UNIVERSIDAD NACIONAL AGRARIA

LA MOLINA

FACULTAD DE CIENCIAS

"AUTOMATIZACIÓN DE LA GESTIÓN DE DATOS METEOROLÓGICOS USANDO PLOTLY DASH"

Trabajo de Suficiencia Profesional para Optar el Título de:

INGENIERA METEORÓLOGA

ROSAMARÍA PÉREZ BELLIDO

Lima – Perú

2024

AUTOMATIZACIÓN DE LA GESTIÓN DE DATOS METEOROLÓGICOS USANDO PLOTLY DASH

Dr. Ernesto Ever Menacho Casimiro

INFORME DE ORIGINALIDAD

10%	10%	2%	%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

ENCONTRAR COINCIDENCIAS CON TODAS LAS FUENTES (SOLO SE IMPRIMIRÁ LA FUENTE SELECCIONADA)

4%

★ www.senamhi.gob.pe

Fuente de Internet

Excluir citas	Activo	Excluir coincidencias	< 10 words
Excluir bibliografía	Activo		

UNIVERSIDAD NACIONAL AGRARIA
LA MOLINA

FACULTAD DE CIENCIAS

“AUTOMATIZACIÓN DE LA GESTIÓN DE DATOS METEOROLÓGICOS USANDO PLOTLY DASH”

Trabajo de Suficiencia Profesional para Optar el Título Profesional de:

INGENIERA METEORÓLOGA

Presentado por:

ROSAMARÍA PÉREZ BELLIDO

Sustentada y aprobado por el siguiente jurado:

Maest.Cs. Alessandri Canchoa Quispe PRESIDENTE	Dr. Alexis Nicolás Ibáñez Blancas MIEMBRO
Mg. Sc. Julio Alfonso Arakaki Kiyán MIEMBRO	Dr. Ernesto Ever Menacho Casimiro ASESOR

DEDICATORIA

A mi familia, por su constante apoyo y amor incondicional durante todo el proceso. Su confianza y palabras de aliento han sido fundamentales para la realización de este trabajo.

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a todas las personas e instituciones que hicieron posible la realización de este trabajo.

A mi asesor, el Dr. Ever Menacho por su motivación durante mi orientación, paciencia y apoyo académico durante todo el proceso de investigación.

Y a mi familia, por su amor incondicional, su comprensión y su constante apoyo emocional a lo largo de este arduo proceso.

A todos, mi más profundo agradecimiento.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

El presente trabajo describe la sistematización de la gestión de información meteorológica mediante la creación de un dashboard utilizando Python y su librería Dash. La herramienta de visualización se desarrolló durante el periodo de agosto de 2019 a septiembre de 2022, mientras se ocupaba el cargo de Analista meteorológico en la Dirección Zonal 3 – Cajamarca del Servicio Nacional de Meteorología e Hidrología (SENAMHI). El dashboard ofrece una visualización clara y concisa de los datos meteorológicos, facilitando las actividades diarias del analista al proporcionar una vista completa de los datos relevantes en un solo lugar. Esto permite realizar análisis en tiempo real, observar patrones con mayor facilidad y tomar decisiones informadas de manera ágil, gracias a los gráficos y mapas interactivos generados mediante un script en Python. Esta herramienta aprovecha las ventajas del lenguaje de programación Python, reconocido por su facilidad de uso y su amplia gama de bibliotecas y herramientas para analizar y visualizar datos, asegurando una implementación eficiente y una interfaz interactiva para el dashboard.

Palabras clave: Python, dashboard, herramienta, visualizador

ABSTRACT

This work describes the systematization of meteorological information management by creating a dashboard using Python and its Dash library. The visualization tool was developed during the period from August 2019 to September 2022, while held the position of Meteorological Analyst in Zonal Directorate 3 – Cajamarca of the National Meteorology and Hydrology Service (SENAMHI). The dashboard offers a clear and concise visualization of weather data, facilitating the analyst's daily activities by providing a complete view of relevant data in one place. This allows you to perform real-time analysis, observe patterns more easily, and make informed decisions quickly, thanks to interactive graphs and maps generated using a Python script. This tool takes advantage of the Python programming language, recognized for its ease of use and its wide range of libraries and tools for analyzing and visualizing data, ensuring efficient implementation and an interactive interface for the dashboard.

Keywords: Python, dashboard, tool, visualizer

I. INTRODUCCIÓN

1.1. Problemática

Crear un *dashboard* como herramienta para el manejo de información meteorológica ofrece numerosas ventajas, como proporcionar una visualización clara y concisa de los datos meteorológicos, lo que facilitará las actividades diarias del analista o pronosticador, ya que le permitirá tener una vista completa de los datos meteorológicos relevantes en un solo lugar, por lo que podrá realizar análisis en tiempo real y observar patrones más fácilmente, para tomar decisiones informadas de manera más ágil, gracias a los gráficos y mapas interactivos generados a través de un script de Python, lenguaje de programación que es reconocido por su facilidad de uso y amplia gama de bibliotecas y herramientas para analizar y visualizar datos, y que asegurará una implementación eficiente y una interfaz interactiva para el *dashboard*.

1.2. Objetivos

1.2.1. Objetivo principal

Construcción de una interfaz visual o *dashboard*, usando la librería *Dash* de Python, para el análisis de la temperatura máxima, mínima y precipitación, información obtenida de las estaciones meteorológicas convencionales y automáticas de la jurisdicción de la Dirección Zonal 3 - Cajamarca sur, del SENAMHI, y recopiladas en la web de Red de voz y data (LVERA) de la institución.

1.2.2. Objetivos específicos

- Elaborar un algoritmo con el software libre Python, para la generación de resúmenes (diario y mensual) de la temperatura máxima, temperatura mínima y precipitación.
- Ejecutar el *dashboard* con información real de las estaciones meteorológicas de la jurisdicción de la Dirección Zonal 3 – Cajamarca sur, del SENAMHI.

II. REVISIÓN DE LITERATURA

2.1. Estación meteorológica

Las estaciones meteorológicas son sitios donde se realizan mediciones y observaciones puntuales de diversos parámetros meteorológicos con instrumentos específicos, con el objetivo de determinar el comportamiento atmosférico (PCE Instruments, 2023).

Para este trabajo se considerará la información de estaciones meteorológicas convencionales y automáticas, que registran variables como temperaturas extremas, precipitación, humedad relativa, dirección y velocidad de viento, presión, radiación solar incidente, entre otras.

2.2. Variables meteorológicas

A continuación, se describen las principales variables meteorológicas a estudiar:

2.2.1. Temperatura

Se utiliza la unidad de medida en grado centígrado (°C), que corresponde a las lecturas directas efectuadas en los termómetros de extremas. La lectura de la temperatura mínima se realiza a las 07:00 horas, y la de la temperatura máxima, a las 19:00 horas.

2.2.2. Precipitación

Se mide la acumulación de lluvia, por unidad de área. Una acumulación de 1 mm corresponde al volumen de 1 litro por metro cuadrado de superficie. Además, se expresa en milímetros (mm) y las lecturas se efectúan diariamente a las 07:00 horas.

2.3. Parámetros estadísticos

2.3.1. Media aritmética.

Es el cociente de la suma de los valores individuales entre el número de valores.

$$Media = \frac{X_1 + X_2 + \cdots + X_3}{n} = \frac{\sum_{k=1}^{n} X_k}{n}$$

Donde:

X1, X2, … Xk: Elementos individuales

n: Total de elementos

2.3.2. Normal climática

Es la media, o promedio aritmético, de datos climatológicos calculados en un periodo de 30 años, por ejemplo, desde el 1 de enero de 1981 al 31 de diciembre de 2010. (OMM, 2007)

2.3.3. Anomalías.

Una anomalía es la diferencia entre un valor observado y la media, a largo plazo y pueden ser positivas o negativas. Se calcula siguiendo la siguiente fórmula:

$$\boldsymbol{Anomalía = observación\ actual - media\ climatológica\ del\ periodo\ base}$$

Además, puede ser calculada a plazos mensuales, semanales, estacionales o de cualquier otra duración.

2.4. Caracterización

Consiste en clasificar los datos diarios de la variable meteorológica (temperatura máxima, mínima o precipitación) en relación con sus umbrales (percentiles) (SENAMHI, 2014). A continuación, se muestran los rangos establecidos por SENAMHI para la caracterización de la precipitación acumulada diaria:

Tabla 1: Caracterización de extremos de precipitación

Umbrales de precipitación	Caracterización
RR/día > p99	Extremadamente lluvioso
p95 < RR/día ≤ p99	Muy lluvioso
P90 < RR/día ≤ p95	Lluvioso

Nota: RR/día es la cantidad acumulada de precipitación en 24 horas.

De manera similar, y provistos por la institución, se usarán los siguientes umbrales para las temperaturas extremas:

Tabla 2: Caracterización de temperatura máxima

Umbrales de temperatura máxima	Caracterización
Tmax > P99	Día extremadamente cálido
P95 < Tmax ≤ P99	Día muy cálido
P90 < Tmax. ≤ P95	Día cálido

Tabla 3: Caracterización de temperatura mínima

Umbrales de temperatura mínima	Caracterización
Tmin < P1	Noche extremadamente fría
P5 < Tmin ≤ P1	Noche muy fría
P10 < Tmin. ≤ P5	Noche fría

2.5. Dashboard

Un *dashboard* es una interfaz visual interactiva que muestra información resumida y visualmente atractiva.

2.6. Python

Python es un lenguaje de programación de alto nivel y de propósito general. Es conocido por su sintaxis clara y legible, lo que lo hace fácil de aprender y utilizar. Python es ampliamente utilizado en la ciencia de datos, desarrollo web, sistematización de tareas, inteligencia artificial y muchos otros campos (*What is python?*, recuperado en julio de 2023). Para este proyecto, se usará este lenguaje para desarrollar el dashboard y realizar los cálculos y visualizaciones relacionados con las variables meteorológicas descritas.

2.6.1. Dash

Es un *framework* de código abierto para crear interfaces de visualización de datos. Fue lanzada en 2017 como biblioteca de Python y ha crecido para incluir implementaciones para otros lenguajes como R, Julia y F#. Dash ayuda a los científicos de datos a crear aplicaciones web analíticas sin necesidad de conocimientos avanzados de desarrollo web. (C. Dylan, 2023)

2.6.2. Pandas

"Es una biblioteca de Python de código abierto que proporciona estructuras de datos y herramientas de análisis de datos fáciles de usar y de alto rendimiento" (*Pandas documentation*, 2023). Esta librería permite trabajar con datos tabulares, similares a hojas de cálculo para explorar datos, limpiarlos y procesarlos. En pandas, una tabla de datos se llama *DataFrame*.

2.6.3. Numpy (Numerical Python)

NumPy (2023), define el paquete como "una biblioteca Python de código abierto que se utiliza en casi todos los campos de la ciencia y la ingeniería. Es el estándar universal para trabajar con datos numéricos en Python y es el núcleo de los ecosistemas científicos de Python y PyData. El API de NumPy se usa ampliamente en Pandas, SciPy, Matplotlib, scikit-learn y la mayoría de los paquetes de ciencia de datos"

2.6.4. XlsxWriter

Es una librería de Python que permite escribir textos, número, entre otros, en hojas de cálculo de Excel. (*Creating Excel files with Python and XlsxWriter*, 2023). Este módulo admite funciones como formato y muchas más, que incluyen:

- Archivos Excel XLSX 100% compatibles.
- Formateo completo
- Validación de listas
- Formato condicional
- Integración con Pandas, entre otros.

2.7. Programación orientada a objetos

La programación orientada a objetos (OPP, por sus siglas en inglés), es un paradigma de programación que proporciona un medio para estructurar programas de modo que las propiedades y los comportamientos se agrupan en objetos individuales (GeeksforGeeks, 2016).

Su objetivo principal es unir los datos y las funciones en una sola unidad para que otra parte del código los reutilice. Este concepto es sumamente importante para el desarrollo del

dashboard, ya que se crearán clases y objetos que serán útiles en múltiples partes del código fuente.

2.8. Web scraping

Consiste en la recopilación de datos de sitios web de forma sistematizada. En el presente trabajo se usará la librería *Requests* de Python, sin embargo, existen muchas otras similares.

2.9. Entorno virtual

Es un entorno aislado que permite tener una versión específica de Python y librerías asociadas a un proyecto, sin afectar el entorno global del sistema operativo. Asimismo, tener un entorno virtual definido es útil para compartir el proyecto con otros desarrolladores o desplegarlo en distintos sistemas. Esto asegura que el proyecto funcione de manera consistente en distintos entornos.

III. DESARROLLO DEL TRABAJO

3.1. Sobre el trabajo

3.1.1. Empresa y/o Servicio

Servicio Nacional de Meteorología e Hidrología del Perú (SENAMHI), en la Dirección Zonal 3 – Cajamarca.

3.1.2. Cargo y periodo de prestación de servicios

La prestación del servicio se hizo bajo el cargo Analista meteorológico, durante el periodo de agosto de 2019 a septiembre del 2022.

3.1.3. Funciones generales del cargo

- Analizar información meteorológica y realizar el seguimiento de imágenes de satélites meteorológicos.
- Interpretar resultados de los modelos numéricos del tiempo (global y regional).
- Participar en la elaboración de pronósticos diarios de tiempo extendido a tres (03) y cinco (05) días para distritos priorizados de la jurisdicción.
- Participar en la elaboración de avisos meteorológicos, sobre la presencia de eventos extremos.
- Elaboración boletines meteorológicos mensuales y reportes meteorológicos diarios, para su difusión.
- Elaboración de pronósticos estacionales.
- Elaboración de boletines diarios y semanales de incendios forestales.
- Atención a medios de prensa (televisión y radio) de manera presencial, por videollamada o vía telefónica.

3.2. Contribuciones

Se desarrolló un visualizador (*dashboard*) con el resumen gráfico de los datos meteorológicos monitoreados diariamente.

Algunas de las funcionalidades adicionales de dicho *dashboard* incluye la exportación de manera sistematizada de un archivo en Excel con el resumen de los datos de las últimas 24 horas, que se usa como base para los reportes meteorológicos diarios. También, un archivo Excel con el resumen de los datos mensuales, que se usa como insumo para la elaboración de los boletines hidrometeorológicos mensuales, y contiene las siguientes hojas:

- *Means*: con los nombres de las estaciones meteorológicas, latitud, longitud, altitud, media mensual de las temperaturas extremas y precipitación acumulada mensual.
- *Anomalies*: con los nombres de las estaciones meteorológicas, latitud, longitud, altitud y anomalías mensuales de temperaturas extremas y precipitación.
- TMAX: con los nombres de todas las estaciones meteorológicas, número de días en lo que va del mes, y valores diarios de la temperatura máxima.
- TMIN: con los nombres de todas las estaciones meteorológicas, número de días en lo que va del mes, y valores diarios de la temperatura mínima.
- PP: con los nombres de todas las estaciones meteorológicas, número de días en lo que va del mes, y valores diarios de precipitación.
- cTmax: con los nombres de las estaciones meteorológicas que cuentan con percentiles de temperatura máxima, así como los valores diarios resaltados por colores según el umbral superado.
- cTmin: con los nombres de las estaciones meteorológicas que cuentan con percentiles de temperatura mínima, así como los valores diarios resaltados por colores según el umbral superado.
- cPP: con los nombres de las estaciones meteorológicas que cuentan con percentiles de precipitación, así como los valores diarios resaltados por colores según el umbral superado.
- Dec: con los nombres de las estaciones meteorológicas que cuentan con normales por decadiaria, así como 3 columnas (1ra-Dec, 2da-Dec, 3ra-Dec), las cuales empiezan con valores de -100% y se van actualizando diariamente.

Adicionalmente, y aunque no es parte del enfoque de este trabajo, se creó una función sencilla en Python para la descarga de datos de reanalysis del ERA5, producto del modelo europeo, a través de su interfaz de programación de aplicaciones (API, por sus siglas en inglés), y para la creación de gráficos para distintas variables atmosféricas como líneas de corriente, divergencia, humedad relativa promedio, humedad específica, y temperatura potencial equivalente. Estas figuras son generadas para la elaboración de la sección de Análisis sinóptico del boletín hidrometeorológico mensual.

3.3. Metodología para la construcción del dashboard

3.3.1. Recopilación de información de estaciones meteorológicas

Consiste en la recopilación de coordenadas de cada estación meteorológica en el área de estudio (sur de Cajamarca), en una hoja de cálculo donde se detalla también la categoría de cada estación, así como 2 columnas adicionales (Anom y Dash) en las que se indica con 0 o 1 si la estación tiene normales climáticas y si se mostrará en el dashboard o no.

Tabla 4: Ubicación de estaciones

Provincia	Distrito	Nombre	Lat	Lon	Categoria	Anom	Dash
CAJABAMBA	CACHACHI	CACHACHI	-7.45106	-78.26855	PE	1	1
CAJABAMBA	CAJABAMBA	CAJABAMBA	-7.62166	-78.05131	CO	1	1
CAJAMARCA	ASUNCION	ASUNCION	-7.32604	-78.51582	CO	1	1
CAJAMARCA	CAJAMARCA	GRANJA PORCON	-7.03753	-78.6334	CO	1	1
CAJAMARCA	CAJAMARCA	AUGUSTO WEBERBAUER	-7.1675	-78.49309	MAP	1	1
CAJAMARCA	COSPAN	COSPAN	-7.42857	-78.54106	CO	1	1
CAJAMARCA	ENCAÑADA	LA ENCAÑADA	-7.12327	-78.33314	CO	1	1
CAJAMARCA	JESUS	JESUS	-7.2457	-78.38841	CO	1	1
CAJAMARCA	MAGDALENA	MAGDALENA	-7.25346	-78.65261	CO	1	1
CAJAMARCA	NAMORA	NAMORA	-7.2006	-78.32782	CO	1	1
CAJAMARCA	SAN JUAN	SAN JUAN	-7.29756	-78.49106	CO	1	1
CELENDIN	CELENDIN	CELENDIN	-6.85292	-78.14485	CO	1	1
CONTUMAZA	CHILETE	CHILETE	-7.21968	-78.83799	PLU	0	0
CONTUMAZA	CONTUMAZA	CONTUMAZA	-7.36521	-78.82273	CO	1	1
CONTUMAZA	CONTUMAZA	CASCABAMBA	-7.38407	-78.72682	EMA	1	1
CONTUMAZA	GUZMANGO	GUZMANGO	-7.38133	-78.9033	EMA	0	1
CONTUMAZA	SAN BENITO	SAN BENITO	-7.42819	-78.92673	CO	1	1
CONTUMAZA	YONAN	MONTE GRANDE	-7.22499	-79.15323	CO	1	1
HUALGAYOC	CHUGUR	CHUGUR	-6.66667	-78.73333	CO	1	1
SAN MARCOS	GREGORIO PITA	SONDOR-MATARA	-7.23687	-78.21262	CO	1	1
SAN MARCOS	PEDRO GALVEZ	SAN MARCOS	-7.32249	-78.1727	CO	1	1
SAN MIGUEL	CATILLUC	QUILCATE	-6.82275	-78.744	CO	1	1
SAN MIGUEL	LLAPA	LLAPA	-6.97833	-78.81119	CO	1	1

Continuación …

Provincia	Distrito	Nombre	Lat	Lon	Categoria	Anom	Dash
SAN MIGUEL	SAN MIGUEL	SAN MIGUEL	-6.99684	-78.85308	CO	1	1
SAN MIGUEL	UNION AGUA BLANCA	LIVES	-7.0802	-79.04045	PLU	1	1
SAN PABLO	SAN PABLO	SAN PABLO	-7.11775	-78.83083	CO	1	1

MAP: Meteorológica agrometeorológica principal
CO: Climatológica ordinaria
PLU: Pluviométrica
EMA: Estación meteorológica automática
PE: Propósitos específicos

3.3.2. Recopilación de umbrales

Se trata de los percentiles y las normales climáticas de las temperaturas extremas y precipitación, proporcionados de manera interna y confidencial por la institución, y su compilación en una sola carpeta llamara "Umbrales" para su acceso desde los scripts de Python.

3.3.3. Diseño del dashboard

La estructura del dashboard se hizo teniendo en cuenta la información analizada diariamente, es decir, un resumen de los datos en las últimas 24 horas, así como distintos gráficos para analizar cada variable de manera individual, para mayor detalle, ver Anexo (Figura 1).

3.3.4. Establecer entorno virtual del proyecto

Para esto, se usó miniconda3, siendo esta una versión más pequeña de Anaconda, y se creó un entorno virtual llamado **dz3,** donde se instaló la versión de Python 3.8.10, y librerías como Dash, Plotly, Pandas, Numpy, entre otras.

requirements.txt

```
brotlipy==0.7.0
dash==2.9.3
dash_bootstrap_components==1.4.1
openpyxl==3.0.10
pandas==1.3.5
numpy==1.24.3
Pillow==9.3.0
py2flowchart==0.0.2
pyzmq==19.0.2
retrying==1.3.3
XlsxWriter==3.1.2
pyyaml==6.0
tqdm==4.65.0
urllib3==2.0.3
requests==2.31.0
lxml==4.9.2
```

Figura 1. Archivo de requerimientos

Además, creó un archivo README.md donde se añadieron instrucciones para instalar las librerías necesarias ya sea en el sistema operativo Linux o Windows. Es importante recalcar que, el desarrollo del dashboard fue en Linux.

Figura 2. Vista previa del archivo README.md

3.3.5. Desarrollo del proyecto

Se usó el IDE (entorno de desarrollo integrado) Visual Studio Code (VS Code) de Microsoft, debido a su libre acceso e interfaz amigable, así como la extensión Jupyter, para probar el código de manera interactiva. Antes de escribir el código se estableció una estructura de las carpetas donde se guardará cada archivo a usar y generar.

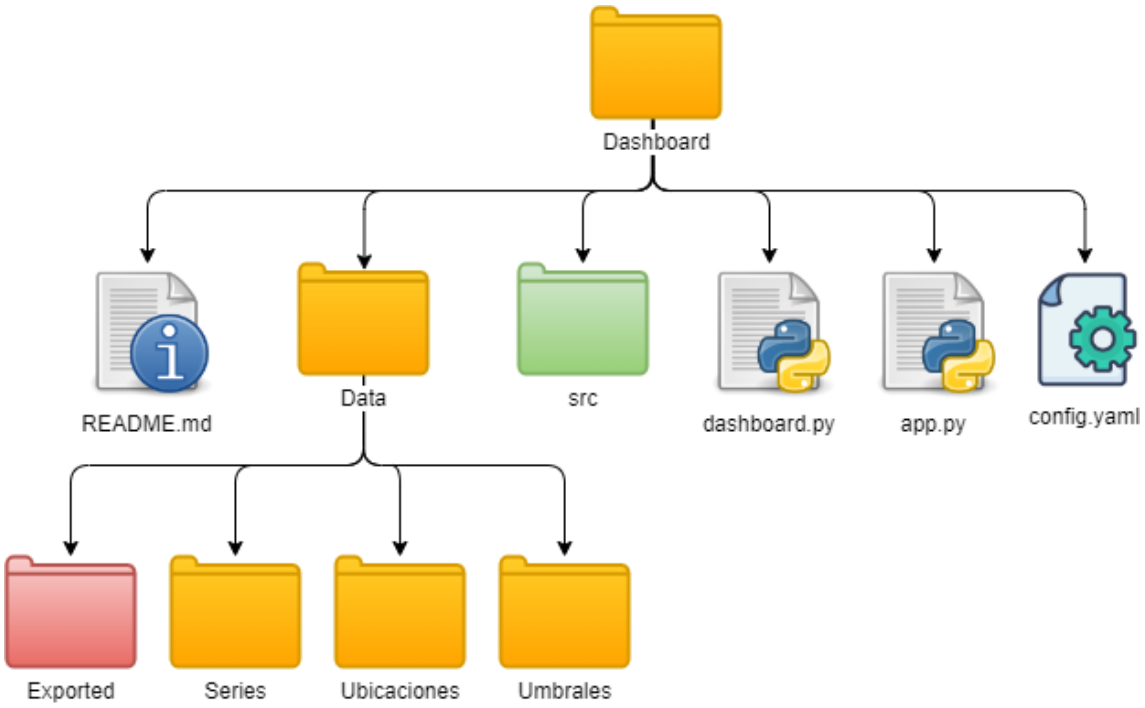


Figura 3. Estructura de directorios

3.3.6. Script para web scraping

Se creó el archivo de python **calculations.py**, donde se elaboró una función llamada *lvera*, la cual permite obtener datos meteorológicos de las estaciones convencionales de la red LVERA (Red de Estaciones Meteorológicas Automatizadas del SENAMHI). El código primero lee una hoja de cálculo que contiene una lista de estaciones y sus códigos (ver **Tabla 4**). Luego, el código solicita la página web de LVERA y extrae una tabla de datos de la página. La tabla de datos contiene información sobre la temperatura máxima, la temperatura mínima y las precipitaciones para cada estación.

Figura 4. Vista previa de la web LVERA

Cod.	Cod_Ant.	Departamento	Provincia	Distrito	Estación	Lat.	Lon.	Alt.	MIN							MAX							PP					
									28	29	30	31	01	02	03	28	29	30	31	01	02		28	29	30	31	01	02
105075	000261	AMAZONAS	BAGUA	ARAMANGO	ARAMANGO	-5.41994	-78.43553	508	16.0	16.5	16.5	16.5	15.4	15.5	13.8	32.5	30.2	-999	33.5	32.2	33.2		35.0	3.5	-999	0.0	20.4	0.0
105104	000229	AMAZONAS	BAGUA	IBAZA	CHIRIACO	-5.16144	-78.28806	323	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999		-999	-999	-999	-999	-999	-999
105079	000272	AMAZONAS	BONGARA	JAZAN	JAZAN	-5.94485	-77.97569	1354	15.1	16.4	16.6	13.1	11.2	11.3	13.1	26.3	24.5	24.7	24.8	25.9	26.5		0.0	8.1	2.2	0.0	0.0	0.0
105011	000375	AMAZONAS	CHACHAPOYAS	CHACHAPOYAS	CHACHAPOYAS	-6.23830	-77.86712	2442	9.6	9.4	9.0	9.8	7.8	7.8	6.4	-999	-999	-999	-999	-999	-999		0.0	0.0	0.0	0.0	0.0	0.0
104060	000256	AMAZONAS	CONDORCANQUI	NIEVA	SANTA MARIA DE NIEVA	-4.83039	-77.93928	225	21.6	22.2	22.4	21.9	22.3	21.5	22.1	-999	32.5	30.2	30.5	32.0	33.6		40.8	0.0	4.8	0.0	14.0	0.0
105068	000253	AMAZONAS	UTCUBAMBA	BAGUA GRANDE	BAGUA CHICA	-5.66148	-78.53396	397	24.0	23.0	22.4	22.2	19.8	22.2	20.8	31.2	29.2	31.6	31.8	32.8	31.6		0.3	0.8	0.0	0.0	19.3	0.0
105042	152206	AMAZONAS	UTCUBAMBA	JAMALCA	JAMALCA	-5.89270	-78.23390	1173	17.8	17.2	16.8	16.4	17.0	17.2	16.6	27.2	25.0	26.0	25.8	28.2	28.4		0.0	0.0	0.0	0.0	0.0	0.0
106121	003332	AMAZONAS	UTCUBAMBA	LONYA GRANDE	EL PALTO	-6.00039	-78.47097	1467	14.8	14.5	14.5	14.5	14.5	15.5	15.6	27.0	24.0	24.7	25.5	27.0	27.5		3.2	1.8	0.0	0.0	0.0	0.0
109016	000440	ANCASH	ALIA	ALIA	ALIA	-9.78164	-77.60674	3486	0.9	0.3	0.9	1.3	0.7	0.1	0.6	22.4	22.8	23.2	23.8	22.6	23.3		0.0	0.0	0.0	0.0	0.0	0.0
110040	150903	ANCASH	BOLOGNESI	CAJACAY	MAYORARCA	-10.15775	-77.34889	3335	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999		0.0	0.0	0.0	0.0	0.0	0.0
110018	000538	ANCASH	BOLOGNESI	CHIQUIAN	CHIQUIAN	-10.14763	-77.15949	3414	3.0	4.6	3.8	2.0	4.7	1.4	3.3	25.2	23.2	21.0	23.4	25.0	23.7		0.0	0.0	0.0	0.0	0.0	0.0
109014	000435	ANCASH	CASMA	BUENA VISTA ALTA	BUENA VISTA	-9.43371	-78.20825	213	19.2	17.4	19.2	16.8	15.7	16.0	16.0	27.8	27.5	25.5	28.2	28.0	25.4		0.0	0.0	0.0	0.0	0.0	0.0
109046	154108	ANCASH	HUARAZ	LA LIBERTAD	CAJAMARQUILLA	-9.63204	-77.74136	3286	7.2	6.8	6.6	7.2	7.6	7.2	6.8	24.4	21.6	19.6	22.4	24.2	22.8		0.0	0.0	0.0	0.0	0.0	0.0
109045	154107	ANCASH	HUARAZ	PARIACOTO	CHACCHAN	-9.53519	-77.77536	2266	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999		0.0	0.0	0.0	0.0	-999	-999
109040	150904	ANCASH	HUARAZ	PARIACOTO	PARIACOTO	-9.55239	-77.88761	1312	10.6	11.0	11.4	10.8	12.6	13.2	13.6	25.0	26.6	26.0	27.6	27.4	27.8		0.0	0.0	0.0	0.0	0.0	0.0
109048	154110	ANCASH	HUARAZ	PIRA	PIRA	-9.58528	-77.70719	3720	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999		0.0	0.0	0.0	0.0	0.0	0.0
109019	000445	ANCASH	HUARI	CHAVIN DE HUANTAR	CHAVIN	-9.58598	-77.17526	3140	10.2	10.2	9.8	7.2	4.4	5.8	5.0	23.0	21.5	23.5	22.5	22.5	23.5		0.0	0.0	2.0	0.0	0.0	0.0
110016	000530	ANCASH	HUARMEY	HUARMEY	HUARMEY	-10.06812	-78.16232	8	20.8	20.6	20.4	20.8	17.4	17.8	15.8	25.4	24.6	23.8	25.4	24.8	24.8		0.0	0.0	0.0	0.0	0.0	0.0
109038	150901	ANCASH	HUARMEY	MALVAS	MALVAS	-9.92723	-77.65519	3009	-999	9.0	9.0	9.2	9.0	8.5	9.4	18.0	18.0	19.0	18.6	19.0	18.0		0.0	0.0	0.0	0.0	0.0	0.0
108103	108103	ANCASH	MARISCAL LUZURIAGA	PISCOBAMBA	PISCOBAMBA II	-8.87045	-77.35072	3278	2.8	6.2	6.8	3.4	1.2	0.4	1.4	19.8	18.8	21.4	21.2	21.2	21.4		0.0	0.0	1.6	0.0	0.0	0.0
110051	155105	ANCASH	OCROS	OCROS	OCROS	-10.40457	-77.40025	3249	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999	-999		0.0	0.0	0.0	0.0	0.0	0.0
108064	004431	ANCASH	PALLASCA	CABANA	CABANA	-8.39058	-78.00461	3364	7.2	5.2	4.8	5.2	6.2	6.0	5.6	20.6	21.0	19.2	20.8	20.0	19.0		0.0	0.0	0.0	0.0	0.0	0.0
108017	000443	ANCASH	POMABAMBA	POMABAMBA	POMABAMBA	-8.82170	-77.45733	2985	3.0	3.4	3.6	3.8	2.0	2.0	2.8	24.6	21.4	23.6	24.6	24.1	24.8		0.0	0.0	0.0	0.0	0.0	0.0
109017	000441	ANCASH	RECUAY	RECUAY	RECUAY	-9.72919	-77.45365	3431	0.2	0.4	2.0	1.4	2.4	-0.6	-1.4	24.4	22.8	23.2	24.2	25.0	24.4		0.0	0.0	0.0	0.0	0.0	0.0
108047	154111	ANCASH	SIHUAS	SIHUAS	SIHUAS	-8.56667	-77.65000	2716	9.1	10.6	11.7	7.3	5.5	7.2	5.0	22.9	25.1	23.1	26.0	29.1	24.5		0.0	0.0	0.0	0.0	0.0	0.0
109018	000444	ANCASH	YUNGAY	YUNGAY	YUNGAY	-9.14189	-77.74998	2466	0.1	0.2	0.4	0.2	-999	-999	-999	26.2	26.8	28.2	28.0	28.4	27.8		0.0	0.0	0.0	0.0	0.3	0.0
113029	000677	APURIMAC	ABANCAY	CURAHUASI	CURAHUASI	-13.55253	-72.73487	2741	7.8	7.6	7.4	10.0	7.6	7.4	7.2	23.2	20.8	22.6	25.0	25.0	25.0		0.0	0.0	0.0	0.0	0.0	0.0
113225	113235	APURIMAC	ABANCAY	TAMBURCO	GRANJA SAN ANTONIO	-13.60490	-72.85690	2772	6.6	5.6	5.2	6.8	7.2	7.2	-999	20.0	20.0	20.4	22.8	22.8	20.8		0.0	0.0	0.0	0.0	0.0	-999
114117	114117	APURIMAC	AYMARAES	CHALHUANCA	AYMARAES	-14.29056	-73.25168	2964	2.4	2.8	-999	2.6	1.3	-2.6	-3.4	27.6	26.8	-999	27.4	27.6	30.1		0.0	-999	-999	1.2	0.0	0.0
113059	000811	APURIMAC	COTABAMBAS	TAMBOBAMBA	TAMBOBAMBA	-13.94492	-72.17522	3279	3.0	5.4	6.6	4.4	4.0	3.8	5.0	20.8	19.6	19.6	21.0	21.8	21.2		0.0	0.0	0.0	0.0	0.0	0.0
114108	114108	APURIMAC	GRAU	CURPAHUASI	CURPAHUASI	-14.06280	-72.66690	3535	10.2	7.4	7.9	10.1	10.3	8.6	9.6	30.1	31.0	28.6	28.7	30.1	29.1		0.0	0.0	0.0	2.3	0.0	2.0

Seguidamente, el código filtra la tabla de datos para incluir solo las estaciones que están en la hoja de cálculo, para luego crear un diccionario que contiene un conjunto de datos para cada estación con valores de temperatura máxima, temperatura mínima y las precipitaciones para los últimos 45 días. Cabe resaltar que, los datos de las estaciones automáticas (EMA) son ingresadas a los archivos CSV manualmente, ya que se requieren credenciales para obtenerlas, lo que requiere un procedimiento distinto para acceder a ellas.

Finalmente, los datos meteorológicos son exportados a un archivo CSV (si la configuración ***export_to_csv*** está establecida en *True*), dependiendo del código, nombre y provincia a la que pertenece la estación.

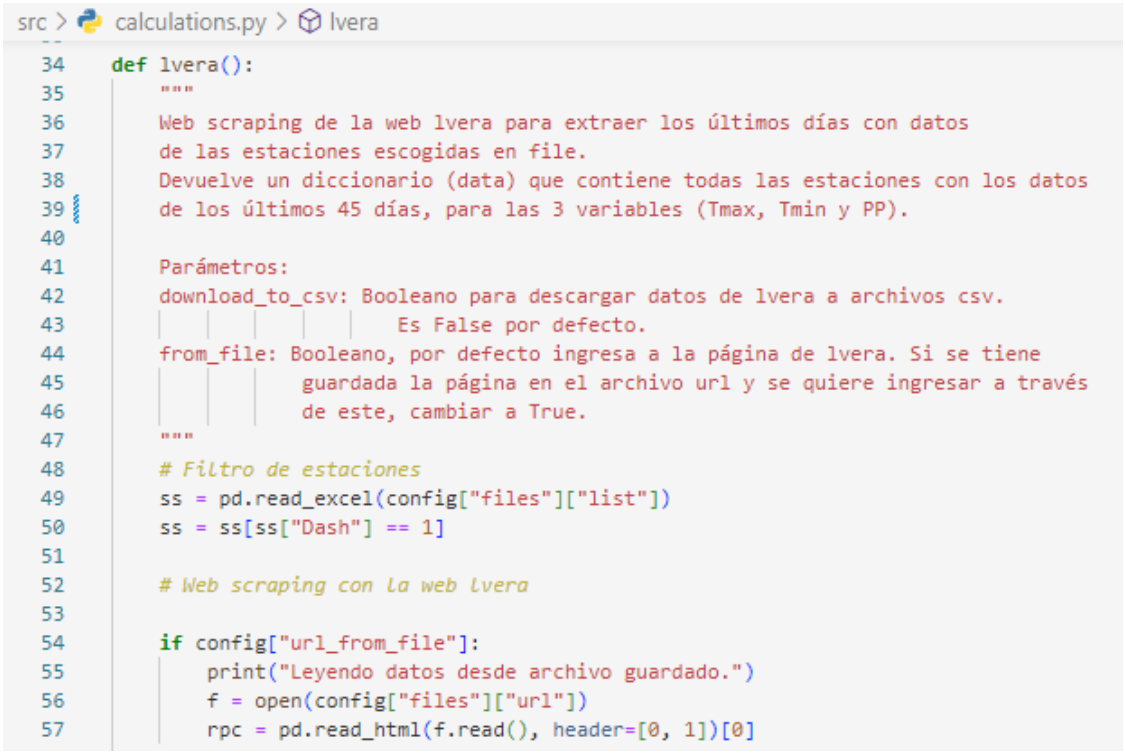


Figura 5. Vista previa del script para web scraping

3.3.7. Archivo de configuración

Para seguir las buenas prácticas en proyectos de ciencias de datos, se creó un archivo de configuración en formato YAML, el cual lleva variables y rutas de archivos que se repetirán en otros códigos, además de opciones. Por defecto y a modo de desarrollo, la opción *url_from_file* es **True,** cuando se quiere leer la web desde un archivo guardado, pero se puede cambiar a False para que se lea directamente desde la dirección HTTP; también, la opción *export_to_csv* es **False,** porque en el modo de desarrollo no es necesario exportar los

datos a csv. Adicionalmente, *paths* y *files* contienen las rutas de los archivos, así como sus nombres, de manera que se pueden modificar sin afectar el código general.

config.yaml

```yaml
# Cambiar a False cuando se quiera leer la web directamente
url_from_file: True

# True, transcribe datos de lvera a Series
export_to_csv: False

# Si solo se quiere leer datos sin generar gráficos, cambiar a False
graphs: True

# Rutas generales de archivos
paths:
  data: Data
  exported: Data/Exported
  umbrales: Data/Umbrales
  series: Data/Series

# Archivos con sus rutas
files:
  url: Data/url.html
  list: Data/Ubicaciones/lista.xlsx
  normals: Data/Umbrales/norm.xlsx
  decadiarias: Data/Umbrales/decadiarias.xlsx
  mapbox_token: Data/.mapbox_token.txt
```

Figura 6. Archivo de configuración

3.3.8. Script para generar series de tiempo

En el mismo archivo **calculations.py**, se añadió la función *series*, la cual crea una gráfica de la temperatura o precipitación para un mes dado. El código toma los siguientes argumentos:

- *var*: La variable a graficar, que puede ser "tmax", "tmin" o "pp".
- *df*: El dataframe que contiene los datos.
- *this_month*: Mes actual, si el día actual es diferente de 1, o mes anterior si el día actual es igual a 1.
- *day*: El número de días a graficar, comenzando desde el día actual.

El código primero verifica el valor del argumento *var*. Si *var* es "tmax" o "tmin", el código crea una gráfica de línea de los datos para cada estación. Luego, establece el eje x con los días del mes y el eje y con los datos de temperatura o precipitación, así como el título de la gráfica y los colores de las líneas. Por otro lado, si *var* es "pp", el código crea una gráfica de barras de la precipitación total para cada día, y continúa con los detalles de los ejes y textos del gráfico. Finalmente, el código retorna la figura.

calculations.py M

src > calculations.py > series

```python
def series(var, df, this_month, day):
    """
    Devuelve gráficos de series de tiempo para las temperaturas extremas,
    y gráfico de barras para precipitación.

    var: 'tmax', 'tmin' o 'pp'.
    df: dataframe (mx, mn o pp).
    this_month: mes anterior si es el día 1 del mes, mes actual para los demás días.
    day: día anterior si es el día 1 del mes, día actual a partir del día 2 del mes.
    """
    var = var.lower()
    if var == "tmax" or var == "tmin":
        ss = []
        for i in range(len(df.columns)):
            ss.append(
                go.Scatter(
                    x=df[df.index.month == this_month].index.day,
                    y=df.iloc[-day:, i],
                    mode="lines+markers",
                    name=df.columns[i],
```

Figura 7. Vista previa del script para generar series de tiempo

3.3.9. Script con clase para calcular anomalías y generar gráficos

La clase *Anomalies*, también en el archivo **calculations.py**, calcula las anomalías mensuales, genera gráficos de barras y mapas, para las variables de interés. Los parámetros de la clase son:

- *var*: 'tmax', 'tmin' o 'pp'.
- *df_mean*: dataframe con la media o acumulado mensual.
- *this_month*: mes anterior si es el día 1 del mes, mes actual para los demás días.
- *file*: archivo 'lista' con todas las estaciones.
- *file_w_normals*: archivo 'norm' con los valores de las normales climáticas.

El método *calculate_anomalies* calcula los valores de las anomalías mensuales para las estaciones especificadas de acuerdo con el parámetro *var*; el método *bars_fig* genera un gráfico de barras de las anomalías; y el método *maps* genera un mapa de las anomalías.

Además, el código utiliza la biblioteca *plotly* para generar los mapas. La biblioteca *plotly* permite crear visualizaciones interactivas, como mapas, que se pueden compartir en línea.

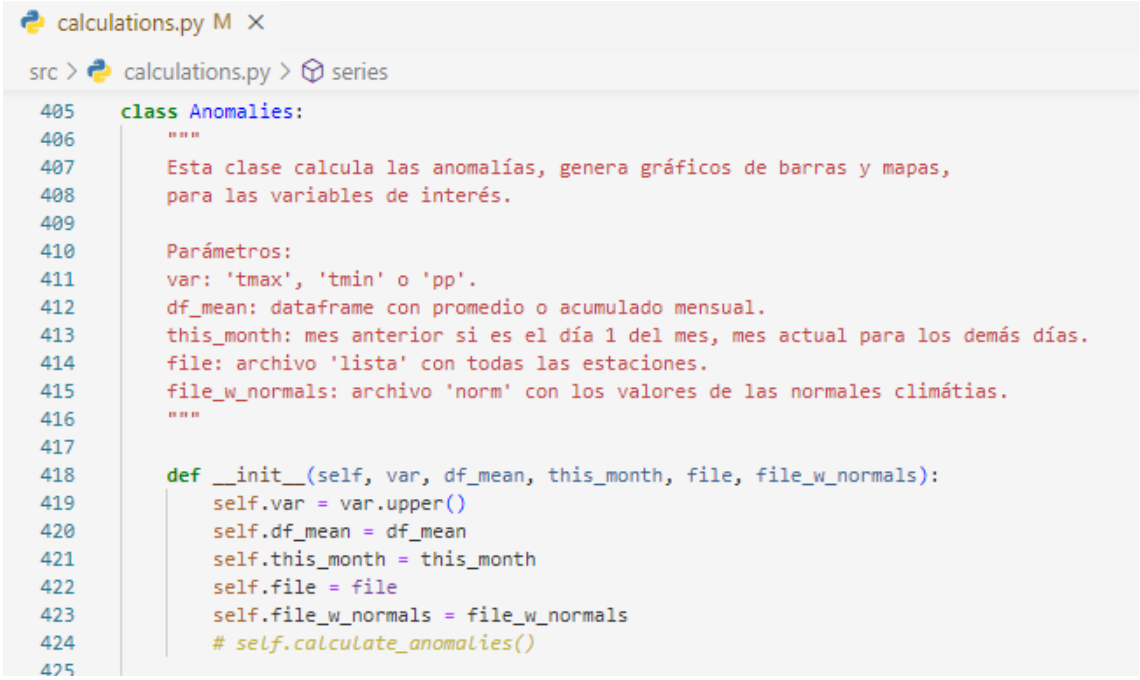


Figura 8. Vista previa del script con clase calcular anomalías

3.3.10. Script con clase para generar caracterizaciones

La clase *Clasification*, también en **calculations.py**, genera tablas que muestran cómo los valores de una variable se comparan con los percentiles y las normales climáticas. Esto puede ser útil para entender la variabilidad de una variable meteorológica y para identificar valores extremos. Dentro de la clase se definieron 4 métodos:

- *get_percentile*
- *prep_dataframe*
- *style_tmin*
- *style_tmax*
- *style_pp*

El método *get_percentile* primero lee los percentiles y las normales climáticas de los archivos de Excel. Luego, crea un nuevo *dataframe* que sólo incluye las estaciones que tienen percentiles y normales climáticas para el mes actual.

El método *prep_dataframe* añade una columna al *dataframe* con el ID de la estación, y luego reinicia el índice, preparándolo para luego añadir estilos según la variable.

Los métodos *style_tmin*, *style_tmax* y *style_pp* utilizan la biblioteca *dash_table* para crear un DataTable con el *dataframe*. También, añaden estilos condicionales al DataTable para resaltar los valores que están dentro o fuera de los percentiles y las normales climáticas.

src > calculations.py > Clasification > get_percentile

```
607  class Clasification:
608      """
609      Esta clase genera las tablas con la caracterización de las variables de interés
610      en base a sus percentiles y normales climáticas.
611
612      Parámetros:
613      var: 'tmax', 'tmin' o 'pp'.
614      df: dataframe (mx, mn o pp).
615      this_month: mes anterior si es el día 1 del mes, mes actual para los demás días.
616      this_month_days_list: Lista con los días (tipo string) del mes anterior si es el día 1,
617                            o del mes actual si es cualquier otro día.
618      """
619
620      def __init__(self, var, df, this_month, this_month_days_list, umb_path):
621          self.var = var
622          self.df = df
623          self.this_month = this_month
624          self.this_month_days_list = this_month_days_list
625          self.umb_path = umb_path
626          self.get_percentile()
627          self.prep_dataframe()
628
629      def get_percentile(self):
630          """
```

Figura 9. Vista previa del script con clase para caracterización

3.3.11. Script con clase para calcular anomalías de precipitación por decadiarias y crear mapas de distribución

La clase *Decadiarias* calcula las anomalías de precipitación por decadiarias y genera los mapas. Los parámetros de la clase son:

- *pp: dataframe 'pp'.*
- *this_month: mes anterior si es el día 1 del mes, mes actual para los demás días.*
- *file: ruta de archivo 'lista' con todas las estaciones.*
- *dec_file: ruta de archivo con decadiarias.*

Asimismo, contiene los siguientes métodos: *df_prep*, que prepara el dataframe “pp” que contiene los datos de los últimos 45 días; *calculate_anom*, que calcula las anomalías por decadiaria y asigna colores de acuerdo con su valor; y *maps*, que genera los mapas de anomalías por decadiarias usando Mapbox (plataforma de código abierto para crear mapas y visualizaciones de datos geoespaciales).

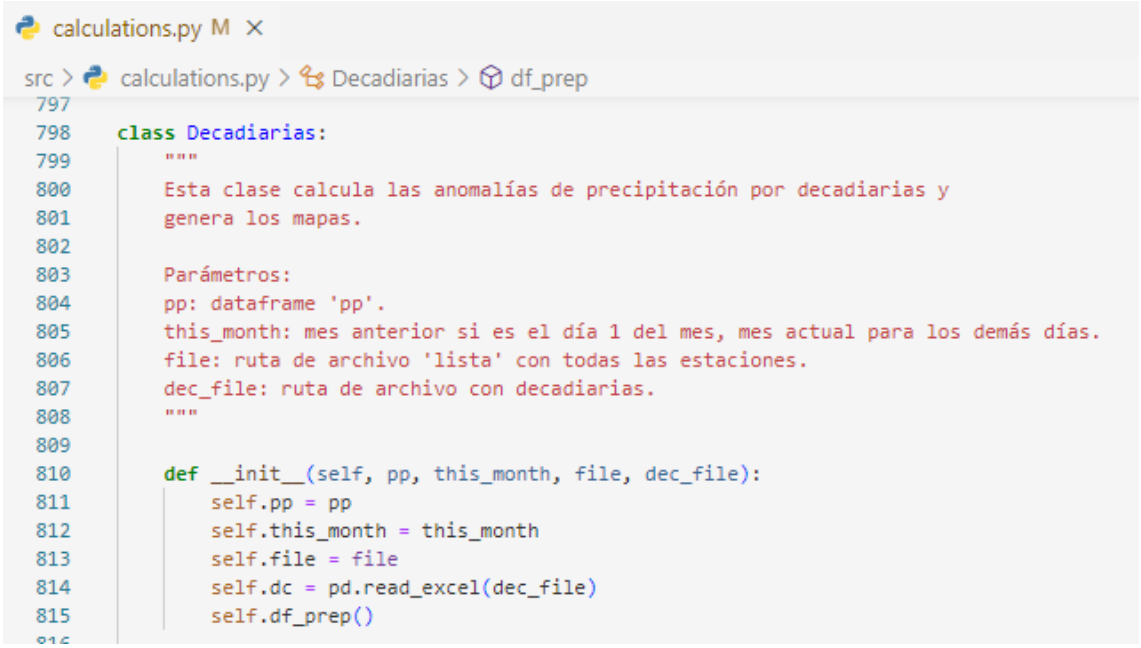


Figura 10. Vista previa del script con clase para calcular anomalías por decadiarias

3.3.12. Script para exportar reporte diario y mensual

Ambos scripts**, daily_report.py** y **monthly_report.py**, tienen una estructura similar. Primero abren el archivo YAML de configuración y lo carga en una variable llamada *config*. Luego, crea una instancia de la clase Dashboard y le pasa el parámetro *graphs=False* para indicar que no se deben generar los gráficos. A continuación, el código obtiene la fecha actual y crea un directorio para el mes actual. Si el directorio ya existe, el código no hace nada. Luego genera el reporte en formato Excel, recorriendo las filas y columnas del *dataframe* y escribiendo los datos en la hoja de trabajo y agregando un borde a las celdas para que el reporte sea más legible. Finalmente, el código cierra el libro de trabajo de Excel.

3.3. Preparando archivos finales para el despliegue del Dashboard

Previo al despliegue, se creó un archivo general llamado **dashboard.py** en la carpeta raíz**,** donde se crearon objetos con las funciones y clases en el archivo **calculations.py.**

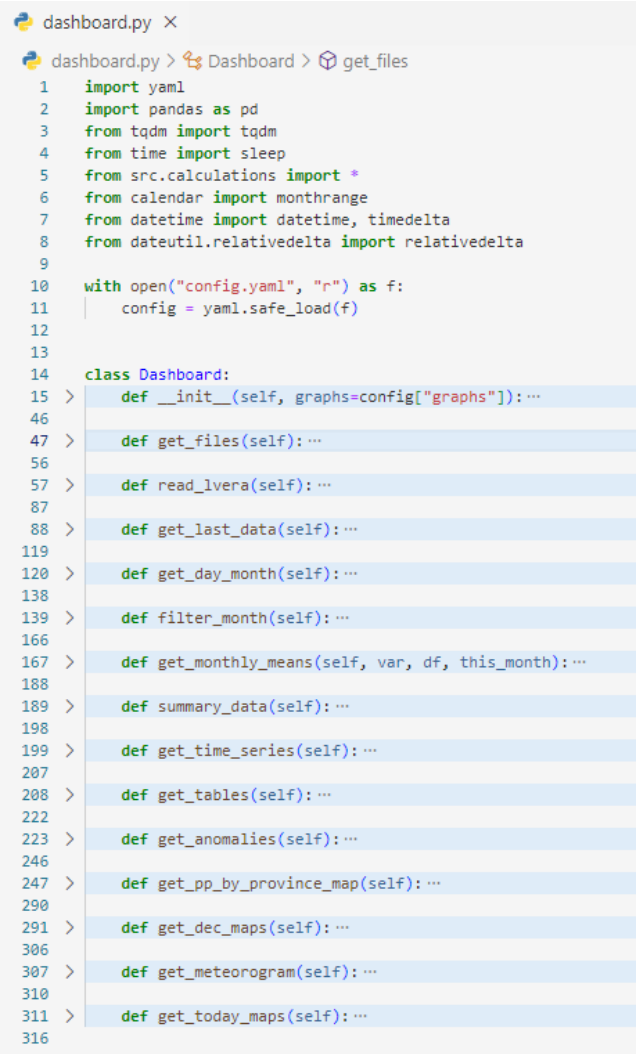


Figura 11. Vista previa del script dashboard.py

Luego, se creó un archivo **layout.py**, donde se hizo la estructura del dashboard en formato HTML, pero usando librerías de Python como *dash_html_components*. Para más detalle sobre las plantillas, ver **Anexo 2**, **Anexo 3** y **Anexo 4**.

Finalmente, se creó el script **app.py**, también en la carpeta raíz, en la cual se indicarán los parámetros a esperar en el comando para ejecutar de acuerdo con lo que se desea, solo dashboard o dashboard con reportes (*daily* y/o *monthly*), para desplegar el proyecto.

3.4. Despliegue de dashboard con información real

A continuación, se describen los comandos disponibles a ejecutar desde la carpeta raíz del proyecto para el despliegue del dashboard:

Generar solo dashboard:

```
python app.py
```

Generar dashboard y reporte diario:

```
python app.py --daily
```

Generar dashboard y reporte mensual:

```
python app.py --monthly
```

Generar dashboard, reporte diario y mensual:

```
python app.py --daily --monthly
```

En caso escoger generar alguno de los reportes, estos son guardados en la carpeta **Data/Exported/** y en las subcarpetas **AÑO/MES,** dependiendo del mes al que corresponden los datos.

Para fines prácticos, se optó por el comando para generar el dashboard, sin los reportes, desde el terminal de Anaconda, instalado previamente, no sin antes haber activado el entorno virtual **dz3**. Esta nos indicará la dirección en la que se verá en proyecto, en este caso es en el puerto 8050. Ver **Figura 12**.

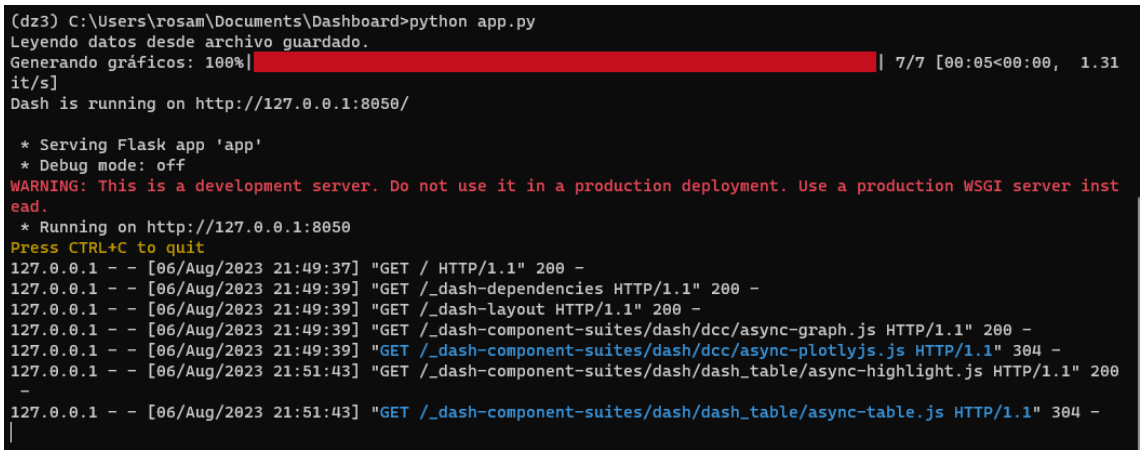


Figura 12. Vista del despliegue del Dashboard desde el terminal de Anaconda

Para visualizar las pestañas del dashboard, ver Anexos 5 al 8.

IV. RESULTADOS Y DISCUSIÓN

Se logró implementar un algoritmo con el lenguaje de programación Python, para la generación automática de los reportes diario y mensual con datos diarios y mensuales de las temperaturas máximas, mínimas y precipitación, para las estaciones en la jurisdicción de la Dirección Zonal 3 – SENAMHI, Cajamarca sur, además de un manual del usuario dentro del proyecto, cuyo contenido se encuentra en un archivo README.md, con las instrucciones para ejecutar el visualizador.

La funcionalidad de cada una de las clases y funciones se probó a través de Jupyter Notebooks, herramienta que permitió ejecutar líneas de código de manera interactiva, logrando ejecutar exitosamente el programa completo, lo que permitió la optimización del tiempo de desarrollo de las tareas rutinarias relacionadas al cargo del Analista lo que conllevó a tener al alcance una herramienta de fácil entendimiento para el usuario, en lugar de presentar datos numéricos únicamente.

Cabe resaltar que, la ejecución del visualizador se hizo y deberá continuar haciéndose de forma manual, aunque puede automatizarse dicho proceso, es necesario que el Analista corrobore los datos luego de revisarlos, en caso sean erróneos inicialmente. Este proceso depende de diversos factores relacionados a la comunicación con los observadores meteorológicos como la señal del móvil.

V. CONCLUSIONES

- Se logró el desarrollo de una herramienta visual para representarla información meteorológica de manera resumida y práctica, usando el lenguaje de software libre Python, de las estaciones meteorológicas de la jurisdicción de la Dirección Zonal 3 – Cajamarca sur, del SENAMHI.
- Se optimizó el tiempo de trabajo ocupado en generar los resúmenes (diario y mensual) de las temperaturas extremas y precipitación, al generar los mismos de manera sistematizada.
- Se sistematizó la exportación de hojas de cálculo de Excel con los datos meteorológicos, cálculos y caracterizaciones, que son base para otras tareas principales como la elaboración de reportes y boletines.

VI. RECOMENDACIONES

Dado que las versiones de Python y sus librerías son actualizadas continuamente, y como pueden aparecer nuevas variables o estaciones meteorológicas que agregar al visualizador, es necesaria la actualización del código periódicamente, así como su correcta documentación.

Además, se recomienda la elaboración de *tests* con Unittest o Pytest, para para probar cada una de las clases y funciones elaboradas, a fin de verificar su funcionamiento o refactorización.

Adicionalmente, se recomienda explorar otras librerías que podrían ayudar a mejorar la presentación y performance del dashboard, como Flask.

Finalmente, la herramienta puede replicarse para otras Direcciones Zonales, o incluso a nivel nacional, sin embargo, ello requiere una arquitectura más detallada del software, por lo que también se recomienda elaborar protocolos o estándares para los archivos que se ingesta al programa, como el formato de los percentiles y normales climáticas.

VII. REFERENCIAS BIBLIOGRÁFICAS

Creating Excel files with Python and XlsxWriter. (s/f). Readthedocs.Io. Recuperado el 11 de julio de 2023, de https://xlsxwriter.readthedocs.io/index.html

OMM (2007). *Función de las normales climatológicas en un clima cambiante*. https://library.wmo.int/doc_num.php?explnum_id=4549

NumPy: the absolute basics for beginners — NumPy v1.25 Manual. (s/f). Numpy.org. Recuperado el 11 de julio de 2023, de https://numpy.org/doc/stable/user/absolute_beginners.html

Pandas documentation — pandas 2.0.3 documentation. (s/f). Pydata.org. Recuperado el 11 de julio de 2023, de https://pandas.pydata.org/docs

PCE Instruments. (2023, mayo, 3). ¿Qué es una estación meteorológica? https://www.pce-iberica.es/medidor-detalles-tecnicos/que-estacion-meteorologica.html

Python OOPs concepts. (2016, mayo 16). GeeksforGeeks. Recuperado de https://www.geeksforgeeks.org/python-oops-concepts/

A. David. (2020, julio 6). *Object-oriented programming (OOP) in Python 3*. Realpython.com; Real Python. https://realpython.com/python3-object-oriented-programming/

C. Dylan. (2023, febrero 20). *Develop data visualization interfaces in Python with Dash*. Realpython.com; Real Python. https://realpython.com/python-dash/

SENAMHI. (2014). Estimación de umbrales de precipitaciones extremas para la emisión de avisos meteorológicos. https://www.senamhi.gob.pe/load/file/01402SENA-6.pdf

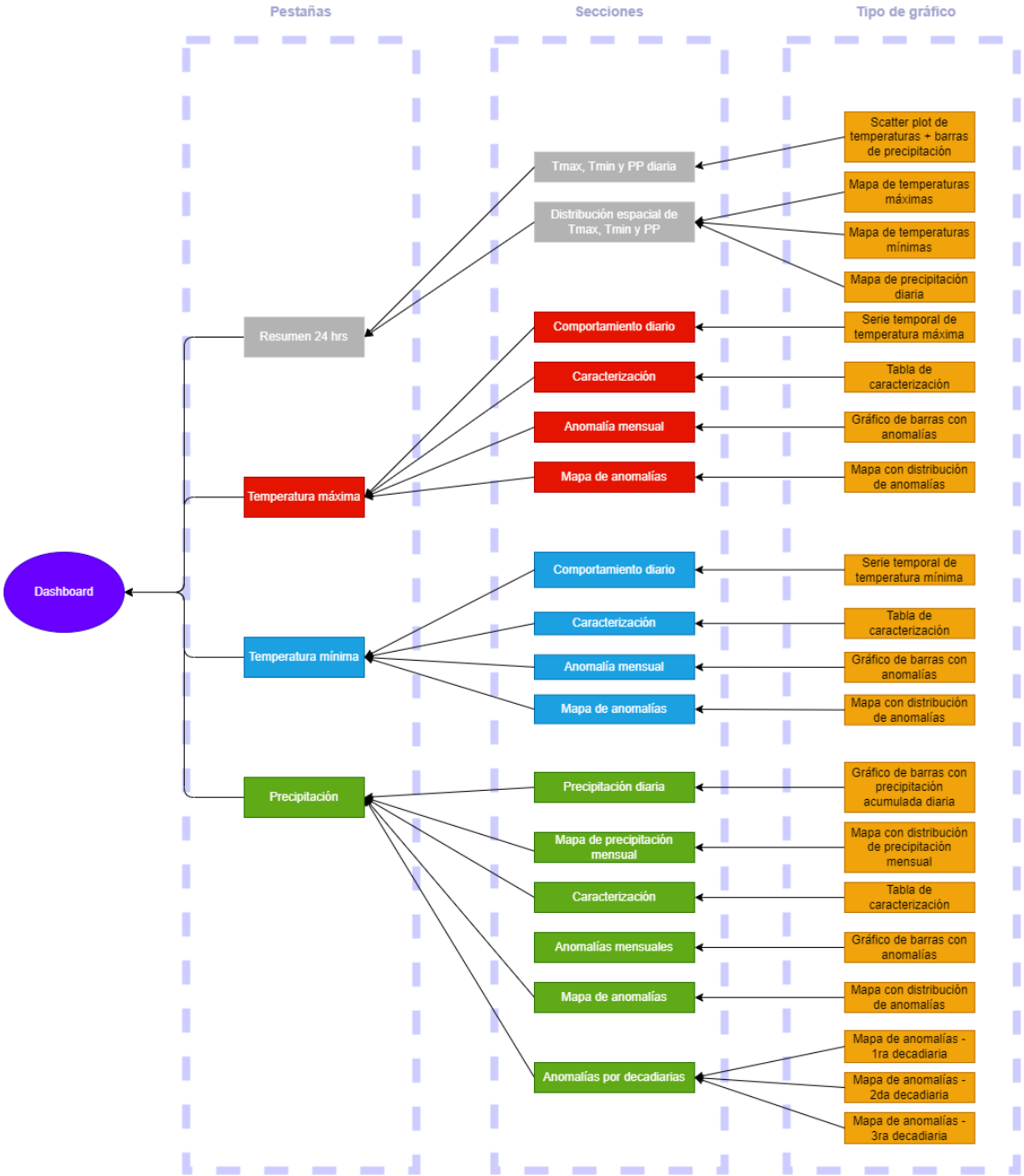
Varios. (s/f). *Nociones de estadística aplicada a la climatología*. Recuperado el 10 de julio de 2023, de https://www.meted.ucar.edu/afwa/climo/stats_es/print.php#page_contributors

What is python? Executive summary. (s/f). Python.org. Recuperado el 11 de julio de 2023. https://www.python.org/doc/essays/blurb/

(S/f). https://www.inec.gob.pa/archivos/P5121generalidades.pdf

VIII. ANEXOS

Anexo 1. Diagrama de flujo para la creación del dashboard

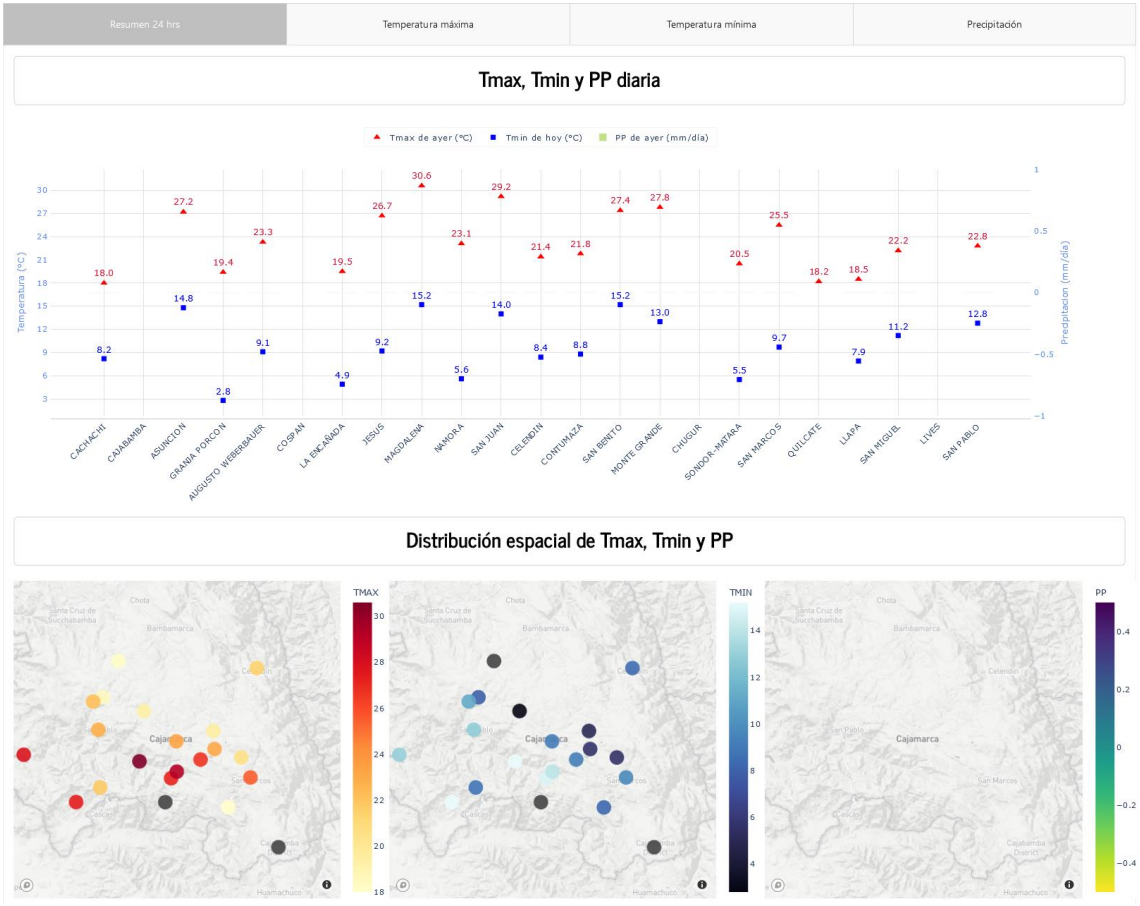


Anexo 2. Plantilla de pestaña de Resumen 24 hrs

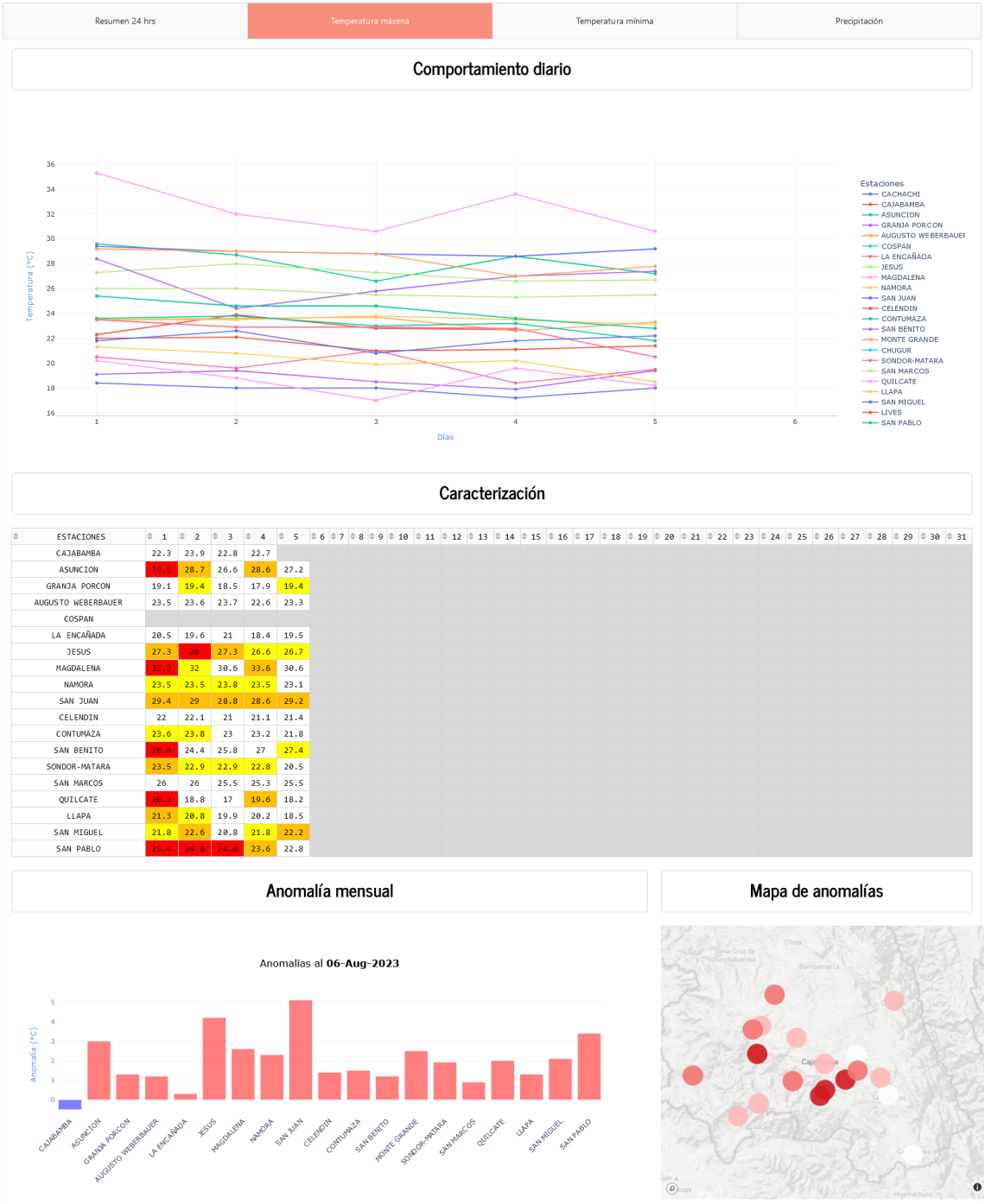
Anexo 3. Plantilla de las pestañas de temperaturas

Anexo 4. Plantilla de la pestaña de Precipitación

Anexo 5. Pestaña de Resumen 24 hrs

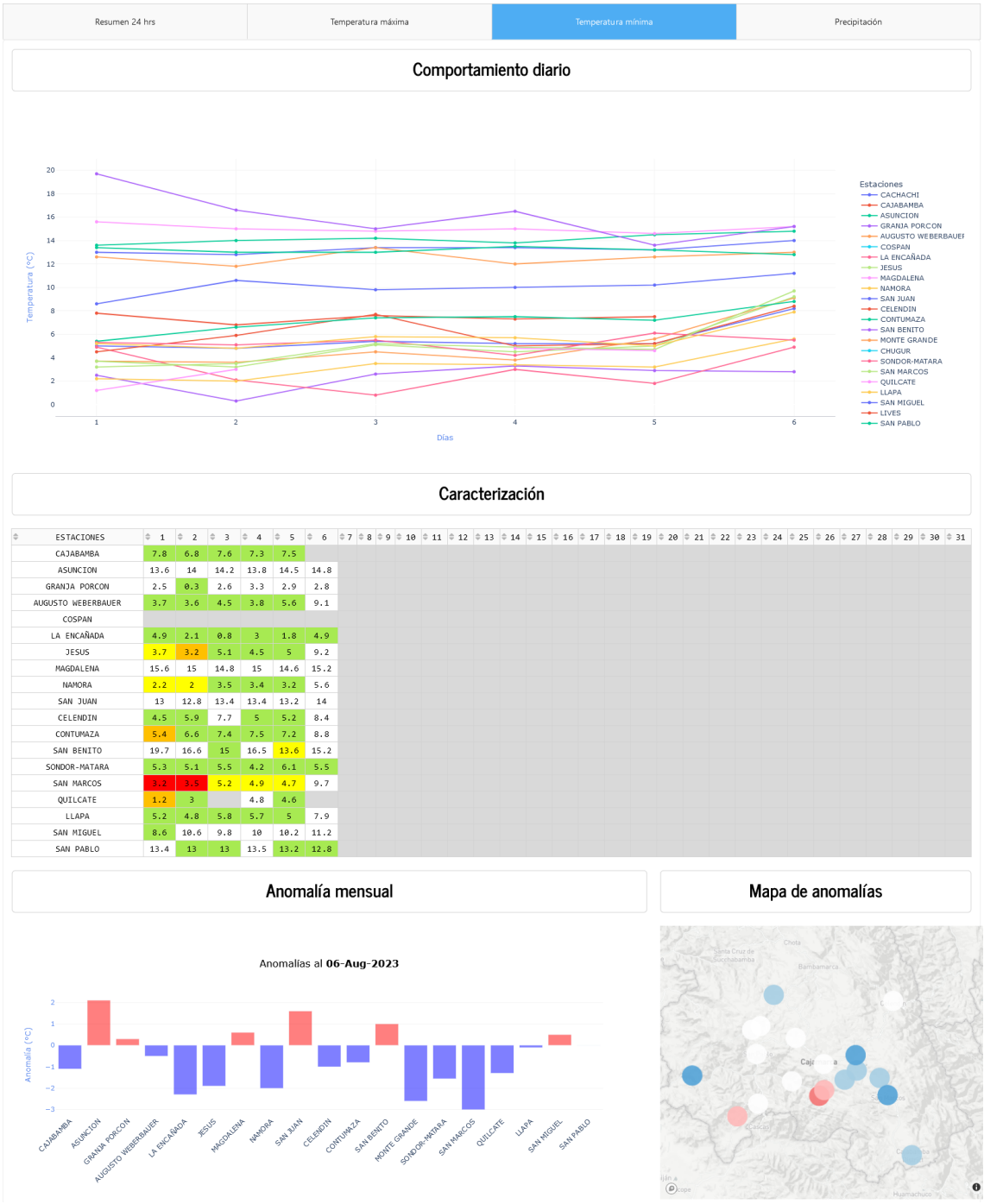


Anexo 6. Pestaña de Temperatura máxima



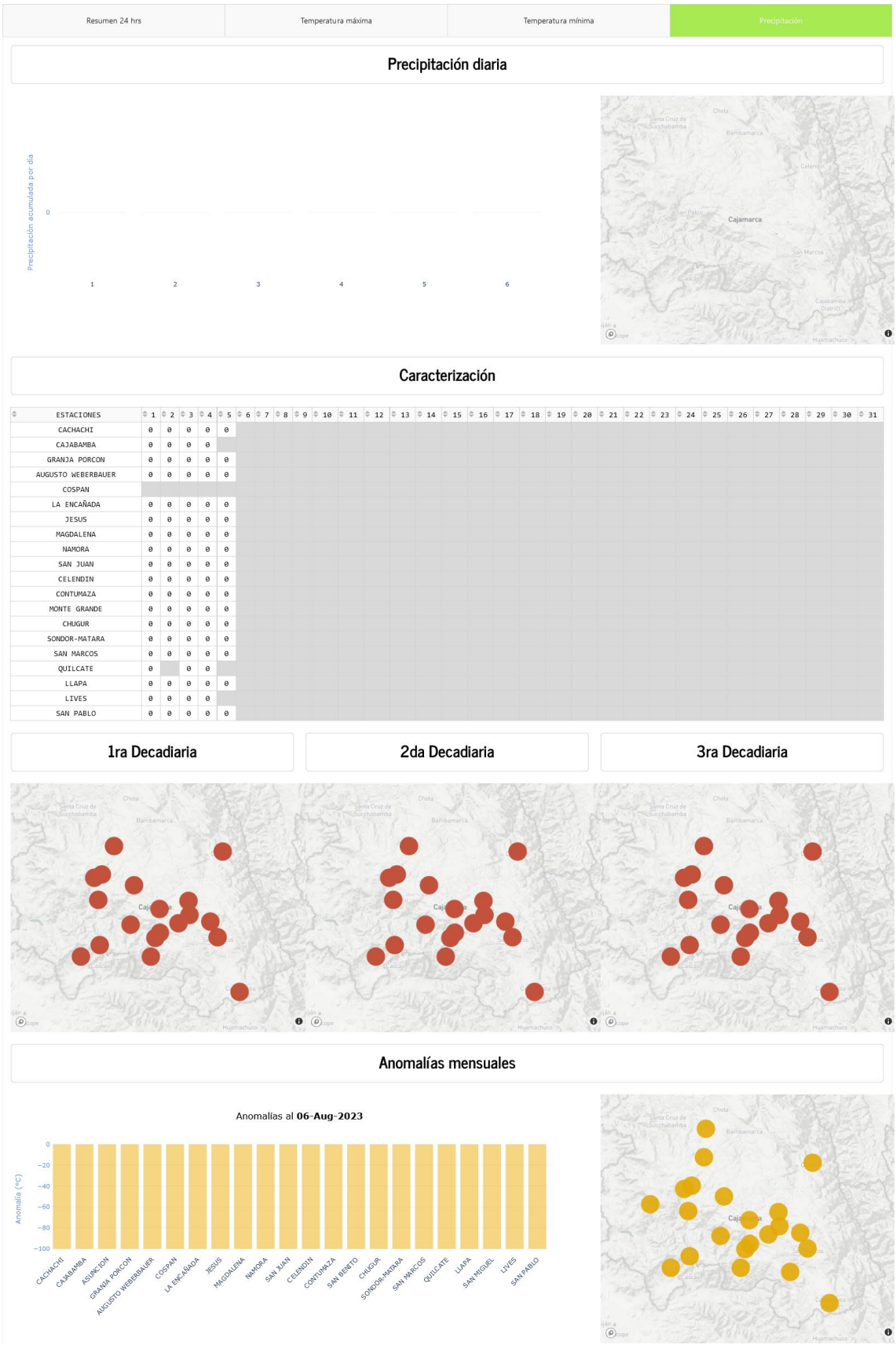
ESTACIONES	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
CAJABAMBA	22.3	23.9	22.8	22.7																											
ASUNCION	29.6	28.7	26.6	28.6	27.2																										
GRANJA PORCON	19.1	19.4	18.5	17.9	19.4																										
AUGUSTO WEBERBAUER	23.5	23.6	23.7	22.6	23.3																										
COSPAN																															
LA ENCAÑADA	20.5	19.6	21	18.4	19.5																										
JESUS	27.3	28	27.3	26.6	26.7																										
MAGDALENA	35.3	32	30.6	33.6	30.6																										
NAMORA	23.5	23.5	23.8	23.5	23.1																										
SAN JUAN	29.4	29	28.8	28.6	29.2																										
CELENDIN	22	22.1	21	21.1	21.4																										
CONTUMAZA	23.6	23.8	23	23.2	21.8																										
SAN BENITO	28.4	24.4	25.8	27	27.4																										
SONDOR-MATARA	23.5	22.9	22.9	22.8	20.5																										
SAN MARCOS	26	26	25.5	25.3	25.5																										
QUILCATE	20.2	18.8	17	19.6	18.2																										
LLAPA	21.3	20.8	19.9	20.2	18.5																										
SAN MIGUEL	21.8	22.6	20.8	21.8	22.2																										
SAN PABLO	25.4	24.6	24.6	23.6	22.8																										

Anexo 7. Pestaña de Temperatura mínima



ESTACIONES	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
CAJABAMBA	7.8	6.8	7.6	7.3	7.5																										
ASUNCION	13.6	14	14.2	13.8	14.5	14.8																									
GRANJA PORCON	2.5	0.3	2.6	3.3	2.9	2.8																									
AUGUSTO WEBERBAUER	3.7	3.6	4.5	3.8	5.6	9.1																									
COSPAN																															
LA ENCAÑADA	4.9	2.1	0.8	3	1.8	4.9																									
JESUS	3.7	3.2	5.1	4.5	5	9.2																									
MAGDALENA	15.6	15	14.8	15	14.6	15.2																									
NAMORA	2.2	2	3.5	3.4	3.2	5.6																									
SAN JUAN	13	12.8	13.4	13.4	13.2	14																									
CELENDIN	4.5	5.9	7.7	5	5.2	8.4																									
CONTUMAZA	5.4	6.6	7.4	7.5	7.2	8.8																									
SAN BENITO	19.7	16.6	15	16.5	13.6	15.2																									
SONDOR-MATARA	5.3	5.1	5.5	4.2	6.1	5.5																									
SAN MARCOS	3.2	3.5	5.2	4.9	4.7	9.7																									
QUILCATE	1.2	3		4.8	4.6																										
LLAPA	5.2	4.8	5.8	5.7	5	7.9																									
SAN MIGUEL	8.6	10.6	9.8	10	10.2	11.2																									
SAN PABLO	13.4	13	13	13.5	13.2	12.8																									

Anexo 8. Pestaña de Precipitación



ESTACIONES	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
CACHACHI	0	0	0	0	0																										
CAJABAMBA	0	0	0	0																											
GRANJA PORCON	0	0	0	0	0																										
AUGUSTO WEBERBAUER	0	0	0	0	0																										
COSPAN																															
LA ENCAÑADA	0	0	0	0	0																										
JESUS	0	0	0	0	0																										
MAGDALENA	0	0	0	0	0																										
NAMORA	0	0	0	0	0																										
SAN JUAN	0	0	0	0	0																										
CELENDIN	0	0	0	0	0																										
CONTUMAZA	0	0	0	0	0																										
MONTE GRANDE	0	0	0	0	0																										
CHUGUR	0	0	0	0	0																										
SONDOR-MATARA	0	0	0	0	0																										
SAN MARCOS	0	0	0	0	0																										
QUILCATE	0		0	0																											
LLAPA	0	0	0	0	0																										
LIVES	0	0	0	0																											
SAN PABLO	0	0	0	0	0																										