

Documentación Técnica: API de Predicción de Temperatura

Equipo de Desarrollo

20 de mayo de 2025

1. Stack Tecnológico

1.1. Dependencias Exactas

- **Python:** 3.10.9
- **joblib:** 1.2.0 (para serialización del modelo)
- **numpy:** 1.24.4 (manejo de arrays numéricos)
- **pandas:** 1.5.3 (procesamiento de datos)
- **scikit-learn:** 1.1.3 (framework de ML)
- **scipy:** 1.10.1 (dependencia de scikit-learn)
- **boto3:** 1.26.0 (acceso a servicios AWS)

1.2. Infraestructura AWS

- **Lambda:** Tiempo de ejecución configurado a 1 minuto
- **Docker:** Imagen basada en `public.ecr.aws/lambda/python:3.10`
- **ECR:** Repositorio privado para la imagen Docker
- **S3:** Bucket `modelpredicts` con modelo en `modelsRepository/modelo_temperatura.pkl`
- **API Gateway:** Endpoint REST con autenticación via API Key

2. Guía de Uso

2.1. Interfaz en Postman

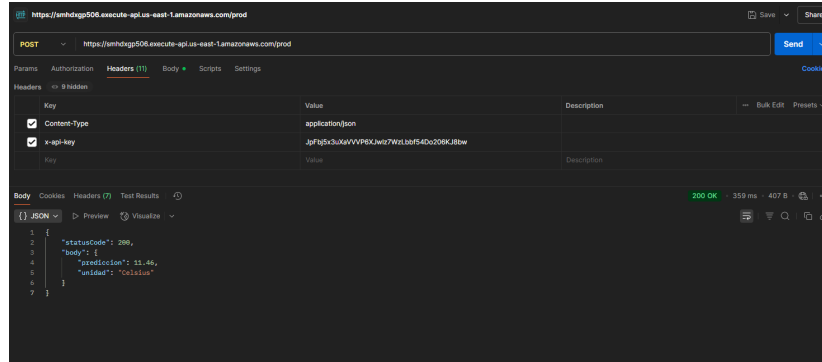


Figura 1: Configuración en Postman: Headers y respuesta

Como se muestra en la Figura 1, la solicitud requiere:

- URL: `https://smhdxgp506.execute-api.us-east-1.amazonaws.com/prod`
- Método: POST
- Headers:
 - Content-Type: `application/json`
 - x-api-key: `JpFbj5x3uXaVVVP6XJwlz7WzLbbf54Do206KJ8bw`

2.2. Cuerpo de la Petición

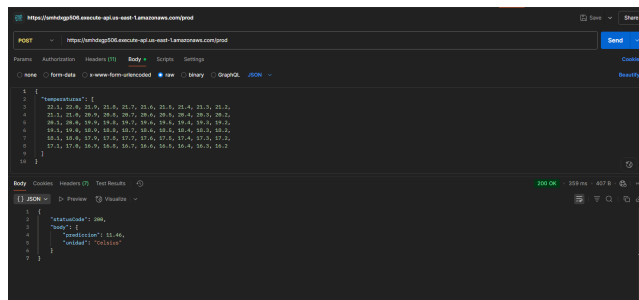


Figura 2: Ejemplo del body en formato JSON

La estructura del body (Figura 2) debe contener exactamente 60 valores:

```

1 {
2   "temperaturas": [
3     22.1, 22.0, 21.9, ..., 16.2 // 60 valores consecutivos
4   ]
5 }

```

3. Flujo de Ejecución

1. El cliente envía solicitud a API Gateway con API Key
2. API Gateway valida la autenticación y redirige a Lambda
3. Lambda descarga el modelo desde S3 (solo en cold start)
4. El modelo procesa los datos de entrada
5. Se devuelve la predicción en formato JSON

4. Ejemplo de Código

4.1. Cliente en Python

```

1 import requests
2
3 API_URL = "https://smhdxgp506.execute-api.us-east-1.amazonaws.com/
4   prod"
5 API_KEY = "JpFbj5x3uXaVVP6XJwIz7WzLbbf54Do206KJ8bw"
6
7 headers = {
8   "x-api-key": API_KEY,
9   "Content-Type": "application/json"
10 }
11
12 data = {
13   "temperaturas": [22.1, 22.0, ..., 16.2] # 60 valores
14 }
15
16 response = requests.post(API_URL, json=data, headers=headers)
17 print("Predicci n:", response.json()["body"]["prediccion"])

```

5. Consideraciones Técnicas

- **Tamaño del modelo:** 45 MB aprox. (requiere configuración especial en Lambda)
- **Tiempo de respuesta:** 300-500 ms después del cold start
- **Límites:**
 - Máximo 60 valores de entrada

- Precisión de 2 decimales
- Unidad Celsius fija