

Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Ciencias de la Tierra

Librería de balance de materiales en yacimientos subsaturados,
utilizando lenguaje de programación Python para la estimación de
reservas de hidrocarburos

Proyecto Integrador

Previo la obtención del Título de:

Ingeniero en Petróleo

Presentado por:

Erick Michael Villarroel Tenelema

Kevin Steeven López Soria

Guayaquil - Ecuador

Año: 2024

DEDICATORIA

El presente proyecto lo dedico en primer lugar, al pilar fundamental de toda mi vida, Johnny Villarroel y Mery Tenelema, mis padres. Sin ellos no hubiera sido posible nada de lo que he logrado hasta ahora. A mi madre por enseñarme la humildad y sabiduría con el amor más puro que he podido conocer. A mi padre, por enseñarme a ser firme, capaz y nunca rendirme.

A ti, abuelita, Esther, que partiste pronto y no pudiste ver todos mis logros, donde sea que estés, te dedico esta empresa, como siempre te gustó.

Finalmente, a cada persona que siempre ha creído en mí, familia y amigos que llevo presente en mis triunfos.

Erick Villarroel

Dedico este proyecto, en primer lugar, a mis padres, Sonia Patricia Soria Zamora y Nel Sandro López Cusme, quienes siempre me han apoyado y brindado ánimos a lo largo de mi carrera. Sin ellos, llegar hasta donde la vida me ha colocado hoy nunca hubiera sido posible. Aún recuerdo las palabras que me dijo mi madre cuando culminé mis estudios de educación general: “Hijo, con la ayuda de Dios y de tus padres lo vas a lograr. Tomes la decisión que tomes, siempre estaremos para ti. Solo debes confiar en que las cosas se darán a su debido tiempo.”

Finalmente, quisiera destacar que hay personas inspiradoras que nos demuestran que fracasar no significa que sea el final del camino. Al contrario, cómo se levantan y vuelven a poner todo de sí mismos en ello les ayuda a alcanzar sus metas. Estas personas son un ejemplo para seguir.

Kevin López

AGRADECIMIENTOS

Mi más sincero agradecimiento a mis padres por su apoyo incondicional, tanto emocional como económico.

También, a mi alma mater, ESPOL. La universidad en la que siempre quise estar y la que me ha formado en valores y conocimientos.

Especial agradecimiento de igual manera al Ingeniero Freddy Carrión, profesor de ESPOL y tutor de esta tesis, quien nos ha sabido guiar por el camino tecnológico y a ser capaces de aprender nuevas cosas que complementan nuestra formación profesional y académica. Igualmente, al Ingeniero Oscar Cortes de SLB y tutor externo voluntario de este trabajo, quién también ha sido un guía y mentor.

Erick Villarroel

Quisiera agradecer a las personas que siempre estuvieron para mí, brindándome su apoyo constante a lo largo de mi carrera profesional. En primer lugar, a mis padres, sin cuya ayuda y aliento, llegar hasta este punto jamás hubiera sido posible. Además, agradezco a los docentes de esta prestigiosa universidad, de quienes tuve la oportunidad de adquirir valiosos conocimientos y enseñanzas.

Hago una mención especial al Msc. Freddy Carrión, nuestro tutor de tesis, y a nuestro tutor externo y voluntario, Oscar Cortes, profesional de SLB. Ellos han sido nuestros principales guías en el uso de herramientas digitales en la industria energética.

Finalmente, agradezco a las amistades que logré forjar durante esta hermosa experiencia. De ellos aprendí muchas cosas más allá de lo académico, y han dejado una gran huella en la persona que soy hoy.

Kevin López

Declaración Expresa

Nosotros **Erick Michael Villarroel Tenelema** y **Kevin Steeven López Soria** acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique a los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 30 de mayo del 2024.

Erick Michael Villarroel Tenelema

Kevin Steeven López Soria

Evaluadores

ANDRES
EDUARDO
GUZMAN
VELASQUEZ

Firmado digitalmente
por ANDRES
EDUARDO GUZMAN
VELASQUEZ
Fecha: 2024.09.09
16:27:21 -05'00'

Msc. Andrés Guzmán

Profesor de Materia



Firmado electrónicamente por:
**FREDDY PAUL CARRION
MALDONADO**

Msc. Freddy Carrión

Tutor de proyecto

Resumen

La industria petrolera enfrenta el desafío constante de estimar con precisión las reservas de crudo, una tarea crítica para la planificación y toma de decisiones. Sin embargo, las herramientas existentes a menudo son complejas, costosas y poco accesibles.

El objetivo de este estudio es desarrollar una herramienta de cálculo de reservas de petróleo accesible y eficiente. Para ello, se ha seleccionado el balance de materiales como metodología debido a que se basa en cuatro principios fundamentales: precisión, adaptabilidad, integración de datos, y proyección. Con el fin de abordar las limitaciones de softwares comerciales, se ha desarrollado una librería en Python denominada Pytank.

El proyecto se dividió en tres etapas: el desarrollo del código base, la creación de una documentación digital para su correcto uso con ejemplos prácticos y el diseño de una interfaz gráfica que funcione con la lógica de la librería.

Para la evaluación y validación de Pytank se calculó las reservas del tanque Espol donde los resultados obtenidos (71.29 MMStb) mostraron una alta correlación con valores proporcionados por un software comercial (70.14 MMStb).

Se concluye que Pytank brinda una solución innovadora y accesible, demostrando su confiabilidad y precisión respecto a otras herramientas que tienen el mismo fin. Además, su carácter de código abierto fomenta la colaboración y el desarrollo continuo de la herramienta.

Palabras Clave: Balance de materiales, reservas de petróleo, Python, Pytank.

Abstract

The oil industry faces the constant challenge of accurately estimating crude oil reserves, a critical task for planning and decision making. However, existing tools are often complex, expensive and inaccessible.

The objective of this study is to develop an accessible and efficient oil reserves calculation tool. To this end, material balance has been selected as the methodology because it is based on four fundamental principles: accuracy, adaptability, data integration, and projection. In order to address the limitations of commercial software, a Python library called Pytank was developed.

The project was divided into three stages: the development of the code base, the creation of a digital documentation for its correct use with practical examples, and the design of a graphical interface that works with the library logic.

For the evaluation and validation of Pytank, the reserves of the Espol tank were calculated where the results obtained (71.29 MMStb) showed a high correlation with values provided by a commercial software (70.14 MMStb).

It is concluded that Pytank provides an innovative and accessible solution, demonstrating its reliability and accuracy with respect to other tools that have the same purpose. In addition, its open-source nature encourages collaboration and continuous development of the tool.

Keywords: Material balance, hydrocarbon reserves, programming, Python, Pytank.

Índice general

Dedicatoria.....	II
Agradecimientos.....	IV
Evaluable.....	VII
Resumen	VIII
<i>Abstract</i>	IX
Índice general.....	X
Abreviaturas.....	XIII
Simbología.....	XIV
Índice de tablas.....	XVI
Índice de Imágenes.....	XVI
Índice de Ilustraciones.....	XVI
Índice de Figuras	XVII
Índice de Gráficos	XVII
Capítulo 1. Introducción	1
1. Identificación del problema.....	2
1.1 Descripción del Problema.....	3
1.2 Justificación del Problema.....	3
1.3 Objetivos.....	4
1.3.1 Objetivo general.....	4
1.3.2 Objetivos específicos.....	4
1.4 Antecedentes	4
1.4.1 Antecedentes Teóricos	5
1.4.2 <i>Antecedentes Históricos</i>	5
1.5 Introducción al balance de materiales en Yacimientos Subsaturados.....	5
1.6 Fundamentos del balance de materiales	7
1.7 Balance de Materiales en Yacimientos Subsaturados	9

1.8	Influjo de Agua en Yacimientos	10
1.9	Modelos para cuantificar el influjo de agua	11
1.10	Método analítico: Presión de yacimiento sintética o calculada	13
1.11	Implementación de Herramientas Computacionales.....	14
Capítulo 2: Metodología		17
2.	Desarrollo de la librería Pytank	18
2.1	Flujo de trabajo general del proyecto	18
2.2	Algoritmo general de la librería	19
2.2.1	Clase Well.....	20
2.2.2	Clase OilModel	21
2.2.3	Clase WaterModel.....	21
2.2.4	Clase Tank	22
2.2.5	Clase Aquifer	24
2.2.6	Clase Analysis.....	25
2.2.7	Clase VectorData	27
2.2.8	Clase PVTSchema.....	28
2.3	Desarrollo integral del código correspondiente a la librería.....	28
2.4	Estructura de la librería	33
2.5	Desarrollo integral del aplicativo web.....	35
2.6	Desarrollo integral de la documentación.....	37
Capítulo 3: Resultados y Análisis		41
3.	Pytank.....	42
3.1	Uso de la librería	42
3.2	Pytank View.....	48
3.2.1	Caso 1: Yacimiento volumétrico sin acuífero.....	50
3.2.2	Caso 2: Yacimiento con Acuífero utilizando Fetkovich	55
3.2.3	Caso 3: Yacimiento con Acuífero utilizando Carter Tracy	57

3.3	Ajuste de presiones.....	59
3.4	Comparación entre data real y obtenida	61
3.5	Comportamiento del tanque o reservorio	62
3.6	Documentación de Pytank	65
3.6.1	Home	67
3.6.2	Getting Started	67
3.6.3	User Guide	68
3.6.4	Development.....	69
3.6.5	API Reference	70
Capítulo 4: Conclusiones y recomendaciones		71
4.	Conclusiones y recomendaciones	72
4.1	Conclusiones.....	72
4.2	Recomendaciones.....	74
Bibliografía		75
Anexos		79

Abreviaturas

API	American Petroleum Institute
CSS	Hojas de estilo en Cascada
ESPOL	Escuela Superior Politécnica del Litoral
GOES	Gas Original En Sitio
HTML	Lenguaje de Marcado de Hipertexto
IDE	Integrated Development Environment
POE	Petróleo Original En Sitio
POO	Programación Orientada a Objetos
PVT	Presión, Volumen y Temperatura
SCV	Sistema de Control de Versiones

Simbología

Boi	Factor volumétrico inicial del petróleo
Bg	Factor volumétrico del gas
Bginy	Factor volumétrico del gas inyectado
Bo	Factor volumétrico del petróleo
Bw	Factor volumétrico del agua
cf	Compresibilidad de la roca
co	Compresibilidad del petróleo
cw	Compresibilidad del agua
D	Darcy
Efw	Expansión de agua connata y formación
Eg	Expansión de gas
Eo	Expansión de petróleo
Et	Expansión total de fluidos
F	Fluidos producidos
ft	Pies
Gp	Gas producido
J	Índice de productividad
m	Capa de gas
N	Reservas de petróleo originales en sitio
Np	Petróleo producido
Pa	Presión promedio
Pb	Presión de Burbuja
Pi	Presión Inicial
P	Presión

Pr	Presión de reservorio
Psi	Libras por pulgada cuadradas
Rp	Relación gas-petróleo acumulado
Rsb	Gas en solución en el punto de burbuja
Rsi	Gas en solución inicial
Rs	Gas en solución actual
Swc	Saturación crítica de agua
tD	Tiempo adimensional
Δp	Variación de presión
Δt	Variación de tiempo
We	Influjo de Agua
Wei	Influjo de agua inicial
Winy	Agua inyectada
Wp	Agua producida
γ_g	Gravedad específica del gas
γ_o	Gravedad específica del petróleo

Índice de tablas

Tabla 1 Propiedades PVT	7
Tabla 2. Resultados reales vs Obtenidos con la librería	62

Índice de Imágenes

Imagen 1. Repositorio en la página online de GitHub	29
Imagen 2. Interfaz de PyCharm	30
Imagen 3. Sección de Pycharm donde se clona el repositorio	30
Imagen 4. Dependencias del ambiente virtual creado por Poetry en el archivo .toml	31
Imagen 5. Sección de Pycharm donde se agrega un interpretador	32
Imagen 6. Sección de Pycharm donde se agrega el ambiente virtual creado con Poetry	32
Imagen 7. Sección de GIT para el control de versiones del proyecto	33
Imagen 8. Directorios del repositorio para la aplicación web en Pycharm	36
Imagen 9. Interfaz gráfica de Pytank View	37
Imagen 10. Esquema general de los directorios que comprenden la documentación en línea	40
Imagen 11. Interfaz gráfica de Pytank View – Módulo Well	49
Imagen 12. Interfaz gráfica de Pytank View – Módulo Fluid Models	49
Imagen 13. Interfaz gráfica de Pytank View – Módulo Tank	50
Imagen 14. Interfaz del modulo Analysis de Pytank View	51
Imagen 15. Interfaz gráfica de Pytank View – Módulo Tank con modelo de Fetkovich	55
Imagen 16. Interfaz gráfica de Pytank View – Módulo Analysis con la opción del método analítico	56
Imagen 17. Resultados en software comercial	61
Imagen 18. Interfaz gráfica de la página de inicio	66
Imagen 19. Interfaz gráfica del módulo Getting Started	67
Imagen 20. Interfaz gráfica del módulo "User Guide"	68
Imagen 21. Interfaz gráfica del módulo "API Reference"	70

Índice de Ilustraciones

Ilustración 1. Presiones Calculadas vs Presiones reales	14
Ilustración 2. Principales comandos de Git	16
Ilustración 3. Flujo de trabajo general del proyecto integrador	18

Ilustración 4. Algoritmo General de la Librería	19
Ilustración 5. Algoritmo para la clase Well	20
Ilustración 6. Algoritmo para la clase OilModel	21
Ilustración 7. Algoritmo para la clase WaterModel	22
Ilustración 8. Algoritmo de la clase Tank.....	23
Ilustración 9. Algoritmo de clase Aquifer.....	24
Ilustración 10. Algoritmo de la clase Analysis.....	26
Ilustración 11. Flujo de trabajo para el desarrollo de la lógica de la librería	28
Ilustración 12. Algoritmo de la librería	33
Ilustración 13. Flujo de trabajo para la creación de la aplicación web.....	35
Ilustración 14. Flujo de trabajo para el desarrollo de la documentación	38
Ilustración 15. Algoritmo para la Clase VectorData	79

Índice de Figuras

Figura 1. Conceptualización de un modelo de tanque para un reservorio de hidrocarburo	6
Figura 2. Gráfica Havlena & Odeh para Yacimientos Subsaturados.....	10
Figura 3. Gráfica de Campbell.....	11
Figura 4. Estados continuo y no continuo	12

Índice de Gráficos

Gráfico 1. Gráfico de Campbell correspondiente al “Tanque” para el aplicativo web vs obtenido en la librería	52
Gráfico 2. Grafica de Havlena y Odeh para el método grafico – Sin aporte de acuífero para el aplicativo web vs obtenido en la librería	53
Gráfico 3. Graficas del comportamiento de la presión promedio acuífero para el aplicativo web vs obtenido en la librería.....	54
Gráfico 4. Método de Havlena & Odeh con aporte de acuífero	55
Gráfico 5. Presiones reales y Sintéticas para el método analítico – Con aporte de acuífero	57
Gráfico 6. Havlena y Odeh para el método gráfico – Con aporte de acuífero.....	58
Gráfico 7. Presiones Reales y Sintéticas para el método analítico – Con aporte de acuífero	59
Gráfico 8. Presiones Promedios ajustada según el usuario.....	60
Gráfico 9. Havlena y Odeh para el método gráfico – Ajuste de presiones promedios	60
Gráfico 10. Presiones Reales y Sintéticas para el método analítico – Ajuste de presiones	61

Gráfico 11. Módulo Analysis con la opción para escoger el tramo deseado en las gráficas de Campbell y Havlena	63
Gráfico 12. Gráficas de Campbell en un tramo seleccionado	64
Gráfico 13. Gráficas de Havlena & Odeh en un tramo seleccionado.....	64
Gráfico 14. Análisis exploratorio con la producción acumulada de petróleo y agua de cada pozo perteneciente al tanque o reservorio	65
Gráfico 15. Gráfico Presiones Observadas vs Tiempo	79
Gráfico 16. Caudales totales del tanque ESPOL a través del tiempo.....	80
Gráfico 17. Presiones promedio vs producción acumulada de agua y petróleo del tanque ESPOL	80
Gráfico 18. Producción acumulada de petróleo y agua a través del tiempo del tanque ESPOL .	80
Gráfico 19. Caudales de petróleo y agua por pozos a través del tiempo del tanque ESPOL	81

Capítulo 1. Introducción

1. Identificación del problema

El petróleo, como fuente de energía, ofrece varias ventajas que radican en su transporte y almacenamiento, en comparación a otras opciones, como el carbón o la energía nuclear (Parra, 2003, como se citó en Toledo, 2010).

En la actualidad, el petróleo y sus derivados son los combustibles que mueven al mundo: gasavión, gasolina, turbosina, diésel, gasóleo ligero y pesado (Chow, 2002, como se citó en Toledo, 2010). Es de suma importancia saber en qué cantidad se dispone de este recurso. Por este motivo, según (Toledo, 2010), las reservas de petróleo y su estimación son uno de los temas principales de este sector.

Existen varias metodologías para el cálculo de reservas, una de las más importantes es mediante la ecuación de balance de materiales. Esta herramienta permite estimar el petróleo y gas en sitio. También evalúa los mecanismos de producción y el historial del yacimiento, permitiendo predecir su comportamiento futuro (Satter & Iqbal, 2016).

Hoy en día la mayoría de los procesos y cálculos están automatizados. Existe un desarrollo constante de nuevos programas informáticos, aplicaciones móviles, software, etc., que optimizan procesos. Conforme a esto, el balance de materiales ha evolucionado codificándose en diferentes lenguajes de programación, dando paso a varios simuladores y aplicaciones que determinen el volumen de hidrocarburos en sitio (Bakhty et al., 2020). Una herramienta para el desarrollo de estos programas es Python, un lenguaje de programación orientado a objetos de muy alto nivel (Martínez Rafael et al., 2014).

Finalmente, un buen prospecto para estimar sus reservas es el campo ESPOL, ubicado en la Cuenca Oriente del Ecuador. Este campo tiene datos de producción, presión y tablas con información PVT, fundamentales para el uso de métodos como el balance de materiales.

1.1 Descripción del Problema

La transformación digital en la Industria 4.0 ha impulsado la necesidad de herramientas informáticas eficientes y productivas (Garrell & Llorenc, 2019), para el sector energético, especialmente en el sector petrolero. A pesar de la variedad de software para diversas tareas, la carencia de opciones software de código abierto y de fácil interpretación limita el desarrollo de programas informáticos dedicados a estimar reservas de petróleo.

Actualmente, para usar un software se requiere de una inversión inicial (licencias comerciales) y su manejo debe con una preparación preliminar compleja, excluyendo a muchos profesionales, investigadores y estudiantes. Además, ahora se requieren soluciones innovadoras que optimicen y automaticen procesos, enfocándose en habilidades de programación. Python, es un lenguaje de programación que está en vanguardia; muchos programas hoy en día están creados a partir de este lenguaje y se necesita que se siga impulsando su aprendizaje y desarrollo (Pérez et al., 2014)

Por último, en la mayoría de los casos, las herramientas gratuitas son asociadas a falta de precisión en sus resultados y a la poca versatilidad que brindan a la hora de llevar a cabo análisis más detallados o enfrentarse a distintos escenarios. Por este motivo, la mayoría de las empresas del sector petrolera tienen sus propias herramientas o dependen de servicios pagados para softwares que satisfagan diferentes necesidades (Zambrano et al., 2017).

1.2 Justificación del Problema

La creación de una librería en Python ayudaría a promover la innovación en la industria energética, no solo a nivel del país, sino mundialmente. Al ser una librería Open-Source, de código abierto, su naturaleza permite que investigadores, estudiantes y profesionales colaboren en su desarrollo y mejora.

De igual manera, ayudaría a las empresas a optimizar el proceso de estimar el volumen de reservas de manera automatizada y precisa, usando el método de balance de materiales. Conjunto

a esto, al mejorar la precisión en su estimación, se puede evitar realizar operaciones de perforación en áreas que no posean reservas significativas, reduciendo la contaminación asociada a las operaciones, evaluando la viabilidad de proyectos petroleros que adopten prácticas sostenibles (Fernández et al., 2020).

Por último, este proyecto influye en la economía, ya que una herramienta digital gratuita ahorra mucho en adquirir softwares comerciales que sirven para el mismo fin (Gómez, 2004) y reducir la desigualdad con el acceso al conocimiento y las oportunidades (Solano et al., 2015). Además de tener una herramienta gratuita que brinde una precisión similar a softwares comerciales con el mismo fin.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar una librería de balance de materiales en yacimientos subsaturados, utilizando el lenguaje de programación Python, para la estimación de reservas de hidrocarburos.

1.3.2 Objetivos específicos

1. Aplicar los principios del balance de materiales y los conceptos fundamentales de programación en Python, para el diseño inicial de la librería.
2. Implementar la librería en Python, para que sea integrada a un manejador de paquetes (PyPi).
3. Evaluar el uso de la librería integrada a un aplicativo web, para la estimación de reservas hidrocarburífera de yacimientos subsaturados.
4. Diseñar un manual de usuario para el manejo correcto de la librería.

1.4 Antecedentes

Como punto de partida para desarrollar este proyecto de tesis, se han considerado los siguientes trabajos previos que abordan aspectos claves del tema de estudio.

1.4.1 Antecedentes Teóricos

- Aguas & Yoon (2023) realizaron un estudio utilizando análisis exploratorio de datos (EDA) con la finalidad de descubrir tendencias en información relacionada con propiedades PVT (presión, volumen y temperatura). Desarrollaron un repositorio que contienen diferentes funciones en lenguaje de programación Python. El propósito de su trabajo fue calcular el volumen de reservas para yacimientos subsaturados con influjo de agua.

1.4.2 Antecedentes Históricos

- Un ejemplo histórico es el trabajo de Canel & Rosbaco (1992), quienes desarrollaron un proyecto exitoso. Utilizaron un modelo composicional diseñado para simular cualquier tipo de fluido: retrógrado, condensado, reinyección de gas, inyección de agua. En su trabajo, simulaban el comportamiento termodinámico de fluidos en los yacimientos aplicando la metodología de balance de materiales con un software con resultados como producción acumulada de petróleo, gas y agua según los cambios de presión y tiempo.

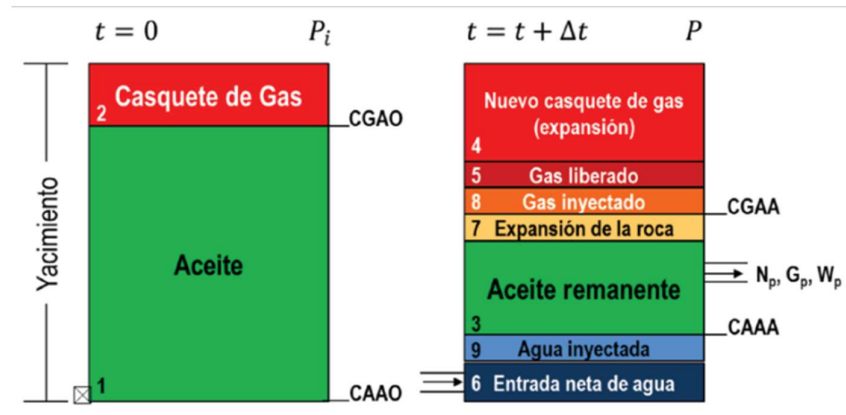
1.5 Introducción al balance de materiales en Yacimientos Subsaturados

Según Ahmed (2018), los yacimientos subsaturados se definen como aquellos reservorios de petróleo donde la presión inicial (P_i) del yacimiento resulta ser mayor a su presión de burbuja (P_b). En estos casos, el petróleo se encontrará en una fase líquida y el gas se estará presente en forma de gas disuelto en el crudo, sin formar una capa de gas segregada.

El volumen de hidrocarburo presente inicialmente en un reservorio, se denomina petróleo original in situ, se denominado POES. Al estimar las reservas totales de un reservorio, se debe considerar diferentes aspectos geológicos, como la distribución de porosidad, la saturación de fluidos del reservorio y la geometría del yacimiento (Jordy & Kevin, 2018).

La ecuación de balance de materiales es un modelo determinístico implementado como una herramienta que permite interpretar y predecir el comportamiento de los yacimientos, bajo el principio de conservación de la masa comparando el volumen residual de petróleo y el agua invadida durante la producción (Dake, 1983).

Figura 1. Conceptualización de un modelo de tanque para un reservorio de hidrocarburo



Fuente: Balance de materiales en yacimientos de aceite negro de arenas lenticulares de Fernanda et al., 2019

Entre las diferentes aplicaciones de la ecuación de balance de materia tenemos (Ferrer, 2009):

- Estimación del POES y GOES sin necesidad de datos de pruebas de producción.
- Ayuda a predecir el comportamiento futuro de un reservorio.
- Proporciona información sobre los mecanismos de producción y recuperación.

Sin embargo, el balance de materia presenta ciertas limitaciones como (Ferrer, 2009):

- Asume que el yacimiento se comporta como un sistema cerrado y homogéneo.
- Requiere información confiable de propiedades PVT de los fluidos del yacimiento.

Para un criterio preciso al evaluar un yacimiento es sumamente importante establecer las propiedades físicas del petróleo, gas y agua en condiciones estáticas y dinámicas, tanto el fondo del pozo y en superficie. Estas propiedades se obtienen con correlaciones empíricas a partir de datos del campo o mediante pruebas experimentales en laboratorio (Wheaton, 2016).

La implementación de correlaciones empíricas es una solución más económica que las pruebas de laboratorio, ya que permiten establecer las propiedades de los fluidos de interés mediante cálculos matemáticos, si la información utilizada representa el reservorio (Escobar, 2012).

La implementación de correlaciones empíricas es una solución más económica en comparación que las pruebas de laboratorio, ya que permiten establecer las propiedades de los fluidos de interés mediante cálculos matemáticos, siempre que la información utilizada sea representativa del reservorio (Escobar, 2012).

Hay diferentes correlaciones empíricas, pero cada una dependerá de la información de la que se disponga. A continuación, se tiene las siguientes correlaciones para el petróleo:

Tabla 1 Propiedades PVT

Propiedades del Petróleo	
Presión del punto de burbuja (Pb): Standing	$P_b = 18,2 \left[\left(\frac{R_s}{\gamma_g} \right)^{0,83} (10^a) - 1,4 \right]$ $a = 0,00091(T - 460) - 0,0125(API)$
Gas en solución (Rs): Standing	$R_s = \gamma_g \left[\left(\frac{P}{18,2} + 1,4 \right) 10^x \right]^{1,204}$
Factor de compresibilidad del petróleo (Co): Vásquez and Beggs	$c_o = \frac{-1433 + 5R_{sb} + 17,2(T - 460) - 1180\gamma_g + 12,61(API)}{10^5 P}$
Factor volumétrico de formación del petróleo (Bo): Standing	$B_o = 0,9759 + 0,000120 \left[R_s \sqrt{\frac{\gamma_g}{\gamma_o}} + 1,25(T - 460) \right]^{1,2}$

Fuente: Fundamentos de Ingeniería de yacimiento, Magdalena P. Ferrer (2009)

1.6 Fundamentos del balance de materiales

La ecuación de balance de materiales se basa en el principio de conservación de masa, relacionando el volumen original de hidrocarburos con los volúmenes producidos y los cambios en sus propiedades de los fluidos durante su producción. La ecuación puede expresarse de la

siguiente manera: Volumen Original de hidrocarburos = Volumen de hidrocarburos producidos + volumen de hidrocarburos remanentes en el yacimiento.

$$N = \frac{N_p[B_o + (R_p - R_s)B_g] - (W_e - W_p B_w) - G_{iny} B_{giny} - W_{iny} B_w}{(B_o - B_{oi}) + (R_{si} - R_s)B_g + mB_{oi}\left(\frac{B_g}{B_{gi}} - 1\right) + B_{oi}(1+m)\left(\frac{c_w S_{wc} + c_f}{1 - S_{wc}}\right)\Delta p} \quad (\text{Ec. 1,1})$$

De la ecuación podemos identificar los siguientes términos:

- Expansión del petróleo y gas disuelto

$$N(R_{si} - R_s)B_g \quad (\text{Ec. 1,2})$$

$$N(B_o - B_{oi}) \quad (\text{Ec. 1,3})$$

- Expansión de capa de gas

$$mNB_{oi}\left(\frac{B_g}{B_{gi}} - 1\right) \quad (\text{Ec. 1,4})$$

- Cambio en el volumen poroso de hidrocarburos debido a expansión del agua y reducción del volumen poroso

$$NB_{oi}(1 + m)\left(\frac{c_w S_{wc} + c_f}{1 - S_{wc}}\right)\Delta p \quad (\text{Ec. 1,5})$$

- Vaciamiento del volumen poroso

$$N_p[B_o + (R_p - R_s)B_g] \quad (\text{Ec. 1,6})$$

Escobar (2012) señala que para aplicar la ecuación de balance de materia se debe considerar lo siguiente:

- Volumen poroso constante.
- Equilibrio termodinámico entre fluidos del yacimiento.
- Procesos a la misma temperatura.
- Efectos de compresibilidad del agua de formación y formación despreciables.
- Sistema cerrado y homogéneo.

- Información PVT representativa.

1.7 Balance de Materiales en Yacimientos Subsaturados

Havlena y Odeh (1963) lograron desarrollar una técnica gráfica para resolver la ecuación general de balance de materiales en yacimientos de hidrocarburo. El método requerirá identificar el mecanismo por el que se encuentra produciendo un reservorio. La exactitud de los resultados dependerá de la veracidad y cantidad de información de la que se disponga (Ferrer, 2009).

La ecuación de balance de materiales se redujo significativamente hasta obtener la siguiente ecuación:

$$F = N (E_o + E_{fw} + mE_g) + W_e \text{ (Ec. 1,7)}$$

En donde cada termino representa:

- F: Fluidos producidos por el yacimiento
- N: Petróleo Original In Situ (POES)
- E_o: Expansión de petróleo
- E_{fw}: Expansión de agua connata y formación
- E_g: Expansión de gas
- m: Capa de gas
- W_e: Influjo de Agua

Para Yacimientos Subsaturados, suponiendo que durante el proceso de producción de un reservorio no existe inyección de fluidos para mejorar su producción, de la ecuación descrita anteriormente, se pueden eliminar varios términos como:

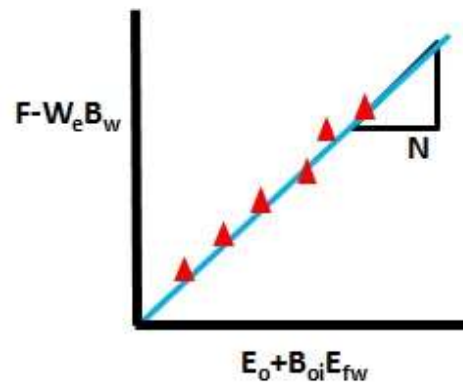
- m: Al ser un yacimiento subsaturado no se tendrá una capa de gas asociada.
- R_s = R_{si} = R_p: El gas se encuentra disuelto en el petróleo.
- W_e: Asumiendo que inicialmente no se tiene presencia de un acuífero.

De tal manera que la ecuación (1.7) quedaría expresada de la siguiente manera:

$$F = N(E_o + E_{fw}) \text{ (Ec.1,8)}$$

De la ecuación (1.8) se obtiene una gráfica teórica que intenta predecir el comportamiento de un yacimiento subsaturado, en la cual se relacionan los fluidos producidos por el reservorio (F) vs la expansión de los fluidos ($E_o + E_{fw}$), en donde se asume que la pendiente debe ser un valor aproximado al POES (N).

Figura 2. Gráfica Havlena & Odeh para Yacimientos Subsaturados



Fuente: Manual Harmony Enterprise, 2019

1.8 Influjo de Agua en Yacimientos

Durante la etapa de producción de un campo petrolero, la presión inicial tenderá a disminuir con el tiempo. Cuando la presión disminuye, el acuífero invade el medio poroso, desplazando el volumen de hidrocarburo original del reservorio.

Por lo tanto, dependiendo de si existe presencia de un acuífero cercano a un reservorio, estos pueden ser clasificados como (Escobar, 2012):

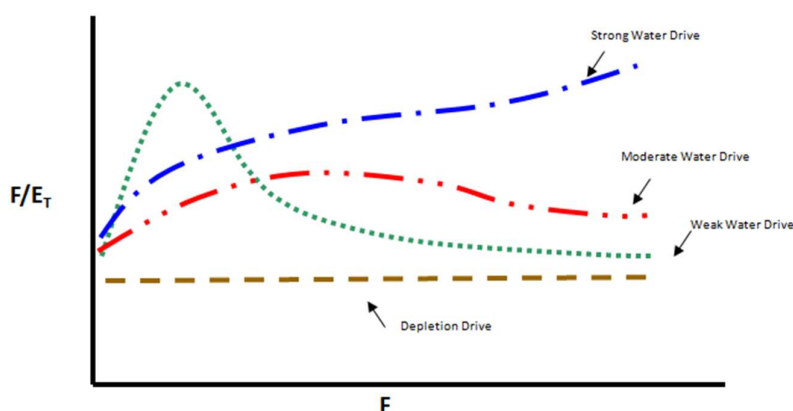
- **Volumétricos:** Son aquellos reservorios que no presentan un acuífero en zonas adyacentes.
- **No volumétricos:** Yacimientos en los que el volumen de hidrocarburo original se reduce por la intrusión de agua proveniente de un acuífero adyacente.

En un caso real, muchas veces no se tendrá información de si existe presencia o no de un acuífero y sus características. Si no se tiene una descripción precisa del acuífero, no se puede

obtener una predicción más real del comportamiento del reservorio. Campbell propuso un método gráfico para determinar si el acuífero tiene un aporte energético hacia el yacimiento.

En el gráfico de Campbell se suele graficar el retiro de fluidos producidos del yacimiento (F) en relación con la expansión total de fluidos (E_T) vs producción acumulada (N_p), en donde no se toma en cuenta el influjo de agua (IHS Markit, 2019).

Figura 3. Gráfica de Campbell



Fuente: Manual Harmony Enterprise, 2019

- Curva marrón: En caso de no existir presencia de un acuífero, la línea tendera a ser constante y horizontal.
- Curva verde: Si se tiene una curva que tiene una declinación apresurada, este se trataría de un aporte energético bajo.
- Curva roja: Si la curva muestra una ligera tendencia a aumentar, significa que se tiene un aporte energético moderado.
- Curva azul: Si hay un alza pronunciada, sería un acuífero con un alto aporte energético

1.9 Modelos para cuantificar el influjo de agua

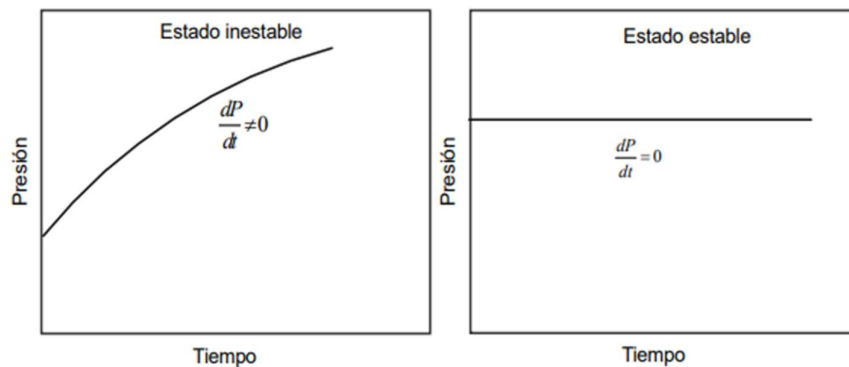
Determinar el régimen de flujo es crucial cuando se desear calcular el influjo de agua. Se puede dividir en tres categorías:

- Continuo/Estable

- Semicontinuo/Pseudoestable
- No continuo/Inestable

En un régimen estable, la variación de presión se distribuye uniformemente en todo el yacimiento, y el acuífero responde inmediatamente para mantener la presión. Si la declinación de la presión tiene un comportamiento lineal en función del tiempo en cualquier posición del yacimiento, y es proporcional a la tasa de producción del yacimiento, se tratará de un régimen semicontinuo. Por último, se tiene un régimen no continuo, cuando la variación de la presión del reservorio no tiende a variar linealmente en función del tiempo (Ferrer, 2009).

Figura 4. Estados continuo y no continuo



Fuente 1: Fundamentos de Ingeniería de yacimientos, F. Escobar (2012)

Los modelos de estado inestable representan de forma más real el comportamiento de los fluidos en un reservorio, ya que toman en cuenta los efectos transitorios de la producción por variaciones de presiones. Entre los modelos de estado no continuo podemos destacar dos: Fetkovich y Carter Tracy.

El modelo propuesto por Fetkovich obtiene resultados similares al propuesto por Hurst & Van Everdingen en 1949 quienes implementaron la ecuación de difusividad para calcular el influjo de agua como un parámetro adimensional. Este modelo propone implementar el índice de productividad (IP), para describir el influjo desde un acuífero finito hacia el reservorio (Escobar,

2012). El método de Fetkovich descarta el uso del principio de superposición lo que resulta en un proceso más simple en comparación con el método de Hurst & Van Everdingen, sin embargo, los resultados obtenidos resultan ser menores a los obtenidos mediante otros métodos debido a que descarta los periodos de tiempo transitorios al inicio de la producción.

$$(\Delta W_e)_n = \frac{W_{ei}}{P_i} * [(\bar{P}_a)_{n-1} - (\bar{P}_r)_n] * \left[1 - \exp\left(\frac{-J P_i \Delta t_n}{W_{ei}}\right) \right]$$

El modelo propuesto por Carter & Tracy (1960) también toma como referencia el modelo propuesto por Hurst & Van Everdingen, sin embargo, a diferencia del método propuesto por Fetkovich, este modelo si toma en cuenta los efectos de periodo transitorio, dando resultados más precisos.

$$(W_e)_n = (W_e)_{n-1} + [(t_D)_n - (t_D)_{n-1}] \left[\frac{B \Delta p_n - (W_e)_{n-1} (p'_D)_n}{(p_D)_n - (t_D)_{n-1} (p'_D)_n} \right]$$

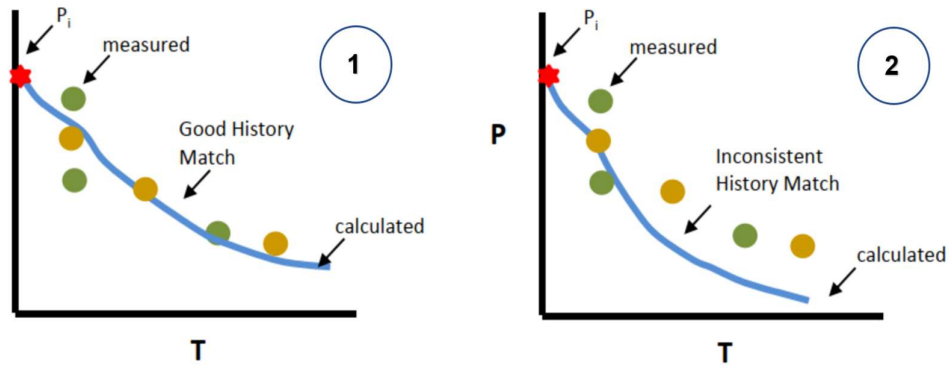
1.10 Método analítico: Presión de yacimiento sintética o calculada

Se basa en un procedimiento iterativo para ajustar la presión del yacimiento en función al tiempo utilizando los valores originales de petróleo in situ POES, gas in situ GOES y agua producida Wp. Una vez calculada la presión sintética, se grafica junto a las presiones estáticas u observadas reales del yacimiento. La gráfica corresponde a las presiones en el *eje y* versus el tiempo en el *eje x*. En esta se compara la diferencia que existe entre el comportamiento de ambas presiones. Esta técnica de balance de materiales es la más sólida debido a dos factores principales en torno a procesos de extracción acumulativa basada en el tiempo:

- La presión en función al tiempo es una tendencia fácil de entender.
- Al usar el tiempo permite analizar los cambios en las tasas de extracción, operaciones de inyección.

Para entender el comportamiento de las presiones y qué se puede deducir de ellas, se presenta la siguiente figura:

Ilustración 1. Presiones Calculadas vs Presiones reales



Fuente: Manual Harmony Enterprise, 2019

En la Ilustración 1. Se puede observar dos graficas donde los puntos representan a las presiones reales y la curva a las presiones sintéticas o calculadas:

- En la gráfica 1 se aprecia que existe un comportamiento similar entre ambas presiones. Entonces, se concluye que la combinación de valores de POES, GOES y W proporcionados para el análisis corresponden a soluciones válidas para la ecuación de balance de materiales.
- En la gráfica 2 se aprecia que existe una inconsistencia en el comportamiento de ambas presiones. Esto puede ser consecuencia de datos PVT incorrectos o con poca precisión, puntos de presión y producciones atípicos o datos de presiones incorrectos.

Las pruebas de presión son importantes en la industria petrolera, sobre todo gracias a esta información se puede caracterizar un yacimiento. La presión promedio es un parámetro importante debido a que permite determinar el comportamiento de un yacimiento a lo largo de toda su vida de producción (Ibagon & Buitrago, 2008).

1.11 Implementación de Herramientas Computacionales

Python es un lenguaje de programación de alto nivel, creado a principios de los años 90 por Guido van Rossum, basado en el lenguaje ABC y soporta múltiples paradigmas de

programación como la orientación a objetos imperativa y funcional (Duque, 2011). Su diseño enfatiza la legibilidad del código mediante el uso de sangría significativa, lo que permite desarrollar lógicas complejas en poco código y de manera legible (Pérez et al., 2014). Algunas de las estructuras que podemos encontrar son:

- Listas
- Dicionarios
- Conjuntos
- Tuplas

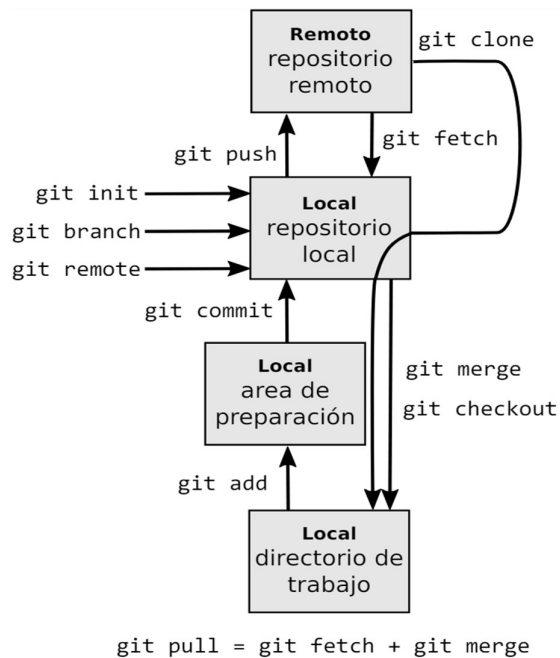
Para desarrollar la librería de este proyecto, es necesario comprender el paradigma de programación orientada a objetos (POO), el cual se diferencia de un enfoque estructurado y secuencial al basar su modelo en ciertos objetos para resolver problemas (Guagliano, 2019). Estos objetos tienen la característica de comunicarse entre sí, mediante mensajes y su funcionamiento se fundamenta en técnicas como: herencia, cohesión, abstracción, polimorfismo y encapsulación.

Poetry es una herramienta que gestiona las dependencias y permite la creación de ambientes virtuales en Python (Palacio, 2023). Permite declarar y administrar las de manera fácil las bibliotecas de las que depende un proyecto, evitando el conflicto entre versiones al utilizar varias dependencias (Jaworski, 2021).

Para el desarrollo del proyecto se implementó Poetry, para gestionar las bibliotecas necesarias en la librería, facilitando su mantenimiento y mejora por parte de otros desarrolladores en el futuro.

Git es un sistema de control de versiones (SCV) que permite gestionar de manera eficiente flujos de trabajo no lineales en proyectos de desarrollo de software. (Lopez et al., 2015). Posee diversos comandos para trabajar adecuadamente, tanto local como remotamente.

Ilustración 2. Principales comandos de Git



Fuente. GitHub como herramienta docente de Lopez et al., 2015

GitHub es una herramienta que sirve para alojar en la nube repositorios remotos. Posee una interfaz web que permite a sus usuarios crear repositorios vacíos, clonar repositorios existentes para colaborar en su trabajo, ya sea de manera local o remota (Lopez et al., 2015).

Un IDE (Integrated Development Environment) es un programa informático que integra diversas herramientas para facilitar el desarrollo de software (Bell & Parr, 2003).

Principales componentes de un IDE:

- Editor de código
- Compilador
- Depurador
- Constructor de interfaz gráfica

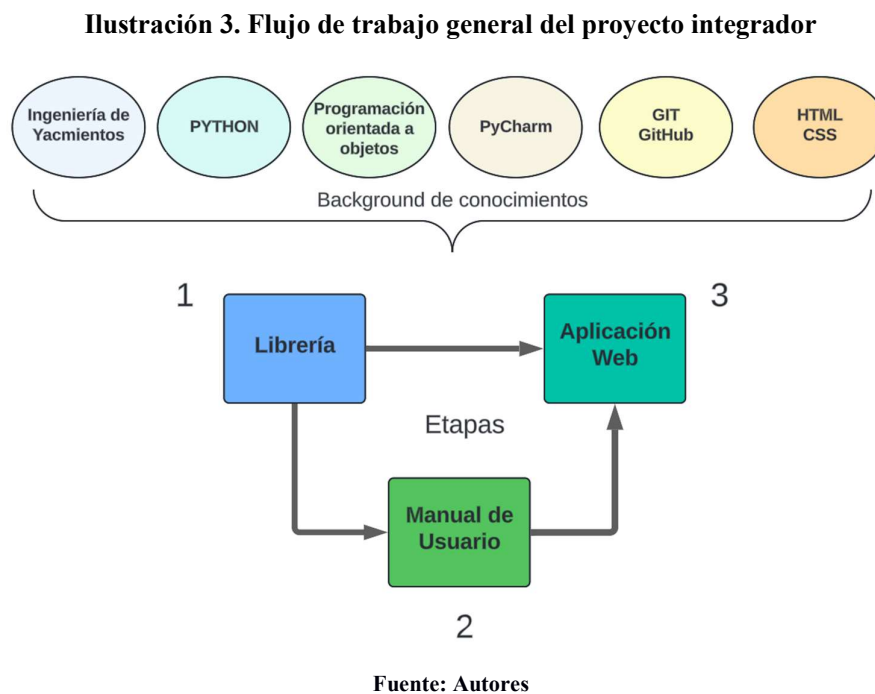
Entre los principales proveedores de sistemas IDE están: Visual Studio, IntelliJ Idea, Pycharm, Xcode, Eclipse, PhpStorm, AWS Cloud 9.

Capítulo 2: Metodología

2. Desarrollo de la librería Pytank

Este trabajo se desarrolló con una metodología cuantitativa, basada en datos históricos de producción de petróleo y agua, de presiones e información de los fluidos del campo ESPOL. El análisis llevado a cabo en esta metodología consistió en un análisis matemático, donde se estudiaron variables presentes en los tres reservorios del campo ESPOL. Estas variables se integraron en ecuaciones matemáticas establecidas a partir de la ecuación de balance de materiales, para modelar el comportamiento del reservorio y calcular las reservas

2.1 Flujo de trabajo general del proyecto



En el flujo de trabajo de la ilustración 3. se aprecia, de manera general, la secuencia utilizada para el desarrollo de este proyecto, junto con las herramientas generales empleadas. Además, este procedimiento permitió obtener progresivamente los resultados esperados.

De manera resumida se pueden identificar 3 diferentes etapas:

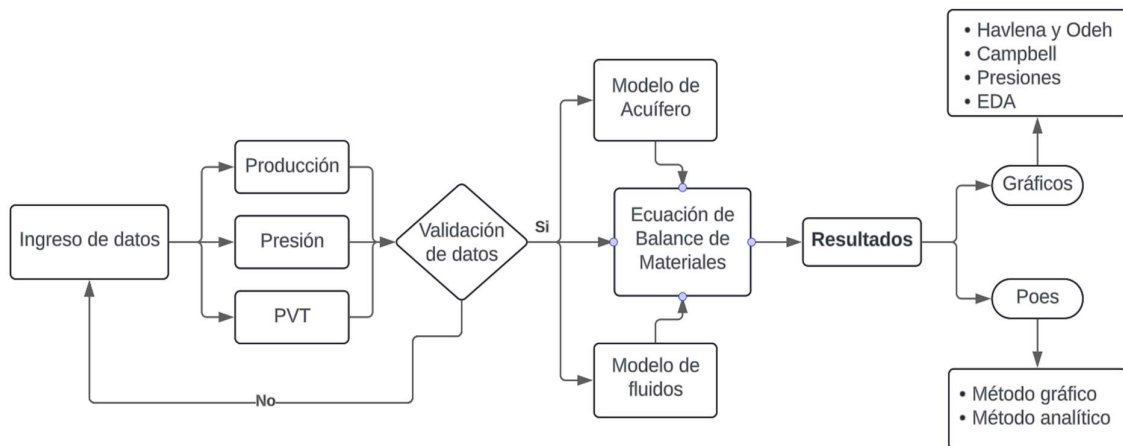
- **Primera etapa (1):** Consta de la realización de la librería en el lenguaje Python.

- **Segunda etapa (2):** Corresponde a la documentación que servirá para el manual de usuario de la librería.
- **Tercera etapa (3):** Corresponde al desarrollo de un aplicativo web que utiliza la lógica de la librería generando un interfaz gráfico

2.2 Algoritmo general de la librería

Este algoritmo identifica las variables y procesos llevados a cabo en la librería para el cálculo del POES determinístico de yacimientos subsaturados, mediante la ecuación de balance de materiales. Se puede observar en la siguiente Ilustración 4.:

Ilustración 4. Algoritmo General de la Librería



Fuente: Autores

La librería está estructurada mediante programación orientada a objetos en clases principales y secundarias, además de funciones necesarias para su estructura y funcionamiento:

Clases principales:

- Clase Well
- Clase OilModel
- Clase WaterModel
- Clase Tank

- Clase Aquifer (Clase Fetkovich y CarterTracy)
- Clase Analysis

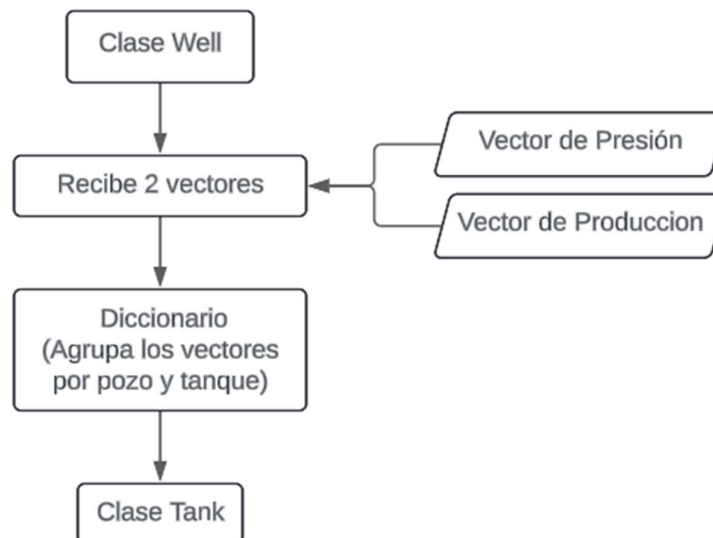
Clases secundarias (clases de validaciones de datos):

- Clase VectorData
- Clase PressVector
- Clase ProdVector
- Clase PVTSchema

2.2.1 Clase Well

En la Ilustración 5 se observa que Well recibe los vectores de presión y producción. A estos vectores los agrupa por pozo y por tanque. Al final se obtiene un diccionario, que tiene como claves los nombres de los tanques (reservorios) y como valores la lista de pozos que le corresponde a cada tanque. A su vez cada pozo contiene sus vectores de presión y producción correspondientes.

Ilustración 5. Algoritmo para la clase Well

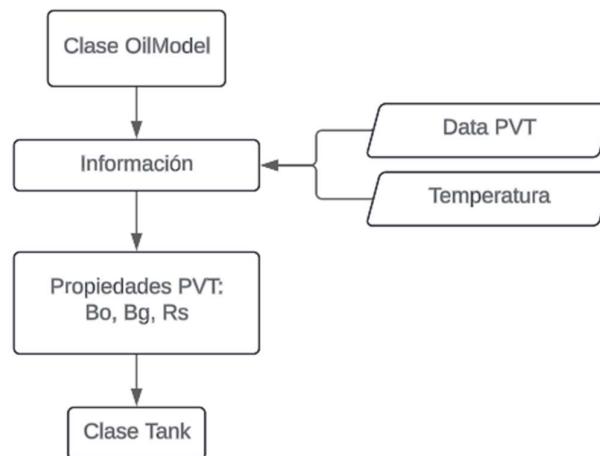


Fuente: Autores

2.2.2 Clase OilModel

La Ilustración 6 muestra que esta clase recibe información PVT y una temperatura. Con esta información calcula las propiedades PVT mediante interpolaciones hechas con el módulo interpolated (interpolación lineal) de la librería scipy de Python a partir de una presión dada.

Ilustración 6. Algoritmo para la clase OilModel



Fuente: Autores

El formato de la data PVT (xls, csv, etc) y de los datos que entran en esta clase:

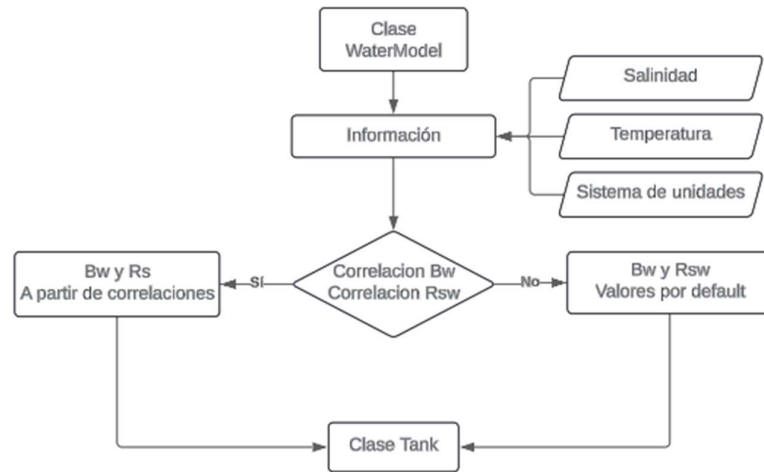
- **Tabla PVT:** DataFrame
- **Temperatura:** float (decimal)

2.2.3 Clase WaterModel

Esta clase corresponde al modelo de agua y permite calcular 2 propiedades PVT del agua necesarias a partir de 2 correlaciones respectivamente, como se aprecia en la Ilustración 7.:

- Factor volumétrico del agua (Bw) [bbls/stb]
- Solubilidad del agua (Rsw) [scf/bbls]

Ilustración 7. Algoritmo para la clase WaterModel



Fuente: Autores

Para el cálculo posterior del balance de materia. Los objetos que recibe son:

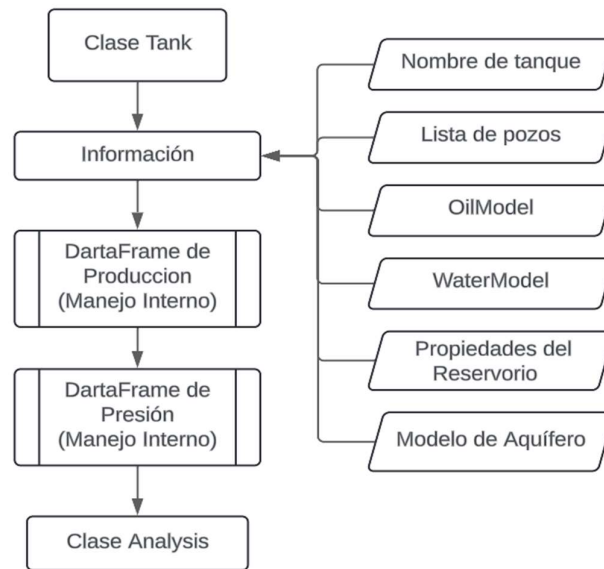
- **Salinidad:** float [ppm]
- **Temperatura:** float [°F]
- **Sistema de unidades:** En caso de poner el número 1 son unidades de campo, en caso de poner 0 son unidades métricas.

En caso de no querer utilizar correlaciones, origina valores por defecto de Bw y Rsw. Los valores son de 1 y 0 correspondientemente.

2.2.4 Clase Tank

Como se observa en la Ilustración 8 la clase Tank se maneja en función a un tanque (reservorio) específico.

Ilustración 8. Algoritmo de la clase Tank



Fuente: Autores

Sus parámetros de funcionamiento son los siguientes:

- **Nombre del tanque:** cadena de caracteres (str), ejemplo: tank_center.
- **Lista de pozos:** Lista de pozos generadas por el usuario u haciendo uso de las funciones de soporte (helpers).
- **OilModel:** Modelo de fluido de petróleo con sus propiedades PVT (Clase OilModel)
- **WaterModel:** Modelo de fluido del agua con sus propiedades PVT (Clase WaterModel)
- **Propiedades del reservorio:** Presión inicial [psi], compresibilidad de la formación [1/psi], compresibilidad de la roca [1/psi] y saturación inicial de agua [decimal].
- **Modelo de acuífero (Clase Fetkovich o Clase Carter Tracy):** Dependiendo si el yacimiento presente o no un aporte energético de un acuífero.

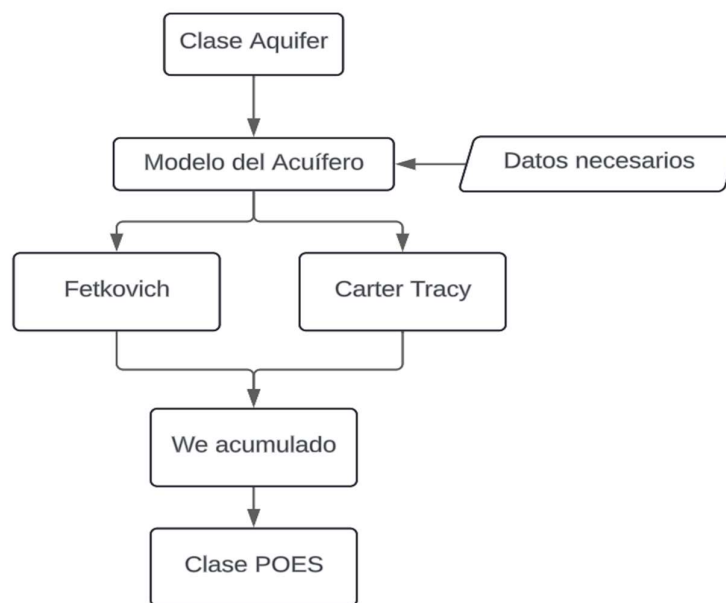
Esta clase es el enlace de todo el procesamiento anterior en las demás clases con la última clase Analysis que calcula el POES, la cual se mencionará más adelante. Su función específica es ser un contenedor de todas las propiedades que posee el tanque (reservorio). Sus métodos no son

para cálculos, sin embargo, posee métodos internos y privados que sirven para el manejo de la información de producción y presión en DataFrame que ayudarán a cálculos posteriores en la Clase Analysis

2.2.5 Clase Aquifer

La ilustración 9. muestra la clase definida con el nombre Aquifer, la cual depende de otras dos clases: Fetkovich y CarterTracy. Su algoritmo está mostrado de manera unificada, debido a que tanto la clase Fetkovich como la de Carter Tracy llegan al mismo fin. Su objetivo final es generar un DataFrame con los valores de influjo de agua (We) acumulado [bbls] con el modelo de los acuíferos ya mencionados.

Ilustración 9. Algoritmo de clase Aquifer



Fuente: Autores

Esto lo hace a partir de los siguientes datos necesarios para el modelo de acuífero:

- **Porosidad del acuífero:** float [decimal]
- **Compresibilidad total:** float [1/psi]
- **Altura del acuífero:** float [ft]
- **Ángulo del acuífero:** float [grados sexagesimales]

- **Permeabilidad del acuífero:** float [Darcy]
- **Viscosidad del agua:** float [cp]
- **Lista de Presiones:** list [psi]
- **Lista de rangos de tiempo:** list [días]

El valor de We acumulado sirve para el cálculo del POES [MMSTB] mediante la clase Analysis. Además, para el modelo de Carter Tracy, no existe un valor de radio del acuífero debido a que este modelo asume que el radio del acuífero es infinito.

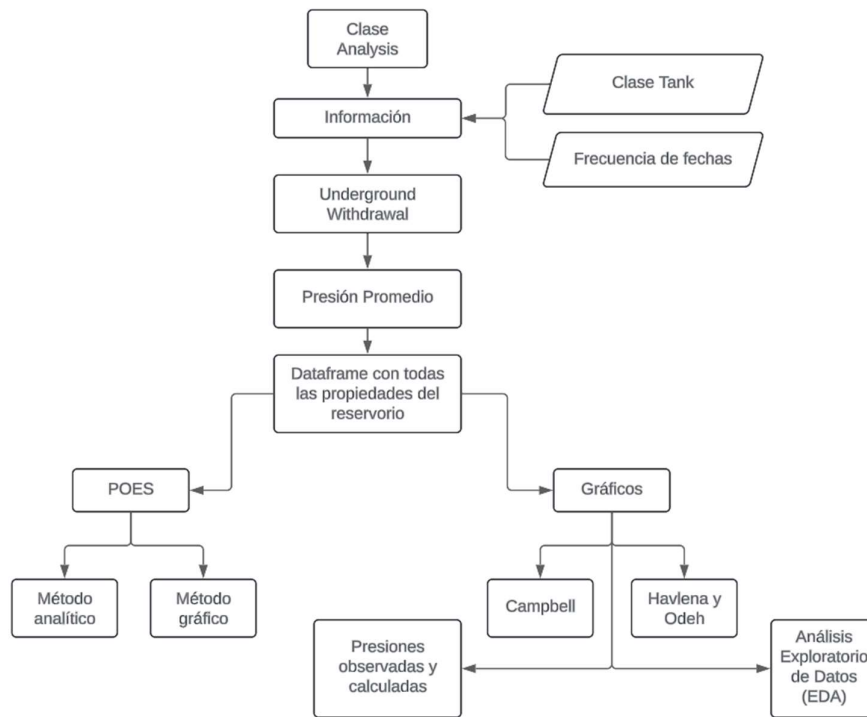
2.2.6 Clase Analysis

Esta clase es la última de la librería en crearse. Tal como muestra en la ilustración 10, a partir de ella se genera lo siguiente:

- Cálculo de Underground Withdrawal (UW): En la ecuación de balance de materiales se conoce como F [bbls].
- Cálculo de la presión promedio [psi].
- DataFrame con la información necesaria para el cálculo del POES por tanque (Reservorio).
- Gráfico de Campbell: Este gráfico brinda al usuario conocimiento acerca del aporte energético que tiene el reservorio por la presencia de un acuífero. Esto sirve de base para modelar un acuífero que está proporcionando un influjo de agua.
- Método gráfico de Havlena y Odeh: Mediante este gráfico se puede calcular el poes [MMSTB]. Aquí existen dos opciones de creación del gráfico. Por un lado, en Campbell se deduce la presencia de un acuífero, lo que implica que se debe trabajar con un yacimiento subsaturado no volumétrico. Si no hay aporte energético, se trabaja con un yacimiento subsaturado volumétrico.
- Método analítico: Este método funciona a través de un poes inferido mediante el cual se vuelve a calcular una presión sintética. El poes inferido se vuelve

significativo cuando la presión observada del reservorio tiene el mismo comportamiento que la presión nueva sintética. Entonces, se puede constatar que ese poes es el correcto para la ecuación de balance de materiales. En la ilustración 1 correspondiente el capítulo 1 del marco teórico se puede ver el comportamiento de ambas presiones en ambos casos, cuando existen similitud entre las presiones y cuando existen inconsistencias y qué se concluye respectivamente.

Ilustración 10. Algoritmo de la clase Analysis



Fuente: Autores

Las propiedades que recibe para estos cálculos son los siguientes:

- Una instancia de la Clase Tank, donde están albergadas todas las propiedades del reservorio que necesita la clase Analysis.
- Una frecuencia que establece la frecuencia de los datos a través de un string. Por ejemplo, para una frecuencia anual, se tendría que escribir “12M” que corresponden a 12 meses, un año.

- Existe otra propiedad llamada *position* que puede quedar vacía, debido a que solo es un arreglo de cómo se establece la frecuencia según los meses mostrados. En caso de querer un tipo de arreglo existen tres, que se pasan a la clase como string: “begin”, “middle” y “end”. Mostrando la frecuencia, en caso de ser “begin”, de la siguiente manera: 2019-01-01, 2020-01-01, y así sucesivamente.
- Adicionalmente, recibe 3 parámetros que sirven para ajustar el comportamiento de la presión en caso de que este se vea afectado por picos pronunciados o por ausencia de datos. Los parámetros son: *smooth* correspondiente a si se desea hacer el ajuste o no con dos opciones True en caso de hacer el ajuste y False en caso de no requerirlo; *k* correspondiente al grado de ajuste, puede ser cuadrático o polinomial, por ejemplo, 2, 3, etc.; *s* correspondiente al *smooth*, valor numérico con el que se desea ajustar más o menos la curva.

2.2.7 Clase *VectorData*

Esta es una clase secundaria. Sin embargo, su importancia no se deja a un lado. Para poder trabajar en la librería se necesitan validar datos. Justamente, este es el objetivo de esta. *VectorData* valida datos de producción y presión mediante las librerías *Pydantic* y *Pandera*. Desde la documentación oficial en línea se puede constatar que ambas librerías sirven para la validación de datos (Pandera developers, 2019; Pydantic developers, n.d.).

Esta clase es heredada por 2 subclases que también son secundarias:

- Clase *PressVector*
- Clase *ProdVector*

Las validaciones que se hacen en estas tres clases radican en fechas y valores que se encuentran dentro de la información de producción y presión. Los datos procesados sirven para futuros cálculos en las demás clases que lo necesiten. Su algoritmo se encuentra en la parte de Anexos como anexo 1.

2.2.8 Clase *PVTSchema*

Al igual que la clase *VectorData* y sus subclases, esta valida la información PVT definiendo un conjunto de reglas para su estructura y valores. Los datos que valida:

- Presión
- Factor volumétrico del petróleo
- Factor volumétrico del gas
- Relación gas-petróleo

Estos datos deben de ser coherentes para ser usados en posteriores cálculos que forman parte de la clase *OilModel* y a su vez ayudan a la clase *Aquifer*. Su algoritmo es similar al de *VectorData*

2.3 Desarrollo integral del código correspondiente a la librería

El desarrollo del código para la librería se divide en varias etapas, descritas a continuación en la siguiente ilustración 11:

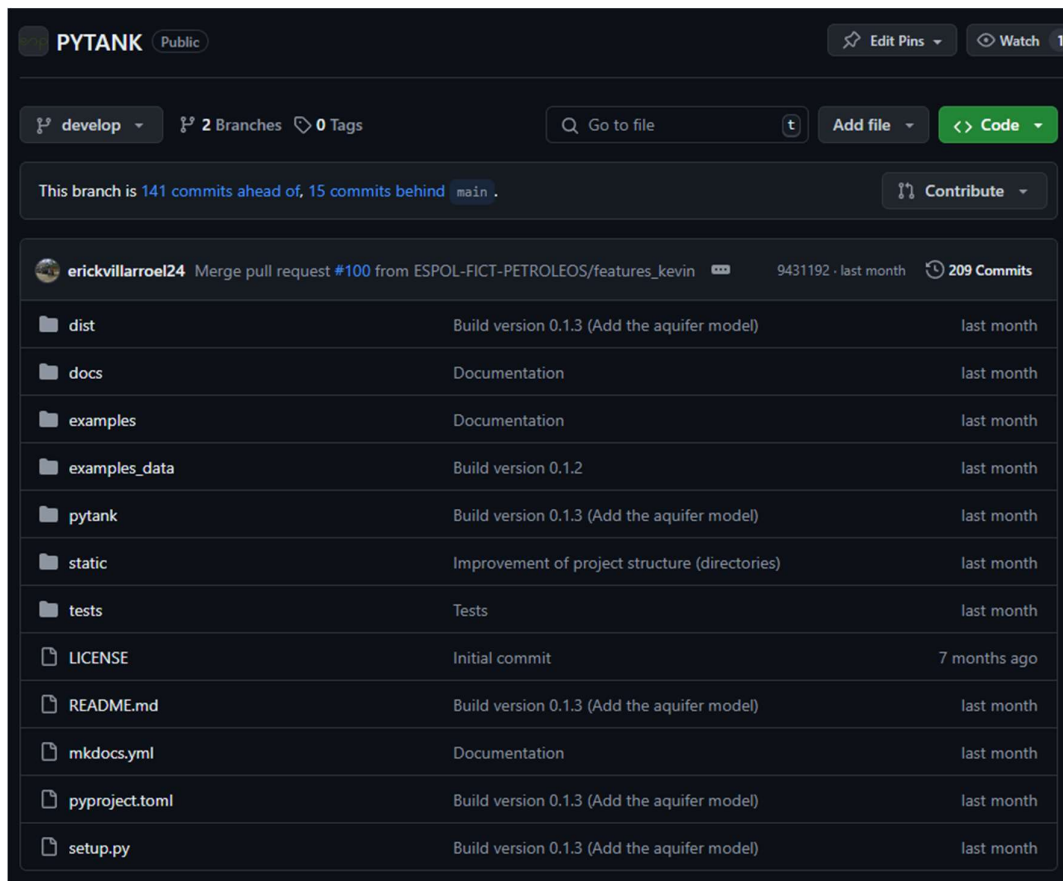
Ilustración 11. Flujo de trabajo para el desarrollo de la lógica de la librería



Fuente: Autores

- Se creó un repositorio en GitHub, donde se alojó todo lo relacionado al código fuente, estructura y datos de la librería desarrollada. De esta manera, se pudo almacenar y supervisar las tareas por los tutores del proyecto. El repositorio se lo denominó como PYTANK, nombre de la librería. En la Imagen 1., se aprecia el repositorio alojado en la plataforma online (GitHub). URL: <https://github.com/ESPOL-FICT-PETROLEOS/PYTANK>

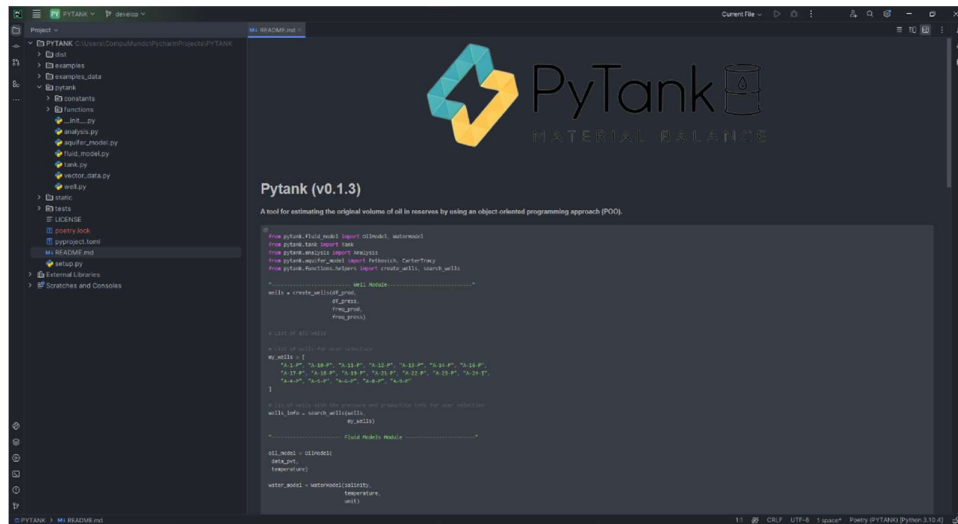
Imagen 1. Repositorio en la página online de GitHub



Fuente: Autores

- Para el desarrollo del código se necesitó de un entorno de desarrollo Integrado (IDE). El escogido fue PyCharm debido a su fácil instalación y a la ventaja de poder usar su versión profesional con una licencia gratuita estudiantil. La interfaz del IDE se puede ver a continuación en la Imagen 2.:

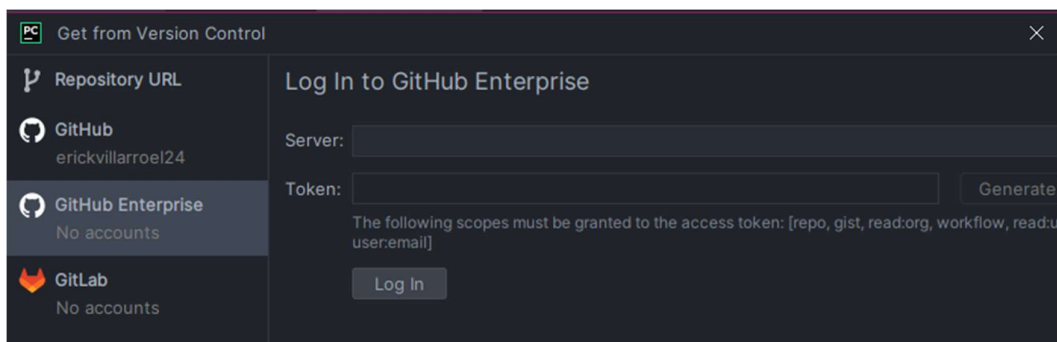
Imagen 2. Interfaz de PyCharm



Fuente: Autores

- Para poder trabajar en Pycharm con el repositorio creado en GitHub es necesario clonarlo en el IDE. Entonces, se debe copiar el URL del repositorio en la página online, y pegarlo en el entorno. Sin embargo, también se puede acceder al repositorio con un Token creado en la cuenta de GitHub, tal como se realizó en este proyecto. Se realizó este procedimiento, debido a que, generando un Token, se pueden obtener los permisos necesarios para los cambios correspondientes en el código. En la Imagen 3. se puede apreciar en qué sección del IDE se realizó este procedimiento.

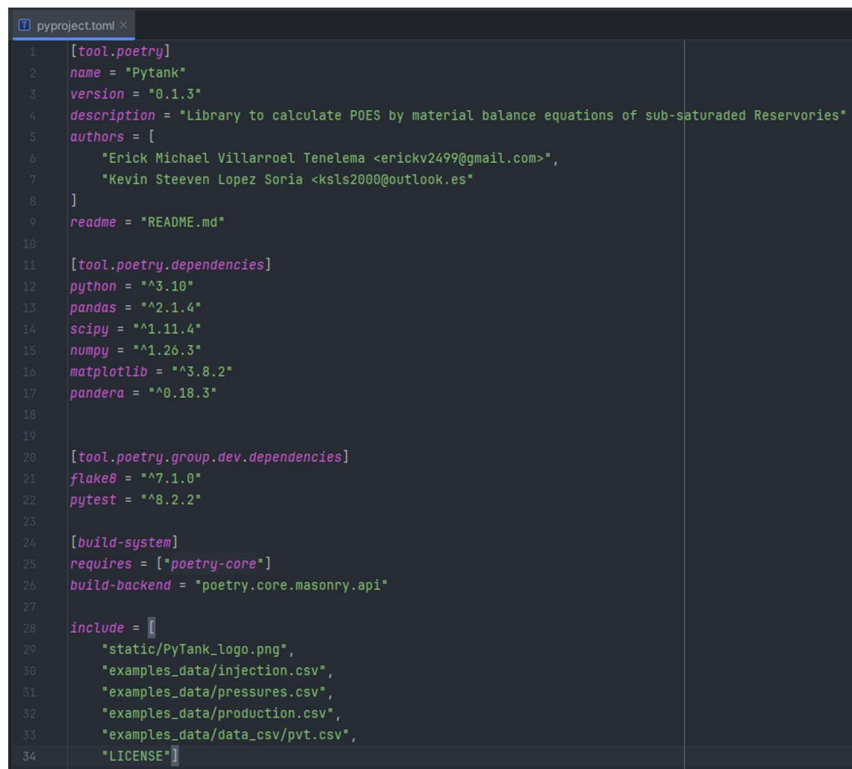
Imagen 3. Sección de Pycharm donde se clona el repositorio



Fuente: Autores

- Una vez clonado el proyecto en el IDE (PyCharm), se creó un ambiente virtual que permitió gestionar las dependencias necesarias para el desarrollo de la librería. Se decidió trabajar con Poetry, la cual es una herramienta que permite manejar las dependencias de forma independiente en cada proyecto. Además, cuenta con la ventaja de crear un archivo donde se puede encontrar la información de todas las librerías con sus versiones a medida que se trabaja. De esta manera, se facilita el manejo de la información para que futuros colaboradores o desarrolladores puedan actualizar el contenido de la librería. Esto se puede apreciar en la siguiente Imagen 4:

Imagen 4. Dependencias del ambiente virtual creado por Poetry en el archivo .toml



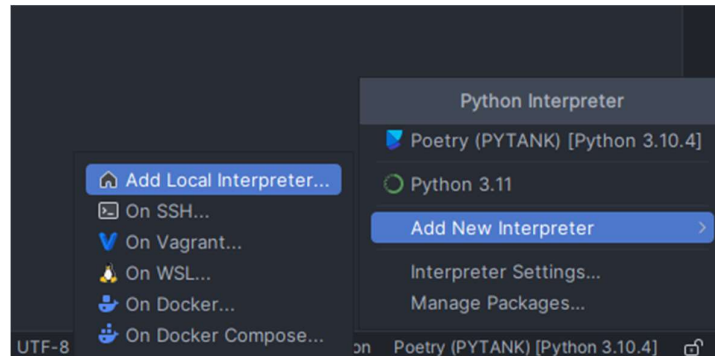
```
1 [tool.poetry]
2 name = "Pytank"
3 version = "0.1.3"
4 description = "Library to calculate POES by material balance equations of sub-saturated Reservoirs"
5 authors = [
6     "Erick Michael Villarroel Tenelema <erickv2499@gmail.com>",
7     "Kevin Steeven Lopez Soria <ksls2000@outlook.es>"
8 ]
9 readme = "README.md"
10
11 [tool.poetry.dependencies]
12 python = "^3.10"
13 pandas = "^2.1.4"
14 scipy = "^1.11.4"
15 numpy = "^1.26.3"
16 matplotlib = "^3.8.2"
17 pandera = "^0.18.3"
18
19 [tool.poetry.group.dev.dependencies]
20 flake8 = "^7.1.0"
21 pytest = "^8.2.2"
22
23 [build-system]
24 requires = ["poetry-core"]
25 build-backend = "poetry.core.masonry.api"
26
27 include = [
28     "static/PyTank_logo.png",
29     "examples_data/injection.csv",
30     "examples_data/pressures.csv",
31     "examples_data/production.csv",
32     "examples_data/data_csv/pvt.csv",
33     "LICENSE"
```

Fuente: Autores

- Se procedió a agregar como interprete al ambiente virtual creado con poetry. Para esto, se selecciona la parte inferior derecha en la interfaz de Pycharm, donde está

la versión de Python utilizada, en este caso, la versión 3.10.4. Una vez seleccionada, se despliega la siguiente opción que se aprecia en la Imagen 5.

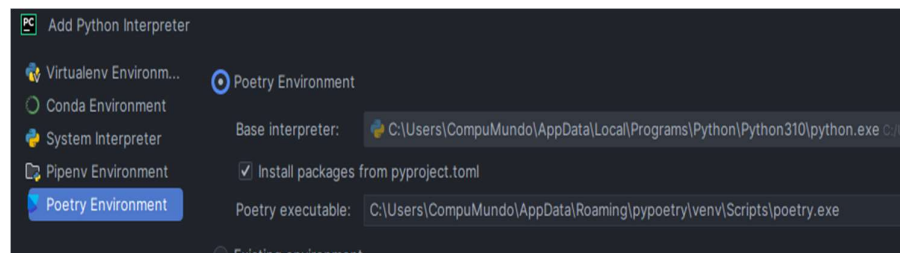
Imagen 5. Sección de Pycharm donde se agrega un interpretador



Fuente: Autores

- En esta se escoge “Add Local Interpreter” y luego en la opción de “Poetry Environment”, tal como se muestra a continuación en la Imagen 6.

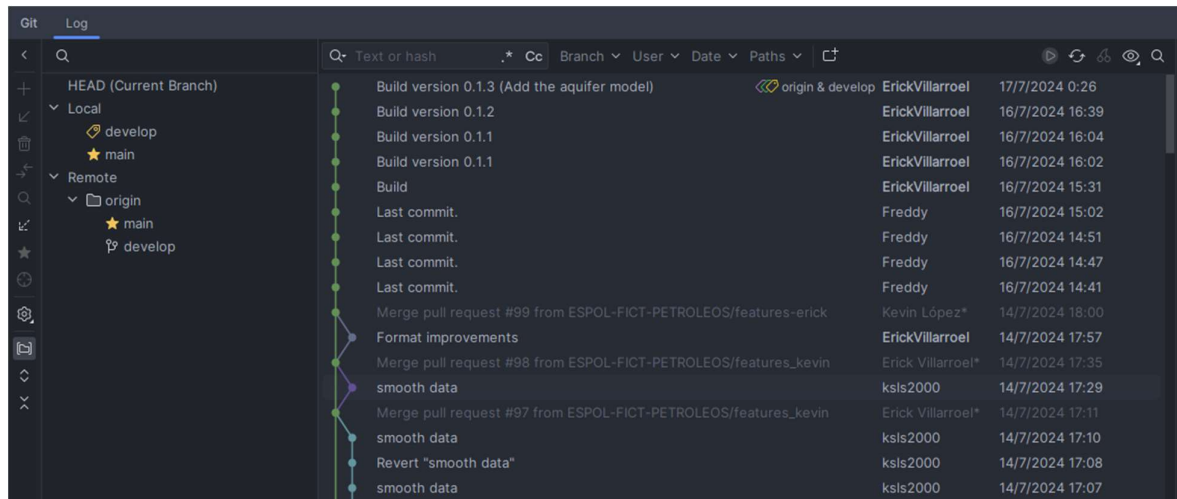
Imagen 6. Sección de Pycharm donde se agrega el ambiente virtual creado con Poetry



Fuente: Autores

- Es muy importante mencionar que el desarrollo de este código fue de manera conjunta. La herramienta utilizada para poder trabajar en colaboración fue GIT, un sistema de control de versionamiento (Lopez et al., 2015). Este permitió administrar cada cambio que se hizo en el código conforme su evolución. Tal como se puede ver en la siguiente Imagen 7.

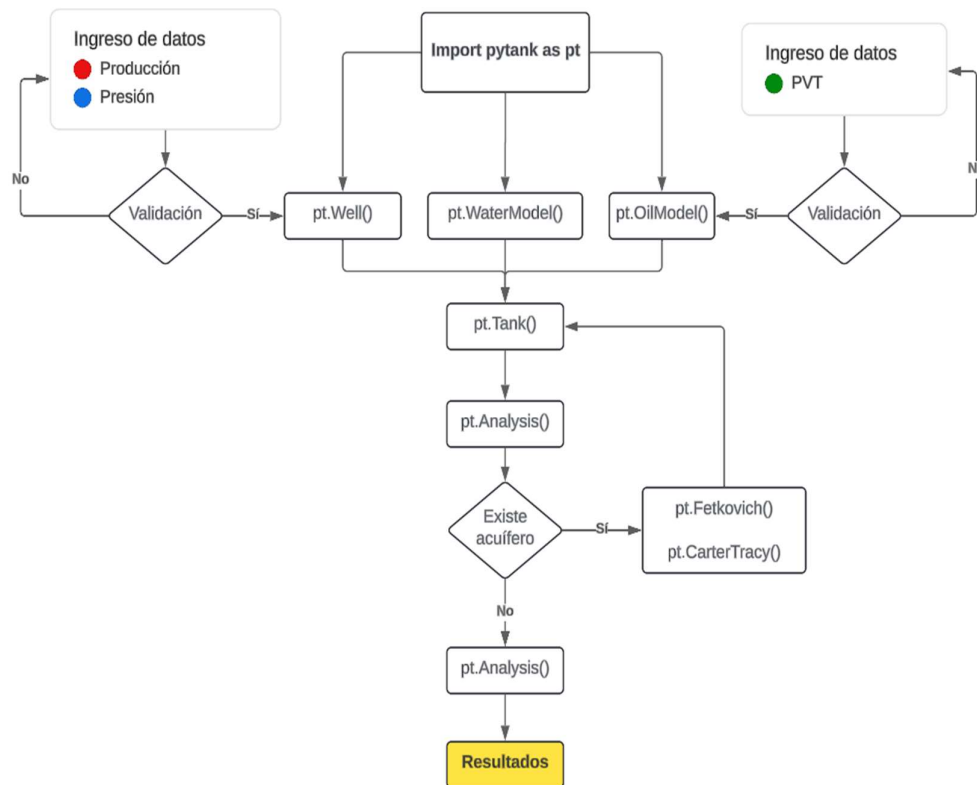
Imagen 7. Sección de GIT para el control de versiones del proyecto



Fuente: Autores

2.4 Estructura de la librería

Ilustración 12. Algoritmo de la librería



Fuente: Autores

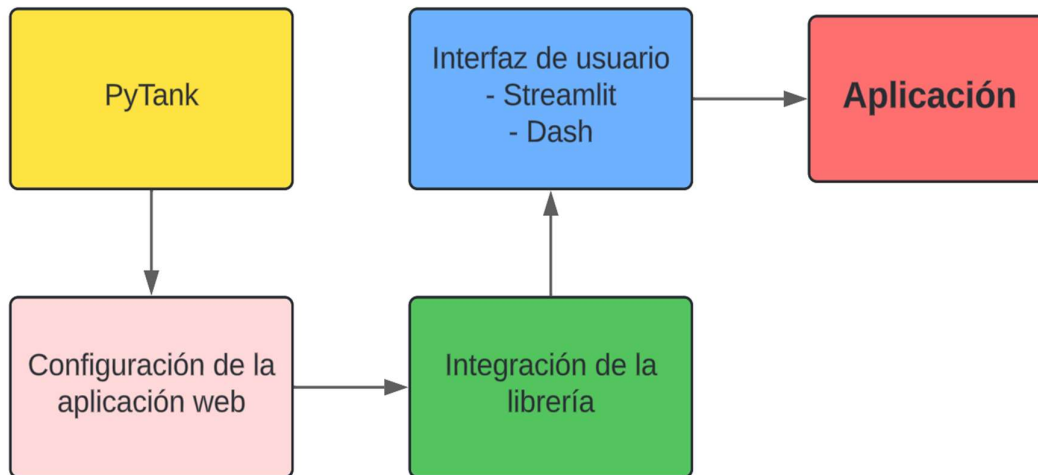
En la Ilustración 12 se observa cómo funciona la lógica del código de la librería. A continuación, se explica el flujo del algoritmo:

- a) Se importa el módulo principal que contiene a los demás.
- b) Se procesan y validan los datos de producción y presión por pozo. Esto es responsabilidad del usuario. Existen módulos de ayuda, de igual manera, como el módulo Well.
- c) Se deben procesar los datos de la información PVT del petróleo mediante el módulo OilModel.
- d) De la misma manera, para el agua mediante el módulo WaterModel.
- e) Se pasa la información procesada de producción y presión de los pozos con las propiedades del agua y petróleo al módulo Tank. En conjunto ese módulo necesita propiedades correspondientes al tanque, como son la presión inicial, saturación inicial del agua, compresibilidad de la roca y del agua. Este módulo funciona como almacenador de propiedades.
- f) Con el tanque y sus propiedades ya listas, se continúa al módulo Analysis que permite obtener el valor del $poes$ [MMSTB]. A Través de este módulo también se puede saber si existe una influencia de acuífero mediante el método Campbell de este módulo.
- g) Sabiendo que existe acuífero, para modelarlo se usa el módulo Fetkovich o CarterTracy que permite obtener el influjo acumulado de agua del tanque.
- h) La instancia del acuífero se ingresa en el módulo Tank para otro análisis.
- i) Finalmente, se vuelve a obtener el $poes$ con los nuevos parámetros y se lo constata con el método analítico a través del módulo Analysis.

2.5 Desarrollo integral del aplicativo web

Si bien el uso de la librería es sencillo, la lógica detrás de ella es muy robusta y técnica. Además, en lo que compete a gráficos, estos se realizaron con la librería matplotlib que genera gráficos estáticos. Es por eso por lo que, para hacer que el usuario tenga una mejor experiencia en lo que compete a uso y resultados, se realizó una aplicación web. Esta interfaz gráfica permite que el usuario tenga una mejor interacción con la lógica de la librería.

Ilustración 13. Flujo de trabajo para la creación de la aplicación web



Fuente: Autores

La ilustración 12 muestra las etapas que se tienen para crear la aplicación. A continuación, una explicación de lo que compete a cada parte:

- **Pytank:** Funciones y lógica de la librería
- **Configuración de la aplicación web:** Esta parte corresponde a la organización e imagen de la estructura del proyecto.
- **Integración de la librería:** Importa y utiliza las funciones y lógica de Pytank en la aplicación web.
- **Interfaz de usuario:** Mediante un framework (dash o Streamlit) se crea el Frontend o interfaz de la aplicación que será vista por el usuario.

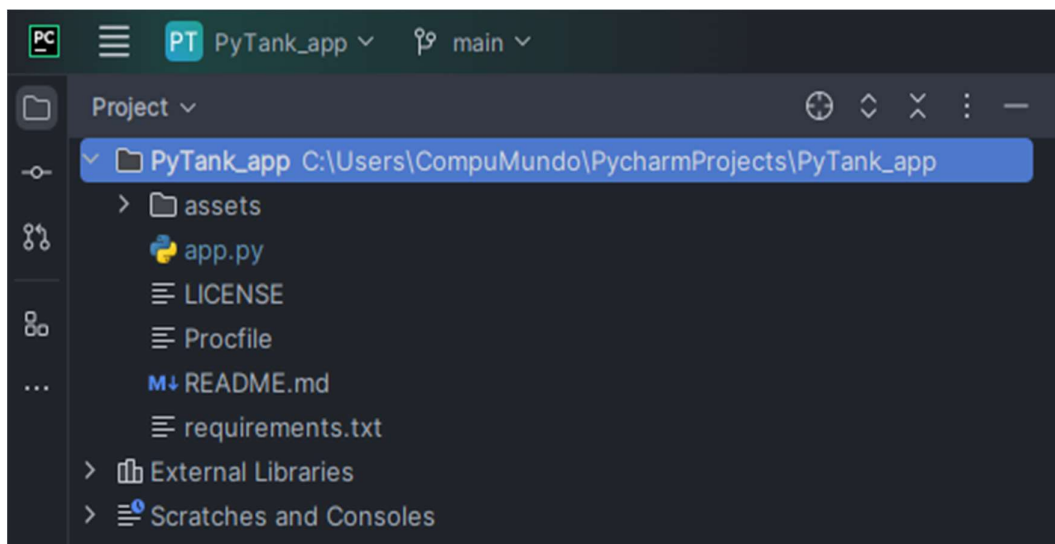
- **Aplicación:** Finalmente, una vez se tenga todo listo, se ejecuta la aplicación.

Para este desarrollo, se creó un repositorio aparte del repositorio principal de la librería. También se alojó en GitHub para su desarrollo conjunto.

URL: https://github.com/erickvillarroel24/PyTank_app

El IDE utilizado nuevamente es Pycharm. Los directorios del repositorio se pueden apreciar en la Imagen 8.

Imagen 8. Directorios del repositorio para la aplicación web en Pycharm



Fuente: Autores

Finalmente, la aplicación fue desarrollada con Dash, un framework que permite realizar aplicativos webs con mayor libertad de modificaciones, con respecto a otros framework como Streamlit.

La aplicación fue desarrollada con HTML, CSS junto con la lógica de la librería Pytank. Su código completo cuenta con aproximadamente 2500 líneas y se presenta en una sola ventana que contiene cinco módulos: Well, Tank, Fluid Models y Analysis. Estos nombres con la finalidad de no perder la esencia propia de la librería Pytank. A continuación, en la Imagen 9 se presenta como quedó el diseño de la aplicación, mostrando como ejemplo el módulo primer módulo, Well.

Imagen 9. Interfaz gráfica de Pytank View

The image shows the Pytank View Well Module interface. It features a blue header with the text 'Pytank View' and a logo. Below the header is a yellow bar with the text 'Well Module'. The main content area is divided into two sections: 'Configuration' on the left and 'Results Area' on the right. The 'Configuration' section includes two file upload sections: 'Upload Production CSV' and 'Upload Pressure CSV'. Each section has a 'Select Files' button and a green button labeled 'Upload production data to start.' Below these are two dropdown menus for 'Production frequency' and 'Pressure frequency'. There is also a text input field labeled 'Enter Well Names Below' and two buttons: 'Add Well' (green) and 'Remove Well' (red). At the bottom of the configuration section is an orange 'Submit' button. The 'Results Area' has a grey header and a text description: 'This section shows the production and pressure information for each well according to time. Make sure you enter the correct parameters.' A blue double-arrow button is located in the bottom right corner of the interface.

Fuente: Autores

2.6 Desarrollo integral de la documentación

A pesar de que el uso de la librería es accesible, su diseño y funcionalidad son complejos al estar fundamentado en principios técnicos sólidos. Para garantizar que los usuarios puedan aprovechar al máximo la librería y los diferentes métodos que posee, es esencial el desarrollo de una documentación integral que explique en detalle cómo utilizar la librería y entender su lógica interna. Esta documentación no solo servirá como guía, además también mejorará la experiencia del usuario al facilitar la comprensión de los métodos y funciones disponibles. El trabajo fue desarrollado en el siguiente enlace:

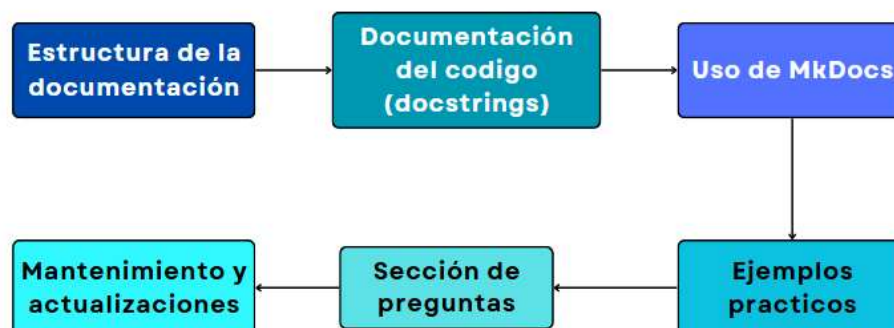
Prerrequisitos:

- **Entorno Virtual (venv):** Se debe crear y configurar un entorno virtual que permita instalar y utilizar los diferentes paquetes según se requiera. En este caso, se utiliza Poetry por su facilidad para gestionar dependencias y su compatibilidad con diversas versiones de Python. La creación de un entorno virtual asegura que las dependencias del proyecto estén aisladas de otros proyectos y del sistema operativo,

evitando conflictos y garantizando un entorno controlado para el desarrollo y la ejecución de la librería.

- **Documentación del código (docstrings):** Para el desarrollo de la documentación en línea de la librería, es indispensable contar con una descripción detallada de las funciones y métodos desarrollados. Esto incluye especificar los parámetros de entrada, los valores de retorno y ejemplos de uso. Por ello, es necesario asegurarse de que las funciones internas de la librería incluyan información clara sobre su uso y funcionamiento. La implementación de docstrings coherentes y bien estructurados no solo facilita la comprensión del código, sino que también permite la generación automática de documentación utilizando herramientas como Sphinx o MkDocs.
- **Repositorio de GitHub:** Se debe crear un repositorio en el que se almacene la estructura del código correspondiente al desarrollo de la librería. GitHub no solo proporciona un espacio para el almacenamiento y la gestión del código, sino que también facilita la colaboración entre desarrolladores, permitiendo el seguimiento de cambios, la revisión de código y la gestión de problemas (issues).

Ilustración 14. Flujo de trabajo para el desarrollo de la documentación



Fuente: Autores

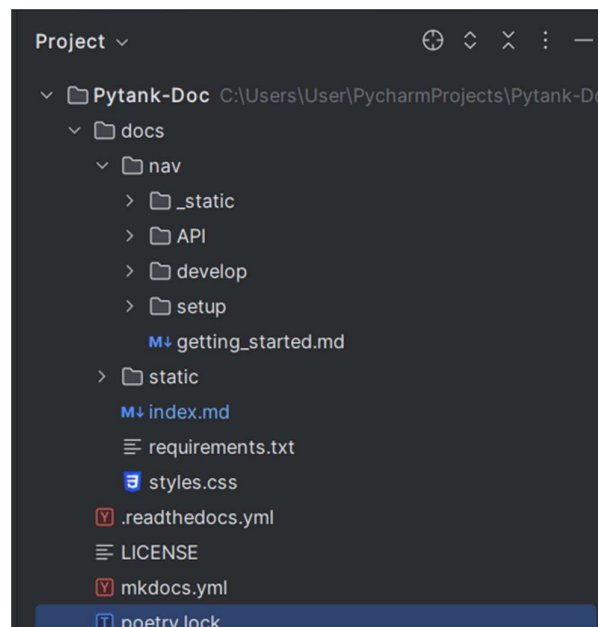
La Ilustración 13 muestra las etapas involucradas en la creación de la documentación de la librería. A continuación, se detalla cada componente del proceso:

- **Estructura de la documentación:** Se establece un esquema claro que organiza la información de manera lógica, facilitando la navegación y el acceso a los contenidos relevantes que comprende la librería.
- **Documentación del código (docstrings):** Cada función y método de la librería se documenta con descripciones precisas que explican su propósito, parámetros de entrada y valores de retorno, lo que ayuda a los usuarios a comprender el uso correcto.
- **Uso de MkDocs:** Se implementa MkDocs para generar un sitio web estático que presenta la documentación de forma atractiva y accesible. Esto incluye la instalación y configuración inicial del paquete, así como la creación de la estructura de archivos necesaria.
- **Ejemplos prácticos:** Se incluyen ejemplos de uso que ilustran como aplicar las funciones de la librería en situaciones reales, lo que permite a los usuarios ver la librería en acción y comprender su aplicación práctica.
- **Sección de preguntas frecuentes (FAQ):** Se agrega una sección que responde a las dudas comunes que pueden tener los usuarios, lo que contribuye a resolver problemas y mejorar la experiencia general.
- **Mantenimiento y actualizaciones:** Se establece un protocolo para mantener la documentación actualizada con cada nueva versión de la librería, asegurando que siempre esté alineada con las ultimas funcionalidades y mejoras.

Al finalizar este proceso, se obtiene un sitio web estático que presenta la documentación que corresponde a la librería Pytank. Se utilizo un formato basado en Markdown y está organizada en diferentes secciones como se muestra en la Imagen 10. las cuales abordan diferentes aspectos de la librería. A continuación, se presenta un resumen de su estructura:

- **Home:** La página principal que da la bienvenida a los usuarios y proporciona una visión general de la librería.
- **Getting Started:** Esta sección incluye las guías iniciales que ayuden a los usuarios a configurar la librería y comenzar a utilizarla rápidamente.
- **User Guide:** Posee diferentes módulos que le muestran al usuario el uso y la creación de los diferentes “objetos” y el uso de las funciones para llevar a cabo un análisis correcto de los datos y obtener resultados, además se presentan diferentes ejemplos prácticos utilizando data real.
- **Development:** Esta sección está destinada a los desarrolladores, en donde se presentan puntos desde una instalación avanzada de la librería, y se presentan las notas de las versiones de la documentan los cambios y mejoras realizadas en la librería.
- **API Reference:** Ofrece a los usuarios detalles de las diferentes funciones y módulos que comprenden el funcionamiento de la librería.

Imagen 10. Esquema general de los directorios que comprenden la documentación en línea



Fuente: Autores

Capítulo 3: Resultados y Análisis

3. Pytank

3.1 Uso de la librería

El primer producto final de este proyecto, cumpliendo uno de los objetivos específicos es alojar la librería en un manejador de paquetes, como otras librerías reconocidas y bien documentadas. En el siguiente url se puede apreciar que se cumplió el objetivo y Pytank ha sido alojada en PyPi.

Url: <https://pypi.org/project/Pytank/>

A continuación, se describe paso a paso el proceso que el usuario debe realizar para que pueda darle uso al producto terminado, la librería, en un IDE, ejemplo PyCharm.

El proceso inicia con la importación de las librerías necesarias. En este caso Pytank y Pandas. Primero, se necesita trabajar con datos históricos de producción y presión. En este caso, los datos son provenientes del campo ESPOL ubicado en la cuenca Oriente. Cabe recalcar que el usuario debe utilizar los datos de su interés. Puesto que, este caso es solo de ejemplo para la validación del proyecto. Todos los datos se obtendrán de este campo. El procesamiento de estos datos por pozo es responsabilidad del usuario, debido a que los usuarios definirán con qué pozos específicos y que ordenamiento requieren realizar para el balance de materiales. Sin embargo, la librería tiene funciones de ayuda *helpers* que le permiten al usuario tener al alcance una herramienta rápida para el procesamiento de datos de todos los pozos. Esto lo hace *create_wells* que almacena en una lista y procesa la información de producción y de presión de todos los pozos que el usuario tenga en su información.

El balance de materiales se realiza para un tanque en específico. Para esto, existe otra función, de nombre *search_wells* que permite obtener la información de los pozos con los cuales el usuario requiera realizar el balance de materiales (destinados al tanque) en la lista de todos sus pozos creados con *create_wells*: datos significativos, para establecer relaciones y resumir el material de los hallazgos.

Sobre la publicación de los datos, se debe de tener cuidado al publicarlos en el documento ya que estos pueden ser personales, sensibles, en estos casos, se recomienda que sean datos agregados, es decir, un conjunto de información recopilada con base a promedios o resumen de un grupo de datos individuales, también se recomienda utilizar técnicas para anonimizar datos con el fin de que se salvaguarde datos de identidad o información sensible.

```
import pytank as pt
import pandas as pd

# Transform the csv to DataFrames
df_pvt = pd.read_csv(
    "C:/Users/User/PycharmProjects/PYTANK/pytank/resources/data_csv/"
    "pvt.csv"
)
df_production = pd.read_csv(
    "C:/Users/User/PycharmProjects/PYTANK/pytank/resources/data_csv/"
    "production.csv"
)
df_pressures = pd.read_csv(
    "C:/Users/User/PycharmProjects/PYTANK/pytank/resources/data_csv/"
    "pressures.csv"
)

wells = pt.create_wells(df_prod=df_production,
                        df_press=df_pressures,
                        freq_prod="MS",
                        freq_press=None)

# List of wells for user selection
my_wells = [
    "A-1-P", "A-10-P", "A-11-P", "A-12-P", "A-13-P", "A-14-P", "A-16-P",
```

```

    "A-17-P", "A-18-P", "A-19-P", "A-21-P", "A-22-P", "A-23-P", "A-24-I",
    "A-4-P", "A-5-P", "A-6-P", "A-8-P", "A-9-P"
]
# list of wells with the pressure and production info for user selection
wells_info = pt.search_wells(wells, my_wells)

```

Sabiendo que un pozo tiene su respectiva información de producción y de presión, también es indispensable obtener información que describa el comportamiento de los fluidos que forman parte del reservorio. Por esto, se desarrollaron los módulos “OilModel” y “WaterModel”. La diferencia de un módulo con otro radica en que el módulo “OilModel” necesita información PVT que ya ha sido validada (la validación no la realiza la librería) para obtener el valor de las propiedades físicas de hidrocarburos (Bo, Bg, Rs), mientras que el módulo “WaterModel” obtiene las propiedades para el agua en base al uso de la correlación de McCain para su factor volumétrico y solubilidad.

```

oil_model = pt.OilModel(
    data_pvt=df_pvt,
    temperature=25,
)
water_model = pt.WaterModel(
    salinity=3000,
    temperature=200,
    unit=1
)

```

Continuando con el proceso de la implementación de la librería, se despliega el módulo “Tank”. Este módulo requiere principalmente de las propiedades del tanque (reservorio) y se le debe designar un nombre.

Una vez se tiene todas las propiedades del tanque, se procede a realizar el primer análisis de este con el balance de materiales, usando el módulo Analysis de la siguiente manera:

```
frequency = "12M"
analysis = pt.Analysis(tank_class=tank1,
                      freq=frequency,
                      position="end"
                      smooth=False)
```

Cabe recalcar que este módulo realiza el análisis con una frecuencia elegida por el usuario:

- Anual (12M o Y)
- Mensual (M)
- Trimestral (3M)
- Semestral (6M)

De la frecuencia depende la cantidad de datos con el que se hará el análisis. Además, este módulo puede ajustar los datos de presión promedio para obtener un mejor comportamiento de estos. Esto se lo puede hacer a partir de las propiedades k (grado de ajuste polinomial) y s (smooth de ajuste). Se lo hace solo si se requiere, ya que en primera instancia se tiene que obtener la data sin ajuste.

Ya teniendo instanciado el módulo Analysis, se procede a usar sus métodos: *campbell_plot()* y *plot_press_avg_time()*. El primer método genera un gráfico de Campbell donde se puede constatar si el tanque tiene un aporte energético por parte de un acuífero, el segundo permite crear un gráfico de presión promedio vs tiempo para visualizar el comportamiento de la presión.

```
camp = analysis.campbell_plot()
camp.show()

plt = pt.analysis.plot_press_avg_time()
plt.show()
```

Como se ha podido observar durante la presentación de este código, existe la variable *tank1* que hace referencia a una instancia de Tank que no posee un acuífero. El objetivo de la librería es proveer una herramienta que permita estimar las reservas para yacimientos subsaturados volumétricos y no volumétricos. Por ello, también es necesario el desarrollo de una clase que ayude a determinar el aporte de un acuífero en caso de que exista la influencia de éste sobre el reservorio revisado con el método *campbell_plot()*.

Los módulos “*Fetkovich*” y “*CarterTracy*” son los encargados de estimar dicho volumen que ingrese al reservorio en caso de existir. Para ello, se requiere de ciertos parámetros del acuífero, y que el usuario debe tener en conocimiento para poder implementarla. El modelo de acuífero que se escoja corresponde a los nombres de las clases, dependiendo del requerimiento del usuario y de los datos que tenga disponibles. La forma correcta para utilizar el módulo viene dada de la siguiente manera:

```
aq_radius = 14000
res_radius = 2000
aq_thickness = 20
phi = 0.25
ct = 0.000007
theta = 290
k = 25
water_visc = 0.6

fet = Fetkovich(
    aq_radius=aq_radius,
    res_radius=res_radius,
    aq_thickness=aq_thickness,
    aq_por=phi,
    ct=ct,
```

```

        theta=theta,

        k=k,

        water_visc=water_visc,
    )

```

Si existe un aporte energético de un acuífero, la instancia de Tank quedaría de la siguiente manera:

```

tank_fet = pt.Tank(name=tank_name,

                    wells=wells_info,

                    oil_model=oil_model,

                    water_model=water_model,

                    pi=pi,

                    swo=0.25,

                    cw=3.5e-6,

                    cf=4.6e-6,

                    aquifer=fet)

```

A continuación, se realiza un nuevo análisis, en este caso, con un acuífero, para usar los demás métodos del módulo Analysis de manera correcta. Se presenta su ejemplo:

```

analysis_fet = pt.Analysis(tank_class=tank_fet, freq=frequency,
position="end")

"Analytic method"
analitic_meth_fet = analysis_fet.analytic_method(poes=67e+6, option="plot")
analitic_meth_fet.show()

"Havlena Method"
havlena_fet = analysis_fet.havlena_odeh_plot()
havlena_fet.show()

```

Como se puede apreciar en el script, esta parte sirve para calcular el POES [MMSTB] de manera gráfica con los métodos de *havlena_odeh_plot()* y de manera analítica con el método de *analytic_method()* de un reservorio subsaturado con aporte energético de un acuífero.

Por último, existen métodos del módulo Analysis que el usuario puede usar para ver el comportamiento del tanque de manera gráfica en relación con su producción y presiones. Por ejemplo, se puede graficar el aporte de caudal que cada pozo otorga al tanque o el comportamiento de su presión observada a través del tiempo de la siguiente manera:

```
plot1 = analysis_carter.plot_flow_rate_tank()  
plot1.show()
```

3.2 Pytank View

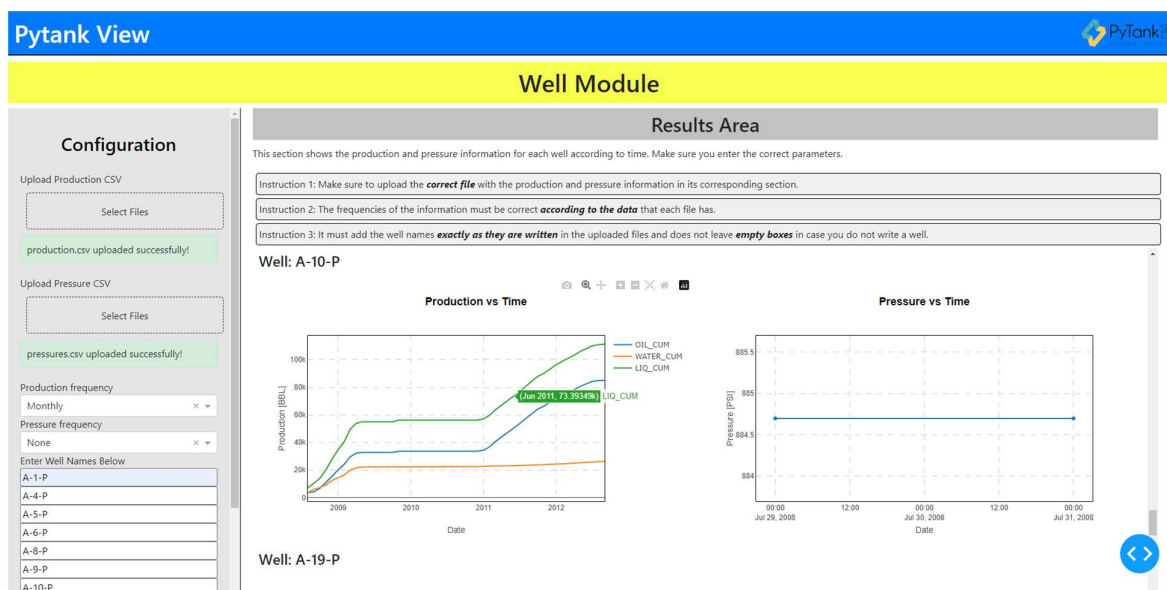
En esta sección se presentará el segundo producto de este proyecto, el cual consiste en una aplicación web que trabaja con la lógica de la librería. Esta herramienta se realizó para que un usuario que no maneje Python pueda usar la lógica de la librería mediante una interfaz gráfica que ofrezca todas las opciones que ofrece la librería para llevar a cabo un análisis.

La aplicación fue realizada con el framework Dash, una librería para crear aplicaciones web, usando código HTML, CSS y Python (también pueden utilizar otros lenguajes de programación). Es importante recalcar que los resultados que se obtengan mediante esta aplicación también se pueden obtener escribiendo código, usando la librería Pytank, tal como se mostró en la sección anterior. La principal diferencia radica en que la aplicación genera resultados de graficas dinámicas gracias al uso de la librería plotly, mientras que con el uso normal de la librería se pueden obtener gráficos estáticos hechos con la librería matplotlib. Para evidenciar la rigurosidad de los resultados de la aplicación y de la librería, las gráficas presentadas a continuación serán mostradas con la aplicación y con la librería para el primer caso.

Aclarando estas se procede con los resultados. Primero, se presenta el módulo de nombre Well, en donde se carga la información histórica de producción y presión necesarias con el fin de

mostrar por pozo (ingresados por el usuario para el balance de materiales) dicha información histórica gráficamente, tal como se muestra en la siguiente Imagen 11:

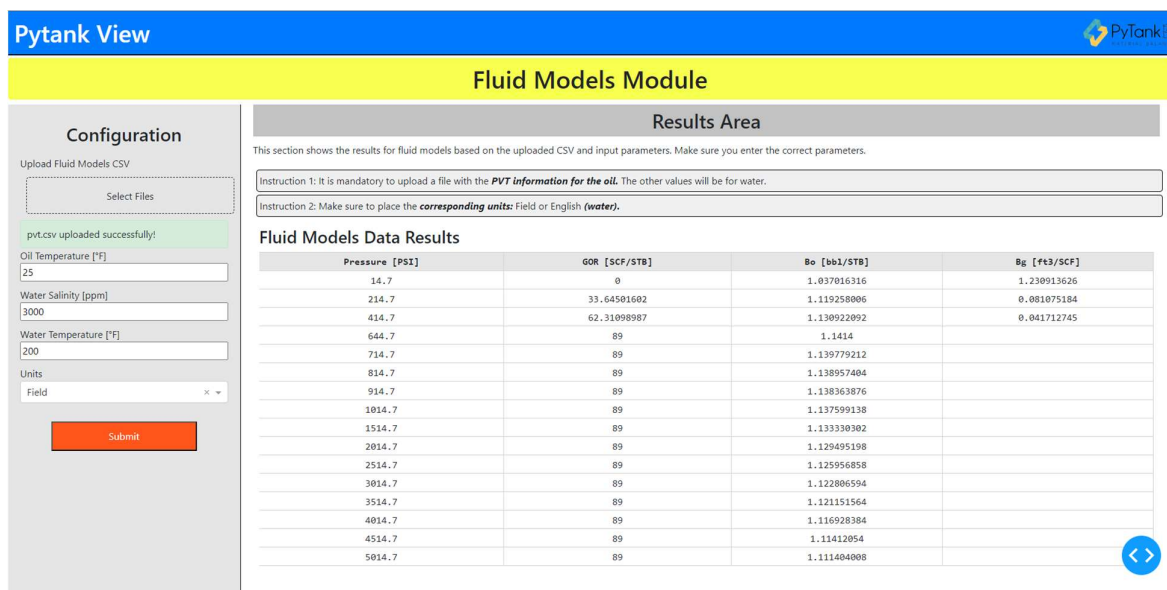
Imagen 11. Interfaz gráfica de Pytank View – Módulo Well



Fuente: Autores

Se debe ingresar también información para modelar las propiedades de petróleo y agua. Esto se lo hace en el módulo de Fluid Models. En la siguiente imagen se observa un ejemplo:

Imagen 12. Interfaz gráfica de Pytank View – Módulo Fluid Models



Fuente: Autores

Después de esta parte, el enfoque se centra en los 2 últimos módulos que permiten llevar a cabo el análisis para el balance de materiales de distintas maneras o análisis, dependiendo de lo que el usuario necesite. Desde este punto ya se presentan los resultados para el cálculo de reservas de petróleo.

Es importante mencionar que, se dividirán a estos resultados en diferentes casos, dependiendo de los análisis que se pueden llevar a cabo.

3.2.1 Caso 1: Yacimiento volumétrico sin acuífero

El primer caso corresponde a un yacimiento subsaturado volumétrico, sin acuífero. En la Imagen 13, se puede observar el módulo Tank de la aplicación con su respectiva configuración. Este módulo en sus resultados muestra las propiedades del tanque con el que se hará el balance de materiales.

Imagen 13. Interfaz gráfica de Pytank View – Módulo Tank

Pytank View

Tank Module

Configuration

Tank name: ESPOL

Initial Pressure: 3700

Initial water saturation: 0.25

Water compressibility: 0.000035

Formation compressibility: 0.000046

Aquifer Model: None

Submit

Results Area

This section displays the tank configuration with specific properties selected. Make sure you put the correct properties

Instruction 1: Be sure to place the values in the **specific units** in each section.

Instruction 2: First work assuming that there is **no aquifer**. If an aquifer exists, make sure to place the appropriate model with its correct properties in the specific units.

Instruction 3: **Check** in the results section if the values entered are correct.

ESPOL

Initial Pressure: 3700

Initial Water Saturation: 0.25

Water Compressibility: 3.5e-06 1/PSI

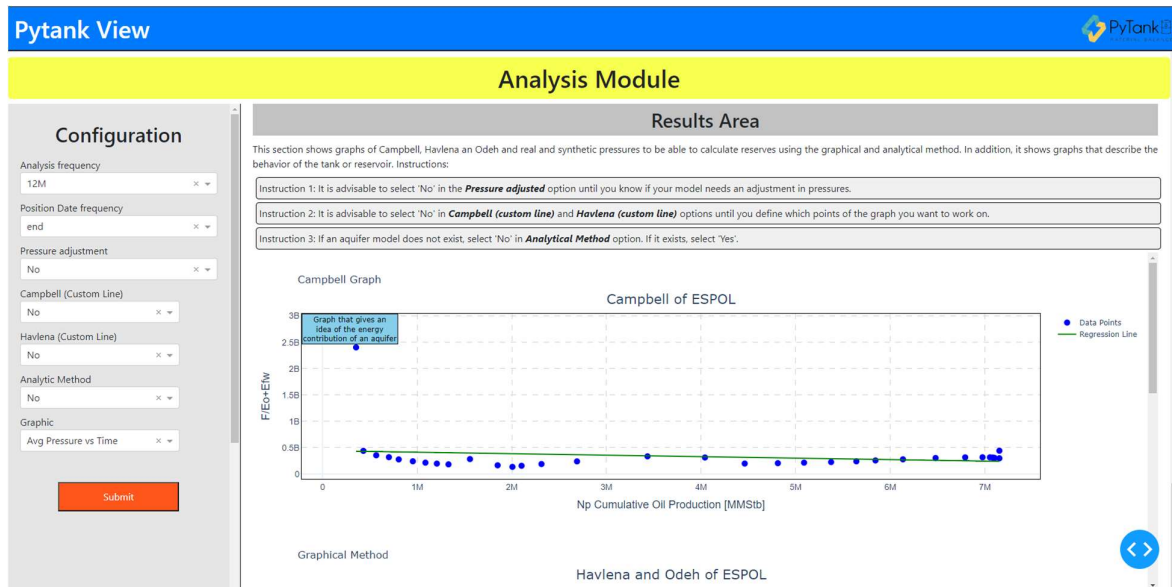
Formation Compressibility: 4.6e-06 1/PSI

Aquifer Model: Without aquifer model

Fuente: Autores

Posteriormente, en el módulo de nombre analysis se procede a colocar los parámetros con los que se requiere realizar el balance de materiales sin acuífero, tal como se puede apreciar en la Imagen 14 obteniendo como resultado los gráficos descritos a continuación:

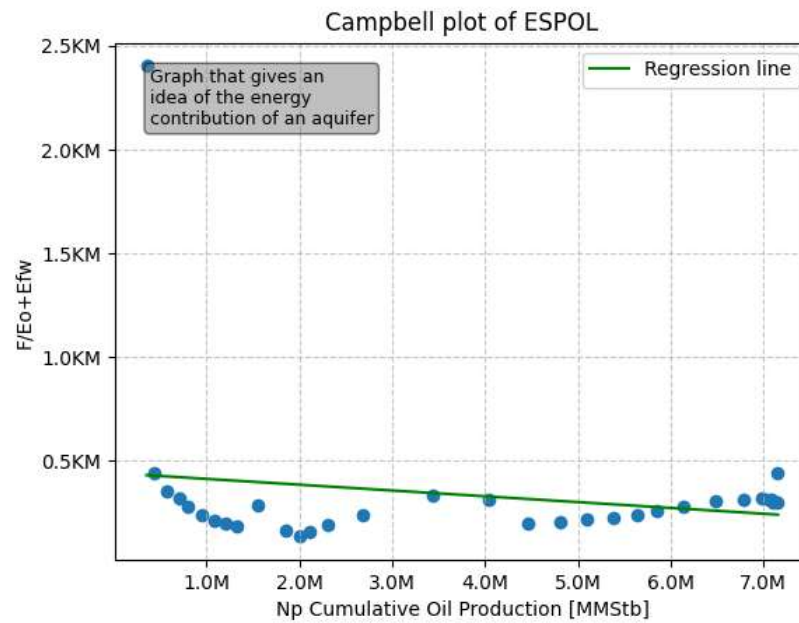
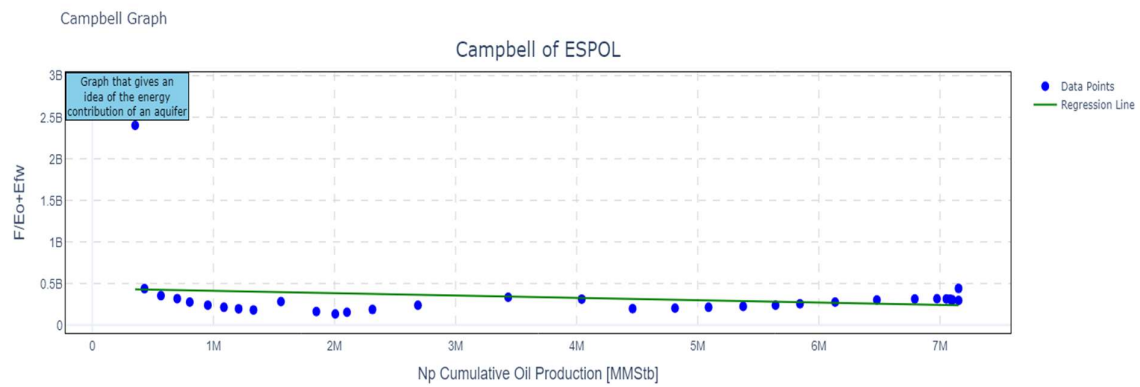
Imagen 14. Interfaz del modulo Analysis de Pytank View



Fuente: Autores

En el gráfico 1. se observa por la tendencia de los puntos que existe la influencia de un acuífero según la data de producción registrada por en el “tanque”, por ende, se debería considerar el definir un modelo de acuífero que logre representar un comportamiento más realista del reservorio.

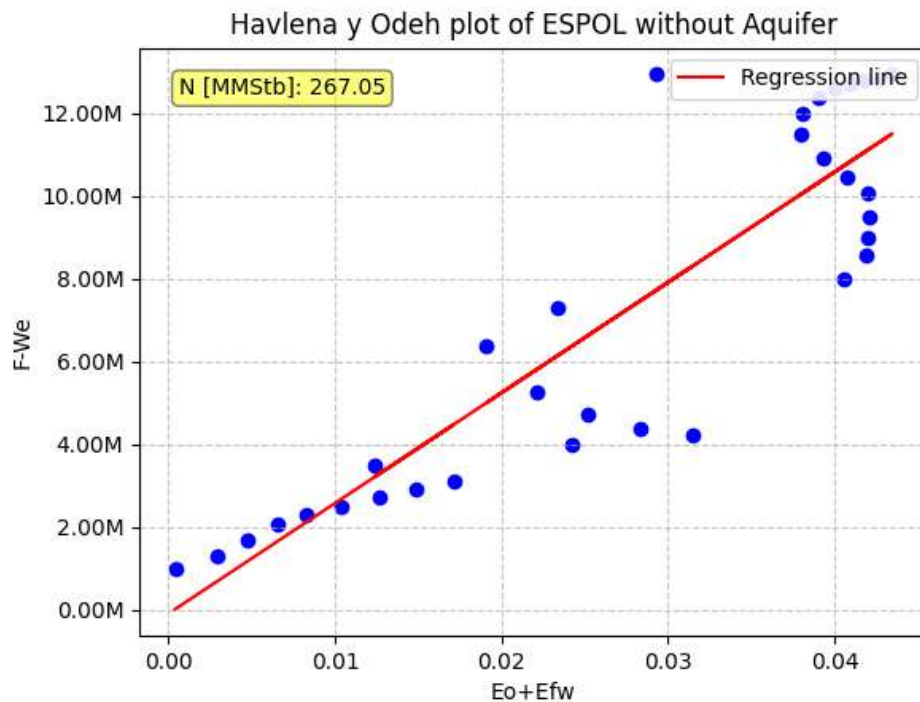
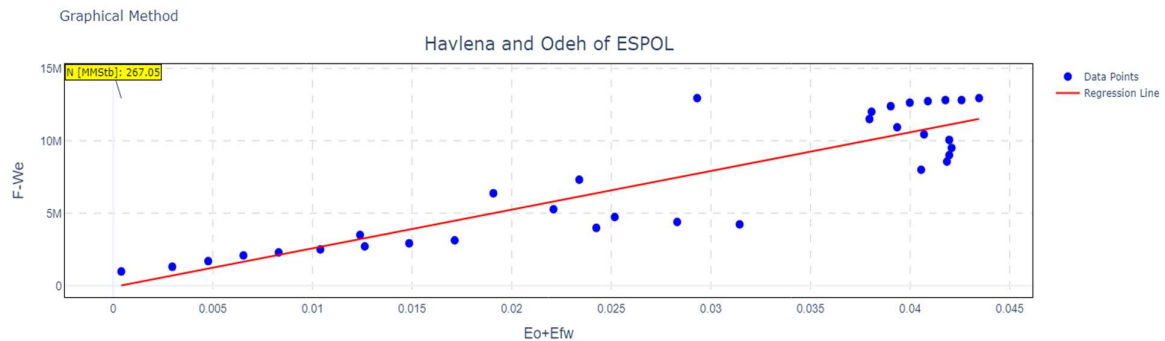
Gráfico 1. Gráfico de Campbell correspondiente al “Tanque” para el aplicativo web vs obtenido en la librería



Fuente: Autores

Otro gráfico que ofrece el módulo Analysis de la aplicación es el siguiente:

Gráfico 2. Grafica de Havlena y Odeh para el método grafico – Sin aporte de acuífero para el aplicativo web vs obtenido en la librería

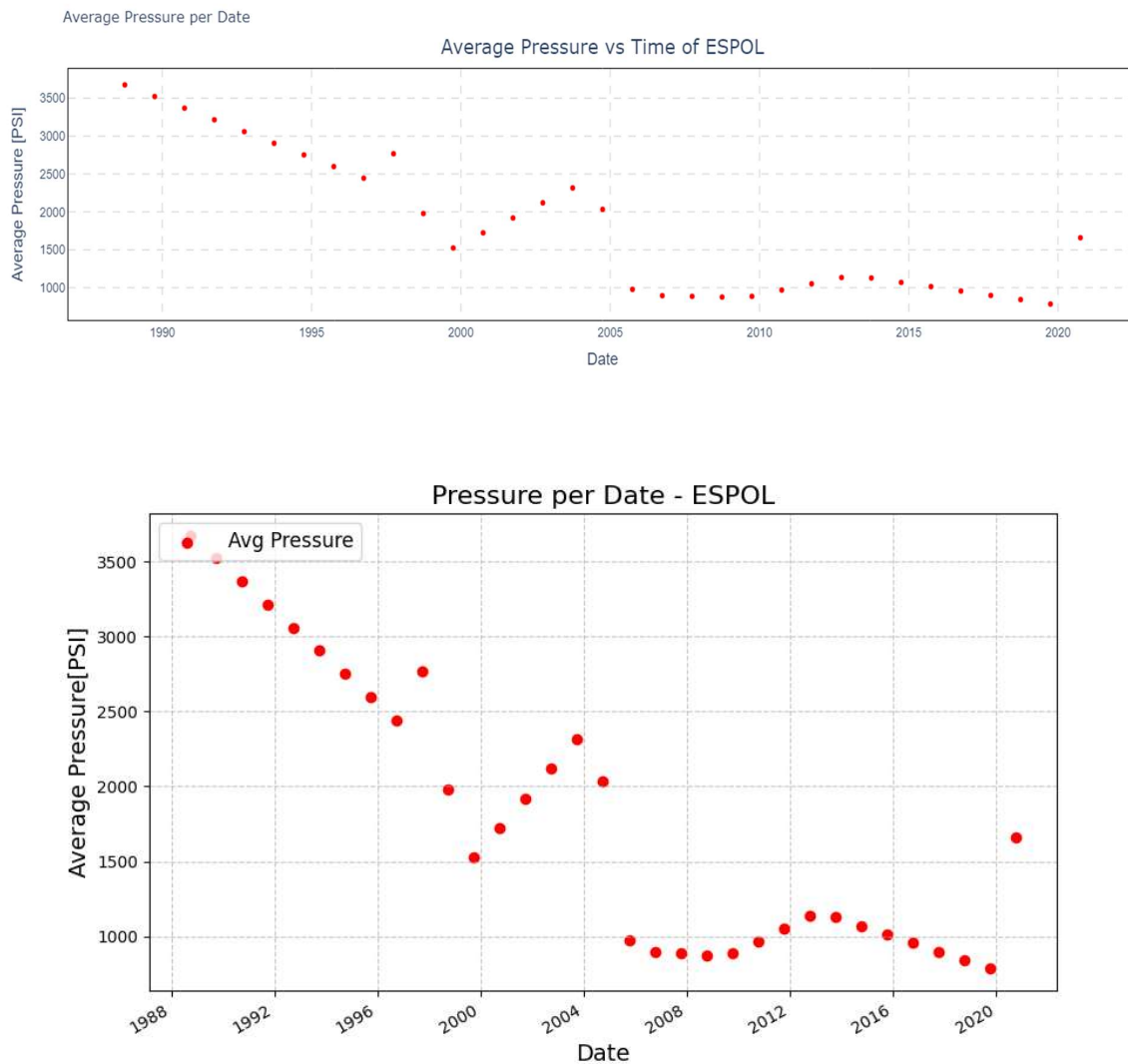


Fuente: Autores

Como se aprecia en el Gráfico 2. que corresponde al método gráfico de Havlena y Odeh, se observa que las reservas de petróleo son de 267.05 millones de barriles en superficie. Sin embargo, gracias a la gráfica de Campbell sabemos que este valor de reservas no es preciso debido a la influencia de un acuífero.

Además, de las dos opciones gráficas descritas anteriormente, existe una tercera opción gráfica que permite visualizar el comportamiento de las presiones a través de tiempo.

Gráfico 3. Graficas del comportamiento de la presión promedio acuífero para el aplicativo web vs obtenido en la librería



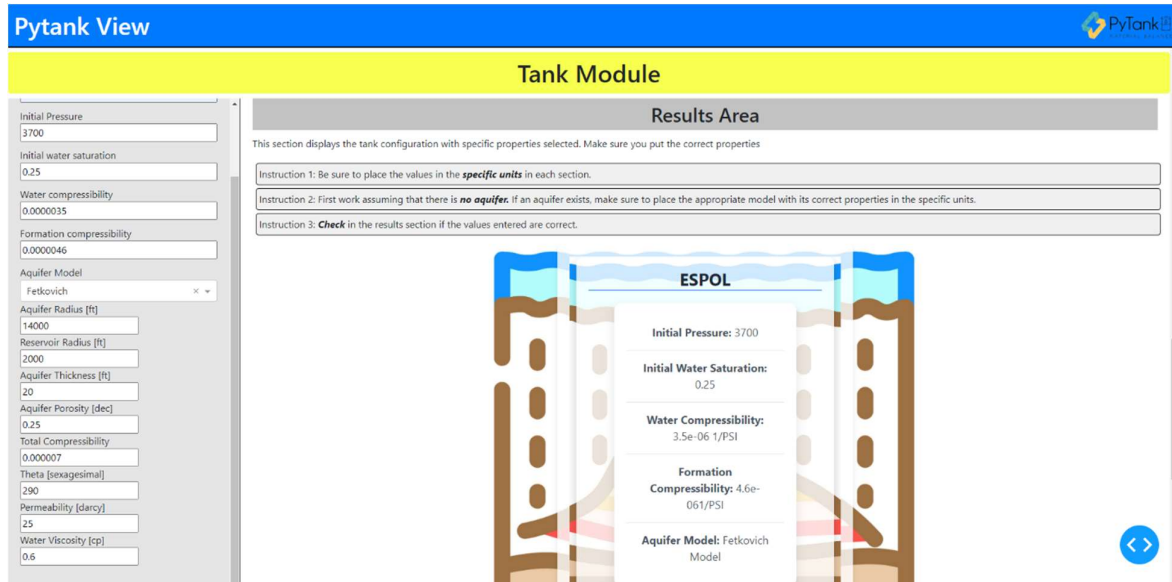
Fuente: Autores

En el Gráfico 3, se aprecia el comportamiento de la presión promedio en relación con el tiempo. Este es relevante porque sugiere si las presiones necesitan un ajuste o no. En este caso, el comportamiento tiene picos pocos pronunciados por lo que se podría deducir que necesita uno. Esto ya es criterio ingenieril por parte del usuario que realiza el análisis para el cálculo de reservas.

3.2.2 Caso 2: Yacimiento con Acuífero utilizando Fetkovich

A continuación, se llevará a cabo un análisis de un yacimiento subsaturado no volumétrico, con un acuífero. Este será modelado con Fetkovich. Entonces para esto modificamos el módulo Tank tal como se aprecia en la siguiente Imagen 15.

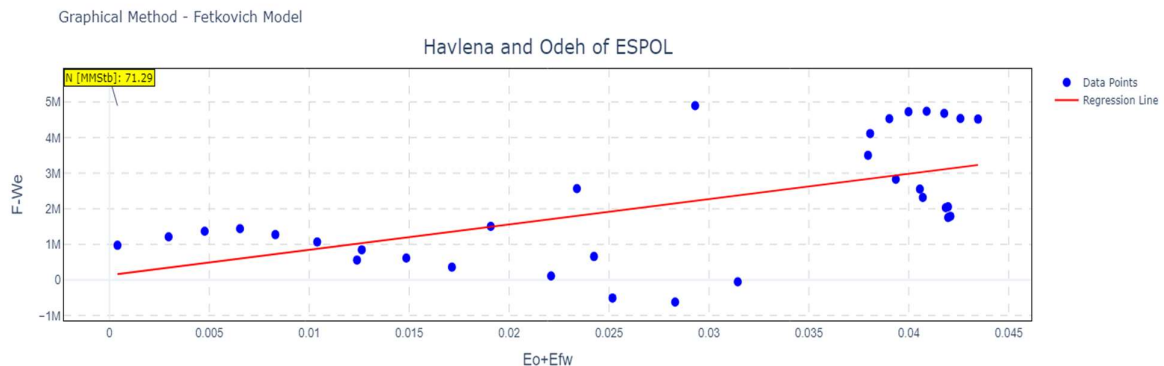
Imagen 15. Interfaz gráfica de Pytank View – Módulo Tank con modelo de Fetkovich



Fuente: Autores

Una vez modelado el acuífero con sus propiedades se procede a obtener las reservas mediante el módulo Analysis de la aplicación. Obteniendo la siguiente gráfica:

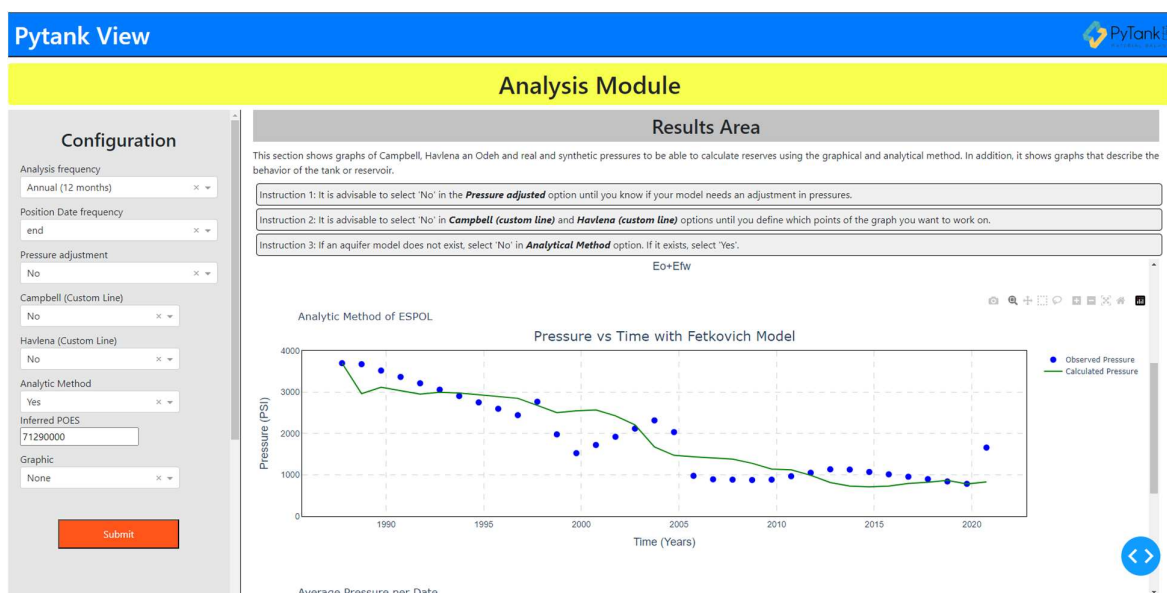
Gráfico 4. Método de Havlena & Odeh con aporte de acuífero



Fuente: Autores

En esta gráfica se aprecia que el valor de las reservas cambia a 71.29 millones de barriles en superficie. Este cambio se debe a que ahora el reservorio cuenta con el aporte de un acuífero. Para poder constatar que este valor es correcto, podemos hacer uso de la opción *Analytic Method* del mismo módulo Analysis. Esta opción necesita de un POES inferido. En este caso, para constatar el POES obtenido del método gráfico, se infiere su valor, tal como se muestra en la siguiente Imagen 16.

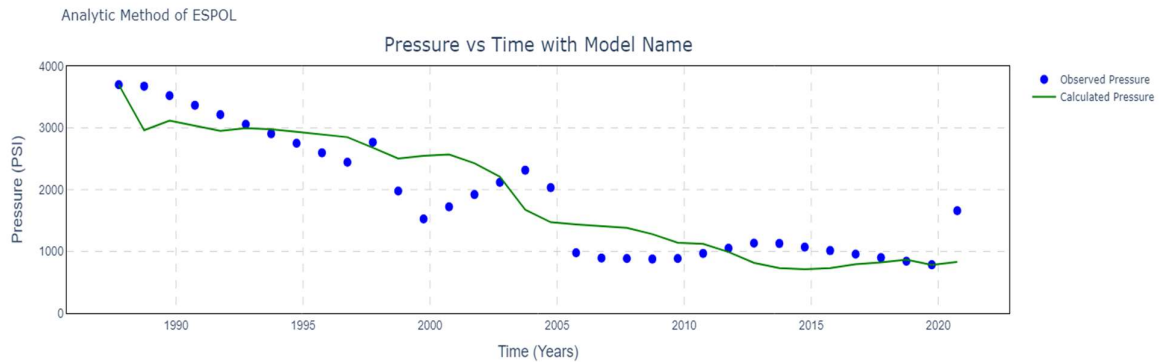
Imagen 16. Interfaz gráfica de Pytank View – Módulo Analysis con la opción del método analítico



Fuente: Autores

A partir de esta opción se obtiene la gráfica de presiones sintéticas y presiones reales. A continuación, se muestra de mejor manera la gráfica:

Gráfico 5. Presiones reales y Sintéticas para el método analítico – Con aporte de acuífero



Fuente: Autores

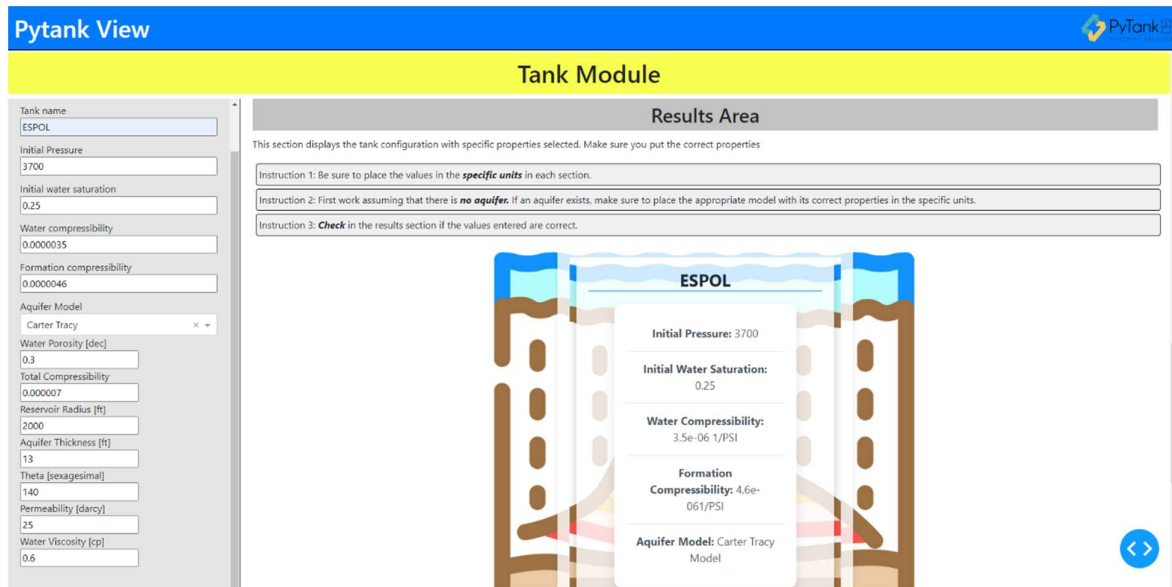
La gráfica 5 permite deducir si el POES que fue obtenido mediante el método gráfico es el correcto. Para generar la gráfica se necesitan las propiedades del acuífero, en conjunto a un POES inferido, dando lugar a presiones sintéticas calculadas a partir de estos parámetros. Para verificar el valor de POES debe haber un comportamiento similar entre las presiones reales (promedios) y las presiones sintéticas. En la gráfica se puede apreciar que existe un comportamiento similar entre las dos presiones, verificando el POES obtenido.

3.2.3 Caso 3: Yacimiento con Acuífero utilizando Carter Tracy

En la sección anterior se escogió modelar el acuífero con Fetkovich. Sin embargo, se puede tener otro modelo, Carter Tracy. Este modelo para el análisis realizado no es la mejor opción, ya que en Carter Tracy se asume un acuífero de radio infinito y tal como se mostró en la gráfica de Campbell, aunque existe aporte energético de un acuífero, este no es tan grande. Por lo tanto, modelar un acuífero infinito no es la mejor opción y el valor de las reservas no sería preciso. A continuación, se verá un ejemplo de su modelamiento.

- Se debe establecer un modelo de acuífero con los siguientes parámetros como se logra ver a continuación:

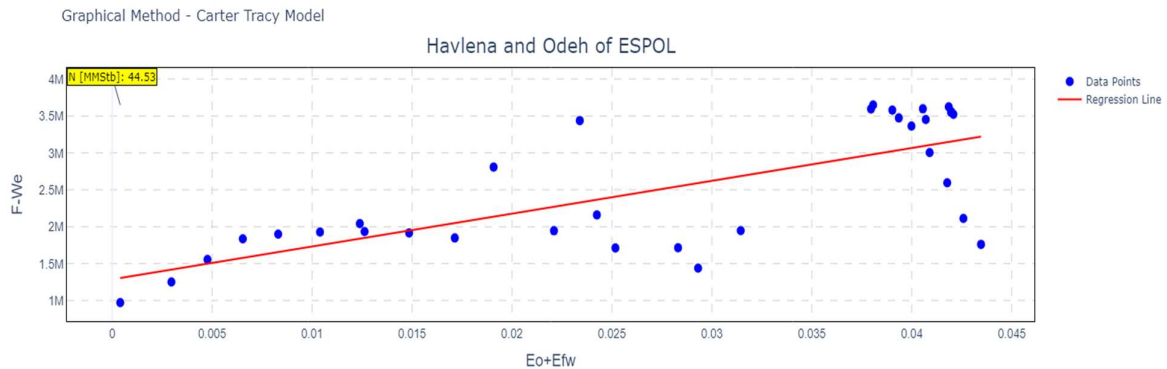
Imagen 17. Módulo Tank con modelo de Carter Tracy para el acuífero



Fuente: Autores

- Los resultados obtenidos según el modelo de acuífero establecido serían los siguientes:

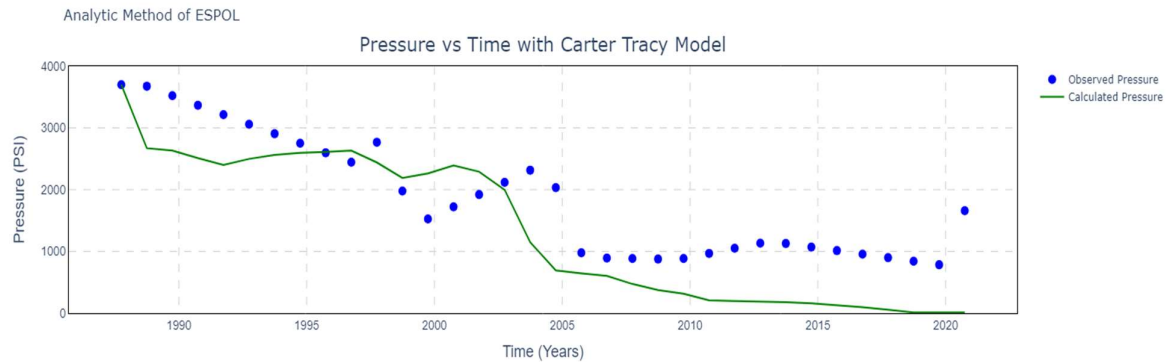
Gráfico 6. Havlena y Odeh para el método gráfico – Con aporte de acuífero



Fuente: Autores

- En el Gráfico 6. se aprecia que el valor de las reservas cambió a 44.53 millones de barriles en superficie. Ahora, se verificará este valor mediante el método analítico, infiriendo el mismo valor de reservas obtenidos mediante el método gráfico.

Gráfico 7. Presiones Reales y Sintéticas para el método analítico – Con aporte de acuífero



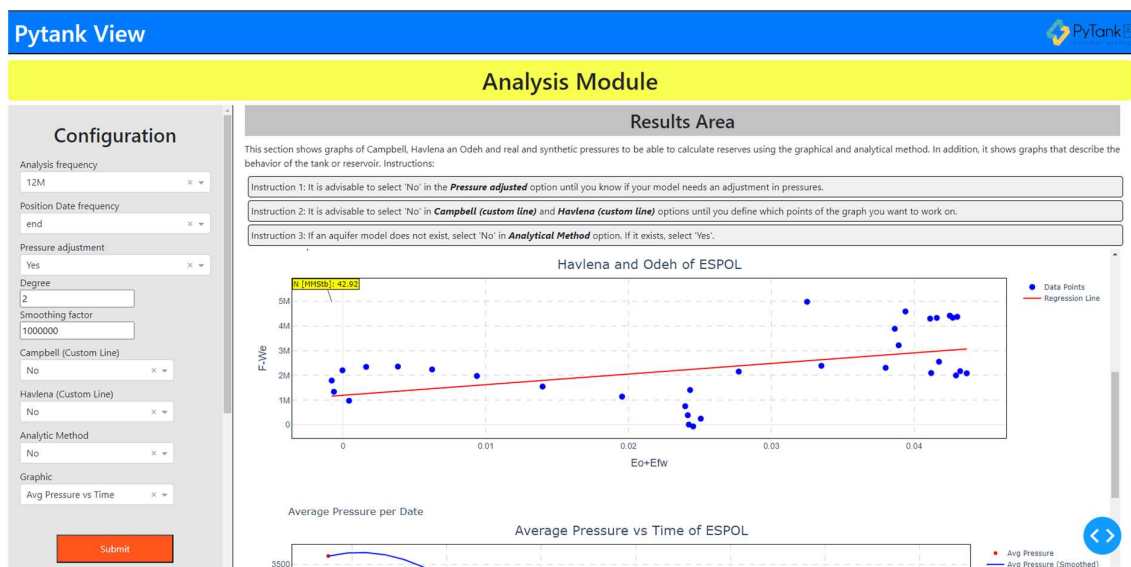
Fuente: Autores

- En el Gráfico 7 se aprecia que el comportamiento entre las presiones difiere en mayor medida a comparación con el modelo de Fetkovich. Como se esperaba, el modelo de Carter Tracy no representa un modelo significativo para representar el acuífero del reservorio analizado.

3.3 Ajuste de presiones

Otro caso se presenta cuando la presión promedio es ajustada. Esta opción es permitida también en el módulo Analysis, tal como se muestra en la siguiente Imagen 18:

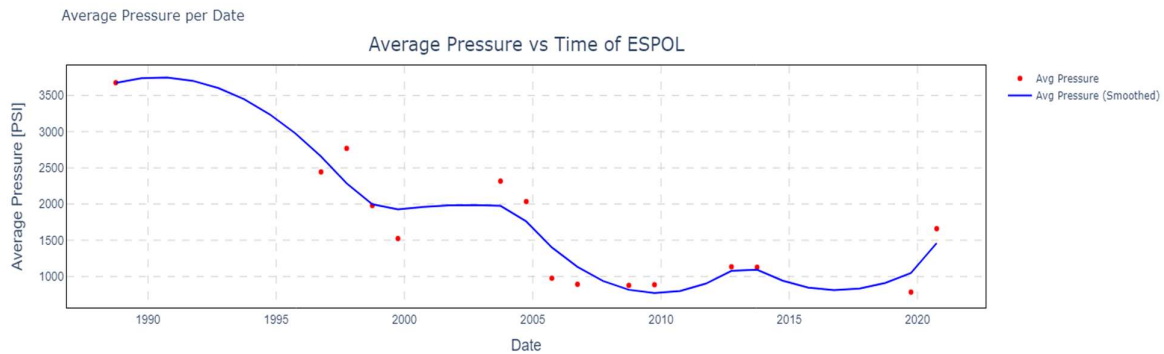
Imagen 18. Módulo Analysis con la opción para ajustar el comportamiento de la presión



Fuente: Autores

Con la presión ajustada y con un modelo de acuífero de Fetkovich se obtiene las siguientes gráficas:

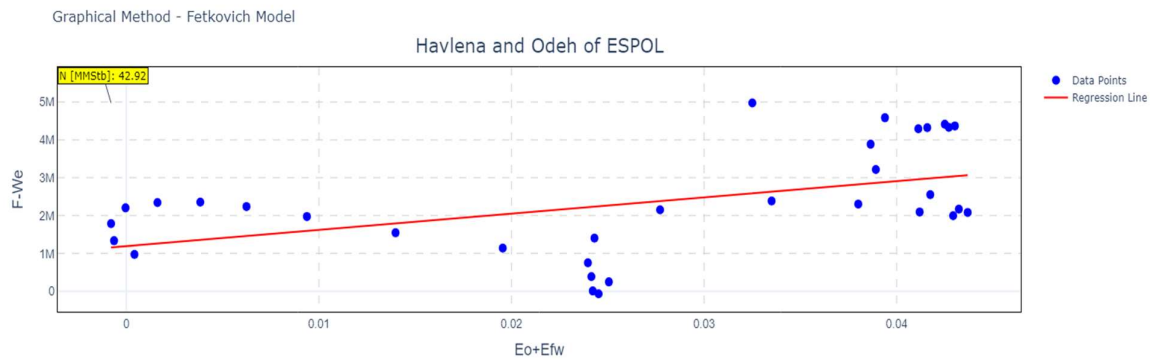
Gráfico 8. Presiones Promedios ajustada según el usuario



Fuente: Autores

En el gráfico 8 vemos como la presión promedio ha sido ajustada y no se aprecian picos pronunciados. Este cambio en el comportamiento de la presión también cambia el valor de las reservas. Por ejemplo, se obtiene el siguiente valor de 42.92 millones de barriles en superficie de reservas de petróleo.

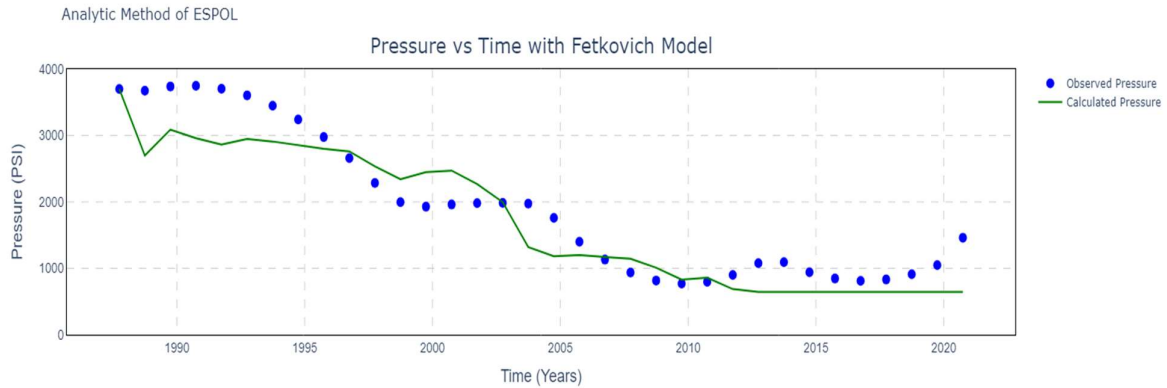
Gráfico 9. Havlena y Odeh para el método gráfico – Ajuste de presiones promedios



Fuente: Autores

Como se aprecia en el Gráfico 9 el valor de las reservas bajó a 42.92 millones de barriles en superficie. Si verificamos este valor mediante el método analítico, obtenemos la siguiente gráfica:

Gráfico 10. Presiones Reales y Sintéticas para el método analítico – Ajuste de presiones



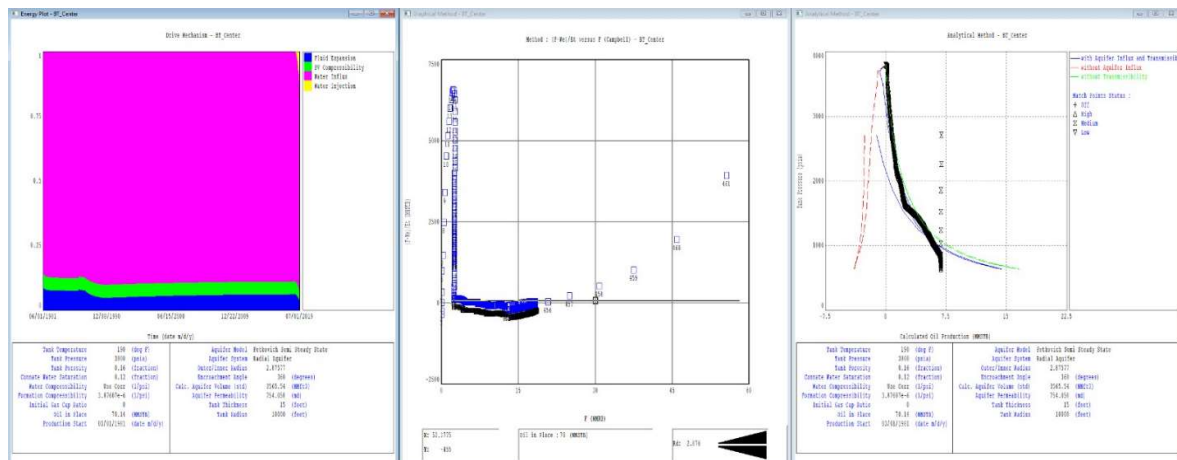
Fuente: Autores

En el Gráfico 10 observamos que con el valor inferido de 42.92 millones de barriles en superficie existe también un comportamiento similar entre las presiones observadas y sintéticas. Ahora, si bien ha sido ajustada la presión, no significa que el ajuste lleve a un valor más preciso. Esto se justificará en la siguiente sección.

3.4 Comparación entre data real y obtenida

A continuación, se presentará una comparación entre los valores teóricos verdaderos de las reservas del campo ESPOL con los valores obtenidos mediante Pytank:

Imagen 17. Resultados en software comercial



Fuente: Proporcionada por la empresa dueña de la información

Tabla 2. Resultados reales vs Obtenidos con la librería

Reservas de Petróleo					
Real [MMStb]			Obtenida [MMStb]		
Sin acuífero	Modelo de Fetkovich	Modelo de Carter-Tracy	Sin acuífero	Modelo de Fetkovich	Modelo de Carter-Tracy
-----	70.14	-----	267.05	71.29	44.53

Fuente: Autores

Como se aprecia en la tabla 2 vemos que el reservorio ESPOL con mayor número de pozo posee un valor real de 70.14 millones de barriles en superficie, obtenidos con un software comercial (data real de la empresa dueña de la información). Con Pytank se obtuvo un valor de 71.29 millones de barriles en superficie, demostrando que la librería cumple con su propósito y que el valor obtenido difiere en un mínimo porcentaje del valor real, exactamente 1.63% de error. Con esta comparativa se puede analizar también el caso respectivo al ajuste de las presiones. En la gráfica 9 se observa que cuando se hace el ajuste, el valor de las reservas disminuye, lo que crea un valor erróneo en su obtención. Por ese motivo se aclaró que el ajuste es ya de criterio ingenieril y que puede no representar como consecuencia una mayor precisión en los resultados. Cabe recalcar que, para ambos análisis, tanto en el software como en Pytank, se ha trabajado con data similar, más no exactas en algunos parámetros debido al anonimato de la información.

3.5 Comportamiento del tanque o reservorio

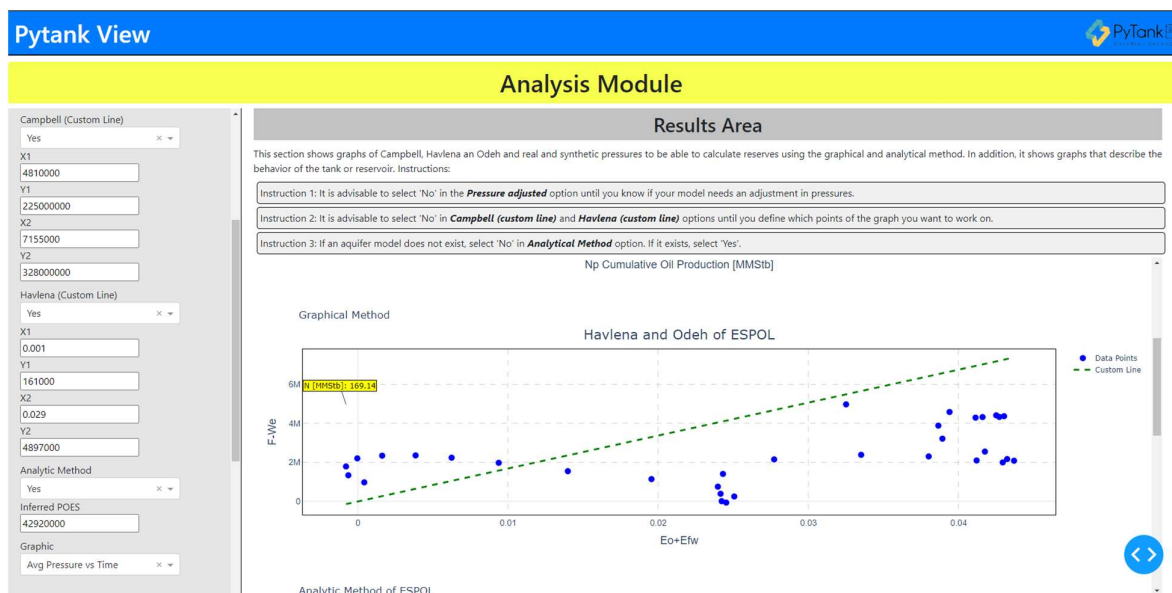
Por último, la librería en conjunto del aplicativo web ofrece una función que pueden ayudar a obtener un valor de reservas más significativo. Esta opción se encuentra en las siguientes gráficas:

- Gráfica de Campbell y Havlena and Odeh: Existe una opción que permite escoger el tramo en el cual se quiere trabajar, a partir de 2 puntos que el usuario debe ingresar para que se genere la regresión lineal.

- También existen otros gráficos de ayuda para visualizar mejor el comportamiento del tanque o reservorio. La opción de *graphic* contiene también a estas. Por ejemplo, el gráfico del aporte de petróleo y agua de cada pozo al tanque.

En el aplicativo web lo podemos encontrar en el módulo de Analysis, tal como se aprecia en el siguiente gráfico 11:

Gráfico 11. Módulo Analysis con la opción para escoger el tramo deseado en las gráficas de Campbell y Havlena

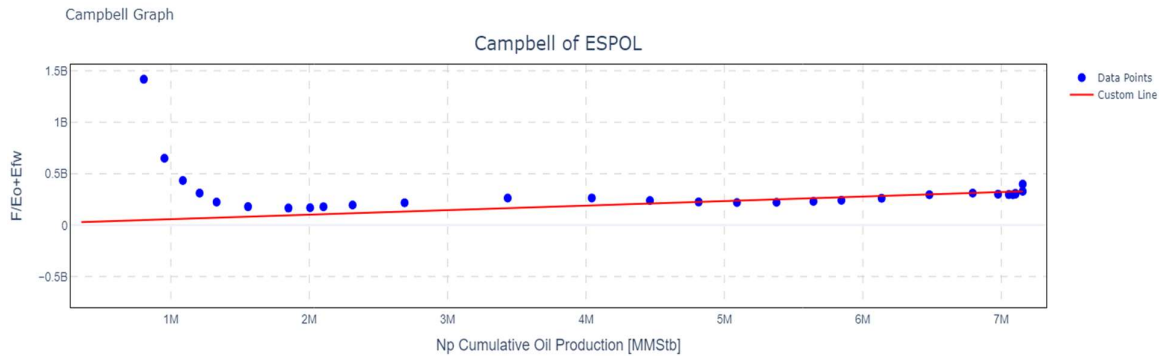


Fuente: Autores

Si se desea ver con el manejo de la librería y no del aplicativo web, quedaría de la siguiente manera, por ejemplo, para el caso de la gráfica de Havlena con la presencia de un acuífero:

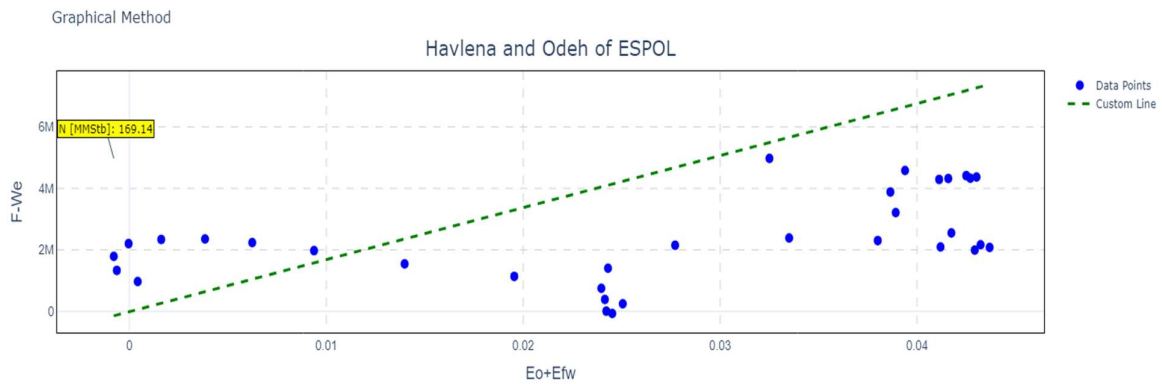
```
havlena_fet_custom = analysis_fet.havlena_odeh_plot(
    custom_line=True,
    x1=0.001,
    y1=161000,
    x2=0.029,
    y2=4897000
)
havlena_fet_custom.show()
```

Gráfico 12. Gráficas de Campbell en un tramo seleccionado



Fuente: Autores

Gráfico 13. Gráficas de Havlena & Odeh en un tramo seleccionado

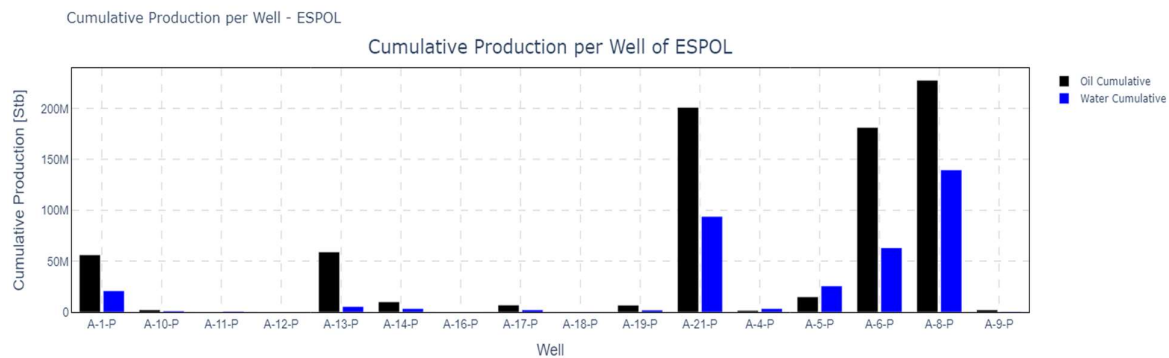


Fuente: Autores

Las gráficas muestran un comportamiento distinto en sus regresiones lineales y eso se debe a que se ha escogido un tramo distinto al que está por defecto. Se recalca que estos tramos son solo de ejemplo para mostrar estos resultados. Puesto que en este caso no fue necesario escoger. Sin embargo, pueden existir escenarios con la presencia de puntos atípicos o dispersos, donde el tramo seleccionado se hace indispensable.

Con respecto a los gráficos de ayuda, tenemos de ejemplo el gráfico 14, de producción acumulada de cada pozo al tanque:

Gráfico 14. Análisis exploratorio con la producción acumulada de petróleo y agua de cada pozo perteneciente al tanque o reservorio



Fuente: Autores

La grafica muestra que el pozo A-8-P es mayor productor de petróleo y de agua hasta la fecha. De igual forma se puede observar otros gráficos que estarán en la parte de anexos.

3.6 Documentación de Pytank

En esta última sección se presenta el tercer y último entregable de este proyecto, para mostrar la documentación correspondiente a la librería desarrollada. Se ha creado un repositorio en GitHub para almacenar el código y la estructura de la documentación, lo que facilita la gestión del proyecto y permite la colaboración con otros desarrolladores.

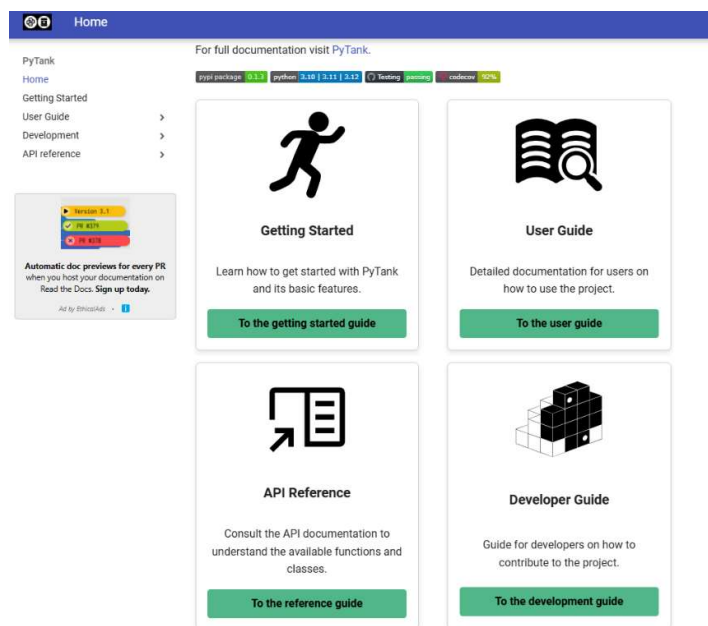
La documentación se presenta como un sitio web estático, desarrollado utilizando MkDocs, una herramienta que permite crear documentación de manera eficiente y atractiva. Además, se implementaron diversas dependencias que contribuyeron a mejorar la funcionalidad y la experiencia del usuario. El proyecto fue alojado en la plataforma Read the Docs, que proporciona un entorno adecuado para gestionar y publicar la documentación de proyectos de software en línea, garantizando su accesibilidad y visibilidad. La documentación se puede visualizar en el siguiente enlace: [PyTank \(pytank-doc.readthedocs.io\)](https://pytank-doc.readthedocs.io)

El objetivo de esta documentación es ofrecer a los usuarios una guía completa sobre las funciones de la librería, así como instrucciones claras para su implementación. Esto no solo facilita

el uso de la librería, sino que también promueve una mayor adopción y comprensión por parte de la comunidad de desarrolladores.

A pesar de que MkDocs utiliza principalmente Markdown para la redacción de la documentación, se ha incorporado HTML en ciertas secciones para mejorar la funcionalidad y la presentación visual, ofreciendo una experiencia más agradable y dinámica para los usuarios. La Imagen 18 ilustra la interfaz gráfica del sitio web estático desde una perspectiva de uso público, destacando su diseño intuitivo y accesible.

Imagen 18. Interfaz gráfica de la página de inicio



Fuente: Autores

La documentación está compuesta por diferentes secciones las cuales son:

- **Home:** Una introducción general a la librería y su propósito.
- **Getting Started:** Instrucciones iniciales sobre cómo instalar y configurar la librería.
- **User Guide:** Una guía detallada que explica cómo utilizar las diferentes funciones de la librería.

- **API Reference:** Una referencia completa de la API, que incluye descripciones de funciones, parámetros y ejemplos de uso.
- **Developer Guide:** Información destinada a desarrolladores que deseen contribuir al proyecto, incluyendo pautas de desarrollo y pruebas.

3.6.1 Home

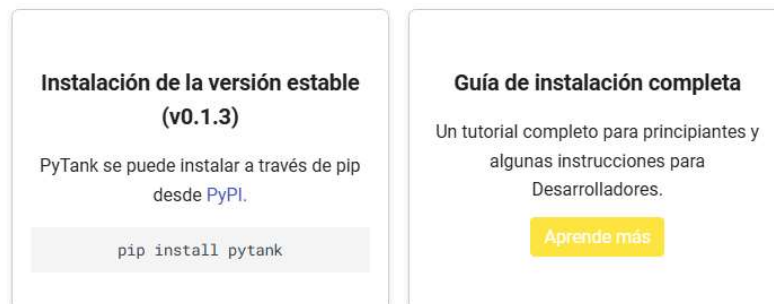
Esta sección proporciona una visión general de la librería como se muestra en la Imagen 18, incluyendo su propósito, características principales y beneficios. También incluye enlaces a las secciones más relevantes, facilitando la navegación.

3.6.2 Getting Started

En esta sección, el objetivo principal es proporcionar a los nuevos usuarios la información necesaria para comenzar a usar Pytank de manera efectiva. Se establece un soporte de versiones de Python, donde se detallan los cambios realizados en la última versión de la librería, así como el acceso a versiones anteriores para aquellos usuarios que deseen trabajar con ellas. Esto garantiza que los usuarios puedan elegir la versión que mejor se adapte a sus necesidades y entorno de desarrollo.

Se presenta un módulo de instrucciones de instalación que permite a los usuarios instalar la librería en su ordenador con un simple comando. Esto simplifica el proceso de instalación, eliminando la necesidad de configuraciones complicadas.

Imagen 19. Interfaz gráfica del módulo Getting Started

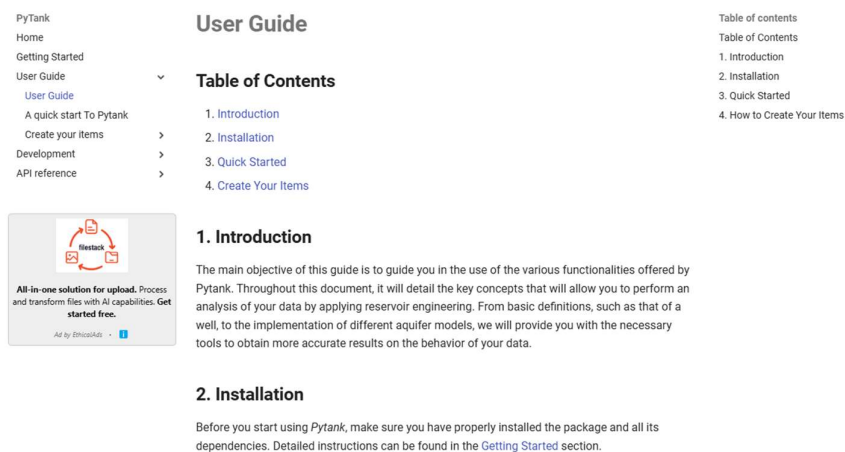


Fuente: Autores

3.6.3 User Guide

En esta pestaña tal como se observa en la Imagen 20 se detallan tanto los métodos principales, así como los secundarios. En este caso existen diversos métodos que tienen varias funciones que cumplen un objetivo distinto de las demás. Además, se presenta una guía de inicio rápido para los usuarios que estén preparados para utilizar la librería.

Imagen 20. Interfaz gráfica del módulo "User Guide"



Fuente: Autores

Esta sección posee diferentes apartados como son:

- **¿Cómo crear un pozo?:** Este apartado explica detalladamente cómo crear un pozo utilizando los métodos disponibles en Pytank. Se proporcionan ejemplos y guías paso a paso para ayudar a los usuarios a familiarizarse con esta funcionalidad fundamental.
- **¿Qué modelos de fluidos se necesitan?:** Se detalla los diferentes modelos de fluidos compatibles con Pytank y se guía a los usuarios sobre cómo desarrollar sus propios modelos en base a la información que posean.
- **¿Cómo crear un tanque?:** Este apartado se centra en el proceso de creación de tanques utilizando Pytank. Se proporcionan instrucciones claras y ejemplos de código para facilitar el uso de los métodos relacionados con tanques.

- **Analizando con Pytank:** En este apartado se explora cómo utilizar Pytank para realizar el análisis pertinente de la información. Se detallan los métodos disponibles para el análisis y se proporcionan ejemplos de cómo interpretar los resultados.
- **Vamos a crear un Acuífero:** Finalmente, se dedica un apartado específico a la creación de acuíferos utilizando Pytank en base a modelos tales como Fetkovich y Carter Tracy. Se explican los pasos necesarios y se proporcionan ejemplos de código para guiar a los usuarios en este proceso.

3.6.4 *Development*

Esta sección dirigida a desarrolladores que deseen contribuir al proyecto de la librería. Su objetivo principal es facilitar la incorporación de nuevos colaboradores al proporcionar pautas claras sobre cómo colaborar, así como información técnica detallada sobre la arquitectura interna de la librería.

Esta sección está dividida en diversas secciones que abordan diferentes aspectos relacionados con el desarrollo y la contribución:

- **Cómo contribuir:** Esta subsección explica el proceso de contribución, incluyendo instrucciones sobre cómo crear un fork del repositorio, realizar cambios en el código y enviar solicitudes de extracción (pull requests). También se detallan las normas y pautas que deben seguirse para garantizar la coherencia y la calidad del código.
- **Documentación interna:** Esta subsección resalta la importancia de mantener una documentación interna actualizada y detallada. Se proporcionan pautas sobre cómo documentar cambios en el código, agregar comentarios explicativos y actualizar la documentación existente para garantizar que los colaboradores tengan acceso a información precisa y relevante.

- Construcción de la documentación: Aquí se explican los pasos necesarios para construir la documentación localmente, utilizando herramientas como MkDocs. Esto permite a los colaboradores verificar sus cambios en la documentación antes de enviarlos, asegurando que todo esté en orden.

3.6.5 API Reference

En esta sección como se muestra en la Imagen 21 se presenta una referencia completa de la API, que incluye descripciones detalladas de cada función, sus parámetros, tipos de retorno y ejemplos de uso. Esta referencia es crucial para desarrolladores que buscan integrar la librería en sus propios proyectos o que necesitan información técnica específica.

Imagen 21. Interfaz gráfica del módulo "API Reference"

Main Method	Functions	Description
Well	no methods	-
Oil_Model	get_bg_at_press(p)	Method to interpolate gas volumetric factor (Bg).
	get_bo_at_press(p)	Method to interpolate oil volumetric factor (Bo).
	get_ro_at_press(p)	Method to interpolate oil solubility (Rs).
Water_Model	get_bw_at_press(p)	Method to calculate the water volumetric factor Bw using correlation.

Fuente: Autores

Capítulo 4: Conclusiones y recomendaciones

4. Conclusiones y recomendaciones

4.1 Conclusiones

- La librería Pytank ha sido desarrollada de una manera integral, optimizando miles de líneas de códigos en pasos simples para cualquier usuario familiarizado con Python que desee realizar cálculos de reservas de petróleo de yacimientos subsaturados. Esto no solo ahorraría tiempo, sino que también simplificaría las instrucciones complejas. Además, al tratarse de un código abierto, permitiría a los usuarios acceder al repositorio y revisar el código, garantizando así su accesibilidad, uso y mejora continua.
- Se verificó que la matemática que engloba al balance de materiales se ha integrado en los diferentes módulos y funciones que comprenden a la librería, asegurando que el cálculo de reservas se realice con precisión al considerar todas las variables representativas que modelan un reservorio.
- La implementación de Pytank es sencilla, debido a que ésta se ha alojado en PyPi, un gestor de paquetes que permite descargarla de manera rápida y eficiente, similar a otras populares como Pandas y Numpy. Esto garantiza que la librería ha sido creada bajo normas y convenciones de buenas prácticas de programación y arquitectura, lo que refleja un enfoque profesional más allá de un simple repositorio.
- El aplicativo web ha sido diseñado como una herramienta digital que permite la flexibilidad del uso de la lógica de la librería. Este enfoque ayuda a que los usuarios, sin necesidad de conocimientos en programación, puedan realizar análisis correctos para el cálculo de reservas, siempre que comprendan los fundamentos teóricos del balance de materiales. Como se evidenció en el capítulo 3, los resultados obtenidos son equivalentes tanto al utilizar la librería como la aplicación. Sin embargo, es importante mencionar que manejando Python (programando) se tiene un mayor dinamismo a la hora de ejecutar los análisis.

- La implementación de una documentación exhaustiva no solo optimiza la accesibilidad y la comprensión de la librería, sino que también establece diversas guías clave que facilitan a los usuarios, tanto principiantes como experimentados, la rápida asimilación de las funcionalidades del paquete. Además, esta documentación proporciona la información necesaria para desarrolladores interesados en colaborar, promoviendo así una comunicación activa y colaborativa en torno al proyecto. Este enfoque no solo fomenta la participación de la comunidad, sino que también contribuye significativamente a la evolución y mejora continua de Pytank.
- Con respecto a los resultados en los valores de las reservas calculados, se concluye que la librería tiene poco error en la precisión, por ejemplo, el valor obtenido con el método gráfico de Havlena y Odeh es de 71.29 MMSTB que se constató con el método analítico, comparándolo con el valor real para el tanque ESPOL que es de 70.14 MMSTB. Este contraste entre los dos valores garantiza la lógica, matemática y confiabilidad de Pytank. De igual manera, se constató el método gráfico de Campbell para demostrar la influencia de un aporte energético. Esto dio paso a que se represente un acuífero para el reservorio mediante el modelo de Fetkovich, permitiendo de esta manera llegar a un resultado de reservas con mayor precisión.
- Finalmente, se concluye que el proyecto es una herramienta confiable para realizar diferentes análisis de un reservorio. La librería permite modelar al mismo en diferentes condiciones, ajustar presiones en caso de valores atípicos o escasos, y seleccionar los tramos más significativos para los métodos gráficos de Campbell y Havlena-Odeh, aplicando criterios de ingeniería que resultan en valores confiables en escenarios que requieren un enfoque más especializado.

4.2 Recomendaciones

- Antes de comenzar a utilizar la librería, es fundamental que se lea detenidamente la documentación, especialmente las secciones “Getting Started” y “User Guide”. Esto permitirá comprender el propósito de la librería, sus funcionalidades claves y como aplicarlas correctamente en el análisis de reservorios.
- Se recomienda a los usuarios asegurarse de tener instalado Python y las dependencias necesarias en su equipo antes de comenzar a utilizar Pytank. Se deben seguir las instrucciones de instalación proporcionadas en la documentación para garantizar un entorno de trabajo adecuado.
- Al empezar con el uso de la librería, es recomendable que los usuarios comiencen con ejemplos simples y bien documentados. Esto les ayudará a familiarizarse con las funciones antes de abordar problemas más complejos.
- Dado que la librería se desarrolló con conjuntos de datos con nombres de columnas específicos, es crucial que los usuarios se aseguren de que sus datos de entrada cumplan con esta estructura. Utilizar nombres de columnas coherentes y correctos facilitará el procesamiento de datos y garantizará que los cálculos se realicen de manera precisa.

Bibliografía

- Aguas, D., & Yoon, S. (2023). Desarrollo de librería en Python para la determinación de reservas de petróleo mediante el balance de materiales [Tesis de Grado]. Escuela Superior Politécnica del Litoral.
- Bakhty, N., Demin, A., & Tupitsin, M. (2020, October 26). Assisted Creation and Usage of Material Balance Models for Production Forecasting as a Part of Integrated Field Management. Day 1 Mon, October 26, 2020. <https://doi.org/10.2118/201958-MS>
- Bell, D., & Parr, M. (2003). Java para estudiantes (Pearson Editorial, Ed.).
- Canel, C. A., & Rosbaco, J. (1992, March 8). Compositional Material Balance: Its Application to the Development of an Oil and Gas Field with Retrograde Condensation. All Days. <https://doi.org/10.2118/23647-MS>
- Dake, L. P. (1983). Fundamentals of Reservoir Engineering (Vol. 8). Elsevier.
- Duque, R. G. (2011). Python PARA TODOS. <http://mundogeek.net/tutorial-python/>
- Escobar, F. H. (2012). FUNDAMENTOS DE INGENIERIA DE YACIMIENTOS.
- Fernández, M., Goiria, J. G., Fernando, A., María, H., Martínez, J., & Karlos Pérez De Armiño, H. (2020). Challenges to development: processes of change towards global justice (Primera). <https://reedes.org>
- Ferrer, M. P. de. (2009). Fundamentos de ingeniería de yacimientos (1st ed., Vol. 1). Ediciones Astro Data S. A.
- Garrell, A., & Llorenc, G. (2019). La Industria 4.0 en la sociedad digital (Gibert Núria, Ed.; 1st ed.). Merge Books. <https://doi.org/10.29057/escs.v9i18.9002>
- Guagliano, C. (2019). Programación en Python 2: programación orientada a objetos (C. Peña, Ed.; Six Ediciones, Vol. 2). RedUsers. <https://books.google.es/books?hl=es&lr=&id=y1yzDwAAQBAJ&oi=fnd&pg=PA5&dq=programaci%C3%B3n+orientada+a+objetos+python&ots=G1yVzIYC4h&sig=IW8TAoNVzIWXzyRj3Qrf3vQi7E#v=onepage&q=programaci%C3%B3n%20orientada%20a%20objetos%20python&f=false>
- Jaworski, M. Z. T. (2021). Expert Python Programming: Master Python by learning the best coding practices and advanced programming concepts (Saby D'silva, Ed.; 4th ed.). Packt Publishing Ltd.,
- Jordy, L., & Kevin, V. (2018). ESTIMACIÓN DE PETRÓLEO ORIGINAL EN SITIO USANDO MÉTODO DE BALANCE DE MATERIALES EN EL YACIMIENTO G-8, CAMPO FICT-GF, CASO DE ESTUDIO.

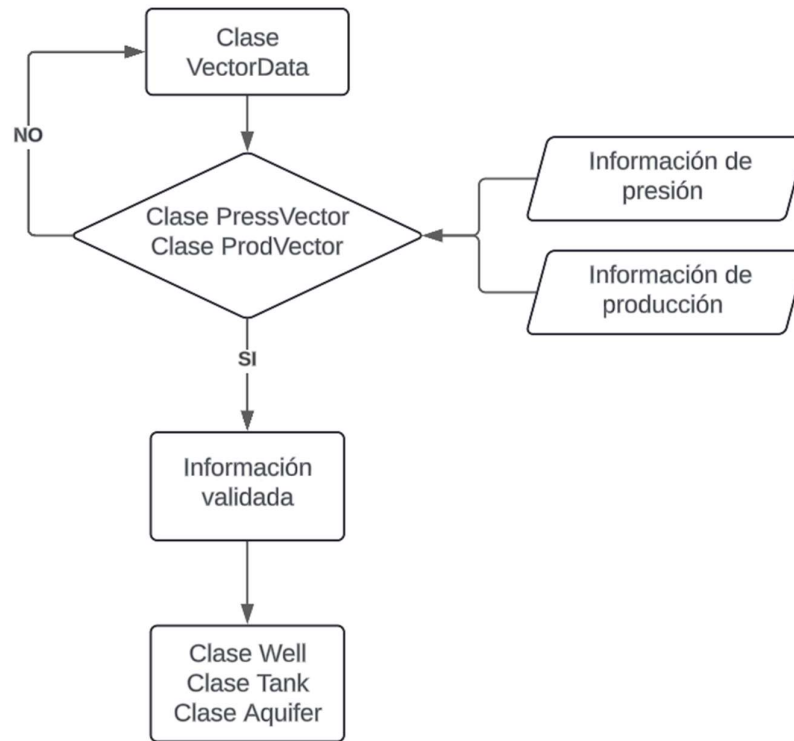
- Lopez, F., Béjar, R., Latre, M., Nogueras, J., & Zarazaga, J. (2015). GitHub como herramienta docente (Universitat Oberta La Salle, Ed.). <http://subversion.apache.org/3http://mercurial.selenic.com/4http://git-scm.com/>
- Martínez Rafael, Pereira Manuel, & González Raimundo. (2014). Viabilidad de Python en la enseñanza de la programación Feasibility of Python in teaching programming. *Revista Científico Pefagógica*, 12(2), 179–186.
- Palacio, E. (2023, November 2). Poetry: Simplificando la Creación y Gestión de Entornos Virtuales en Python. *LikendIn*. <https://www.linkedin.com/pulse/poetry-simplificando-la-creaci%C3%B3n-y-gesti%C3%B3n-de-en-palacio-gaviria-zucpe/>
- Pandera developers. (2019). UNION PANDERA. The Open-Source Framework for Precision Data Testing. <https://pandera.readthedocs.io/en/stable/>
- Pérez, I., Ricardo, Y., & García Roberto. (2014). El lenguaje de programación Python. *Ciencias Holguín*, XX(2), 1–13. <http://www.linuxjournal.com/article/2959>
- Pydantic developers. (n.d.). Pydantic. Retrieved June 15, 2024, from <https://docs.pydantic.dev/latest/>
- Satter, A., & Iqbal, G. M. (2016). Reservoir performance analysis by the classical material balance method. In *Reservoir Engineering* (pp. 233–245). <https://doi.org/10.1016/b978-0-12-800219-3.00014-0>
- Toledo, Á. (2010). Claves para Interpretar la Evolución de la Industria Petrolera Internacional bajo el Neoliberalismo [Tesis de Doctorado, Universidad Autónoma de Zacatecas]. <https://estudiosdeldesarrollo.mx/wp-content/uploads/2021/04/Angel-Toledo-Tolentino.pdf>
- Wheaton, R. (2016). Analytical Methods for Prediction of Reservoir Performance. In *Fundamentals of Applied Reservoir Engineering* (pp. 75–105). Elsevier. <https://doi.org/10.1016/B978-0-08-101019-8.00004-1>
- Zambrano, O., Alexandra, J., Mariuxi, Lady, Tapia, S., Alexandra, J., & Santillán, A. (2017). IMPORTANCIA DE LA PROGRAMACIÓN EN LA FORMACIÓN DE LOS INGENIEROS DE SISTEMAS COMPUTACIONALES IMPORTANCE OF PROGRAMMING IN THE TRAINING OF COMPUTER SYSTEMS ENGINEERS. *Opuntia Brava*, 9(4), 57.
- Aguas, D., & Yoon, S. (2023). Desarrollo de librería en Python para la determinación de reservas de petróleo mediante el balance de materiales [Tesis de Grado]. Escuela Superior Politécnica del Litoral.

- Bakhty, N., Demin, A., & Tupitsin, M. (2020, October 26). Assisted Creation and Usage of Material Balance Models for Production Forecasting as a Part of Integrated Field Management. Day 1 Mon, October 26, 2020. <https://doi.org/10.2118/201958-MS>
- Bell, D., & Parr, M. (2003). Java para estudiantes (Pearson Editorial, Ed.).
- Canel, C. A., & Rosbaco, J. (1992, March 8). Compositional Material Balance: Its Application to the Development of an Oil and Gas Field with Retrograde Condensation. All Days. <https://doi.org/10.2118/23647-MS>
- Dake, L. P. (1983). Fundamentals of Reservoir Engineering (Vol. 8). Elsevier.
- Duque, R. G. (2011). Python PARA TODOS. <http://mundogeek.net/tutorial-python/>
- Escobar, F. H. (2012). FUNDAMENTOS DE INGENIERIA DE YACIMIENTOS.
- Fernández, M., Goiria, J. G., Fernando, A., María, H., Martínez, J., & Karlos Pérez De Armiño, H. (2020). Challenges to development: processes of change towards global justice (Primera). <https://reedes.org>
- Ferrer, M. P. de. (2009). Fundamentos de ingeniería de yacimientos (1st ed., Vol. 1). Ediciones Astro Data S. A.
- Garrell, A., & Llorenc, G. (2019). La Industria 4.0 en la sociedad digital (Gibert Núria, Ed.; 1st ed.). Merge Books. <https://doi.org/10.29057/escs.v9i18.9002>
- Guagliano, C. (2019). Programación en Python 2: programación orientada a objetos (C. Peña, Ed.; Six Ediciones, Vol. 2). RedUsers. <https://books.google.es/books?hl=es&lr=&id=y1yzDwAAQBAJ&oi=fnd&pg=PA5&dq=programaci%C3%B3n+orientada+a+objetos+python&ots=G1yVzIYC4h&sig=IW8TAoNVzIWXzyRj3Qrf3vQi7E#v=onepage&q=programaci%C3%B3n%20orientada%20a%20objetos%20python&f=false>
- Jaworski, M. Z. T. (2021). Expert Python Programming: Master Python by learning the best coding practices and advanced programming concepts (Saby D'silva, Ed.; 4th ed.). Packt Publishing Ltd.,
- Jordy, L., & Kevin, V. (2018). ESTIMACIÓN DE PETRÓLEO ORIGINAL EN SITIO USANDO MÉTODO DE BALANCE DE MATERIALES EN EL YACIMIENTO G-8, CAMPO FICT-GF, CASO DE ESTUDIO.
- Lopez, F., Béjar, R., Latre, M., Nogueras, J., & Zarazaga, J. (2015). GitHub como herramienta docente (Universitat Oberta La Salle, Ed.). <http://subversion.apache.org/3http://mercurial.selenic.com/4http://git-scm.com/>

- Martínez Rafael, Pereira Manuel, & González Raimundo. (2014). Viabilidad de Python en la enseñanza de la programación Feasibility of Python in teaching programming. *Revista Científico Pefagógica*, 12(2), 179–186.
- Palacio, E. (2023, November 2). Poetry: Simplificando la Creación y Gestión de Entornos Virtuales en Python. *LikendIn*. <https://www.linkedin.com/pulse/poetry-simplificando-la-creaci%C3%B3n-y-gesti%C3%B3n-de-en-palacio-gaviria-zucpe/>
- Pandera developers. (2019). UNION PANDERA. The Open-Source Framework for Precision Data Testing. <https://pandera.readthedocs.io/en/stable/>
- Pérez, I., Ricardo, Y., & García Roberto. (2014). El lenguaje de programación Python. *Ciencias Holguín*, XX(2), 1–13. <http://www.linuxjournal.com/article/2959>
- Pydantic developers. (n.d.). Pydantic. Retrieved June 15, 2024, from <https://docs.pydantic.dev/latest/>
- Satter, A., & Iqbal, G. M. (2016). Reservoir performance analysis by the classical material balance method. In *Reservoir Engineering* (pp. 233–245). <https://doi.org/10.1016/b978-0-12-800219-3.00014-0>
- Toledo, Á. (2010). Claves para Interpretar la Evolución de la Industria Petrolera Internacional bajo el Neoliberalismo [Tesis de Doctorado, Universidad Autónoma de Zacatecas]. <https://estudiosdeldesarrollo.mx/wp-content/uploads/2021/04/Angel-Toledo-Tolentino.pdf>
- Wheaton, R. (2016). Analytical Methods for Prediction of Reservoir Performance. In *Fundamentals of Applied Reservoir Engineering* (pp. 75–105). Elsevier. <https://doi.org/10.1016/B978-0-08-101019-8.00004-1>
- Zambrano, O., Alexandra, J., Mariuxi, Lady, Tapia, S., Alexandra, J., & Santillán, A. (2017). IMPORTANCIA DE LA PROGRAMACIÓN EN LA FORMACIÓN DE LOS INGENIEROS DE SISTEMAS COMPUTACIONALES IMPORTANCE OF PROGRAMMING IN THE TRAINING OF COMPUTER SYSTEMS ENGINEERS. *Opuntia Brava*, 9(4), 57.

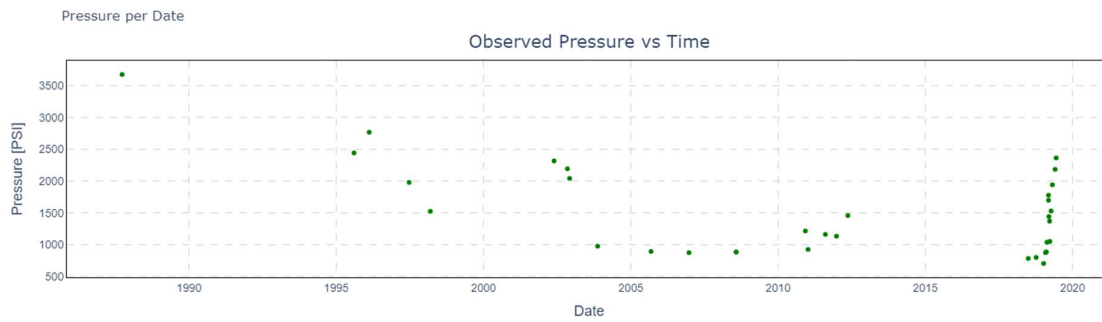
Anexos

Ilustración 15. Algoritmo para la Clase VectorData



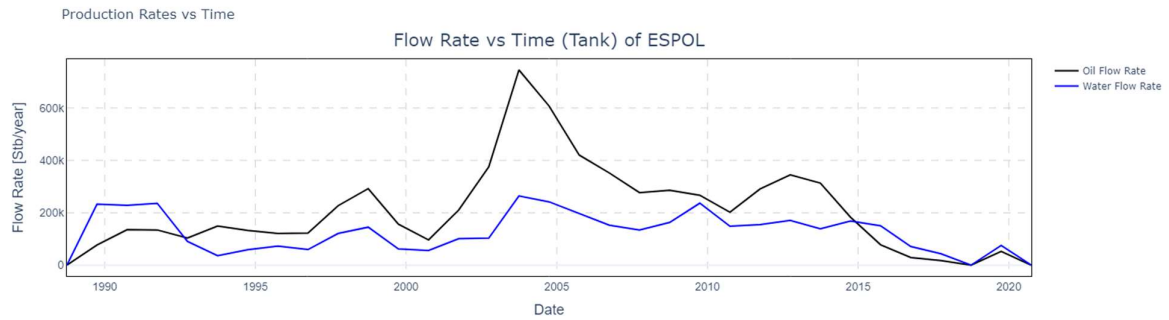
Fuente: Autores

Gráfico 15. Gráfico Presiones Observadas vs Tiempo



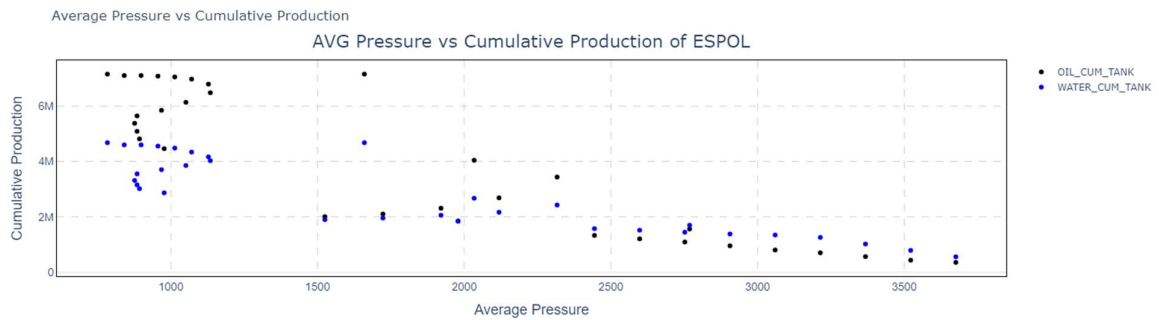
Fuente: Autores

Gráfico 16. Caudales totales del tanque ESPOL a través del tiempo



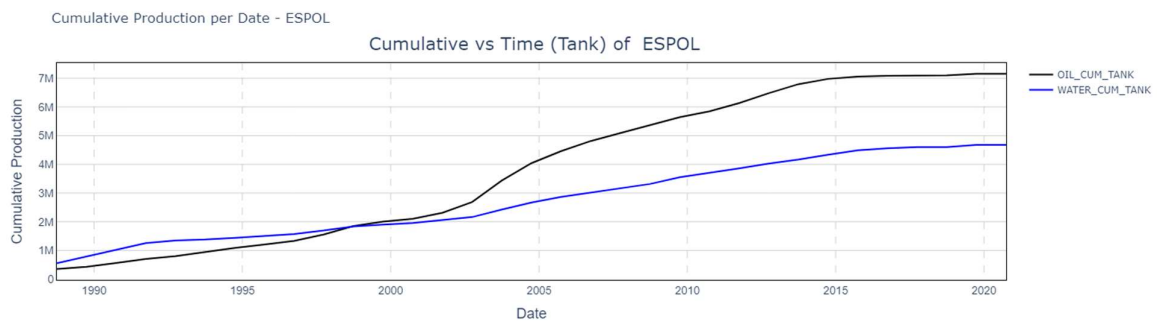
Fuente: Autores

Gráfico 17. Presiones promedio vs producción acumulada de agua y petróleo del tanque ESPOL



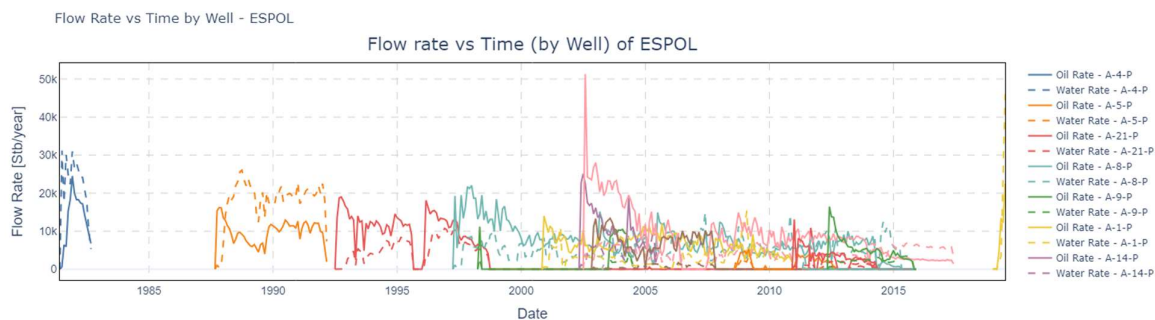
Fuente: Autores

Gráfico 18. Producción acumulada de petróleo y agua a través del tiempo del tanque ESPOL



Fuente: Autores

Gráfico 19. Caudales de petróleo y agua por pozos a través del tiempo del tanque ESPOL



Fuente: Autores

PYTANK: PYTHON PARA LA ESTIMACIÓN DE RECURSOS HIDROCARBURÍFEROS

PROBLEMA

Los estudiantes, universidades, académicos y pequeñas empresas enfrentan dificultades para acceder a herramientas de estimación de reservas debido a los altos costos y la falta de flexibilidad de las soluciones comerciales, lo que limita su capacidad para adquirir experiencia práctica y adaptarse a las demandas del mercado.



**COSTOS
ELEVADOS**



**NO
FLEXIBILIDAD**



**ACCESO
LIMITADO**

OBJETIVO GENERAL

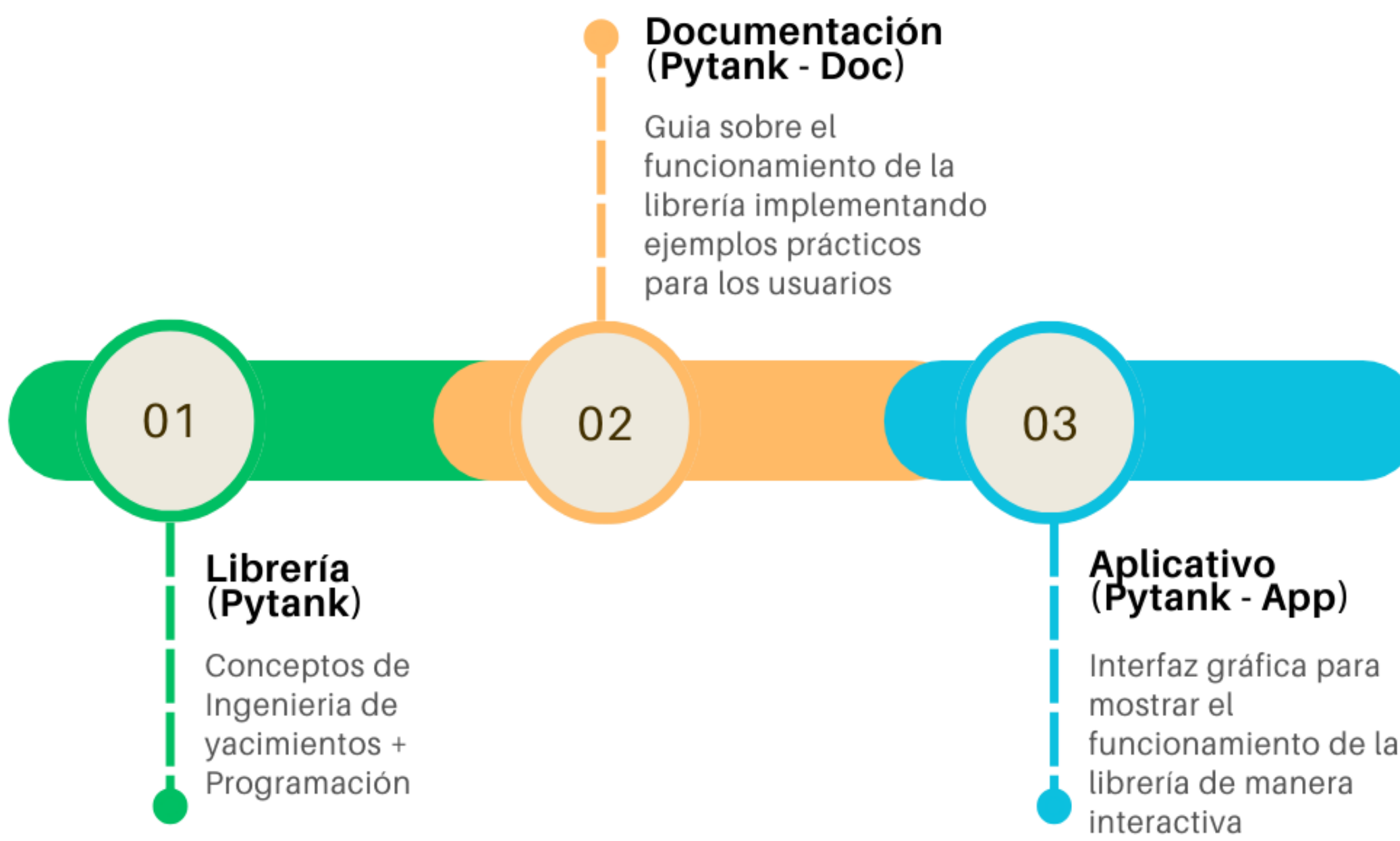
Desarrollar una librería en Python utilizando balance de materiales en yacimientos subsaturados, enfocada en la estimación de reservas de petróleo.



PROPUESTA

Para llevar a cabo el desarrollo de este proyecto, se lo dividió en tres etapas principales que se complementan entre sí para lograr un resultado óptimo y eficiente:

ETAPAS DEL PROYECTO



Librería cargada en PyPi:



Documentación de Pytank:



Aplicativo Web de Pytank:



PROPUESTA DE VALOR

COSTOS

Pytank a diferencia de otras herramientas, asegura su uso de manera gratuita y libre.

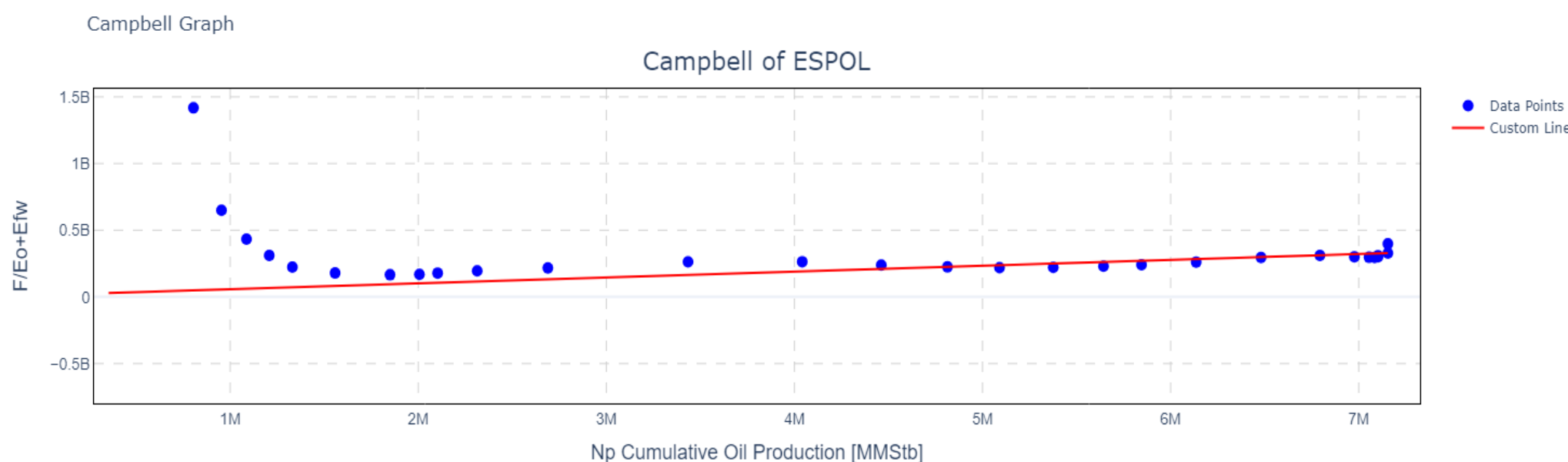
ADAPTABILIDAD

Altamente adaptable, lo que facilita la colaboración y evolución constante de la librería para satisfacer diversas necesidades en el cálculo de reservas.

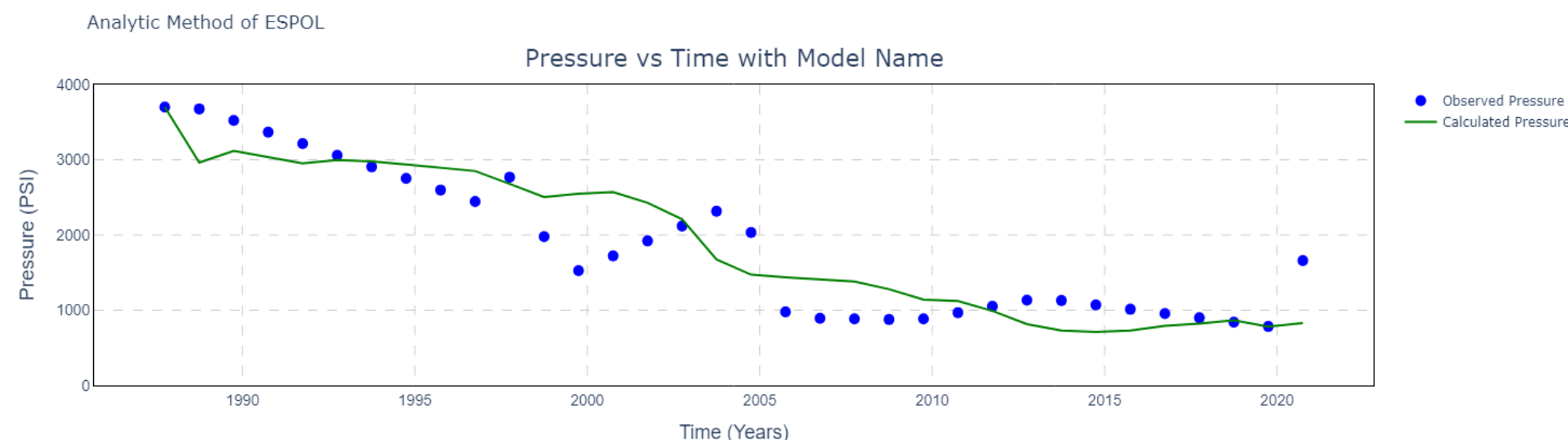
CALIDAD

La librería demuestra alta precisión en el cálculo de reservas, con mínimos errores en comparación a softwares comerciales, validando su lógica y confiabilidad.

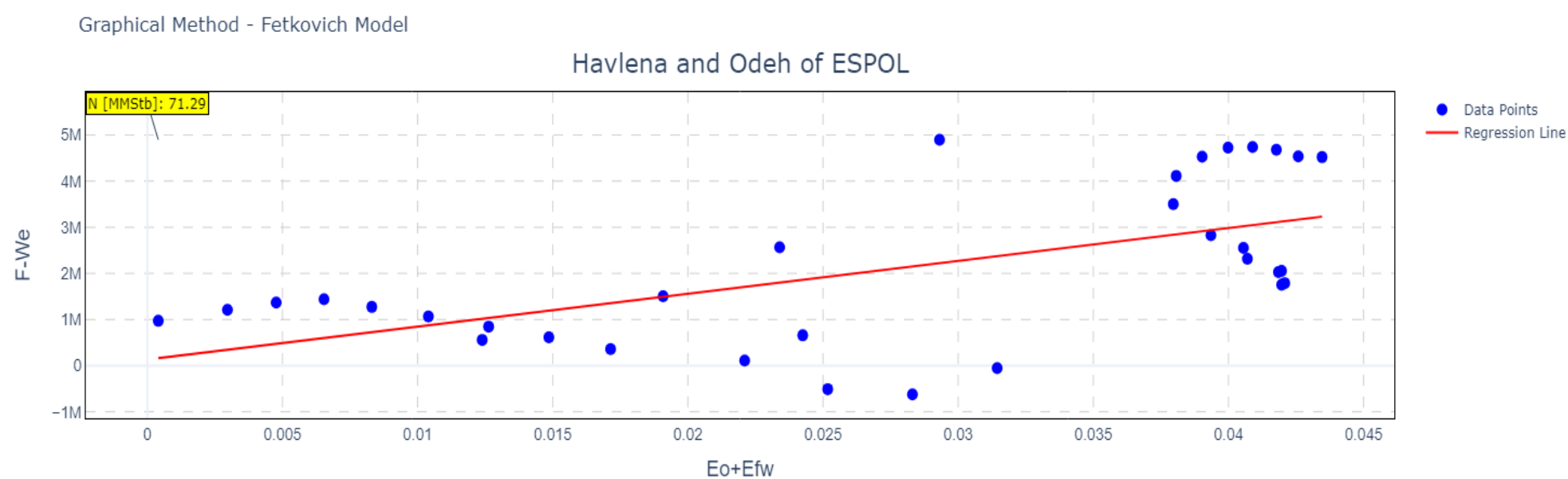
RESULTADOS



Gráfica 1. Método gráfico de Campbell que verifica la presencia de un aporte energético debido a un acuífero.



Gráfica 3. Método analítico de presiones reales y sintéticas que verifica el valor de las reservas obtenidos mediante el método gráfico.



Gráfica 2. Método gráfico de Havlena y Odeh para el cálculo de las reservas de petróleo de un yacimiento subsaturado con acuífero, dando como resultado 71.29 MMSTB.

Reservas de petróleo [MMSTB]	
Software Comercial	Pytank
70.14	71.29

Tabla 1. Comparación entre valores de reservas (POES) obtenidos a través de un software comercial y de Pytank, verificando la precisión de sus resultados.

CONCLUSIONES

- La librería Pytank a diferencia de otras herramientas simplifica cálculos de reservas de petróleo optimizando el tiempo y asegurando su precisión.
- Al ser de código abierto, Pytank asegura a los usuarios tener acceso al código fuente, garantizando su accesibilidad, uso y mejora continua.
- La lógica, cálculos matemáticos y confiabilidad de Pytank han sido validadas al comparar el valor del POES calculado de 71.29 MMSTB obtenido mediante la librería, con valor real para el reservorio ESPOL de 70.14 MMSTB, demostrando la precisión de la librería.