

Generador de Traslados Cielito - Aplicación Web

Aplicación web moderna para generar recomendaciones de traslados de inventario de forma automatizada.

Requisitos Previos

- Python 3.11+
- Base de datos SQL Server configurada
- Archivo `.env` con credenciales

Instalación

1. Instalar dependencias adicionales

```
bash

pip install -r requirements_web.txt
```

2. Crear estructura de directorios

```
bash

# Desde la raíz del proyecto (PROYECTO_TRASLADOS2/)
mkdir -p app/static/css
mkdir -p app/static/js
mkdir -p app/static/images
mkdir -p app/templates
mkdir -p app/output
```

3. Copiar archivos

Coloca los archivos en la siguiente estructura:

```
PROYECTO_TRASLADOS2/
├── app/
│   ├── __init__.py (crear vacío)
│   ├── main.py (servidor FastAPI)
│   ├── static/
│   │   ├── css/
│   │   │   └── style.css
│   │   ├── js/
│   │   │   └── app.js
│   │   └── images/
│   │       └── logo_cielito.png (tu logo)
│   └── templates/
│       └── index.html
├── config/
├── processors/
├── traslados/
├── db/
├── main.py (tu script principal)
└── .env
```

4. Agregar logo

Coloca tu logo de Cielito en:

```
app/static/images/logo_cielito.png
```

Ejecución

Modo desarrollo (con auto-reload)

```
bash
cd app
python main.py
```

O usando uvicorn directamente:

```
bash
uvicorn app.main:app --reload --host 0.0.0.0 --port 8000
```

Modo producción

```
bash
uvicorn app.main:app --host 0.0.0.0 --port 8000 --workers 4
```

Acceso a la Aplicación

Una vez iniciado el servidor, abre tu navegador en:

```
http://localhost:8000
```

Uso de la Aplicación

1. Configuración Básica

- **Meses de análisis:** Selecciona cuántos meses de historial usar (1-12)
- **Días de cobertura:** Define el rango objetivo de inventario en días
- **Ratio de seguridad:** Porcentaje de inventario a conservar en bodega

2. Opciones Avanzadas

- ☒ **Modo Debug:** Genera logs detallados
- ☒ **Guardar intermedios:** Exporta archivos de ventas y stock procesados
- ☒ **Permitir siembra:** Permite enviar productos nuevos a tiendas

3. Generar Traslados

1. Configura los parámetros deseados
2. Haz clic en "Generar Traslados"
3. Espera mientras se procesa (verás una barra de progreso)

4. Descarga los archivos generados automáticamente

Archivos Generados

La aplicación genera 2 archivos Excel:

1. `Traslados_final_[timestamp].xlsx`

- **Hoja 1 - Traslados:** Detalle completo de todos los traslados
- **Hoja 2 - Stock Final:** Inventario resultante después de aplicar traslados

2. `Traslados_final_resumen_[timestamp].xlsx`

- **Hoja 1 - Por Tienda:** Resumen agregado por tienda destino
- **Hoja 2 - Por Fase:** Resumen por fase del algoritmo
- **Hoja 3 - Top 50:** Referencias más trasladadas

Solución de Problemas

Error de conexión a base de datos

Verifica que tu archivo `.env` tenga las credenciales correctas:

```
env

SQL_SERVER=tu_servidor
SQL_DATABASE=tu_base_de_datos
SQL_USER=tu_usuario
SQL_PASSWORD=tu_contraseña
SQL_DRIVER={ODBC Driver 18 for SQL Server}
```

Puerto 8000 ya en uso

Cambia el puerto en el comando de inicio:

```
bash


uvicorn app.main:app --reload --port 8080
```

Logo no se muestra

Verifica que el archivo esté en:

```
app/static/images/logo_cielito.png
```

Seguridad

 **IMPORTANTE:** Esta aplicación está diseñada para uso interno. Para producción:

1. Agrega autenticación (OAuth, JWT, etc.)
2. Usa HTTPS
3. Implementa rate limiting
4. Valida y sanitiza todas las entradas
5. Configura CORS apropiadamente

Monitoreo

Los logs se guardan en:

- Consola del servidor (tiempo real)
- `traslados.log` (si está habilitado debug)

Personalización

Cambiar colores

Edita las variables CSS en `app/static/css/style.css`:

```
css

:root {
  --primary-color: #2563eb; /* Color principal */
  --success-color: #10b981; /* Color de éxito */
  --error-color: #ef4444; /* Color de error */
}
```

Cambiar textos

Edita `app/templates/index.html` para modificar títulos y descripciones.

Despliegue en Producción

Para desplegar en un servidor:

Usando Docker (recomendado)

```
dockerfile

FROM python:3.11-slim

WORKDIR /app

COPY requirements.txt requirements_web.txt ./
RUN pip install -r requirements.txt -r requirements_web.txt

COPY . .

EXPOSE 8000

CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

Usando Gunicorn

```
bash

pip install gunicorn
gunicorn app.main:app --workers 4 --worker-class uvicorn.workers.UvicornWorker --bind 0.0.0.0:8000
```

Soporte

Para problemas o preguntas, contacta al equipo de desarrollo.

Versión: 2.0

Última actualización: Octubre 2025