ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN GEOLOGÍA Y PETRÓLEOS

INTEGRACIÓN DIGITAL DEL MODELO DE CHOKE DE UN CAMPO DE LA REGIÓN AMAZÓNICA DEL ECUADOR

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN PETRÓLEOS

OPCIÓN: ARTÍCULO ACADÉMICO

LESLY MICHELLE RIVERA SILVA

lesly.rivera@epn.edu.ec

DIRECTOR: MSc. VINICIO RENÉ MELO GORDILLO

vinicio.melo@epn.edu.ec

Quito, febrero 2022

DECLARACIÓN

Yo, Lesly Michelle Rivera Silva, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

_____

Lesly Michelle Rivera Silva

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Lesly Michelle Rivera Silva, bajo mi supervisión.

MSc. Vinicio René Melo Gordillo
Director del Trabajo de Titulación

AGRADECIMIENTOS

A mis padres Edwin y Doris por haber sido un pilar fundamental en mi vida, por todos los valores inculcados, el amor y paciencia que me han sabido brindar, los consejos y motivación que siempre me dieron durante toda mi vida estudiantil, agradezco a Dios por tenerlos en mi vida, de ustedes he aprendido a no rendirme nunca y que los sueños se hacen realidad si uno trabaja con ahínco y pasión.

A mi hermano Andrés, quien me ha escuchado y ha estado conmigo en todo momento.

A mis abuelitos Jorge y Esther por ser mis segundos padres, todo el esfuerzo y los frutos que voy cosechando se los dedico, son unas personas ejemplares de quienes he aprendido y ahora soy quien soy, el amor que me han brindado me ha ayudado para no decaer nunca y seguir con pie firme en todas las metas que me he propuesto.

A Luis quien ha sido mi apoyo durante toda mi instancia en la Universidad, agradezco por los concejos y palabras motivadoras que me supo dar, por la dedicación y determinación que pusimos en aprender siempre nuevas cosas que nos servirán para nuestra vida profesional.

A mi director de tesis, MSc. Vinicio Melo por compartir sus conocimientos y apoyarme durante este último peldaño para alcanzar mi meta más anhelada.

A mi mentora por parte de la empresa Schlumberger, Ing. Estefi Batallas quién ha sido un soporte fundamental para la realización de mi trabajo de titulación, agradezco por la paciencia y calidad humana que ha demostrado durante este proceso.

A la empresa Schlumberger por permitirme realizar este trabajo de titulación y a sus ingenieras quienes lo hicieron posible Annalyn Azancot y Angélica Vargas.

A la Facultad de Ingeniería en Geología y Petróleos por permitirme desempeñar un cargo fundamental de liderazgo al ser parte del Capítulo SPE-EPN y a mis maestros quienes me han preparado para la vida profesional brindándome todos sus conocimientos, agradezco todas las palabras de motivación que permitieron que me envuelva más la pasión por mi carrera.

DEDICATORIA

Se lo dedico a Dios por haberme regalado la vida y la salud, me ha permitido soñar y luchar por todos los objetivos que me he trazado, he podido vivir y disfrutar de las maravillas de la vida rodeada de personas a quienes amo y por quién me esfuerzo cada día por ser mejor.

A mis padres y hermano por guiarme y nunca dejarme sola, han festejado conmigo mis victorias y en las debilidades que se me han presentado han estado ahí para nunca dejarme vencer, me han recordado que si uno sueña en grande puede lograr todo en esta vida y que nunca se debe dar por vencido, que de las caídas se aprende y con más fuerza se levanta para no cometer el mismo error.

A toda mi familia, tíos y primos quienes me han sabido apoyar durante mi vida estudiantil, cada concejo que me han brindado ha sido fundamental para crecer y ser la persona que soy.

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

RESUMEN

Los proyectos de monitoreo continuo de la producción son cada vez más comunes y requiere del uso de modelos confiables como es el modelo de choke, que permite obtener escenarios en donde se puede identificar el potencial de producción por reservorio, pozo y facilidad. Por lo tanto, desarrollar una automatización del modelo es necesario para ahorrar tiempo y potenciar de una manera efectiva y confiable la producción del campo.

El trabajo busca realizar la integración digital en lenguaje Python, para optimizar el tiempo de los ingenieros y tener los modelos actualizados, reemplazando dentro del flujo de trabajo la carga manual de los datos en PIPESIM.

De las bases de datos existentes de cada pozo que opera en un campo de la Región Amazónica del Ecuador, se utilizó los datos del mes de agosto del año 2021. Con la información de cada pozo se procedió a realizar un cargador de datos en Excel para elaborar el modelo de choke mediante la codificación en Canopy con lenguaje Python.

Se utilizó el software PIPESIM de manera indirecta, puesto que, la codificación realizada permite generar los modelos de pozo en el software sin la necesidad de abrir la interfaz. El código está diseñado para trasladar los valores generados en PIPESIM de los nodales A, C y D a un documento Excel.

Finalmente, se construyó el modelo de choke en Power BI, donde se alcanzó un 90 % de optimización en el tiempo que se empleaba para armar el modelo. Los resultados indican las oportunidades existentes de optimización que se tiene a nivel de reservorio, pozo y facilidades del campo.

Palabras Clave: Modelo de choke, integración digital, lenguaje Python, PIPESIM.

ABSTRACT

Continuous production monitoring projects are increasingly common and require the use of reliable models such as the choke model, that allows get scenarios where the production potential by reservoir, well and facility can be identified. Therefore, developing model automation is necessary to save time and improve the field production effectively.

The present work seeks digital data integration in Python language, to optimize the engineers time and have updated models, replacing the manual loading of the data in PIPESIM within the workflow.

From the existing databases of each well that operates in a field of Amazon Region of Ecuador, only the data form August 2021 were used. With the information of each well, a data loader was generated in Excel to develop the choke model by coding in Canopy with Python language.

The PIPESIM software was used indirectly since the coding carried out allows generating the well models in PIPESIM without opening the interface. The code is designed to transfer the values generated in PIPESIM of the nodals A, C and D to an Excel file.

Finally, the choke model was built in Power BI, where 90% optimization was achieved in the time that was used to build the model. The results show the existing opportunities for optimization at the reservoir level, well and field facilities.

Key Words: Choke model, digital integration, Python language, PIPESIM.

1. INTRODUCCIÓN

A lo largo del tiempo se han desarrollado métodos para optimizar la producción de un campo junto con las oportunidades que pueden existir tanto en reservorio, pozo y facilidades. Los pozos candidatos a la implementación del modelo de choke automatizado se encuentran actualmente trabajando con un levantamiento artificial por bombeo electrosumergible (BES) y con un valor de daño.

El potencial de producción a nivel de pozo se realiza mediante una sensibilización a la presión (Pwf) actual, reduciendo su valor hasta una presión objetivo mediante un incremento en la frecuencia (Hz) de la bomba. Mientras que, para los pozos que poseen bombas trabajando al límite de su rango de operación, el potencial de producción se obtiene reemplazando la bomba por otra de mayor rango.

El potencial de producción a nivel de reservorio se determina mediante la simulación con un daño igual a cero, la oportunidad de optimización en la producción será determinada si el pozo no se encuentra estimulado.

Todas las sensibilidades se las realizará con la simulación del perfil de Presión/Temperatura (*P/T Profile*). Tanto los perfiles de temperatura como los de presión se generan nodo por nodo para el sistema analizado. La presión de entrada y salida siempre hace referencia a los límites del sistema (Schlumberger, 2014).

En el texto se presentará el análisis del perfil de Presión/Temperatura como un enfoque sistemático para determinar la capacidad de producción de cada pozo dentro de las facilidades de superficie. Una vez identificada y estudiada cada variable que afecta el comportamiento, solo resta decidir qué cambios mejorarán el potencial del pozo.

En ciertas ocasiones, la previsión de producción a corto y largo plazo se lo realiza de una manera simple, debido a que, el tiempo que conlleva la carga manual de datos no permite realizarlo de manera consecutiva.

Por este motivo los ingenieros de producción se han visto en la necesidad de realizar una integración digital del campo, para obtener el modelo de choke actualizado a las condiciones de cada pozo.

Para optimizar el tiempo y tener los modelos actualizados, se realiza la codificación en Python, la aplicación de algoritmos ayuda al desarrollo del modelo para realizar la predicción de la producción. Además, que es un lenguaje sencillo que fácilmente puede adaptarse a las herramientas predeterminadas de otros softwares como PIPESIM.

1.1 Objetivos

1.1.1 Objetivo General

Automatizar el modelo de choke mediante codificación en Python en un campo de la Región Amazónica del Ecuador.

1.1.2 Objetivos Específicos

- Desarrollar el modelo de choke para pozo, reservorio y facilidades en un campo de la Región Amazónica del Ecuador.
- Generar un cargador automático de todos los inputs en un campo de la Región Amazónica del Ecuador.
- Mejorar la eficiencia del tiempo de trabajo de ingeniería con la automatización del proceso.
- Identificar las oportunidades de optimización del campo.

1.1.3 Alcance

El proyecto abarca el desarrollo digital del modelo de choke para reservorio, pozo y facilidades en 18 pozos de un campo de la Región Amazónica del Ecuador.

2. MATERIALES

2.1 Generalidades

2.1.1 Modelo de Choke

El sistema de producción está conformado por el reservorio, los pozos y las instalaciones. SPE (2016) menciona que las restricciones dentro del sistema pueden estar asociadas a cualquiera de los fluidos producidos (petróleo, gas o agua) o a su vez pueden darse por una combinación entre ellos.

Para conocer la capacidad de oferta base del yacimiento, es necesario considerar los factores que pueden intervenir en la producción como son el daño de formación de pozo (*skin*), la permeabilidad de la formación productora, el espesor y las capacidades de las instalaciones (SPE, 2016). Es útil definir un modelo para un sistema de producción para comprender completamente el potencial y las limitaciones existentes.

SPE (2016) indica que el modelo de choke permite que los activos integrados se concentren en los segmentos del sistema de producción que están limitando el potencial (baja eficiencia) y que potencialmente podrían mejorarse, ya sea con un cambio en las condiciones operativas o con costos operativos adicionales; este proceso se denomina optimización de la producción y eliminación de cuellos de botella.

Los aumentos en el corte de agua, GOR, etc., tienen un impacto en las eficiencias de los sistemas del modelo. El diagrama de choke se puede utilizar para comprender el rango de potencial actual y las limitaciones asociadas del sistema integrado para asegurar un pronóstico realista a corto plazo (SPE, 2016).

Spencer y Morgan (1998) mencionan que el modelo fue introducido por primera vez por el grupo de producción en Aberdeen y considera al reservorio – pozo – facilidad – exportación como un sistema análogo a una tubería con varios estranguladores que restringen el flujo, en la Figura 1 se puede ver representado el esquema.

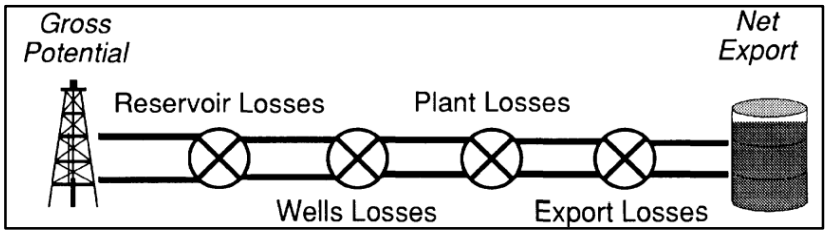


Figura 1. *Concepto de modelo de estrangulador genérico*
Fuente: (Palen & Goodwin, 1996)

El esquema de la Figura 2 indica los puntos que se deben obtener para generar el modelo de choke. El punto A se obtiene a partir de las condiciones iniciales del reservorio, el punto B mediante el incremento de la frecuencia a la que está operando la bomba en el pozo, el punto C es obtenido a partir de una Pwf objetivo (bajos valores de Pwf) y el punto D con la simulación de un daño cero.

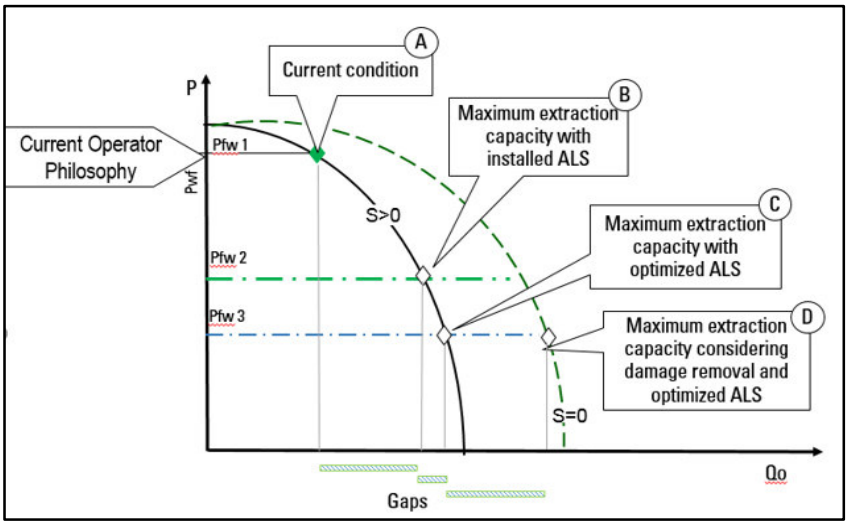


Figura 2. *Puntos para la elaboración del modelo de choke.*
Fuente: Estefi Batallas - Schlumberger

En la Figura 3 se observa el potencial máximo de producción que se obtiene a partir del modelo de choke a nivel de reservorio, pozo y facilidades (planta).

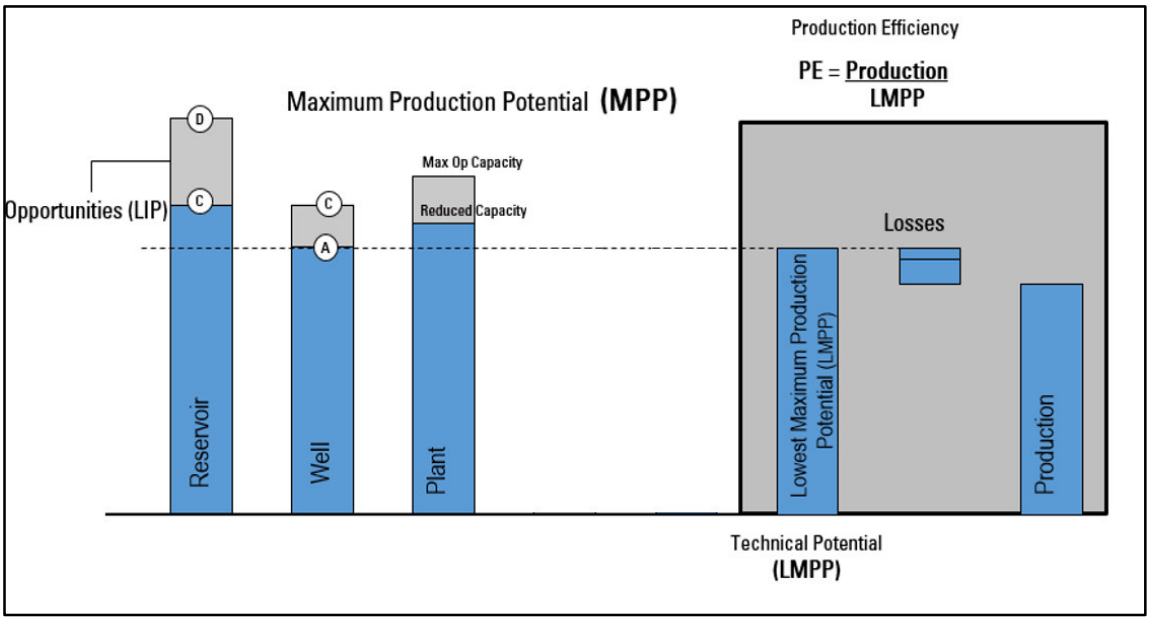


Figura 3. *Modelo de choke.*
Fuente: Estefi Batallas - Schlumberger

2.1.2 Elementos de una Completación Sencilla de Pozo

Completación sencilla: Ortega (2016) denomina a esta completación cuando se tiene solo una tubería de producción, para mayor entendimiento se puede visualizar en la Figura 4.

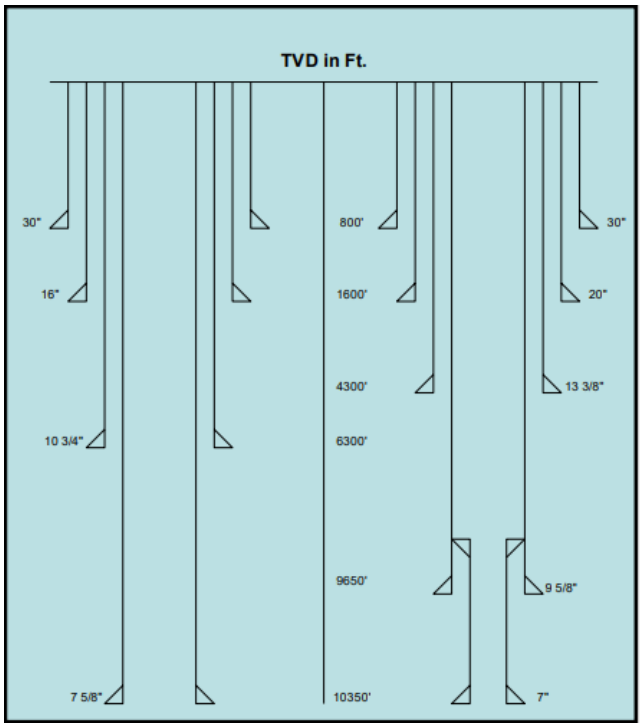


Figura 4. *Completación sencilla, el primero con tubing y el segundo con liner.*
Fuente: (Ortega, 2016)

Casing: También llamada tubería de revestimiento el esquema está representado en la Figura 5, la tubería es utilizada para recubrir la parte interna del pozo perforado, dando estabilidad y evitando que las paredes del pozo colapsen durante la extracción del crudo (Schlumberger, 2022). Se baja para proteger formaciones de agua dulce, aislar zonas que tienen pérdida de circulación o a su vez aislar formaciones con gradientes de presión significativamente diferentes**.**

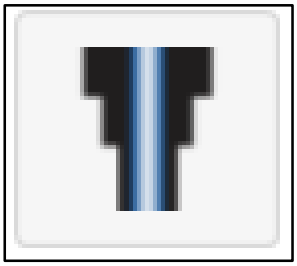


Figura 5. *Casing.*
Fuente: Software Pipesim 2019.1-Schlumberger

En la Figura 6 se puede observar los diferentes tipos de casing para pozos *Onshore.*

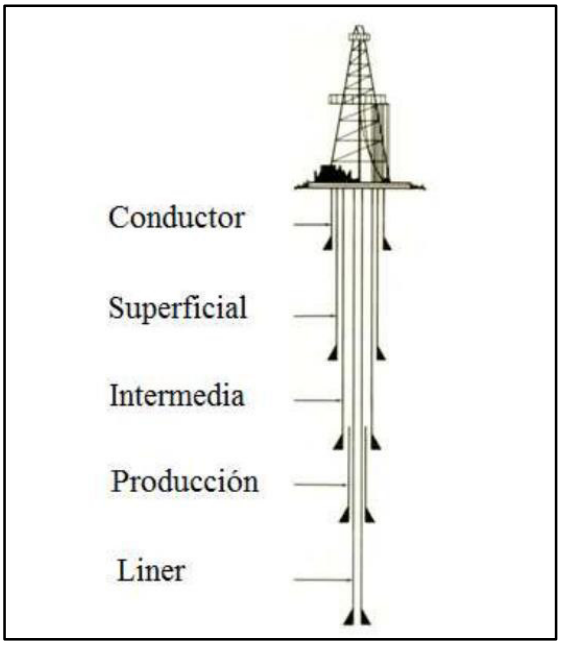


Figura 6. *Tuberías de revestimiento en pozos Onshore.*
Fuente: (Schlumberger, 2004)

Casing Conductor: Es la primera tubería que puede ser cementada el esquema se encuentra representado en la Figura 7; generalmente se instala antes que el taladro de perforación llegue a la locación. Sirve para prevenir derrumbes alrededor de la base del taladro, da soporte al siguiente casing, y sella formaciones someras no consolidadas (Ortega, 2016).

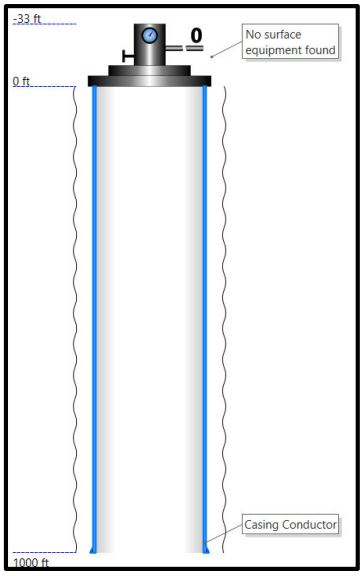


Figura 7. *Casing conductor.*
Fuente: Software Pipesim 2019.1-Schlumberger

Casing Superficial: Es diseñado para soportar cargas de comprensión, en la Figura 8 se encuentra representado el esquema. Sirve para aislar zonas someras poco consolidadas, aísla zonas de agua fresca, soporta el cabezal de pozo y los equipos de BOP (Ortega, 2016).

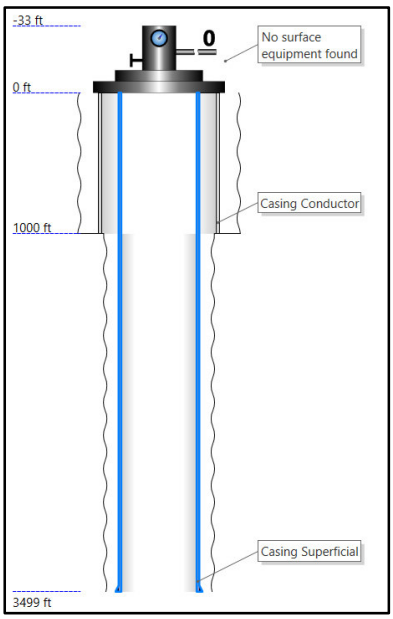


Figura 8. *Casing superficial.*
Fuente: Software Pipesim 2019.1-Schlumberger

Casing Intermedio: Es diseñado para aislar zonas que son inestables, puede o no ser cementado, el esquema se encuentra representado en la Figura 9. Previene el ensanchamiento del hoyo que puede darse por el lavado del fluido de perforación, da soporte al liner (Ortega, 2016).

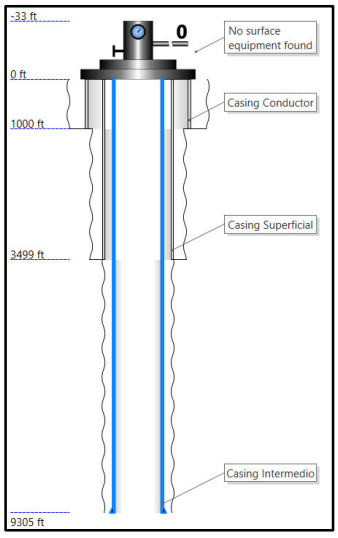


Figura 9. *Casing intermedio.*
Fuente: Software Pipesim 2019.1-Schlumberger

Casing de Producción: Es diseñado para aislar zonas de producción, en la Figura 10 se puede observar el esquema. Soporta la máxima presión del reservorio, resiste las presiones de fractura, debe ser resistente a la corrosión, protege al tubing (Ortega, 2016).

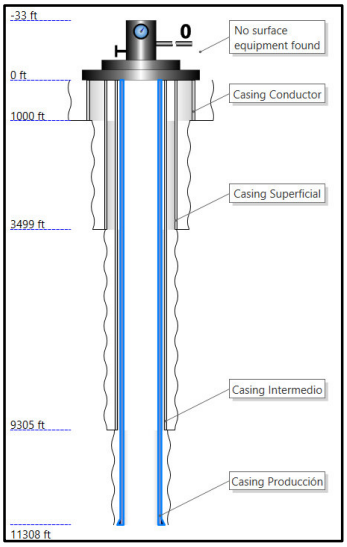


Figura 10. *Casing de producción.*
Fuente: Software Pipesim 2019.1-Schlumberger

Liner: Es una sarta de revestimiento que no se extiende hasta el cabezal del pozo, sino que se encuentra anclada o suspendida desde la sarta de revestimiento previa, el esquema se encuentra representado en la Figura 11. Es económica y de instalación rápida, permite proteger el desgaste de la tubería de revestimiento (Ortega, 2016).

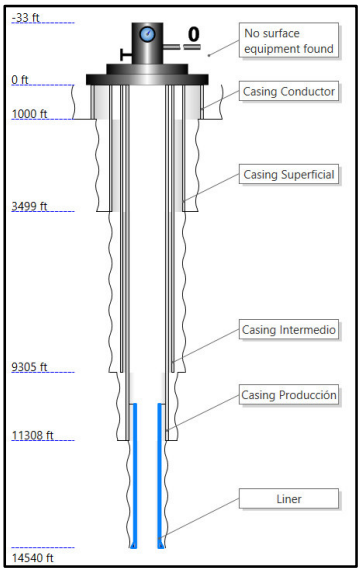


Figura 11. *Liner.*
Fuente: Software Pipesim 2019.1-Schlumberger

Tubing: Se denomina tubería de producción, tiene un diámetro más pequeño y se introduce por el casing (Steel, 2018). En la Figura 12 se muestra el esquema.

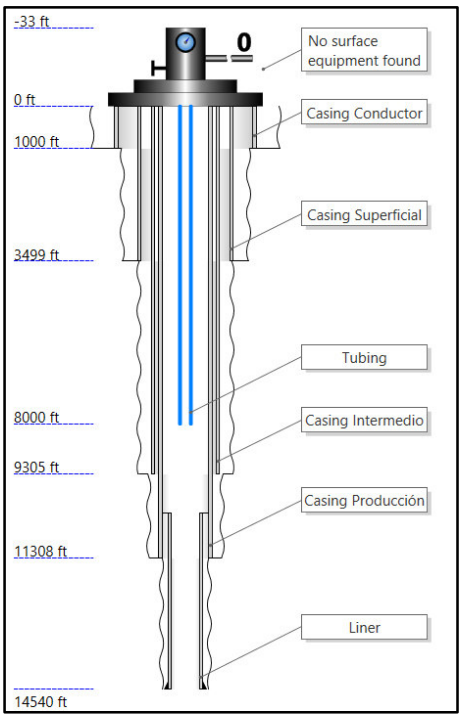


Figura 12. *Tubing.*
Fuente: Software Pipesim 2019.1-Schlumberger

2.1.3 Levantamiento Artificial por Bombeo Electrosumergible

Es utilizado para extraer grandes cantidades de fluido en pozos profundos. Vargas (2008) menciona que el sistema posee una ventaja al tener pocas instalaciones de superficie, por lo que, puede trabajar en áreas urbanas; además es utilizado en condiciones de pozo poco favorables como: abrasivos, corrosivos, altas temperaturas, alta viscosidad en los fluidos, elevadas cantidades de gas, pozos direccionales y horizontales (dependiendo del ángulo de construcción, generalmente < 9°/100 ft).

Su funcionamiento es mediante la rotación centrífuga de la bomba electrosumergible, un motor eléctrico ubicado en el fondo del pozo proporciona la potencia que necesita la bomba; la corriente eléctrica del motor es suministrada y controlada mediante variadores de velocidad instalados en la superficie del pozo, la corriente es conducida a través del cable de potencia hasta el motor (Vargas, 2008).

En la Figura 13 se puede observar los componentes del levantamiento artificial por bombeo electrosumergible.

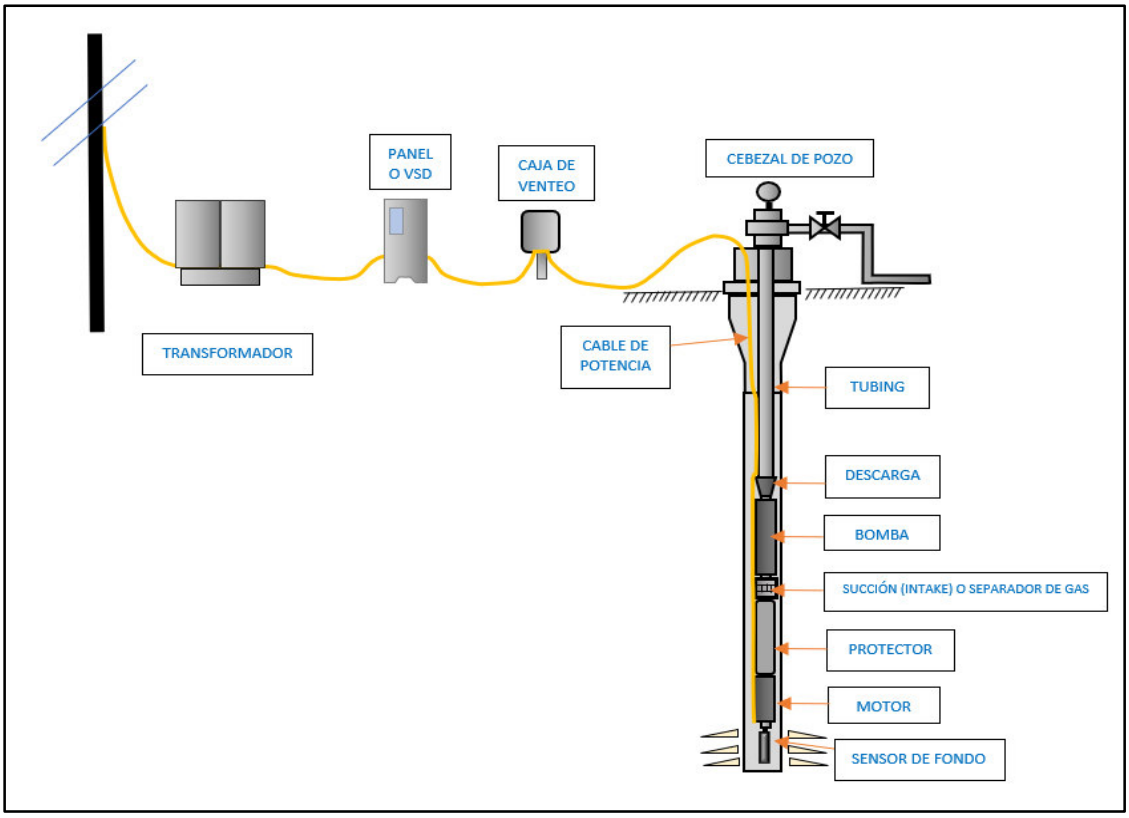


Figura 13. *Esquema del sistema de bombeo Electrosumergible.*
Elaborado por: Lesly Rivera

2.1.4 Permeabilidades Relativas

La permeabilidad relativa permite comparar dos fluidos inmiscibles en un medio poroso, ya que un fluido en presencia de otro inhibe el flujo, es decir es usada para describir de manera cuantitativa el movimiento simultáneo de dos o más fluidos inmiscibles a través de una roca porosa (Schlumberger, 2022).

Se obtiene mediante la relación que existe entre la permeabilidad efectiva y la permeabilidad absoluta (Madrid, 2019). Se calcula mediante la siguiente ecuación:

$$Kr = \frac{Ke}{K} \qquad \textit{Ecuación 1}$$

Donde:

Kr = Permeabilidad relativa

Ke = Permeabilidad efectiva

K = Permeabilidad absoluta

2.2 Software y Elementos de Programación

2.2.1 Software PIPESIM

Por más de 30 años, el simulador PIPESIM ha venido evolucionando continuamente con las últimas innovaciones que se presentan en la industria del petróleo y gas (Schlumberger S. , 2008), además que la interfaz cada vez es más amigable con el usuario como se puede ver en la Figura 14.

Las últimas versiones del simulador ayudan a la integración digital, generando un ahorro de tiempo en los ingenieros, todos los procesos manuales se los puede realizar de manera rápida y eficiente mediante codificación.

PIPESIM no sólo modela el flujo multifásico desde el yacimiento hasta el cabezal del pozo, sino que además tiene en cuenta el desempeño de la línea de flujo y de las instalaciones de superficie para proveer un análisis integral del sistema de producción (Schlumberger S. , 2008).

Schlumberger (2009) menciona que el software PIPESIM permite a los usuarios:

- Generar un análisis nodal integral en cualquier punto de un sistema de producción utilizando múltiples parámetros de sensibilidad.
- Diseñar pozos nuevos y analizar los pozos verticales, horizontales y multilaterales existentes.
- Diseñar sistemas de levantamiento artificial con BES para optimizar la producción.
- Generar una simulación con el perfil de Presión/Temperatura.
- Conectarse a OFM para identificar los candidatos de un campo para estudios adicionales o tratamientos con fines de remediación.

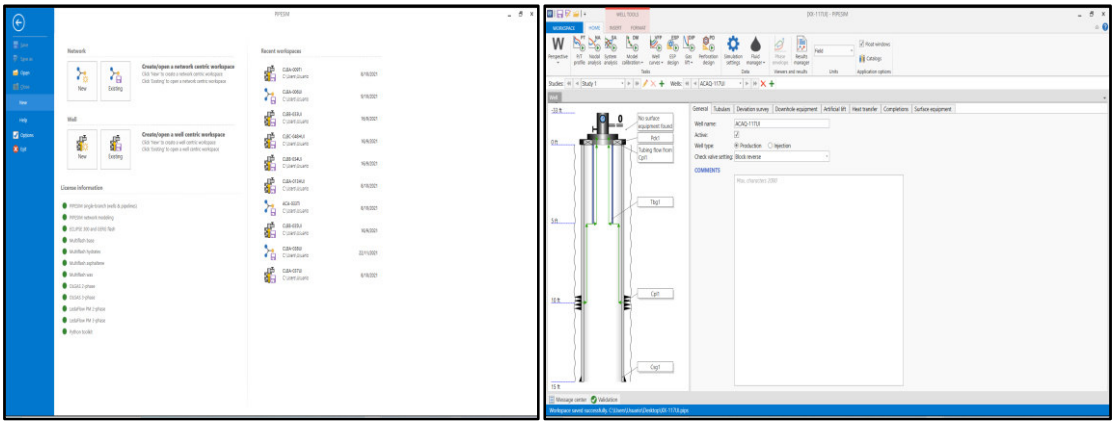


Figura 14. *Interfaz de PIPESIM 2019.1*
Fuente: Software PIPESIM 2019.1-Schlumberger

2.2.2 Programación

Es el conjunto de instrucciones consecutivas y ordenadas que llevan a la computadora a ejecutar una tarea específica (Robles, 2021). Con el pasar de los años, la programación ha tomado más fuerza ya que tareas que eran repetitivas y rutinarias se las puede reemplazar mediante la automatización de los procesos utilizando la programación.

Robles (2021) expone que la lógica de la programación está conformada por tres elementos sustanciales que son: la creatividad, la lógica y el razonamiento; estos elementos ayudan a crear líneas de código que a su vez facilitan los procesos que se ha venido trabajando por años en la industria del Petróleo y Gas.

La programación requiere de una visión general de un problema y la solución que se le pueda dar utilizando la computadora, para ello Robles (2021) define las fases que se debe seguir para programar, las mismas que son:

- **Análisis del problema:** Esta fase comprende en analizar la problemática a la que se le va a dar solución, se debe identificar todos los recursos con los que se cuenta (tecnológicos, humanos, entre otros). En este punto se define como trabaja el programa y se identifica los requerimientos del usuario.
- **Diseño del algoritmo:** Aquí se ejecuta la visualización gráfica que permita mostrar la solución del problema; se lo puede desarrollar mediante un diagrama de flujo; donde se identifican los datos de entrada, el proceso de ejecución y la salida o resultado que se va a obtener con el diseño.

- **Codificación:** En este punto se puede elegir el lenguaje de programación que desee trabajar; debe seguirse la metodología adecuada para ejecutar la codificación.
- **Ejecución y prueba:** En esta fase se realizan todas las pruebas necesarias para garantizar el buen funcionamiento del programa.

2.2.3 Interfaz Enthought Canopy

Enthought Canopy es un completo entorno de análisis basado en Python para científicos, ingenieros y analistas. Es de fácil instalación y es compatible con los paquetes de Python, lo cual genera una rápida recopilación, manipulación, análisis y visualización de datos. Con esta herramienta la creación de secuencias de comandos se vuelve más sencilla (Quintagroup, 2020).

Quintagroup (2020) da a conocer las características principales que Enthought Canopy incluye en su interfaz, como son:

- Un editor de texto avanzado con resaltado de sintaxis, autocompletado de código Python y verificación de errores.
- Implementación de paquetes de Python con un administrador de paquetes gráfico, que también notifica las actualizaciones, ayuda a revertir las versiones de los paquetes e informa de errores.
- Práctico navegador de documentación con guía del usuario y ejemplos de código.

2.2.4 Lenguaje de Programación Python

Fue creado por Guido van Rossum, al final de la década de los 80 (Python, 2020). El lenguaje de programación ha tomado más fuerza en los últimos años ya que permite procesar fácilmente todo tipo de estructuras de datos, tanto de texto como numéricos.

Es utilizado en universidades y empresas ya que es fácil, amigable y de uso libre; Python se puede usar en varios dispositivos y sistemas operativos ya que se han creado intérpretes para Unix, Linux, Windows y sistemas Mac Os (Mochales, 2020).

Machuca (2021) expone que Python posee tipado dinámico, permitiéndole ser más amigable al momento de definir las variables que se van a utilizar en la codificación, no se necesita decir de que tipo son los datos (si es *string, int, float,* etc.) facilitando la creación de software.

Gracias a la aceptación y popularidad que tiene el lenguaje, se puede contar con una gran variedad de funciones y librerías que se encuentran ya desarrolladas, por ende, podemos acceder a la extensa biblioteca existente. Posee soporte para la múltiple variedad de bases de datos y es de libre distribución (Python, 2020).

2.2.5 Librerías de Python

Las librerías son fuentes con varias funcionalidades. Los desarrolladores evitan escribir el código para sí mismos al utilizar el código precisamente escrito y bien definido de diferentes librerías. Ayudando a desarrollar el código de manera eficiente y ahorrando tiempo al no tener que escribir un *script* desde cero (Laura, 2021).

Laura (2021) menciona algunas de las librerías que son utilizadas para el cálculo numérico y análisis de datos, como son:

- **NumPy:** Proporciona una estructura de datos universal que permite el análisis he intercambio entre distintos algoritmos. Las estructuras de datos que implementa son vectores multidimensionales y matrices con capacidad para gran cantidad de datos (Laura, 2021).
- **SciPy:** Proporciona rutinas numéricas eficientes, fáciles de usar y opera en las mismas estructuras de datos proporcionadas por NumPy. Sus módulos proporcionan funcionalidad para resolver el cómputo de tareas analíticas y científicas como: integración numérica, optimización, interpolación, transformadas de Fourier, álgebra lineal, estadística, etc. (Laura, 2021).
- **Pandas:** Las estructuras de datos principales en pandas son series para una dimensión y DataFrame para dos dimensiones. Son las estructuras de datos más usadas en muchos campos tales como finanzas, estadística y muchas áreas de

ingeniería. Pandas destaca por lo fácil y flexible que hace la manipulación y el análisis de datos (Laura, 2021).

- **Xlwings:** es una librería diseñada para el análisis de información avanzado a base del lenguaje de programación Python y, por supuesto, compatible con Excel (Odin, 2020).

2.2.6 PythonToolkit (PTK)

PythonToolkit (PTK) es un entorno interactivo para Python. Originalmente fue diseñado para proporcionar un entorno similar a Matlab para científicos e ingenieros cuando se usa junto con los paquetes de Python: numpy, scipy y matplotlib (SOURCEFORGE, 2021). Sin embargo, también se puede utilizar como un entorno interactivo de propósito general, especialmente para la programación de interfaz gráfica de usuario interactiva.

Está construido alrededor de una ventana de consola, un editor de código y un sistema de complementos de herramientas para que se puedan agregar fácilmente características adicionales de Python (SOURCEFORGE, 2021).

2.3 Diagrama de Flujo para Algoritmos / Programación

Son útiles para escribir un programa o algoritmo como ayuda a la explicación que se debe dar a otras personas respecto al desarrollo y funcionamiento del código. Se puede usar para explicar detalladamente la lógica detrás de un programa antes de empezar a codificar el proceso automatizado. Puede ayudar a organizar una perspectiva general y ofrecer una guía cuando llega el momento de codificar. (Lucidchart, 2020).

Con los diagramas de flujo se puede:

- Mostrar la organización en la que se ejecutó el código.
- Visualizar la ejecución del código dentro de un programa.
- Mostrar la estructura de aplicación.
- Comprender cómo los usuarios van a utilizar el código dentro del programa.

3. METODOLOGÍA

Previo a la codificación se realizó una recopilación de los datos que se tiene de los 18 pozos que conforman un campo de la Región Amazónica del Ecuador; los modelos de choke eran realizados manualmente en PIPESIM 2012.3.

La versión 2012.3 al no poseer librerías para codificar, fue necesario migrar los modelos a PIPESIM 2019.1; en donde se realizó codificación para crear los modelos desde cero y generar el modelo de choke totalmente automatizado. A continuación, en la Figura 15 se detalla todo el procedimiento que se empleó para tener una automatización completa del modelo.

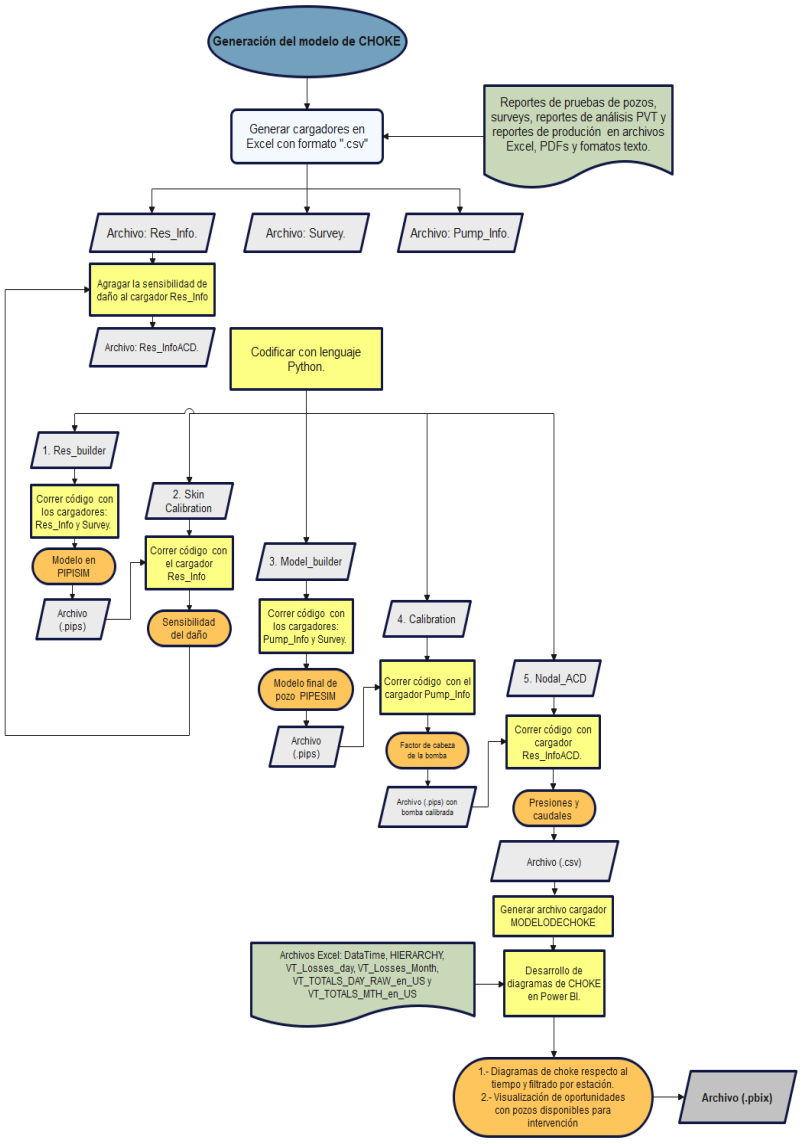


Figura 15. *Diagrama de flujo de la metodología aplicada para generar el modelo de choke. Elaborado por: Lesly Rivera*

3.1 Desarrollo del Modelo de Pozo Base en PIPESIM.

Se realiza la codificación para generar el modelo de pozo en PIPESIM mediante la base de datos (Res_Info & survey); el flujo de trabajo se describe en la Figura 16.

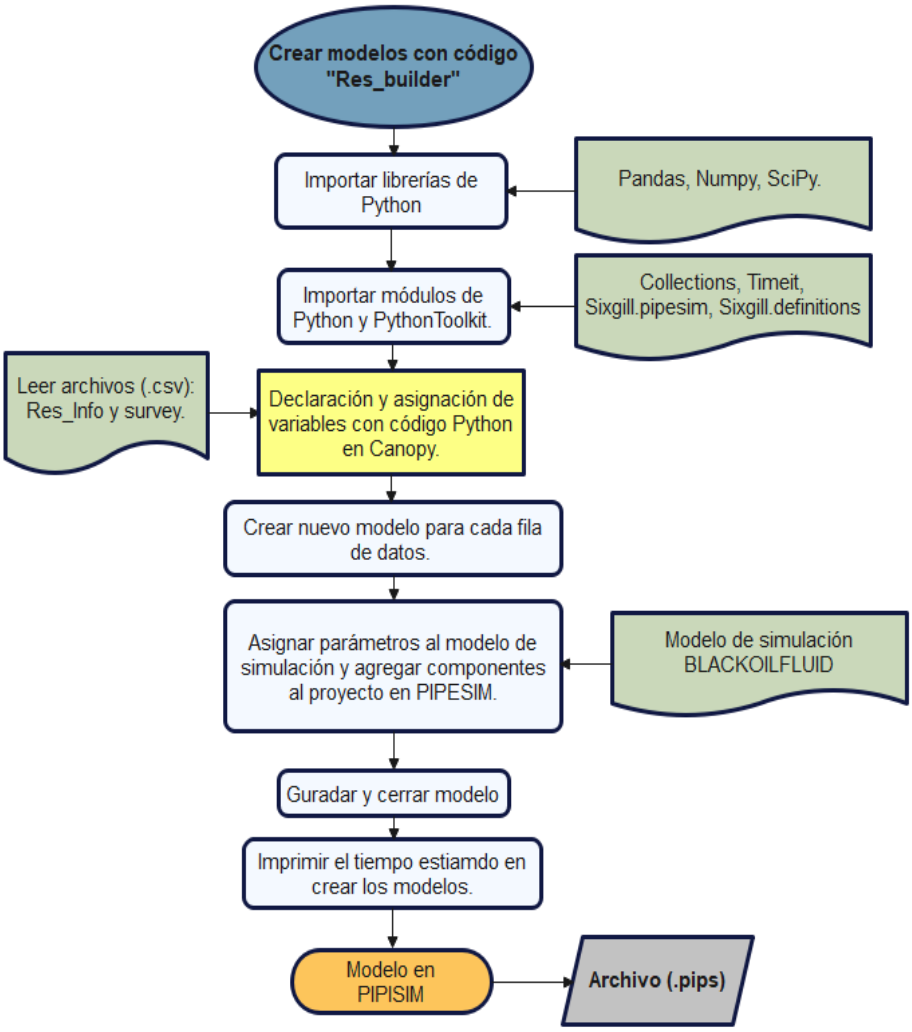


Figura 16. *Diagrama de flujo Res_builder para elaborar el modelo de pozo base en PIPESIM. Elaborado por: Lesly Rivera*

Las bases de datos (Res_Info & survey) son generadas en un archivo Excel con la información que se posee de cada pozo, se genera el código “Res_builder” para crear el modelo base en PIPESIM. El código no se encuentra adjunto por temas de confidencialidad de la empresa.

En la Tabla 1 se puede ver los datos de reservorio y datos estándar de completación de cada pozo los mismos que se encuentran generados en un archivo llamado “Res_Info”.

Tabla 1. *Datos de reservorio y completación generados en el archivo Res_Info.*
Elaborado por: Lesly Rivera

WELL	CASING_START	CASING_END	CASING_ID	CASING_WT	CASING_DIAM	LINER_START	LINER_END	LINER_ID	LINER_WT	LINER_DIAM	TUBING_START	TUBING_END	TUBING_ID	TUBING_WT	ESP_TYPE
AB-057HI	0	15	6.28	0.36	10						0	5	2.99	0.25	
AC-063TI	0	15	6.28	0.36	10						0	5	2.99	0.25	
AC-127TI	0	15	6.28	0.36	10						0	5	2.99	0.25	
AC-165TI	0	15	6.28	0.36	10						0	5	2.99	0.25	
AC-170TI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XA-009TI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XA-010HS1UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XA-013HUI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XA-037UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XB-012UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XB-021UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XB-033UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XB-035UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XC-043HUI	0	15	6.28	0.36	10						0	5	2.99	0.25	
YA-015UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
YD-023HUI	0	15	6.28	0.36	10						0	5	2.99	0.25	
Z-003UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
Z-004TI	0	15	6.28	0.36	10						0	5	2.99	0.25	

WELL	ESP_STAGES	CPL_START	CPL_PR	CPL_TR	CPL_THICKNESS	CPL_K	CPL_S	LIQUID	WCUT	GOR	API	FREQ	WHP	PWF	BRCH	PWF_OBJAVOCET
AB-057HI		10	1700	236	10	614	0	755	74	9.02	27.6	77.5	110	741.31	AB-057HI - Wellhead	450
AC-063TI		10	1490.51	220	55.946	519.24	0	1944	89	229	22.4	59	20	838.46	AC-063TI - Wellhead	676.26
AC-127TI		10	1481	220	60.337	83	0	1334	88	229.01	23.9	61	30	476.25	AC-127TI - Wellhead	620
AC-165TI		10	1427.21	220	73.358	113.29	0	780	86	229.03	23.3	56	30	654.16	AC-165TI - Wellhead	477.43
AC-170TI		10	1478.77	220	78.99	160	0	1446	87	229.01	22.1	57	20	661.89	AC-170TI - Wellhead	779.3
XA-009TI		10	2064	226	8	55	0	174	3	150	26.3	48.5	60	376.88	XA-009TI - Wellhead	450
XA-010HS1UI		10	1387.94	220	28	254	0	584	24	163	18.5	57.6	120	443.1	XA-010HS1UI - Wellhead	510
XA-013HUI		10	1431.4	220	53	213	0	1099	1	150	19.8	57	200	556.81	XA-013HUI - Wellhead	550
XA-037UI		10	1551.98	220	30.91	77	0	228	36	160	19.1	61	105	291.15	XA-037UI - Wellhead	287
XB-012UI		10	1343.15	220	46.25	330	0	1365	65	160	20.1	61	25	384.7847534	XB-012UI - Wellhead	330
XB-021UI		10	1283.91	220	36	125.23	0	406	90	160	17.8	57	260	340.0720655	XB-021UI - Wellhead	300
XB-033UI		10	1095.7	220	65.37	482	0	1060	48	160	18.6	51	25	514.7659761	XB-033UI - Wellhead	280
XB-035UI		10	1249.52	220	40	65	0	145	3	160	18.5	52	100	291.3196234	XB-035UI - Wellhead	240
XC-043HUI		10	1500	220	31.48	481.68	0	1720	64	163	17.8	56	160	1101.26	XC-043HUI - Wellhead	1185
YA-015UI		10	1500	224	37	155	0	340	64	146.98	18.7	57.5	130	370.95	YA-015UI - Wellhead	380
YD-023HUI		10	1201.41	224	36.62	1116	0	2040	44	147	18.5	64.7	200	692.1	YD-023HUI - Wellhead	700
Z-003UI		10	1820	220	33.5	90	0	162	3	145.98	18.3	43	30	415.16	Z-003UI - Wellhead	275
Z-004TI		10	1729	226	19.76	150	0	444	6	123.01	27.1	54.5	150	306.44	Z-004TI - Wellhead	450

En la Tabla 2 se puede visualizar el cargador “survey” que cuenta con datos de las trayectorias de cada pozo (MD & TVD).

Tabla 2. *Datos de trayectoria generados en el archivo survey*
Realizado por: Lesly Rivera

XA-009TI_MD	XA-009TI_TVD	XA-010HS1UI_MD	XA-010HS1UI_TVD	XA-013HUI_MD	XA-013HUI_TVD	XA-037UI_MD	XA-037UI_TVD	XB-012UI_MD	XB-012UI_TVD	XB-021UI_MD	XB-021UI_TVD	XB-035UI_MD	XB-035UI_TVD	XB-033UI_MD	XB-033UI_TVD	XC-043HUI_MD	XC-043HUI_TVD
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	46.99999053	100	99.99983551	50	49.9998284	36.8	36.8	216	215.9817516	100	100	36.6	36.6	18.3	18.3	28.5	28.5
533	532.9859825	200	199.9994392	100	99.99924322	100	100	380	379.9006175	200	200	100	100	36.6	36.6	64.25	64.25
730	729.8591321	300	299.9980268	121.39	121.3889962	200	200	436	435.8499757	300	299.99	200	200	68.3	68.3	100	100
821	820.6293189	400	399.9972063	150	149.9987843	240	240	528	527.6649316	400	399.99	300	299.99	100	100	150	149.99
912	911.2232594	500	499.9956213	200	199.9984432	300	300	619	618.328612	500	499.99	400	399.99	150	150	200	199.98
1006	1004.559605	600	599.9939775	250	249.9981138	400	399.99	742	740.4910982	600	599.99	500	499.98	200	200	250	249.97
1100	1097.572802	700	699.9919915	300	299.9976322	500	499.99	833	830.3394795	700	699.99	600	599.98	250	249.99	300	299.97
1195	1191.173472	767	766.9884506	350	349.9968564	600	599.99	917	912.7014754	800	799.99	700	699.98	300	299.99	350	349.96
1289	1283.398039	885	884.9863012	400	399.9962653	700	699.99	1011	1004.130386	900	899.99	800	799.95	350	349.99	400	399.96
1384	1376.316508	976	975.9696776	500	499.977283	800	799.98	1105	1094.585958	1000	999.99	900	899.86	400	399.98	450	449.96
1479	1468.756922	1070	1069.908528	600	599.8832645	900	899.98	1199	1183.859997	1100	1099.99	1000	999.67	450	449.98	500	499.96
1573	1559.639068	1164	1163.776742	700	699.6871873	1000	999.97	1293	1271.721173	1200	1199.99	1100	1099.37	500	499.98	550	549.96
1668	1650.868282	1259	1258.552811	800	799.3062604	1100	1099.97	1388	1358.788063	1300	1299.99	1200	1198.96	550	549.97	600	599.96
1762	1740.250514	1353	1352.263743	900	898.6860186	1200	1199.96	1482	1443.156112	1400	1399.99	1300	1298.42	600	599.97	650	649.96
1857	1829.496696	1448	1446.878365	1000	997.7511313	1300	1299.95	1572	1522.184411	1500	1499.99	1400	1397.61	650	649.97	700	699.96
1951	1916.639097	1543	1541.332233	1100	1096.334539	1400	1399.94	1577	1526.523887	1600	1599.99	1500	1496.49	700	699.96	750	749.96
2046	2003.745889	1637	1634.568284	1200	1194.374866	1500	1499.94	1673	1609.737607	1698.85	1698.84	1600	1594.91	750	749.96	800	799.96
2141	2090.50576	1732	1728.608837	1300	1291.867494	1600	1599.92	1769	1693.845317	1793.41	1793.39	1700	1692.71	800	799.96	850	849.96
2236	2177.403438	1826	1821.633771	1400	1389.178055	1700	1699.88	1865	1778.568596	1883.18	1883.07	1800	1789.88	850	849.96	900	899.96
2330	2263.412879	1921	1915.68598	1500	1486.855357	1800	1799.83	1961	1862.433418	1980.65	1980.27	1900	1886.4	900	899.96	950	949.96
2425	2350.24348	2015	2008.707523	1570	1555.517064	1829	1828.81	2056	1944.205952	2076.83	2075.92	2000	1982.2	950	949.96	1000	999.96
2519	2436.156751	2110	2102.728916	1681.64	1665.387623	1900	1899.77	2152	2026.216664	2169.6	2167.89	2100	2077.12	1000	999.96	1050	1049.96
2614	2523.057804	2205	2196.852422	1776.84	1759.367804	2000	1999.76	2248	2109.0066	2268	2265.14	2200	2171.09	1050	1049.96	1100	1099.96
2708	2609.077233	2300	2291.027801	1870.42	1851.89088	2100	2099.75	2341	2189.279583	2363.85	2359.55	2300	2264.34	1100	1099.96	1150	1149.96
2803	2696.085109	2394	2384.253364	1961.88	1942.418214	2200	2199.71	2435	2269.806845	2457.08	2451.03	2400	2356.7	1150	1149.96	1200	1199.96

YA-015UI_MD	YA-015UI_TVD	YD-023HUI_MD	YD-023HUI_TVD	Z-003UI_MD	Z-003UI_TVD	Z-004TI_MD	Z-004TI_TVD	AB-057HI_MD	AB-057HI_TVD	AC-063TI_MD	AC-063TI_TVD	AC-127TI_MD	AC-127TI_TVD	AC-165TI_MD	AC-165TI_TVD	AC-170TI_MD	AC-170TI_TVD
0	0	0	1.14E-13	0	0	0	0	0	0	0	-0.000016174	0	-0.000028203	0	0	0	-0.000028203
283	282.99824	37.5	37.5	37	37	37	37	373	372.999	288.0005824	288.0005598	36.0000728	36.0000438	100	100	100.0001987	99.99968352
374	373.997796	100	99.99967508	100	100	100	100	465	464.998	380.0007678	379.9960746	100.0001987	100.0001718	200	200	200.0003975	199.9975642
464	463.9808851	200	199.9965052	200	200	200	200	556	555.997	655.0012924	654.6020621	200.0003975	200.0001887	300	300	300.0006088	299.9919048
554	553.8717388	250	249.9934836	300	300	230	230	646	645.997	732.0014573	731.2471739	300.0006088	299.9999615	400	400	400.000795	399.9817288
644	643.6128822	300	299.9907338	400	400	267	267	736	735.996	825.0016741	823.5500318	400.000795	399.9997037	500	500	500.0009812	499.9715528
736	735.1007525	400	399.9894431	500	500	300	300	828	827.972	917.0018594	914.4137389	500.0009812	499.9990797	600	600	600.0012175	599.9613768
826	824.3798122	500	499.9814889	600	600	400	400	919	918.886	1008.002013	1003.889321	600.0012175	599.997998	700	700	603.0012116	602.9610166
917	914.4849675	600	599.9669351	700	700	500	500	1009	1008.75	1099.002167	1092.897311	700.0014037	699.9976487	800	800	700.0014037	699.9233077
1011	1007.138608	700	699.9509694	800	800	600	600	1101	1100.52	1187.002327	1178.279318	800.00159	799.9974824	900	900	800.00159	799.7836148
1105	1099.107455	800	799.9305185	900	900	700	700	1192	1191.12	1276.002519	1263.827585	900.0018263	899.9971331	1000	1000	883.0017931	882.533292
1199	1190.357735	900	899.9153416	1000	1000	800	800	1286	1284.42	1372.002729	1355.12452	1000.001962	999.9966007	1100	1100	978.0019725	977.0381808
1294	1282.142088	1000	999.9102276	1100	1100	900	900	1380	1377.38	1467.002909	1444.392033	1100.002199	1099.99558	1200	1200	1073.002152	1071.236978
1388	1372.615943	1100	1099.906575	1200	1200	1000	1000	1475	1471.04	1562.003088	1532.780759	1200.002435	1199.993705	1300	1300	1168.002331	1164.909284
1482	1462.278827	1200	1199.902028	1300	1300	1100	1100	1569	1563.27	1657.003268	1620.225759	1278.002581	1277.99203	1400	1400	1263.002511	1258.118943
1577	1551.77288	1300	1299.896375	1400	1399	1200	1200	1663	1654.84	1752.003547	1706.671122	1406.002796	1405.988868	1500	1500	1357.002759	1350.007921
1671	1639.084038	1400	1399.889937	1500	1498	1300	1300	1757	1745.68	1847.003727	1792.415922	1501.002975	1500.982954	1600	1600	1452.002938	1442.503223
1765	1725.424379	1500	1499.881265	1600	1597	1400	1400	1851	1835.86	1943.003937	1879.063434	1596.003154	1595.95104	1700	1700	1547.003118	1534.798452
1859	1811.476147	1600	1599.867588	1700	1695	1500	1500	1945	1925.4	2038.004117	1965.163216	1691.003434	1690.870297	1800	1800	1642.003297	1626.699636
1953	1897.597737	1700	1699.85209	1794	1786	1600	1600	2039	2014.11	2133.004296	2051.608579	1786.003613	1785.71265	1900	1900	1737.003477	1718.180044
2048	1984.377153	1800	1799.836863	1888	1877	1700	1700	2134	2102.63	2229.004507	2138.967397	1880.003761	1879.407417	2000	1999.99	1831.003625	1808.525073
2142	2069.245487	1900	1899.825871	1980	1966	1800	1800	2228	2189.09	2324.004686	2224.712197	1975.003941	1973.919691	2100	2099.94	1926.003804	1899.690295
2236	2153.438733	2000	1999.822395	2077	2057	1900	1900	2322	2274.52	2419.004866	2309.731654	2070.00412	2068.269245	2200	2199.76	2021.003984	1990.738084
2330	2237.820398	2100	2099.819735	2169	2143	2000	2000	2417	2359.79	2514.005045	2394.749891	2166.004331	2163.422023	2300	2299.42	2115.004132	2080.690657
2425	2323.365288	2200	2199.81593	2265	2232	2100	2100	2511	2443.73	2610.005256	2481.034243	2260.004479	2256.383879	2443.44	2441.93	2209.00448	2170.753825
2519	2408.218638	2300	2299.811916	2355	2315	2200	2200	2605	2527.69	2705.005436	2566.419912	2355.004659	2350.086703	2538.74	2536.08	2303.004628	2260.954933

Una vez realizados los cargadores, en la interfaz de Canopy, se empieza a llamar los datos que se encuentran en cada archivo Excel. Se procede a enlazar los paquetes que tiene PIPESIM con Python Toolkit; se importa las librerías de Pandas y el tiempo de ejecución del código. En la Figura 17 a manera de ejemplo se puede observar un fragmento del código.

```
from sixgill.pipesim import Model
from sixgill.definitions import ModelComponents, Parameters, Constants
import pandas as pd
from timeit import default_timer as timer
```

Figura 17. *Librerías de Python y Python Toolkit.*
Fuente: Canopy versión 2.1.9.30

A continuación, se traslada los datos que se tiene en el archivo “Res_Info & survey” hacia el software PIPESIM, esta acción se realiza con la librería de Pandas.

Para empezar a crear el modelo en PIPESIM se define variables que son de ayuda durante toda la programación; ya que hace referencia a los datos que son guardados en cajas llamadas “objetos”, ayudando a recordar la variable utilizada durante toda la codificación.

Las variables que se utilizaron para crear el modelo en PIPESIM son “wellname” que hace referencia al nombre del pozo y “filename” que permitirá abrir y crear el modelo PIPESIM. En la Figura 18 se puede visualizar la corrida del código Res_builder y en la Figura 19 se puede observar los archivos creados en PIPESIM a partir de la ejecución del código.

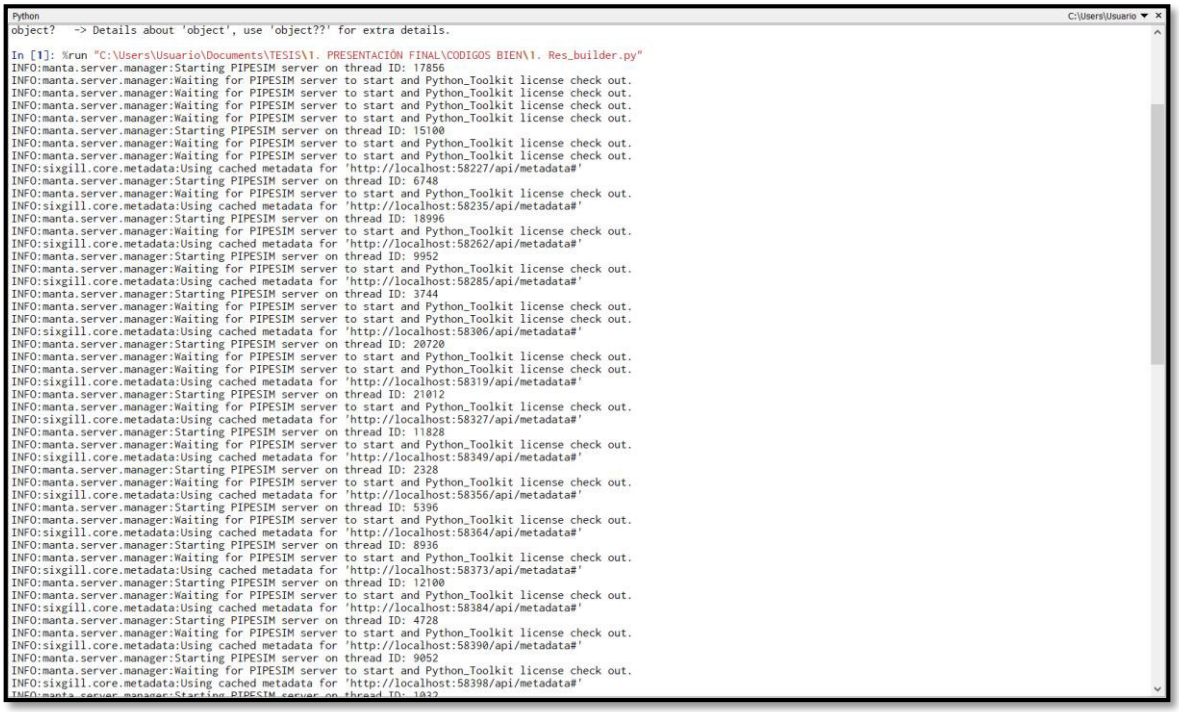


Figura 18. *Corrida del Código para crear los archivos en PIPESIM (.pips)*
Fuente: Canopy versión 2.1.9.30

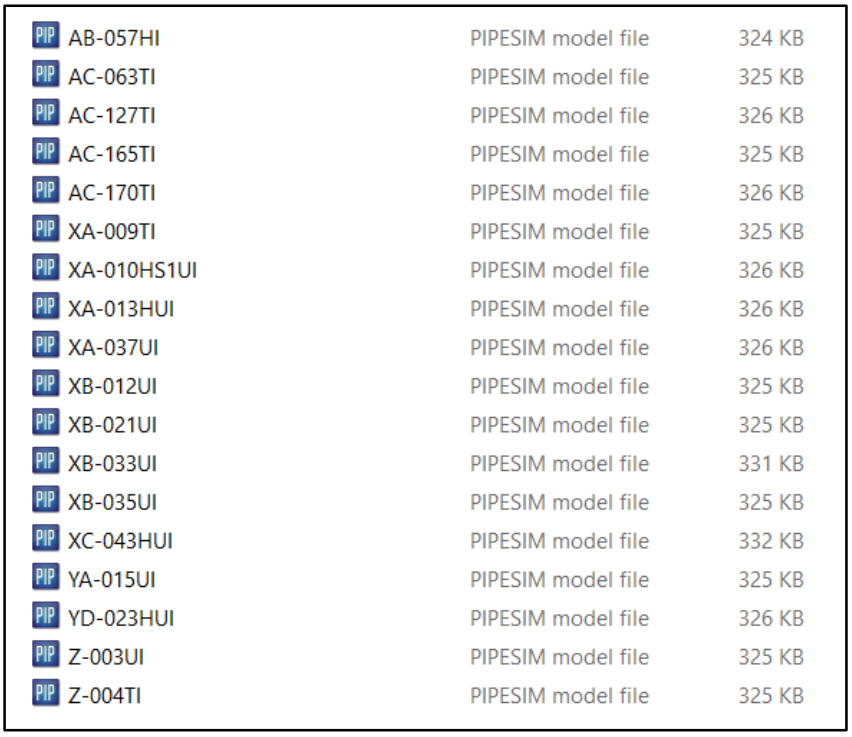


Figura 19. *Archivos PIPESIM (.pips) generados a partir de la corrida del código. Fuente: Software PIPESIM 2019.1-Schlumberger*

Para generar el modelo de pozo en PIPESIM como se muestra en la Figura 20; en el código se debe crear variables que permitan enlazar los datos con los nombres establecidos en el encabezado de las tablas generadas en el archivo Excel (CASING_START, CASING_END, CPL_PR, etc.); estas variables deben ajustarse a los *keywords* de las librerías de Python Toolkit para que el modelo de PIPESIM pueda ser creado y no rebote ningún error al momento de cargar los datos en la interfaz del software.

Una vez definidas las variables a utilizar, se empieza a codificar la construcción del modelo mediante los parámetros establecidos en Python Toolkit, se puede hacer referencia a los parámetros desde el contexto de nivel superior, siempre que sean únicos.

Se utiliza “model.add” para añadir los componentes del modelo como argumentos de la codificación. La trayectoria del pozo se maneja como arreglo, es decir, se asigna a un “DataFrame” que es una estructura de datos con dos dimensiones en la cual se puede guardar distintos tipos (como caracteres, enteros, valores de punto flotante, factores y más) (Carpentries, 2021).

Se cargan los datos de “completion”, “casing”, “packer”, “tubing” y propiedades del fluido con las variables de cada elemento, los datos serán transferidos al *software* mediante “model.add”, generándose de esta manera el modelo de pozo en PIPESIM.

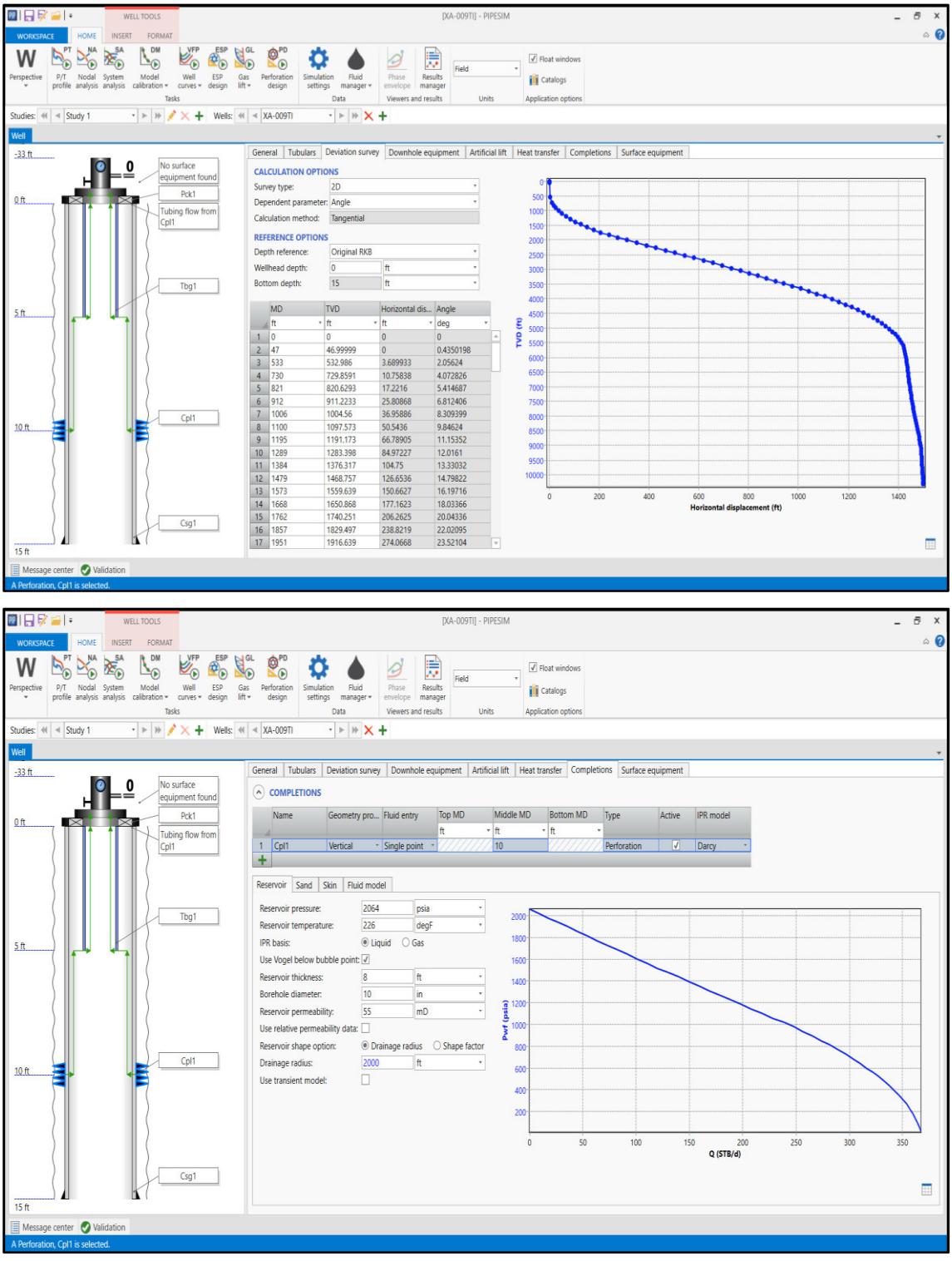


Figura 20. *Creación del modelo de pozo en PIPESIM.*
Fuente: Software PIPESIM 2019.1-Schlumberger

Una vez creado el modelo base en PIPESIM, se procede a ingresar de manera manual los datos de permeabilidades relativas de cada pozo. Las permeabilidades relativas son las más afectadas con el daño de formación por invasión de finos, la geometría de los poros y asociación con el área superficial (Islas, 1991). En la Figura 21 se puede observar el cambio que se da en la tasa de producción (STB/d) cuando se genera la curva IPR con datos de permeabilidades relativas. Finalmente se guarda y cierra el modelo.

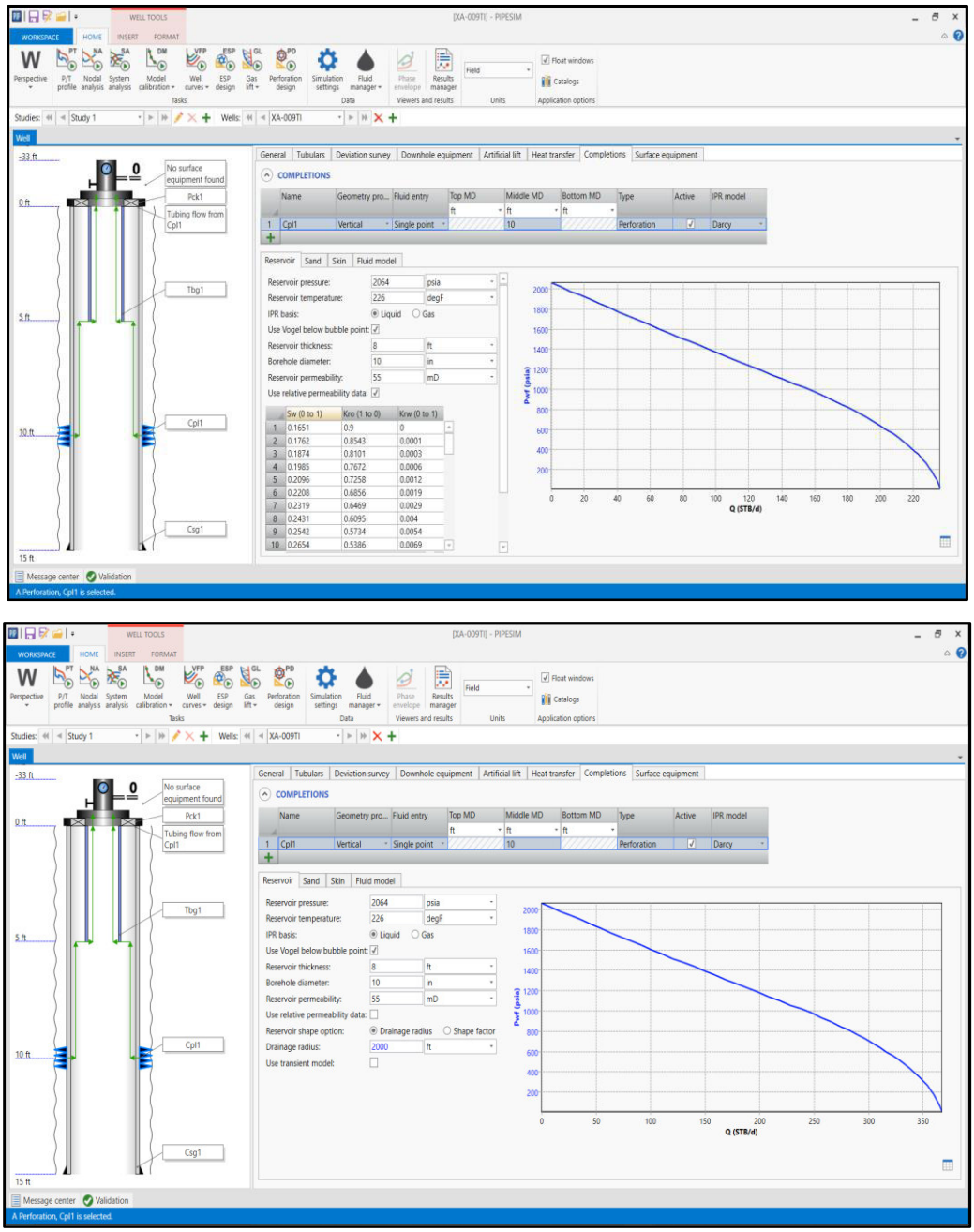


Figura 21. *Generación de la curva IPR con & sin datos de Kr.*
Fuente: Software PIPESIM 2019.1-Schlumberger

3.2 Calibración del Daño de Pozo.

En la Figura 22 se encuentra detallado el flujo de trabajo respecto a la codificación realizada para obtener el valor del daño a las condiciones de reservorio.

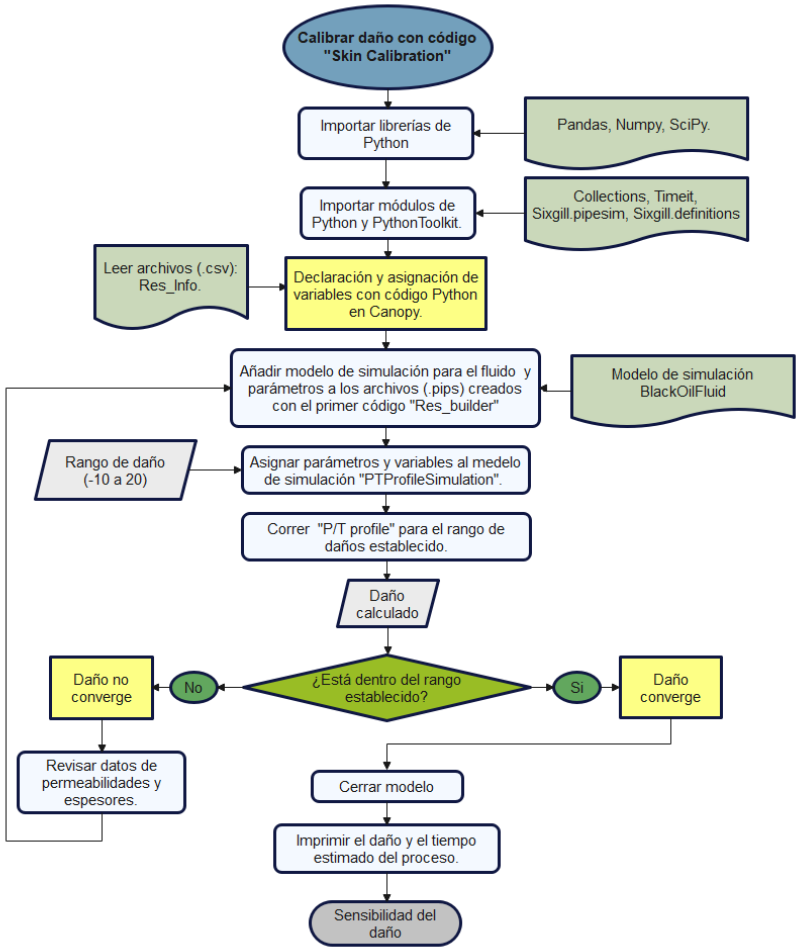


Figura 22. *Diagrama de flujo para el código Skin Calibration.*
Elaborado por: Lesly Rivera

Una vez creado el modelo del pozo y cargadas las permeabilidades relativas, se procede a calibrar el daño de formación. Mediante la elaboración del código “Skin Calibration” se puede realizar la sensibilidad del daño permitiendo obtener un mejor ajuste en el valor obtenido, en base a las condiciones de reservorio que posee cada pozo. El código no se encuentra disponible por cuestiones de confidencialidad con la empresa.

Para comenzar con la codificación en la interfaz de Canopy se debe importar:

- El modelo

- Parámetros
- Variables
- Constantes

Mediante las librerías de Python Toolkit, Pandas, Numpy, y el tiempo de ejecución se genera la codificación de la sensibilidad del daño de formación; además se establece el módulo *"collections"* que permite implementar tipos de datos de contenedores especializados, proporcionando alternativas a los contenedores integrados de uso general de Python (Python Software Foundation, 2022).

Con el módulo *"collections"* se puede importar "defaultdict" que ayuda, a que, si no se encuentra una clave en el diccionario de Python, en lugar de generar un *KeyError* hace de crear una nueva entrada (Python Software Foundation, 2022) permitiendo la generación del modelo al momento de correr el código.

Con "scipy.interpolate" se puede interpolar los datos, que van desde la interpolación unidimensional simple hasta soluciones que son más complejas como interpolación multidimensional (DelfStack, 2020); en la Figura 23 a modo de ejemplo se presenta un fragmento de código de las librerías importadas.

```
from sixgill.pipesim import Model
from sixgill.definitions import Parameters,SystemVariables, ProfileVariables, Constants,ModelComponents
from collections import defaultdict
import pandas as pd
import numpy as np
from scipy.interpolate import interp1d
from timeit import default_timer as timer
```

Figura 23. *Librerías, módulos y funciones importadas al código.*
Fuente: Canopy versión 2.1.9.30

Para realizar la calibración del daño, el código abre el modelo base creado anteriormente y los datos que se encuentran en el archivo de Excel "Res_Info" son codificados con las librerías de Pandas.

Se empieza definiendo variables que son de ayuda durante toda la programación; las variables deben pertenecer al conjunto de componentes del modelo para realizar la simulación con el perfil de presión y temperatura en PIPESIM.

Las variables que se utilizarán en la codificación pertenecen a las propiedades del reservorio, puesto que, para calibrar el daño se debe utilizar los datos de: presión de reservorio, presión de fondo fluyente, presión de cabeza, temperatura de reservorio, corte de agua, frecuencia, y la tasa del líquido.

Una vez definidas las variables se accede a los parámetros “parameters” de los componentes del modelo para generar la simulación con el perfil de presión y temperatura. La trayectoria del pozo se maneja como arreglo siendo asignada a un “DataFrame”.

Posteriormente se realiza la simulación con el comando “PTProfileSimulation”; el código requiere de un rango en que pueda converger el modelo con la calibración del daño, para ello se debe colocar el valor mínimo “MINVALUE” y el valor máximo “MAXVALUE” dentro de las líneas de código, estos valores pueden ser modificados antes de correr el modelo; para definir el rango se deberá tener presente los datos petrofísicos del pozo.

Finalmente se definen las variables del sistema de salida para ejecutar el estudio de simulación de red mediante “parameters”, “profile_variables” y “system_variables”; obteniendo como resultado el valor calibrado de daño a las condiciones de reservorio; en la Figura 24 se muestra la interfaz de Canopy con el valor de cada uno de los daños generados por pozo y en la Figura 25 se muestra el archivo con los datos cargados en PIPESIM a partir de la corrida del código. El modelo es guardado y cerrado.

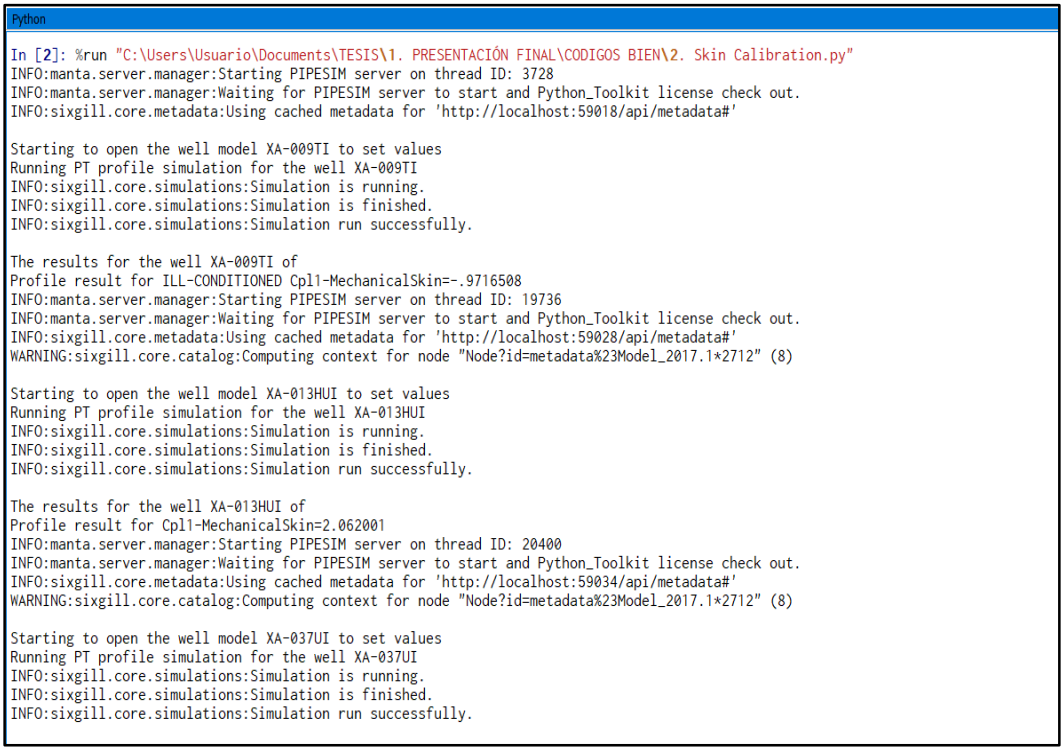


Figura 24. *Daño calibrado, valores generados en Canopy.*
Fuente: Canopy versión 2.1.9.30

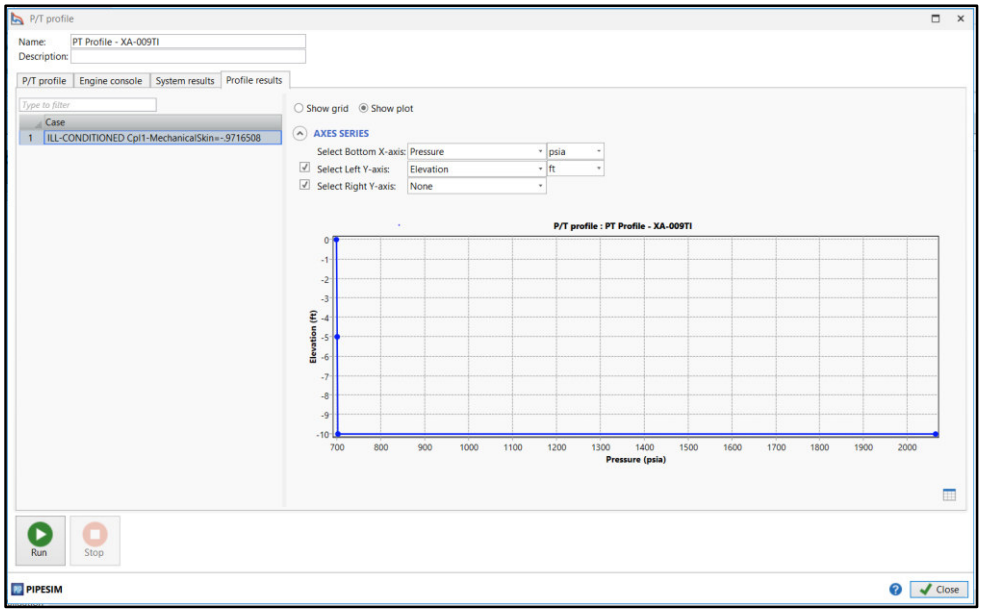


Figura 25. *Daño simulado y calibrado en PIPESIM.*
Fuente: Software PIPESIM 2019.1-Schlumberger

En la Tabla 3 se resume los valores de daño obtenidos con la simulación de cada pozo.

Tabla 3. *Valores de daño obtenidos a partir de la simulación con el perfil Presión/Temperatura.*
Elaborado por: Lesly Rivera

WELL	SKIN
AB-057HI	5.621881
AC-063TI	9.326328
AC-127TI	-1.867956
AC-165TI	5.244057
AC-170TI	3.275128
XA-009TI	-0.9716508
XA-010HS1UI	0.6524745
XA-013HUI	2.062001
XA-037UI	2.279646
XB-012UI	2.060279
XB-021UI	5.250495
XB-033UI	6.01192
XB-035UI	5.091334
XC-043HUI	-3.973143
YA-015UI	9.966104
YD-023HUI	1.13443
Z-003UI	15.71066
Z-004TI	8.368603

3.3 Generación del Modelo de Pozo.

Se genera el modelo de pozo añadiendo la bomba BES, se realiza la codificación para obtener el modelo final del pozo utilizando la base de datos “Pump_Info” & survey”; el flujo de trabajo se describe en la Figura 26.

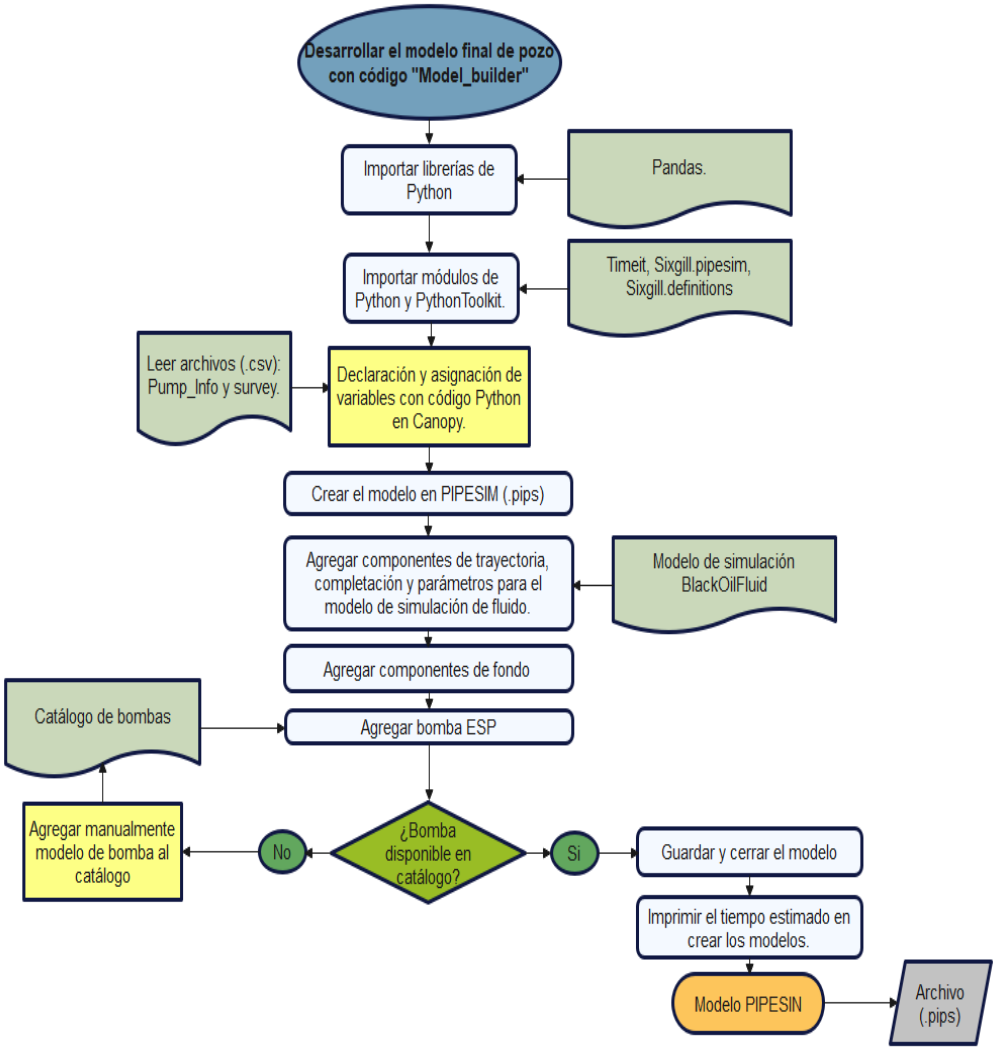


Figura 26. *Diagrama de flujo para crear el modelo final de pozo con bomba BES mediante el código Model_builder.*
Elaborado por: Lesly Rivera

Para generar los pozos con la información de profundidad de asentamiento de la bomba y profundidad de cada elemento del pozo se genera en Excel el cargador de datos “Pump_Info” como se muestra en la Tabla 4. El valor del daño será el obtenido a partir de la corrida del código “Skin Calibration”.

Tabla 4. *Datos de cargador Pump_Info.*
Elaborado por: Lesly Rivera

WELL	CASING_START	CASING_END	CASING_ID	CASING_WT	CASING_DIAM	LINER_START	LINER_END	LINER_ID	LINER_WT	LINER_DIAM	TUBING_START	TUBING_END	TUBING_ID	TUBING_WT	ESP_TYPE
AB-057HI	0	11610	6.276	0.362	8.5						0	10098	2.992	0.254	D460N
AC-063TI	0	9801	8.535	0.545	12.5	9662	1009	6.276	0.362	8.5	0	9439	2.992	0.254	SN3600
AC-127TI	0	10065	8.535	0.545	12.5	9870	1198	6.276	0.362	8.5	0	10245	2.992	0.254	DN1750
AC-165TI	0	11060	8.535	0.545	12.5	10862	946	6.276	0.362	8.5	0	11313.44	2.992	0.254	DN1750
AC-170TI	0	9256	8.535	0.545	12.5	9058	3387	6.276	0.362	8.5	0	11456.51	2.992	0.254	DN1750
XA-009TI	0	10550	6.276	0.362	8.5						0	9676.24	2.992	0.254	D460N
XA-010HS1UI	0	10820	6.276	0.362	8.5	9628	1612	4.276	0.362	6.25	0	8956.2	2.992	0.254	DN1750
XA-013HUI	0	9176	8.535	0.545	12.5	9058	2955	6.276	0.362	8.5	0	8760.93	2.992	0.254	DN1750
XA-037UI	0	11040	8.535	0.545	12.5	10874	714	6.276	0.362	8.5	0	10935.45	2.992	0.254	D460N
XB-012UI	0	10356	8.535	0.545	12.5	10027	938	6.276	0.362	8.5	0	9885.02	2.992	0.254	DN1750
XB-021UI	0	10562	8.681	0.545	12.5	10380	808	6.276	0.362	8.5	0	10459.91	2.992	0.254	D460N
XB-033UI	0	10945	8.681	0.472	12.25	10776	749	6.276	0.362	8.5	0	10109	2.99	0.25	DN1750
XB-035UI	0	11536	8.681	0.545	12.5	11389	583	6.276	0.362	8.5	0	11420	2.992	0.254	D460N
XC-043HUI	0	9874	8.681	0.472	12.25	9643	1217	6.276	0.362	8.5	0	8114	2.99	0.25	DN1750
YA-015UI	0	10445	6.276	0.362	8.535						0	9724	2.992	0.254	D460N
YD-023HUI	0	10215	8.535	0.362	12.5	10056	1454	6.276	0.362	8.5	0	9989	2.99	0.25	SN2600
Z-003UI	0	10082	8.681	0.472	12.25	9944	10782	6.276	0.362	8.5	0	9914	2.992	0.254	D460N
Z-004TI	0	11670	8.681	0.472	12.25	11472	12380	6.276	0.362	8.5	0	11684	2.992	0.254	DN1750

WELL	ESP_STAGES	CPL_START	CPL_PR	CPL_TR	CPL_THICKNESS	CPL_K	CPL_S	LIQUID	WCUT	GOR	API	FREQ	WHP	PWF	BRCH	PWF_OBJAVOCET
AB-057HI	230	10527	1700	236	10	614	5.621881	755	74	9.02	27.6	77.5	110	741.31	AB-057HI - Wellhead	450
AC-063TI	184	10307.5	1490.51	220	55.946	519.24	9.326328	1944	89	229	22.4	59	20	838.46	AC-063TI - Wellhead	676.26
AC-127TI	444	10685	1481	220	60.337	83	-1.867956	1334	88	229.01	23.9	61	30	476.25	AC-127TI - Wellhead	620
AC-165TI	468	11558.5	1427.21	220	73.358	113.289	5.244057	780	86	229.03	23.3	56	30	654.16	AC-165TI - Wellhead	477.43
AC-170TI	468	12172.5	1478.77	220	78.99	160	3.275128	1446	87	229.01	22.1	57	20	661.89	AC-170TI - Wellhead	779.3
XA-009TI	340	9952	2064	226	8	55	-0.9716508	174	3	150	26.3	48.5	60	376.88	XA-009TI - Wellhead	450
XA-010HS1UI	468	11086	1387.94	220	28	254	0.6524745	584	24	163	18.5	57.6	120	443.1	XA-010HS1UI - Wellhead	510
XA-013HUI	468	11416.5	1431.4	220	53	213	2.062001	1099	1	150	19.8	57	200	556.81	XA-013HUI - Wellhead	550
XA-037UI	274	11176.5	1551.98	220	30.91	77	2.279646	228	36	160	19.1	61	105	291.15	XA-037UI - Wellhead	287
XB-012UI	468	10436.5	1343.15	220	46.25	330	2.060279	1365	65	160	20.1	61	25	384.7847534	XB-012UI - Wellhead	330
XB-021UI	288	10756.5	1283.91	220	36	125.2333	5.250495	406	90	160	17.8	57	260	340.0720655	XB-021UI - Wellhead	300
XB-033UI	468	11089	1095.7	220	65.37	482	6.01192	1060	48	160	18.6	51	25	514.7659761	XB-033UI - Wellhead	280
XB-035UI	328	11731.5	1249.52	220	40	65	5.091334	145	3	160	18.5	52	100	291.3196234	XB-035UI - Wellhead	240
XC-043HUI	468	10640	1500	220	31.48	481.68	-3.973143	1720	64	163	17.8	56	160	1101.26	XC-043HUI - Wellhead	1185
YA-015UI	328	9888	1500	224	37	155	9.966104	340	64	146.98	18.7	57.5	130	370.95	YA-015UI - Wellhead	380
YD-023HUI	170	11121	1201.41	224	36.62	1115.95	1.13443	2040	44	147	18.5	64.7	200	692.1	YD-023HUI - Wellhead	700
Z-003UI	492	10193.5	1820	220	33.5	90	15.71066	162	3	145.98	18.3	43	30	415.16	Z-003UI - Wellhead	275
Z-004TI	404	12110	1729	226	19.76	150	8.368603	444	6	123.01	27.1	54.5	150	306.44	Z-004TI - Wellhead	450

El código “Model_builder” empieza con la importación de las librerías de Pandas y Pyhthon Toolkit junto con el tiempo de ejecución; se procede a importar los componentes, parámetros y constantes que serán utilizados más adelante en los *keywords* del modelo.

En la Figura 27 se observa a modo de ejemplo el fragmento de código para importar las librerías. Cabe mencionar que el código no se encuentra adjunto por cuestiones de confidencialidad con la empresa.

```
from sixgill.pipesim import Model
from sixgill.definitions import ModelComponents, Parameters, Constants
import pandas as pd
from timeit import default_timer as timer
```

Figura 27. *Librerías, módulos y funciones importadas de Pandas y Python Toolkit.*
Fuente: Canopy versión 2.1.9.30

Los valores que se encuentran en el archivo Excel son cargados en PIPESIM con la corrida del código mediante las librerías de Pandas, la trayectoria del pozo se maneja como arreglo, es decir, se asigna a un “DataFrame”.

A continuación, se define las variables que se van a utilizar en el código; cada una debe estar asignada a una clase. Se debe acceder a los parámetros de los componentes los mismos que permiten leer y escribir los valores respectivamente en PIPESIM.

Se define los parámetros dentro del modelo utilizando las clases respectivas dependiendo de los valores a ser ingresados; para esta acción se utiliza las siguientes clases:

- **WellTrajectory:** Se refiere a la carga de los datos referentes a la trayectoria del pozo tanto en *measured depth* (MD) como *true vertical depth* (TVD). En la Figura 28 se puede observar los datos cargados en PIPESIM a partir de la ejecución del código.

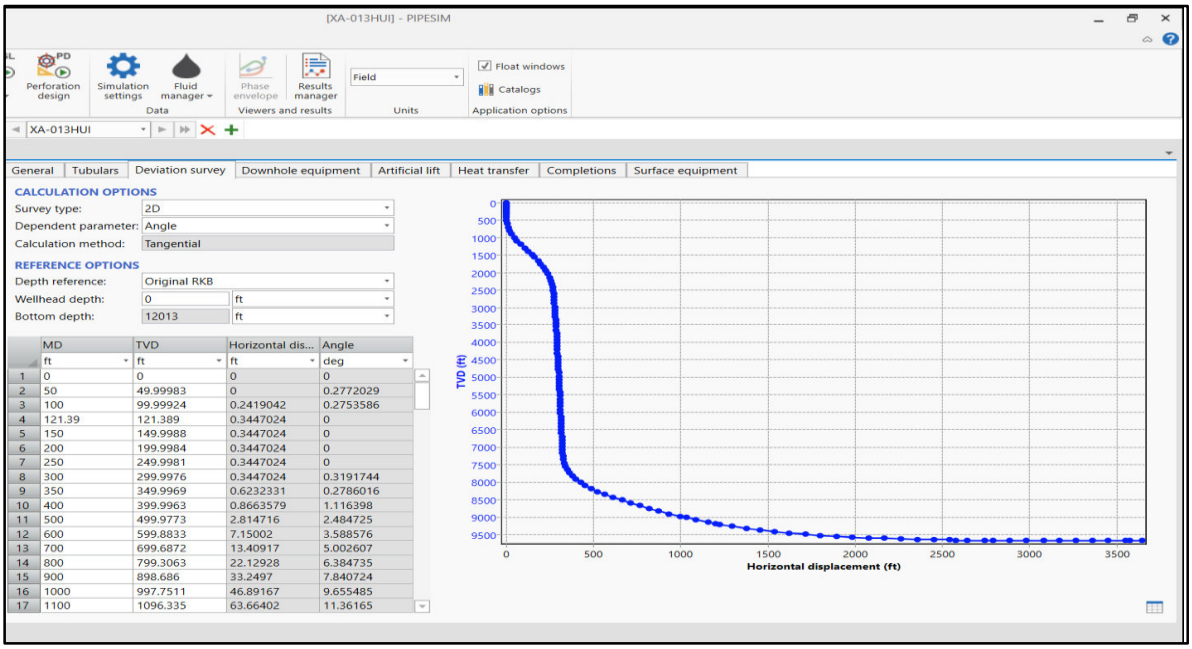


Figura 28. *Datos de Survey cargados en PIPESIM a partir de la ejecución del código.*
Fuente: Software PIPESIM 2019.1-Schlumberger

- **BlackOilFluid:** Carga los valores respectivos de las propiedades del fluido que son: corte de agua (WATERCUT), relación gas-petróleo (GOR) y la gravedad específica del petróleo (API). En la Figura 29 se puede observar los componentes cargados en PIPESIM una vez que se ejecuta el código.

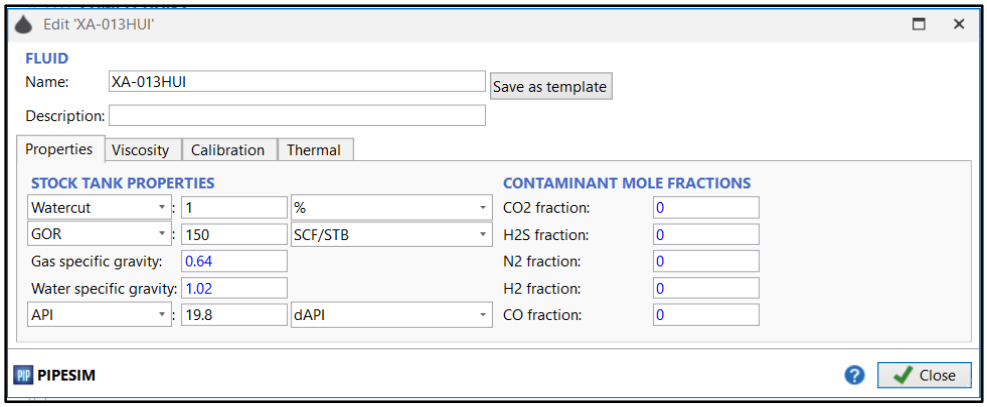


Figura 29. *Componentes de Black Oil Fluid cargados en PIPESIM a partir de la corrida del código. Fuente: Software PIPESIM 2019.1-Schlumberger*

- **Casing:** Carga los valores del casing de producción, para generar el modelo, el software PIPESIM requiere de los siguientes datos: tope, la profundidad total, el diámetro interno, diámetro del hoyo y el espesor. En la Figura 30 se puede observar el diagrama generado en PIPESIM después de ser corrido el código.

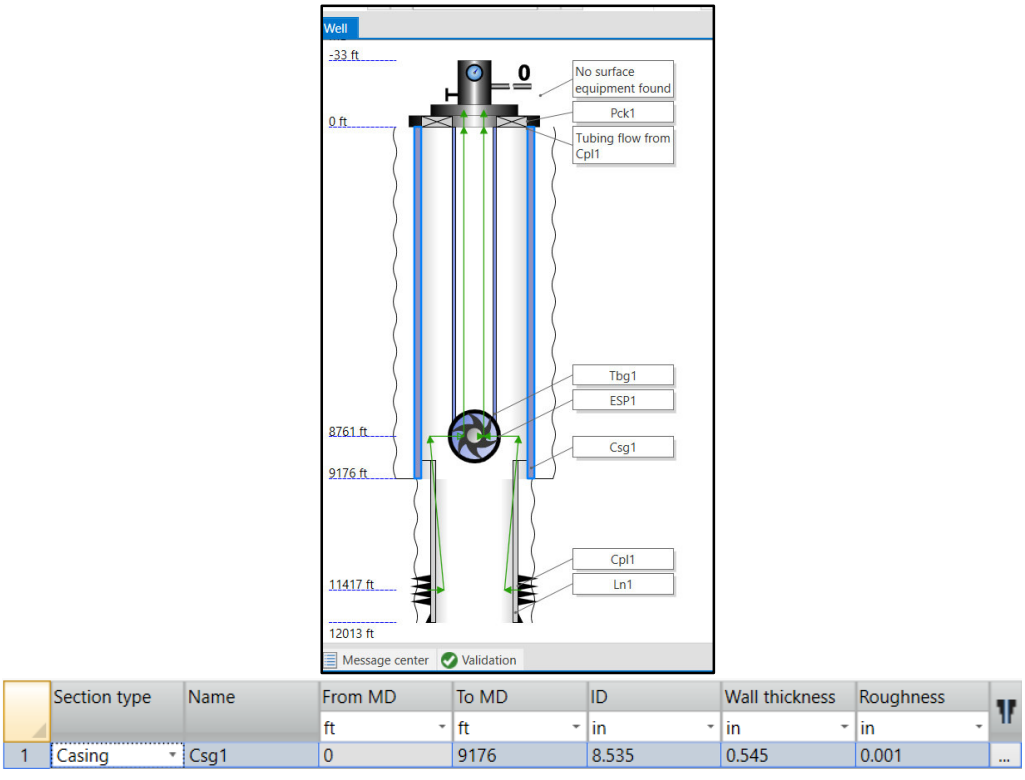
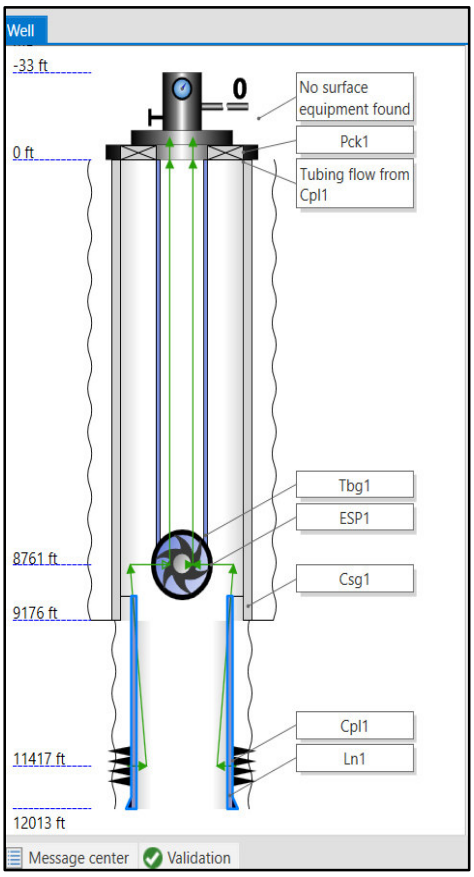


Figura 30. *Casing generado en PIPESIM a partir de la corrida del código. Fuente: Software PIPESIM 2019.1-Schlumberger*

- **Liner:** Carga los valores del liner, para generar el modelo, el software PIPESIM requiere de los siguientes datos: tope, la profundidad total, el diámetro interno,

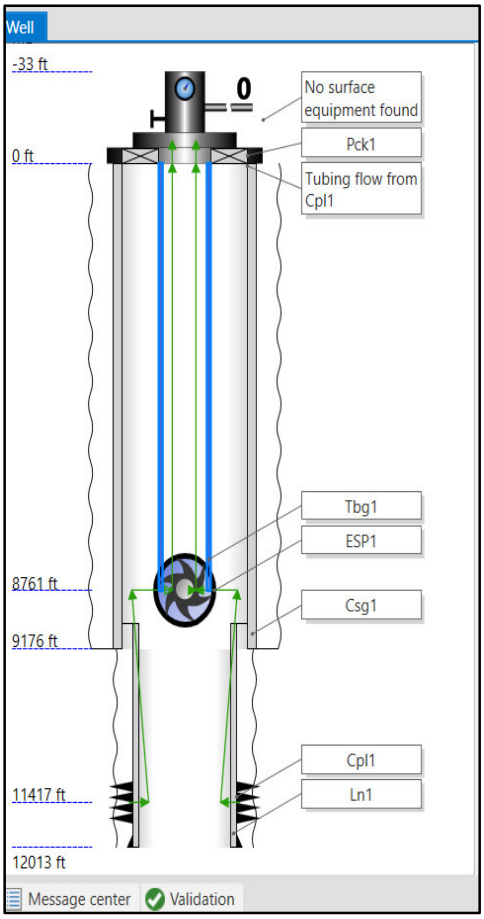
diámetro del hoyo y el espesor. En la Figura 31 se puede observar el diagrama generado en PIPESIM después de ser corrido el código.



	Section type	Name	From MD	To MD	ID	Wall thickness	Roughness	
			ft	ft	in	in	in	
1	Casing	Csg1	0	9176	8.535	0.545	0.001	...
2	Liner	Ln1	9058	12013	6.276	0.362	0.001	...

Figura 31. *Liner generado en PIPESIM a partir de la corrida del código.*
Fuente: Software PIPESIM 2019.1-Schlumberger

- **Tubing:** Carga los valores de tubing, para generar el modelo, el software PIPESIM requiere de los siguientes datos: tope, la profundidad total, el diámetro interno y el espesor. En la Figura 32 se puede observar el diagrama generado en PIPESIM después de ser corrido el código.



	Name	To MD	ID	Wall thickness	Roughness	
		ft	in	in	in	
1	Tbg1	8760.93	2.992	0.254	0.001	...

Figura 32. *Tubing generado en PIPESIM a partir de la corrida del código.*
Fuente: Software PIPESIM 2019.1-Schlumberger

- **BES:** Permite cargar los datos del equipo BES dentro de PIPESIM. El equipo debe tener configurada la bomba con el Fabricante y Modelo para que se pueda identificar la entrada en el catálogo. Fallará si se encuentra más de un elemento en el contexto.

  Los datos que se debe ingresar son: profundidad a la que se encuentra ubicada la BES, tipo de bomba, modelo, frecuencia de operación, factor de cabeza, factor de flujo y número de etapas. En la Figura 33 se puede observar la bomba BES generada en PIPESIM después de ejecutar el código.

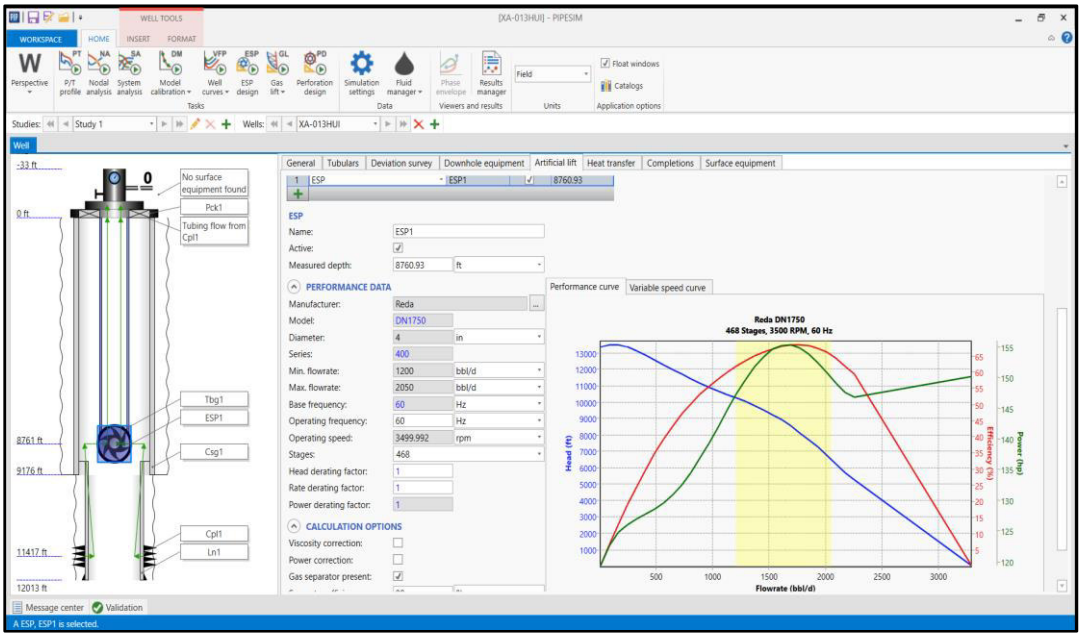


Figura 33. *Equipo BES cargado en PIPESIM a partir de la corrida del código.*
Fuente: Software PIPESIM 2019.1-Schlumberger

- **Completion:** Permite cargar datos de presión y temperatura del reservorio. En la Figura 34 se puede observar los valores en PIPESIM después de ejecutar el código.

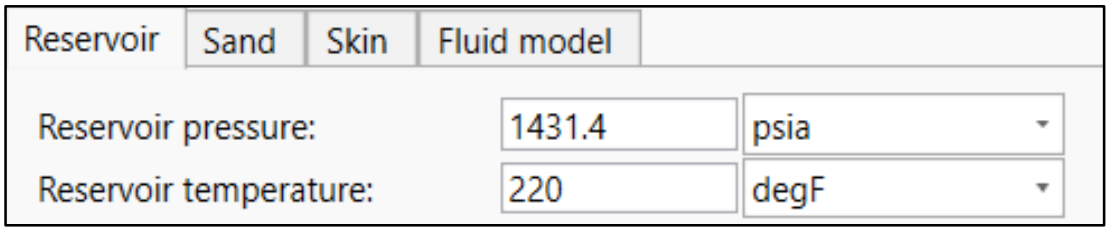
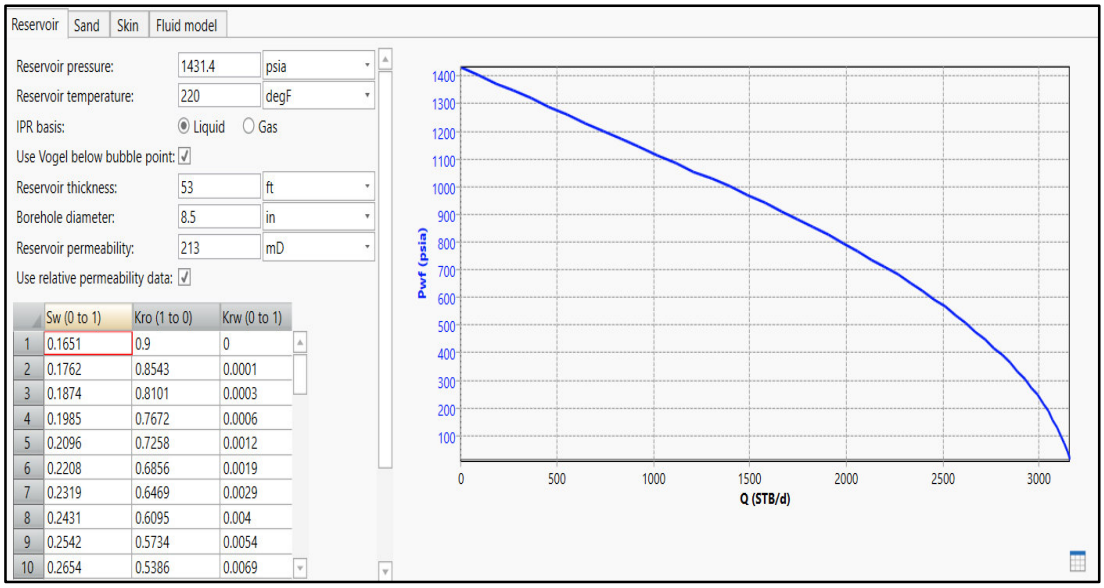


Figura 34. *Carga de los datos de reservorio en PIPESIM a partir de la corrida del código.*
Fuente: Software PIPESIM 2019.1-Schlumberger

- **IPRDarcy:** Se genera la curva IPR del modelo de cada pozo. Los valores que se necesitan son: daño, permeabilidad, espesor de arena, y habilitar la opción de "USEVOGELBELOWBUBBLEPOINT". Ver Figura 35.



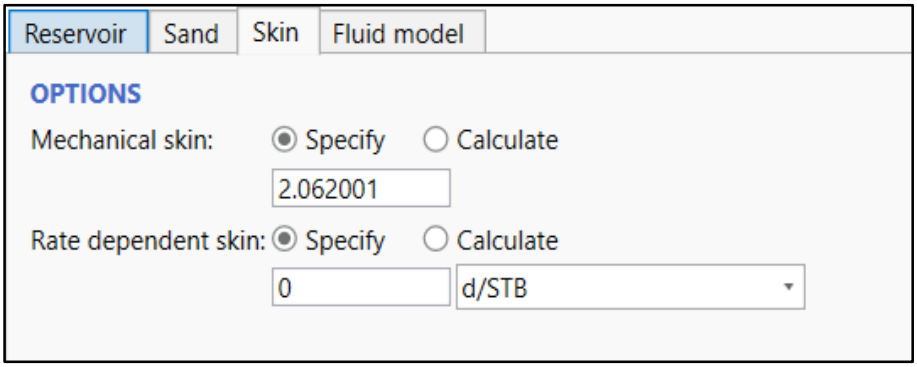


Figura 35. *Carga de los datos para la creación de la curva IPR por el método de Darcy en PIPESIM a partir de la corrida del código.*
Fuente: Software PIPESIM 2019.1-Schlumberger

De esta manera se genera el modelo de pozo en PIPESIM mediante codificación, posteriormente se abre los archivos generados y se procede a cargar de manera manual los datos de permeabilidades relativas. Finalmente se guarda y se cierra el modelo.

3.4 Calibración de la Bomba BES.

Para generar el modelo de pozo con la bomba BES, se procede a elaborar el código “Calibration” utilizando la base de datos “Pump_Info”; En la Figura 36 se puede observar el flujo de trabajo que servirá como base para el desarrollo del código.

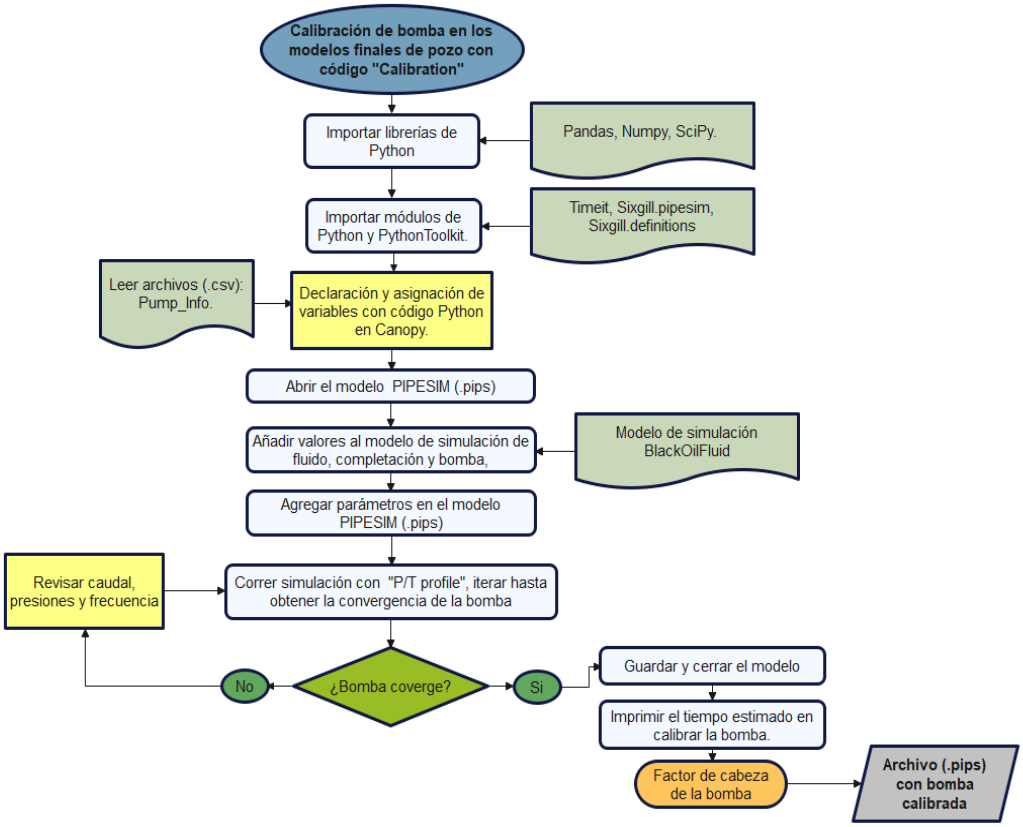


Figura 36. *Diagrama de flujo para el desarrollo del código Calibration.*
Elaborado por: Lesly Rivera

Se trabaja sobre el modelo creado anteriormente con el código “Model_builder”. La codificación comienza con la importación de las librerías de Pyhthon Toolkit, Pandas, Numpy y el tiempo de ejecución del código; se define los parámetros y constantes que serán utilizadas en los *keywords,* para la construcción del modelo de cada uno de los pozos.

En la Figura 37 se presenta un fragmento del código a modo de ejemplo, por temas de confidencialidad con la empresa no se anexa el código completo.

```
from sixgill.pipesim import Model
from sixgill.definitions import Parameters
import pandas as pd
import numpy as np
from scipy.interpolate import interp1d
from timeit import default_timer as timer
```

Figura 37. *Librerías, módulos y funciones importadas de Python Toolkit, Pandas y Numpy.*
Fuente: Canopy versión 2.1.9.30

Los valores que se encuentran en el archivo de datos “Pump_Info” de Excel se los lee con las librerías de Pandas. A continuación, se define las variables que van a ser utilizadas en el código; cada una debe estar asignada a una clase.

Se debe acceder a los parámetros de los componentes a través de los métodos “get_value() & set_value()”. Con los parámetros definidos, se puede enlazar correctamente las clases para acceder a cada función del software; de esta manera, se está asegurando que todos los valores que se encuentran en Excel sean ingresados correctamente en el software PIPESIM para generar el modelo de cada pozo.

La bomba debe ser calibrada mediante la sensibilidad a las condiciones de: presión en cabeza, frecuencia a la que está trabajando y el caudal de líquido que aporta cada pozo; el método a utilizar será el perfil de Presión/Temperatura; el resultado de la simulación se verá reflejado en el factor de cabeza de la bomba (*well head factor*) y el tiempo que tardó en calibrarse.

El perfil de Presión/Temperatura permite factorizar la eficiencia y ajustar el cabezal de la bomba (predeterminado =1) para que coincida mejor con los datos de rendimiento del pozo o a su vez tener en cuenta el desgaste e ineficiencias que se pueda estar teniendo en el sistema.

En la Figura 38 se puede observar la corrida del código y la impresión en la interfaz de Canopy del tiempo que se tarda la codificación en generar el modelo y obtener los valores del factor de cabeza mediante la sensibilización de la bomba BES para cada pozo.

```
In [2]: %run "c:\users\usuario\appdata\local\temp\tmpvsaly3.py"
INFO:manta.server.manager:Starting PIPESIM server on thread ID: 27732
INFO:manta.server.manager:Waiting for PIPESIM server to start and Python_Toolkit license check out.
INFO:sixgill.core.metadata:Using cached metadata for 'http://localhost:59728/api/metadata#'
WARNING:sixgill.core.catalog:Computing context for node "Node?id=metadata%23Model_2017.1*2712" (8)
INFO:sixgill.core.simulations:Simulation is running.
INFO:sixgill.core.simulations:Simulation is finished.
INFO:sixgill.core.simulations:Simulation run successfully.
INFO:sixgill.core.simulations:Simulation is running.
INFO:sixgill.core.simulations:Simulation is running.
INFO:sixgill.core.simulations:Simulation is running.
INFO:sixgill.core.simulations:Simulation is running.
INFO:sixgill.core.simulations:Simulation is finished.
INFO:sixgill.core.simulations:Simulation run successfully.
INFO:sixgill.core.simulations:Simulation is running.
INFO:sixgill.core.simulations:Simulation is running.
INFO:sixgill.core.simulations:Simulation is running.
INFO:sixgill.core.simulations:Simulation is running.
INFO:sixgill.core.simulations:Simulation is finished.
INFO:sixgill.core.simulations:Simulation run successfully.
1098.997515536248 0.9981498634149822
333.66586370000005
```

Figura 38. *Tiempo y factor de cabeza de la bomba con la simulación del perfil de Presión/Temperatura. Fuente: Canopy versión 2.1.9.30*

En la Figura 39 se observa el archivo PIPESIM generado a partir de la corrida del código, el modelo se genera con los datos que se encuentran cargados en el archivo Excel, el diagrama de pozo cuenta con profundidades de casing y tubing y con el funcionamiento de la bomba BES. Se puede observar que el factor de cabeza se obtiene con la sensibilización de la bomba.

Con la ejecución del código se aprecia que no existen ineficiencias en el trabajo que está desempeñando la bomba en el pozo y que el rendimiento es óptimo para seguir trabajando en el campo. Finalmente se guarda y cierra el modelo.

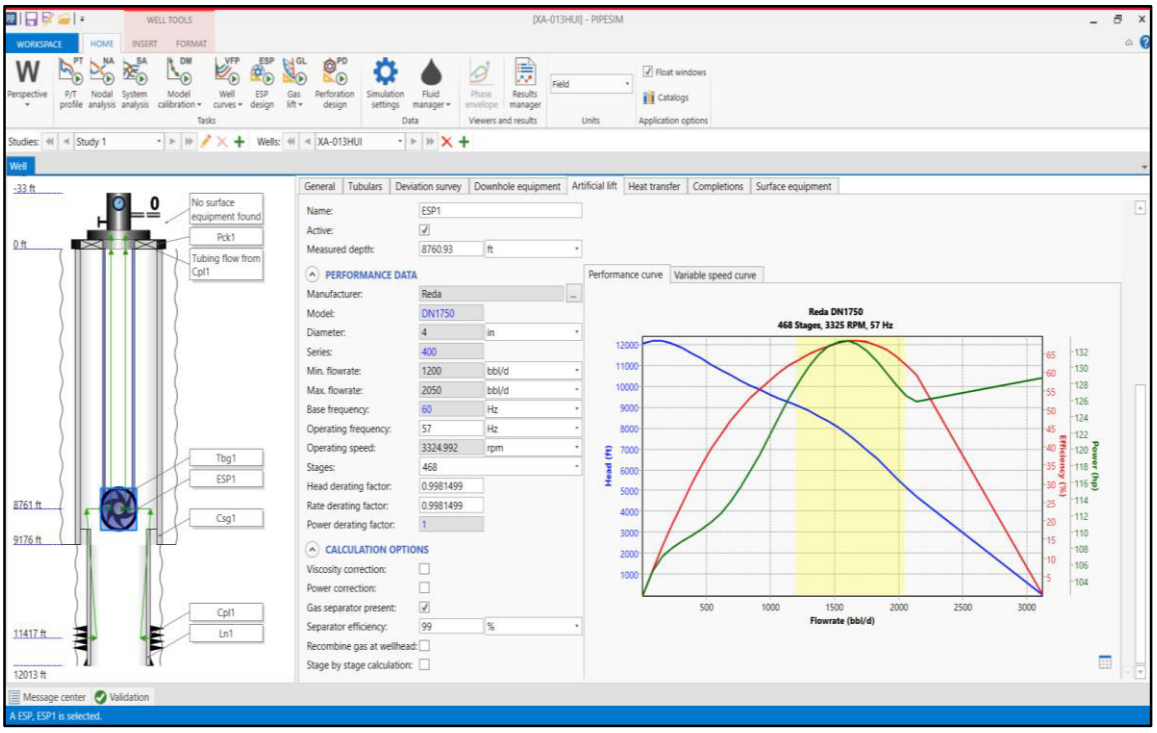


Figura 39. *Archivo PIPESIM con Bomba BES calibrada a partir de la corrida del código.*
Fuente: Software PIPESIM 2019.1-Schlumberger

3.5 Obtención de Presiones y Caudales para cada Punto del Modelo de Choke.

La Figura 40 describe el flujo de trabajo que se utilizará para generar el código "Nodal_ACD" de los puntos nodales (A, C y D), cabe mencionar que por cuestiones de confidencialidad con la empresa el código no puede ser publicado.

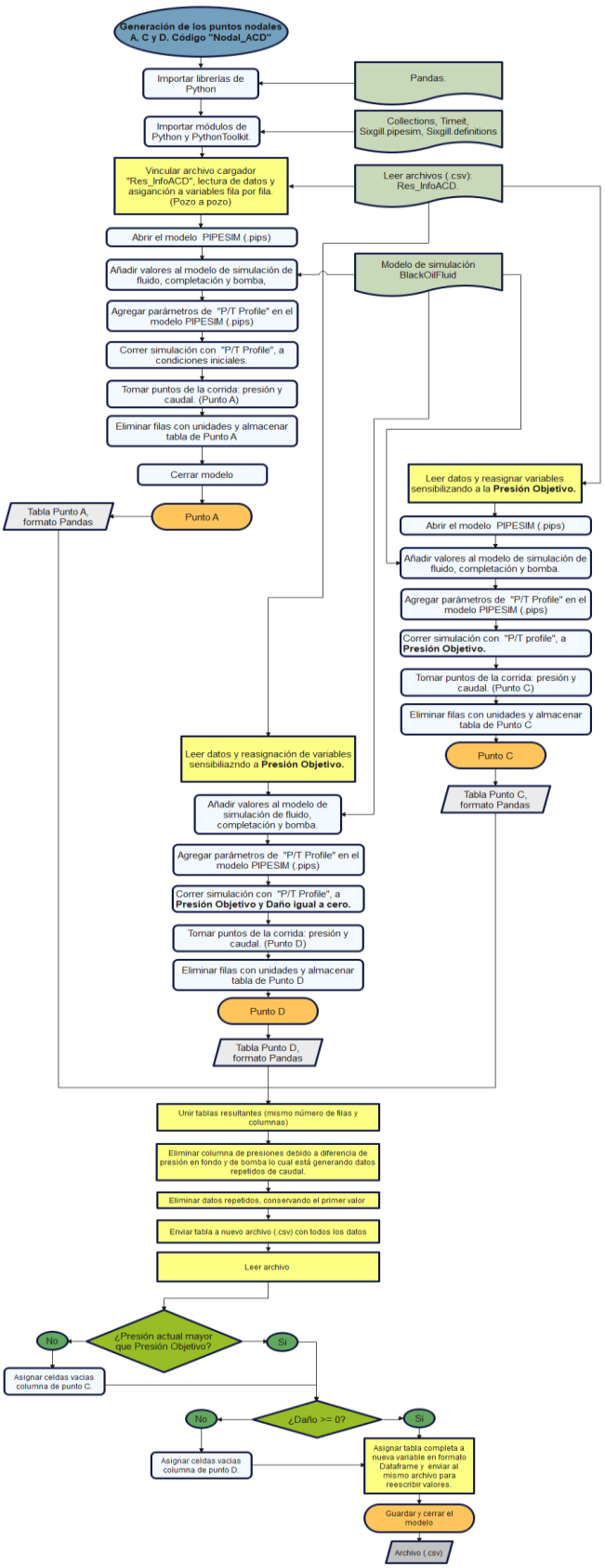


Figura 40. *Diagrama de flujo para los puntos nodales A, C y D.*
Elaborado por: Lesly Rivera

3.5.1 Caso Base, punto A

El caso base representa la producción más las perdidas calculadas a lo largo de la línea de producción dando como resultado el *"Lowest Maximum Production Potential"*. Se puede realizar un control de calidad de la simulación, comparando los resultados obtenidos con los datos de producción, obteniendo el ajuste correcto para el modelo.

Para el punto A las presiones y caudales están dadas a partir de las condiciones iniciales en la que está operando cada pozo, el código "Nodal_ACD" permitirá ejecutar la simulación del perfil de Presión/Temperatura en PIPESIM. Los resultados que se obtenga de la simulación serán trasladados a un documento Excel.

La base de datos a utilizar se encuentra en un archivo Excel llamado "Res_InfoACD", el cargador se lo puede ver en la Tabla 5. Los valores del cargador son los mismos del archivo "Res_Info" cambiando el valor del daño por los valores que se encuentran en la Tabla 3 obtenidos a partir de la corrida del código "Skin Calibration".

Tabla 5. *Datos del cargador Res_InfoACD.*
Elaborado por: Lesly Rivera

WELL	CASING_START	CASING_END	CASING_ID	CASING_WT	CASING_DIAM	LINER_START	LINER_END	LINER_ID	LINER_WT	LINER_DIAM	TUBING_START	TUBING_END	TUBING_ID	TUBING_WT	ESP_TYPE
AB-057HI	0	15	6.28	0.36	10						0	5	2.99	0.25	
AC-063TI	0	15	6.28	0.36	10						0	5	2.99	0.25	
AC-127TI	0	15	6.28	0.36	10						0	5	2.99	0.25	
AC-165TI	0	15	6.28	0.36	10						0	5	2.99	0.25	
AC-170TI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XA-009TI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XA-010HS1UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XA-013HUI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XA-037UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XB-012UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XB-021UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XB-033UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XB-035UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
XC-043HUI	0	15	6.28	0.36	10						0	5	2.99	0.25	
YA-015UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
YD-023HUI	0	15	6.28	0.36	10						0	5	2.99	0.25	
Z-003UI	0	15	6.28	0.36	10						0	5	2.99	0.25	
Z-004TI	0	15	6.28	0.36	10						0	5	2.99	0.25	

WELL	ESP_STAGES	CPL_START	CPL_PR	CPL_TR	CPL_THICKNESS	CPL_K	CPL_S	LIQUID	WCUT	GOR	API	FREQ	WHP	PWF	BRCH	PWF_OBJAVOCET
AB-057HI		10	1700	236	10	614	5.6219	755	74	9.02	27.6	77.5	110	741.31	AB-057HI - Wellhead	450
AC-063TI		10	1490.51	220	55.946	519.24	9.3263	1944	89	229	22.4	59	20	838.46	AC-063TI - Wellhead	676.26
AC-127TI		10	1481	220	60.337	83	-1.868	1334	88	229.01	23.9	61	30	476.25	AC-127TI - Wellhead	620
AC-165TI		10	1427.21	220	73.358	113.29	5.2441	780	86	229.03	23.3	56	30	654.16	AC-165TI - Wellhead	477.43
AC-170TI		10	1478.77	220	78.99	160	3.2751	1446	87	229.01	22.1	57	20	661.89	AC-170TI - Wellhead	779.3
XA-009TI		10	2064	226	8	55	-0.9717	174	3	150	26.3	48.5	60	376.88	XA-009TI - Wellhead	450
XA-010HS1UI		10	1387.94	220	28	254	0.6525	584	24	163	18.5	57.6	120	443.1	XA-010HS1UI - Wellhead	510
XA-013HUI		10	1431.4	220	53	213	2.062	1099	1	150	19.8	57	200	556.81	XA-013HUI - Wellhead	550
XA-037UI		10	1551.98	220	30.91	77	2.2796	228	36	160	19.1	61	105	291.15	XA-037UI - Wellhead	287
XB-012UI		10	1343.15	220	46.25	330	2.0603	1365	65	160	20.1	61	25	384.7847534	XB-012UI - Wellhead	330
XB-021UI		10	1283.91	220	36	125.23	5.2505	406	90	160	17.8	57	260	340.0720655	XB-021UI - Wellhead	300
XB-033UI		10	1095.7	220	65.37	482	6.0119	1060	48	160	18.6	51	25	514.7659761	XB-033UI - Wellhead	280
XB-035UI		10	1249.52	220	40	65	5.0913	145	3	160	18.5	52	100	291.3196234	XB-035UI - Wellhead	240
XC-043HUI		10	1500	220	31.48	481.68	-3.9731	1720	64	163	17.8	56	160	1101.26	XC-043HUI - Wellhead	1185
YA-015UI		10	1500	224	37	155	9.9661	340	64	146.98	18.7	57.5	130	370.95	YA-015UI - Wellhead	380
YD-023HUI		10	1201.41	224	36.62	1116	1.1344	2040	44	147	18.5	64.7	200	692.1	YD-023HUI - Wellhead	700
Z-003UI		10	1820	220	33.5	90	15.711	162	3	145.98	18.3	43	30	415.16	Z-003UI - Wellhead	275
Z-004TI		10	1729	226	19.76	150	8.3686	444	6	123.01	27.1	54.5	150	306.44	Z-004TI - Wellhead	450

3.5.2 Caso I, punto C

Una vez obtenido el punto A el código correrá la simulación en PIPESIM con el método del perfil de Presión/Temperatura para obtener el punto C; utilizando los datos del cargador “Res_InfoACD”.

La *Outlet Pressure* que utilizará el modelo será la Presión Objetivo asignada para cada pozo, dicha presión es un dato generado a partir de los reservoristas en campo, puesto que, para establecer la presión se debe analizar todas las condiciones en la que cada pozo está trabajando y en base a ello se debe generar una reducción en la presión, la misma que no debe ser menor a la presión de burbuja (Pb) para resguardar la integridad de cada pozo.

Se debe tener cuidado en el valor de la presión generada, ya que, trabajar con una presión extremadamente baja provocará que el pozo opere con un nivel de fluido dinámico menor a la profundidad de asentamiento de la bomba BES y esto a su vez genera un problema en el motor siendo propenso a quemarse.

Se conoce que la eficiencia de trabajo del equipo BES se basa en tener un nivel de fluido que le permita generar enfriamiento al sistema, además, que se estará trabajando por debajo de la presión de burbuja lo cual generará cavitación en las bombas BES.

El código posee un condicional que muestra los resultados obtenidos si cumple la condición: Presión Objetivo menor que la presión con la que está trabajando el pozo. De esta manera se conoce los pozos que son candidatos para ser optimizados.

3.5.3 Caso II, punto D

El punto D hace referencia a un trabajo de reacondicionamiento en el pozo para remover el daño de formación permitiendo restituir las condiciones naturales de producción.

El código de manera automática realiza la simulación del perfil Presión/Temperatura con los datos que se tienen en el cargador “Res_InfoACD”; el valor del daño con el que se correrá el modelo se encuentra programado al valor de cero haciendo referencia a los resultados que se obtendrían al remover el daño de formación; cabe mencionar que el código será ejecutado con la presión objetivo que fue establecida para el punto C.

Los pozos que carecen de daño hacen referencia a un trabajo de estimulación en la arena, para dichos pozos el código está diseñado para no mapear los resultados y

continuar con los siguientes, los datos obtenidos serán solo de los pozos que tengan un daño de formación mayor a cero, de esta manera se podrá conocer las oportunidades que se tiene en la optimización de la producción.

3.6 Elaboración de Diagramas de Choke en Power BI.

Después de haber desarrollado el código para obtener los puntos A, C y D se procede a desarrollar el modelo de choke; los valores de los puntos de cada pozo se los almacena en un cargador de Excel para luego proceder a armar los diagramas en Power BI. En la Figura 41 se encuentra detallado el proceso.

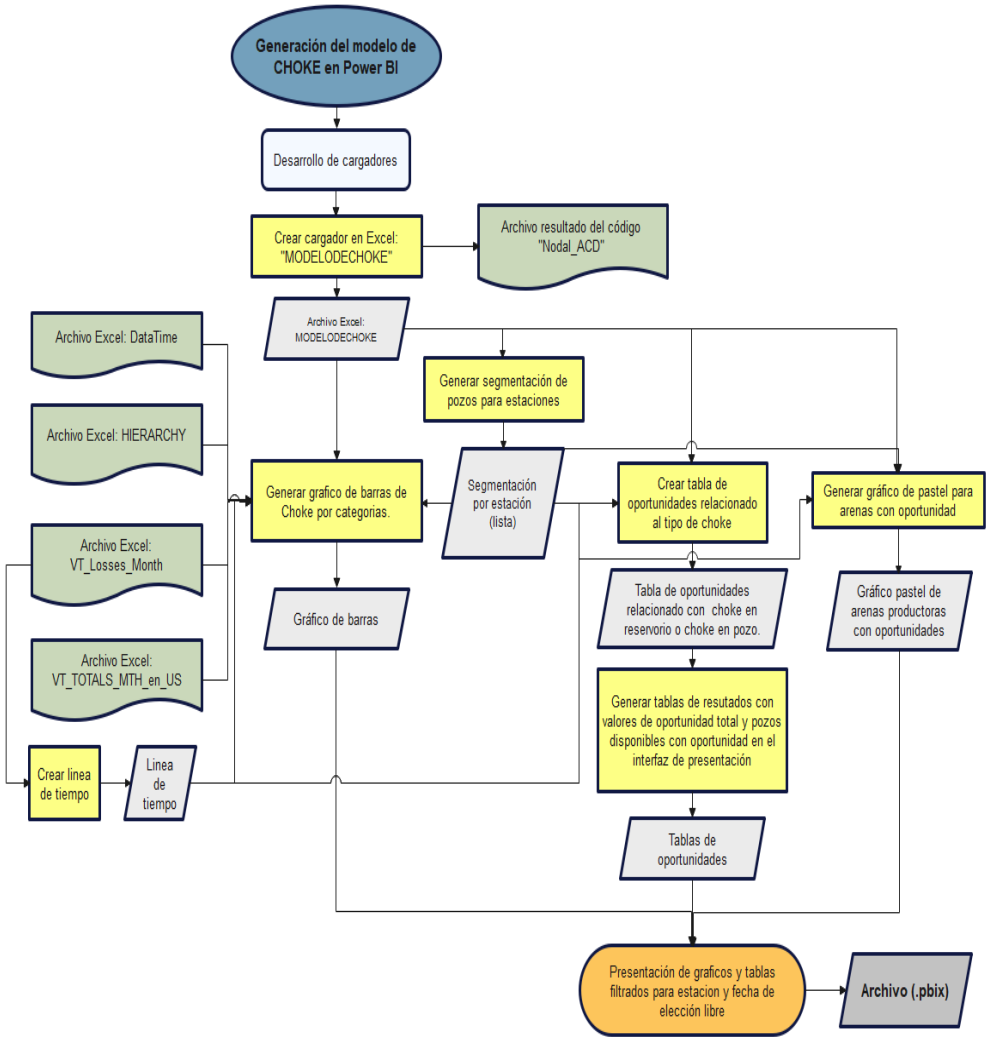


Figura 41. *Diagrama de flujo para modelo de choke en Power BI.*
Elaborado por: Lesly Rivera

Se optó trabajar en Power BI ya que es una herramienta que permite visualizar datos, diagramas y gráficos de una manera más versátil e interactiva, también se puede elaborar informes personalizados dando paso a una mejor comprensión al momento de presentar los resultados. Además, con toda la información que se va actualizando instantáneamente se puede tomar decisiones acertadas y crear una mejor interacción en los entornos de trabajo.

En la Figura 42 se puede observar el modelo de choke generado en Power BI.

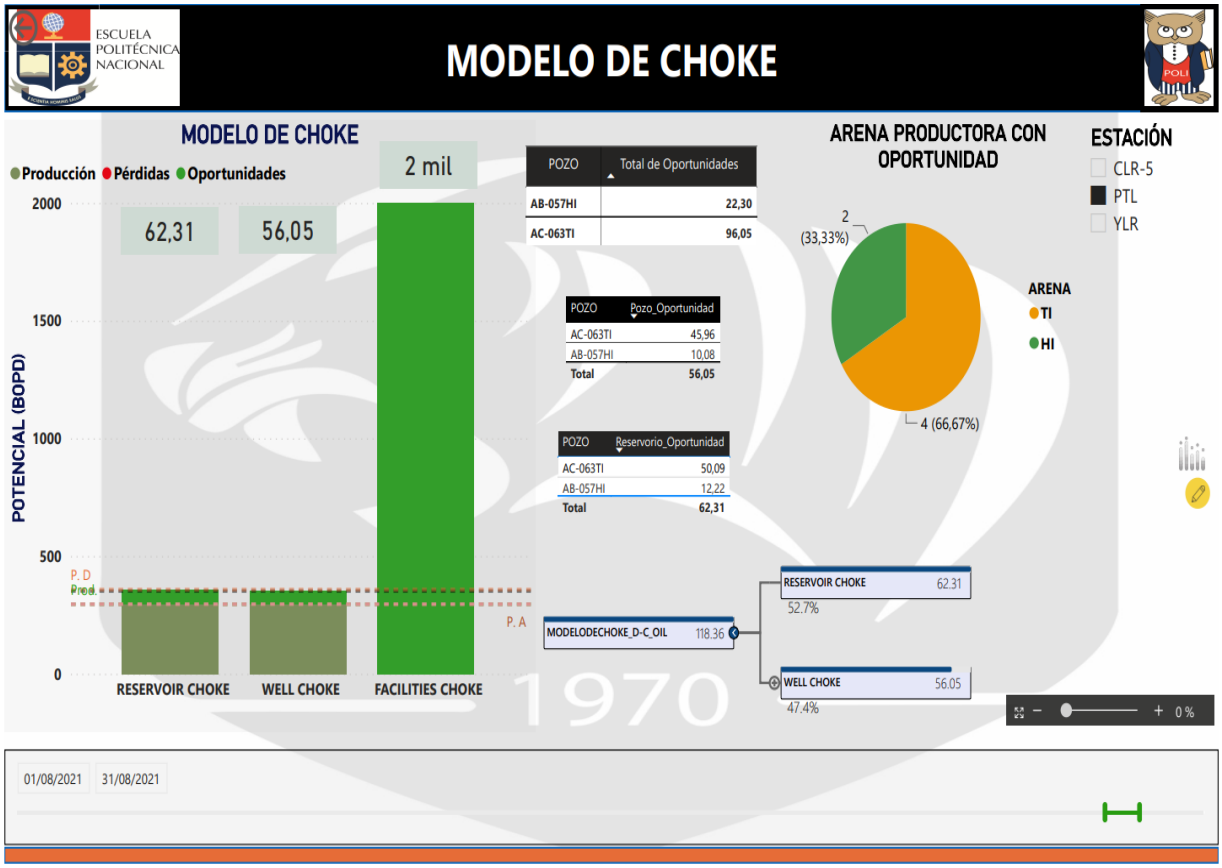


Figura 42. *Modelo de choke generado en Power BI.*
Elaborado por: Lesly Rivera

4. RESULTADOS Y DISCUSIÓN

A partir del código y la simulación con el perfil de Presión/Temperatura se obtienen los siguientes valores de caudales para los respectivos puntos como se evidencia en la Tabla 6.

Tabla 6. *Valores obtenidos de los puntos nodales A, C y D con el código en Python.*
Elaborado por: Lesly Rivera

WELL	PWF (A)	QL (A)	QL (A) OUTLETP=Pwf	QOIL (A)	QOIL (A) OUTLETP=Pwf	QL (C)	QOIL (C)	QL(D)	QOIL(D)
AB-057HI	741.3	755.0	675.6	196.3	175.7	714.4	185.8	761.4	198.0
AC-063TI	838.5	1944.0	489.8	213.8	53.9	907.7	99.8	1363.0	149.9
AC-127TI	476.3	1334.0	1150.3	160.1	138.0				
AC-165TI	654.2	780.0	516.2	109.2	72.3	586.5	82.1	716.2	100.3
AC-170TI	661.9	1446.0	1104.4	188.0	143.6			1598.1	207.8
XA-009TI	376.9	174.0	139.7	168.8	135.5				
XA-010HS1UI	443.1	584.0	532.9	443.8	405.0			624.9	475.0
XA-013HUI	556.8	1099.0	938.1	1088.0	928.7	941.5	932.1	1048.2	1037.7
XA-037UI	291.2	228.0	218.4	145.9	139.7	218.6	139.9	272.5	174.4
XB-012UI	384.8	1365.0	1230.0	477.8	430.5	1252.1	438.2	1366.0	478.1
XB-021UI	340.1	406.0	395.7	40.6	39.6	401.0	40.1	502.2	50.2
XB-033UI	514.8	1060.0	752.5	551.2	391.3	889.0	462.3	1069.8	556.3
XB-035UI	291.3	145.0	133.3	140.7	129.3	136.9	132.8	213.6	207.2
XC-043HUI	1101.3	1720.0	1213.4	619.2	436.8				
YA-015UI	371.0	340.0	316.1	122.4	113.8			479.3	172.6
YD-023HUI	692.1	2040.0	1421.1	1142.4	795.8			2136.4	1196.4
Z-003UI	415.2	162.0	136.4	157.1	132.3	146.5	142.1	307.8	298.5
Z-004TI	306.4	444.0	728.3	417.4	684.6			796.8	749.0

El tiempo que tardó en crearse el modelo y generar los puntos nodales A, C y D con codificación en Python en los 18 pozos fue de 2701.2956 segundos dando un aproximado de 45.216 minutos.

Se añade 3 minutos al tiempo obtenido de la corrida, necesarios para que el operador pueda ejecutar el código y revisar que toda la información este correcta antes de crear el modelo.

Podemos ver que se reduce significativamente el tiempo utilizado para generar el modelo de choke, si el modelo es creado desde cero de manera manual, todo el proceso invertido será aproximadamente de 50 minutos por pozo, si se genera para todos los

pozos que tiene el campo (18 pozos) se tendrá 900 minutos invertidos en crear los modelos en PIPESIM y a esto se agregaría unos 30 minutos más en generar el modelo de choke en Power BI, obteniendo un tiempo invertido de 930 minutos.

En la Tabla 7 se encuentra el tiempo que tarda en correr cada código para crear el modelo de pozo y se lo compara con el tiempo implementado al realizar el proceso de forma manual.

Tabla 7. *Tiempo de corrida de cada código para generar el modelo de pozo y armado del modelo de choke*
Elaborado por: Lesly Rivera

CÓDIGO	TIMPO DE EJECUCIÓN AUTOMÁTICA POR POZO (min)	TIEMPO DE EJECUCIÓN MANUAL POR POZO (min)
Res_builder	0.7494	118
Skin Calibration	0.6204	90
Model_builder	0.6421	360
Calibration	15.0322	450
Nodal_ACD	2.5969	415
Total	19.641	**1433**

Con la automatización del proceso se obtiene la actualización del modelo de choke, indispensable para realizar un seguimiento continuo de las oportunidades de optimización de la producción que se vayan presentando en el campo.

El modelo de choke para la estación CRL-5 se encuentra representado en la Figura 43 e indica las oportunidades de optimización en la producción a nivel de:

- Facilidades (barra FACILITIES CHOKE) con un valor de: 5000 BPPD.
- Pozo (barra WELL CHOKE) con un valor de: 96.27 BPPD.
- Reservorio (barra RESERVOIR CHOKE) con un valor de: 376.69 BPPD.

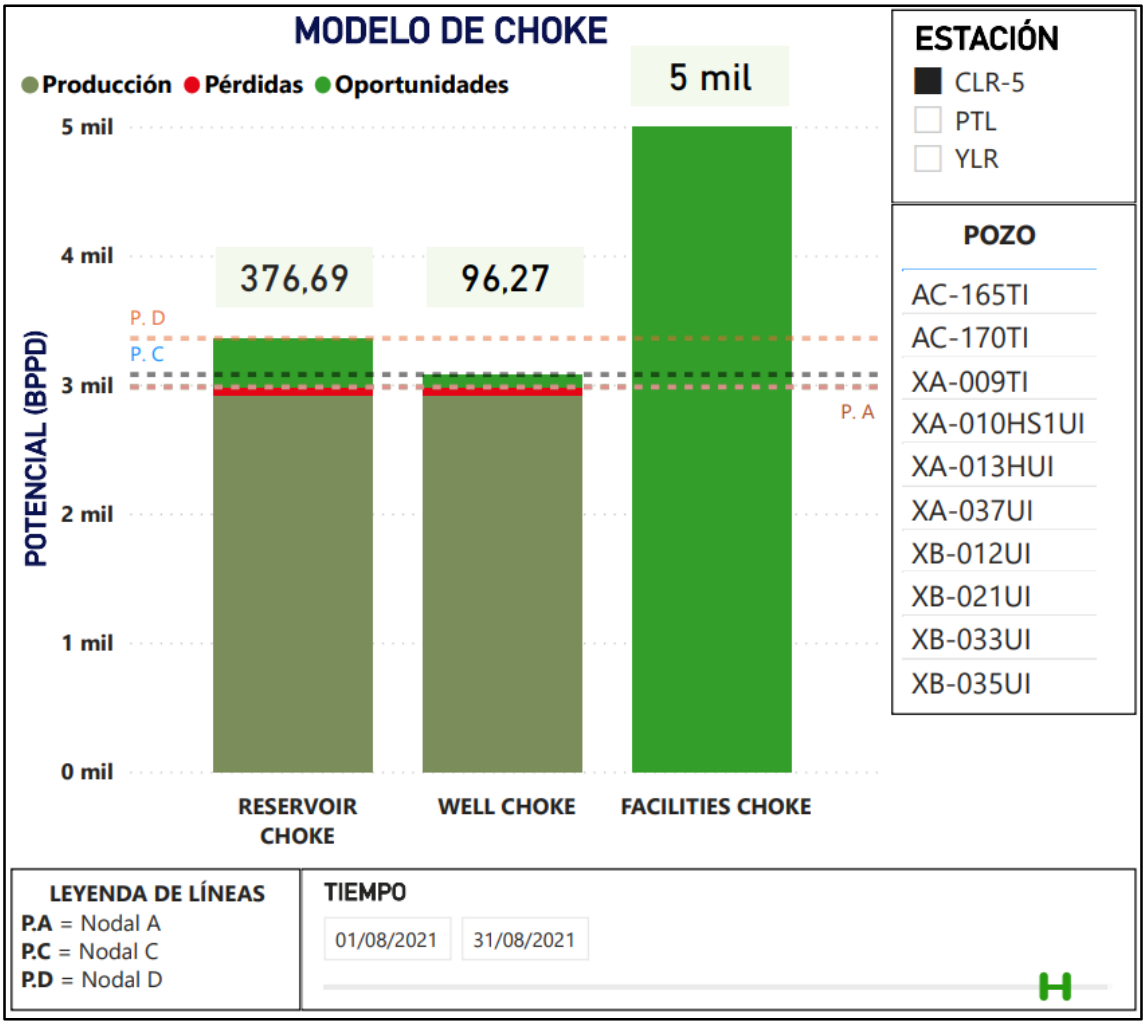


Figura 43. *Modelo de choke para la estación CRL-5.*
Elaborado por: Lesly Rivera

El mismo procedimiento que se realizó para la estación CRL-5 se aplica para la estación PTL que se encuentra en la Figura 44, las oportunidades de optimización en la producción a nivel de facilidades, pozo y reservorio se encuentran mapeadas en el mismo gráfico.

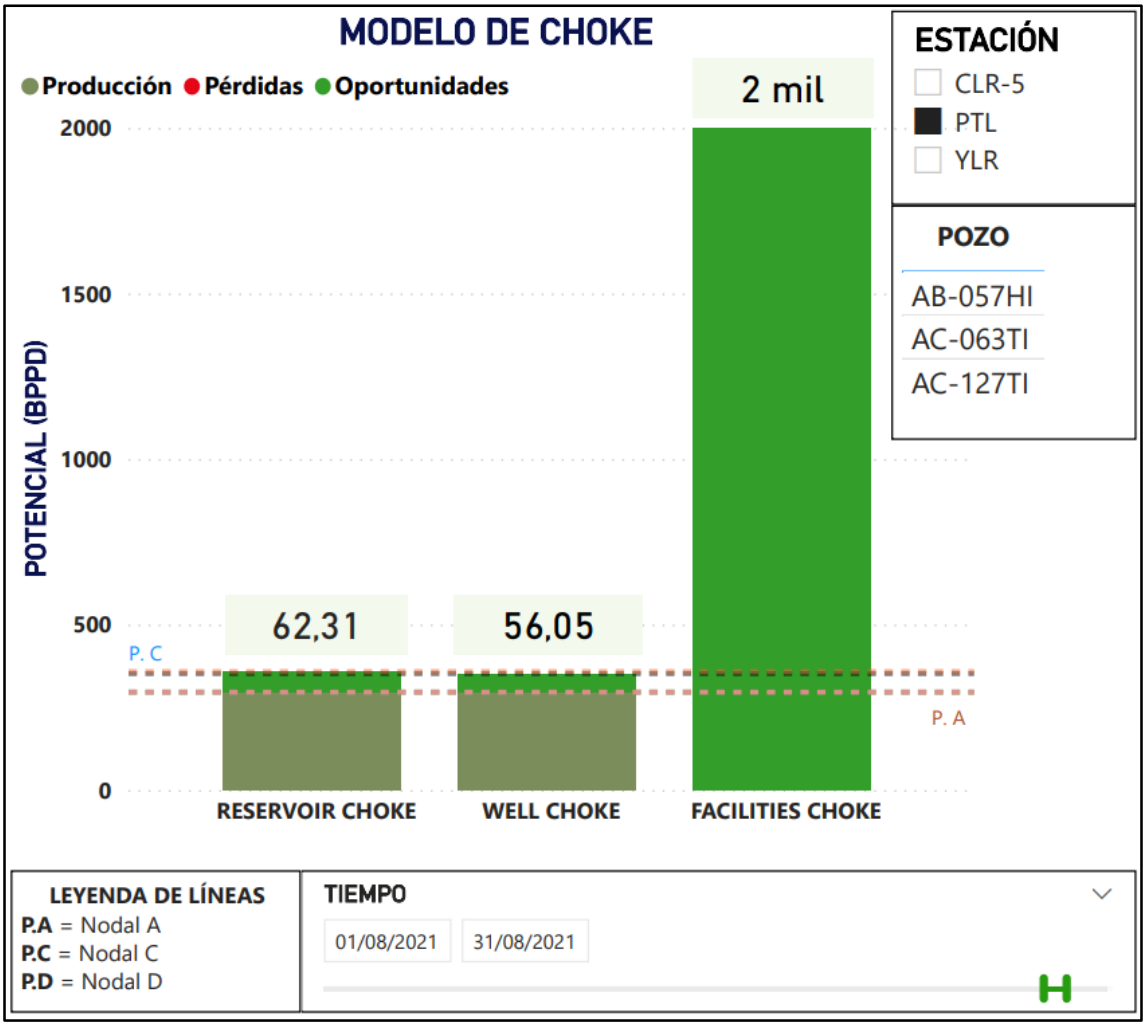


Figura 44. *Modelo de choke para la estación PTL.*
Elaborado por: Lesly Rivera

El modelo de choke de la estación YLR se lo puede observar en la Figura 45, el análisis será el mismo que se aplicó para la estación PTL, los datos de optimización en la producción se encuentran en el mismo gráfico.

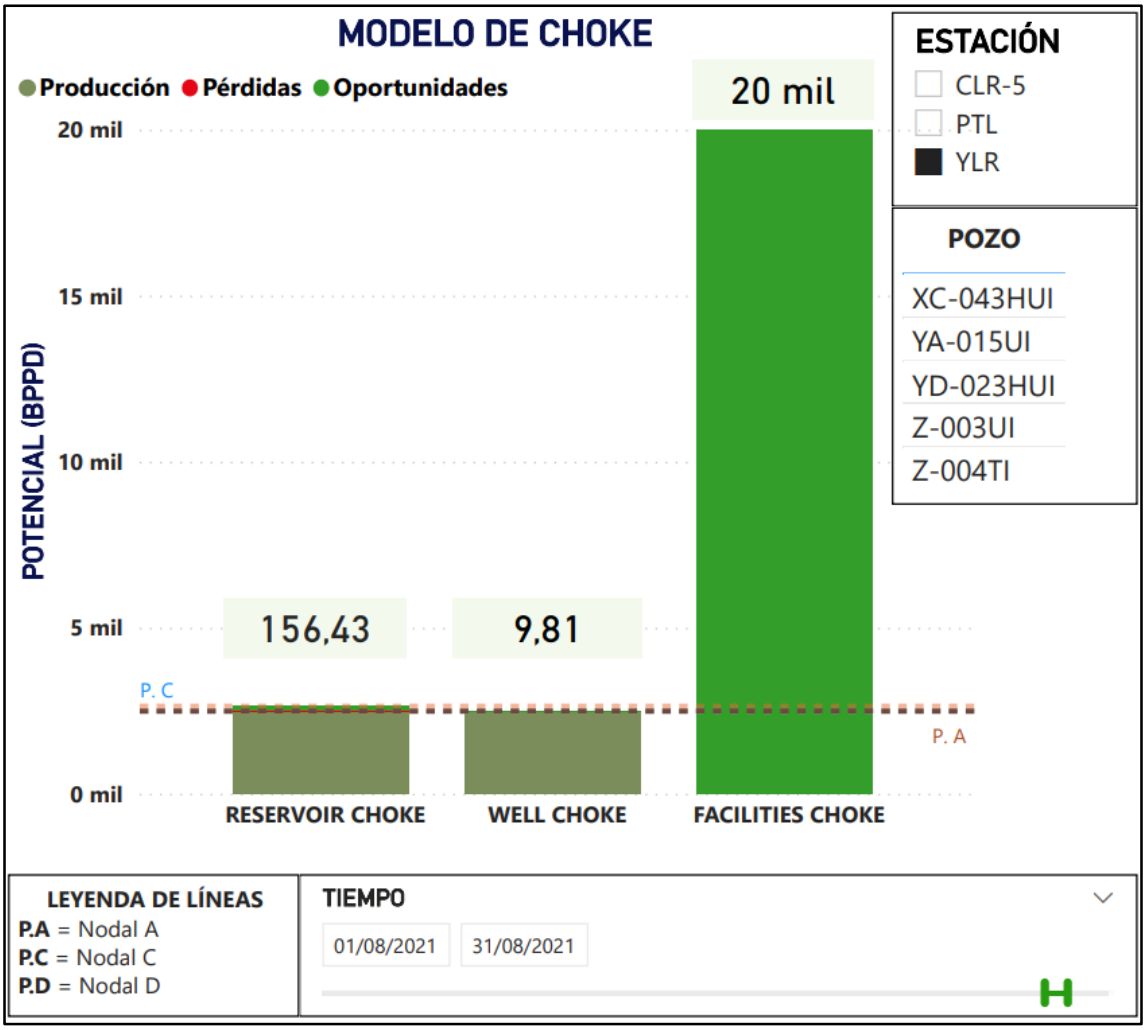


Figura 45. *Modelo de choke para la estación YLR.*
Elaborado por: Lesly Rivera

Los valores de la oportunidad de **optimización de la producción por pozo** se encuentran en las siguientes tablas:

En la Tabla 8 se encuentran mapeados los valores para la estación CLR-5:

Tabla 8. *Valores de caudal de petróleo a partir de la oportunidad de pozo para la estación CLR-5.*
Elaborado por: Lesly Rivera

POZO	OPORTUNIDAD POR POZO (BPPD)
XA-037UI	0.18
XB-021UI	0.53
XA-013HUI	3.44
XB-035UI	3.52
XB-012UI	7.74
AC-165TI	9.85
XB-033UI	71.01
TOTAL	96.27

- La estación cuenta con el aporte de 10 pozos, sin embargo, se puede evidenciar que solo a 7 de ellos se les puede realizar una intervención para cambiar o aumentar frecuencia en la bomba y obtener un aporte significativo de crudo para la estación.

En la Tabla 9 se muestran los valores de oportunidad para la estación PTL:

Tabla 9. *Valores de caudal de petróleo a partir de la oportunidad de pozo para la estación PTL. Elaborado por: Lesly Rivera*

POZO	OPORTUNIDAD POR POZO (BPPD)
AB-057HI	10.08
AC-063TI	45.96
TOTAL	56.04

- Son 3 pozos los que aportan a la estación, sin embargo, solo a 2 de ellos se les puede realizar una intervención en el sistema BES para obtener un incremento en la producción de crudo.

Se muestran los valores de oportunidades para la estación YLR en la Tabla 10:

Tabla 10. *Valores de caudal de petróleo a partir de la oportunidad de pozo para la estación YLR. Elaborado por: Lesly Rivera*

POZO	OPORTUNIDAD POR POZO (BPPD)
Z-003UI	9.81
TOTAL	9.81

- La estación cuenta con el aporte de 3 pozos, pero solo a 1 se le puede realizar una intervención en la bomba BES para obtener un aporte significativo de crudo.

Los valores de la oportunidad de **optimización de la producción por reservorio** se encuentran en las siguientes tablas:

La Tabla 11 cuenta con valores de oportunidad para la estación CLR-5:

Tabla 11. *Valores de caudal de petróleo a partir de la oportunidad de reservorio para la estación CLR-5. Elaborado por: Lesly Rivera*

POZO	OPORTUNIDAD POR POZO (BPPD)
XB-021UI	10.13
AB-057HI	12.22
AC-165TI	18.16
XA-037UI	34.49
XB-012UI	39.88
AC-063TI	50.09
XB-035UI	74.42
XB-033UI	94.04
XA-013HUI	105.57
Z-003UI	156.43
TOTAL	595.43

- Se puede apreciar que todos los pozos de la estación son candidatos para realizar una intervención a la arena productora con el objetivo de eliminar el daño de formación, obteniendo una producción significativa de crudo.

Los valores de oportunidad para la estación PTL se encuentran mapeados en Tabla 12:

Tabla 12. *Valores de caudal de petróleo a partir de la oportunidad de reservorio para la estación PTL. Elaborado por: Lesly Rivera*

POZO	OPORTUNIDAD POR POZO (BPPD)
AB-057HI	12.22
AC-063TI	50.09
TOTAL	62.31

- El daño de formación solo puede ser removido en 2 pozos de la estación, los mismos que son: AB-057HI y AC-063TI.

La Tabla 13 contiene los datos de la oportunidad de producción de crudo en la estación YLR:

Tabla 13. *Valores de caudal de petróleo a partir de la oportunidad de reservorio para la estación YLR. Elaborado por: Lesly Rivera*

POZO	OPORTUNIDAD POR POZO (BPPD)
Z-003UI	156.43
TOTAL	156.43

- Se puede observar que la intervención para eliminar el daño de formación solo se puede realizar en un pozo, obteniendo un aporte de crudo significativo para la estación.

5. CONCLUSIONES Y RECOMENDACIONES

A partir de la automatización de los procesos para generar el modelo de choke mediante codificación Python se concluye que:

EN EL CÓDIGO

- La automatización de procesos permite ahorrar tiempo significativo en los operadores, la probabilidad de errores en el mapeo de datos es mínima existiendo confiabilidad en los resultados obtenidos.
- La optimización del tiempo para generar los diagramas de choke tanto para pozo, reservorio y facilidad se da en un 90%. Sustituyendo en un 100% la carga manual de datos para las corridas de simulación en el software PIPESIM.
- El código Python al ser un lenguaje de fácil uso y compatible con la mayoría de los programas existentes en la industria del petróleo y gas, permite crear un nexo de codificación con el software PIPESIM para realizar simulaciones en pozos desde cero, permitiendo modificar y actualizar los modelos existentes si fuese el caso.
- Se obtienen datos exactos de los puntos nodales A, C y D; permitiendo mapear las oportunidades que se presentan en pozo y reservorio.

EN LOS RESULTADOS

- Se evidencia la presencia de pozos que pueden ser optimizados con un trabajo de Workover; por ejemplo, se puede realizar intervención a la bomba BES o se puede remover el daño de formación.
- La optimización de la producción solo se da si existe la oportunidad de reducir la Pwf y si se tiene daños de formación que aún no han sido estimulados, caso contrario la optimización por parte del pozo estará realizada en un 100%; en este caso se deberá evaluar otros métodos de aporte para la producción del campo.
- El total de oportunidades para producir en la estación CLR-5 es de 472.96 BPPD, siendo de esta producción un 79.6% (376.69 BPPD) las oportunidades

desbloqueadas en reservorio y el 20.4% (96.27 BPPD) siendo oportunidades desbloqueadas en pozo.

- El total de oportunidades para producir en la estación PTL es de 118.36 BPPD, siendo de esta producción un 52.6% (62.31 BPPD) las oportunidades desbloqueadas en reservorio y el 47.4% (56.05 BPPD) siendo oportunidades desbloqueadas en pozo.
- El total de oportunidades para producir en la estación YLR es de 166.24 BPPD, siendo de esta producción un 94.1% (156.43 BPPD) las oportunidades desbloqueadas en reservorio y el 5.9% (9.81 BPPD) siendo oportunidades desbloqueadas en pozo.

En base a los desafíos que se presentaron durante el trabajo y esperando que en un futuro se obtengan mejores resultados para las oportunidades de producción en el campo, se recomienda:

- Obtener valores de reservorio más exactos, la permeabilidad y espesor de formación juegan un rol fundamental al momento de realizar la sensibilidad del daño de formación, si se obtienen datos más precisos se puede desbloquear de manera más eficiente las oportunidades de producción en el campo.
- Generar una metodología más eficiente para la carga de permeabilidades relativas a los modelos en PIPESIM (.pips).
- Migrar el código a las versiones más recientes del software PIPESIM, puesto que, las nuevas versiones poseen una interfaz que permite ser automatizada de manera completa con codificación Python.
- Modificar el código para optimizar el tiempo de corrida evitando la impresión de valores en el archivo Excel y solo imprimir estos valores en la interfaz de Canopy; de esta manera se reduce el código final a la mitad de su extensión, obteniendo los puntos C y D en una misma iteración.

6. REFERENCIAS BIBLIOGRÁFICAS

Carpentries, T. (2021). *Análisis y visualización de datos usando Python*. Obtenido de https://datacarpentry.org/python-ecology-lesson-es/02-starting-with-data/index.html

DelfStack. (2020). *SciPy scipy.interpolate.interp1d Función*. Obtenido de https://www.delftstack.com/es/api/scipy/scipy-scipy.interpolate.interp1d-method/

Islas, C. (1991). *MANUAL DE ESTIMULACIÓN MATRICIAL DE POZOS PETROLEROS.* Mexico: Ediciones Gráficas "Z", S.A.

Laura, M. (2021). *Librerías Python: ¿Qué son y para qué sirven cada una de ellas?* Obtenido de https://es.bitdegree.org/tutoriales/librerias-python/

Lucidchart. (2020). *¿Qué es un diagrama de flujo?* Obtenido de https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo

Machuca, F. (20 de 5 de 2021). *¿Qué es Python? El lenguaje de programación más popular para aprender en 2021.* Obtenido de crehana: https://www.crehana.com/blog/desarrollo-web/que-es-python/

Madrid, M. (2019). *Curvas de Permeabilidad Relativa Parte I: Introducción*. Obtenido de https://www.portaldelpetroleo.com/2012/12/curvas-de-permeabilidad-relativa-parte.html

Mochales, M. (28 de 09 de 2020). *Python, el lenguaje de programación más popular de 2021.* Obtenido de profile: https://profile.es/blog/python/

Odin, A. (2020). *xlwings: Usa herramientas de análisis de Python en Excel*. Obtenido de https://excelcute.com/xlwings-introduccion-excel/

Ortega, G. (2016). *COMPLETACIÓN DE POZOS.* Quito.

Palen, W., & Goodwin, A. (1996). *Increasing Production in a Mature Basin: "The Choke Model".* Milan, Italy 22-24: Society of Petroleum Engineers. Inc.

Python. (2020). *PYTHON*. Obtenido de https://lenguajesdeprogramacion.net/python/

Python Software Foundation. (2022). *collections — Container datatype*, Versión 2. Obtenido de Documentation 3.10.2: https://docs.python.org/3/library/collections.html

Quintagroup. (2020). *CANOPY - ENTORNO CIENTÍFICO Y ANALÍTICO DE PYTHON*. Obtenido de https://quintagroup.com/cms/python/canopy

Robles, M. (2021). *¿Para qué sirve la Programación?* Obtenido de https://blogs.unitec.mx/vida-universitaria/para-que-sirve-la-programacion/

Schlumberger. (2004). *Introducción al Equipo de Perforación*. Obtenido de ACADEMIA: https://www.academia.edu/6455135/Schlumberger_Introduccion_al_equipo_de_perforacion

Schlumberger. (2009). *PIPESIM.* Obtenido de LA COMUNIDAD PETROLERA: https://lacomunidadpetrolera.com/2009/01/pipesim-3.html

Schlumberger. (2014). *PIPESIM - User Guide.* Houston, USA.

Schlumberger. (2022). *Permeabilidad relativa*. Obtenido de Oilfield Glossary: https://glossary.oilfield.slb.com/es/terms/r/relative_permeability

Schlumberger. (2022). *Tubería de revestimiento*. Recuperado el 20 de 2 de 2022, de Oilfield Glossary: https://glossary.oilfield.slb.com/es/terms/c/casing

Schlumberger, S. (2008). *Simulador de Flujo Multifásico de Estado Estable PIPESIM*. Obtenido de https://www.software.slb.com/products/pipesim

SOURCEFORGE. (2021). *PythonToolKit (PTK)*. Obtenido de https://sourceforge.net/projects/pythontoolkit/

SPE. (2016). *Production forecasting system constraints*. Obtenido de https://petrowiki.spe.org/Production_forecasting_system_constraints

Spencer, J. A., & Morgan, D. T. (1998). *Application of Forecasting and Uncertainty Methods to Production.* New Orleans: Society of Petroleum Engineers. Inc.

Steel, C. (2018). *CASING Y TUBING*. Obtenido de https://www.coltubossteel.com.co/casing.html

Vargas, E. (2008). *Método de Levantamiento Artificial por Bombeo Electrosumergible (BES)*. Obtenido de https://www.monografias.com/trabajos63/levantamiento-artificial-bombeo/levantamiento-artificial-bombeo2.shtml