

Manual de Procesos

Publicación del Dashboard EMBI con Dash y Render

Fecha de elaboración: 19/09/2025

Índice

1. Objetivo
2. Flujo General
3. Estructura del Proyecto
4. Configuración en GitHub
5. Despliegue en Render
6. Actualizaciones
7. Verificación del Servicio
8. Mantenimiento
9. Próximos pasos sugeridos

1. Objetivo

El objetivo es implementar un dashboard interactivo en Python (Dash + Plotly) que muestre la evolución del EMBI (Emerging Market Bond Index) por país, utilizando como fuente el archivo oficial del Banco Central de la República Dominicana. El dashboard debe ser accesible en la web mediante un servicio de despliegue en Render.

2. Flujo General

El proceso completo sigue los pasos: 1. Preparar los archivos de la aplicación: app.py, requirements.txt y Procfile. 2. Crear un repositorio en GitHub y subir los archivos. 3. Configurar un Web Service en Render. 4. Render ejecuta automáticamente el script y publica el dashboard. 5. Cada actualización en GitHub dispara un nuevo despliegue en Render.

3. Estructura del Proyecto

La carpeta debe contener: - app.py: Script principal con el dashboard (incluye carga desde la URL del Excel, limpieza y visualización). - requirements.txt: Lista de librerías necesarias. - Procfile: Instrucción para iniciar la aplicación en Render. - .gitignore: Opcional, para excluir archivos temporales o entornos locales.

4. Configuración en GitHub

1. Crear un nuevo repositorio en GitHub (público o privado). 2. Inicializar Git en la carpeta del proyecto con 'git init'. 3. Conectar el repositorio local con GitHub mediante 'git remote add origin'. 4. Subir archivos con 'git add .', 'git commit' y 'git push'. 5. Verificar en GitHub que los archivos estén visibles.

5. Despliegue en Render

1. Ingresar a <https://render.com>. 2. Crear un nuevo Web Service. 3. Seleccionar el repositorio 'embi-dash'. 4. Configurar: - Build Command: pip install -r requirements.txt - Start Command: gunicorn app:server 5. Render instalará dependencias, ejecutará el script y entregará una URL pública accesible.

6. Actualizaciones

Cada vez que se realicen cambios en el código: - Ejecutar 'git add .', 'git commit -m "mensaje"' y 'git push'. - Render detectará el cambio y redeployará automáticamente. Actualización de datos: El Excel se descarga en vivo desde la URL oficial, por lo que no se requiere subir archivos manualmente.

7. Verificación del Servicio

Para comprobar que el servicio está activo: - Abrir la URL pública entregada por Render y verificar el funcionamiento del dashboard. - Consultar la pestaña 'Logs' en Render para revisar que el archivo Excel fue descargado y procesado. - Errores comunes: Error 500 si la URL del Excel no responde; tiempo de espera si el plan gratuito suspende el servicio por inactividad.

8. Mantenimiento

Buenas prácticas de mantenimiento: - Documentar cambios con commits claros en GitHub. - Revisar periódicamente que la URL del Excel siga vigente. - Si la URL cambia, modificar la variable 'url' en app.py.

9. Próximos pasos sugeridos

Posibles mejoras: - Agregar exportación de gráficos en formato imagen. - Implementar caché local para reducir tiempos de carga. - Migrar a Railway o Fly.io para mayor estabilidad y escalabilidad. Resultado esperado: Un dashboard público accesible desde cualquier dispositivo, que siempre muestre los últimos datos del EMBI con filtros interactivos por país y fecha.