**Sistema de control automatizado de datos extremos de precipitación de estaciones automáticas usando inteligencia artificial**

**Manual Técnico: MA-DMA-001**

**Versión: 01**

|  |  |
| --- | --- |
| **Elaborado por:**    Diego Ernesto Paredes Chilón | **Firma:** |

**INDICE**

[1. OBJETIVO 3](#_heading=h.30j0zll)

[2. ALCANCE 3](#_heading=h.3znysh7)

[3. DEFINICIONES Y SIGLAS 3](#_heading=h.2et92p0)

[4. DESARROLLO 3](#_heading=h.tyjcwt)

[**4.1 Requerimientos Técnicos: Hardware 3**](#_heading=)

[**4.2 Requerimientos Técnicos: Software 3**](#_heading=)

[**4.3 Herramientas para el desarrollo 3**](#_heading=)

[**4.4 Instalación de la Aplicación 4**](#_heading=)

[**4.5 Arquitectura y Módulos 4**](#_heading=h.3dy6vkm)

[**4.6 Diseño de la base de datos 4**](#_heading=)

[5. TABLA HISTÓRICA DE CAMBIOS 4](#_heading=h.1t3h5sf)

# OBJETIVO

El presente documento se realizó con la finalidad de detallar los requerimientos básicos, instalación, arquitectura y diseño del servicio web que integra el modelo para el control de calidad de datos de precipitaciones extremas de estaciones meteorológicas automáticas de SENAMHI.

# ALCANCE

El documento abarca la instalación del servicio web de manera local, así como el detalle del diseño y arquitectura del mismo.

# DEFINICIONES Y SIGLAS

* UI: Interfaz de usuario (User Interface)

# DESARROLLO

# Requerimientos Técnicos: Hardware

El servicio web ha sido probado satisfactoriamente con un procesador AMD Ryzen 5 3600 y 8 GB de memoria ram.

Adicionalmente, el equipo donde se instale/desarrolle el servicio web debe tener acceso a internet para poder obtener las imágenes satelitales.

# Requerimientos Técnicos: Software

Los software necesarios para la instalación son los siguientes:

* Windows 10
* Python 3.10
* PIP (Se instala por defecto con python)

En caso de las demás librerías, estas se especifican en el archivo requirements.txt, las cuales se instalarán con PIP y Python.

# Herramientas para el desarrollo

* Para el desarrollo del servicio web se utilizó PyCharm, sin embargo se puede desarrollar en cualquier IDE o editor de código.
* Un navegador web para realizar las pruebas (Google Chrome, FireFox, etc.)

# Instalación de la Aplicación

Para el correcto funcionamiento del servicio web se necesita primero tener instalado:

* Python (Recomendado 3.10)
* Pip (Recomendado última versión disponible)

Es recomendado crear un *virtual environment* antes de comenzar las instalaciones. Mas información en <https://docs.python.org/3/tutorial/venv.html>

Una vez instalado las dos herramientas previas e inicializado el virtual environment, se debe descargar el código fuente en una carpeta (GitHub del repositorio: [https://github.com/ShounenGakusei/QC](https://github.com/ShounenGakusei/SistemaQC) )

Ahora se instalarán las librerías auxiliares. Esto se puede realizar con la herramienta pypi, ejecutando en la línea de comandos: pip install -r requirements.txt



# Arquitectura y Módulos

El servicio web consta de un solo módulo. En cuanto a la estructura de archivos, se siguió la estructura básica de Flask (Herramienta para desarrollar servicios webs):

* Carpeta Templates: En esta carpeta se guardan los archivos HTML (UI).
* Carpeta static: En esta carpeta se guarda Los archivos CSS y JS

Adicionalmente se tiene carpetas propias:

* Carpeta Modelos: Carpeta que almacena el modelo de deep-learning
* Carpeta logs: Se guardan los mensajes y errores del sistema
* Imágenes: Se utiliza para guardar las imágenes satelitales históricas.
* Utils: En esta carpeta se guardan los métodos auxiliares para el funcionamiento del servicio web.

Finalmente toda la lógica general del servicio web se encuentra en el archivo Deploy.py.

# Diseño de la base de datos

No aplica.

# TABLA HISTÓRICA DE CAMBIOS

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Detalle de cambios** |
| 01 | 2023/05/24 | Versión inicial |
| 02 | 2025/12/19 | Versión desplegada |