**🚀 Instructivo detallado: Desplegar Flask + SQL Server + React en Azure App Service**

**✅ Objetivo**

Este instructivo te guiará paso a paso para desplegar tu backend en **Azure App Service** con **SQL Server**, y conectar tu **frontend en React**. También se considera la ejecución de la versión de consola.

**🔹 1️⃣ Configurar la Base de Datos en Azure SQL Server**

Como tu backend usa **SQL Server**, primero necesitamos **crear la base de datos** en Azure.

**📌 Crear SQL Server en Azure**

1. **Ir al portal de Azure**: [https://portal.azure.com](https://portal.azure.com/)
2. **Buscar "SQL Databases"** y hacer clic en Crear → Base de datos SQL.
3. **Configurar la base de datos**:
   * **Nombre de la base de datos**: productos\_db
   * **Servidor**: Crear nuevo
   * **Nombre del servidor**: sqlserver-backend
   * **Región**: East US *(o la que prefieras)*
   * **Usuario administrador**: adminsql
   * **Contraseña**: TuContraseñaSegura
   * **Tipo de base de datos**: Azure SQL Database
4. **Hacer clic en "Siguiente" y Crear**.
5. **Habilitar acceso externo**: Ve al servidor SQL creado (sqlserver-backend.database.windows.net) y en Firewall, **agrega tu IP pública** (Permitir acceso a servicios de Azure).
6. **Copiar la cadena de conexión** en Cadenas de conexión > ADO.NET:
7. Server=tcp:sqlserver-backend.database.windows.net,1433;Database=productos\_db;User ID=adminsql;Password=TuContraseñaSegura;

**🔹 2️⃣ Configurar el Backend en Azure App Service**

Tu backend en Flask debe ejecutarse en **Azure App Service**.

**📌 1. Subir tu backend a GitHub**

Si no lo tienes en **GitHub**, súbelo:

git init

git add .

git commit -m "Subiendo backend a Azure"

git branch -M main

git remote add origin https://github.com/TU\_USUARIO/TU\_REPO.git

git push -u origin main

**📌 2. Crear Azure App Service**

1. **Ir a Azure Portal** > Buscar App Services.
2. **Clic en Crear > Aplicación web**.
3. **Configurar App Service**:
   * **Nombre de la aplicación**: backend-productos
   * **Sistema operativo**: Linux
   * **Publicación**: Código
   * **Pila de ejecución**: Python 3.9
   * **Plan de hospedaje**: Gratis (F1) o Básico (B1)
4. **Ir a "Revisar y crear"** y **crear la app**.

**🔹 3️⃣ Configurar Variables de Entorno**

1. **Ir a Configuración > Configuración de Aplicaciones**.
2. **Agregar las siguientes variables de entorno**:
   * SQL\_SERVER = sqlserver-backend.database.windows.net
   * SQL\_DATABASE = productos\_db
   * SQL\_USER = adminsql
   * SQL\_PASSWORD = TuContraseñaSegura
   * FLASK\_ENV = production

**🔹 4️⃣ Configurar el Despliegue en Azure**

1. **Ir a Implementación > GitHub Actions**.
2. **Seleccionar el repositorio donde subiste el backend**.
3. **Clic en Configurar y Guardar**.

Azure configurará automáticamente una **GitHub Action** para desplegar el backend cada vez que hagas un git push.

**🔹 5️⃣ Conectar React con el Backend**

Actualmente, tu frontend en **React** (App.js) usa:

const API\_URL = "http://localhost:5000";

🔹 **Reemplázalo con la URL de Azure App Service**:

const API\_URL = "https://backend-productos.azurewebsites.net";

**🔹 6️⃣ Desplegar React en Azure Static Web Apps**

1. **Ir a Azure Portal** > Buscar Static Web Apps y crear una nueva.
2. **Seleccionar el repositorio donde está el frontend en GitHub**.
3. **Configurar la ruta de salida del build (/build)**.
4. **Esperar que Azure despliegue React automáticamente**.

Cuando termine, la app se podrá acceder desde:

https://frontend-productos.azurestaticapps.net

**🔹 7️⃣ Ejecutar la Consola en Windows**

Si tu consola interactúa con el backend, solo necesitas cambiar en UI\_console.py:

API\_URL = "https://backend-productos.azurewebsites.net"

Luego, puedes ejecutarlo en **Windows** con:

python UI\_console.py

**🚀 ¡Listo! Ahora tu aplicación está 100% en la nube de Azure**

✅ **Backend en Flask en Azure App Service**  
✅ **Base de datos en Azure SQL Database**  
✅ **Frontend en React en Azure Static Web Apps**  
✅ **Consola en Python funcionando con Azure**

📌 **Dime si necesitas ayuda con algún paso específico o si algo no funciona como esperas.** 🚀