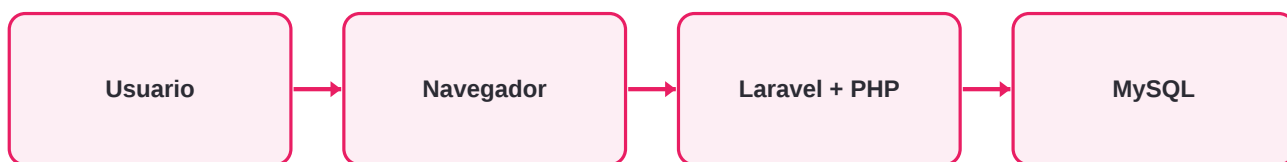


ModaSuite

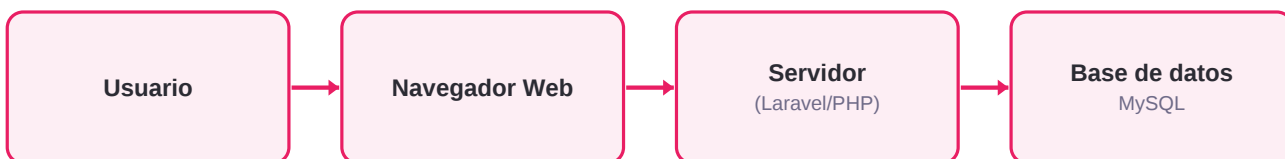


1. ¿Con qué tecnología está desarrollado?

El sistema es una **aplicación web** construida con el framework **Laravel 11** (lenguaje **PHP 8.2**), base de datos **MySQL** y vistas del lado del servidor con **Blade**. La interfaz usa **HTML5**, **CSS3** (tema propio) y **JavaScript** para las partes dinámicas (POS, buscadores, gráficos con Chart.js).

Componente	Tecnología
Lenguaje de servidor	PHP 8.2+
Framework	Laravel 11 (patrón MVC)
Base de datos	MySQL 8 / MariaDB
Motor de plantillas	Blade (renderizado del lado del servidor)
Frontend	HTML5, CSS3 (tema propio), JavaScript, Chart.js
ORM	Eloquent
Servidor web	Apache o Nginx (o el servidor embebido para pruebas)
Gestor de dependencias	Composer
Arquitectura	SaaS multi-negocio (una base de datos, datos aislados por empresa)

Arquitectura (cómo fluye una petición)



En pocas palabras: el navegador pide una página, Laravel procesa la lógica (controladores) y consulta MySQL con Eloquent, y devuelve el HTML ya armado con Blade.

2. Funcionalidades del sistema

Administración de la plataforma (Super Admin)

- Panel global con métricas de todas las empresas.
- Empresas Cliente: alta, edición, módulos visibles por empresa y branding.
- Planes de suscripción (precio, límites de usuarios/productos y módulos incluidos).
- Gestión de suscripciones con ticket e historial (Pagos / Suscripción).
- Usuarios y roles a nivel plataforma.

Operación de cada tienda (empresa)

- Catálogo: categorías, marcas, colecciones y productos con variantes (talla/color/numeración).
- Inventario con kardex de movimientos (entradas, salidas, ajustes).
- Compras y proveedores (suman stock y actualizan costo).
- Punto de Venta (POS) con carrito, medios de pago, descuentos y promociones automáticas.
- Ventas/Facturación con comprobante imprimible; Devoluciones que reponen stock.

- Caja y Arqueo (apertura, ingresos/egresos, cierre con diferencia).
- Promociones, Fidelización por puntos y Clientes.
- Reportes (ventas y stock valorizado) con exportación a CSV.
- Multi-tienda: locales y traslados de stock entre ellos.
- Configuración (logo, moneda, meta) y Mantenimiento (respaldo/restauración/reinicio por empresa).

Módulos por rubro

Rubro	Módulos propios
Prendas / Deportiva	Tallas y Colores, Colecciones y Temporadas.
Calzado	Numeración de calzado (variantes por número).
Accesorios / Bisutería	Variantes por color; Materiales.
Boutique	Colecciones y Citas / Probadores.
Joyería	Materiales y Metales, Órdenes de Reparación, Certificados/Garantías, Seguros / Alto Valor.

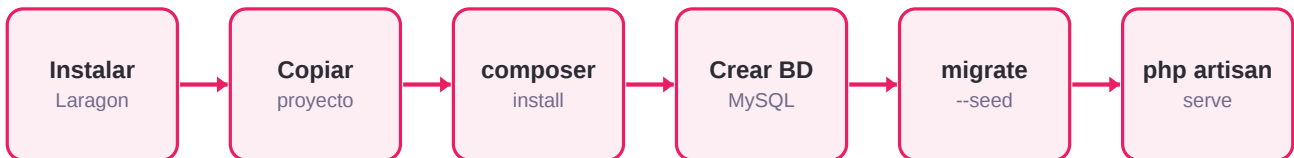
3. Instalar el sistema en una computadora nueva

Para ejecutarlo en tu PC (Windows) necesitas un entorno con PHP, MySQL y Composer. La forma más sencilla es instalar **Laragon** (o XAMPP), que ya trae Apache, PHP y MySQL juntos.

3.1 Programas a descargar e instalar

Programa	Para qué	Dónde
Laragon (recomendado) o XAMPP	Trae Apache + PHP 8.2 + MySQL en un solo instalador.	laragon.org / apachefriends.org
Composer	Instala las dependencias de Laravel.	getcomposer.org
Git (opcional)	Descargar/actualizar el proyecto por control de versiones.	git-scm.com
Editor de código (opcional)	Ver/editar el código (VS Code).	code.visualstudio.com

3.2 Flujo de instalación



3.3 Pasos detallados

- 1 Instala **Laragon** (incluye PHP, MySQL y Apache). Ábrelo y pulsa **Iniciar todo**.
- 2 Copia la carpeta del proyecto **MODAS** dentro de C:\laragon\www\ (o descomprime el ZIP allí).
- 3 Instala **Composer**. Abre la terminal de Laragon (menú **Terminal**) y ubícate en la carpeta del proyecto.
- 4 Ejecuta `composer install` para descargar las dependencias de Laravel.
- 5 Copia `.env.example` a `.env` y ejecuta `php artisan key:generate`.
- 6 Crea la base de datos `suite_saas_modas_integral` (en Laragon: menú **MySQL** → o con el comando de abajo).
- 7 Configura en `.env` los datos de MySQL (usuario **root**, contraseña vacía por defecto).
- 8 Ejecuta `php artisan migrate --seed` y luego `php artisan storage:link`.
- 9 Opcional: `php artisan db:seed --class=DemoDataSeeder` para datos de ejemplo.
- 10 Arranca con `php artisan serve` y abre **http://localhost:8000**.

Comandos (resumen)

```
composer install
copy .env.example .env (Windows)
php artisan key:generate
mysql -u root -e "CREATE DATABASE suite_saas_modas_integral CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;"
php artisan migrate --seed
php artisan storage:link
php artisan serve
```

Accesos de prueba: Super admin **superadmin@modasuite.test / password** · Admin de tienda **admin@modasuite.test / password**

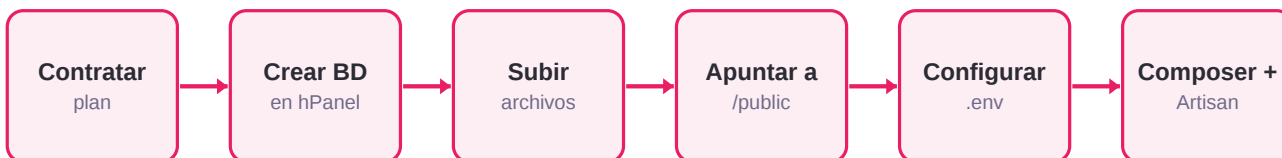
4. Hosting web sugerido

Una aplicación Laravel necesita un hosting con **PHP 8.2+**, **MySQL**, **Composer** y **acceso a consola (SSH)**. Estas son las mejores opciones según su facilidad y costo:

Hosting	Tipo	Ventajas	Ideal para
Hostinger (recomendado)	Compartido/VP S	Instalación de Laravel en 1 clic, hPanel simple, MySQL incluido, SSH; VPS desde ~US\$ 5/mes.	PyMEs que quieren publicar fácil y barato.
DigitalOcean	VPS	Imagen de Laravel en el Marketplace (1 clic), control total (root).	Quien quiere más control y escalar.
Railway / Render	PaaS	Despliegue por Git muy sencillo, escalable.	Despliegue rápido desde repositorio.
Laravel Cloud / Forge	Gestionado	Optimizado para Laravel por sus creadores.	Proyectos que crecen y requieren rendimiento.

Recomendación: para tu caso (una PyME en Perú que quiere publicar rápido y a bajo costo), **Hostinger** es la opción más recomendada por su panel sencillo (hPanel), soporte de Laravel e instalación en un clic. Para más control y varias tiendas con mucho tráfico, un **VPS** (Hostinger VPS o DigitalOcean) es lo ideal.

5. Cómo subir la aplicación a Hostinger (paso a paso)



5.1 Preparar la cuenta y la base de datos

- 1 Contrata un plan **Business/Cloud** (compartido con SSH) o un **VPS** en Hostinger, y registra o conecta tu dominio.
- 2 En **hPanel** → **Bases de datos MySQL**, crea una base de datos, un usuario y una contraseña. Anótalos.

5.2 Subir el proyecto

- 1 Genera un **.zip** del proyecto (sin la carpeta vendor ni .env).
- 2 En **hPanel** → **Administrador de archivos** sube el ZIP a la carpeta de tu dominio y descomprímelo. (Alternativa: clonar con **Git** desde la Terminal SSH.)
- 3 Haz que el dominio apunte a la carpeta **public** del proyecto: en hPanel puedes cambiar la **“Document Root”** del dominio a **/tu-proyecto/public**.

5.3 Configurar y activar

1

Crea el archivo `.env` (copiando `.env.example`) y pon los datos de la BD de Hostinger. Define `APP_ENV=production`, `APP_DEBUG=false` y `APP_URL=https://tudominio.com`.

2

Abre la **Terminal SSH** (hPanel → Avanzado → SSH) y entra a la carpeta del proyecto.

3

Ejecuta los comandos de instalación (ver abajo): dependencias, clave, migraciones, enlace de almacenamiento y cachés.

4

Da permisos de escritura a `storage` y `bootstrap/cache`.

5

Abre tu dominio en el navegador: la aplicación debe cargar el login. Ingresa y cambia las contraseñas por defecto.

Comandos en el servidor (SSH)

```
cd ~/domains/tudominio.com/tu-proyecto
composer install --no-dev --optimize-autoloader
cp .env.example .env && nano .env # configurar BD y APP_URL
php artisan key:generate
php artisan migrate --force --seed
php artisan storage:link
chmod -R 775 storage bootstrap/cache
php artisan config:cache && php artisan route:cache && php artisan view:cache
```

Si tu plan no tiene SSH: sube el proyecto ya con la carpeta `vendor` incluida y ejecuta las migraciones desde un panel/adminer, o usa un plan Business/VPS que sí incluya SSH (recomendado para Laravel).

Seguridad en producción: usa **HTTPS** (SSL gratis en Hostinger), mantén `APP_DEBUG=false`, cambia las contraseñas de prueba y programa copias de seguridad periódicas.

6. Resumen

El sistema está hecho con **Laravel 11 + PHP + MySQL**. Localmente se instala con Laragon/XAMPP + Composer en pocos pasos, y en producción el hosting más recomendado por facilidad y costo es **Hostinger** (con SSH), o un **VPS** si necesitas más potencia y múltiples tiendas con alto tráfico.