**Plan de pruebas Current Weather Data API**

MEDELLIN | COLOMBIA

**ELABORADO POR:**

Mario ALEJANDRO Benitez Orozco

2025

Tabla de contenido

[INTRODUCCIÓN 2](#_Toc199975513)

[OBJETIVO 2](#_Toc199975514)

[JUSTIFICACIÓN 2](#_Toc199975515)

[ALCANCE 2](#_Toc199975516)

[ESTRATEGIA DE PRUEBAS 3](#_Toc199975517)

[HERRAMIENTAS 3](#_Toc199975518)

[TIPOS DE VALIDACIONES/ASERCIONES 3](#_Toc199975519)

[CRITERIOS DE ACEPTACIÓN 3](#_Toc199975520)

[RIESGOS Y CONSIDERACIONES 3](#_Toc199975521)

[INTEGRACIÓN CI/CD RECOMENDADA – ARQUITECTURA & COSTO CERO 4](#_Toc199975522)

[ESCENARIOS DE PRUEBA DETALLADOS 5](#_Toc199975523)

[CONCLUSIONES 5](#_Toc199975524)

# **INTRODUCCIÓN**

Este documento presenta el plan de pruebas para el endpoint Current Weather Data API de OpenWeatherMap. El propósito es asegurar la calidad, funcionalidad, robustez y precisión de la API frente a diversos escenarios de consulta.

Así también tener claridad de una de tantas posibles soluciones que pueden proponerse para darle solución a este caso.

# **OBJETIVO**

Validar el comportamiento y la respuesta del endpoint /weather para solicitudes tanto válidas como inválidas, garantizando la entrega de información confiable y el manejo adecuado de errores.

# **JUSTIFICACIÓN**

La correcta validación de servicios externos como OpenWeatherMap es crucial para sistemas que dependen de información meteorológica en tiempo real. Un plan de pruebas sólido asegura que cualquier integración será resiliente a cambios, errores y condiciones inesperadas, minimizando el riesgo de fallos en producción.

# **ALCANCE**

* Endpoint: GET /weather
* Entradas: ciudad, coordenadas, código postal, API Key, unidades, idioma.
* Cobertura: Respuestas exitosas, manejo de errores, validación de datos, tiempos de respuesta.

# **ESTRATEGIA DE PRUEBAS**

Se automatizarán todos los escenarios con Cypress, abarcando happy paths y edge cases, con ejecución local y en CI/CD.

# **HERRAMIENTAS**

* Cypress (automatización principal).
* Postman (manual/exploratorio, opcional)
* GitHub Actions / Jenkins (CI/CD)
* Docker (entorno controlado)

# **TIPOS DE VALIDACIONES/ASERCIONES**

* Códigos HTTP: 200, 400, 401, 404
* Validación de esquema JSON
* Precisión de datos meteorológicos
* Manejo de errores y mensajes claros
* Headers correctos (Content-Type)

# **CRITERIOS DE ACEPTACIÓN**

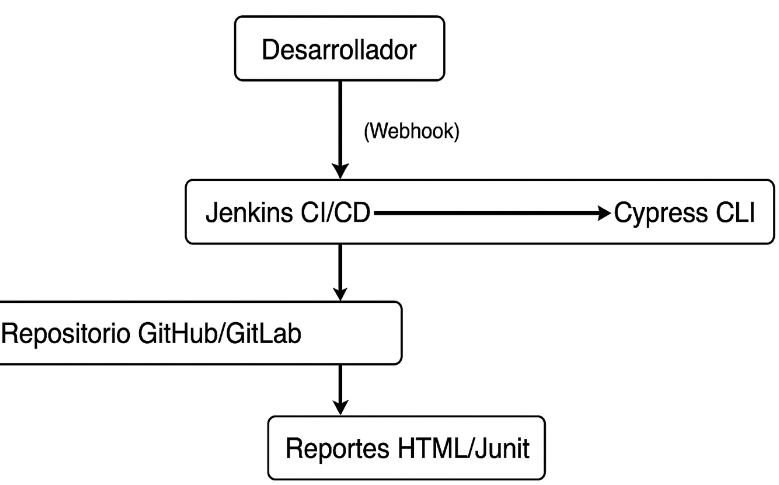
El endpoint debe responder correctamente a entradas válidas y manejar adecuadamente los errores, con datos meteorológicos completos y exactos según la documentación oficial.

# **RIESGOS Y CONSIDERACIONES**

* Cambios inesperados en la API externa
* Límites de uso por API Key
* Diferencias regionales de datos meteorológicos

# **INTEGRACIÓN CI/CD RECOMENDADA – ARQUITECTURA & COSTO CERO**

Se recomienda integrar Cypress en un pipeline CI/CD gratuito usando Jenkins y Docker. La arquitectura contempla triggers automáticos al realizar un push o pull request, construcción de imágenes, ejecución de pruebas y generación de reportes, con notificación al equipo.



# **ESCENARIOS DE PRUEBA DETALLADOS**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Escenario** | **Entrada de ejemplo** | **Resultado esperado** |
| 1 | Consulta por ciudad válida | q=London | 200 OK, datos meteorológicos correctos |
| 2 | Consulta por coordenadas válidas | lat=40.71&lon=-74.01 | 200 OK, datos correspondientes |
| 3 | Consulta por código postal válido | zip=94040,us | 200 OK, datos correctos |
| 4 | Consulta en diferentes unidades | units=imperial / units=metric | 200 OK, datos en unidad solicitada |
| 5 | Consulta en idioma diferente | lang=es | 200 OK, respuesta en español |
| 6 | Ciudad inexistente | q=NoExisteCiudad | 404 Not Found, mensaje de error |
| 7 | Coordenadas fuera de rango | lat=999&lon=999 | 400 Bad Request o error adecuado |
| 8 | API Key inválida o ausente | Sin appid o valor incorrecto | 401 Unauthorized, error de autenticación |
| 9 | Falta de parámetros obligatorios | Solo appid | 400 Bad Request, error descriptivo |
| 10 | Formato incorrecto en parámetros | q=12345!@# | 400 Bad Request o error adecuado |
| 11 | Nombre de ciudad extremadamente largo | q=xxxxxxxxxxxxxxxxxx... | 400/404, manejo correcto de error |
| 12 | Verificación de headers de respuesta | Cualquier consulta | Header Content-Type: application/json |

# **CONCLUSIONES**

El plan propuesto cubre de manera integral la funcionalidad y resiliencia del endpoint de clima actual. La automatización con Cypress, combinada con una integración CI/CD de costo cero, garantiza calidad continua y feedback rápido para cualquier equipo de desarrollo.