**Proyecto de Investigación Aplicada Meteo- Huascarán: Ecoturismo seguro con monitoreo y pronóstico meteorológico automatizado con machine learning en el Parque Nacional Huascarán**

**Pruebas Unitarias y de estrés**

**SERVICIO DE CONTROL DE CALIDAD AUTOMATIZADO DE DATOS EXTREMOS DE PRECIPITACIÓN DE ESTACIONES AUTOMÁTICAS USANDO INTELIGENCIA ARTIFICIAL**

**Versión 01**

| **Elaborado por:**  Diego Ernesto Paredes Chilon | **Firma:** |
| --- | --- |
| **Revisado por:** | **Firma:** |
| **Aprobado por:** | **Firma:** |

**INDICE**

[1. TÍTULO DEL PROYECTO 3](#_30j0zll)

[2. DESCRIPCIÓN 4](#_3znysh7)

[3. CONSIDERACIONES 4](#_2et92p0)

[4. MOCKUPS DEL SERVICIO WEB 4](#_tyjcwt)

[5. PRUEBAS Y GESTIÓN DE ERRORES 6](#_3dy6vkm)

[6. TABLA HISTÓRICA DE CAMBIOS 6](#_1t3h5sf)

# **TÍTULO DEL PROYECTO**

Proyecto de Investigación Aplicada “Meteo-Huascarán: Ecoturismo seguro con monitoreo y pronóstico meteorológico automatizado con machine learning en el Parque Nacional Huascarán”.

# **DESCRIPCIÓN**

Se realizaron las pruebas unitarias y pruebas de estrés del servicio web que integra los modelos algorítmicos para el control de calidad de datos de precipitaciones.

# **CONSIDERACIONES**

* 1. Supuestos

El servicio web desplegado tendrá acceso a internet y estará disponible las 24 horas del día, esto debido a que internamente se realiza un proceso asíncrono que descarga las imágenes satelitales de manera horaria, y así reducir el tiempo de predicción.

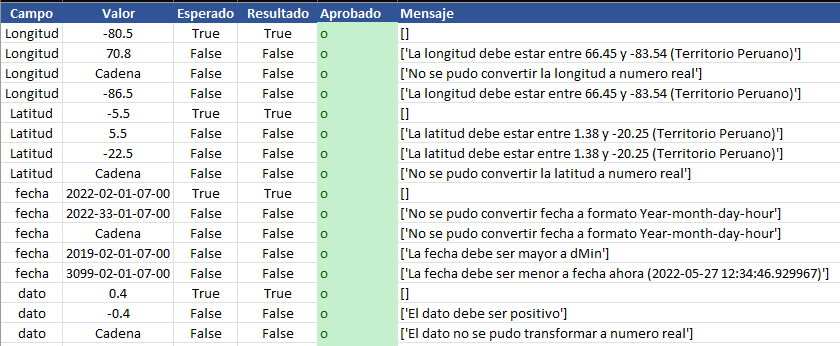
* 1. Dependencias

El servicio web se desarrollará con lenguajes de programación Python Versión 3.10. El servidor donde se instale este servicio web debe tener una versión igual o mayor.

Por otra parte, para poder descargar las imágenes satelitales, el servidor debe tener conexión a internet en todo momento. Así mismo, este servicio web depende de la disponibilidad del servidor que almacena las imágenes satelitales.

# **PRUEBAS UNITARIAS**

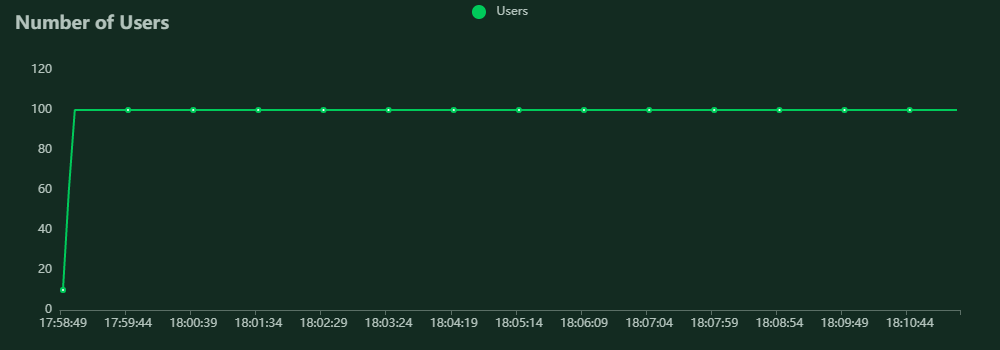
En cuanto a las pruebas unitarias se realizó con lenguaje Python puro. Los resultados se guardaron en un archivo Excel. La Tabla 14 muestra las pruebas que se realizaron con diferentes valores en los parámetros de entrada, como se puede visualizar todas las pruebas fueron satisfactorias, además los mensajes devueltos por el servicio web indica cuál fue el error que sucedió.

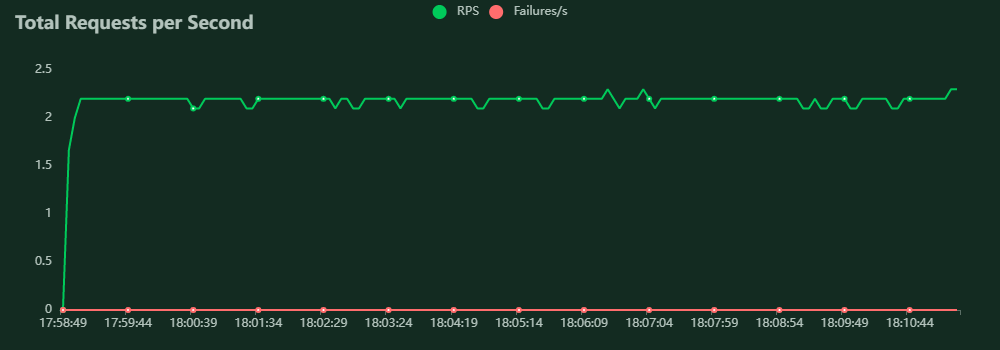
**

# **PRUEBAS DE ESTRÉS**

Para las pruebas de estrés, se realizó con Locust, una librería de Python que permite realizar pruebas de estrés. Las pruebas se realizaron a lo largo de aproximadamente 10 minutos, este comenzó a las 17:58 horas y terminó a las 18:10 horas, con un máximo de 100 peticiones al mismo tiempo. Con la herramienta Locust se pudo simular a 100 usuarios realizar peticiones al mismo tiempo al servicio web. Se comenzó incrementando los usuarios de 10 en 10 hasta llegar a tener 100 usuarios constantes. En total se logró ejecutar 1660 peticiones dentro de los diez minutos que se realizó el experimento, en los cuales no se encontró peticiones fallidas.

La primera figura muestra la cantidad de usuarios que realizaban peticiones para cada tiempo., mientras que la segunda figura muestra los pedidos que se hicieron por segundo al servicio web para cada tiempo, la línea roja indica las peticiones exitosas, mientras la línea roja los fallos.





# **TABLA HISTÓRICA DE CAMBIOS**

| **Versión** | **Fecha** | **Detalle de cambios** |
| --- | --- | --- |
| 01 | 2024/12/05 | Versión inicial |