

¡Hola! Sí, después de revisar detenidamente el código y los archivos que proporcionaste, he identificado un error clave que es muy probablemente la causa por la que no estás recibiendo los mensajes en Telegram.

El problema reside en una inconsistencia entre cómo se leen las variables de entorno en tu código fuente de TypeScript (functions/src/index.ts) y el código JavaScript compilado (functions/lib/index.js) que finalmente se ejecuta en Firebase.

El Error Principal: Acceso a las Variables de Entorno

1. Lo que se pretendía (y está en el prompt.txt):

En tu archivo de instrucciones prompt.txt y en el código fuente functions/src/index.ts, las credenciales de Telegram se obtienen correctamente de la configuración de Firebase así:

TypeScript

```
const TELEGRAM_BOT_TOKEN = functions.config().telegram.token;
```

```
const TELEGRAM_CHAT_ID = functions.config().telegram.chat_id;
```

2. El problema real (en el código que se ejecuta):

Sin embargo, en el archivo JavaScript compilado (functions/lib/index.js) que Firebase ejecuta, el código intenta acceder a estas variables de una manera diferente:

JavaScript

```
const TELEGRAM_BOT_TOKEN = process.env.TELEGRAM_TOKEN;
```

```
const TELEGRAM_CHAT_ID = process.env.TELEGRAM_CHAT_ID;
```

Cuando configuras las variables con firebase functions:config:set telegram.token="...", Firebase **no** las expone en process.env. Por lo tanto, TELEGRAM_BOT_TOKEN y TELEGRAM_CHAT_ID son undefined en tu función.

Debido a esto, la función sendTelegramAlert se detiene prematuramente en esta comprobación, y nunca llega a enviar el mensaje:

TypeScript

```
async function sendTelegramAlert(deviceId: string, measurement: Measurement) {
```

```
  if (!TELEGRAM_BOT_TOKEN || !TELEGRAM_CHAT_ID) {
```

```
    console.error("Error: El token del bot de Telegram o el ID del chat no están configurados en las variables de entorno.");
```

```
    return; // La función termina aquí y no envía nada
```

```
  }
```

```
// ... el resto del código de envío  
}
```

Esto también explica por qué no ves nada en los registros de Google Cloud con el filtro "Telegram API response": la función nunca intenta hacer la llamada a la API.

Cómo Solucionarlo

La solución es asegurarse de que el código que se compila y despliega utilice el método `functions.config()` para leer las variables. El código en `functions/src/index.ts` parece ser una versión diferente a la que generó `functions/lib/index.js`.

Por favor, asegúrate de que el contenido de tu archivo `functions/src/index.ts` sea exactamente este y luego vuelve a desplegarlo:

TypeScript

```
// functions/src/index.ts
```

```
import * as functions from "firebase-functions";
```

```
import * as admin from "firebase-admin";
```

```
import * as https from "https"; // Importa el módulo https de Node.js
```

```
// Initialize Firebase Admin SDK
```

```
admin.initializeApp();
```

```
const db = admin.database();
```

```
// Obtén las credenciales de Telegram desde la configuración de Firebase
```

```
const TELEGRAM_BOT_TOKEN = functions.config().telegram.token;
```

```
const TELEGRAM_CHAT_ID = functions.config().telegram.chat_id;
```

```
const NOISE_THRESHOLD_DB = 85.0; // Umbral para estado "Rojo"
```

```
const DURATION_MINUTES = 1;
```

```
const DURATION_SECONDS = DURATION_MINUTES * 60;
```

```

/**
 * Interface for the structure of a single measurement in the database.
 */
interface Measurement {
    estado: string;
    fecha: string;
    nivel_dB: string;
    vibracion_ms2: string;
}

/**
 * This Cloud Function triggers whenever new data is written to /mediciones/{deviceId}.
 * It checks if the noise level has been consistently high for a specified duration
 * and sends a Telegram alert if the condition is met.
 */
export const checkNoiseLevel = functions.database
    .ref("/mediciones/{deviceId}/{timestamp}")
    .onWrite(async (change, context) => {
        if (!change.after.exists()) {
            return null;
        }

        const { deviceId } = context.params;
        const newMeasurement = change.after.val() as Measurement;
        const alertStatusRef = db.ref(`/alert_status/${deviceId}`);

        if (newMeasurement.estado.toLowerCase() === "rojo") {
            const alertStatusSnapshot = await alertStatusRef.once("value");

```

```

    if (alertStatusSnapshot.val()?.alerted === true) {
        console.log(`[${deviceId}] Ya se ha enviado una alerta. No se necesita acción.`);
        return null;
    }

    console.log(`[${deviceId}] Estado 'Rojo' detectado. Enviando alerta...`);
    await sendTelegramAlert(deviceId, newMeasurement);
    await alertStatusRef.set({ alerted: true });
} else {
    await alertStatusRef.set({ alerted: false });
    console.log(`[${deviceId}] El estado no es 'Rojo'. Se reinicia el estado de la alerta.`);
}

return null;
});

/**
 * Sends an alert message to a Telegram chat using the Telegram Bot API.
 * @param {string} deviceId The ID of the device that triggered the alert.
 * @param {Measurement} measurement The latest measurement data.
 */
async function sendTelegramAlert(deviceId: string, measurement: Measurement) {
    if (!TELEGRAM_BOT_TOKEN || !TELEGRAM_CHAT_ID) {
        console.error("Error: El token del bot de Telegram o el ID del chat no están configurados en las variables de entorno.");
        return;
    }

    const message = `Alerta de Ruido: ${deviceId} - Nivel: ${measurement.nivel_dB} - Fecha: ${measurement.fecha}`;

```

```
const data = JSON.stringify({  
  chat_id: TELEGRAM_CHAT_ID,  
  text: message,  
});
```

```
const options = {  
  hostname: 'api.telegram.org',  
  port: 443,  
  path: `/bot${TELEGRAM_BOT_TOKEN}/sendMessage`,  
  method: 'POST',  
  headers: {  
    'Content-Type': 'application/json',  
    'Content-Length': data.length,  
  },  
};
```

```
return new Promise<void>((resolve, reject) => {  
  const req = https.request(options, (res) => {  
    let responseBody = "";  
    res.on('data', (chunk) => {  
      responseBody += chunk;  
    });  
    res.on('end', () => {  
      console.log(`[${deviceId}] Telegram API response:`, responseBody);  
      if (res.statusCode === 200) {  
        console.log(`[${deviceId}] Alerta de Telegram enviada con éxito.`);  
        resolve();  
      }  
    });  
  });  
});
```

```

    } else {
        console.error(`[${deviceId}] Error al enviar la alerta de Telegram: ${res.statusCode}`,
        responseBody);
        reject(new Error(`Telegram API responded with status code ${res.statusCode}`));
    }
    });
});

req.on('error', (error) => {
    console.error(`[${deviceId}] Error en la solicitud a la API de Telegram:`, error);
    reject(error);
});

req.write(data);
req.end();
});
}

```

Pasos a Seguir

1. **Reemplaza** el contenido de functions/src/index.ts con el código corregido que te he proporcionado.
2. **Abre tu terminal** en el directorio raíz de tu proyecto.
3. **Despliega tus funciones** nuevamente con el comando:

Bash

```
firebase deploy --only functions
```

Después de desplegar, el sistema debería funcionar correctamente. Si aún tienes problemas, te recomiendo quitar cualquier filtro en los registros de Google Cloud para poder ver todos los mensajes, incluyendo el console.error que te indicará si las variables de entorno se están leyendo como nulas.