

# Manual de Usuario – Proyecto OPTIGAS

## 1. Clonación del Repositorio

---

Antes de poder utilizar el sistema OPTIGAS, es necesario obtener todos los archivos y carpetas que conforman el proyecto. Estos archivos están almacenados en un repositorio remoto en la plataforma GitHub. GitHub es una herramienta ampliamente utilizada para almacenar y gestionar proyectos de software, permitiendo que cualquier persona con los permisos adecuados pueda acceder al código, descargarlo, modificarlo o ejecutarlo.

El proceso para obtener una copia local del proyecto desde GitHub se denomina 'clonar un repositorio'. Este proceso copia todos los archivos del proyecto a su computador, incluyendo scripts, bases de datos de ejemplo, configuraciones y documentación.

### Pasos detallados:

---

#### Requisitos previos

Antes de comenzar, asegúrese de tener instalado en su equipo el programa Git. Si no lo tiene, puede descargarlo desde: <https://git-scm.com/>

#### Instrucciones

##### 1. Abrir una terminal de comandos:

- En Windows: puede buscar 'Símbolo del sistema' o 'CMD'.
- En macOS o Linux: puede abrir la aplicación llamada 'Terminal'.

##### 2. Ejecutar el comando de clonación:

```
git clone https://github.com/jsmoralesc/optigas-contugas.git
```

Este comando descargará todo el contenido del proyecto en una nueva carpeta llamada 'optigas-contugas'.

##### 3. Entrar a la carpeta del proyecto:

```
cd optigas-contugas
```

Esto le permitirá ubicarse dentro del proyecto, donde podrá ejecutar los siguientes pasos de instalación, configuración y despliegue.

## 2. Instalación del Entorno Virtual

---

Antes de ejecutar cualquier parte del proyecto, es muy importante configurar correctamente el entorno de trabajo en tu computador. Esto garantiza que todas las herramientas, librerías y versiones necesarias estén disponibles y no entren en conflicto con otros proyectos que puedas tener.

### ¿Qué es un entorno virtual?

Un entorno virtual es como una “burbuja” que aísla todo lo que el proyecto necesita para funcionar: versiones específicas de Python y librerías como pandas, numpy, streamlit, entre otras. Gracias a esto, puedes trabajar con total seguridad sin afectar otros programas instalados en tu computador.

### ¿Qué herramienta usaremos?

Para este proyecto se utiliza Conda, una herramienta muy común en ciencia de datos que facilita la creación y gestión de entornos virtuales. Conda forma parte de Anaconda o Miniconda, y si no la tienes instalada, puedes descargarla desde:

<https://docs.conda.io/en/latest/miniconda.html>

### Pasos para crear y activar el entorno

---

#### 1. Abre la terminal o consola

- En Windows: busca “Anaconda Prompt” o abre PowerShell (si tienes Conda configurado).
- En Mac/Linux: abre la Terminal.

#### 2. Dirígete a la carpeta del proyecto

`cd optigas-contugas`

#### 3. Crea el entorno con el archivo environment.yml

Este archivo contiene toda la configuración necesaria para instalar las librerías correctas, con sus respectivas versiones.

Ejecuta el siguiente comando:

`conda env create -f environment.yml`

Esto creará automáticamente un entorno llamado 'optigas' con todas las dependencias necesarias.

#### 4. Activa el entorno

`conda activate optigas`

Notarás que el nombre del entorno ('optigas') aparece entre paréntesis al inicio de la línea en tu terminal, lo que indica que está activo correctamente:

(optigas) \$

### 3. Configuración de la Base de Datos

---

El proyecto OPTIGAS utiliza una base de datos local tipo SQLite, un archivo liviano que actúa como una mini base de datos que no necesita instalar servidores adicionales. Esta base de datos se encarga de almacenar la información histórica de cada cliente, organizada por hora, incluyendo las variables de:

- Presión
- Temperatura
- Volumen

Esta información es esencial, ya que alimenta los gráficos y análisis que verás más adelante en el dashboard.

#### Archivo de base de datos

El archivo de la base de datos se llama por defecto 'optigas.db' y se ubica en la carpeta 'db/' del proyecto. Si al clonar el proyecto esta base de datos no existe aún, puedes generarla automáticamente utilizando un script que ya viene incluido en el repositorio.

## 4. Ejecución de los Códigos

---

Una vez se ha configurado correctamente el entorno de trabajo y la base de datos, es momento de poner en marcha los componentes principales del sistema. Esta etapa incluye dos pasos fundamentales:

1. Preparar los datos para su análisis (preprocesamiento)
2. Entrenar el modelo de detección de anomalías usando técnicas de inteligencia artificial

Estos pasos permiten transformar los datos crudos en información útil para la toma de decisiones y habilitar el análisis automático de eventos anómalos en el consumo de gas por parte de los clientes.

### 4.1. Preprocesamiento de Datos

El preprocesamiento es una fase crítica, ya que garantiza que los datos estén organizados, limpios y listos para el análisis y visualización. Dado que se trata de registros generados automáticamente cada hora, pueden presentarse casos como:

- Datos faltantes: por desconexión de sensores o fallas de comunicación.
- Inconsistencias horarias: por saltos o duplicados en los tiempos.
- Valores fuera de rango: que deben ser corregidos o marcados para análisis posterior.

Este paso incluye las siguientes tareas:

- Reindexación temporal: se asegura de que cada cliente tenga un registro por cada hora dentro del rango histórico (ej. 2019–2023).
- Imputación de valores faltantes: se rellena inteligentemente la información que falta.
- Interpolación de variables: se suavizan las curvas de presión, temperatura y volumen.

Para ejecutar este proceso, basta con correr el siguiente comando en la terminal:

`python scripts/etl_raw_to_gold.py`

Después de ejecutarlo, los datos estarán disponibles en la carpeta `data/gold/` y en la base de datos 'optigas.db', listos para alimentar los modelos de inteligencia artificial y para ser visualizados en el dashboard.

## 4.2. Entrenamiento del Modelo de Anomalías

El siguiente paso es la construcción de un modelo automático que permita detectar comportamientos inusuales en los datos de consumo, como caídas inesperadas de presión, aumentos abruptos de volumen o registros fuera de los rangos operativos.

Para esto, se utilizan técnicas de machine learning (aprendizaje automático). En este proyecto, el modelo principal es DBSCAN, una técnica especializada en identificar anomalías sin necesidad de etiquetas previas.

Este modelo aprende a identificar qué registros se comportan de manera "normal" y cuáles podrían representar un evento atípico, basándose en patrones históricos. Esto permite:

- Detectar posibles fallas en sensores.
- Identificar consumos irregulares.
- Anticipar riesgos operativos o de seguridad.

El modelo se entrena automáticamente a partir de los datos preprocesados ejecutando el siguiente comando:

```
python scripts/anomaly_detection.py
```

Este comando genera un archivo de modelo entrenado, almacenado en la carpeta `modelos/` y una tabla en la base de datos llamada 'gold\_anomalías' con el resultado de la modelación para cada cliente, que luego será utilizado por el dashboard para clasificar nuevos registros en tiempo real o al momento de su visualización.

## 5. Despliegue del Dashboard



Pantalla inicial del dashboard abierto en el navegador, mostrando el título principal y las primeras secciones visibles.

El dashboard es una aplicación visual e interactiva que ha sido diseñada como el componente principal de este proyecto. Su propósito es permitir a los usuarios explorar los datos de presión, temperatura y volumen registrados por cliente, así como identificar de forma clara y rápida posibles anomalías en el comportamiento de estas variables. A través de gráficos, tablas y filtros, se puede navegar por la información de forma intuitiva y obtener insights útiles para la toma de decisiones operativas.

### ¿Cómo se accede al dashboard?

Una vez que se han seguido los pasos anteriores (clonación del repositorio, configuración del entorno y ejecución de scripts de preprocesamiento), es momento de lanzar la aplicación del dashboard.

Para ello, asegúrese de estar ubicado dentro de la carpeta raíz del proyecto y con el entorno virtual activado. Luego, en la terminal (o consola), escriba el siguiente comando:

```
streamlit run app/main.py
```

### ¿Qué sucede después de ejecutar este comando?

1. El sistema iniciará el servidor local de Streamlit, que es la herramienta utilizada para construir el dashboard.
2. Automáticamente se abrirá una nueva ventana o pestaña en su navegador predeterminado (por ejemplo, Google Chrome o Edge).
3. La dirección que verá suele ser <http://localhost:8501> o similar. Este enlace es único para su computador y permite acceder al dashboard de forma local, sin necesidad de conexión a internet.

### ¿Qué se puede hacer en el dashboard?

Desde el momento en que se abre la interfaz, el usuario podrá:

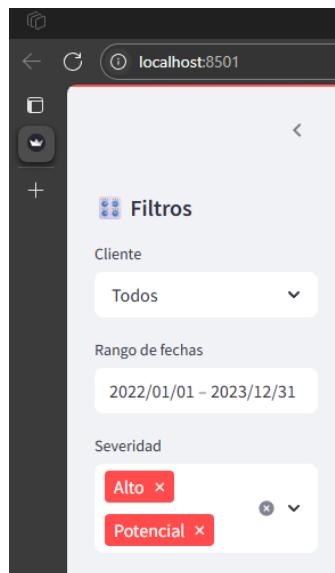
- Seleccionar un cliente para analizar su comportamiento operativo.
- Elegir una variable (presión, temperatura, volumen) para profundizar en su comportamiento histórico.
- Ver gráficamente los datos, identificar desviaciones o picos inusuales y confirmar visualmente la existencia de anomalías.
- Consultar un resumen numérico del comportamiento del cliente en un rango de fechas determinado.
- Descargar datos o tablas para un análisis posterior.

### ¿Por qué es importante esta herramienta?

El dashboard no solo presenta los datos, sino que los contextualiza: combina reglas de negocio, criterios técnicos y algoritmos de aprendizaje automático para marcar comportamientos inusuales. Esto hace que sea mucho más fácil para cualquier persona del equipo —sin necesidad de conocimientos en programación o estadística— entender qué está ocurriendo y cuándo podría necesitarse una revisión o intervención.

## 6. Uso del Dashboard – Guía Interactiva

### 6.1. Filtros Laterales



Panel lateral con filtros de cliente, fecha y variable.

Desde el panel lateral podrá seleccionar el cliente, el rango de fechas y la variable que desea analizar. Esto le permitirá enfocar el análisis a períodos específicos o a variables particulares como volumen, presión o temperatura.

### 6.2. KPIs Principales



Vista de las tarjetas con indicadores clave (promedios y número de anomalías).

El dashboard muestra indicadores clave de desempeño (KPIs) como el volumen promedio, presión media y temperatura promedio, además del número total de anomalías detectadas. Estos ayudan a tener una visión rápida del estado del cliente.

### 6.3. Vista rápida de las anomalías más recientes y top de clientes con mayor anomalías

### Alertas recientes de anomalías (últimas 50)

	timestamp	cliente_id	presión	temperatura	volumen	severidad
175176	2023-12-31 00:00:00	CLIENTE10	17.4718	24.0633	0	Alto
262776	2023-12-31 00:00:00	CLIENTE15	17.4824	28.3148	141.3136	Alto
192696	2023-12-31 00:00:00	CLIENTE11	3.4798	29.6913	128.5313	Alto
52536	2023-12-31 00:00:00	CLIENTE3	3.6602	28.7263	121.259	Alto
227736	2023-12-31 00:00:00	CLIENTE13	17.5548	22.2305	0	Alto
332856	2023-12-31 00:00:00	CLIENTE19	16.2943	27.267	0.0195	Potencial
157656	2023-12-31 00:00:00	CLIENTE9	17.4809	23.53	38.1111	Alto
157655	2023-12-30 23:00:00	CLIENTE9	17.4802	23.6696	4.636	Alto
105095	2023-12-30 23:00:00	CLIENTE6	15.8742	24.9763	67.4409	Potencial
175175	2023-12-30 23:00:00	CLIENTE10	17.4721	23.741	0	Alto

### Visualización de variables operativas y anomalías (top 5 clientes con más alertas en los últimos 5 días)

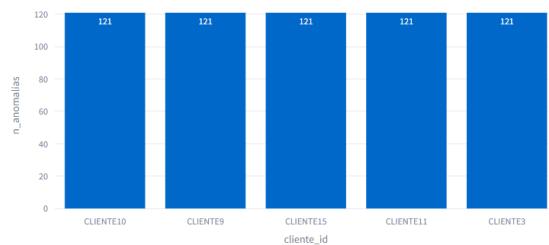


Tabla de anomalías recientes y gráfico de líneas de la variable seleccionada con puntos de anomalías.

Distribución - Volumen (últimos 5 días - Top 5)



Top 5 clientes con más alertas (últimos 5 días)



Gráficos descriptivos de la información de los top clientes con mayores anomalías.

Visualice cómo varían los gráficos a lo largo del tiempo y al seleccionar la variable a visualizar entre Volumen, presión, temperatura.

## 6.4. Detalles Adicionales

Para poder desplegar un análisis detallado de un cliente es necesario en los filtros del dashboard en la parte superior izquierda seleccionar un cliente específico, si no se selecciona un cliente específico por defecto toma el valor de todos y se informa en el dashboard.

Selecciona un cliente para ver su análisis detallado.

Aviso de selección de cliente.

Se puede consultar una tabla con los registros más recientes donde se muestra información detallada de cada lectura, incluyendo fecha, hora, presión, temperatura, volumen y la clasificación de severidad.

## 6.5. Análisis detallado de cliente

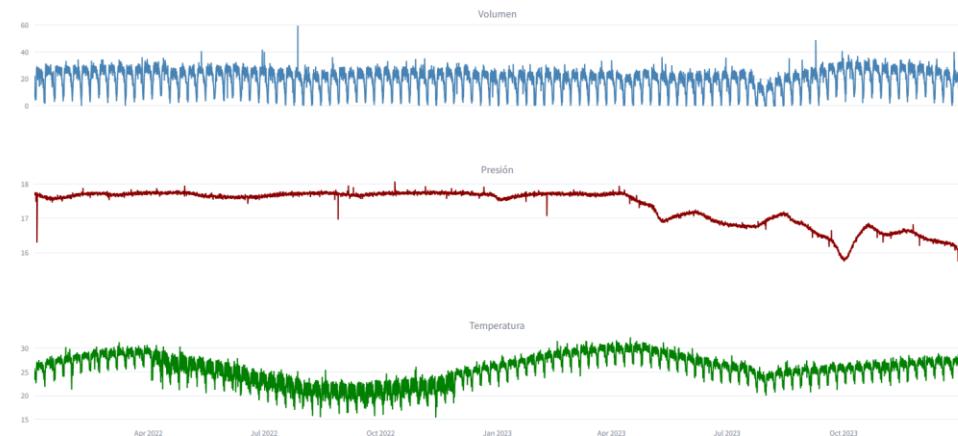
La sección cuenta con dos pestañas informativas correspondientes a ‘Evolucion historica’ y ‘Anomalias’ donde se estructura la información particular del cliente teniendo una vista histórica del comportamiento del cliente y otra vista específica de las anomalías que se identifican en el comportamiento de dicho cliente.

- **‘Evolucion historica’**



#### Indicadores históricos de comportamiento del cliente.

- Evolución de Variables Operativas**



#### Evolución histórica de las variables del cliente.

- Distribución de Variables**



#### Gráfico de distribución de las variables del cliente.

- ‘**Anomalías**’:

Para el manejo de las anomalías se implementó una clasificación dada por la variable severidad la cual clasifica cada observación horaria (lectura de presión, temperatura y volumen por cliente) según el **nivel de anomalía detectada**. Esta categorización se construye combinando dos enfoques:

Categoría	Descripción técnica	Fuente de la anomalía
Alto	Anomalía detectada por <b>reglas de negocio</b> (por ejemplo, valores atípicos según el IQR o ceros inconsistentes).	Reglas estadísticas / dominio (IQR, ceros)
Potencial	Anomalía detectada <b>solo por el modelo de machine learning</b> (DBSCAN).	ML no supervisado (clustering)
OK	No se detectó ninguna anomalía ni por reglas ni por el modelo.	Lectura considerada normal

### ¿Por qué combinar reglas + ML?

- **Reglas de negocio (IQR)** permiten detectar outliers claros y errores físicos (como presión en cero con volumen > 0).
- **Modelos como DBSCAN** detectan patrones de comportamiento anormal que no son evidentes con reglas simples.

Al combinar ambos, se obtiene una clasificación más robusta y útil para:

- Priorización de alertas (Alto primero).
- Análisis operativo y mantenimiento preventivo.
- Visualizaciones históricas en dashboards como OPTIGAS.

#### 🔍 Análisis detallado – CLIENTE1

🕒 Evolución Histórica

⚠ Anomalías

#### ⚠ Métricas de Anomalías

🕒 Rango Anomalías

2022-01-01 → 2023-12-26

⚠ Total Anomalías

11710

🕒 Días con Anomalías

529

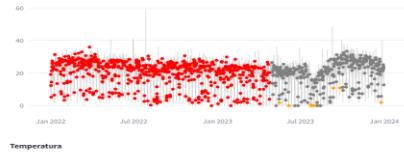
🕒 Prom. Anomalías/Día

22.1

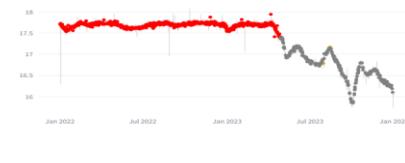
#### Indicadores anomalías del cliente.

##### 🕒 Evolución Temporal

Volumen



Volumen



Temperatura

15

20

25

30

Presión

16

17

18

19

Temperatura

16

17

18

19

#### Evolución de severidad de las anomalías del cliente.

#### Detalles Adicionales

	timestamp	volumen	presión	temperatura	severidad
17378	2023-12-26 02:00:00	2.2614	16.2592	23.6522	Potencial
17377	2023-12-26 01:00:00	4.663	16.2819	23.7648	Potencial
17376	2023-12-26 00:00:00	3.0866	16.2937	23.6185	Potencial
17375	2023-12-25 23:00:00	2.9008	16.2974	23.946	Potencial
17374	2023-12-25 22:00:00	2.6134	16.2827	24.0676	Potencial
17372	2023-12-25 20:00:00	2.8805	16.2596	24.2075	Potencial
17370	2023-12-25 18:00:00	1.783	16.2849	24.2933	Potencial
17369	2023-12-25 17:00:00	...	...	...	Potencial

#### Detalle de los registros con anomalías del cliente.

#### Distribución 3D de Anomalías

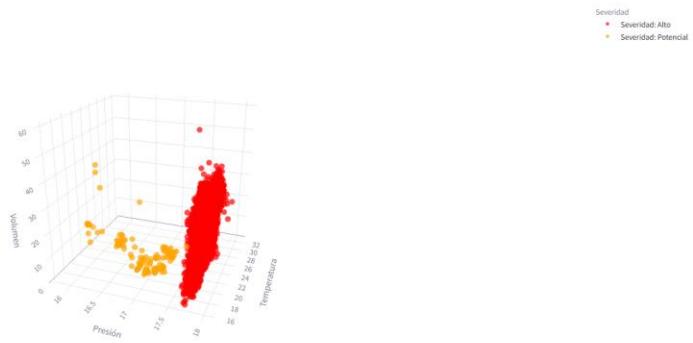


Gráfico 3D interactivo de las anomalías por severidad.

## 6.6. Resumen Descriptivo por Cliente

Se presenta un resumen descriptivo del cliente con métricas de interés tales como:

#### **consumo\_promedio** Promedio horario del volumen de gas.

<b>consumo_minimo</b>	Mínimo volumen de gas registrado.
<b>consumo_maximo</b>	Máximo volumen de gas registrado.
<b>desviacion</b>	Desviación estándar del consumo.
<b>CUM</b>	Consumo total en los últimos 30 días.
<b>rel_CUM</b>	Relación entre CUM y el promedio histórico.
<b>num_Alto</b>	Número de anomalías de severidad Alto.
<b>num_Potencial</b>	Número de anomalías de severidad Potencial.
<b>num_normales</b>	Número de lecturas normales.
<b>total_anomalias</b>	Total de alertas (Alto o Potencial).
<b>porcentaje_anomalias</b>	Porcentaje de registros con anomalías (Alto o Potencial).

Nota: en el dashboard se tiene la funcionalidad de poder descargar el resultado en formato CSV.

## 7. Consideraciones sobre Despliegue en Producción

Este proyecto ha sido diseñado de manera modular y escalable, lo cual permite su **despliegue en entornos de producción** si así se requiere. Gracias al uso de tecnologías como **Streamlit**, **SQLite**, y la estructura de procesamiento en Python, es posible adaptar el sistema para operar de forma continua en servidores locales o en la nube (como AWS, GCP o Azure), brindando acceso remoto y seguro al dashboard analítico para equipos operativos y administrativos.

**DEMO:** En el siguiente link se puede consultar el DEMO del tablero control si se desea ver con mayor detalle: <https://www.youtube.com/watch?v=o55bggUeFD8>

