**Bot de Pedidos para Telegram con Node.js, Render y AppSheet**

Este proyecto implementa un bot de chat para Telegram que permite a los usuarios realizar pedidos. El bot está construido con Node.js, utiliza la librería node-telegram-bot-api para comunicarse y AppSheet como base de datos gratuita para gestionar productos y pedidos. Está diseñado para ser desplegado fácilmente en la plataforma de hosting Render.

**Características**

* **Flujo de Conversación Guiado**: El bot guía al usuario desde el saludo inicial hasta la finalización del pedido.
* **Recopilación de Datos del Cliente**: Solicita y almacena el nombre, dirección y celular del cliente.
* **Búsqueda de Productos**: Busca productos en una tabla de AppSheet.
* **Manejo de Múltiples Coincidencias**: Si la búsqueda arroja varios resultados, le permite al usuario elegir el correcto.
* **Carrito de Compras**: Los usuarios pueden añadir múltiples productos a un solo pedido.
* **Resumen de Pedido**: Muestra un resumen detallado antes de finalizar y al confirmar el pedido.
* **Persistencia de Datos**: Guarda toda la información del pedido en tablas de AppSheet.
* **Costo Cero**: Funciona con los planes gratuitos de Telegram, AppSheet y Render.
* **Operación 24/7**: Al usar el método de "polling", el bot se mantiene activo en Render sin dormirse por inactividad.

**📋 Requisitos Previos**

1. **Node.js**: Instalado en tu máquina local (v16 o superior).
2. **Cuenta de Telegram**: Para crear y administrar el bot.
3. **Cuenta de AppSheet**: Donde crearás tu aplicación y tus tablas.
4. **Cuenta de Render**: Para desplegar la aplicación.
5. **Git y cuenta de GitHub**: Para subir tu código y desplegar en Render.

**🛠️ Paso 1: Crear tu Bot de Telegram**

1. Abre Telegram y busca al bot llamado BotFather.
2. Inicia una conversación con él y envía el comando /newbot.
3. Sigue sus instrucciones: elige un nombre para tu bot (ej. "Bot de Pedidos de Mi Tienda") y luego un nombre de usuario único que debe terminar en "bot" (ej. "MiTiendaPedidosBot").
4. ¡Listo! BotFather te dará un **token de acceso a la API**. Es una cadena larga de caracteres.
5. **Copia y guarda este token en un lugar seguro**. Lo necesitarás en un momento.

**⚙️ Paso 2: Configuración de AppSheet**

Esta parte no cambia. Tu base de datos vivirá en AppSheet. Si ya la tenías configurada, puedes saltar al paso 3.

**2.1. Crear las Tablas**

Crea una nueva aplicación en AppSheet y añade las siguientes tres tablas (puedes usar Google Sheets como fuente).

**Tabla: Productos** | Nombre de Columna | Tipo de Dato | Notas | | :--- | :--- | :--- | | nombreProducto | Text | (Clave) Nombre y descripción del producto. | | valor | Number | Precio del producto. |

**Tabla: enc\_pedido (Encabezado del Pedido)** | Nombre de Columna | Tipo de Dato | Notas | | :--- | :--- | :--- | | pedidoid | Text | (Clave) El ID único del pedido. | | enc\_total | Number | El valor total de todos los productos del pedido. | | fecha | Date | La fecha en que se realizó el pedido. | | cliente | Text | Nombre completo del cliente. | | direccion | Address | Dirección de entrega del cliente. | | celular | Phone | Número de contacto del cliente. |

**Tabla: Pedido (Detalle del Pedido)** | Nombre de Columna | Tipo de Dato | Notas | | :--- | :--- | :--- | | RowID | Text | (Clave) ID autogenerado. | | pedidoid | Text | El mismo ID de la tabla enc\_pedido para relacionarlos. | | fecha | DateTime | Fecha y hora exactas en que se añadió el producto. | | nombreProducto | Text | Nombre del producto pedido. | | cantidadProducto | Number | Cantidad de unidades de este producto. | | valor\_unit | Number | Precio unitario del producto. | | valor | Number | Valor total (cantidad \* valor\_unit). |

**2.2. Habilitar la API y Obtener Claves**

1. En el editor de AppSheet, ve a **Manage > Integrations**.
2. Haz clic en **"Enable"** para la sección de API.
3. Se generará una **Application Access Key**. Cópiala y guárdala.
4. También necesitarás tu **App ID**. Lo puedes encontrar en la URL del editor (.../appName/yourapp-12345/) o en **Manage > Integrations**.

**🚀 Paso 3: Configuración y Despliegue en Render**

**3.1. Preparar el Código Localmente**

1. **Organiza tus archivos**: Coloca index.js, appsheet.js, package.json, y .gitignore en una carpeta.
2. **Crea el archivo .env**:
   * Dentro de la carpeta, crea un archivo llamado .env (puedes copiar y renombrar .env.example).
   * Abre el archivo .env y pega lo siguiente, reemplazando los valores de ejemplo con tus credenciales reales:
3. # Credenciales de AppSheet
4. APPSHEET\_APP\_ID=tu\_id\_de\_app\_aqui
5. APPSHEET\_ACCESS\_KEY=tu\_access\_key\_aqui
6. # Token del Bot de Telegram
7. TELEGRAM\_BOT\_TOKEN=el\_token\_que\_te\_dio\_botfather
8. # (Opcional) Usuario de Telegram del asesor
9. TELEGRAM\_ASESOR\_USERNAME=usuario\_asesor\_sin\_arroba
10. **Instala las dependencias**:
    * Abre una terminal en la carpeta del proyecto.
    * Ejecuta el comando npm install.

**3.2. Subir el Código a GitHub**

1. Crea un nuevo repositorio en GitHub.
2. Sube tus archivos (index.js, appsheet.js, package.json, .gitignore) al repositorio. **Asegúrate de que .env esté en tu .gitignore para no subir tus claves secretas.**

**3.3. Desplegar en Render**

1. Inicia sesión en Render y haz clic en **New + > Web Service**.
2. Conecta tu repositorio de GitHub.
3. Configura el servicio web:
   * **Name**: Dale un nombre único (ej: mi-bot-telegram).
   * **Runtime**: Node.
   * **Build Command**: npm install.
   * **Start Command**: node index.js.
   * **Instance Type**: El plan Free es suficiente.
4. **Añade las Variables de Entorno**:
   * Ve a la sección **"Environment"** y haz clic en **"Add Environment Variable"**.
   * Añade cada una de las claves de tu archivo .env una por una:
     + **Key**: APPSHEET\_APP\_ID, **Value**: tu\_id\_de\_app\_aqui
     + **Key**: APPSHEET\_ACCESS\_KEY, **Value**: tu\_access\_key\_aqui
     + **Key**: TELEGRAM\_BOT\_TOKEN, **Value**: el\_token\_que\_te\_dio\_botfather
     + **Key**: TELEGRAM\_ASESOR\_USERNAME, **Value**: usuario\_asesor\_sin\_arroba (opcional)
5. Haz clic en **"Create Web Service"**. Render comenzará el despliegue. Si todo va bien, verás en el log el mensaje "Bot de Telegram iniciado y esperando mensajes...".

**✅ ¡Listo! Cómo Probar tu Bot**

1. Abre Telegram y busca el nombre de usuario de tu bot (el que creaste con BotFather, ej. MiTiendaPedidosBot).
2. Inicia una conversación con él.
3. Envía /start o hola para iniciar el flujo de pedidos.
4. Sigue las instrucciones del bot para realizar una prueba.
5. Verifica que los datos del pedido se guarden correctamente en tus tablas de AppSheet.