

## Aplicaciones Distribuidas

### Práctica

Realice las siguientes actividades

1. Se usará el Servicio Web que genera números aleatorios. Debe ajustarlo para que genere números entre 15 y 45. Estos valores serán tomados como posibles temperaturas.
2. Genere un programa en NodeJS, que usando Axios, llame al Servicio de números aleatorios cada 30 segundos. Si la temperatura recibida es mayor a 39 “tres veces seguidas”, se debe enviar un correo electrónico notificando este evento.
3. Haga lo mismo, pero ahora usando como medio de notificación a alguno de los siguientes ambientes:
  - a. Telegram
  - b. Discord
4. Documente sus códigos incluyendo evidencias de sus pruebas
5. Comparta el documento y sus códigos por un mensaje privado en el Canal de Discord

## Código

```
const axios = require('axios');

// Configuración del servicio de números aleatorios
const RANDOM_API_URL = 'https://www.random.org/integers/';
const MIN_TEMP = 15;
const MAX_TEMP = 45;
const TEMP_UMBRAL = 39;
const INTERVALO_MS = 3000; // 3 segundos

const DISCORD_WEBHOOK_URL =
'https://discord.com/api/webhooks/1422614151607484437/Kga_cQfZoG4qP6bE2887Ec76eL4bQjtK2YzwNqCUqNRYHhOEqx6JB8cMGXcoTUsOIVoY';

// Array para almacenar las últimas 3 temperaturas
let historialTemperaturas = [];

/**
 * Función para obtener temperatura aleatoria del servicio web
 */
async function obtenerTemperatura() {
  try {
    const response = await axios.get(RANDOM_API_URL, {
      params: {
        num: 1,
        min: MIN_TEMP,
        max: MAX_TEMP,
        col: 1,
        base: 10,
        format: 'plain',
        rnd: 'new'
      }
    });
  });

  // Convertir la respuesta a string si no lo es
  let temperaturaStr = response.data;
  if (typeof temperaturaStr !== 'string') {
    temperaturaStr = String(temperaturaStr);
  }

  const temperatura = parseInt(temperaturaStr.trim());
  console.log(` Temperatura actual: ${temperatura}°C`);
  return temperatura;
}
```

```

    } catch (error) {
        console.error(' Error al obtener temperatura:', error.message);
        console.error('Detalles:', error.response?.data || error);
        return null;
    }
}

/**
 * Función para enviar alerta a Discord usando Webhook
 */
async function enviarAlertaDiscord(temperaturas) {
    const embed = {
        title: ' ALERTA: Temperatura Alta Detectada',
        description: `Se han detectado **3 temperaturas consecutivas** superiores a
${TEMP_UMBRAL}°C`,
        color: 15158332, // Color rojo (en decimal)
        fields: [
            {
                name: ' Lecturas Consecutivas',
                value: ` Lectura 1: **${temperaturas[0]}°C**\nLectura 2: **${temperaturas[1]}°C**\n
Lectura 3: **${temperaturas[2]}°C**`,
                inline: false
            },
            {
                name: ' Fecha y Hora',
                value: new Date().toLocaleString('es-MX', {
                    timeZone: 'America/Mexico_City',
                    dateStyle: 'full',
                    timeStyle: 'long'
                }),
                inline: false
            }
        ],
        footer: {
            text: 'Sistema de Monitoreo de Temperatura'
        },
        timestamp: new Date().toISOString()
    };

    const payload = {
        username: 'Monitor de Temperatura',
        avatar_url: 'https://cdn-icons-png.flaticon.com/512/1684/1684371.png',
        embeds: [embed]
    };

```

```

    try {
      await axios.post(DISCORD_WEBHOOK_URL, payload);
      console.log('Alerta enviada a Discord exitosamente');
    } catch (error) {
      console.error('Error al enviar mensaje a Discord:', error.message);
      if (error.response) {
        console.error('Detalles del error:', error.response.data);
      }
    }
  }
}

/**
 * Función para verificar si hay 3 temperaturas altas consecutivas
 */
function verificarAlertaTemperatura() {
  if (historialTemperaturas.length >= 3) {
    // Verificar las últimas 3 temperaturas
    const ultimas3 = historialTemperaturas.slice(-3);
    const todasAltas = ultimas3.every(temp => temp > TEMP_UMBRAL);

    if (todasAltas) {
      console.log('¡ALERTA! Tres temperaturas consecutivas superiores a', TEMP_UMBRAL +
        '°C');
      enviarAlertaDiscord(ultimas3);

      // Limpiar el historial después de enviar la alerta
      historialTemperaturas = [];
      console.log('Historial limpiado. Reiniciando monitoreo...\n');
      return true;
    }
  }
  return false;
}

/**
 * Función para esperar (sleep)
 */
function sleep(ms) {
  return new Promise(resolve => setTimeout(resolve, ms));
}

/**
 * Función principal de monitoreo continuo

```

```

*/
async function monitorearTemperatura() {
  let i = 1;

  console.log('Iniciando monitoreo de temperatura...');
  console.log(`Configuración: Rango  ${MIN_TEMP}-${MAX_TEMP}°C    |    Umbral:
${TEMP_UMBRAL}°C | Intervalo: ${INTERVALO_MS/1000}s`);
  console.log(`📄 Notificaciones: Discord Webhook\n`);

  // Verificar que el webhook esté configurado
  if (DISCORD_WEBHOOK_URL === 'TU_WEBHOOK_URL_AQUI') {
    console.error(' ERROR: Debes configurar tu DISCORD_WEBHOOK_URL en la línea 11');
    console.error(' Sigue las instrucciones en los comentarios del código\n');
    process.exit(1);
  }

  while (i > 0) {
    // Obtener temperatura actual
    const temperatura = await obtenerTemperatura();

    if (temperatura !== null) {
      // Agregar temperatura al historial
      historialTemperaturas.push(temperatura);

      // Mostrar estado del historial
      console.log(` Historial actual: [${historialTemperaturas.slice(-3).join(', ')}]`);

      // Verificar si hay alerta
      const alertaEnviada = verificarAlertaTemperatura();

      if (!alertaEnviada && historialTemperaturas.length > 3) {
        // Mantener solo las últimas 3 lecturas si no hay alerta
        historialTemperaturas = historialTemperaturas.slice(-3);
      }
    }

    // Esperar antes de la siguiente lectura
    console.log(` Esperando ${INTERVALO_MS/1000} segundos para la siguiente medición...\n`);
    await sleep(INTERVALO_MS);
  }
}

// Manejo de errores global
process.on('unhandledRejection', (error) => {

```

```
    console.error(' Error no manejado:', error);  
  });  
  
  // Iniciar el monitoreo  
  monitorearTemperatura().catch(error => {  
    console.error(' Error fatal en el monitoreo:', error);  
  });
```

## Resultados

```
🌡 Temperatura actual: 43°C
📄 Historial actual: [43]
⌚ Esperando 3 segundos para la siguiente medición...

🌡 Temperatura actual: 18°C
📄 Historial actual: [43, 18]
⌚ Esperando 3 segundos para la siguiente medición...

🌡 Temperatura actual: 31°C
📄 Historial actual: [43, 18, 31]
🚨 ¡ALERTA! Tres temperaturas consecutivas superiores a 10°C
🔄 Historial limpiado. Reiniciando monitoreo...

⌚ Esperando 3 segundos para la siguiente medición...

✅ Alerta enviada a Discord exitosamente
🌡 Temperatura actual: 41°C
📄 Historial actual: [41]
⌚ Esperando 3 segundos para la siguiente medición...

🌡 Temperatura actual: 34°C
📄 Historial actual: [41, 34]
⌚ Esperando 3 segundos para la siguiente medición...

🌡 Temperatura actual: 26°C
📄 Historial actual: [41, 34, 26]
🚨 ¡ALERTA! Tres temperaturas consecutivas superiores a 10°C
🔄 Historial limpiado. Reiniciando monitoreo...

⌚ Esperando 3 segundos para la siguiente medición...

✅ Alerta enviada a Discord exitosamente
🌡 Temperatura actual: 25°C
📄 Historial actual: [25]
⌚ Esperando 3 segundos para la siguiente medición...
```



## Monitor de Temperatura APP 22:07



### ALERTA: Temperatura Alta Detectada

Se han detectado **3 temperaturas consecutivas** superiores a 10°C



#### Lecturas Consecutivas



Lectura 1: **43°C**



Lectura 2: **18°C**



Lectura 3: **31°C**



#### Fecha y Hora

martes, 30 de septiembre de 2025, 10:07:52 p.m. GMT-6

Sistema de Monitoreo de Temperatura • hoy a las 22:07



### ALERTA: Temperatura Alta Detectada

Se han detectado **3 temperaturas consecutivas** superiores a 10°C



#### Lecturas Consecutivas



Lectura 1: **41°C**



Lectura 2: **34°C**



Lectura 3: **26°C**



#### Fecha y Hora

martes, 30 de septiembre de 2025, 10:08:03 p.m. GMT-6

Sistema de Monitoreo de Temperatura • hoy a las 22:08