# PREDICCIÓN DEL RUN-LIFE EN BOMBAS DE CAVIDADES PROGRESIVAS EMPLEANDO INTELIGENCIA ARTIFICIAL EN EL CAMPO CASABE

**ELISA MARÍA ANGULO VANEGAS**

**ALVARO ANDRÉS MARTÍN ROJAS**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER**

**FACULTAD DE INGENIERÍAS FISICOQUÍMICAS**

**ESCUELA DE INGENIERÍA DE PETRÓLEOS**

**BUCARAMANGA**

**2020**

# PREDICCIÓN DEL RUN-LIFE EN BOMBAS DE CAVIDADES PROGRESIVAS EMPLEANDO INTELIGENCIA ARTIFICIAL EN EL CAMPO CASABE

**ELISA MARÍA ANGULO VANEGAS**

**ALVARO ANDRÉS MARTÍN ROJAS**

**Trabajo de grado para optar por el título de Ingeniero de Petróleos**

**Director**

**EDISON ODILIO GARCIA NAVAS**

**Ingeniería de Hidrocarburos, M.sc**

**Codirector**

**JORGE MARIO DOVAL DORADO**

**Ingeniero de Petróleos**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER**

**FACULTAD DE INGENIERÍAS FISICOQUÍMICAS**

**ESCUELA DE INGENIERÍA DE PETRÓLEOS**

**BUCARAMANGA**

**2020**

## DEDICATORIA

*Principalmente agradezco a Dios, por todas las bendiciones que ha puesto en mi camino, por darme día a día un motivo para sonreír y disfrutar de este hermoso viaje llamado vida. Todo el honor y la gloria son para ti Padre Santo.*

*A mi madre Alix Vanegas Zambrano, estoy eternamente agradecida contigo, muchas gracias por confiar en mí, apoyarme, ser mi bastón que me permite seguir adelante y nunca desfallecer.*

*A mi padre Adolfo Angulo Espitia, Padre lindo, aunque ahora no estas con nosotros en este mundo te quiero agradecer por todo el amor que me brindaste, por ser mi ejemplo a seguir, mi consejero y mi amigo. Te amare por siempre.*

*A mis hermanos, Adolfo y Andrés por ser tan especiales, cada uno en su forma me completa, me apoya y me hace muy feliz.*

*A toda mi familia mis abuelos, mis tías, tíos y primos, por brindarme todo su amor y estar conmigo en cada momento de mi vida. Los adoro mucho y estoy eternamente agradecida con cada uno.*

*A todos mis amigos, en especial a aquellos que se han convertido en familia y a quienes les tengo un cariño inmenso: Fernanda Pachón, Julián Moreno, Alex Parra, Anderzon Gonzales, Wilmar Ramírez, Tatiana Muñoz, Camilo Parra, Gracias por cada momento vivido con ustedes, los tendré siempre en mi corazón*

*A las personas que brindaron su apoyo para la realización de nuestro proyecto de grado, en especial a mi director de tesis el ingeniero Edison Odilio García, Camilo Valenzuela, Felipe Consuegra, al Ingeniero Santander Bernal y a los ingenieros de producción de Ecopetrol en Casabe especialmente, al ingeniero Ariel Patiño, Jorge Doval y Hoover Arce.*

*Elisa María Angulo Vanegas*

*A Dios por darme la oportunidad de llegar hasta aquí y continuar este camino de la vida.*

*A mi madre Martha, mi padre Alvaro y mis hermanos Andrea y Oscar, que me han apoyado incondicionalmente durante toda mi vida. Sin ellos, no estaría aquí.*

*A mi Alma Mater, a cada uno de los profesores, compañeros y amigos, que aportaron su granito de arena para poder realizar este proyecto y que aportaron con mi formación académica.*

*La vida es una y el tiempo premia, por ende, hay que dirigir el rumbo del camino donde la felicidad sea plena.*

*Alvaro Andrés Martin Rojas*

## AGRADECIMIENTOS

Los autores expresan sus más sinceros agradecimientos a:

A nuestras familias por su amor, apoyo incondicional y motivación durante todos estos años. Todos nuestros logros son suyos.

A Ecopetrol S.A. por proporcionarnos la información necesaria para llevar a cabo este proyecto y al ingeniero Jorge Doval, por ser el guía para conseguirlo.

Al ingeniero Edison Odilio García, por aportar con su orientación y conocimiento para la culminación de este trabajo de grado.

A la Universidad Industrial de Santander y a la Escuela de Ingeniería de Petróleos por sus enseñanzas, experiencias vividas y momentos inolvidables.

**CONTENIDO**

Pág

## TABLA DE FIGURAS

Pág

## LISTA DE TABLAS

Pág

## LISTA DE CUADROS

Pág

## LISTA DE ANEXOS

Pág.

## RESUMEN

**TITULO:** PREDICCIÓN DEL RUN-LIFE EN BOMBAS DE CAVIDADES PROGRESIVAS EMPLEANDO INTELIGENCIA ARTIFICIAL EN EL CAMPO CASABE*

**AUTORES**: ELISA MARÍA ANGULO VANEGAS, ALVARO ANDRÉS MARTÍN ROJAS**

**PALABRAS CLAVE**: Bombas PCP, arenamiento, Campo Casabe, *Run-life,* componentes de falla, intervención, inteligencia artificial, predicción de variables.

**DESCRIPCIÓN:** El campo Casabe ubicado en el Valle Medio del Magdalena Colombiano, operado por la empresa estatal Ecopetrol S.A, ha implementado en gran medida el sistema de bombeo artificial PCP. Una de sus principales razones, su alto contenido de sólidos en sus pozos productores. Ahora bien, aunque se lleva registro del *Run Life* de las bombas PCP y sus principales componentes de falla, actualmente, no se realiza un análisis para la estimación del posible tiempo de vida media de estas bombas cuando son instaladas. Por lo anterior, cuando se presenta falla en estos sistemas, algunas veces hay demoras en el inicio de la intervención, aspecto negativo, dado a que se presenta una diferida en la producción, ocasionando pérdidas de producción y de dinero. Esta situación podría mejorar, estimando el run-life de la bomba, para así, programar una intervención en el menor tiempo posible de falla y así, no perder producción por disponibilidad de equipos.

En este contexto, se identificó la oportunidad de generar un modelo de predicción usando la inteligencia artificial para la estimación del tiempo de vida media (run life) de las bombas PCP, tomando como base los registros que los operadores realizaron desde el año 2016 al 2018. Posteriormente, se evaluó por medio de algoritmos las características más relevantes en el desempeño de las bombas PCP del campo. Dichas características fueron usadas como datos de entrada en cinco diferentes modelos de predicción (*Decision Tree*, *Random Forest, Support Vector Machine, k-nearest Neighbor* y Redes Neuronales, Naive Bayes Gaussian), dando como resultado, un mejor comportamiento en el modelo de *Random Forest* dado que este algoritmo trabaja de una mejor manera con el número de datos que se tenían disponibles para el desarrollo del trabajo (236 datos de falla).


---

*Proyecto de Grado


**Facultad de Ingeniería Fisico-Quimicas, Escuela de Ingeniería de Petróleos. Director: Edison Odilio Navas, Codirector: Jorge Mario Doval Dorado

## ABSTRACT

**TITLE:** RUN-LIFE PREDICTION IN PROGRESSIVE CAVITY PUMPS USING ARTIFICIAL INTELLIGENCE IN THE CASABE FIELD*

**AUTHORS**: ELISA MARÍA ANGULO VANEGAS, ALVARO ANDRÉS MARTÍN ROJAS**

**KEY WORDS**: PCP Pumps, sandblasting, Casabe Field, Run life, Failure components, Artificial Intelligence, variable prediction.

**DESCRIPTION**: The Casabe field located in the Middle Magdalena Valley in Colombia, operated by Ecopetrol S.A, has largely implemented the PCP artificial pumping system. One of the main reasons is its high solids content in the producing wells. However, although the Run Life record of PCP pumps and their main failure components are kept, currently, an analysis is not performed to estimate the possible half-life of these pumps when they are installed. Therefore, when a failure occurs in these systems, there are sometimes delays in the start of the intervention, a negative aspect, given that there is a delay in production, causing production and money losses. This situation could improve, estimating the run-life of the pump, in order to schedule an intervention in the shortest possible time of failure and thus, not lose production due to equipment availability.

In this context, the opportunity to generate a prediction model using artificial intelligence for estimating the average lifetime (run life) of PCP pumps was identified, based on the records that operators made from 2016 to 2018 Subsequently, the most relevant characteristics in the performance of field PCP pumps were evaluated using algorithms. These characteristics were used as input data in five different prediction models (Decision Tree, Random Forest, Support Vector Machine, k-nearest Neighbor and Neural Networks, Naïve Bayes Gaussian), resulting in better behavior in the Random Forest model since this algorithm works in a better way with the number of data that were available for the development of the work (236 failure data).


---

[1] *Graduation Project

**Physical-Chemistry Engineering Faculty. Petroleum Engineering School. Director: Edison Odilio Navas, Codirector: Jorge Mario Doval Dorado

# INTRODUCCIÓN

El desempeño óptimo del sistema artificial de bombas de cavidades progresivas (PCP), es determinado por múltiples factores operativos de cada pozo y por las características geológicas del yacimiento. Algunos factores son más relevantes que otros en dicho funcionamiento, e identificarlos, dará al operador un mejor criterio para la evaluación del tiempo de vida media de estos equipos.

Dadas las características geológicas y operativas del Campo Casabe, el sistema PCP ha sido una excelente opción para el manejo de sólidos en la producción de los pozos, característica principal del campo. Es por ello, que este método de levantamiento artificial se ha convertido en el más usado en el campo. No obstante, se han reportado reiteradas fallas en el sistema, las cuales han sido registradas y se ha llevado un seguimiento del tiempo de vida media (Run Life) de las bombas PCP. Sin embargo, no se ha realizado un análisis para la predicción del posible tiempo de vida media de estas bombas. Por ende, cuando se presenta falla en estos sistemas, algunas veces hay retrasos en el inicio de la intervención. Esta demora en la intervención es un aspecto negativo, presentando diferidas en la producción, ocasionando a la compañía pérdidas de dinero.

En este contexto se identificó la oportunidad de evaluar una metodología para la predicción del Run life en el campo Casabe haciendo uso de técnicas modernas tales como los algoritmos de inteligencia artificial.

En el capítulo inicial de este trabajo, se realizó una contextualización sobre la operación en el campo Casabe, su historia, características de campo, una descripción de las bombas de cavidades progresivas (PCP) en donde se detallan los componentes básicos del sistema, de igual manera, se da a conocer la distribución de los sistemas de levantamiento artificial en el campo.

El capítulo posterior, consiste en la recopilación de los datos de fallas de las bombas de cavidades progresivas, los cuales fueron obtenidos desde enero del 2016 hasta diciembre del 2018, estos datos provienen del archivo de seguimiento de fallas del

campo Casabe, propiedad de Ecopetrol S.A. Por otro lado, se obtuvo información de los reportes de las pruebas de producción del campo para los años de estudio y también se obtiene información de las operaciones realizadas en los pozos a analizar por medio del software Open Wells.

En el siguiente capítulo se realiza la selección de los parámetros que presentan una mayor influencia en la predicción del Run life en el campo Casabe, esto, basado en el estudio de diferentes campos con características similares al campo Casabe y con la característica de que tienen instalados bombas PCP en sus pozos. Se compara y se detalla el perfil de las fallas de estas bombas en estos campos.

En el quinto capítulo se describen los algoritmos de inteligencia artificial que fueron analizados para la predicción del Run life del campo Casabe. Se estudiaron seis modelos los cuales son: Redes neuronales artificiales (RNA), máquina de soporte vectorial (SVM), Naive bayes Gaussiano, K-vecinos más cercanos (KNN), Árboles de decisión (Decisión tree), Bosques aleatorios (Random Forest). Se detalla el funcionamiento de estos algoritmos a la hora de la predicción de datos.

En el sexto capítulo se establece el diseño de un modelo que permite dar solución al problema presentado, donde inicialmente se describe la construcción del Dataset el cual contiene 236 datos (registros de pozos que fallaron por PCP). Se evalúa por medio del programa PYTHON las características de pozo más relevantes para cada modelo, estás comprenden: Máxima desviación del pozo, profundidad de la bomba, torque, revoluciones por minuto, nivel de fluido, %BSW, caudal y THP. De igual manera, se categorizan cuatro diferentes clases de salida (0,1,2 y 3). El valor predicho por el modelo, representará el cuatrimestre en que falla la bomba. Esta categorización de solo cuatro salidas, hace que el modelo de predicción sea de clasificación y no de regresión. Luego se realiza el procesamiento de los datos en

cada uno de los modelos para observar el comportamiento de dichos modelos con el dataset con el que se cuenta.

Finalmente, se obtuvo el porcentaje de predicción de cada uno de los algoritmos de inteligencia artificial que fueron analizados para la determinación del Run life del campo Casabe, obteniendo como resultado, una mejor predicción en el modelo de Bosques aleatorios (Random Forest).

# 1. OBJETIVOS

## 1.1 OBJETIVO GENERAL

Predecir el run life en las bombas de cavidades progresivas mediante el uso de inteligencia artificial en el Campo Casabe propiedad de Ecopetrol S.A

.

## 1.2 OBJETIVOS ESPECÍFICOS

Realizar una recopilación de los datos de falla en las bombas de cavidades progresivas del campo Casabe durante los últimos tres años.

Analizar los parámetros que afectan el Run life de la bomba PCP en el campo Casabe, con la finalidad de crear un *dataset* que permita la obtención de datos de entrada al modelo de predicción con inteligencia artificial.

Estudiar los diferentes algoritmos de inteligencia artificial, con el fin de elegir el modelo que mejor se ajuste en la predicción de fallas en bombas PCP del campo Casabe.

Diseñar un modelo capaz de predecir el Run life de las bombas PCP, en el campo Casabe, usando datos suministrados por Ecopetrol S.A, de los últimos tres años.

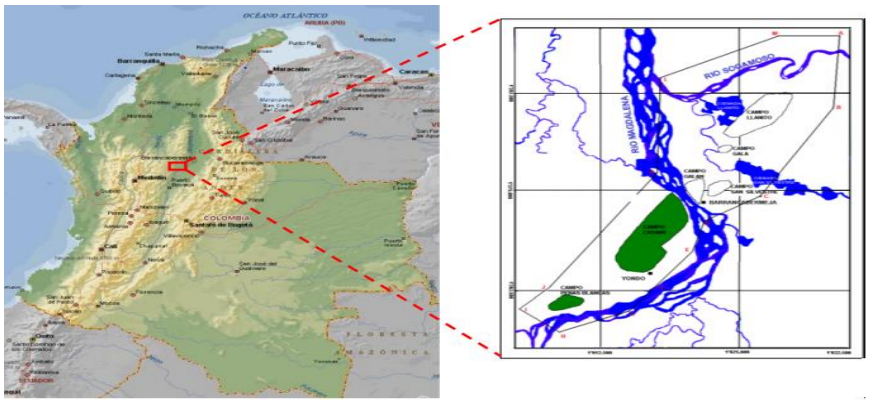
# 2. GENERALIDADES

En este capítulo se describen las generalidades más importantes del campo Casabe y características de las bombas PCP, sus principales componentes y la función que tienen. Por último, se mencionará la distribución de los sistemas de levantamiento utilizados en el campo.

## 2.1 CAMPO CASABE

El campo Casabe, actualmente propiedad de Ecopetrol S.A. se descubre en el año 1941, con el pozo casabe 1 (CSBE-1) a través de la compañía Royal Dutch Shell. Este campo colombiano maduro, está localizado en la cuenca del Valle Medio del Magdalena (VMM), específicamente en el municipio de Yondó, departamento de Antioquia, Colombia[1]. *(Figura 1).*

Figura 1.Localización del campo Casabe



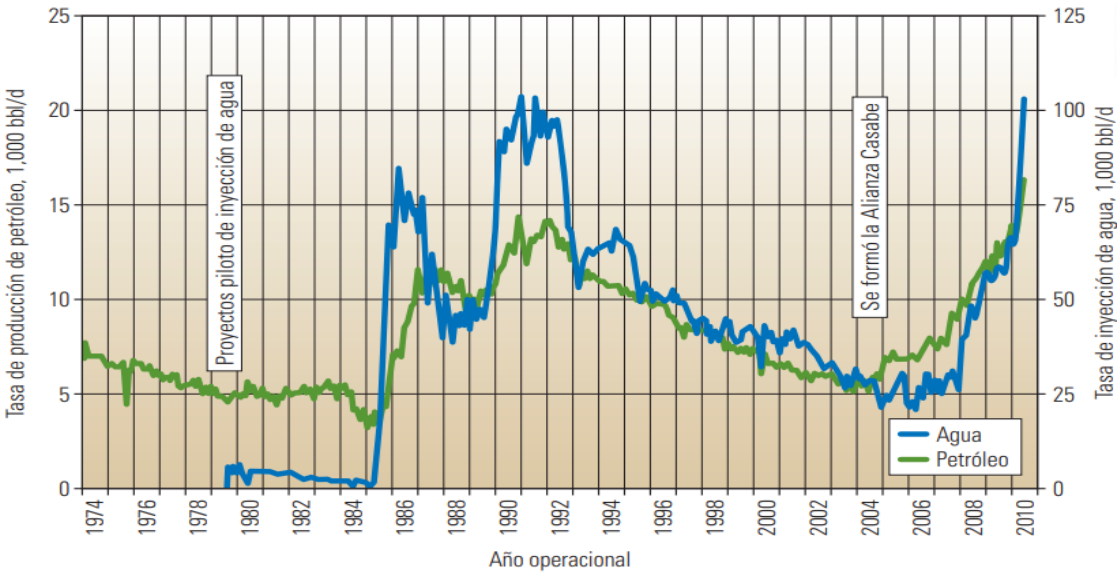

Fuente: ACOSTA, T. et al. Recuperación mejorada en un yacimiento de alta complejidad estratigráfica: Campo Casabe (Caso de estudio). Acipet, 2017.

[1] ACOSTA, T. et al. Recuperación mejorada en un yacimiento de alta complejidad estratigráfica: Campo Casabe (Caso de estudio). Acipet, 2017.

En junio de 1945 se inicia la explotación comercial del campo Casabe y para el año de 1958 se logra su máximo desarrollo con la perforación de 414 pozos de los cuales, tan solo 10 de estos resultaron secos. La tasa máxima de producción que llego a alcanzar llegó a hacer de 46 000 BOPD, y los principales mecanismos de producción durante este periodo fueron el agotamiento natural y empuje de un acuífero débil.

Para el año de 1970, se presenta una declinación significativa del campo, reduciéndose a 5000 BOPD la producción en sus valores promedio[2] (Figura 2). Con la finalidad de dar solución a esta declinación, la empresa colombiana de petróleo, Ecopetrol S.A, determinó en el año de 1979, la explotación secundaria del campo, mediante pilotos de inyección de agua proveniente de la formación La Mesa.

Figura 2.Tasa de inyección de agua y producción de crudo en Casabe.




Fuente: AMAYA. M, et al. Revitalización de un campo maduro. Oilfield review primavera de 2010:22, no. 1. Copyright 2010 Schulumberger.

Teniendo en cuenta los resultados favorables obtenidos, Ecopetrol S.A determinó en el año 1985, extender el piloto de inyección de agua a todo el campo, mediante

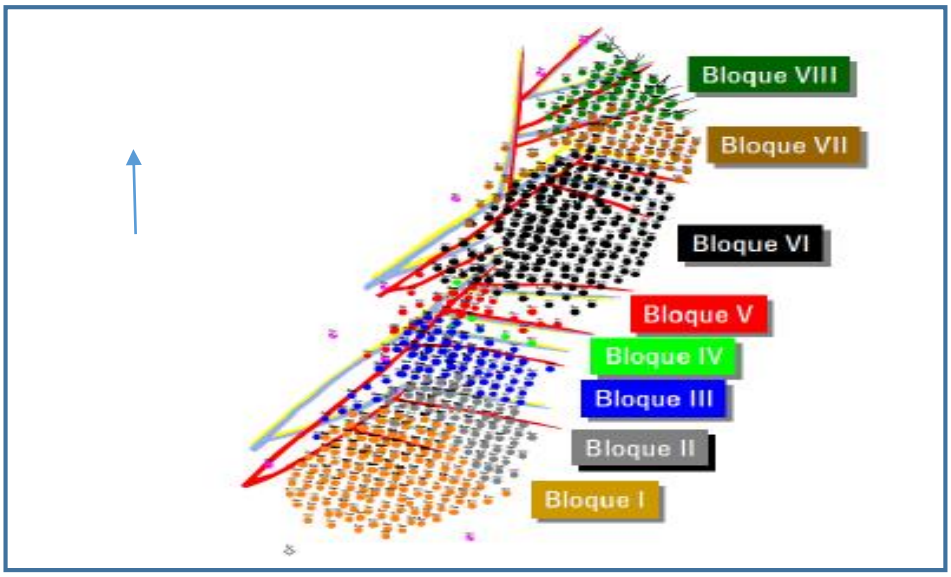
[2] AMAYA. M, et al. Revitalización de un campo maduro. Oilfield review primavera de 2010:22, no. 1. Copyright 2010 Schlumberger.

patrones de cinco puntos. A pesar de que el sistema de inyección implementado logró aumentar significativamente la producción del campo, las complejidades estructurales, la heterogeneidad de las arenas y las lutitas hinchables, ocasionaron que se presentara una irrupción temprana del frente de inyección en los pozos productores, trayendo consigo una alta producción de sedimentos que causaron fallas en los equipos de fondo y baja efectividad en el proceso. Además, se evidenciaron colapsos del casing en algunos pozos, asociados a la alta producción de arena, lo que generó pérdidas económicas muy significativas.

Debido a la constante declinación de la producción y en búsqueda de mitigar los problemas ocasionados por la inyección descontrolada de agua en el campo, se firma la Alianza tecnológica entre Ecopetrol S.A y la multinacional Schlumberger en el año 2004. El objetivo principal de este convenio fue la de incrementar la producción del campo, mediante la implementación de nuevas tecnologías, técnicas de gerenciamiento de yacimientos y reducción de costos operativos. Gracias a estos cambios, se logra una producción de campo de hasta 18 000 BOPD con un caudal de inyección de agua de 110 000 BWPD.

**2.1.1 Estructura Del Campo.** El campo Casabe, presenta una estructura anticlinal asimétrica, con buzamiento moderado hacia el Oriente. Hacia el oeste, limita con la Falla de Casabe y además presenta una distribución de fallas normales en dirección este-oeste, lo que hace que el campo se encuentre dividido en ocho bloques, lo cual ha sido aprovechado operativamente para su desarrollo. (Figura 3). Las formaciones productoras corresponden a Colorado, Mugrosa y La Paz. La profundidad de estas formaciones oscila entre 2 200 y 5 600 ft.

Figura 3. Distribución de bloques a través del campo Casabe.




Fuente: ACOSTA. T, et al (2017).

### 2.1.2 Propiedades Petrofísicas y del fluido del Campo Casabe

Tabla 1. Propiedades petrofísicas del campo casabe.

| Parámetro | Fm. Colorado | Fm. Mugrosa |
|---|---|---|
| Área (acres) | 4570 | 2030 |
| Profundidad prom (ft.s.n.m) | 2600 | 3900 |
| Espesor neto (ft) | 76 | 43 |
| Porosidad prom (%) | 24 | 25 |
| Permeabilidad prom (md) | 225 | 385 |
| Swi (%) | 23 | 23 |

Fuente: Ecopetrol S.A.

Tabla 2. Propiedades de los fluidos del campo Casabe.

| Parámetro | Rango |
|---|---|
| Gravedad API | 19-21 |
| Viscosidad (cp) | 43-21 |
| Factor volumétrico inicial | 1,083-1,117 |
| Factor volumétrico | 1,055-1,07 |
| GOR (SCF/STB) | 180-250 |
| Presión a Pb | 1350-2200 |

Fuente: Ecopetrol S.A

## 2.2 BOMBAS DE CAVIDADES PROGRESIVAS

Las bombas de cavidades progresivas (*Progressive Cavity Pump* - PCP) son bombas de desplazamiento positivo que consisten en un rotor de acero helicoidal dentro de un elastómero sintético, que se encuentra pegado internamente a un tubo de acero (estator). El estator se instala en el pozo conectándolo al fondo de la tubería de producción. Por otro lado, el rotor se encuentra conectado al final de la sarta de varillas. La rotación de esta sarta desde superficie por accionamiento de una fuente de energía permite que el rotor gire dentro del estator fijo, permitiendo

que las cavidades progresen haciendo que el fluido se desplace verticalmente hasta superficie[3]. (Figura 4 y Figura 5)

Tabla 3. Algunas ventajas y desventajas de las Bombas PCP.

| **Ventajas del sistema PCP** | **Desventajas del sistema PCP** |
|---|---|
| Producción de fluidos altamente viscosos. | Capacidad máxima de 5 000 bbl/día |
| Producción con alto corte de arena | Capacidad de levantamiento max 10 500 ft |
| Tolera altos porcentajes de gas libre | Resistencia térmica de max 350 °F |
| Ausencia de válvulas (evita bloqueos y desgaste) | Sensibilidad a los fluidos (elastómeros hinchados o deteriorados con ciertos fluidos por periodos de tiempo prolongados |
| Resistencia a la abrasión | |
| Bajos costos de inversión inicial | Bajas capacidades volumétricas con cantidades de gas libre considerable |
| Bajos costos de energía | Daño del estator cuando la bomba trabaja en seco por largos periodos de tiempo |
| Simple instalación y operación | Desgaste por contacto de varillas y/o tubería de producción (pozos alto dogleg y horizontales) |

[3] CIULLA F**.** Fundamentos para diseño de Cavidades progresivas. Weatherford, 2003.

| | |
|---|---|
| Pequeñas dimensiones en superficie | Frecuentemente se requiere remoción de la tubería para remover la bomba. |
| Bajo mantenimiento | |
| Bajo nivel de ruido | A altas velocidades de trabajo se presentan vibraciones* |

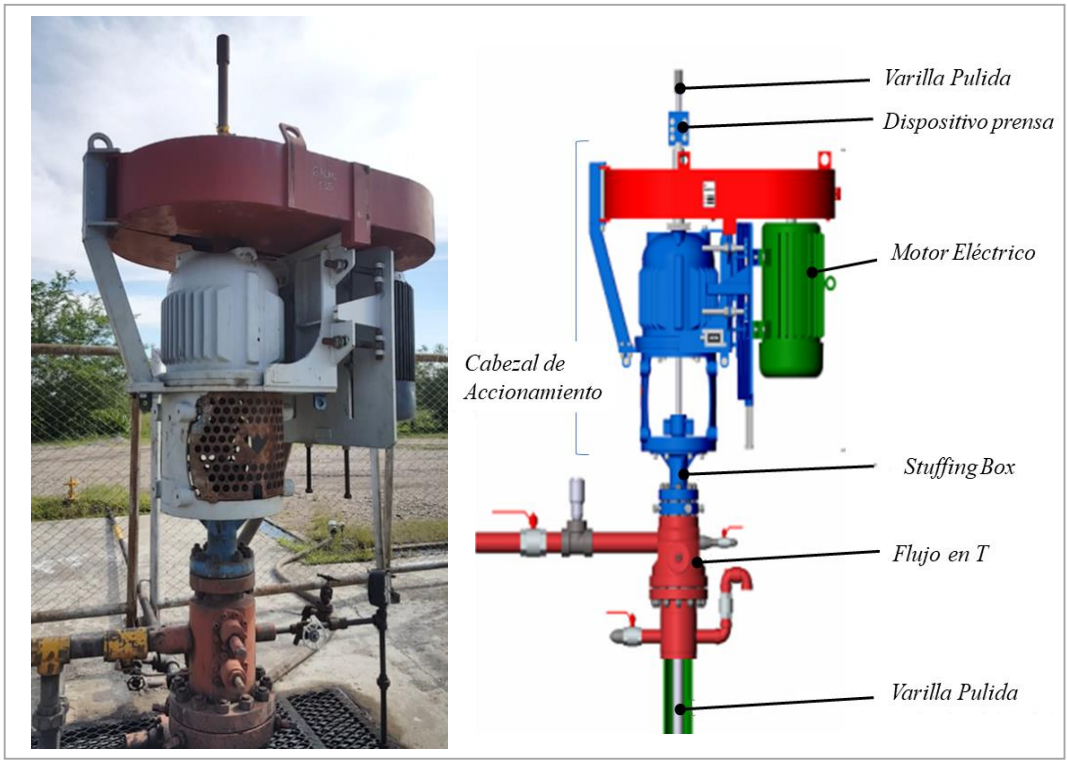
*Al tener altas vibraciones, se recomienda instalar anclas de tuberías y estabilizadores o centralizadores de varillas.

Fuente: CIULLA F. (2003).

La alta eficiencia de alrededor de 50 o 60 % en estas bombas es una gran ventaja respecto a otros tipos de levantamiento artificial. En la tabla 3, se evidencian algunas ventajas y desventajas de este sistema.

Como se había expresado anteriormente, las bombas PCP utilizan un rotor de forma helicoidal con *n* lóbulos, dentro de un estator de forma helicoide con *n+1* lóbulos. Las dimensiones del rotor y estator están diseñadas de tal manera que producen interferencias, las cuales crean líneas de sello que definen las cavidades (debido a que las cavidades están hidráulicamente selladas entre sí, el tipo de bombeo es de desplazamiento positivo). Al girar el rotor, estas cavidades se desplazan (progresan) en un movimiento combinado de traslación y rotación, lo cual se manifiesta en desplazamiento helicoidal de las cavidades (figura 6), desde la succión hasta la descarga.

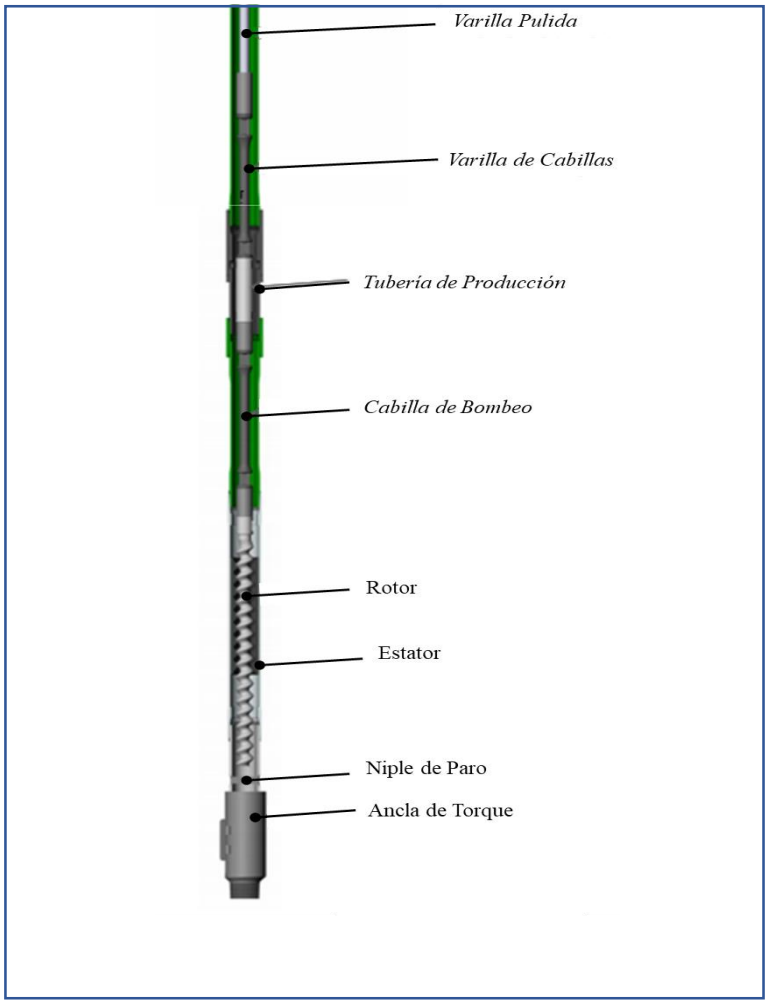
Figura 4.Componentes de superficie del sistema PCP.




**2.2.1 Componentes de subsuelo de las bombas PCP.** El sistema de fondo incluye:

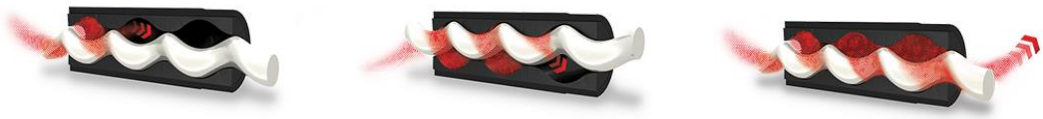
- ✓ Varilla Pulida
- ✓ Sarta de Varillas
- ✓ Rotor
- ✓ Estator
- ✓ Elastómero
- ✓ Centralizadores
- ✓ Niple de Paro
- ✓ Otros accesorios

Figura 5.Componentes de fondo del sistema PCP




Fuente: Weatherford. (2005)

Figura 6.Funcionamiento de las cavidades progresivas

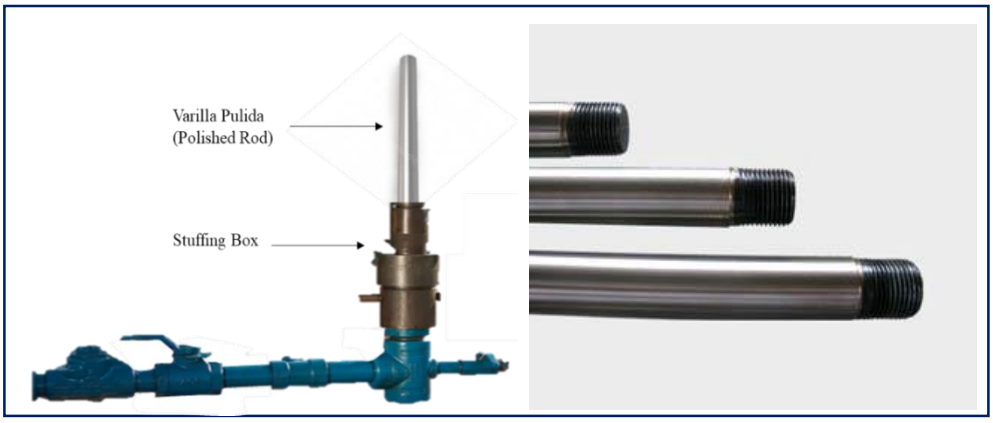


Fuente: Soluciones PCM (2012)

Al estar el rotor introducido en el estátor, se crea una doble cadena de cavidades hexagonales herméticas. Cuando el rotor gira en el interior del estátor, el panel avanza en forma de espiral a lo largo del eje de la bomba, sin cambiar de forma ni de volumen. De esta manera se transfiere el fluido.

**Varilla Pulida (*Polished rod*):** Es la junta más superior que tiene el sistema a de varillas. Es la unión directa entre la sarta de varillas y la superficie. La varilla Pulida pasa a través de la caja de sellos y permite un eficiente sello hidráulico. Generalmente, está fabricada con acero, molibdeno, manganeso y níquel, superficialmente terminada en acabado espejo, con el propósito que no dañe los sellos. Puede ser hueca o totalmente sólida. El diámetro de la varilla dependerá del tipo de cabeza de accionamiento se tenga (los rangos normalmente usados son de 7/8”, 1”, 1 1/8”; 1 ¼”; 1 ½”; 1.9”).

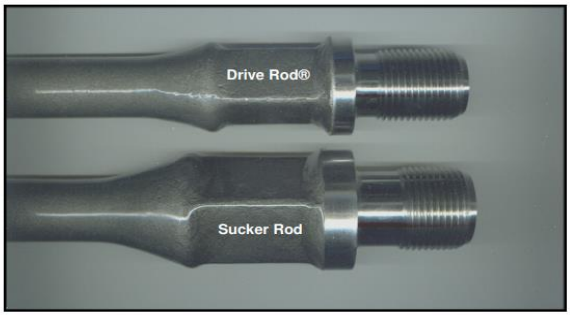
Figura 7.Varilla Pulida




Fuente: NELGAR Services (2018)

**Sarta de varillas:** Es un conjunto de varillas las cuales su función principal es transmitir la rotación desde la superficie (desde la varilla pulida) hasta el rotor. Su diámetro está limitado al diámetro interior del *tubing* de producción (normalmente se usan diámetros reducidos para evitar rozamiento con el tubing). El máximo esfuerzo soportado sobre las varillas es en el tope de la sarta.

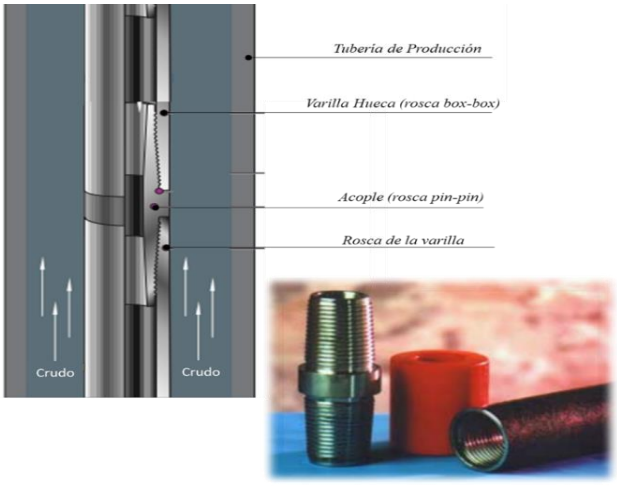
Figura 8.Comparación Varillas convencionales y modificadas (Drive Rods vs Sucker Rods).




Fuente: DOVER Artificial Lift (2014).

Existen una gran variedad de tipos de varillas como lo son: Varillas Convencionales (*Sucker Rods*), varilla hueca (*Hollow rods*), varilla continua o tubería flexible (*continuous rod, coiled rod*).

Figura 9.Varillas huecas.




Fuente: OLMOS & Co. (2003)

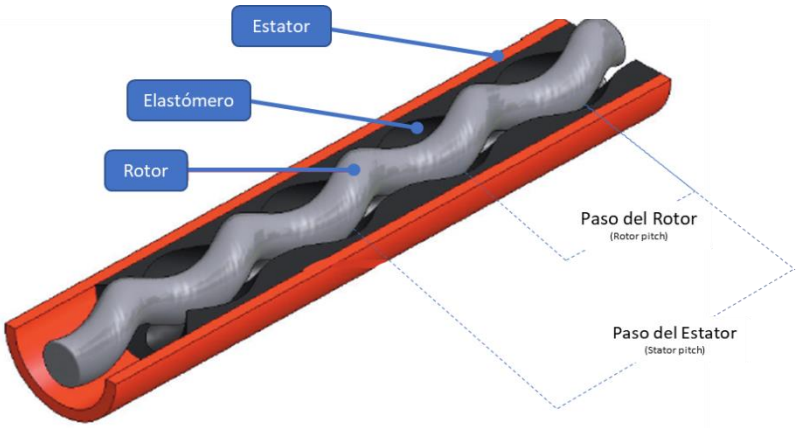
**Rotor:** Es el único elemento que se mueve en la bomba y se encuentra conectado a la sarta de varillas. Los rotores se construyen de acero de alta resistencia y son recubiertos por una capa de material resistente a la abrasión. Se han probado materiales como el carburo de tungsteno, carburo de silicio, óxido de titanio, y oxido de cromo. Sin embargo, ninguno de ellos ha mostrado tan alta resistencia como el cromo endurecido. Como ya se ha dicho anteriormente, el rotor tiene como función principal, bombear el fluido girando de modo excéntrico dentro del estator, creando cavidades que progresan de forma ascendente.

El diámetro del rotor es función del posible hinchamiento que sufrirá el elastómero del estator por efectos de presión, temperatura y reacción química con los fluidos producidos.

Un paso del rotor menor y un diámetro menor mejora el manejo del torque. Las cavidades reducidas, también hacen que se reduzca la velocidad a través de la

bomba, dando así, una menor erosión y una vida (run-life) mayor de la bomba. A su vez, las cavidades de mayor sección permiten un mayor desplazamiento de fluido[4].

Figura 10.Elementos de la bomba.

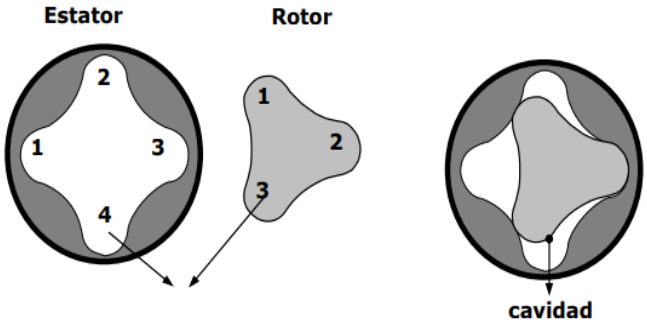



Fuente: Cameron (2013).

**Estator:** Este elemento se encuentra conectado usualmente al tubing de producción. Son tubos de acero con un elastómero sintético en su interior moldeado precisamente en forma de hélice. Como se había explicado anteriormente, los lóbulos del estator son n+1 lóbulos de rotor. Siempre el estator tiene un lóbulo más que el rotor. (figura 11). En la industria, la geometría más usada es la 1:2.

[4] KODU PCP. *Complete progressing cavity pump system*. Calgary, Canada. 2010

Figura 11.Lóbulos del Estator y del Rotor.




Fuente: HIRCHFELDT M. (2008)

Para la fabricación del estator, se recubre con un adhesivo la parte interna del tubo de acero para permitir la unión metal-elastómero. Posteriormente se inyecta controladamente el elastómero a alta temperatura y alta presión entre la camisa de acero y un núcleo. Dicho núcleo tiene un perfil interno inverso al del estator, hace la función de rotor de dos lóbulos. Posteriormente el estator pasa a un proceso de vulcanizado para lograr las propiedades deseadas del elastómero. Finalmente se deja enfriar lo que hace que el elastómero se contraiga, permitiendo así, la extracción del núcleo. La magnitud de la contracción depende del tipo de elastómero.[5]

Es importante que cada cierta cantidad de estatores fabricados se tome una muestra del elastómero, con el fin de verificar que todas sus propiedades mecánicas estén en los rangos permitidos, ya que es precisamente el elastómero el mayor responsable de la calidad del producto final.

[5] HIRCHFELDT M. Manual de bombeo de cavidades progresivas. Volumen I. 2008.

**Elastómeros**: Es un polímero de alto peso molecular el cual se encuentra en forma de espiral la cual es adherida a la parte interna del tubo de acero para así conformar el estator. El elastómero posee una propiedad esencial para el proceso y es su capacidad de recobrar rápidamente sus dimensiones una vez que la fuerza es removida (resiliencia). Gracias a esta propiedad, es posible que se cree la interferencia necesaria entre el rotor y el estator la cual determina la hermeticidad entre cavidades contiguas y en consecuencia la eficiencia de la bomba.

Varias fallas de las bombas PCP se deben a la falta de integridad del elastómero y a menudo resulta ser afectado física o químicamente por las condiciones a las que trabaja. La temperatura es uno de los factores a considerar, puesto que, aunque varía para cada yacimiento, ésta puede tener un rango entre los 60 °F a 360 °F (15 a 200 °C). De igual manera, la bomba puede trabajar con una alta presión de fondo. El fluido producido también es otro factor que puede disminuir la integridad del elastómero, debido a que puede contener solidos (arena), gases ($CH_4$, $CO_2$, $H_2S$) y un variado tipo de otros constituyentes, incluyendo agua, parafinas, naftenos, asfáltenos, y aromáticos. Adicionalmente, existen fluidos en contacto con el elastómero, como los usados en estimulación de pozos, tratamientos, las trazas lodo usado en la perforación que permanecen en la formación, inhibidores de corrosión entre otros[6].

Debido a lo anterior, la industria día a día busca desarrollar nuevos tipos de elastómeros cada vez más resistentes a estas condiciones de operación.

Los elastómeros más usados en la aplicación de las bombas PCP, son:

- ✓ Base Nitrílica o caucho NBR (*nitrile butadine rubber*)
- ✓ Nitrilo Hidrogenado o HNBR (*Hydrogenated nitrile butadine rubber*)
- ✓ Fluoelastómeros

Los cambios más comunes en las propiedades mecánica de los elastómeros son:

[6] SPE, Petrowiki. Petroleum Engineering Handbook (PEH). Octubre de 2018

- ✓ Hinchamiento, lo cual provoca una excesiva interferencia entre rotor y estator.
- ✓ Endurecimiento, lo cual provoca la pérdida de resiliencia del elastómero.

**Centralizadores:** Su uso puede ser adicional, no obstante, se recomienda su uso por encima y por debajo de la bomba, cuando el sistema esté trabajando con altas velocidades (mayor a 350 RPM), esto para minimizar el efecto de la vibración excesiva debido al movimiento excéntrico del rotor y a su vez para centralizar la bomba dentro de la *tubing* de producción.

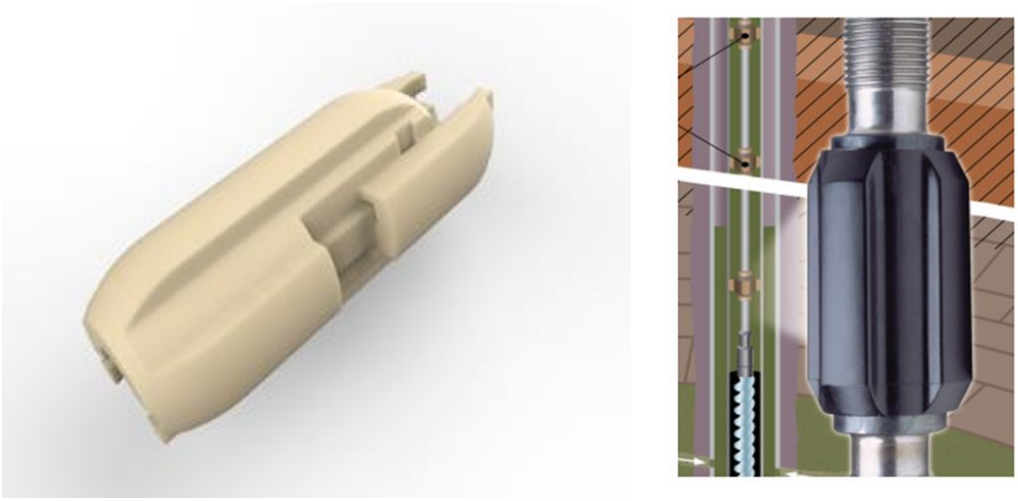
Se recomienda usar centralizadores de tubería cuando se tiene una curvatura mayor a 1º/100 pies en la sección donde será instalada la bomba, para permitir la correcta linealidad entre el rotor y el estator y así, evitar una distribución de esfuerzos no homogéneos en la bomba.

Existen diferentes configuraciones, en los que se tiene:

**Dispositivos Rotativos:** Elemento de una sola pieza montado sobre el cuerpo de la varilla con recubrimiento externo de material suave (poliuretano), rota con la varilla.

**Dispositivos no rotativos o de Giro:** No rota con la varilla, se mantiene en contacto con ella. Elemento de dos piezas cuyo componente interno esta adjuntado o montado sobre la varilla y la camisa externa independiente. Normalmente el poliuretano está en contacto con la camisa.

Figura 12.Centralizadores de varilla no rotatorios.



Fuente: KODU (2016)

Niple de paro: un elemento de paro es un tubo de longitud corta (~ 2 ft) el cual va roscado a la parte inferior del estator. Normalmente tiene rosca-caja en la parte superior (para conectarlo a la bomba) y en la parte inferior un pin para permitir conectar el ancla de torsión u otro dispositivo. Sus funciones principales son:

- ✓ Servir de punto tope al rotor cuando se realiza el espaciamiento del mismo.
- ✓ Brindar un espacio libre al rotor, de manera que permita la libre elongación de la sarta de varillas durante la operación.
- ✓ Servir de un retenedor para impedir que el rotor y/o cabillas lleguen a fondo en caso tal que se presenta desconexión de éstas últimas.
- ✓ Servir de punto de conexión para accesorios tales como anclas de gas o anti-torque.

Figura 13.Niple de Paro.

Fuente: Chacín Nelvy (2018)

**Trozo de maniobra:** Este elemento es indispensable en la configuración del sistema PCP. Se encuentra conectado justo encima del rotor, en lugar de una cabilla, debido a que, si se instala una varilla, debido a su longitud y a su movimiento excéntrico del rotor que se trasmite directamente a ella, tiende a doblarse y a rozar con las paredes de la tubería de producción. El trozo de maniobra al ser menos de la mitad de largo de la varilla se dobla menos o no se dobla (dependiendo del diámetro) impidiendo así, que roce.

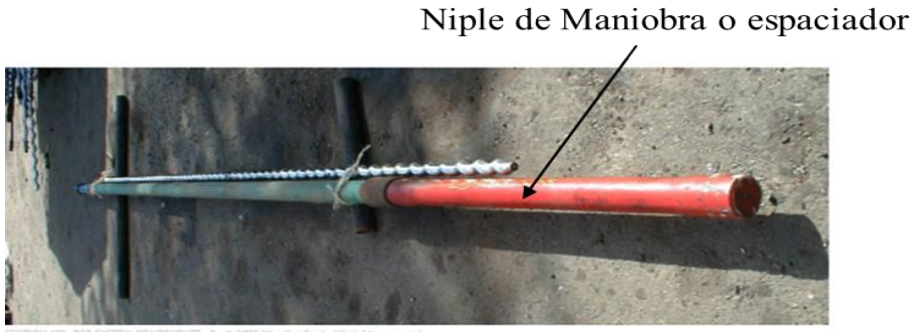
**Niple de maniobra (Niple distanciador):** Se encuentra conectado justo encima del estator y su diámetro interno puede que sea mayor que el de la tubería de producción. Tanto la cabeza del rotor, como el trozo de maniobra se deslazan en dos o más direcciones al mismo tiempo que rotan, este desplazamiento producto de la excentricidad, puede originar un roce entre estos componentes y la tubería de producción, es por eso que se hace necesario el niple distanciador que permita esta libertad de movimiento.

El diámetro que permite este movimiento es D+2E, donde: “D” es el mayor de los diámetros entre la cabeza del rotor y el diámetro externo del acople de varillas. “E” es la excentricidad de la bomba (dato suministrado por el fabricante).

El niple debe contar con diámetro interno mayor que el valor de D+2E. La longitud debe ser la suficiente para que la cabeza del rotor este en el interior de dicho niple.

Otra gran ventaja que presenta los niples de maniobra o niple intermedio (o espaciador) es que durante las operaciones de *workover*, las cuñas, mordazas, llaves de apriete, entre otras, se colocarán en él, y no en el cuerpo del estator, evitando cualquier daño a este último. [7]

Figura 14.Niple de Maniobra.




Fuente: Weatherford. (2003).

**Ancla de torsión:** Cuando el rotor gira en sentido de las manecillas del reloj (visto desde arriba) su fricción con el estator tiende a rotar este último en el mismo sentido y por consiguiente el *tubing* de producción tiene el riesgo de que se desconecte. El

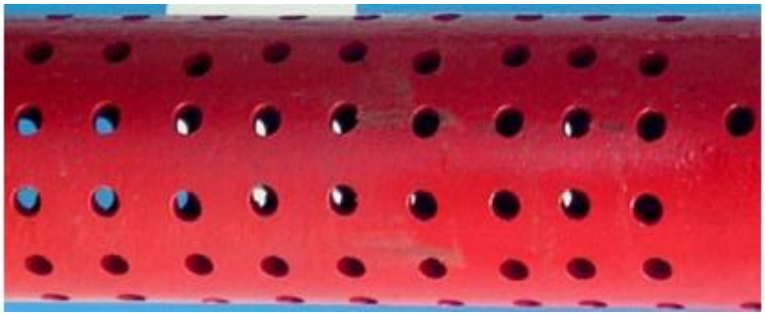
[7] Weatherford. Bombas de Cavidades Crogresivas, componentes del sistema. Programa de entrenamiento. 2003

ancla de torsión evita este problema debido a que cuando más tiende a desenroscarse, el ancla se ajusta más. Este elemento se instala debajo del estator.[8]

**Filtro para Arena/Solidos:** Este dispositivo disminuye o evita que partículas sólidas (rocas, restos de elastómeros, etc.) lleguen a la bomba. Es importante tener en cuenta que, para crudos viscosos, se debe prestar atención en el diseño de los orificios, de manera que no representen una obstrucción al flujo.

Se debe conocer la granulometría de la arena para así, tener un diseño optimo del filtro.

Figura 15.Filtro para Arena/Solidos



Fuente: Weatherford. (2003)

**2.2.2 Componentes de Superficie del Sistema PCP** El equipo de superficie consiste en un pequeño cabezal de rotación y un motor eléctrico de bajo poder. (figura 6). El cabezal alberga la caja de cambios, un sistema de frenos integrado y un eje impulsor (varilla pulida). Las varillas están sujetas al impulsor del cabezal de rotación (o varilla pulida) el cual está ensamblado directamente sobre el cabezal del pozo. El movimiento del eje es provocado por un sistema polea-banda. (Figura 16).

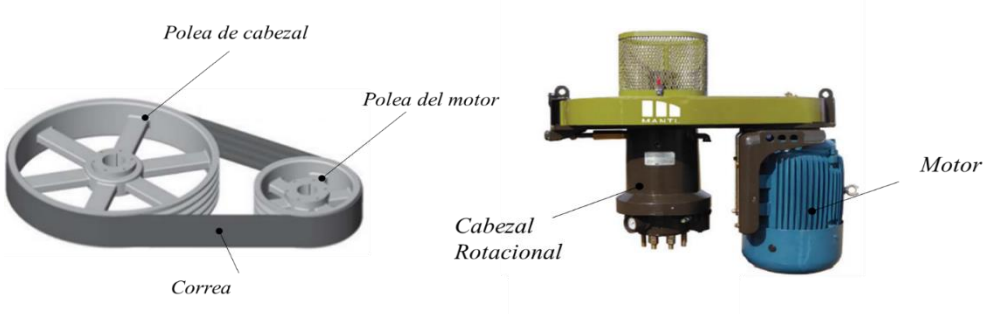
Las funciones principales del equipo de superficie son:

- ✓ Suspender la sarta de varillas y soportar la carga axial del equipo de fondo

[8] HASSAN A, AHMED H, SULEIMAN H, ALI H. *Prediction Production performance of progressive cavity pump for high viscous oil. Sudan University of science and technology. Page 19. 2017*

- ✓ Entregar el torque requerido a la varilla lisa (*polished rod*)
- ✓ Entregar la rotación segura y a la velocidad requerida
- ✓ Prevenir la fuga de fluidos en la superficie
- ✓ Proveer la relajación del sistema de la energía almacenada, durante las paradas *(shutdowns)*

Figura 16.Funcionamiento cabezal.




Fuente: MANTL COMPANY. Production line. Driveheads.

**Motor primario:** En la mayoría de las veces se usa un motor eléctrico. Sin embargo, en lugares apartados donde resulta costoso llevar energía eléctrica, se opta por motores de combustión interna. En general el sistema de PCP, trabaja con baja velocidad, la selección de la unidad motriz incluye especificar un método para reducir la velocidad del motor o un motor de baja velocidad. Las transmisiones con bandas o cadenas son aceptables para reducir la velocidad del motor. También se usa un reductor de engranes de acoplamiento directo para unidades grandes. El reductor disminuirá la velocidad del motor a la velocidad requerida.

Es común el empleo de motores que tienen reductores de velocidad integrales y poleas de paso variable. Aunque estas unidades tienen un mayor costo inicial que

los sistemas de velocidad fija, a menudo, ahorran energía y pueden ser la elección más económica con el pasar del tiempo.  Existen motores de frecuencia variable, ahorran energía si son del tipo correcto. El método más confiable es coincidir la velocidad necesaria en la bomba con la del motor, por ejemplo, 1750, 1775 u 870 rpm para circuitos de 60 Hz, no es necesario de bandas, cadenas ni engranajes, pero puede que se necesite una bomba un poco más grande. Aunque hay que tener en cuenta que rara vez es posible coincidir los requisitos del proceso con la velocidad del motor[9] .

Figura 17.Motor MARATHON 20HP 1200RPM.

Fuente: MARATHON (2018)

Los rangos de operación de los motores eléctricos van desde 7.5 kW hasta 75 kW (10 a 100 hp). Los motores de combustión interna varían su capacidad desde 22.5 kW hasta 225 kW (30 a 300 hp). Como se había comentado anteriormente, los motores utilizados por lo general son eléctricos de baja potencia y de eficiencia *premium* como pueden ser motores de 20 hp para aplicaciones estándar o de 10 hp para bombas pequeñas que manejan bajos gastos. Las grandes bombas (que manejan caudales de 1200 bpd) pueden utilizar motores de 20 hp de potencia.

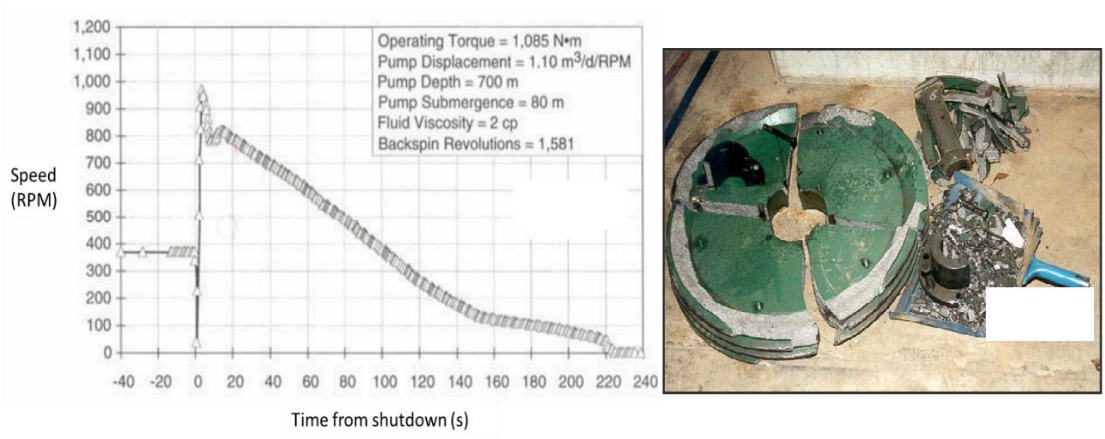
Generalmente en la práctica el rango de operación de las PCP es entre 40 a 350 rpm. Al girar, los motores eléctricos a una velocidad nominal pueden llegar a 1800

[9] ESQUIVEL O. Sistema de bombeo por cavidades progresivas aplicado a pozos desviados. Universidad Nacional autónoma de México, Facultad de ingeniería. México, DF; septiembre 2009.

rpm, se hace necesario contar con una caja reductora de una relación de transmisión adecuada para llevar la velocidad angular del motor a velocidades cercanas a la requerida por la bomba.

**Sistema de frenado:** Una función importante de cabezal es el frenado que requiere el sistema cuando éste rota en marcha inversa (back-spin). Cuando el sistema PCP está en operación, una cantidad energía se acumula en forma de torsión en las varillas. Si en algún momento el sistema se para repentinamente, la sarta de varillas libera la energía acumulada girando en forma inversa para liberar la torsión. Adicionalmente, a esta rotación se le suma la generada por la igualación de niveles en el tubing de producción y espacio anular, al momento de la parada "Ibit",p.[10] Durante ese proceso de Back-Spin, el sistema puede llegar a alcanzar velocidades de rotación muy altas (Figura 18).

Figura 18.Falla del sistema de Frenado.




Fuente: HIRCHFELDT M. (2008)

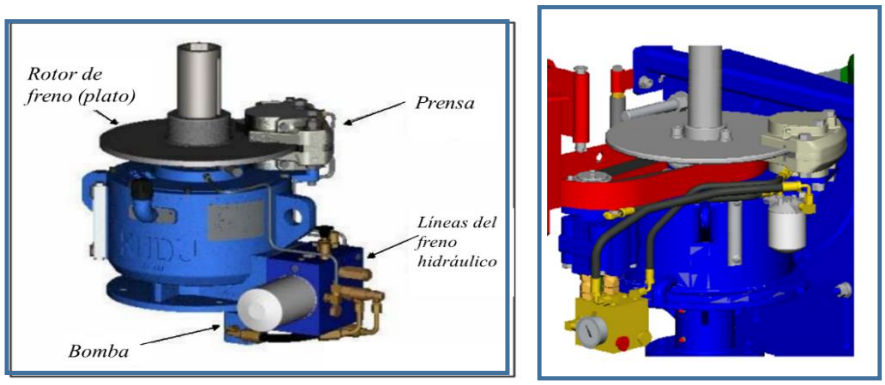
[10] HIRCHFELDT M..,p. 31.

Al perder el control del back-spin, las velocidades son tan altas que causan daños irremediables en los equipos de superficie e inclusive desconexión de la sarta de varillas. Es por ello, que es de gran importancia el uso de frenos, para impedir que las velocidades del back-spin sean tan altas.

Del sistema de frenado, se destacan dos por su eficiencia de funcionamiento:

- ✓ *Freno de accionamiento por fricción*: compuesto tradicionalmente de un sistema de disco y pastillas de fricción, accionadas hidráulicamente o mecánicamente cuando se ejecuta el giro inverso. Este tipo de freno es utilizado generalmente para motores menores a 75 hp.
- ✓ *Freno de accionamiento Hidráulico*: Es muy utilizado debido a su mayor eficiencia de acción. Es un sistema integrado al cuerpo del cabezal que consiste en un plato rotatorio adaptado al eje del cabezal que gira libremente en el sentido de las agujas del reloj (operación de la PCP). Al ocurrir el Back-Spin, el plato acciona un mecanismo hidráulico que genera resistencia al movimiento inverso, lo que permite que se reduzca considerablemente la velocidad inversa y se disipe la energía acumulada. Dependiendo del diseño del cabezal, este mecanismo hidráulico puede accionarse con juegos de válvulas de drenaje, embragues mecánicos, entre otros.

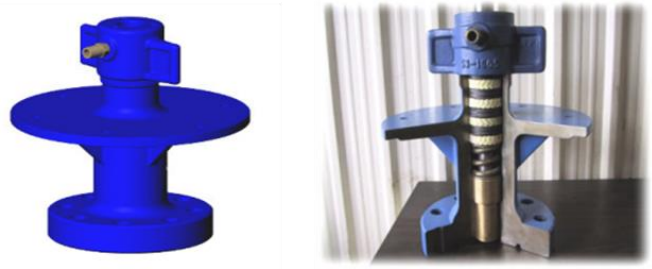
Figura 19.Freno de accionamiento hidráulico para evitar las velocidades inversas.




Fuente: HIRCHFELDT M. (2008)

**Sello mecánico (*Stuffing box*):** Una de las funciones principales de los elementos de superficie es prevenir la fuga de fluidos en la superficie. Los elementos que ayudan a esta función son los prensaestopas o sello mecánico (*stuffing box*). La configuración básica consiste en un niple corto con un sistema de empaquetaduras sintéticas y/o de bronce instaladas en serie. Este dispositivo sella el espacio anular en torno a la barra pulida de manera que pueda girar, pero evitando a la vez la fuga del fluido al exterior (ambiente).

Figura 20.Diseño Convencional del Stuffing Box.



Tomado de: Weatherford. (2003)

## 2.3 SISTEMAS DE LEVANTAMIENTO ARTIFICIAL EN EL CAMPO CASABE

El campo Casabe cuenta con diferentes tipos de sistemas de levantamiento artificial (SLA), los cuales han permitido mantener el campo en producción, dada la declinación que este presenta. El primer tipo de sistema de levantamiento que se utilizó en Casabe, fue el bombeo mecánico, pero debido a que el campo presentó problemas de arenamiento, muchos de estos sistemas se cambiaron a bombas de cavidades progresivas, ya que permiten realizar un mejor manejo de arena.

La implementación del sistema de bombas de cavidades progresivas fue notoria en los últimos años. Para agosto de 2007 el campo Casabe contaba con 34 pozos

productores que contaban con este tipo de levantamiento artificial, y para junio del 2018 se tenían 163 pozos con este sistema, lo que evidencia un incremento aproximado del 500% de este sistema en solo 11 años.

Para el año 2018, el campo Casabe cuenta con 270 pozos productores activos con diferentes tipos de sistemas de levantamiento artificial (tabla 4) entre ellos se encuentran los sistemas de bombeo mecánico (BM), bombas electro-sumergible (ESP), bombas de cavidades progresivas (PCP) y bombeo electro-sumergible combinado con bombeo de cavidades progresivas (ESPCP).

Tabla 4. Distribución de los sistemas de levantamiento artificial empleados en los pozos campo Casabe

| **SLA** | **Número de Pozos** | **Porcentaje (%)** |
|---|---|---|
| BM | 85 | 31,48 |
| PCP | 163 | 60,37 |
| ESP | 19 | 7,04 |
| ESPCP | 3 | 1,11 |

Fuente: Ecopetrol S.A (2018)

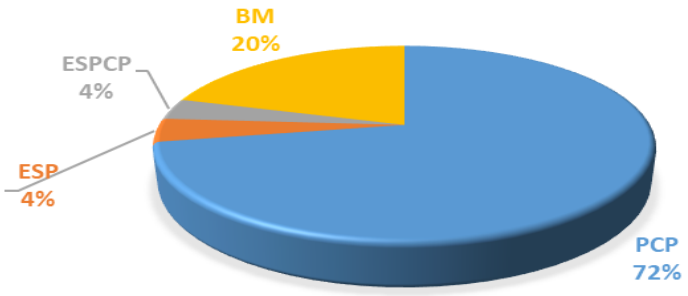
En este contexto, la selección, operación y optimización de los diferentes sistemas juegan un rol muy importante en el desarrollo del campo, donde los esfuerzos se han focalizado principalmente en incrementar los límites técnicos y el *run life* de cada sistema de levantamiento. La instalación de las bombas PCP en el campo, se implementó debido al incremento de las fallas en los pozos con sistema por bombeo mecánico causado por la alta cantidad de bombas pegadas y arenadas. Más tarde, pozos con alto potencial (500 BFPD) no podían ser producidos de forma confiable

con bombeo mecánico, por lo cual los sistemas PCP fueron instalados como una buena opción para el manejo de arena a esas tasas de flujo[11].

## 3. RECOPILACIÓN DE DATOS DE FALLA EN BOMBAS PCP DEL CAMPO CASABE

Como punto de partida para el desarrollo de este proyecto se realizó una recopilación de los datos de fallas de las bombas PCP en el campo Casabe. Se elige usar este sistema de levantamiento artificial, debido a que en el campo este sistema es el que más predomina por su versatilidad en el manejo de arenas. De igual manera, otro criterio de elección del método de levantamiento fue el número de fallas, en las cuales, las fallas de las bombas PCP representan un 72% de los datos registrados (Figura 21).

Figura 21.Fallas de los Sistemas de Levantamiento Artificial para el año 2017 en el campo Casabe.



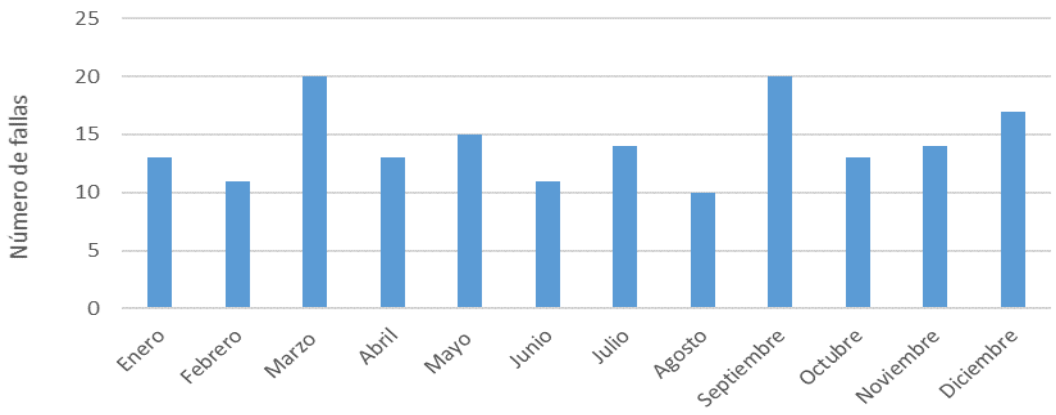

Fuente: Ecopetrol S.A

[11] Ecopetrol S.A (2018)

Cabe resaltar que Ecopetrol S.A ha llevado un seguimiento de las fallas del campo y demás eventos que ocurren en cada uno de los pozos productores del campo Casabe desde el 2016, dicho seguimiento se ha consolidado teniendo en cuenta fechas de instalación, paradas de pozos, análisis de falla, causas de falla, tipos de bombas, componente fallado, entre otros. Para dar solución al presente proyecto se hizo uso de esta información, de la cual la empresa tenía datos desde enero del 2016 hasta diciembre del 2018.

Este archivo se desarrolló en el campo para tener un registro de la cantidad de pozos que presentaban fallas y caracterizar con qué frecuencia ocurría el daño. La (Figura 22) muestra la cantidad de pozos que presentaron algún tipo de falla para el año 2017.

Figura 22.Histograma de las fallas generadas en el año 2017 de las bombas PCP en el Campo Casabe.




Fuente: Ecopetrol S.A

Con la información suministrada por Ecopetrol S.A se construye una parte del *dataset,* la cual fue usada más adelante para la predicción de las fallas de las bombas PCP del campo. Cabe resaltar, que solo se contaba con el reporte de tres años de seguimiento de las fallas y que no se contaba con información característica

del pozo como producción, BSW%, profundidad de la bomba, *dog-leg*, entre otras características, razón por la cual, en primera medida se tuvo que obtener información del software *Open Wells* de la empresa Landmark de HALLIBURTON*,* en el cual, ECOPETROL S.A, reporta todas las operaciones que se realizan en el campo Casabe. De este software se buscó la información referente a la desviación del pozo y profundidad a la cual se ubicó la bomba PCP.

Por otro lado, para obtener la información de las condiciones de operación de los pozos antes de que se genere la falla, se investiga el reporte de las últimas pruebas de producción de pozo que facilitó ECOPETROL S.A. De este archivo se pudo determinar la producción de los pozos, el %BSW, torque de la bomba, revoluciones por minuto (RPM), entre otras características.

Esta información se recopiló junto con los datos de las fallas de las bombas PCP para el mismo periodo (2016-2018). La tabla 5 muestra un archivo de Excel en el cual se encuentra la recopilación de las fallas de las bombas por pozo. Esta Información se abarcará con mayor detalle en los siguientes capítulos, además se evidenciará el uso de este *dataset* para crear el modelo de predicción de fallas PCP para el campo Casabe.

Tabla 5. Base de datos de las fallas de las bombas PCP en el campo Casabe.

| Pozo | Fecha de instalación | Run life (días) | Año de la falla | Mes de la falla | Primer componente fallado | Causa general de la falla |
|---|---|---|---|---|---|---|
| 1 | 19-jun-16 | 228 | 2016 | 6 | Tubería | Instalación |
| 2 | 03-0ct-16 | 113 | 2016 | 10 | Estator | Operación |
| 3 | 20-sep-16 | 639 | 2016 | 9 | Varillas | Desgaste normal |
| 4 | 15-mar-16 | 365 | 2016 | 3 | Tubería | Fluidos o yacimiento |

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | 22-mar-16 | 361 | 2016 | 3 | Tubería | Fluidos o yacimiento |
| 6 | 19-oct-16 | 127 | 2016 | 10 | Estator | Desconocido |
| 7 | 05-feb-17 | 27 | 2017 | 2 | Rotor | Operación |

Fuente: Modificado del archivo de Excel Estadística de fallas ECP, suministrado por Ecopetrol S.A.

# 4. SELECCIÓN DE PARÁMETROS OPERACIONALES Y CONDICIONES DEL POZO QUE MÁS INFLUYEN EN EL *RUN LIFE* DE LAS BOMBAS PCP EN EL CAMPO CASABE

El proceso que se presenta en el presente capítulo fue producto del ejercicio conjunto entre los autores del proyecto y los ingenieros del departamento de producción del campo Casabe y teniendo en cuenta el análisis presente en otros campos. Teniendo en cuenta el seguimiento de fallas que se presentaron en los últimos años, los datos de campo, la experiencia de los ingenieros, se procede a evaluar los parámetros que pueden estar influenciando el Run Life de las bombas PCP en el campo.

## 4.1 CAUSA DE FALLA EN DIFERENTES CAMPOS QUE CUENTAN CON BOMBAS PCP

El primer paso para seleccionar los parámetros que más influenciaran en el Run Life de las bombas en el campo de estudio, fue analizar el comportamiento de otros campos (casos de referencia) que tenían bombas de cavidades progresivas como SLA. La selección de los pozos que se mostrarán a continuación fue elegida por la similitud de las formaciones que tienen con el campo Casabe.

**4.1.1 Campo Bhagyam, India** El campo de Bhagyam está localizado en noreste de india, este campo posee 150 pozos productores y 40 pozos inyectores, el campo contiene un crudo dulce con una gravedad API de 27°, una viscosidad que está en el rango de [50-450] Cp. La máxima temperatura que alcanza el yacimiento es de 53°C[12].

Este campo tiene en su mayoría bombas PCP como sistema de levantamiento, las cuales han presentado diversas fallas en sus componentes[13]. Tales como:

**Falla en la tubería:** Esta falla esta ocasionada debido a la desviación que presentan los pozos del campo Bhagyam (Figura 23).

**Falla en el estator**: Las fallas en el estator eran causadas debido al movimiento forzado que hacían las partículas sólidas que están siendo arrastradas por el crudo a lo largo de las cavidades del estator, generando una fuerte abrasión.

**Falla en la cavillas:** Este componente estaba presentando fallas debido a las torsiones que sufría el equipo y la fatiga a la cual se estaba forzando.

[12] Agarwal. S. et.al. (2016, November 30). *Advance in Completion Design to Improve Bhagyam PCP Run Life. Society of Petroleum Engineers*.

[13] Agarwal. S.Op.cit., p.49.

Figura 23.Falla ocasionada por el contacto entre el Tubing y el Rod-tubing.

Fuente Agarwal. S. et.al. (2016)[16]

**4.1.1.1 Análisis de Fallas del Campo Bhagyam.** En el análisis de fallas presentes en el campo se obtuvo que la mayoría estas se estaban generando principalmente por estas causas:

- Alto RPM.
- Desviación que presentan los pozos (Alto dogleg y perfil en forma de S).
- Contraste de las durezas ente en rod sucker y el material del tubing.

**4.1.2 Campo Bonanza, Colombia**. El campo Bonanza, está localizado en la cuenca del Valle Medio del Magdalena, Departamento de Santander, municipio de Rio negro. Forma parte del activo provincia, es operado por Ecopetrol S.A. La (Figura 24) muestra la localización del campo Bonanza "Ibid".p. [14]

[14] Ibid.., p. 44

Figura 24.Localización Geográfica del Campo Bonanza.


Fuente. Informe plan de desarrollo del campo bonanza 2010.

El campo está constituido por 32 pozos activos, 26 de estos son productores. Presenta la siguiente distribución en cuanto a levantamiento artificial (Tabla 6).

Tabla 6. Distribución de los sistemas de levantamiento artificial en el campo Bonanza año 2013.

| SLA | Número de pozos |
|---|---|
| PCP | 21 |
| Gas lift | 2 |
| BM | 2 |
| ESP | 1 |

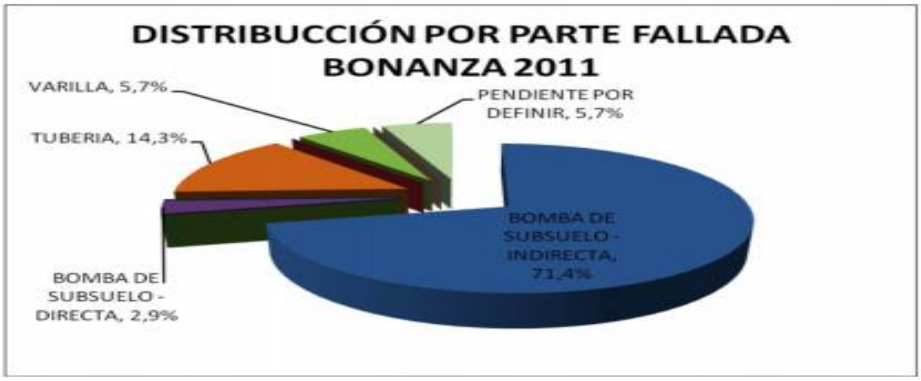
Fuente Ecopetrol S.A.

Como se puede apreciar la mayor distribución del tipo de sistema de levantamiento artificial corresponde a bombas PCP. De acuerdo a los reportes la mayoría de las

fallas son asociadas específicamente a tres componentes de subsuelo en el pozo (Figura 25):

- Bomba.
- Varilla de bombeo.
- Tubería de producción.

Figura 25.Distribución de fallas del campo Bonanza, 2011.




Fuente. Ecopetrol. S.A

En este campo se han presentado fallas de las bombas PCP, donde la mayor frecuencia de ocurrencia se encuentra distribuida en Rotores, Estatores, Tubería.

## 4.2 CAUSA DE FALLA EN LAS BOMBAS PCP DEL CAMPO CASABE.

Teniendo una idea del comportamiento de otros campos, se buscó identificar cuáles son los componentes que más fallan en las bombas PCP del campo Casabe y a su vez tener una noción de cuál puede ser la causa de estas fallas. Esta información

será indispensable para crear el *dataset* que se usará para analizar los algoritmos de inteligencia artificial y a su vez determinar el *Run Life* de las bombas PCP (Tabla 7).

Tabla 7. Porcentaje de Causas Específicas de falla en los pozos del Campo Casabe**.**

| **Causa especifica de falla** | **Numero de Pozos** | **Valor porcentual** |
|---|---|---|
| Arenamiento | 41 | 28% |
| Configuración del Sistema | 1 | 1% |
| Datos Inapropiados Usados en el Diseño | 1 | 1% |
| Desconocido | 2 | 1% |
| Desgaste Normal o Esperado | 19 | 13% |
| Fluidos Corrosivos | 4 | 3% |
| Instalación – Servicio de Campo | 9 | 6% |
| Limitación de la Tecnología | 20 | 14% |
| Operación de Otros Pozos del Campo | 17 | 11% |
| Problemas de Fabricación | 1 | 1% |
| Procedimiento de Operación | 1 | 1% |
| Sabotaje / Vandalismo | 1 | 1% |
| Scale | 1 | 1% |
| Selección de Materiales | 12 | 8% |
| Monitoreo Inadecuado | 18 | 12% |
| **Total** | 148 | 100% |

Fuente Ecopetrol S.A. (2017)

Se procede a analizar la base de datos que se tiene para el seguimiento de fallas la cual se habló en el capítulo 2 de este libro. De la información contenida en esta base, se estableció cuáles eran los mayores problemas que generaban daño en las bombas PCP. De lo anterior se pudo determinar las causas principales de generación de fallas en el campo Casabe. Se destaca que un 28% de estas fallas es debido al alto arenamiento que producen los pozos. Es importante tener en

cuenta, que la implementación de gran cantidad de bombas PCP en el campo, se debió a que estas permiten un gran manejo de arena, ya que los pozos de Casabe presentan un alto BSW% “Ibid”.p.[15].

Como se puede evidenciar, para el año 2017, 41 pozos presentaron fallas por arenamiento, lo cual corresponde al 28% de las fallas, el porcentaje más alto de las causas. De igual manera las limitaciones tecnológicas presentan un alto porcentaje con 14%, el desgaste de los equipos con un 13% y el monitoreo inadecuado con un 12%. Así bien, para tener un mayor entendimiento de estas causas se describirán a continuación.

**Arenamiento**: Es una de las principales causas de falla con los que cuenta el campo Casabe. Considerado como uno de los mayores problemas que se presenta en el proceso de producción de un campo, causando daños en los equipos de la bomba. Las partículas de arena generan erosión al equipo, ocasionando un desgaste por abrasión, provocando problemas de roturas en los equipos o fracturamiento. Un ejemplo de ello es la evidencia de incrustaciones de arena en algunos elastómeros. Por lo anterior, trae consigo perdidas en la producción y disminución del *run life* en los equipos.

**Desgaste normal o esperado:** Hace referencia a las fallas ocasionadas después de que el sistema cumplió con su ciclo de vida medio, en este caso, el equipo tuvo un buen desempeño durante su vida útil.

**Fluidos corrosivos:** Hace referencia a fallas que ocurrieron en ciertos componentes del sistema al quedar expuestos a fluidos corrosivos del yacimiento, causando así, que el equipo sufra alteraciones y no pueda cumplir con su tiempo de vida útil.

**Limitaciones de la tecnología:** Algunos equipos a instalar no están disponibles o las empresas proveedoras demoran un tiempo considerable en su entrega. Es por

[15] Ibid.., p.44.

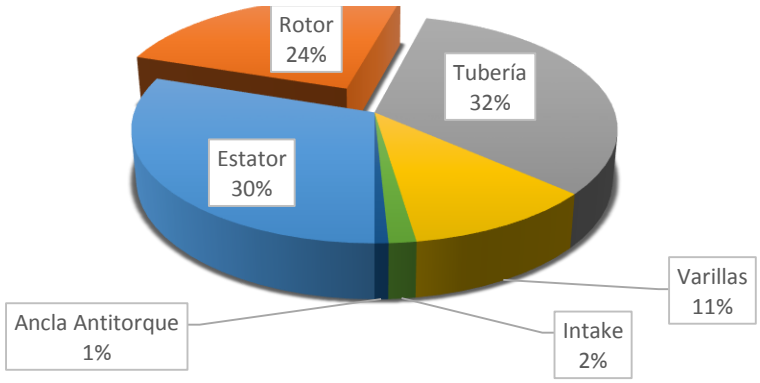
ello, que con el fin de mantener los índices de producción en el campo se plantea la instalación de otro equipo que pueda satisfacer la necesidad, pero no presenta la eficiencia adecuada. En otros casos, no se cuenta con el equipo en su totalidad, por lo tanto, se reutilizan algunos componentes de otros pozos, aumentando la probabilidad de que se presenten fallas en el sistema en menores tiempos.

**Selección de materiales:** La causa de esta falla se debe a que los materiales de los componentes de la bomba, no eran los más apropiados para el pozo. Aspectos a tener en cuenta tales como compatibilidad con fluidos del yacimiento, *dogleg*, temperaturas altas, presencia de arena, entre otros, son vitales a la hora de un diseño del sistema PCP.

De acuerdo a las causas anteriores e informes obtenidos por Ecopetrol S.A, se pudo establecer que la mayor concentración de las fallas de las bombas de cavidades progresivas se presenta en los siguientes componentes (Figura 26):

- Tubería
- Estator
- Varillas
- Rotor

Figura 26.Componentes que presentan mayor falla en las bombas PCP en el campo Casabe.




Fuente: Ecopetrol S.A (2017)

Teniendo en cuenta las causas principales de las fallas en el sistema PCP del campo Casabe se propone una agrupación de las fallas y sus causas para los distintos componentes, del siguiente modo:

**Fallas generadas por la acumulación de arena:** Estas dan lugar al arenamiento del pozo y también al atascamiento de la bomba.

**Fallas de la tubería:** Abarcan tanto la ruptura (contacto *coupling-tubing*) como la desconexión de la tubería.

**Fallas de la varilla:** Generan las desconexiones y las rupturas de las mismas.

**Falla del equipo de subsuelo:** Incluye las fallas por desgarramiento e hinchamiento del elastómero, pero así mismo la falla del ancla de tubería o un mal procedimiento de espaciamiento.

**Fallas de superficie:** Incluyen ruptura de correas, cambio de motor eléctrico, falla del cabezal de accionamiento y fallas asociadas con el variador.

Por otro lado, al analizar cómo pueden fallar los componentes permite tener una noción de los aspectos que se deben tener en cuenta para poder realizar una predicción del *Run Life* de los equipos. A continuación, se mencionarán las principales consecuencias que se presentan en las fallas de los estatores y rotores, los cuales como vimos anteriormente son componentes que fallan con mayor frecuencia en el campo Casabe.

**4.2.1 Fallas Recurrentes en Estatores** Estas son algunas de las fallas más comunes que se generan en el estator de la bomba PCP[16]:

**Falla Por trabajar en seco:** Falla que ocurre cuando no hay lubricación en el estator por largos periodos de tiempo. Causando un aumento de la temperatura, lo cual genera que el elastómero se endurezca, se quiebre y agriete[17]. En algunos casos este puede llegar a desprenderse del estator. Esta falla es ocasionada por la baja productividad del pozo, interferencia del gas u obstrucciones en la succión de la bomba (Figura 27).

[16] Saveth, K. (1998). *General Guidelines for Failure Analysis Of Downhole Progressing Cavity Pumps*.

[17] B.T Wagg. *Progressing cavity pump inspection and reporting*-C.Fer technologies

Figura 27.Elastómero dañado por altas temperaturas.

Fuente. B.T Wagg*. Progressing cavity pump inspection and reporting*-C.Fer technologies.

**Falla por histéresis:** Esta falla se caracteriza por el desprendimiento del elastómero en la línea de sello entre en rotor y el estator. La figura 28 muestra una falla en el elastómero debida al fenómeno histéresis. La histéresis es un proceso normal que ocurre en la vida útil del elastómero por una variedad de razones que al final son ocasionadas por la sobrepresión. Aunque hay algunos factores que haces que este tipo de falla se genere prematuramente tales como:

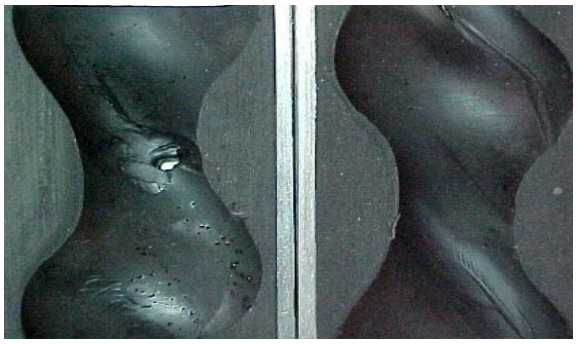
- Alta interferencia entre el rotor y estator.
- Elastómero sometido a alta presión.
- Alta temperatura/poca disipación del calor.

Figura 28.Secuencia de falla en un elastómero, debido a la histéresis.

Fuente: Hirschfelsdt.M.

**Falla por abrasión:** La severidad de esta falla puede depender de: abrasividad de las partículas, cantidad, velocidad lineal del fluido dentro de la bomba y a través de la sección transversal de la cavidad. A medida que el porcentaje de materia abrasivo incrementa, de la misma medida lo hará el desgaste de la línea desello formado por la interferencia entre el rotor-estator, causando que aumente el deslizamiento y con ello una disminución de la producción (Figura 29).

Figura 29.Elastómero dañado por la abrasión.



Fuente: Hirschfelsdt.M.

**4.2.2. Fallas Recurrentes En Rotores** Estas son algunas de las fallas más comunes que se generan en el rotor de la bomba PCP

**Desgaste Abrasivo:** Esta falla se genera cuando el recubrimiento de cromo en el rotor empieza a desgastarse, debido a una producción con componentes abrasivos. Este tipo de desgaste puede llegar solo a afectar el recubrimiento de cromo o afectar el material base, cabe destacar que en cualquiera de las dos situaciones se afecta el rendimiento de la bomba, puesto que modifica la interferencia entre el rotor y el estator[18].

El desgaste abrasivo usualmente ocurre sobre las crestas del rotor, aunque también, se puede presentar en otras secciones del mismo (Figura 30).

Figura 30.Desgaste abrasivo sobre el rotor.

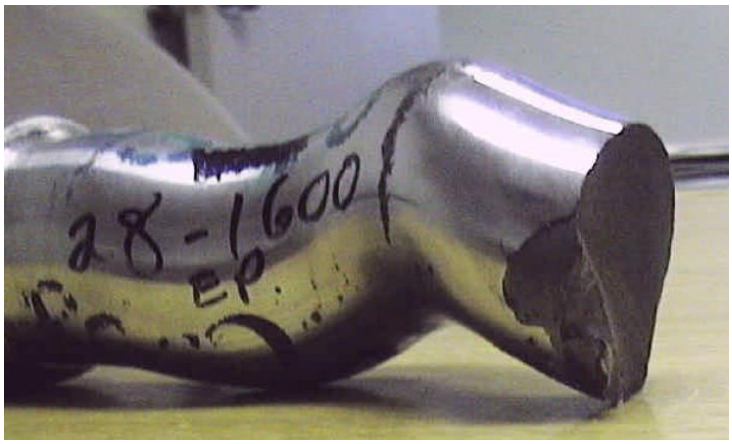
Fuente: B.T Wagg

**Rompimiento del rotor:** Puede ocasionarse por desgaste del rotor o fatiga, el rompimiento por torsión en el rotor generalmente comienza en un extremo y se propagan hasta que las fuerzas que están generando el torque rompen el

[18]Wangg.B. Artificial, W., & Systems, *Base Metal Bent Broken General Wear Good Heat Checked Pitted Pressure Washed Rippled Scored Flat Spots Base Metal*.

componente (Figura 31). Generalmente se genera durante la operación, debido al sobredimensionamiento la capacidad de producción que puede manejar la bomba y al uso excesivo de fuerza para liberar la pieza atascada durante las operaciones en campo. “Ibid”.p.[19]

Figura 31.Rompimiento del rotor.




Fuente: (B.T Wagg)

**Fatiga o torsión:** Esta falla es ocasionada por stress cíclico al cual es sometido el material. Empiezan como pequeñas grietas que van aumentando a medida que se genera el stress cíclico. Sucede cuando el rotor no se inserta adecuadamente durante la instalación inicial, causando que la bomba experimente un movimiento excéntrico más alto al que fue diseñado, resultando finalmente en una falla por fatiga.

También se ocasiona por el arenamiento, cuando la bomba se bloquea, causa que el rotor también lo haga. Como el rotor no es de material flexible, la porción intenta moverse ocasionando que llegue un punto en el cual se quiebra (Figura 32).

[19] Ibid.., p.60.

Figura 32.Falla por torsión en el rotor.

Fuente. B.T Wagg

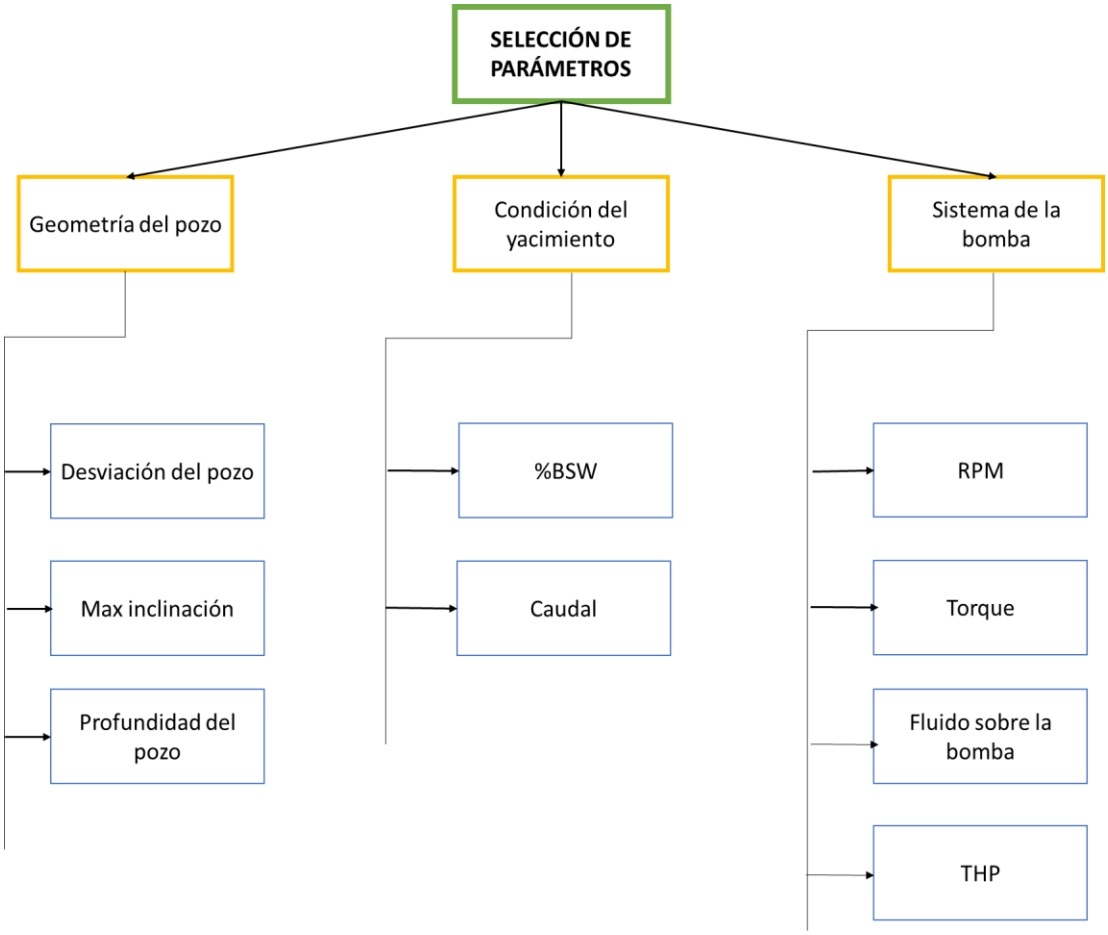
Teniendo en cuenta lo mencionado a lo largo de este capítulo, se estableció que son 4 componentes los que presentan mayor índice de falla en las bombas PCP, por tanto, el análisis de los parámetros operacionales y condiciones del pozo se establecieron bajo las posibles causas que incurren en el daño de estos cuatro elementos.

Para ello fue necesario realizar un análisis de las variables del yacimiento, del pozo y del sistema de bombeo (tanto de fondo como de superficie), que pudiesen estar afectando el tiempo de vida medio de los componentes del equipo y de esta manera generando las fallas. Para ello, se dispuso de la información de la base de datos creada anteriormente.

De esta manera, se establece que para la predicción de las fallas en el campo Casabe se tengan en cuenta las siguientes características y parámetros caudal del pozo, RPM, BSW, torque, profundidad del pozo, THP, dogleg, Liquido sobre la bomba. La (Figura 33), muestra la subdivisión de estos parámetros de acuerdo a

las regiones donde se encuentren. Estos datos, se seleccionaron para la creación del *dataset* de entrada para la predicción las fallas de las bombas mediante algoritmos de inteligencia artificial.

Figura 33.Parámetros usados para crear el modelo de predicción de las fallas en el campo Casabe.




## 5. ALGORITMOS DE INTELIGENCIA ARTIFICIAL ANALIZADOS PARA LA PREDICCIÓN DE LAS FALLAS PCP EN EL CAMPO CASABE

En este capítulo se describen los modelos de inteligencia artificial que los autores del proyecto estudiaron para seleccionar el mejor algoritmo que permita la creación de

un modelo para la predicción de fallas que se presentan en las bombas PCP del campo de Casabe y de esta manera garantizar mejores resultados.

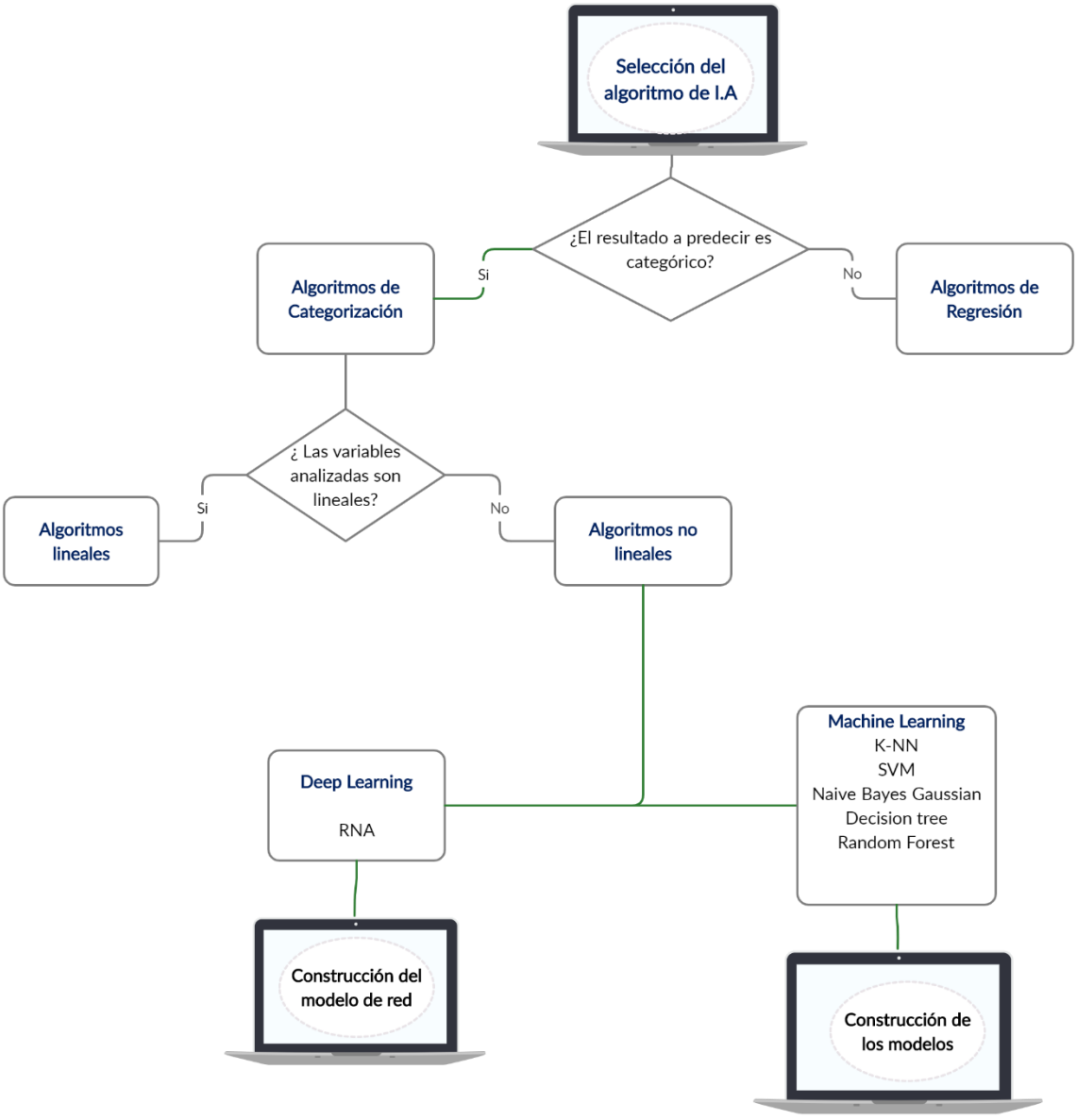
Cabe resaltar que los algoritmos de inteligencia artificial basado en aprendizaje automático como el que se va a utilizar para dar solución al presente problema de determinación del *Run Life* es de tipo Categórico y no es lineal, ya que las variables no presentaban este tipo de comportamiento. Los algoritmos a usar se determinaron a partir del siguiente flujograma de decisiones (Figura 34).

## 5.1. ALGORITMOS DE REGRESIÓN

En cuanto a los algoritmos de regresión ya sean lineales o no lineales, se utilizan para predecir un valor real, un ejemplo de ello puede ser el salario de una persona. Si su variable independiente es el tiempo, entonces está pronosticando valores futuros, de lo contrario su modelo está prediciendo valores presentes pero desconocidos.

El resultado de esta técnica generara un valor numérico, dentro de un conjunto infinito de posibles resultados

Figura 34.Flujograma para la escogencia de los algoritmos a usar para la predicción del Run life de las bombas PCP en el Campo Casabe.



## 5.2. ALGORITMOS DE CLASIFICACIÓN

Estos algoritmos son utilizados cuando se busca que el algoritmo sea capaz de clasificar el resultado en una clase, dentro de un número limitado de estas. Las clases hacen referencia a categorías arbitrarias según el tipo de problema.
Por ejemplo, si se quiere detectar si un correo es spam o no, sólo hay 2 clases, y el algoritmo de *Machine Learning* de clasificación, solo podrá obtener dos posibles salidas.

Por otro lado, diversos algoritmos de *Machine Learning* pueden generar su resultado de clasificación a partir de probabilidades, Es decir, nos pueden decir que un correo es spam con una probabilidad del 89%. O incluso que una imagen tiene un 67% de probabilidades ser un perro, un 18% de ser un gato, un 9% de ser una oveja, etc. Normalmente cuando se presenten estos tipos de problemas se eligen las clases que han presentado una mayor probabilidad de ocurrencia.

### Técnicas de Machine Learning para Clasificación

Existe una amplia variedad de aplicaciones en las cuales se pueden usar métodos de clasificación para la solución de problemas, los cuales pueden ir desde problemas que involucren el campo de la medicina, predicción de fallas en equipos e incluso en marketing. Los modelos de clasificación pueden incluir modelos tales como:

- Regresión Logística
- K vecinos más cercanos (K-NN)
- Máquina de soporte vectorial (SVM)
- Naive Bayes Gaussian
- Árboles de decisión (Decision tree)

- Bosques aleatorios (Random Forest)
- Redes neuronales (RNA)

En presente trabajo busca predecir el Run life de las bombas PCP en el campo Casabe en un determinado rango de tiempo, es decir, predecir las fallas mediante cuatrimestres, por lo tanto, la solución del problema hace parte de un problema categórico, para el caso específico de este proyecto se estudiaron 6 de las 7 técnicas mencionadas anteriormente debido a que las regresiones logísticas son utilizadas cuando los datos presentan un comportamiento lineal, lo cual en el caso específico del dataset a utilizar no sigue este tipo de tendencia. Los algoritmos de clasificación utilizados se muestran en la figura 35. Estos algoritmos también han sido escogidos gracias a la potencia de análisis que presentan, por ser prácticos y por manejar un código de escritura sencilla.

Con la finalidad de entender cada una de las técnicas analizadas para el desarrollo de este proyecto, se procede a realizar una revisión de cada uno de los algoritmos de inteligencia artificial usados en el mismo, donde se busca dar a conocer el principio básico de funcionamiento, sus características, entre otros aspectos.

Figura 35.Métodos de Predicción analizados para predecir el run life de las bombas PCP en el campo Casabe.


## 5.3. RED NEURONAL ARTIFICIAL (RNA)

Este fue el primer algoritmo que se probó para predecir las fallas de las bombas PCP en el campo Casabe, dado a la potencia que presenta a la hora de predecir datos. Este es uno de los métodos más potentes de *Machine Learning*, además, de que se está implementando en gran medida en la industria petrolera durante los últimos años[20].

Para comprender como funciona este algoritmo es necesario retomar un poco la teoría de su funcionamiento. Las redes neuronales artificiales, representan una clase de modelos de aprendizaje automático, lo que quiere decir, que estas van aprendiendo por sí solas a media que se les introduce información. Un ejemplo de ello puede ser cuando se usa algún buscador para consultar algún tema, se escriben palabras claves y se obtiene diversos documentos que hablan sobre el tema que

[20] Antonio Gulli, S. P. (2017). *Deep Learning with Keras-Packt (2017)*.

precisa. Esto funciona, porque estos buscadores han sido diseñados con redes neuronales las cuales van codificando la información suministrada y buscando dentro de su base de datos los temas que más se van relacionando a la búsqueda, y a medida que se van introduciendo más datos, esta red se va haciendo más poderosa. Es por ello que para que una red neuronal funcione a la perfección se necesita de una base de datos muy grande.

Es interesante observar, que las redes neuronales se basan en estudios sobre los sistemas nerviosos centrales de los mamíferos, el cual está compuesto de diversas neuronas biológicas. Cada red está a su vez constituida por varias neuronas interconectadas, organizadas en capas, las cuales intercambian mensajes, que son codificados para obtener una salida o respuesta.

**5.3.1. Elementos principales de una red neuronal artificial** Las redes neuronales están constituidas básicamente por neuronas o datos de entrada, las uniones de estas forman capas y la unión de estas últimas constituyen una red neuronal, (Figura 36).

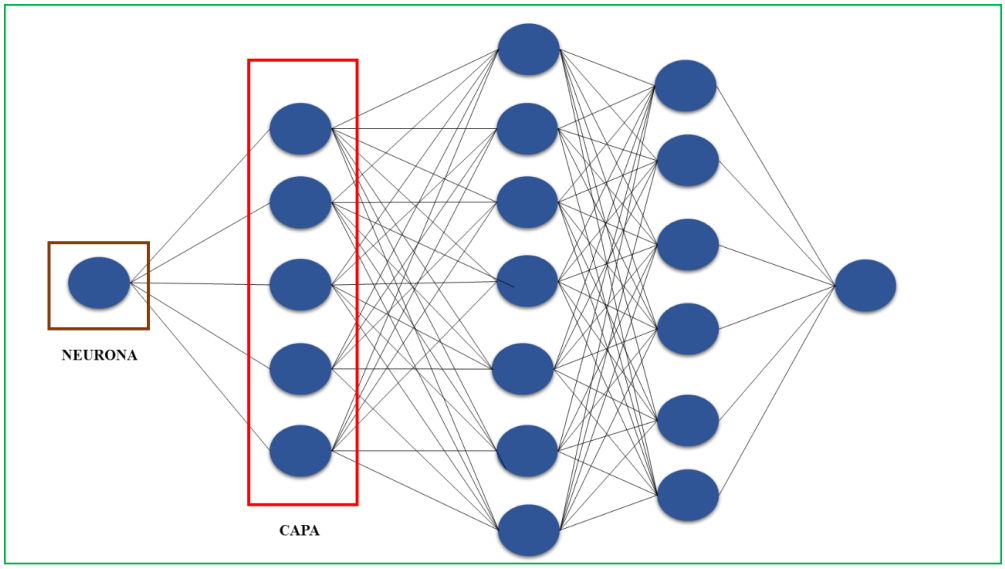
Las redes neuronales se encuentran agrupadas por capas las cuales pueden agruparse en distintos tipos tales como:

**Capa de entrada**: Esta en la que recibe las señales de entrada en la red, es decir esta se encarga de recibir los datos, que más adelante harán parte del proceso de la misma red.

**Capas ocultas:** Estas capas pueden tener diferentes conexiones, son las encargadas de generar la topología de la red.

**Capa de salida:** Recibe la información proveniente de la interacción de las capas ocultas y trasmite una respuesta, es la salida de la red neuronal.

Figura 36.Elementos principales de una red neuronal.




**5.3.2. Estructura Básica De Una Neurona Artificial.** La neurona artificial es una unidad procesadora interna en la RNA que funciona con cuatro elementos:

**Elemento receptor:** Este corresponde al punto en el cual llegan una o varias señales de entrada Xi, que generalmente provienen de otras neuronas y que son atenuadas o amplificadas. Cada una de ellas con arreglo a un factor de peso Wi, que constituye la conectividad entre la neurona fuente de donde provienen y la neurona de destino en cuestión.

**Elemento sumado:** Efectúa la suma algebraica ponderada de las señales de entrada, de acuerdo con su peso, aplicando la siguiente expresión.

$$S = \sum Wi * Xi$$

**Función de activación:** Este elemento aplica una función no lineal de umbral (que frecuentemente es una función escalón o una curva logística) a la salida del sumador para decidir si la neurona se activa, generando una salida o no.

**Elemento de salida:** Este produce la señal, de acuerdo con el elemento anterior, que constituye la salida de la neurona.

## 5.4. MÁQUINA DE SOPORTE VECTORIAL (SVM)

Este algoritmo de inteligencia artificial, fue desarrollado en 1998 y desde entonces se ha aplicado con éxito en la resolución de diversos problemas en diferentes áreas, las cuales van desde la predicción de series temporales, reconocimiento de rostro e incluso el procesamiento de datos biológicos para el diagnóstico médico.

En muchas aplicaciones, las SVM han mostrado tener gran desempeño, más que las máquinas de aprendizaje tradicional como las redes neuronales y han sido introducidas como herramientas poderosas para resolver problemas de clasificación[21].

La SVM es un modelo que parte de un dataset de entrenamiento, busca etiquetar los datos en diferentes clases y representar dichos datos en puntos en el espacio, para así, crear diferentes grupos con una distancia de separación definida. Esto, con el objetivo de cuando se ingrese el dato de muestra (dato a predecir), se ponga en correspondencia con el modelo para ser clasificado correctamente en función de su proximidad y dar como resultado el grupo categórico al que corresponde.

En otras palabras, una SVM primero mapea los puntos de entrada a un espacio de características de una dimensión mayor y encuentra un hiper-plano que los separa

[21] Gustavo, A. (2014). Las máquinas de soporte vectorial *(SMS) Redalyc*. *January 2014*.

y maximiza generando un margen de distanciamiento denotado con la letra m, separación que realiza entre las clases que existen en el espacio, (Figura 37). El hiper-plano separador esta dado de manera general por:

Ecuación 2. Hiper-plano separador del modelo SVV

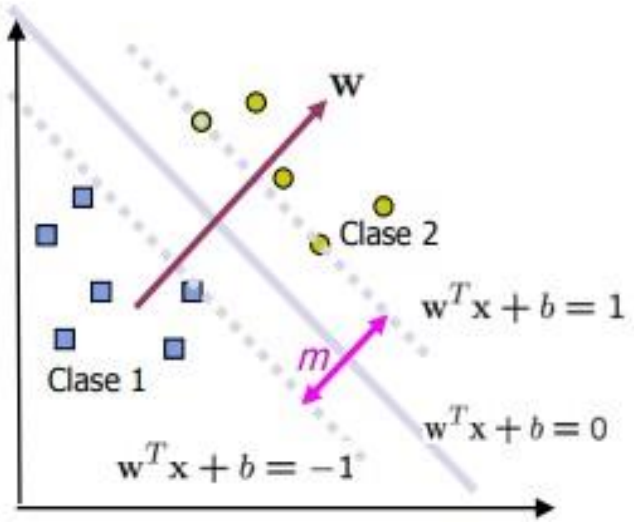
$$(w * x) + b = 0 \quad donde\ w, x\ \in R^n\ , b\ \in R.$$

El modelo, encuentra el vector *w* de pesos, que contiene la ponderación de cada atributo (en el caso de este trabajo, encontrar el peso de cada característica de pozo), e indica qué tanto aportan en el proceso de predicción. En tanto que *b*, define el umbral de decisión, llamado usualmente *bias* en inglés[22].

Como se observa en la figura 36, la manera más simple de realizar la separación es mediante una línea recta, un plano recto o un hiper-plano N-dimensional, pero los universos a clasificar no se suelen presentar en forma ideal con dos dimensiones como ocurre en el ejemplo gráfico, sino que un algoritmo SVM debe tratar con más de dos variables predictoras, curvas no lineales de separación, casos donde los conjuntos de datos no pueden ser completamente separados, clasificaciones en más de dos categorías. La representación por medio de funciones núcleo o *Kernel* ofrece una solución a este problema, proyectando la información a un espacio de características de mayor dimensión el cual aumenta la capacidad computacional de las máquinas de aprendizaje lineal (Parra, 2019).

[22] JIMEMNEZ. L, RENGIFO. P. Al interior de una máquina de soporte vectorial. Facultad de ciencias Naturales y exactas. Universidad del Valle. Octubre 2010. Pg 74.

Figura 37. Máquina de soporte vectorial.




Fuente: Gustavo, A. (2014). Las máquinas de soporte vectorial *(SVMs) Redalyc*. *January 2014*.

**Ventajas y desventajas del algoritmo**

**Ventajas:**

- Eficientes en espacios de grandes dimensiones.
- Puede hacer frente a problemas no lineales mediante el método de kernel.
- Eficaces en los casos en que el número de dimensiones es mayor que el número de muestras.
- Utiliza un subconjunto de puntos de entrenamiento en la función de decisión (llamados vectores de apoyo), por lo que también es eficiente en la memoria.
- se pueden especificar diferentes funciones del núcleo para la función de decisión. Se proporcionan núcleos comunes, pero también es posible especificar núcleos personalizados.
- Garantiza la convergencia.

**Desventajas:**

- Si el número de características es mucho mayor que el número de muestras, se evita el sobreajuste en la elección de las funciones del kernel y el término de regularización es crucial.
- Las máquinas de vectores de apoyo no proporcionan directamente estimaciones de probabilidad.
- Las máquinas de soporte vectorial en scikit-learn admiten como entrada tanto vectores de muestra densos (numpy.ndarray y convertibles a ese por numpy.asarray) como escasos (cualquier scipy.sparse). Sin embargo, para utilizar una SVM para hacer predicciones de datos dispersos, debe haberse ajustado a esos datos. Para un rendimiento óptimo, se debe utilizar el orden C numpy.ndarray (denso) o **scipy.sparse.csr_matrix (disperso)**.

Debido a las ventajas mencionadas anteriormente, las aplicaciones de la SVM se encuentran fácilmente en varios campos de la ciencia y la ingeniería, como la vigilancia de las condiciones y diagnóstico de fallos en equipos.

**5.4.1. Atributos para el desarrollo del modelo SVM en la librería sklearn de Python.** Sklearn maneja dos clases de atributos los cuales corresponden a C y Kernel.

**C:** Equilibra la correcta clasificación de los datos de entrenamiento con la maximización del margen de la función de decisión. Para valores mayores de **C**, se aceptará un margen menor si la función de decisión es mejor para clasificar correctamente todos los puntos de entrenamiento, si se aumenta el parámetro C, se está apostando a que los datos de entrenamiento contienen las observaciones más extremas posibles, es decir que las futuras observaciones estarán más lejos

de los límites que los puntos en los que has entrenado el modelo. Un valor de **C** más bajo fomentará un margen más grande, por lo tanto, una función de decisión más simple, a costa de la precisión de la formación. El atributo **C** se comporta como un parámetro de regularización en el SVM, le indica al algoritmo cuánto importaran los puntos mal clasificados.

**KERNEL:** El algoritmo SVM se implementa en la práctica utilizando un kernel, el cual transforma un espacio de datos de entrada en la forma requerida. SVM utiliza una técnica llamada el kernel trick. Aquí, el núcleo toma un espacio de entrada de baja dimensión y lo transforma en un espacio de mayor dimensión. Linear: El núcleo lineal se utiliza cuando los datos son separables linealmente, es decir, se pueden separar usando una sola línea. Es uno de los núcleos más comunes que se utilizan. Se utiliza principalmente cuando hay un gran número de características en un conjunto de datos en particular.

- **Kernel polinómico:** Es una forma más generalizada del núcleo lineal. El núcleo polinómico puede distinguir el espacio de entrada curvo o no lineal.
- **Kernel de función de base radial:** Es una función de núcleo popular que se utiliza comúnmente en la clasificación de SVM. RBF puede mapear un espacio de entrada en un espacio dimensional infinito. utiliza curvas normales alrededor de los puntos de datos, y las sumas de manera que el límite de decisión puede ser definido por un tipo de condición de topología como las curvas en las que la suma es superior a un valor de 0,5.

## 5.5. K-VECINOS MÁS CERCANOS (K-NN)

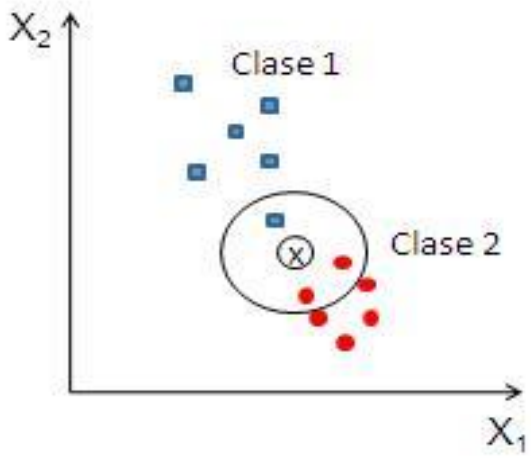
El algoritmo clasifica cada dato nuevo en el grupo que corresponda, según la cercanía que este tenga con los datos que se encuentran a su alrededor. Es decir, calcula la distancia del elemento nuevo respecto a cada uno de los datos ya existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el

grupo al cual pertenece. Este grupo será, por tanto, el de mayor frecuencia con menores distancias.

El funcionamiento de este método de clasificación puede detallarse en la figura 38 en el cual se representan 12 muestras las cuales pertenecen a 2 clases distintas, la clase 1 formada por seis muestras representadas de color azul y, la clase 2, por seis muestras representadas de color rojo. Para el ejemplo se han seleccionado tres vecinos, k=3.

Al analizar la muestra x, el modelo estima un radio a analizar y como se observa en la figura, distingue 1 vecino de la clase 1 y 2 vecinos de la clase 2. Como la regla tomada es 3-nn, la muestra a analizar se asignará a la clase 2. Cabe aclarar que si la muestra se analiza con un k= 1 (1-nn), la clase asignada a la muestra x, sería la clase 1, ya que el vecino más cercano a la muestra pertenece a la Clase 1.

Figura 38.Representación del espacio de K-Vecinos más cercanos.




Fuente: PARRA. F. Estadística y Machine Learning con R. Universidad Nacional de Educación a Distancia. España, 2019.

El K-NN es un algoritmo de aprendizaje supervisado, es decir, que a partir de un juego de datos inicial su objetivo será el de clasificar correctamente todas las instancias nuevas[23].

De igual forma, el método trabaja en dos fases:

**Fase de entrenamiento**, en el cual se almacena los vectores característicos y las etiquetas de las clases de los ejemplos de entrenamiento. Para el caso de este trabajo, se toma el dataset con cada una de las características de cada pozo y se introduce en el modelo para su respectivo entrenamiento.

**Fase de test**, aquí, la evaluación de la muestra (de la cual, no se conoce su clase) se representa por un vector en el espacio característico. Se calcula la distancia entre los vectores almacenados (datos usados del entrenamiento) y el nuevo vector (dato a predecir), y se seleccionan los k ejemplos más cercanos. La muestra (dato de entrada a predecir) es clasificada con la clase que más se repite en los vectores seleccionados.

**5.5.1. Atributos para el desarrollo del modelo K-NN en la librería sklearn de Python.** La clasificación de vecinos en el **KNeighborsClassifier** es la técnica más usada. La elección óptima del valor depende en gran medida de los datos: en general, un valor mayor suprime los efectos del ruido, pero hace que los límites de la clasificación sean menos claros.

**RadiusNeighborsClassifier:** Implementa el aprendizaje basado en el número de vecinos dentro de un radio fijo de cada punto de consulta, donde es un valor de punto flotante especificado por el usuario.

[23] García, C., & Gómez, I. (2006). Algoritmos de aprendizaje: knn & kmeans. *Universidad Carlos III de Madrid*.

En los casos en que los datos no se muestrean de manera uniforme, la clasificación de vecinos basada en el radio en el **RadiusNeighborsClassifier** puede ser una mejor elección. El usuario especifica un radio fijo, de tal manera que los puntos en vecindarios más escasos utilizan menos vecinos cercanos para la clasificación. Para los espacios de parámetros de alta dimensión, este método se vuelve menos efectivo debido dimensionalidad.

La clasificación básica de vecinos más cercanos utiliza pesos uniformes: es decir, el valor asignado a un punto de consulta se calcula a partir de la mayoría simple de los votos de los vecinos más cercanos. En algunas circunstancias, es mejor ponderar los vecinos de tal manera que los vecinos más cercanos contribuyan más al ajuste. Esto se puede lograr a través de la palabra clave de ponderación. El valor por defecto, pesos = 'uniforme', asigna pesos uniformes a cada vecino. pesos = 'distancia' asigna pesos proporcionales al inverso de la distancia del punto de consulta. Alternativamente, se puede suministrar una función de la distancia definida por el usuario para calcular los pesos.

El clasificador del **NearestCentroid** más cercano es algoritmo que representa cada clase por el centroide de sus miembros. No necesita parámetros para elegir, lo que lo convierte en un buen clasificador de línea base. Sin embargo, sufre en las clases no convexas, así como cuando las clases tienen varianzas diferentes, ya que se supone una varianza igual en todas las dimensiones, por otro lado, este método solo funciona para problemas de regresión.

### 5.6. NAIVE BAYES GAUSSIAN

Naive Bayes es uno de los clasificadores más utilizados por su simplicidad y rapidez. Se basa en una técnica de clasificación y predicción supervisada que construye

modelos que predicen la probabilidad de posibles resultados, en base al Teorema de Bayes, también conocido como teorema de la probabilidad condicionada[24].

Ecuación 3. Fórmula de NAIVE Bayes Gaussian

$$P(A/B) = \frac{P(B/A) * P(A)}{P(B)}$$

Donde:

- P(A): Probabilidad de que la hipótesis A sea cierta (independientemente de los datos).

- P(B): Probabilidad de los datos (independientemente de la hipótesis). Esto se conoce como probabilidad previa.

- P(A|B): Probabilidad de la hipótesis A dada los datos B. Esto se conoce como la probabilidad posterior.

- P(B|A): es la probabilidad de los datos b dado que la hipótesis A era cierta. Esto se conoce como probabilidad posterior (Parra, 2019).

Caben destacar dos partes en el algoritmo. El modelo presenta dos fases, la primera es la construcción del modelo, y la segunda clasificar una muestra con dicho modelo:

**Fase 1:** Creación del modelo. Para ello se necesitan cuatro pasos:

1. Calcular las probabilidades a priori de cada clase.
2. Para cada clase, realizar un recuento de los valores de atributos que toma cada ejemplo. Se debe distribuir cada clase por separado para mayor comodidad y eficiencia del algoritmo.

[24] PARRA. F. Estadística y Machine Learning con R. Universidad Nacional de Educación a Distancia. España, 2019.

3. Aplicar la Corrección de Laplace, para que los valores “cero” no den problemas.
4. Normalizar para obtener un rango de valores [0,1].

**Fase 2:** Calcular la predicción del modelo. Para ello se necesitan dos pasos:

1. Para cada clase disponible, se determinan los valores de probabilidad de cada valor de los atributos del nuevo ejemplo.
2. Aplicar la fórmula de Naive Bayes. (Ecuación 3)

Es importante tener en cuenta, que algunas ventajas del modelo son su facilidad y su rápida predicción de datos. Se caracteriza por un mejor comportamiento en la predicción de variables categóricas comparado con variables numéricas. No obstante, una de sus desventajas es que, si la variable categórica tiene una categoría en el dataset de prueba, que no se observó en el dataset de entrenamiento, el Naive Bayes asignará una probabilidad de 0 y no podrá hacer una predicción. Esto se conoce como frecuencia cero. Para resolver esto, se puede utilizar la técnica de alisamiento.

Por otro lado, una de las asunciones del modelo es la de predictores independientes. En la práctica es difícil observar un conjunto de predictores que sean completamente independientes.

**5.6.1. Atributos para el desarrollo del modelo Naive Bayes Gaussian en la librería sklearn de Python.** Los atributos con los que se pueden construir los algoritmos de Naive Bayes Gaussian puede ser de dos tipos var_smoothing y Priors[25].

[25] Documentación de la librería sklearn de Python. Fuente: https://scikit-learn.org/0.15/modules/generated/sklearn.ensemble.RandomForestClassifier.html

**var_smoothing:** Porción de la mayor variación de todas las características que se añade a las variaciones para la estabilidad del cálculo. La variable, var_smoothing, añade artificialmente un valor definido por el usuario a la varianza de la distribución (cuyo valor por defecto se deriva del conjunto de datos de formación). Esto amplía esencialmente (o "suaviza") la curva y da cuenta de más muestras que están más alejadas de la media de la distribución.

**Priors:** Probabilidades previas de las clases. Si se especifica, las probabilidades previas no se ajustan según los datos.

## 5.7. ÁRBOLES DE DECISIÓN (DECISION TREE)

Los árboles de decisión son un modelo de predicción cuyo objetivo principal es el aprendizaje inductivo basado en observaciones y construcciones lógicas, los cuales sirven para representar y categorizar una serie de condiciones que suceden sucesivamente y permiten dar solución a un problema[26].

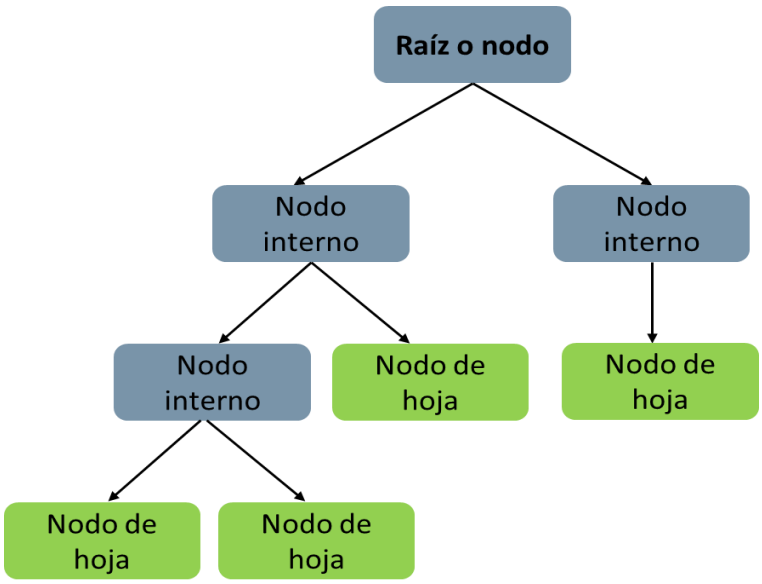
Estos tipos de algoritmos pueden ser usados para resolver problemas de tipo categórico como de regresión. Para el caso específico del problema de predicción de fallas a resolver se usó este algoritmo como un problema de clasificación en el cual se ingresarían las características de entrada del *Dataset* y se obtendría de esta un valor del cuatrimestre en el cual la bomba PCP fallara. Este modelo divide los datos múltiples veces de acuerdo a ciertos parámetros de corte lo cual crea diferentes subdivisiones de los datos con características en común.

[26] Barrientos, R., Cruz, N., Acosta, H., Rabatte, I., Gogeascoechea, M., Pavón, P., & Blázquez, S. (2009). *Arbol de decisión como herramienta en el diagnóstico médico*. 20–24.

**5.7.1. Estructura del algoritmo Decision Tree.** Un árbol de decisión es un mapa de los posibles resultados de una serie de decisiones relacionadas. Permite realizar una comparación de posibles acciones entre sí según sus costos, probabilidades y beneficios.

Se denominan árboles debido a que la representación gráfica que forman se asemeja a estos, debido a que presentan ramificaciones, en la Figura 39, se muestra un esquema básico de este algoritmo.

Figura 39.Representación gráfica del modelo de árboles de decisión.




Estos modelos contienen un nodo principal que se puede considerar como la raíz del modelo, el cual es el atributo a partir del cual se inicia el proceso de clasificación; consecuentemente se encuentran los nodos internos que corresponden a cada una de las preguntas acerca del atributo en particular del problema. Cada respuesta de los cuestionamientos es representada mediante un nodo de hoja. Las ramas que

salen de cada uno de estos nodos se encuentran etiquetadas con los posibles valores del atributo. Los nodos finales o nodos hoja corresponden a una decisión, la cual coincide con una de las variables clase del problema a resolver. La solución al problema puede tomar valores discretos o continuos dependiendo de las variables usadas en el proceso.

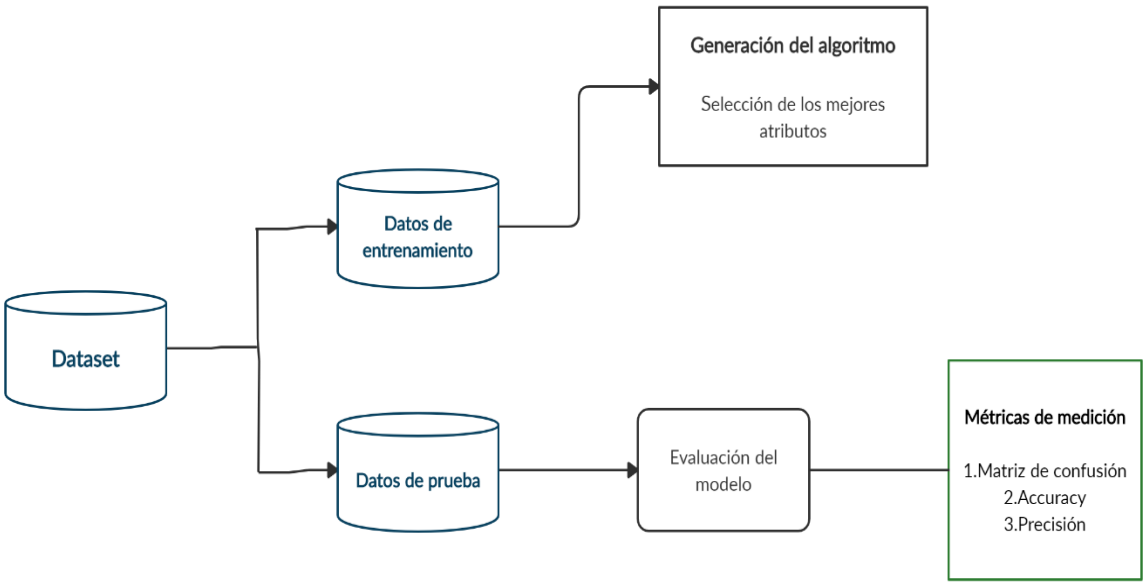
**5.7.2. Funcionamiento del algoritmo.** El funcionamiento del algoritmo de *decision tree* consiste en:

1. Seleccionar el mejor atributo utilizando Medidas de selección de atributos para dividir los datos de un Dataset.
2. Hacer que ese atributo sea un nodo de decisión y se divida el conjunto de datos en subconjuntos más pequeños. recursivamente para cada nodo hasta que una de las condiciones coincida en que:

- Todas las tuplas pertenecen al mismo valor de atributo.
- No quedan más atributos, la Figura 40 muestrea el proceso de un algoritmo general por *decision tree*[27].

[27] Documentación de la librería sklearn de Python. Fuente: https://scikit learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Figura 40.Esquema de la representación de los procesos al momento de crear un algoritmo de Decision tree.




**5.7.3. Atributos para el desarrollo del modelo de decision tree en la librería sklearn de Python.** Para los modelos de Decision tree fueron evaluados los atributos criterios y Divisor

**Criterios:** Dentro de los criterios del algoritmo de árboles de decisión se encuentras dos tipos: gini y entropy, el método de gini es el que viene ya por default en este algoritmo.

- **Gini** consiste en la ganancia de información la cual es una propiedad estadística que mide qué tan bien un atributo dado separa los ejemplos de entrenamiento de acuerdo con la clasificación objetivo.
- **Entropy** Mide la impureza del conjunto de entrada. la entropía se conoce como aleatoriedad o impureza en el sistema. En teoría de la información, se refiere a la impureza en un grupo de ejemplos. La ganancia de información es una disminución de la entropía.

**Divisor:** Para los divisores de encuentran best y random, donde best es el divisor que viene por default.

La estrategia utilizada para elegir la división en cada nodo es la siguiente, las estrategias soportadas con "best" sirven para elegir la mejor división y "random" para elegir la mejor división aleatoria.

## 5.8. RANDOM FOREST

El algoritmo de inteligencia artificial de árboles aleatorios (*Random Forest*), es un tipo de algoritmo no supervizado que presenta una gran versatilidad a la hora de resolver problemas tanto categóricos como de regresión. Es uno de los métodos más novedosos, sus inicios datan del año 2001. Este método se basa en el uso de dos algoritmos, el *bootstrap* y el *bagging.*

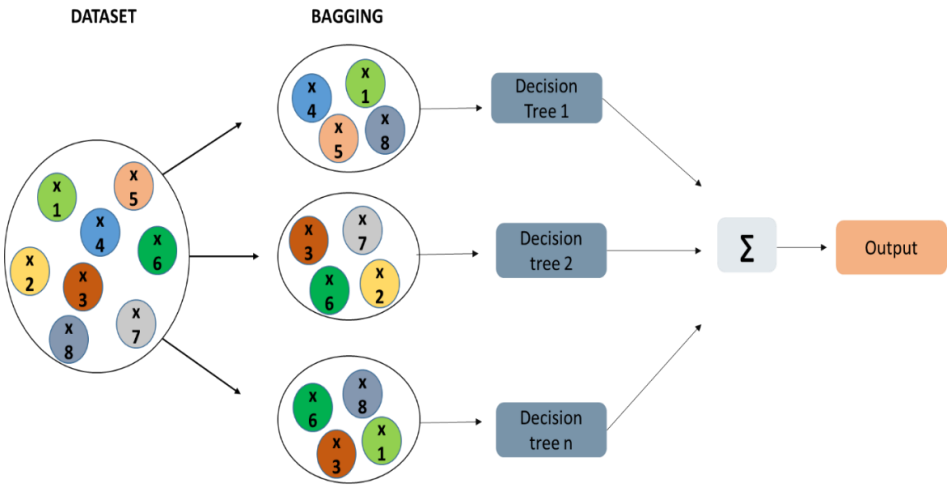
**5.8.1. Algoritmos de Bagging y Boosting.** Estas técnicas se basan en el uso de modelos individuales para el desarrollo de un mejor método. En *Bagging* se da a los modelos un peso igual. Mientras que el Boosting se dan distintos pesos a los modelos con el objetivo de brindarles mayor ponderación a aquellos que resaltan más.

**Algoritmos de Bagging:** El principal objetivo de los algoritmos de *Bagging* es la reducción de la varianza. Son métodos donde los algoritmos simples son usados en paralelo. El principal objetivo de los métodos en paralelo es el de aprovecharse de la independencia que hay entre los algoritmos simples, ya que el error se puede reducir bastante al promediar las salidas de los modelos simples.

Con el fin de entender mejor el *Bagging*, suponga que se tiene un conjunto de datos del mismo tamaño, con los cuales se construirá un árbol de decisión para cada conjunto de datos. Se podría pensar que dado a que los datos son iguales cada árbol será casi idéntico a otro y que, por lo tanto, se obtendrá una misma predicción para un nuevo caso; pero no es cierto, más aún, si los conjuntos de datos son reducidos. Si los datos de entrenamiento sufrieran algún cambio, entonces se tendría como resultado fácilmente un atributo diferente, lo cual implica que existe la posibilidad de que en los casos de prueba para algunos árboles de decisión se produzcan predicciones correctas y otras no.

*Bagging* es un método de aprendizaje estadístico cuyo procedimiento tiene como propósito la reducción de la varianza, una forma natural de reducir la varianza, y por ende, aumentar la precisión de la predicción de un método de aprendizaje estadístico, es seleccionar una gran cantidad de conjuntos de entrenamiento de la población y construir un modelo de predicción independiente utilizando cada conjunto de entrenamiento. La figura 41 permite comprender este algoritmo de una forma ilustrativa.

Figura 41.Esquema Interno del Funcionamiento del modelo de Random Forest.



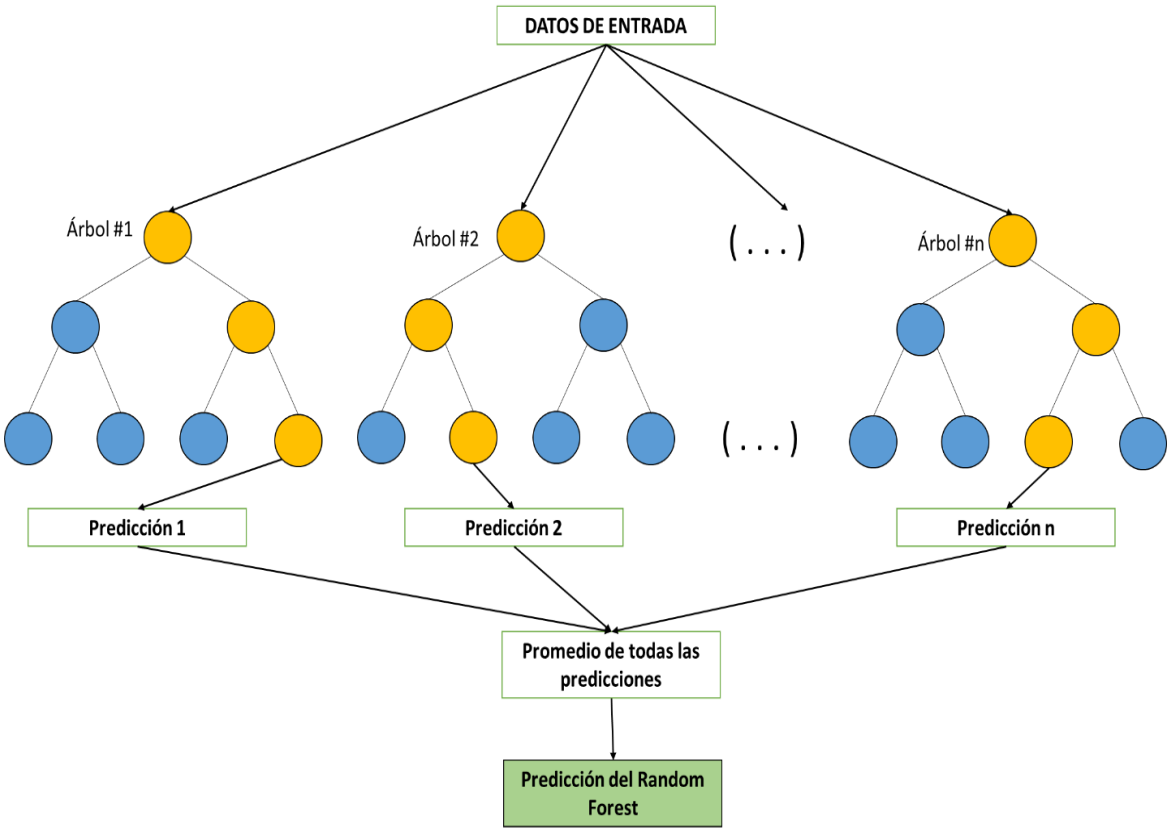
**Algoritmos de Boosting:** El principal objetivo de los algoritmos de *boosting* es el de la reducción del sesgo, En estos algoritmos, los modelos simples son utilizados secuencialmente, es decir, cada modelo simple va delante o detrás de otro modelo simple. El principal objetivo de los métodos secuenciales es el de aprovecharse de la dependencia entre los modelos simples. El rendimiento general puede ser mejorado haciendo que un modelo simple posterior genere más importancia a los errores cometidos por un modelo simple previo. Poniendo un ejemplo, es como si al resolver un problema, se aprovechara el conocimiento de los errores de otros, para no cometerlos nuevamente.

Las predicciones de cada modelo simple, se combinan por medio de una caracterización de las respuestas con la mayor tendencia (Problemas de clasificación) o por medio de una suma ponderada (Problemas de regresión) para producir la predicción final. La diferencia con el *bagging* es que en el *boosting* los algoritmos no se entrenan independientemente, sino que se ponderan según los errores de los anteriores.

**5.8.2. Funcionamiento del algoritmo Random Forest.** Una vez descritos estos algoritmos bases se detalla el funcionamiento del algoritmo de aprendizaje no supervisado de *Random Forest*. Este se puede comprender de una mejor si se piensa de forma literal, es decir, pensar en un bosque en el cual, en su interior tienen un gran conjunto de árboles. Lo que quiere decir que este algoritmo trabajara con diversos árboles de decisión (*Decision tree*). Cabe resaltar, que entre más árboles se tengan en el modelo mejor será la predicción que se obtenga. Estos árboles de decisión se generan aleatoriamente y a partir de cada uno se obtendrá una respuesta, la mayor frecuencia obtenida de todos los árboles será la solución del

modelo de *Random Forest*, en la figura 42, se muestra una representación gráfica de este modelo[28].

Figura 42. Representación gráfica del modelo de Random Forest.




Cabe resaltar que una ventaja que se presenta a la hora de utilizar el algoritmo de *Random Forest,* es que combina la facilidad de los algoritmos de *Decision tree*, los cuales son fáciles de construir, usar e interpretar, pero los cuales en la práctica pueden llegar a generar problemas, puesto que trabajan correctamente con la data

[28] Breiman, L. (2001). ST4_Method_Random_Forest. *Machine Learning*, *45*(1), 5–32.

entrenada, pero al momento de clasificar nuevas muestras puede generar errores. Mientras que los algoritmos de *Random Forest*, al combinar diversos *Decision tree* pueden generar una respuesta más flexible y mejorar las predicciones.

### 5.8.3. Pasos para la creación de un modelo de *Random Forest*

**Primer paso:** Construcción de un *Boostrapped Dataset*. El cual consiste en crear un nuevo Dataset con nueva organización del que se tenía en el Dataset original, esto se puede hacer de manera aleatoria cambiando uno o varios conjuntos de datos.

Para comprender de una mejor manera este paso se mostrará un ejemplo sencillo, donde se muestra un *Dataset* original y el *Dataset Boostrap* creado a partir de este, se puede observar que los conjuntos se mantienen durante los cambios efectuados, pero la única variación que existe en el orden de las posiciones que han tomado, es decir en el dataset original en primer conjunto era el del array cero, mientras que en el dataset nuevo toma el array número cuatro. Por otro lado, cabe resaltar que se puede hacer un solo cambio en el orden o diversos, lo que importa es que ninguno sea igual al anterior.

Tabla 8. Dataset original de ejemplo explicativo

| | DOG LEG | TORQUE | RPM | % BSW | CAUDAL (BBL/D) | RUN LIFE |
|---|---|---|---|---|---|---|
| 0 | 1.44 | 0,55 | 185 | 35 | 45 | 1 |
| 1 | 2.14 | 0.35 | 125 | 85 | 45 | 4 |
| 2 | 2.68 | 0.45 | 175 | 75 | 195 | 2 |
| 3 | 2.58 | 0.48 | 105 | 85 | 75 | 4 |
| 4 | 1.77 | 0.15 | 85 | 85 | 15 | 1 |

Tabla 9. Dataset Boostrap de ejemplo explicativo

| | DOG LEG | TORQUE | RPM | % BSW | CAUDAL (BBL/D) | RUN LIFE |
|---|---|---|---|---|---|---|
| 4 | 1.77 | 0,15 | 85 | 85 | 14 | 1 |
| 2 | 2.68 | 0.45 | 175 | 75 | 195 | 2 |
| 1 | 2.14 | 0.35 | 125 | 85 | 45 | 4 |
| 3 | 2.58 | 0.48 | 105 | 85 | 75 | 4 |
| 0 | 1.44 | 0.55 | 185 | 35 | 45 | 1 |

**Segundo paso:** Crear un árbol de decisión a partir del *Boostrapped Dataset*, pero usando un solo subconjunto aleatorio (Columnas) en cada paso.

### 5.8.3. Ventajas y desventajas de *Random Forest*.

**Ventajas**

- Pocas suposiciones y por lo tanto la preparación de los datos son mínima.
- Puede manejar hasta miles de variables de entrada e identificar las más significativas. Método de reducción de dimensionalidad.
- Una de las salidas del modelo es la importancia de variables**.**
- Incorpora métodos efectivos para estimar valores faltantes.

**Desventajas**

- Pérdida de interpretación
- Bueno para clasificación, no tanto para regresión. Las predicciones no son de naturaleza continua.
- En regresión, no puede predecir más allá del rango de valores del conjunto de entrenamiento.
- Poco control en lo que hace el modelo (modelo caja negra para modeladores estadísticos).

**5.8.5. Atributos para el desarrollo del modelo de Random Forest en la librería sklearn de Python**. Los parámetros en el bosque aleatorio (*Random Forest*) sirven para aumentar el poder de predicción del modelo o para facilitar el entrenamiento del modelo. A continuación, se presentan los parámetros que se usan en Python para estos[29].

**n_estimadores**: Este es el número de árboles que quieres construir antes de tomar el máximo de votos o los promedios de las predicciones. Un **mayor número de**

[29] Documentación de la librería sklearn de Python. Fuente: https://scikit-learn.org/0.15/modules/generated/sklearn.ensemble.RandomForestClassifier.html

**árboles** da un **mejor rendimiento**, pero hace que tu código sea más lento. Se debe elegir el valor más alto que tu procesador pueda manejar porque esto hace que tus predicciones sean más fuertes y más estables.

**max_features**: Aumentar max_features generalmente mejora el rendimiento del modelo ya que en cada nodo se tendrá un mayor número de opciones a considerar. Sin embargo, esto no es necesariamente cierto ya que esto disminuye la diversidad de los árboles individuales del bosque aleatorio. además, se disminuye la velocidad del algoritmo al aumentar las max_features. Por lo tanto, se necesita encontrar el equilibrio correcto y elegir las máximas características óptimas.

Tipos características a considerar cuando se busca la mejor división:

- **log2**, entonces max_features=log2(n_features).
- **sqrt,** entoces max_features=sqrt(n_features).

## 5.9. MÉTRICAS DE RENDIMIENTO PARA LOS PROBLEMAS DE CLASIFICACIÓN EN ALGORITMOS DE MACHINE LEARNING.

Una vez implementado el modelo y obtenido el resultado en forma de una probabilidad o una clase, se procede determinar si el modelo realizado es óptimo de acuerdo al conjunto de datos de prueba.

Para ello se utilizan diferentes métricas de rendimiento para evaluar diferentes algoritmos de aprendizaje automático. A continuación, se mencionar las más importantes y las que fueron usadas para la comprobación de la predicción del Run life de las bombas PCP en el campo Casabe.

**5.9.1. MATRIZ DE CONFUSIÓN.** La matriz de confusión es una de las métricas más intuitivas y fáciles utilizadas para encontrar la precisión del modelo. Son

utilizadas cuando el problema a solucionar es de clasificación, en el que el resultado puede ser de dos o más tipos de clases.

En el campo de la inteligencia artificial y el aprendizaje automático una matriz de confusión es una herramienta que permite visualizar el desempeño de un algoritmo de aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real, es decir permite ver qué tipos de aciertos y errores está teniendo el modelo a la hora de pasar por el proceso de aprendizaje con los datos.

Para entender mejor ello, se propondrá un ejemplo básico en el cual se evaluará si una persona tiene cáncer o no. Donde 0 hace referencia a si la persona no tiene cáncer y 1 si lo tiene. Para esto se desarrollará una matriz de confusión, la cual es una tabla con dos dimensiones ("Dato Real" y "Dato Previsto"), y conjuntos de "clases" en ambas dimensiones. Las clasificaciones de "Dato Real" se representa mediante columnas y las de "Dato Previsto" en filas (Tabla 10).

Tabla 10. Matriz de confusión caso ejemplo

| | | **Datos Reales** | |
|---|---|---|---|
| | | Positivos (1) | Negativos (0) |
| **Datos Predichos** | Positivos (1) | **VP** | **FP** |
| | Negativos (0) | **FN** | **VN** |

Donde:

**Verdaderos positivos (VP):** Los verdaderos positivos son los casos en que la clase real del punto de datos era 1(Verdadero) y el predicho es también 1 (Verdadero), para el caso base significa que una persona tiene realmente cáncer (1) y el modelo que clasifica su caso como cáncer (1) se clasifica como Verdadero positivo.

**Verdaderos Negativos (VN):** Los verdaderos negativos son los casos en los que la clase real del punto de datos era 0 (Falso) y la predicción es también 0 (Falso), donde en el caso en el que una persona que NO tiene cáncer y en el cual el modelo que clasifica su caso como No cáncer se encuentra dentro de los Verdaderos Negativos.

**Falsos positivos (FP):** Los falsos positivos son los casos en los que la clase real del punto de datos era 0 (Falso) y la predicción es 1 (Verdadero). Falso es porque el modelo ha predicho incorrectamente y positivo porque la clase predicha fue positiva (1), donde para de ejemplo representa si una persona que NO tiene cáncer y el modelo que clasifica su caso como cáncer se clasificara como Falsos Positivos.

**Falsos negativos (FN):** Los falsos negativos son los casos en los que la clase real del punto de datos era 1 (Verdadero) y la predicción es 0 (Falso). Falso es porque el modelo ha predicho incorrectamente y negativo porque la clase predicha fue negativa. (0). Para el ejemplo representa a una persona con cáncer y el modelo que clasifica su caso como No-Cáncer si se encuentra bajo Falsos Negativos.

Un modelo se puede considerar optimo cuando se reduzca la cantidad de falsos positivos y falsos negativos, ya si estos tienen la mayor ocurrencia quiere decir que el modelo clasifica las cosas incorrectamente en comparación con la clase real.

**5.9.2. ACCURACY.** El accuracy en los problemas de clasificación corresponde a el número de predicciones correctas hechas por el modelo sobre todo tipo de predicciones hechas. Este es determinado mediante la ecuación 4.

Ecuación 4. Fórmula para la determinación de Accuracy.

$$Accuracy = \frac{VP + VN}{VP + FP + FN + VN}$$

Donde en el Numerador, se deben colocar las predicciones que fueron correctas (Verdaderos positivos y Verdaderos negativos y en el denominador, se colocan todas las predicciones hechas por el algoritmo (Tanto las correctas como las incorrectas).

**5.9.2.1. CUANDO USAR LA MÉTRICA ACCURACY.** El accuracy es una buena medida cuando las clases de variables objetivo en los datos están casi equilibradas. Por ejemplo, el 60% de las clases en nuestras imágenes de frutas son manzanas y el 40% son naranjas.

Por el contrario, esta métrica **NUNCA** debe ser usada como una medida cuando las clases de variables objetivo en los datos son la mayoría de una clase. Por ejemplo, cuando para detectar si una persona puede tener casos y se tiene datos de 100 personas que no tienen cáncer, mientras que sólo 5 personas tienen cáncer.

**5.9.3. PRECISIÓN.** La precisión es una medida que establece la proporción de los datos que fueron predichos y que al momento de verificarlos con los datos reales logran generar un ajuste. La métrica de predicción se calcula mediante la ecuación 5.

Ecuación 5. Fórmula para determinar la Precisión de un modelo.

$$\text{P}recisión = \frac{VP}{VP+FP}$$

Un ejemplo de esta métrica es cuando se tiene una muestra de cáncer con 100 personas y sólo 5 personas tienen cáncer. Este modelo es muy malo y predice cada caso como Cáncer. Ya que se está prediciendo que todos tienen cáncer, en el denominador (Verdaderos positivos y Falsos positivos) es 100 y el numerador, persona que tiene cáncer y el modelo que predice su caso como cáncer es 5. Así que, en este ejemplo, podemos decir que la precisión de dicho modelo es del 5%.

# 6. DISEÑO DE UN MODELO CAPAZ DE PREDECIR LAS FALLAS OPERACIONALES DE LAS BOMBAS PCP, EN EL CAMPO CASABE

En este capítulo se mencionará el proceso que se realizó para la construcción del modelo de predicción de las fallas de las bombas PCP para el campo Casabe. Se inicia con la estructuración del *Dataset* a usar en la predicción, de igual manera, se explica cómo se deben manejar las variables de entrada y salida del *Dataset.* Seguidamente se muestra el resultado de cada uno de los 6 algoritmos de inteligencia artificial mencionados en el capítulo 4 y, por último, se elige el mejor algoritmo para modelar el *Run Life* de las bombas PCP.

## 6.1. METODOLOGÍA DE APLICACIÓN

Para dar inicio a la construcción del modelo de predicción del Run life en el campo Casabe, se debió realizar una estructuración de cada uno de los pasos necesarios para abordar dicho problema, en la figura 43 se muestra un diagrama con los procesos realizados.

Figura 43.Flujograma de la metodología empleada para la construcción del modelo.

### 6.1.1. SELECCIÓN DEL SOFTWARE PYTHON PARA LA PREDICCIÓN DEL RUN LIFE EN EL CAMPO CASABE.

Para el desarrollo de este proyecto, se ha utilizado el software Python, este se selección debido a que es el más completo y versátil para trabajar con problemas de inteligencia artificial y Deep Learning. Aunque existen otros tipos de software como R que han tenido una gran acogida para trabajar inteligencia artificial, Python es el que mejor se posiciona dado que tiene en su base una gran cantidad de librerías exclusivas para trabajar con redes neuronales y algoritmos de inteligencia artificial que facilitan el trabajo y optimizan los procesos.

A continuación, se describirán las librerías que fueron utilizadas durante el proyecto y la funcionalidad de cada una de ellas.

#### 6.1.1.1. Librerías para algoritmos de inteligencia artificial en Python.

Principalmente se debe seleccionar una de las tres librerías con las que Python cuanta para el manejo de Deep Learning las cuales son: TensorFlow, PyTorch y Keras.

- **TensorFlow**. Es una de las plataformas de Deep Learning más importante más importante a nivel mundial desarrollada por Google cuenta con una gran flexibilidad y gran comunidad de desarrolladores lo ha posicionado como la herramienta líder en el sector del Deep Learning. Es una biblioteca de código abierto para computación numérica, que utiliza gráficos de flujo de datos. Los nodos en las gráficas representan operaciones matemáticas, mientras que los bordes de las gráficas representan las matrices de datos multidimensionales (tensores) comunicadas entre ellos.

- TensorFlow es una gran plataforma para construir y entrenar redes neuronales, que permiten detectar y descifrar patrones y correlaciones, análogos al aprendizaje y razonamiento usados por los humanos.
- **PyTorch** PyTorch es una biblioteca de inteligencia artificial de código abierto basada en la biblioteca Torch, utilizada para aplicaciones como la visión por computador y el procesamiento de lenguaje natural, desarrollada principalmente por el laboratorio de investigación de Facebook. Es un software libre y de código abierto publicado bajo la licencia BSD modificada. Aunque la interfaz de Python está más pulida y es el principal foco de desarrollo, PyTorch también tiene una interfaz C++.

- **Keras** Es un framework de alto nivel para el aprendizaje, escrito en Python capaz de trabajar sobre **TensorFlow**, **CNTK**, o **Theano**. Fue desarrollado con el objeto de facilitar un proceso de experimentación rápida. Esta librería es muy usada y conocida dado que permite trabajar muy bien en el desarrollo de redes neuronales y algoritmos de inteligencia artificial, a su vez el código que maneja permite que sea más entendible y es más fácil de manejar que TensorFlow, a su vez que permite obtener respuestas en menos tiempo y con una mayor claridad.

Cabe resaltar que para el caso específico de este proyecto se decidió hacer uso de la librería Keras para la creación de un modelo de red neuronal, dado que Keras presenta una mayor facilidad a la hora de trabajar, además que presenta resultados efectivos en el manejo de Deep Learning.

**6.1.1.2. Librería de Python para Machine Learning.** En Python existe la librería ***scikit-learn***, mediante la cual se pueden desarrollar algoritmos tanto categóricos como de regresión, en el capítulo cinco se explicó que para predecir las fallas en el

campo Casabe en el caso específico de este proyecto se utilizaran algoritmos de tipo categórico, los cuales fácilmente se pueden solucionar mediante el uso de esta librería.

Scikit-learn es una librería de Python para Machine Learning y Análisis de Datos. Está basada en NumPy, SciPy y Matplotlib. La ventaja principal de scikit-learn es la facilidad de uso y la gran cantidad de técnicas de aprendizaje automático que implementa. Es muy fácil de usar porque tiene una interfaz simple y muy consistente.

**6.1.1.3. Librerías de Python para calculo numérico y análisis de datos.** Mediante el uso de estas librerías se pueden realizar operaciones matemáticas de cualquier tipo, permiten en manejo de datos (Dataset), para poder ser ejecutados mediante algoritmos de inteligencia artificial, dentro de estas librerías se encuentran: NumPy, SciPy, Pandas.

- **Librería NumPy** Esta librería proporciona una estructura de datos matemáticos universales lo cual permite el análisis de datos y el intercambio de datos entre distintos algoritmos. Las estructuras de datos que implementa son vectores multidimensionales y matrices con capacidad para gran cantidad de datos. Además, esta librería proporciona funciones matemáticas de alto nivel que operan en estas estructuras de datos.

- **Librería SciPy** Proporciona rutinas numéricas eficientes fáciles de usar y opera en las mismas estructuras de datos proporcionadas por NumPy. Con SciPy se puede dar solución a problemas de integración numérica, optimización, interpolación, transformadas de Fourier, álgebra lineal, estadística, etc.

- **Librería Pandas** Pandas es una librería de Python destinada al análisis de datos, proporciona estructuras de datos flexibles y que permiten trabajar con ellos de forma muy eficiente. Pandas ofrece las siguientes estructuras de datos:

    - **Series:** Son arrays unidimensionales con indexación (arrays con índice), similar a los diccionarios. Pueden generarse a partir de diccionarios o listas.
    - **DataFrame:** son conjuntos de series, es decir estructuras de dos dimensiones.

Estas son las estructuras de datos más usadas en muchos campos tales como finanzas, estadística, ciencias sociales y muchas áreas de ingeniería. Pandas destaca por lo fácil y flexible que hace la manipulación de datos y el análisis de datos.

Para el manejo de los datos para la predicción de *Run Life* en el campo Casabe es necesario hacer uso de un *Dataset* en forma de *DataFrame*, dado que este tendrá dos dimensiones la primera consiste en los datos de entrada del modelo, estos corresponden a los parámetros operacionales que fueron seleccionados en el capítulo cuatro como los más representativos para dar solución a este problema y la segunda dimensión hace referencia a los datos de salida del modelo, los cuales consisten en las clases o categorías que se establecieron representados en cuatrimestres que pueden tomar los conjuntos de datos.

**6.1.1.4. Librerías de Python para visualización** Visualizar un problema mediante graficas siempre ha permitido entender el comportamiento de los resultados que involucran problemas complejos, tales como procesos que involucran la inteligencia artificial, es por ello que Python cuenta con tres librerías para el

manejo de datos mediante gráficos, entre ellos se cuenta con: Matplotlib, Seabon y Bokeh. Para este proyecto se hizo uso de la librería Matplotlib ya que es la más completa.

- **Matplotlib** Es la librería gráfica de Python estándar y la más conocida. Se puedes usar Matplotlib para generar gráficos de calidad necesaria para publicarlas tanto en papel como digitalmente. Con Matplotlib se pueden crear diversos tipos de gráficos entre ellos se encuentran las series temporales, histogramas, espectros de potencia, diagramas de barras, diagramas de errores, etc.

Por otro lado, PYTHON también cuenta con diversos entornos de desarrollo que permiten un uso práctico para desarrollar modelos, para este proyecto se usaron dos de ellos: Spyder, el cual se obtiene del servidor anaconda, trabajando como aplicación en un ordenador y, Google Colaboratory, proporcionado por Google para poder trabajar online y guardar directamente en la nube toda la información que se introduce.

**6.1.2. CREACIÓN DEL DATASET PARA LA PREDICCIÓN DE FALLAS DE LAS BOMBAS PCP.** La base fundamental de este proyecto de grado, una vez identificados los seis tipos de algoritmos de inteligencia artificial a evaluar, consistió en la creación de un *Dataset* que permitiera realizar tanto el entrenamiento de los modelos como realizar la prueba de los mismos y de esta manera poder dar una predicción del *Run life* de las bombas PCP en el campo Casabe.

Este *Dataset* debe tener, tanto las variables de entrada (Parámetros para la predicción), como las de salida del modelo (Run life de las bombas PCP, establecido en 4 clases). Quizá, se preguntará, por qué es necesario los datos de salida si es lo que se quiere predecir. Respuesta a ello, es que todos los algoritmos de inteligencia artificial se basan en el aprendizaje de datos, es decir, ellos analizan los datos de

entrada del modelo y los van asociando a una salida conocida. Entre más características similares se presenten con la misma respuesta, permitirá que luego, cuando solo se ingresen características de entrada, automáticamente dará la salida que le corresponda según el previo entrenamiento que tuvo. Esta es una de las razones por las cuales entre más robusto sea el *dataset* mejor será la predicción del modelo.

Para que la predicción sea confiable, se necesita que los datos de entrada en el modelo sean los más adecuados, razón por la cual, en el capítulo 3 se mencionaron los parámetros tanto operacionales como de yacimiento que se consideran pertinentes para determinar el *Run Life* de las bombas. Estos serán utilizados en la creación del *dataset* y serán las variables de entrada en el modelo de predicción.

Se destacan que son 8 las características de entrada en el modelo (Cuadro 1). Además de ello el *dataset* se compone de 236 datos de fallas en bombas PCP del campo Casabe. A cada vector de características, en otras palabras, a cada fila de datos le corresponderá un valor del *Run Life* que tuvo la bomba antes de fallar. Este estará medido en cuatrimestres ya que permite un mejor manejo de los datos y se puede dar solución al problema como un caso categórico y no de regresión, ya que, si se considera el hecho de dar solución al problema por regresión, el *Run Life* debería estar medido en días, y debido a que se tienen datos muy dispersos no es conveniente dar solución a este problema de manera regresiva.

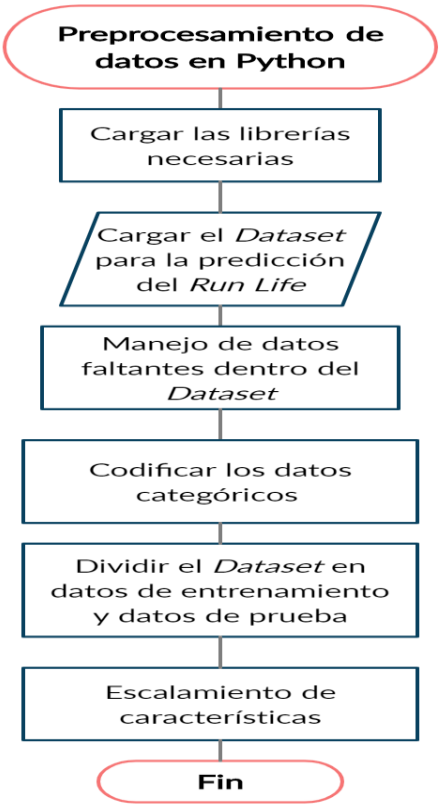
Cuadro 1. Características contenidas en el dataset para la predicción de fallas en el Campo Casabe.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Max desviación del pozo | Profundidad de la bomba | Torque | Revoluciones por minuto (RPM) | Nivel de fluido sobre la bomba | %BSW | Caudal | THP |

## 6.2. PREPROCESAMIENTO DE LOS DATOS PARA LA PREDICCIÓN DEL *RUN LIFE*.

Para el diseño del modelo de predicción de fallas para el campo Casabe o cualquier desarrollo que se realice en *Machine Learning* o Deep Learning se necesita tener un preprocesamiento de dato, este es uno de los pasos más importantes a la hora de realizar modelos de inteligencia artificial mediante aprendizaje automático, ya que los datos que se ingresen representaran la base del desarrollo. Este preprocesamiento se debe realizar de la manera correcta para que el entrenamiento sea el adecuado. La figura 44 muestra la correcta forma de hacer un procesamiento de datos para cualquier Dataset para un problema que tenga datos categóricos.

Figura 44.Flujograma para el procesamiento de los datos para la predicción



**Primer paso:** Importar las librerías básicas para realizar el preprocesamiento de datos las cuales son Numpy, Pandas, Matplotlib.

**Segundo paso:** Cargar el Dataset con los parámetros de entrada y salida desarrollados para la predicción de fallas en bombas PCP en el campo Casabe. Cabe resaltar que para que Python pueda entender los datos estos deben estar en un formato de tipo .CSV.

Para importar el *Dataset* se hace uso de la librería Pandas, lo primero que se debe hacer es crear una variable con el nombre que se le desea colocar a los datos.

Una vez cargado el dataset se procede a crear dos identidades, la primera hace referencia a la matriz de características, la cual corresponde a los parámetros que fueron seleccionados para predecir el Run life de las bombas PCP en el campo Casabe y la segunda identidad corresponde al vector que forma la variable dependiente, que para el caso específico de este proyecto corresponde al Run life de las bombas por cuatrimestres. El cuadro 2 representa la división de los datos.

Cuadro 2. Datos de entrada y de salida para la creación de los algoritmos de predicción**.**

| X | | | | | | | | Y |
|---|---|---|---|---|---|---|---|---|
| **Max desviación del pozo** | **Profundidad de la bomba** | **Torque** | **Revoluciones por minuto (RPM)** | **Nivel de fluido sobre la bomba (ft)** | **%BSW** | **Caudal (bbl/d)** | **THP** | **Run Life (cuatrimestre)** |
| 1.44 | 3700 | 0.55 | 185 | 575 | 35 | | 160 | 1 |
| 1.44 | 3700 | 0.45 | 205 | 525 | 100 | 205 | 120 | 4 |
| 2.14 | 2900 | 0.33 | 125 | 675 | 85 | 45 | 80 | 1 |
| 2.2 | 3900 | 0.45 | 165 | 1025 | 85 | 65 | 135 | 4 |
| 2.68 | 2900 | 0.35 | 175 | 1125 | 85 | 290 | 100 | 2 |
| 2.68 | 2900 | 0.25 | 125 | 1025 | 85 | 195 | 120 | 1 |
| 2.58 | 2300 | 0.15 | 125 | 1675 | 85 | 75 | 100 | 2 |

**Tercer paso:** Manejo de datos faltantes dentro del *Dataset,* este paso se suele utilizar cuando dentro del *Dataset* ingresado alguno de los parámetros dentro de un conjunto de datos o array se encuentre vacío, ya sea porque las muestras no pudieron ser tomadas o simplemente no se reportaron.

Lo primero que se reviso es si el Dataset para la predicción del *Run life* presentaba datos faltantes.

De lo cual se obtuvo lo siguiente:

```
Datos faltantes:
Max Dogleg                16
PUMP Depth                0
Torque                    0
RPM                       0
fluido sobre la bomba     0
BSW                       0
Caudal                    2
THP                       1
RL                        0
```

Lo que quiere decir que el Dataset no cuenta con 16 datos de máximo *Dogleg*, 2 datos del caudal y un dato en el THP de la bomba, esto teniendo en cuenta que el conjunto total de datos corresponde a 236 datos.

Cuando existen datos que faltan dentro de un Dataset se pueden generar errores al momento de entrenar un modelo, por tanto, es necesario que estos datos se puedan rellenar con la información óptima para efectuar el proceso satisfactoriamente. Una de las formas de solucionar este problema es simplemente eliminado los conjuntos que tengan datos vacíos, esto se puede hacer solo si el *Dataset* que se está manejando cuenta con un gran número de datos y la información eliminada solo representa el 1% de los datos totales. Este no es el caso del Dataset que se tiene para lo cual existe una segunda forma y es rellenando los datos faltantes con la media de los datos que conforman una misma característica o parámetro.

Luego de establecer que los datos faltantes tomen el promedio de los datos se vuelven a imprimir si existen datos faltantes de lo que se obtiene que todos los datos han quedado rellanados satisfactoriamente.

**Cuarto paso:** Codificación de los datos categóricos, esta parte consiste en si se tienen datos de tipo *string* en un dataset, estos se deben codificar para que se pueda manejar todo el *Dataset* con un solo tipo de datos.

**Quito paso:** División del Dataset en datos de entrenamiento y prueba, esta parte es fundamental al momento de tratar con problemas de inteligencia artificial, para ello lo que se debe hacer es subdividir el Dataset en dos uno que contendrá los datos que serán entrenados en cada uno de los modelos y otro con los datos que serán probados una vez el entrenamiento se realice, con el fin de determinar si el modelo efectuado permite realizar una predicción aceptable (Figura 45).

Lo primero que se debe realizar es que al Dataset se le debe realizar una organización aleatoria de los datos, esto se realiza con la finalidad de que si existían datos con la misma categoría organizados uno tras otro al momento de dividir el dataset cada una de las clases logren ser distribuidas, esto permite que se disminuya el error en las predicciones.

Posteriormente se necesita saber el porcentaje de los datos que serán tomados para el entrenamiento y cual, para la prueba, la mejor división que es reportada por los expertos de Machine Learning es considerar el 80% de los datos para entrenar los modelos y el 20% para que puedan ser probados, esta configuración permite que no se genere overfitting en los modelos, es decir que esto no vayan a presentar un sobreajuste o se presente un sobre-entrenamiento que no es recomendable (Figura 46) “Ibid”.p [30]

[30] Ibid.., p.65.

Figura 45.Código para crear los datos de entrada y salida en el modelo en Google Colaboratory.

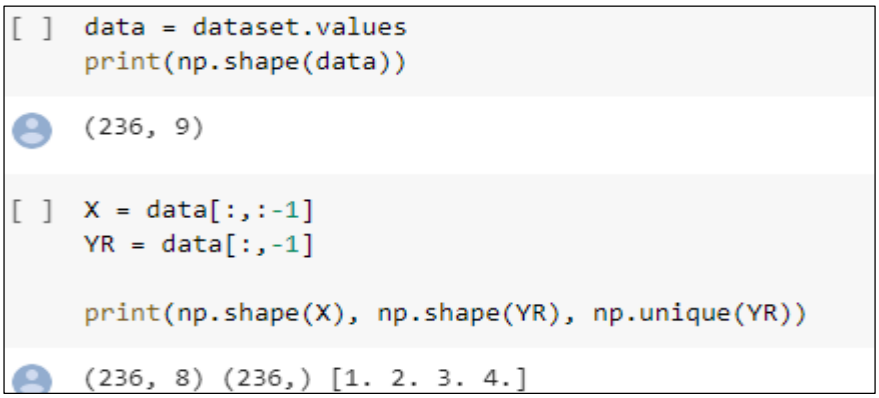
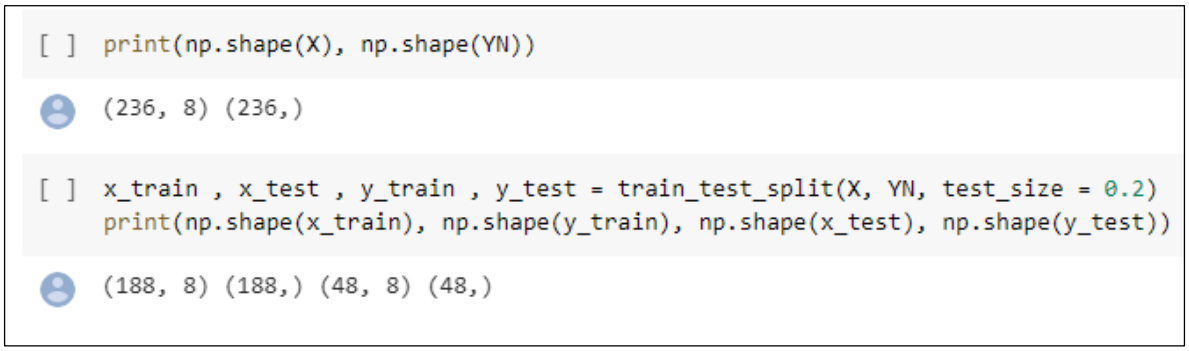



Figura 46.Código de la división de los datos para el entrenamiento y prueba.




**Sexto paso:** Escalamiento de las características, esta parte consiste en que los datos puedan quedar en una misma escala con la finalidad de que no existan características que estén presentando un mayor dominio que otras y evitar que las características no dominantes no sean consideradas por el algoritmo de inteligencia artificial.

Para el escalado de características existen dos formas de hacerlo, uno es mediante la estandarización y el otro es normalizando los datos (Tabla 11).

Tabla 11. Tipos de sistema para escalar los datos dentro de un *Dataset.*

| ESTANDARIZACIÓN | NORMALIZACIÓN |
|---|---|
| $X_{stand} = \frac{x - mean(x)}{desviación\ estandar(x)}$ | $X_{norm} = \frac{x - min(x)}{max(x) - \min(x)}$ |

La escogencia de estos dos tipos de datos se presenta respecto a la distribución que presentan los datos, cuando la distribución es de tipo normal se utiliza únicamente el método de normalización, mientras que el método de estandarización puede trabajar tanto si los datos presentan un tipo de distribución como cuando no la presentan.

Cuadro 3. Datos de entrada y de salida para la creación de los algoritmos de predicción**.**

| X | | | | | | | | Y |
|---|---|---|---|---|---|---|---|---|
| Max desviación del pozo | Profundidad de la bomba | Torque | Revoluciones por minuto (RPM) | Nivel de fluido sobre la bomba (ft) | %BSW | Caudal (bbl/d) | THP | Run Life (cuatrimestre) |
| 1.44 | 3700 | 0.55 | 185 | 575 | 35 | | 160 | 1 |
| 1.44 | 3700 | 0.45 | 205 | 525 | 100 | 205 | 120 | 4 |
| 2.14 | 2900 | 0.33 | 125 | 675 | 85 | 45 | 80 | 1 |
| 2.2 | 3900 | 0.45 | 165 | 1025 | 85 | 65 | 135 | 4 |
| 2.68 | 2900 | 0.35 | 175 | 1125 | 85 | 290 | 100 | 2 |
| 2.68 | 2900 | 0.25 | 125 | 1025 | 85 | 195 | 120 | 1 |
| 2.58 | 2300 | 0.15 | 125 | 1675 | 85 | 75 | 100 | 2 |

## 6.3. CREACIÓN DE LOS MODELOS DE PREDICCIÓN PARA PREDECIR LAS FALLAS EN EL CAMPO CASABE

Una vez que el Dataset ha tenido el correcto tratado de datos mediante la etapa del preprocesamiento, se empieza a ser el diseño de los algoritmos para la predicción del *Run life* en las bombas PCP del campo Casabe. Para este proyecto como se mencionó en el capítulo 5 se probarán 6 modelos uno será mediante aplicación de Deep Learning con el uso de redes neuronales y los otros corresponden a algoritmos de Machine Learning, donde se creará un algoritmo para K- vecinos más cercanos, Maquina de soporte vectorial (SVM), Naive Bayes Gaussian, árboles de decisión (*Decision tree*) y Bosques aleatorios (*Random Forest*).

**6.3.1. Manejo de las clases del Dataset para la predicción del Run life.** Antes de realizar cada uno de los algoritmos, se procedió a hacer un manejo de datos de las clases de *Run life,* con el fin de que los modelos no generen problemas debido a estas durante la ejecución. Como se comentó en la sección de la construcción del *Dataset,* para este proyecto el Run life consta de cuatro clases las cuales se clasificaron como: Falla en el primer cuatrimestre (1), Falla en segundo cuatrimestre (2), Falla en el tercer cuatrimestre (3), Falla en el cuarto cuatrimestre (4).

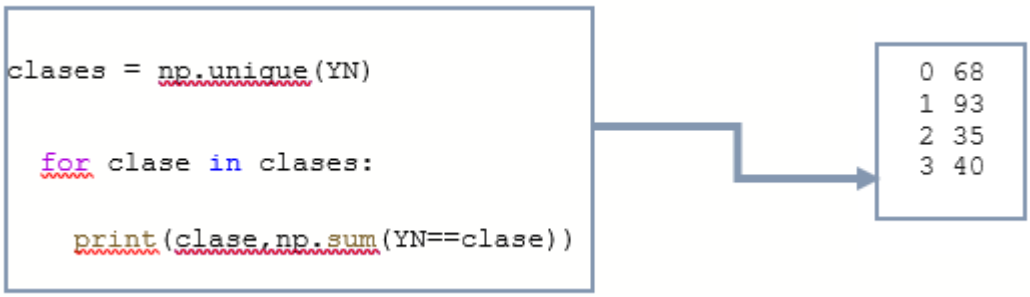
Se decidió iniciar las clases desde 0, es decir que para el primer cuatrimestre el algoritmo lo reconocerá como la clase 0, el segundo cuatrimestre como la clase 1 y así sucesivamente, esto se realizó con la finalidad de que el algoritmo no le diera mayor prioridad a una clase en particular (Figura 47).

Figura 47.Código para la inicialización de clases.

Seguidamente es necesario determinar cómo están distribuidas cada una de las clases del Dataset, es decir cuántos datos se tienen de cada una de estas, este paso es muy importante al momento de poder realizar los modelos y validarlos, ya que si existen diferencias considerables de una clase respecto a otra puede recurrir a posibles errores y por tanto es posible que el algoritmo este prediciendo de forma incorrecta.  Una muy buena distribución de las clases es que todos sus elementos estén equilibrados es decir que tengan la misma cantidad de datos, aunque también se considera un buen equilibrio si estas están manteniendo una proximidad de una clase a la otra en cuanto a la cantidad de datos, por ejemplo, cuando una clase representa el 60% y el otro el 40%.

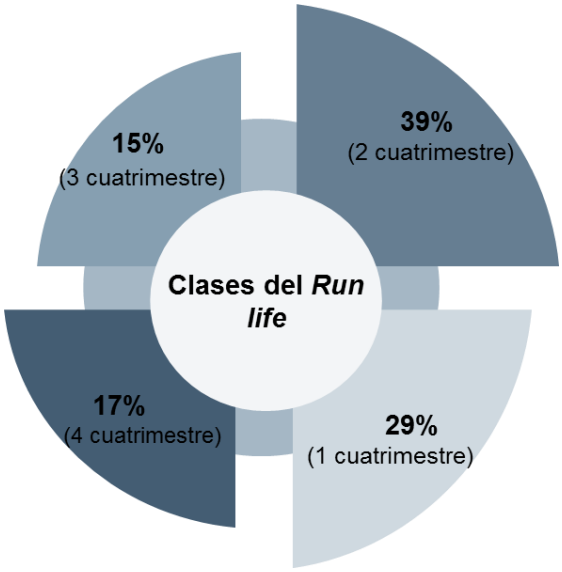
Para el Dataset de predicción del presente proyecto se cuenta con la siguiente distribución de las clases (Figura 48).

Figura 48.Código para determinar la cantidad de datos de cada clase.




La que quiere decir que las clases del modelo no se encuentran equilibradas y esto como se mencionó anteriormente puede generar errores en la predicción de los modelos, La figura 49 muestra el porcentaje de cada una de las clases.

Figura 49.Distribución en porcentaje de cada una de las clases del Dataset.




Ahora se procederán a balancear las clases del Dataset, para poder hacer los modelos correctamente (Figura 50).

Figura 50.Código para balancear cada una de las clases.

```
def balancear (X, y):

    import numpy as np

    datos = []
    lbl = []
    X = np.array(X)

    class = np.unique(y)
    menor = 999999999
    for i in range(len(clases)):
      if np.sum(np.array(y)==i)<= menor:
        menor = np.sum(np.array(y)==i)
    #print(menor)
    for clase in class:
      indice = 0
      contador = 0
      while contador < menor:

        if y[indice] == clase:

          datos.append(X[indice,:])
          lbl.append(y[indice])
          contador += 1
        indice += 1


    return datos, lbl
```

Una vez se han se han balanceado las clases, se proceden a cargar las librerías necesarias para realizar los modelos (Figura 51).

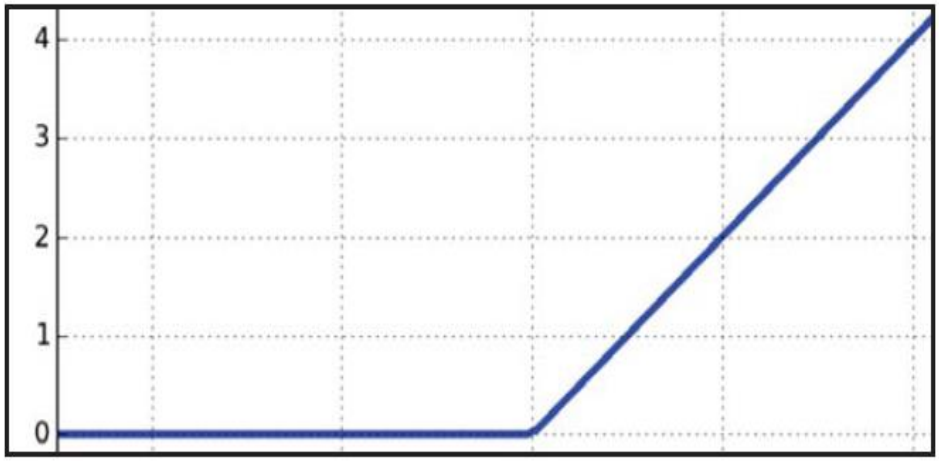
Figura 51.Código para importar las librerías para la construcción de los algoritmos.

```
from sklearn.ensemble           import RandomForestClassifier
from sklearn.svm                import SVC
from sklearn.neighbors          import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes      import GaussianNB
```

**6.3.2. Construcción del modelo de Redes Neuronales Artificiales (RNA).** El primer modelo que se decidió probar fue una red neuronal la cual tiene las siguientes características, presenta una capa de entrada, 10 capas ocultas y una capa de salida, el sistema de activación es ReLu, la cual se explica a continuación.

**Función de activación ReLU:** La función unidad lineal rectificad (ReLU, por sus siglas en ingles), es una función de activación muy popular en el uso de redes neuronales artificiales, porque esta genera muy buenos resultados experimentales. Esta función se define por la función F(x)= Max (0, x), es una función no lineal, esta toma valor de cero cuando los datos son negativos y tiene un comportamiento ascendente cuando estos son positivos, la figura 52, representa gráficamente esta función de activación.

Figura 52.Función de activación unidad lineal rectificada (ReLU).




Fuente: Gulli. A y Sujit. P. (2017)[31]

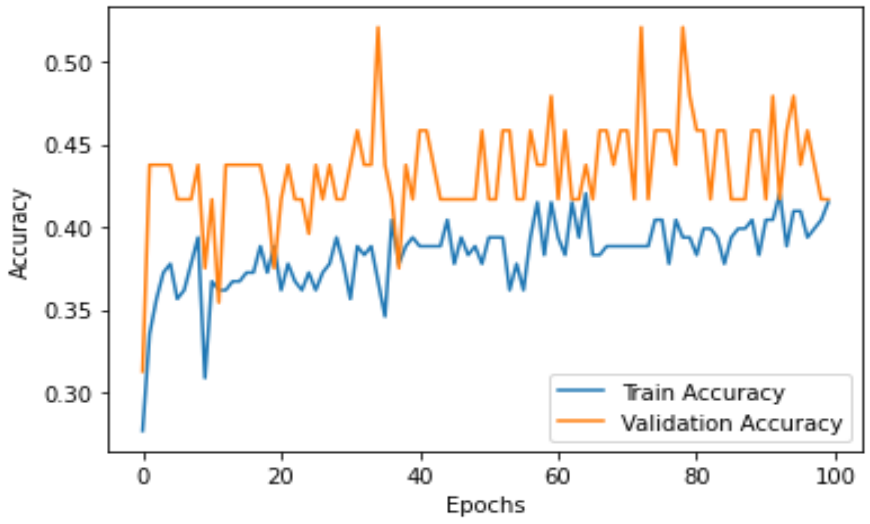
El desarrollo de esta red neuronal de detalla en la figura 53. Con este diseño, se obtuvo una predicción del **41,6%** (Figura 54), esto se debe a que las redes neuronales a pesar de ser modelos muy eficientes para la predicción de datos necesitan que se cuente con bastantes datos para que se pueda considerar viable su uso.

[31] Birmingham-Mumbai.Gulli. A y Sujit. P. (2017) Deep learning with Keras, implement neural networks with keras on Theano and TensorFlow. Birmingham-Mumbai.

Figura 53.Código diseñado para la creación de una red neuronal artificial.

```
model = keras.models.Sequential([

    keras.layers.Dense(dim[1], input_shape=[dim[1]]),
    keras.layers.Dense(10, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.relu),
    keras.layers.Dense(20, activation=tf.nn.relu),
    keras.layers.Dense(20, activation=tf.nn.relu),
    keras.layers.Dense(30, activation=tf.nn.relu),
    keras.layers.Dense(30, activation=tf.nn.relu),
    keras.layers.Dense(50, activation=tf.nn.relu),
    keras.layers.Dense(50, activation=tf.nn.relu),
    keras.layers.Dense(4, activation=tf.nn.softmax)])
opt = keras.optimizers.SGD(lr=0.001, momentum=0.1)
test_loss, test_acc = model.evaluate(x_test, y_test, verbose=0)
print('Test accuracy:', test_acc)

Test accuracy: 0.4166666567325592
```

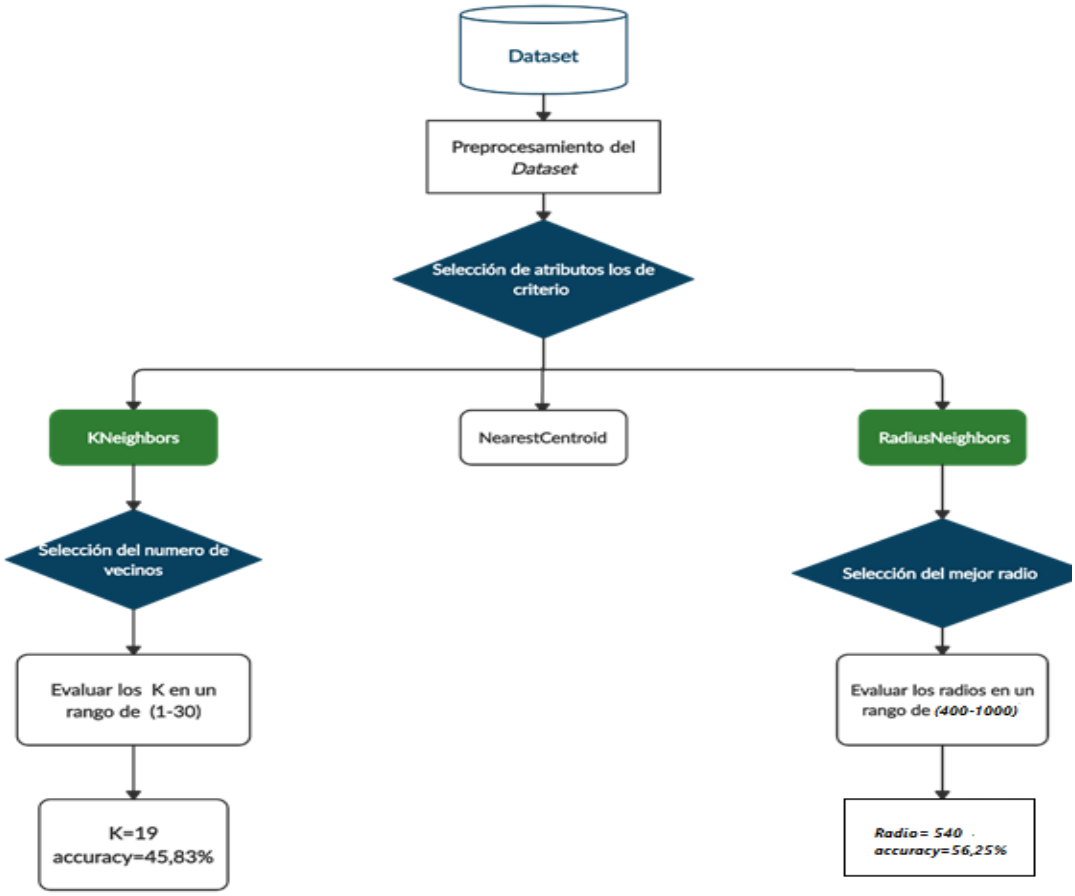
Figura 54.Validación de la red neuronal. La línea de color amarillo indica en valor de la predicción del 41,6% obtenido en el modelo.




**6.3.3. Construcción del modelo de K-Vecinos más Cercanos (K-NN).** Para la construcción del modelo de K-NN se determinó que se estudiaran dos de los tres

atributos (*Descritos en el capítulo 5*) para la construcción de este algoritmo los cuales corresponden a KNeighborsClassifier y RadiusNeighborsClassifier se descartó la prueba con el atributo NearestCentroid, ya que este solo permite generar un buen modelo cuando el problema es de tipo de regresión lo cual no funciona para el desarrollo de este proyecto dado que se tiene un problema de clasificación en la figura 55 se muestra el flujograma para la creación de los algoritmos de K-NN.

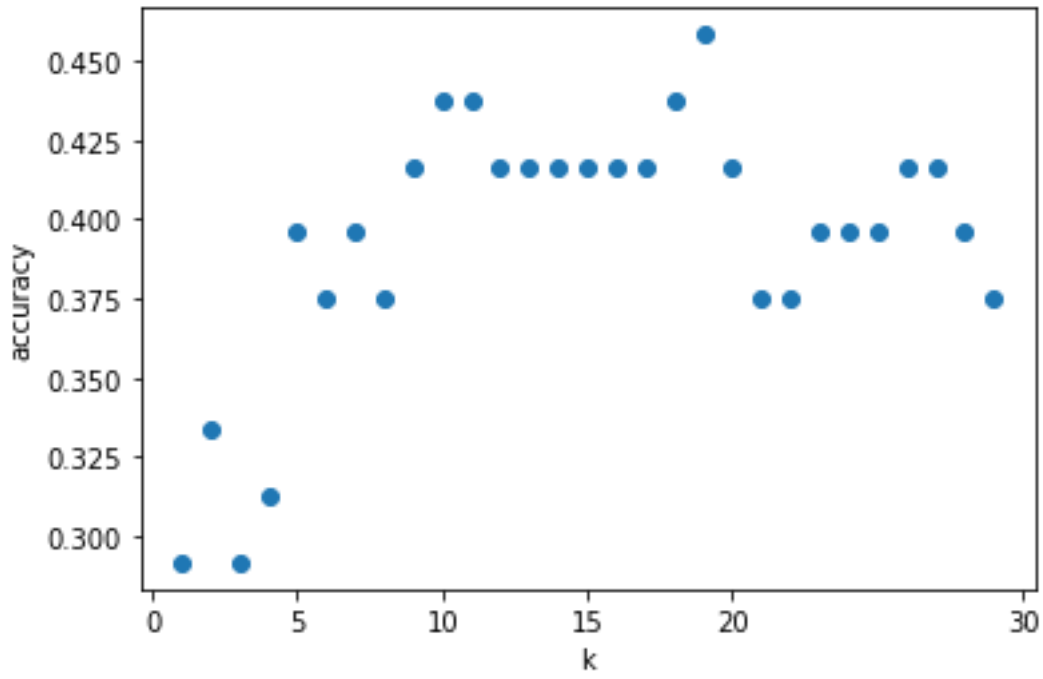
Figura 55. Flujograma del desarrollo del modelo de K-NN




Primentamente se realiza la construcción del algoritmo utilizando **KNeighborsClassifier**, para este algoritmo se requiere saber el valor que se le asignara a la variable K, la cual hace referencia a la cantidad de datos más cercanos

con los cuales se va a evaluar el modelo, para ello se decidieron evaluar variar la K en 30 algoritmos los cuales se evaluaron en un rango de uno a treinta y luego se imprimían los datos respecto al resultado obtenido del accuracy tal y como se observa en la figura 56.

Figura 56. Variación de los K del modelo en un rango de uno a treinta respecto al accuracy.




De esta grafica se puede observar que la selección del mejor número de vecinos más cercanos corresponde a **K=19,** puesto que es la que logra el mejor acurracy, entonces se procede a conocer con exactitud el valor de este (Figura 57).

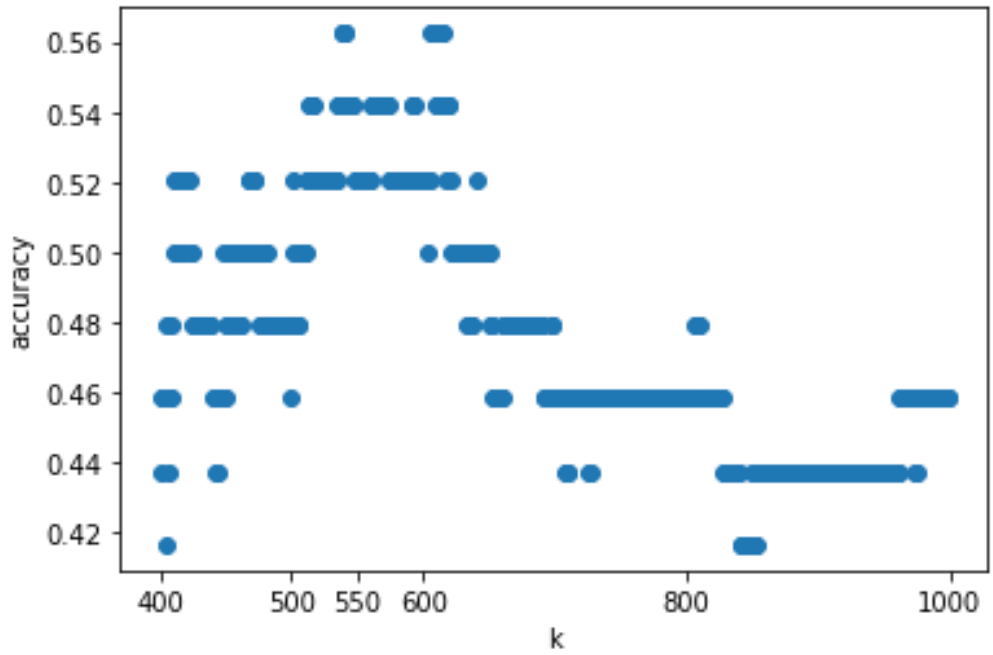
Figura 57.Código del algoritmo de K-NN con un K=19.

```
modelo = KNeighborsClassifier(n_neighbors = 19)
modelo.fit(x_train,y_train)
print(modelo.score(x_test,y_test))

y_predict = modelo.predict(X)
0.4583333333333333
```

De acá se obtuvo que la predicción del modelo utilizando el atributo **KNeighborsClassifier** con un **K=19** equivale al **45,83%** de predicción del Run life de las bombas PCP en el campo Casabe.

Luego de ello se procede a crear el algoritmo evaluando el atributo **RadiusNeighborsClassifier** donde se probó variar el radio desde 400 a 1000, de acá se obtuvo lo siguiente (Figura 58).

Figura 58.Variación de los radios del modelo en un rango de 400 a 1000 respecto al accuracy.



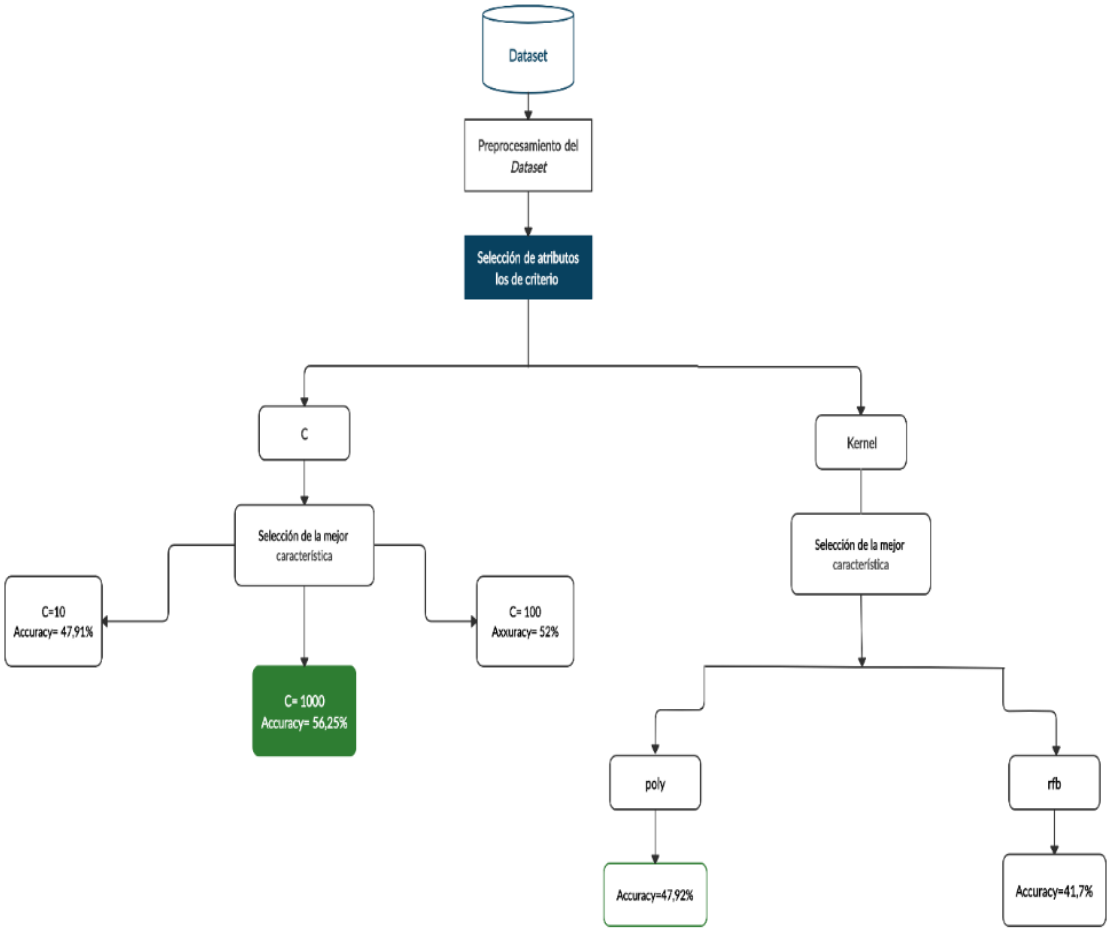
De acá se obtiene que el mejor radio para la predicción el cual corresponde a **Radio=540** obteniendo un accuracy de **56,25%** (Figura 59)**.**

Figura 59.Código del algoritmo de K-NN con un radio = 540.

```
 modelo = RadiusNeighborsClassifier(radius= 540)
modelo.fit(x_train,y_train)
print(modelo.score(x_test,y_test))
y_predict = modelo.predict(X)
0.5625
```

**6.3.4. Construcción del modelo de Maquina de Soporte Vectorial (SVM).** Para la construcción de este modelo se probaron realizaron diversos algoritmos probando los atributos C y Kernel (Mencionados en el capítulo 5), la figura 60 muestra el diagrama de flujo que se realizó para la construcción del modelo.

Figura 60.Flujograma del desarrollo del modelo de SVM.




Primeramente, se inició realizando la construcción del modelo de Maquina de Soporte vectorial probando el atributo C variando el parámetro en 10, 100, 1000 y 10000 (Figura 61).

Figura 61.Código del para la construcción de los algoritmos de SVM variando el atributo C.

**C= 10**

```
modelo = SVC(C= 10)
s = cross_val_score(modeloDeForest, x_train, y_train, cv=KFold(5, shuf
fle=True), scoring=make_scorer(accuracy_score))
print('Precision en la fase de train:',np.mean(s), np.std(s))
modelo.fit(x_train, y_train)
modelo.score(x_test,y_test)

Precision en la fase de train: 0.38321479374110956 0.06899576166702302
0.4791666666666667
```

**C= 100**

```
modelo = SVC(C= 100 , kernel= 'poly')
s = cross_val_score(modeloDeForest, x_train, y_train, cv=KFold(5, shuf
fle=True), scoring=make_scorer(accuracy_score))
print('Precision en la fase de train:',np.mean(s), np.std(s))
modelo.fit(x_train, y_train)
modelo.score(x_test,y_test)

Precision en la fase de train: 0.37652916073968706 0.10078173389180409
0.5208333333333334
```

**C=1000**

```
modelo = SVC(C= 1000)
s = cross_val_score(modeloDeForest, x_train, y_train, cv=KFold(5, shuf
fle=True), scoring=make_scorer(accuracy_score))
print('Precision en la fase de train:',np.mean(s), np.std(s))
modelo.fit(x_train, y_train)
modelo.score(x_test,y_test)
Precision en la fase de train: 0.3672830725462305 0.06668753850825734
0.5625
```

De aca se obtiene que el mejor modelo variando el atributo C, se genera cuando este toma en valor de **C=1000**, obteniendo un Accuracy de **56,25%**.

Porteriormente se procede a probar el atributo Kerner variando los parametros usando kernel polinomico (poly) y kernel en función de base radial (rbg) (Figura 62).

Figura 62. Código del para la construcción de los algoritmos de SVM variando el atributo Kernel

**Kernel = poly**

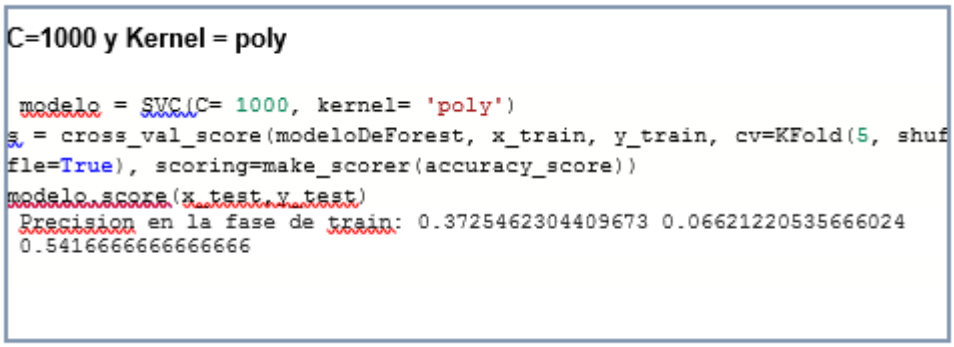
```
modelo = SVC(kernel='poly')
s = cross_val_score(modelo, x_train, y_train, cv=KFold(5, shuffle=True
), scoring=make_scorer(accuracy_score))
print('Precision en la fase de train:',np.mean(s), np.std(s))
modelo.fit(x_train, y_train)
modelo.score(x_test,y_test)

Precision en la fase de train: 0.38250355618776677 0.04560447205579259
0.4791666666666667
```

**Kernel= rfb**

```
modelo = SVC(kernel='rbf')

s = cross_val_score(modeloDeForest, x_train, y_train, cv=KFold(5, shuf
fle=True), scoring=make_scorer(accuracy_score))

print('Precision en la fase de train:',np.mean(s), np.std(s))

modelo.fit(x_train, y_train)

modelo.score(x_test,y_test)
Precision en la fase de train: 0.3721194879089616 0.07144366177222147
0.4166666666666667
```

De acá se obtuvo que de los dos kernel que se probaron el que obtenía un mejor resultado era el de Kernel = poly obteniendo un accuracy de 47,92%, posteriormente se opta por probar un algoritmo configurando los dos atributos que dieron el mejor resultado los cuales fueron C=1000 y Kernel = poly (Figura 63).

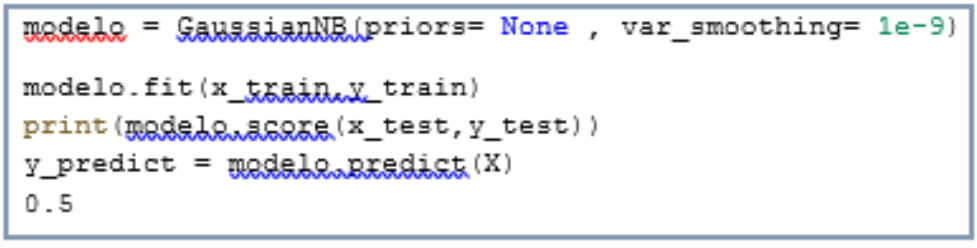
Figura 63.Código del para la construcción de los algoritmos de SVM con C=1000 y Kernel= poly




Lo que permite observar que es mejor utilizar el atributo C=1000 ya que fue el que tuvo el mejor accuracy de los modelos probados para el algoritmo de Maquina de Soporte Vectorial con un accuracy de **56,25%**.

**6.3.5. Construcción del modelo de Naive Bayes Gaussian.** Para la construcción de este modelo se decidiero hacer un algoritmo usando los atributos por default de var_smoothing y Priors, con lo cual se pudo obtener un accuracy de 50% (Figura 64).

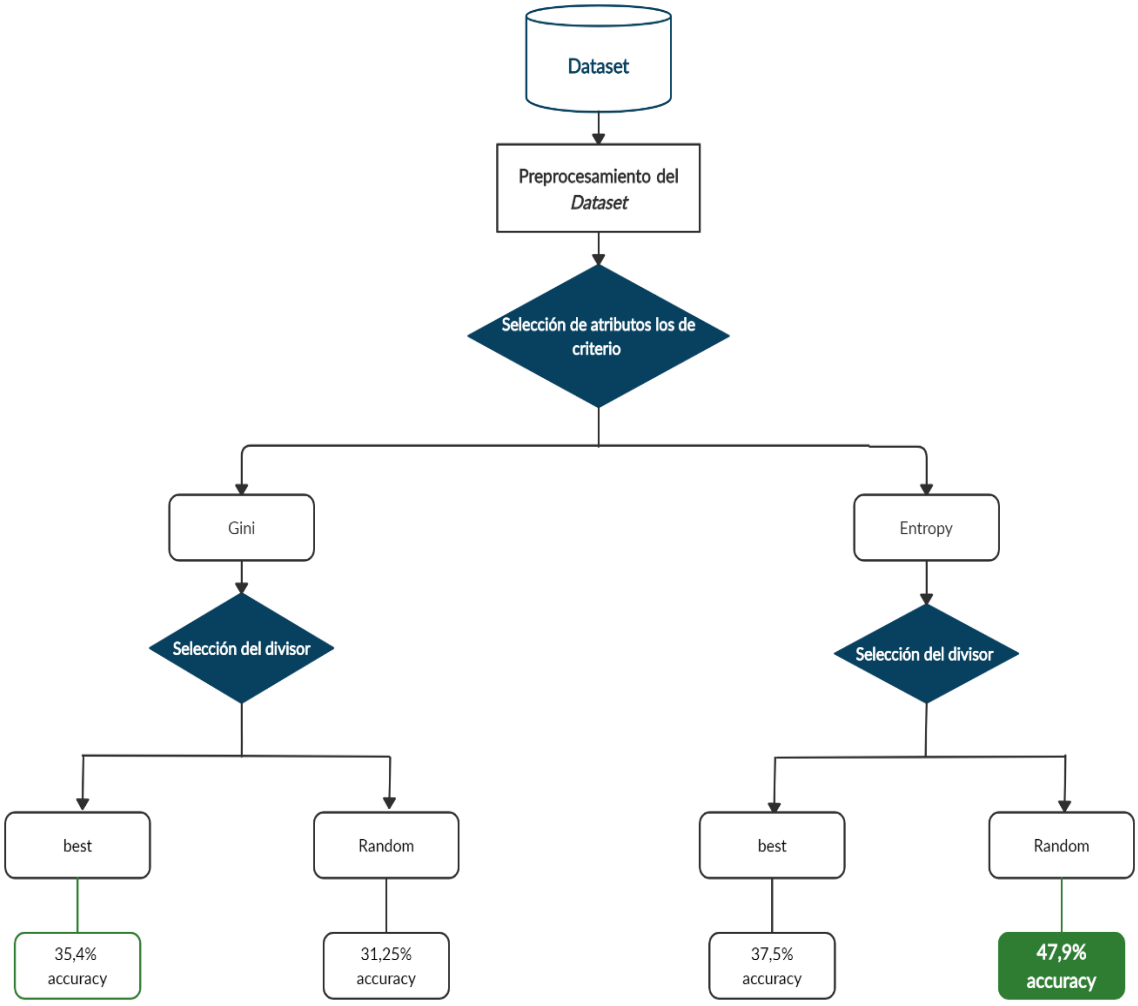
Figura 64.Código del para la construcción de los algoritmos de Naive Bayes Gaussian




**6.3.6. Construcción del modelo de árboles de decisión (Decision tree).** Para la construcción del modelo de árboles de decisión se probaron 4 configuraciones teniendo variaciones entre los dos tipos de criterios que maneja el algoritmo el cual

es gini y entropy y entre los dos tipos de divisores los cuales son best y random, la figura 65 muestra cómo se seleccionó el modelo.

Figura 65.Flujograma del desarrollo del modelo de decision tree.



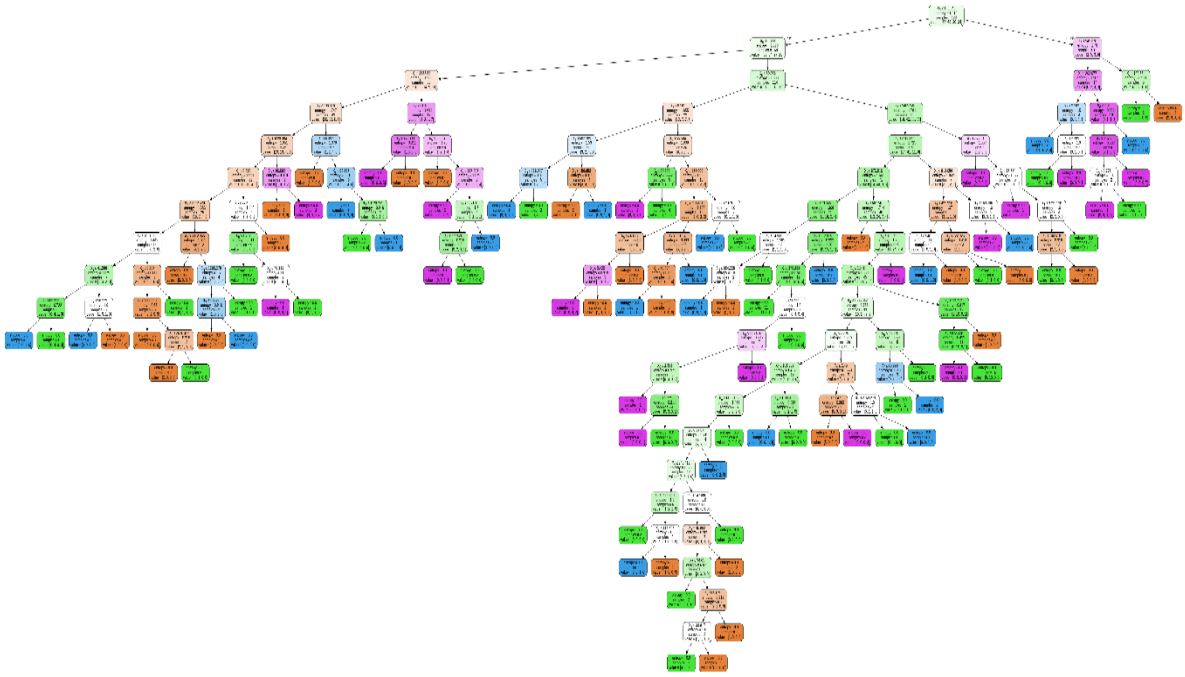

Para este algoritmo se pudo determinar que, de las cuatro configuraciones realizadas (Figura 66), la que estaba presentando un mayor accuracy se da mediante la escogencia del criterio **entropy** junto con el divisor **random**, el cual obtuvo un resultado de 47,9%, posteriormente a el modelo seleccionado se le

extraer el esquema la estructura de los arboles realizadas, para visualizar el modelo (Figura 67).

Figura 66.Código del para la construcción de los algoritmos de Decision tree.

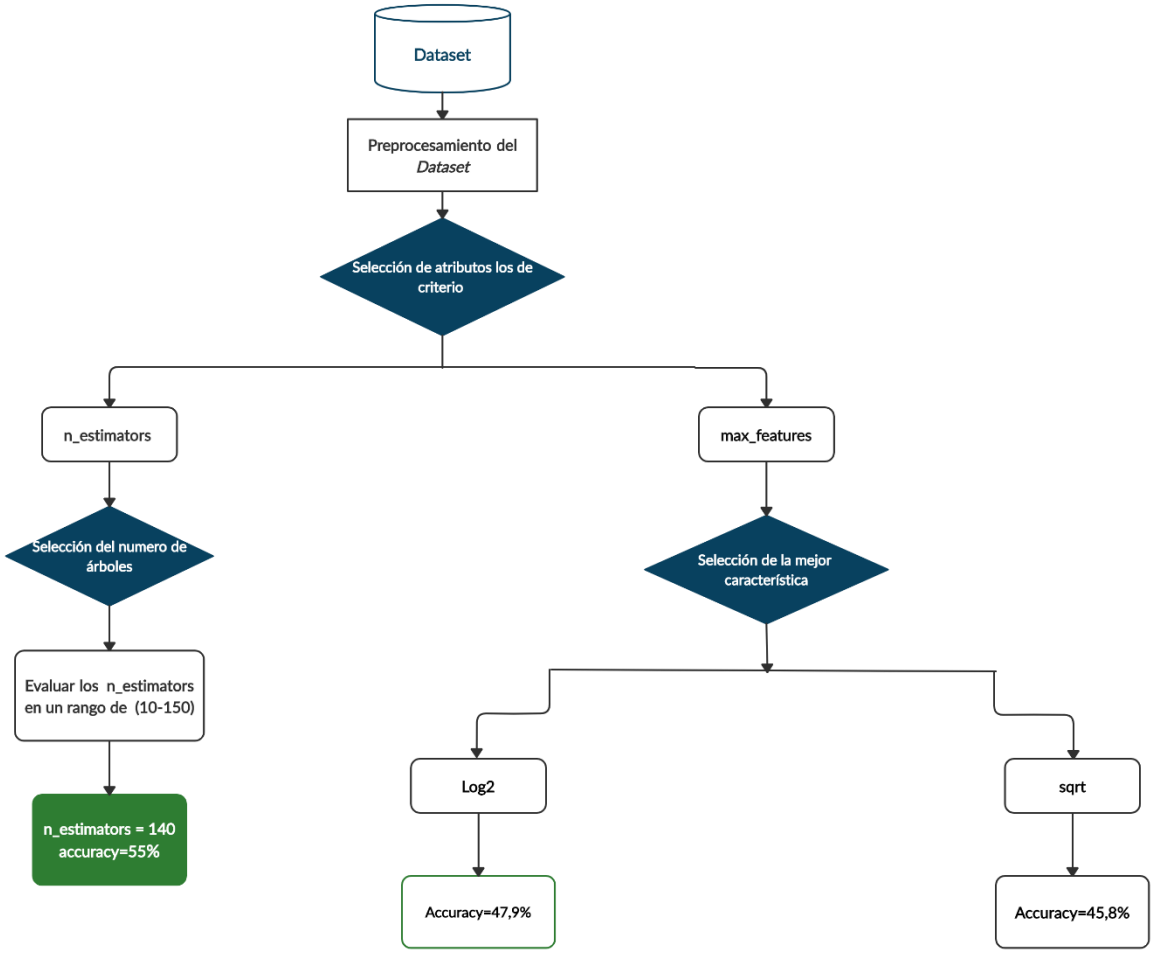
```
modelo = DecisionTreeClassifier(criterion='gini',splitter='best')
modelo.fit(x_train,y_train)
print(modelo.score(x_test,y_test))
y_predict = modelo.predict(X)
0.3541666666666667

modelo = DecisionTreeClassifier(criterion='gini',splitter='random' )
modelo.fit(x_train,y_train)
print(modelo.score(x_test,y_test))
y_predict = modelo.predict(X)
0.3125

Modelo =DecisionTreeClassifier(criterion='entropy',splitter='random')
modelo.fit(x_train,y_train)
print(modelo.score(x_test,y_test))
y_predict = modelo.predict(X)
0.4793333333333333

modelo = DecisionTreeClassifier(criterion='entropy',splitter='best' )
modelo.fit(x_train,y_train)
print(modelo.score(x_test,y_test))
y_predict = modelo.predict(X)
0.375
```

Figura 67.Estructura del modelo seleccionado de decision tree con los atributos entropy y random.



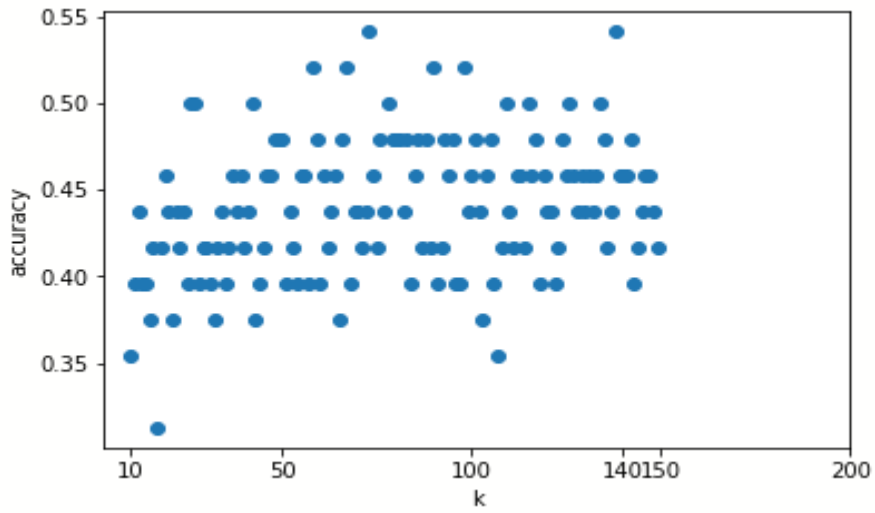
**6.3.7. Construcción del modelo de Bosques aleatorios (Random Forest).** Para la creación del algoritmo de Random Forest se van a realizar comparaciones entre los atributos de n_estimators y de max_features con el fin de determinar con cual se debería realizar el modelo de predicción, la figura 68 se muestra el flujograma para la creación de los algoritmos de Random Forest.

Figura 68.Flujograma del desarrollo del modelo de Random Forest.




Para la construcción de este modelo se probaron los atributos **n_estimators** (Mencionado en el capítulo 5), este método consiste en la creación de árboles, para el modelo se varió el parámetro en un rango de 10 a 150, la figura 69 muestra el resultado.

*Figura 69.Variación de los n_estimators del modelo en un rango de 10 a 150 respecto al accuracy..*




De acá se puede observar que la mayor predicción hace referencia a un valor de **55%** mediante el uso de **n_estimators = 140,** una vez obtenido se probara el parámetro de **max_features** en sus valores de ( log2 y sqrt) (Figura 70).

Figura 70.Código del para la construcción de los algoritmos de Random Forest con el atributo max_features

```
modelo = RandomForestClassifier(max_features= "log2")
modelo.fit(x_train,y_train)
print(modelo.score(x_test,y_test))
y_predict = modelo.predict(X)
0.4791666666666667

modelo = RandomForestClassifier( max_features= "sqrt")
modelo.fit(x_train,y_train)
print(modelo.score(x_test,y_test))
y_predict = modelo.predict(X)
0.4583333333333333
```

Mediante el atributo **max_features** con la que se logra un mejor modelo es con el método **log2** con un valor de accuracy de **47,9%,** lo que quiere decir que para este algoritmo se usara el atriburo **n_estimators = 140** ya que con este se obtuvo un accuracy de **55%.**

### 6.4. Optimización de los modelos creados

Una vez se ha definido los atributos de cada uno de los algoritmos, es necesario realizar una optimización de ellos con la finalidad de poder realizar la predicción del *Run life*. La tabla 11 muestra los algoritmos anteriormente expuestos con la mejor configuración de los mismos.

Tabla 11. Representación de los mejores algoritmos seleccionados de cada modelo con sus respectivos atributos o configuraciones.

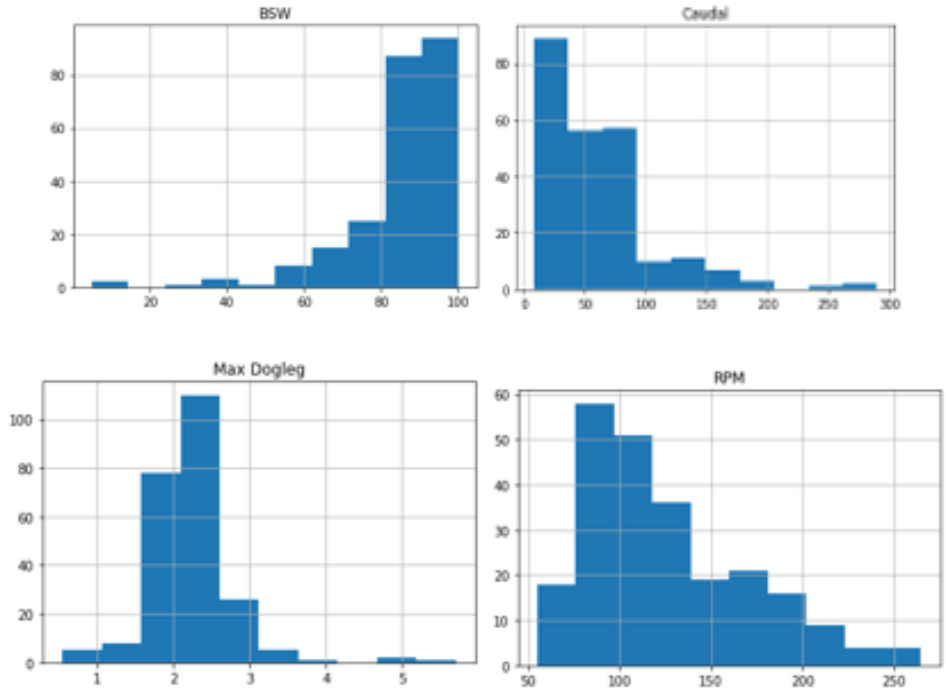
| Algoritmo | Configuración | Accuracy |
|---|---|---|
| SVM | C=1000 | 56,25% |
| K-NN | radius= 540 | 56,25% |
| Naive Bayes G | var_smoothing y Priors(Default) | 50% |
| Decision tree | criterion= entropy Splitter= random | 47,9% |
| Random Forest | n_estimators = 140 | 55% |
| RNA | 10 capas ocultas, ReLu | 41% |

De la tabla se puede destacar que los algoritmos que presentaron un mayor accuracy pertenecen al modelo de Random Forest, K-NN y SVM, los cuales serán

a los cuales se les hará un proceso de optimización para mejorar la predicción del modelo y determinar cuál de estos podría llegar a ser considerado.

Una de las formas que se pensó para poder disminuir el error de la predicción de los algoritmos fue mediante el análisis de la distribución de los parámetros de entrada en el Dataset, para ello se realizaron histogramas de cada uno de estos como se pueden detallar en la figura 71, de acá se pueden observar que los datos no presentan una distribución, esto puede ser lo que este ocasionando que los algoritmos planteados no estén presentando optimas predicciones.

Figura 71.Histogramas de cada uno de los parámetros incluidos en el Dataset para la predicción del Run life.

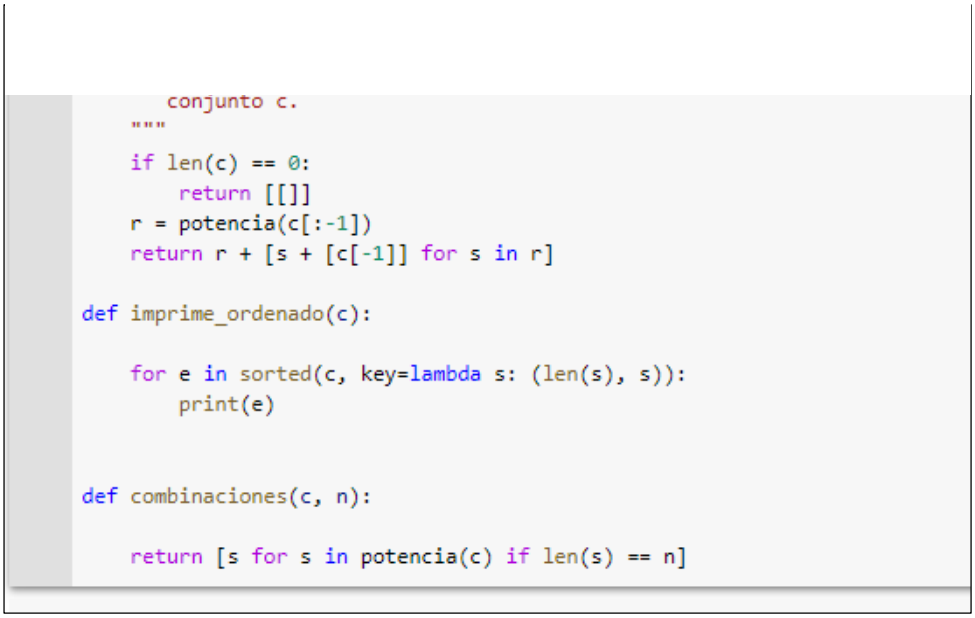



### 6.4.1. Creación de la matriz con los mejores parámetros para cada algoritmo.

Como se observó anteriormente los datos de los parámetros del Dataset no están presentando una distribución, lo que posiblemente sea un factor por el cual no se

ha obtenido una buena predicción del Run life de las bombas de campo Casabe. Con la finalidad de disminuir estos errores se ha planteado realizar la construcción de una matriz de características donde cada modelo evaluará cada uno de los parámetros y creara *subdataset* para cada uno de los algoritmos creados, donde cada uno trabajara con los parámetros con los cuales se logró un mejor accuracy. Las figuras 72 y 73 muestran el código que se utilizó para este proceso.

Figura 72.Código para crear las combinaciones de las mejores características.



```
        conjunto c.
    """
    if len(c) == 0:
        return [[]]
    r = potencia(c[:-1])
    return r + [s + [c[-1]] for s in r]

def imprime_ordenado(c):

    for e in sorted(c, key=lambda s: (len(s), s)):
        print(e)


def combinaciones(c, n):

    return [s for s in potencia(c) if len(s) == n]
```

Figura 73.Código del buscador de las características más relevantes.

```
    #s = cross_val_score(modelo, xPrueba, y_train, cv=KFold(5, shuffle=True),
    #precisionDePrueba = np.mean(s)
    modelo.fit(xPrueba,y_train)
    precisionDePrueba = modelo.score(xTPrueba,y_test)

    if precisionDePrueba > Best['acc']:

      Best['acc'] = precisionDePrueba
      Best['XTR'] = xPrueba
      Best['XT']  = xTPrueba
      Best['caracteristicas'] = combI

time.sleep(1)
print('Precision'      , Best['acc'])
print('caracteristicas', Best['caracteristicas'])

return Best
```

Finalmente, cuando se ha diseñado el buscador de las características más relevantes de entrada del modelo, se procede a utilizar este en cada uno de los algoritmos de inteligencia artificial (Figura 74).

Figura 74.Código del accuracy y las características obtenidas por los tres modelos seleccionados.

```
[34] best = busqueda(SVC(C= 1000))

100%|██████████| 6/6 [00:07<00:00,  1.27s/it]
Precision 0.4791666666666667
caracteristicas [5, 6, 7]
```

```
[26] best = busqueda(KNeighborsClassifier(radius= 540))

100%|██████████| 6/6 [00:00<00:00,  7.49it/s]
Precision 0.5416666666666666
caracteristicas [4, 6]
```

```
[29] best = busqueda(RandomForestClassifier(n_estimators= 140))

100%|██████████| 6/6 [00:40<00:00,  6.76s/it]
Precision 0.6458333333333334
caracteristicas [0, 1, 4, 5, 6, 7]
```

Una vez realizado el modelo con las mejores características se puede observar que el modelo de *Random forest* es el que presenta la mayor predicción obteniendo un Accuracy de **65%**, **mediante el uso de las características**

0 = Máxima desviación del pozo

1 = Profundidad de la bomba

4 = Nivel de fluido sobre la bomba

5 = Porcentaje de BSW

6= Caudal

7=THP

El modelo *Random Forest* con el atributo n_estimators= 140 se escoge como el mejor algoritmo de los que se estudiaron para determinar las fallas en el campo Casabe, puesto que este fue el que genero el mayor resultado tal como se aprecia en la figura 74, dando un valor de predicción en el entrenamiento de valor de 65% mientras que los otros métodos evaluados tuvieron resultados menores o iguales al 55 %.

Es posible que este método tuviese un mejor entrenamiento ya que como se mencionó en el capítulo 5 este algoritmo se basa en la creación de muchos árboles de decisión (Decision Tree), lo que quiere decir que crea múltiples combinaciones dando lugar a que pueda obtener una mayor predicción tal como se evidencio en este caso.

Una que se escoge este algoritmo con el uso de las características mencionadas anteriormente, se procede a realizar la verificación del modelo.

## 6.5 VERIFICACIÓN DEL MODELO

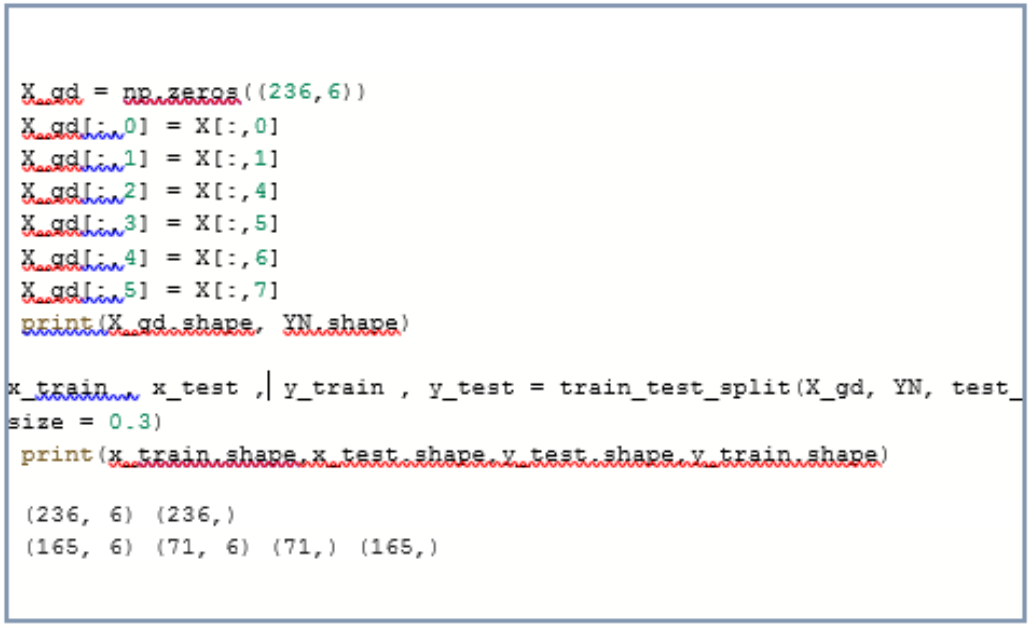
Como se menciono en el capitulo 5 existen tres metodos para poder verificar si un modelo es adecuado o no para cierto problema, esto se hace mediante el uso de tres medidores.

- Matriz de confusión
- Accuracy

Los cuales serán usados para la verificación del modelo de Random Forest seleccionado.

Primeramente se procede a hacer la reconstrucción del dataset original para dejar solamente las caracteristicas con las cuales el modelo random forest permitio generar una mayor predicción,codigo mostrado en la figura 75.

Figura 75.Código para la reconstrucción del dataset teniendo en cuenta las características obtenidas para el modelo de Random Forest.



```
X_gd = np.zeros((236,6))
X_gd[:,0] = X[:,0]
X_gd[:,1] = X[:,1]
X_gd[:,2] = X[:,4]
X_gd[:,3] = X[:,5]
X_gd[:,4] = X[:,6]
X_gd[:,5] = X[:,7]
print(X_gd.shape, YN.shape)

x_train, x_test , y_train , y_test = train_test_split(X_gd, YN, test_
size = 0.3)
print(x_train.shape,x_test.shape,y_test.shape,y_train.shape)

(236, 6) (236,)
(165, 6) (71, 6) (71,) (165,)
```

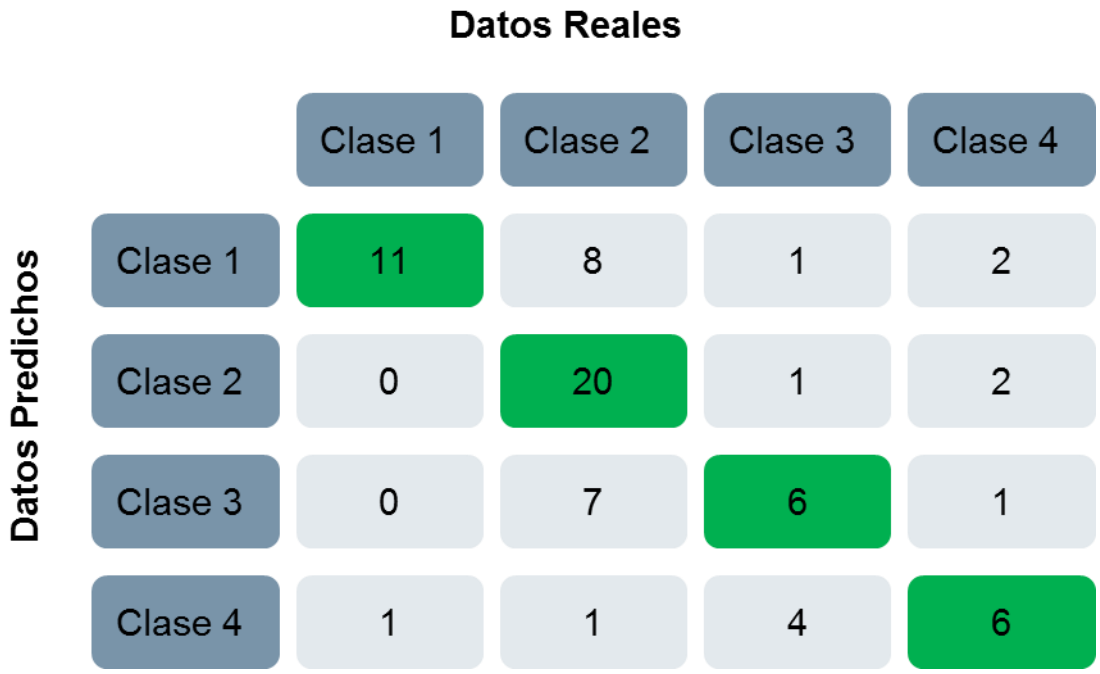
**6.5.1 Matríz de Confusión** Una vez se crea el nuevo dataset se procede nuevamente a cargar el algoritmo de *Random Forest* y a crear la matriz de confusión, como se puede observar en la figura 76.

Figura 76.Código para la construcción de la matriz de confusión

```
modelo = RandomForestClassifier(n_estimators= 140)
modelo.fit(x_train,y_train)
print(modelo.score(x_test,y_test))
y_predict = modelo.predict(X_gd)
0.6458333333333334
y_predict = modelo.predict(x_test)
confusion_matrix = confusion_matrix(y_test, y_predict)
confusion_matrix

array([[11,  8,  1,  2],
       [ 0, 20,  1,  2],
       [ 0,  7,  6,  1],
       [ 1,  1,  3,  6]])
```

Como se explico en el capitulo 5 una matriz de confusion permite detallar si realmente la clase que predijo el modelo cumple o no, en la figura 76 se muestra el comportamiento que presento el algortimo realizando la prueba del comportamiento de las muestras que se predijeron mediante el modelo y los datos reales, cabe aclarar que la clase 1 corresponde al primer cutrimestre en el que las bombas fallan, la clase 2 al segundo cuatrimestre y asi sucesivamente.

Figura 77.Representación de la matriz de confusión del algoritmo Random Forest para los datos de Casabe.




De la matriz de confución que se logro extraer del modelo se puede observar que de 22 muestras observadas para el fallo en las bombas PCP 11 de ellas fueron clacificadas correctamente y 8 fueron clasificadas en el 2 cuatrimestre, lo que muestra que el modelo no se desvio tanto ya que estuvieron cerca a la clase 1.

Para la Clase 2 que corresponde al segundo cuatrimestre se puede observar que presento una predicción muy buena ya que de las 23 muestras 20 fueron correctas, lo cual es un muy buen indicador.

Para la Clase 3 que corresponde al cuarto cuatrimestre se puede observar que el modelo no cumplio con el objetivo ya que logro predecir solo 6 muestras de las 15 que se tenian para esta clase, esto pudo deverse a que como los datos que se presentan en el dataset no siguen un distribución el modelo puede generar dificultad a la hora de predecir correctamente.

Para la Clase 4 que corresponde al cuarto cuatrimestre de las 11 clases, 6 fueron clasificadas correctamente y 4 de ellas fueron clasificadas como si estivieran en el cuatrimestre 3.

Por otro lado se puede observar un comportamiento en el cual la segunda mayor cantidad de datos se encuentra en el cuatrimestre siguiente, esto puede permitir pensar que aunque los datos estan dispersos ellos encuntran tendensias de similitud.

**6.5.2 ACCURACY** El modelo de random forest seleccionado obtuvo un accuracy de 65% como se observa en la apartir de la figura 78.

Figura 78.Código para la determinación del accuracy en el modelo.

```
modelo = RandomForestClassifier(n_estimators= 140)
modelo.fit(x_train,y_train)
print(modelo.score(x_test,y_test))
y_predict = modelo.predict(X_gd)
0.6458333333333334
```

Tanto de la matriz de confusión como del accuracy obtenido del modelo se puede observar que a pesar de que se realizó un arduo trabajo para la obtención de un modelo óptimo para la predicción del Run life en el campo Casabe el mayor valor que se logra obtener es de un 65%, esto puede estar siendo provocado por problemas en los datos que se están manejando dentro del dataset, ya sea porque la dispersión de los datos no permite que el modelo pueda llegar a tener un mejor resultado, a pesar de que estos fueron tratados para generar una mejor distribución y hacerle un proceso de normalización. Por otro lado, se debería verificar como

Ecopetrol está llevando el archivo del seguimiento de los datos de falla en el Campo ya que el mal muestreo de los datos puede generar que el modelo no sea del todo confiable.

Por otro lado, como Ecopetrol para el campo Casabe presenta pocos datos para las fallas, con solo 236 muestreos y considerando que de estos 188 fueron utilizados para la construcción del modelo, puede que no sean tan representativos como se esperaba.

# 7. ANÁLISIS PRESUPUESTAL

En el campo Casabe se presenta una gran cantidad de pozos con diferidas de producción, lo que hace que la empresa Ecopetrol S.A. deje de recibir dinero por esta causa. A través de este proyecto, se busca determinar previamente el posible fallo de un pozo que tenga instalado una bomba de cavidades progresivas como levantamiento artificial. Así, se podrá decidir intervenir el pozo antes de que falle del sistema, con lo que ocasionará una reducción del tiempo en el que el pozo deja de producir por la falla, lo cual, representa una disminución en la diferida, Automáticamente se evidencia en una ganancia económica para la empresa.

De la base de datos de Ecopetrol S.A en el campo Casabe, se eligieron 20 pozos (Tabla 13) para el análisis financiero. Se buscó que los pozos elegidos tuvieran tiempos de diferidas que representaran las diferidas promedio del campo. Este análisis, dará una perspectiva del dinero que está dejando de recibir la empresa debido a tiempos sin producción y mostrará lo beneficioso que puede llegar a hacer la predicción de fallas en las bombas de cavidades progresivas en el campo Casabe por medio de inteligencia artificial.

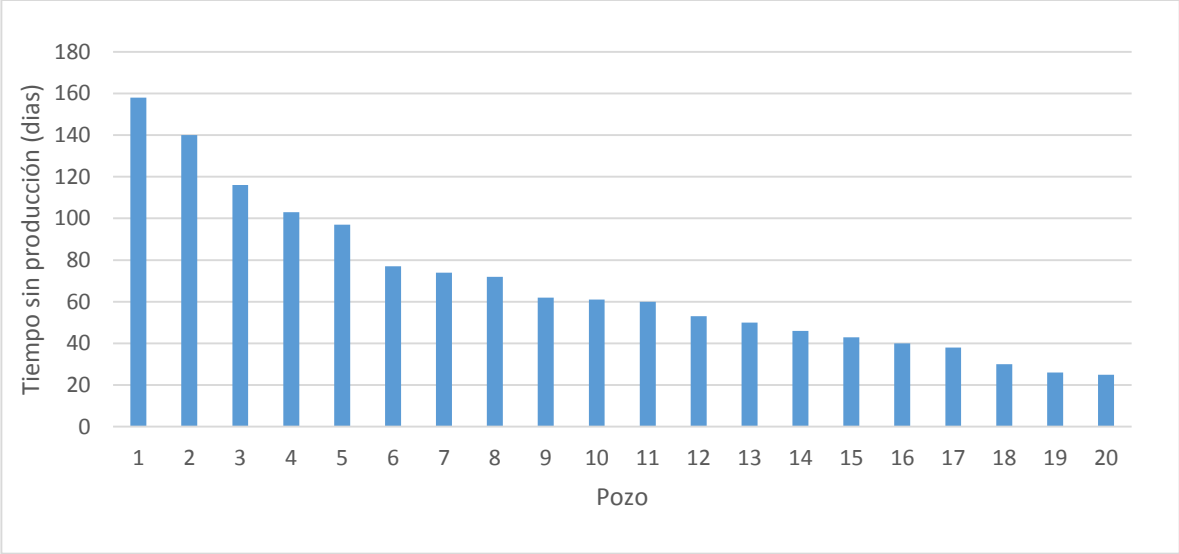
Tabla 13. Pozos seleccionados para el análisis presupuestal. El caudal de producción se determina con base en los últimos registros de producción del pozo.

| Pozo | Días sin Producción | Caudal Promedio (bopd) |
|---|---|---|
| 1 | 158 | 60 |
| 2 | 140 | 87 |
| 3 | 116 | 57 |
| 4 | 103 | 37 |
| 5 | 97 | 21 |

| | | |
|---|---|---|
| 6 | 77 | 51 |
| 7 | 74 | 43 |
| 8 | 72 | 33 |
| 9 | 62 | 40 |
| 10 | 61 | 36 |
| 11 | 60 | 43 |
| 12 | 53 | 25 |
| 13 | 50 | 35 |
| 14 | 46 | 41 |
| 15 | 43 | 64 |
| 16 | 40 | 48 |
| 17 | 38 | 45 |
| 18 | 30 | 76 |
| 19 | 26 | 82 |
| 20 | 25 | 75 |

Como se evidencia en la figura 77, existen pozos con tiempos de paro de más de 4 meses. Sin embargo, la mayoría de los pozos se interviene en el transcurso de los 3 meses posteriores a la falla. Como se ha mencionado anteriormente, algunas razones de la demora para ser ejecutada la intervención pueden ser por la falta de disponibilidad del componente fallado del sistema (trámites de importación o reparación del mismo), equipo de *workover* ejecutando otra intervención, entre otros aspectos.

Figura 79.Pozos seleccionados para el análisis presupuestal.




Luego de la recopilación de los pozos a analizar. Se tiene que tener en cuenta dos variables. El costo del crudo promedio de venta (USD/BBL) durante las diferidas de los pozos. Con esto, determinamos el dinero que dejó de recibir la empresa. Sin embargo, a este valor, se le debe sustraer el costo de producción de cada barril (*Lifting Cost*), las regalías por producción y el costo de tratamiento de agua por barril. Estas últimas variables, son determinantes, puesto que nos dan un indicio del dinero que dejó de "gastar" la empresa por la falta de producción.

A continuación, se presenta una breve explicación de estas variables empleadas en análisis presupuestal.

## 7.1 CONCEPTOS EMPLEADOS EN EL ANÁLISIS PRESUPUESTAL

**7.1.1 Valor Bruto del Aceite** Es el precio recibido por venta en campo, este se expresa en dólares por barril (USD/BBL). El valor del barril, viene determinado por costos variables y otros fijos. Los costos variables dependen directamente de la

extracción de dicho barril. Si el crudo no se produce el costo no es grabado (regalías, impuestos, consumo eléctrico, tratamiento). Por otro lado, los costos fijos no están influenciados por la producción de un campo o pozo específico.

**7.1.2 Regalías** Es el pago que debe realizar una empresa al Estado por explotar un recurso natural no renovable dentro de la nación. El valor del pago se determina de acuerdo a la producción del campo, según lo indica el artículo 16 de la Ley 756 de 2002. Así: *"Establece como regalía por la explotación de recursos naturales no renovables de propiedad nacional, sobre el valor de la producción en boca o borde de mina o pozo…".*

Este pago corresponde entre el 8% y el 25% del valor de producción del crudo en el pago y está determinado según la tabla 10.

Tabla 14. Pago de Regalías de un campo según su producción mensual.

| **Producción Promedio del campo en el mes (bbl)** | **Porcentaje de Regalías** |
|---|---|
| Menor o igual a 5.000 | 8% |
| 5.000 – 125.000 | 8% + (Producción Promedio Mes Kopd – 5 Kopd) *(0.1) |
| 125.000 – 400.000 | 20 % |
| 400.000 – 600.000 | 20 % + (Producción Promedio Mes Kopd – 400 Kopd) *(0.025) |
| Mayor a 600.000 | 25 % |

El promedio mensual para el campo Casabe comprendidos entre el año 2016 al 2018 fue de 11,276 BPD. De acuerdo a la tabla 11 se realiza el cálculo del porcentaje de regalías así:

$$\text{Porcentaje de Regalías} = 8\,\% + (\,11.276\ \text{KBPD} - 5.0\ \text{KBBPD}) * (0.1)$$

$$\text{Porcentaje de Regalías} = 8\,\% + 0.6276\,\% = 8.6276\%$$

Según el anterior dato, el porcentaje de regalías promedio que debe pagar Ecopetrol SA por el campo Casabe, es de 8.6276 %. Este dato se tendrá en cuenta en el análisis presupuestal. Tabla 15.

**7.1.3 Lifting Cost** Se refiere al costo de producción de un barril de crudo en el campo luego de la fase de perforación y completamiento. Estos costos pueden incluir mantenimiento de los equipos de levantamiento y de las facilidades de superficie, costos por energía, costos de transporte, entre otros. Todos estos, se determinan de acuerdo a un barril de crudo (USD/bbl). Para el campo Casabe, se tiene un estimado de 13 USD/bbl.

**7.1.4 Tratamiento de Agua** Como se ha mencionado anteriormente, el campo Casabe presenta como método de recobro secundario la inyección de agua, la cual ha sido implementada por más de 30 años. Es por ello, que se presenta un BSW tan elevado (85-95%). Debido a esto, se debe tener en cuenta el tratamiento del volumen de agua producida por pozo, el cual se tiene un estimado de 0.3 USD/bbl para el campo Casabe.

## 7.2 ANÁLISIS PRESUPUESTAL

Con base en la tabla 9 se determinan los barriles totales que no fueron producidos, multiplicando los días sin producción por el caudal promedio del pozo. (columna 3, tabla 15). Ahora bien, para conocer el valor en USD de los barriles dejados de producir (costo del valor bruto), se determina el promedio del costo del barril durante los años 2016 y 2017, encontrándose un promedio de 55 USD/bbl.

Como se puede observar en la tabla 15, se puede inferir que Ecopetrol SA dejó de vender aproximadamente 3.66 millones de dólares en solo 20 pozos que tuvieron fallas durante el año 2016 al 2018.

De igual manera, el dinero que dejó de ganar la empresa, se puede determinar descontando al valor anterior los costos de producción (Lifting Cost, regalías, tratamiento de agua). Estos valores pueden denominarse costos que la empresa no tuvo que pagar, ya que no se produjeron estos barriles.

Como se determinó anteriormente, las regalías aproximadas que Ecopetrol da a la nación por ventas del crudo en el campo Casabe son del 8.6 %. El Lifting Cost del Campo aproximadamente es del 13% y el tratamiento del agua es del 0.3 dólares por barril. En la tabla 15, se evidencian los resultados.

Se puede evidenciar Ecopetrol dejó de ganar 2.8 millones de dólares en solo 20 pozos que no produjeron por falta de una intervención temprana.

Tabla 15. Costos totales por Regalías, Lifting Cost y Tratamiento de Agua para los 20 pozos del análisis presupuestal. Además, saldo neto del dinero que Ecopetrol SA dejó de recibir por el tiempo en que estos pozos que no produjeron por daño de la bomba PCP.

| ***Pozo*** | ***Días sin Producción*** | ***Caudal Promedio del pozo (bbl)*** | ***Total Barriles no Producidos (bbl)*** | ***USD de Crudo Bruto**** | ***Regalías*** | ***Lifting Cost*** | ***Tratamiento del Agua (USD)*** |
|---|---|---|---|---|---|---|---|
| 1 | 158 | 60 | 9480 | 521400 | | | 2844 |
| 2 | 140 | 87 | 12180 | 669900 | | | 3654 |
| 3 | 116 | 57 | 6612 | 363660 | | | 1984 |
| 4 | 103 | 37 | 3811 | 209605 | | | 1143 |
| 5 | 97 | 21 | 2037 | 112035 | | | 611 |
| 6 | 77 | 51 | 3927 | 215985 | | | 1178 |
| 7 | 74 | 43 | 3182 | 175010 | | | 955 |
| 8 | 72 | 33 | 2376 | 130680 | | | 713 |
| 9 | 62 | 40 | 2480 | 136400 | | | 744 |
| 10 | 61 | 36 | 2196 | 120780 | 8.6% | 13% | 659 |
| 11 | 60 | 43 | 2580 | 141900 | | | 774 |
| 12 | 53 | 25 | 1325 | 72875 | | | 398 |
| 13 | 50 | 35 | 1750 | 96250 | | | 525 |
| 14 | 46 | 32 | 1472 | 80960 | | | 442 |
| 15 | 43 | 39 | 1677 | 92235 | | | 503 |
| 16 | 40 | 36 | 1440 | 79200 | | | 432 |
| 17 | 38 | 45 | 1710 | 94050 | | | 513 |
| 18 | 30 | 76 | 2280 | 125400 | | | 684 |
| 19 | 26 | 82 | 2132 | 117260 | | | 640 |
| 20 | 25 | 75 | 1875 | 103125 | | | 563 |
| Total | | | 66522 | **3658710** | 315659 | 475632 | 19957 |
| | *Saldo neto descontando los costos de producción* | | | | *3343051* | 2867419 | ***2847462*** |

*Tasa de cambio promedio del periodo 2016-1027, de 55 USD/BBl

# 8. CONCLUSIONES

Las bombas de cavidades progresivas son el método de levantamiento artificial más usado en el campo Casabe con un valor de más del 60% en comparación con los otros métodos de levantamiento artificial. Ahora bien, de acuerdo con el análisis de fallas en los registros recopilados y compartidos por Ecopetrol SA, se puede concluir que los componentes de las bombas PCP que presentan falla más a menudo son la tubería, el estator, el rotor y la varilla.

A través del análisis de datos y el desarrollo de algoritmos en el programa PYTHON, se concluye que las características más relevantes en el run-life de las bombas PCP en el campo Casabe son: Máximo *Dog-Leg*, profundidad de la bomba, Torque, BSW%, altura de fluido sobre la bomba y el caudal del pozo.

Antes de iniciar el desarrollo de un modelo mediante algoritmos de inteligencia artificial es necesario realizar un preprocesamiento de los datos contenidos en el Dataset que va a servir para el modelo ya que de esta manera se puede contribuir a que los modelos sean óptimos, para ello este debe ser un paso indispensable para la construcción de algoritmos.

Mediante la construcción y pruebas de atributos de los diferentes algoritmos de inteligencia artificial que fueron probados para la predicción del Run life en el campo Casabe se determinó que el algoritmo con el cual se obtenían los mejores resultados se lograba mediante el uso Random Forest usando el atributo de *n_estimators* igual 140, usando los parámetros de entrada de Máximo *Dog-Leg*, profundidad de la bomba, BSW%, altura de fluido sobre la bomba, caudal del pozo y el THP, obteniéndose una predicción del 65%.

El modelo de *Random Forest* para la predicción del run life de las bombas PCP del campo Casabe realizado en este proyecto de grado tiene un porcentaje de

predicción del 65%. Este valor puede estar reflejado por el tipo de datos con el que se trabajó, ya que estos, presentaban alta dispersión lo que no permite que el modelo pueda llegar a tener un mejor resultado, a pesar de que estos fueron tratados para generar una mejor distribución luego de realizar un proceso de normalización. Este resultado da un indicio del tiempo de vida media de la bomba, para así poder tener una alerta de posible falla. Es importante aclarar que la literatura expresa que el aumento del porcentaje en la predicción se logra con una mayor cantidad de datos de entrada. Para este caso, el modelo mejorará su predicción en el tiempo, a medida que se presenten nuevas fallas, las cuales, aumentarán el *dataset de* entrada al modelo.

Una vez que se ha seleccionado un algoritmo de inteligencia artificial se debe proceder a realizar una verificación del modelo lo cual se hace mediante la implementación de métricas de rendimiento de las cuales se destacan la matriz de confusión y el *Accuracy*, estas permiten determinar la predicción del modelo, comparando los datos reales respecto a los datos estimados del algoritmo.

Finalmente se concluye, que una predicción del momento en que fallará una bomba PCP (*run life del equipo*) ayuda a programar una intervención en el pozo antes de que falle el sistema, lo que ocasionará una reducción del tiempo en el que el pozo deja de producir debido a la falla, lo cual, representa una disminución en la diferida y automáticamente se evidencia en una ganancia económica para la empresa, como se puede observar al analizar sólo 20 pozos promedio del campo, en los cuales, Ecopetrol dejó de ganar 2.8 millones de dólares, que no produjeron por falta de una intervención temprana.

# 9. RECOMENDACIONES

Continuar con la recopilación de datos de falla de las bombas PCP en el campo Casabe y registrar su Run-Life, con el fin de obtener un *dataset* mucho más amplio para ingresar al modelo de Random Forest y así aumentar la predicción del run life de las bombas PCP en el campo. Esto traerá mayor confiabilidad en el modelo para la toma de decisiones de una posible intervención predictiva.

Programar una interfaz amigable para un fácil manejo del programa con el código, y así evitar que se realice por el programa Python el cual tiene un manejo más complejo. Esto permitirá que técnicos de Ecopetrol S.A. puedan usar el modelo y aplicarlo a la predicción del run-life de las bombas PCP en el campo Casabe.

Luego de obtener un mayor *dataset*, programar una red neuronal para evaluar estos datos. Las redes neuronales procesan y predicen de una mejor manera grandes cantidades de datos y esto puede que ayudar a incrementar la exactitud de la predicción del run life en las bombas PCP del campo Casabe.

# BIBLIOGRAFÍA


ACOSTA, T. et al. Recuperación mejorada en un yacimiento de alta complejidad estratigráfica: Campo Casabe (Caso de estudio). ACIPET 2017.

AGARWAL.S. et. al. *Advance in Completion Design to Improve Bhagyam PCP Run Life. Society of Petroleum Engineers* (SPE) 2016.

AMAYA. M. et. al. (2010) Casabe: Revitalización de un campo maduro*. En: Revista Oilfield review* Schlumberger. No. 1. (Abril-Mayo 2010)

AMERICAN PETROLEUM INSTITUTE, *Recommended Practice for Case and Use of Casing and Tubing.* Washington, SPEC. 5CT.1980.

BASOGAIN. X. Redes neuronales artificiales y sus aplicaciones. Bilbao (2009). Universidad del país Vasco (EUH). Escuela superior de ingeniería de Bilbao. Departamento de ingeniería de sistemas y automática.

CRUZ. N, ACOSTA. H, CARRILLO. H & BARRIENTOS. R. *Comparison of the Performance of Seven Classifiers as Effective Decision Support Tools for the Cytodiagnosis of Breast Cancer: A Case Study. Analysis and Design of Intelligent Systems using Soft Computing Techniques*. Serie de Libros: *Advances in Soft Computing*; (AINSC, volumen 41)41: 79 – 87.

ECOPETROL S.A Estadísticas de falla en pozos productores. Departamento de ingeniería y control de producción: Casabe. Barrancabermeja, Colombia. 2018.

GULLI.A y SUJIT. P. *Deep learning with Kers, implement neural networks with keras on Theano and TensorFlow*. Birmingham-Mumbai. 2017.

HASSAN A, AHMED H, SULEIMAN H y ALI H. *Prediction Production performance of progressive cavity pump for high viscous oil. Sudan University of science and technology. October 2017.*

HIRCHFELDT M. Manual de bombeo de cavidades progresivas. Volumen I. C-FER. 2008

KODU PCP. Complete progressing cavity pump system. Calgary, Canada. 2010.

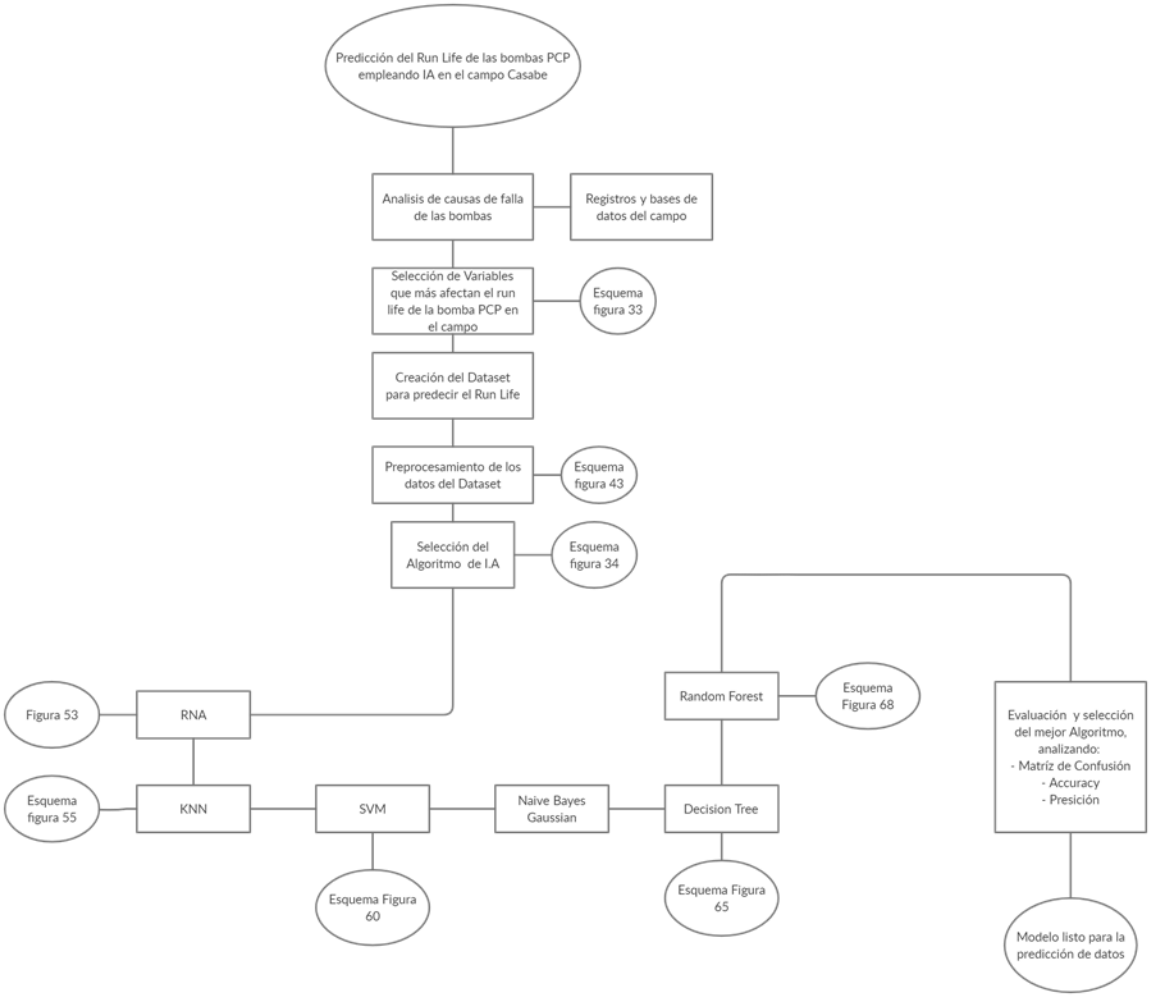
SUAREZ R, ZALAZAR M, SALINAS D y BESSONE J. Fallas de Recubrimientos metálicos empleados en equipos de producción de petróleo. Jornadas SAM/CONAMET/SIMPOSIO materia 2003[online], 2003.

WEATHERFORD. Bombas de cavidades progresivas, componentes del sistema. Programa de entrenamiento. 2003

YAÑEZ, M et al. Nota de Curso. Confiabilidad Integral. Capitulo II. Ingeniería de Confiabilidad de Equipos. 2007

# ANEXOS

## ANEXO A. Diagrama De flujo



## ANEXO B. *Dataset* de entrada de los modelos de predicción.

| Max Dogleg | PUMP Depth | Torque | RPM | Nivel de fluido sobre pump (ft) | BSW | Q (bbl/d) | THP | RL (Cutrimestre) |
|---|---|---|---|---|---|---|---|---|
| 1.44 | 3700 | 0.55 | 185 | 575 | 35 | | 160 | 1 |
| 1.44 | 3700 | 0.45 | 205 | 525 | 100 | 205 | 120 | 4 |
| 2.14 | 2900 | 0.35 | 125 | 675 | 85 | 45 | 80 | 1 |
| 2.2 | 3900 | 0.45 | 165 | 1025 | 85 | 65 | 135 | 4 |
| 2.68 | 2900 | 0.45 | 175 | 1125 | 75 | 290 | 100 | 2 |
| 2.68 | 2900 | 0.35 | 125 | 1025 | 85 | 195 | 120 | 1 |
| 2.68 | 2900 | 0.25 | 125 | 1675 | 85 | 75 | 100 | 2 |
| 2.58 | 2300 | 0.15 | 125 | 1875 | 85 | 45 | 90 | 2 |
| 2.58 | 2300 | 0.55 | 125 | 2225 | 85 | 15 | 140 | 1 |
| 1.69 | 2900 | 0.35 | 115 | 275 | 85 | 65 | 50 | 1 |
| 1.69 | 2900 | 0.35 | 105 | 1225 | 85 | 35 | 100 | 1 |
| 1.77 | 3100 | 0.25 | 135 | 1125 | 85 | 55 | 90 | 3 |
| 1.77 | 2500 | 0.15 | 105 | 1275 | 75 | 75 | 90 | 1 |
| 1.77 | 2700 | 0.45 | 155 | 525 | 95 | 45 | 100 | 2 |
| 1.77 | 3100 | 0.65 | 225 | 975 | 95 | 85 | 90 | 1 |
| 1.77 | 2900 | 0.25 | 145 | 925 | 95 | 55 | 90 | 2 |
| 1.77 | 3900 | 0.65 | 235 | 425 | 85 | 135 | 75 | 4 |
| 1.77 | 2900 | 0.45 | 115 | 1075 | 95 | 15 | 60 | 4 |
| 1.85 | 3300 | 0.55 | 145 | 375 | 95 | 8 | 90 | 3 |
| 2.58 | 2900 | 0.45 | 105 | 425 | 85 | 75 | 55 | 4 |
| 1.77 | 2700 | 0.05 | 85 | 2325 | 65 | 45 | 70 | 2 |
| 1.77 | 3700 | 0.35 | 175 | 1325 | 85 | 75 | 75 | 2 |
| 1.77 | 2500 | 0.15 | 85 | 2125 | 95 | 15 | 55 | 1 |
| 1.77 | 4500 | 0.45 | 85 | 325 | 35 | 145 | 50 | 4 |
| 1.76 | 2900 | 0.35 | 95 | 725 | 55 | 115 | 100 | 1 |
| 2.58 | 3100 | 0.65 | 125 | 1025 | 95 | 55 | 90 | 2 |
| 1.99 | 3300 | 0.25 | 205 | 425 | 85 | 45 | 60 | 2 |
| 2.38 | 2900 | 0.25 | 95 | 1975 | 95 | 8 | 85 | 1 |
| 1.77 | 2900 | 0.45 | 65 | 575 | 95 | 25 | 70 | 2 |
| 2.55 | 3100 | 0.45 | 105 | 1375 | 85 | 65 | 70 | 2 |
| 1.77 | 2900 | 0.15 | 75 | 825 | 95 | 8 | 80 | 1 |
| 1.77 | 2900 | 0.45 | 105 | 475 | 75 | 65 | 70 | 4 |
| 2.45 | 2700 | 0.35 | 85 | 1675 | 85 | 15 | 50 | 2 |
| 2.18 | 2500 | 0.55 | 165 | 275 | 85 | 55 | 80 | 2 |
| 2.18 | 2700 | 0.15 | 75 | 1025 | 95 | 35 | 60 | 1 |
| 2.18 | 2700 | 0.25 | 95 | 1175 | 95 | 25 | 60 | 2 |
| 2.18 | 3100 | 0.45 | 75 | 1875 | 95 | 25 | 60 | 2 |
| 2.18 | 3100 | 0.75 | 185 | 875 | 95 | 105 | 70 | 2 |
| 2.18 | 2300 | 0.35 | 85 | 875 | 85 | 35 | 60 | 2 |
| 2.18 | 2300 | 0.75 | 125 | 1225 | 95 | 15 | 70 | 1 |
| 2.18 | 2500 | 0.35 | 95 | 1375 | 85 | 65 | 80 | 1 |
| 2.18 | 2700 | 0.25 | 85 | 1275 | 85 | 25 | 80 | 2 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2.18 | 2700 | 0.65 | 165 | 375 | 95 | 75 | 100 | 2 |
| 2.18 | 2500 | 0.25 | 125 | 775 | 85 | 55 | 55 | 2 |
| 2.92 | 2700 | 0.25 | 105 | 625 | 85 | 35 | 60 | 2 |
| 2.19 | 2500 | 0.05 | 105 | 2225 | 85 | 15 | 90 | 3 |
| 2.25 | 3100 | 0.55 | 125 | 725 | 85 | 45 | 90 | 1 |
| 2.36 | 2700 | 0.25 | 95 | 1675 | 55 | 165 | 100 | 3 |
| 2.5 | 2700 | 0.35 | 85 | 1125 | 75 | 65 | 90 | 1 |
| 4.7 | 3300 | 0.25 | 155 | 875 | 75 | 35 | 85 | 2 |
| 4.7 | 3300 | 0.25 | 105 | 375 | 85 | 45 | 70 | 1 |
| 2.3 | 2700 | 0.15 | 165 | 2025 | 95 | 25 | 90 | 2 |
| 2.3 | 2700 | 0.25 | 85 | 1025 | 85 | 45 | 90 | 1 |
| 1.99 | 2700 | 0.15 | 125 | 625 | 95 | 8 | 70 | 1 |
| 2.59 | 2900 | 0.15 | 95 | 1675 | 95 | 15 | 70 | 2 |
| 2.36 | 2900 | 0.15 | 65 | 375 | 65 | 55 | 70 | 3 |
| 2.46 | 2900 | 0.35 | 175 | 225 | 95 | 35 | 100 | 4 |
| 1.88 | 2700 | 0.25 | 105 | 875 | 95 | 35 | 70 | 2 |
| 2.93 | 2900 | 0.45 | 125 | 525 | 85 | 35 | 60 | 4 |
| 2.93 | 2900 | 0.45 | 95 | 325 | 85 | 45 | 87,9 | 2 |
| 3.52 | 3300 | 0.25 | 85 | 1025 | 75 | 45 | 110 | 2 |
| 2.83 | 2700 | 0.15 | 105 | 2225 | 65 | 75 | 80 | 1 |
| 2.83 | 2700 | 0.05 | 55 | 2675 | 85 | 75 | 100 | 1 |
| 2.2 | 3700 | 0.25 | 125 | 1275 | 85 | 35 | 120 | 1 |
| 2.2 | 3700 | 0.15 | 115 | 1125 | 75 | 45 | 110 | 1 |
| 2.74 | 2700 | 0.75 | 145 | 775 | 85 | 155 | 80 | 3 |
| 1.43 | 2500 | 0.35 | 95 | 825 | 95 | 35 | 90 | 2 |
| 2.58 | 2300 | 0.25 | 75 | 1375 | 85 | 25 | 70 | 2 |
| 2.83 | 2700 | 0.35 | 85 | 425 | 85 | 55 | 100 | 2 |
| 2.83 | 2700 | 0.25 | 95 | 1325 | 85 | 25 | 70 | 2 |
| 1.77 | 2700 | 0.25 | 105 | 1725 | 85 | 45 | 100 | 1 |
| 1.77 | 2900 | 0.35 | 95 | 575 | 85 | 45 | 80 | 1 |
| 1.46 | 3900 | 0.45 | 85 | 925 | 85 | 65 | 100 | 4 |
| 1.77 | 2500 | 0.55 | 115 | 925 | 95 | 25 | 60 | 1 |
| 1.85 | 3300 | 0.55 | 95 | 475 | 95 | 15 | 95 | 1 |
| 1.77 | 3500 | 0.45 | 205 | 475 | 95 | 45 | 70 | 1 |
| 1.77 | 2700 | 0.15 | 85 | 1675 | 65 | 55 | 50 | 1 |
| 2.09 | 3300 | 0.25 | 195 | 225 | 85 | 35 | 100 | 3 |
| 1.77 | 2500 | 0.15 | 85 | 2175 | 95 | 15 | 40 | 1 |
| 2.18 | 3300 | 0.25 | 95 | 1375 | 65 | 75 | 80 | 1 |
| 2.18 | 3100 | 0.35 | 165 | 775 | 75 | 125 | 100 | 1 |
| 1.77 | 3700 | 0.25 | 115 | 1425 | 65 | 75 | 120 | 4 |
| 2.12 | 2900 | 0.15 | 145 | 525 | 55 | 75 | 75 | 2 |
| 2.45 | 2700 | 0.35 | 85 | 2775 | 85 | 45 | 80 | 2 |
| 2.45 | 3100 | 0.45 | 85 | 625 | 55 | 85 | 80 | 1 |
| 1.76 | 2900 | 0.35 | 175 | 275 | 75 | 45 | 0 | 1 |
| 1.77 | 3100 | 0.25 | 105 | 1725 | 95 | 15 | 70 | 2 |
| 1.77 | 3100 | 0.35 | 115 | 1025 | 95 | 15 | 40 | 1 |
| 2.25 | 3100 | 0.15 | 135 | 1075 | 95 | 15 | 70 | 1 |
| 2.38 | 2900 | 0.25 | 85 | 1225 | 75 | 65 | 80 | 1 |
| 2.38 | 2900 | 0.25 | 95 | 1425 | 95 | 25 | 80 | 2 |
| 1.77 | 2900 | 0.35 | 95 | 2175 | 85 | 15 | 80 | 1 |
| 2.18 | 3100 | 0.45 | 215 | 1275 | 95 | 35 | 130 | 1 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2.18 | 2700 | 0.25 | 95 | 1125 | 95 | 65 | 60 | 1 |
| 2.18 | 2700 | 0.25 | 65 | 675 | 95 | 15 | 70 | 1 |
| 0.54 | 3100 | 0.45 | 95 | 475 | 75 | 65 | 70 | 2 |
| 2.18 | 3100 | 0.45 | 85 | 150 | 75 | 95 | 50 | 1 |
| 2.15 | 2700 | 0.35 | 115 | 675 | 85 | 55 | 80 | 3 |
| 2.18 | 2500 | 0.55 | 115 | 825 | 85 | 15 | 75 | 2 |
| 1.8 | 3900 | 0.15 | 85 | 1875 | 75 | 25 | 70 | 1 |
| 2.25 | 3100 | 0.55 | 125 | 675 | 95 | 55 | 60 | 1 |
| 2.25 | 3100 | 0.35 | 135 | 925 | 85 | 45 | 75 | 1 |
| 2.43 | 2700 | 0.25 | 75 | 150 | 75 | 65 | 100 | 1 |
| 2.5 | 2500 | 0.25 | 65 | 1525 | 75 | 55 | 80 | 2 |
| 1.99 | 2700 | 0.25 | 145 | 825 | 85 | 65 | 80 | 1 |
| 1.99 | 2700 | 0.15 | 115 | 875 | 55 | 55 | 70 | 1 |
| 1.79 | 2900 | 0.25 | 85 | 2075 | 95 | 35 | 0 | 1 |
| 1.79 | 2900 | 0.35 | 125 | 675 | 95 | 25 | 70 | 2 |
| 2.73 | 2500 | 0.55 | 165 | 675 | 95 | 25 | 120 | 4 |
| 2.65 | 2900 | 0.45 | 125 | 575 | 85 | 75 | 90 | 4 |
| 2.14 | 2900 | 0.25 | 95 | 1625 | 85 | 35 | 90 | 2 |
| 2.79 | 2500 | 0.45 | 105 | 225 | 95 | 65 | 90 | 2 |
| 2.49 | 2300 | 0.35 | 125 | 725 | 95 | 45 | 80 | 2 |
| 2.49 | 2300 | 0.25 | 105 | 1075 | 95 | 15 | 90 | 2 |
| 2.74 | 3700 | 0.45 | 105 | 1625 | 85 | 85 | 80 | 1 |
| 2.74 | 2500 | 0.25 | 135 | 225 | 85 | 55 | 95 | 2 |
| 2.66 | 2700 | 0.45 | 95 | 1175 | 95 | 25 | 80 | 2 |
| 2.39 | 2900 | 0.55 | 105 | 1275 | 95 | 35 | 70 | 4 |
| 1.43 | 2500 | 0.35 | 75 | 875 | 95 | 35 | 90 | 3 |
| 3.31 | 2500 | 0.35 | 135 | 575 | 95 | 35 | 60 | 2 |
| 3.31 | 2500 | 0.25 | 95 | 1175 | 85 | 85 | 60 | 2 |
| 1.71 | 3300 | 0.05 | 105 | 775 | 5 | 65 | 65 | 2 |
| 1.77 | 3100 | 0.45 | 185 | 575 | 95 | 55 | 90 | 2 |
| 2.63 | 3100 | 0.35 | 105 | 1975 | 95 | 55 | 100 | 2 |
| 1.77 | 2700 | 0.25 | 105 | 825 | 85 | 45 | 70 | 1 |
| 1.83 | 2700 | 0.55 | 265 | 675 | 95 | 75 | 90 | 4 |
| 1.77 | 2900 | 0.45 | 175 | 975 | 95 | 65 | 80 | 2 |
| 1.77 | 2500 | 0.45 | 145 | 1175 | 95 | 55 | 65 | 2 |
| 1.77 | 3900 | 0.55 | 125 | 1775 | 85 | 85 | 100 | 3 |
| 1.77 | 2500 | 0.35 | 135 | 925 | 95 | 35 | 100 | 2 |
| 1.77 | 2900 | 0.65 | 165 | 475 | 85 | 75 | 70 | 3 |
| 2.46 | 3500 | 0.35 | 95 | 1175 | 95 | 8 | 90 | 4 |
| 1.77 | 3100 | 0.65 | 155 | 775 | 75 | 65 | 70 | 3 |
| 1.77 | 3300 | 0.55 | 115 | 475 | 95 | 35 | 65 | 2 |
| 1.77 | 3500 | 0.55 | 145 | 525 | 75 | 175 | 95 | 4 |
| 1.77 | 3700 | 0.25 | 95 | 2025 | 85 | 55 | 50 | 4 |
| 1.77 | 3300 | 0.25 | 115 | 1175 | 85 | 35 | 100 | 4 |
| 2.18 | 3300 | 0.45 | 265 | 375 | 65 | 125 | 197 | 4 |
| 2.18 | 3100 | 0.35 | 185 | 375 | 85 | 65 | 90 | 2 |
| 1.77 | 2700 | 0.25 | 235 | 275 | 65 | 145 | 74 | 4 |
| 1.77 | 2700 | 0.35 | 205 | 225 | 75 | 175 | 122 | 4 |
| 2.58 | 3100 | 0.65 | 125 | 375 | 95 | 25 | 80 | 2 |
| 2.58 | 3100 | 0.65 | 105 | 575 | 95 | 15 | 80 | 2 |
| 1.77 | 3100 | 0.25 | 105 | 475 | 85 | 85 | 90 | 1 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1.92 | 3100 | 0.35 | 115 | 325 | 95 | 45 | 80 | 2 |
| 1.96 | 2700 | 0.35 | 145 | 725 | 65 | 75 | 90 | 4 |
| 2.49 | 3300 | 0.65 | 115 | 575 | 95 | 35 | 110 | 2 |
| 2.49 | 3300 | 0.55 | 135 | 375 | 95 | 35 | 110 | 3 |
| 1.99 | 3300 | 0.35 | 195 | 1475 | 85 | 75 | 65 | 2 |
| 1.99 | 3300 | 0.45 | 165 | 575 | 75 | 75 | 60 | 1 |
| 2.25 | 3100 | 0.65 | 155 | 975 | 95 | 25 | 65 | 1 |
| 2.25 | 3100 | 0.55 | 135 | 325 | 95 | 25 | 115 | 2 |
| 2.27 | 3300 | 0.65 | 225 | 475 | 95 | 85 | 80 | 4 |
| 2.55 | 3100 | 0.55 | 155 | 875 | 85 | 55 | 70 | 2 |
| 1.99 | 3700 | 0.35 | 75 | 525 | 35 | 85 | 105 | 4 |
| 1.77 | 4100 | 0.65 | 115 | 625 | 85 | 95 | 50 | 4 |
| 2.22 | 3900 | 0.75 | 125 | 1375 | 85 | 65 | 60 | 4 |
| 2.45 | 2700 | 0.35 | 85 | 1525 | 85 | 35 | 60 | 2 |
| 2.14 | 2200 | 0.25 | 145 | 175 | 75 | 55 | 100 | 2 |
| 2.18 | 2300 | 0.15 | 95 | 975 | 95 | 35 | 90 | 3 |
| 2.18 | 2900 | 0.05 | 55 | 1975 | 5 | | | 1 |
| 2.18 | 2700 | 0.45 | 195 | 925 | 85 | 125 | 70 | 2 |
| 2.18 | 3700 | 0.65 | 245 | 1225 | 95 | 155 | 80 | 2 |
| 2.18 | 2300 | 0.35 | 85 | 1275 | 85 | 65 | 85 | 2 |
| 2.18 | 2900 | 0.65 | 175 | 425 | 95 | 95 | 50 | 2 |
| 2.18 | 2900 | 0.75 | 115 | 925 | 85 | 75 | 110 | 3 |
| 3.94 | 3100 | 0.75 | 255 | 725 | 95 | 105 | 65 | 4 |
| **5.7** | 3300 | 0.55 | 215 | 625 | 95 | 105 | 115 | 3 |
| 2.18 | 4500 | 0.35 | 95 | 1725 | 75 | 95 | 79 | 1 |
| 2.18 | 3300 | 0.55 | 185 | 975 | 95 | 65 | 45 | 3 |
| 2.18 | 3300 | 0.45 | 165 | 875 | 95 | 290 | 80 | 1 |
| 0.54 | 3100 | 0.45 | 85 | 475 | 65 | 125 | 50 | 2 |
| 2.66 | 3300 | 0.85 | 195 | 825 | 95 | 35 | 93 | 4 |
| 2.18 | 2500 | 0.15 | 185 | 525 | 65 | 55 | 65 | 2 |
| 2.18 | 2500 | 0.05 | 55 | 425 | 55 | 55 | 60 | 3 |
| 3.2 | 2700 | 0.15 | 175 | 1275 | 55 | 125 | 70 | 3 |
| 2.18 | 2700 | 0.25 | 85 | 2425 | 85 | 35 | 75 | 3 |
| 2.2 | 3100 | 0.35 | 145 | 1375 | 85 | 25 | 90 | 3 |
| 1.89 | 3100 | 0.65 | 195 | 875 | 95 | 115 | 90 | 3 |
| 1.46 | 3300 | 0.55 | 165 | 375 | 85 | 75 | 70 | 3 |
| 2.61 | 3100 | 0.55 | 115 | 675 | 85 | 145 | 50 | 4 |
| 2.84 | 3100 | 0.55 | 95 | 575 | 85 | 75 | 90 | 2 |
| 1.7 | 3500 | 0.45 | 185 | 475 | 85 | 65 | 50 | 4 |
| 2.18 | 2700 | 0.35 | 105 | 150 | 85 | 55 | 90 | 2 |
| 2.26 | 3300 | 0.35 | 85 | 1425 | 75 | 65 | 70 | 3 |
| 3.33 | 3900 | 0.35 | 105 | 925 | 65 | 35 | 65 | 4 |
| 2.41 | 2700 | 0.25 | 115 | 1825 | 85 | 25 | 90 | 3 |
| 1.93 | 3700 | 0.35 | 135 | 1425 | 65 | 75 | 140 | 3 |
| 2.74 | 3700 | 0.45 | 195 | 1525 | 85 | 175 | 78 | 2 |
| 1.9 | 3500 | 0.55 | 195 | 1425 | 95 | 125 | 90 | 3 |
| 1.47 | 3100 | 0.25 | 115 | 925 | 95 | 15 | 100 | 3 |
| 1.77 | 2700 | 0.35 | 95 | 1375 | 85 | 55 | 90 | 2 |
| 1.84 | 2900 | 0.45 | 95 | 1125 | 95 | 15 | 70 | 1 |
| 2.68 | 2500 | 0.55 | 155 | 975 | 95 | 45 | 70 | 2 |
| 2.23 | 3500 | 0.75 | 105 | 325 | 95 | 75 | 70 | 4 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1.77 | 3500 | 0.35 | 165 | 825 | 95 | 45 | 105 | 2 |
| 1.77 | 2700 | 0.15 | 95 | 2225 | 75 | 45 | 80 | 1 |
| 1.77 | 2500 | 0.25 | 125 | 725 | 95 | 45 | 50 | 3 |
| 1.77 | 3700 | 0.55 | 125 | 925 | 85 | 145 | 100 | 2 |
| 1.77 | 3100 | 0.55 | 105 | 625 | 65 | 255 | 90 | 1 |
| 1.77 | 3100 | 0.55 | 85 | 625 | 95 | 55 | 80 | 1 |
| 2.45 | 3100 | 0.25 | 165 | 725 | 75 | 35 | 75 | 2 |
| 1.76 | 2900 | 0.25 | 85 | 975 | 95 | 15 | 80 | 1 |
| 1.99 | 2900 | 0.45 | 95 | 575 | 95 | 35 | 60 | 2 |
| 2.18 | 3100 | 0.45 | 195 | 1325 | 95 | 15 | 110 | 1 |
| 2.18 | 2300 | 0.45 | 185 | 275 | 85 | 165 | 100 | 3 |
| 2.18 | 2300 | 0.35 | 105 | 1575 | 85 | 45 | 100 | 2 |
| 2.18 | 2500 | 0.05 | 75 | 575 | 85 | 35 | 90 | 4 |
| 2.18 | 3100 | 0.65 | 165 | 575 | 55 | 115 | 70 | 2 |
| 2.18 | 2900 | 0.15 | 145 | 2625 | 95 | 75 | 80 | 2 |
| 2.18 | 3300 | 0.55 | 105 | 925 | 85 | 75 | 80 | 4 |
| 2.18 | 3300 | 0.75 | 155 | 775 | 95 | 75 | 60 | 1 |
| 2.18 | 3300 | 0.55 | 205 | 1075 | 95 | 55 | 155 | 1 |
| 2.18 | 2500 | 0.25 | 75 | 150 | 85 | 25 | 70 | 2 |
| 2.66 | 3300 | 0.65 | 145 | 1425 | 95 | 35 | 80 | 2 |
| 2.39 | 2300 | 0.35 | 125 | 575 | 85 | 55 | 85 | 2 |
| 0.848 | 2900 | 0.15 | 105 | 575 | 85 | 25 | 90 | 3 |
| 2.3 | 2700 | 0.25 | 135 | 975 | 95 | 35 | 90 | 1 |
| 2.18 | 3500 | 0.35 | 85 | 1425 | 85 | 55 | 80 | 2 |
| 2.05 | 2700 | 0.35 | 205 | 1325 | 85 | 75 | 100 | 2 |
| 2.14 | 2900 | 0.35 | 115 | 1025 | 85 | 35 | 110 | 2 |
| 2.41 | 2700 | 0.25 | 85 | 1325 | 95 | 25 | 80 | 4 |
| 2.68 | 2900 | 0.35 | 95 | 1675 | 85 | 25 | 100 | 2 |
| 2.66 | 3900 | 0.45 | 135 | 150 | 95 | 65 | 85 | 2 |
| 1.43 | 2300 | 0.45 | 125 | 525 | 30 | 25 | 90 | 1 |
| 2.6 | 2700 | 0.25 | 135 | 575 | 95 | 8 | 85 | 3 |
| 1.77 | 3300 | 0.85 | 65 | 475 | 75 | 75 | 117 | 4 |
| 2.12 | 2900 | 0.15 | 115 | 275 | 45 | 55 | 90 | 3 |
| 2.18 | 2900 | 0.35 | 115 | 1025 | 100 | 35 | 110 | 2 |
| 2.18 | 3100 | 0.45 | 205 | 825 | 95 | 55 | 131 | 4 |
| 2.18 | 2500 | 0.35 | 75 | 375 | 95 | 25 | 80 | 2 |
| 1.01 | 2500 | 0.15 | 115 | 825 | 85 | 15 | 80 | 2 |
| 2.44 | 2500 | 0.45 | 115 | 525 | 95 | 35 | 130 | 4 |
| 2.49 | 2900 | 0.55 | 125 | 975 | 95 | 25 | 55 | 3 |
| 0.55 | 3500 | 0.55 | 95 | 625 | 85 | 25 | 80 | 4 |
| 2.08 | 2300 | 0.25 | 175 | 325 | 65 | 195 | 80 | 3 |