

Prototipo de Herramienta para la Identificación Automática de Tiempos de Operación

Durante la Perforación en un Campo de Ecopetrol

Leonardo Fabio Cortés Vergara y Santiago Herrera Parra

Trabajo de Grado para Optar el Título de Ingeniero de Petróleos

Director

Néstor Fernando Saavedra Trujillo

Magíster en Ingeniería de Petróleos

Codirector (es)

Hernán Darío Mantilla Hernández

Magíster en Geotecnia

Álvaro José Osorio Benavides

Ingeniero de Petróleos

Universidad Industrial de Santander

Facultad de Ingenierías Fisicoquímicas

Escuela de Ingeniería de Petróleos

Bucaramanga

2021

Dedicatoria

Le dedico este libro primeramente a Dios, por permitirme culminar de manera correcta mis estudios y permitirme culminar este proyecto de grado.

A mi madre Cielo Vergara, por brindarme su apoyo incondicional en todo momento, ser mi consejera en todo lugar, y por brindarme su infinito amor.

A mi padre Leonardo Cortés, por toda la enseña y sacrificios que realizo para yo poder culminar y cumplir este objetivo.

A mi otro padre Jorge Cortés, por su apoyo, su amor, sus enseñanzas, consejos, por siempre estar allí para mí.

A mi novia y compañera de vida Vivian Bueno, por siempre estar allí para mí, por apoyarme y brindarme su amor, por motivarme a salir adelante en cada momento.

A mis demás familiares por estar siempre al pendiente de todo mi desarrollo profesional.

A mi amigo y compañero de tesis Santiago Herrera por hacer realidad este objetivo poder culminar satisfactoriamente nuestro sueño.

A mis los más grandes amigos que me pudo dejar la vida universitaria, por compartir momentos inolvidables, anécdotas, horas de estudio: Fernando, Andrés, Camilo, Santiago, Carlos, Jefriey, Oscar, Daniela, Ricardo duran (Q.E.P.D). Y a todos aquellos compañeros y/o amigos que hicieron un aporte significativo en mi vida como estudiante.

LEONARDO CORTÉS

Agradecimientos

Queremos agradecer en primera instancia a nuestra alma mater la Universidad industrial de Santander, al Grupo de Investigación de Estabilidad de Pozos y a la Escuela de Ingeniería de Petróleos, por brindarles a lo largo de nuestra formación académica y profesional un entorno acogedor, propicio y con todos los recursos que pudimos tener para el desarrollo de seres humanos y profesionales íntegros y éticos. Agradecemos al Instituto Colombiano del Petróleo y los excelentes profesionales que allí laboran, por brindarnos la oportunidad de realizar nuestro trabajo de grado con el apoyo de ellos.

Agradecer a nuestro director ingeniero Néstor Saavedra, por su apoyo, comprensión, guía y paciencia en el desarrollo de este trabajo de grado.

Agradecer a los ingenieros Hernán Mantilla, Álvaro Osorio y a la ingeniera Luz Angela Bohórquez por su apoyo, tiempo invertido en reuniones, socializaciones, arreglos, aporte de ideas, contenidos y su excelente disposición para sacar adelante este trabajo de grado.

Por último, agradecer a cada uno de nuestros compañeros con los que compartimos jornadas de estudio, momentos de ocio y demás actividades que contribuyeron en nuestro objetivo e hicieron de nuestro paso por la universidad algo de no olvidar.

Tabla de Contenido

	Pág.
Introducción	13
1. Objetivos	14
1.1 Objetivo General	14
1.2 Objetivos Específicos.....	14
2. Estado del Arte.....	15
2.1 Tiempos utilizados en la perforación	15
2.1.1 Tiempos reales	15
2.1.2 Tiempo no productivo (NPT).....	16
2.1.3 Tiempo invisible (ILT)	16
2.1.4 Tiempos planos	16
2.2 Metodologías para identificación de tiempos	16
2.2.1 Límite técnico (LT).....	16
2.2.2 Best in class (BIC)	17
2.2.3 Best of best (BOB).....	17
2.2.4 Best observed (BO).....	18
2.2.5 Average well duration.....	18
2.2.6 Percentil 10 y percentil 50 (P10 - P50).....	18
2.2.7 Máximum theoretical performance (MTP).....	18
2.3 Operaciones.....	19
2.3.1 Backreaming	19

2.3.2 Circulating.....	19
2.3.3 Drilling.....	20
2.3.4 On surface	20
2.3.5 Pull out of hole (POOH)	20
2.3.6 Pull out of hole with pump (POOH/WPUMP)	20
2.3.7 Reaming	20
2.3.8 Run in hole (RIH)	20
2.3.9 Run in hole with pump (RIH/WPUMP)	20
2.3.10 Sarta quieta.....	20
2.3.11 Sliding.....	21
2.3.12 Undefined.....	21
2.3.13 Unregistred.....	21
 3. Información del Prototipo de Herramienta	 21
3.1 Variables que Miden los Sensores en Perforación.....	21
3.2 Parámetros en Perforación	22
3.2.1 Depth of the well measured (DMEA).....	22
3.2.2 Measured depth of bit (DBTM)	22
3.2.3 Revolutions per minute (RPM).....	22
3.2.4 Rate of penetration (ROP)	23
3.2.5 Torque (TRQ)	23
3.2.6 Weight on bit (WOB).....	23
3.2.7 Mud flow in (MFI).....	23
3.2.8 Hook load (HKLA)	24

3.3 Definición de los Lenguajes y Bases de Datos	24
3.4 Implementación del Algoritmo en los Lenguajes y Código del Programa	31
3.4.1 Tiempo real (Azure Cosmos DB)	32
3.4.2 Tiempo histórico (Microsoft SQL Server).....	37
3.4.3 Aplicación EcoAge web	39
4. Funcionamiento del Prototipo de Herramienta	40
4.1 Componente Excel del Prototipo de Herramienta	41
4.1.1 Excel ILT automatización.....	41
4.1.2 Software PSeInt	44
4.1.3 Desarrollo en Excel.....	49
4.1.2 Presentación en Diagrama de Flujo	60
5. Aplicación del Prototipo de Herramienta al Campo de Estudio y Resultados	62
5.1 Reseña del Campo.....	62
5.2 Aplicación	63
5.2.1 Pozo 1.....	63
5.2.2 Pozo 2.....	67
6. Conclusiones.....	70
7. Recomendaciones	72
Referencias Bibliográficas	73

Lista de Tablas

	Pág.
Tabla 1 Duración del pozo 1	63
Tabla 2 Tiempo de las operaciones del pozo 1	64
Tabla 3 Duración del pozo 2	67
Tabla 4 Tiempo de las operaciones del pozo 2	67

Lista de Figuras

	Pág.
Figura 1. Representación de las relaciones entre los tiempos de perforación	15
Figura 2. Comparación del tiempo de construcción de un pozo entre el concepto tradicional vs límite técnico.....	17
Figura 3. Diagrama de flujo de algoritmo de operaciones versión 2	31
Figura 4. Trama JSON entrada a cosmos DB	32
Figura 5. Configuración inicial de la base de datos Cosmos DB.....	33
Figura 6. Programación faImportacionJsonCosmos en C#.....	35
Figura 7. Configuración de credenciales de acceso de Cosmos DB.....	36
Figura 8. Programa de consola en C# que ejecuta algoritmo de operaciones versión 2	36
Figura 9. Ejecucion algoritmo de operaciones version 2 en POSTMAN	37
Figura 10. Tabla de datos_salida de resultados de algoritmo de operación versión 2.....	38
Figura 11. Ejecución algoritmo de operaciones versión 2 en POSTMAN	38
Figura 12. Archivos de la implementación del algoritmo de operaciones en la aplicación EcoAge Web	39
Figura 13. Archivo de la implementación del algoritmo de operaciones en la aplicación EcoAge Web	40
Figura 14. Interfaz de Macro de Excel ILT automatización	41
Figura 15. Macro de Excel ILT automatización ejecutada	42
Figura 16. Visual Basic de la Macro de Excel ILT automatización	44
Figura 17. Barra de herramientas de PSeInt	45

Figura 18. Diagrama de flujo creado en el software PSeInt	47
Figura 19. Zoom o vista ampliada de la figura 19	47
Figura 20. Algoritmo creado en el software PSeInt.....	48
Figura 21. Desarrollo en Excel prueba A.....	50
Figura 22. Desarrollo en Excel, ensayo B.....	53
Figura 23. Desarrollo en Excel, adición de la operación 'Sarta quieta' en el ensayo B	54
Figura 24. Desarrollo en Excel, archivo C.....	57
Figura 25. Desarrollo en Excel, archivo C con su última actualización	59
Figura 26. Diagrama de flujo de la base de datos del ICP	60
Figura 27. Diagrama de flujo creado y propuesto para la base de datos del ICP	61
Figura 28. Distribución porcentual de las operaciones del pozo 1	65
Figura 29. Distribución porcentual de las operaciones del pozo 2	69

Lista de Apéndices

“Los apéndices están adjuntos y puede visualizarlos en la base de datos de la biblioteca UIS”

Apéndice A. Ensayo A

Apéndice B. Ensayo B

Apéndice C. Ensayo C

Apéndice D. Aplicación en dos pozos de un campo de Ecopetrol

Resumen

Título: Prototipo de Herramienta para la Identificación Automática de Tiempos de Operación Durante la Perforación en un Campo de Ecopetrol *

Autor: Leonardo Fabio Cortés Vergara, Santiago Herrera Parra **

Palabras Clave: Operación, Prototipo de Herramienta, Perforación, Parámetros, Identificación, Lenguaje de Programación.

Descripción: Las campañas de perforación son una inversión importante para las empresas de la industria Oil&Gas, de las cuales unas resultan más exitosas o eficientes que otras en cuanto a tiempo, por ende en costos. A partir de estas se identifican oportunidades de mejora continua para conseguir buenas prácticas en cada una de las operaciones realizadas en la perforación de un pozo logrando campañas con mejores desempeños que traen consigo una reducción de tiempo y un mejor manejo del capital.

Este proyecto de grado plantea una herramienta prototipo para la identificación de las operaciones hechas en los taladros relacionadas en todo lo que compone la perforación de un pozo petrolero a partir de los parámetros que arrojan dichas operaciones durante el proceso. Así mismo se ejecutan unas pruebas de identificación de parámetros para posteriormente hacer un análisis de las operaciones arrojadas y su respectivo tiempo con lo que finalmente se presentan las conclusiones pertinentes para servir de utilidad en la búsqueda de oportunidades de mejora.

* Trabajo de Grado

** Facultad de Ingenierías Físicoquímicas. Escuela de Ingeniería de Petróleos. Director: Néstor Fernando Saavedra Trujillo. Magíster en Ingeniería de Petróleos. Codirector (es): Hernán Darío Mantilla Hernández. Magíster en Geotecnia. Álvaro José Osorio Benavides. Ingeniero de Petróleos.

Abstract

Title: Prototype of a Tool for Automatic Identification of Operation Times During Drilling in an Ecopetrol Field *

Author: Leonardo Fabio Cortés Vergara, Santiago Herrera Parra. **

Key Words: Operation, Tool Prototype, Drilling, Parameters, Identification, Programming Language.

Description: Drilling campaigns are an important investment for companies in the oil and gas industry, some of which are more successful or efficient than others in terms of time and, therefore, costs. Based on these, opportunities for continuous improvement are identified to achieve good practices in each of the operations carried out in the drilling of a well, achieving campaigns with better performance that result in a reduction of time and better capital management.

This degree project starts with the identification of the operations performed in the drills and that are immersed in everything that makes up the drilling of an oil well from the parameters that yield these operations during the process. Likewise, some identification tests are carried out to later make an analysis of the operations and their respective time, and finally the pertinent conclusions are presented to serve as useful in the search for improvement opportunities.

* Degree Work

** Physical-Chemical Engineering Faculty. Petroleum Engineering School. Director: Néstor Fernando Saavedra Trujillo. Master in Petroleum Engineering. Codirector (es): Hernán Darío Mantilla Hernández. Master in Geotechnics. Álvaro José Osorio Benavides. Petroleum Engineer.

Introducción

A lo largo de los años en la industria de los hidrocarburos ha sido fundamental para las empresas el manejo de su capital en sus inversiones y operaciones, lo que lleva consigo a tener en estas últimas unos costos bajos que signifiquen una mayor liquidez o utilidad. Estos bajos costos conllevan una mejoría progresiva y una mayor eficacia en cada uno de los procesos, segmentos o actividades que se desarrollan en la industria petrolera para llegar a este fin.

La perforación de pozos es una de las actividades donde se puede obtener mayor eficacia pues en esta los costos están directamente relacionados con el tiempo que tarde en construirse un pozo, esto significa, mejorar de alguna forma las prácticas empleadas en cada una de las actividades que se desarrollan en la perforación. En este sentido es conveniente optimizar la detección de operaciones para de esta forma tener un control en caso de que exista inconsistencia con los reportes que se realizan en campo, con el fin de llevar un registro detallado y exhaustivo que permita mejorar la eficiencia y desarrollo de la perforación, mitigando errores cometidos y no tenidos en cuenta en anteriores campañas de perforación de Ecopetrol.

El presente trabajo dividido en 4 capítulos muestra el desarrollo de un prototipo de herramienta que permite la identificación automática de las operaciones realizadas en la perforación, por ende los tiempos de estas mediante parámetros físicos obtenidos de los sensores con los que cuentan las herramientas que intervienen en este proceso. A su vez se realiza la aplicación y prueba en pozos ya perforados por Ecopetrol con el objetivo de comprobar el correcto funcionamiento del prototipo de herramienta propuesto.

1. Objetivos

1.1 Objetivo General

Proponer prototipo de herramienta automática que identifique y clasifique los tiempos de operación en la perforación de un campo de Ecopetrol.

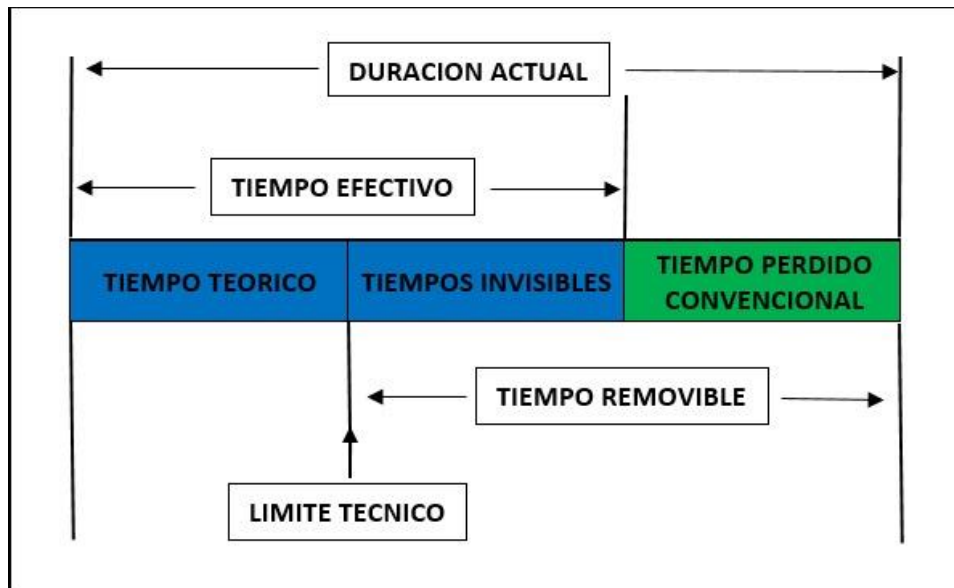
1.2 Objetivos Específicos

- Revisar los mecanismos empleados en la identificación de tiempos de operación en la perforación y seleccionar al menos uno de ellos.
- Entender y aplicar el método seleccionado para la identificación de los tiempos de operación.
- Construir el prototipo de herramienta para la identificación de los tiempos de operación en la perforación.
- Aplicar el prototipo de herramienta en un campo de Ecopetrol.

2. Estado del Arte

2.1 Tiempos utilizados en la perforación

Figura 1. Representación de las relaciones entre los tiempos de perforación



Fuente: Adaptado de Bond, D, F.; Scott, P. W; Windham, T, M. Applying Technical Limit Methodology for Step Change in Understanding and Performance. New Orleans, Louisiana, 1225 March. 1996

Dentro de las diferentes metodologías para el análisis de los tiempos de operación es necesario identificar cada uno de los tiempos presentes durante la ejecución de las actividades que se realizan en el proceso de perforación. En la figura mostrada anteriormente se evidencia la relación que tienen estos tiempos, los cuales serán definidos a continuación.

2.1.1 Tiempos reales

Se refiere a la duración total que tuvo el proceso de perforación, incluyendo todas las actividades realizadas por la cuadrilla, desde el recibimiento de equipo hasta la última cementación programada. Esta definición abarca los tiempos no productivos, tiempos invisibles y tiempos productivos que tuvo la construcción del pozo.

2.1.2 Tiempo no productivo (NPT)

En inglés No Productive Time. Son aquellos tiempos de perforación que se toman un tiempo extra al planeado, por causa de una actividad o evento externo que afecta el oficio de perforar.

2.1.3 Tiempo invisible (ILT)

Traducido del inglés “Invisible lost time”, son definidos como el tiempo perdido por ineficiencia del personal o equipo en actividades ya programadas. Su clasificación como invisible es debido a que no está considerado como tiempo no productivo, pero está oculto en los reportes de los tiempos productivos de perforación.

2.1.4 Tiempos planos

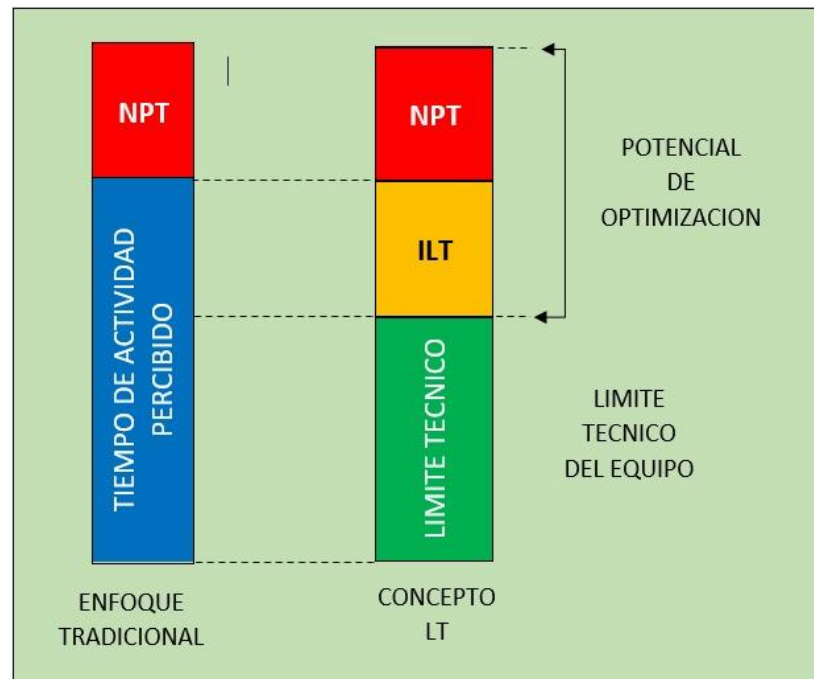
Son conocidos de manera similar a los tiempos invisibles, pero con una diferencia en las actividades a los que son atribuidos, ya que son provenientes de operaciones que se realizan sin emplear sarta de perforación.

2.2 Metodologías para identificación de tiempos

2.2.1 Límite técnico (LT)

Este concepto se introduce en la perforación de pozos como la construcción de un pozo teórico con tiempo operativo perfecto teniendo como base el conocimiento actual, la experiencia colectiva y la tecnología de diseño disponible.

El límite técnico tiene como objetivo principal reducir los costos y tiempos de perforación, estandarizar las actividades y herramientas a usar en la operación, con el fin de obtener un máximo desempeño posible.

Figura 2. Comparación del tiempo de construcción de un pozo entre el concepto tradicional vs límite técnico

Fuente: Adaptado de Cantarelli, E.; Kojadinovic, N.; Pham, K.; Youssef, G.; Invisible Lost Time Reduction and Drilling Risk Management Optimization in United Arab Emirates Onshore Field.

2.2.2 Best in class (BIC)

En español, Mejor en la clase. Compara el desempeño propio de una empresa con el de cualquier operador con mejores prácticas de perforación en una serie de pozos similares.

El BIC no refleja el mejor desempeño que se pueda alcanzar, ya que usa en su comparación rendimientos anteriormente logrados, sin embargo, suministra una medida contra el rendimiento actual.

2.2.3 Best of best (BOB)

En español, Mejor de lo mejor. Es la medida del mejor rendimiento interno de una empresa en el desarrollo de una operación o actividad en una serie de pozos similares. Tiene como limitante comparar el rendimiento obtenido en pozos perforados por ellos mismos.

2.2.4 Best observed (BO)

En español, Mejor observado. Es el mejor tiempo observado por algún operador en una serie de pozos comparables.

2.2.5 Average well duration

En español, Duración promedio pozo. Se define como la duración media de una serie de pozos sustancialmente similares.

2.2.6 Percentil 10 y percentil 50 (P10 - P50)

El estudio de percentiles es muy común cuando se analizan estadísticamente conjuntos grandes de datos.

Las razones por la cual se aconseja trabajar con rangos son:

- Las cuadrillas trabajan más eficientemente cuando se apunta a un rango que cuando lo hacen apuntando a un límite.
- En campos maduros resulta difícil romper una marca propuesta.
- Pasos pequeños han dado resultados mayores que cambios abruptos.

2.2.7 Máximum theoretical performance (MTP)

En español, Máximo desempeño teórico. Es definido como el tiempo mínimo en el que un pozo podría ser perforado y su cálculo consiste en factores físicos que restringen o limitan el desarrollo de la perforación, tales como, la energía específica de la roca, límites operacionales, número de tuberías de revestimiento, tamaño del hueco, litología, profundidades, etc.

Del mismo modo hay que tener las siguientes consideraciones para el desarrollo del modelo:

- ROP: limitada por la potencia del taladro.
- No hay viajes de limpieza.

- No se repasan las secciones del hueco.
- Perfecta cementación y corrida de revestimiento
- Tiempos de circulación.

2.3 Operaciones

Una vez se lleva a cabo el alistamiento de la locación, el izaje de la torre y la instalación de los demás equipos requeridos, se inicia la perforación en la que es necesario tener un plan y un programa de perforación, hecho posteriormente a la realización de diversos estudios geológicos e ingenieriles, que son la base para hacer de esta una operación eficaz y segura. En este sentido en el proceso de la perforación son varias acciones o actividades determinantes las que se deben realizar para seguir de la mejor manera dicho plan de perforación con el fin de construir un pozo estable y óptimo para su posterior producción.

A partir de esto se definieron las operaciones que son parte de la herramienta y engloban los parámetros de perforación con los que se trabajaron, las cuales son las más comunes e importantes en el taladro:

2.3.1 Backreaming

Es la práctica de circular y rotar la sarta de perforación mientras simultáneamente se saca del pozo y a su vez rectificar este mismo. Se complementa con la operación Reaming.

2.3.2 Circulating

Ciclo del fluido de perforación donde se mueve desde la superficie, fuera de los tanques de lodo, a través de la tubería, arriba del anular y de regreso a los tanques de lodo.

2.3.3 Drilling

Operación de penetrar el subsuelo rotando, circulando, y bajando sarta de perforación con el objetivo de desarrollar un pozo seguro y conectar el yacimiento con la superficie.

2.3.4 On surface

Operación definida cuando la sarta de perforación se encuentra en el piso del taladro de perforación o cuando no se ha iniciado la perforación de un pozo.

2.3.5 Pull out of hole (POOH)

Retirar o sacar la sarta de perforación del pozo sin flujo de lodo.

2.3.6 Pull out of hole with pump (POOH/ WPUMP)

Retirar o sacar la sarta de perforación del pozo mientras se realiza la circulación de lodo.

2.3.7 Reaming

Hace referencia a las actividades de rectificar el pozo, agrandar el agujero al tamaño deseado, enderezar las patas de perro (dog legs), y ayudar en la perforación direccional.

2.3.8 Run in hole (RIH)

Entrar o bajar tubería conectada entre sí, ya sea tubería de perforación, tubería de producción, etc., en el pozo sin circulación de lodo.

2.3.9 Run in hole with pump (RIH/ WPUMP)

Entrar o bajar tubería conectada entre sí, ya sea tubería de perforación, tubería de producción, etc., en el pozo con flujo de lodo a través de este.

2.3.10 Sarta quieta

Hace referencia a la acción de mantener la sarta de perforación quieta en una profundidad determinada sin accionar el equipo en superficie generando torque, RPM, flujo de lodo, etc., en ella.

2.3.11 Sliding

Es la actividad de perforar con un motor de lodo que hace rotar la broca en el fondo del pozo sin rotar la sarta de perforación desde la superficie, por lo que no se van a presentar torque ni RPM's (revoluciones por minuto) en el piso de la torre de perforación.

2.3.12 Undefined

Es referente a que no se cumple ningún criterio condicional en las herramientas de perforación, por lo tanto, no hay operación determinada.

2.3.13 Unregistred

Se presenta cuando las variables arrojadas por las herramientas de perforación tienen valores nulos, como por ejemplo (-999,25).

3. Información del Prototipo de Herramienta

3.1 Variables que Miden los Sensores en Perforación

En la actividad de perforar existen diversos sensores que miden y garantizan que todo el proceso sea llevado a cabo de forma correcta y precisa, estos sensores generalmente están ubicados en el taladro, broca o caseta del lodero. En muchos casos para la toma de medidas por parte de los sensores se agrupan datos de diferente fuente con el objetivo de generar menos incertidumbre y error.

Aunque bien dicho los sensores son por lo general precisos y automatizados, aun se aplican y utilizan técnicas para la toma de parámetros que no son estrictamente arrojadas por un sensor, resultan en técnicas un poco ortodoxas pero eficientes que no dan un valor exacto pero si muy acercado a la realidad que está ocurriendo en el momento de perforar.

3.2 Parámetros en Perforación

En el proceso de perforación inciden muchos factores que de una u otra forma facilitan y ayudan a tener un mayor control y eficacia en la operación, estos factores en el proceso son denominados parámetros de perforación, los cuales permiten ya sea tener un control preciso sobre la sarta y la broca como de los fluidos que intervienen al momento de perforar. Los parámetros en un taladro que cumple un proceso de perforar son muchos y muy variados, para efectos prácticos del desarrollo del prototipo de esta herramienta se tuvieron en cuenta 8 parámetros que afectaban las operaciones seleccionadas, las siglas empleadas de estos parámetros son: DMEA, DBTM, RPM, ROP, TRQ, WOB, MFI, HKLA.

3.2.1 Depth of the well measured (DMEA)

Se considera una medición a lo largo de la dirección en la perforación, esta indica la profundidad del agujero y es identificada en registros como una curva que no aumenta, DMEA representa en todos los casos el máximo de profundidad alcanzado por la broca, mecha o barrena hasta la hora actual.

3.2.2 Measured depth of bit (DBTM)

Es un indicativo de la posición en tiempo real de la broca, mecha o barrera a lo largo de un agujero y de este modo a través de registros apunta a una coordenada curvilínea (subida y bajada de la sarta a lo largo de la perforación).

3.2.3 Revolutions per minute (RPM)

Se define como el número de rotaciones que se tienen en el taladro, y pueden ser tomadas en el top drive o en la broca si esta cuenta con un motor de fondo, son adquiridas a lo largo de un minuto y va directamente relacionado con la velocidad de giro que puede alcanzar el objeto y con el tipo de formación que se tiene.

3.2.4 Rate of penetration (ROP)

Se define como la velocidad con la que la broca, mecha o barrena puede romper la roca que se encuentra por debajo de la misma y de ese modo profundizar el pozo. Esta velocidad se indica habitualmente en unidades de pies por hora o metros por hora.

3.2.5 Torque (TRQ)

Es una medida del torque giratorio aplicado a la sarta de perforación, tomada particularmente en la mesa giratoria o en el cable del motor de accionamiento superior, en base a los registros se pueden observar casos de picos altos en las lecturas debido a actividades rutinarias como también operación de difícil perforación. Durante la perforación el valor promedio de torque es 0 “zero” y tiende a aumentar.

3.2.6 Weight on bit (WOB)

Es la cantidad de fuerza ejercida hacia abajo sobre la broca y resulta de todos los componentes de la sarta de perforación, Este parámetro es calculado en medida a ciertos datos en conjunto como HKLA, RPM, Sarta de perforación, etc. Y el operador lo establece para que coincida con las características de la formación que se desea perforar.

3.2.7 Mud flow in (MFI)

Hace referencia a la cantidad de fluido por unidad de tiempo que se encuentra en proceso de utilización al momento de desarrollar la perforación, también se determina como la cantidad de fluido de perforación que se encuentra circulando en el pozo y logra el objetivo de estabilizar el hueco.

3.2.8 Hook load (HKLA)

Se encarga de medir la tensión en el cable de trabajo cuando se encuentra estirado, configurado previamente de forma manual por un operador. Se define también como la diferencia entre el peso de toda la cuerda y la fuerza aplicada a él.

3.3 Definición de los Lenguajes y Bases de Datos

Los lenguajes utilizados para escribir programas de computadoras que puedan ser entendidos por ellas se denominan lenguajes de programación. También se dice que un programa es un conjunto de órdenes o instrucciones que resuelven un problema específico basado en un lenguaje de programación. Los lenguajes de programación se clasifican según su base desarrollo y su uso en:

- a) Lenguaje basado en cálculos numéricos en lo que se mencionan el Fortan, Maple, Matlab y Algol.
- b) Lenguajes para Negocios como Cobol.
- c) Lenguajes para la inteligencia Artificial como Prolog, Adal, Lisp y Logo y 18
- d) lenguajes para sistemas como C y el ensamblador.

También se pueden clasificar en tres grandes categorías: a) Máquinas, b) Bajo nivel y Alto nivel.

a) Los Lenguajes Máquina: Son aquellos cuya instrucción son directamente entendibles por la computadora y no necesitan traducción posterior para que la UCP, pueda entender y ejecutar el programa. La programación en lenguaje máquina es difícil, por ello se necesitan lenguajes que permitan simplificar este proceso. Los lenguajes de bajo nivel han sido diseñados para ese fin.

b) Los Lenguajes de Programación de Alto Nivel: Son los más utilizados como lenguajes de Programación, en los que las instrucciones o sentencias a la computadora son escritas con palabras similares a los lenguajes humanos ejemplos de estos lenguajes de alto nivel son (C y Ada, BASIC, COBOL, SNOBOL, LISP Y PROLOG, APL Y FORTAN, PASCAL.) en general lenguaje inglés, como lo es el caso de Visual Basic, lo que facilita la escritura y la fácil comprensión por el programador. En el sistema se utilizará el siguiente lenguaje, en sus seis versiones, Visual Basic ha pasado de ser el lenguaje de programación más simple de Microsoft Windows a un entorno de desarrollo sumamente complejo, capaz de proporcionar cualquier tipo de aplicación, desde las utilidades más sencillas a grandes aplicaciones cliente / servidor para “n” usuarios. [Bonnet, 1992].

En el mundo de la programación existen diferentes tipos y formas de desarrollarla, siendo algunas más eficaces que otras o inclusive algunas más funcionales en cierto tipo requerido, acompañado esto de bases de datos que de una u otra forma facilitan la implementación de códigos y algoritmos para una correcta ejecución.

3.3.1 JSON

JavaScript Object Notation es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

3.3.2 C#

C# es un lenguaje orientado a objetos sencillo, moderno, amigable, intuitivo y fácilmente legible que ha sido diseñado por Microsoft con el ambicioso objetivo de recoger las mejores características de muchos otros lenguajes, fundamentalmente Visual Basic, Java y C++, y combinarlas en uno sólo en el que se unan la alta productividad y facilidad de aprendizaje de Visual Basic con la potencia de C++.

Aunque con la beta actualmente disponible del lenguaje no se consiguen prestaciones en velocidad tan altas como usando el C++ tradicional, Microsoft asegura que en futuras versiones se irá aumentando la eficiencia del lenguaje, gracias fundamentalmente a la posibilidad de generar dinámicamente código optimizado para aprovechar las características de la máquina sobre la que se ejecute el programa.

Quizás el más directo competidor de C# es Java, lenguaje con el que guarda un enorme parecido en su sintaxis y características. En este aspecto, es importante señalar que C# incorpora muchos elementos de los que Java carece (sistema de tipos homogéneo, propiedades,

indexadores, tablas multidimensionales, operadores redefinibles etc.) y que según los benchmarks realizados la velocidad de ejecución del código escrito en C# es ligeramente superior a su respectiva versión en Java.

A continuación, se recoge de manera resumida las principales características de C#:

- Dispone de todas las características propias de cualquier lenguaje orientado a objetos: encapsulación, herencia y polimorfismo.
- Ofrece un modelo de programación orientada a objetos homogéneo, en el que todo el código se escribe dentro de clases y todos los tipos de datos, incluso los básicos, son clases que heredan de `System.Object` (por lo que los métodos definidos en esta son comunes a todos los tipos del lenguaje).
- Permite definir estructuras, que son clases un tanto especiales: sus objetos se almacenan en pila, por lo que se trabaja con ellos directamente y no referencias al montículo, lo que permite accederlos más rápido. Sin embargo, esta mayor eficiencia en sus accesos tiene también sus inconvenientes, fundamentalmente que el tiempo necesario para pasarlas como parámetros a métodos es mayor (hay que copiar su valor completo y no sólo una referencia) y no admiten herencia (aunque sí implementación de interfaces).
- Es un lenguaje fuertemente tipado, lo que significa se controla que todas las conversiones entre tipos se realicen de forma compatible, lo que asegura que nunca se acceda fuera del espacio de memoria ocupado por un objeto. Así se evitan frecuentes errores de programación y se consigue que los programas no puedan poner en peligro la integridad de otras aplicaciones.

- Tiene a su disposición un recolector de basura que libera al programador de la tarea de tener que eliminar las referencias a objetos que dejen de ser útiles, encargándose de ello éste y evitándose así que se agote la memoria porque al programador olvide liberar objetos inútiles o que se produzcan errores porque el programador libere áreas de memoria ya liberadas y reasignadas.
- Incluye soporte nativo para eventos y delegados. Los delegados son similares a los punteros a funciones de otros lenguajes como C++ aunque más cercanos a la orientación a objetos, y los eventos son mecanismos mediante los cuales los objetos pueden notificar de la ocurrencia de sucesos. Los eventos suelen usarse en combinación con los delegados para el diseño de interfaces gráficas de usuario, con lo que se proporciona al programador un mecanismo cómodo para escribir códigos de respuesta a los diferentes eventos que puedan surgir a lo largo de la ejecución de la aplicación. (pulsación de un botón, modificación de un texto, etc.)
- Incorpora propiedades, que son un mecanismo que permite el acceso controlado a miembros de una clase tal y como si de campos públicos se tratasen. Gracias a ellas se evita la pérdida de legibilidad que en otros lenguajes causa la utilización de métodos Set() y Get() pero se mantienen todas las ventajas de un acceso controlado por estos proporcionada.
- Permite la definición del significado de los operadores básicos del lenguaje (+, -, *, &, ==, etc.) para nuestros propios tipos de datos, lo que facilita enormemente tanto la legibilidad de las aplicaciones como el esfuerzo necesario para

escribirlas. Es más, se puede incluso definir el significado del operador [] en cualquier clase, lo que permite acceder a sus objetos tal y como si fuesen tablas.

A la definición de este último operador se le denomina indizador, y es especialmente útil a la hora de escribir o trabajar con colecciones de objetos.

- Admite unos elementos llamados atributos que no son miembros de las clases sino información sobre éstas que podemos incluir en su declaración. Por ejemplo, indican si un miembro de una clase ha de aparecer en la ventana de propiedades de Visual Studio.NET, cuáles son los valores admitidos para cada miembro en ésta, etc. [URL05, 2004].

3.3.3 Bases de datos

Una base de datos o un conjunto de datos en reconocido es un conjunto de datos que pertenecen a un mismo modelo o contexto de aplicación y que se encuentran almacenados de forma precisa y de manera sistemática para su uso final. Dicho esto, es corrector afirmar que una biblioteca es una base de datos compuestas por libros y/o documentos escritos que se encuentran anexados para una consulta sustancial. Para efectos del presente y con la llegada de la revolución tecnológica, podemos ya denominar bases de datos como formatos virtuales o electrónicos que permiten de cierta forma ofrecer un gran catálogo de soluciones al problema de tener espacio en el que almacenar datos.

3.3.3.1 Azure cosmos DB

Azure Cosmos DB es una base de datos NoSQL escalable y distribuida globalmente totalmente gestionada. Nos permite usar varios tipos de modelos de datos, como documentos, clave-valor o grafos. También, permite almacenar datos con y sin esquema definido y realizar consultas en el lenguaje SQL.

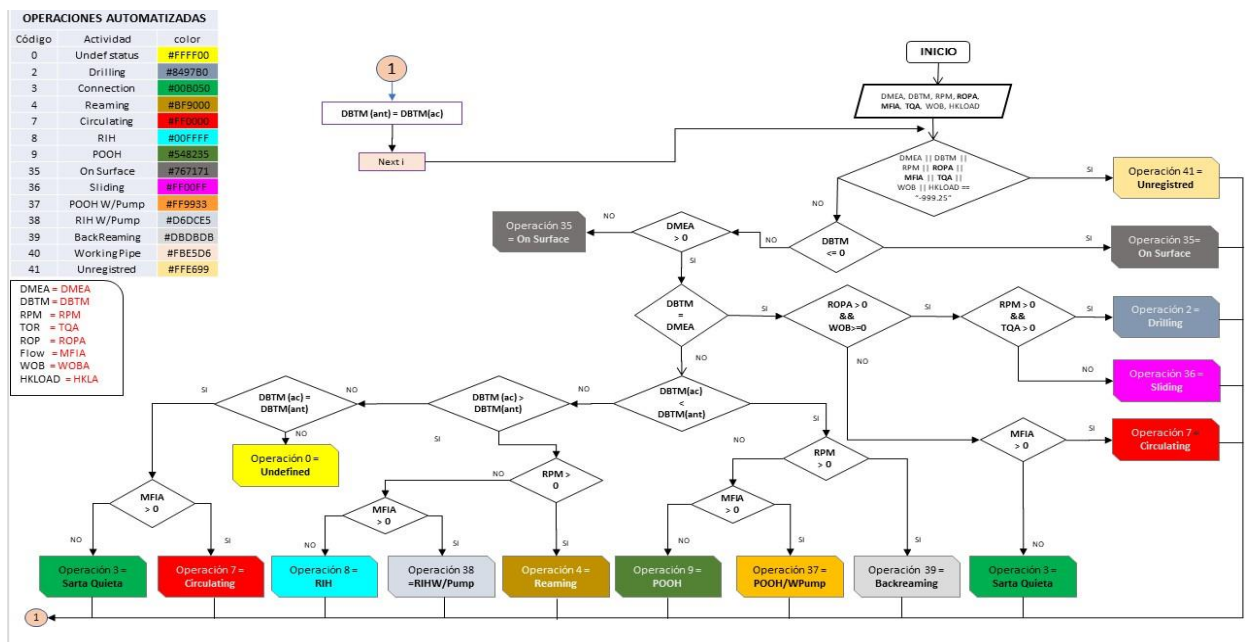
Cosmos DB tiene varias opciones para almacenar y acceder a los datos. Presenta una serie de API's disponibles para que poder elegir la que más convenga en función del proyecto que se desea implementar.

- **SQL:** Es la API por defecto en Cosmos DB. Usa el lenguaje SQL para realizar las consultas a los datos y almacenar documentos JSON. También nos permite leer valores especificando su id y clave de partición, lo que reduce el tiempo de consulta.
- **MongoDB:** Esta API soporta el protocolo de MongoDB, por lo que es compatible directamente con estos clientes con unos cambios mínimos.
- **Cassandra:** API compatible con el protocolo de **Cassandra** y los drivers de Apache CQLv4. Generalmente, bastará con cambiar en nuestra aplicación la cadena de conexión para realizar la migración.
- **Table:** Compatible con las aplicaciones que interaccionan con Azure Tables.
- **Gremlin:** Compatible con el lenguaje de consulta Gremlin y datos de grafos con el framework Apache TinkerPop (Gremlin)

3.4 Implementación del Algoritmo en los Lenguajes y Código del Programa

El algoritmo de operaciones es un proceso que se realiza cuando se reciben los datos (históricos o en tiempo real) que tiene como objetivo principal identificar 14 operaciones de acuerdo las variables de entrada DMEA, DBTM, RPM, TOR, ROP, FLOW, WOB y HKLOAD. El trabajo de Bohórquez, B. L. es parte sustancial en la transformación, a través de la base de datos, de la información que se recibe para ser leída o ejecutada por el prototipo de herramienta creado en este proyecto y que se mostrará más adelante. A continuación, se visualiza el diagrama de flujo implementado en MS SQL Server (Tiempo histórico) y en COSMOSDB (Tiempo real).

Figura 3. Diagrama de flujo de algoritmo de operaciones versión 2

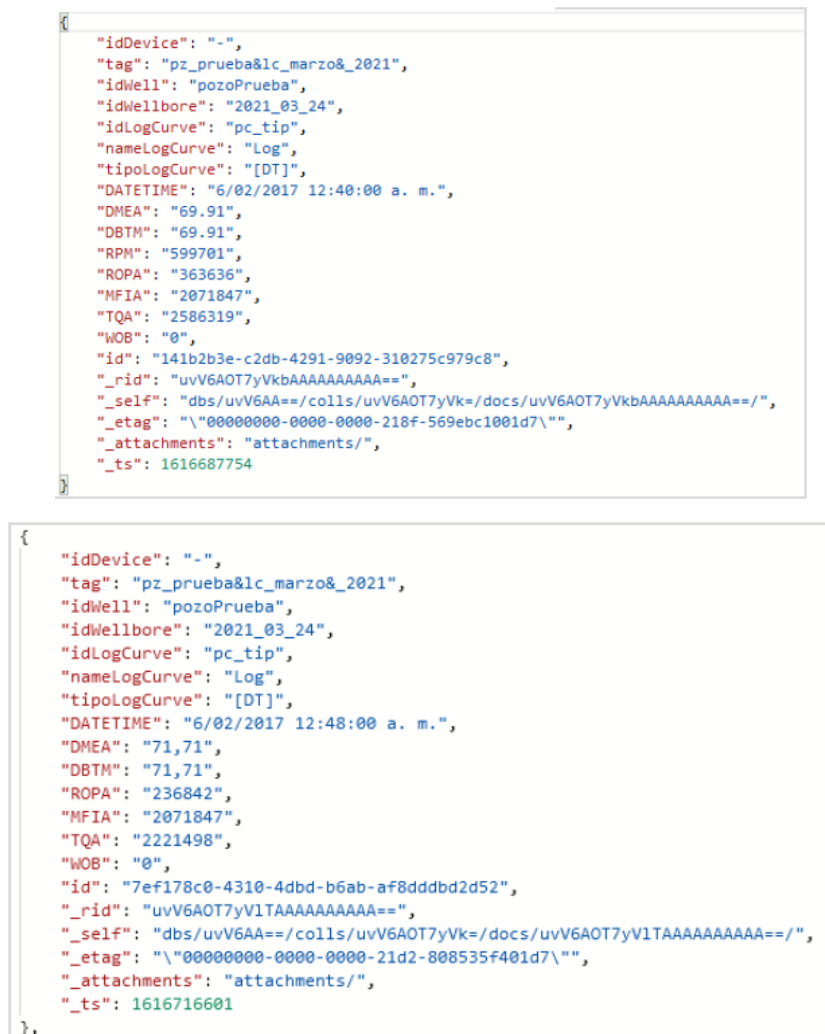


Fuente: Tomado de BOHORQUEZ, B. L.; TIEMPO REAL (COSMOS BD) IMPLEMENTACIÓN DE VERSIÓN 2 DE ALGORITMO DE OPERACIONES EN ENTORNO AZURE TIEMPO HISTÓRICO (SQL SERVER). 2021

3.4.1 Tiempo real (Azure Cosmos DB)

Para la implementación en tiempo real del algoritmo de operaciones se debe tener en cuenta la estructura de la trama de JSON que recibe del streaming analytics (ver Figura 4). Cuando un dato de entrada su valor es NULL, lo que permite que no tenga inconvenientes en tomar el primer dato y muestre una operación errónea, este no se evidencia en la trama JSON como se observa en la trama inferior de la Figura 4.

Figura 4. Trama JSON entrada a cosmos DB



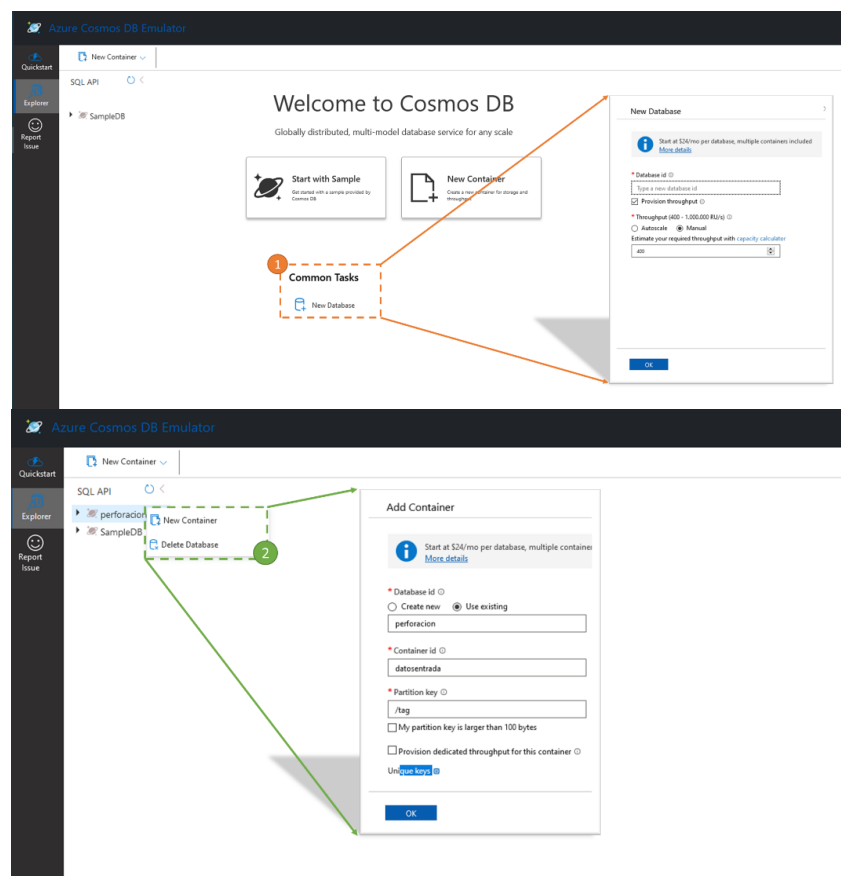
```
{
  "idDevice": "-",
  "tag": "pz_prueba&lc_marzo&_2021",
  "idWell": "pozoPrueba",
  "idWellbore": "2021_03_24",
  "idLogCurve": "pc_tip",
  "nameLogCurve": "Log",
  "tipoLogCurve": "[DT]",
  "DATETIME": "6/02/2017 12:40:00 a. m.",
  "DMEA": "69.91",
  "DBTM": "69.91",
  "RPM": "599701",
  "ROPA": "363636",
  "MFIA": "2071847",
  "TQA": "2586319",
  "WOB": "0",
  "id": "141b2b3e-c2db-4291-9092-310275c979c8",
  "_rid": "uvV6AOT7yVkbAAAAAAAAA==",
  "_self": "dbs/uvV6AA==/colls/uvV6AOT7yVk=/docs/uvV6AOT7yVkbAAAAAAAAA==/",
  "_etag": "\"00000000-0000-0000-218f-569ebc1001d7\"",
  "_attachments": "attachments/",
  "_ts": 1616687754
}
```

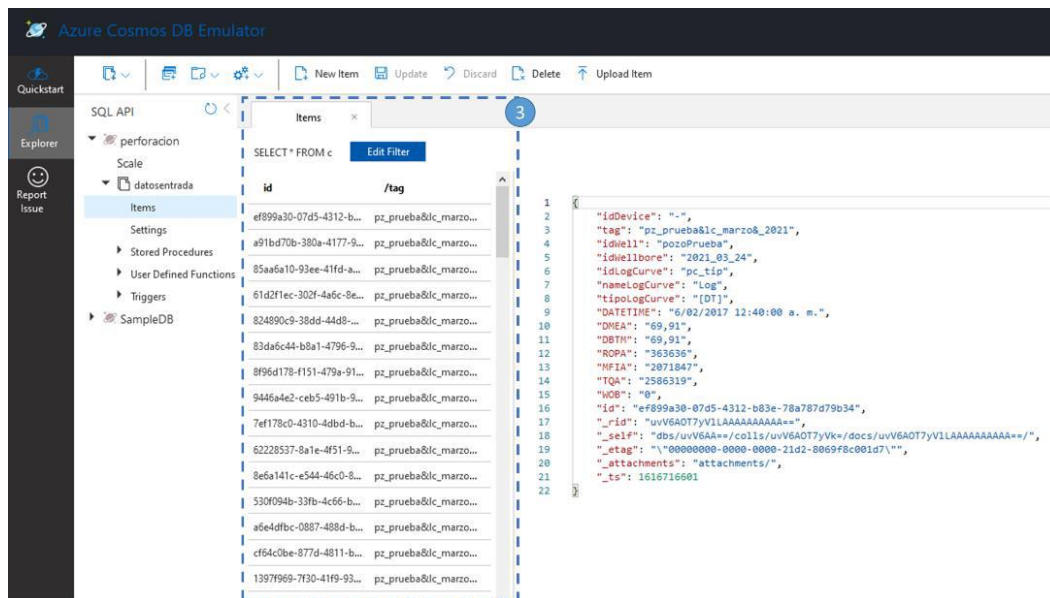
```
{
  "idDevice": "-",
  "tag": "pz_prueba&lc_marzo&_2021",
  "idWell": "pozoPrueba",
  "idWellbore": "2021_03_24",
  "idLogCurve": "pc_tip",
  "nameLogCurve": "Log",
  "tipoLogCurve": "[DT]",
  "DATETIME": "6/02/2017 12:48:00 a. m.",
  "DMEA": "71,71",
  "DBTM": "71,71",
  "ROPA": "236842",
  "MFIA": "2071847",
  "TQA": "2221498",
  "WOB": "0",
  "id": "7ef178c0-4310-4dbd-b6ab-af8dddbd2d52",
  "_rid": "uvV6AOT7yV1TAAAAAAAAA==",
  "_self": "dbs/uvV6AA==/colls/uvV6AOT7yVk=/docs/uvV6AOT7yV1TAAAAAAAAA==/",
  "_etag": "\"00000000-0000-0000-21d2-808535f401d7\"",
  "_attachments": "attachments/",
  "_ts": 1616716601
},
```

Fuente: Tomado de BOHORQUEZ, B, L.; TIEMPO REAL (COSMOS BD) IMPLEMENTACIÓN DE VERSIÓN 2 DE ALGORITMO DE OPERACIONES EN ENTORNO AZURE TIEMPO HISTÓRICO (SQL SERVER). 2021

Debido a que no se cuenta con una cuenta de Microsoft Azure se realizó esta actividad de forma local, en la Figura 5 se observa en la pantalla superior como se crea una nueva base de datos, la cual despliega una ventana modal en donde se nombra la base de datos para este proyecto, que para efectos prácticos se denominará: perforación. En la pantalla media se observa en el panel izquierdo que aparece el icono con el nombre de la base de datos creada y al hacer clic derecho sale el menú que permite crear el container el cual se denominará: datosentrada y tendrá como PartitionKey: /tag. Finalizando la Figura 5 se despliega el container datosentrada el cual permite crear los ítems el cual son los documentos que cada uno de ellos contiene una trama de datos.

Figura 5. Configuración inicial de la base de datos Cosmos DB





Fuente: Tomado de BOHORQUEZ, B, L.; TIEMPO REAL (COSMOS BD) IMPLEMENTACIÓN DE VERSIÓN 2 DE ALGORITMO DE OPERACIONES EN ENTORNO AZURE TIEMPO HISTÓRICO (SQL SERVER). 2021

Una vez arquitectura de la Nube de Azure para el proyecto de EcoAge Web, el paso continuo es la emulación del transporte de los datos, estando en la Nube se hace a través del Stremming Analytcs, para hacer las pruebas una vez configurada la base de datos de Cosmos DB con la misma configuración de la arquitectura de la Nube de Azure para el proyecto de EcoAge Web, el paso continuo es la emulación del transporte de los datos, estando en la Nube se hace a través del Stremming Analytcs, para hacer las pruebas requeridas y tener una muestra de datos significativa se creó el programa por consola en C# denominado faImportacionJsonCosmos que se muestra en la Figura 6.

Figura 6. Programación faImportacionJsonCosmos en C#

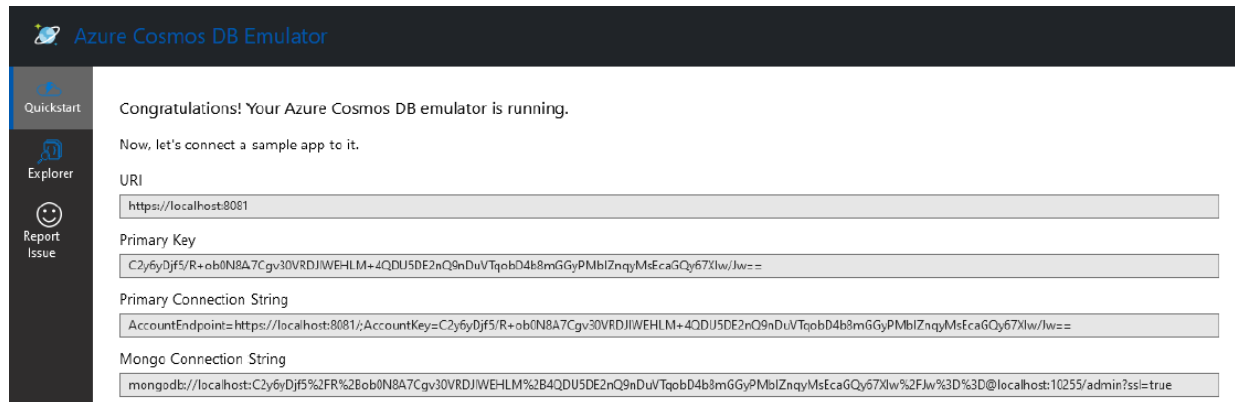
```

1  using System;
2  using System.Threading.Tasks;
3  using Microsoft.Azure.Cosmos;
4  using Microsoft.Azure.Documents.Client;
5  using Newtonsoft.Json.Linq;
6
7  namespace faImportacionJsonCosmos
8  {
9      2 referencias
10     class Program
11     {
12         private string EndpointUrl = "https://localhost:8081";
13         private string PrimaryKey = "C2y6yDjff5/R+ob0N8A7Cgv30VRDJIWEHLH+4QDU5DE2nQ9nDuVTqobD4b8mGGyPMbIZnqyMsEcaGQy67XIw/Jw==";
14         private string databaseId = "perforacion";
15         private string containerId = "datosentrada";
16
17         1 referencia
18         public async Task GetStartedDemoAsync()
19         {
20             DocumentClient client = new DocumentClient(new Uri(EndpointUrl), PrimaryKey);
21             Uri collectionLink = UriFactory.CreateDocumentCollectionUri(databaseId, containerId);
22             string[] lines = System.IO.File.ReadAllLines(@"D:\files_json\datos_completos.json");
23             foreach (string line in lines)
24             {
25                 var details = JObject.Parse(line);
26                 await client.CreateDocumentAsync(collectionLink, details);
27             }
28
29             0 referencias
30             public static async Task Main(string[] args)
31             {
32                 try
33                 {
34                     Console.WriteLine("Beginning operations...\n");
35                     Program p = new Program();
36                     await p.GetStartedDemoAsync();
37                 }
38                 catch (CosmosException de)
39                 {
40                     Exception baseException = de.GetBaseException();
41                     Console.WriteLine("{0} error occurred: {1}", de.StatusCode, de);
42                 }
43                 catch (Exception e)
44                 {
45                     Console.WriteLine("Error: {0}", e);
46                 }
47                 finally
48                 {
49                     Console.WriteLine("End of demo, press any key to exit.");
50                     Console.ReadKey();
51                 }
52             }
53         }
54     }
55 }

```

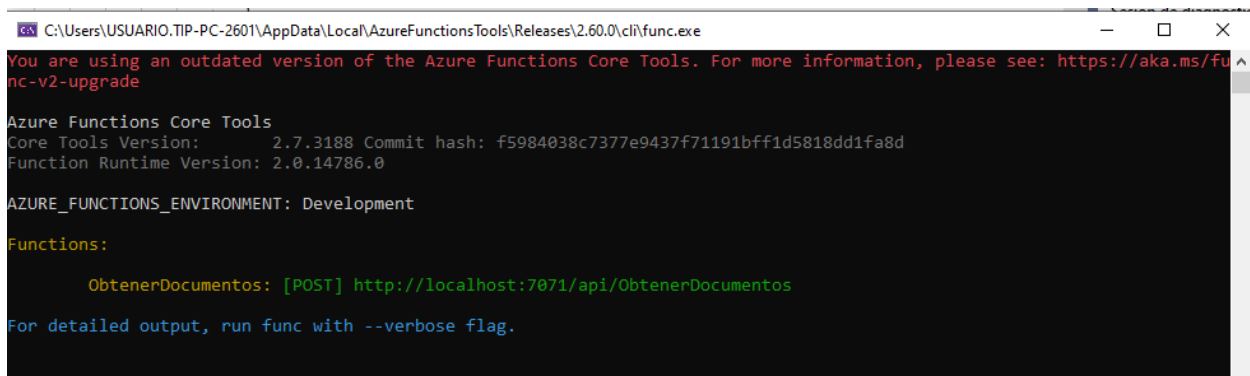
Fuente: Tomado de BOHORQUEZ, B. L.; TIEMPO REAL (COSMOS BD) IMPLEMENTACIÓN DE VERSIÓN 2 DE ALGORITMO DE OPERACIONES EN ENTORNO AZURE TIEMPO HISTÓRICO (SQL SERVER). 2021

Desde la línea 11 a la línea 14 se evidencia los parámetros de configuración de la base de datos de Cosmos DB la cual se puede extraer de la pestaña Quickstart que se visualiza en la Figura 7.

Figura 7. Configuración de credenciales de acceso de Cosmos DB

Fuente: Tomado de BOHORQUEZ, B, L.; TIEMPO REAL (COSMOS BD) IMPLEMENTACIÓN DE VERSIÓN 2 DE ALGORITMO DE OPERACIONES EN ENTORNO AZURE TIEMPO HISTÓRICO (SQL SERVER). 2021

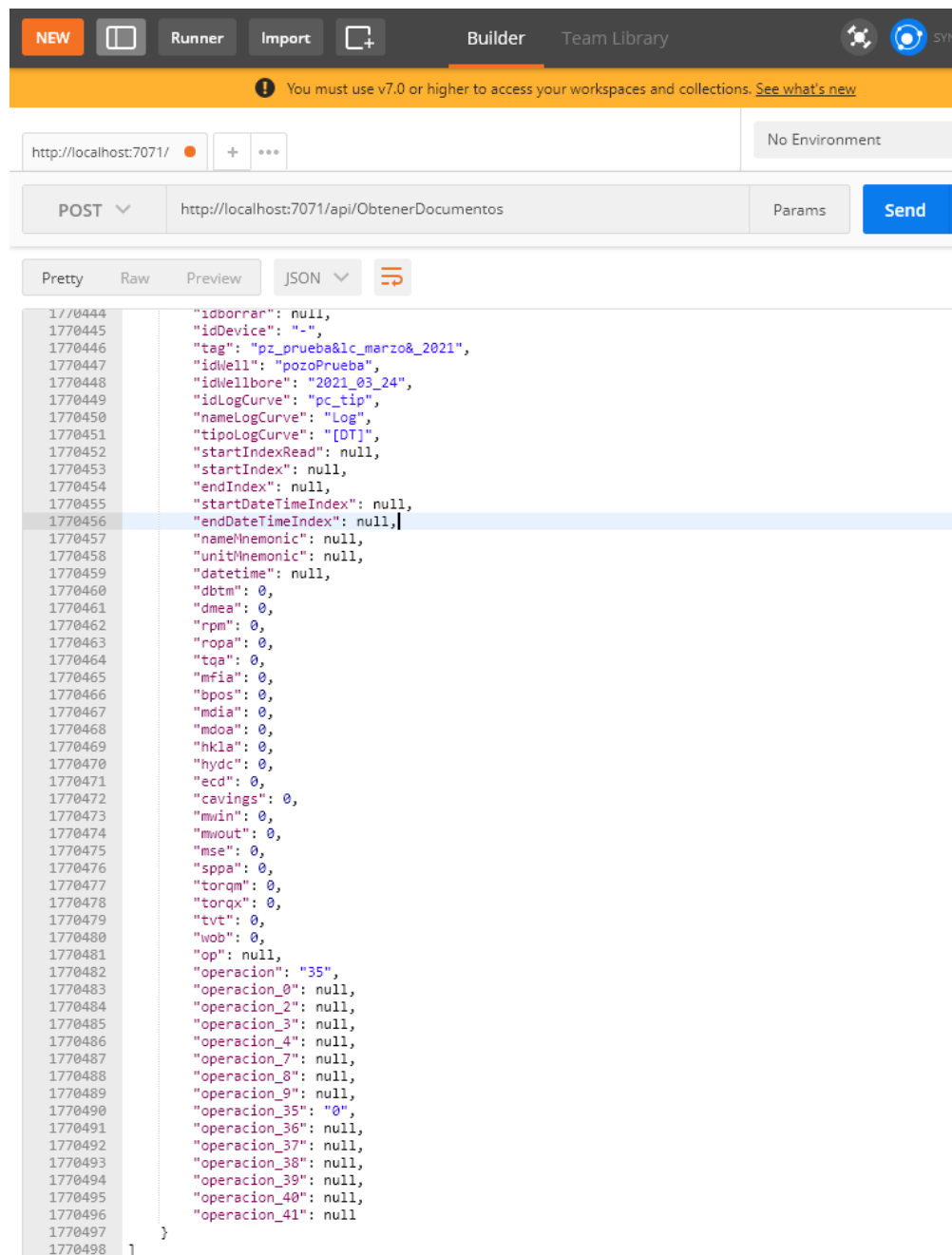
La implementación del algoritmo de operaciones se hace mediante el programa en C# `faOperacionPerforacion` una vez implementado el diagrama de flujo (Ver Figura 3) se visualiza una ventana de comando (Ver Figura 8) en donde muestra la url en método POST que debe ser ejecutada para la obtención de los resultados de salida del algoritmo de operaciones.

Figura 8. Programa de consola en C# que ejecuta algoritmo de operaciones versión 2

Fuente: Tomado de BOHORQUEZ, B, L.; TIEMPO REAL (COSMOS BD) IMPLEMENTACIÓN DE VERSIÓN 2 DE ALGORITMO DE OPERACIONES EN ENTORNO AZURE TIEMPO HISTÓRICO (SQL SERVER). 2021

Mediante el programa POSTMAN se ejecuta la URL <http://localhost:7071/api/ObtenerDocumentos>, en la Figura 9 se evidencia la trama resultante de la ejecución de la URL.

Figura 9. Ejecucion algoritmo de operaciones version 2 en POSTMAN



Fuente: Tomado de BOHORQUEZ, B, L.; TIEMPO REAL (COSMOS BD) IMPLEMENTACIÓN DE VERSIÓN 2 DE ALGORITMO DE OPERACIONES EN ENTORNO AZURE TIEMPO HISTÓRICO (SQL SERVER). 2021

3.4.2 Tiempo histórico (Microsoft SQL Server)

Para tiempo histórico se actualizo el procedimiento `dbo.ejecutar_etl_operaciones` con la implementación del algoritmo de operaciones versión 2, el resultado de la ejecución se evidencia

3.4.3 Aplicación EcoAge web

Debido a los requerimientos de cargue de información externa (archivos .las) , los cuales no provienen del proceso de la telemetría se hizo necesario implementar en la aplicación web una función que permita la identificación de operaciones con los datos que el usuario final seleccione desde la interfaz gráfica del aplicativo web de EcoAge, por consiguiente se crea en el backend dos archivos que permite la ejecución de este proceso.

Figura 12. Archivos de la implementación del algoritmo de operaciones en la aplicación EcoAge Web

```

// utilities.js
exports.AlgoritmoOperaciones = function(DBTH, DHEA, DBTH, RPM, ROPA, MFIA, TQA, MOB)
{
    var Operacion = '';
    var CampoOper = '';
    var salida = '';

    if (DBTH == -999.25 || DHEA == -999.25 || MFIA == -999.25 || RPM == -999.25 || TQA == -999.25 || ROPA == -999.25 || MOB == -999.25)
    {
        Operacion = 41;
        CampoOper = 'Unregistered';
        //continue Final;
    }

    if (DBTH <= 0)
    {
        Operacion = 35;
        CampoOper = 'On Surface';
        //continue Final;
    }
    else
    {
        if (DHEA > 0)
        {
            if (DBTH == DHEA)
            {
                if ((ROPA > 0) && (TQA > 0))
                {
                    if ((RPM > 0) && (TQA > 0))
                    {
                        Operacion = 2;
                        CampoOper = 'Drilling';
                    }
                    else
                    {
                        Operacion = 36;
                        CampoOper = 'Sliding';
                    }
                }
                else
                {
                    if (MFIA > 0)
                    {
                        Operacion = 7;
                        CampoOper = 'Circulating';
                    }
                    else
                    {
                        Operacion = 3;
                        CampoOper = 'Connection';
                    }
                }
            }
        }
        else
        {
            //continue Final;
        }
    }
}

// algoritmooperaciones.js
var estructuras = require("../controllers/estructuras");
var httpsigs = require("../core/httpsigs");
var util = require("../core/utilities");

exports.routerAlgoritmoOperaciones = function (req, resp) {
    switch (req.method) {
        case "POST":
            if (req.url === "/algoritmooperaciones") {
                var reqBody = '';
                req.on("data", function (data) {
                    reqBody += data;
                });
                if (reqBody.length > 1e7) {
                    httpsigs.show413(req, resp);
                }
                req.on("end", function () {
                    var jsons = JSON.parse(reqBody);
                    var DBTH = jsons.DBTH;
                    var DHEA = jsons.DHEA;
                    var DBTH = jsons.DBTH;
                    var RPM = jsons.RPM;
                    var ROPA = jsons.ROPA;
                    var MFIA = jsons.MFIA;
                    var TQA = jsons.TQA;
                    var MOB = jsons.MOB;
                    var resultado = util.AlgoritmoOperaciones(DBTH, DHEA, DBTH, RPM, ROPA, MFIA, TQA, MOB);
                    httpsigs.sendJson2(req, resp, resultado);
                });
            }
        default:
            httpsigs.show404(req, resp);
            break;
    }
    if ('OPTIONS' === req.method) {
        httpsigs.send200(req, resp);
    }
    else {
        httpsigs.show405(req, resp);
        break;
    }
}

```

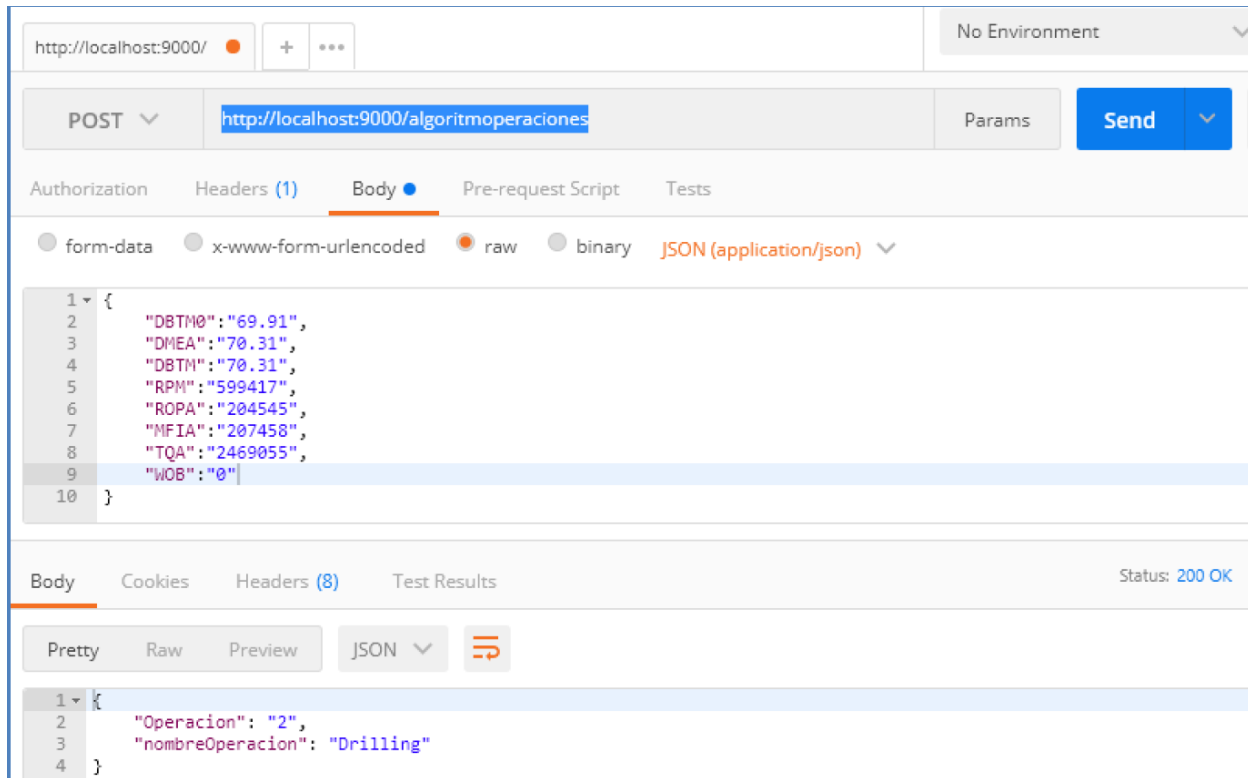
Archivos de la implementación del algoritmo de operaciones en la aplicación EcoAge Web

Fuente: Tomado de BOHORQUEZ, B, L.; TIEMPO REAL (COSMOS BD) IMPLEMENTACIÓN DE VERSIÓN 2 DE ALGORITMO DE OPERACIONES EN ENTORNO AZURE TIEMPO HISTÓRICO (SQL SERVER). 2021

En la Figura 12 se visualiza en la pantalla izquierda el archivo utilities.js donde la función exports.AlgoritmoOperaciones tiene la codificación del proceso para determinar la operación específica de un conjunto de parámetros. En la pantalla derecha se ve el código para la ejecución del proceso que se hace mediante una URL que tiene los parámetros requeridos.

En la Figura 13 se evidencia en la ventana POSTMAN el envío de los parámetros tipo JSON a la url <http://localhost:9000/algoritmooperaciones> y como respuesta en la parte inferior la respuesta obtenida del proceso de operaciones ejecutado.

Figura 13. Archivo de la implementación del algoritmo de operaciones en la aplicación EcoAge Web



Fuente: Tomado de BOHORQUEZ, B, L.; TIEMPO REAL (COSMOS BD) IMPLEMENTACIÓN DE VERSIÓN 2 DE ALGORITMO DE OPERACIONES EN ENTORNO AZURE TIEMPO HISTÓRICO (SQL SERVER). 2021

4. Funcionamiento del Prototipo de Herramienta

La creación y/o propuesta del prototipo de herramienta comprendió el pasaje por diferentes documentos o archivos compartidos para su análisis y oportunidad de mejora de manejo de los parámetros de perforación (ya mencionados anteriormente) para identificar operaciones, tiempos, y profundidad durante la realización de la perforación, así como el manejo

“amateur” de programas que contribuyeran a las ideas y formas que pudiera adoptar el prototipo de herramienta; del mismo modo se tuvo el proceso con archivos generados propiamente donde se llevó progresivamente la creación para cumplir los objetivos. Lo de descrito en las líneas anteriores está reflejado a continuación:

4.1 Componente Excel del Prototipo de Herramienta

4.1.1 Excel *ILT* automatización

El primer paso dentro de este desarrollo consistió en la revisión y análisis de la macro de Excel, que tiene por nombre el de este subtitulo, compartida dentro del convenio de investigación GIEP - ICP. Este archivo representa un primer desarrollo por parte de ese equipo de trabajo donde también es posible identificar qué operación se está llevando a cabo según los parámetros que se introduzcan.

Figura 14. Interfaz de Macro de Excel *ILT* automatización

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
2						Total	Motor	Table													
3	WID	Date	TIME	DBTM	DMEA	RPMA	MMRP	RPMS	ROPA	TQ	MFI	SPPI	WOB	HKLI	Well Activity Code	Activity Code Run					
4																					
5																					
6																					
7																					
8																					
9																					
10																					
11																					
12																					
13																					
14																					
15																					
16																					
17																					
18																					
19																					
20																					
21																					
22																					
23																					
24																					
25																					

Fuente: Geomecánica y perforación ICP

Como un archivo inicial y macro de identificación la intención fue correrla para poder observar su funcionamiento exterior en cuanto a la identificación correcta o no de las

operaciones, y así mismo interiormente entender el código de programación con el que está formada.

En lo que respecta a la primera parte se cargó una parte de la data suministrada de un pozo A de Ecopetrol. Para este caso no se alimentaron todas las columnas de parámetros de la macro, sin embargo, esto no afectó la ejecución en su programación, los parámetros a los cuales se les colocaron data fueron los siguientes:

- DBTM
- DMEA
- RPMA
- ROPA
- TQ
- MFI
- WOBI
- HKLI

Figura 15. Macro de Excel ILT automatización ejecutada

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15					
2					Total	Motor	Table												
	WID	Date	TIME	DBTM	DMEA	RPMA	MMRP	RPMS	ROPA	TQ	MFI	SPPI	WOBI	HKLI	Well Activity Code	Activity Code Run			
3																			
16				4992.36	4992.36	2.994.489			218.18	136.303.964	8.701.155		4.54	163.81	Unregistered	Drilling	RIH		
17				4994.5	4994.5	2.972.406			264	121.131.726	8.693.785		8.26	160.09	Unregistered	POOH	RIHW Pump		
18				4996.7	4996.7	2.968.272			266.67	136.956.026	8.693.785		8.71	159.64	Unregistered	RIH	Sliding		
19				4998.17	4998.17	2.980.019			266.67	99.499.085	8.715.902			0.170.8	Unregistered	Drilling	Undefined		
20				4996.63	4998.17	2.980.019			266.67	19.859.028	8.715.902			0.174.43	Unregistered	Undefined	Working Pipe		
21				4981.28	4998.17	2.175.291			266.67	-12.204.344	8.701.163			0.186.68	Unregistered	Undefined			
22				4979.32	4998.17	2.178.976			266.67	-18.429.087	8.715.902			0.194.76	Unregistered	Drilling			
23				4981.4	4998.17	2.177.125			266.67	-12.001.362	8.708.499			0.178.42	Unregistered	Undefined			
24				4991.14	4998.17	2.177.125			266.67	-11.444.511	8.708.499			0.166.26	Unregistered	Undefined			
25				4985.77	4998.17	2.183.158			266.67	-17.368.419	8.715.886			0.173.61	Unregistered	Drilling			
26				4997.02	4998.17	2.427.018			266.67	4.099.534	8.715.902			0.169.44	Unregistered	Drilling			
27				4998.36	4998.36	2.446.106			266.67	2.724.487	8.715.902			0.171.89	Unregistered	Drilling			
28				4998.77	4998.77	2.451.832			266.67	7.397.012	8.699.996		1.72	166.63	Unregistered	Drilling			
29				4999.37	4999.37	2.177.127			31.93	17.308.032	8.708.508			3.165.35	Unregistered	RIH			
30				4999.97	4999.97	2.177.127			31.93	17.688.735	8.708.508			163.81	Unregistered	RIH			
31				5000.58	5000.58	2.177.127			72	19.142.816	8.708.508			3.36	164.99	Unregistered	RIH		
32				5001.19	5001.19	2.177.127			73.47	15.198.958	8.708.558			2.81	165.54	Unregistered	RIH		
33				5001.79	5001.79	2.177.127			73.47	20.176.623	8.708.508			3.36	164.99	Unregistered	Drilling		
34				5002.36	5002.36	2.175.289			71.29	22.977.979	8.701.155			2.81	165.54	Unregistered	RIH		
35				5002.93	5002.93	2.177.125			71.29	16.603.033	8.708.499			2.27	166.08	Unregistered	Drilling		
36				5003.55	5003.55	2.176.821			67.92	22.235.058	8.707.286			2.63	165.72	Unregistered	RIH		
37				5004.17	5004.17	2.177.125			75.79	1.821.439	8.708.499			0.91	167.44	Unregistered	Drilling		
38				5004.79	5004.79	2.175.289			75.79	19.065.264	8.701.155			4.54	163.81	Unregistered	RIH		
39				5005.15	5005.15	2.175.289			59.02	17.677.667	8.701.155			2.81	165.54	Unregistered	RIH		
40				5005.74	5005.74	2.178.976			59.02	17.568.697	8.715.902			0.168.99	Unregistered	Drilling			
41				5006.37	5006.37	2.394.288			72.73	14.830.061	9.577.152		1.09	167.26	Unregistered	Drilling			
42				5007	5007	2.175.289			72.73	19.333.282	8.701.155			3.09	165.26	Unregistered	POOH		
43				5007.66	5007.66	2.177.127			75	21.081.951	8.708.508			2.63	165.72	Unregistered	Drilling		
44				5008.33	5008.33	2.177.127			79.91	1.117.088	8.708.508			1.73	168.44	Unregistered	RIH		
45				5009	5009	2.175.291			79.91	25.748.372	8.701.163			5.36	164.81	Unregistered	POOH		
46				5009.72	5009.72	2.177.127			80.9	1.245.713	8.708.508			3.73	166.44	Unregistered	Drilling		
47				5010.03	5010.03	2.180.003			61.45	28.546.996	8.720.012			2.36	167.81	Unregistered	Drilling		

Fuente: Geomecánica y perforación ICP

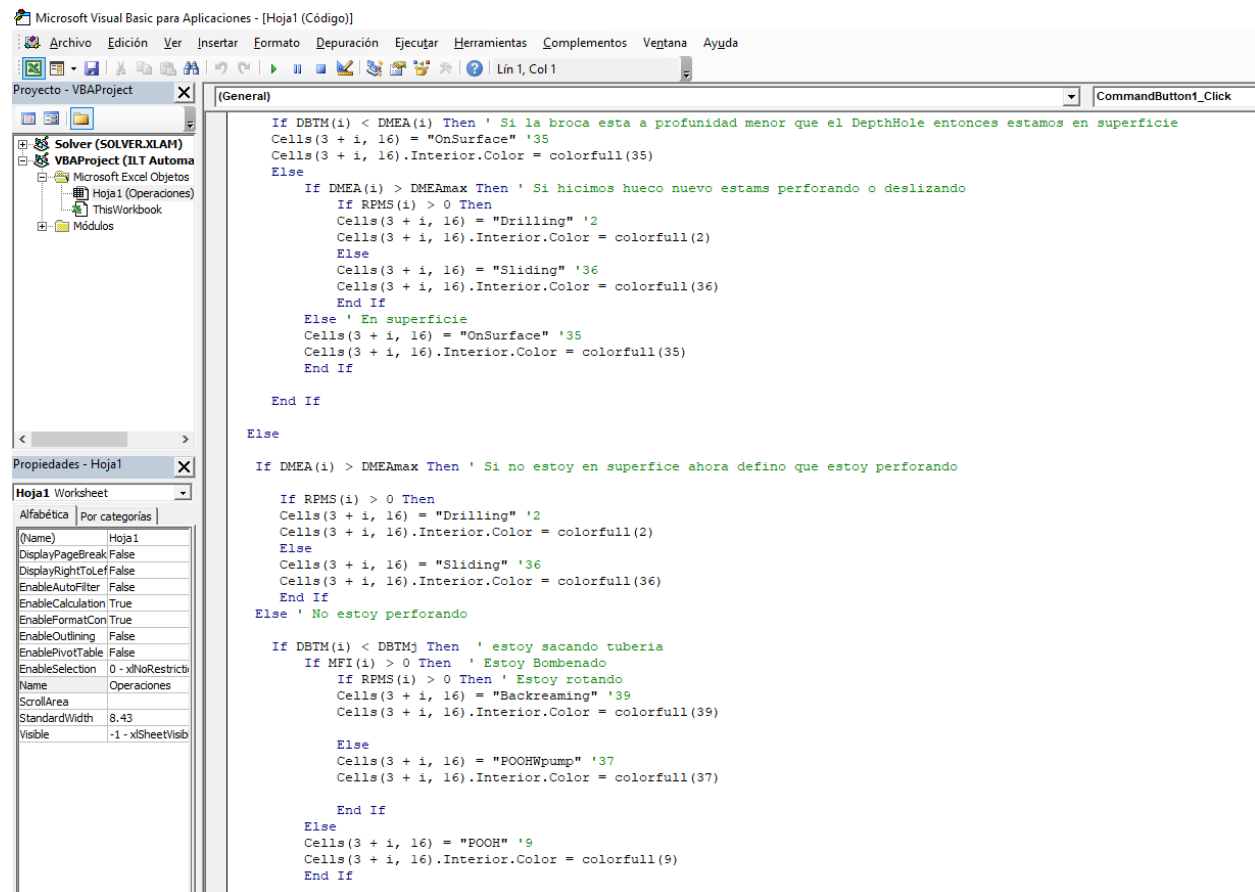
Como se evidencia en las filas contenidas entre las dos líneas rojas (41 a 45) de la figura 15 este fue uno de los segmentos donde se encontró una identificación errónea de las operaciones de acuerdo con los parámetros correspondientes. Se analizó la secuencia de las operaciones que se fueron identificando y como según esto venía dándose la construcción del pozo, a partir de esto se sacaron las siguientes observaciones para tenerlas como puntos importantes en las oportunidades de mejora en el prototipo a desarrollar:

- De acuerdo con la profundidad de la fila 41 (5006.37) no hubo concordancia con pasar de Drilling a POOH (Put Out Of Hole), fila 42, con una profundidad mayor a la perforada en la última operación antes de POOH.
- Ya que la operación anterior estuvo errónea, la siguiente (fila 43) a POOH no fue lógicamente correcta al pasar a Drilling con una profundidad mayor a la anterior sin haberla perforado.
- La operación de la fila 44, RIH (Run in Hole), fue incorrecta ya que no es posible correr o bajar la tubería a una profundidad mayor a la perforada anteriormente.
- De acuerdo con la profundidad de la fila 45 y la operación que fue identificada antes, no es posible que al aumentar la profundidad la operación identificada haya sido POOH, por lo tanto, siguiendo con el orden lógico de las actividades la operación identificada debió haber sido RIH o Drilling.

Posterior a lo analizado en la primera parte, se procedió al interior de la macro, es decir, al componente Visual Basic para poder observar su programación o codificación para el funcionamiento que se ve cada vez que se ingresa al archivo. El propósito de estudiar las diferentes líneas de código para asignar un color predeterminado de acuerdo con la operación a identificar, los condicionales para saber si es nulo o no un parámetro al momento de entrada y la

definición de los rangos de valores para cada parámetro en esta función, así como los condicionales sucesivos comenzando por los valores de profundidad de los cuales la macro parte para identificar la operación correspondiente; fue tenerlas como punto de partida, primera noción, y ejemplo para los siguientes avances que se debían hacer en el desarrollo del proyecto.

Figura 16. Visual Basic de la Macro de Excel ILT automatización



Fuente: Geomecánica y perforación ICP

4.1.2 Software PSeInt

Se utilizó el software libre de programación básica PSeInt, en el cual se intentó plasmar las primeras ideas que fueron generadas y los puntos a corregir a partir del análisis de la macro anteriormente explicada. Esta herramienta se utilizó ya que tiene una interfaz sencilla de

entender y manejar para personas que no están familiarizadas o apenas tienen unos conceptos primarios de la programación.

Figura 17. Barra de herramientas de PSeInt



Fuente: Software PSeInt

Se definieron cuales de los parámetros iban a intervenir o tenían lugar en cada una de las operaciones y con qué valores con el fin de correr un código de programación o algoritmo completo dónde pasara a identificar qué operación se cumplía de acuerdo con los datos recibidos. Esto se hizo con el manejo del lenguaje lógico, por lo tanto, las operaciones se determinaron de la siguiente manera:

- **Drilling:** $DBTM > 0$; $DBTM_{actual} > DBTM_{anterior}$;
 $DMEA > 0$; $DMEA_{actual} > DBTM_{anterior}$;
 $DMEA = DBTM$;
 $RPM > 0$; $ROP > 0$; $FlowIn > 0$; $TQ > 0$; $WOB > 0$; $Hookload > 0$
- **Reaming:** $DBTM > 0$; $DBTM_{actual} > DBTM_{anterior}$;
 $DMEA > 0$; $DMEA_{actual} = DMEA_{anterior}$;
 $RPM > 0$; $FlowIn > 0$; $TQ > 0$; $Hookload > 0$
- **Backreaming:** $DBTM > 0$; $DBTM_{actual} < DBTM_{anterior}$;
 $DMEA > 0$; $DMEA_{actual} = DMEA_{anterior}$;
 $RPM > 0$; $ROP = 0$; $FlowIn > 0$; $Hookload > 0$
- **Sliding:** $DBTM > 0$; $DBTM_{actual} > DBTM_{anterior}$;
 $DMEA > 0$; $DMEA_{actual} > DMEA_{anterior}$;

$$DMEA = DBTM ;$$

$$WOB > 0 ; ROP > 0 ; FlowIn > 0 ; TQ > 0 ; Hookload > 0$$

- **Circulating:** $DBTM > 0 ;$

$$DMEA > 0 ;$$

$$FlowIn > 0 ; Hookload > 0$$

- **POOH:** $DBTM > 0 ; DBTM \text{ actual} < DBTM \text{ anterior} ;$

$$DMEA > 0 ; DMEA \text{ actual} = DMEA \text{ anterior} ;$$

$$Hookload > 0$$

- **POOHWPump:** $DBTM > 0 ; DBTM \text{ actual} < DBTM \text{ anterior} ;$

$$DMEA > 0 ; DMEA \text{ actual} = DMEA \text{ anterior} ;$$

$$FlowIn > 0 ; Hookload > 0$$

- **RIH:** $DBTM > 0 ; DBTM \text{ actual} > DBTM \text{ anterior} ;$

$$DMEA > 0 ; DMEA \text{ actual} = DMEA \text{ anterior} ;$$

$$Hookload > 0$$

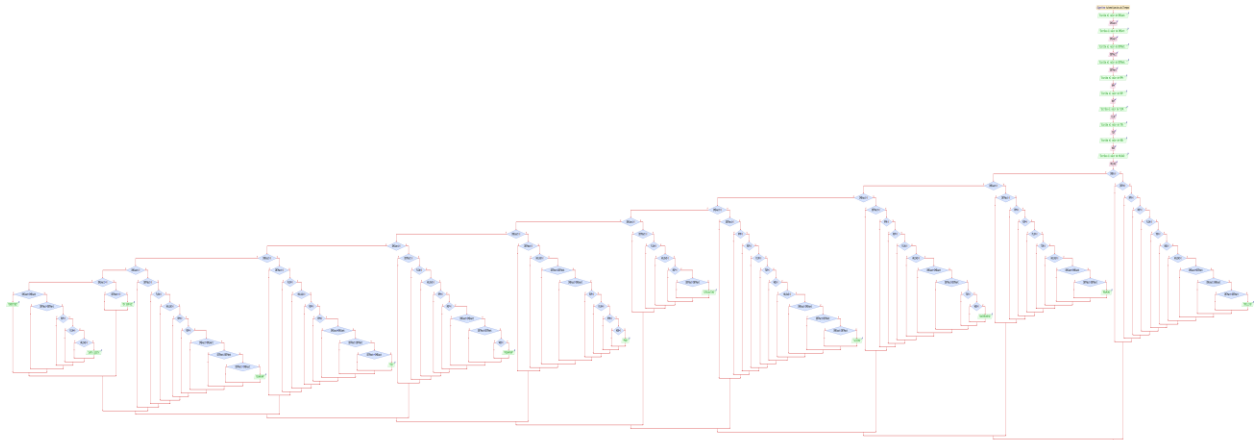
- **RIHWPump:** $DBTM > 0 ; DBTM \text{ actual} > DBTM \text{ anterior} ;$

$$DMEA > 0 ; DMEA \text{ actual} = DMEA \text{ anterior} ;$$

$$FlowIn > 0 ; Hookload > 0$$

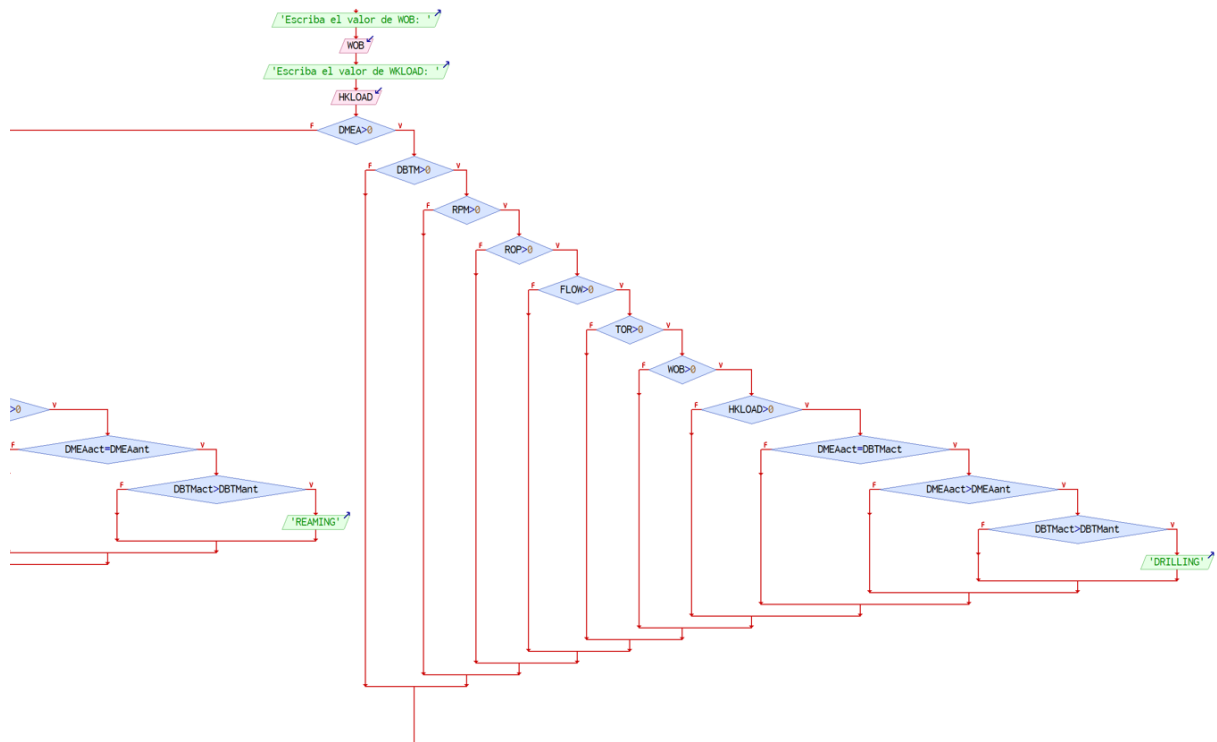
Una vez se definieron las operaciones, se escribió y ejecutó todo el algoritmo de programación en el formato PSeInt desde su ingreso de datos y/o validación hasta el final de cada uno de los ciclos para la identificación de la operación. La ejecución se representó de dos maneras, mediante el algoritmo y mediante un diagrama de flujo o flujograma.

Figura 18. Diagrama de flujo creado en el software PSeInt



Fuente: Autores

Figura 19. Zoom o vista ampliada de la figura 19

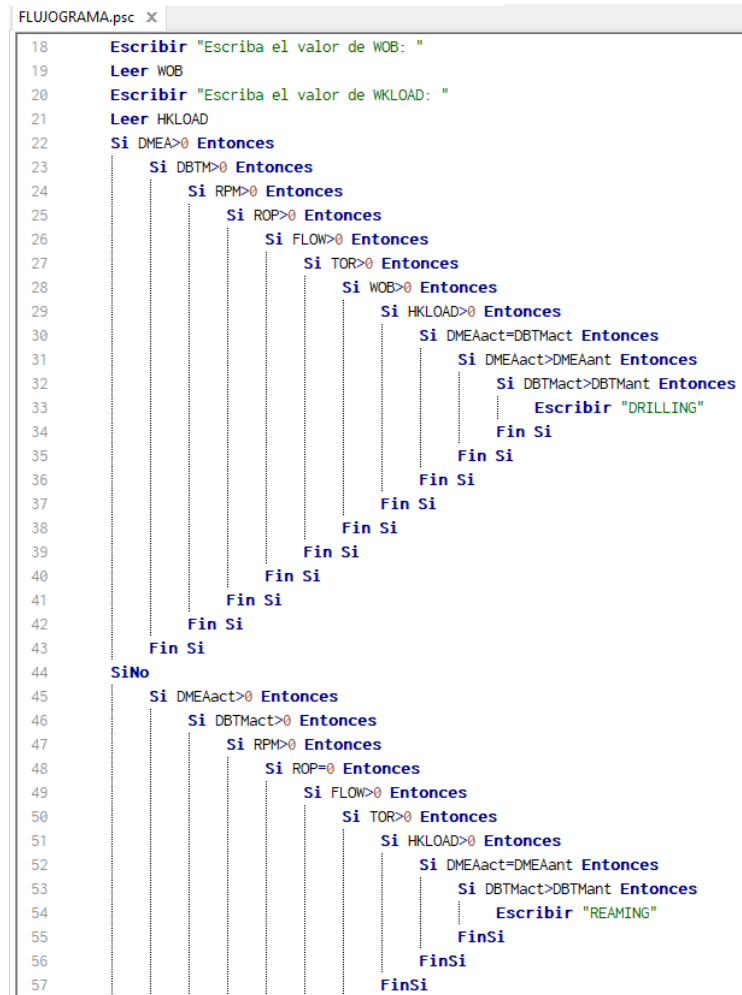


Fuente: Autores.

Como se puede apreciar en las figuras 18 y 19 el resultado fue un proceso largo para cada operación, a partir de esto se concluyó que no era recomendable continuar desarrollando la herramienta de esta forma y con la ayuda de este software de programación ya que el algoritmo hace ciclos sucesivos, repetitivos y constantes en la búsqueda de identificar cada operación, por

lo que, como se dijo anteriormente, terminaba siendo un proceso extenso en términos de programación; en adición a esto se observó que el algoritmo al tomar valores de ingreso uno a uno no discretizaba de manera correcta y pasaba de una operación a otra sin lograr identificar acertadamente.

Figura 20. Algoritmo creado en el software PSeInt



Fuente: Autores.

Sin embargo, la utilización del software PSeInt contribuyó en esta etapa para el estudio y definición de cada una de las operaciones a incluir en la herramienta, así como para hacer una mejora continua en este desarrollo.

4.1.3 Desarrollo en Excel

Posterior al trabajo realizado en PSeInt se optó por intentar reflejar la propuesta de algoritmo con las observaciones u oportunidades de mejora encontradas anteriormente en Excel siendo una opción sencilla y práctica para trasladar el algoritmo a otra herramienta o interfaz en caso de así requerirlo. Esta “sección” donde se abordó la realización del prototipo de herramienta en Excel resultó ser el más acertado, práctico y donde se pudo conseguir un avance constante y significativo en la entrega del objetivo.

El proceso constó de un cambio progresivo en cuanto a cómo se definía cada operación a medida que se fue estudiando detalladamente cada una de ellas, del mismo modo se analizó como introducir otras “operaciones” para seguir con el patrón de identificación correspondiente a parámetros inútiles dentro de la data, y así mismo se determinó la posibilidad de añadir o agrupar otras operaciones con el fin de cubrir todo lo que respecta a la construcción de un pozo petrolero. A continuación, se muestra en detalle la evolución del proceso descrito anteriormente:

En el punto de partida (prueba, ensayo o archivo A) se trató una porción pequeña de los datos adquiridos de un pozo A del año 2019 de Ecopetrol, ya que iba a ser la primera de todas las pruebas. Así mismo, con el lenguaje previamente descrito que iba a definir cada operación se construyó una tabla rellena de ‘SI’ y ‘NO’, ‘SI’ para cuando la operación correspondiera a los parámetros tratados en esa fila de los datos, y en caso contrario se iba a ver el ‘NO’.

Figura 21. Desarrollo en Excel prueba A

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
DMEA	DBTM	RPM	ROP	FLOWin	TQ	WOB	Hload		Drilling	Reaming	Backreaming	Sliding	Circulating	POOH	POOHwpump	RIH	RIHwpump	Onsurface
5132.41	5130.75	0	97.71	331.5286	75.8412	0	44.8379		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5132.41	5130.75	0	97.71	728.9664	40.9798	0	197.9428		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5132.41	5130.75	0	97.71	852.2522	27.0533	0	195.8144		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5132.41	5130.75	0	97.71	852.1338	54.4515	0	199.3394		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5132.41	5130.808	0	97.71	851.9385	41.4348	0	202.55		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5132.41	5132.292	80.0394	97.71	852.3501	5255.291	0	197.3382		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5132.41	5134.099	80	97.71	852.015	5235.581	0	196.5817		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5133.99	5135.325	80	940	852.4469	5299.725	0	197.3641		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5135.05	5135.726	79.9607	151.5024	851.1941	5468.724	12.0261	180.7039		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5135.996	5136.913	80	160	851.5836	5312.462	0.4801	193.8545		SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
5136.604	5138.273	80	135.8587	850.8051	6042.902	10.2306	182.4994		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5137.953	5139.63	80	160.7561	850.6873	5674.975	11.5121	181.2179		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5139.307	5140.912	80.0256	160	851.0752	5839.539	12.476	180.254		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5140.625	5142.23	80	150.0488	850.5688	6950.808	18.1629	174.5671		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5141.917	5143.502	80.0394	148.5951	850.0625	5971.044	17.1121	175.6179		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5143.223	5144.833	80	161.9512	850.5689	7080.235	18.8015	173.9285		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5144.503	5146.198	79.9354	164.2683	850.2566	6766.718	16.1159	176.6141		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5145.883	5147.435	80	166.8415	850.3747	6930.438	14.1617	178.5683		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5147.187	5148.583	79.896	166.8415	850.2568	7410.135	14.2259	178.5041		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
5148.283	5149.674	80	124.039	850.7629	6626.989	11.7445	180.9855		NO	NO	NO	NO	NO	NO	NO	NO	NO	NO

Fuente: Autores.

Como se observa en la figura 21, a las operaciones ya descritas en este capítulo se le añadió la ‘operación’ On Surface, esta se definió inicialmente así:

- **On Surface:** DMEA ≥ 0 ;

$$DBTM = 0$$

Del mismo modo fue evidente que el objetivo de esta prueba no fue cumplido, ya que al introducir los datos la codificación solo identificó tres operaciones de las diez programadas, y en efecto, solo ocho filas de datos de veintiuno ingresadas, de lo cual se pudo decir que es ilógico y erróneo puesto que en esa porción de datos existen más operaciones a identificar si el lenguaje lógico utilizado hubiera sido correctamente elaborado.

Dando continuidad a lo trabajado y sumando los errores a anotar se realizó el ensayo B en Excel del prototipo de herramienta. En este se cambió la data ingresada por la de un pozo B, perforado en el 2019, mucho más densa y de la cual también se tomó una porción, claramente más grande que la anterior. Como todo ensayo trae su estudio, se volvieron a examinar las operaciones para que fueran más exactas con la identificación al trabajar con una data más grande y así ir reduciendo la cantidad de parámetros no identificados pero que si tenían operación correspondiente, por lo que a todas las operaciones, exceptuando ‘Drilling’ y ‘On

surface' se les ampliaron los parámetros por los cuales iban a ser definidas en esta prueba. Las operaciones con los nuevos parámetros quedaron definidas de la siguiente forma:

- **Reaming:** $DBTM > 0$; $DBTM_{actual} > DBTM_{anterior}$;
 $DMEA > 0$; $DMEA_{actual} = DMEA_{anterior}$;
 $ROP_{actual} = ROP_{anterior}^{**}$
 $RPM > 0$; $FlowIn > 0$; $TQ > 0$; $Hookload > 0$
- **Backreaming:** $DBTM > 0$; $DBTM_{actual} < DBTM_{anterior}$;
 $DMEA > 0$; $DMEA_{actual} = DMEA_{anterior}$;
 $ROP_{actual} = ROP_{anterior}^{*}$
 $RPM > 0$; $FlowIn > 0$; $TQ > 0^{*}$; $Hookload > 0$
- **Sliding:** $DBTM > 0$; $DBTM_{actual} > DBTM_{anterior}$;
 $DMEA > 0$; $DMEA_{actual} > DMEA_{anterior}$;
 $DMEA = DBTM$;
 $RPM > 0^{*}$; $WOB > 0$; $ROP > 0$; $FlowIn > 0$; $TQ = 0^{*}$; $Hookload > 0$
- **Circulating:** $DBTM > 0$; $DBTM_{actual} > DBTM_{anterior}^{*}$;
 $DMEA > 0$;
 $ROP_{actual} = ROP_{anterior}^{*}$;
 $FlowIn > 0$; $Hookload > 0$
- **POOH:** $DBTM > 0$; $DBTM_{actual} < DBTM_{anterior}$;
 $DMEA > 0$; $DMEA_{actual} = DMEA_{anterior}$;
 $ROP_{actual} = ROP_{anterior}^{*}$;

* Los parámetros subrayados son los añadidos o modificados para la prueba B

$$\underline{RPM = 0^*} ; \underline{FlowIn = 0^*} ; Hookload > 0$$

- **POOHWPump:** $DBTM > 0$; $DBTM_{actual} < DBTM_{anterior}$;

$$DMEA > 0 ; DMEA_{actual} = DMEA_{anterior} ;$$

$$\underline{ROP_{actual} = ROP_{anterior}^*} ;$$

$$\underline{RPM = 0^*} ; FlowIn > 0 ; Hookload > 0$$

- **RIH:** $DBTM > 0$; $DBTM_{actual} > DBTM_{anterior}$;

$$DMEA > 0 ; DMEA_{actual} = DMEA_{anterior} ;$$

$$\underline{DBTM_{actual} \leq DMEA_{actual}^*} ;$$

$$\underline{ROP_{actual} = ROP_{anterior}^*} ;$$

$$\underline{RPM = 0^*} ; \underline{FlowIn = 0^*} ; Hookload > 0$$

- **RIHWPump:** $DBTM > 0$; $DBTM_{actual} > DBTM_{anterior}$;

$$DMEA > 0 ; DMEA_{actual} = DMEA_{anterior} ;$$

$$\underline{DBTM_{actual} \leq DMEA_{actual}^*} ;$$

$$\underline{ROP_{actual} = ROP_{anterior}^*} ;$$

$$\underline{RPM = 0^*} ; FlowIn > 0 ; Hookload > 0$$

* Los parámetros subrayados son los añadidos o modificados para la prueba B

Figura 22. Desarrollo en Excel, ensayo B

DMEA	DBTM	RPM	ROP	FLOWin	TQ	WOB	Hload	OPERACIÓN	Drilling	Reaming	Backreaming	Sliding	Circulating	POOH	POOHWPump	RH	RHWpump	Onsurface
4976.5	4976.5	3.037.314	259.71	9.006.541	157.350.861	5.71	158.64	Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
4977.12	4977.12	3.057.614	254.57	9.034.618	55.232.111		0.174.98	Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
4977.12	4977.12	3.484.236	254.57	10.728.588	-105.192.305		0.175.52	Circulating	NO	NO	NO	NO	SI	NO	NO	NO	NO	NO
4977.12	4977.12	4.953.917	254.57	16.619.828	2.619.361		0.174.07	Backreaming	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
4977.62	4977.62	2.968.845	254.57	8.708.508	16.119.781	4.81	163.54	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4979.63	4979.63	2.982.111	236.9	8.710.743	99.892.081	5.17	160.18	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4981.47	4981.47	2.985.846	198.49	8.701.163	98.816.783	11.71	156.64	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4982.94	4982.94	2.963.811	188.76	8.708.508	168.093.575	4.63	163.72	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4984.9	4984.9	2.977.635	262.29	8.693.785	155.535.579	5.54	162.81	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4986.78	4986.78	297.948	213.88	8.701.163	137.306.634	11.25	157.1	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4988.69	4988.69	2.978.429	262.29	8.701.163	126.459.148	7.99	160.36	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4990.63	4990.63	2.973.204	209.83	8.701.113	116.467.116	10.53	157.82	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4992.36	4992.36	2.994.489	218.18	8.701.155	136.303.964	4.54	163.81	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4994.5	4994.5	2.972.406	264	8.693.785	121.131.726	6.26	160.09	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4996.7	4996.7	2.968.272	266.67	8.693.785	136.956.026	6.71	159.64	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4998.17	4998.17	2.980.019	266.67	8.715.902	99.499.085		0.170.8	Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
4998.17	4998.17	2.980.019	266.67	8.715.902	19.859.028		0.174.43	Backreaming	NO	NO	NO	SI	NO	NO	NO	NO	NO	NO
4998.17	4998.17	2.175.291	266.67	8.701.163	-12.204.344		0.186.68	Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
4998.17	4998.17	2.178.976	266.67	8.715.902	-18.429.087		0.194.76	Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
4998.17	4998.17	2.177.125	266.67	8.708.499	-12.001.362		0.178.42	Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
4998.17	4998.17	2.177.125	266.67	8.708.499	-11.444.511		0.166.26	Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
4998.17	4998.17	2.183.158	266.67	8.715.886	-17.368.419		0.173.61	Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
4998.17	4998.17	2.427.018	266.67	8.715.902	4.099.534		0.169.44	Reaming	NO	SI	NO	NO	NO	NO	NO	NO	NO	NO
4998.36	4998.36	2.446.106	266.67	8.715.902	2.724.487		0.171.89	Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
4998.77	4998.77	2.451.832	266.67	8.699.996	7.397.012	1.72	166.63	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4999.37	4999.37	2.177.127	51.93	8.708.508	17.308.032		165.35	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
4999.97	4999.97	2.177.127	51.93	8.708.508	17.688.735	4.54	163.81	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
5000.58	5000.58	2.177.127	72	8.708.508	19.142.816	5.36	164.99	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
5001.19	5001.19	2.177.127	73.47	8.708.558	15.198.958	5.81	165.54	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
5001.79	5001.79	2.177.127	73.47	8.708.508	20.176.623	5.36	164.99	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
5002.36	5002.36	2.175.289	71.29	8.701.155	22.977.979	5.81	165.54	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
5002.93	5002.93	2.177.125	71.29	8.708.499	16.603.033	2.27	166.08	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
5003.55	5003.55	2.176.821	67.42	8.707.286	22.235.058	5.63	165.72	Drilling	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO

Fuente: Autores.

El archivo B se realizó con una porción de mil datos, la cual fue tomada desde una fila aleatoria de todos los datos del pozo B de Ecopetrol.

En la figura 22 se aprecia también la adición de una columna a la tabla, la función de esta columna ‘OPERACIÓN’ fue la de mostrar la operación identificada con su respectivo nombre además de identificarla con ‘SI’ y ‘NO’ (simultáneamente hecho en las demás columnas del lado). En ella se juntaron las definiciones de todas las operaciones en su lenguaje lógico y se agregó la operación ‘Undefined’ (ya definida en el capítulo 2) para cuando la combinación de parámetros que se lee no corresponde o aplica a ninguna operación. Por consiguiente, como se aumentó la data, el número de conjuntos de parámetros identificados con ‘Undefined’ aumentó, sin embargo, estos fueron menores a la cuarta parte de la data tratada y se sumó este punto a las observaciones de mejora pues los parámetros eran correctos para ser identificados por otra operación.

Posteriormente se trabajó la totalidad de la data del pozo B lo que provocó el mismo efecto anterior. Y a las operaciones ‘Reaming’, ‘Backreaming’, ‘POOH’, ‘POOHWPump’,

‘RIH’, ‘RIHWpump’ se les agregó el parámetro WOB = 0 con la intención de disminuir las apariciones de la actividad ‘Undefined’.

Siguiendo con la misma idea a las operaciones que ya se plasmaron se les sumó la operación creada ‘Sarta quieta’ con motivo de encerrar dos operaciones estipuladas en la macro *ILT Automatización* por los autores de esta, ‘Connection’ y ‘Working pipe’, pues estas generaban vacíos de identificación al estar cubiertas por los mismos parámetros y en las mismas condiciones además de, en el caso de ‘Working pipe’, no tener un concepto oficial o estándar.

Cabe resaltar que estas pruebas y las mostradas con anterioridad se corrieron con una frecuencia de data en tiempo de cada 30 segundos (medio minuto).

Figura 23. Desarrollo en Excel, adición de la operación 'Sarta quieta' en el ensayo B

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
DMEA	DBTM	RPMI	ROPI	MFH	TQI	WOB	HKLI		OPERACIÓN	Drilling	Reaming	Backreaming	Sliding	Circulating	POOH	POOHWPump	RIH	RIHWpump	On surface	Sarta quieta
2514_1056.57		0	0	0	-108.018.085	0	46.01		Sarta quieta	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI
2514_1056.57		0	0	0	-108.212.423	0	44.83		Sarta quieta	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI
2514_1056.57		0	0	0	-108.178.008	0	48.12		Sarta quieta	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI
2514_1056.57		0	0	0	-108.229.633	0	45.48		Sarta quieta	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI
2514_1056.57		0	0	0	-108.195.215	0	45.71		Sarta quieta	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI
2514_1056.57		0	0	0	-108.109.175	0	45.02		Sarta quieta	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI
2514_1056.57	0.2779	0	0	0	-108.109.175	0	45.55		Sarta quieta	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI
2514_1056.57	200.588	0	0	0	-87.834.915	0	43.02		Sarta quieta	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI
2514_1056.57	20.085	0	0	0	-60.657.332	0	43.6		Sarta quieta	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI
2514_1056.57	0	0	0	12.107	-101.987.407	0	43.45		Circulating	NO	NO	NO	NO	SI	NO	NO	NO	NO	NO	NO
2514_1055.84	0	0	0	167.996	-105.579.538	0	88.79		POOHWPump	NO	NO	NO	NO	NO	SI	NO	NO	NO	NO	NO
2514_1055.84	0	0	0	1.678.319	-101.831.626	0	88.56		Circulating	NO	NO	NO	NO	SI	NO	NO	NO	NO	NO	NO
2514_1055.84	200.065	0	0	3.518.459	-67.929.431	0	88.99		Circulating	NO	NO	NO	NO	SI	NO	NO	NO	NO	NO	NO
2514_1055.84	299.854	0	0	4.458.095	-1.861.669	0	88.53		Circulating	NO	NO	NO	NO	SI	NO	NO	NO	NO	NO	NO
2514_1056.13	300.063	0	0	5.031.673	-8.165.162	0	89.22		Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
2514_1058.59	300.147	0	0	5.034.133	29.603.171	0	88.37		Reaming	NO	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
2514_1061.79	299.269	0	0	5.034.133	-7.843.631	0	87.3		Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
2514_1065.35	299.854	0	0	5.036.594	-715.64	0	88.72		Reaming	NO	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
2514_1071	299.854	0	0	5.031.673	808.666	0	87.91		Backreaming	NO	NO	SI	NO	NO	NO	NO	NO	NO	NO	NO
2514_1077.98	300.293	0	0	5.036.594	11.120.735	0	85.65		Reaming	NO	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
2514_1086.37	30.044	0	0	5.034.133	-6.365.633	0	84.62		Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
2514_1092.37	300.147	0	0	5.036.594	-5.771.331	0	85		Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
2514_1098.08	299.707	0	0	5.036.594	39.241.399	0	88.49		Reaming	NO	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
2514_1104.38	300.293	0	0	5.036.594	-5.506.039	0	88.26		Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
2514_1110.52	299.415	0	0	5.036.602	7.593.603	0	84.81		Reaming	NO	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
2514_1117.57	300.147	0	0	5.034.133	772.971	0	85.96		Reaming	NO	SI	NO	NO	NO	NO	NO	NO	NO	NO	NO
2514_1122.94	30	0	0	5.036.602	-2.916.999	0	84.16		Undefined	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO

Fuente: Autores.

Una vez se corrió la data completa con la nueva operación incluida se redujeron los registros de ‘Undefined’ a un poco menos de la cuarta parte del total de la data, 7.323 de 29.865.

Sin embargo, nuevamente se observó un aspecto importante dentro los datos corridos, aun habiendo mejorado la identificación de las operaciones iba a seguir apareciendo ‘Undefined’ en

cierta cantidad de filas puesto que en esta data había porciones de valores negativos, lo que se presenta con los datos de cualquier pozo perforado.

En la nueva prueba, prueba C, se hicieron nuevamente dos cambios antes de ejecutarla para, como su nombre lo indica, probar o evidenciar el avance que se iba a obtener. El primero de ellos fue el conjunto de datos, un poco más denso, de un nuevo pozo perforado en el año 2019 por Ecopetrol, el cual se llamó pozo C. Y el segundo cambio fue, dentro del constante balance de las operaciones con los parámetros, la nueva definición de estas de la siguiente manera:

- **Drilling:** $DMEA > 0$;

$$DBTM > 0 ;$$

$$DMEA = DBTM ;$$

$$ROP > 0 ; RPM > 0 ; TQ > 0 ; WOB \geq 0$$

- **Reaming:** $DBTM > 0$; $DBTM_{actual} > DBTM_{anterior}$;

$$DMEA > 0 ;$$

$$RPM > 0$$

- **Backreaming:** $DBTM > 0$; $DBTM_{actual} < DBTM_{anterior}$;

$$DMEA > 0 ;$$

$$RPM > 0$$

- **Sliding:** $DBTM > 0$;

$$DMEA > 0 ;$$

$$DMEA = DBTM ;$$

$$RPM = 0 ; WOB \geq 0 ; ROP > 0 ; TQ = 0$$

- **Circulating:** $DBTM > 0$; $DBTM_{actual} = DBTM_{anterior}$;

$$DMEA > 0 ;$$

$$\text{ROP} = 0 ;$$

$$\text{FlowIn} > 0 ; \text{WOB} \leq 0$$

- **POOH:** $\text{DBTM} > 0 ; \text{DBTM actual} < \text{DBTM anterior} ;$

$$\text{DMEA} > 0 ;$$

$$\text{RPM} = 0 ; \text{FlowIn} = 0$$

- **POOHWPump:** $\text{DBTM} > 0 ; \text{DBTM actual} < \text{DBTM anterior} ;$

$$\text{DMEA} > 0 ;$$

$$\text{RPM} = 0 ; \text{FlowIn} > 0$$

- **RIH:** $\text{DBTM} > 0 ; \text{DBTM actual} > \text{DBTM anterior} ;$

$$\text{DMEA} > 0 ;$$

$$\text{RPM} = 0 ; \text{FlowIn} = 0$$

- **RIHWPump:** $\text{DBTM} > 0 ; \text{DBTM actual} > \text{DBTM anterior} ;$

$$\text{DMEA} > 0 ;$$

$$\text{RPM} = 0 ; \text{FlowIn} = 0$$

- **On Surface:** $\text{DMEA} \geq 0 ;$

$$\text{DBTM} \leq 0$$

- **Sarta quieta:** $\text{DMEA} > 0 ;$

$$\text{DBTM} > 0 ; \text{DBTM actual} = \text{DBTM anterior} ;$$

$$\text{ROP} = 0 ; \text{FlowIn} = 0 ; \text{WOB} = 0$$

La prueba se corrió con los cambios mencionados y simultáneamente se ejecutó en la base de datos del ICP para observar con mayor veracidad el avance y concordancia con lo producido en el archivo de Excel, por lo tanto, el porcentaje de errores encontrados hasta ese momento. El pozo con el que se trabajó tenía un total de 31.604 filas de conjuntos de parámetros

con una frecuencia en tiempo de cada 1 minuto, de los cuales se tuvieron 9.000 identificaciones equivocadas según la comparación con la prueba ejecutada en la base de datos que se adjuntó en el archivo C de Excel, como se puede apreciar en la figura 24, en las columnas AB y AD con nombre *Resultado de código implementado*.

Figura 24. Desarrollo en Excel, archivo C

F	G	H	I	J	K	L	M	N	O	AA	AB	AD	AE	AF	AG	AH
DMEA (ft)	Bit Depth_WITS 0 (ft)	RPM Time_WITS 0 (RPM)	ROP Time_WITS 0 (ft/h)	Flowin_T (GPM)	TOR_T (klb/ft)	WOB_T (klb)	Hookload_W ITS0 (klb)		OPERACIÓN		Resultado de código implementado	Resultado de código implementado		Resultado de código implementado	registro de cambio de respecto a la prueba 1	Resultado
											Prueba 1	Resultado		Prueba 2		
96.11	96.11	59.946	176.471	2.072.119	2.026.059	0	51.5		Drilling		error	Circulating		ok	CAMBIO PRB2	Drilling
96.31	96.31	59.356	176.471	2.071.086	9.661.911	0	50.87		Drilling		error	Circulating		ok	CAMBIO PRB2	Drilling
96.51	96.51	59.958	176.471	2.070.291	9.661.911	0.3	49.62		Drilling			Drilling		ok		Drilling
96.91	96.91	59.946	176.471	2.071.847	1250	0.15	49.77		Drilling			Drilling		ok		Drilling
97.11	97.11	59.355	315.789	2.073.517	2.203.583	2.09	47.83		Drilling			Drilling		ok		Drilling
97.31	97.31	600.086	315.789	207.217	2.052.117	0	52.43		Drilling		error	Circulating		ok	CAMBIO PRB2	Drilling
97.51	97.51	599.417	315.789	2.071.847	1.876.923	0	51.04		Drilling		error	Circulating		ok	CAMBIO PRB2	Drilling
97.51	97.51	599.941	315.789	2.072.224	9.661.911	0	51.13		Drilling		error	Circulating		ok	CAMBIO PRB2	Drilling
97.91	97.91	598.809	315.789	2.071.981	2.273.615	0	50.21		Drilling		error	Circulating		ok	CAMBIO PRB2	Drilling
98.11	98.11	59.721	22.5	2.067.558	1.507.692	0	49.95		Drilling		error	Circulating		ok	CAMBIO PRB2	Drilling
98.31	96.91	598.898	0	2.072.527	9.036.766	0	51.97		BackReaming			BackReaming		ok		BackReaming
98.31	96.91	599.249	0	2.072.408	9.649.153	0	51.43		Circulating			Circulating		ok		Circulating
98.31	96.91	60.021	0	2.070.178	2.716.612	0	51.12		Circulating			Circulating		ok		Circulating
98.31	89.71	0	0	1.649.238	0.9605	0	51.51		POOH W/Pump			POOH W/Pump		ok		POOH W/Pump
98.31	75.31	0	0	1.659.365	0.9552	0	51.65		POOH W/Pump			POOH W/Pump		ok		POOH W/Pump
98.31	65.11	0	0	124.451	0.9591	0	51.41		POOH W/Pump			POOH W/Pump		ok		POOH W/Pump
98.31	53.11	0	0	1.243.309	0.9677	0	50.07		POOH W/Pump			POOH W/Pump		ok		POOH W/Pump
98.31	62.91	598.375	0	1.283.341	9.815.008	0	52.26		Reaming			Reaming		ok		Reaming
98.31	76.71	598.544	0	2.072.538	1.492.308	0	51.18		Reaming			Reaming		ok		Reaming
98.31	910766	59.958	0	2.071.847	1.046.154	0	517.333		Reaming			Reaming		ok		Reaming
98.31	89.51	599.822	0	2.072.479	1.252	0	51.57		BackReaming			BackReaming		ok		BackReaming
98.31	79.71	0	0	1.659.754	0.9815	0	51.17		POOH W/Pump			POOH W/Pump		ok		POOH W/Pump
98.31	60.91	0	0	1.243.908	0.989	0	51.22		POOH W/Pump			POOH W/Pump		ok		POOH W/Pump
98.31	46.11	0	0	45.685	0.981	0	51.97		POOH W/Pump			POOH W/Pump		ok		POOH W/Pump
98.31	30.71	0	0	0	0.9826	0	51.01		POOH			POOH		ok		POOH
98.31	28.71	0	0	0	0.9773	0	52.38		POOH			POOH		ok		POOH
98.31	41.51	0	0	0	0.9807	0	52.16		RIH			RIH		ok		RIH
98.31	526599	0	0	82.904	0.9752	0	51.482		RIH W/Pump			RIH W/Pump		ok		RIH W/Pump
98.31	66.71	599.372	0	165.847	9.968.105	0	49.9		BackReaming			BackReaming		ok		BackReaming

Fuente: Autores.

De la prueba realizada se concluyó que hubo un factor importante por el cual se tuvo un número considerado de errores, este fue el formato numérico de los datos. La conclusión radicó en que ya sea en Excel o en la base de datos los datos no tenían el mismo formato de miles y/o decimales, si todos estaban definidos para unidades de miles con la coma (,) y si todos estaban definidos para unidades decimales con el punto (.) o, al contrario. Solucionado este aspecto se corrió una segunda prueba en la base de datos, también adjuntada en el archivo C y que se puede identificar en la figura 24 en las columnas AF-AG-AH, de donde se obtuvo una disminución considerable en las identificaciones erróneas pasando a una cifra de 4923. Llegados a este punto solo faltaba incorporar una operación de las también establecidas desde un comienzo,

‘Unregistred’, para el caso donde se presentaban valores nulos dentro de los datos ingresados. Gracias a la añadidura de esta actividad a la identificación de los parámetros los errores se pudieron reducir a más del 50% para un total de 836.

Avanzando en el desarrollo del tema y habiendo obtenido una porción mínima de errores respecto a la cantidad de data tratada se continuó analizando cómo se podía optimizar el reconocimiento de las operaciones, para lo que se encontró una opción práctica que fue agregar condicionales a ciertos parámetros, respecto a si los valores en esa fila son mayores o menores que 0, para que siguieran haciendo parte de los criterios (en cuanto a parámetros) de la operación correspondiente, esto debido a la variabilidad de los valores de la data de una fila de tiempo a otra. Los condicionales, parámetros removidos y las operaciones a las que se les hizo esto fueron:

- **Drilling:** se le agregó $\text{FlowIn} > 0$

- **Sliding:** se le agregó $\text{FlowIn} > 0$

se le cambió $\text{TQ} = 0$ por un condicional para torque ($\text{TQ} \leq 0$ o $\text{TQ} > 0$)

- **Circulating:** se le removió $\text{WOB} \leq 0$

se le agregó un condicional para ROP ($\text{ROP} \geq 0$ o $\text{ROP actual} = \text{ROP anterior}$)

- **Sarta quieta:** se le removió $\text{WOB} = 0$

se le cambió $\text{ROP} = 0$ por un condicional para ROP ($\text{ROP} = 0$ o $\text{ROP actual} = \text{ROP anterior}$)

De manera análoga esto trajo consigo un nuevo avance que significó la reducción de los 836 errores anteriores a tan solo 19 de estos.

Se hizo la última actualización a algunas operaciones buscando corregir totalmente los errores ya disminuidos, pero aún existentes. Las operaciones a las que se le hizo alguna modificación fueron las siguientes:

- **Circulating:** se cambió DBTM actual = DBTM anterior por un condicional para DBTM (DBTM actual = DBTM anterior o DBTM actual = DMEA actual)
- **On Surface:** se le retiró DMEA ≥ 0 , por lo tanto quedó solo con DBTM ≤ 0
- **Sarta quieta:** se le removió el condicional de ROP puesto en la corrida anterior (ROP = 0 o ROP actual = ROP anterior)

Figura 25. Desarrollo en Excel, archivo C con su última actualización

F	G	H	I	J	K	L	M	N	O	AA	AF	AG	AH	go implementado	Resultado
DMEA (ft)	Bit Depth_WITS 0 (ft)	RPM Time_WITS 0 (RPM)	ROP Time_WITS 0 (ft/h)	FlowIn_T (GPM)	TOR_T (klb/ft)	WOB_T (klb)	Hookload_W ITS0 (klb)		OPERACIÓN		Resultado de código implementado	registro de cambio con respecto a la prueba 1	Resultado	Resultado	Prueba 2
97.51	97.51	599.417	315.789	2.071.847	1.876.923	0	51.04		Drilling		ok	CAMBIO PRB2	Drilling		
97.51	97.51	599.941	315.789	2.072.224	9.661.911	0	51.13		Drilling		ok	CAMBIO PRB2	Drilling		
97.91	97.91	598.809	315.789	2.071.981	2.273.615	0	50.21		Drilling		ok	CAMBIO PRB2	Drilling		
98.11	98.11	59.721	22.5	2.067.558	1.507.692	0	49.95		Drilling		ok	CAMBIO PRB2	Drilling		
98.31	96.91	598.898	0	2.072.927	9.036.766	0	51.97		BackReaming		ok		BackReaming		
98.31	96.91	599.249	0	2.072.408	9.649.153	0	51.43		Circulating		ok		Circulating		
98.31	96.91	60.021	0	2.070.178	2.716.612	0	51.12		Circulating		ok		Circulating		
98.31	89.71	0	0	1.649.238	0.9605	0	51.51		POOH W/Pump		ok		POOH W/Pump		
98.31	75.31	0	0	1.659.365	0.9552	0	51.65		POOH W/Pump		ok		POOH W/Pump		
98.31	65.11	0	0	1.24.451	0.9591	0	51.41		POOH W/Pump		ok		POOH W/Pump		
98.31	53.11	0	0	1.243.309	0.9677	0	50.07		POOH W/Pump		ok		POOH W/Pump		
98.31	62.91	598.375	0	1.283.341	9.815.008	0	52.26		Reaming		ok		Reaming		
98.31	76.71	598.544	0	2.072.538	1.492.308	0	51.18		Reaming		ok		Reaming		
98.31	910766	59.958	0	2.071.847	1.046.154	0	517.333		Reaming		ok		Reaming		
98.31	89.51	599.822	0	2.072.479	1.252	0	51.57		BackReaming		ok		BackReaming		
98.31	79.71	0	0	1.659.754	0.9815	0	51.17		POOH W/Pump		ok		POOH W/Pump		
98.31	60.91	0	0	1.243.908	0.989	0	51.22		POOH W/Pump		ok		POOH W/Pump		
98.31	46.11	0	0	45.685	0.981	0	51.97		POOH W/Pump		ok		POOH W/Pump		
98.31	30.71	0	0	0	0.9826	0	51.01		POOH		ok		POOH		
98.31	28.71	0	0	0	0.9773	0	52.38		POOH		ok		POOH		
98.31	41.51	0	0	0	0.9807	0	52.16		RIH		ok		RIH		
98.31	526599	0	0	82.904	0.9752	0	51.482		RIH W/Pump		ok		RIH W/Pump		
98.31	66.71	599.372	0	165.847	9.968.105	0	49.9		BackReaming		ok		BackReaming		

Ordenar de A a Z

Ordenar de Z a A

Ordenar por color

Vista de Hoja

Borrar filtro de "Prueba 2"

Filtrar por color

Filtros de texto

Buscar

☒ (Seleccionar todo)

☒ ok

ACEPTAR

Cancelar

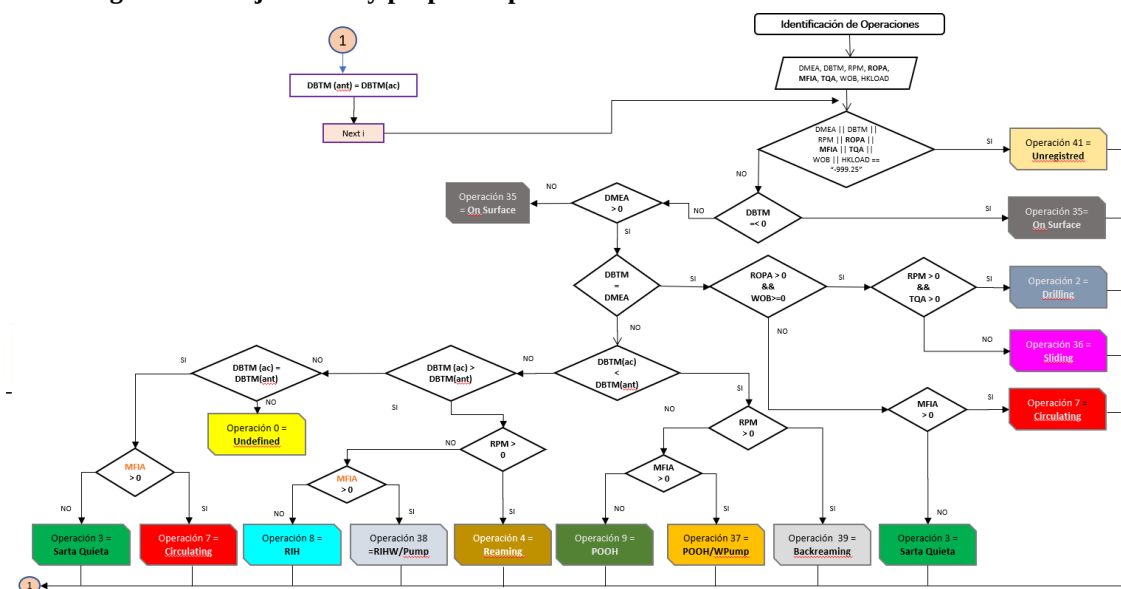
Fuente: Autores.

Como se muestra en la figura 25 y como se esperaba no existieron errores después de esta última corrección, de esta manera se cumplió con lo trazado y el archivo en Excel quedó alineado con la prueba corrida en la base de datos para verificar su correcto desempeño.

El propósito, al igual que con la macro mencionada, fue determinar los errores dentro del diagrama al pasar por cada uno de los ciclos condicionales para llegar a la operación asignada

Una de las equivocaciones que se encontraron se puede apreciar dentro del cuadro rojo en la figura 26 específicamente en el condicional que se ve primero, el cual describe que si el flujo (MFIA) es mayor a cero ($MFIA > 0$) la operación reconocida es POOH, por el contrario, si no es mayor a cero pasa al siguiente condicional ($RPMS > 0$), en este si no es mayor a cero la operación identificada es POOHWPump. Se afirmó que el proceso es erróneo porque si la operación determinada es con bomba, en este caso POOHWPump, quiere decir que el valor ingresado de flujo (MFIA) va a ser mayor a cero, por lo tanto, la operación identificada en la parte ‘SI’ de este primer condicional debió ser POOHWPump y no POOH.

Figura 27. Diagrama de flujo creado y propuesto para la base de datos del ICP



Fuente: Autores.

Finalmente se transformaron las ultimas definiciones de las operaciones, las ideas plasmadas para tener un nuevo algoritmo y todo lo creado en Excel al diagrama de flujo mostrado en la figura 27. Así mismo a diferencia del primer diagrama de flujo en este se incorporó el concepto de la operación ‘Sarta quieta’ que también ayudó a tener un diagrama más corto y simple al cumplir y/o reemplazar las funciones de ‘Connection’ y ‘Working pipe’.

De esta forma el prototipo de herramienta quedó ajustado a las condiciones necesarias para ser “instalado” en la base de datos y que esta funcione a partir de este algoritmo finalizado.

5. Aplicación del Prototipo de Herramienta al Campo de Estudio y Resultados

5.1 Reseña del Campo

El campo fue descubierto en 1969 por Chevron con un pozo que alcanzó una profundidad de 7347 ft (pies) y que produjo crudo pesado de las formaciones Mirador (10° API) Guadalupe y Une (13 7° API). Produce de las areniscas de Une y Gacheta. También fue probada la Formación Mirador como productora de crudo pesado. Esta formación no ha sido explotada comercialmente y se tiene como reserva para ser completada una vez se agoten las reservas de las formaciones Une y Gacheta. El área productiva máxima calculada por Ecopetrol es de 11000 acres (Agencia Nacional de Hidrocarburos – ANH, 2018).

Después de haberse descubierto el campo inició producción casi 7 años después. En diciembre de 2005 se reportó una producción diaria de 52837 barriles con un corte de agua de 63%. Los últimos pozos muestran tasas de producción entre 900 y 1000 barriles. Esta última tasa se logró con la perforación de nuevos pozos y a una mayor disponibilidad de las facilidades de

producción y tratamiento. El mecanismo de producción para las formaciones Une y Gacheta es un acuífero activo (ANH, 2018).

5.2 Aplicación

Para las pruebas del correcto funcionamiento del prototipo de herramienta se trabajaron data's reales siendo estas de entrada unas primeras aplicaciones con varios pozos utilizados para el objetivo. Sin embargo, una vez se finalizó el diseño del prototipo se procedió a correr los parámetros de dos pozos del campo de estudio mencionado, esto con el fin de verificar que al cambiar la información puesta en el prototipo de herramienta de manera aleatoria continuara operando acertadamente. Lo hecho con cada uno de los pozos se muestra a continuación:

5.2.1 Pozo 1

El primero de los dos pozos trabajados fue un pozo perforado dentro de una de las campañas de 2009 de Ecopetrol. El pozo tuvo una profundidad medida de 7005.2 ft.

A partir de los resultados arrojados por el prototipo de herramienta, que en esta ejecución marchó identificando parámetros con una frecuencia de cada 5 minutos, se pudo primeramente determinar la duración de toda la construcción del pozo.

Tabla 1
Duración del pozo 1

Duración del Pozo	
Minutos	212275
Horas	3537,92
Días	147,4

Fuete: Autores

Así mismo y como lo indica el nombre de este proyecto se obtuvo el tiempo total de intervención de cada una de las operaciones que tuvieron lugar en la perforación del pozo, el cual se evidencia en la siguiente tabla:

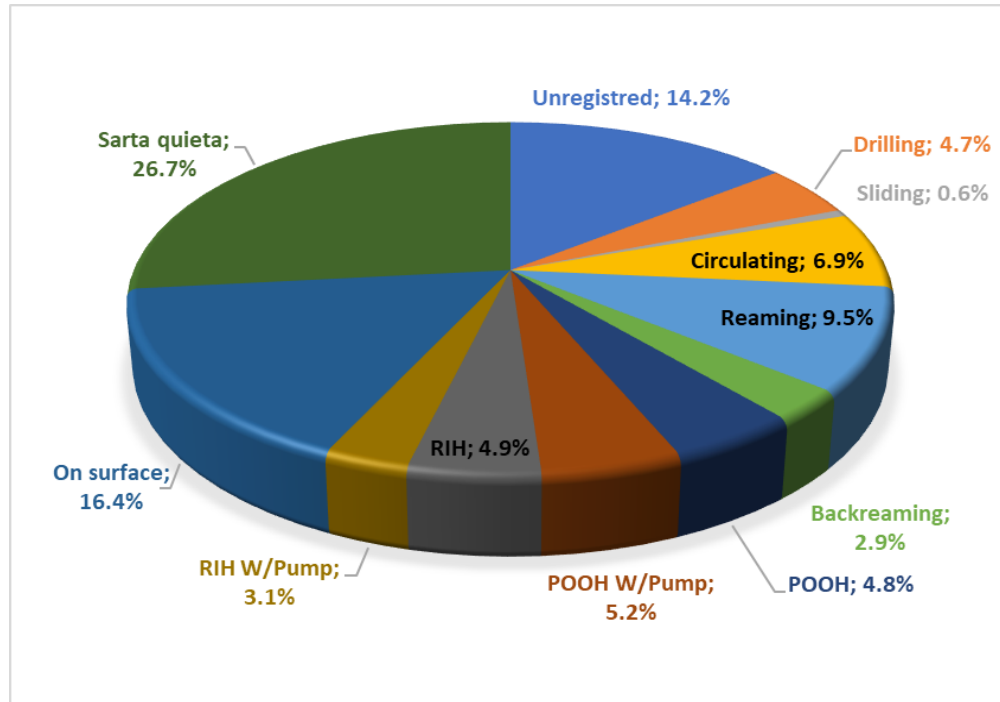
Tabla 2
Tiempo de las operaciones del pozo 1

OPERACIÓN	HORAS	DÍAS
DRILLING	166.67	6.9
SLIDING	21.92	0.91
CIRCULATING	245.17	10.2
REAMING	336.5	14
BACKREAMING	100.8	4.2
POOH	170.67	7.1
POOH W/PUMP	183.8	7.66
RIH	172.3	7.2
RIH W/PUMP	111.3	4.64
SARTA QUIETA	944.8	39.37
ON SURFACE	579.92	24.16
UNREGISTERED	503.92	21
UNDEFINED	0	0

Fuente: Autores

De estos tiempos conseguidos se realizó un análisis porcentual para observar de manera más sencilla cómo fue la distribución de las operaciones y en qué medida en todo el desarrollo del pozo finalizado.

Figura 28. Distribución porcentual de las operaciones del pozo 1



Fuente: Autores.

La idea y la facilidad de tener estos tiempos es que permiten hacer un análisis y observar el desempeño del pozo y así mismo y en su defecto un posterior estudio sobre el por que de determinada cantidad o porcentaje de ciertas operaciones y cómo estas afectaron positiva o negativamente la duración de la perforación del pozo en discusión. Siguiendo con este plan se hicieron las siguientes afirmaciones respecto a las cantidades de tiempo encontradas:

- Las operaciones con mayor porcentaje de tiempo son ‘Sarta quieta’ y ‘On surface’ con 26,7% y 16.4% respectivamente, lo que significa que durante la perforación del pozo hubo eventos externos tales como, tormenta eléctrica o fallas en alguno de los equipos que detuvieron la actividad en desarrollo e hicieron parar la tubería y/o volver a superficie para esperar a solucionar el imprevisto que se pudo haber presentado. Así mismo estos eventos que hacen aumentar el tiempo de estas operaciones específicas y, en efecto, de todo el pozo se pueden categorizar como

NPT's por lo que habría que entrar en el detalle de los reportes diarios del pozo para poder clasificar los tipos de NPT's que se pudieron haber presentado.

- La tercera operación con más tiempo fue 'Unregistred', esto quiere decir que una gran cantidad de los parámetros recibidos y puestos en el prototipo de herramienta vienen con valores nulos (-999.25, -99.925) afectando la identificación de la operación que se hizo en su momento y del mismo modo reduciendo el rendimiento del pozo.
- Dentro de las operaciones normales o cotidianas de la perforación de un pozo las que tuvieron una mayor participación en este caso son 'Reaming' y 'Circulating' de lo que se pudo deducir que el pozo tuvo inconvenientes con reducciones en el agujero en ciertas profundidades lo que hicieron repasar el hueco varias veces, así como obstrucciones al momento de avanzar en la profundidad del pozo. Seguido de estas acciones en el pozo y como bien se sabe, se debe circular para limpiar el agujero y llevar todos estos ripios a superficie, sin embargo, puede que las circulaciones hayan sido cortas y al intentar salir del pozo se encontraran con camas de cortes que no llegaron a la superficie y que eran arrastrados por los componentes del BHA con mayor diámetro externo aumentando así la frecuencia de la circulación.
- Posterior a estas operaciones siguen 'POOH W/Pump', 'RIH', y 'POOH', lo que refuerza la idea expuesta en el punto anterior y entendiendo que cuando se repasaba y circulaba el pozo seguido de esto se sacaba la tubería con circulación y después se volvía a introducir en esa sección donde se pudo presentar una

obstrucción. También se deduce que este proceso se hizo de manera sucesiva en determinadas profundidades hasta lograr avanzar de ese punto.

De esta manera se aplicó el prototipo de herramienta al Pozo 1 y se analizó con los planteamientos anteriormente mostrados.

5.2.2 Pozo 2

El segundo de los dos pozos estudiados fue un pozo perforado en la campaña de 2010 de Ecopetrol. El pozo tuvo una profundidad medida de 7020.2 ft.

A partir de los resultados arrojados por el prototipo de herramienta, que en esta ejecución marchó identificando parámetros con una frecuencia de cada 5 minutos, se pudo primeramente determinar la duración de toda la construcción del pozo.

Tabla 3
Duración del pozo 2

Duración del Pozo	
Minutos	187070
Horas	3117,8
Días	129,9

Fuente: Autores

Continuando con el proceso implementado en el Pozo 1, se logró obtener el tiempo total de intervención de todas las operaciones que tuvieron lugar en la perforación del pozo, los cuales se evidencian en la siguiente tabla:

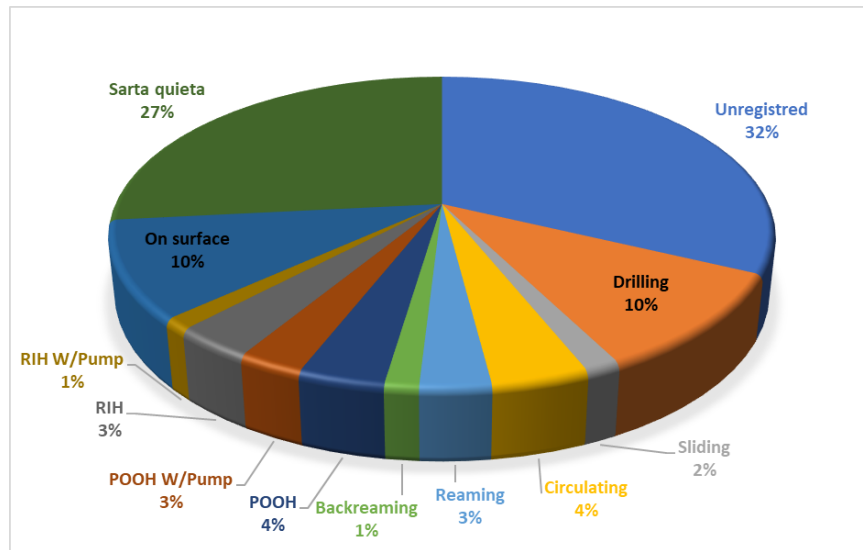
Tabla 4
Tiempo de las operaciones del pozo 2

OPERACIÓN	HORAS	DÍAS
DRILLING	322.6	13.4

SLIDING	45.3	1.88
CIRCULATING	126.7	5.28
REAMING	92	3.83
BACKREAMING	44.2	1.84
POOH	112.7	4.69
POOH W/PUMP	85.83	3.57
RIH	104.25	4.34
RIH W/PUMP	37.08	1.54
SARTA QUIETA	830.5	34.6
ON SURFACE	315.4	13.1
UNREGISTERED	1001	41.7
UNDEFINED	0	0

Fuente: Autores

De estos tiempos obtenidos se decidió realizar nuevamente un análisis porcentual para poder determinar de forma más sencilla y concisa la distribución en tiempos de cada operación que influyo en el desarrollo total del pozo.

Figura 29. Distribución porcentual de las operaciones del pozo 2

Fuente: Autores.

Teniendo en cuenta la distribución de los datos obtenidos se pueden afirmar ciertos criterios en el análisis porcentual del tiempo de perforación, como lo son:

- Las operaciones que denotaron un mayor consumo de porcentaje en tiempo son ‘Unregistred’ y ‘Sarta quieta’ con 32.1% y 26.6% respectivamente, lo cual significa que durante el tiempo que tomo la perforación del pozo, los sensores que miden los parámetros necesarios para la corrida del prototipo obtuvieron medidas erróneas o nulas que tenían valores de (-999.25, -99.925), valor por el cual el prototipo de herramienta que se encarga de la identificación de estas operaciones no lograba agruparlas en ningún valor programado anteriormente, de igual forma también implica factores externos tales como, tormenta eléctrica o fallas en alguno de los equipos que detuvieron la actividad en desarrollo e hicieron parar la sarta de tubería para esperar a solucionar el imprevisto que se pudo haber presentado.

- La tercera operación con más tiempo fue ‘Drilling’, esto pudo indicar que fue un pozo con pocos problemas en cuanto a inestabilidad y con buen avanza en cuanto a profundidad perforada, ya que esto indica que la relación de avance en profundidad fue muy buena comparada en el pozo tomado en primera instancia.
- Siguiendo con la secuencia de operaciones otra que se logró destacar fue la de ‘On Surface’, esto se puede deducir que fue debido a el tema de el porque se encontraba la sarta quieta y tuvieron que subir a superficie para solucionar algún inconveniente en la (broca, mecha o barrera), o para corregir algún inconveniente de la sarta de tuberías como tal, o con la cuadrilla de trabajo.

Teniendo en cuenta los puntos expuestos anteriormente y al poder implementar el prototipo de herramienta en más de un pozo de las campañas realizadas por Ecopetrol, se puede dar por terminado el diseño del prototipo de herramienta y corroborar su buen funcionamiento en cuanto a lo que fue programado.

6. Conclusiones

El tiempo de los dos pozos analizados de la campaña de perforación de 2009 y 2010 fue de 130 días y 148 días respectivamente, esto a partir de la cantidad de datos recibidos y el intervalo de tiempo que había entre cada uno de ellos. Se relaciona una baja eficiencia a la cantidad de días con los que se llevaron a cabo las perforaciones de estos pozos, ya que desde el punto de vista de tiempo y teniendo como referencia que para la campaña de 2019 del campo en estudio los pozos se perforan en un promedio de 30 días hay alrededor de un 78% menos de eficiencia en la campaña 2009, es decir, se pudo haber perforado 3 pozos adicionales. Lo que fue debido tanto a problemas operacionales en la segunda fase o sección de los pozos donde se

evidencia la mayor cantidad de identificaciones de las operaciones “Reaming”, “Backreaming”, “POOH” y “POOH/Pump” además de estar entre estas, después de ciertos intervalos, las operaciones “On Surface” y “Sarta quieta”; como a eventos sociales o climáticos que causan la suspensión de la operación.

La creación de la operación ‘Sarta quieta’ en el prototipo de herramienta es de gran utilidad para una identificación más simple, efectiva y específica de acuerdo con los parámetros que se reciben y que estaban asignados a las operaciones anteriores que se venían trabajando a este proyecto, ‘Connection’ y ‘Working pipe’, los cuales para la detección de una u otra no tenían un valor diferente definido generando error en la identificación del prototipo de herramienta. Además de ser, basado en los trabajos o actividades realizados en campo, una operación que agrupa a las dos anteriores y que resulta ser más práctica.

No debe confundirse para la aplicación en el prototipo de herramienta el tipo de información que se va a ejecutar respecto a tiempos. Es decir, la data ejecutada es enviada de manera simultanea a la perforación del pozo por lo que corresponde a una data de tiempos reales en la que están inmersos los tiempos no productivos, tiempos invisibles y tiempos productivos que tuvo la construcción del pozo, sin ningún estudio o clasificación previa.

Es fundamental contar con información confiable para la correcta ejecución e identificación del prototipo de herramienta ya que una gran cantidad de esta es recibida con errores de decimales, unidades de mil o valores negativos provocando la no identificación de una posible operación que se está llevando a cabo y por el contrario se determina en el prototipo de herramienta como ‘Unregistred’ o ‘Undefined’. Así mismo, es la base del análisis hecho a los tiempos gastados por cada una de las operaciones durante la construcción del pozo para obtener

planes que contribuyan a las mejores prácticas y así conseguir una reducción de tiempo en la campaña de perforación con una mayor eficiencia.

7. Recomendaciones

Determinar y seleccionar operaciones puntuales hechas en el taladro que se puedan definir a partir de los parámetros recibidos con el fin de lograr una mayor discretización y un mayor análisis en la identificación de las actividades realizadas en la perforación, ejemplo de esto es la cementación donde el parámetro fundamental es el flujo que va dirigido hacia el pozo.

Realizar inspecciones de diagnóstico, prevención y mantenimiento de los sensores y demás herramientas que obtienen los parámetros con los que se trabajan ya que esta es una de las razones con mayor influencia por las que hay porciones grandes donde no es posible identificar operaciones y en las que por la secuencia y lógica de las actividades que se van identificando está la posibilidad de continuar el proceso de identificación. Además de hacer un control de calidad a los parámetros registrados pues pueden contener errores en ciertas cantidades de datos.

Asociar a la información ejecutada en el prototipo de herramienta los proyectos desarrollados hacia las metodologías de identificación y manejo de NPT's e ILT's para alcanzar un mejor análisis y clasificación de los tiempos estudiados siendo parte de la mejora continua para próximas campañas de perforación.

Para futuros proyectos la utilización de una herramienta informática como Python o de programación facilita el procesamiento de big data a diferencia de Excel. En el mismo sentido realizar pruebas de desarrollo para el prototipo de herramienta en Machine learning que optimiza el funcionamiento del prototipo de herramienta.

Referencias Bibliográficas

Al Shamsi, J.; Al Nauimi, M.; Al Hosani, F.; Al Sairi, A. (2018). “Invisible Lost Time Initiative
ILT in Drilling Toward More Efficiency, High Performance and High Profitability Do
More with Less” SPE-192969-MS.

Andersen, K.; Per Arild, S.; Maidla, E.; King, B.; Thonhauser, G.; Zollner, P. (2009). “Case
History: Automated Drilling Performance Measurement of Crews and Drilling
Equipment” SPE/IADC 119746.

Ansari, A.; Watfa, M. (2018). “A Novel Strategy to Identify Potential Savings in Digitized
Oilfields Through Automated Drilling Data Analysis” SPE-192821-MS.

Bohorquez, B, L. (2021). TIEMPO REAL (COSMOS BD) IMPLEMENTACIÓN DE VERSIÓN
2 DE ALGORITMO DE OPERACIONES EN ENTORNO AZURE TIEMPO
HISTÓRICO (SQL SERVER).

Bond, D.F.; Scott, P.W.; Page, P.E & Windham, T.M. (1998). “Applying Technical Limit
Methodology for Step Change in Understanding and Performance”. Woodside Offshore
Petroleum. SPE 51181.

Cantarelli, E.; Kojadinovic, N; Le Dang Pham, K.; Salem Youssef, G. (2017). “Invisible Lost Time Reduction and Drilling Risk Management Optimization in United Arab Emirates Onshore Field” SPE-188640-MS.

DE TOGNI, Riccardo. (S.F.) Predictive Model for Drilling Phase Duration of Oil & Gas Wells.

De Wardt, J. & Rushmore, P. (2016). “True lies: Measuring Drilling and Completion Efficiency” SPE-178850-MS.

El Afifi, S.; Albassam, B.; Fahmy, F.A. (2015). “Enhance the Drilling & Tripping Performance on Automated Rigs with fully automated performance measurement”. SPE-176786-MS.

Ford Brett, J. (2066). “The Perfect Well Ratio: Defining and using the theoretically minimum duration to improve drilling performance” OGS International. AADE-06-DF-HO-13.

G. Yarim, G. Martin, R. B. May. (2008). A Guide to Successful Bakcreaming Real Time Case Histories.

IADC Drilling Manual. (2000). eBook versión 11.

Kumar, D.; Jarrett, C.; Smith, G.; Smith, M.; Cheatham, C.; Kolstad, E.; Brooks, S. (2017). “Real Time Drilling Optimization and Rig Activity Based Models Deliver Best In Class Drilling Performance: Case History” SPE/IADC-184744-MS.

Lakhanpal, V.; Robello, S. (2017). “A New Approach to Harness Data for Measuring Invisible Lost Time in Drilling Operations” SPE-187270-MS.

Osorio, A. (2018). Evaluación de tiempos históricos de perforación a partir de una metodología teórica para la optimización de áreas operacionales. Caso de estudio, campo colombiano. Tesis de grado Universidad Industrial de Santander, Escuela Ingeniería de Petróleos.

Ouahrani, L.; Haris Abdul Nahil, A.; Suluru, S.; Chiha, A.; Al Fakih, A. (2018). “Invisible Lost Time measurement and reduction contributes to optimizing total well time by improving ROP and reduction flat time” SPE-192319-MS.

Raza, S.A.; Al-Braik H. and Attalah M. (2017). “Performance Enhancement of Drilling and Completions Operations in Giant Offshore Field Abu Dhabi by Tracking and Monitoring Invisible Lost Time and Defined KPIs”. SPE 1883228.

SCHLUMBERGER LIMITED. The Oilfield Glossary. [en línea]. (Recuperado en 17 de octubre de 2020). Disponible en: <https://www.glossary.oilfield.slb.com/>