



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Aprendizaje Automático y Ciencia de Datos en la Ingeniería Petrolera: Enfoque al Stick & Slip

TESIS

Que para obtener el título de
Ingeniero Petrolero

P R E S E N T A

Luis Enrique Navarro Morales

DIRECTOR DE TESIS

Ing. Daniel Marure Valdez



Ciudad Universitaria, Cd. Mx., 2022

*[...] que la posesión y el beneficio matan,
mientras que la búsqueda,
desligada de cualquier utilitarismo,
puede hacer a la humanidad más libre,
más tolerante y más humana.*

Nuccio Ordine

Agradecimientos

A mis padres, Rocio y Antonio, por enseñarme el valor de la perseverancia, el cariño y la dedicación.

A mi hermana, Adriana, por estar presente a lo largo de mi formación y ser mi compañera de vida.

A mi familia, por todo lo que hemos compartido.

A Ana, por el tiempo, las experiencias y el conocimiento que compartimos y adquirimos juntos.

A los maestros y profesionistas que contribuyen a la formación profesional de nuestro país.

Al Ing. Daniel Marure y a la M.I. Berenice Anell por su apoyo en la revisión, elaboración y seguimiento de esta tesis.

Al Dr. Leonardo Teja por sus sabios consejos.

A la Universidad Nacional Autónoma de México y la Facultad de ingeniería por ser los bastiones de formación profesional y humana.

Y para aquellos que formaron parte de mi crecimiento y que partimos por caminos diferentes.

Acknowledgements

To my dear friends Misael, Nina, Johao, Natasha, Rodrigo, Diego and Kevin for all our experiences and knowledge we've gained together.

To my mentors, Eng. Alejandro Primera whose confidence and support has been invaluable, Eng. Stephen Fell for the hours and knowledge imprinted to this project, Eng. Erik Norman for his trust, Msc. Hesham Moubarak, Eng. Hazem Said, Eng. Darlington Etaje, Eng. Krishna Nischal and Dr. Rashid S. for their assistance.

Índice de Figuras

Figura 1.1.1 : Ubicación de Volve.....	3
Figura 1.1.2 : Producción de Volve	3
Figura 1.1.3 : Formaciones de la Estructura Theta Vest (Discovery Report, 1993)	4
Figura 1.1.4 : Modelo dinámico del yacimiento Volve	6
Figura 1.2.1 : Crecimiento de lenguajes de programación (TrustDimension, 2021)	7
Figura 1.2.2 : "IDE" de "Jupyter Notebook"	8
Figura 2.1.1 : Sistemas en la perforación (Salid232, 2019).....	9
Figura 2.1.2 : Aparejo de fondo (Mayer, 2007).....	10
Figura 2.1.3 Principios de desviación (Azar, 2007)	11
Figura 2.1.4 : BHA de 8 ½" del pozo F-14 (Drilling program F-14, 2008)	13
Figura 2.2.1 : Toma de survey (Frontier MWD Systems, 2014)	14
Figura 2.2.2 : "Survey" de perforación pozo F5	14
Figura 2.2.3 : Registro de Temperatura-Profundidad pozo F10	15
Figura 2.2.4 : Registro de corrida de perforación	17
Figura 2.3.1 : Pulso positivo (Morley, 1998)	19
Figura 2.3.2 : Pulso negativo (Morley, 1998)	19
Figura 2.3.3 : Onda continua (Morley, 1998).....	19
Figura 2.3.4 : Transmisión de datos (Azar, 2007).....	20
Figura 2.4.1 : Registros WITSML pozo F10.....	21
Figura 2.4.2 : Registros LAS de pozo F10	22
Figura 3.1 : Subramas de la Inteligencia Artificial.....	23
Figura 3.2 : Aprendizaje Automático (Mitchell T. M., 1997).....	24
Figura 3.1.1 : Aprendizaje supervisado.....	25
Figura 3.1.2 : Diagrama de flujo para ML (Belyadi, 2021. Modificado)	26
Figura 3.2.1 : Diagrama Venn para ciencia de datos (Conway, 2010)	27
Figura 3.2.2 : Pairplot de registros de perforación	28
Figura 3.2.3 : Relación de gráficos de violín y caja.....	28
Figura 3.2.4 : Relación Peso-Profundidad y secciones.....	29
Figura 3.2.5 : Gráfico de violín de TQA para pozo F5.....	29
Figura 3.2.6 : PDA (Mohaghegh, 2020. Modificado).....	30
Figura 3.3.1 : Ejemplo Árbol de decisión (Breiman,1984. Modificado)	32
Figura 3.3.1.1 : Elementos del Árbol de decisión (Müller, 2017).....	33
Figura 3.4.1 : Boosting (QuantDare, 2016)	35
Figura 3.4.2 : Bagging (QuantDare, 2016).....	35
Figura 3.4.1.1 : Elementos del Random Forest (Jagannath, 2017)	36
Figura 4.1.1 : Vibraciones de la sarta (Guangjian & Ping, 2016).....	37
Figura 4.1.2 : Vibraciones del BHA (SLB DrillingQuickReference, 2013).....	38
Figura 4.1.3 : S&S (Mayer, 2017)	39
Figura 4.1.4 : Medición S&S (Green, 2005).....	39
Figura 4.1.5 : Evento de S&S (Mayer, 2017)	41
Figura 4.2.1 : Daños por severidades altas	42
Figura 4.2.2 : Remoción de vibraciones	43

Figura 4.2.3 : Diagrama de flujo para remediación (SLB Drilling Quick Reference, 2013. Modificado)	43
Figura 4.3.1 : Documentos entregables en la perforación pozo F4	45
Figura 4.3.2 : Pozos de análisis.....	46
Figura 4.3.1.1 : Barrena 12 ¼" pozo F14	47
Figura 4.3.1.2 : Barrena 8 ½" pozo F14	48
Figura 4.3.1.3 : Parámetros direccionales Pozo F14	48
Figura 4.3.2.1 : Corrida 1, Barrena 8 ½" pozo F10	49
Figura 4.3.2.2 : Corrida 2, Barrena 8 ½" pozo F10	50
Figura 4.3.2.3 : Parámetros direccionales Pozo F10	50
Figura 4.3.3.1 : Barrena 12 ¼" pozo F5	51
Figura 4.3.3.2 : Corrida 1, Barrena 8 ½" pozo F5	52
Figura 4.3.3.3 : Corrida 2, Barrena 8 ½" pozo F5	52
Figura 4.3.3.4 : Parámetros direccionales Pozo F5	53
Figura 4.3.4.1 : Barrena 12 ¼" pozo F4	54
Figura 4.3.4.2 : Corrida 1, Barrena 8 ½" pozo F4	54
Figura 4.3.4.3 : MWD RECORD pozo F4	55
Figura 4.3.4.4 : Corrida 2, Barrena 8 ½" pozo F4	55
Figura 4.3.4.5 : Parámetros direccionales Pozo F4	56
Figura 4.3.4.6 : S&S en secciones de perforación	56
Figura 4.3.4.7 : Severidad S&S categórica por pozo	57
Figura 4.3.4.8 : Severidad S&S categórica para Azimut	58
Figura 4.3.4.9 : Severidad S&S categórica para Inclinación	58
Figura 4.3.4.10 : Severidad S&S categórica para DLS	59
Figura 4.3.4.11 : Pairplot de S&S categórico.....	61
Figura 5.1.1 : Archivos LAS de pozo F14.....	62
Figura 5.2.1 : BONB de etapa de barrena de 12 ¼" F-4	63
Figura 5.2.2 : Gráfico de caja por pozo	64
Figura 5.2.3 : Histograma de severidades categóricas.....	65
Figura 5.3.1 : Coeficientes de correlación de Pearson para árbol de decisión.....	66
Figura 5.3.2 : Sets de datos para árbol de decisión	66
Figura 5.3.1.1 : Árbol de decisión resultante	67
Figura 5.3.2.1 : Relevancia parámetros para árbol de decisión.....	68
Figura 5.3.2.2 : Matriz de confusión para árbol de decisión.....	69
Figura 5.4.1 : Coeficientes de Pearson para RF.....	70
Figura 5.4.2 : Sets de datos para RF.....	70
Figura 5.4.1.1 : RF gráfico.....	72
Figura 5.4.2.1 : Relevancia parámetros para modelo Random Forest	73
Figura 5.4.2.2 : Matriz de confusión para RF	73
Figura 5.5.1 : Matriz de confusión para "rf"	75
Figura 5.5.2 : Pozo multilateral F15	76
Figura 5.5.3 : Registros MWD disponibles para F15A.....	76
Figura 5.5.4 : Plan Pozo F15A.....	77
Figura 5.5.5 : Comparación de modelos	78

Índice de tablas

Tabla 1 : Acumulados de producción e inyección.....	4
Tabla 2 : Litología de la región	5
Tabla 3 : Registros de perforación	16
Tabla 4 : Machine Learning	24
Tabla 5 : Vertientes de razonamiento.....	31
Tabla 6 : Clasificación por severidad.....	40
Tabla 7 : Prevención de S&S (Green, 2005)	44
Tabla 8 : Análisis correspondiente al Pairplot.....	60
Tabla 9 : Resumen de modelos	75
Tabla 10 : Aplicación de modelos Random Forest.....	77

Contenido

Resumen	1
Abstract.....	2
Capítulo 1. Aspectos Fundamentales.....	3
1.1 Campo Volve	3
1.2 Python.....	7
Capítulo 2. Herramientas de Perforación	9
2.1 Sarta de Perforación	9
2.2 Adquisición de datos.....	14
2.3 Transmisión de datos.....	19
2.4 Archivos WITSML y LAS	21
Capítulo 3. <i>Machine Learning</i>	23
3.1 Aprendizaje supervisado.....	25
3.2 Ciencia de Datos.....	27
3.3 Árbol de decisión	32
3.3.1 Entrenamiento de árbol.....	33
3.4 Métodos de Ensamble	35
3.4.1 <i>Random Forest</i>	36
Capítulo 4. Stick & Slip	37
4.1 Mecanismos de vibración	37
4.2 Consecuencias y Remediación	42
4.3 Relevancia operativa.....	45
4.3.1 Análisis de Pozo F14.....	47
4.3.2 Análisis de Pozo F10.....	49
4.3.3 Análisis de Pozo F5.....	51
4.3.4 Análisis de Pozo F4.....	53
Capítulo 5. Predicción de Vibraciones Torsionales	62
5.1 Recopilación de datos	62
5.2 Limpieza y construcción de base de datos.....	63
5.3 Construcción de árbol de decisión.....	66
5.3.1 Parámetros del modelo	67
5.3.2 Interpretación de resultados	68
5.4 Construcción de Random Forest.....	70
5.4.1 Parámetros del modelo	71
5.4.2 Interpretación de resultados	73
5.5 Aplicación de modelo de predicción.....	75
Capítulo 6. Conclusiones	79
6.1 <i>Stick & Slip</i>	79
6.2 Uso de datos	82
6.3 Aplicación del algoritmo RF	83
Oportunidades de Implementación	84
Bibliografía	85
Anexo 1. Términos y Condiciones	87
Anexo 2. Códigos Elementales	88

Resumen

El presente trabajo busca aplicar herramientas de análisis de datos y predicción de la rama del aprendizaje automático o Machine Learning, específicamente los algoritmos de *Árbol de Decisión* y *Random Forest*, a un fenómeno de vibración torsional en la sarta de perforación, para que sea una herramienta adicional en la mitigación y posible predicción de eventos de esta índole.

Por lo que se requiere una introducción básica a estas disciplinas, en función de su relevancia y potencial apoyo a la actividad del ingeniero petrolero; en el Capítulo 1. Aspectos Fundamentales, se busca dar una breve introducción a los datos de tiempo real utilizados en el presente para desarrollar los modelos de predicción, siendo estos de libre acceso, y de un campo real noruego llamado *Volve* así como la descripción del lenguaje de programación denominado Python el cual ha tenido un auge entre la comunidad científica y digital por su versatilidad y herramientas de código libre.

El Capítulo 2. Herramientas de Perforación busca explicar los componentes esenciales de la sarta de perforación, y, por ende, los dispositivos enfocados en la medición de diversos parámetros recabados en el fondo y en la superficie, así como su presentación gráfica y transmisión, mientras que el Capítulo 3. *Machine Learning* se enfoca en el desarrollo de estos algoritmos, sus principales vertientes de aplicación y clasificación, también se detallan los elementos de análisis más relevantes de la ciencia de datos, ya que serán de utilidad para la construcción de modelos de predicción.

El Capítulo 4. *Stick & Slip (S&S)* muestra los fenómenos vibratorios presentes en la sarta en el proceso de perforación, enfocándose principalmente en la vibración de **S&S**, su presentación en los registros de **MWD**, así como su medición y relevancia en el campo, esta última recabada de reportes de actividades de pozos perforados en el campo noruego, mientras que el Capítulo 5. Predicción de Vibraciones Torsionales muestra el proceso para construir modelos de predicción de **ML** utilizando la librería de Python llamada *Scikit-learn* a partir de estos registros, y los *survey's*; resulta de importancia mencionar que en este capítulo se analizan mediante la ciencia de datos estos parámetros de perforación, para encontrar información que pudiera resultar útil en el diseño y selección de parámetros de futuros pozos. Los modelos construidos fueron tres, estos consistieron en un árbol de decisión con parámetros de peso y rotación, para mostrar únicamente el proceso de construcción y predicción del algoritmo siguiente, el *Random Forest*, para el que se elaboraron dos modelos, uno de ellos con parámetros de perforación, y otro agregándole datos direccionales, resultando de la siguiente manera.

Nombre modelo	Predictores	Dimensiones		Precisión
		Training set Y	Testing set Y _{pred}	
rf	[BPOS, BVEL, DEPT, GR, HKLD, ROP5, RPM, SPPA, SWOB, TFLO, TQA]	197 823	49 456	99.888%
rf_direccional	[BPOS, BVEL, DEPT, GR, HKLD, ROP5, RPM, SPPA, SWOB, TFLO, TQA, INCL, AZI, DLS]	8 476	2 119	97.352%

Resaltando la relevancia de la adquisición de datos para la construcción y entrenamiento de modelos de **ML** y el potencial beneficio que puede aportar para el diseño de siguientes pozos y mitigación oportuna de eventos de **S&S**.

Abstract

The work presented seeks to apply tools from the Data Science and Machine Learning (**ML**), more precise the Decision tree as a milestone of the Random Forest, which are applied to predict torsional vibrations in the drillstring, to function as a tool to mitigate and if possible, predict Stick & Slips events.

In order to accomplish this task, an introduction must be done into this fields, according to the benefits they can provide to the petroleum engineer duties; the first chapter gives an introduction to python programming language, its importance and open-source capabilities, also is introduced the data available to construct such models, open data initiative from Equinor’s *Volve* field.

Chapter 2 explains the elements that conform the bottom-hole assembly, therefore the measuring tools downhole, but also in the surface, these measures being presented graphically and its transmission protocols; Chapter 3 content s focus on the **ML** algorithms, the main purpose of them and its classification, also elements of Data science are introduced to analyze and visualize data that later on will be analyzed and needed to construct the prediction models.

Chapter 4 titled Stick & Slip (**S&S**) focus on the description of several drillstring vibrations in which the S&S is part of, also its representation in the **MWD** data and its relevance mentioned in the End Of Well Reports of each analyzed well, this will be useful in the next Chapter 5, that focus on the workflow to build **ML** models with the real time and surveying data, using the Scikit-Learn library, using as well, Data science techniques to analyze **S&S** data and seeking useful insights from them and could be used to design and select drilling parameters for future wells.

Three models were created to predict **S&S** events, the first of was built using a decision tree with weight on bit and **RPM**’s as predictors, to show the building process of the random forest algorithm, to which 2 models were built using solely **MWD** data and one that has directional parameters, the results as follows

Model name	Predictors	Dimensions		Accuracy
		“Training set”	“Testing set”	
		X	X	
Rf	[BPOS, BVEL, DEPT, GR, HKLD, ROP5, RPM, SPPA, SWOB, TFLO, TQA]	197 823	49 456	99.888%
rf_directional	[BPOS, BVEL, DEPT, GR, HKLD, ROP5, RPM, SPPA, SWOB, TFLO, TQA, INCL, AZI, DLS]	8 476	2 119	97.352%

Showing the importance of the data gathering process to build and develop training sets, as the potential benefits that the applications of this methods could carry to design, mitigate and avoid **S&S** events.

Capítulo 1. Aspectos Fundamentales

1.1 Campo Volve

Este fue un campo localizado en el mar territorial de Noruega y desarrollado por su compañía nacional o **NOC** llamada en ese momento Statoil, renombrada Equinor en el año 2018, su localización se muestra en la Figura 1.1.1 este yacimiento fue descubierto en 1993 y su plan de desarrollo se aprobó en 2005, produciendo crudo hasta el año 2016 en aguas someras con un tirante de agua de 80[m] y cuya formación productora se encuentra en la formación *Hugin* del grupo *Vestland* del jurásico medio, estableciendo el yacimiento a una profundidad de 2,750 – 3,120 [m] como se muestra en Figura 1.1.3.

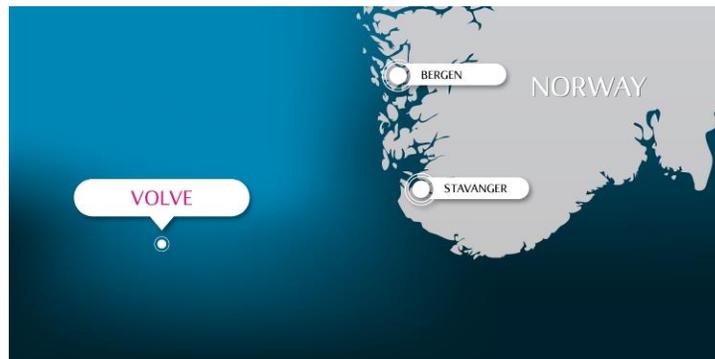


Figura 1.1.1 : Ubicación de Volve

La producción de este yacimiento se muestra en la Figura 1.1.2, en esta es posible apreciar que existe producción de gas asociado (amarillo) a lo largo de la vida productiva del yacimiento, así mismo, al inicio de producción de crudo (verde) se tienen pozos de inyección de agua (azul oscuro) para la preservación de la presión; el aporte de agua (azul) ocurre de manera significativa a la producción después del año 2009, mostrándose las producciones e inyecciones acumuladas en la Tabla 1.

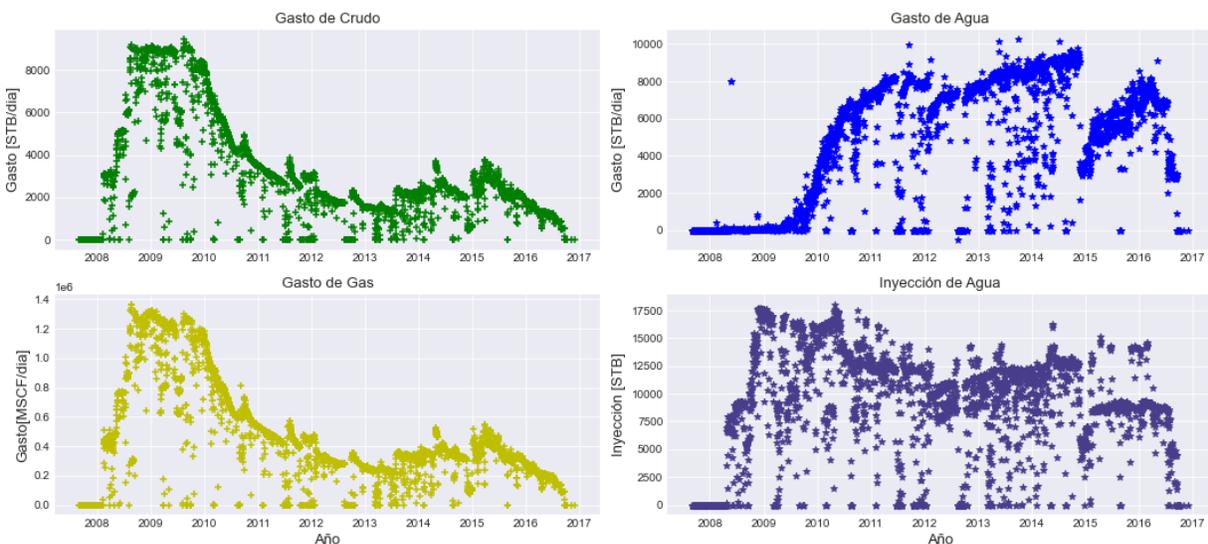


Figura 1.1.2 : Producción de Volve

Tabla 1 : Acumulados de producción e inyección

Modo	Fluido	Acumulado
Producción	Aceite	10.0370 [MMSTB]
	Agua	15.3195 [MMSTB]
	Gas	1,475.3704 [MMSCF]
Inyección	Agua	30.3301 [MMSTB]

En el año 2018, una vez abandonado, la operadora petrolera otorgó la licencia de uso y visualización al público de los datos referentes a este yacimiento, entre los cuales se encuentran los registros de perforación, petrofísica, producción, así como de simulación y sísmica para el estudio por parte de estudiantes o instituciones de investigación y de esta forma contribuir al mejoramiento de prácticas y procesos primordialmente del *Upstream*.

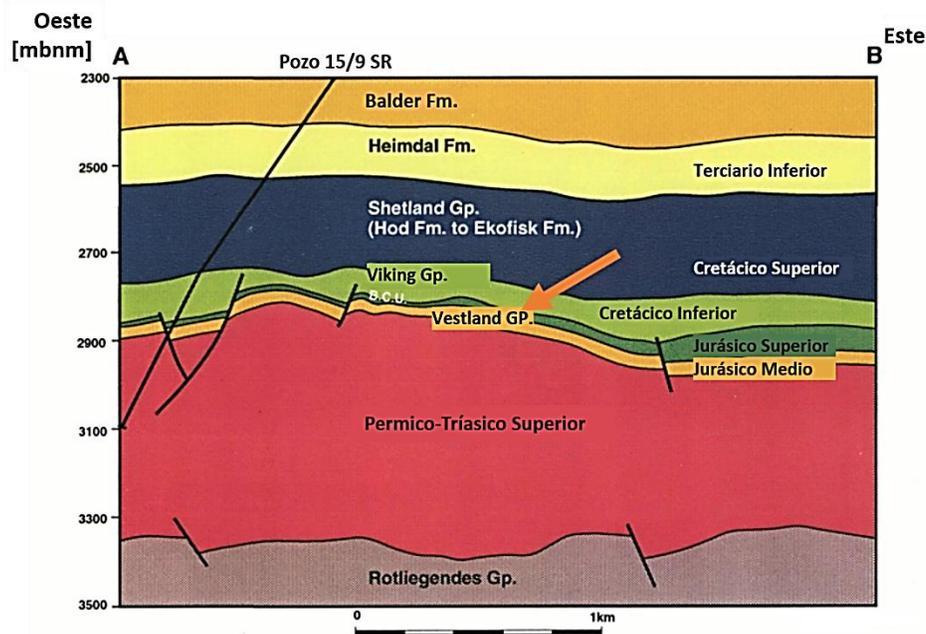


Figura 1.1.3 : Formaciones de la Estructura Theta Vest (Discovery Report, 1993)

La perforación del pozo **15/9-19 SR** en el año de 1992 tuvo como propósito la exploración de la formación *Heimdal*, la cual se le denomina *Sele* actualmente y se encuentra en el Paleoceno, que hasta ese momento se conocía como productora de gas y condensado, así como exploración de la formación *Hugin* del Jurásico/Triásico, cuyos horizontes no habían sido definidos, mostrándose con la flecha en la Figura 1.1.3.

En la Tabla 2 se muestra la litología definida de esta región, la cual sitúa a la formación productora en el grupo de areniscas gradadas de *Vestland*, y que también se encuentra marcada de color naranja.

Tabla 2 : Litología de la región

Grupo	Formación	Descripción
	Lecho Marino	Arenas densas con arcilla
	Cuaternaria	Arcilla con intercalaciones de arenas
Nordland	Utsira	Arenisca gradada con intercalaciones de limos y caliza
Hordaland	Skade	Arenisca moderadamente gradada con roca calcárea cementada
	Grid	Arenisca fina
Rogaland	Balder	Lutita arcillosa con intercalaciones de caliza
	Sele	Lutita arcillosa con intercalaciones de calizas y trazas tobeáceas
	Lista	Lutita arcillosa no calcárea con trazas de calizas
	Ty	Arenisca gradada fina a pobre, con intercalaciones de lutita arcillosa, <i>siltstone</i> y caliza
Shetland	Ekofisk	Caliza moderadamente dura con restos de arcilla y arenisca
	Tor	Caliza dura con trazas de lutita arcillosa
	Hod	Caliza calcárea a marga con glauconita
	Blodøks	Marga gris a oscura con laminaciones arcillosas
	Hidra	Piedra caliza moderadamente dura
Cromer Knoll	Rødby	Roca marga con laminaciones arcillosas
	Sola	Roca marga con intercalaciones de lutita arcillosa
	Åsgard	Intercalaciones de marga y calizas con arcilla y limolita
Viking	Draupne	Arcilla orgánica con trazas de pirita
	Heather	Lutita arcillosa con caliza e intercalaciones arcillosas
Vestland	Hugin	Arenisca de grano fino a grueso con trazas de lutita arcillosa
	Sleipner	Arenisca de grano fino a grueso con intercalaciones de carbón
Hegre	Skagerrak	Arenisca fina con laminaciones de limos
	Smith Bank	Lutita arcillosa con trazas de arenisca

Resaltando que la formación *Hugin* tiene una litología propia de arenas de depósito marino de alta energía, en la cual se tienen los pozos productores **F1**, **F11**, **F12**, **F14**, **F15** y **F5** y los pozos inyectores **F4** y la reconversión del pozo **F5**; los cuales pueden verse en el modelo de simulación de la Figura 1.1.4 donde las condiciones al inicio de la producción se encuentra en la imagen superior, y la saturación remanente de crudo (verde) al final de su vida productiva se muestra en la imagen inferior, el factor de recobro para este yacimiento fue de 54%.

1.2 Python

Este es un lenguaje de programación de alto nivel, es decir, su estructuración y sus algoritmos se expresan de una forma entendible por la capacidad cognitiva humana; creado en 1980 por Guido Von Rossum, Python ha adquirido una atención y soporte gracias a su versatilidad y multi-soporte entre plataformas tecnológicas.

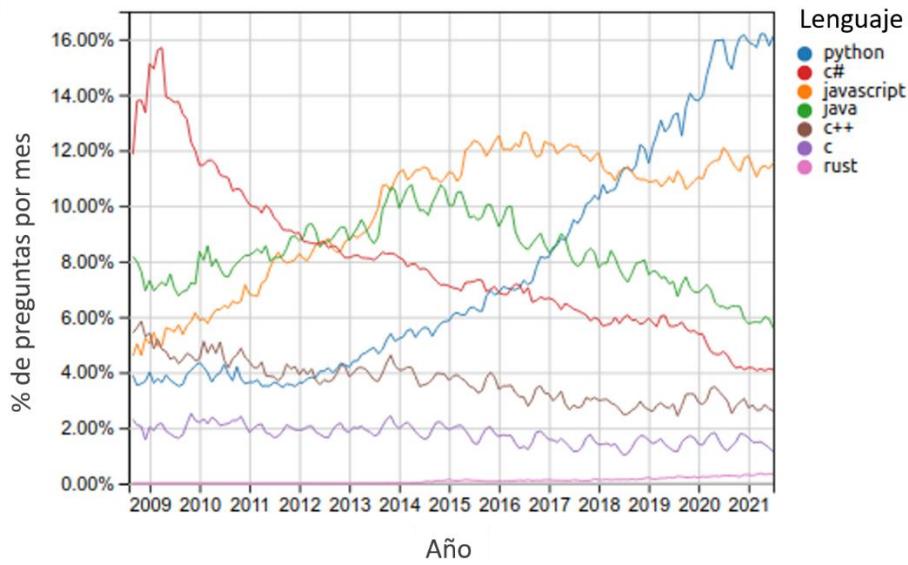


Figura 1.2.1 : Crecimiento de lenguajes de programación (TrustDimension, 2021)

La Figura 1.2.1 muestra el gráfico de crecimiento de distintos lenguajes de programación a lo largo de los años en función del porcentaje de preguntas realizadas al mes en la comunidad de *Stack-Overflow*, plataforma publica de preguntas y respuestas para programadores profesionales y aficionados

La popularidad de este lenguaje de programación en el rubro de *Machine Learning (ML)* radica en la cantidad de bibliotecas disponibles y de código libre, es decir, libre para edición y distribución; entre las que se destaca la biblioteca *Scikit-learn* la cual contiene algoritmos actualizados y optimizados continuamente de **ML**, aunado a estas herramientas, en este lenguaje se pueden hacer actividades desde la importación de archivos hasta la visualización y análisis especializado de estos.

La plataforma denominada Anaconda, es un manejador de paquetes y bibliotecas de Python, en esta se encuentran diferentes interfaces graficas para programar o **IDE's** en este lenguaje, siendo *Jupyter Notebook* la más relevante para el análisis de datos, ya que su versatilidad, estructura y sencillez facilita las tareas de análisis y manejo de datos.

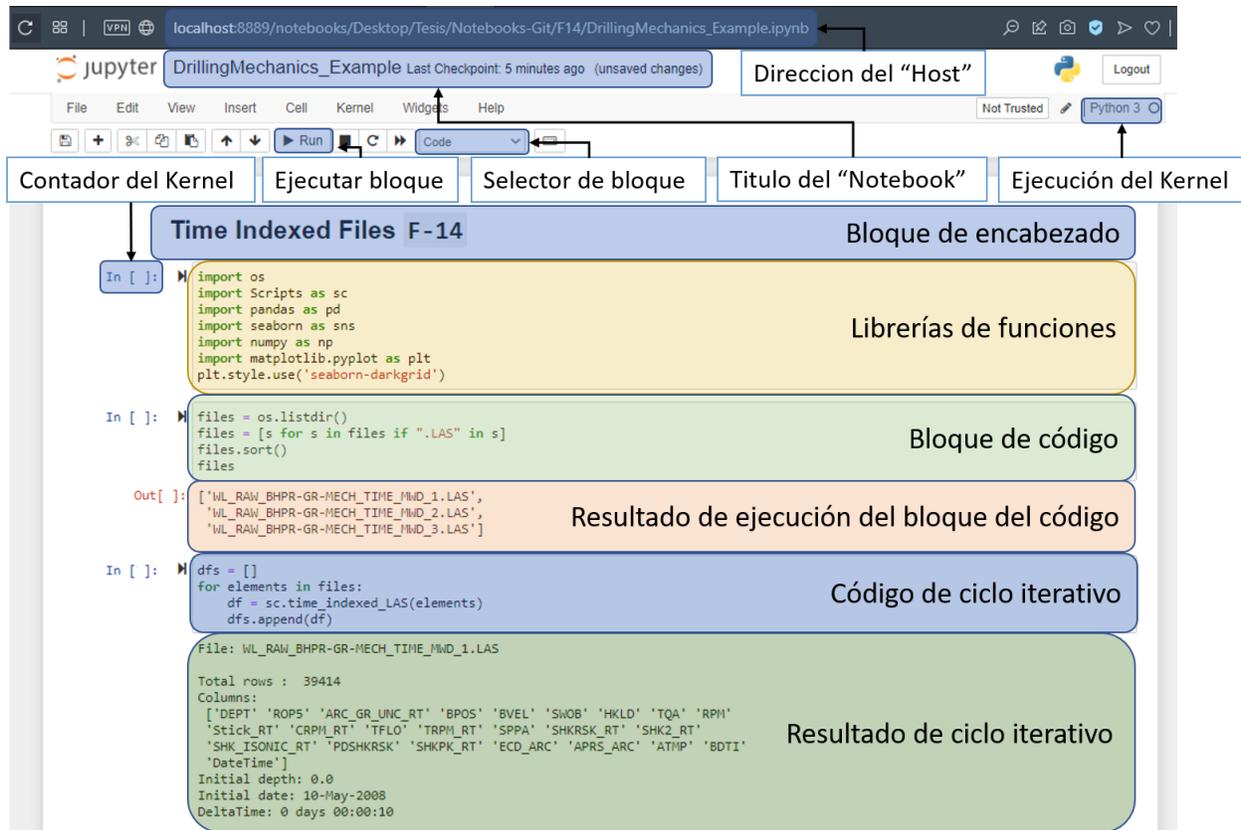


Figura 1.2.2 : "IDE" de "Jupyter Notebook"

En la Figura 1.2.2 se muestran los elementos esenciales para el usuario de la plataforma *Jupyter Notebook*, siendo esta, una aplicación web y cuya estructura se enfoca en emular un flujo de trabajo mediante bloques de código seguido de su bloque de ejecución, y para permitir una mejor visualización, bloques de encabezado, así como, también se muestra en el bloque verde la función "sc.time_indexed_LAS", función fundamental para la elaboración de este trabajo, dicho algoritmo se encuentra anexado en el Anexo 2. Códigos Elementales, el cual realiza a partir de los registros de perforación en tiempo, una base de datos para su posterior análisis e interpretación.

Capítulo 2. Herramientas de Perforación

2.1 Sarta de Perforación

Los hidrocarburos, fundamento energético del último siglo, se encuentran en formaciones rocosas en el subsuelo, para poder acceder a este recurso es necesario la creación de un conducto que permita la producción de estos hidrocarburos a la superficie.

Este proceso requiere la construcción de un pozo, cuya profundidad dependerá de la formación contenedora del recurso, de la localización superficial desde donde se realizará dicho pozo, así como de la infraestructura disponible; esta última incluye al equipo de perforación, que está conformado convencionalmente por un sistema de izaje, un sistema de potencia, un sistema de circulación, prevención de brotes, los cuales son mostrados en la Figura 2.1.1.

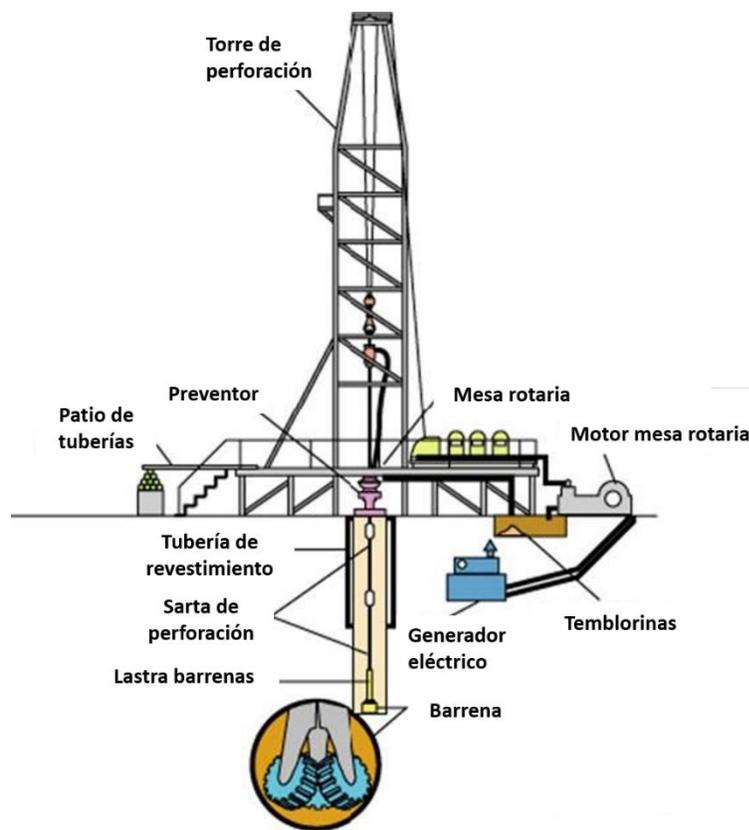


Figura 2.1.1 : Sistemas en la perforación (Salid232, 2019)

Por último, el tipo y diseño del equipo de perforación dependerá de la localización, pudiendo ser terrestre *onshore* o costa afuera *offshore*.

Este sistema de rotación se conforma por:

- Mesa rotaria o Top Drive, **TDS** por sus siglas en inglés, *Top Drive System*
Se enfoca en la transmisión de potencia y torque hacia la sarta
- Tubería de perforación **DP**
Transmite el movimiento rotatorio del equipo superficial a la barrena y permite la circulación del lodo por la sección interna de la tubería.
- Tubería de perforación extrapesada **HWDP**
Proveer de peso a la barrena, así como transicionar entre las tuberías de perforación y lastra barrenas.
- Lastra barrenas **DC**
Adicionar peso a la barrena y mantener la sección de la tubería de perforación en tensión durante la perforación, en la Figura 2.1.2 se muestra el punto neutro, este, es la transición entre las cargas de tensión para las secciones superiores y compresión para las inferiores.
- Estabilizadores
Proveen una distancia fija a la pared del pozo, manteniendo el aparejo concéntrico en el pozo, y de esta forma reducir el pandeo o flexión.
- Percusores o martillos de perforación
Dispositivos mecánicos e hidráulicos encargados de liberar algún componente del aparejo que se haya atascado, a partir de la generación de un impacto a sí mismo.
- **MWD o LWD**
Dispositivos de medición de las condiciones y parámetros de perforación, así como registros petrofísicos simultáneos a la perforación, estos se describirán a detalle más adelante.
- Barrena
Herramienta para fallar la roca.

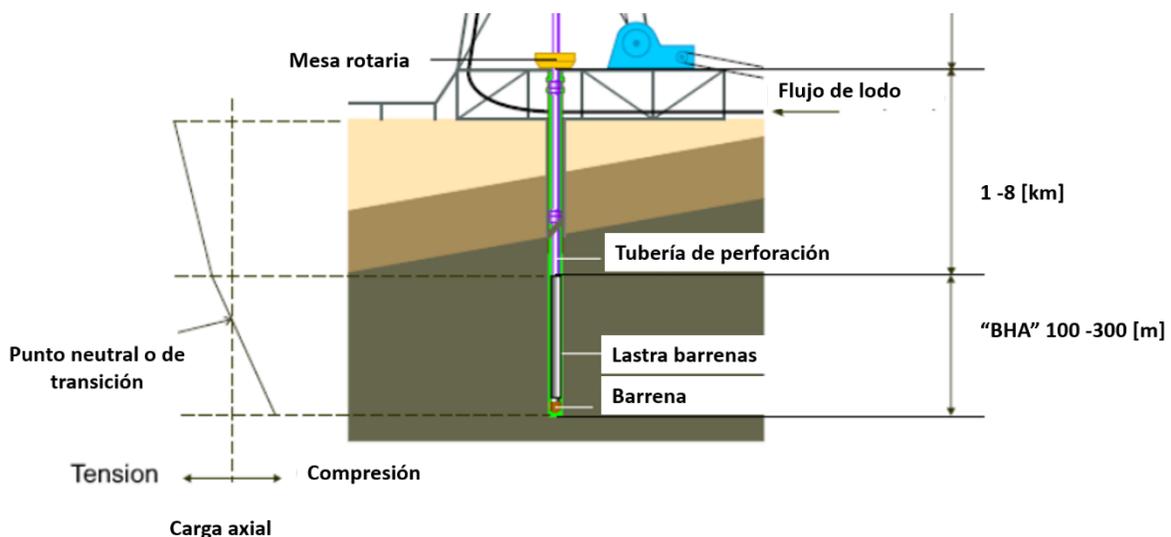


Figura 2.1.2 : Aparejo de fondo (Mayer, 2007)

El ensamble de fondo o **BHA** por sus siglas en inglés, consiste en los componentes inferiores al **HWDP** como se muestra en la Figura 2.1.2, siendo estos los más relevantes para tomar en cuenta al momento del diseño, ya que estos podrán ser sometidos a las siguientes condiciones operativas:

- Esfuerzos de colapso y ruptura
- Esfuerzo de tensión y compresión
- Torsión
- Presencia de componentes abrasivos
- Fatiga
- Fricción
- Vibración

Aunado a estas condiciones operativas, pueden existir condiciones direccionales que han de ser tomadas en cuenta, ya que el diseño de la trayectoria al objetivo geológico puede variar, por lo que es necesario la implementación de herramientas adicionales y técnicas, enfocadas en cambiar la dirección del aparejo, estos funcionan primordialmente en la aplicación de una fuerza lateral y la generación de un ángulo de inclinación al eje del **BHA** y como se muestra en la Figura 2.1.3

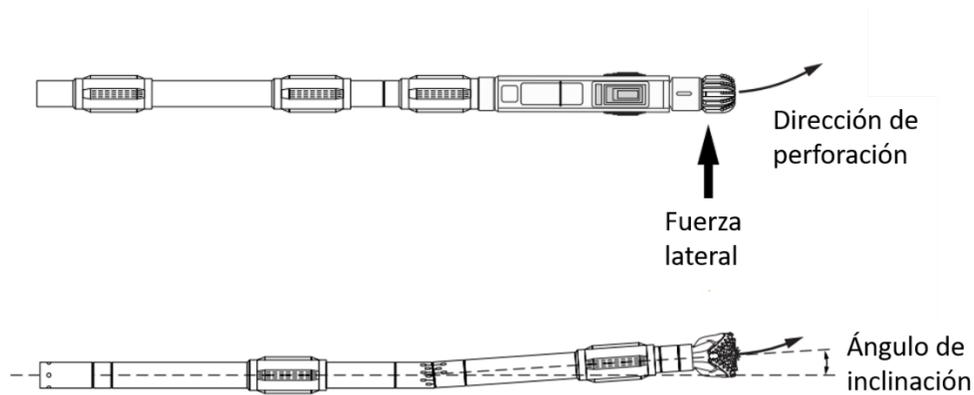


Figura 2.1.3 Principios de desviación (Azar, 2007)

En general, para llevar a cabo el control direccional de un pozo se emplea un ensamble de fondo direccional, pudiendo ser:

- *Bent Sub* y motor de fondo
Estos dispositivos se utilizan simultáneamente, ya que el *bent sub* o codo desviador ajustable es una sección de tubería que tiene una desviación menor a 3° a diferencia de su lado contrario, acotando su uso a porciones de *build up* o *drop off* de la trayectoria, es decir, secciones cuyos ángulos son modificados para modificar la trayectoria.
- Sistema rotatorio direccional (**RSS**)
Herramienta capaz de hacer cambios a la inclinación y al azimut mientras la sarta rota en forma continua, a través de aletas que aplican una fuerza lateral.

En la Figura 2.1.3, recuadro amarillo se muestra el diagrama del **BHA** del pozo **F14** correspondiente a la sección de la barrena de 8 ½”, este contiene gráficamente el ensamble de fondo; las especificaciones de la herramienta direccional **PD675** y cuya tubería de perforación tiene un diámetro de 5”, datos mostrados en el recuadro gris, en el recuadro verde, la distribución de los componentes enlistados, notando una barrena de **PDC**, así como dispositivos para realizar las mediciones de **LWD** y **MWD**, martillos o *Jar* para liberar la barrena ante cualquier eventualidad, y varios estabilizadores, los cuales también son mostrados en el esquema, en la columna o recuadro azul, la longitud acumulada; estos diagramas se encuentran en el programa de perforación de cada pozo, y es común encontrar dos diseños diferentes por cada sección o diámetro de la barrena, siendo el segundo diseño más drástico que el utilizado de primera instancia, y esto es por motivos operativos y contingentes.

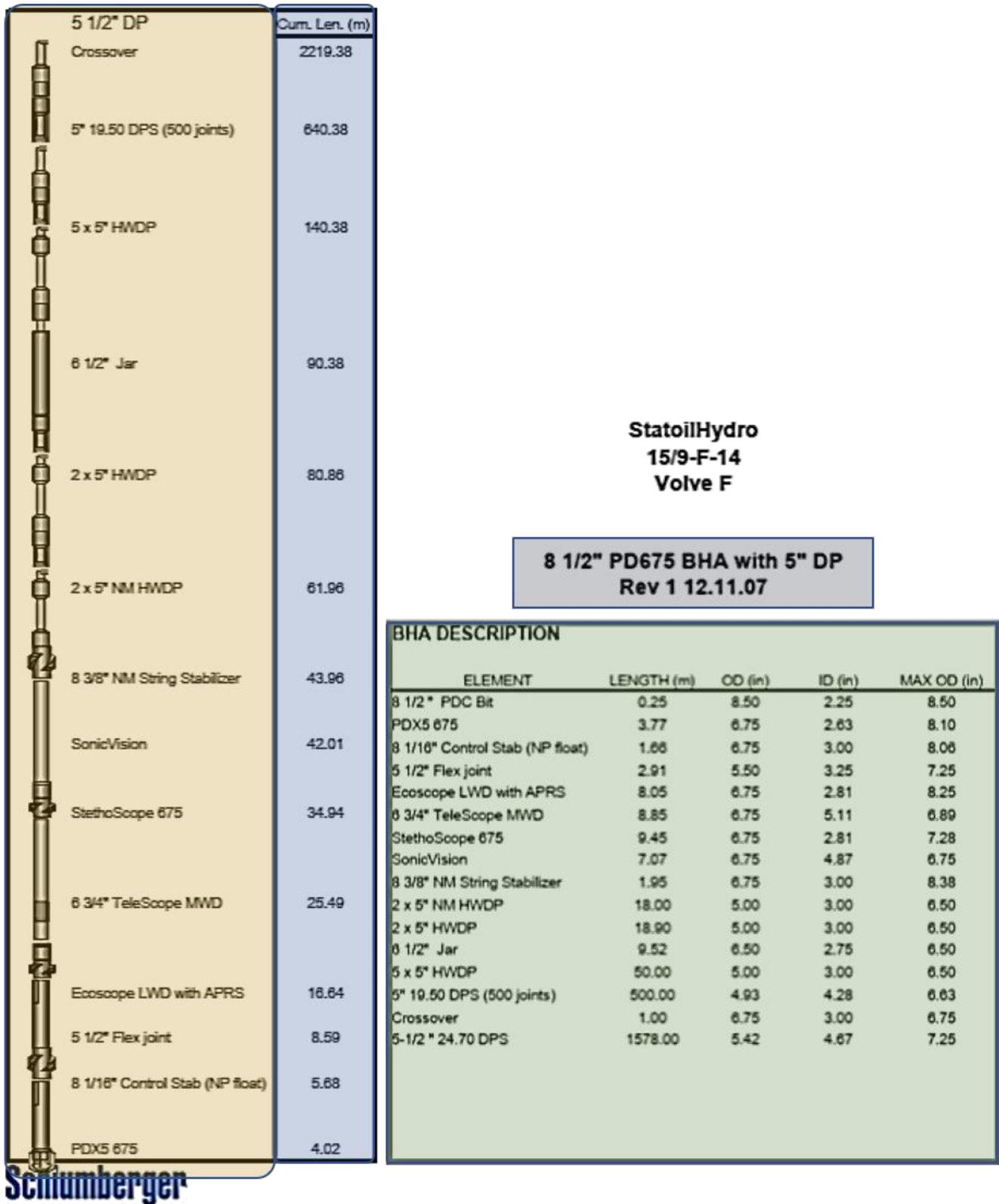


Figura 2.1.4 : BHA de 8 1/2" del pozo F-14 (Drilling program F-14, 2008)

2.2 Adquisición de datos

En el mar de norte, a finales de los años setentas, debido a los cambios a la regulación en ámbito de perforación en la industria petrolera, se desarrollaron herramientas que buscaban hacer más eficiente dicho proceso, ya que este consistía en, perforar un par de metros, retirar la sarta de perforación, tomar el registro cableado de *survey* para determinar la inclinación y el azimut que iba adquiriendo el pozo; este método resulto ser incosteable debido al tiempo no productivo o **NPT** por sus siglas en inglés, por lo que se intentó acoplar dicha herramienta en el aparejo de perforación y así evitar los viajes para toma de *survey*.

En la Figura 2.2.1 se muestra el carrete de cable metálico o *wireline* con el cual era bajado el dispositivo medidor en los años de 1930.

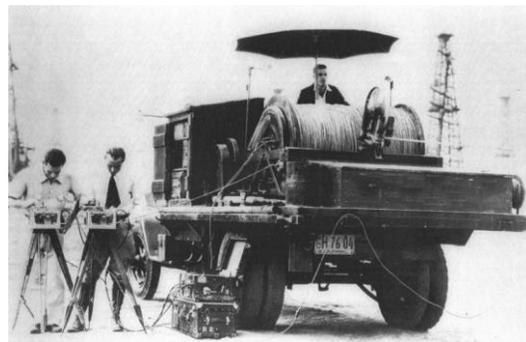


Figura 2.2.1 : Toma de *survey* (Frontier MWD Systems, 2014)

Estas mediciones de trayectoria o *survey* son tomadas hoy en día de forma puntual, después de la conexión de una nueva lingada a través de pulsos de presión o dependiendo de la herramienta utilizada, hasta que la presión de trabajo por el lodo sea alcanzado y de esta forma obtener mediciones de azimut o **AZI**, siendo esta la dirección del pozo con referencia al norte magnético; inclinación o **INCL**, la desviación angular del pozo a la vertical; y el *dog leg severity* o **DLS**, que indica los cambios repentinos experimentados la trayectoria; por mencionar algunos.

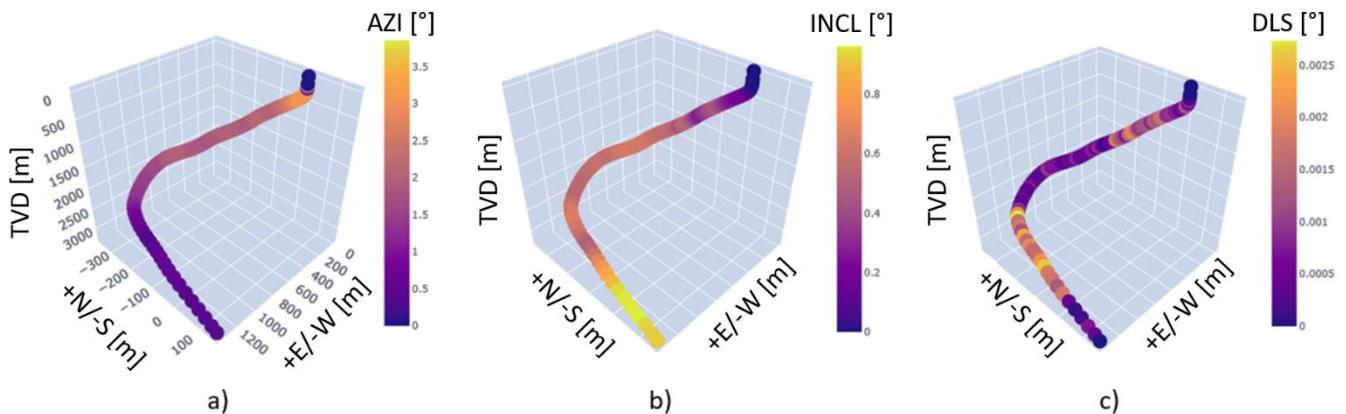


Figura 2.2.2 : "Survey" de perforación pozo F5

En la Figura 2.2.2 se muestran los *survey's* correspondientes al pozo **F5**, donde en el inciso (a) se muestra el azimut de la herramienta en cada medición y donde es posible apreciar un ángulo elevado justo después del **KOP** o punto de desviación; el (b) muestra el ángulo de inclinación, notando que en la sección más profunda del pozo aumenta este ángulo lo que permite la horizontalidad del mismo; y por último, el (c) muestra la severidad del **DLS**, el cual es la variación de la inclinación entre las estaciones de medición y cuyo intervalo de 2,500-3,000 [m] en **TVD** o en la profundidad vertical muestra el esfuerzo del perforador por mantener la trayectoria establecida.

En el desarrollo de esta herramienta, se intentó acoplar también otras mediciones o registros para aumentar la eficiencia en tiempo, y la recopilación de datos en cada pozo, existiendo las herramientas de:

- Mediciones al perforar o **MWD**
Esta herramienta aporta las mediciones de las condiciones mecánicas de la perforación, tanto en los sistemas superficiales y subsuperficiales.
- Registros al perforar o **LWD**
Estos consisten en la toma de registros petrofísicos mientras el proceso de perforación se lleva a cabo.

Algunas de las mediciones llevadas a cabo por estas herramientas son realizadas por sensores localizados en el **BHA** y pueden ser almacenados o transmitidos para ser analizados y realizar mejoras, mitigaciones de problemas por parte del equipo de perforación, en la Figura 2.2.3 se muestra el perfil de temperatura en los datos de perforación del pozo **F10**, con la flecha verde se muestra el aumento gradual de la temperatura y profundidad, mientras que la flecha roja nos muestra el perfil de disminución de la temperatura al ir removiendo el aparejo del pozo.

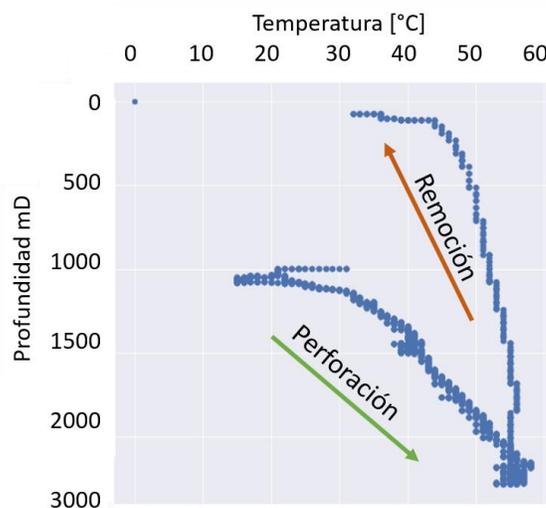


Figura 2.2.3 : Registro de Temperatura-Profundidad pozo F10

En estos registros de medición **MWD**, es posible apreciar mediciones de fondo, y superficiales, agrupadas por protocolos de transmisión de datos que más adelante se describirán; en la Figura 2.2.4 se presentan los registros descritos en la Tabla 3.

Tabla 3 : Registros de perforación

Nombre	Mnemónico	Unidad	Descripción
Presión de fondo	APRS_ARC	psi	Mediciones de la presión anular en fondo
Gamma Ray	ARC_GR_UNC	°api	El registro de emisión de elementos radioactivos endógenos a la roca
Temperatura de fondo	ATMP	°C	Medición de temperatura de espacio anular en fondo
Posición del bloque viajero	BPOS	m	La altura a la cual se encuentra el bloque viajero en el mástil o torre de perforación
Velocidad del bloque viajero	BVEL	m/s	La velocidad de desplazamiento del bloque y el sistema de izaje
Revoluciones de la sarta de perforación	CRPM_RT	rpm	Revoluciones del BHA a nivel del lastraberrenas
Profundidad	DEPT	m	Registro de profundidad en metros desarrollados o MD .
Densidad equivalente	ECD	gr/cm ³	La densidad del fluido en fondo del pozo
Carga sobre el gancho	HKLD	ton	Medición de la carga en el gancho por parte de la sarta de trabajo y perforación
Tasa de penetración	ROP5	m/hr	El avance de perforación en distancia y tiempo promediado cada 5 [ft].
Revoluciones por minuto	RPM	rpm	Revolución del sistema de rotación, top drive o mesa rotaria
Presión de bombeo	SPPA	kPa	Las pérdidas por fricción en el sistema hidráulico
Peso sobre la barrena	SWOB	ton	Medición del peso ejercido por la sarta a la barrena y por ende a la formación
“Stick & Slip”	Stick_RT	rpm	Vibración torsional
Flujo total	TFLO	lpm	Razón de flujo de lodo de perforación al pozo
Torque	TQA	KMDN	Torque ejercido por el TDS derivado por de la rotación de la sarta en unidades de kilómetro decanewton
Revoluciones de la turbina de fondo	TRPM_RT	rpm	Medición de las revoluciones de la turbina contenida en el BHA para energizar componentes de fondo

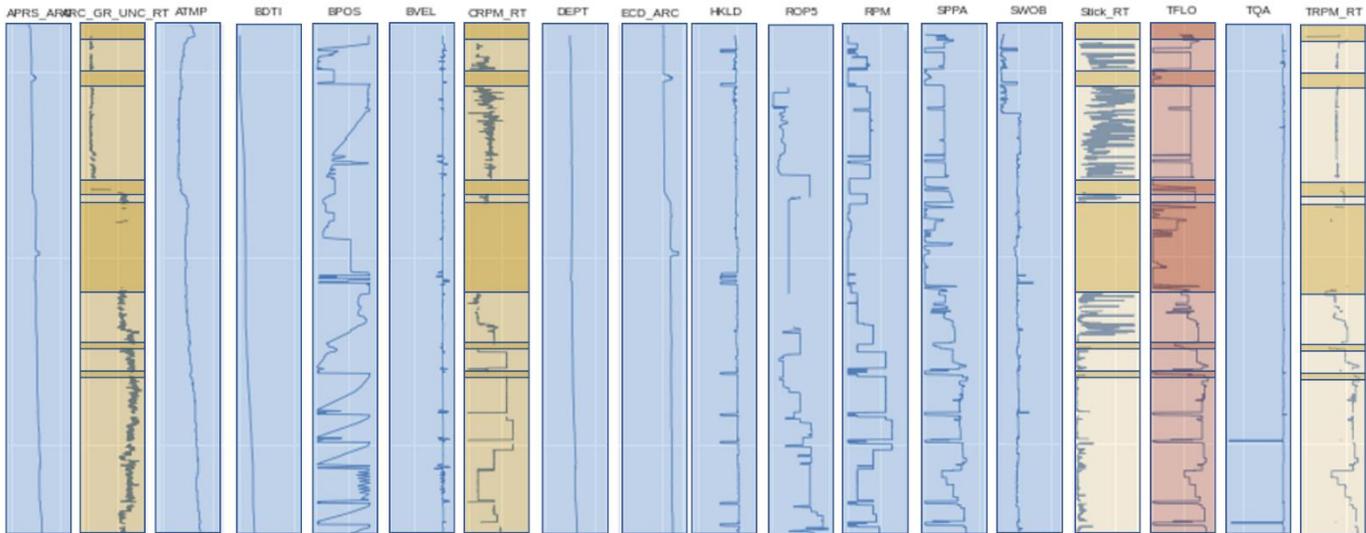


Figura 2.2.4 : Registro de corrida de perforación

Resulta relevante mencionar, que la nomenclatura para titular cada registro puede variar dependiendo de la herramienta o compañía, sin embargo, es común que el nombre contenga indicios de la medición que esta presenta.

A partir de la Figura 2.2.4 se pueden realizar varios comentarios, estos registros pueden contener contingencias, actividades adicionales, que se desarrollaron en el periodo de toma de datos, idealmente estos registros deberán contener desde que se mete el aparejo al pozo, hasta el momento en que se remueve del mismo, aunque esto no necesariamente ocurra; por otra parte, algunos registros pueden mostrar datos faltantes, los cuales se muestran en tono oscuro.

Estos registros pueden ser analizados en la posteridad, para determinar el **NPT**, que es una medida de productividad y desempeño de la perforación de determinado pozo, en la Figura 2.2.5 se muestran dos secciones de toma de registros, en el recuadro azul, se muestra el proceso de *Run In Hole* o **RIH**, cuyo propósito es insertar la sarta el pozo, en los recuadros más pequeños se muestran varios parámetros que varían en dicho proceso, la carga sobre el gancho aumenta gradualmente (b), se le puede dotar de rotación (a) y existir flujo ocasional de lodo o romper circulación como se le denomina en campo (a); el morado muestra la preparación del proceso *drill out* o rebajar cemento y accesorios, el cual consiste en la perforación del *plug* y del cemento remanente de la sección anterior; en el amarillo se muestra el proceso llamado *Circulating Bottoms-Up* o circular el pozo, actividad realizada para remover los recortes del fondo y preparar el fondo para la zapata de la tubería de revestimiento y que se caracteriza por la elevación cíclica de la sarta (d), junto con una rotación de la sarta variable (a); por último se muestra en la verde, la remoción de la sarta conocido como *Pull Out Of Hole* o **POOH**, donde también se aprecia en (f) la remoción de la sarta variando el peso, esto pudiendo indicar la presencia de *drag* o arrastre por falta de limpieza o por la construcción del pozo y después la remoción constante ininterrumpida de la sarta (g).

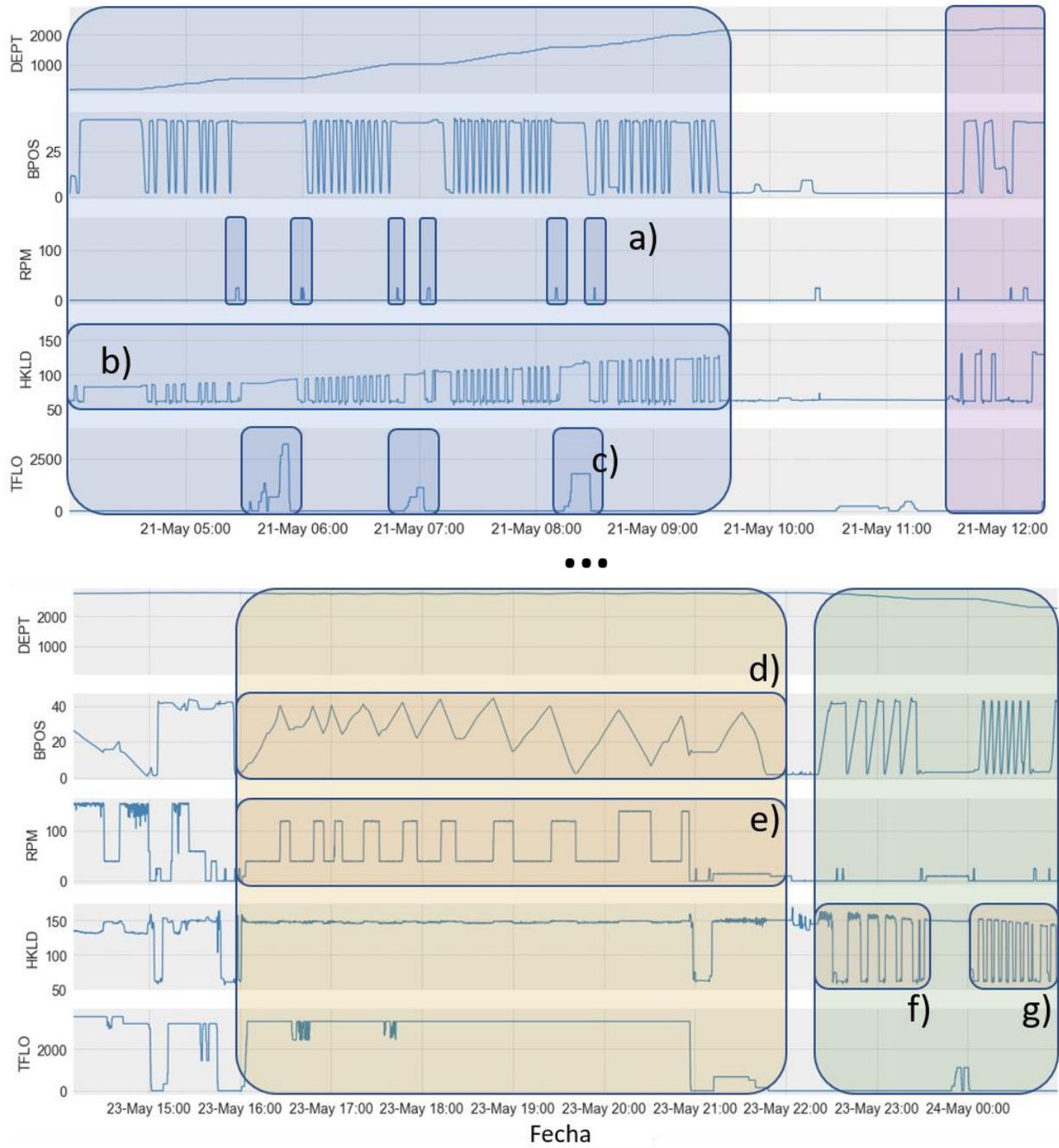


Figura 2.2.5 : Análisis de corrida de perforación

2.3 Transmisión de datos

El sistema que hace posible la transmisión de determinados datos del **BHA**, llámese **LWD** y **MWD**, a la superficie se hace gracias a los pulsos telemétricos en el lodo, por sus siglas en inglés **MDT**; este sistema emite pulsaciones de presión a través del lodo de perforación, señales analógicas producidas por la herramienta que son convertidas a señales digitales y transmitidas a la superficie mediante las siguientes modalidades:

- Pulso positivo telemétrico
Un elemento restrictor de flujo produce un cambio en la presión del lodo de modo que se creen pulsos, como se muestra en la siguiente Figura 2.3.1

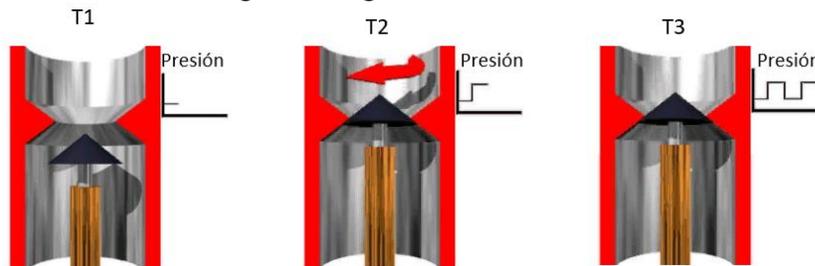


Figura 2.3.1 : Pulso positivo (Morley, 1998)

- Pulso negativo telemétrico
Una válvula desviadora de flujo ocasiona caídas en la presión del lodo, mostrándose en la Figura 2.3.2 una abertura al espacio anular.

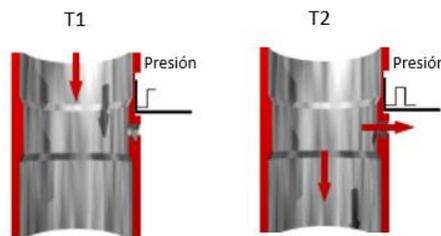


Figura 2.3.2 : Pulso negativo (Morley, 1998)

- Onda telemétrica continua
La rotación de unas placas genera una variación en la presión, como se muestra en la Figura 2.3.3.

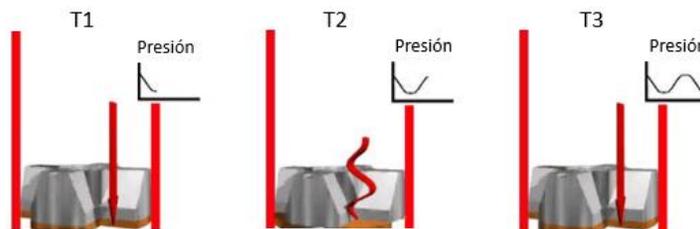


Figura 2.3.3 : Onda continua (Morley, 1998)

Una vez transmitidos los pulsos a través del lodo por el espacio anular, se utiliza un transductor o sensor en la superficie convirtiendo dichos pulsos en señales eléctricas, a continuación, se recaban los datos de superficie para luego un equipo de cómputo decodifique dichas señales eléctricas en información que puede ser interpretado por el personal del piso de perforación e ingenieros como se muestra en la Figura 2.3.4; y luego ser agrupada en protocolos de transmisión de datos en tiempo real para las compañías.

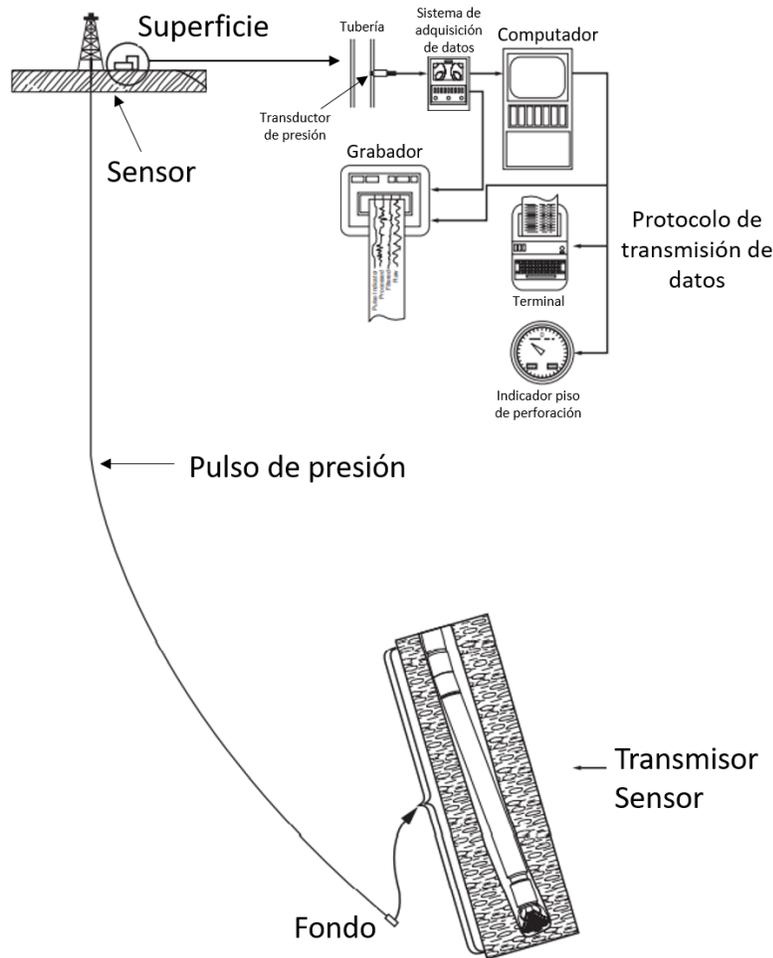


Figura 2.3.4 : Transmisión de datos (Azar, 2007)

Así mismo, algunas herramientas disponen de memorias internas que permiten almacenar las mediciones y, al ser retiradas del pozo, pueden ser obtenidos mediante software para ser analizados en una mayor resolución.

Como se mostró en la Figura 2.2.4, existe una medición denominada **TRPM_RT** la cual muestra la rotación de una turbina que se encuentra dentro del **BHA**, el propósito de esta, es la de dotar de energía a los componentes electrónicos que realizan algunas mediciones, ya que algunos componentes no cuentan con una batería; por lo que al no existir flujo de lodo **TFLO**, rojo oscuro, no existe carga eléctrica generada, y por ende, no existirá medición en algunos carriles, amarillo oscuro.

2.4 Archivos WITSML y LAS

Los registros obtenidos en las corridas al perforar un pozo pueden ser de dos formatos:

- Registros **WITSML**

El propósito del *Wellsite Information Transfer Standard Markup Language* es la estandarización de los datos en sitio sin importar su procedencia, construcción del pozo, planeación y ejecución de otras actividades, para ser transferidos a las oficinas de compañías de servicios, operadoras y terceros a través de protocolos de transferencia siendo estos los que determinaran la transmisión en tiempo real de los datos; estos archivos vienen contenidos en formato **XML**, que permite una gran cantidad de almacenamiento en una estructura de nodos, en la Figura 2.4.1 se muestran las dos presentaciones disponibles de estos archivos, indexados por profundidad en el inciso (a) y aquellos cuya medición fue hecha en función del tiempo, (b), como se muestra en los recuadros rojos; así mismo, en el recuadro gris, la versión de la estructura **XML**; en azul la información referente al pozo; en naranja la sección de perforación; en verde la compañía de servicio; y en morado los intervalos que abarcan los datos almacenados.

```

a) <logs xmlns="http://www.witsml.org/schemas/1series" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.4.1.1">
  <log uidWell="W-924688" uidWellbore="B-924688" uid="L-957090-MD">
    <nameWell>15/9-F-10</nameWell>
    <nameWellbore>15/9-F-10 - Main Wellbore</nameWellbore>
    <name>8.5 in Section - MD Log</name>
    <objectGrowing>>false</objectGrowing>
    <serviceCompany>Schlumberger</serviceCompany>
    <runNumber>9,10</runNumber>
    <pass>Drilling</pass>
    <creationDate>2009-05-17T06:09:03.000Z</creationDate>
    <indexType>measured depth</indexType>
    <startIndex uom="m">4802.124</startIndex>
    <endIndex uom="m">5062.765</endIndex>
    <direction>increasing</direction>
    <indexCurve>DEPTH</indexCurve>
  </log>
</logs>

b) <logs xmlns="http://www.witsml.org/schemas/1series" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.4.1.1">
  <log uidWell="W-924688" uidWellbore="B-924688" uid="L-957090-Time">
    <nameWell>15/9-F-10</nameWell>
    <nameWellbore>15/9-F-10 - Main Wellbore</nameWellbore>
    <name>8.5 in Section - Time Log</name>
    <serviceCompany>Schlumberger</serviceCompany>
    <runNumber>9,10</runNumber>
    <pass>Drilling</pass>
    <creationDate>2009-05-17T06:09:03.000Z</creationDate>
    <indexType>date time</indexType>
    <startDateTimeIndex>2009-06-27T14:29:47.000Z</startDateTimeIndex>
    <endDateTimeIndex>2009-06-27T16:48:59.000Z</endDateTimeIndex>
    <direction>increasing</direction>
    <indexCurve>TIME</indexCurve>
  </log>
</logs>

```

Figura 2.4.1 : Registros WITSML pozo F10

- Registros **LAS**

Estos registros son llevados a cabo por compañías prestadoras de servicios y refinados para su análisis posterior cada 10[seg], esta presentación de registros fue introducida por la Canadian *Well Logging Society* para estandarizar los registros digitales a través de computadores personales en formato *Log ASCII Standard* o **LAS**, en la Figura 2.4.2 se muestran dichos archivos con sus respectivos marcos de referencia o índices, profundidad en (a) y tiempo en (b), como se muestra en los recuadros rojos; en los recuadros amarillos se muestran las fechas de generación de dichos archivos, sin embargo es importante mencionar que las discrepancias entre estas es debido a que los registros de perforación por tiempo comúnmente se muestran por corridas individuales, mientras que los de profundidad muestran la sección completa, tal es el caso mostrado; así mismo, la información mostrada resulta insuficiente, para determinar secciones, corridas por lo que es también de utilidad tener los archivos de encabezados o **ASC**, el cual se muestra en (c).

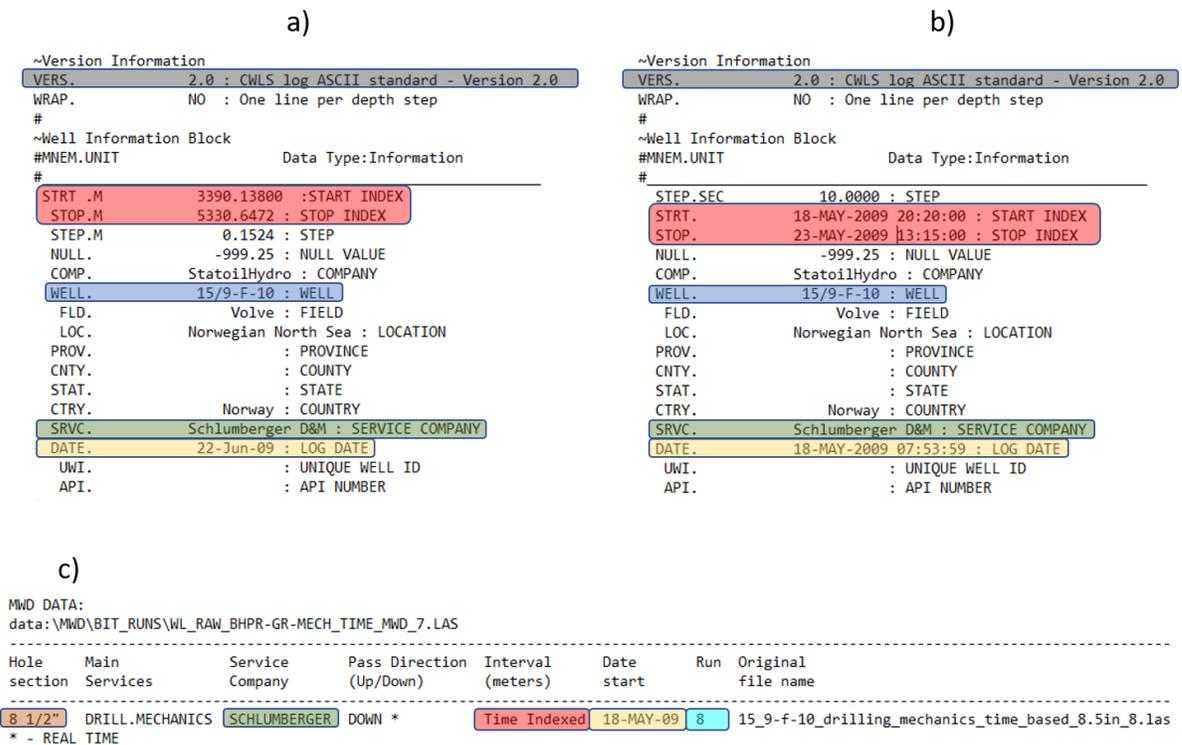


Figura 2.4.2 : Registros LAS de pozo F10

Los datos de **MWD** contenidos en los archivos de **WITSML** son muy robustos y extensos por lo que requieren una capacidad de procesamiento muy amplia, en estos se suelen encontrar más de 100 carriles de registros, muchos datos de los cuales se encuentran faltantes.

Capítulo 3. *Machine Learning*

Alan Turing, precursor de la informática moderna, en su artículo titulado *Computing machinery and intelligence* de 1950 abordó la temática de la inteligencia artificial o **AI** por sus siglas en inglés, a partir de la pregunta *Can machines think?*, a lo largo de este texto plantea escenarios que poco a poco han tomado cabida en la realidad gracias a los avances en el procesamiento y en la elaboración de algoritmos cada vez más complejos, y que pesar de la abstracción realizada resulta de mucho valor lo siguiente:

“One must experiment with teaching one such machine and see how well it learns. One can then try another and see if it is better or worse. There is an obvious connection between this process and evolution [...]” (Turing , 1950)

Este texto nos muestra la variedad de métodos de enseñanza, y la selección de los más adecuados en diferentes tareas, para que con el tiempo se consiga satisfactoriamente la realización de alguna disciplina humana, como el ajedrez.

La inteligencia artificial es una disciplina de la ciencia de la computación, que busca mediante el empleo de determinados algoritmos replicar los procesos que se llevan a cabo en el cerebro humano, por otra parte, el *Machine Learning*, aprendizaje automático o **ML** es un subconjunto de algoritmos que mejoran a lo largo del tiempo, estas disciplinas se muestran en la Figura 3.2.4.1, así como la ciencia de datos, de la que se hablará más adelante.

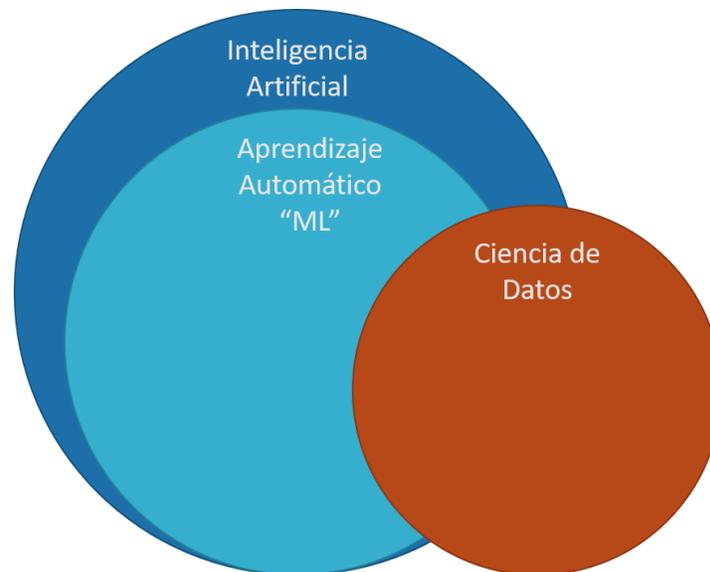


Figura 3.2.4.1 : Subramas de la Inteligencia Artificial

3.1 Aprendizaje supervisado

Como se mencionó anteriormente, en este tipo de aprendizaje se le otorga al algoritmo las etiquetas o *labels* de los datos, en otras palabras, las respuestas correctas, esta es la tarea esencial de estos algoritmos, el aportar nuevas respuestas; en la Figura 3.1.1, se muestra este proceso, con la adición del proceso de evaluación, cuyo propósito es mostrar el grado de exactitud que el algoritmo obtuvo comparándolo con la variable original.

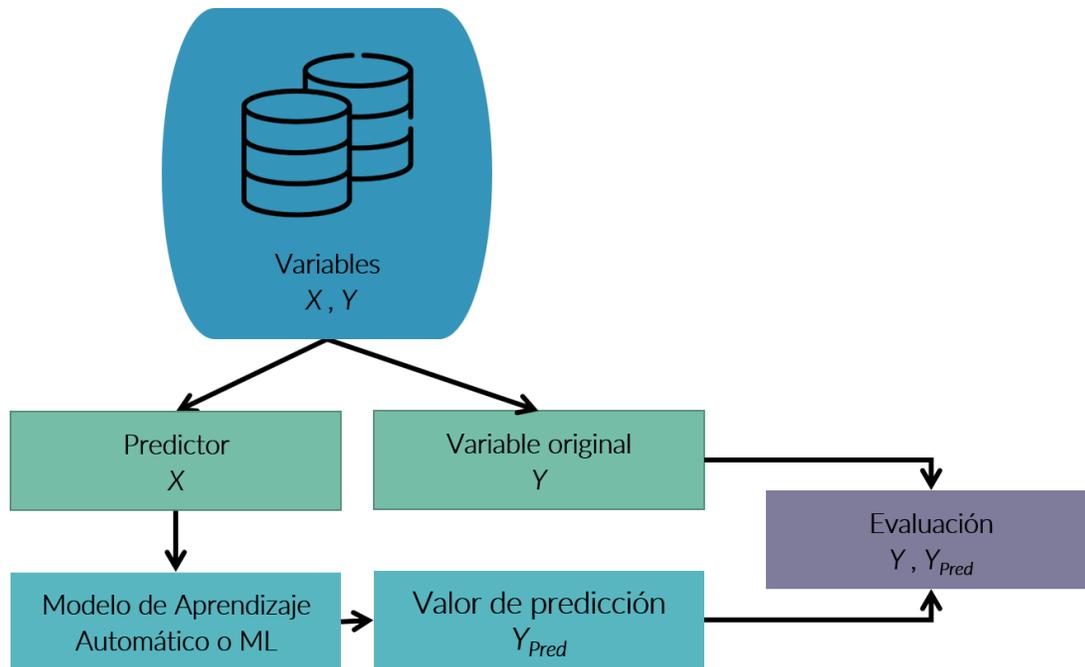


Figura 3.1.1 : Aprendizaje supervisado

Entre los algoritmos de aprendizaje supervisado se pueden generalizar los siguientes tipos de problemas:

- Clasificación
Estos algoritmos buscan predecir los *labels* de la clase categórica con base en los datos dados, entre los que se encuentran el algoritmo de *Support Vector Machines*, árboles de decisión, entre otros.
- Regresión
Predecir un valor continuo a partir de variables, la regresión lineal es el algoritmo más ampliamente conocido, sin embargo, también existen las regresiones polinomiales, entre otros.

Para llevar a cabo satisfactoriamente estos algoritmos es necesario enfocarnos en los preparativos que estos requieren, aunque el flujo de trabajo varíe dadas algunas condiciones y características que necesitan tener los predictores, tales como normalización o estandarización de los datos; el flujo general para aplicar algoritmos de **ML** y resolver problemas de clasificación y regresión se muestra en la Figura 3.1.2

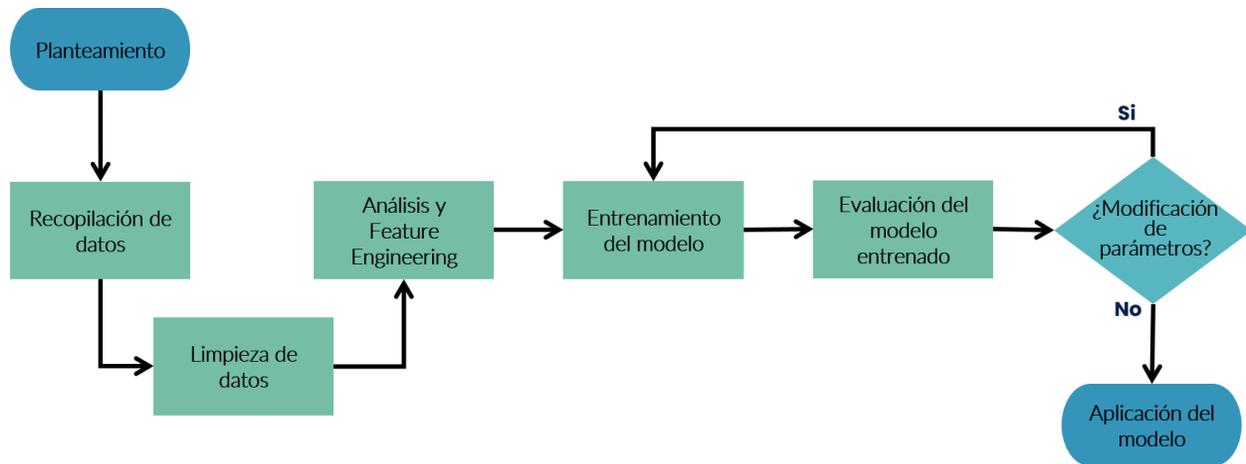


Figura 3.1.2 : Diagrama de flujo para ML (Belyadi, 2021. Modificado)

De esta forma se busca la construcción, y satisfactoria aplicación de un modelo de **ML**, repasándolo con más detalle se tiene el siguiente proceso:

1. Planteamiento: Enfocar la construcción del modelo a un fenómeno específico.
2. Recopilación de datos: La procedencia de los datos puede ser variada y la cantidad de estos es piedra angular para un modelo robusto y de utilidad.
3. Limpieza de datos: Los datos recabados pueden contener información innecesaria, redundante, *outliers* que son mediciones atípicas, y deberán de ser analizados y depurados si es necesario para mejorar la construcción del modelo.
4. *Análisis y Feature Engineering*: De presentarse datos faltantes, estos deberán ser rellenados aplicando diferentes métodos o ser eliminados; así mismo, se podría generar nuevos datos a partir de los iniciales y de esta forma obtener información relevante adicional para el modelo.
5. Entrenamiento del modelo: Se deberán particionar los datos en entrenamiento y *testing*, y construir el algoritmo o algoritmos de elección, a partir de los de entrenamiento.
6. Evaluación del modelo entrenado: Una vez construido el modelo, se deberá evaluar la precisión de este con los datos de *testing*, y de esta forma determinar el número de aciertos que tuvo el modelo con respecto a los originales.
7. Modificación de parámetros: Si el modelo resulta deficiente, se podrán editar algunos parámetros de estos, desde variar el tamaño de los datos de entrenamiento hasta valores de parámetros propios del algoritmo.
8. Aplicación del modelo: El modelo construido ya está listo para su aplicación y continua actualización.

Como es posible ver en este flujo, es de suma importancia la calidad de los datos utilizados para construir y entrenar un modelo de **ML**, por lo que resulta muy conveniente tener nociones básicas de ciencia de datos, el cual es el propósito del subtema siguiente.

3.2 Ciencia de Datos

El *Data Science* o ciencia de datos, es la disciplina de extraer, limpiar, transformar, visualizar, transformar los datos, esta tarea es llevada a cabo gracias a los enfoques que lo conforman, en la Figura 3.2.1 se muestran sus tres elementos más importantes, así como las áreas donde convergen.



Figura 3.2.1 : Diagrama Venn para ciencia de datos (Conway, 2010)

La programación resulta esencial para el manejo de comandos digitales y de esta forma realizar operaciones y algoritmos, a los datos obtenidos a través de determinados sistemas; en el ámbito petrolero estos sistemas están conformados y son variados dependiendo de la etapa de la cadena de valor que se esté tratando o estudiando, por ejemplo, en el *upstream* se encuentra la perforación, de donde a través de las mediciones tomadas en sitio se obtiene la materia prima para estos análisis y que pueden retomarse del tema 2.4 Archivos **WITSML** y **LAS**; las matemáticas y estadística son el pilar para el análisis de estos extensos datos así como encontrar las relaciones existentes entre estos, el análisis exploratorio de datos o **EDA** tiene como fundamento el análisis utilizando la estadística y poder satisfacer los siguientes propósitos:

- Limpieza de datos.
- Mayor entendimiento de las variables.
- Determinar la relación que existe entre las variables.

Este análisis exploratorio tiene herramientas de visualización capaces de manejar una extensa cantidad de datos y de esta forma facilitar su comprensión e interpretación:

- *Pairplot*

En este se muestra la relación que existe entre las variables representado en una matriz gráfica como la Figura 3.2.2 anteriormente en la Figura 2.2.3 se mostró el perfil de temperatura del pozo **F10** que fue obtenido de esta matriz de gráficos, en la diagonal principal de esta se muestra el histograma del parámetro en cuestión.

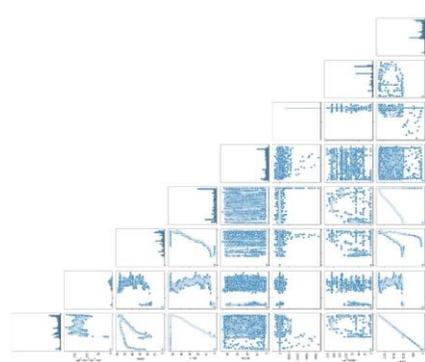


Figura 3.2.2 : Pairplot de registros de perforación

- Gráfico de caja y de violín

Estos gráficos muestran las principales propiedades estadísticas de los datos, los cuartiles, la mediana, los *outliers*, así como los *whiskers* inferiores y superiores que son los valores mínimos y máximos respectivamente, este gráfico se muestra del lado izquierdo de la Figura 3.2.3, el gráfico de violín es homólogo, con la distinción de que en este es posible ver si los datos son multimodales mediante un *kernel* de densidad que le agrega la librería de visualización de Python, *seaborn*, adicionalmente se muestra el histograma como referencia visual.

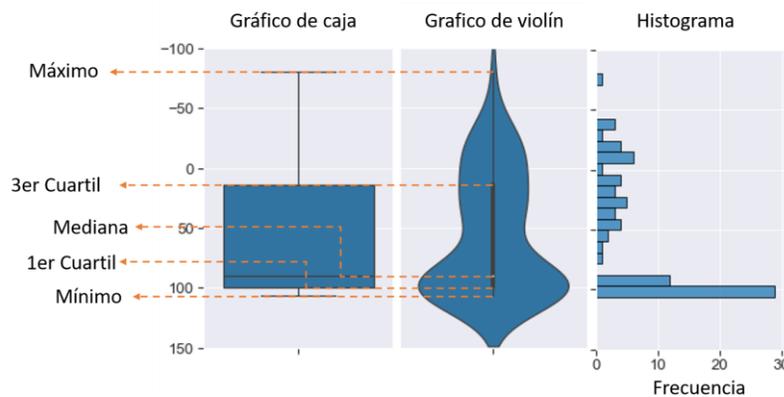


Figura 3.2.3 : Relación de gráficos de violín y caja

Para ejemplificar la utilidad de estas dos representaciones gráficas de los datos, se propone analizar las siguientes figuras.

En la Figura 3.2.4 se muestra una relación obtenida a partir del análisis de una corrida de perforación, donde se encuentra en colores las secciones de perforación del pozo **F10**, en azul los registros o mediciones realizadas en la sección más profunda que es de 8 1/2", por lo que el peso sobre el gancho o **HKLD** refleja un valor superior al color morado que es una sección más somera, siendo esta de 17 1/2", y por ende una sección donde el peso de la sarta es menor.

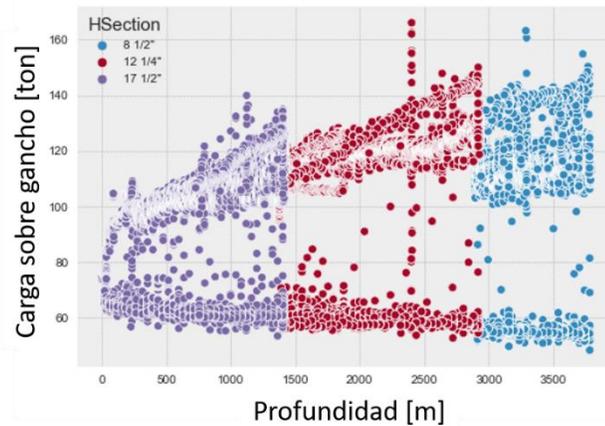


Figura 3.2.4 : Relación Peso-Profundidad y secciones

Por otra parte, si se emplea el gráfico de violín para el análisis de un parámetro de perforación, la Figura 3.2.5 muestra el torque o **TQA** ejercido a la sarta de la sección de perforación de 12 1/4" del pozo **F5**, en donde se observa una comparación con los datos reportados en el *Final Well Report* sobre el resumen de parámetros; se tiene un torque mínimo y máximo que compaginan con los obtenidos a partir del reporte; mientras que el promedio muestra una discrepancia leve, como también puede comprobarse con el histograma.

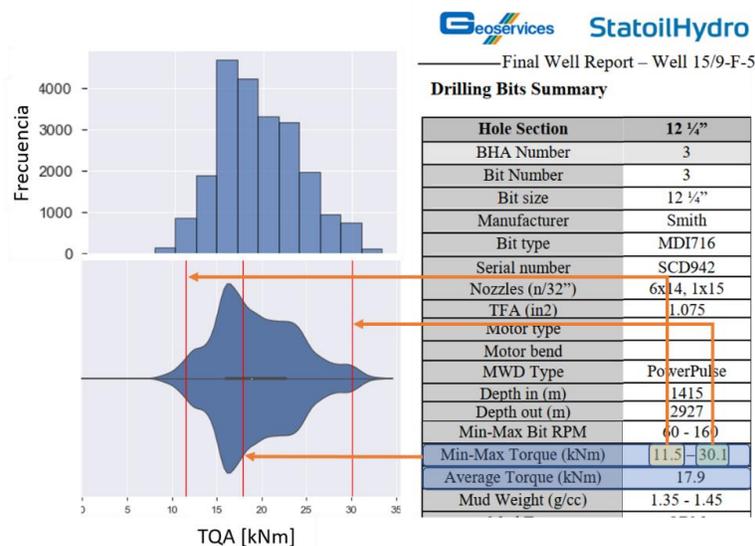


Figura 3.2.5 : Gráfico de violín de TQA para pozo F5

A lo largo de estas definiciones, y ejemplos de aplicación se ha mostrado la relevancia del conocimiento especializado, el último rubro de la Figura 3.2.1, siendo este la rama en donde se aplicarán las herramientas y metodologías antes mencionadas.

El conocimiento especializado o *Domain expertise* como se le conoce comúnmente, es el conocimiento *a priori* o previamente adquirido de alguna rama de conocimiento en especial, en la cual se están aplicando estas técnicas de análisis, en la Ingeniería Petrolera da pie al *Petroleum Data Analytics* o **PDA** por sus siglas en inglés, esta utiliza los datos recabados como pieza fundamental para el análisis, la creación de flujos de trabajo y toma de decisiones, como se muestra en la Figura 3.2.6, de esta, se puede interpretar que a partir de los datos obtenidos de algún fenómeno físico que se quiere modelar, a través de algoritmos de **ML** es posible obtener un modelo a partir de estos, el cual puede entrar en conflicto con la resolución de problemas de ingeniería a través de modelos físicos y matemáticos.

El futuro de las operaciones de perforación puede ser resumido en una continua implementación de “data driven models” para optimizar y disminuir la incertidumbre de los ambientes perforados.

(Ikram Noshi & Jacob Schubert , 2019)

Esta vertiente se enfoca en los datos de la siguiente forma:

- Los datos no se disponen para nuestro entendimiento de la física.
- La física es modelada utilizando los datos disponibles.

A partir de este enfoque se busca retirar presunciones, simplificaciones y nociones preconcebidas para modelar algún comportamiento.

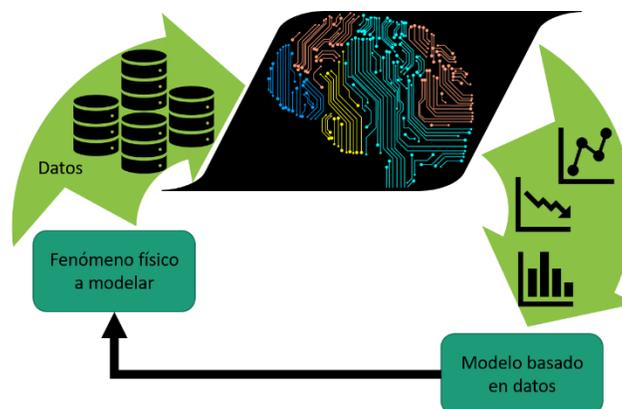


Figura 3.2.6 : PDA (Mohaghegh, 2020. Modificado)

Un modelo es la representación cualitativa o cuantitativa de un sistema o fenómeno, y está conformado por:

- Parámetros
Propiedades o atribuciones que permanecen constantes durante el análisis.
- Variables
Atribuciones del sistema que cambian durante el estudio.
- Relaciones funcionales
Procesos físicos o relaciones que se llevan a cabo entre las variables y sus repercusiones por los parámetros.

Dada estas consideraciones, la precisión de estos modelos físicos está dada por la exactitud de los datos iniciales, el planteamiento de las ecuaciones que rigen el fenómeno y su aproximación, analítica o numérica.

Por lo que el **PDA** aporta un replanteamiento completo de la formulación y solución a los problemas de ingeniería, de donde se podría definir dos enfoques de pensamiento:

- Deductivo
 Aquel que parte de una teoría capaz de representar determinado fenómeno.

- Inductivo
 A partir de las consecuencias de determinado fenómeno se busca el planteamiento de una conjetura.

Pudiéndose expresar de mejor forma comparando ambas vertientes como se muestra en la Tabla 5.

Tabla 5 : Vertientes de razonamiento

Pensamiento deductivo	Pensamiento inductivo
Derivación automática de un modelo explicativo	Acumulación de observaciones
De lo general a lo particular	De lo particular a lo general
Concepto de validez	Concepto de probabilidad
Las conclusiones se siguen de las premisas necesarias	La conclusión es incierta y discutible

El pensamiento deductivo representa la solución a los problemas de ingeniería mediante una formulación matemática y su posterior solución, mientras que el inductivo plantea una solución que puede obtenida a partir de la aplicación de modelos basados en datos, y por ende una solución utilizando herramientas como las que ofrece el **ML**.

3.3 Árbol de decisión

También denominado algoritmo **CART**, por sus siglas en inglés *Classification And Regression Trees*, término acuñado por Leo Breiman en 1984, y cuyo fundamento es la creación de un árbol binario con las que se busca repartir las observaciones o datos dados en función de sus atributos y predecir el valor de la variable deseada, existiendo como ya se mencionó, problemas de clasificación y regresión.

Ventajas:

- Fácil de interpretar.
- No se requiere estandarización, establecer el valor de los datos en un intervalo comúnmente [0,1].
- No influencia significativa por parte de *outliers*.

Desventajas:

- Sensibles a datos de entrenamiento desbalanceados
- No extrapolan

Para que un determinado problema de clasificación sea satisfactorio, es necesario establecer una serie de condiciones que particione los datos en las clases correspondientes, en la Figura 3.3.1 se muestra un ejemplo, el diagrama de partición para determinar la clasificación de macro-actividades o el *rig state*, pudiendo ser "A" o "B" de una determinada observación, a partir de tres características, el **ROP**, **TFLO** y **BPOS**, acrónimos ya discutidos en el subtema 2.2 Adquisición de datos.

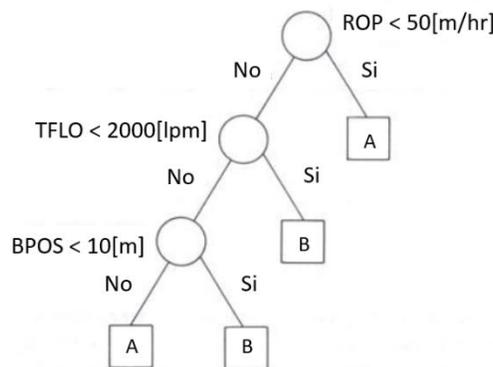


Figura 3.3.1 : Ejemplo Árbol de decisión (Breiman,1984. Modificado)

Así mismo, se pueden establecer una serie de reglas para la clasificación utilizando las condicionales:

<i>If</i>	ROP < 50[m/hr]	<i>Then</i>	A
<i>If</i>	ROP >= 50[m/hr]	<i>And</i>	TFLO < 2000[lpm]
		<i>Then</i>	B
<i>If</i>	ROP >= 50[m/hr]	<i>And</i>	TFLO >= 2000[lpm]
		<i>And</i>	BPOS < 10[m]
		<i>Then</i>	B
<i>If</i>	ROP >= 50[m/hr]	<i>And</i>	TFLO >= 2000[lpm]
		<i>And</i>	BPOS >= 10[m]
		<i>Then</i>	A

En recuadro azul, se muestran las condicionales, en verde los argumentos de estas condicionales y por último, en naranja las clases asignadas o las micro-actividades.

3.3.1 Entrenamiento de árbol

Como se vio primeramente, la construcción de un árbol consiste en particionar las observaciones dadas, por lo que se puede definir una función objetivo, para un problema de clasificación, esta función se le denomina como *Gini*, y se puede obtener mediante la Ecuación 1.

$$Gini(\text{particion}) = \sum_{i=1}^n (p_i \times (1 - p_i)) \quad (1)$$

Donde P_i es la probabilidad de que una observación sea de una clase i , por lo que el algoritmo buscará obtener la partición con mayor pureza, o con un menor índice *Gini*, y de esta forma se tiene una limitante en la generación de nodos de decisión, en cuanto el índice sea cero, las observaciones contenidas en ese nodo corresponderán a una sola clase.

La Figura 3.3.1.1 muestra el algoritmo antes mencionado, este se estructura en nodos de decisión, siendo estos las condicionales para la partición de los datos, subsecuente a estos nodos, el árbol aumenta de profundidad, pudiendo continuar en nodos o en hojas finales, donde la partición termina.

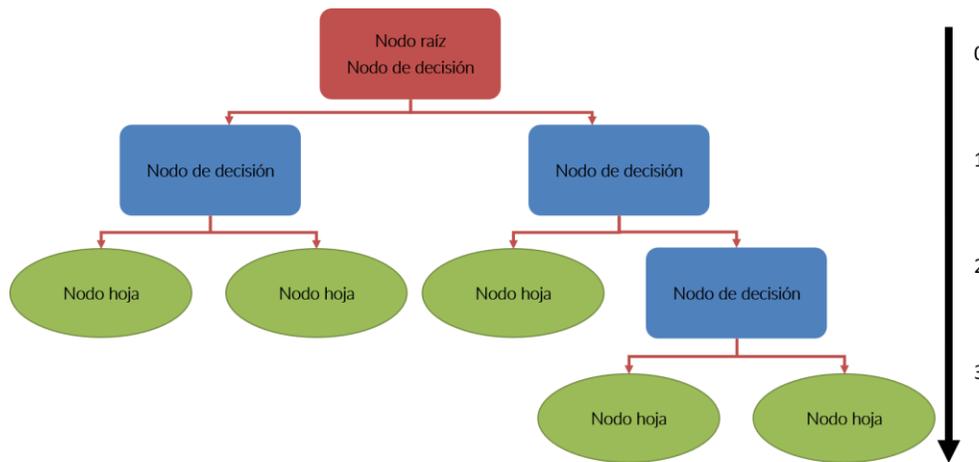
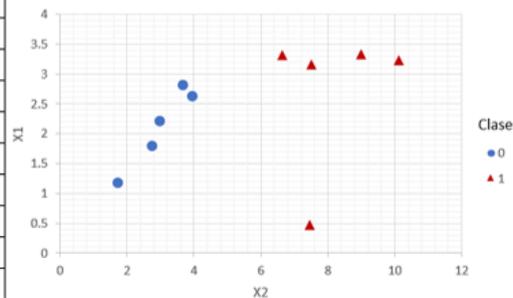


Figura 3.3.1.1 : Elementos del Árbol de decisión (Müller, 2017)

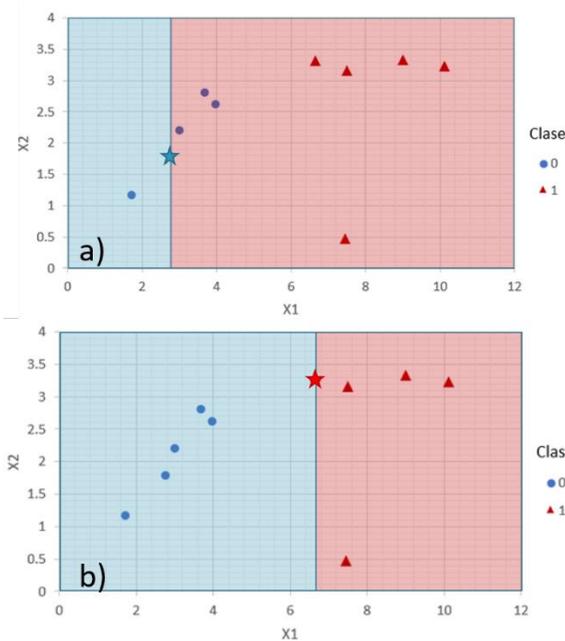
Para visualizar de mejor forma este procedimiento, se utilizará el siguiente ejemplo:

X1	X2	Y
2.77124472	1.7847839	0
1.72857131	1.1697614	0
3.67831985	2.8128136	0
3.96104336	2.6199503	0
2.99920892	2.2090142	0
7.49754587	3.1629535	1
9.00220326	3.3390472	1
7.44454233	0.4766834	1
10.124939	3.234551	1
6.64228735	3.3199838	1



Se tiene la distribución de datos mostrada, a estos se les busca clasificar con respecto a la clase "Y", teniendo atributos "X1" y "X2".

Se buscará particionar los datos como se muestra más adelante y clasificarlos mediante el criterio "Gini".



En el inciso (a) se muestra una primera partición, esta se encuentra en "X1 = 2.7712447" una de las coordenadas de los puntos mostrado con símbolo de estrella por lo que la condición de partición queda de la siguiente forma:

$$\begin{aligned} & \text{If "X1 < 2.7712447" Then Clase 0} \\ & \text{If "X1 \geq 2.7712447" Then Clase 1} \end{aligned}$$

Por lo que analizando la precisión de esta condición y con base en las particiones del (a):

Recuadro Azul, Clase 0:

$$\begin{aligned} Y = 0 & \rightarrow \frac{1}{1} = 1 \\ Y = 1 & \rightarrow \frac{0}{1} = 0 \end{aligned}$$

Recuadro Rojo, Clase 1:

$$\begin{aligned} Y = 0 & \rightarrow \frac{4}{9} = 0.444 \\ Y = 1 & \rightarrow \frac{5}{9} = 0.555 \end{aligned}$$

Por lo que el índice *Gini*:

$$Gini(X1 = 2.7712) = (1x(1 - 1)) + (0x(1 - 0)) + (.44x(1 - .44)) + (.55x(1 - .55)) = 0.4938$$

Al realizar la partición del inciso (b) se tienen las siguientes condiciones:

$$\begin{aligned} & \text{If "X1 < 6.6422" Then Clase 0} \\ & \text{If "X1 \geq 6.6422" Then Clase 1} \end{aligned}$$

Por lo que analizando la precisión de esta condición y con base en la partición del:

Recuadro Azul, Clase 0:

$$\begin{aligned} Y = 0 & \rightarrow \frac{5}{5} = 1 \\ Y = 1 & \rightarrow \frac{0}{5} = 0 \end{aligned}$$

Recuadro Rojo, Clase 1:

$$\begin{aligned} Y = 0 & \rightarrow \frac{0}{5} = 0 \\ Y = 1 & \rightarrow \frac{5}{5} = 1 \end{aligned}$$

Obteniendo un índice *Gini* de:

$$Gini(X1 = 6.6422) = (1x(1 - 1)) + (0x(1 - 0)) + (1x(1 - 1)) + (0x(1 - 0)) = 0.0$$

Este valor del índice indica una partición perfecta entre las clases, por lo que el algoritmo finaliza en este punto, ya que no existen elementos en los datos los cuales se encuentren sin asignar.

3.4 Métodos de Ensamble

Estos métodos, se fundamentan en conjuntar diversos modelos de **ML**, su estructuración depende del arreglo de los modelos, por lo que se destacan:

- *Boosting*

El ajuste de múltiples modelos sencillos llamados *weak learners* mostrados en el recuadro naranja de la Figura 3.4.1 para que, aplicándolos de una forma secuencial, con cada modelo siguiente existe una mejora a comparación del que le antecede, los modelos estandarte de esta clasificación son los métodos de *AdaBoost* y *GradientBoosting*.

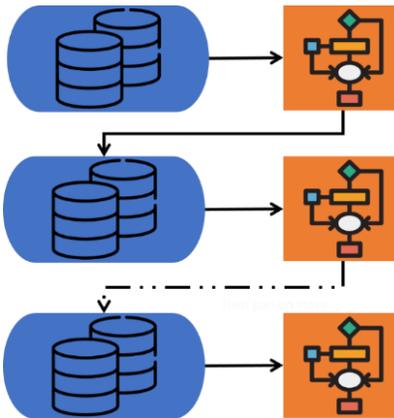


Figura 3.4.1 : Boosting (QuantDare, 2016)

- *Bagging*

También denominado *Bootstrap Aggregation*, ya que se ajustan múltiples modelos a un subconjunto de los datos de entrenamiento, para obtener una predicción compuesta por todos estos modelos, en la Figura 3.4.2, se muestra este proceso de aprendizaje paralelo entre los modelos, el algoritmo que mejor lo representa es el *Random Forest* o **RF**, por sus iniciales en inglés.

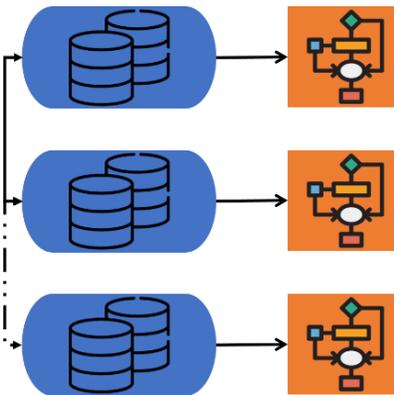


Figura 3.4.2 : Bagging (QuantDare, 2016)

3.4.1 Random Forest

Este algoritmo es uno de los más populares y más poderosos del repertorio de **ML**, ya que es un conjunto de árboles de decisión individuales, recordando que este modelo forma parte de la clasificación de *Bagging* para los modelos de ensemble; donde cada uno de ellos es entrenado por observaciones particulares y de esta forma, la predicción será el resultado de agregar las predicciones particulares de cada árbol.

De esta forma, pueden solucionarse problemas de regresión, donde la predicción resulta en un promedio del resultado de cada árbol de decisión; mientras que los de clasificación resultan de un contador, siendo la clase más votada, la utilizada en dicha predicción.

En la Figura 3.4.1.1, se muestra el procedimiento de construcción de un modelo de *Random Forest* o **RF** y su respectiva predicción.

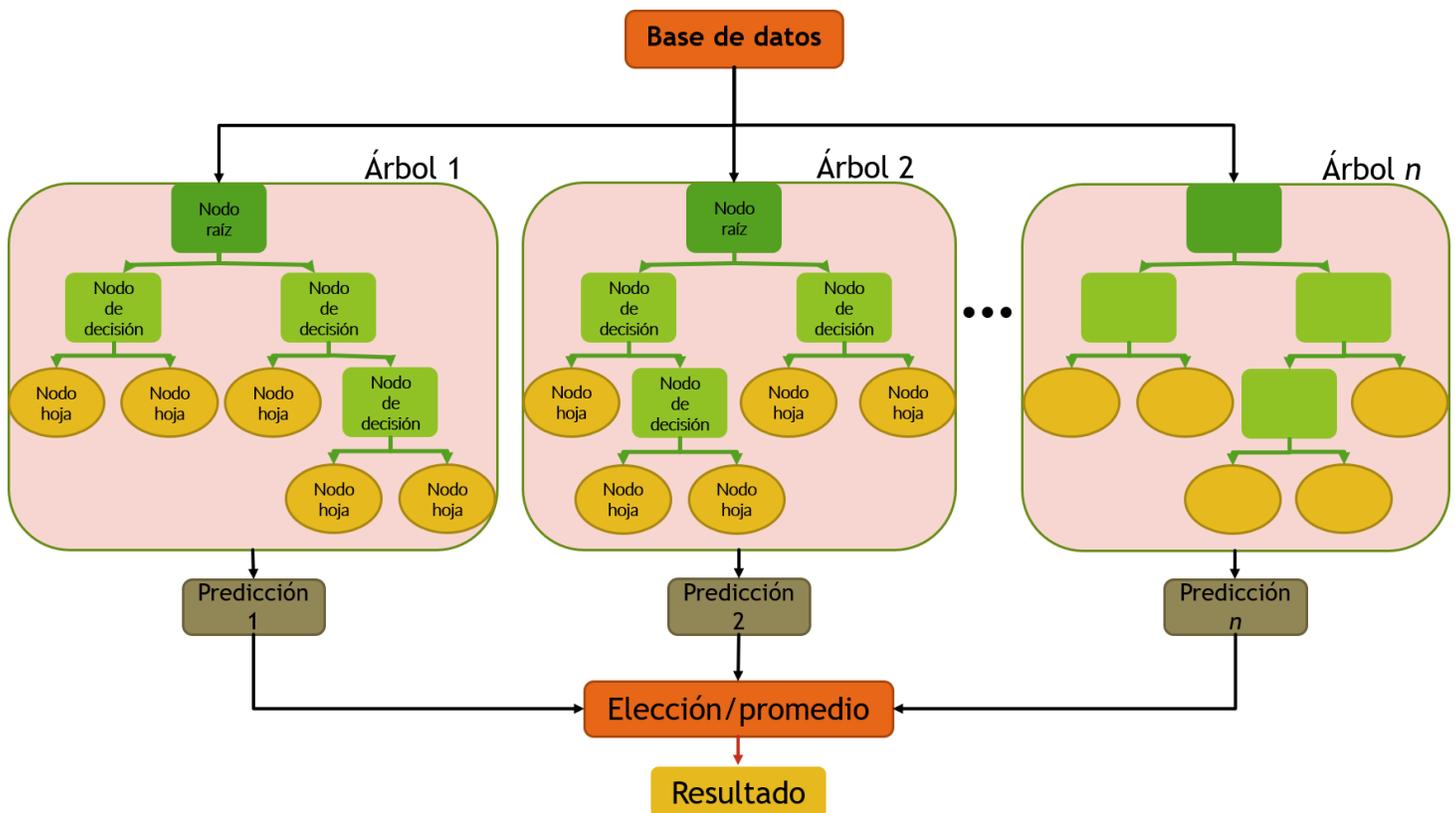


Figura 3.4.1.1 : Elementos del Random Forest (Jagannath, 2017)

Capítulo 4. Stick & Slip

4.1 Mecanismos de vibración

En el proceso de perforación, la barrena falla la roca definiendo el diámetro del pozo, mientras que, entre este y la sarta, se encuentra un espacio denominado anular, en el cual el lodo de perforación desplaza los recortes hasta la superficie; a lo largo de la longitud del **BHA** se encuentran un número limitado de estabilizadores, enfocados en centrar la sarta en el medio del pozo para de esta forma generar una eficiente transferencia del peso a la barrena.

De esta forma, al existir una distribución inadecuada de los esfuerzos, condiciones operativas que permitan el libre movimiento en el pozo, existirán vibraciones que pueden visualizarse en pozos horizontales y verticales como se muestra en la Figura 4.1.1

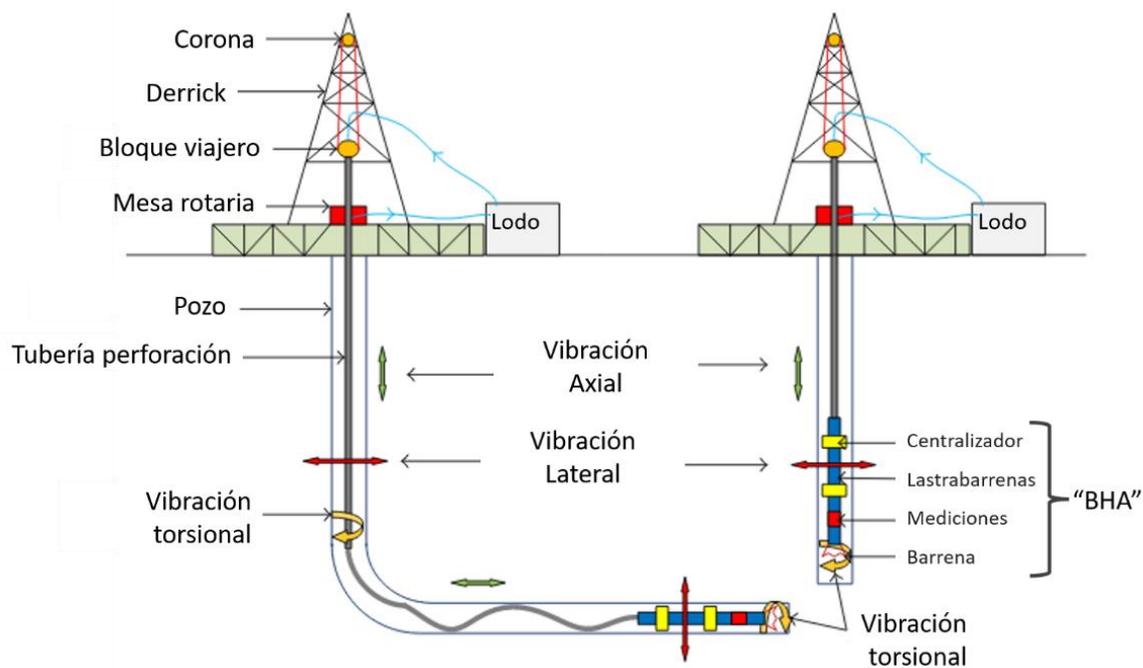


Figura 4.1.1 : Vibraciones de la sarta (Guangjian & Ping, 2016)

Dependiendo de la naturaleza de estas vibraciones, su frecuencia y su magnitud estas pueden representar un problema operativo y a la infraestructura, ya que, la composición de la sarta de perforación consiste en esencia de acero, este material posee propiedades elásticas que permiten que, a determinada carga, el material regrese a sus condiciones originales o se deforme si se excede, por lo que es de suma importancia no exceder este límite elástico.

Definiendo más a fondo las vibraciones, estas pueden presentarse de la siguiente forma:

- Vibraciones axiales
Estas ocasionan el rebote de la barrena en el fondo del pozo, y posterior daño en los cortadores de los mismo, así como fallas en el **BHA**; mostrándose en la Figura 4.1.2 inciso (a).
- Vibraciones laterales
Estas vibraciones resultan ser las más relevantes en cuanto a daños severos al **BHA** ya que ocasionan pandeos, choques y *whirls* a niveles de barrena y del ensamble; se muestra en la Figura 4.1.2 inciso (b) y (c), respectivamente.
- Vibraciones torsionales
Esta vibración es el enfoque del presente trabajo, por lo que se detallará más adelante, sin embargo, puede observarse en la Figura 4.1.2 inciso (d).

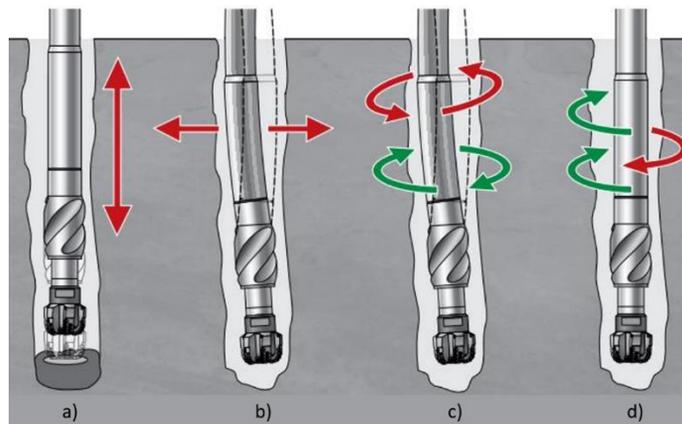


Figura 4.1.2 : Vibraciones del BHA (SLB DrillingQuickReference, 2013)

Las vibraciones torsionales, *Stick & Slip* o **S&S** por sus siglas, ocurren gracias a que en la superficie el sistema de rotación proporciona este a la sarta de perforación de una forma constante, mientras que en el fondo del pozo esta oscila, debido a la rigidez del material con el que se encuentra elaborada la sarta, provocando una deficiente transmisión del movimiento y torque de superficie a fondo; así mismo, por las interacciones entre la barrena y la formación, en estas pueden existir instantes en el que la fricción presente evite la revolución o *stick* en inglés; para luego, en la subsecuente, dicha sarta deberá rotar de manera más acelerada y de esta forma compensar aquellas ya llevadas a cabo en la superficie o *slip*, como se muestra en la Figura 4.1.3.

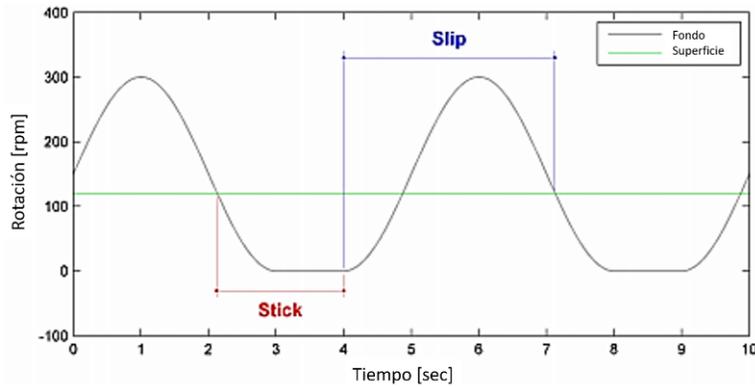


Figura 4.1.3 : S&S (Mayer, 2017)

Este fenómeno vibratorio se ve afectado y acentuado por los siguientes parámetros:

- Longitud y diámetro de sarta de perforación
- Condiciones superficiales del equipo de rotación
- Alto **WOB** y bajo **RPM** en fondo
- Barrenas **PDC** con bajo número de aletas
- Condiciones direccionales del pozo
- Litología

El primer parámetro mencionado determina la rigidez torsional de la sarta, ocasionando una tendencia al fenómeno, así como la frecuencia, una sarta pequeña y con gran diámetro no será proclive a generar este fenómeno torsional o **S&S**.

La medición de este parámetro se lleva a cabo internamente en un intervalo de 5 segundos y en el sistema de **MWD**, esta consiste en la Ecuación 2 que se muestra a continuación.

$$Stick_RT [rpm] = [RPM_{maxima} - RPM_{minima}]_{5[sec]} \quad (2)$$

De una forma gráfica puede representarse como la Figura 4.1.4, este registro se titula *Stick Real Time*, como se muestra adicionalmente en la Figura 2.2.4, correspondiente a los registros de perforación.

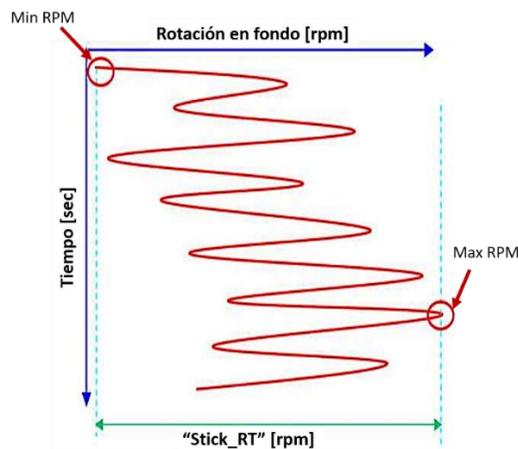


Figura 4.1.4 : Medición S&S (Green, 2005)

A partir de la metodología de análisis de Schlumberger contenida en el *Drilling and Measurement Technologies* para analizar el **S&S**, se establece un valor porcentual y una clasificación, como se muestra en la Tabla 6.

Tabla 6 : Clasificación por severidad

S&S [%]	Modalidad	Severidad	Límite para remediación	Severidad categórica
0 – 40	Sin	Nula / Baja	Sin	0
40 - 60	Vibración torsional	Media	Sin	1
60 -80			Recomendar mitigación	
80 – 100	S&S	Alta	Obligatorio mitigar	2
> 100	S&S	Severa	30 [min] Reporte formal	3

En la primer columna se muestra el porcentaje de severidad del **S&S**, este es una medida sobre la rotación del **BHA** en comparación con la superficie, es decir, el porcentaje adicional de rotación al que es sometida la sarta en un determinado momento; así mismo, como elemento de predicción utilizado más adelante en el presente trabajo y su forma de obtención se muestra en la Ecuación 3.

$$S\&S \text{ [%]} = \left(\frac{Stick_RT}{2 * RPM} \right) x 100 \quad (3)$$

Dependiendo el porcentaje de **S&S**, se establece que menor al 40% no se presenta fenómeno vibratorio o es muy pequeño, por lo que la severidad o el riesgo que representa para el **BHA** es nulo o bajo, mientras que entre 40-80 % se presentan eventos de riesgo medio, los cuales se recomienda mitigar; ente el porcentaje de 80-100% se establece la obligación por parte del equipo de perforación, aplicar medidas de corrección ya que estos representan eventos de riesgo alto; por último, superior al 100% se presentan eventos severos, los cuales tendrán repercusiones en la integridad del **BHA**.

Para realizar un modelo de predicción, es de suma importancia establecer una variable categórica siendo esta una variable numérica que, represente los riesgos o severidades, por lo que se propuso una severidad categórica, la cual consiste en los primeros tres numerales, como se muestra en la última columna de la Tabla 6.

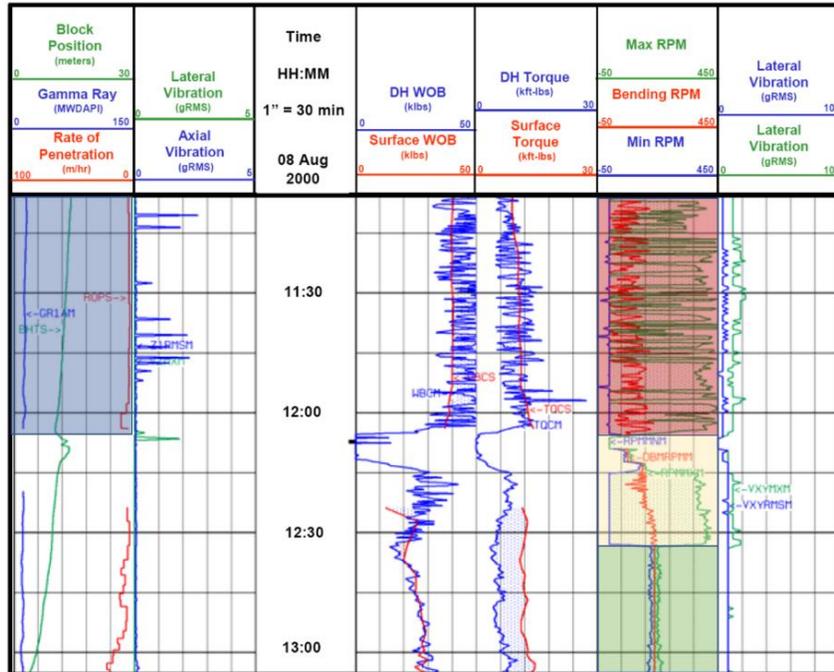


Figura 4.1.5 : Evento de S&S (Mayer, 2017)

En la Figura 4.1.5, se muestra en rojo la ocurrencia severa de **S&S**, que, recordando la definición antes mencionada, es la diferencia entre los valores máximos (+/- 300[rpm]) y los mínimos (+/- 0[rpm]) por lo que el área entre la línea verde y azul indica eventos severos, mientras que, en el recuadro amarillo, se realizó una actividad de remediación que se vio reflejada en la eliminación del evento de **S&S** en el recuadro verde conforme avanzó el tiempo.

4.2 Consecuencias y Remediación

De existir un manejo deficiente en los parámetros de perforación, y estos generen eventos constantes y altos de **S&S** se pueden presentar complicaciones como las que se mencionan a continuación:

- Daños a la barrena
Pudiéndose presentar desde cortadores rotos (a) a desprendimientos (b) por la aceleración tan marcada en el *slip*, también, de presentarse más fenómenos vibratorios, daños severos a esta (c).
- Daños a las herramientas direccionales
Los *pads* contenidos en los ensambles direccionales son móviles y pueden desgastarse (d) hasta fracturarse y quedar inservibles (f).
- Efectos de sobre torque
Las cajas y pines contenidos en toda la sarta, pueden ser sometidos a un esfuerzo mayor al de diseño, lo que puede ocasionar problemas en las conexiones (e).

Estas corresponden a las consecuencias hacia la infraestructura, Figura 4.2.1, sin embargo, también se puede tener bajo desempeño en el **ROP**, efectos de arrastre e interferencias en la telemetría.

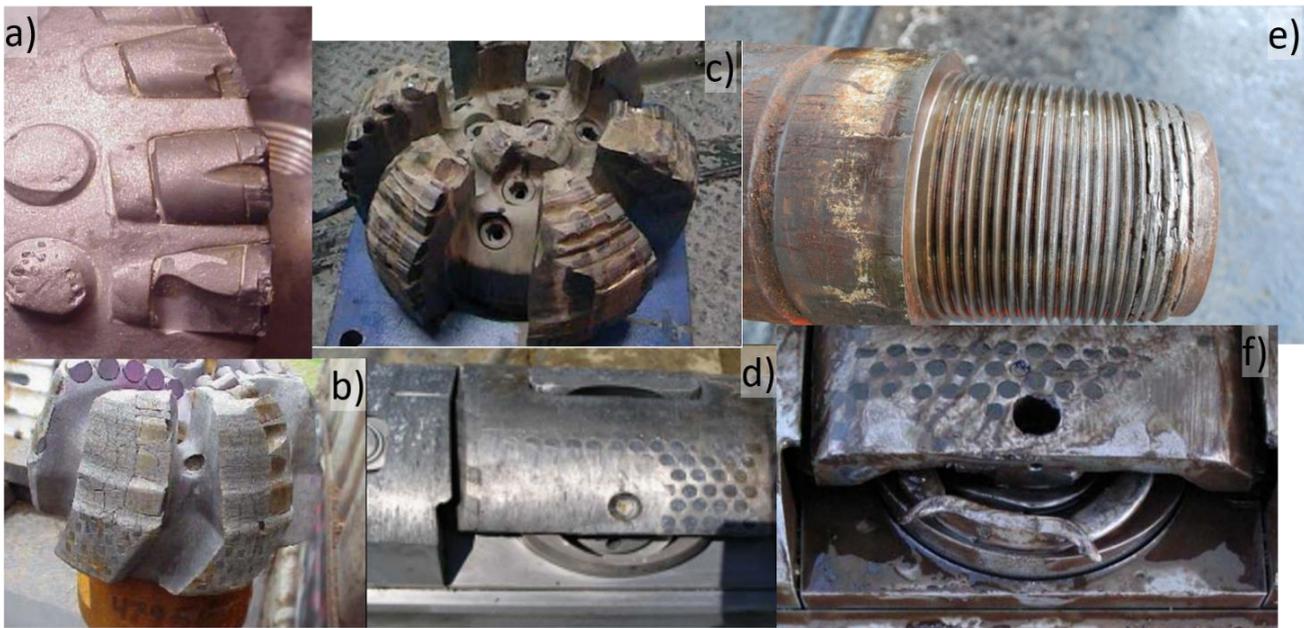


Figura 4.2.1 : Daños por severidades altas

Estas repercusiones, pueden evitarse gracias a un control constante de los parámetros de la perforación, el flujo de trabajo de la Figura 4.2.2, muestra el recorrido de la medición en fondo, la transmisión del síntoma a través del **MWD**, la visualización por parte del equipo de perforación en la cabina, y posteriormente su conveniente remediación.



Figura 4.2.2 : Remoción de vibraciones

La metodología de remediación para estas vibraciones radica en un manejo óptimo de las revoluciones y peso sobre la barrena por lo que, en campo, se sugiere seguir el flujo de trabajo presentado en la Figura 4.2.3

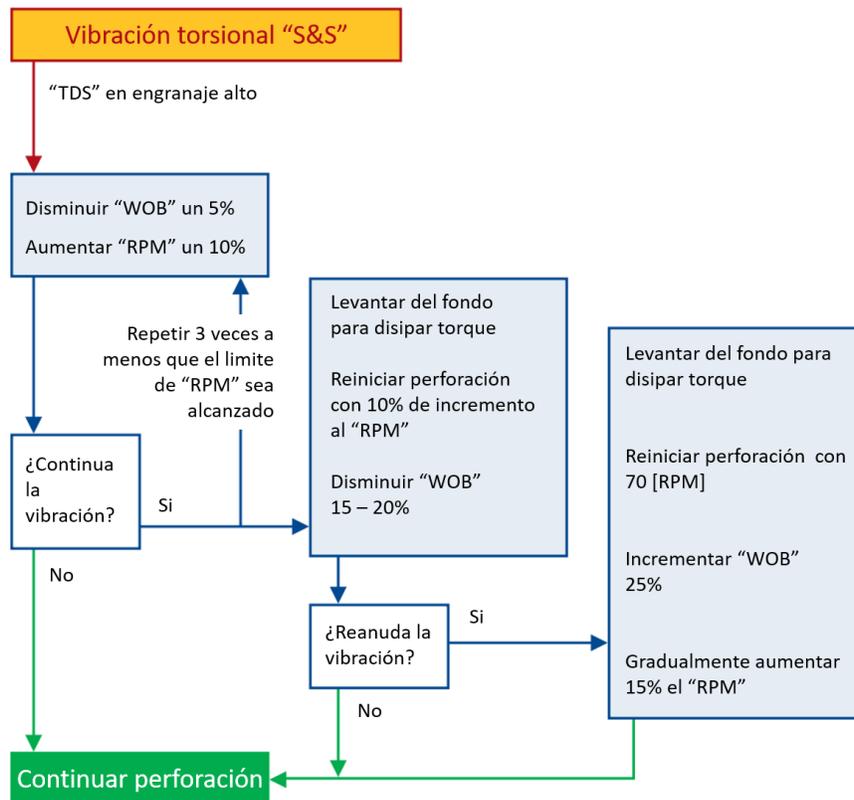


Figura 4.2.3 : Diagrama de flujo para remediación (SLB Drilling Quick Reference, 2013. Modificado)

Como se aprecia en esta Figura 4.2.3, la remediación consiste en minimizar el peso sobre la barrena, y minimizar la rotación acelerada de la sarta, aumentando las rotaciones en superficie para que de esta forma se disminuya los efectos de la aceleración en el *slip*; Si esto no funciona, comienzan las afectaciones al **NPT**, ya que la siguiente estrategia de mitigación conlleva una elevación de la sarta en fondo para de esta forma eliminar la fricción entre la barrena y la formación.

La solución a este fenómeno vibratorio recae en el correcto manejo por parte del perforador y la edición de los parámetros de perforación, sin embargo, es importante mencionar, que existen condiciones direccionales y fenómenos vibratorios que suceden simultáneamente que dificultarán su remediación.

También se pueden tomar en consideración la siguiente Tabla 7, para prevenir estos fenómenos torsionales.

Tabla 7 : Prevención de S&S (Green, 2005)

Componente o parámetro	Prevención
Barrena	Mayor número de aleta y cortadores pequeños
BHA	Centralizadores y dispositivos <i>roller reamers</i>
Tortuosidad o angulo	Angulo bajo y tortuosidad baja
Lodo de perforación	Lodos base aceite para mejor lubricidad
TDS	Sistema en engranaje alto y <i>soft-torque</i>

4.3 Relevancia operativa

Se decidió analizar la relevancia de estos eventos de vibración a partir de los documentos denominados *End of well report* o por sus siglas en inglés **EOWR**, los *Final well report* o **FWR**, en estos se encuentra una breve descripción del desempeño, actividades y pormenores en el proceso de perforación de determinado pozo, así mismo, contienen evaluaciones de desempeño, análisis de *survey's*, entre otros; también, se analizó el *Drilling Program* o programa de perforación, estos documentos se muestran en la Figura 4.3.1

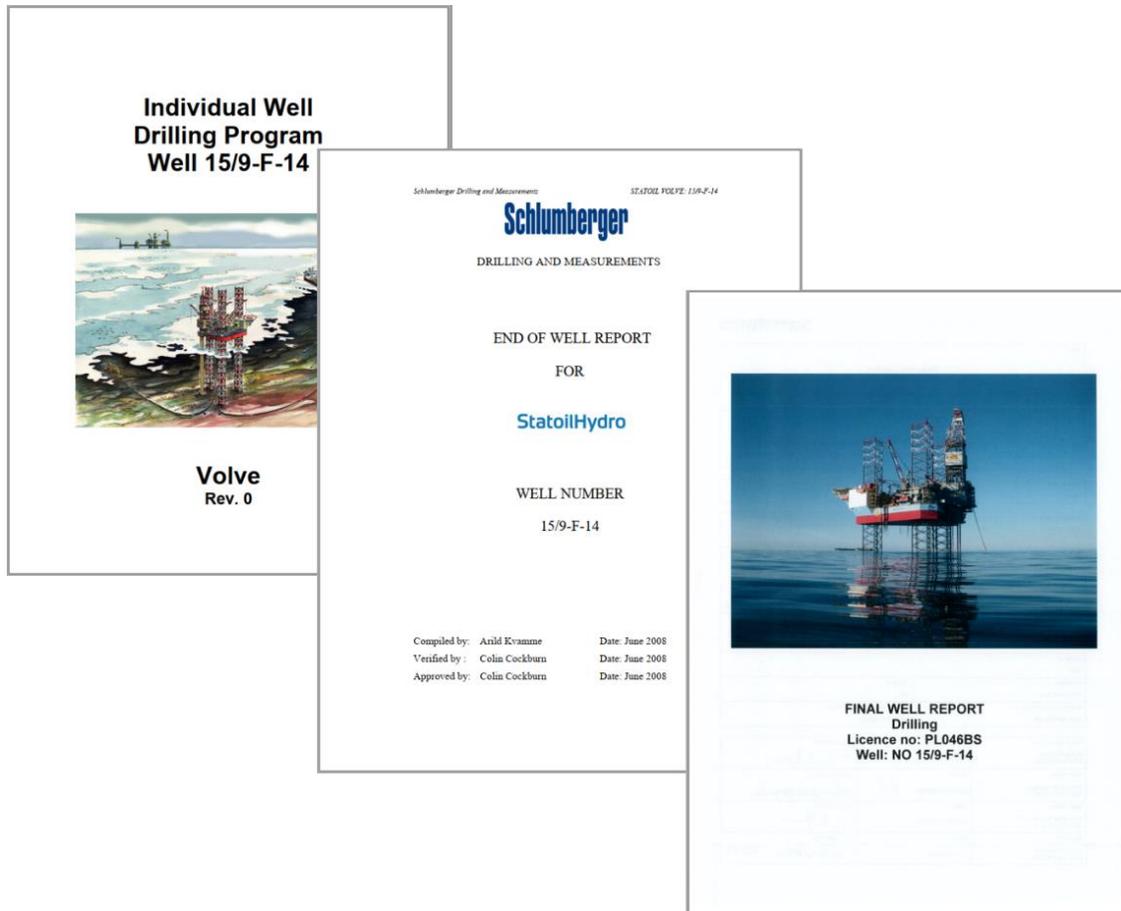


Figura 4.3.1 : Documentos entregables en la perforación pozo F4

A partir de la descripción de eventos de **S&S** contenidos en estos se analizaron los pozos con los que se construirán los modelos de predicción, los pozos **F4** terminado el 15 de febrero de 2008, **F5** el 30 de julio de 2008, **F10** el 16 de junio de 2009 y **F14** el 15 de mayo de 2008, ya que estos no son pozos multilaterales

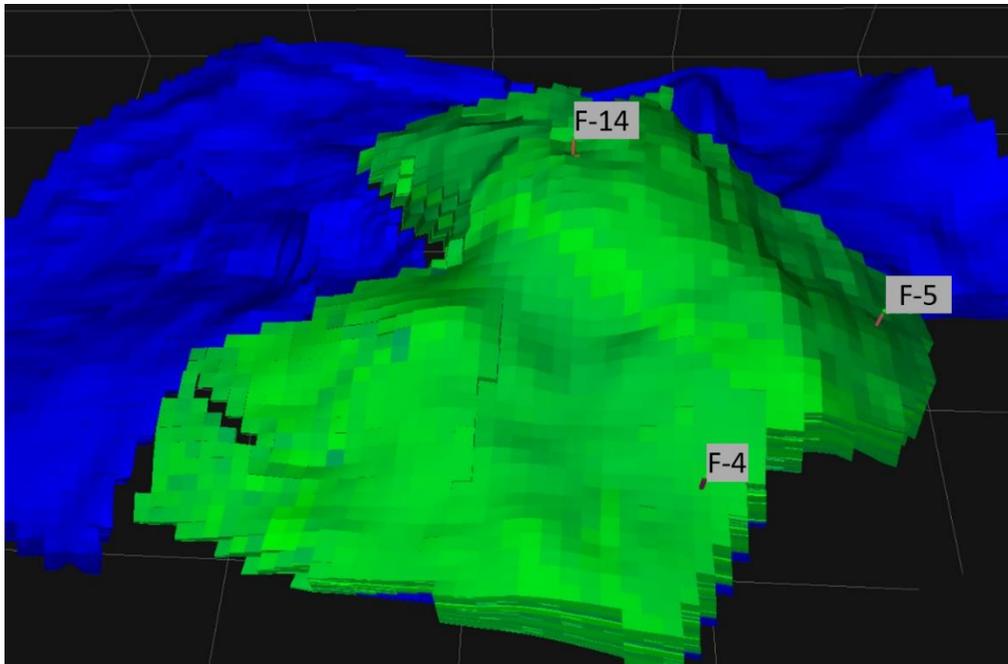


Figura 4.3.2 : Pozos de análisis

En la Figura 4.3.2 se muestra el modelo de simulación con los pozos previamente mencionados, sin embargo, el pozo **F10** al ser exploratorio, este no se encuentra en el modelo de simulación.

En el análisis de estos pozos, primero se muestran los registros de **MWD** para las secciones más afectadas por este fenómeno vibratorio, y en estas figuras, ser posible apreciar recuadros, los cuales tienen un significado particular, en el de color rojo, se mostraran los eventos de **S&S** severos; en recuadro amarillo, las posibles causas de estos fenómenos, cabiendo destacar que estas se concentran en la interacción **SWOB** y **RPM** para que; en el recuadro verde se muestren las posibles remediaciones que se llevaron a cabo; en recuadro azul se encuentran menciones relevantes a lo largo del registro de perforación; en el recuadro gris se muestra la columna geológica por las cuales se está perforando, remarcando que la Tabla 2 contiene la descripción de cada formación.

En la última sección de cada análisis, se mostrarán los registros de *survey* o trayectoria, para terminar, asociando eventos de **S&S** a algunos parámetros direccionales.

4.3.1 Análisis de Pozo F14

En el reporte de perforación correspondiente a la etapa de barrena de 12 ¼”, se menciona que en el intervalo de 2,281 – 2,662 mD, o metros desarrollados, se sufrieron eventos leves de **S&S**, y severos al superar los 2,662 [m], estos pueden observarse en los recuadros de la Figura 4.3.1.1, la cual muestra, el registro **MWD**, así mismo, es posible notar que en esta corrida solo se aprecia un evento que pudo haber causado el **S&S**, siendo este el incremento acelerado de peso, y un par de horas después se aprecia el intento de mitigación de este.

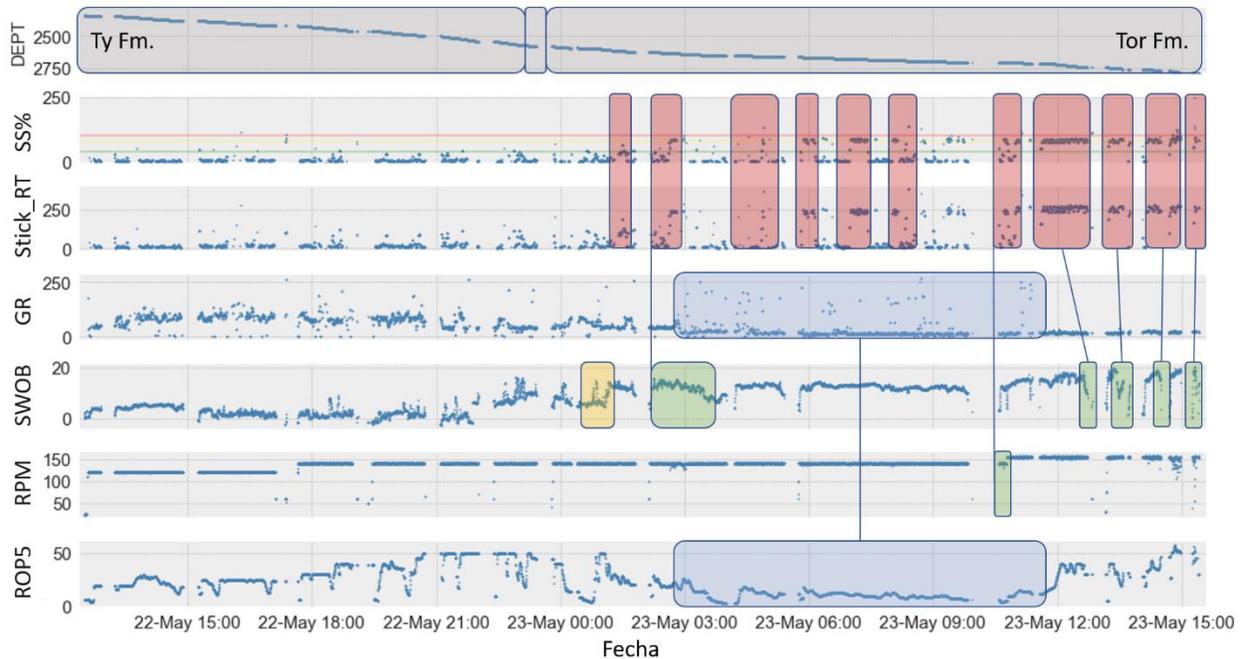


Figura 4.3.1.1 : Barrena 12 ¼” pozo F14

A partir de este análisis también resulta muy importante mencionar, que, en la Figura 4.2.3, se establece que, al existir eventos continuos, se sugiere levantar la barrena del fondo, esto puede interpretarse como una disminución pronunciada en el peso sobre la barrena o **SWOB**, existiendo evidencias a partir del día 23 de mayo a las 12:00 del registro.

A lo largo de esta sección de perforación, se cruzaron los horizontes de *Ty*, *Ekofisk* y *Tor*, siendo este último, la formación que posee los eventos más severos y relevantes, recordando que se trata de roca caliza con intercalaciones arcillosas, estas intercalaciones se muestran de forma clara el recuadro azul correspondiente al carril de **GR**, así mismo, una respuesta característica de las areniscas, **GR** bajo le corresponden los eventos más frecuentes y altos de **S&S**; una vez finalizada esta sección, se procedió a utilizar la barrena de 8 ½” para la siguiente sección, la cual se muestra en la Figura 4.3.1.2, esta sección comienza con severos eventos de **S&S**, los cuales coinciden con la formación *Tor*, siendo esta la que ocasionaba eventos constantes en la sección anterior; la formación *Blodoeks* también muestra eventos severos pudiendo ser ocasionados por una excesiva carga en el fondo, para luego hacer intentos de mitigación controlando el **SWOB** y un aumento de **RPM**; la formación *Draupne* muestra un incremento en las ocurrencias, siendo esta, carbonatos con lutita arcillosa, al presentarse eventos severos se intentó mitigarlos disminuyendo el WOB a la par de realizar varias remociones de fondo.

Desafortunadamente el registro de **GR** no aporta mucha interpretación, este se encuentra con mediciones inferiores a 100[°API]; en la formación de *Smith Bank*, ocurre un evento repentino severo, pudiendo apreciar un aumento muy pronunciado de **SWOB**, esto, realizado para compensar la caída del **ROP** mostrada en el recuadro azul, evento seguido, un intento de mitigación, el cual no afecta de manera significativa el **ROP**, mostrando que no necesariamente los eventos de mitigación van a repercutir negativamente en el desempeño de perforación.

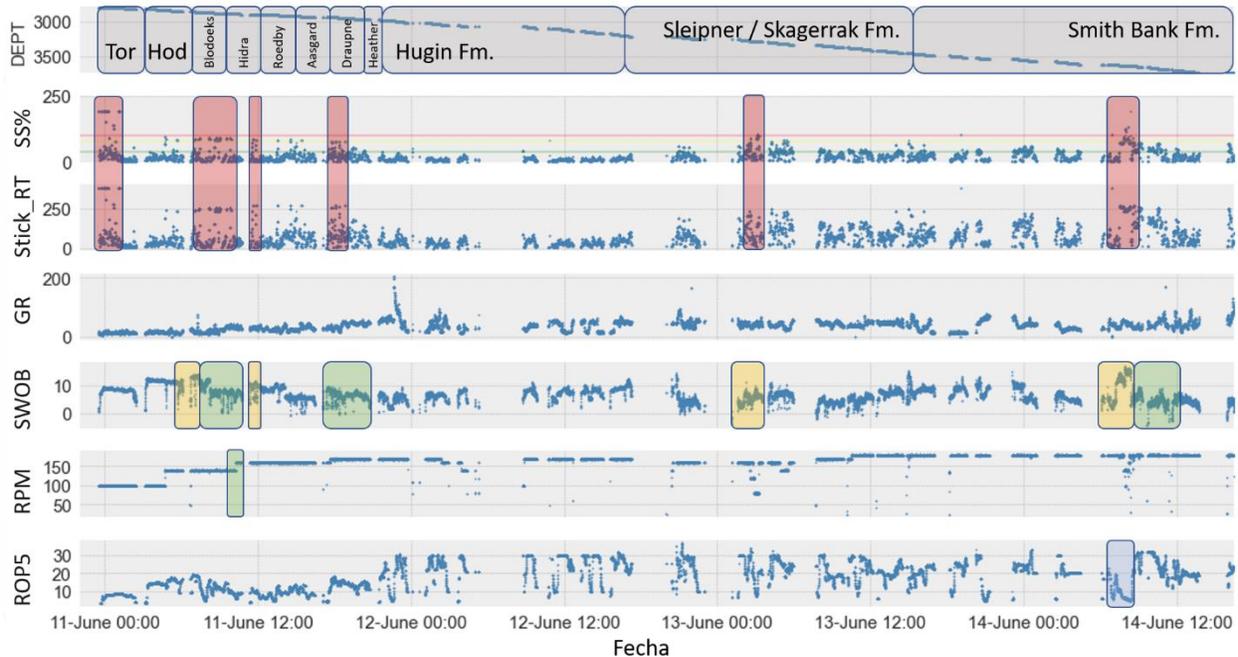


Figura 4.3.1.2 : Barrena 8 ½" pozo F14

Los registros de trayectoria se muestran en la Figura 4.3.1.3, en el primer carril la sección vertical en función de la profundidad verdadera, este, siendo un indicativo de distancia del punto de *survey* al eje vertical del pozo; en el segundo, los grados de inclinación; tercero, el *dog leg severity* y cuarto, el azimut de la barrena, recordando que estos parámetros ya fueron definidos en el desarrollo de la sección 2.2 Adquisición de datos, en este registro es posible apreciar que en la primer sección analizada, a los eventos severos de la formación *Tor*, le corresponde un **DLS** de grados intermedios, así como una inclinación incremental, la cual es indicativo de una sección de construcción de ángulo o *build-up*.

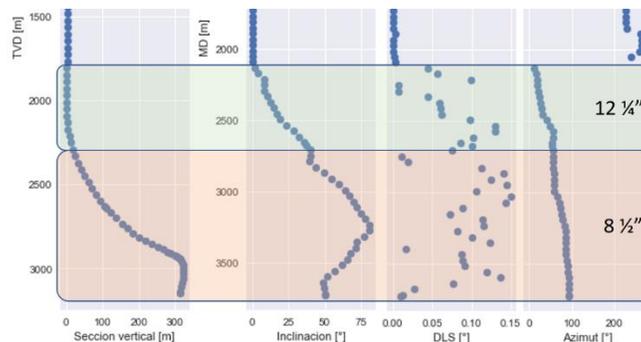


Figura 4.3.1.3 : Parámetros direccionales Pozo F14

Así mismo, es posible apreciar que en la etapa de barrena de 8 ½” se muestra una disminución constante de inclinación correspondiente a la caída o *drop*, seguida de un segmento constante de 50° conocido como tangente, en esta última sección de perforación, se tienen puntuales eventos de **S&S**.

4.3.2 Análisis de Pozo F10

Los eventos severos de este pozo se encuentran concentrados en la primer corrida de la etapa de barrena de 8 ½”, muestra de esto, en el **FWR** se menciona que, desde los inicios de la perforación, se aprecia bajo **ROP**, lo cual se corrobora con la Figura 4.3.2.1, en esta sección se atravesaron las formaciones *Ekofisk* y *Tor*, las cuales ya habían generado eventos de relevancia en el pozo anteriormente analizado.

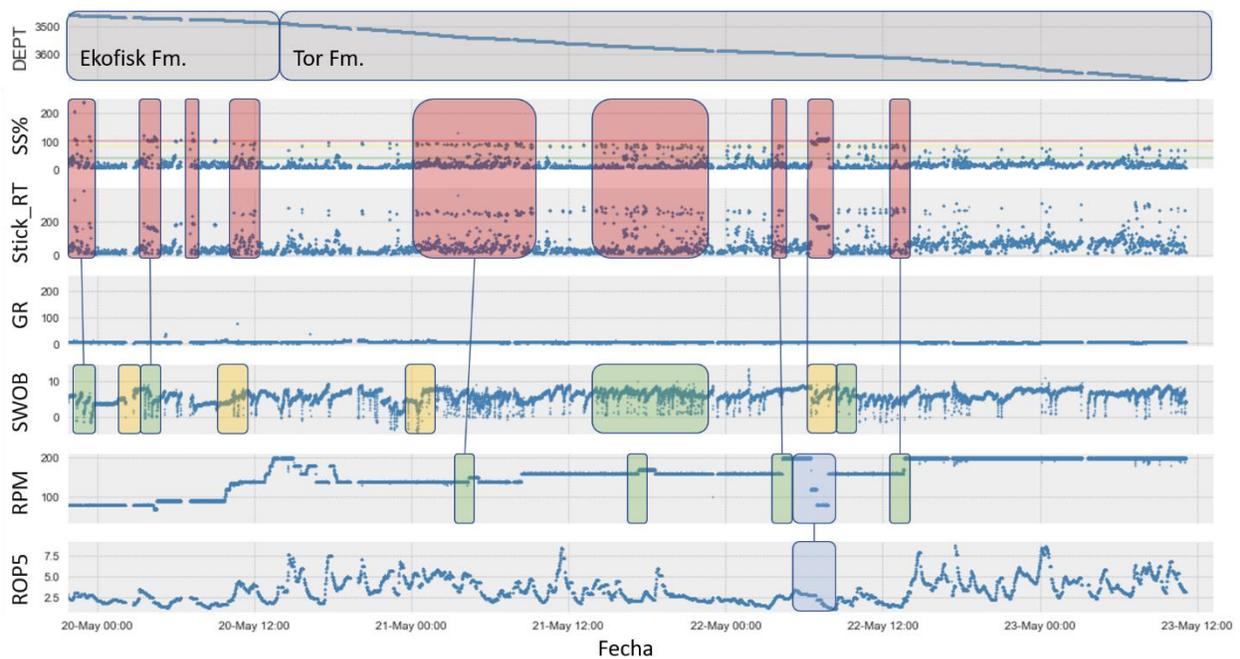


Figura 4.3.2.1 : Corrida 1, Barrena 8 ½” pozo F10

Los eventos más constantes y severos se encuentran en la formación *Tor*, así mismo, en esta formación es posible apreciarse la disminución constante y abrupta del **SWOB** existiendo momentos en los que este valor toma valores negativos, a pesar de estas medidas de mitigación, resulta muy interesante ver que existen cuatro eventos en los cuales se aumentó el **RPM** en esta formación.

En el recuadro azul puede apreciarse el efecto de disminuir las **RPM** de 200 a 70, una caída marcada en el ya bajo **ROP**.

Por motivos de desempeño, se determinó cambiar el **BHA** por otra barrena **PDC**, aunque con un mayor número de aletas, este registro de la segunda corrida, se muestra en la Figura 4.3.2.2, la primer diferencia visible es el ausencia del registro de **GR**, ya que este no se encuentra en los datos recabados y tampoco se hace mención de esta ausencia en el **FWR**; la segunda diferencia es la presentación de los datos, estos no son puntuales y constantes, sino abarcan una amplitud específica, siendo esto evidencia de una técnica realizada desde la cabina del perforador, la cual consiste en permitir la variación de las revoluciones en el **TDS** al recargar la barrena en el fondo, repitiendo dicho proceso por el tiempo que el perforador lo

requiera, en el recuadro azul se muestra el comportamiento del **ROP** utilizando esta técnica, mientras que al concluir dicha metodología (morado) es posible apreciar la mejora en el **ROP**, el cambio de perfil de **SWOB** y la disminución de los eventos de **S&S**, pudiendo indicar que algunas metodologías de perforación pueden tener una contraparte en la generación de vibraciones.

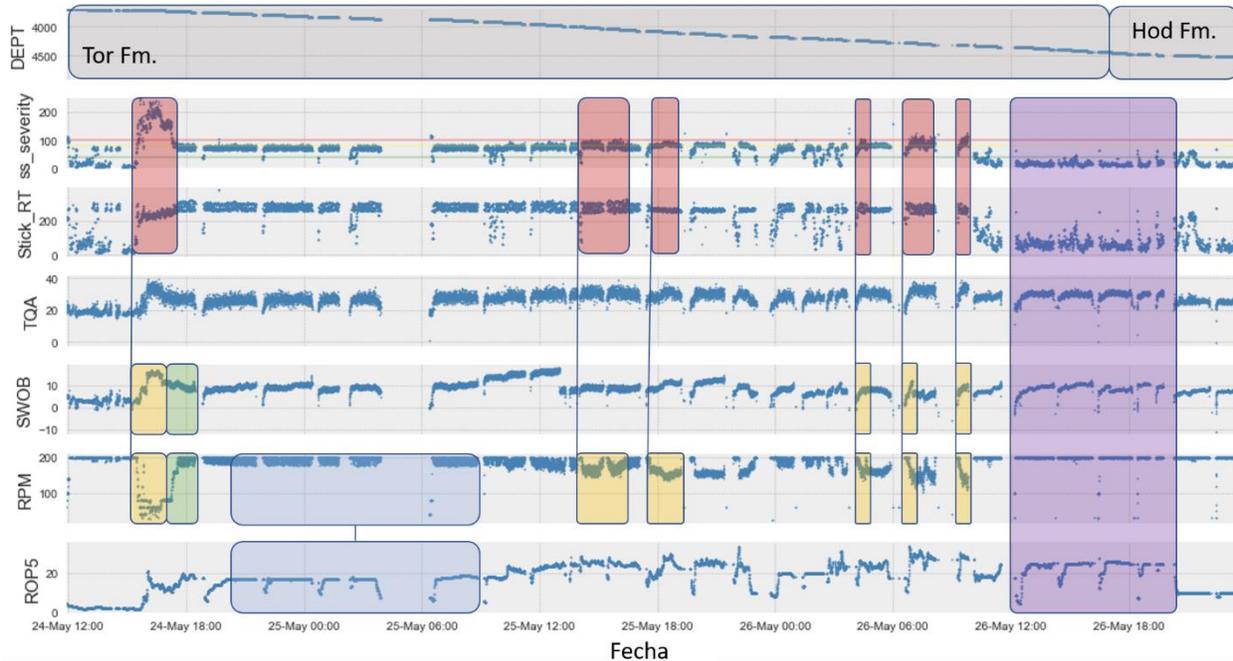


Figura 4.3.2.2 : Corrida 2, Barrena 8 ½” pozo F10

En esta segunda corrida o **BHA** es posible asociar con mayor facilidad las potenciales causas de los eventos de **S&S** severos.

Al realizar la comparación y el análisis de trayectoria, en la primer corrida, el **DLS** se encuentra en valores bajos ya que, la inclinación y el azimut muestran ángulos constantes, mientras que en la corrida 2, el **DLS** se comporta con una amplitud más marcada, y el registro **MWD** muestra eventos severos constantes y al final la perforación al declinar la inclinación, los eventos disminuyen.

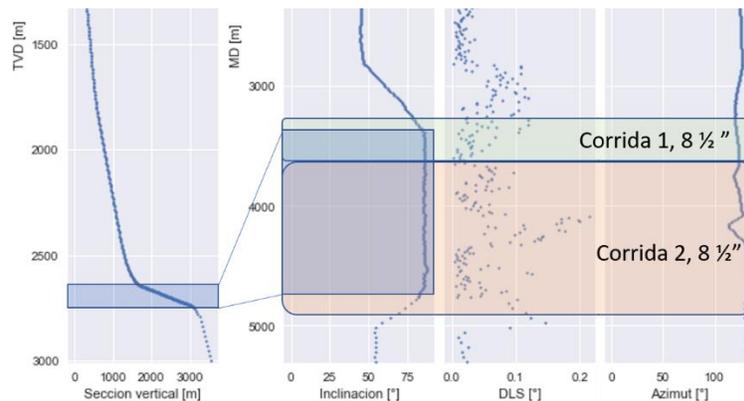


Figura 4.3.2.3 : Parámetros direccionales Pozo F10

En el carril correspondiente a la sección vertical, es posible ver un desfase que puede corroborarse en los recuadros azules.

4.3.3 Análisis de Pozo F5

Analizando la etapa de barrena de 12 ¼” mostrada en la Figura 4.3.3.1, se aprecian eventos de relevancia y cuyas posibles causas puede apreciarse con facilidad, así como la remediación realizada por parte del equipo de perforación; aunque en todas las formaciones es posible apreciar eventos severos, nuevamente la formación *Tor* muestra un comportamiento reiterativo de eventos severos, pudiendo detectar 4 intentos de remediación, 1 disminución gradual de **SWOB**, dos remociones de fondo y un aumento gradual en las **RPM**, los cuales no mejoran las severidades.

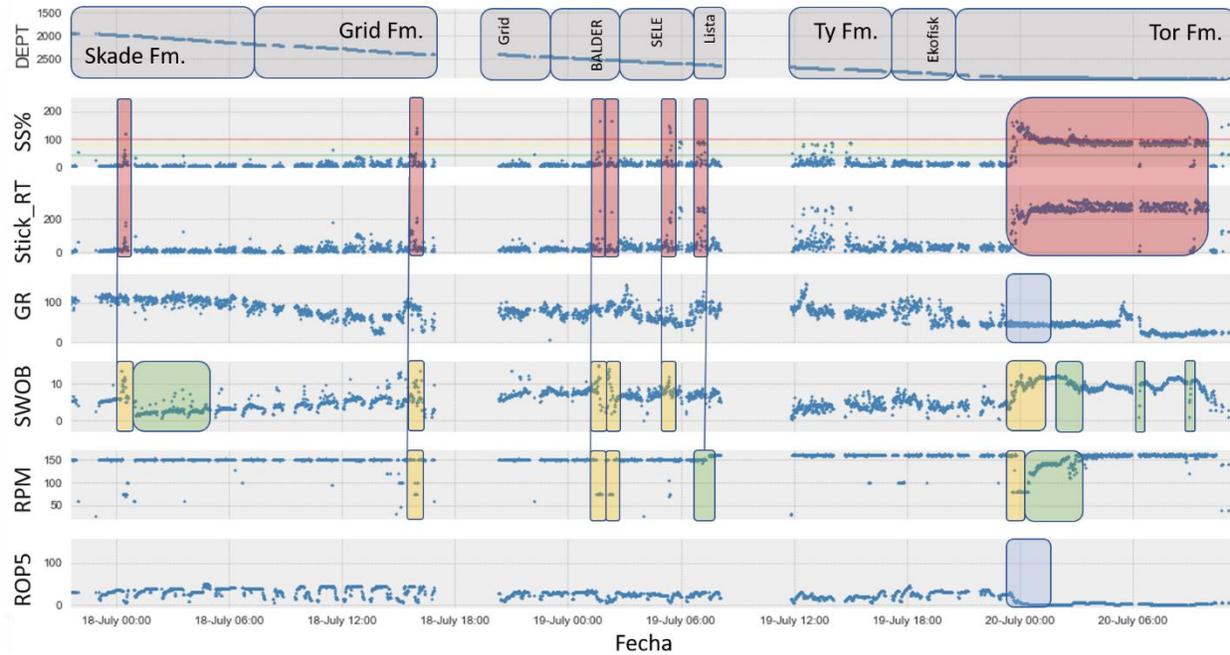


Figura 4.3.3.1 : Barrena 12 ¼” pozo F5

Así mismo, en los recuadros azules se muestra un comportamiento bastante peculiar al entrar en la formación *Tor*, y como esta formación repercute en la caída del **ROP**.

La etapa de barrena de 8 ½” fue perforada en dos corridas y dos ensambles homólogos, en la Figura 4.3.3.2 se muestra la primer corrida con este diámetro, en el **EOWR** se reportó que a partir de los 3,000 mD hubieron aumentos en los valores del **S&S**, los cuales pueden detectarse en las formaciones *Hod* a *Roedby*, estas ocurrencias tuvieron intentos de mitigación los cuales se expresan en el documento como sigue:

“[...] We tried to mitigate the sticknslip by increasing the Surface RPM, picking off bottom. Also it was tried by stopping both rotation and picking off bottom, releasing the torque in the string, but without any sucess. [...]”

(EOWR_F5, 2008)

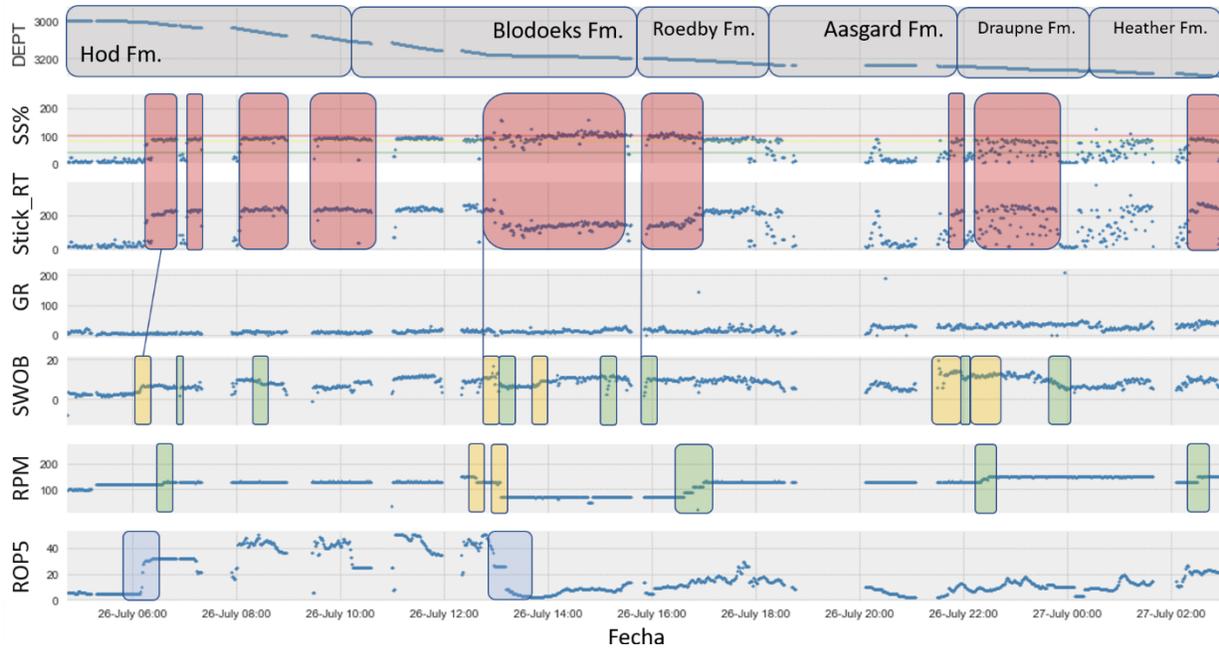


Figura 4.3.3.2 : Corrida 1, Barrena 8 1/2" pozo F5

En los recuadros azul es posible ver eventos de aumento y disminución de **ROP**, el primero dado un aumento de **WOB** ocasionando un incremento evidente de vibraciones torsionales; el segundo por una disminución de peso aunado a una disminución de **RPM**, marcando eventos severos y un bajo desempeño en la perforación.

En la Figura 4.3.3.3 es posible apreciar vibraciones severas puntuales en la formación *Hugin*, a los que se les pueden atribuir un incremento de peso, así como también dos eventos visualmente reconocibles de disminución significativa de **RPM** pudiendo ocasionar algunas ocurrencias de **S&S**.

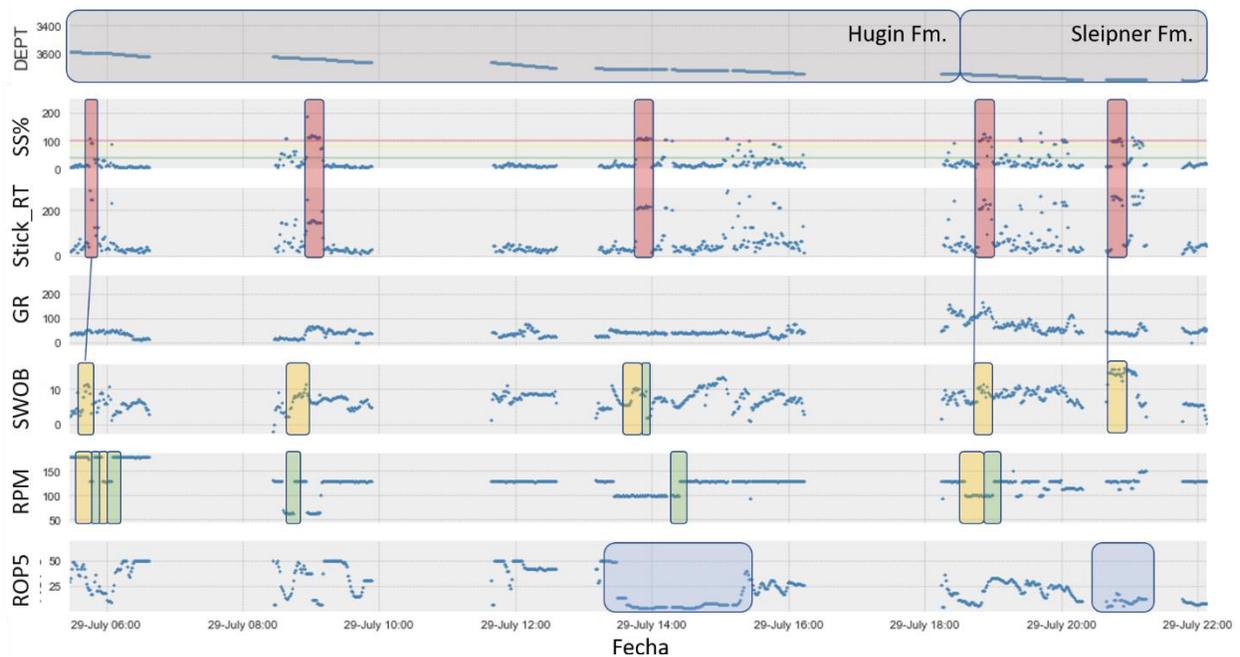


Figura 4.3.3.3 : Corrida 2, Barrena 8 1/2" pozo F5

Nuevamente, en los recuadros azules se muestran comportamientos relevantes en el **ROP**, una caída catastrófica que pudo ser ocasionada por una disminución de **RPM** y **SWOB**, por lo que se intenta mitigar liberando peso y más adelante aumentando rotaciones, remediaciones insatisfactorias ya que no mejora el desempeño; el siguiente recuadro muestra una tasa de penetración baja a pesar de que se recargó la barrena en fondo a casi 20[ton], por lo que se ocasionan eventos altos de **S&S** a pesar de que el **ROP** no ve mejoría.

Analizando los parámetros direccionales de la etapa de barrena de 12 ¼" se muestra una inclinación constante, un azimut que disminuye conforme se va perforando y valores de **DLS** inferiores a 0.001[°], con base en el **MWD** la formación *Tor* comienza a perforarse al finalizar esta sección, momento en el que la inclinación comienza a aumentar, y por consiguiente, el **DLS**.

Con la corrida 1 de diámetro 8 ½" pueden observarse eventos en los **MWD** altos constantes, mientras que la inclinación disminuye, al igual que el azimut, lo que se opone al comportamiento del pozo anterior; con respecto a la siguiente corrida, los eventos de **S&S** son de poca duración y ocasionales, los datos direccionales nos indican un aumento de inclinación y un **DLS** a la baja, sin embargo, en el primer carril, al que le corresponde la sección vertical previamente definida, se aprecian valores negativos por lo que estos son erróneos, esto pudo haber sido ocasionados por alguna falla del sistema medidor o en la transmisión del mismo.

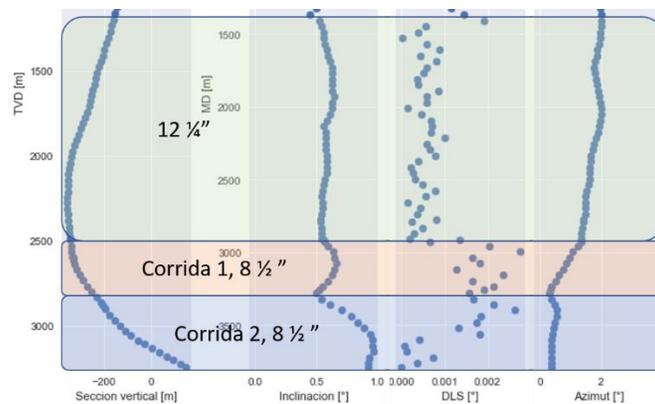


Figura 4.3.3.4 : Parámetros direccionales Pozo F5

4.3.4 Análisis de Pozo F4

El último pozo de análisis fue el **F4**, pozo cuyas etapas de barrenas disponibles son de 12 ¼" mostrada en la Figura 4.3.4.1, y las dos corridas correspondientes a la barrena de 8 ½" mostradas en la Figura 4.3.4.2 y en la Figura 4.3.4.4.

Al analizar los datos relevantes en la documentación, existe una omisión de eventos de esta índole, sin embargo, en la Figura 4.3.4.1, puede apreciarse pocos eventos de relevancia de **S&S**, un desempeño en el **ROP** muy bueno, únicamente disminuye notablemente a mediados de la formación *Lista*, donde es posible apreciar eventos más largos y remediaciones que recaen en el parámetro de **RPM**, al iniciar estos últimos cuatro eventos, puede apreciarse un cambio en la respuesta del **GR**, donde se establece cercana a la línea de arenas.

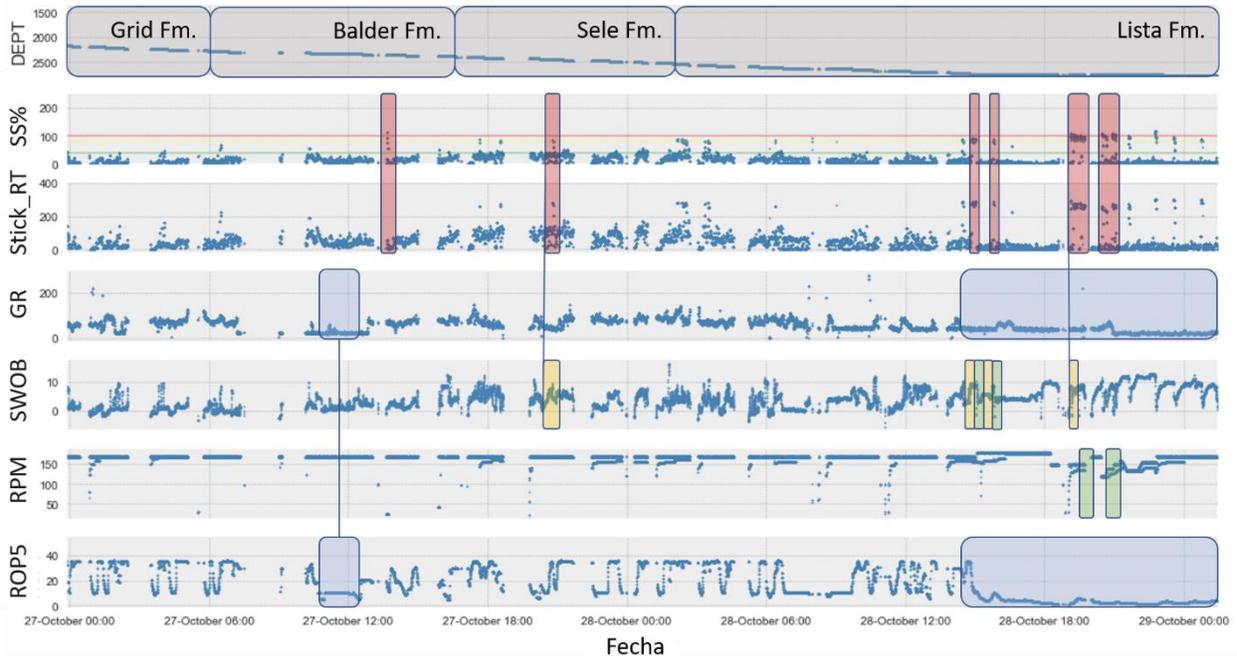


Figura 4.3.4.1 : Barrena 12 ¼" pozo F4

La Figura 4.3.4.2 correspondiente a la primer corrida de diámetro 8 ½" muestra nuevamente la tendencia de eventos severos en la formación *Tor*; sin embargo, estos eventos no se expresan en los reportes a pesar de que es posible ver intentos de mitigación, estas ocurrencias son reconocidas por un aumento marcado de **SWOB**.

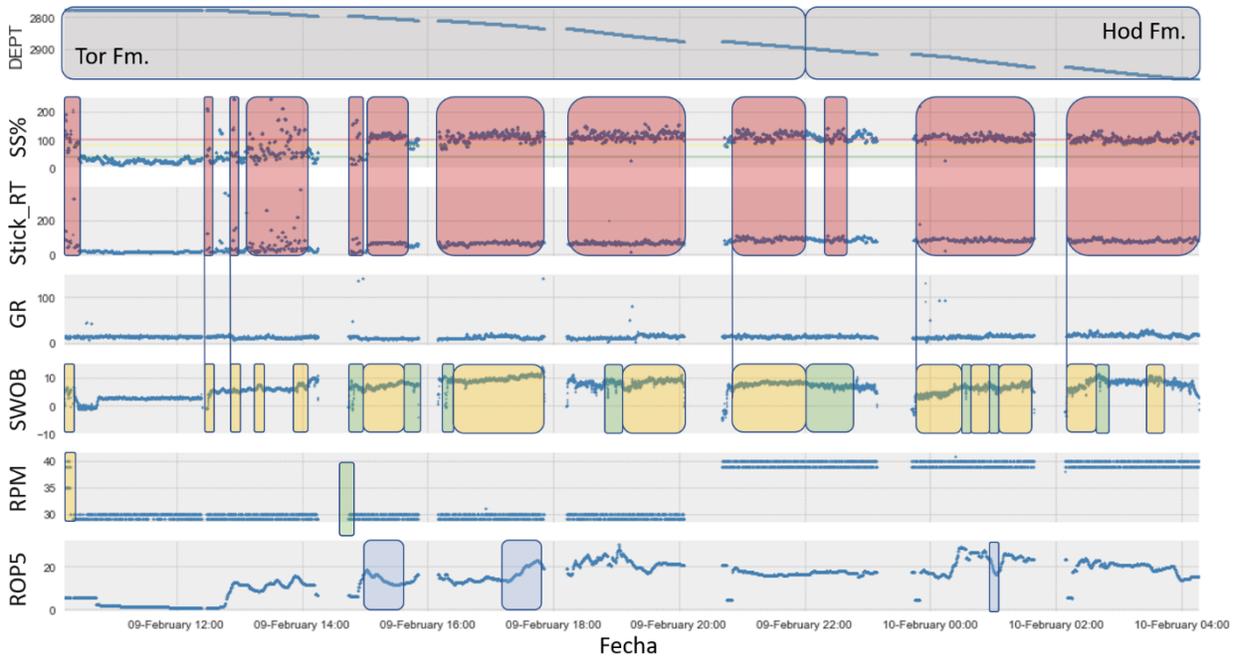


Figura 4.3.4.2 : Corrida 1, Barrena 8 ½" pozo F4

Es importante realizar la mención del carril correspondiente al **RPM**, el cual es visiblemente diferente a los previamente analizados, esto debido a que esta corrida fue realizada con un motor de fondo cuyas

características se muestran en los recuadros azules del *MWD RECORD* presente en el **FWR** del pozo, mostrado en la Figura 4.3.4.3.

Schlumberger 15/9-F-4 MWD RECORD (MWD Runs 1-)

MWD RUN #		8.5" 15/9-F-4	
BHA #	LWD RUN #	4	2
DRILLING PARAMETERS			
DEPTH IN (m)	DEPTH OUT (m)	2770.00	2993.00
DATE IN	DATE OUT	08.feb.08	10.feb.08
DRILLING HOURS	PUMP HOURS	11.60	28.00
ROP (m/hr)	WOB (kkg)	19.20	7.00
RPM (surface/total)	TORQUE (kNm)	30 / 200	16.50
FLOW (lpm)	PRESSURE (bar)	2060.00	240.00
BIT MAKE	BIT TYPE	Smith	PDC
MOTOR MAKE	MOTOR TYPE	D&M	Vortex
SURVEY PARAMETERS			
INCL START	INCL FINAL	29.63	34.44
AZIM START	AZIM FINAL	133.70	162.51
MWD OPERATIONAL PARAMETERS			
COLLAR #	TOOL #	AB Z258	P100
Stethoscope	ADN	N/A	N/A
CDR/ARC/Ecoscope	ISONIC/GVR	811-VE40	42255
TURBINE RANGE (gpr)	GAP (in.)	400 - 800	0.10
Telemetry Frequency	Bitrate	12	6
Tur RPM @ lpm	Bit - D&I (m)	2930@2060	32.23
Bit - Res. (m)	Bit - Gamma (m)	26.18	23.14
Bit - DT (m)	Bit - APWD (m)	41.65	23.30
Bit - Porosity (m)	Bit - Density (m)	26.59	24.40

Figura 4.3.4.3 : MWD RECORD pozo F4

En la Figura 4.3.4.4 al inicio de la perforación, se expresa en el reporte que se tuvieron eventos severos lo que disminuyo el "ROP", al inicio de la formación *Heather* se vio reducido significativamente los eventos de "S&S", siendo esta primordialmente lutita arcillosa con intercalaciones arcillosas y de calizas.

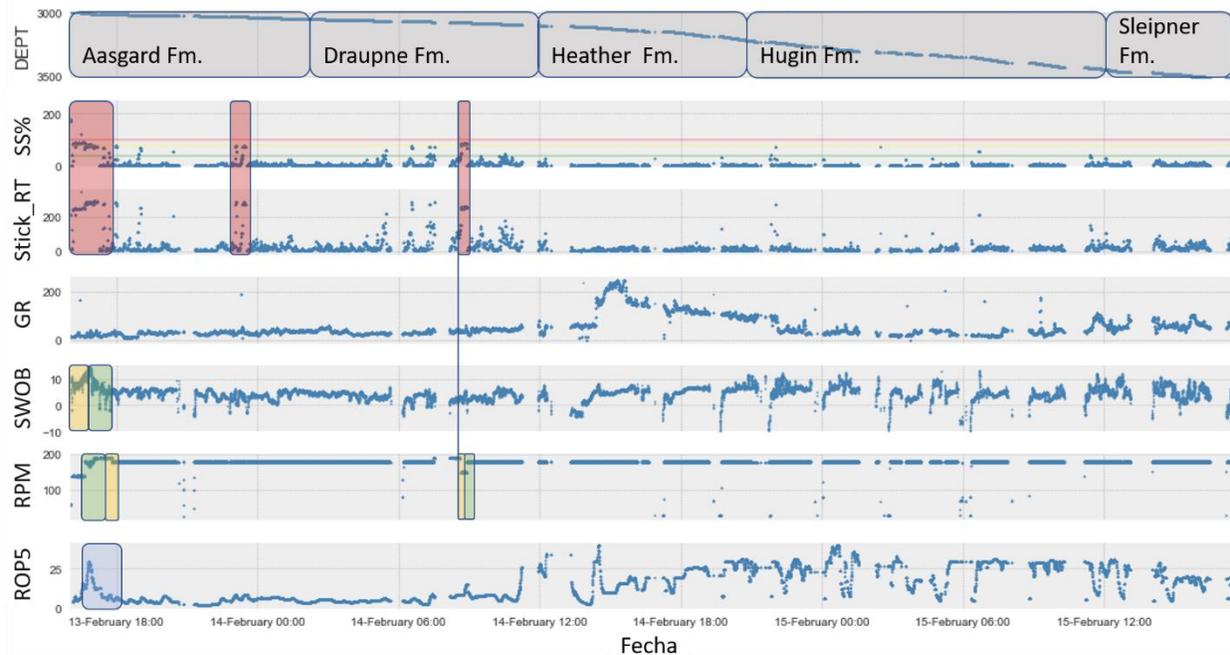


Figura 4.3.4.4 : Corrida 2, Barrena 8 1/2" pozo F4

Al analizar los parámetros direccionales, la barrena de 12 ¼" muestra parámetros constantes, mientras que el **MWD** indica presencia de pocos eventos de relevancia; en la primer corrida de 8 ½" es posible ver un incremento severo de **DLS**, la inclinación y el azimut muestran una tendencia incremental, en el registro de vibraciones los eventos son largos y altos de **S&S**; en la última corrida los eventos son ocasionales y el **DLS** muestra decrementos.

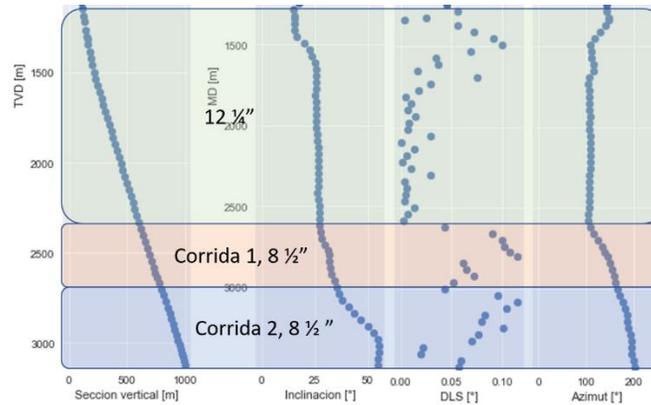


Figura 4.3.4.5 : Parámetros direccionales Pozo F4

Y nuevamente, en el primer carril no se muestra correctamente la respuesta vista en el carril de la inclinación, sugiriendo un error en su registro.

A lo largo del presente capítulo, se fue tratando de una forma práctica las implicaciones que tiene este evento vibratorio en el desempeño de perforación, sus evidencias y posibles consecuencias, a continuación, se abordarán los mismos datos, pero desde un enfoque de la ciencia de datos y de **EDA**.

En la Figura 4.3.4.6, se muestra el grafico de violín, donde se analiza el comportamiento de cada sección de perforación en función del valor del **Stick_RT**, siendo este el valor obtenido en campo. En esta representación gráfica es posible discernir una tendencia de importancia, con base en el subtema 3.2 Ciencia de Datos el valor de mayor ocurrencia es mostrado como una protuberancia; estas se concentran en las secciones de menor diámetro, por ende, a mayor profundidad, como se muestra en los recuadros rojos.

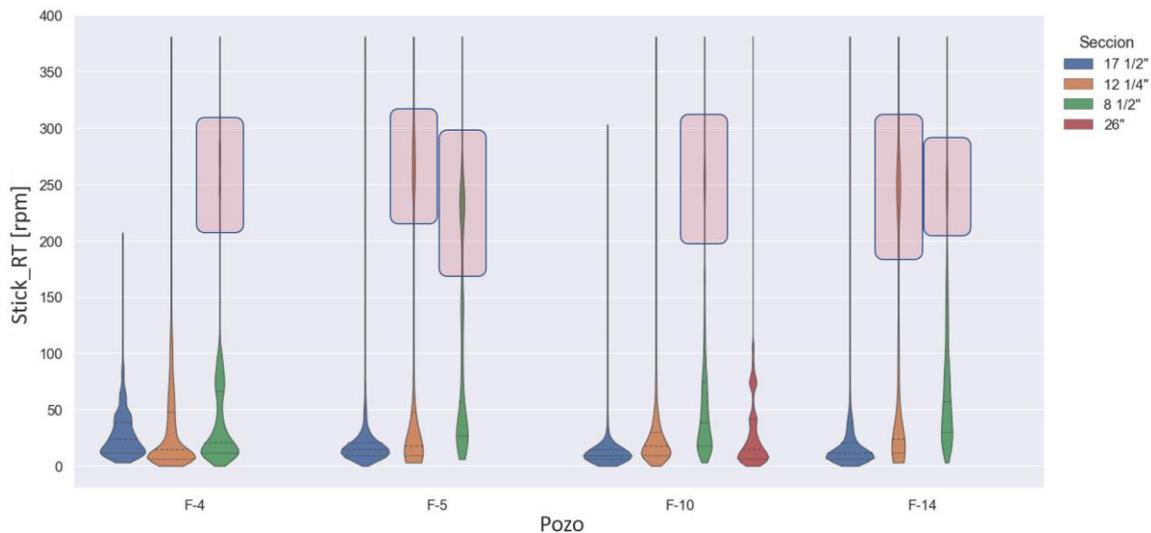


Figura 4.3.4.6 : S&S en secciones de perforación

En otras palabras, los eventos cuyos valores de **Stick_RT** son mayores a 200[rpm], se concentran principalmente en secciones de 8 ½”, así mismo para las secciones con barrena de 12 ¼”, ya que existen eventos en los pozos **F14** y **F5** y que son fácilmente reconocibles; en los pozos **F4** y **F10** no sucede esto, ya que la línea no adquiere un área.

Realizando una variación en la presentación de los datos, se puede obtener el gráfico de la Figura 4.3.4.7.

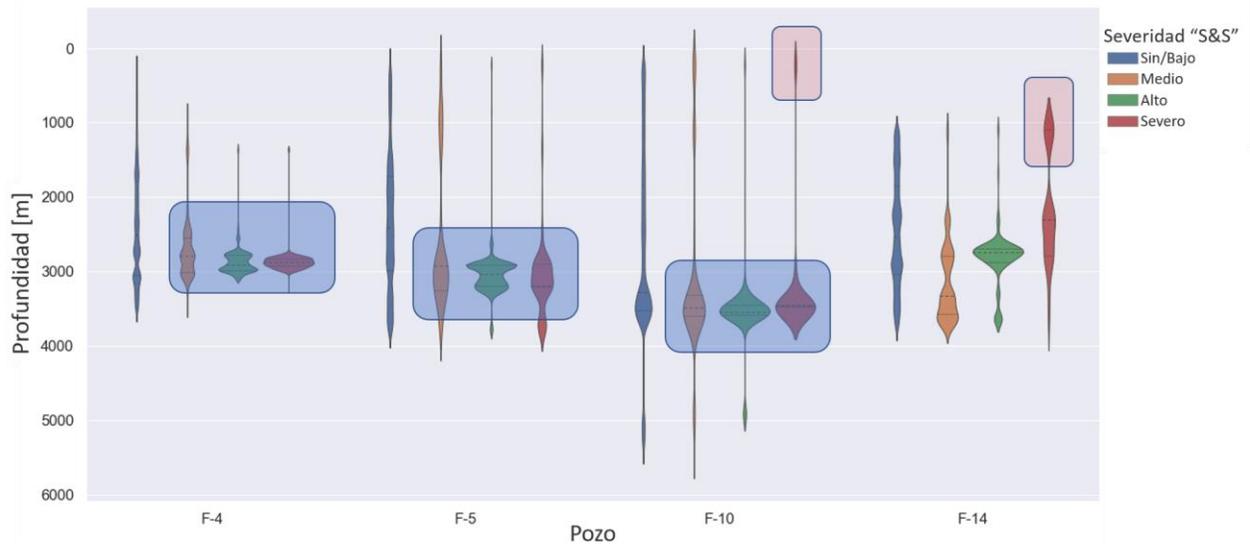


Figura 4.3.4.7 : Severidad S&S categórica por pozo

En esta es posible reconocer una tendencia sumamente interesante y de relevancia, se muestra la severidad categórica en función de la profundidad, para los pozos **F4**, **F5** y **F19**, los eventos severos están reservados en el intervalo 2,500 – 4,000 mD, y estos se encuentran precedidos por eventos altos, es decir, en el pozo **F10** y **F4**, el área de los eventos altos abarca los eventos severos, esto también puede decirse de los eventos medios mientras que en los pozos **F5** y **F14** no existe dicha tendencia.

Para analizar los datos direccionales, se obtuvieron las siguientes representación gráficas, con la distinción que en estas se decidió presentar a la par, el histograma, ya que el intervalo de grados necesita mayor detalle, en la Figura 4.3.4.8 se muestra la severidad de **S&S** en cada survey realizado, encuadrando en rojo los eventos de relevancia, en este es posible apreciar que se tuvieron eventos severos en el intervalo 140-160[°] de azimut, es decir, cuando el eje entre la barrena y el norte magnético tomaba estos valores, existieron vibraciones torsionales de relevancia, esto tiene connotaciones interesantes en cuanto a la geomecánica de las formaciones en las cuales se perforaron, sin embargo el alcance de este trabajo y de los datos disponibles no nos permite explorar esas posibles razones; así mismo, existieron 152 eventos medios específicamente a los 0[°] de azimut.

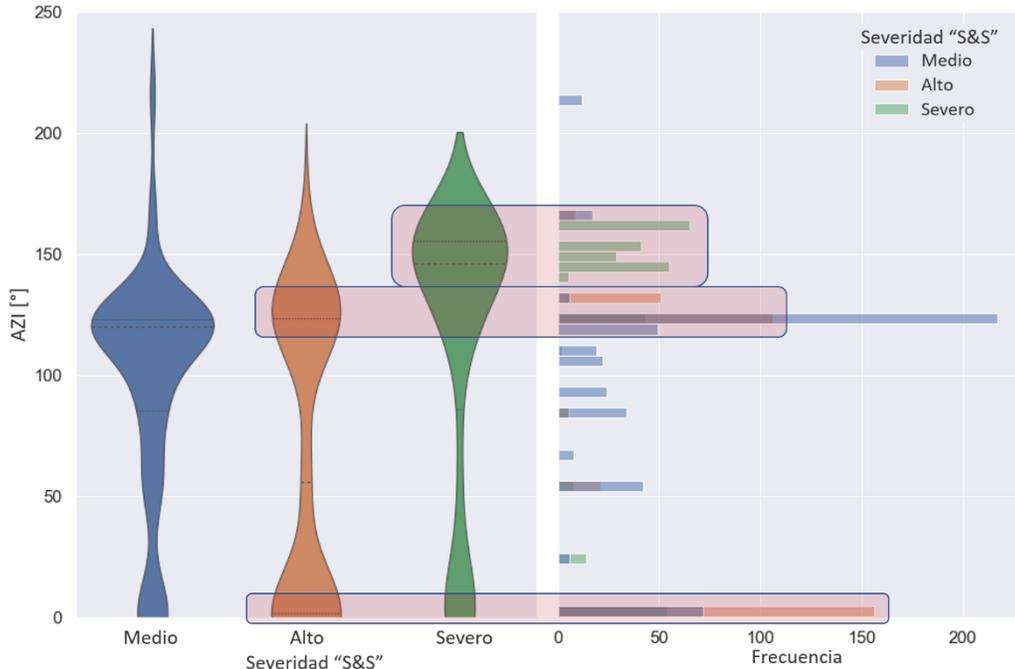


Figura 4.3.4.8 : Severidad S&S categórica para Azimut

En la Figura 4.3.4.9 se muestran más de 150 eventos severos cuando el ángulo entre el eje de la barrena y el vertical del pozo tomaba el valor de 33[°] de inclinación, recordando que este ángulo de deflexión es uno de los parámetros de relevancia cuando se establece la trayectoria de un pozo, por lo que este ángulo genera vibraciones potencialmente dañinas para el **BHA** y este fue de diseño; también, cuando el pozo adquiere una verticalidad, se generan vibraciones altas, a la par que 55 vibraciones severas.

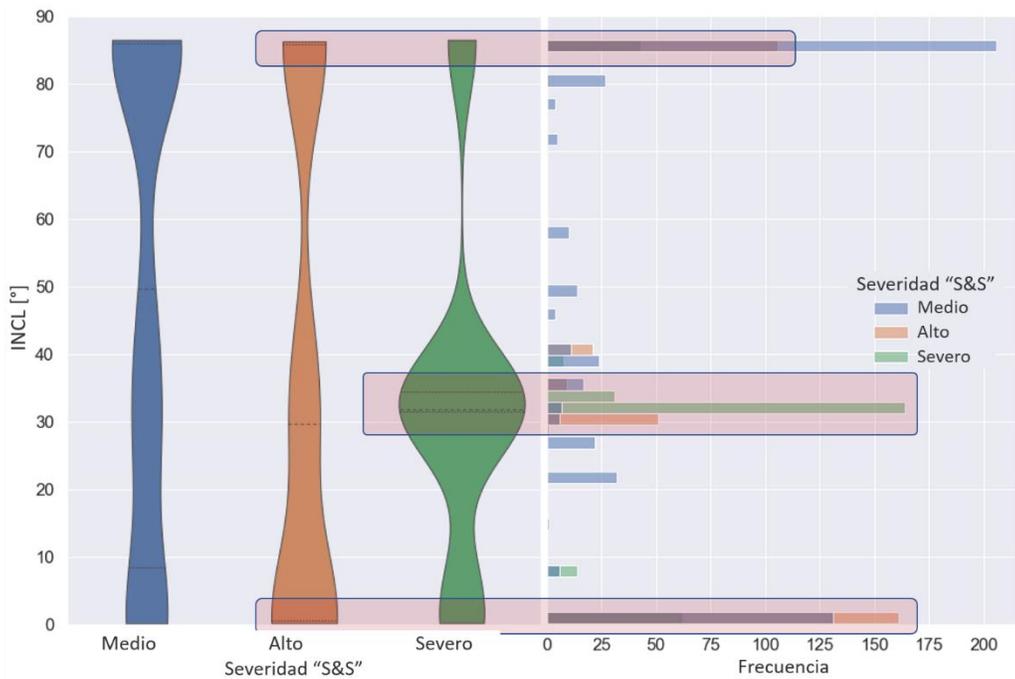


Figura 4.3.4.9 : Severidad S&S categórica para Inclinación

La Figura 4.3.4.10, muestra el cambio en la inclinación y azimut, mejor conocido como **DLS**, este indica que los eventos severos (gráfico verde) prevalecieron en tres intervalos, a los .06, .12 y a los 0[°/m], cuando los altos principalmente a los 0 y a los 0.10[°/m].

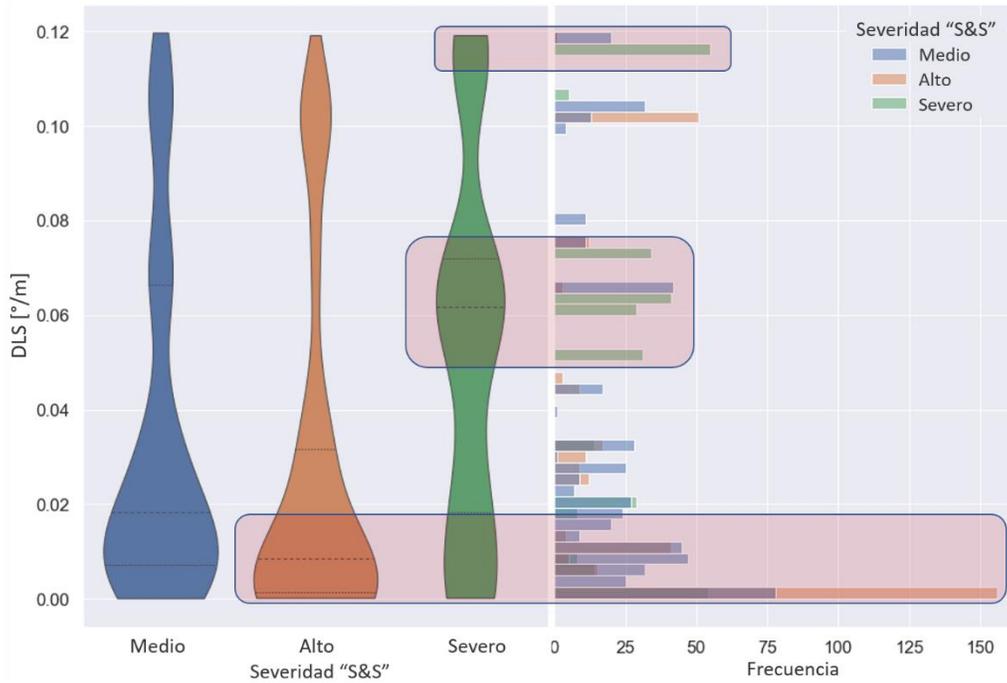


Figura 4.3.4.10 : Severidad S&S categórica para DLS

Es importante mencionar que para realizar el análisis de la presentación gráfica *pairplot*, la cual se hará a continuación, se debe realizar tomando en cuenta la diferencia que existe entre la correlación, que es una medida estadística que existe entre dos variables, y la relación causal, siendo esta la ocurrencia de un evento un efecto dado una causa. Por lo que, una correlación entre dos variables no implica causalidad, sin embargo, de existir una relación causal entre ellas, estas deben de estar correlacionadas.

Hasta este momento, los gráficos Figura 2.2.3 y Figura 3.2.4, muestran relaciones causales entre las variables de temperatura y peso con la profundidad, pudiendo haber relaciones más complejas que poseen una correlación mas no una causalidad.

Analizando estos eventos utilizando la herramienta *pairplot*, se puede obtener la Tabla 8.

Tabla 8 : Análisis correspondiente al Pairplot

Inciso	Relación	Descripción
a	SWOB RPM	Los eventos severos se encuentran concentrados en revoluciones inferiores a 150[rpm] y en su mayoría a pesos inferiores a 12 [ton], mientras que los altos a revoluciones superiores.
b	RPM	El histograma muestra que los eventos severos se concentran en revoluciones cercanas a 70 [rpm], mientras que los altos en 130 [rpm].
c	ROP5	Los eventos severos se concentran cerca 20 e inferiores a 5 [m/hr], mientras que los eventos medios en 5 [m/hr].
d	GR	Los eventos se concentran en respuestas valores inferiores a 25 [°API], mientras que los altos se presentan de igual forma a los 25 y 50 [°API].
e	TFLO DEPT	Cuando se tienen gastos inferiores a 2,500 [lpm] y profundidades cercanas a los 3,000 mD se tendrán eventos severos; mientras que en gastos superiores a la misma profundidad se tienen eventos altos.
f	RPM TQA	Los eventos medios se encuentran homogéneamente distribuidos en función del TQA , mientras que, los eventos severos se concentran en revoluciones inferiores a 125 pero con elevado TQA .
g	DEPT AZI	Los eventos severos se concentran en azimut superiores a 120 [°]; los eventos altos y medios, se encuentran a profundidades superiores a 2,500 mD.
h	TQA AZI	Se tendrán eventos severos si se tiene un azimut superior de 120 [°], y si el TQA se encuentra en el intervalo 12-22 [kNm].
i	AZI	Los eventos severos se concentran en 0 y en 160 [°], mientras que los altos únicamente a los 150 [°]
j	DEPT DLS	Los eventos altos se encuentran en valores de DLS inferiores a 0.7 [°] y por debajo de los 2 ,500 mD.
k	DLS	Se tienen los mismos eventos categóricos en ángulos 0.1 [°/m], los medios y altos se encuentran principalmente cercanos a 0 [°/m]
l	RPM DLS	Existe una ausencia de valores altos y medios en valores inferiores a 125[rpm] y en el intervalo de DLS comprendido entre 0.025 - 0.1 [°].
m	INCL DLS	Los eventos altos se encuentran en valores de DLS menores a 0.5 y a valores de ángulos mayores a 25 [°], mientras que los severos se encuentran en el intervalo de 20 – 50 [°].
n	INCL	Los eventos severos se concentran en ángulos de 25 – 30 [°], mientras que los altos se encuentran cercano a los 90 y a los 0 [°].
o	SWOB INCL	los eventos severos se encuentran en “SWOB” inferiores a 10 [ton] y en ángulos inferiores a 30° de INCL, mientras que los eventos altos se encuentran a un peso superior de 12 [ton].

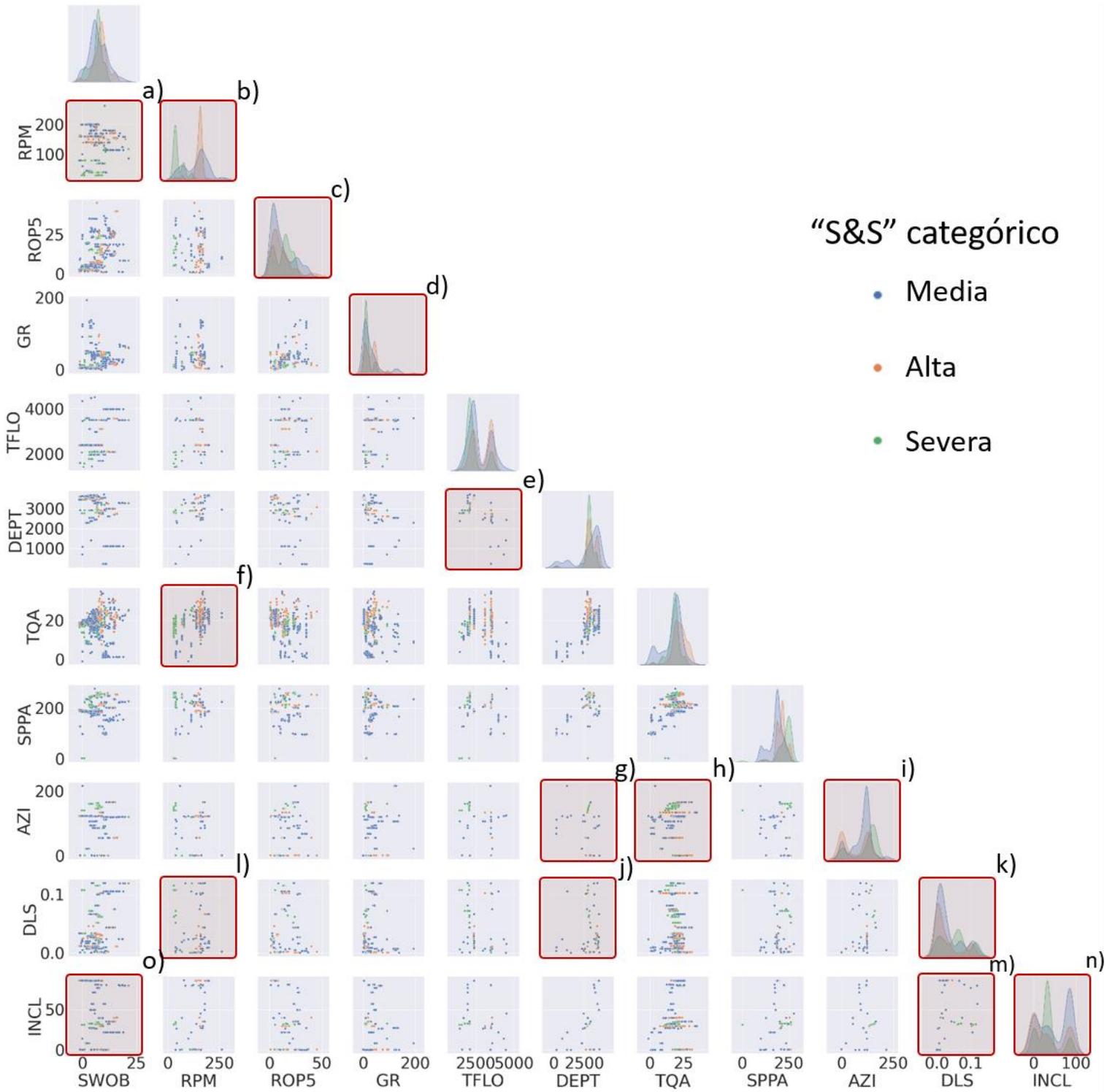


Figura 4.3.4.11 : Pairplot de S&S categórico

Capítulo 5. Predicción de Vibraciones Torsionales

Retomando la Figura 3.1.2, el primer paso para la creación satisfactoria de un modelo de predicción utilizando un algoritmo de **ML** es el planteamiento, por lo que decidió establecer el siguiente:

La construcción de dos modelos de *Random Forest* para predecir severidades de **S&S**, a partir de datos de parámetros de perforación y datos direccionales.

5.1 Recopilación de datos

Para la construcción del modelo de predicción, se utilizaron los parámetros de perforación y direccionales de los pozos **F4**, **F5**, **F10** y **F14**, los cuales consisten en los archivos **LAS** y los **WITSML** de trayectoria respectivamente.

La Figura 5.1.1 muestra las secciones de perforación disponibles para el pozo **F14**, registros ya explicados en el subtema 2.2 Adquisición de datos.

```
MWD DATA:
data:\MWD\BIT_RUNS\WL_RAW_BHPR-GR-MECH_TIME_MWD_1.LAS
-----
Hole      Main      Service   Pass Direction  Interval   Date       Run   Original
section  Services  Company   (Up/Down)      (meters)  (start)   Run   file name
-----
17 1/2"  DRILL.MECHANICS  SCHLUMBERGER  DOWN          TIME-INDEXED  10-MAY-08  3
07sca0093_volve_15_9_f_14_drill_mech_time_based_175in.las
data:\MWD\BIT_RUNS\WL_RAW_BHPR-GR-MECH_TIME_MWD_2.LAS
-----
Hole      Main      Service   Pass Direction  Interval   Date       Run   Original
section  Services  Company   (Up/Down)      (meters)  (start)   Run   file name
-----
12 1/4"  DRILL.MECHANICS  SCHLUMBERGER  DOWN          TIME-INDEXED  21-MAY-08  4   mud_log_time.las
data:\MWD\BIT_RUNS\WL_RAW_BHPR-GR-MECH_TIME_MWD_3.LAS
-----
Hole      Main      Service   Pass Direction  Interval   Date       Run   Original
section  Services  Company   (Up/Down)      (meters)  (start)   Run   file name
-----
8 1/2"   DRILL.MECHANICS  SCHLUMBERGER  DOWN          TIME-INDEXED  10-JUN-08  5   15_9_f_14_85in_run5_dm_apwd_time.las
```

Figura 5.1.1 : Archivos LAS de pozo F14

Estos archivos al ser indexados por tiempo, las bibliotecas disponibles en Python para la interpretación de los *well logs* pudiendo ser *lasio* o *welly*, no tienen el soporte para su interpretación y análisis, por lo que se procedió programar una función capaz de construir una base de datos a partir de estos archivos para obtener una por cada archivo **LAS**, resultando en tres estructuras de datos por pozo, esta función se encuentra en el Anexo 2. Códigos Elementales, bajo el nombre “time_indexed_LAS”.

Por otra parte, los registros de trayectoria por su formato, resultan más complejos de visualizar, sin embargo, existe la biblioteca *BeautifulSoup* que permiten su visualización e interpretación y posteriormente la creación de bases de datos, a partir de los *survey’s* contenidos en cada archivo, la función “read_xml” tiene este propósito, y la cual se encuentra en el Anexo 2. Códigos Elementales.

Con estas funciones se obtuvo una base de datos a partir del **MWD** del pozo, y una que contiene los *survey’s*; es importante mencionar que los primeros contienen procesos adicionales a la perforación, por lo que resulta de suma importancia acotar dichos datos.

5.2 Limpieza y construcción de base de datos

Este proceso consistió en la refinación y aglomeración de las bases de datos, para terminar con una sola, siendo esta la estructura que servirá de entrenamiento para nuestros modelos de predicción.

Se necesitó filtrar y suprimir procesos ajenos a la perforación, en otras palabras, mantener las mediciones únicamente cuando la barrena se encontraba en fondo y perforando por lo que se procedió a utilizar la función “drilling_process_fell”, la cual, con base en los registros de inicio de perforación agrega un indicador de *Bit On Buttom* o **BONB**, para que de esta forma se pudiera realizar este filtrado, resultando en la Figura 5.2.1.

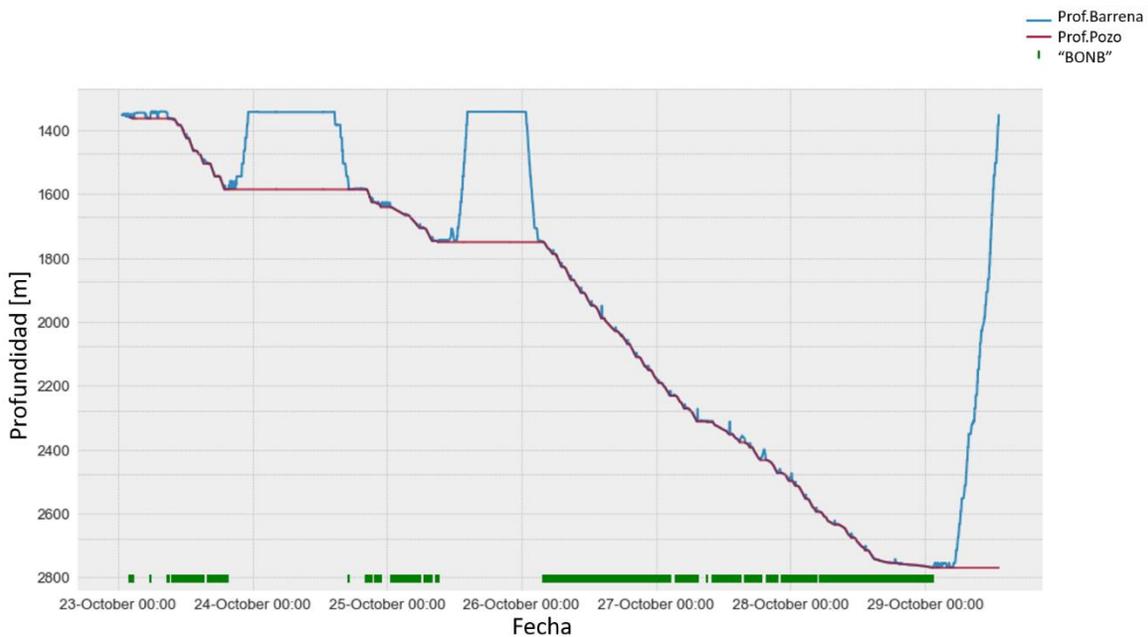


Figura 5.2.1 : BONB de etapa de barrena de 12 ¼" F-4

En esta, se muestra gráficamente el proceso de perforación en el tiempo, así como la profundidad de barrena (azul), existiendo momentos en los cuales se retira del fondo para realizar algún procedimiento; la profundidad del pozo (rojo), la cual suprime estas remociones y por último el valor booleano de **BONB** en la parte inferior, siendo una línea verde cuando se perfora, y una línea transparente cuando no.

Cabe recordar que por pozo se tienen los registros de perforación de las etapas de perforación de las barrenas de 17 ½", la de 12 ¼" y finalmente de 8 ½", entre estos se cotejaron los registros presentes en cada uno de ellos, hasta terminar con los siguientes:

[GR, BPOS, BVEL, DEPT, ECD, HKLD, ROP5, RPM, SPPA, Stick_RT, SWOB, TFLO, TQA]

Por lo que es necesario concatenar estas secciones para conformar una base de datos por pozo, para esto se analizó estadísticamente revisando la coherencia en estas mediciones, en la Figura 5.2.2 se muestra el gráfico de caja por pozo para el indicador de **HKLD** en el cual se observa que las mediciones tienen una coherencia y continuidad con respecto a magnitudes, a pesar de que en el pozo **F5** se ve una disminución en la carga ejercida por el gancho.

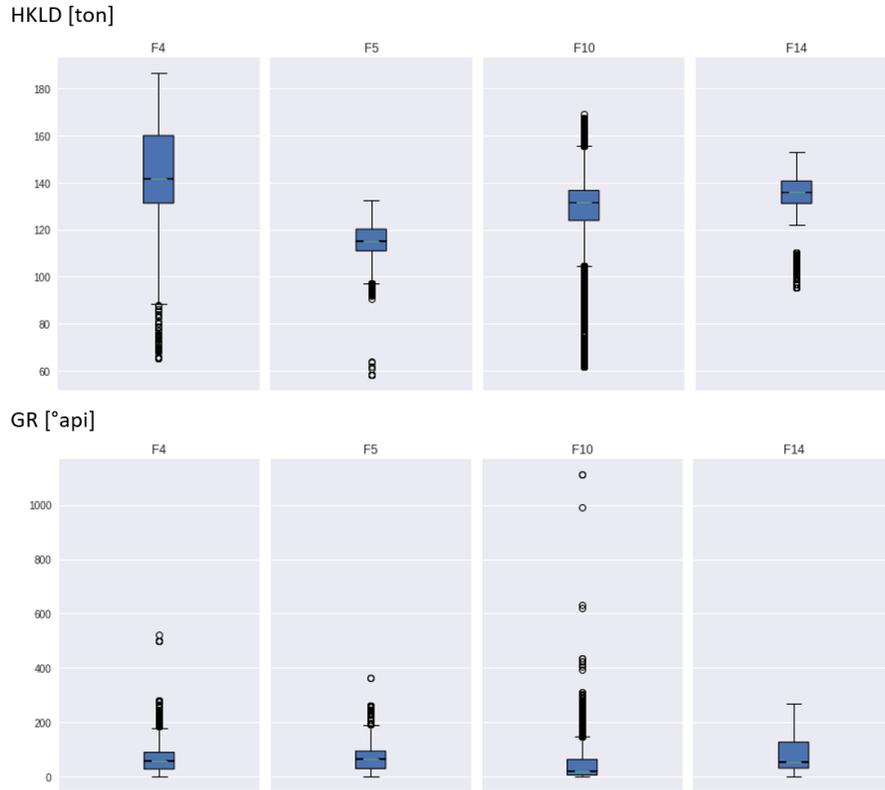


Figura 5.2.2 : Gráfico de caja por pozo

También en esta se muestra el carril correspondiente a la medición de rayos gamma o **GR**, en el pozo **F10** es posible apreciar una gran cantidad de *outliers* que, por la naturaleza de la medición, se mantendrán a lo largo de la construcción de la base de datos para *entrenamiento* y *testing*.

Adicionalmente, se decidió acoplar datos direccionales, para tener la mayor cantidad de datos útiles posible, de esta forma la inclinación, el azimut y el **DLS** fueron agregados a esta base de datos, sin embargo, al realizar este proceso de unión, se perdieron datos, ya que, hay que recordar que los datos de *survey* son tomados ocasionalmente, y en las profundidades intermedias no se tienen datos, por lo que se obtuvo lo siguiente:

Pre-acople de datos direccionales → 247,279 elementos
 Post-acople de datos direccionales → 10,595 elementos

Una vez conformada la nueva base de datos, se procedió a aplicar la función “*ss_severity*”, la cual aplica las variables categóricas establecidas en la Tabla 6 y de esta forma realizar el análisis del Capítulo 4. *Stick & Slip*.

En este punto, resulta importante puntualizar que las bases de datos creadas poseen una cualidad que puede visualizarse en la Figura 5.2.3

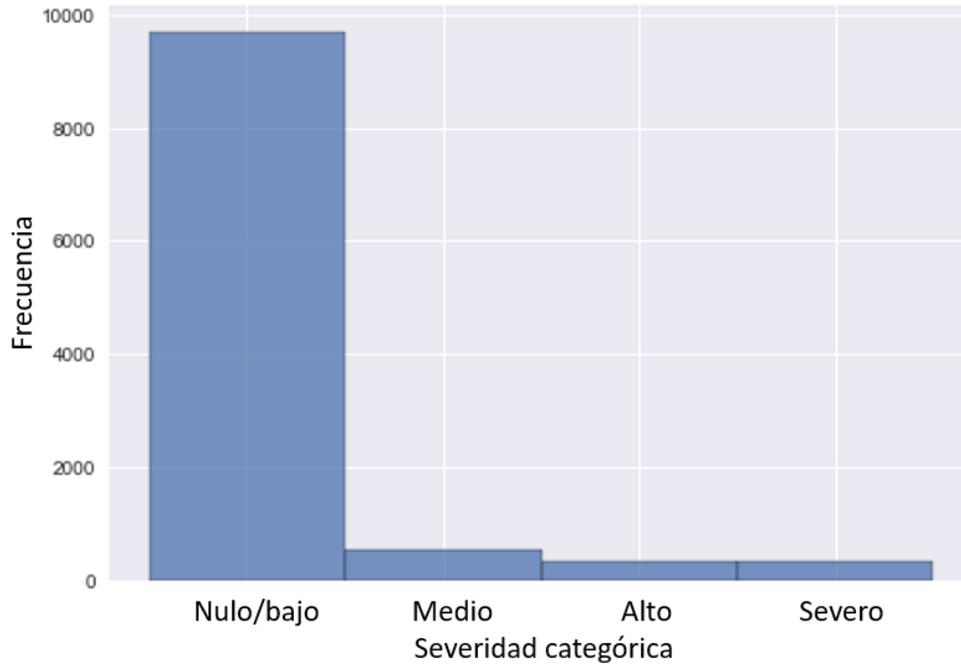


Figura 5.2.3 : Histograma de severidades categóricas

Esta nos muestra el total de ocurrencias de cada una de las severidades del **S&S**, existiendo un conjunto de datos desbalanceado, es decir, sus clases no poseen un mismo número de elementos, sin embargo, como se ha visto a lo largo de los análisis previos, las vibraciones torsionales medias a severas, son eventualidades en la perforación, ocurren puntualmente dadas ciertas condiciones, por lo que esta condición es inherente al fenómeno que queremos predecir.

A pesar de este contratiempo, se decidió avanzar y desarrollar los algoritmos de Árbol de decisión y *Random Forest*.

5.3 Construcción de árbol de decisión

Ya que el árbol de decisión es el fundamento del algoritmo *Random Forest*, se decidió realizar un modelo en el cual, mediante los parámetros de **SWOB** y **RPM** se intente predecir la severidad categórica del **S&S**, esto con propósitos didácticos y de esta forma analizar la construcción de este modelo utilizando la herramienta de *Scikit-learn* de Python enfocada en arboles de decisión.

Como parte del diagrama de flujo mostrado en la Figura 3.1.2, se necesita analizar si estas variables poseen una relación de colinealidad, y de esta forma evitar utilizar parámetros que aporten la misma información para la predicción de la severidad de S&S.

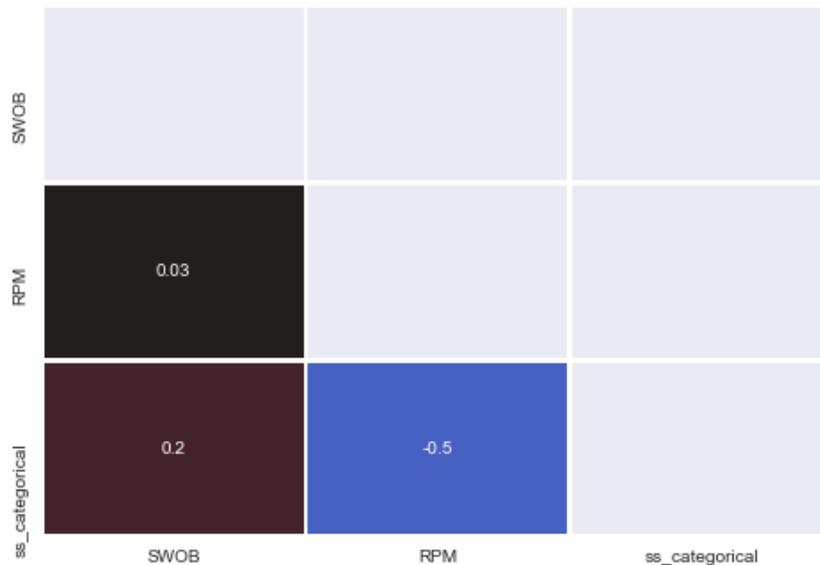


Figura 5.3.1 : Coeficientes de correlación de Pearson para árbol de decisión

En esta Figura 5.3.1 se muestran los coeficientes de correlación de Pearson, para determinar si existe alguna relación lineal entre las variables que conformaran los datos de entrenamiento para el modelo, al ser estos valores inferiores a la unidad, estos no se encuentran relacionados entre sí, lo que permitirá un buen desempeño en la construcción y entrenamiento del modelo.

```
Predictores:
    ['RPM', 'SWOB']
Tamaños:
  Training:
    X: (197823, 2)
    Y: (197823,)
  Testing:
    X: (49456, 2)
    Y: (49456,)
```

Figura 5.3.2 : Sets de datos para árbol de decisión

En la Figura 5.3.2 se muestran los tamaños finales de los conjuntos o *sets* de datos, utilizando un 80% de ellos para entrenar el modelo, y 20% para evaluarlo o realizar el *testing*.

5.3.1 Parámetros del modelo

Como ya se mencionó anteriormente, se utilizó la función “DecisionTreeClassifier” de la librería de código libre *Scikit-learn* resultando el comando siguiente

```
%%time
dtc = DecisionTreeClassifier(random_state=2,max_leaf_nodes = 5,max_depth = 4).fit(x_train,y_train)
```

Definiendo los argumentos:

- *random_state*
Controla la aleatoriedad del algoritmo, se definió un valor para obtener resultados replicables.
- *max_leaf_nodes*
Establece el máximo número de nodos de decisión, estableciendo 5 como máximo para que el árbol resultante sea didáctico.
- *max_depth*
El máximo número de profundidad del árbol, estableciendo un máximo de 4 niveles.

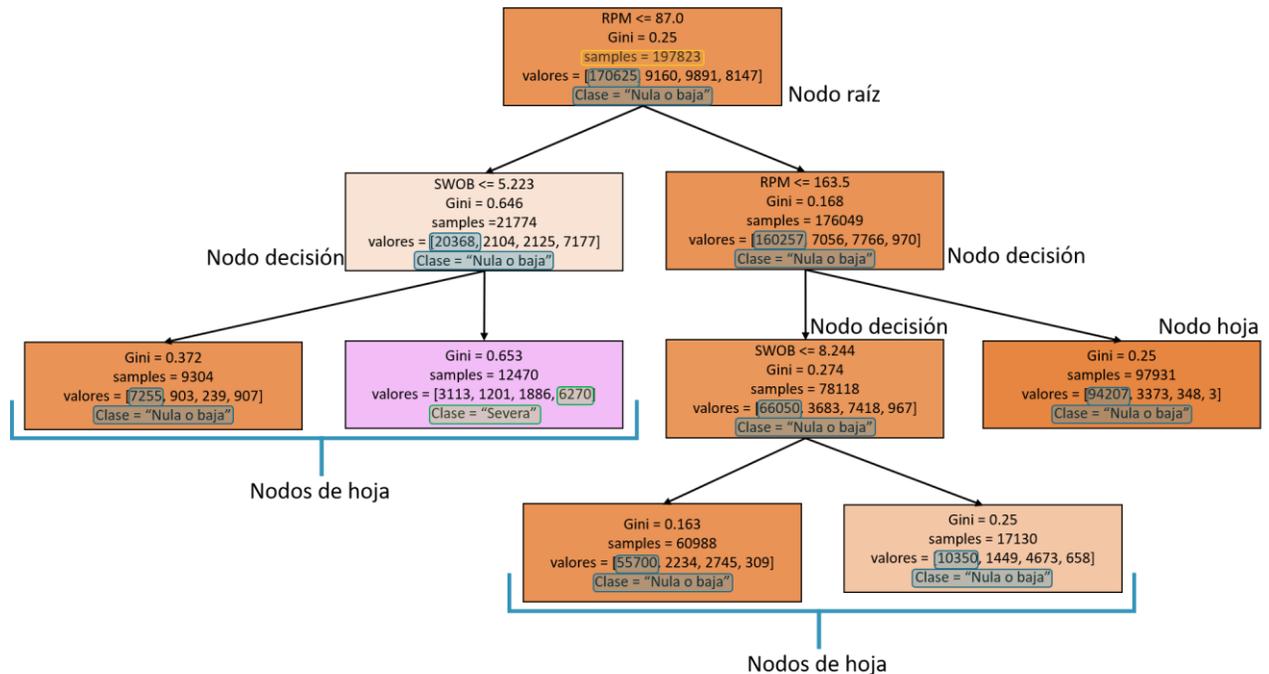


Figura 5.3.1.0.1 : Árbol de decisión resultante

En esta Figura 5.3.1.0.1 se muestra la solución grafica del modelo, un árbol de decisión como el mostrado y descrito anteriormente en la Figura 3.3.1.1 pudiendo encuadrar componentes de relevancia; en el recuadro amarillo del nodo raíz, se muestra el total de datos de entrenamiento, siendo la primera pregunta **RPM** menor o igual a 87, de donde se obtiene en la partición mostrada en el recuadro azul, 170,625 elementos que corresponden a una severidad nula o baja; en la siguiente partición o profundidad 1, la pregunta del primer nodo de decisión hace referencia al **SWOB** menor a 5.2, habiendo nuevamente una mayoría de elementos de clase nula o baja, sin embargo, de este nodo de decisión, el modelo es capaz de asignar una clasificación nula o baja y una severa, en color naranja y rosa respectivamente, en este último nodo existen 6,270 ocurrencias, sobrepasando a los 3113 eventos nulos; desplazándonos a la derecha en

el nodo de decisión 2 de la profundidad 1, se aprecia un nodo de decisión y otro de hoja a la misma profundidad, teniendo el primero una menor cantidad de elementos o *simples* en la clase nula o baja, por lo que al realizar una partición de este solo se obtienen estos eventos, y aunado a que en los argumentos establecimos 5 como máximo número de nodos de hoja, el proceso trunca.

Al aplicar comandos básicos de Python y la función cuyo propósito es determinar la precisión del modelo mediante la comparación de elementos predichos y originales, se obtiene el siguiente bloque de código y bloque de resultado.

```
print("Precisión del modelo:",round((metrics.accuracy_score(y_test, y_pred_test)*100),4,"%")  
Precisión del modelo: 87.498 %
```

5.3.2 Interpretación de resultados

Una ventaja de este algoritmo es la obtención de la relevancia de los parámetros, este nos indica cuál de los parámetros de predicción, el **RPM** y el **SWOB** tienen una mayor relevancia para predecir la variable seleccionada, que en este caso es la severidad categórica de **S&S**, la Figura 5.3.2.1 muestra que el peso sobre la barrena resulta más importante que las revoluciones para simular este comportamiento a partir del modelo construido.

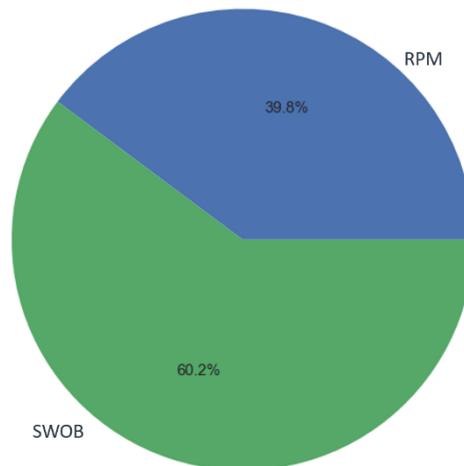


Figura 5.3.2.1 : Relevancia parámetros para árbol de decisión

Al analizar la Figura 5.3.1.0.1 el árbol de decisión resultante el cual resulta de mucho valor didáctico, como ya se describió con anterioridad, únicamente es capaz de predecir las clases nula o baja y de **S&S** severo, en el diagrama de flujo de la Figura 3.1.2 se establece una toma de decisión, sobre la modificación de los parámetros o también de los argumentos utilizados para crear el *DecisionTreeClassifier*, por lo que un análisis más a detalle es necesario, el cual puede ser llevado a cabo a través de la matriz de confusión que se muestra en la Figura 5.3.2.2.

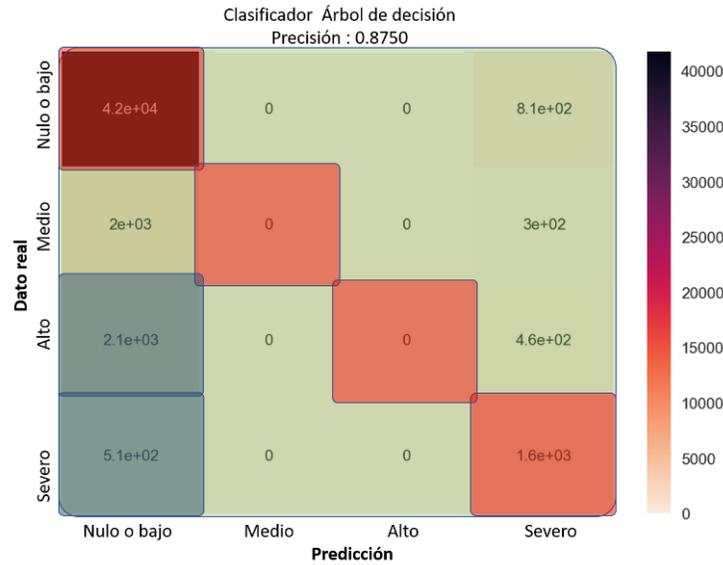


Figura 5.3.2.2 : Matriz de confusión para árbol de decisión

Esta herramienta es útil para determinar el desempeño de problemas de clasificación binaria, o multi-clase, que es el problema que se tiene, aproximar cuatro variables categóricas; en la Figura 5.3.2.2, se comparan los valores reales (eje y) contra los valores obtenidos a partir de la predicción (eje x), por lo que, en los recuadros naranja o diagonal principal, se muestra el número de datos cuya predicción de clase fue correcta, es decir, se tuvieron 1,600 eventos severos y 42,000 eventos nulos o bajos correctamente clasificados; las predicciones erróneas, se encuentran en la diagonal superior e inferior o en los recuadros verdes.

Como ya habíamos comentado a partir de la Figura 5.3.1.0.1, este modelo solo predijo eventos nulos y severos, por lo que la columna en la matriz correspondiente a las clases de predicción medio y alto se encuentran en cero; por otro lado, en los recuadros azules, es de suma importancia notar, que este modelo asigna como evento nulo o bajo a 510 eventos severos, y a 2,100 eventos altos, estos valores resultan los más relevantes para determinar el cambio de argumentos, sin embargo dado que este algoritmo se realizó como una forma de ejemplificación y contextualización para el modelo *Random Forest*, este modelo resulto finalizado.

5.4 Construcción de Random Forest

Una vez explicado el comportamiento de un árbol de decisión, se decidió aplicar el *Random Forest* un método de ensamble el cual, como ya se vio en el tema 3.4.1 *Random Forest*, es un conjunto de árboles de decisión para realizar una determinada predicción; para este modelo se decidieron ocupar los predictores direccionales, junto con los parámetros de perforación ya descritos en capítulos anteriores.



Figura 5.4.1 : Coeficientes de Pearson para RF

En esta Figura 5.4.1, puede apreciarse que las variables de predicción no tienen una colinealidad entre ellas, a pesar de que, en los recuadros rojos, el **TQA** o torque tiene una correlación con la profundidad o **DEPT**, con un valor de 0.8; así mismo con la presión de bombeo o **SPPA**, sin embargo, por motivos académicos, se mantendrán en el modelo; la profundidad también posee una relación muy directa con la inclinación o **INCL**, aunque este parámetro se mantendrá también.

```

Predictores:
  ['AZI' 'BPOS' 'BVEL' 'DEPT' 'DLS' 'GR' 'HKLD']
  ['INCL' 'ROP5' 'RPM' 'SPPA' 'SWOB' 'TFLO' 'TQA']
Tamaños:
  Training:
    X: (8476, 14)
    Y: (8476,)
  Testing:
    X: (2119, 14)
    Y: (2119,)
    
```

Figura 5.4.2 : Sets de datos para RF

La Figura 5.4.2 muestra los predictores finales para predecir la severidad categórica mostrada en la Tabla 6.

5.4.1 Parámetros del modelo

La biblioteca de *Scikit-learn* ofrece la función de “*RandomForestClassifier*”, la cual contiene el algoritmo para realizar un problema de clasificación aplicando este modelo de ensamble, por lo que ejecutando la siguiente línea de código se estaría construyendo dicho modelo.

```
▶ %%time
rf_direccional = RandomForestClassifier(n_estimators=50,max_depth=None, min_samples_split=2,
                                     min_samples_leaf=2, max_features= "auto", bootstrap=True,
                                     n_jobs=-1,random_state=2).fit(x_train,y_train)
```

Los argumentos dados son los siguientes:

- *n_estimators*
Número de árboles en el modelo de **RF**.
- *max_depth*
Máximo nodos de decisión por árbol, el predefinido es hasta que el valor **Gini** sea cero.
- *min_samples_split*
Número de “samples” o datos requeridos para partir un nodo, se estableció 4 por el número de severidades categóricas.
- *min_samples_leaf*
Número de “samples” requeridos por hoja.
- *max_features*
Número de predictores considerados en cada partición.
- *n_jobs*
Número de núcleos de procesamiento empleados para el modelo, todos los disponibles.

Si se pudiera visualizar los árboles del modelo **RF**, como se muestra en la Figura 3.3.1.1, se obtendría la siguiente Figura 5.4.1.1, donde a partir de los argumentos dados al algoritmo, se crearán 50 árboles de decisión para la predicción de las variables categóricas del “S&S”.

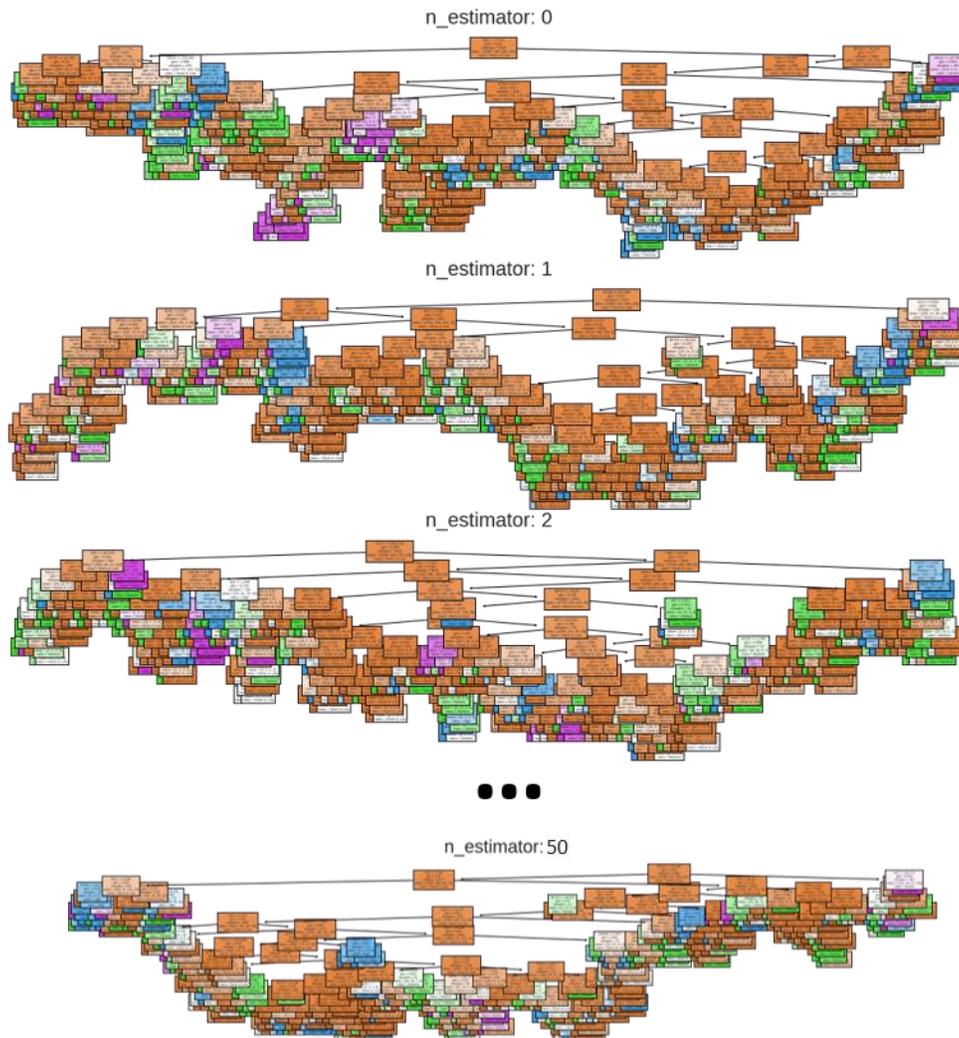


Figura 5.4.1.1 : RF gráfico

De esta forma podemos obtener la precisión de nuestro modelo, comparando las variables categóricas del *testing set*, predichas (`y_pred`) y originales (`y_test`).

```
print("Precisión del modelo:", round((accuracy_score(y_test, y_pred)*100),4), "%")
```

Precisión del modelo: 97.3572 %

Este valor nos indica el porcentaje de datos correctos que obtuvo el modelo, es decir, el algoritmo "rf_direccional" fue entrenado con 8,476 datos distribuidos en 14 parámetros de predicción, estos del *Training set*, y ocupando esos datos para predecir los valores categóricos.

5.4.2 Interpretación de resultados

Resulta de mucha conveniencia tener en cuenta que este modelo se trata de un “Data Driven Model”, o, un modelo construido a partir de datos, por lo que, no necesariamente vamos a observar la relevancia que la bibliografía pudiera aportarnos.

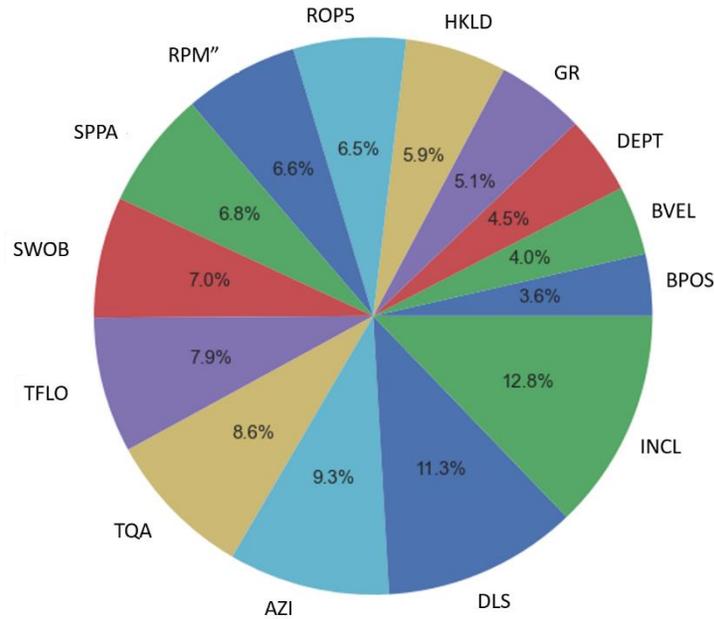


Figura 5.4.2.1 : Relevancia parámetros para modelo Random Forest

En la Figura 5.4.2.1, se muestra que la mayor relevancia para predecir la severidad del **S&S** a partir del modelo de predicción, son los parámetros direccionales, la inclinación, el **DLS** y el azimut, para luego los parámetros de perforación como el torque y el flujo total de lodo al pozo, quedando las revoluciones en 8va posición.

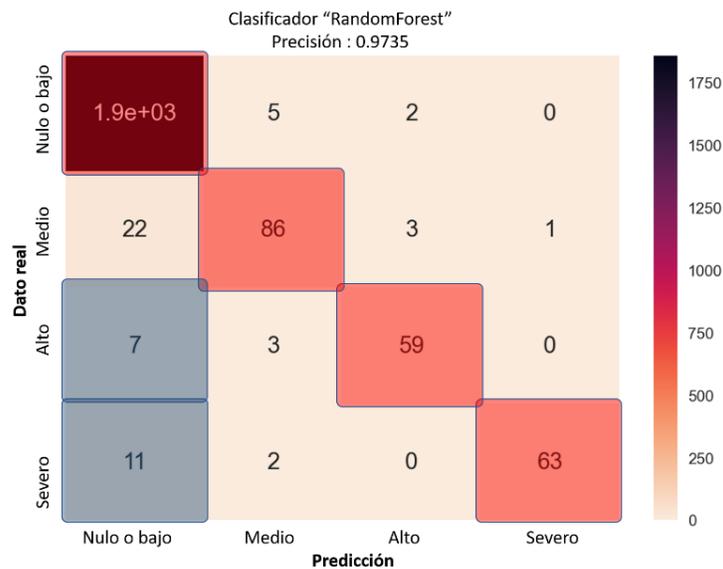


Figura 5.4.2.2 : Matriz de confusión para RF

Al analizar la matriz de confusión mostrada en la Figura 5.4.2.2 en su diagonal principal se encuentran las clasificaciones correctas, donde podemos destacar que el modelo predijo correctamente 63 elementos cuya clasificación corresponde a eventos severos, sin embargo, es importante puntualizar que en los recuadros azules son mostradas las siguientes aseveraciones:

- El modelo clasifico 11 elementos como nulo o bajo, cuando en la realidad los elementos eran eventos severos.
- Se clasificaron 7 elementos como nulos o bajo, cuando son eventos altos de **S&S**.

Estos hechos resultan críticos, ya que, la finalidad de estos métodos de predicción, es asignar correctamente los eventos severos, sin embargo, tomando en cuenta que el tamaño del *testing set* es de 2119, estos 11 elementos mal clasificados representan el 0.5 de estos, por lo que se seguirá utilizando este modelo para la aplicación a un pozo nuevo o *blind set*.

5.5 Aplicación de modelo de predicción

A partir de la Figura 3.1.2, el último procedimiento para la elaboración y aplicación de un modelo de **ML** es aplicarlo a un determinado conjunto de datos o *Blind set*, el cual debe ser un conjunto de datos ajeno a los sets de entrenamiento y *testing*, este modelo "rf_direccional" fue el visto en el subtema 5.4 Construcción de Random Forest y con este predecir los eventos de **S&S** del set adicional, el cual será un pozo.

Sin embargo, previo a la elaboración de este trabajo, se construyó un modelo de *Random Forest* a partir de la base de datos denominada "pre-acople de datos direccionales" descrita en el subtema 5.2 Limpieza y construcción de base de datos, este modelo llamado "rf" obtuvo una precisión de 99.8% y una matriz como se muestra en la Figura 5.5.1.

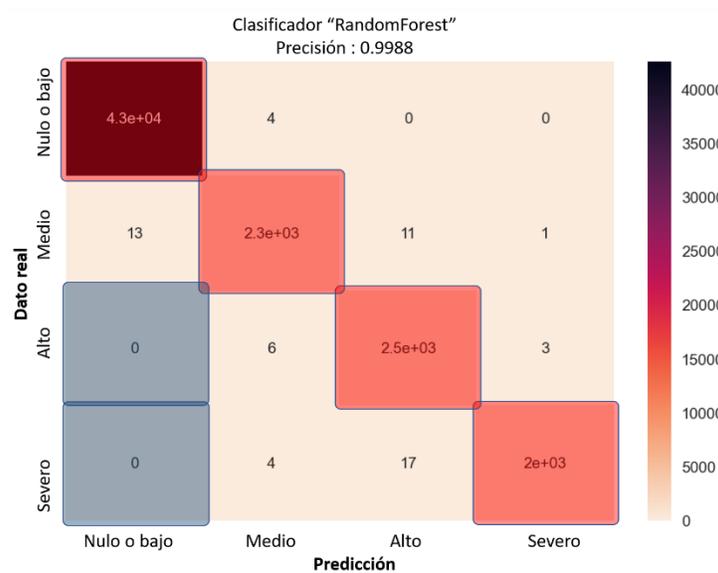


Figura 5.5.1 : Matriz de confusión para "rf"

Esta matriz muestra una respuesta ideal de cualquier modelo, en la diagonal principal, se tienen valores altos, mientras que en las diagonales superiores e inferiores se muestra un total de 53 elementos mal clasificados, aunado a esto, en los recuadros azul, locaciones que muestran las ocurrencias severas clasificadas como nulo o bajo por el modelo, se encuentran vacías.

Resumiendo ambos modelos se puede construir la Tabla 9.

Tabla 9 : Resumen de modelos

Data set	Nombre del modelo	Predictores	Dimensiones		Precisión
			Set Entrenamiento	Testing Set	
			Y	Y _{pred}	
Pre-Acople	rf	[BPOS, BVEL, DEPT, GR, HKLD, ROP5, RPM, SPPA, SWOB, TFLO, TQA]	197,823	49,456	99.883%
Post-Acople	rf_direccional	[BPOS, BVEL, DEPT, GR, HKLD, ROP5, RPM, SPPA, SWOB, TFLO, TQA, INCL, AZI, DLS]	8,476	2,119	97.352%

Así mismo, resulta de mucha importancia notar las magnitudes de datos de entrenamiento, que existen entre los modelos, el primero tiene cerca de 100mil datos para entrenar la asignación de las variables categóricas del **S&S**, mientras que el modelo con parámetros direccionales tiene menos del 5% de estos.

En ambos modelos se probó la predicción de severidades categóricas del **S&S** a través de los registros de perforación y *survey* del pozo **F-15A**, siendo este un pozo multilateral que parte del **F15**, como se muestra en la Figura 5.5.2, siendo aledaño a los pozos de entrenamiento.

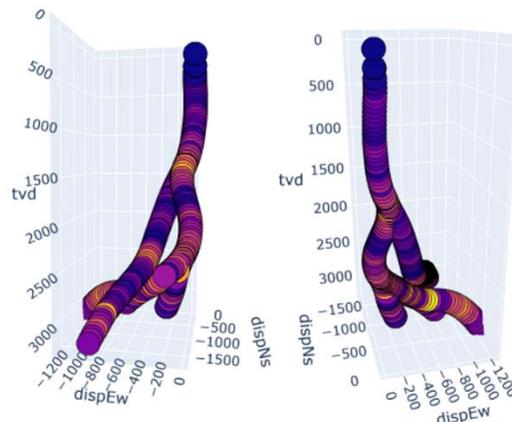


Figura 5.5.2 : Pozo multilateral F15

Los datos disponibles para este pozo **F-15A** son a partir de la tubería de revestimiento de 20”, o corridas inferiores a la barrena de 17 ½”, como se muestra en la Figura 5.5.3, mientras que en el plan de perforación es mostrado en la Figura 5.5.4 es posible apreciar que este pozo atraviesa las formaciones que se han ido nombrando y analizando con anterioridad, y en el mapa de contorno su localización en función de los pozos de entrenamiento.

```

OPERATOR      : STATOILHYDRO ASA
WELL          : 15/9-F-15
WELLBORE     : 15/9-F-15 A
FIELD        : VOLVE
RIG          : MERSK INSPIRER
COUNTRY      : NORWAY
DRILL PERMIT#: 2937-P

MWD DATA:

data:\MWD\BIT_RUNS\WL_RAW_BHPR-GR-MECH_TIME_MWD_1.LAS
-----
Hole   Main      Service   Pass Direction Interval   Date      Run   Original
section Services Company    (Up/Down)   (meters)  start    start  file name
-----
17 1/2" DRILL.MECHANICS SCHLUMBERGER DOWN/RealTime TIME INDEXED 09-DEC-08 7 15_9_f_15_175in_dm_time.las
Related plots:
data:\MWD\BIT_RUNS\WL_RAW_BHPR-GR-MECH_TIME_MWD_PLOT_1.PDF (Original file name: drillmech_2inhr_time_175in.pds)

data:\MWD\BIT_RUNS\WL_RAW_BHPR-GR-MECH_TIME_MWD_2.LAS
-----
Hole   Main      Service   Pass Direction Interval   Date      Run   Original
section Services Company    (Up/Down)   (meters)  start    start  file name
-----
8 1/2" DRILL.MECHANICS SCHLUMBERGER DOWN/RealTime TIME INDEXED 28-DEC-08 9 run9_mud_log_time.las
Related plots:
data:\MWD\WLC_RAW_BHPR-GR-MECH_TIME_MWD_PLOT_1.PDS (Original file name: mud_log_time.pds)

data:\MWD\BIT_RUNS\WL_RAW_BHPR-GR-MECH_TIME_MWD_3.LAS
-----
Hole   Main      Service   Pass Direction Interval   Date      Run   Original
section Services Company    (Up/Down)   (meters)  start    start  file name
-----
8 1/2" DRILL.MECHANICS SCHLUMBERGER DOWN/RealTime TIME INDEXED 02-JAN-09 10 run10_mud_log_time.las
Related plots:
data:\MWD\WLC_RAW_BHPR-GR-MECH_TIME_MWD_PLOT_1.PDS (Original file name: mud_log_time.pds)
    
```

Figura 5.5.3 : Registros MWD disponibles para F15A

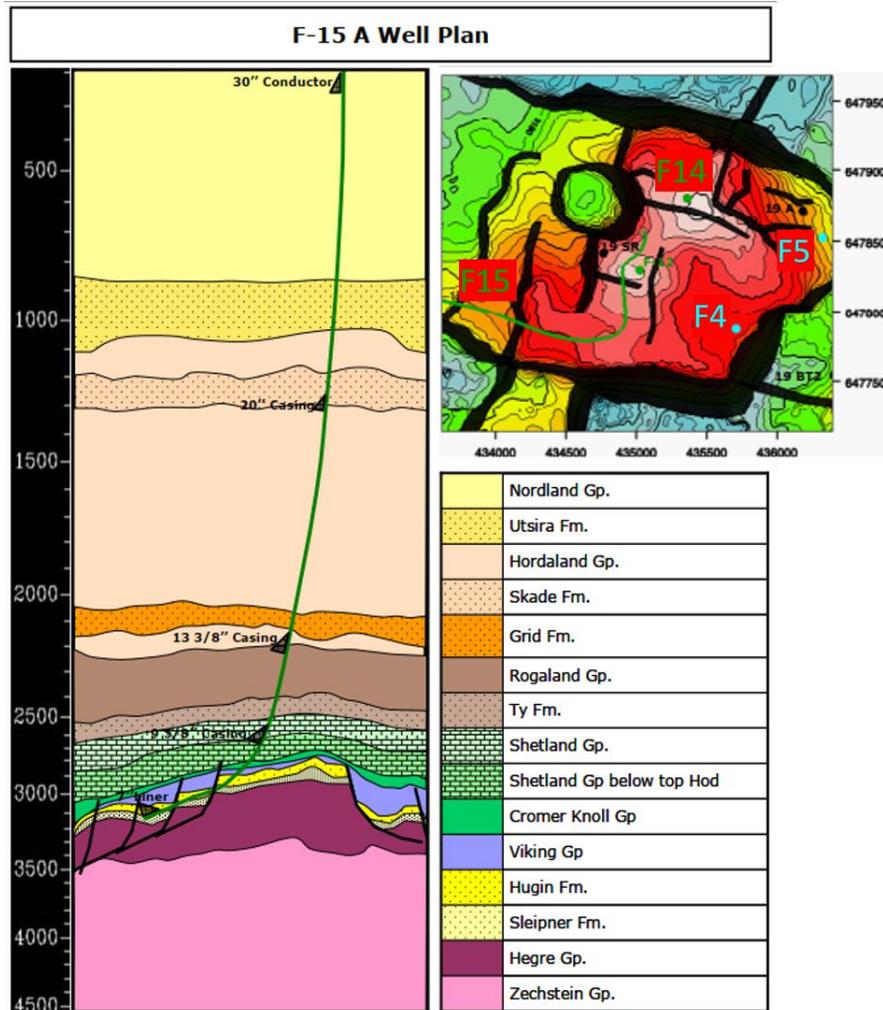


Figura 5.5.4 : Plan Pozo F15A

Al aplicar los modelos de predicción se obtiene la siguiente Tabla 10, donde se muestran los datos dados al modelo para predecir la severidad categórica del **S&S**.

Tabla 10 : Aplicación de modelos Random Forest

Modelo	Datos	Precisión
rf	53 505	94.5388 %
rf_direccional	1 089	71.3499 %

El desempeño del modelo de "rf_direccional" es deficiente, esto puede deberse al reducido número de datos de entrenamiento condiciones dadas por la frecuencia de la toma de *survey's*, en contraste con el modelo "rf", el cual posee las condiciones mecánicas de la perforación tiene una precisión muy alta la cual se muestra en la Figura 5.5.5.

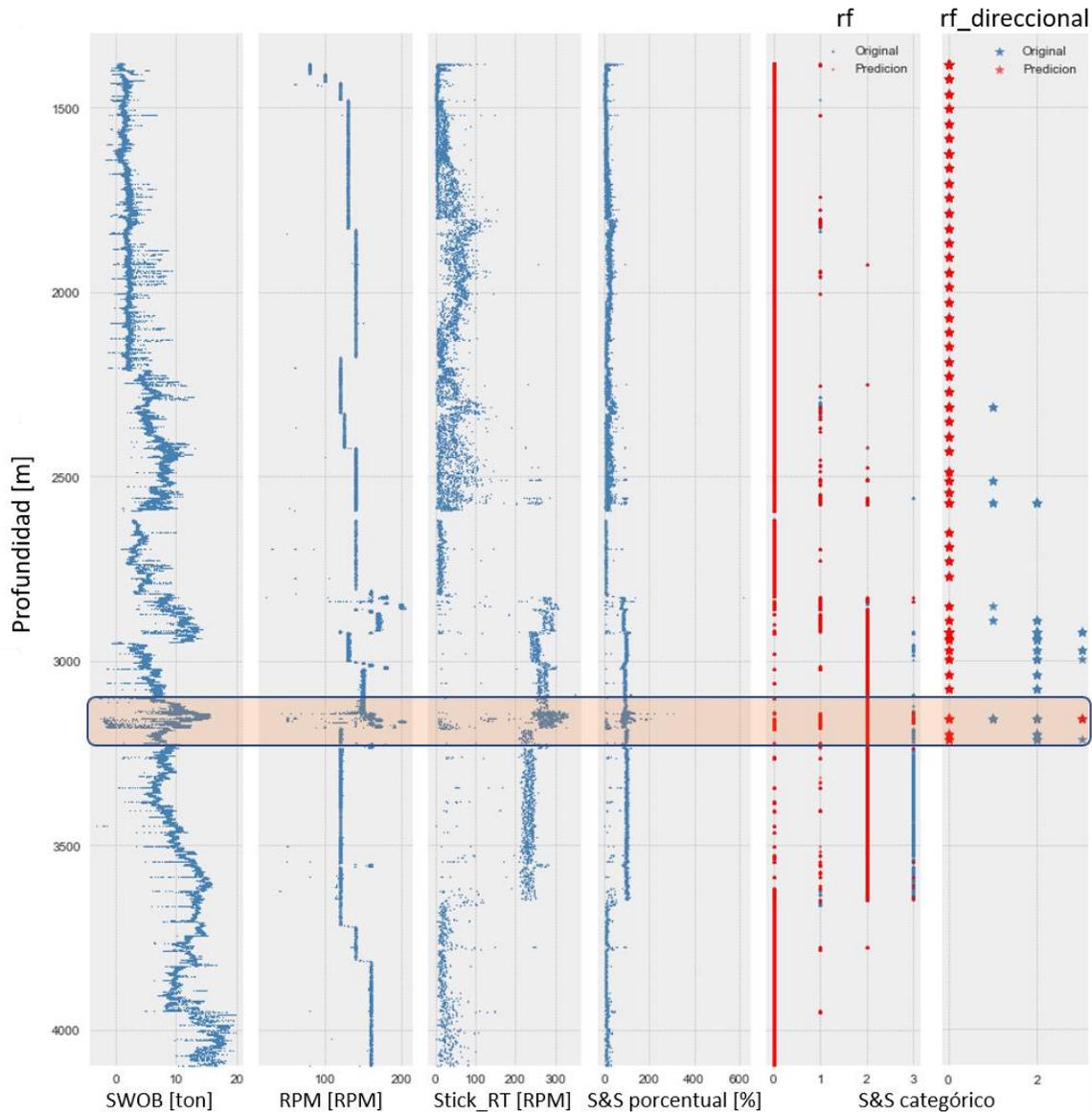


Figura 5.5.5 : Comparación de modelos

A partir de esta figura es importante mencionar, que la predicción de las severidades categóricas de nula hasta alta se encuentra bien definidas en el modelo “rf”, mientras que el direccional únicamente se predicen los eventos nulos y un evento severo; en el recuadro naranja, es fácil reconocer un evento severo mostrado interpretado en el carril de S&S porcentual, y que el modelo lo predijo satisfactoriamente, sin embargo es importante recordar en algunos caos los eventos severos estaban precedidos por altos, por lo que con el desempeño mostrado, mitigando estos se evitarían los severos.

Capítulo 6. Conclusiones

6.1 Stick & Slip

A partir del análisis realizado con base en los reportes de perforación disponibles, se pueden realizar las siguientes puntualizaciones:

- Uno de los hallazgos de mayor relevancia lo aporta la Figura 4.3.4.7 : Severidad S&S categórica por pozo, la cual nos muestra que los eventos severos de **S&S** están comúnmente precedidos por eventos altos y, a su vez, estos por eventos medios, lo que nos indica que al tener una detección oportuna en la generación de eventos, podríamos disminuir las incidencias de eventos severos, por lo que contar con una herramienta capaz de predecir eventos altos, es una ventaja para evitar el incremento de estos y por ende daños y **NPT**.
- En la perforación del pozo **F10**, la Figura 4.3.2.1 : Corrida 1, Barrena 8 ½" pozo F10 muestra un **ROP** bajo, así como severidades de **S&S** superiores al 100%, indicando la presencia de vibraciones severas, y que al correlacionar con la columna geológica, se encuentra en la formación *Tor*, la cual es una caliza dura con trazas de lutita arcillosa y que ha causado eventos severos por su naturaleza, se decidió seguir la perforación, resultando en 200 [m] en tres días, y con un **ROP** que no supero las 7.5 [m/hr]. Esto resulta muy importante de analizar, ya que, al cambiar la barrena, mejoro significativamente la penetración, y el intervalo de **S&S** se mantuvo en valores inferiores a 100%; por lo que en el diseño de la sarta habrá de tomarse en cuenta el comportamiento de las barrenas con la formación y vibraciones que estas generen a partir de la evidencia de otros pozos.
- En la Figura 4.3.3.1 : Barrena 12 ¼" pozo F5, al tener una caída en el **ROP** el perforador optó por incrementar el **SWOB** hecho que se identifica con facilidad en el registro, correlacionando geológicamente obtenemos nuevamente la formación *Tor*, por lo que la pregunta es obvia, ¿Qué genero la caída tan brusca de "ROP"?
La respuesta dista de obviedad, en el registro es posible ver una disminución de **RPM** que pudo haber acrecentado el fenómeno vibratorio que también se presentó de forma drástica, así como la interacción entre la barrena seleccionada y las propiedades petrofísicas de la formación, por lo que un incremento de peso más que soluciones aporó problemas en forma de severidades cercanas a 150%; otro caso similar es la Figura 4.3.1.2 : Barrena 8 ½" pozo F14, donde es posible apreciar un cambio repentino de peso para compensar una caída de **ROP**, momentos después comienzan las severidades superiores a 100%, que mediante remediaciones como retirar de fondo mitigaron y las disminuyeron a 50%.

-

- Los eventos de remediación podrían interpretarse de mejor manera, teniendo evidencia de un *log* de vibraciones, para que de esta forma puedan analizarse y modelarse si es posible estos eventos, en la Figura 4.3.3.2 : Corrida 1, Barrena 8 ½" pozo F5, se expresa un periodo cercano a una hora de pausa completa debido a vibraciones severas presentes en las formaciones *Blodoeks* y *Roedby*, una vez reanudada la perforación, la formación *Draupne*, ya había sido perforada en el pozo **F4** con parámetros inferiores en peso y superiores en revoluciones con pocos eventos de **S&S**, recalcando la importancia de diseñar a partir de parámetros ya conocidos, los siguientes pozos.
- Los parámetros direccionales resultan de una complejidad particular, ya que al cotejar las secciones de perforación con sus respectivas condiciones direccionales, los gráficos de la Figura 4.3.4.5 : Parámetros direccionales Pozo F4 y la Figura 4.3.1.3 : Parámetros direccionales Pozo F14, muestran que al existir un incremento constante de inclinación se tienen secciones con altas incidencias de **S&S**, esto aunado a la Figura 4.3.4.9 : Severidad S&S categórica para Inclinación, se podría resumir que, al existir incrementos constantes de 30 [°] las incidencias de eventos severos son altas; mientras que las ocurrencias en función del azimut, no son concluyentes ni se diferencia algún comportamiento constante, sin embargo al analizar los grados de azimut puntuales por medio de la Figura 4.3.4.8 : Severidad S&S categórica para Azimut el intervalo de 140-160 [°] muestra incidencias relevantes de vibraciones; el **DLS** al ser una razón de cambio entre los dos parámetros anteriores, muestra discrepancias en cuanto a ocurrencias de **S&S** como se muestra en la Figura 4.3.4.10 : Severidad S&S categórica para DLS.
- Probablemente la Figura 4.3.4.11 : Pairplot de S&S categórico, resulte ser la más abstracta y complicada de interpretar, con las menciones de que estas relaciones pueden no tener una causalidad física con el fenómeno de **S&S**, por lo que se deberán estudiar más a fondo, sin embargo por fines académicos se mencionan los siguiente, el inciso (e) muestra la incidencia de eventos severos en profundidades de 3,000[m] y con gastos inferiores a 2,500[lpm], esto podría ser un indicativo de un deficiente acarreo de recortes, y que estos podrían repercutir en la disminución de rotaciones de la sarta ya que estos estarían localizados en el espacio muerto de los componentes, esta inferencia podría corroborarse con el **ECD**; el inciso (g) muestra que los eventos severos se concentran en ángulos de azimut superiores a 120 y en profundidades superiores a los 2,500 mD; el inciso (l) muestra una probable ventana operativa, cuando se tienen valores de **DLS** entre 0.025 y 0.1[°] cuando se tienen valores de **RPM** mayores a 60; por último el inciso (o) que muestra el ángulo de diseño o desviación **INCL**, el cual muestra los eventos severos en cargas inferiores a 10[ton] y ángulos por debajo de 30[°].

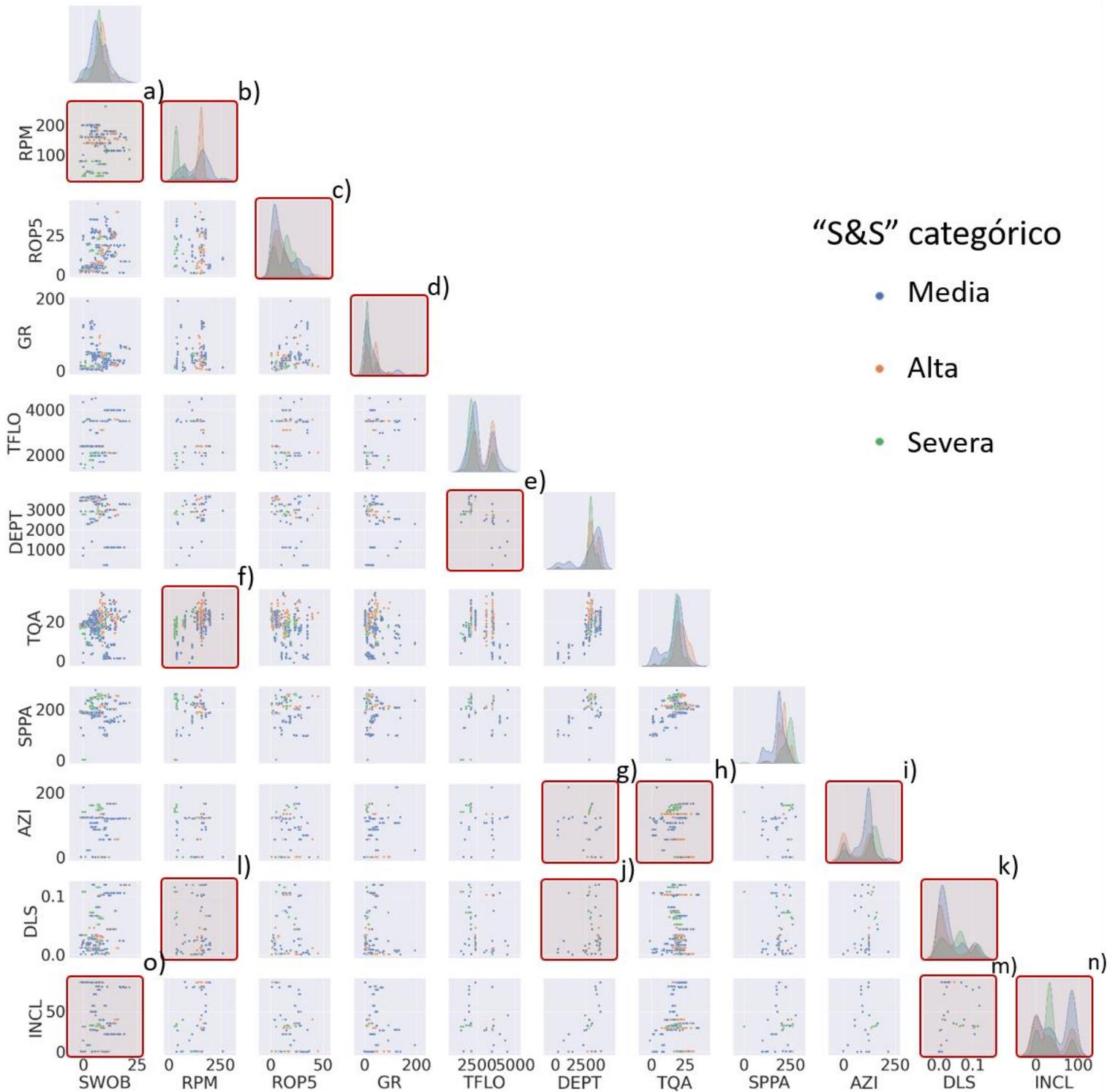


Figura 4.3.4.11 : Pairplot de S&S categórico

6.2 Uso de datos

Las conclusiones de este rubro se enfocan en el manejo, obtención y filtrado de datos, actividad de suma importancia para el modelado de *Data Driven Models*, por lo que se podrían concluir lo siguiente:

- Los datos que se pueden recabar para construir los modelos de **ML** pueden venir en múltiples formatos y presentaciones, dependiendo de la metodología de medición, de la etapa en la cadena de valor, y en muchas ocasiones, dependiendo de la empresa que las realiza, complicando de esta forma su obtención, por lo que se sugiere una estandarización, en esfuerzo multidisciplinario para establecer una norma sobre la disposición de datos, para que de esta forma, los modelos contengan la mayor cantidad de datos útiles posibles.
- El punto anterior solo puede ser valido existiendo personal calificado con conocimientos en los rubros de programación, análisis de datos y con nociones básicas de **ML** para que este pueda cerciorarse de la correcta agrupación de datos, la Figura 4.3.2.2 : Corrida 2, Barrena 8 ½” pozo F10, es ejemplo de esto, ya que esta corrida carece de medición de rayos gamma, probablemente por motivos económicos o imperfectos en la herramienta, desafortunadamente no se expresa en el **FWR**, sin embargo, la etapa de barrena era meritoria de la recabación de la mayor cantidad de información posible, pero que en este caso, debió de suprimirse por su ausencia.
- Los eventos de severos, como se vio en la Figura 4.3.4.7 : Severidad S&S categórica por pozo, se encuentran concentrados mas no aislados en grandes profundidades, también existen vibraciones en segmentos verticales como lo indica la Figura 4.3.4.9 : Severidad S&S categórica para Inclinación, por lo que el tener datos de secciones más someras resultaría en un beneficio para la construcción de un modelo capaz de predecir eventos en múltiples condiciones, en la Figura 5.1.1 : Archivos LAS de pozo F14 se muestra la disponibilidad de las secciones inferiores a 17 ½”.
- Las diversas visualizaciones de datos resultan muy importantes ya que, dependiendo de las magnitudes y el comportamiento de los parámetros, deberán emplearse diversas variaciones a los gráficos, en el análisis de la Figura 4.3.4.8 : Severidad S&S categórica para Azimut y de la Figura 4.3.4.10 : Severidad S&S categórica para DLS, resultaba de mayor utilidad visualizar el histograma para determinar los ángulos puntales, ya que el gráfico de violín posee el *kernel* de densidad que puede confundir en la interpretación, ya que se mostrarían valores negativos, los cuales no tienen un sustento físico.
- Las bibliotecas de *Seaborn* para análisis estadístico, *Pandas* para el manejo de datos y *Matplotlib* para visualizaciones gráficas resultan de mucha utilidad para los flujos de trabajo de análisis y visualización como el que se presentó en el desarrollo de esta tesis, y que pueden ser visualizados en el Anexo 2. Códigos Elementales.
- Existen una gran cantidad de datos que son recabados a lo largo de toda la cadena de valor, que pueden ser parte toral de los modelos basados en datos o como se muestra en la Figura 3.2.6 : PDA (Mohaghegh, 2020. Modificado)

6.3 Aplicación del algoritmo RF

Los modelos realizados muestran una clara distinción, la cantidad de datos de entrenamiento, así como la cantidad de datos presentes en el *Blind Set*, sin embargo, se pueden realizar las siguientes menciones:

- El *Random Forest* permite algo muy útil para el ingeniero petrolero, la construcción de modelos a partir de una base de datos que contienen valores cuyas unidades campo, tienen magnitudes diferentes, desde $[g/cm^3]$ para la densidad de un lodo, hasta $[ton]$ para el peso del gancho en el sistema de izaje, estas cantidades pueden ser manejadas sin necesidad de una normalización por el **RF**, es decir, sin la necesidad de un preprocesamiento de datos.
- A partir de Figura 3.2.6 : PDA (Mohaghegh, 2020. Modificado), se pueden construir modelos de regresión y clasificación, siendo estos muy útiles para determinar las severidades de movimientos vibratorios en la sarta, ya que estos se clasifican para su comprensión y mitigación.
- Ocupando el principio de Pareto, podría establecerse que, el 80% del tiempo fue invertido para los procedimientos 1-4 de la Figura 3.1.2 : Diagrama de flujo para ML (Belyadi, 2021. Modificado), mientras que el restante 20% se utilizó para la aplicación de los algoritmos, su interpretación y modificación de parámetros, procedimientos 5-8, por lo que la creación de flujos de trabajo facilitarían la implementación de estas técnicas en las actividades de campo.
- A partir de la Tabla 9 : Resumen de modelos, es posible determinar que la cantidad de datos en un modelo basado en datos es sumamente importante, el modelo "rf" tuvo más de 190 mil datos para entrenar y crear 50 árboles de decisión obteniendo una Figura 5.5.1 : Matriz de confusión para "rf", envidiable, mientras que el modelo "rf_direccional" tuvo menos de 10 mil para intentar asignar cada valor a la clase correspondiente, sin embargo, este mostró deficiencias claras en la Figura 5.4.2.2 : Matriz de confusión para RF, y esto es reflejado en el desempeño al momento de predecir un nuevo pozo a pesar de que este sea aledaño, perforara la misma columna geológica y con condiciones semejantes de infraestructura, el desempeño del modelo "rf" es claro en la Figura 5.5.5 : Comparación de modelos, prediciendo un 94% de las ocurrencias, mientras que el "rf_direccional" un 71% como se muestra en la Tabla 10 : Aplicación de modelos Random Forest; visualmente, esto se reflejaría como el registro de perforación, el primer modelo capaz de predecir todo el intervalo de severidades, mientras que el segundo, únicamente los extremos.
Sin embargo, es importante mencionar la zona comprendida entre 3,200 – 3,600 mD tiene una deficiencia en la predicción de eventos severos, aunque como ya se mencionó en el subtema 6.1 *Stick & Slip*, de existir eventos altos, la tarea de mitigación adquiere relevancia para evitar que estos progresen a severos.
- Estos algoritmos requieren antecedentes conceptuales complejos, pero su aplicación resulta muy práctica, sin embargo, es una herramienta que va mejorando si se le permite entrenar con nuevos datos, por ende, una herramienta que le agregara valor a la actividad del ingeniero petrolero.

Oportunidades de Implementación

1. Acoplar datos de otros campos para visualizar el desempeño con otras litologías y revisar su repercusión en los modelos.
2. Agregar variables categóricas como columnas, tipo de barrena, tipo de lodo, grado de tubería de perforación, así como otras condiciones de perforación para aumentar el número de predictores y ver su relación con los eventos de **S&S**.
3. Realizar una base de datos a partir de los registros contenidos en los archivos **WITSML** ya que estos son tomados cada 5[sec] y existen más registros de vibraciones.
4. Realizar una base de datos con datos direccionales interpolados, para que en el proceso de acople con los registros **MWD** estos no sufran pérdidas, tomando en cuenta las repercusiones de datos sintéticos en el subsecuente modelo de desarrollado.
5. Probar algún pozo mexicano, para determinar su grado de similitud en la predicción de estos eventos.
6. Acoplar datos más precisos sobre la litología, en el presente solo se utilizó el **GR** como indicador litológico, pero de estar disponibles, los registros de **LWD**, acústicos o eléctricos, por mencionar algunos, aportarían información de utilidad.
7. Analizar la factibilidad de tomas más continuas de *survey's* y de esta forma tener una base de datos más robusta utilizando el flujo de trabajo aquí mostrado.
8. Utilizar los registros de **ECD** los cuales pueden aportar información sobre la eficiencia de acarreo de recortes y ver si este valor repercute en la generación de eventos de **S&S**.
9. Mediante el mnemónico **INCL_CONT_RT**, el cual es una medición indirecta de la inclinación de la barrena en el proceso de perforación, realizar un modelo de **ML** para ver su similitud con el *survey* de inclinación y su posible apoyo en la predicción.
10. Acoplar una variable categórica cuyo propósito sea indicar si hubo una acción correctiva, para que de esta forma se pueda establecer cuales eventos severos fueron los reportados, y que acción correctiva fue aplicada.
11. Realizar un algoritmo robusto capaz de asignar a detalle la formación correspondiente al dato de perforación y de esta forma tener la formación como una variable categórica en el modelo.
12. La evidencia física, sobre las condiciones finales de la barrena en cada etapa de perforación, para determinar de forma más tangible las repercusiones de los eventos antes descritos.
13. Establecer una representación gráfica más practica para mostrar los parámetros direccionales, como la sección vertical, **AZI**, **INCL** y **DLS**.

Bibliografía

- Adams, N. J. (1985). *Drilling Engineering*. USA: PenWellBooks.
- Amat Joaquín, R. (Octubre de 2020). *cienciadedatos.net*. Obtenido de *Árboles de decisión con Python: regresión y clasificación*:
https://www.cienciadedatos.net/documentos/py07_arboles_decision_python.html
- Amat Rodrigo, J. (Octubre de 2020). *cienciadedatos.net*. Obtenido de *Random Forest con Python*:
https://www.cienciadedatos.net/documentos/py08_random_forest_python.html
- Baker, R. (2001). *A Primer of Oilwell Drilling*. Houston, USA: Petroleum Extension Service.
- Barberii, E. E. (1998). *El Pozo Ilustrado*. Venezuela: Fondo Editorial del Centro Internacional de Educación y Desarrollo.
- Belyadi, H., & Haghighat, A. (2021). *Machine Learning Guide for Oil and Gas Using Python*. USA: Elsevier.
- Belyadi, H., Fathi, E., & Belyadi, F. (2019). *Hydraulic Fracturing in Unconventional Reservoirs*. USA: Elsevier.
- Brownlee, J. (2016). *Master Machine Learning Algorithms*. Australia.
- Guangjian Dong, Ping Chen, "A Review of the Evaluation, Control, and Application Technologies for Drill String Vibrations and Shocks in Oil and Gas Well", *Shock and Vibration*, vol. 2016, Article ID 7418635, 34 pages, 2016. <https://doi.org/10.1155/2016/7418635>
- Gupta, Soumya, Chatar, Crispin, and José R. Celaya. "Machine Learning Lessons Learnt in Stick-Slip Prediction." Paper presented at the Abu Dhabi International Petroleum Exhibition & Conference, Abu Dhabi, UAE, November 2019. doi: <https://doi.org/10.2118/197584-MS>
- Hernández Cortez, F. D. (2018). *Aplicación de la Teoría de Peforación Direccional*. Mexico: Tesis UNAM.
- Ikram Noshi, C., & Jacob Schubert, J. (2019). *Application of Data Science and Machine Learning Algorithms for ROP Prediction: Turning Data into Knowledge*. Offshore Technology Conference (OTC-29288-MS).
- INTEQ, Baker Hughes. (1995). *Drilling Engineering Workbook*. USA: Baker Hughes INTEQ.
- Kuri Abdala, J. A. (NA.). *Apuntes de Planeación*. Mexico: UNAM.
- Kyllingstad, Aage "A Comparison of Stick-Slip Mitigation Tools." Paper presented at the SPE/IADC Drilling Conference and Exhibition, The Hague, The Netherlands, March 2017.
doi: <https://doi.org/10.2118/184658-MS>

- Ledgerwood, L. W., Hoffmann, Olivier J., Jain, Jayesh R., El Hakam, Carmel, Herbig, Christian, and Reed W. Spencer. "Downhole Vibration Measurement, Monitoring, and Modeling Reveal Stick/Slip as a Primary Cause of PDC-Bit Damage in Today." Paper presented at the SPE Annual Technical Conference and Exhibition, Florence, Italy, September 2010. doi: <https://doi.org/10.2118/134488-MS>
- Mayer, H. (2007). *Analysis and Development of Algorithms for Identification and Classification of Dynamic Drillstring Dysfunctions*. Austria: University of Leoben.
- Mitchell, B. (1995). *Advanced Oilwell Drilling Engineering*. Colorado, USA.
- Mitchell, T. M. (1997). *Machine Learning*. USA: Mc.GrawHill.
- Millan, Ekaterina, Ringer, Maurice, Boualleg, Riadh, and Denis Li. "Real-Time Drillstring Vibration Characterization Using Machine Learning." Paper presented at the SPE/IADC International Drilling Conference and Exhibition, The Hague, The Netherlands, March 2019. doi: <https://doi.org/10.2118/194061-MS>
- Morley, J. (1998). *Introduction to Measurement While Drilling (MWD)*. SLC IntroToMWD.
- Noshi, Christine Ikram. "Application of Data Science and Machine Learning Algorithms for ROP Optimization in West Texas: Turning Data into Knowledge." Paper presented at the Offshore Technology Conference, Houston, Texas, May 2019. doi: <https://doi.org/10.4043/29288-MS>
- R.Reyes, A. (27 de Enero de 2019). *Oil Gains Analytics*. Obtenido de <http://blog.oilgainsanalytics.com/publications/exploring-drilling-data-from-voive-dataset-witsml-r/>
- Ricco Macedo, G. A. (2012). *Tesis : Principios de medición de los registros geofisicos de pozos*. Mexico: Universidad Nacional Autónoma de México.
- Singh, S. (24 de Agosto de 2018). *Medium, Why correlation does not imply cautaion?* Obtenido de <https://medium.com/@seema.singh/why-correlation-does-not-imply-causation-5b99790df07e>
- Smith services. (2001). *Drilling Assembly Handbook*. USA: DRILCO GROUP.
- Turing, A. (1950). *Computing machinery and intelligence*. (O. Academic, Ed.) Mind.
- Varsity Tutors. (s.f.). *Hotmath*. Obtenido de *Correlacion y relación causal*: https://www.varsitytutors.com/hotmath/hotmath_help/spanish/topics/correlation-and-causal-relation
- Vromen, T. (2015). *Control of stick-slip vibrations in drilling systems*. Holanda: Shell Global Solutions.
- Zakuan, Ahmad, Junaida, Azreen, Subroto, Bramanta, Hermawan, Heru, Fatakh, Abdul, and Abdul Halim. "Stick Slip Mitigation Plan to Improve Drilling." Paper presented at the SPE Asia Pacific Oil and Gas Conference and Exhibition, Jakarta, Indonesia, September 2011. doi: <https://doi.org/10.2118/141988-MS>

Anexo 1. Términos y Condiciones

La compañía *Equinor* permite el acceso y uso de los datos presentados en el presente trabajo bajo las condiciones siguientes:

You are free to:

- Download and use the Licensed Material for non-commercial and commercial purposes
- Create, produce and reproduce Adapted Material
- Share the Licensed Material and/or the Adapted Material

Under the following terms:

- You may not sell the Licensed Material.
- You must give Equinor and the *Vo/ve* license partners credit, and provide a link to these terms and conditions, as well as a copyright notice if applicable
- You may not share Adapted Material under a license that prevents recipients from complying with these terms and conditions
- You shall not use the Licensed Material in a manner that appears misleading nor present the Licensed Material in a distorted or incorrect manner.

Por lo que el uso dado en el presente se encuentra dentro de los límites establecidos.

De requerir los términos y condiciones extendidos, se encuentran expresados en el documento:

<https://datavillagesa.blob.core.windows.net/disclaimers/HRS%20and%20Terms%20and%20conditions%20for%20license%20to%20data%20-%20Volve.pdf>

Para el estudiante que desee acceso a estos datos, se le recomienda visitar la siguiente liga:

[“https://discovervolve.com/2020/03/05/how-to-access-the-volve-dataset/”](https://discovervolve.com/2020/03/05/how-to-access-the-volve-dataset/)

Anexo 2. Códigos Elementales

A continuación, se muestran los códigos de las funciones utilizadas para la elaboración del presente, así como algunos *notebooks* que resultarán de mucha utilidad para el lector que desee ingresar en el rubro de la programación aplicada al análisis de datos de perforación:

Librerías

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.style.use("seaborn")
import seaborn as sns
from bs4 import BeautifulSoup
from mpl_toolkits import mplot3d
import matplotlib.dates as mdates
import math
```

Funciones

```
def read_xml(file):
    print("File : ",file)
    with open(file) as f:
        data = f.read()
    data_xml = BeautifulSoup(data, 'xml')
    print("\nWellbore name:",data_xml.find_all("nameWell")[0].text)
    print("Wellbore section:",data_xml.find_all("name")[0].text)
    print("Start date:",data_xml.find_all('dTimTrajEnd')[0].text)
    len_ = len(data_xml.find_all('statusTrajStation'))
    print("Final md :",data_xml.find_all('md')[len_-1].text,"[m]")
    print("Survey's:",len_,"\n\n")
    tvd = data_xml.find_all('incl')
    cols = ['md','tvd','dispNs','dispEw','incl','azi','dls']
    well = pd.DataFrame()
    for col in cols:
        well[col] = [float(x.text) for x in data_xml.find_all(col)]
    well["neg_tvd"] = well["tvd"]*-1
    return(well)
```

```
def time_indexed_LS(file):
    #print("File:",file,"\n")
    with open(file, 'r',encoding = "latin1") as f:
        text_ = f.readlines()[0:]
    i = 0;
    for line in text_:
        #print("Line {}: {}".format(i,line.strip()))
        if "# " in line.strip():
            i_columns = i
            #print(text_[i])
            #print(text_[i+5:])
            break
        i+=1
    cols = str(text_[i_columns]).split()
    del cols[0]
```

```

data = (text_[i+5:])
data_list = []
for i in range(len(data)):
    act = data[i].split()
    data_list.append(act)
    i+=1
df = pd.DataFrame(data_list,columns = cols)
df["DateTime"] = pd.to_datetime(df["TIME"])
#del df["TIME"]
df = df.replace(-999.250000, np.nan)
df = df.replace(-999.25, np.nan)
new_df = df.apply(pd.to_numeric, errors='ignore')
new_df["DateTime"] = df["DateTime"]
print("Total rows : ",len(df))
print("Columns:\n",df.columns.values)
print("Initial depth:",df["DEPTH"][0])
print("Initial date:", df["DateTime"][0].strftime('%d-% %Y'))
print("DeltaTime:",df["DateTime"][1]-df["DateTime"][0],"\n\n")
del new_df["TIME"]
new_df = new_df.replace(-999.250000, np.nan)
new_dfdf = new_df.replace(-999.25, np.nan)
return(new_df)

def well_plt(data,info):
    plt.style.use("seaborn")
    date_format = mdates.DateFormatter("%d-% %H:%M")
    print("Plot:",info[0],"\tSection:",info[1])
    plots = list(data.columns.difference(['DateTime']))
    fig,axes = plt.subplots(1,len(plots), figsize=(20,20),sharey=True)
    for i,log in enumerate(plots):
        axes[i].plot(data[log],data['DateTime'],linewidth = .8)
        axes[i].set_ylim(data['DateTime'].min(),data['DateTime'].max())
        axes[i].invert_yaxis()
        axes[i].set_title(log,fontsize=8)
        axes[i].yaxis.set_major_formatter(date_format)
    plt.show()
    print("\n\n")

def date_plts(list_data,info,inf,sup):
    print("Plot:",info[0],"\tSection:",info[1])
    plt.style.use("bmh")
    date_format = mdates.DateFormatter("%d-% %H:%M")
    fig,axes = plt.subplots(len(list_data),1,figsize=(15,8),sharex=True)
    for i,log in enumerate(list_data):
        axes[i].plot(data["DateTime"],data[list_data[i]],linewidth = 0.5,color="steelblue")
        axes[i].set_ylabel(log,fontsize=15)
        axes[i].xaxis.set_major_formatter(date_format)
    axes[i].set_xlabel("DateTime",fontsize=15)
    axes[i].set_xlim(inf,sup)
    plt.tight_layout()

def drilling_process_fell(data,start_depth):
    """by Stephen Fell
    HDTV : Hole Depth
    ON : it on ottom """
    data["DateTime"] = pd.to_datetime(data["DateTime"])
    data['HDTH'] = data["DEPT"].cummax().apply(lambda x: x if (x > start_depth) else np.nan)

```

```

data[' ON '] = data['HDTH'] - data['DEPT']
data[' ON '] = data[' ON '].apply(lambda x: 0 if (x < 0.5) else 1)
tflo = data["TFLO"].to_list()
rpm = data["RPM"].to_list()
bonb = data[" ON "].to_list()
flg = []
for i in range(len(data)):
    if (bonb[i] == 0) and (tflo[i] > 20) and (rpm[i] > 20):
        flg.append(1)
    else :
        flg.append(0)
    i += 1
data["Flag"] = flg
return(data)

```

```

def ss_severity(data):
    ss_list = data["Stick_RT"].to_list()
    rpm_list = data["RPM"].to_list()
    ss_severity = []
    ss_categorical = []
    ss_categorical_txt = []
    for i in range(len(data)):
        try:
            s_v = (ss_list[i]/(2*rpm_list[i]))*100
        except ZeroDivisionError:
            s_v = 0
        ss_severity.append(s_v)
        if (s_v < 40) or (math.isnan(s_v)):
            cat_v = 0 #none
            cat_t = "None or Low"
        elif ((s_v > 40) and (s_v < 80)):
            cat_v = 1 # Medium
            cat_t = "Medium"
        elif ((s_v > 80) and (s_v < 100)):
            cat_v = 2 # High
            cat_t = "High"
        elif (s_v > 100):
            cat_v = 3 #Severe
            cat_t = "Severe"
        ss_categorical.append(cat_v)
        ss_categorical_txt.append(cat_t)
    data["ss_severity"] = ss_severity
    data["ss_categorical"] = ss_categorical
    data["ss_categorical_txt"] = ss_categorical_txt
    return(data)

```

Archivos indexados por tiempo pozo F10

```
WMD DATA:
-----
data:WMD\BIT_BOMS\WL_RAW_BHPR-GR-MECH_TIME_MMD_1.LAS
-----
file section Main Service Pass Direction Interval Date Original
      (meters) (Up/Down) (meters) start start file name
-----
26 .....MECHANICS SCHLUMBERGER DOWN *
   TIME INDEXED 10-Apr-09 2 09scad071_volve_15-9-f10_26fu_pgamma_dm_time.las
-----
* REAL TIME
```

```
In [1]: cd ~/Desktop/Tesis/Notebooks-Git
```

```
In [2]: import Script as sc
```

```
In [3]: cd ~/Desktop/Tesis/F10
```

```
In [4]: ls
```

Archivos referentes al pozo

```
Volume in drive C has no label.
Volume Serial Number is 90DA-519E

Directory of C:\Users\LuisH\Desktop\Tesis\F10

05/12/2021 10:36 PM <DIR>
05/12/2021 10:36 PM <DIR>
10/13/2021 04:12 PM <DIR>
09/14/2021 10:33 PM
05/12/2021 10:35 PM 492,582 FWR_MUD_nr_002.pdf
05/06/2021 02:19 AM 7,801,350 WL_RAW_BHPR-GR-MECH_TIME_MMD_1.LAS
05/06/2021 02:17 AM 7,951 WL_RAW_BHPR-GR-MECH_TIME_MMD_1-9_INF_1.ASC
05/06/2021 02:18 AM 3,437,168 WL_RAW_BHPR-GR-MECH_TIME_MMD_2.LAS
05/06/2021 02:20 AM 14,195,588 WL_RAW_BHPR-GR-MECH_TIME_MMD_3.LAS
05/06/2021 02:18 AM 3,354,638 WL_RAW_BHPR-GR-MECH_TIME_MMD_4.LAS
05/06/2021 02:19 AM 6,200,785 WL_RAW_BHPR-GR-MECH_TIME_MMD_5.LAS
05/06/2021 02:19 AM 6,021,866 WL_RAW_BHPR-GR-MECH_TIME_MMD_6.LAS
05/06/2021 02:20 AM 9,556,509 WL_RAW_BHPR-GR-MECH_TIME_MMD_7.LAS
05/06/2021 02:19 AM 8,125,959 WL_RAW_BHPR-GR-MECH_TIME_MMD_8.LAS
05/06/2021 02:19 AM 5,567,619 WL_RAW_BHPR-GR-MECH_TIME_MMD_9.LAS
12 File(s) 151,383,446 bytes
4 Dir(s) 47,803,834,368 bytes free
```

Bibliotecas a utilizar

```
In [5]: import os
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')
```

Registros de perforación

```
In [6]: files = os.listdir()
files = [s for s in files if ".LAS" in s]
files.sort()
files
```

```
Out [6]: ['WL_RAW_BHPR-GR-MECH_TIME_MMD_1.LAS',
'WL_RAW_BHPR-GR-MECH_TIME_MMD_2.LAS',
'WL_RAW_BHPR-GR-MECH_TIME_MMD_3.LAS',
'WL_RAW_BHPR-GR-MECH_TIME_MMD_4.LAS',
'WL_RAW_BHPR-GR-MECH_TIME_MMD_5.LAS',
'WL_RAW_BHPR-GR-MECH_TIME_MMD_6.LAS',
'WL_RAW_BHPR-GR-MECH_TIME_MMD_7.LAS',
'WL_RAW_BHPR-GR-MECH_TIME_MMD_8.LAS',
'WL_RAW_BHPR-GR-MECH_TIME_MMD_9.LAS']
```

Loop para construir bases de datos por medio de funcion time_indexed_LAS

```
In [7]: dfs = []
for elements in files:
    try:
        df = sc.time_indexed_LAS(elements)
    except:
        df = sc.new_time_indexed_LAS(elements)
    dfs.append(df)
```

```
File: WL_RAW_BHPR-GR-MECH_TIME_MMD_1.LAS
Total rows : 34201
Columns:
['DEPT', 'BPOS', 'BVEL', 'BONB', 'GRM1', 'HKLD', 'ROPS', 'SMOB', 'RPM', 'TOA',
'SPM1', 'SPM2', 'SPM3', 'SPPA', 'TFLO', 'TRPM_RT', 'STUCK_RT', 'STICK_RT',
'SHKPK_RT', 'DateTime']
Initial depth: 176.0605
Initial date: 10-April-2009
DeltaTime: 0 days 00:00:10

File: WL_RAW_BHPR-GR-MECH_TIME_MMD_2.LAS
Total rows : 13142
Columns:
['TIME', 'DEPTH', 'BLKP', 'ECD', 'PVEL', 'ROP', 'RPM', 'SHKRSK', 'SPPA', 'STICK',
'SMOB', 'TFLO', 'TRPM', 'GR_CAL', 'STOR', 'HKLA', 'DHAT', 'DateTime']
Initial depth: 1009.8475
Initial date: 21-April-2009
DeltaTime: 0 days 00:00:10
```

```
File: WL_RAW_BHPR-GR-MECH_TIME_MMD_3.LAS
Total rows : 54362
Total rows : 54361
Columns:
['TIME', 'DEPTH', 'BLKP', 'ECD', 'PVEL', 'ROP', 'RPM', 'SHKRSK', 'SPPA', 'STICK',
'SMOB', 'TFLO', 'TRPM', 'GR_CAL', 'STOR', 'HKLA', 'DHAT', 'DateTime']
Initial depth: -999.25
Initial date: 23-April-2009
DeltaTime: 0 days 00:00:10
```

```
[ 'DEPT', 'ROPS', 'INCL_CONT_RT', 'GRMA_BHC_RT', 'HKLD', 'SMOB', 'TQA', 'RPM',
'Stick_RT', 'SHKRSK_RT', 'DHAT', 'TRPM_RT', 'TFLO', 'SPPA', 'DHAP', 'ECD', 'BPOS',
'BEVEL', 'UCAV_DH_ECO_RT', 'DateTime' ]
Initial depth: 3429.7584
Initial date: 31-May-2009
DeltaTime: 0 days 00:00:10
```

```
In [8]:
run_2 = dfs[0]
run_3 = dfs[1]
run_4 = dfs[2]
run_5 = dfs[3]
run_6 = dfs[4]
run_7 = dfs[5]
run_8 = dfs[6]
run_9 = dfs[7]
run_10 = dfs[8]
```

Creacion de Flags de seccion de perforacion

```
In [9]:
runs = [run_2, run_3, run_4, run_5, run_6, run_7, run_8, run_9, run_10]
runs_txt = ["run_2", "run_3", "run_4", "run_5", "run_6", "run_7", "run_8", "run_9", "run_10"]
sections = ['26"', '17 1/2"', '17 1/2"', '17 1/2"', '12 1/4"', '12 1/4"', '12 1/4"', '12 1/4"', '8 1/2"', '8 1/2"']
info = list(zip(runs_txt, sections))
```

```
In [10]:
print("Archivo\t\t\t\tSeccion\t\t\t\tIndice")
list(zip(files, info, np.arange(len(info))))
```

```
Out[10]:
Archivo Seccion Indice
[('WL_RAW_BHPR-GR-MECH_TIME_MWD_1.LAS', ('run_2', '26"'), 0),
 ('WL_RAW_BHPR-GR-MECH_TIME_MWD_2.LAS', ('run_3', '17 1/2"'), 1),
 ('WL_RAW_BHPR-GR-MECH_TIME_MWD_3.LAS', ('run_4', '17 1/2"'), 2),
 ('WL_RAW_BHPR-GR-MECH_TIME_MWD_4.LAS', ('run_5', '12 1/4"'), 3),
 ('WL_RAW_BHPR-GR-MECH_TIME_MWD_5.LAS', ('run_6', '12 1/4"'), 4),
 ('WL_RAW_BHPR-GR-MECH_TIME_MWD_6.LAS', ('run_7', '12 1/4"'), 5),
 ('WL_RAW_BHPR-GR-MECH_TIME_MWD_7.LAS', ('run_8', '8 1/2"'), 6),
 ('WL_RAW_BHPR-GR-MECH_TIME_MWD_8.LAS', ('run_9', '8 1/2"'), 7),
 ('WL_RAW_BHPR-GR-MECH_TIME_MWD_9.LAS', ('run_10', '8 1/2"'), 8)]
```

Corridas de perforacion

```
In [11]:
%%time
i = 0
for run in runs:
    sc.well_pit(run, info[i])
    i += 1
```

Plot: run_2 Section: 26"

```
File: WL_RAW_BHPR-GR-MECH_TIME_MWD_4.LAS
Total rows : 14698
Columns:
[ 'DEPT', 'ROPS', 'ARC_GR_RT', 'BPOS', 'BEVEL', 'SMOB', 'HKLD', 'TQA', 'RPM',
'Stick_RT', 'CRPM_RT', 'TFLO', 'TRPM_RT', 'SPPA', 'SHKRSK_RT', 'SHKPK_RT',
'ECD_ARC', 'APRS_ARC', 'ATMP', 'DateTime' ]
Initial depth: 296.6532
Initial date: 05-May-2009
DeltaTime: 0 days 00:00:10
```

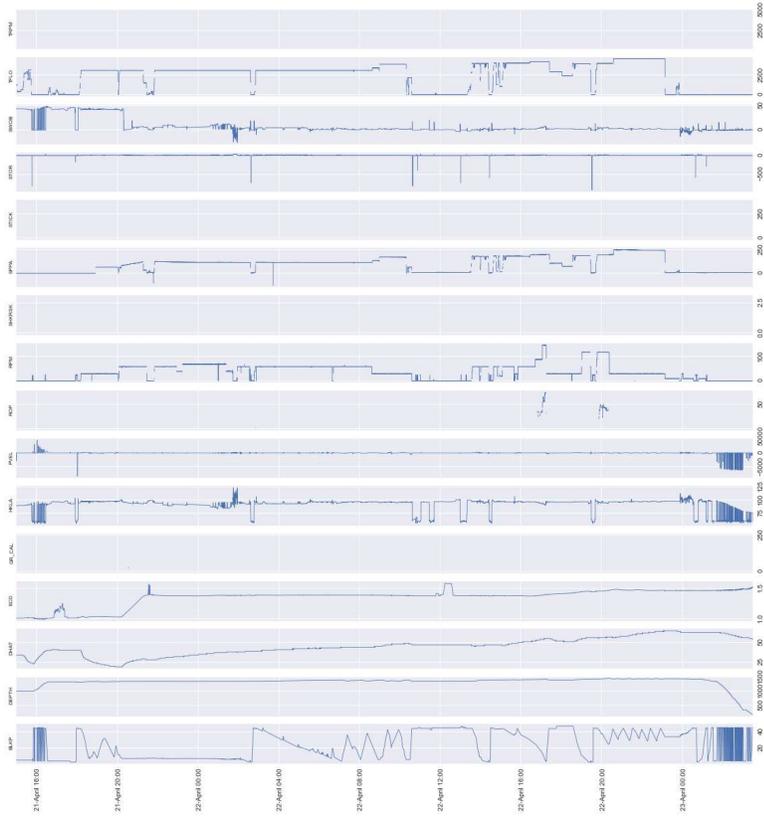
```
File: WL_RAW_BHPR-GR-MECH_TIME_MWD_5.LAS
Total rows : 27181
Columns:
[ 'DEPT', 'ROPS', 'ARC_GR_RT', 'BPOS', 'BEVEL', 'SMOB', 'HKLD', 'TQA', 'RPM',
'Stick_RT', 'CRPM_RT', 'TFLO', 'TRPM_RT', 'SPPA', 'SHKRSK_RT', 'SHKPK_RT',
'ECD_ARC', 'APRS_ARC', 'ATMP', 'DateTime' ]
Initial depth: 527.9707
Initial date: 07-May-2009
DeltaTime: 0 days 00:00:10
```

```
File: WL_RAW_BHPR-GR-MECH_TIME_MWD_6.LAS
Total rows : 26281
Columns:
[ 'DEPT', 'ROPS', 'ARC_GR_RT', 'BPOS', 'BEVEL', 'SMOB', 'HKLD', 'TQA', 'RPM',
'Stick_RT', 'CRPM_RT', 'TFLO', 'TRPM_RT', 'SPPA', 'SHKRSK_RT', 'SHKPK_RT',
'ECD_ARC', 'APRS_ARC', 'ATMP', 'DateTime' ]
Initial depth: nan
Initial date: 11-May-2009
DeltaTime: 0 days 00:00:10
```

```
File: WL_RAW_BHPR-GR-MECH_TIME_MWD_7.LAS
Total rows : 40651
Columns:
[ 'DEPT', 'ROPS', 'INCL_CONT_RT', 'GRMA_BHC_RT', 'HKLD', 'SMOB', 'TQA', 'RPM',
'Stick_RT', 'SHKRSK_RT', 'DHAT', 'TRPM_RT', 'TFLO', 'SPPA', 'DHAP', 'ECD', 'BPOS',
'BEVEL', 'UCAV_DH_ECO_RT', 'DateTime' ]
Initial depth: 3394.5204
Initial date: 18-May-2009
DeltaTime: 0 days 00:00:10
```

```
File: WL_RAW_BHPR-GR-MECH_TIME_MWD_8.LAS
Total rows : 46153
Columns:
[ 'DEPT', 'ROPS', 'INCL_CONT_RT', 'HKLD', 'SMOB', 'TQA', 'RPM', 'Stick_RT',
'SHKRSK_RT', 'TRPM_RT', 'TFLO', 'SPPA', 'BPOS', 'BEVEL', 'DateTime' ]
Initial depth: 3430.0329
Initial date: 24-May-2009
DeltaTime: 0 days 00:00:10
```

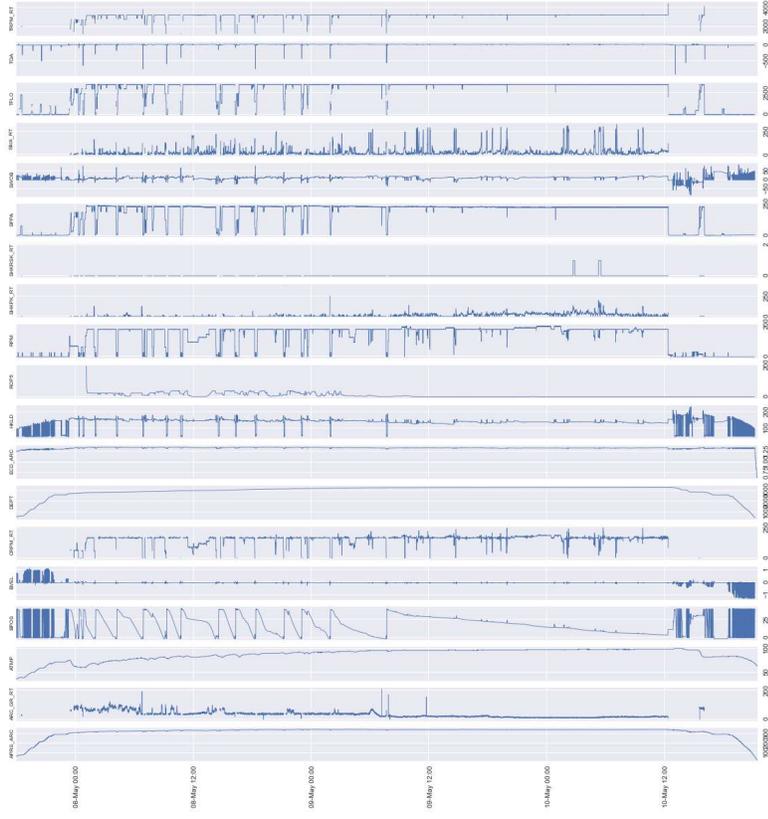
```
File: WL_RAW_BHPR-GR-MECH_TIME_MWD_9.LAS
Total rows : 23677
Columns:
```



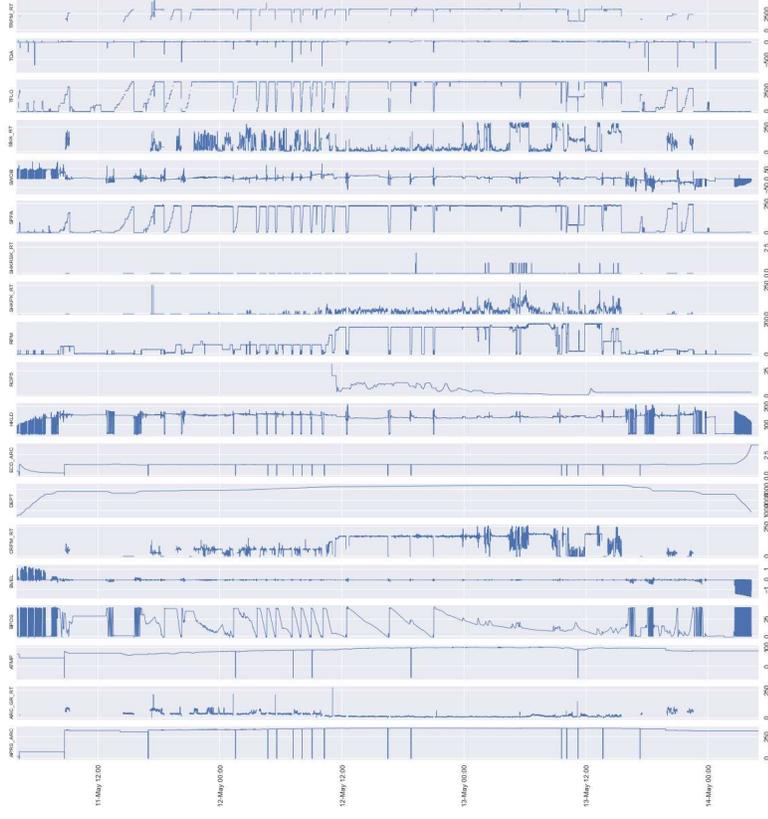
Plot: run_4 Section: 17 1/2"



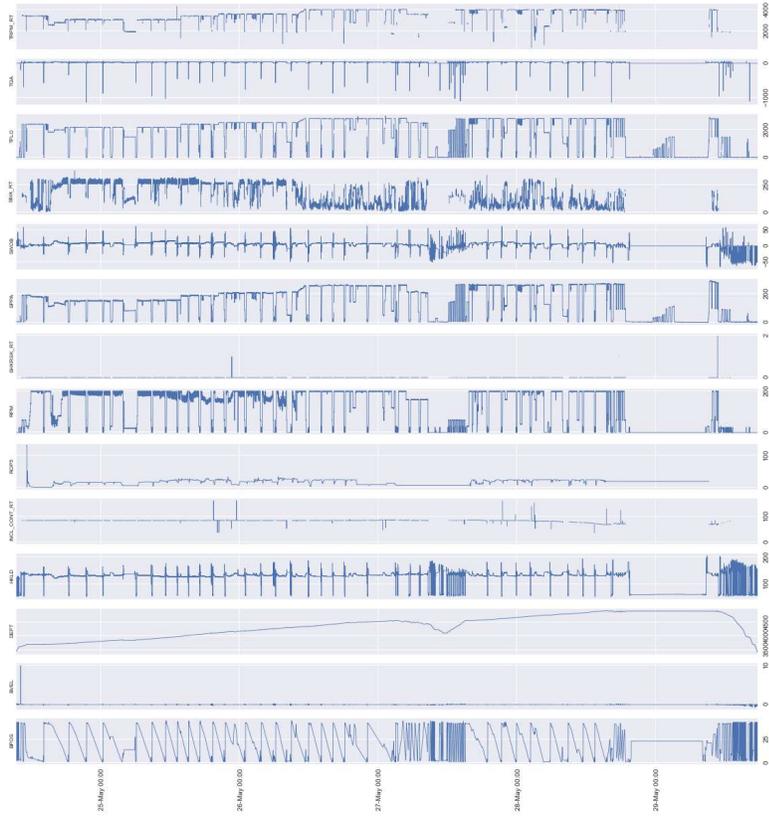
Plot: run_3 Section: 17 1/2"



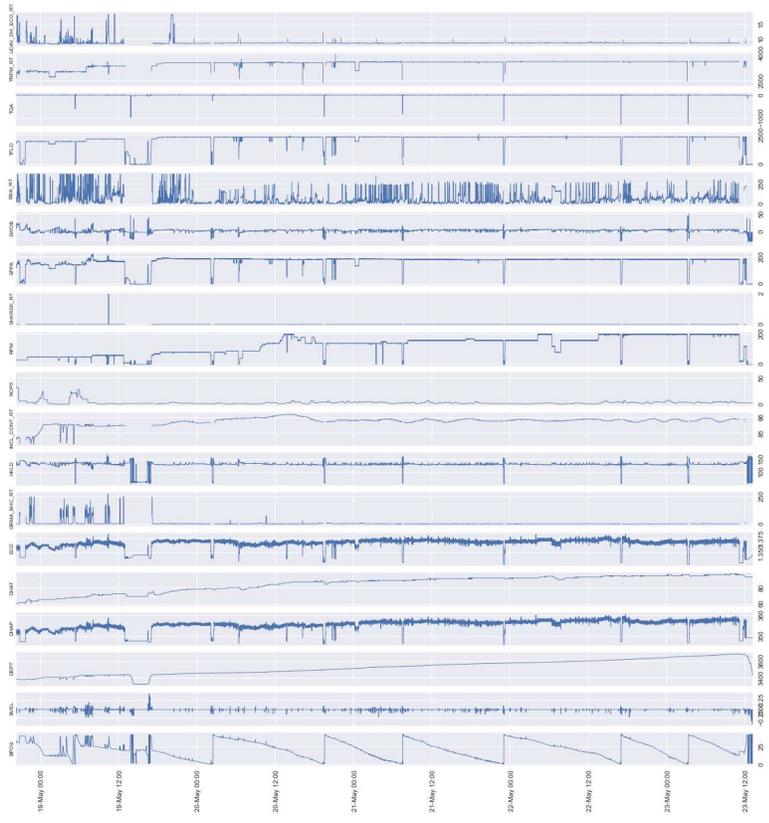
Plot: run_7 Section: 12 1/4"



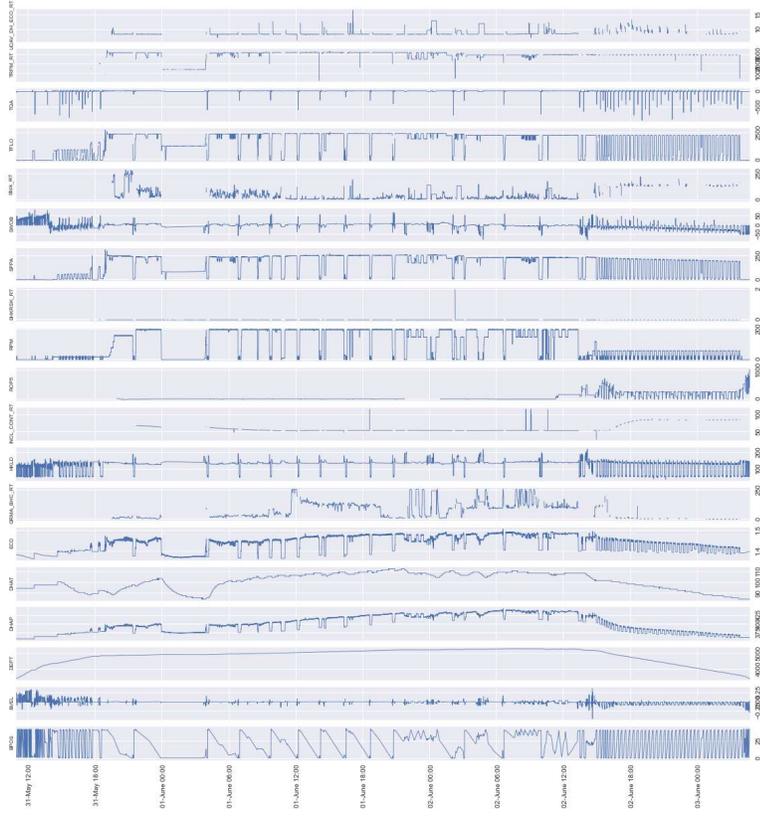
Plot: run_8 Section: 8 1/2"



Plot: run_10 Section: 8 1/2"



Plot: run_9 Section: 8 1/2"



Wall time: 1min 30s

Selector de Fechas

- ipywidgets necesario

La funcion es necesaria para seleccionar un rango dentro de este notebook

```
In [12]:
import ipywidgets as widgets
def print_date_range(date_range):
    global inf ; inf = date_range[0]
    global sup ; sup = date_range[1]
    return(inf, sup)
date_slider(data):
dates = data.set_index("DateTime", index ; options = [(date.strftime("%d %m %H:%M"))
selection_date_slider = widgets.SelectionRangeSlider(options=dates, index=index, or
widgets.interact(print_date_range, date_range=selection_date_slider)
```

Análisis de Registros

In [20]:

```
print("Archivo\t\t\t\t\tSeccion\t\t\t\t\tIndice")
List(zip(files, info, np.arange(len(info))))
```

Out[20]:

```
Archivo Seccion Indice
[('WL_RAW_BHPR-GR-MECH_TIME_MMD_1.LAS', ('run_2', '26"', 0),
('WL_RAW_BHPR-GR-MECH_TIME_MMD_2.LAS', ('run_3', '17 1/2"', 1),
('WL_RAW_BHPR-GR-MECH_TIME_MMD_3.LAS', ('run_4', '17 1/2"', 2),
('WL_RAW_BHPR-GR-MECH_TIME_MMD_4.LAS', ('run_5', '12 1/4"', 3),
('WL_RAW_BHPR-GR-MECH_TIME_MMD_5.LAS', ('run_6', '12 1/4"', 4),
('WL_RAW_BHPR-GR-MECH_TIME_MMD_6.LAS', ('run_7', '12 1/4"', 5),
('WL_RAW_BHPR-GR-MECH_TIME_MMD_7.LAS', ('run_8', '8 1/2"', 6),
('WL_RAW_BHPR-GR-MECH_TIME_MMD_8.LAS', ('run_9', '8 1/2"', 7),
('WL_RAW_BHPR-GR-MECH_TIME_MMD_9.LAS', ('run_10', '8 1/2"', 8)]
```

Corrida 9

Base de datos

In [19]:

```
runs[7]
```

Out[19]:

```
DEPT  ROPS  INCL_CONT_RT  HKLD  SWOB  TQA  RPM  Stick_RT  SHKRSK_RT  TRPN
```

0	3430.0329	NaN	NaN	123.4083	11.4065	0.0273	0.0	NaN	NaN
1	3432.7112	NaN	NaN	123.7613	11.0534	0.0256	0.0	NaN	NaN
2	3434.7227	NaN	NaN	123.0905	11.7242	0.0230	0.0	NaN	NaN
3	3435.8878	NaN	NaN	125.0726	9.7421	0.0260	0.0	NaN	NaN
4	3436.0723	NaN	NaN	124.3252	10.4895	0.0288	0.0	NaN	NaN
...
46147	3432.0700	NaN	NaN	150.3643	-44.1485	0.0221	0.0	NaN	NaN
46148	3429.4581	NaN	NaN	149.9109	-43.6951	0.0231	0.0	NaN	NaN
46149	3426.8152	NaN	NaN	150.1218	-43.9060	0.0237	0.0	NaN	NaN
46150	3423.1287	NaN	NaN	150.3671	-44.1513	0.0226	0.0	NaN	NaN

DEPT ROP5 INCL_CONT_RT HKLD SWOB TOA RPM SICK_RT SHKRSK_RT TRPM

46151 3420.5448 NaN NaN 150.4841 -44.2682 0.0227 0.0 NaN NaN NaN

46152 rows x 15 columns

Selector de fechas

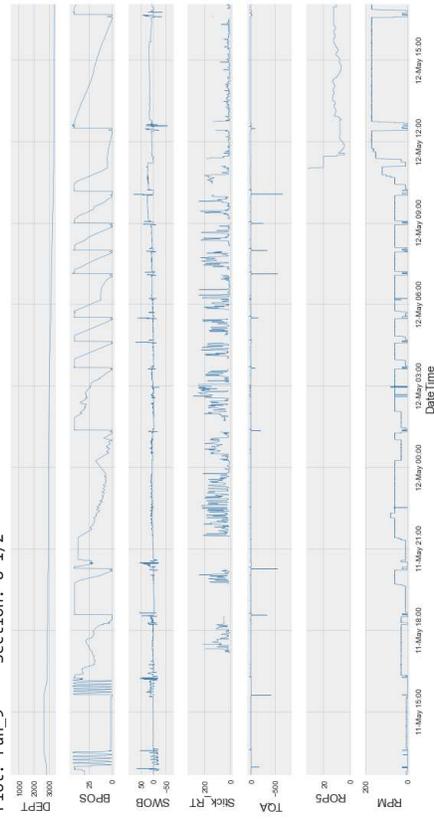
In [16]:

```
date_slider(run_7)
```

In [18]:

```
sc.date_plt(["DEPTH", "BPOS", "SMOB", "Stick_RT", "TOA", "RPM", "run_7", info[7], inf, sup
```

Plot: run_9 Section: 8 1/2"



Adición de etiquetas de pozo y seccion

In [25]:

```
i = 0
for run in runs:
    run["well"] = "F-10"
    run["Run"] = runs_txt[i]
    i += 1
```

Profundidades de las secciones de perforación

In [26]:

```
print("Archivo\t\t\t\t\tSeccion\t\t\tIndice")
list(zip(files, info, np.arange(len(info))))
```

Out[26]:

```
Archivo Seccion Indice
[('WL_RAW_BHPR-GR-MECH_TIME_WMD_1.LAS', ('run_2', '26"'), 0),
 ('WL_RAW_BHPR-GR-MECH_TIME_WMD_2.LAS', ('run_3', '17 1/2"'), 1),
 ('WL_RAW_BHPR-GR-MECH_TIME_WMD_3.LAS', ('run_4', '17 1/2"'), 2),
 ('WL_RAW_BHPR-GR-MECH_TIME_WMD_4.LAS', ('run_5', '12 1/4"'), 3),
```

```
('WL_RAW_BHPR-GR-MECH_TIME_WMD_5.LAS', ('run_6', '12 1/4"'), 4),
 ('WL_RAW_BHPR-GR-MECH_TIME_WMD_6.LAS', ('run_7', '12 1/4"'), 5),
 ('WL_RAW_BHPR-GR-MECH_TIME_WMD_7.LAS', ('run_8', '8 1/2"'), 6),
 ('WL_RAW_BHPR-GR-MECH_TIME_WMD_8.LAS', ('run_9', '8 1/2"'), 7),
 ('WL_RAW_BHPR-GR-MECH_TIME_WMD_9.LAS', ('run_10', '8 1/2"'), 8)]
```

62. Drilling Bits Summary

Bit System	Bit Number	12 1/2"	12 3/4"	13 1/2"	14 1/2"	15 1/2"	16 1/2"	17 1/2"	18 1/2"	19 1/2"	20 1/2"	21 1/2"	22 1/2"	23 1/2"	24 1/2"	25 1/2"	26 1/2"	27 1/2"	28 1/2"	29 1/2"	30 1/2"	31 1/2"	32 1/2"	33 1/2"	34 1/2"	35 1/2"	36 1/2"	37 1/2"	38 1/2"	39 1/2"	40 1/2"	41 1/2"	42 1/2"	43 1/2"	44 1/2"	45 1/2"	46 1/2"	47 1/2"	48 1/2"	49 1/2"	50 1/2"	51 1/2"	52 1/2"	53 1/2"	54 1/2"	55 1/2"	56 1/2"	57 1/2"	58 1/2"	59 1/2"	60 1/2"	61 1/2"	62 1/2"	63 1/2"	64 1/2"	65 1/2"	66 1/2"	67 1/2"	68 1/2"	69 1/2"	70 1/2"	71 1/2"	72 1/2"	73 1/2"	74 1/2"	75 1/2"	76 1/2"	77 1/2"	78 1/2"	79 1/2"	80 1/2"	81 1/2"	82 1/2"	83 1/2"	84 1/2"	85 1/2"	86 1/2"	87 1/2"	88 1/2"	89 1/2"	90 1/2"	91 1/2"	92 1/2"	93 1/2"	94 1/2"	95 1/2"	96 1/2"	97 1/2"	98 1/2"	99 1/2"	100 1/2"									
Bit System	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

63. FWR_MUD_nr_002.pdf

Bit System	Bit Number	12 1/2"	12 3/4"	13 1/2"	14 1/2"	15 1/2"	16 1/2"	17 1/2"	18 1/2"	19 1/2"	20 1/2"	21 1/2"	22 1/2"	23 1/2"	24 1/2"	25 1/2"	26 1/2"	27 1/2"	28 1/2"	29 1/2"	30 1/2"	31 1/2"	32 1/2"	33 1/2"	34 1/2"	35 1/2"	36 1/2"	37 1/2"	38 1/2"	39 1/2"	40 1/2"	41 1/2"	42 1/2"	43 1/2"	44 1/2"	45 1/2"	46 1/2"	47 1/2"	48 1/2"	49 1/2"	50 1/2"	51 1/2"	52 1/2"	53 1/2"	54 1/2"	55 1/2"	56 1/2"	57 1/2"	58 1/2"	59 1/2"	60 1/2"	61 1/2"	62 1/2"	63 1/2"	64 1/2"	65 1/2"	66 1/2"	67 1/2"	68 1/2"	69 1/2"	70 1/2"	71 1/2"	72 1/2"	73 1/2"	74 1/2"	75 1/2"	76 1/2"	77 1/2"	78 1/2"	79 1/2"	80 1/2"	81 1/2"	82 1/2"	83 1/2"	84 1/2"	85 1/2"	86 1/2"	87 1/2"	88 1/2"	89 1/2"	90 1/2"	91 1/2"	92 1/2"	93 1/2"	94 1/2"	95 1/2"	96 1/2"	97 1/2"	98 1/2"	99 1/2"	100 1/2"									
Bit System	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

In [27]:

```
depths = [207, 1400, 1463, 2616, 2825, 3319, 3442, 3695, 4911]
```

Renombre de columnas de profundidad

In [28]:

```
runs[1] = runs[1].rename({"DEPTH": "DEPTH"}, axis=1)
runs[2] = runs[2].rename({"DEPTH": "DEPTH"}, axis=1)
```

Adición de BONB a los datasets

In [30]:

```
drill_op = []
i = 0
for run in runs:
    df = sc.drilling_process_fell(run, depths[i])
    sc.drill_flag(df, info[i])
    df = df[df["flag"] == 1]
    drill_op.append(df)
    i += 1
```

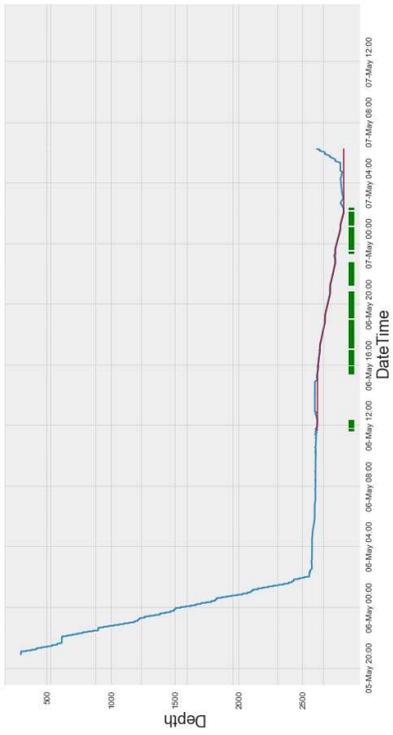
Plot: run_2 Section: 26"

Bit Depth (DEPTH)
Hole Depth (HDEPTH)
Drilling Flag



Plot: run_5 Section: 12 1/4"

Bit Depth (DEPTH)
Hole Depth (HDEPTH)
Drilling Flag



Plot: run_6 Section: 12 1/4"

Bit Depth (DEPTH)
Hole Depth (HDEPTH)
Drilling Flag



Plot: run_3 Section: 17 1/2"

Bit Depth (DEPTH)
Hole Depth (HDEPTH)
Drilling Flag



Plot: run_4 Section: 17 1/2"

BI Depth (DEPT)
Hole Depth (HDDTH)
Drilling Flag



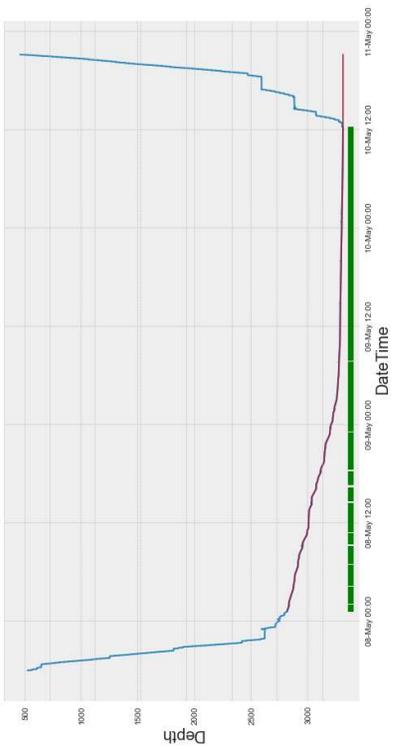
Plot: run_9 Section: 8 1/2"

BI Depth (DEPT)
Hole Depth (HDDTH)
Drilling Flag



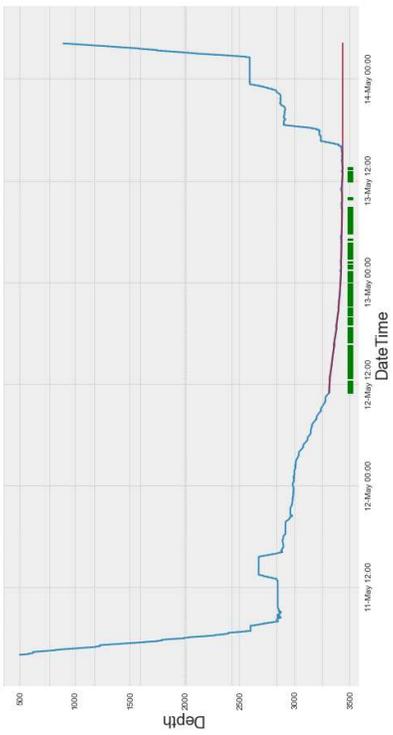
Plot: run_10 Section: 8 1/2"

BI Depth (DEPT)
Hole Depth (HDDTH)
Drilling Flag



Plot: run_7 Section: 12 1/4"

BI Depth (DEPT)
Hole Depth (HDDTH)
Drilling Flag



Plot: run_8 Section: 8 1/2"

— Bit Depth (BDDT)
— Rate Depth (RDT74)
| Drilling Flag



Analisis de corridas de perforación

```
In [33]: print("Archivo\t\t\t\tSeccion\t\tIndice")
List(zip(files,info,np.arange(len(info))))

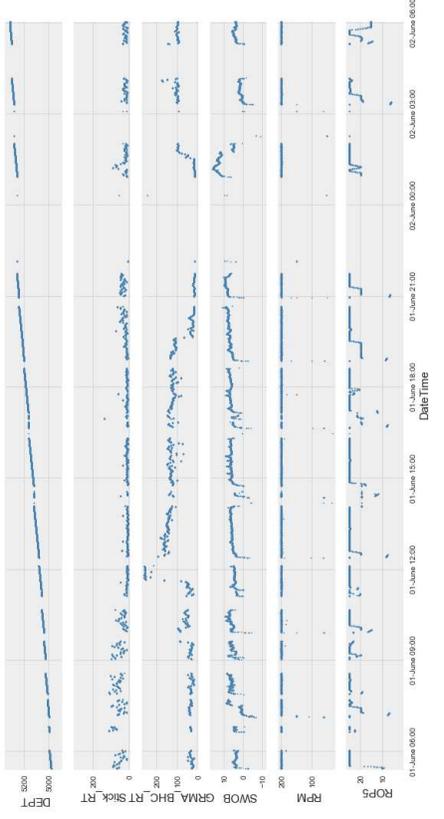
Archivo Seccion Indice
[('WL_RAW_BHPR-GR-MECH_TIME_MMD_1.LAS', ('run_2', '26"), 0),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_2.LAS', ('run_3', '17 1/2"), 1),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_3.LAS', ('run_4', '17 1/2"), 2),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_4.LAS', ('run_5', '12 1/4"), 3),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_5.LAS', ('run_6', '12 1/4"), 4),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_6.LAS', ('run_7', '12 1/4"), 5),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_7.LAS', ('run_8', '8 1/2"), 6),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_8.LAS', ('run_9', '8 1/2"), 7),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_9.LAS', ('run_10', '8 1/2"), 8)]
```

Perforación de la sección de 8 1/2" de pozo F10

Registros disponibles

```
In [34]: runs[8].columns
Out[34]: Index(['DEPT', 'ROPS', 'INCL_CONT_RT', 'GRMA_BHC_RT', 'HKLD', 'SMOB', 'TOA',
        'RPM', 'stick_RT', 'SHKRSK_RT', 'DHAT', 'TRPM_RT', 'TFLO', 'SPPA',
        'DHAP', 'ECD', 'BPOS', 'BVEL', 'UCAV_DH_ECO_RT', 'DateTime', 'Well',
        'Run', 'HDTH', 'BONB', 'Fflag',
        dtype='object')
In [36]: date_slider(runs[8])
```

```
In [40]: sc.date_scatter_pits(['DEPT', 'stick_RT', 'GRMA_BHC_RT', 'SMOB', 'RPM', 'ROPS'],drill_op[8],
Plot: run_10 Section: 8 1/2"
```



Creacion de archivo csv de las secciones del pozo F10

```
In [42]: print("Archivo\t\t\t\tSeccion\t\tIndice")
List(zip(files,info,np.arange(len(info))))

Archivo Seccion Indice
[('WL_RAW_BHPR-GR-MECH_TIME_MMD_1.LAS', ('run_2', '26"), 0),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_2.LAS', ('run_3', '17 1/2"), 1),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_3.LAS', ('run_4', '17 1/2"), 2),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_4.LAS', ('run_5', '12 1/4"), 3),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_5.LAS', ('run_6', '12 1/4"), 4),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_6.LAS', ('run_7', '12 1/4"), 5),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_7.LAS', ('run_8', '8 1/2"), 6),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_8.LAS', ('run_9', '8 1/2"), 7),
 ('WL_RAW_BHPR-GR-MECH_TIME_MMD_9.LAS', ('run_10', '8 1/2"), 8)]

In [43]: filepath = "/home/luis/Desktop/Tesis/F10/CSVs"

i = 2 for file in drill_op: file.to_csv(filepath+"/run_"+str(i)+".csv",index=False) i+=1

In [ ]:
```

Acople de datos direccionales y registros de vibración para pozo F-5

```
In [1]: cd ~/Desktop/Tesis/Notebooks-Git
```

```
C:\Users\LuisH\Desktop\Tesis\Notebooks-Git
```

```
In [2]: import Script as sc
```

```
In [3]: cd ~/Desktop/Tesis/F5/trajectory
```

```
C:\Users\LuisH\Desktop\Tesis\F5\trajectory
```

```
In [4]: import os
from bs4 import BeautifulSoup
import pandas as pd
import seaborn as sns
import plotly.express as px
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')
```

Registros de trayectoria WITSMIL

en formato .XML

```
In [5]: ls
```

```
Volume in drive C has no label.
Volume Serial Number is 90DA-519E

Directory of C:\Users\LuisH\Desktop\Tesis\F5\trajectory

09/14/2021 05:14 PM <DIR>
09/14/2021 05:14 PM <DIR>
04/27/2021 06:46 PM 118,363 1.xml
04/28/2021 12:07 AM 9,371 Book1.xlsx
04/27/2021 06:46 PM 32 MetadataFileInfo.txt
05/01/2021 02:48 AM 6,004,223 WellboreF5.csv
4 File(s) 6,131,989 bytes
2 Dir(s) 47,901,167,616 bytes free
```

```
In [6]: files = os.listdir()
files = [s for s in files if ".xml" in s]
files
```

```
Out[6]: ['1.xml']
```

Lector de archivos xml y construcción de base de datos

```
In [7]: for file in files:
sc.read_xml(file)
```

File : 1.xml

```
Wellbore name: NO 15/9-F-5
Wellbore section: MWD-15/9-F-5
Start date: 2016-08-28T10:32:12.000Z
Final md : 3761.77001953125 [m]
Surveys: 97
```

```
In [8]: traj_wellbore = sc.read_xml(files[0])
```

File : 1.xml

```
Wellbore name: NO 15/9-F-5
Wellbore section: MWD-15/9-F-5
Start date: 2016-08-28T10:32:12.000Z
Final md : 3761.77001953125 [m]
Surveys: 97
```

```
In [9]: traj_wellbore
```

```
Out[9]:
```

	md	tvdt	dispsNs	dispEw	incl	azi	dls	vertSect	n
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	-0.
1	145.899994	145.899994	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	-145.
2	146.000000	146.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	-146.
3	154.899994	154.899982	-0.009374	-0.008158	0.002793	3.857701	0.000314	-0.009374	-154.
4	195.300003	195.299828	-0.065922	0.052531	0.004887	1.713041	0.000169	-0.065922	-195.
...
93	3642.070068	3157.971685	32.573648	1307.338473	0.957139	0.386590	0.000080	32.573648	-3157.
94	3682.429932	3181.129784	63.179152	1319.825282	0.962200	0.388161	0.000129	63.179152	-3181.
95	3722.879883	3204.733187	93.604499	1332.205129	0.933228	0.384671	0.000720	93.604499	-3204.
96	3761.770020	3228.099703	122.420063	1343.870781	0.919265	0.384671	0.000359	122.420063	-3228.
97	3792.000000	3246.431310	144.701001	1352.890964	0.919265	0.384671	0.000000	144.701001	-3246.

98 rows x 9 columns

Gráficos

```
In [29]: fig, axes = plt.subplots(1,4, figsize=(10,8))
axes[0].scatter(traj_wellbore["vertSect"], traj_wellbore["tvdt"])
axes[0].invert_yaxis()
axes[0].set_ylim(max(traj_wellbore["tvdt"])+10,0)
axes[0].set_xlabel("Seccion vertical [m]")
axes[0].set_ylabel("TVD [m]")

axes[1].scatter(traj_wellbore["incl"],traj_wellbore["md"])
axes[1].invert_yaxis()
axes[1].set_ylim(max(traj_wellbore["md"])+10,0)
axes[1].set_xlabel("Inclinacion [°]")
```

```

axes[1].set_ylabel("MD [m]")
axes[1].tick_params(axis='y', which='minor')

axes[2].scatter(traj_wellbore["dis"], traj_wellbore["md"])
axes[2].invert_yaxis()
axes[2].set_ylim(max(traj_wellbore["md"])+10, 0)
axes[2].set_xlabel("DLS [°]")
axes[2].set(yticklabels=[])

axes[3].scatter(traj_wellbore["azi"], traj_wellbore["md"])
axes[3].invert_yaxis()
axes[3].set_ylim(max(traj_wellbore["md"])+10, 0)
axes[3].set_xlabel("Azimut [°]")
axes[3].set(yticklabels=[])

```

```
plt.tight_layout()
```

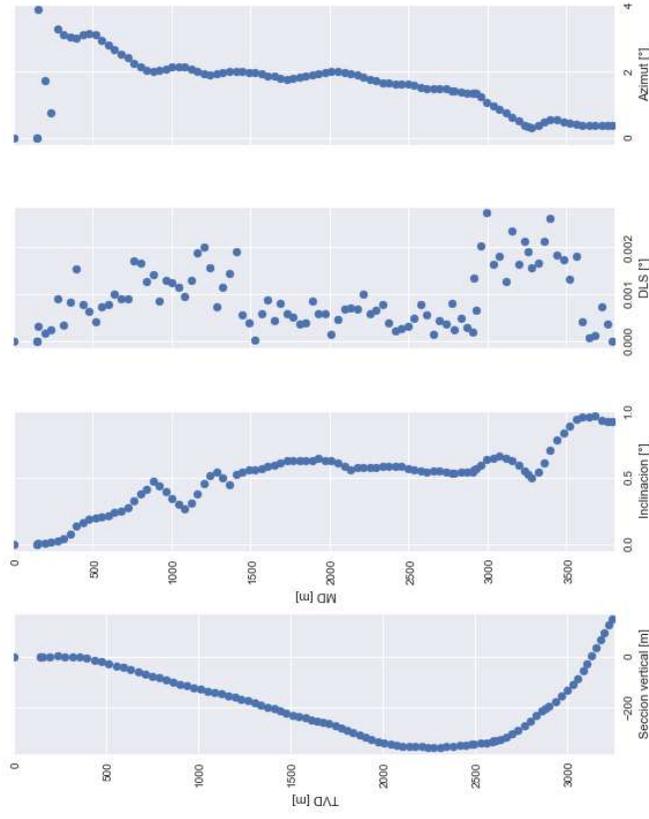


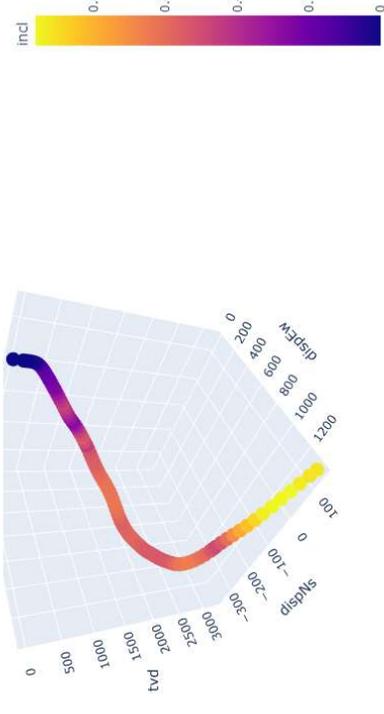
Gráfico 3D

- Inclinação

```

x = traj_wellbore["dispEw"].values
y = traj_wellbore["dispNs"].values
z = traj_wellbore["tvd"].values
fig = px.scatter_3d(traj_wellbore, x = "dispEw", y = "dispNs", z = "tvd", color = "incl")
fig.update_scenes(zaxis_aurange = "reversed")
fig.show()

```

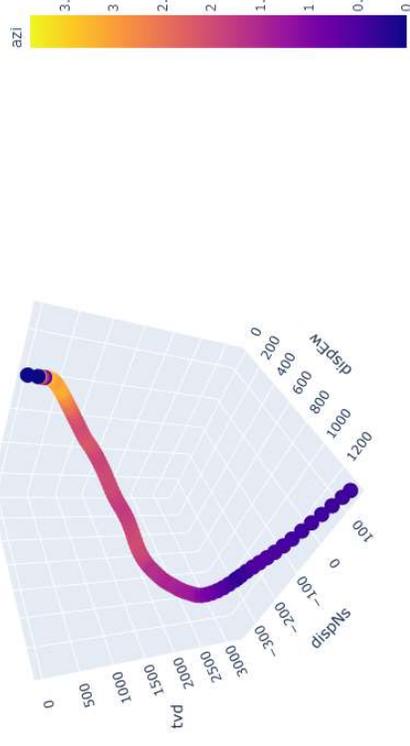


- Azimut

```

fig = px.scatter_3d(traj_wellbore, x = "dispEw", y = "dispNs", z = "tvd", color = "azi")
fig.update_scenes(zaxis_aurange = "reversed")
fig.show()

```

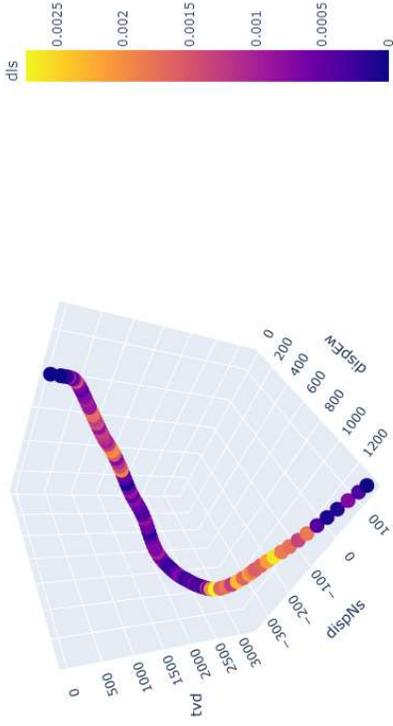


- DLS

```

fig = px.scatter_3d(traj_wellbore, "dispEw", "dispNs", "tvd", color = "dis")
fig.update_scenes(zaxis_aurange = "reversed")
fig.show()

```

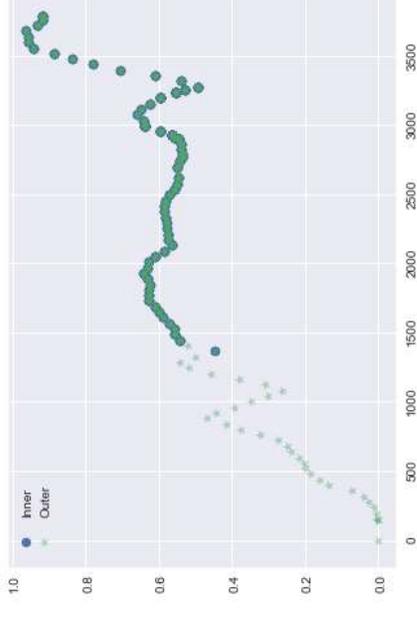


Funcion merge

```
In [18]:
df_inner = pd.merge(survey, mwd, how="inner")
df_outer = pd.merge(survey, mwd, how="outer")
```

Tipos de combinación

```
In [19]:
plt.scatter(df_inner["DEPT"], df_inner["incl"], label="Inner")
plt.scatter(df_outer["DEPT"], df_outer["incl"], label="Outer", alpha=.4, marker="*")
plt.legend()
plt.show()
```



Registros de Stick & Slip acoplados a los survey's

```
fig = px.scatter_3d(df_outer, df_outer["dispEw"], df_outer["dispNs"], df_outer["tvd"], color =
"Stick_RT")
```

```
fig.update_scenes(zaxis_autorange = "reversed") fig.show()
```

Datos MWD

```
In [13]: f_f5 = "C:/Users/LuisH/Desktop/Tesis/F5/CSVs/WellboreF5.csv"
```

```
In [14]: df = pd.read_csv(f_f5)
```

```
In [15]: survey = traj_wellbore[["md", "dispNs", "incl", "tvd", "dis", "azi"]]
```

```
mwd = df
```

```
In [16]: survey.rename(columns = {"md": "DEPT"}, inplace=True)
```

```
C:\Users\LuisH\anaconda3\lib\site-packages\pandas\core\frame.py:4296: SettingWithCopyWarning:
  A value is trying to be set on a copy of a slice from a DataFrame
```

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/user-guide/indexing.html#returning-a-view-versus-a-copy>

```
ing:
A value is trying to be set on a copy of a slice from a DataFrame
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user-guide/indexing.html#returning-a-view-versus-a-copy
```

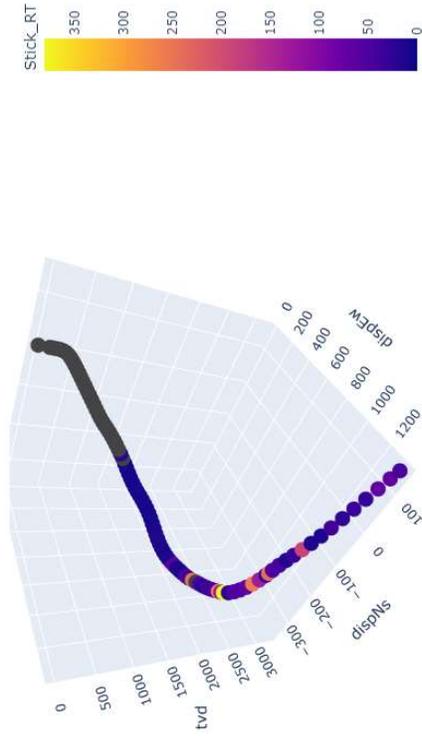
Acople de profundidades de survey y MWD

```
In [17]: survey["DEPT"] = survey["DEPT"].astype(int)
mwd["DEPT"] = mwd["DEPT"].astype(int)
```

```
<ipython-input-17-4b3ecb5f3e78>:3: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame. Try using `.loc[row_indexer,col_indexer] = value` instead

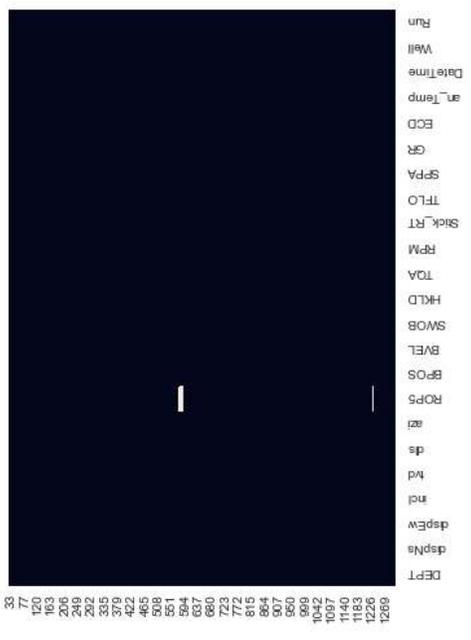
```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user-guide/indexing.html#returning-a-view-versus-a-copy
```



```
In [23]: df_outer.dropna(subset=["Stick_RT", "incl", inplace=True)
```

```
In [24]: %%time
print("Missing data:")
sns.heatmap(df_outer.isnull(), cbar=False)
plt.xlabel("")
plt.show()
```

Missing data:



Wall time: 4.84 s
df_outer.to_csv(r"\\WellboreF5_Incl.csv", index=False)

```
In [1]: cd ~/Desktop/Tesis/Notebooks-Git
In [2]: C:\Users\LuisH\Desktop\Tesis\Notebooks-Git
import script as sc
```

Análisis de vibraciones Stick & Slip

```
In [3]: cd ~/Desktop/Tesis/Notebooks-Git\Training\StickSlip
C:\Users\LuisH\Desktop\Tesis\Notebooks-Git\Training\StickSlip
1s
Volume in drive C has no label.
Volume Serial Number is 90DA-519E
Directory of C:\Users\LuisH\Desktop\Tesis\Notebooks-Git\Training\StickSlip
```

```
10/13/2021 05:13 PM <DIR>
10/13/2021 05:13 PM <DIR>
10/13/2021 05:13 PM <DIR>
09/15/2021 08:21 AM 9, 293, 645 Blind_F15A.csv
09/15/2021 09:03 AM 272, 832 Blind_F15A_incl.csv
09/01/2021 07:55 PM 9, 108, 286 BlindSet.csv
09/15/2021 10:29 AM 131, 051 BlindSet.ipynb
09/01/2021 07:55 PM 270, 654 BlindSet_Incl.csv
09/26/2021 05:36 PM 4, 314, 036 DecisionTree.ipynb
09/14/2021 06:24 PM 593, 865 DecisionTree_Copy1.ipynb
09/11/2021 12:45 AM 1, 098 dtc.dot
10/06/2021 12:28 PM 400, 007 RandomForest.ipynb
10/06/2021 12:58 PM 4, 350, 260 RandomForest_Incl.ipynb
09/14/2021 09:04 PM 2, 534, 049 rf_5trees.png
09/15/2021 09:13 AM 273, 255 rf_incl.csv
09/15/2021 08:59 AM 10, 367, 424 rf_imgd.csv
09/01/2021 07:55 PM 592, 460 S85.ipynb
10/06/2021 03:42 PM 333, 604 S85_Incl.ipynb
10/06/2021 03:43 PM 317, 178 S85_Copy1.ipynb
09/13/2021 03:54 PM 34, 876, 686 Training_original.csv
09/14/2021 02:52 PM 46, 633, 045 TrainingVss.csv
09/01/2021 07:55 PM 2, 626, 454 TrainingVss_Inc.csv
09/14/2021 08:29 PM 48, 384 tree.dot
20 File(s) 127,338,273 bytes
3 Dir(s) 47,728,865,280 bytes free
```

Bibliotecas a utilizar

```
In [5]: import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')
```

Base de datos

```
In [8]: df = pd.read_csv("Training_original.csv")
In [9]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 287689 entries, 0 to 287688
Data columns (total 15 columns):
 # Column Non-Null Count Dtype
---
0 BPOS 287689 non-null float64
1 BVEL 287689 non-null float64
2 DEPT 287689 non-null float64
3 DateTime 287689 non-null object
4 GR 269090 non-null float64
5 HKLD 287689 non-null float64
6 ROPS 271935 non-null float64
7 RPM 287689 non-null float64
8 Run 287689 non-null object
9 SPPA 287689 non-null float64
10 SWOB 287689 non-null float64
11 Stick_RT 251095 non-null float64
12 TFLO 287689 non-null float64
13 TOA 287689 non-null float64
14 Well 287689 non-null object
dtypes: float64(12), object(3)
memory usage: 32.9+ MB
```

```
In [10]: ## Descripción estadística de la base de datos
```

```
In [11]: df.describe().transpose()
```

Out[11]:

	count	mean	std	min	25%	50%	75%	max
BPOS	287689.0	22.748777	12.572350	-1.4588	12.1300	22.7600	33.3732	53.3183
BVEL	287689.0	3.315810	9.876834	-1.960587	0.0014	0.0055	0.0100	490.3050
DEPT	287689.0	2.568207792	982.628218	2.069035	1908.2309	2760.1376	3298.9891	5330.7276
GR	269090.0	58.410470	47.832776	0.0000	18.0625	45.5625	89.3978	1111.8508
HKLD	287689.0	130.963900	18.702178	58.2302	121.2820	132.3900	141.1500	186.5285
ROPS	271935.0	19.989163	33.037100	0.6112	6.1901	16.9036	29.6500	1433.18210
RPM	287689.0	147.443451	42.581193	21.0000	131.0000	160.0000	170.0000	263.0000
SPPA	287689.0	195.613315	44.719825	-7.284633	178.6700	200.0174	224.0930	439.0982
SWOB	287689.0	5.559202	4.283104	-133.8011	2.6030	5.0872	7.6653	87.7975
Stick_RT	251095.0	48.299659	64.021792	0.0000	12.0000	24.0000	57.0000	381.0000
TFLO	287689.0	3084.866275	828.774211	44.3100	2214.9961	3491.4305	3500.4523	4583.4420
TOA	287689.0	14.760415	13.309971	-971.0449	10.9175	15.8170	19.7612	34.8810

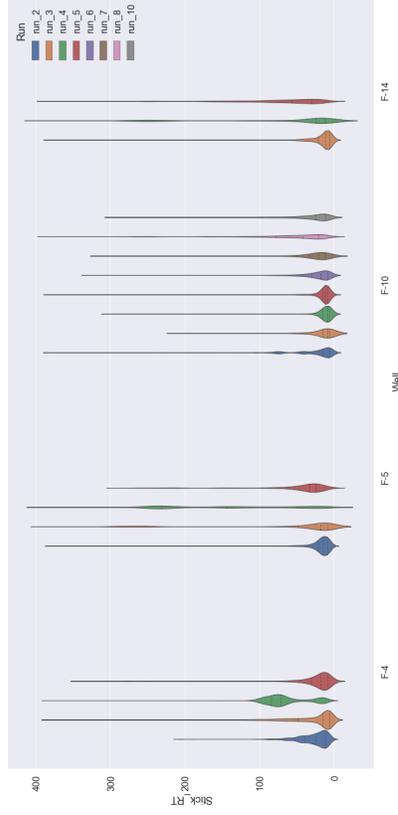
```
df.drop(["DateTime"],axis=1,inplace=True)
```

Análisis de gráfico de violín

por corrida

```
In [13]: sns.set(font_scale=1.5)
plt.figure(figsize=(20,10))
sns.violinplot(data=df.drop(["DateTime"],axis=1), x="Well", y="Stick_RT", hue="Run", i
```

```
sns.despine(left=True)
plt.tight_layout()
```



Identificación de vibraciones por seccion

```
In [15]: df_f10 = df[df["Well"]=="F-10"]
```

```
In [16]: Runs = df_f10["Run"].tolist()
sections = []
for element in Runs:
    if element == "run_2":
        v = '26"'
    elif element == "run_3" or element == "run_4":
        v = '17 1/2"'
    elif element == "run_5" or element == "run_6" or element == "run_7":
        v = '12 1/4"'
    elif element == "run_8" or element == "run_9" or element == "run_10":
        v = '8 1/2"'
    sections.append(v)
```

```
In [17]: df_f10["Sections"] = sections
```

```
<ipython-input-17-b10f3f0edfa5>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
df_f10["Sections"] = sections
```

```
In [18]: df_f4 = df[df["Well"]=="F-4"]
```

```
In [19]: Runs = df_f4["Run"].tolist()
sections = []
for element in Runs:
    if element == "run_2":
        v = '17 1/2"' #Double check
    elif element == "run_3":
        v = '12 1/4"'
```

```
elif element == "run_4" or element == "run_5":
    v = '8 1/2"'
sections.append(v)
```

```
In [20]: df_f4["Sections"] = sections
```

```
<ipython-input-20-eb8c15972f62>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
df_f4["Sections"] = sections
```

```
In [21]: df_f5 = df[df["Well"]=="F-5"]
```

```
In [22]: Runs = df_f5["Run"].tolist()
sections = []
for element in Runs:
    if element == "run_2":
        v = '17 1/2"'
    elif element == "run_3":
        v = '12 1/4"'
    elif element == "run_4" or element == "run_5":
        v = '8 1/2"'
    sections.append(v)
```

```
In [23]: df_f5["Sections"] = sections
```

```
<ipython-input-23-ed1f17070074>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
df_f5["Sections"] = sections
```

```
In [24]: df_f14 = df[df["Well"]=="F-14"]
```

```
In [25]: Runs = df_f14["Run"].tolist()
sections = []
for element in Runs:
    if element == "run_3":
        v = '17 1/2"'
    elif element == "run_4":
        v = '12 1/4"'
    elif element == "run_5":
        v = '8 1/2"'
    sections.append(v)
```

```
In [26]: df_f14["Sections"] = sections
```

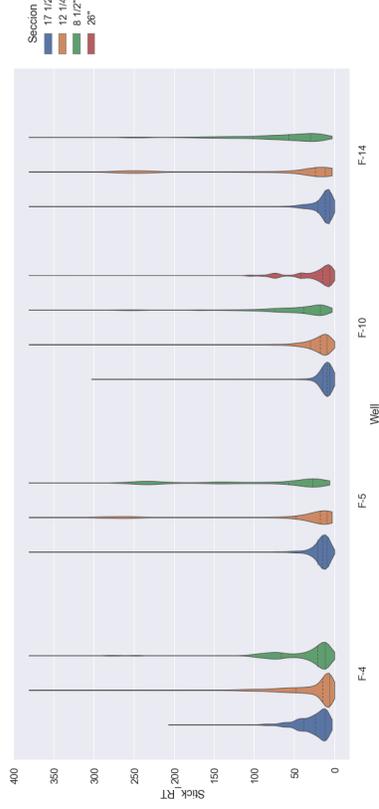
```
<ipython-input-26-7a1eed7a2cbb>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
df_f14["Sections"] = sections
```

```
In [27]: frames = [df_f4, df_f5, df_f10, df_f14]
df_mod = pd.concat(frames)
```

Por pozo y seccion

```
In [29]: sns.set(font_scale=1.5)
plt.figure(figsize=(20, 10))
sns.violinplot(data=df_mod.drop(["DateTime"], axis=1), x="Well", y="Stick_RT", hue="Seccion",
#plt.legend(loc="upper left", title="Seccion")
sns.despine(left=True)
```



Severidad de Stick & Slip

```
In [30]: df_ss = sc.ss_severity(df_mod)
```

```
In [31]: df_ss.info()
```

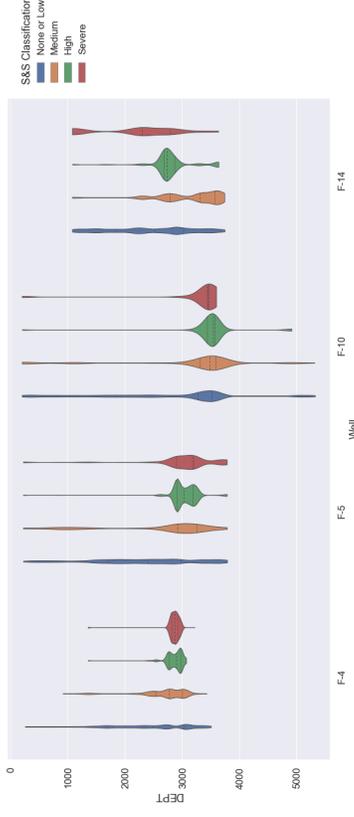
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 287689 entries, 0 to 287688
Data columns (total 19 columns):
 # Column Non-Null Count Dtype
---
 0 BPOS 287689 non-null float64
 1 BVEL 287689 non-null float64
 2 DEPT 287689 non-null float64
 3 DateTime 287689 non-null object
 4 GR 269090 non-null float64
 5 HKLD 287689 non-null float64
 6 ROPS 271935 non-null float64
 7 RPM 287689 non-null float64
 8 Run 287689 non-null object
 9 SPPA 287689 non-null float64
10 SWOB 287689 non-null float64
11 Stick_RT 251095 non-null float64
12 TFLO 287689 non-null float64
13 TQA 287689 non-null float64
14 Well 287689 non-null object
15 Sections 287689 non-null object
16 ss_severity 251095 non-null float64
```

```
17 ss_categorical 287689 non-null int64
18 ss_categorical_txt 287689 non-null object
dtypes: float64(13), int64(1), object(5)
memory usage: 53.9+ MB
```

Severidad por pozo

```
In [32]: sns.set(font_scale=1.5)
plt.figure(figsize=(20, 10))
sns.violinplot(data=(df_ss.drop(["DateTime"], axis=1)), x="Well", y="DEPT", hue="ss_cate",
plt.gca().invert_yaxis()
plt.legend(bbox_to_anchor=(1, 1), loc='upper left', borderaxespad=1, title='S&S Classifi')
#sns.despine(left=True)
```

```
Out[32]: <matplotlib.legend.Legend at 0x2be69e45ac0>
```



Base de datos con parámetros adicionales de vibración

```
df_ss.to_csv(r"TrainingVss.csv", index=False)
```

```
In [1]: cd ~/Desktop/Tesis/Notebooks-Git
```

```
C:\Users\LuisH\Desktop\Tesis\Notebooks-Git
```

```
In [2]: import script as sc
```

Random Forest

for mwd and survey dataset

```
In [3]: cd ~/Desktop/Tesis/Notebooks-Git/Training/SticksSlip
```

```
C:\Users\LuisH\Desktop\Tesis\Notebooks-Git\Training\SticksSlip
```

```
In [4]: import pandas as pd
```

```
import os
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

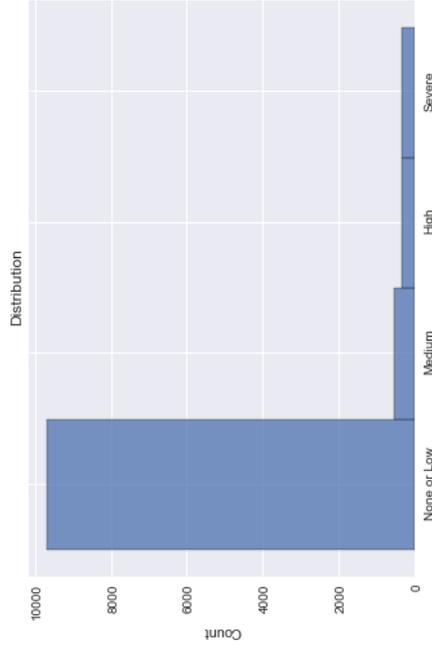
```
plt.style.use('seaborn-darkgrid')
```

```
import seaborn as sns
```

```
In [5]: df = pd.read_csv("TrainingVss_Inc.csv")
```

```
In [6]: df.rename(columns = {"azi": "AZI", "incl": "INCL", "dls": "DLS"}, inplace=True)
```

```
In [7]: sns.histplot(df['ss_categorical_txt'], discrete=True)
plt.title("Distribution")
plt.legend()
plt.show()
```



```
In [8]: plt.figure(figsize=(15,10))
sns.heatmap(df.isnull(), cbar=False)
plt.show()
```

```
202
606
808
1010
1212
1414
1616
1818
2020
2222
2424
2626
2828
3030
3232
3434
3636
3838
4040
4242
4444
4646
4848
5050
5252
5454
5656
5858
6060
6262
6464
6666
6868
7070
7272
7474
7676
7878
8080
8282
8484
8686
8888
9090
9292
9494
9696
9898
10100
10302
10504
10706
```

```
EPoS
BEVL
DEPT
DateTime
GR
HQLD
ROP5
RPM
SPPA
SWOB
Stick_RT
TFLO
TQA
WELL
AZI
dispEW
display
DLS
NCL
NID
Sections
ss_severity
ss_categorical_txt
ss_categorical_txt
```

```
In [9]: df.dropna(inplace=True)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 10595 entries, 0 to 10892
```

```
Data columns (total 25 columns):
```

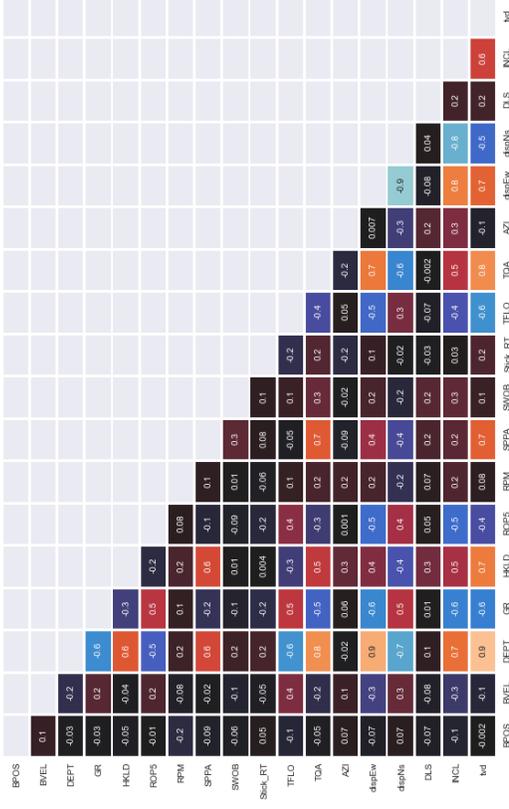
#	Column	Non-Null Count	Dtype
0	BPOS	10595 non-null	float64
1	BEVL	10595 non-null	float64
2	DEPT	10595 non-null	float64
3	DateTime	10595 non-null	object
4	GR	10595 non-null	float64
5	HQLD	10595 non-null	float64
6	ROP5	10595 non-null	float64
7	RPM	10595 non-null	float64
8	Run	10595 non-null	object
9	SPPA	10595 non-null	float64
10	SWOB	10595 non-null	float64
11	Stick_RT	10595 non-null	float64
12	TFLO	10595 non-null	float64
13	TQA	10595 non-null	float64
14	Well	10595 non-null	object
15	AZI	10595 non-null	float64
16	dispEW	10595 non-null	float64
17	dispNS	10595 non-null	float64
18	DLS	10595 non-null	float64
19	INCL	10595 non-null	float64
20	tvd	10595 non-null	float64
21	Sections	10595 non-null	object
22	ss_severity	10595 non-null	float64
23	ss_categorical	10595 non-null	int64
24	ss_categorical_txt	10595 non-null	object

```
dtypes: float64(19), int64(1), object(5)
memory usage: 2.1+ MB
```

```
In [10]: #df.drop(['dispEw', 'dispNs', 'tvd'], axis=1, inplace=True)
df.columns
```

```
Out[10]: Index(['BPOS', 'BVEL', 'DEPT', 'DateTime', 'GR', 'HKLD', 'ROP5', 'RPM', 'Run',
      'SPPA', 'SWOB', 'Stick_RT', 'TFLO', 'TQA', 'Well', 'AZI', 'dispEw',
      'dispNs', 'DLS', 'INCL', 'tvd', 'Sections', 'ss_severity',
      'ss_categorical', 'ss_categorical_txt'],
      dtype='object')
```

```
In [11]: plt.figure(figsize=(15,10))
matrix = np.triu(df.drop(['ss_severity', 'ss_categorical'], axis=1).corr())
sns.heatmap(df.drop(['ss_severity', 'ss_categorical'], axis=1).corr(), annot=True, mask=
plt.show())
```



```
In [12]: df.columns
```

```
Out[12]: Index(['BPOS', 'BVEL', 'DEPT', 'DateTime', 'GR', 'HKLD', 'ROP5', 'RPM', 'Run',
      'SPPA', 'SWOB', 'Stick_RT', 'TFLO', 'TQA', 'Well', 'AZI', 'dispEw',
      'dispNs', 'DLS', 'INCL', 'tvd', 'Sections', 'ss_severity',
      'ss_categorical', 'ss_categorical_txt'],
      dtype='object')
```

```
In [13]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10595 entries, 0 to 10892
Data columns (total 25 columns):
 #   Column              Non-Null Count  Dtype
---  --
 0   BPOS                10595 non-null  float64
 1   BVEL                10595 non-null  float64
 2   DEPT                10595 non-null  float64
 3   DateTime            10595 non-null  object
```

```
4   GR                10595 non-null  float64
5   HKLD              10595 non-null  float64
6   ROP5              10595 non-null  float64
7   RPM               10595 non-null  float64
8   Run               10595 non-null  object
9   SPPA              10595 non-null  float64
10  SWOB              10595 non-null  float64
11  Stick_RT          10595 non-null  float64
12  TFLO              10595 non-null  float64
13  TQA               10595 non-null  float64
14  Well              10595 non-null  object
15  AZI               10595 non-null  float64
16  dispEw            10595 non-null  float64
17  dispNs            10595 non-null  float64
18  DLS               10595 non-null  float64
19  INCL              10595 non-null  float64
20  tvd               10595 non-null  float64
21  Sections          10595 non-null  object
22  ss_severity       10595 non-null  float64
23  ss_categorical    10595 non-null  int64
24  ss_categorical_txt 10595 non-null  object
dtypes: float64(19), int64(1), object(5)
memory usage: 2.1+ MB
```

```
In [14]: df.columns
```

```
Out[14]: Index(['BPOS', 'BVEL', 'DEPT', 'DateTime', 'GR', 'HKLD', 'ROP5', 'RPM', 'Run',
      'SPPA', 'SWOB', 'Stick_RT', 'TFLO', 'TQA', 'Well', 'AZI', 'dispEw',
      'dispNs', 'DLS', 'INCL', 'tvd', 'Sections', 'ss_severity',
      'ss_categorical', 'ss_categorical_txt'],
      dtype='object')
```

Splitting sets

```
In [15]: from sklearn.model_selection import train_test_split
```

```
In [16]: x_cols = ['BPOS', 'BVEL', 'DEPT', 'GR', 'HKLD', 'ROP5', 'RPM', 'SPPA', 'SWOB', 'TFLO',
      '#x_cols = df.columns.difference(['Well', 'Sections', 'ss_severity', 'ss_categorical', 'ss_c
```

```
In [17]: x_cols
```

```
Out[17]: ['BPOS',
      'BVEL',
      'DEPT',
      'GR',
      'HKLD',
      'ROP5',
      'RPM',
      'SPPA',
      'SWOB',
      'TFLO',
      'TQA',
      'AZI',
      'DLS',
      'INCL']
```

```
In [18]: X = df[x_cols]
      Y = df["ss_categorical"]
      x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=.2, random_state=2)
```

```
In [19]: print("Predictores:\n\t", x_cols[0:7])
```

```
print("\t", x_cois[7:])
print("\tTamaños:\t")
print("\t\tTraining:\t\t\t\t", x_train.shape, "\t\t\t\t", y_train.shape)
print("\t\tTesting:\t\t\t\t", x_test.shape, "\t\t\t\t", y_test.shape)
```

Predictores:

```
['BPOS', 'BVEL', 'DEPT', 'GR', 'HKLD', 'ROPS', 'RPM']
['SPPA', 'SWOB', 'TFLO', 'TOA', 'AZI', 'DLS', 'INCL']
```

Tamaños:

```
Training: X: (8476, 14)
          Y: (8476,)
Testing:  X: (2119, 14)
          Y: (2119,)
```

Random Forest Classifier

for training set

```
In [20]: from sklearn.ensemble import RandomForestClassifier
         from sklearn.metrics import accuracy_score
```

```
In [23]: %%time
         rf = RandomForestClassifier(n_estimators=50, max_depth=None, min_samples_split=2,
                                   min_samples_leaf=2, max_features="auto", bootstrap=True,
                                   n_jobs=-1, random_state=2).fit(x_train, y_train)
```

Wall time: 902 ms

```
In [24]: y_pred = rf.predict(x_test)
```

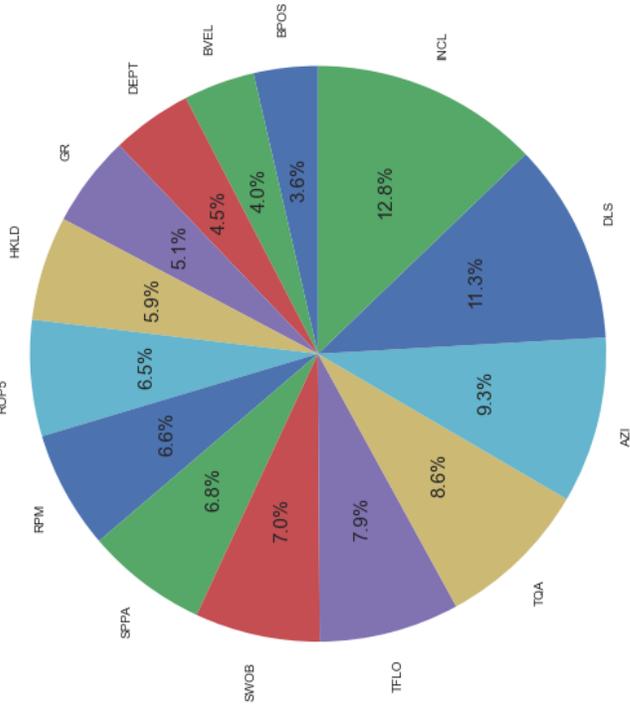
```
In [25]: print("Precision del modelo:", round((accuracy_score(y_test, y_pred)*100), 4), "%")
```

Precision del modelo: 97.3572 %

```
In [27]: feat = np.sort(rf.feature_importances_)
```

```
In [28]: plt.figure(figsize=(10, 15))
         plt.rcParams.update({'font.size': 15})
         plt.pie(feat, labels=x_cois, autopct='%1.1f%%')
         plt.title("Feature importance")
         plt.show()
```

Feature importance

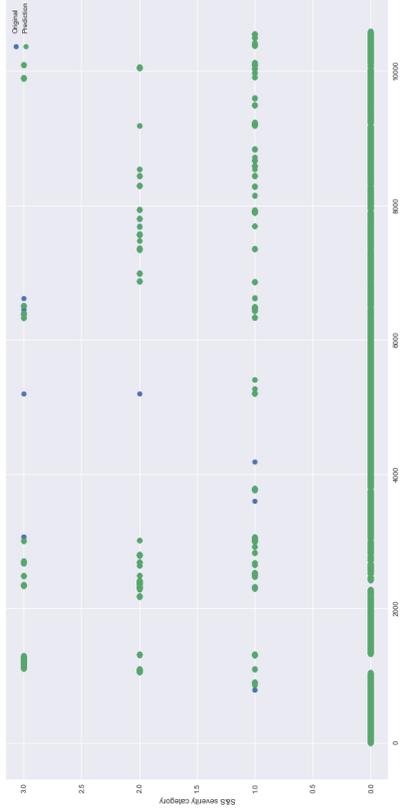


Prediccion

```
In [29]: df_rf = df
         df_rf["Count"] = np.arange(len(df))
```

```
In [30]: df_rf["rf"] = rf.predict(X)
```

```
In [31]: plt.figure(figsize=(20, 10))
         plt.scatter(df_rf["Count"], df_rf["ss_categorical"], label="Original")
         plt.scatter(df_rf["Count"], df_rf["rf"], label="Prediction")
         plt.ylabel("$S&S$ severity category")
         plt.legend()
         plt.show()
```



In [32]:

```
class_names = df["ss_categorical_txt"].unique()
```

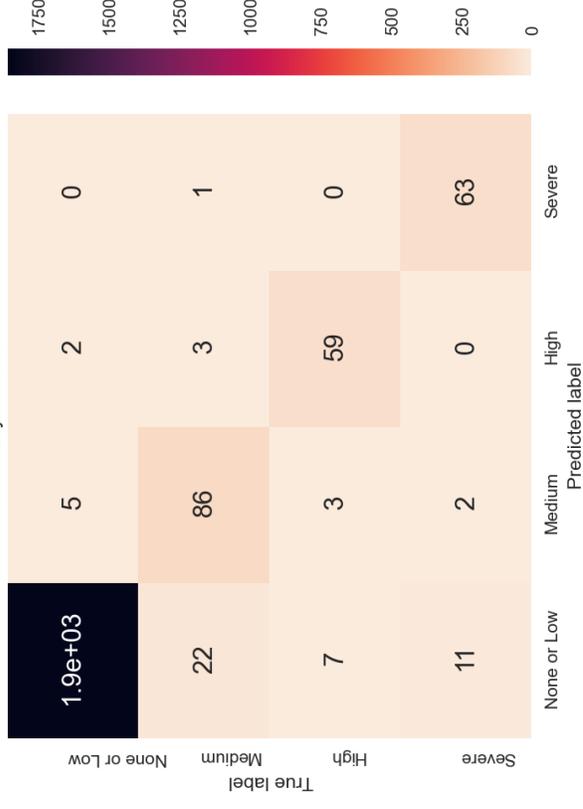
In [33]:

```
from sklearn.metrics import accuracy_score, confusion_matrix, precision_recall_fscore_s
cm = confusion_matrix(y_test, y_pred)
cm_df = pd.DataFrame(cm, index = [class_names], columns = [class_names])
```

In [34]:

```
plt.figure(dpi=150)
sns.heatmap(cm_df, annot=True, cmap="rocket_r")
plt.title('Random Forest Classifier \nAccuracy: {:.4f}'.format(accuracy_score(y_test, y_pred)))
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.show()
```

Random Forest Classifier
Accuracy:0.9736



Blind Set

predicting F15A well

In [35]:

```
b1 = pd.read_csv(r"Blind_F15A_incl.csv")
```

In [36]:

```
b1.columns
```

Out[36]:

```
Index(['DEPT', 'disps', 'dispEw', 'INCL', 'tyd', 'DLS', 'AZI', 'ROP5', 'GR', 'BPOS', 'BVEL', 'SWOB', 'TQA', 'RPM', 'Stick_RT', 'TFLO', 'SPPA', 'ECD', 'HKLD', 'Dateime', 'Well', 'Run', 'ss_severity', 'ss_categorical', 'ss_categorical_txt'], dtype='object')
```

In [37]:

```
b1_predict = pd.DataFrame(b1, columns=x_test.columns)
```

In [38]:

```
print("Predictors:\n\t", b1_predict.columns.values)
print("\nSet shape:", b1_predict.shape)
```

Predictors:

```
['BPOS', 'BVEL', 'DEPT', 'GR', 'HKLD', 'ROP5', 'RPM', 'SPPA', 'SWOB', 'TFLO', 'TQA', 'AZI', 'DLS', 'INCL']
```

Set shape: (1089, 14)

In [39]:

```
y_b1 = pd.to_numeric(b1["ss_categorical"])
```

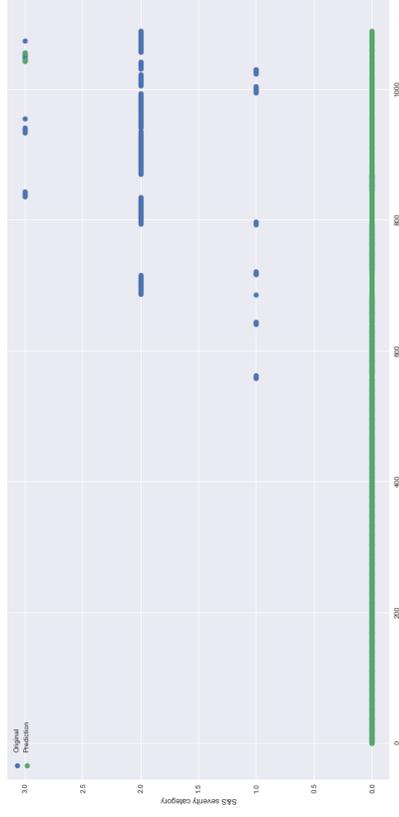
```
In [40]: bl_pred = rf.predict(bl_predict)
bl["Prediction"] = bl_pred
```

```
In [41]: print("Accuracy: ", round((accuracy_score(y_bl, bl_pred)*100),4), "%")
```

Accuracy: 71.3499 %

```
In [42]: bl["Count"] = np.arange(len(bl))
```

```
In [43]: plt.figure(figsize = (20,10))
plt.scatter(bl["Count"],bl["ss_categorical"],label="Original")
plt.scatter(bl["Count"],bl["Prediction"],label="Prediction")
plt.ylabel("SS&S severity category")
plt.legend()
plt.show()
```



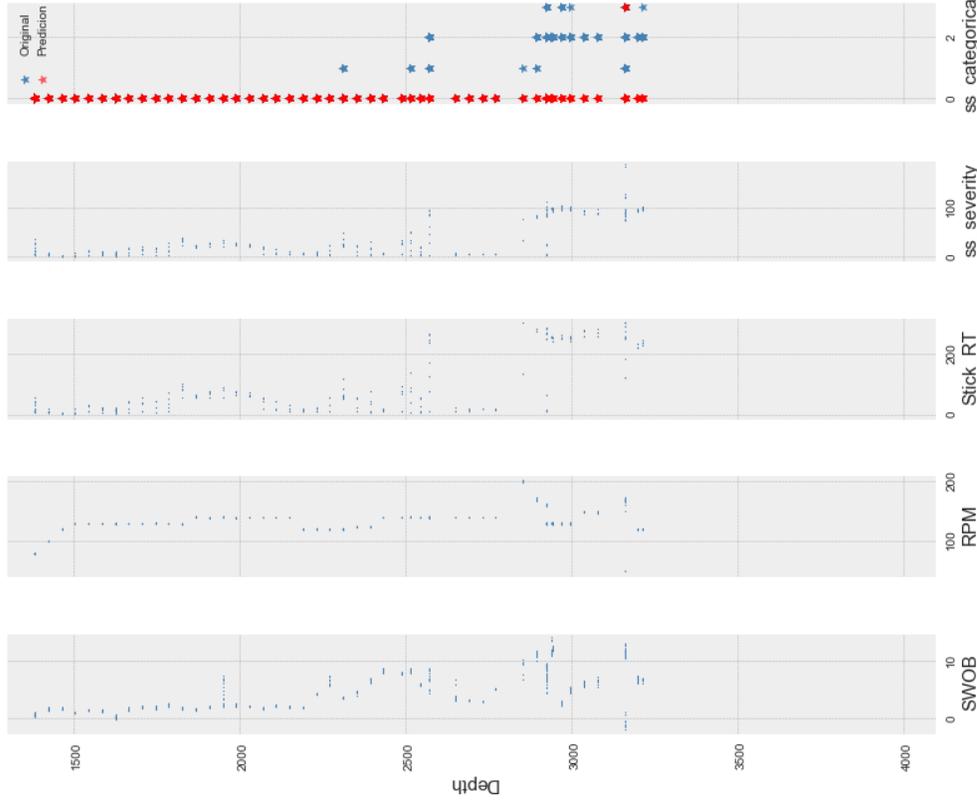
```
In [44]: def depth_plts(data,list_):
plt.style.use("bmh")
plt.rcParams.update({'font.size': 40})
fig,axes = plt.subplots(1,len(list_),figsize=(10,12),sharey=True)
for i,log in enumerate(list_):
axes[i].scatter(data[list_[i]],data["DEPT"],marker = "*",s = 0.5,color="steelblue")
axes[i].set_xlabel(log,fontsize=15)
axes[i].invert_yaxis()
if log == "ss_categorical":
axes[i].set_ylim(4095.6890000000003, 1300)
axes[i].scatter(data[list_[i]],data["DEPT"],marker = "*",color="steelblue")
axes[i].scatter(data["Prediction"],data["DEPT"],marker = "*",label = "Predi")
axes[i].set_xlabel(log, fontsize=15)
axes[i].legend()
#axes[i].invert_yaxis()

axes[0].set_ylabel("Depth", fontsize=15)
plt.tight_layout()
```

```
In [45]: (4095.6890000000003, 1381.0034)
```

```
Out[45]: (4095.6890000000003, 1381.0034)
```

```
In [46]: depth_plts(bl,["SWOB", "RPM", "Stick_RT", "ss_severity", "ss_categorical"])
```



```
blto_csv("rf_incl.csv")
```

```
In [1]: cd ~/Desktop/Tesis/Notebooks-Git
C:\Users\LuisH\Desktop\Tesis\Notebooks-Git
In [2]: import script as sc
```

Random Forest

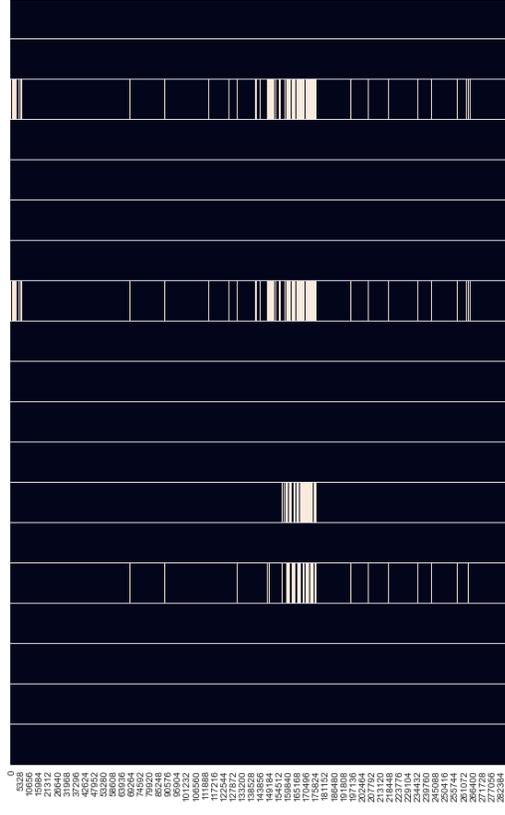
for MMD dataset

```
In [3]: cd ~/Desktop/Tesis/Notebooks-Git/Training/SticksSlip
C:\Users\LuisH\Desktop\Tesis\Notebooks-Git\Training\SticksSlip
```

```
In [4]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')
import seaborn as sns
```

```
In [5]: df = pd.read_csv("TrainingVss.csv")
```

```
In [6]: plt.figure(figsize=(15,10))
sns.heatmap(df.isnull(), cbar=False)
plt.show()
```



```
In [7]: df.dropna(inplace=True)
localhost:8888/nbconvert//Desktop/Tesis/Notebooks-Git/Training/SticksSlip/RandomForest.ipynb?download=false
```

```
In [8]: plt.figure(figsize=(15,10))
matrix = np.triu(df.corr())
sns.heatmap(df.corr(), annot=True, mask=matrix, fmt = '.1g', linewidths=2, vmin=-1, vmax=1
plt.show()
```



```
In [9]: df.columns
```

```
Out[9]: Index(['BPOS', 'BVEL', 'DEPT', 'DateTime', 'GR', 'HKLD', 'ROPS', 'RPM', 'Run',
'SPPA', 'SWOB', 'Stick_RT', 'TFO', 'TQA', 'Well', 'Sections',
'ss_severity', 'ss_categorical', 'ss_categorical_txt'],
dtype='object')
```

Splitting sets

```
In [10]: from sklearn.model_selection import train_test_split
```

```
In [11]: #X_cols = df.columns.difference(['Well', 'Sections', 'ss_severity', 'ss_categorical', 'ss_c
x_cols = ['Stick_RT', 'BPOS', 'BVEL', 'DEPT', 'GR', 'HKLD', 'ROPS', 'RPM', 'SPPA', 'SWOB', 'TFO',
```

```
In [12]: X = df[x_cols]
Y = df["ss_categorical"]
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=.2, random_state=2)
```

```
In [13]: print("Predictors:\n\t", x_cols)
print("\nset shapes:\t")
print("\tTraining:\n\t\t\t", x_train.shape, "\n\t\t\t", y_train.shape)
print("\tTesting:\n\t\t\t", x_test.shape, "\n\t\t\t", y_test.shape)
```

```
Predictors:
['Stick_RT', 'BPOS', 'BVEL', 'DEPT', 'GR', 'HKLD', 'ROPS', 'RPM', 'SPPA', 'SMO
B', 'TFO', 'TQA']
```

```
Set shapes:
```

```

Training: X: (197823, 12)
          Y: (197823,)
Testing:  X: (49456, 12)
          Y: (49456,)

```

Random Forest Classifier

for training set

```

In [14]: from sklearn.ensemble import RandomForestClassifier
         from sklearn.metrics import accuracy_score

In [15]: %%time
         rf = RandomForestClassifier(n_estimators=50,max_depth=None, min_samples_split=4, min_sa
Wall time: 31.1 s

In [16]: y_pred = rf.predict(x_test)

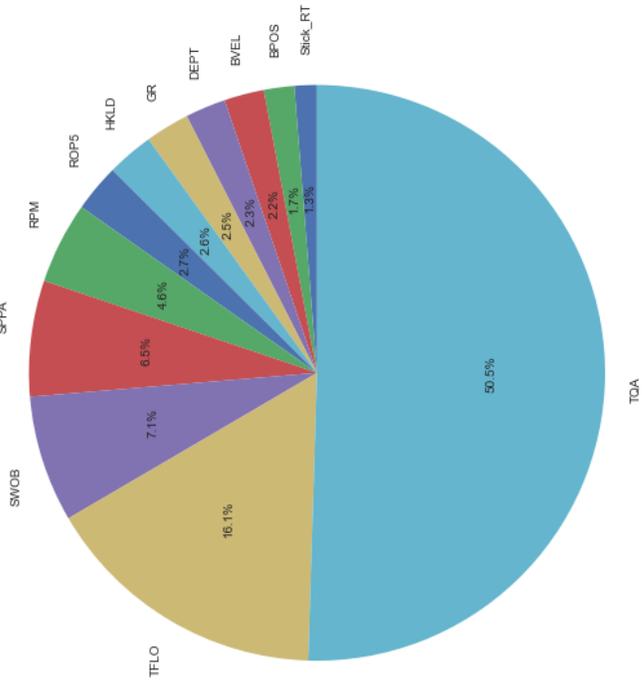
In [17]: print("Accuracy: ", round((accuracy_score(y_test, y_pred)*100),4), "%")
Accuracy: 99.8807 %

In [18]: feat = np.sort(rf.feature_importances_)

In [19]: plt.figure(figsize=(10,15))
         plt.pie(feat,labels=x_cols,autopct= '%1.1f%%')
         plt.title("Feature importance")
         plt.show()

```

Feature importance



Prediction

```

In [20]: df_rf = df
         df_rf["Count"] = np.arange(len(df))

In [21]: df_rf["rf"] = rf.predict(X)

In [22]: plt.figure(figsize=(20,10))
         plt.scatter(df_rf["Count"],df_rf["ss_categorical"],label="Original")
         plt.scatter(df_rf["Count"],df_rf["rf"],label="Prediction")
         plt.ylabel("S&S severity category")
         plt.legend()
         plt.show()

```



In [23]:

```
class_names = df["ss_categorical_txt"].unique()
```

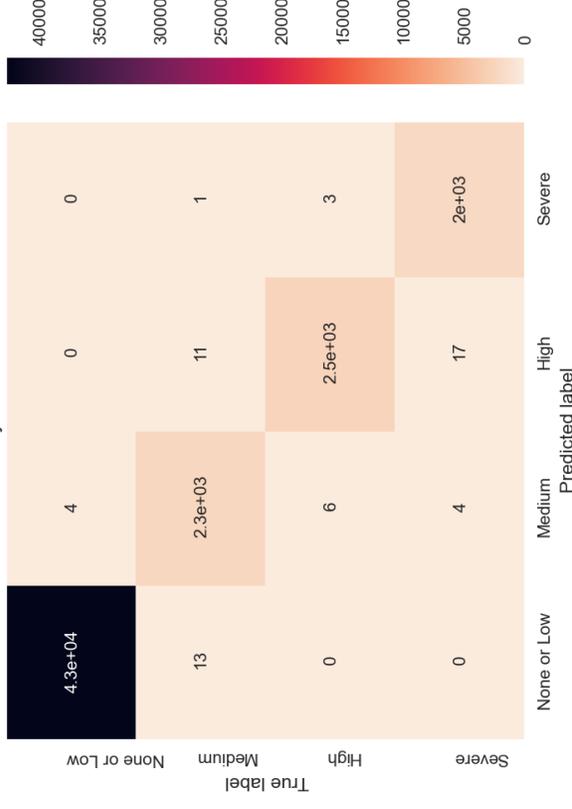
In [24]:

```
from sklearn.metrics import accuracy_score, confusion_matrix, precision_recall_fscore_s
cm = confusion_matrix(y_test, y_pred)
cm_df = pd.DataFrame(cm, index = [class_names], columns = [class_names])
```

In [25]:

```
plt.figure(dpi=150)
sns.heatmap(cm_df, annot=True, cmap="rocket_r")
plt.title('Random Forest Classifier \nAccuracy: {:.4f}'.format(accuracy_score(y_test, y_pred)))
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.show()
```

Random Forest Classifier
Accuracy:0.9988



Blind Set

predicting F-15A well

In [26]:

```
b1 = pd.read_csv(r"Blind_F15A.csv")
```

In [27]:

```
b1.head()
```

Out[27]:

	DEPT	ROP5	GR	BPOS	BVEL	SWOB	TQA	RPM	Stick_RT	TFLO	SPPA	ECD	HI
0	1381.0034	3.494	36.000	25.013	0.001	0.624	7.540	800	57.0	4010.012	188.061	1.431	124.
1	1381.0164	3.494	36.000	25.000	0.001	0.604	7.540	800	57.0	4010.012	188.061	1.431	124.
2	1381.0339	3.494	43.891	24.983	0.001	0.592	6.970	800	12.0	4010.012	188.807	1.432	124.
3	1381.0464	3.494	43.891	24.970	0.001	0.685	7.062	800	12.0	4010.012	188.764	1.431	124.

```
DEPT  ROP5  GR  BPOS  BVEL  SWOB  TQA  RPM  Stick_RT  TFLO  SPPA  ECD  HI
```

```
4  1381.0597  3.979  43.891  24.957  0.001  0.549  7.787  800  12.0  4010.012  188.782  1.431  124.
```

```
< | >
```

```
In [28]: bl_predict = pd.DataFrame(bl, columns=x_test.columns)
```

```
In [29]: bl_predict.head()
```

```
Out[29]:
```

	Stick_RT	BPOS	BVEL	DEPT	GR	HKLD	ROP5	RPM	SPPA	SWOB	TFLO	TQA
0	57.0	25.013	0.001	1381.0034	36.000	124.219	3.494	800	188.061	0.624	4010.012	7.540
1	57.0	25.000	0.001	1381.0164	36.000	124.239	3.494	800	188.061	0.604	4010.012	7.540
2	12.0	24.983	0.001	1381.0339	43.891	124.251	3.494	800	188.807	0.592	4010.012	6.970
3	12.0	24.970	0.001	1381.0464	43.891	124.159	3.494	800	188.764	0.685	4010.012	7.062
4	12.0	24.957	0.001	1381.0597	43.891	124.294	3.979	800	188.782	0.549	4010.012	7.787

```
In [30]: print("Predictors:\n\t", bl_predict.columns.values)
print("\nSet shape:", bl_predict.shape)
```

```
Predictors:
['Stick_RT' 'BPOS' 'BVEL' 'DEPT' 'GR' 'HKLD' 'ROP5' 'RPM' 'SPPA' 'SWOB'
'TFLO' 'TQA']
```

```
Set shape: (53505, 12)
```

```
In [31]: y_bl = pd.to_numeric(bl["ss_categorical"])
```

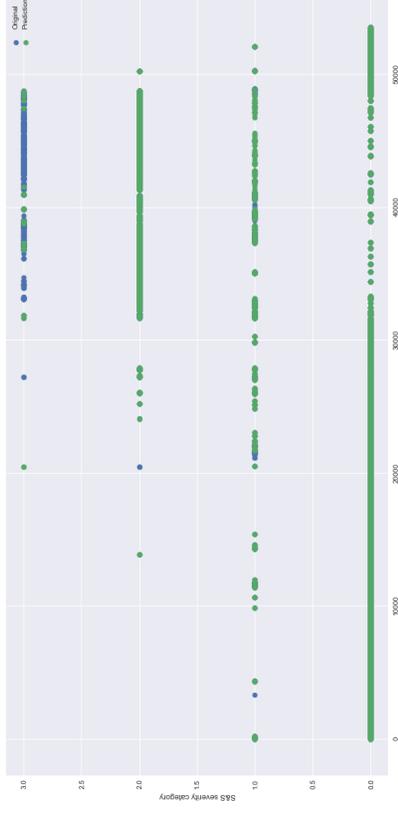
```
In [32]: bl_pred = rf.predict(bl_predict)
bl["Prediction"] = bl_pred
```

```
In [33]: print("Accuracy: ", round((accuracy_score(y_bl, bl_pred)*100),4), "%")
```

```
Accuracy: 94.2547 %
```

```
In [34]: bl["Count"] = np.arange(len(bl))
```

```
In [35]: plt.figure(figsize = (20,10))
plt.scatter(bl["Count"], bl["ss_categorical"], label="Original")
plt.scatter(bl["Count"], bl["Prediction"], label="Prediction")
plt.ylabel("S&S severity category")
plt.legend()
plt.show()
```



```
In [36]: bl["DEPT"].max(), bl["DEPT"].min()
```

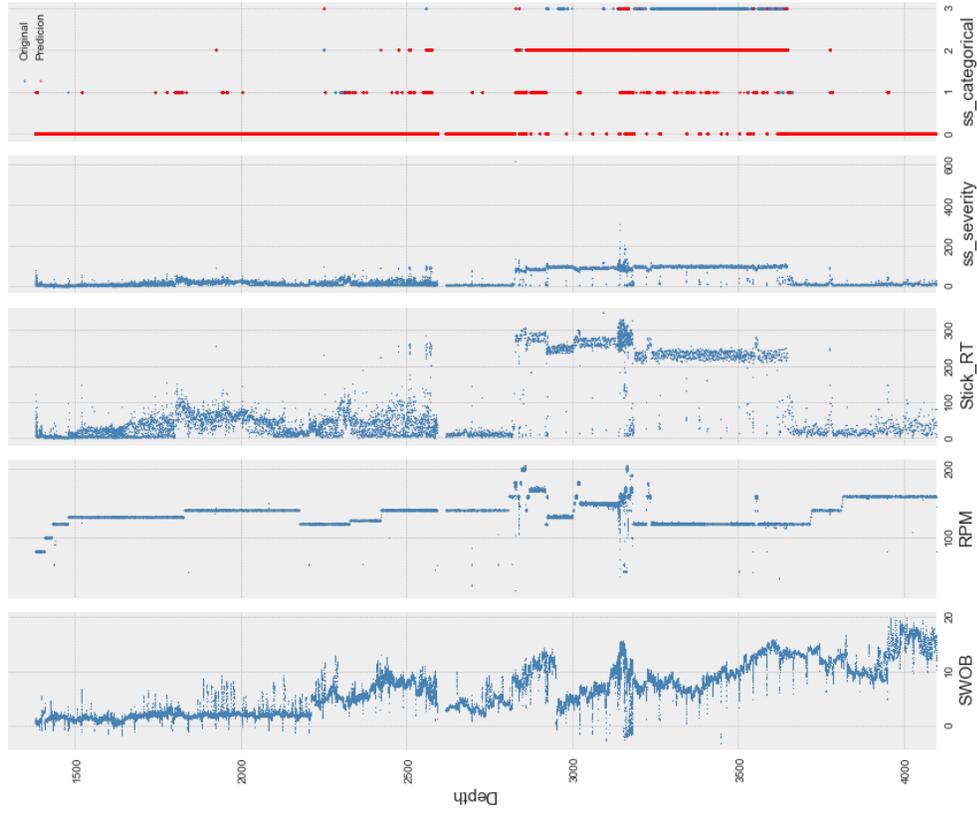
```
Out[36]: (4095.6890000000003, 1381.0034)
```

```
In [37]: def depth_plts(data, list_):
plt.style.use("bmh")
fig, axes = plt.subplots(1, len(list_), figsize=(10, 12), sharey=True)
for i, log in enumerate(list_):
axes[i].set_ylim(4095.6890000000003, 1300)
axes[i].scatter(data[list_[i]], data["DEPT"], marker = "*", s = 0.5, color="steelblue")
axes[i].set_xlabel(log, fontsize=15)
axes[i].invert_yaxis()
if log == "ss_categorical":
axes[i].set_ylim(4095.6890000000003, 1300)
axes[i].scatter(data[list_[i]], data["DEPT"], marker = "*", color="steelblue",
axes[i].scatter(data["Prediction"], data["DEPT"], marker = "*", label = "Predi
axes[i].set_xlabel(log, fontsize=15)
axes[i].legend(loc="upper right", fontsize=10)
#axes[i].invert_yaxis()

axes[0].set_ylabel("Depth", fontsize=15)
plt.tight_layout()
```

```
In [38]: depth_plts(bl, ["SWOB", "RPM", "Stick_RT", "ss_severity", "ss_categorical"])
```

RandomForest



```
In [39]: b1.to_csv("rf_mwd.csv")
```

```
In [ ]:
```