



# Virtual Services Platform (SaaS)

Plataforma integral de gestión para pequeños negocios (Barberías, Talleres, Consultorios) que automatiza la atención al cliente mediante Filas Virtuales, Citas y Notificaciones por WhatsApp.



## Características Principales

- **Fila Virtual Inteligente:** Kiosco de autoservicio (QR) para que los clientes se registren.
- **Notificaciones WhatsApp:** Avisos automáticos ("Tu turno está listo", "Tu equipo fue reparado") usando la API de Meta.
- **Gestión de Taller:** Control de equipos, fallas, costos y estatus de reparación.
- **Punto de Venta (POS):** Registro de ventas, cortes de caja y métodos de pago.
- **Seguridad Robusta:** Protección CSRF, Rate Limiting (anti-fuerza bruta), Hashing de contraseñas y Cookies seguras.
- **Infraestructura:** Despliegue contenerizado con Docker y Nginx como Proxy Inverso con SSL (HTTPS).



## Requisitos Previos

- Un servidor (AWS EC2, DigitalOcean, etc.) con Ubuntu.
- [Docker](#) y Docker Compose instalados.
- Un dominio apuntando a la IP del servidor.
- Cuenta de Meta Developers (para WhatsApp API).



## Instalación y Despliegue

### 1. Clonar el repositorio:

```
git clone  
[https://github.com/tu-usuario/virtual-services.git](https://github.com/tu-usuario/virtual-services.git)  
cd Virtual_Services
```

### 2. Configurar Variables de Entorno:

Crea un archivo .env basado en el ejemplo y configura tus credenciales.  
nano .env

### 3. Iniciar los contenedores:

```
sudo docker compose up -d --build
```

### 4. Verificar estado:

```
sudo docker ps
```

## Credenciales de Acceso

### Usuario Demo (Para Clientes)

- **URL:** <https://tudominio.com/login>
- **Usuario:** 5500000000
- **Contraseña:** demo
- **Nota:** *Este usuario tiene restringido el cambio de contraseña.*

### Super Admin (Para el Dueño del SaaS)

- **URL:** <https://tudominio.com/login>
- **Usuario:** admin
- **Contraseña:** (*La definida en tu variable PASSWORD\_ADMIN*)

## Estructura del Proyecto

```
└── app.py          # Lógica principal (Flask)
└── backup_db.sh    # Script de respaldo automático
└── docker-compose.yml # Orquestación de contenedores
└── Dockerfile       # Configuración de imagen Python
└── requirements.txt # Dependencias (Flask, Gunicorn, etc.)
└── static/          # Archivos públicos (Favicon, CSS, JS)
    └── favicon.svg
└── templates/       # Vistas HTML (Jinja2)
    ├── admin_dashboard.html
    ├── configuracion.html
    ├── landing.html
    ├── login.html
    ├── kiosco/          # Vistas públicas del negocio
    └── legal/           # Términos y privacidad
```

## Seguridad Implementada

- **CSRF Protection:** Todos los formularios requieren token de seguridad.
- **Rate Limiting:** Bloqueo de IP tras 5 intentos fallidos de login por minuto.
- **Password Hashing:** Encriptación automática de contraseñas con werkzeug.security.
- **Secure Cookies:** Cookies configuradas como HttpOnly y Secure.
- **Webhook Signature:** Validación de firma X-Hub-Signature-256 para mensajes de WhatsApp.

## Respaldos (Backups)

El sistema incluye un script de respaldo automático para la base de datos MySQL.

**Ejecución manual:**

`./backup_db.sh`

Configuración automática (Cron):

Se ejecuta diariamente a las 3:00 AM.

`0 3 * * * /home/ubuntu/Virtual_Services/backup_db.sh >> /home/ubuntu/backup.log 2>&1`

## Soporte

Para soporte técnico o dudas sobre la implementación:

- **Contacto:** emilianosully@gmail.com
- **Web:** <https://virtualservices.com.mx>

© 2025 Virtual Services Platform. Todos los derechos reservados.