

1. Introducción

Este documento presenta el **Manual de Usuario** del prototipo de la aplicación desarrollada para la detección automática de valores atípicos en los datos operativos provenientes de la **distribuidora de gas Contugas S.A.C en Perú**. El objetivo principal del prototipo es **apoyar la supervisión técnica y operativa** mediante la aplicación de técnicas de machine learning que permiten identificar comportamientos anómalos en variables como **volumen de gas, presión y temperatura**. También se profundizará sobre el funcionamiento del tablero de control, el cual permite visualizar los datos más recientes para los clientes, como también un análisis más detallado para cada una de las variables correspondientes a cada cliente.

El presente manual proporciona una guía concisa sobre el funcionamiento del artefacto, los pasos necesarios para su instalación y uso, así como los casos de uso previstos. Está dirigido a usuarios técnicos, principalmente con conocimientos básicos de análisis de datos y programación en Python, interesados en incorporar herramientas de inteligencia artificial al monitoreo de redes de distribución de gas.

2. Descripción General del Artefacto

2.1 ¿Qué es y qué hace ConGas-IA?

Este es un **prototipo de aplicación de machine learning** con apoyo de herramientas visuales diseñado para apoyar el monitoreo de datos operativos en una **distribuidora de gas en Perú**. Su función principal es:

- Detectar **valores atípicos** (anomalías) en variables clave: **volumen de gas, presión y temperatura**.
- Visualizar los resultados mediante un **dashboard interactivo**, el cual permite adicionalmente:
 - Visualizar datos históricos de las variables para cada cliente.
 - Desplegar un resumen descriptivo por cliente de su comportamiento histórico.
 - Identificar consumos anómalos o variaciones atípicas en las variables operacionales.

2.2 Ventajas Explícitas e Implícitas

Las ventajas que ofrece este prototipo se pueden dividir en implícitas y explícitas.

Ventajas explícitas

- **Detección temprana** de posibles fallos o comportamientos anómalos en la red de distribución.
- **Automatización** del análisis de grandes volúmenes de datos históricos y generación de modelos basados en algoritmos de análisis de anomalías.
- **Visualización clara** que facilita la toma de decisiones por parte del personal técnico.
- Permite la **exportación de los datos** en formato CSV para análisis externos adicionales.
- El tablero de visualización de anomalías puede **ser accedido fácilmente** desde cualquier navegador de internet, ya que está construido en herramientas open source.

Ventajas implícitas

Estas ventajas son en gran medida beneficios de los que pueden sacar provecho las áreas operativas de Contugas.

- **Mejora en la eficiencia operativa:** La capacidad de detectar anomalías en tiempo real permitirá a Contugas actuar de manera proactiva, realizando mantenimientos preventivos y evitando fallas mayores en la red de distribución. Esto reducirá el tiempo de inactividad y los costos operativos asociados a reparaciones inesperadas.
- **Reducción y mitigación de riesgos operacionales:** La detección temprana de irregularidades operativas podrá ayudar a Contugas crear nuevas políticas operacionales basadas en la predicción del modelo para prevenir incidentes como fugas o fallos estructurales, mejorando la seguridad tanto para los empleados como para los clientes y toda la red en general.
- Favorece la **toma de decisiones basada en datos**, en lugar de intuiciones.

2.3 Limitaciones Actuales

- El prototipo **no reemplaza validaciones operativas humanas**, con lo cual el alcance de este prototipo es solo facilitar la identificación y visualización de comportamientos anómalos de las medidas operativas.
- Su desempeño depende de la **calidad y cantidad de los datos históricos**, como también de la calidad de estos al tomar las medidas y la recolección de estos.
- No incorpora en esta versión alertas en tiempo real ni integración directa con sistemas externos.
- Los análisis y modelos construidos están actualmente elaborados con datos históricos, con lo cual hasta el momento solo funcionarán con datos estáticos, y para aprovechar su funcionamiento a tiempo real requerirá modificaciones y adiciones al prototipo actual.

2.4 Advertencias de Uso

- Se recomienda validar los valores detectados como atípicos antes de tomar decisiones operativas. Para esto se debe apoyar de las personas expertas en mediciones y también del área operativa que corrobore los valores detectados. Como atípicos, corresponden a un evento en particular.
- La versión actual es **experimental** y no está optimizada para grandes volúmenes de datos en tiempo real.

3. Requisitos técnicos

Para la ejecución de los procesos de análisis, generación de modelos y etiquetado de datos atípicos se requiere:

- Sistema operativo: Windows 10/11, Linux o MAC OS.
- Python 3.9+ instalado.
- Conocimientos básicos de:
 - Python y manejo de notebooks.
 - Machine learning (nivel básico).
 - Visualización de datos con Streamlit.

Para la ejecución del tablero en Streamlit:

- Repositorio o clon del proyecto en GitHub.
- Cuenta en Streamlit Cloud (versión libre).

4. Instalación y Configuración

1. Clonado del Repositorio desde GitHub

Clonar el repositorio del proyecto:

```
git clone https://github.com/josesu92/proyectoUAndes.git
cd proyectoUAndes/
```

2. Descripción de la estructura de carpetas y archivos principales

Al clonar el repositorio, podrá encontrar los siguientes archivos:

Manual de usuario

ConGas-IA

📁 .devcontainer	Added Dev Container Folder	2 hours ago
📁 Dashboard	Mejorando la visualizacion de la imagen	2 hours ago
📄 ProyectoFinal.ipynb	Se genera la primera version	5 hours ago
📄 ProyectoFinalExploracion.ipynb	Se genera la primera version	5 hours ago
📄 datos_contugas.xlsx	Se genera la primera version	5 hours ago

Dashboard: Esta carpeta contiene los scripts y los recursos necesarios para que el tablero se pueda visualizar con los datos históricos y los valores atípicos, haciendo uso de la librería Streamlit. Dentro de esta carpeta se encuentran los siguientes archivos:

📁 pages	Adicionando una imagen.	16 hours ago
📄 Bienvenido.py	Mejorando la visualización de la imagen	15 hours ago
📄 image_proyectoFinal.png	Adicionando una imagen.	16 hours ago
📄 requirements.txt	Se actualiza la ubicacion de los archivos	17 hours ago
📄 streamlit.toml	Mejorando la visualizacion de la imagen	16 hours ago

Pages: Esta carpeta hace referencia a los archivos python que definen las páginas que muestran las visualizaciones en el tablero.

📄 1_📄_Vista_General.py	Adicionando una imagen.	16 hours ago
📄 2_👤_Perfil_de_Cliente.py	Actualizando los nombres de los archivos	17 hours ago
📄 anomaliasDetectadas.csv	Se actualiza la ubicacion de los archivos	17 hours ago

Vista_General.py: Este archivo contiene el código del tablero que contiene a todos los clientes en cada una de sus variables para la semana más reciente.

Perfil_de_Cliente.py: Este archivo contiene el código del tablero que permite profundizar en los datos de cada cliente.

anomaliasDetectadas.csv: Este archivo CSV contiene los datos con las etiquetas que identifican si un registro muestra anomalías.

Bienvenido.py: Este archivo se encarga de mostrar la página inicial del tablero y adicionalmente es el script principal que permite invocar las demás páginas.

image_proyectoFinal.png: Es la imagen que se encuentra en la página de bienvenida y tiene el logo de la aplicación ConGAS-IA.

Manual de usuario

ConGas-IA

requirements.txt: En este archivo se definen los paquetes que necesita el contenedor que usa Streamlit Cloud para montar el dashboard. Aquí debes actualizar los paquetes de Python nuevos que vayas a agregar para mejorar o modificar las visualizaciones.

streamlit.toml: Este es el archivo que define los parámetros de inicialización del tablero, como el tema (en este caso, por defecto es 'light' o claro).

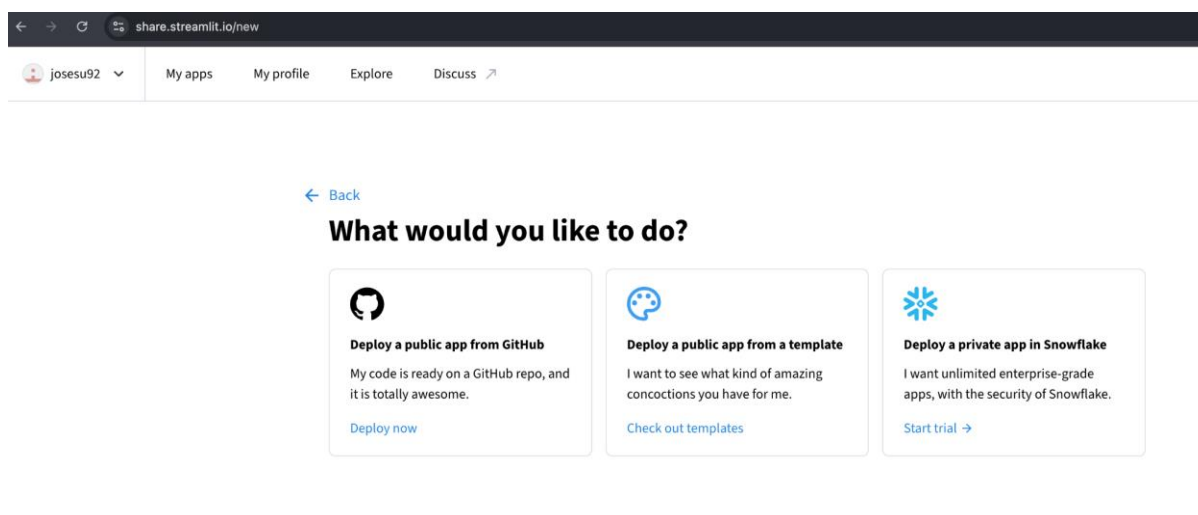
ProyectoFinal.ipynb: Este archivo es un notebook en JupyterNotebooks para generar los modelos; genera la identificación de los valores atípicos.

ProyectoFinalExploracion.ipynb: Este archivo es un notebook en jupyterNotebooks que permite realizar la exploración de los datos de entrada y permite dar un primer análisis a los datos que se encuentran allí.

datos_contugas.xlsx: Son los datos históricos que tienen las métricas de las tres variables (presión, volumen y temperatura).

3. Instalación de Dashboard

1. Se debe tener acceso a Streamlit Cloud, creando una cuenta que puede ser totalmente gratis (en este caso el tablero quedará público).
2. Se debe tener el repositorio alojado en GitHub para que Streamlit pueda acceder.
3. En Streamlit se crea una app configurándose con un repositorio de GitHub.



4. Al seleccionar la opción de 'Desplegar una app pública desde Github', solicitará el acceso y también definir dónde se encuentra el archivo fuente o base del reporte de Streamlit. A continuación, puede ver un ejemplo:

[← Back](#)

Deploy an app

Repository ⓘ

[Paste GitHub URL](#)

josesu92/repo

Branch

master

Main file path

streamlit_app.py

App URL (optional)

.streamlit.app

[Advanced settings](#)[Deploy](#)

5. Clic en desplegar. Esto creará la app con la cual puede ejecutar el reporte que se conecta al repositorio de Github.

5. Tipos de Usuario y Funcionalidades

El prototipo ConGas-IA está diseñado para ser utilizado tanto por usuarios operativos como técnicos, con funcionalidades adaptadas a sus necesidades y niveles de conocimiento:

Usuario técnico (científico de datos, analista de datos):

- Entrenar y calibrar modelos desde los notebooks de Python incluidos en el repositorio ([ProyectoFinal.ipynb](#), [ProyectoFinalExploracion.ipynb](#)).
- Subir nuevos datos históricos, realizar limpieza, imputación y actualización de etiquetas de anomalía.
- Actualizar el archivo [anomaliasDetectadas.csv](#) para reflejar nuevas predicciones en el dashboard.
- Modificar o extender el código del dashboard para incluir nuevas visualizaciones o ajustes de parámetros.

Usuario operativo (personal de mantenimiento, supervisores, jefes de planta):

- Acceder al dashboard a través del navegador web.
- Visualizar el comportamiento histórico de presión, temperatura y volumen para cada cliente.
- Identificar visualmente las anomalías detectadas en cada variable.
- Usar filtros por fecha y cliente para focalizar su análisis en zonas críticas o periodos específicos.

6. Casos de uso y flujo de trabajo

6.1 Casos de uso soportados

1. La app permite, desde diferentes fases, analizar los datos históricos de las variables de cada cliente (volumen, presión y temperatura).
2. Generar modelos con base en series temporales de manera automática para las tres variables de los clientes.
3. Permite la visualización a través de un tablero de seguimiento de las variables y de los clientes, en el cual se podrán visualizar los datos históricos y las anomalías detectadas por los modelos.

6.2 Paso a paso de ejecución

A continuación, se describe el flujo general de ejecución del sistema analítico desde el notebook [ProyectoFinal.ipynb](#), que permite la detección de anomalías por cliente en las variables de presión, temperatura y volumen.

1. Cargar los datos

- Leer el archivo maestro (datos_contugas.xlsx) que contiene las series históricas para cada cliente.
- Consolidar las hojas en un único DataFrame y validar el formato, consistencia y completitud de los datos.

2. Exploración, limpieza y análisis de datos

- Validar la cantidad de registros por cliente y la existencia de valores cero o atípicos.
- Clasificar clientes activos e inactivos según comportamiento del volumen.
- Realizar análisis exploratorio (boxplots, líneas de tiempo) por cliente y variable.

3. Verificación de supuestos

- Aplicar pruebas de estacionalidad (ACF y picos) y estacionariedad (ADF) para cada serie por cliente y variable.
- Clasificar automáticamente las series en estacionarias (para modelar con SARIMA) y estacionales (para modelar con Prophet).

Manual de usuario

ConGas-IA

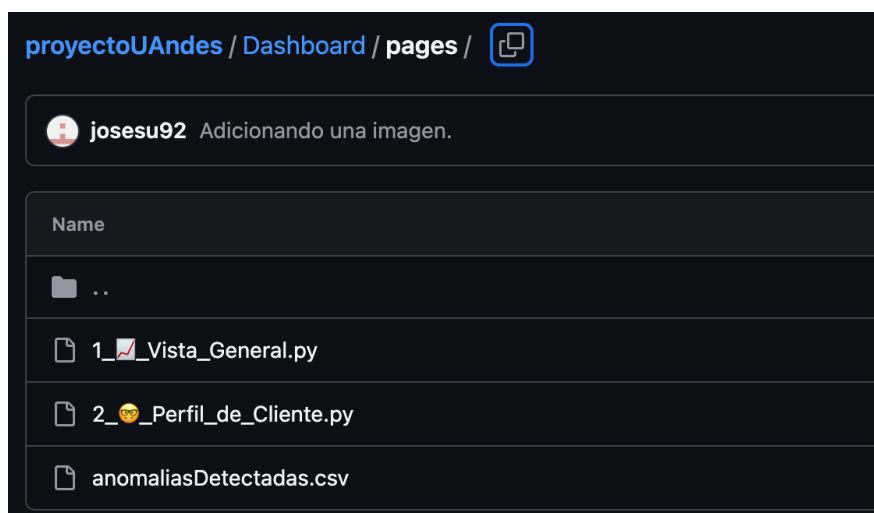
4. Entrenamiento del modelo

- Ejecutar el modelado automáticamente por cliente y por variable:
 - i. SARIMA para series estacionarias.
 - ii. Prophet para series estacionales.
- Calcular residuos y marcar como anomalías los puntos que exceden 2.5 desviaciones estándar.

5. Exportar resultados

- Consolidar todos los resultados en el DataFrame `df_completo` con marcas de anomalías por variable.
- Exportar el archivo final `anomaliasDetectadas.csv` con las detecciones por cliente y fecha.
- Validar los resultados con consultas puntuales por cliente si se desea.

Los resultados después de aplicar el modelo se ponen en la carpeta del repositorio en gitHub en la ruta **Dashboard/pages** con el nombre **anomaliasDetectadas.csv** para que pueda ser consumida por la app de Streamlit. Como se puede ver a continuación:



7. Uso del Dashboard

1. Acceso al dashboard

El dashboard de ConGas-IA es una aplicación web desarrollada con Streamlit. Puede ser accedida desde cualquier navegador moderno (Chrome, Edge, Firefox) sin necesidad de instalación local, siempre que el repositorio esté correctamente desplegado en Streamlit Cloud. El usuario solo debe ingresar al enlace del prototipo para comenzar a interactuar con la visualización.

<https://proyectouandes-kukn8x3ysysm2iybfuwgx2.streamlit.app/>

Manual de usuario

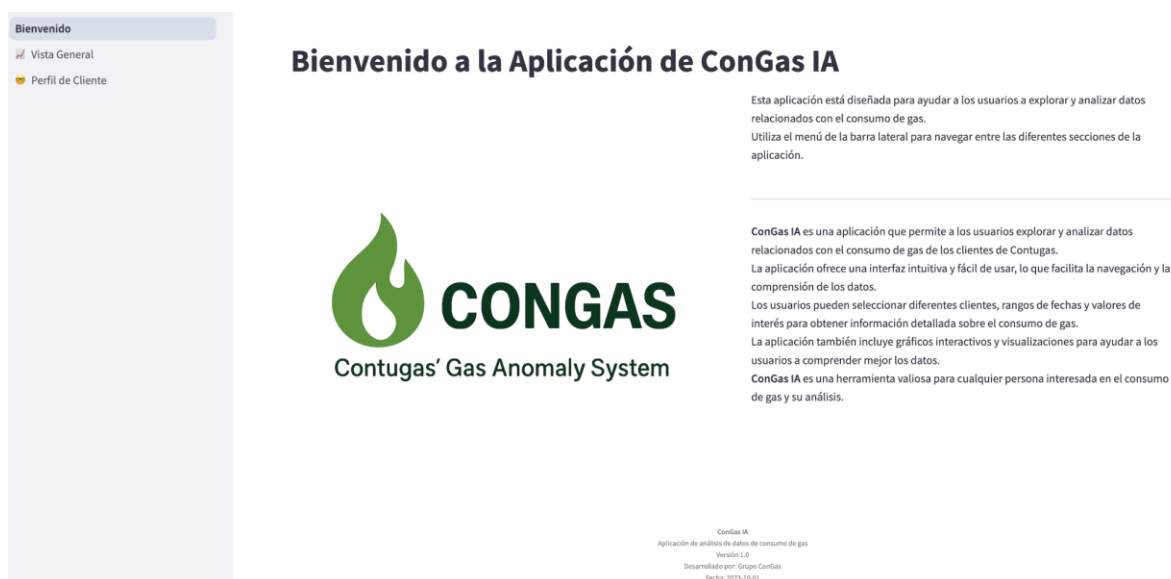
ConGas-IA

2. Estructura y Descripción de las Páginas

El dashboard está organizado en tres secciones principales, accesibles desde un menú lateral:

1. Página de Bienvenida:

- Introduce al usuario al sistema ConGas-IA.
- Presenta el propósito de la herramienta y una visual representativa del prototipo.

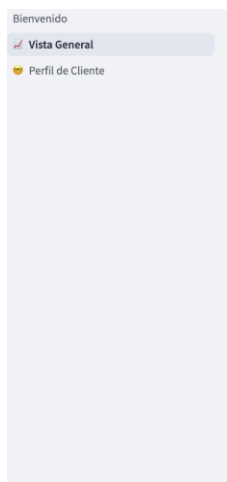


2. Vista General

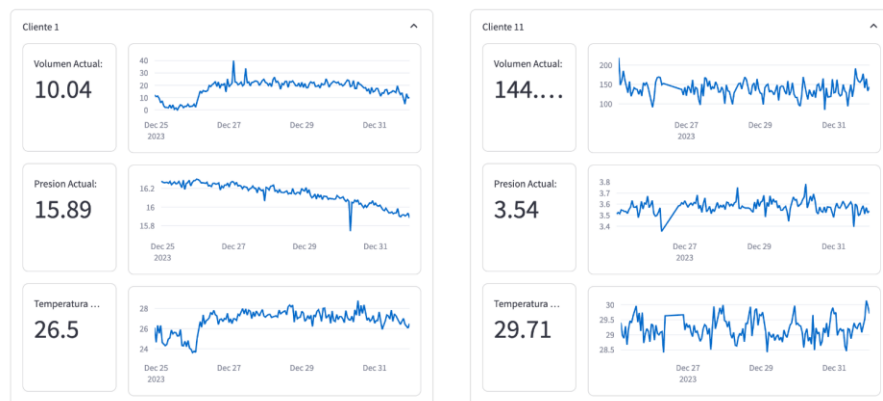
- Muestra el análisis de los datos más recientes (última semana disponible).
- Incluye histogramas, boxplots, matriz de correlación y tabla de exploración de datos con etiquetas de anomalía.
- Útil para diagnóstico rápido del comportamiento general de los clientes y detección de anomalías recientes.

Manual de usuario

ConGas-IA

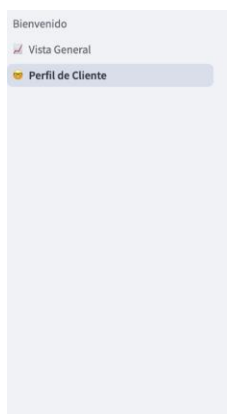


Sistema de Alertas de ConGas IA



3. Perfil de Cliente

- Permite explorar y analizar los datos históricos, como también las anomalías detectadas por los modelos.
- Permite hacer una mayor exploración estadística
- Permite descargar los datos históricos con las etiquetas para cada una de las anomalías.



Perfilado del Cliente

Seleccione el cliente de interés:

CLIENTE 1

Seleccione el rango de fechas de interés:

01.14.2019 - 12.31.2023

Seleccione el rango del volumen de interés:

0.00 - 65.94

Seleccione el rango de la presión de interés:

0.00 - 18.07

Seleccione el rango de la temperatura de interés:

0.00 - 32.87

Resumen del Cliente

ID Cliente: CLIENTE 1

Fecha de inicio: 2019-01-14 00:00:00

Fecha final: 2023-12-31 23:00:00

Total registros: 43412

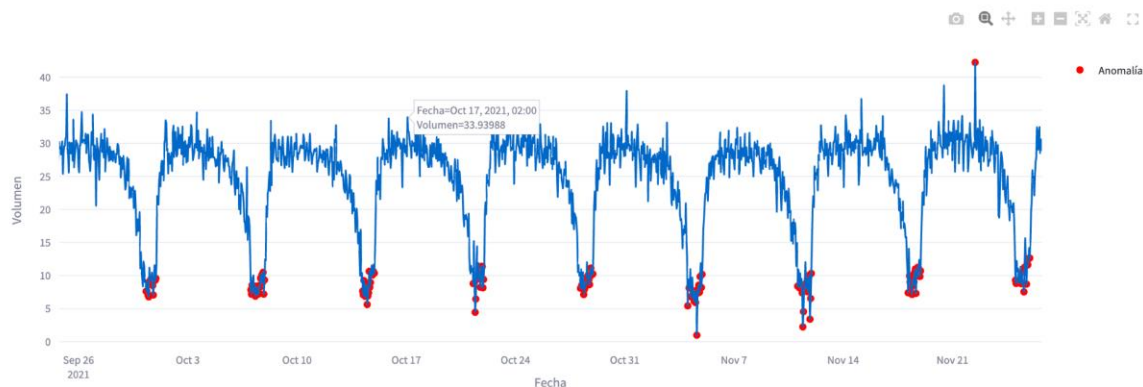
Estadísticas Históricas

	Media	Desviación Estándar	Mínimo	Máximo
Presión	17.5359	0.3583	15.7423	18.0743
Temperatura	25.5759	2.7562	15.4018	32.8691
Volumen	19.9764	7.9392	0	65.9366

Analisis temporal

Volumen Temperatura Presion

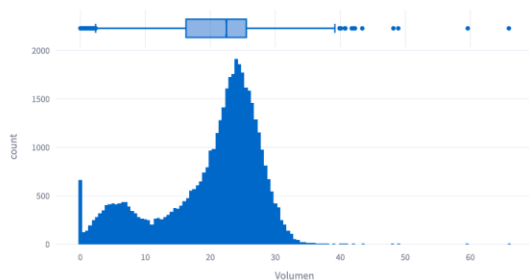
Volumen...



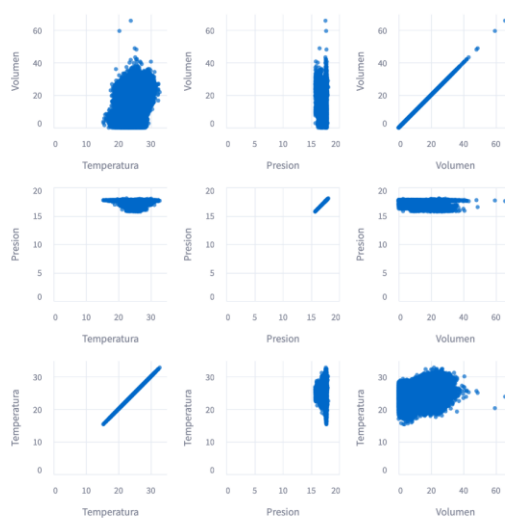
Analisis estadistico

Volumen Temperatura Presion

Volumen...



Correlograma:



Exploracion de datos

	Fecha	Presion	Temperatura	Volumen	AnomaliaPresion	AnomaliaTemperatura	AnomaliaVolumen
0	2019-01-14 00:00:00		17.7326	28.2094	20.9698	<input type="checkbox"/>	<input type="checkbox"/>
1	2019-01-14 01:00:00		17.7478	28.5186	17.8457	<input type="checkbox"/>	<input type="checkbox"/>
2	2019-01-14 02:00:00		17.7589	28.2302	20.9759	<input type="checkbox"/>	<input type="checkbox"/>
3	2019-01-14 03:00:00		17.7279	27.8115	20.5923	<input type="checkbox"/>	<input type="checkbox"/>
4	2019-01-14 04:00:00		17.7465	27.7953	21.6906	<input type="checkbox"/>	<input type="checkbox"/>
5	2019-01-14 05:00:00		17.7355	27.6665	19.0917	<input type="checkbox"/>	<input type="checkbox"/>
6	2019-01-14 06:00:00		17.7318	27.5323	18.9419	<input type="checkbox"/>	<input type="checkbox"/>
7	2019-01-14 07:00:00		17.7153	27.2796	14.9941	<input type="checkbox"/>	<input type="checkbox"/>
8	2019-01-14 08:00:00		17.7029	27.085	19.2999	<input type="checkbox"/>	<input type="checkbox"/>
9	2019-01-14 09:00:00		17.7115	28.8253	19.4087	<input type="checkbox"/>	<input type="checkbox"/>

4. Filtros Disponibles

Los filtros permiten adaptar el análisis a diferentes necesidades:

- Rango de fechas: permite seleccionar el periodo a analizar.

- Cliente: selección individual de clientes por ID.
- Rangos de interés para cada variable: filtros dinámicos para presión, temperatura y volumen.

Esto habilita un análisis más focalizado, por ejemplo: estudiar un cliente con alto consumo en un periodo específico o identificar eventos de presión anómala.

5. Exportación de los datos en CSV

Desde la página de Perfil de Cliente, es posible descargar los datos históricos con las etiquetas de anomalías identificadas. El archivo se exporta en formato CSV, lo que permite:

- Compartir los resultados con otros equipos operativos.
- Realizar análisis adicionales fuera del dashboard (ej. en Excel o herramientas de BI).
- Conservar un registro auditable del comportamiento por cliente.

8. Actualización y Mantenimiento

8.1 Cómo actualizar el archivo de anomalías con datos recientes

Para actualizar los resultados de detección de anomalías en el dashboard, debe reemplazarse el archivo ``anomaliasDetectadas.csv`` ubicado en la carpeta ``Dashboard/pages/`` por uno nuevo, generado con datos recientes. Una vez actualizado el archivo en el repositorio, el dashboard reflejará los nuevos registros automáticamente.

8.2 Cómo re entrenar modelos si hay nuevos datos

Para entrenar nuevamente los modelos con datos nuevos:

1. Cargar los nuevos datos en el archivo ``datos_contugas.xlsx``.
2. Ejecutar el notebook ``ProyectoFinal.ipynb`` para generar los modelos y etiquetar las anomalías.
3. Exportar el nuevo archivo ``anomaliasDetectadas.csv`` y moverlo a la carpeta

`Dashboard/pages/`.

4. Confirmar que los cambios se reflejan correctamente en el dashboard al reiniciar la aplicación.

8.3 Mantenimiento de la aplicación

- Revisar periódicamente la vigencia y calidad de los datos utilizados.
- Actualizar las librerías de Python listadas en `requirements.txt` para asegurar compatibilidad y seguridad.
- Verificar que los modelos sigan cumpliendo con la precisión esperada en nuevos escenarios operativos.
- Planear un reentrenamiento periódico del modelo, por ejemplo, cada 3 o 6 meses, dependiendo del volumen y estacionalidad de los datos.

9. Preguntas Frecuentes

- ¿Qué pasa si el dashboard no carga?

Verifique su conexión a internet. Asegúrese de que el repositorio esté bien desplegado en Streamlit Cloud. Si el error persiste, revise el archivo `streamlit.toml` y que el archivo principal del dashboard esté correctamente referenciado.

- ¿Qué hacer si no aparecen datos?

Revise que el archivo `anomaliasDetectadas.csv` esté presente en la ruta correcta y con los datos completos. Verifique que los filtros no estén ocultando toda la información (por fecha o cliente).

- ¿Cómo actualizar los datos con nuevas mediciones?

Agregue los nuevos datos al archivo `datos_contugas.xlsx`, ejecute el notebook `ProyectoFinal.ipynb`, y reemplace el archivo `anomaliasDetectadas.csv` con los nuevos resultados. Luego reinicie la app.

- ¿Qué hacer si se detectan anomalías que no parecen correctas?

Consulte con el equipo operativo para validar los datos. Las anomalías se detectan estadísticamente y podrían incluir registros atípicos que no representen fallos reales. Ajuste el umbral de sensibilidad o reentrene el modelo si es necesario.

10. Contacto y Soporte

Si presenta dudas, errores durante el uso del prototipo o desea realizar sugerencias de mejora, puede comunicarse con el equipo desarrollador a través de los siguientes canales:

- **Correo electrónico:**
jd.suarezg12@uniandes.edu.co
n.parrado@uniandes.edu.co
- **Repositorio del proyecto en GitHub:**
<https://github.com/josesu92/proyectoUAndes>
Allí encontrará el código fuente, documentación técnica y ejemplos de uso.
- **Versión desplegada del prototipo en Streamlit:**
<https://proyectouandes-kukn8x3ysysm2iybfuwgx2.streamlit.app>