Guía Didáctica: Aplicación del Clima con Node-RED

Introducción: ¿Qué vamos a construir?

Crearemos una aplicación web que permite ingresar el nombre de cualquier ciudad y obtener información meteorológica actual. La aplicación tiene dos pasos principales:

1. **Geocodificación**: Convertir el nombre de la ciudad en coordenadas (latitud y longitud)
2. **Consulta del clima**: Usar esas coordenadas para obtener datos meteorológicos

---

Conceptos Clave: Entendiendo las APIs

¿Qué es una API?

Como bien dijiste, una API es un medio para compartir información entre aplicaciones. En nuestro caso:

- **Tu aplicación Node-RED** solicita datos
- **OpenWeatherMap** responde con información del clima

¿Por qué necesitamos dos llamadas API?

- **API de Geocodificación**: "Paraná" → Coordenadas (31.7319°S, 60.5297°W)
- **API del Clima**: Coordenadas → Datos meteorológicos

Esto es común en aplicaciones reales: a menudo necesitas combinar diferentes servicios para obtener la información completa.

Paso 1: Configuración Inicial

Nodos Necesarios

Antes de empezar, asegúrate de tener instalados estos nodos adicionales:

```
npm install node-red-dashboard
npm install @flowfuse/node-red-dashboard
```

Paso 2: Creando la Interfaz de Usuario

2.1 Nodo de Entrada de Texto

1. Arrastra un nodo **ui-text-input** al canvas
2. Configúralo así:
   - **Nombre**: "Ubicación"
   - **Etiqueta**: "Ingresar ubicación"
   - **Grupo**: Crear nuevo grupo "Información general"

¿Por qué empezamos aquí? La interfaz de usuario es el punto de entrada. El usuario escribe una ciudad aquí.

Paso 3: Primera API - Geocodificación

3.1 Preparando la URL

1. Agrega un nodo **template**
2. Configúralo:

- **Nombre**: "Armar URL Geocoding"
- **Propiedad**: `msg.url`
- **Plantilla**:

```
http://api.openweathermap.org/geo/1.0/direct?q=
{{payload}}&limit=5&appid=029e2971fd993ff3081a6fdf47ad6f19&lang=es
```

Explicación de la URL:

- `q={{payload}}`: El nombre de la ciudad que escribió el usuario
- `limit=5`: Máximo 5 resultados
- `appid=...`: Tu clave API (necesitas registrarte en OpenWeatherMap)
- `lang=es`: Respuestas en español

3.2 Realizando la Llamada API

1. Agrega un nodo **http request**
2. Configúralo:
   - **Nombre**: "Geocoding API"
   - **Método**: GET
   - **Devolver**: objeto JSON parseado

¿Qué sucede aquí? Node-RED toma la URL que armamos y hace una petición HTTP. OpenWeatherMap devuelve un JSON con información de ubicaciones que coinciden con tu búsqueda.

3.3 Procesando la Respuesta

1. Agrega un nodo **change**

2. Configúralo para extraer datos:
   - **Nombre**: "Extraer datos"
   - Reglas:
     - `msg.lat` = `msg.payload[0].lat`
     - `msg.lon` = `msg.payload[0].lon`
     - `msg.ciudad` = `msg.payload[0].name`
     - `msg.provincia` = `msg.payload[0].state`
     - `msg.pais` = `msg.payload[0].country`

¿Por qué `payload[0]`? La API devuelve un array. Tomamos el primer resultado `[0]` que suele ser el más relevante.

Paso 4: Segunda API - Datos del Clima

4.1 Preparando la URL del Clima

1. Agrega otro nodo **template**
2. Configúralo:
   - **Nombre**: "Current Weather URL"
   - **Propiedad**: `msg.url`
   - **Plantilla**:

```
https://api.openweathermap.org/data/2.5/weather?lat={{lat}}&lon=
{{lon}}&appid=029e2971fd993ff3081a6fdf47ad6f19&units=metric&lang=sp
```

Fíjate: Ahora usamos `{{lat}}` y `{{lon}}` que obtuvimos de la primera API.

4.2 Segunda Llamada API

1. Agrega un nodo **http request**
2. Configúralo:
   - **Nombre**: "OpenWeather API"
   - **Método**: GET
   - **Devolver**: objeto JSON parseado

Paso 5: Mostrando los Datos

5.1 Ubicación Mostrada

1. Agrega un nodo **function**:

```
javascript

msg.payload = msg.ciudad + ", " + msg.provincia + ", " + msg.pais;
return msg;
```

2. Conecta a un nodo **ui-text** para mostrar la ubicación

5.2 Datos Meteorológicos

Crea nodos **template** para formatear cada dato:

Temperatura:

```
{{payload.main.temp}} ºC
```

Sensación Térmica:

```
{{payload.main.feels_like}} °C
```

Humedad (usando function):

```
javascript

msg.payload = msg.payload.main.humidity + " %";
return msg;
```

Presión (usando function):

```
javascript

msg.payload = msg.payload.main.pressure + " hPa";
return msg;
```

Viento (usando function):

```
javascript

msg.payload = msg.payload.wind.speed + " m/s";
return msg;
```

Descripción:

```
javascript

msg.payload = msg.payload.weather[0].description;
return msg;
```

5.3 Ícono del Clima

Agrega un nodo **ui-template**:

```html
<img
  :src="`https://openweathermap.org/img/wn/${msg.payload.weather[0].icon}@2x.png`"
  style="width: 70px; height: 70px;"
  alt="weather icon"
/>
```

Paso 6: Organizando el Flujo con Links

6.1 ¿Por qué usar Link Nodes?

Los **link in** y **link out** mantienen el canvas limpio cuando necesitas enviar datos a múltiples destinos.

1. Después de "Extraer datos", agrega un **link out**
2. Conecta a un **link in** que alimente los nodos de visualización

---

Paso 7: Configuración de la Interfaz

7.1 Creando el Dashboard

1. En la configuración, crea:
   - **UI Base**: "My Dashboard"
   - **UI Page**: "Weather App"
   - **UI Group**: "Información general"

7.2 Ordenando Elementos

Configura el orden de visualización:

1. Campo de entrada
2. Ubicación mostrada
3. Ícono del clima
4. Temperatura
5. Sensación térmica
6. Viento
7. Humedad
8. Presión
9. Descripción

Conceptos Avanzados Explicados

JSON y Navegación de Datos

Cuando la API devuelve:

```json
```

```
{
  "main": {
    "temp": 25.3,
    "humidity": 65
  },
  "weather": [
    {
      "description": "cielo claro",
      "icon": "01d"
    }
  ]
}
```


Accedes así:

- Temperatura: `msg.payload.main.temp`
- Descripción: `msg.payload.weather[0].description`

Manejo de Errores

Pregunta para reflexionar: ¿Qué pasaría si el usuario escribe una ciudad que no existe?

Considera agregar nodos **catch** para manejar errores de API.

Rate Limiting

Las APIs tienen límites. OpenWeatherMap permite 1000 llamadas gratis por día.

Preguntas de Comprensión

1. **¿Por qué necesitamos la clave API (`appid`)?**

2. **¿Qué ventaja tiene usar `units=metric`?**
3. **¿Por qué usamos templates para algunas transformaciones y functions para otras?**
4. **¿Cómo modificarías el flujo para mostrar el pronóstico de 5 días?**

Próximos Pasos

Una vez que tengas funcionando esta aplicación básica, podrías:

1. **Agregar más datos**: UV index, visibilidad, punto de rocío
2. **Mejorar el diseño**: Colores, íconos personalizados
3. **Añadir pronóstico**: API de forecast de OpenWeatherMap
4. **Guardar favoritos**: Usar context storage
5. **Gráficos**: Mostrar tendencias de temperatura

Recursos Adicionales

- OpenWeatherMap API Documentation
- Node-RED Cookbook
- Node-RED Dashboard Guide

¡Felicidades! Has creado una aplicación completa que combina múltiples APIs, procesamiento de datos y una interfaz web interactiva. Este es el tipo de integración que verás en aplicaciones profesionales.