



Manual del Sistema

Sistema de Gestión para Botica / Farmacia

Tecnología, funcionalidades, instalación y publicación en internet

Ficha del documento

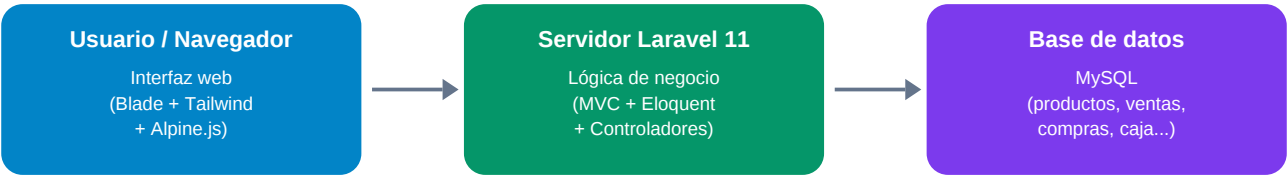
APLICACIÓN	Mi Botica – SaaS de gestión farmacéutica
TECNOLOGÍA	Laravel 11 · PHP 8.2+ · MySQL
VERSIÓN	1.0
FECHA	16/06/2026
PREPARADO PARA	Vito

1 Tecnologías utilizadas

El sistema fue desarrollado con un stack moderno de desarrollo web, basado en el framework **Laravel** sobre el lenguaje **PHP**, con base de datos **MySQL** y una interfaz responsiva.

Capa	Tecnología	Función
Backend	PHP 8.2+ / Laravel 11	Framework MVC: lógica de negocio, rutas y seguridad
Base de datos	MySQL 5.7+ / MariaDB	Almacena productos, ventas, compras, caja, etc.
ORM	Eloquent	Acceso a datos y relaciones entre tablas
Plantillas	Blade	Generación del HTML de las vistas
Estilos	Tailwind CSS	Diseño visual moderno y responsivo
Interactividad	Alpine.js	Menús, modales y carrito sin recargar
Gráficos	Chart.js	Tableros y reportes visuales
Dependencias	Composer	Gestor de paquetes de PHP
Servidor web	Apache / Nginx	Publicación de la aplicación

Arquitectura general



Peticiones HTTP → Respuestas HTML / JSON

Incluye además: control de acceso por **roles** (administrador, farmacéutico, cajero, vendedor), **auditoría automática** de acciones, migraciones de base de datos y datos de demostración (seeders).

2 Funcionalidades del sistema

El sistema cubre el flujo completo de una botica, organizado en cinco áreas:

Visión general

- **Dashboard** con indicadores del día (ingresos, ventas, clientes, alertas) y gráficos: ingresos de los últimos 7 días, distribución por método de pago, top productos y top categorías.

Comercio & Ventas

- **Punto de Venta (POS):** búsqueda de productos, carrito, métodos de pago (Efectivo, Tarjeta, Yape, Plin, Transferencia), descuento, IGV y **ticket imprimible**.
- **Historial de Ventas:** filtros por fecha, método, cajero y estado; **anulación** que devuelve el stock.
- **Clientes:** registro con documento e historial de compras y total gastado.
- **Gestión de Caja:** apertura y cierre con arqueo (esperado vs. contado) y movimientos de ingreso/egreso.

Logística & Inventario

- **Catálogo:** productos (precio, stock, lote, vencimiento, principio activo, receta) y categorías.
- **Compras:** registro a proveedores que ingresa mercadería y actualiza stock y costos.
- **Proveedores:** datos de contacto y condiciones de pago.
- **Inventario:** stock valorizado, lotes y vencimientos (criterio FEFO) y ajustes de stock con trazabilidad.

Gerencia & Control

- **Alertas Sanitarias:** productos vencidos, próximos a vencer y con stock bajo.
- **Reportes:** ventas, productos más vendidos e inventario valorizado, con **exportación a Excel (CSV)** e impresión / PDF.

Ajustes & Sistema

- **Gestión de Personal:** usuarios y roles con permisos (solo administrador).
- **Configuración General:** datos de la empresa, IGV y series de comprobantes (alimentan el ticket del POS).
- **Logs de Auditoría:** registro automático de acciones (quién, qué y cuándo).
- **Base de Datos & Respaldos:** genera y descarga copias de seguridad .sql.
- **Facturación Electrónica:** módulo opcional (integración con SUNAT).

3 Instalación en una computadora nueva (Windows)

Programas necesarios

Programa	Para qué sirve	Descarga
Laragon (recomendado)	Incluye PHP 8.2+, MySQL, Apache/Nginx y Composer; crea dominios .test	laragon.org/download
XAMPP (alternativa)	Incluye PHP, Apache y MySQL	apachefriends.org
Composer	Gestor de dependencias de PHP	getcomposer.org/download
Git (opcional)	Clonar / actualizar el proyecto	git-scm.com

Con **Laragon** no necesitas instalar PHP, MySQL ni Composer por separado: ya vienen incluidos. Es la opción más sencilla para Laravel.

Pasos de instalación

1 Instalar Laragon
Descargar 'Laragon Full', instalar con las opciones por defecto, abrirlo y pulsar 'Start All' (enciende Apache y MySQL).

2 Copiar el proyecto
Colocar la carpeta del sistema dentro de C:\laragon\www\saas_botica (si usas Laragon) o donde prefieras.

3 Abrir la terminal en la carpeta del proyecto e instalar dependencias

```
composer install
```

4 Revisar el archivo de entorno y generar la llave
El archivo .env ya viene incluido; verifica los datos de MySQL (root, sin contraseña, base 'saas_botica').

```
php artisan key:generate
```

5 Crear la base de datos
En HeidiSQL (incluido en Laragon) o phpMyAdmin crear una base 'saas_botica'. O por consola:

```
mysql -u root -e "CREATE DATABASE saas_botica CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode
```

6 Crear las tablas y cargar datos de demostración

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

7 Iniciar el servidor y abrir en el navegador

```
php artisan serve  
# Abrir: http://localhost:8000  
# (con Laragon tambien: http://saas_botica.test)
```

Acceso de prueba: usuario admin@mibotica.test · contraseña password

4 Hosting web sugerido

Para publicar la aplicación en internet existen tres tipos de alojamiento. El costo típico de un hosting para Laravel va de **4 a 50 USD al mes** según el tipo.

Tipo	Ideal para	Ejemplos	Precio aprox.
Hosting compartido	Empezar; bajo tráfico; bajo costo	Hostinger (Premium / Business)	3 – 6 USD/mes
VPS	Más control y rendimiento	Hostinger VPS (KVM), DigitalOcean, Contabo	5 – 20 USD/mes
Cloud / PaaS	Despliegue automático y escalable	Laravel Cloud, Railway, Render	Variable

RECOMENDADO

Hostinger es la opción más recomendada para una botica: panel **hPanel en español**, soporte 24/7, precios competitivos, compatibilidad con varias versiones de PHP y de Laravel, e instalación asistida. Para empezar, un plan **compartido Business** es suficiente; si el negocio crece, un **VPS KVM 2** (desde ~6,49 USD/mes) ofrece más potencia y control.

5 Cómo subirlo a Hostinger (paso a paso)

Guía para publicar el sistema en un hosting **Hostinger** con hPanel. Requiere un plan con acceso **SSH** y **Composer** (Premium o Business).

1

Contratar el plan y el dominio

Elegir un plan Premium/Business o un VPS y asociar tu dominio (ej. mibotica.com).

2

Crear la base de datos MySQL

En hPanel: 'Bases de datos MySQL'. Crear base, usuario y contraseña, y anotarlos.

3

Subir el proyecto

Comprimir el proyecto en .zip SIN las carpetas /vendor ni /node_modules. Subirlo con el Administrador de archivos a una carpeta (ej. /domains/tudominio/laravel) y descomprimir. Alternativa: clonar con Git.

4

Apuntar el dominio a la carpeta public

En hPanel cambiar la 'raíz del documento' del dominio para que apunte a la carpeta /public del proyecto (lo más seguro).

5

Instalar dependencias por SSH

```
cd ~/domains/tudominio/laravel
composer install --optimize-autoloader --no-dev
```

6

Configurar el archivo .env de producción

```
APP_ENV=production
APP_DEBUG=false
APP_URL=https://tudominio.com
DB_DATABASE=tu_base
DB_USERNAME=tu_usuario
DB_PASSWORD=tu_clave
```

7

Generar llave, migrar y optimizar

```
php artisan key:generate
php artisan migrate --seed --force
php artisan config:cache
php artisan route:cache
```

8

Dar permisos a las carpetas de escritura

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

9

Activar HTTPS (SSL gratis)

En hPanel activar el certificado SSL (Let's Encrypt) y forzar HTTPS.

1
0

Probar

Abrir <https://tudominio.com>, iniciar sesión y cambiar la contraseña del administrador.

Recomendación de seguridad: en producción mantener `APP_DEBUG=false`, cambiar las contraseñas de demostración y programar respaldos periódicos desde el módulo 'Base de Datos & Respaldos'.