- Instalar **Composer** (Gestor de dependencias de PHP).
- Instalar **Node.js** (Gestor de paquetes NPM) y Git.

- Un servidor web (Apache, Nginx) o simplemente el servidor embebido de PHP.

Pasos de Instalación:

1. **Clonar o copiar el proyecto** en la carpeta de destino.
2. **Abrir la terminal (cmd/PowerShell)** en la ruta del proyecto.
3. Instalar las dependencias de PHP:

```
composer install
```

4. Instalar las dependencias de Node.js:

```
npm install
```

5. **Configurar el entorno:** Copiar el archivo `.env.example` y renombrarlo a `.env`. (Se puede hacer con el comando en Windows: `copy .env.example .env`). *Abrir el archivo `.env` y configurar las credenciales de la base de datos si utiliza MySQL. Por defecto está listo para SQLite.*
6. Generar la clave de la aplicación:

```
php artisan key:generate
```

7. Preparar la base de datos (Migraciones y Semillas):

```
php artisan migrate --seed
```

8. Compilar los assets del frontend:

```
npm run build
```

4. Ejecución de la Aplicación (Local)

Para ejecutar la aplicación desde la consola (CMD) en su computador:

1. Abra una terminal en el directorio del proyecto.
2. Ejecute el comando integrado de inicio rápido:

```
composer run dev
```

Este comando iniciará de forma simultánea el servidor de Laravel (`php artisan serve`) y el servidor Vite (`npm run dev`).

3. Abra su navegador web e ingrese a la dirección: `http://localhost:8000`

5. Credenciales de Acceso (Usuario Administrador)

Tras ejecutar los comandos de configuración e inicializar la base de datos, el sistema creará un usuario administrador por defecto con acceso total a todos los módulos:

- **URL de Acceso:** `http://localhost:8000/login`
- **Correo Electrónico:** `admin@panaderia.com`

- **Contraseña:** password
-

6. Hosting Ideal y Guía de Despliegue Paso a Paso

Para una aplicación desarrollada en Laravel 12.0, los servicios de hosting compartido tradicionales (como cPanel básico) pueden ser complejos de configurar y estar limitados en recursos.

La mejor y más profesional sugerencia de alojamiento es utilizar un servidor VPS (Servidor Privado Virtual) administrado a través de Laravel Forge.

Proveedores recomendados:

1. **DigitalOcean (Recomendado)** - Excelente rendimiento por precio.
2. **Linode / AWS** - Alta escalabilidad.

Guía Paso a Paso para Subir la Aplicación (Droplet Ubuntu en DigitalOcean + Laravel Forge)

Paso 1: Preparación del Código

1. Asegúrese de subir todo su proyecto a un repositorio privado en **GitHub** o **GitLab**. No incluya los directorios `vendor/` , `node_modules/` , ni el archivo `.env` .

Paso 2: Crear el Servidor (DigitalOcean)

1. Cree una cuenta en DigitalOcean.
2. Genere un API Token temporal (opcional si usa Forge para conectar la cuenta).

Paso 3: Configurar Laravel Forge

1. Regístrese en forge.laravel.com.
2. Conecte su cuenta de DigitalOcean o su proveedor de nube elegido.
3. Cree un nuevo "Server" en Forge. Seleccione **PHP 8.2+**, base de datos (MySQL) y tamaño del servidor.
4. Forge aprovisionará automáticamente el servidor (Instalará Nginx, PHP, MySQL, Redis, Composer y Node automáticamente). Esto tarda unos minutos.

Paso 4: Crear e Implementar el Sitio

1. Dentro del servidor en Laravel Forge, vaya a la sección **"Sites"**.
2. Añada el dominio de su panadería (ej: `sistema.mipanaderia.com`). Asegúrese de tener el directorio web apuntando a `/public` .
3. Seleccione el repositorio de GitHub donde alojó el código e indique la rama (ej. `main`).
4. Presione **Install Repository**.

Paso 5: Configurar el Entorno (.env)

1. En Laravel Forge, dentro de su recién creado sitio, vaya a la pestaña **Environment**.
2. Forge generará un `.env` . Ajuste las variables de la base de datos con las credenciales que Forge le proporcionará, y asegúrese de que la URL (`APP_URL`) coincida con su dominio y que el sistema esté en producción (`APP_ENV=production` , `APP_DEBUG=false`).

Paso 6: Configurar el Script de Despliegue

1. En la pestaña **App**, busque el apartado **Deploy Script**. Asegúrese de que termine de la siguiente manera:

```
composer install --no-interaction --prefer-dist --optimize-autoloader
npm ci
```

```
npm run build  
php artisan migrate --force
```

2. Presione **Deploy Now**. Forge ejecutará los comandos, compilará los estilos e inicializará el sistema.

Paso 7: Listo para Operar Ingrese a su dominio en el navegador. El sistema estará operativo con conectividad segura (SSL/HTTPS se puede activar con 1 clic usando Let's Encrypt desde la pestaña SSL de Forge).