

**UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ**

**Colegio de Ciencias e Ingenierías**

**Proyecto Integrador: Desarrollo de una plataforma  
para el Análisis de Causa Raíz a Través de la  
Extracción Automática de Datos en la industria de  
Petróleo y Gas.**

**Gabriel Alexander Pástor Villacís**

**Ingeniería en Ciencias de la Computación**

Trabajo de fin de carrera presentado como requisito  
para la obtención del título de  
Ingeniería en Ciencias de la Computación

Quito, 08 de octubre de 2023

**UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ**  
**Colegio de Ciencias e Ingenierías**

**HOJA DE CALIFICACIÓN**  
**DE TRABAJO DE FIN DE CARRERA**

**Proyecto Integrador: Desarrollo de una plataforma para Análisis de Causa Raíz, extracción y automatización de datos en la industria de petróleo y gas.**

**Gabriel Alexander Pástor Villacís**

**Nombre del profesor, Título académico**

**Daniel Fellig, MS,**  
**Ingeniería Eléctrica y Computacional**

Quito, 08 de octubre de 2023

## © DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombres y apellidos: Gabriel Alexander Pástor Villacís

Código: 00211253

Cédula de identidad: 1753755410

Lugar y fecha: Quito, 08 de octubre de 2023

## ACLARACIÓN PARA PUBLICACIÓN

**Nota:** El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en <http://bit.ly/COPETHeses>.

## UNPUBLISHED DOCUMENT

**Note:** The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around these publishing available on <http://bit.ly/COPETHeses>.

## RESUMEN

En la industria de petróleo y gas, así como en otras diversas esferas, se suscitan incidentes relacionados con la maquinaria en operación, los cuales pueden resultar en pérdidas significativas de días de trabajo y de barriles por día. Para abordar esta problemática, la implementación de informes de Análisis Causa Raíz (RCA por sus siglas en inglés) constituye una práctica para analizar causas subyacentes de tales incidentes con el propósito de orientar a decisiones futuras. No obstante, una de las facetas desafiantes y fundamentales del proceso radica en la recopilación y conservación de datos relacionados con los incidentes, particularmente debido al procedimiento manual aplicado para compilar la información pertinente en bases de datos. Ante esto, el presente documento se centra en el desarrollo de una plataforma web diseñada con la finalidad de automatizar la extracción de datos de informes de campo. Este proceso se lleva a cabo mediante parseo y reconocimiento óptico de caracteres (OCR). Adicionalmente, se incorpora una sección destinada a la visualización de los datos obtenidos, utilizando tablas y gráficos con el propósito de facilitar el desarrollo de informes de RCA, dentro del área de petróleo y gas.

**Palabras clave:** Extracción, automatización, eficiencia, almacenamiento, pulling, hallazgos, levantamiento artificial, producción, expresiones regulares, parseo, interfaz gráfica, visualizaciones.

## ABSTRACT

In the oil and gas industry, as well as in various other sectors, incidents related to operating machinery frequently occur, leading to significant losses in workdays and daily oil barrel production. To address this challenge, the implementation of Root Cause Analysis (RCA) reports designed to analyze the underlying causes of such incidents to guide future decisions is a common practice. However, one of the challenging and fundamental aspects of this process lies in the collection and preservation of data related to these incidents, particularly due to the manual procedures used to compile relevant information into databases. In response to this issue, this document focuses on the development of a web-based platform designed to automate the extraction of data from field reports. This process is accomplished through parsing and optical character recognition (OCR). Additionally, it incorporates a section dedicated to visualizing the obtained data using tables and charts to facilitate the creation of RCA reports within the field of oil and gas.

**Key words:** Extraction, automation, efficiency, storage, pulling, findings, artificial lift, production, regular expressions, parsing, GUI, visualizations.

## TABLA DE CONTENIDO

<b>1.</b>	<b>INTRODUCCIÓN .....</b>	<b>9</b>
<b>2.</b>	<b>ESTADO DEL ARTE .....</b>	<b>18</b>
<b>3.</b>	<b>DEFINICIÓN DE PROPUESTA.....</b>	<b>22</b>
<b>4.</b>	<b>DISEÑO DEL SISTEMA.....</b>	<b>30</b>
<b>5.</b>	<b>Desarrollo de prototipo.....</b>	<b>50</b>
<b>6.</b>	<b>evaluación y validación .....</b>	<b>70</b>
<b>7.</b>	<b>Conclusiones .....</b>	<b>75</b>
<b>8.</b>	<b>Referencias bibliográficas .....</b>	<b>82</b>

## ÍNDICE DE FIGURAS

<b>Figura 1</b>	<b>Arquitectura del sistema .....</b>	<b>26</b>
<b>Figura 2</b>	<b>Diagrama de casos de uso .....</b>	<b>31</b>
<b>Figura 3</b>	<b>Diagrama de registro.....</b>	<b>32</b>
<b>Figura 4</b>	<b>Diagrama de inicio de sesión .....</b>	<b>33</b>
<b>Figura 5</b>	<b>Diagrama de cierre de sesión .....</b>	<b>34</b>
<b>Figura 6</b>	<b>Diagrama de escaneo de PDF .....</b>	<b>36</b>
<b>Figura 7</b>	<b>Diagrama de query BD .....</b>	<b>37</b>
<b>Figura 8</b>	<b>Diagrama de presentar tablas .....</b>	<b>38</b>
<b>Figura 8</b>	<b>Diagrama de presentar gráficos.....</b>	<b>39</b>
<b>Figura 9</b>	<b>Diagrama de extraer tablas .....</b>	<b>40</b>
<b>Figura 9</b>	<b>Diagrama de procesar PDF .....</b>	<b>41</b>
<b>Figura 10</b>	<b>Diagrama de actividad de registro .....</b>	<b>42</b>
<b>Figura 11</b>	<b>Diagrama de actividad de logeo .....</b>	<b>43</b>
<b>Figura 12</b>	<b>Diagrama de actividad de escaneo .....</b>	<b>45</b>
<b>Figura 13</b>	<b>Diagrama de actividad de tablas .....</b>	<b>46</b>
<b>Figura 14</b>	<b>Diagrama de actividad de gráficos .....</b>	<b>47</b>
<b>Figura 15</b>	<b>Diagrama de actividad de procesamiento de PDF.....</b>	<b>48</b>
<b>Figura 16</b>	<b>Diagrama de secuencia de registro .....</b>	<b>¡Error! Marcador no definido.</b>
<b>Figura 17</b>	<b>Diagrama de secuencia de logeo .....</b>	<b>¡Error! Marcador no definido.</b>
<b>Figura 18</b>	<b>Diagrama de secuencia de escaneo .....</b>	<b>¡Error! Marcador no definido.</b>

<b>Figura 19 Diagrama de secuencia de tablas .....</b>	<b>¡Error! Marcador no definido.</b>
<b>Figura 20 Diagrama de secuencia de gráficos .....</b>	<b>¡Error! Marcador no definido.</b>
<b>Figura 21 Diagrama de secuencia de procesamiento de PDF.....</b>	<b>¡Error! Marcador no definido.</b>
<b>Figura 22 Diagrama de clases .....</b>	<b>49</b>
<b>Figura 23 Resultados de extracción, con coordenadas en Firebase (v1) .....</b>	<b>¡Error! Marcador no definido.</b>
<b>Figura 24 Interfaz principal de streamlit canvas (v1) .....</b>	<b>53</b>
<b>Figura 25 Vista previa de la extracción de datos con canvas (v1) .....</b>	<b>54</b>
<b>Figura 26 Ruta y contenido almacenado en Firebase (v1) .</b>	<b>¡Error! Marcador no definido.</b>
<b>Figura 27 Almacenamiento de datos extraídos de PDF (v1).....</b>	<b>58</b>
<b>Figura 28 Obtención de mail bajo el asunto ‘mailer’ (v1).....</b>	<b>62</b>
<b>Figura 29 Correo recibido con el asunto ‘mailer’ (v1) .....</b>	<b>62</b>
<b>Figura 30 Presentación de tablas (v2) .....</b>	<b>63</b>
<b>Figura 31 Presentación de tabla general y tabla de bombas, en sección de prueba (v2).....</b>	<b>64</b>
<b>Figura 32 Vista previa en extracción de texto (v2) .....</b>	<b>¡Error! Marcador no definido.</b>
<b>Figura 33 Página inicial (v2) .....</b>	<b>67</b>
<b>Figura 34 Página de registro (v2) .....</b>	<b>67</b>
<b>Figura 35 Notificación de correo duplicado (v2).....</b>	<b>68</b>
<b>Figura 36 Página de inicio de sesión (v2) .....</b>	<b>68</b>



## 1. INTRODUCCIÓN

### 1.1. Producción petrolera y la continua extracción de crudo.

La explotación de recursos es un tema de mucha relevancia en algunos países del mundo, dado que el petróleo y gas es uno de los recursos más abundantes y de los que pueden generar mayor cantidad de ingreso a un país por exportaciones.

En cierto grado, la industria de petróleo y gas (o también considerada de hidrocarburos), tiene una importante participación en el Ecuador, considerando que ha dado cerca de un 13% promedio al PIB. De esta manera, se puede considerar que el Producto Interno Bruto, que es una medida para evaluar la salud económica del país, ha experimentado una notable influencia de la industria de petróleo y gas. Y es gracias a esto que esta área sigue siendo clave en el desarrollo económico del Ecuador (Chacon Cruz & Riaño Amaya, 2020, p. 30). De este modo, se considera un área de mucho interés y que motiva al aprovechamiento de los recursos de yacimientos en el país.

A pesar de ello, puede ocurrir que un yacimiento con el tiempo vaya perdiendo su energía natural, y de ese modo también la presión del mismo se vea afectada (Del Pezo Yagual, 2021, p. 1). Es ante estas circunstancias que se considera implementar medios que permitan incrementar la presión del pozo, especialmente pozos maduros, con el fin de alargar la vida útil de estos y que sigan operando (p. 2). Es por ello que se aplica el levantamiento artificial, con el fin de poder extraer el crudo dentro de un pozo hacia la superficie.

Cuando un equipo de levantamiento artificial es usado para continuar con el ciclo de vida de un pozo o yacimiento petrolero, se debe tomar en cuenta el nivel de corrosión y

mantenimiento constante que recibe cada parte del sistema de bombeo que lo conforma. Pues es debido a una mala gestión en mantenimiento, ya sea por un chequeo del estado del equipo en largos periodos de tiempo, o por condiciones internas en el yacimiento, ya sea por la presión de gas que debilita al sistema de protección de un motor insertado a profundidad, por ejemplo, que se tiene posteriormente fallas en la estructura del sistema de levantamiento artificial. Entre las más comunes se puede considerar fallas en el sistema dada la corrosión, fallas mecánicas, fallas externas al equipo, etc. De este modo, y también fuera de la cuestión que aborda el levantamiento artificial, las fallas que ocurren en este campo profesional pueden llegar a ser más graves dependiendo las métricas de operaciones y los cuidados aplicados.

## **1.2.Cuestión de fallas en el área de petróleo y gas.**

Durante las actividades realizadas en el campo petrolero, como en cualquier otro campo, se pueden dar fallas en el equipo de operaciones durante las actividades de producción. Esto puede significar que se sufran fallos de diferentes tipos: corrosión en la capa protectora del motor, la bomba o motor se sobre exigen demasiado al momento de generar presión en el yacimiento, la posibilidad que ocurra una descarga eléctrica, una perforación en el sistema de tuberías o las paredes de la bomba por corrosión, etc. Aunque solo se traten de fallas mecánicas o de corrosión generalmente, siempre terminan generando pérdidas, sean estas desde humanas hasta económicas (Castro-Castro & Cendales-Ladino, 2019, Introducción, párr. 1), y es debido a esto que, ya desde el siglo XX, se ha recurrido a entender cuáles pueden ser las razones de un evento que resulte catastrófico o negativo, con el fin de no volver a cometer dichos errores a futuro o al menos

para mitigar el riesgo de formas alternativas (Castro-Castro & Cendales-Ladino, 2019, Análisis de causa raíz, párr. 1-3).

En el país, la industria petrolera lleva sus operaciones en diferentes puntos, y en ciertos casos se hace mención de derrames de hidrocarburos debido a problemas en la maquinaria. Según el estudio (Mayorga-Mayorga & Reyes-Bueno, 2022), el caso del campo Ancón, en el bloque Gustavo Galindo Velasco, tuvieron esta problemática principalmente por la falta de mantenimiento y procedimientos manuales obsoletos (Introducción, párr. 6). En consecuencia, se han registrado un total de 297 derrames ocurridos entre 2014 y 2018, siendo 2015 el año con más derrames, siendo un total de 70 ese año (Derrames de petróleo ocurridos en el campo Ancón, párr. 1-3).

Dada esta cuestión, cuando se estudia por qué algo falla en la extracción de petróleo, hay un enfoque principalmente en entender cómo funcionan las máquinas que se usan. Con el tiempo, los pozos pueden perder energía o presión, por eso es importante usar métodos para impulsar la extracción. Sin embargo, dichos métodos y equipos implementados siempre se encuentran expuestos a fallos en operaciones de producción o extracción de líquido de un yacimiento, y conocer las causas que provocan dichas fallas es importante para mitigar estas causas a futuro. El análisis causa raíz es una forma de averiguar exactamente las causas de los problemas en esta parte de la producción de petróleo, y determinar cambios en la producción y procesos de mantenimiento y operación de la maquinaria para reducir los fallos a futuro.

### **1.3. Análisis de causa raíz en el área de petróleo y gas.**

Gracias a la implementación del análisis de causa raíz, se puede verificar un conjunto de problemáticas que afectan a la actividad regular en el área petrolera. Un caso

considerable está en el estudio sobre un incidente de tubería ASTM A53 (Khattak et al., 2016), de la cual se pudo determinar que la falla en tubería de esta se debía a una fuga de gas, la cual fue determinada por una grieta dada por la pérdida de espesor causada por un exceso de pulido del material (Background of the incident, párr. 1). Ante esto, se tomó la medida de incrementar el nivel de inspección en los equipos utilizados para la extracción y transporte (Conclusion and recommendations, párr. 1-2).

A pesar de ello, para extraer la información es necesario el uso de diferentes fuentes, que puedan ir desde encuestas hasta reportes internos de la empresa, aunque en ocasiones la información es de acceso restringido y las instalaciones no tiene permitido el acceso libre a cualquier persona (de la Cruz Ventura, 2019, pp. 8-10). En el caso del área de petróleo y gas, se pueden determinar diferentes tipos de reportes, dadas las actividades realizadas en el sector. Dichas actividades implican la extracción del equipo de operaciones en el pozo o yacimiento, un estudio a profundidad de la maquinaria y descripción de hallazgos encontrados, pruebas de producción y simulación del rendimiento del equipo implementado en diferentes condiciones.

#### **1.4. Levantamiento artificial y las condiciones del equipo en operación.**

Al hablar del levantamiento artificial, es necesario comprender de este que se implementa dados los casos donde las condiciones naturales de un pozo se han visto afectadas con el paso del tiempo. Esto implica, principalmente, que dado un pozo o yacimiento que ya haya tenido un tiempo de producción, puede irse perdiendo la presión y la energía implementada para subir desde el subsuelo a la superficie el líquido para ser procesado. Es por ello que los sistemas de levantamiento artificial se aplican, con el fin de

proporcionar energía al pozo y que de este se pueda seguir extrayendo crudo y mantener un nivel aceptable de producción (Lagla Paneluisa, 2023, p. 4).

En si el levantamiento artificial no se define como un proceso y maquinaria que es único, sino que puede diferenciarse principalmente por el medio que se utiliza para ya sea ejercer presión o extraer el líquido o crudo hacia la superficie. Entre los más considerados se tiene el sistema de Bombeo Electro Sumergible (BES). Dicho sistema comprende de una bomba y motor incorporados con un sistema de cableado eléctrico, de modo que la bomba se suspende hacia el fondo del pozo o yacimiento, implementando el motor para su accionar, que se conecta a un sistema de energía eléctrica, transformando la energía recibida en torque. En si este sistema puede generar en promedio 10.000 barriles por día (BPD), haciéndolo uno de los sistemas más efectivos actualmente (p. 5)

Así como se tiene al sistema BES (Bombeo Electro Sumergible), existen otros métodos de levantamiento artificial, tales como levantamiento por gas (Gas Lift), que implementa gas comprimido para aumentar la presión y así subir el crudo a la superficie. Otro es el sistema por bombeo mecánico, que se compone de una manivela que se mueve constantemente, ejerciendo presión hacia el fondo del pozo, y siendo uno de los métodos más antiguos aplicados. Finalmente, se aplica el bombeo hidráulico, que aplica un líquido para aumentar la presión en el interior.

Dentro del área de petróleo y gas, puede haber diferentes tipos de reportes que se implementan para explicar el estado de las actividades, entre estos, se pueden considerar los siguientes: Informe de Evaluación de Riesgos y Mitigación (para identificar riesgos en las operaciones petroleras), Informe de Pruebas de Equipos (documentación de pruebas y análisis en equipos petroleros), Informe de Mantenimiento Correctivo (que contiene

acciones y medidas tomadas para corregir una falla), etc. En si los reportes para hablar de fallas o paro de actividades en un yacimiento o pozo petrolero, para levantamiento artificial, serían: DIFA, Pull, Tear Down y Matching.

#### **1.4.1. DIFA**

En el caso de DIFA (Dismantle, Inspection and Failure Analysis), se trata de un reporte específicamente para sistemas de bombeo electro sumergible. Dentro de este se toma en cuenta no solo los equipos utilizados en la extracción de crudo hacia la superficie, sino que realiza un análisis enfocado en entender los antecedentes y eventos posteriores tras el paro de actividades en un pozo o yacimiento petrolero (Karnik et al., 2021).

Es por eso que este reporte, se vuelve fundamental para la detección de fallas, y también en otros aspectos:

- Determinación de los eventos más importantes tras el paro de actividades.
- Determinación de los antecedentes de un pozo y verificar que condiciones o problemas se vieron al momento de operar.
- Determinar el rendimiento del pozo durante el periodo de actividades regulares.

#### **1.4.2. Pull**

El proceso de Pulling se involucra con el fin de extraer el equipo del fondo del pozo, con el fin de realizar mantenimiento y continuar así de manera efectiva con las operaciones (Figueroa, 2013, Descripción general, párr. 1). Aunque este proceso de extracción de equipo afuera de un pozo suela ser muy normal dentro del área de petróleo y gas, eso no implica que el proceso de pulling no se aplique cuando haya un paro en las actividades dado un incidente o inconveniente con las partes involucradas en la producción, y especialmente con el equipo de levantamiento artificial.

Dadas esas circunstancias, un reporte de Pull se involucra con el fin de obtener detalles y observaciones a primera vista del equipo. Dados esos puntos, se pueden obtener tablas de un reporte de pull sobre cada componente, como sería: cabeza de descarga, bomba, motor, cables, sensor, intake, protector y observaciones. De igual forma, se pueden considerar otros atributos, aunque estos dependerán de la empresa en cuestión.

#### ***1.4.3. Tear Down***

El reporte de Tear Down trata los hallazgos encontrados en maquinaria del equipo de levantamiento artificial, tras realizarse el pulling (Trujillo Coral, 2018, p. 69). Esta técnica se utiliza comúnmente en la industria manufacturera y en la ingeniería para entender las condiciones en las que se encuentra actualmente y en las que estaba operando el equipo, cuando este se encontraba instalado y sus componentes en el fondo del yacimiento. Algunas razones para la importancia de este tipo de informe en RCA son:

- **Identificación de Fallas:** El desmontaje permite la identificación de posibles fallas o defectos en los componentes internos de un sistema. Esto es esencial para determinar si una falla se debió a un problema en uno de los componentes internos.
- **Análisis de Diseño:** El Tear Down proporciona información valiosa sobre el diseño y la calidad de los componentes. Esto es útil para evaluar si un diseño inadecuado contribuyó al problema.
- **Muestra de Evidencia:** Los resultados del desmontaje se pueden utilizar como evidencia en un proceso de RCA, respaldando las conclusiones y recomendaciones.

#### ***1.4.4. Matching history***

Matching history implica la recreación de una situación con un componente de producción, implementando el historial de producción de un equipo dado después de parar

las actividades. Este tipo de informe es valioso para comprender el rendimiento que tendría un equipo en ciertas condiciones de trabajo, que implica trabajar a diferentes frecuencias, proporcionadas por los componentes eléctricos, y así verificar si la eficiencia de los componentes se mantiene o no. Las razones para la importancia de este tipo de informe en RCA son las siguientes:

- **Reconstrucción de Eventos:** El Levantamiento Artificial permite reconstruir un evento o situación, lo que es esencial para comprender cómo se desarrolló un problema en particular.
- **Datos Contextuales:** Se considera el estado más reciente en el que se trabaja con un motor o bomba, dado que son los componentes más importantes en el equipo de levantamiento artificial, o al menos en el equipo BES (Bombeo Electro Sumergible).
- **Validación de Hipótesis:** A través del Levantamiento Artificial, es posible probar y validar hipótesis sobre cómo ocurrió un problema y cuáles fueron las causas subyacentes.
- **Modelado de Escenarios:** Los datos generados pueden utilizarse para modelar diferentes escenarios y evaluar cómo podrían haber contribuido al problema.

En resumen, tanto los reportes de Tear Down como los de Levantamiento Artificial son herramientas esenciales para el proceso de RCA. Mientras que el Tear Down proporciona una comprensión profunda de los componentes internos y las posibles fallas de un sistema, el Levantamiento Artificial permite reconstruir eventos y situaciones para identificar las causas raíz. La combinación de estos dos enfoques brinda una visión completa y sólida para abordar los problemas y tomar medidas preventivas efectivas.



### **1.5.Recolección de datos**

De cierta forma, la recolección de datos es importante para poder definir qué tipo de antecedentes y posibles causas se tengan ante un problema en levantamiento artificial. De esta forma, se considera que cada reporte de fallas o paro de actividades pueda dar una idea fundamental de lo sucedido en un yacimiento o pozo petrolero. Es por ello que la recopilación y extracción de datos importantes es clave para poder representar de buena forma la información para reportes de RCA, y así se tomen decisiones acertadas con el fin de reducir el daño a futuro y mitigar riesgos.

Uno de los medios por los cuales fluye la información es la red, especialmente el correo electrónico, que es un medio por donde se envían reportes de fallas al personal especializado de una empresa. Es en este punto donde la recolección de datos se vuelve una tarea desafiante, y mucho más cuando se realiza de forma manual, lo que hace que sea necesario optimizar el proceso de obtención de datos para evitar el error humano.

### **1.6.Software en el área de petróleo y gas.**

En tanto al campo de software, la implementación de este en la industria petrolera ya tiene un buen tiempo implementándose con ciertos proyectos. Tal como sería el caso del desarrollo de una aplicación de capacitación en el control de pozos petroleros, implementando Visual Basic (Jiménez Moreno, Hernández Barajas, Jiménez Hernández & Plazas Quiroga, 2022). Para estos casos, se considera una tarea compleja el desarrollo de software, aunque en su caso implicaba una aplicación con fines educativos, se toma muy en cuenta el conocimiento adquirido por actividades en el programador o desarrollador del sistema (pp. 4-5). De la misma forma, se han implementado proyectos enfocados en generar software para analizar registros de producción con el mismo entorno (Escobar,

Caviedes Ramírez & Enciso, 2010), considerando una etapa de recolección de datos más la aplicación de lenguaje de programación gráfico en Visual Basic (p. 94).

A pesar de la existencia de diferentes proyectos de software, la implementación de estos es diferente a la que buscaría realizar, que implica el uso de reportes internos, de los cuales se extraen datos que luego servirán para la generación de reportes generales con información cuantitativa y cualitativa para estudiar un pozo. Gracias a esto, se planea un proyecto que se enfoque específicamente en la muestra de datos relacionados a la actividad de campo, y con la finalidad de analizar el rendimiento de la maquinaria implementada.

## **2. ESTADO DEL ARTE**

### **2.1. Parseo y su implementación**

La implementación de algoritmos de parseo ya tiene un buen tiempo de desarrollo, de modo que su implementación se ha dado en diferentes áreas, además que ya se puede considerar como una función integrada de ciertas herramientas. Por otro lado, su vínculo con ciencias de la computación puede radicar en la inteligencia artificial. Un ejemplo de ello sería un software para generar datos de entrenamiento para un sistema de IA y que este analice documentos y los convierta en archivos JSON (Norsworthy & NAVAL RESEARCH LAB WASHINGTON DC, 2022). En si la propuesta sirve para entrenar una inteligencia artificial en el proceso de parseo, sin embargo, el objetivo del proyecto radica en la implementación de datos sintéticos para entrenamiento como producto de la programación en Java y Python (pp. 1-3).

En un inicio, las expresiones regulares toman una importancia significativa en este estudio. Dicho componente implica una herramienta versátil en el procesamiento de texto, además de filtrar o extraer porciones a nivel abstracto desde caracteres alfabéticos hasta puntuaciones. En este caso, Python tiene una implementación clave, siendo la librería estándar “re” (Agarwal, n.d.), considerando la similitud en el lenguaje con Perl que tiene su destreza en el procesado de texto (pp. 7-10). Es por ello que Python puede considerarse como la herramienta adecuada para la implementación de un sistema de parseo y detección de expresiones reglares, principalmente la segunda para realizar una limpieza sobre el texto que se extrae de un documento que se quiera procesar, y así solamente obtener las expresiones o el conjunto de texto objetivo.

## **2.2.OCR y su importancia en la obtención de datos.**

El funcionamiento de un sistema de parseo, que implique dicha extracción de datos de documentos tiene varias áreas de implementación, tal como sería el sistema de extracción de datos por medio de la inteligencia artificial ya entrenada, eliminando así el trabajo tedioso de extracción de datos de forma manual (Bhatt, 2022, p. 1365), siendo prioritariamente en documentos estructurados donde se puede ver un mejor manejo de la información, siendo un ejemplo el de facturas, el dominio del Reconocimiento Óptico de Caracteres (OCR) sobre estas (p. 1366).

La implementación del OCR se puede encontrar también en proyectos de ámbitos diferentes, como en la extracción de datos de directorios de información de estaciones de gas, con el fin de obtener sus datos y geo codificarles para obtener la latitud y altitud de varios negocios de estos (Bell et al., 2020, p. 2). Para este tipo de proyectos se

implementaron datos de la biblioteca de Boston y de Providence, siendo entre 1936 y 1980 (p. 2), consiguiendo una tasa de éxito del 94.4% (pp. 6-7).

A pesar de ello, un sistema de OCR podría tener dificultades aún al implementarse sobre documentos complejos. Al hacer mención de esto, se hace referencia a documentos con caracteres tan minúsculos que dificultarían la buena percepción de la herramienta de OCR en cuestión a aplicarse. No se puede negar que su aplicación es efectiva, sin embargo, en ocasiones puede no tener los resultados esperados, y especialmente dado el hecho de que los caracteres más pequeños se vuelven un desafío grande para su implementación, además que estos siempre pueden estar presentes en reportes de todo tipo.

### **2.3.Implementaciones en la gestión de datos.**

Para el análisis de datos, ya hay implementaciones significativas en software, y en ciertos casos para la recolección de datos. En cuanto a la definición de la muestra, un estudio Giraldo Ramírez, Álvarez Cadavid & Navarro Plazas, (2020) considera el uso de las tecnologías de la información y comunicación en “Selección de casos a través de métodos mixtos (apoyado en software estadístico)”. En si se consideran al menos a SurveyMonkey y Google Drive entre las mencionadas (Fase del diseño de instrumentos y recolección de información, párr. 1), como también el “Uso de representaciones gráficas para exploración y presentación de información”, además que con esto se “ha permitido no solo incorporar sino optimizar la visualización de los datos” (párr. 2). Implementaciones para generar software que ayude a crear reportes ya tienen su aparición varios años atrás. Un ejemplar (Zumba Rosero, 2014) muestra el desarrollo de software web para la generación de reportes, implementando SQL, JavaScript y Dreamweaver principalmente. Sin embargo, este sistema se basa principalmente en la gestión de datos y que contenga

una base de datos sólida (pp. 52-57), es decir, no se involucra la recolección de datos para su almacenamiento y recuperación posteriormente.

#### **2.4. Power Bi y Power Automate en la gestión de la información**

Actualmente se considera a Power BI como una herramienta de apoyo en la generación de reportes, la cual se enfoca en el diseño de reportes y la presentación de la información, además de elementos interactivos para mostrar datos según quiera el usuario. De igual forma, se puede integrar otras herramientas de trabajo, como también bases de datos SQL (maggiesMSFT, n.d.). Sin embargo, su implementación es general y solamente enfocada en la presentación de la información, siendo este un elemento que integra a la plataforma como tal, de esta manera no tiene un enfoque profundo en la representación de datos relacionados a fallas en equipos de levantamiento artificial, a pesar de que puede servir de apoyo para ello.

Por otra parte, Power Automate entra en el proceso de generación de flujos, automatizando procesos repetitivos o de mucho tiempo de ejecución (Georgiostrantzias, n.d.), aunque esto implica que el funcionamiento del flujo es determinado por el usuario y no definido de forma automática por el programa, de modo que el usuario por su cuenta puede explorar diferentes formas de crear operaciones a realizar, aunque esto implique tiempos de diseño y prueba de dichos flujos de información Power, Apps & Automate, 2020, pp. 1-2). Adicionalmente, se requiere de un flujo ya probado y funcionando correctamente, y cuando se involucra Power Automate el riesgo que se corre radica principalmente en el usuario, con la posibilidad de que ocurran fallos en la optimización de actividades. Es por ello que se requiere hacer cambios constantes con base en las actualizaciones de Office 365.

## **2.5.Otras implementaciones.**

Las herramientas de análisis de causa raíz pueden ser muchas en el mercado, tales como Tableau que implementa su funcionalidad en que el usuario decida qué información analizar y conectar diferentes fuentes de datos (Batt, Grealis, Harmon & Tomolonis, 2020, p. 322), aunque no es gratuita y su enfoque se basa más en el análisis de datos. Por otro lado, se tiene Minitab que se encarga del análisis estadístico de la información (Lesik, 2018, pp. 10-11), pero no se evidencia dentro de si la implementación de algoritmos de parseo para extraer datos de otros reportes.

Es debido a estas consideraciones que se pueden explorar diferentes avances en cada aspecto que involucrará el software a diseñar. De esta forma, la importancia de este proyecto se encamina en la integración de componentes de parseo, la automatización de actividades de extracción y almacenamiento de datos, y la representación de la información almacenada. Todo esto, enfocado en el área de petróleo y gas, especialmente sobre fallas en levantamiento artificial. Cabe mencionar que dentro del área de petróleo y gas no se han diseñado aplicaciones enfocadas en este ámbito o no han tenido tal relevancia que se puedan automatizar los procesos de análisis de causa raíz en pozos petroleros.

## **3. DEFINICIÓN DE PROPUESTA**

### **3.1.Descripción del proyecto aplicado.**

El proyecto en cuestión consiste de un sistema integrado, el cual se compone de una página web y un flujo automatizado, para brindar apoyo en la recolección y representación de información para reportes de RCA.

La página web se compone de tres partes fundamentales: una sección para definir los parámetros de extracción por parseo, una sección para presentar la información extraída hasta el momento por tablas, y una sección de dashboards donde se presenta la información más relevante almacenada.

En la sección de definición de parámetros de extracción, se considerará cuatro opciones, correspondientes a los reportes de fallas mencionados anteriormente en levantamiento artificial específicamente, y para cada sección el usuario define un asunto de correo el cual servirá para realizar un chequeo en su bandeja de entrada, y así inspeccionar los correos de interés. Dada una de las opciones, el usuario accederá a dos secciones, una después de otra, e ingresando primeramente un ejemplar del reporte en PDF: definición de parámetros de extracción de texto y una sección para los parámetros de las tablas. Es en estas secciones donde el usuario hace selección de las áreas donde se encuentra campos de texto y tablas relevantes respectivamente.

Por otro lado, la sección de presentación de datos extraídos será el espacio donde el usuario podrá revisar la información extraída y almacenada en la base de datos por el accionar del flujo automatizado, que se compondrá de subsecciones para cada tipo de reporte considerado, es decir, para DIFA, Tear Down, Matching history y Pull. Dicha sección implementa pandas con el fin de obtener la información, y representarla como DataFrame cada tabla.

Seguido de ello, se tiene la sección de dashboard de información relevante, en la cual se representan los datos más relevantes y de apoyo al usuario, para la determinación de causas y datos relevantes, que sirvan de apoyo en la creación de reportes de RCA. Para esto, se considera en esta sección presentar: los datos del equipo mencionado y detallado

en reportes de Pull; tablas de antecedentes, de eventos y gráficos lineales de producción de reportes de DIFA (específicamente para equipo de bombeo electro sumergible – BES); tabla compiladas de hallazgos de cada componente del equipo de levantamiento artificial, que se aplica en reportes de Tear Down; y por último gráficos de eficiencia y rendimiento de motor y bomba, a diferentes frecuencias simuladas, de los reportes de Matching history.

Finalmente, y como elemento clave para el buen funcionamiento de todo el sistema, se comprende de un flujo automatizado, el cual implica el monitoreo de la base de datos por nuevos parámetros de extracción. Esto en caso de que haya nuevos parámetros, iniciará el chequeo de correo electrónico de la cuenta del usuario registrado, implementando IMAP v4 principalmente. Este proceso se hará en paralelo para cada usuario registrado y por asuntos, los cuales el usuario definió en la sección de parámetros de extracción. EN caso de que un correo con asunto de interés llegue, se procederá a extraer la información relevante, con base en los parámetros que el mismo usuario definió, y así la información recolectada se guarda en una base de datos.

### **3.2.Objetivos.**

Determinados los puntos anteriores, se definen los siguientes objetivos para el proyecto en cuestión.

#### **3.2.1. General**

Crear un sistema integrado, de una página web y un flujo automatizado, junto con una base de datos e implementaciones en la extracción de datos, para el procesamiento de reportes por correo, su extracción, almacenamiento y representación de datos importantes para ayudar en la generación de reportes de RCA.

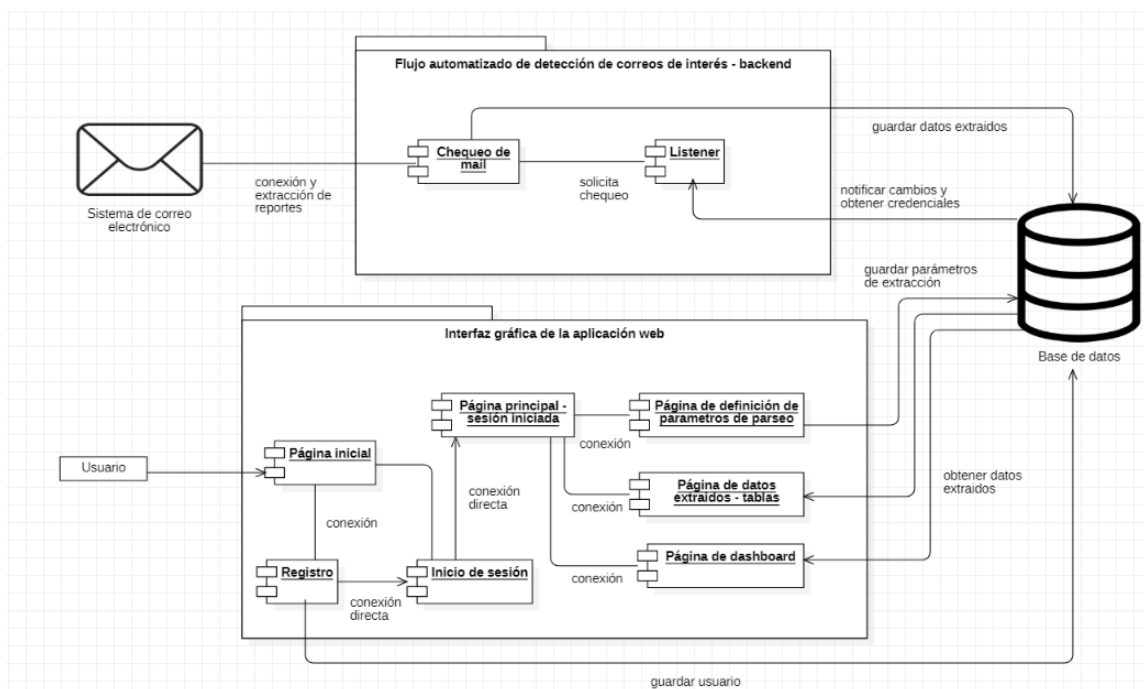
#### **3.2.2. Específicos**



- Investigar y seleccionar herramientas e implementaciones para la extracción de datos por parseo.
- Determinar los componentes más importantes, por medio de diagramas y casos de uso, para ser aplicados en la creación de una página web para visualizar datos extraídos de reportes de fallas.
- Desarrollar un flujo automatizado, generando una conexión por correo y a una base de datos, para la detección de archivos/reportes de fallas y la posterior extracción y almacenamiento de datos.
- Determinar los atributos más importantes, dentro de los reportes de fallas, para la creación de un dashboard para presentar la información recolectada por el flujo automatizado. Arquitectura del proyecto.

### **3.3.Arquitectura del sistema**

En cuanto a la arquitectura del sistema, se debe tomar en cuenta al menos tres componentes que están separados, que serían la aplicación web, el proceso automatizado con mail y la base de datos. Cada uno de dichos componentes con su respectiva relación entre elementos, se tiene entonces la figura a continuación.



*Figura 1 Arquitectura del sistema*

Por medio de la figura mostrada, se piensa aplicar un sistema de logeo al inicio cuando el usuario quiere acceder a la página. De esta forma, se piensa dar acceso restringido a un grupo de personas específicas de una empresa dada. Esta implementación es clave ya que la información a recolectar es confidencial de un área específica, y se busca que los datos sean manejados por el personal escogido y especializado.

De igual forma, cuando el usuario inicia sesión, se puede encontrar con tres secciones o pantallas disponibles, ya mencionadas previamente para: definir los parámetros de extracción de datos, visualización de la información extraída y para representar la información más relevante. Cabe señalar que la segunda y tercera sección se diferencian en la cantidad y tipo de información a presentar, puesto que en la segunda se presentan todos los datos extraídos y almacenados en la base de datos, mientras que la tercera se enfoca más en mostrar los datos más relevantes almacenados, y filtrando la información

por pozo (esto implica que en la sección de dashboards, que es la tercera sección, se mostrará información por pozo y no de todo lo extraído).

### **3.4. Selección de tecnologías y herramientas.**

La implementación de un sistema integrado podría aplicarse usando diferentes lenguajes, y claramente uno a la vez. Las implementaciones que puede tener Python dan mayor capacidad de realizar un trabajo que suene más efectivo y que comprenda una variedad de componentes, en ocasiones implementando código con menor cantidad de líneas a escribir. Es por ello que para esta aplicación se tomará a Python como lenguaje de programación para su escritura y desarrollo, además de librerías y otras implementaciones de apoyo para su ejecución y funcionamiento.

#### ***3.4.1. Parseo y extracción de texto y tablas***

En este punto, el parseo puede ser uno de los elementos más importantes a tomar en cuenta, y para ello se requiere analizar formas en las cuales se puede pasar el texto de una PDF a texto en cadena, además de que en ocasiones el resultado puede estar lleno de segmentos de información irrelevante o patrones innecesarios. Ante ello, se implementa RegEx (*re*) y Tabula (*tabula-py*) para la extracción de texto de un fichero en PDF.

Tabula, dada esta situación, es una herramienta enfocada en la extracción de tablas de archivos PDF, la cual permite definir que tablas y datos se deben extraer de un documento seleccionando el área de interés dentro de cada página. Sucede que existe un wrapper para Python de esta aplicación (*tabula-py*), que encapsula los métodos más importantes para la extracción de tablas, la cual trabaja de igual forma con parámetros de área y pagina, requiriendo también la ruta en la que se ubique el archivo para la extracción (Tabula — Tabula-py Documentation, n.d.).

Por otro lado, RegEx (*re*) es una librería de Python, aplicada para trabajar con expresiones regulares. La idea de implementar tanto *tabula* como RegEx (*re*) recae sobre la extracción de campos de texto más que sobre la extracción de tablas. Siendo así importante para determinar un conjunto de datos a tomar de una selección o área determinada de un PDF. Es así como se podría considera usar *tabula* para extraer en formato de tabla un capo de texto, luego pasarlo a cadena de caracteres, y por medio de expresiones regulares, ir eliminando estas para así solo tener el texto deseado.

#### ***3.4.2. Acceso al correo electrónico***

Para acceder al correo es necesario aplicar el protocolo IMAP, en este caso v4, con el fin de poder no solo acceder al correo por medio de credenciales, sino también para poder acceder a mensajes y recuperar información de estos de forma remota, y sin la necesidad de almacenamiento en el dispositivo local directamente, sino que se accede a los datos de forma remota. Por otro lado, existen implementaciones específicas para ciertos servicios de correo electrónico o proveedores de este, y es así como se inserta a discusión Microsoft Exchange, siendo esta una plataforma de servidor de correo electrónico.

IMAP como tal puede no ser un problema en Python, dada la implementación de la librería *imaplib*, la cual será de suma importancia para poder acceder a la bandeja de entrada de un usuario que haya ingresado las credenciales. Dado que el sistema de correos está presente en este proyecto, al menos para ser revisado y verificando correos electrónicos de interés por asunto determinado, se requiere de la aplicación de *imaplib* y *email*. La librería *email* será de ayuda para poder acceder al contenido de un correo, y así poder extraer un archivo adjunto, ósea un reporte para su extracción (*Imaplib — Protocolo Del Cliente IMAP4*, n.d.).

Sin embargo, sucede que muchos de los usuarios que podría usar este tipo de aplicación, o que al menos requerirían de un sistema de apoyo en estas circunstancias, usarían un servicio de correo de Microsoft, ósea Outlook, es por ello que se considera necesaria una interacción entre Python y el servidor de correo ya mencionado (Microsoft Exchange), y una idea clave aquí sería *exchangelib*. Esta librería es una implementación para Python de código abierto, que interacciona con los servicios de Microsoft Exchange, accediendo al buzón de correo y al contenido de un correo en específico (Exchangelib, 2023).

### **3.4.3. Base de datos**

Para este caso en especial, se puede considerar una variedad de servicios en la nube para guardar la información. Entre esas AWS y Firebase, aunque para este proyecto se tomará más en cuenta la última dada la documentación disponible para Python, especialmente de Firestore, además del manejo de la información en formato de diccionario.

Firestore, de Firebase, se trata de un servicio de base de datos no SQL, que sirve de apoyo para el desarrollo de aplicaciones, brindando almacenamiento y sincronización de datos entre el cliente y una aplicación en tiempo real (Comience Con Cloud Firestore | Firebase, n.d.). El funcionamiento de este servicio es con colecciones y documentos, de modo que la información se organiza dentro de los documentos y estos dentro de las colecciones.

Dado que existe una implementación para Python, que sería *firebase-admin*, se puede utilizar esta para acceder a las colecciones de los datos extraídos, las credenciales de usuario y los parámetros de extracción. Básicamente, accediendo a colecciones, y de estas

a documentos, se puede recuperar y almacenar información. Esta implementación puede ser factible gracias a los métodos *set* y *get*, del componente de Firestore dentro de la librería de *firebase-admin* a aplicar en este proyecto.

#### **3.4.4. Aplicación web**

Streamlit es un framework de código abierto, aplicado para Python, el cual sirve principalmente para la visualización de datos y análisis de estos. Puesto que uno de los puntos clave es la creación de un espacio para visualizar datos, Streamlit puede ser el componente indicado.

Streamlit como tal se consideró puesto que ya tiene métodos y elementos por defecto que hacen más fácil el desarrollo de una aplicación web, además de que existe una librería adicional de la de *streamlit* para poder presentar una página de un documento y seleccionar áreas de interés, que se llama *streamlit-drawable-canvas*. La intención de usar esta extensión es para poner visualizar cada página de un ejemplar PDF y dentro del panel en el que se muestra, por medio de figuras rectangulares en dicho panel o pizarra de dibujo.

## **4. DISEÑO DEL SISTEMA**

### **4.1. Casos de uso.**

Para este sistema, se involucran tres actores principales: el usuario, la base de datos y el sistema de correo electrónico (email). El usuario se encuentra involucrado en todos los casos de uso que sean de la interfaz gráfica de usuario, que es la página o aplicación web, mientras que el sistema de correo electrónico se vincula directamente con el flujo automatizado y el caso de extracción y almacenamiento de la información. Finalmente, la

base de datos se divide en dos componentes principales: base de datos de cuentas de usuario y base de datos de datos extraídos.

De esta forma, para las cuentas se involucra tanto en el sistema de logeo como en el chequeo constante de correos, brindando datos sobre el usuario registrado y los parámetros de extracción que este había definido (esto incluye las áreas y paginas donde se quiere que se haga extracción en un documento que llegue por correo y el asunto para identificar al correo de interés). Mientras tanto, la sección de datos extraídos de la base de datos implica operaciones de almacenamiento de la información extraída por correo y la recuperación de datos de esta.

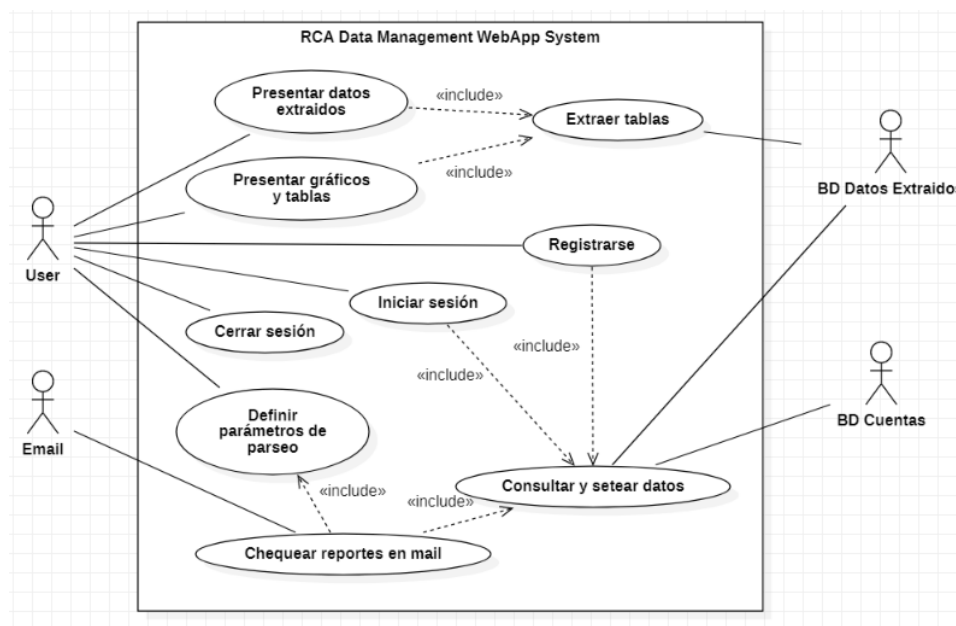


Figura 2 Diagrama de casos de uso

## 4.2. Diagramas robustos.

Por medio del diagrama robusto, se han definido un grupo de casos de uso a considerar para este proyecto, los cuales ser consideraron para el producto final. Esto ha implicado que se tomara en cuenta un grupo de detalles para el funcionamiento de cada componente del sistema, como sería la interacción del usuario con la página web y compilar

en un mismo caso de uso la extracción e ingreso y modificación de información en la base de datos, siendo este caso de uso aplicado a la base de datos como un actor único, aunque trabajando con sus subpartes dadas las situaciones correspondientes (BD de cuentas para casos de logeo y chequeo y extracción de datos del correo electrónico, y BD de datos extraídos para presentar gráficos y tablas).

#### 4.2.1. Registrarse

El usuario ingresa a la pantalla de registrarse e ingresa sus datos, esto del sistema pasa a consultarse en la base de datos, en la sección de cuentas. En este punto, se busca saber si el contenido de registro este duplicado, es decir, si ya existe al menos un usuario que tenga el nombre de usuario. En caso de haber duplicados, se retorna un mensaje solicitando que ingrese otros datos porque los que ingreso ya están registrados. Al momento de ingresar credenciales, no haber duplicados y guardarse la cuenta de usuario se le pasa a la pantalla de inicio de sesión, donde el usuario debe ingresar sus credenciales para poder ingresar al sistema.

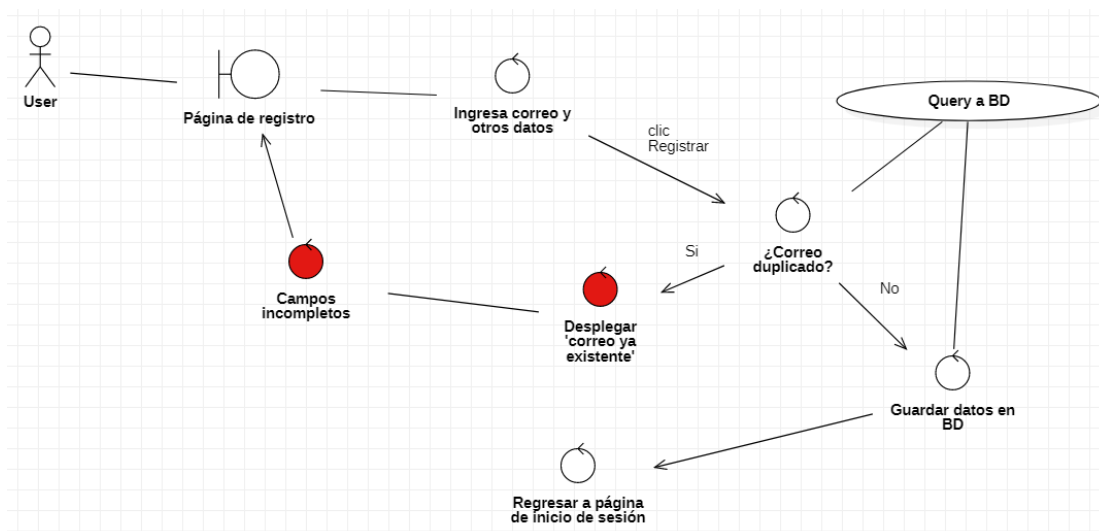


Figura 3 Diagrama de registro

#### 4.2.2. Iniciar sesión



Ingresa el usuario a la pantalla de inicio de sesión, en la cual ingresa sus credenciales. Cuando el usuario ingresa sus credenciales, que sería su correo electrónico y contraseña, se pasa estos datos a verificar en la base de datos, en la sección de cuentas, para ver si las credenciales son correctas. En el caso de credenciales correctas, se pasa a la pantalla principal del sistema, dando la bienvenida al usuario.

En cuanto a situaciones alternativas, si un usuario no existe o las credenciales son incorrectos, se manda un mensaje de vuelta indicando que el nombre o contraseña no son correctos.

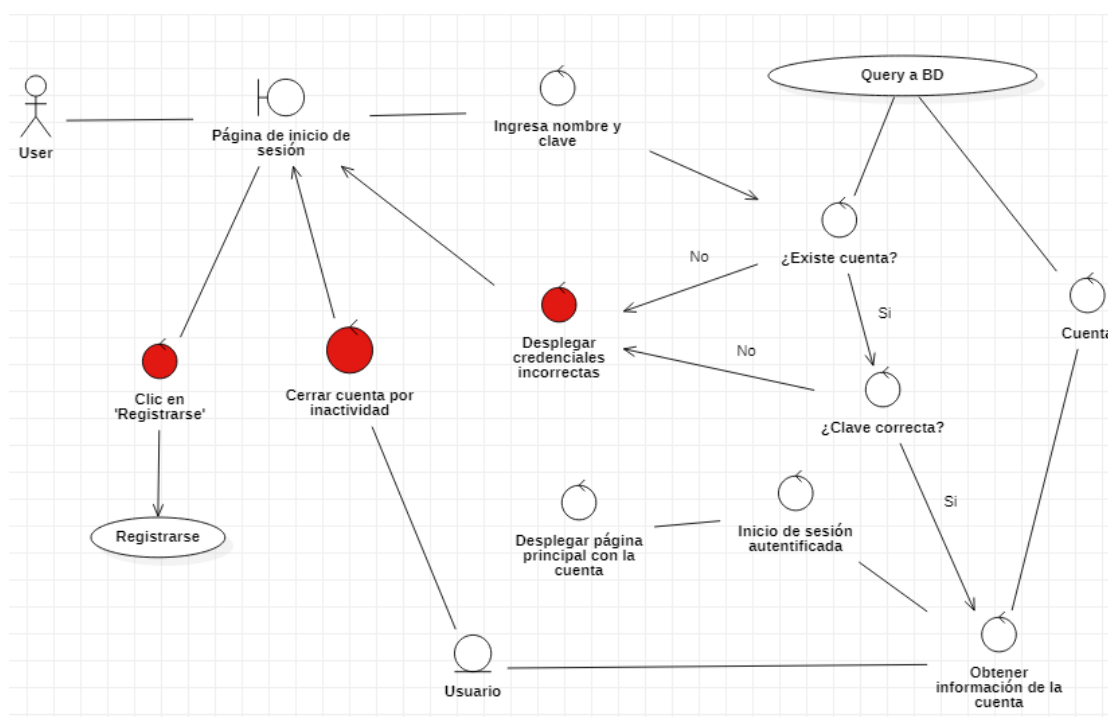
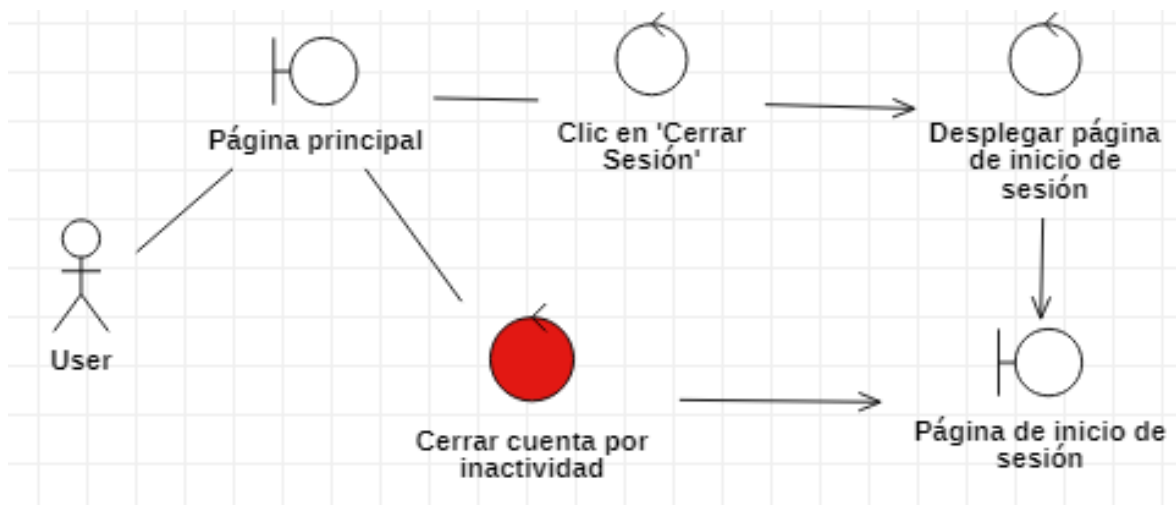


Figura 4 Diagrama de inicio de sesión

#### 4.2.3. Cerrar sesión

El usuario se encuentra en la pantalla principal, da clic en el botón de cerrar sesión y se dirige a la pantalla de iniciar sesión nuevamente.



*Figura 5 Diagrama de cierre de sesión*

#### **4.2.4. Definición de parámetros de extracción de datos**

Esta es la sección, por cantidad de implementaciones, más extensa del sistema. Aquí el usuario clikea en la sección de Parsing (que es la sección para definir los parámetros de extracción de datos, de un documento PDF, por parseo).

Al momento de tener la credencial de acceso al correo registrada en la base de datos, se accede a la sección de selección de tipo de reporte, en esta sección el usuario selecciona uno de los reportes disponibles a escanear, que serían entre Match (Matching history), DIFA, Tear Down y Pull. Posteriormente, define un asunto para registrarlo en la base de datos, lo cual será útil para que cuando se ejecuta el flujo automatizado se verifique en el correo del usuario si ha llegado un correo con el asunto mencionado. Luego se pasa a la sección de extracción de campos de texto, aquí se inserta un documento adjunto, y se despliega un canvas, donde el usuario selecciona las áreas de los campos de texto que se quieran extraer.

Cuando ya se tengan los parámetros de extracción de texto que se definieron con la selección de áreas en el documento, se pasan a otra sección, que es para los parámetros de extracción de tablas. Dicha sección funciona similar a la anterior, solo que aquí la selección de áreas implicará tratar de extraer de ahí una tabla de datos. Cuando el usuario haya pasado por todas las páginas y haya clicado en guardar, se almacenan dichos parámetros y se retorna a la página de selección de tipo de reporte nuevamente.

Entre los casos alternos más importantes que se involucran en este caso de uso, se implica que cuando el usuario cliquee el botón de cancelar, que está disponible cada fase de la definición de parámetros de extracción, se regresa a la sección de selección de tipo de reporte y se cancela todo el proceso de establecimiento de parámetros de extracción de datos. Por otro lado, se considera que en caso de que el usuario no haya ingresado la credencial de acceso al correo, se pondrá una pantalla en la que se solicita el ingreso de esta credencial (esto es necesario ya que el flujo automatizado usa *imaplib* y se requiere de la credencial de correo para acceder). Adicionalmente, en el caso de que no ingrese ningún asunto, se tomara uno por defecto (Report\_sample). Y finalmente, si el usuario no ha seleccionado al menos un área para extracción cuando pase por todas las páginas saldrá un mensaje diciendo que no se guardó nada en la base de datos.

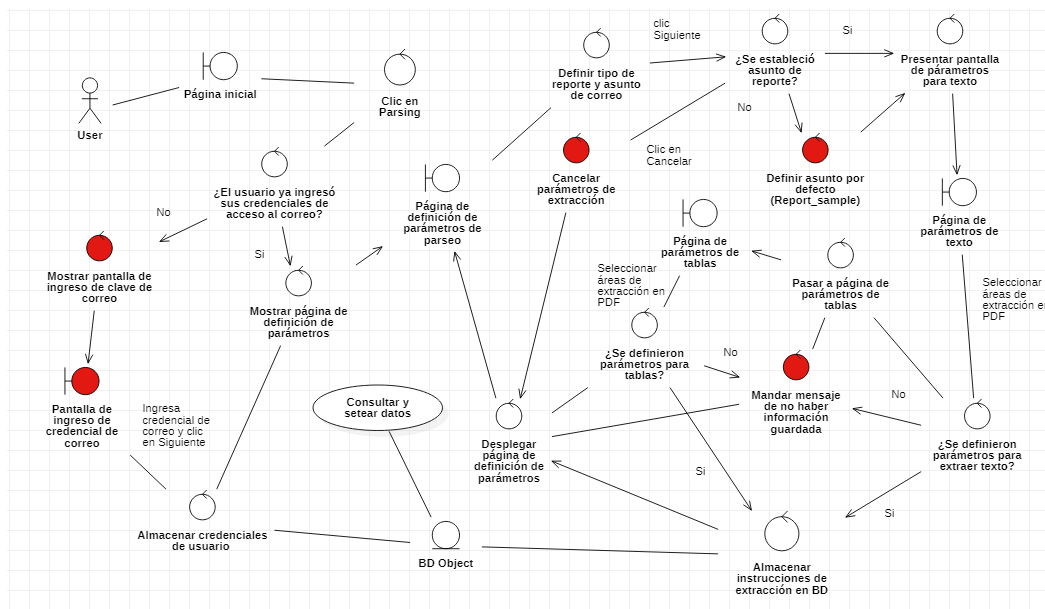


Figura 6 Diagrama de definición de parámetros de extracción por parseo

#### 4.2.5. Consulta a BD

Este caso de uso involucra tanto al caso de uso de consultar e inicializar datos, de modo que se accede a la base de datos para consultar en un directorio específico.

En este punto, se puede actualizar información en dicho directorio, como sería las credenciales de usuario. De igual forma, se puede eliminar un documento o colección, además de insertar datos en un documento o colección.

Como casos alternos, se considera que, en caso de no existir el documento o colección para insertar datos, se genera uno de estos dado el caso, y de ahí se guarda la información. Para el resto de los casos, se lanza un error o indicador (como sería un valor entero), que indique que la información solicitada no pudo ser encontrada. Esto se podría deber a que no exista un documento o colección, o el elemento que se quiere modificar no existe en documento o colección.

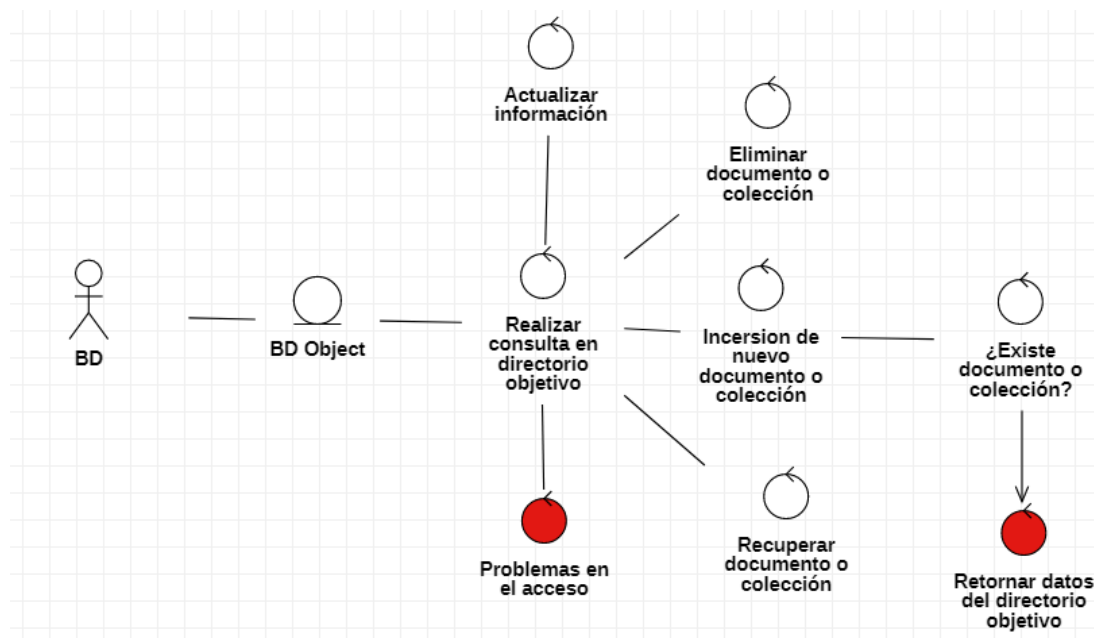


Figura 7 Diagrama de consulta a BD

#### 4.2.6. Presentar datos en tablas

El usuario está en la página principal, selecciona la opción de ver tablas. Al hacer esto, se presenta la página de datos, en esta se selecciona un tipo de reporte. Acto seguido, se procede a consultar en la base de datos los nodos vinculados con dicho reporte, eso se pasa en forma de dataframe y luego se presenta en forma de tabla en la página principal. En caso de que se retorne un resultados vacío o nulo, se avisa y presenta en la página de datos que la tabla está vacía.

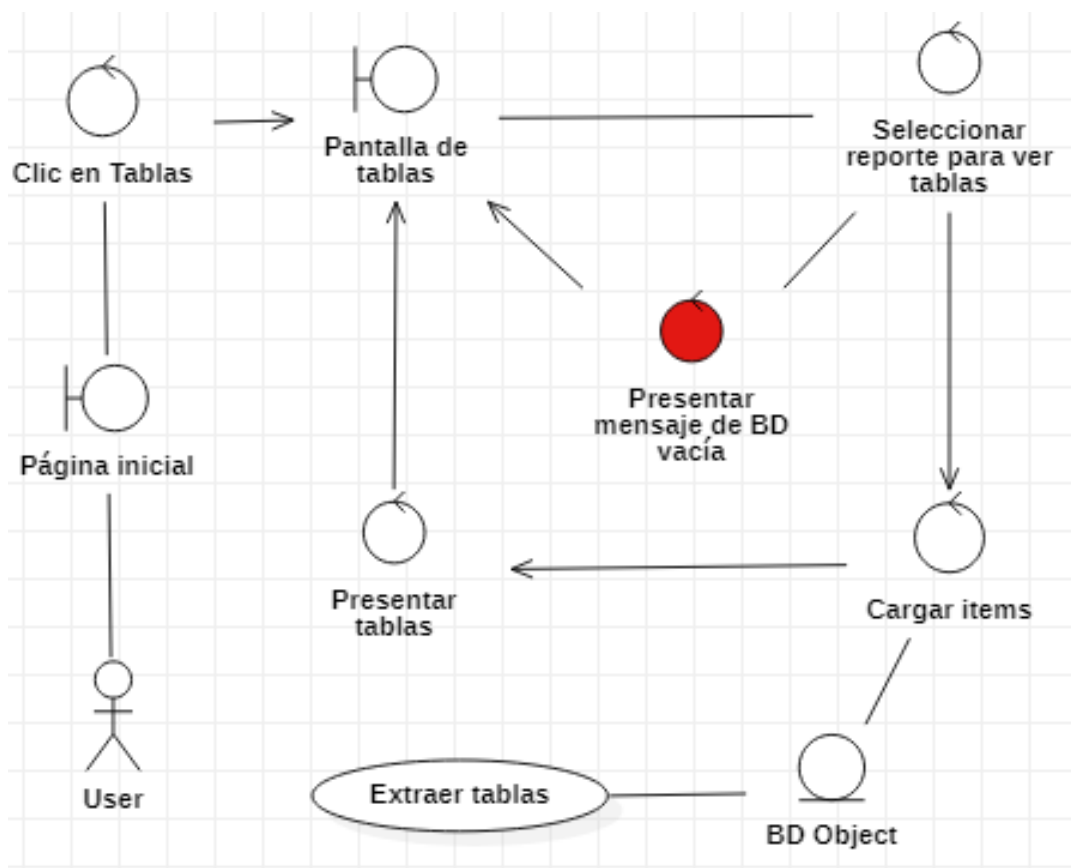


Figura 8 Diagrama de presentar datos en tablas

#### 4.2.7. Generar gráficos en visualizaciones

El usuario pasa de la página principal a la página de visualizaciones, en la cual se presentan gráficos predeterminados, los cuales son gráficos lineales de eficiencia de motor y bomba, de reportes de Match, y un gráfico lineal de los valores de producción de reportes de DIFA. En este caso, si el usuario desea, puede filtrar la información de los gráficos y se generan nuevamente. En caso de no haber información con los filtros se presenta el mensaje de que no hay datos.

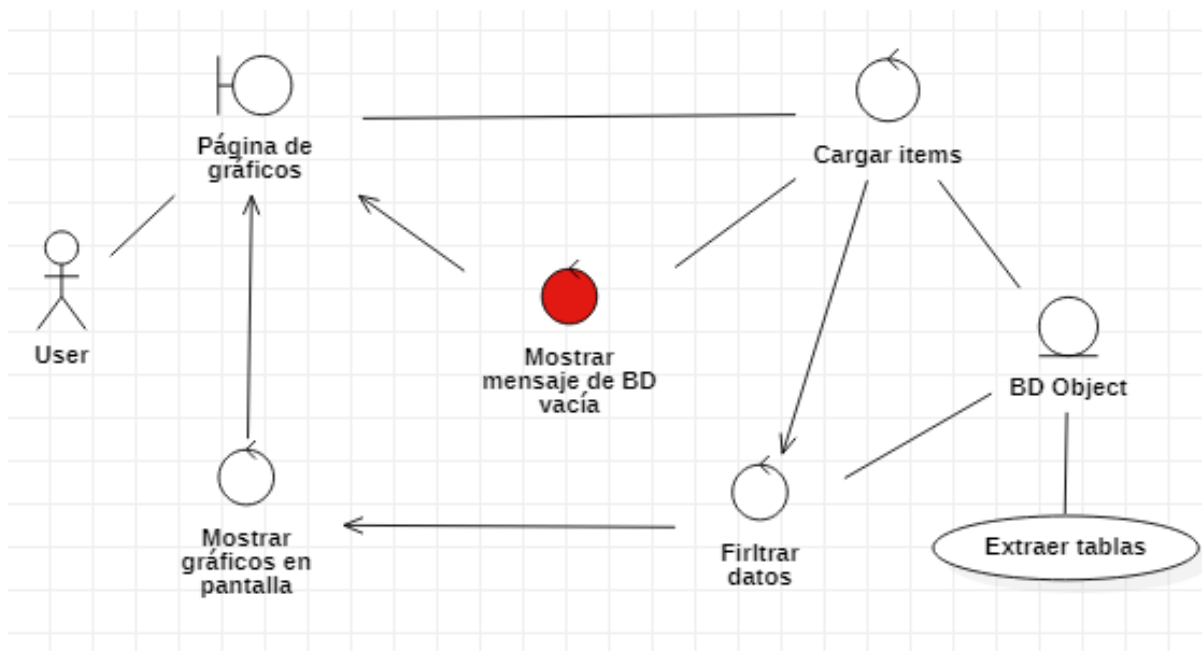


Figura 8 Diagrama de generar gráficos en visualizaciones

#### 4.2.8. Extraer tablas

A pesar de que el caso de uso de consulta a base de datos puede usarse para cualquier sección de la misma, sucede que el caso de uso especialmente para obtener los datos extraídos puede ser un poco diferente.

Se manda una dirección o un conjunto de valores, que conforman la ruta de una colección en la cual se encuentran documentos con la información requerida. Al obtener los nombres de las colecciones, en caso de haber más de una, se procede a ver sus documentos internos.

Cuando ya hay acceso a la ruta de cada documento, se procede a ver y tomar la información sobre los datos extraídos hasta la fecha. Esta información luego es retornada para que sea vista o procesada, dependiendo el caso.

En el caso de excepciones, se considera principalmente que, de no haber una colección o conjunto de colecciones en una ruta determinada, se procede a enviar un mensaje de error.

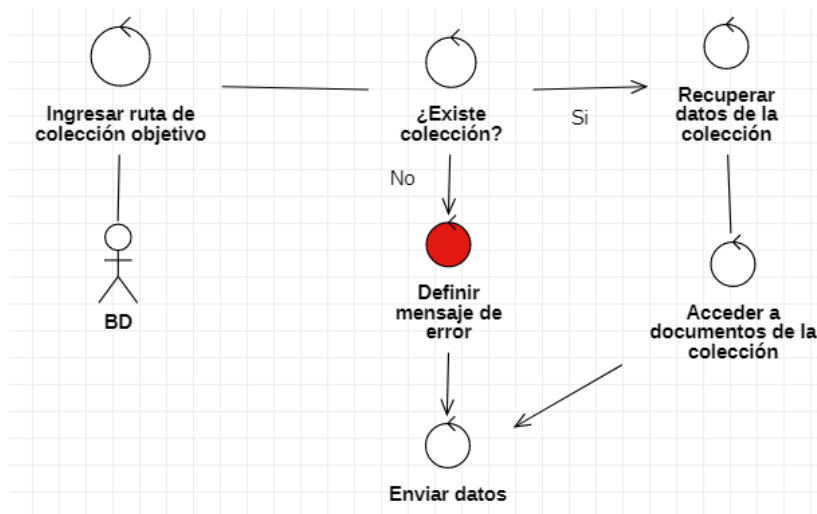


Figura 9 Diagrama de extraer tablas

#### 4.2.9. Proceso automatizado de extracción de datos

La idea de esta implementación radica en que se evidencie que exista dos componentes clave: las credenciales de acceso al correo electrónico del usuario registrado y al menos un asunto. Para ello, se hace un monitoreo constante del sistema de correo electrónico y la base de datos, específicamente sobre la sección de cuentas de esta última.

En el instante en el que se accede por primera vez a la base de datos, se chequea cada usuario y credenciales, de modo que luego se filtra para tener solo a los usuarios que tengan las credenciales de acceso y algún asunto para chequear. Luego de ello, se inicia el chequeo de mail para cada usuario de forma constante (24/7), y esperando a que aparezca un mail con el asunto de interés.

Cuando un mail llega, el sistema verifica si concuerda con el asunto de algunos de los reportes y parámetros de extracción registrados. Se verifica también si tiene adjuntos.



En caso de tener adjuntos, se pasa el documento al sistema y se extraen los datos, dichos datos se insertan en la base de datos, específicamente en la sección de datos extraídos. Posteriormente, tras terminar todo el proceso, se envía un mensaje por consola indicando que el proceso de extracción sobre un documento con asunto dado fue exitoso, indicando que la información extraída fue almacenada.

En el caso de que no se tengan adjuntos, como paso alternativo se ignora el correo, esto mismo aplicaría para correos que no contienen en un asunto de interés.

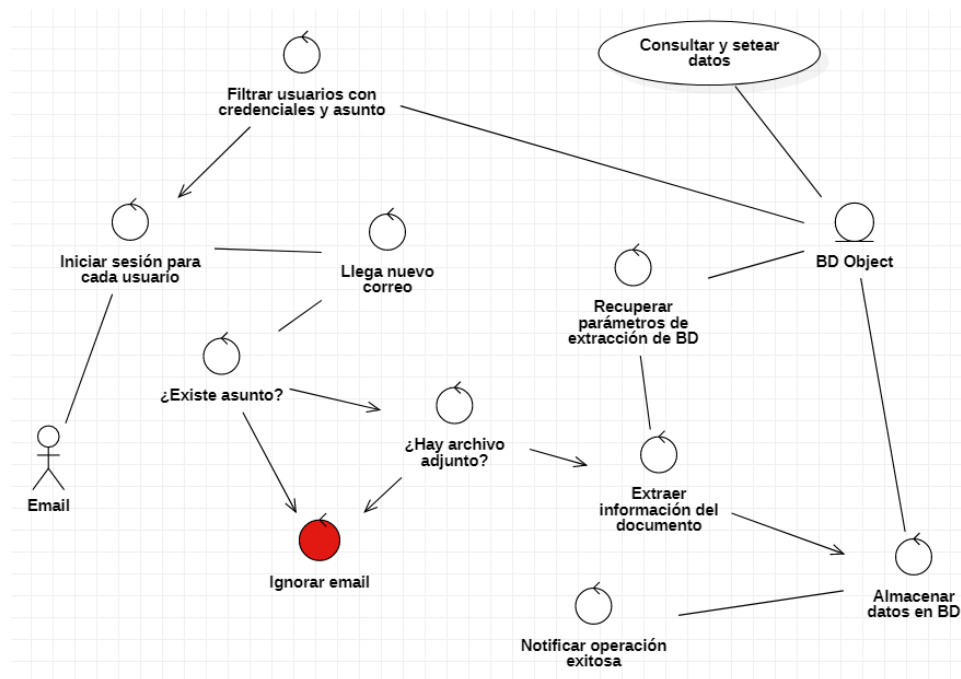


Figura 9 Diagrama de proceso automatizado de extracción de datos

### 4.3. Diagramas de actividades.

Aplicando los casos de uso se pueden definir diferentes actividades, muchas de ellas en conjunto (dados los diferentes casos de uso). En esta situación, las operaciones de inicio y cierre de sesión se consideran una sola actividad, dado que el ingreso de credenciales se aplica en una misma sección de la interfaz gráfica de la aplicación web.

#### 4.3.1. Registro

El usuario pasa a la página de registro, en la cual ingresa sus credenciales, principalmente el correo, una contraseña de acceso a la aplicación web, y un nombre de usuario. Las credenciales son enviadas a la base de datos, con el fin de verificar primero si ya existe un usuario con el correo ingresado, posteriormente, accede a la colección de cuentas de usuario. Ahí se ingresan las credenciales, y se establece el acceso al correo del usuario como denegado (False), al no haberse ingresado aún la credencial de correo (contraseña de correo electrónico). Posteriormente, se notifica al usuario que se ha registrado con éxito, y se pasa a la página de inicio de sesión.

En caso de que haya algún error con el ingreso de las credenciales, sea por duplicado o correo no permitido (ya que se busca solo trabajar con Outlook y Gmail), se notifica al usuario en la misma página.

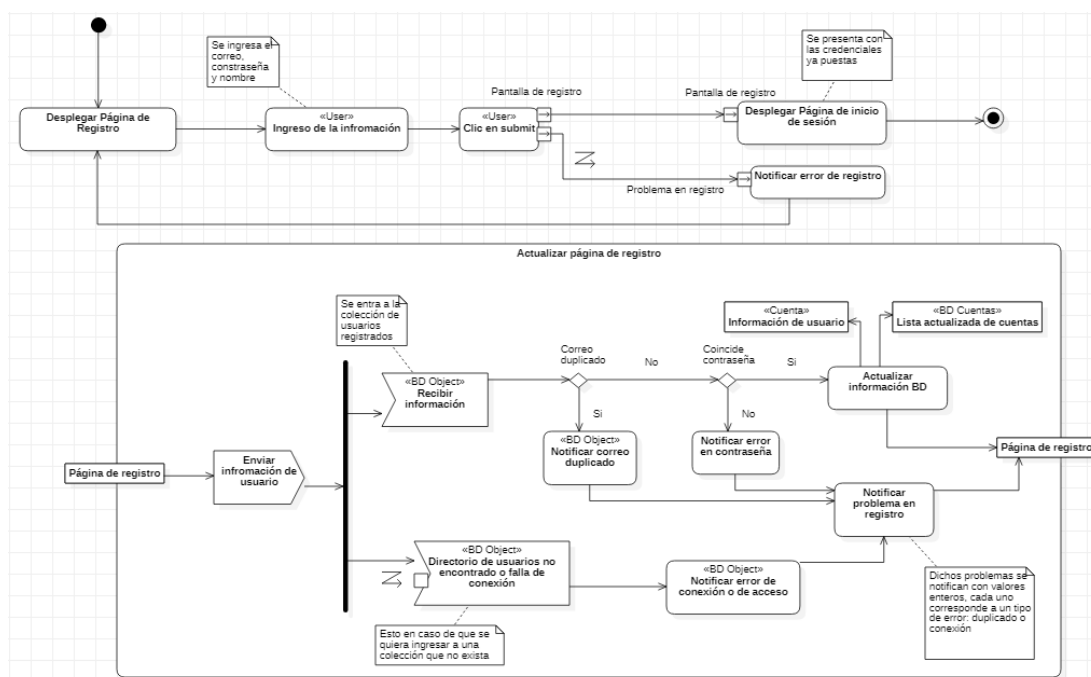


Figura 10 Diagrama de actividad de registro

#### 4.3.2. Logeo

El inicio de sesión no varía mucho en este diagrama, puesto que se mantiene el funcionamiento tal cual se definió para el caso de uso. Aquí las credenciales se ingresan, para luego que el sistema entre a la base de datos y verifique si existe algún usuario con dichas credenciales. Al haber las credenciales, se ingresa al sistema con un mensaje de bienvenida. En caso de que se vaya a cerrar sesión, se espera 20 minutos o que el dispositivo este sin actividad después de un buen tiempo, además del caso en que el usuario cierre sesión.

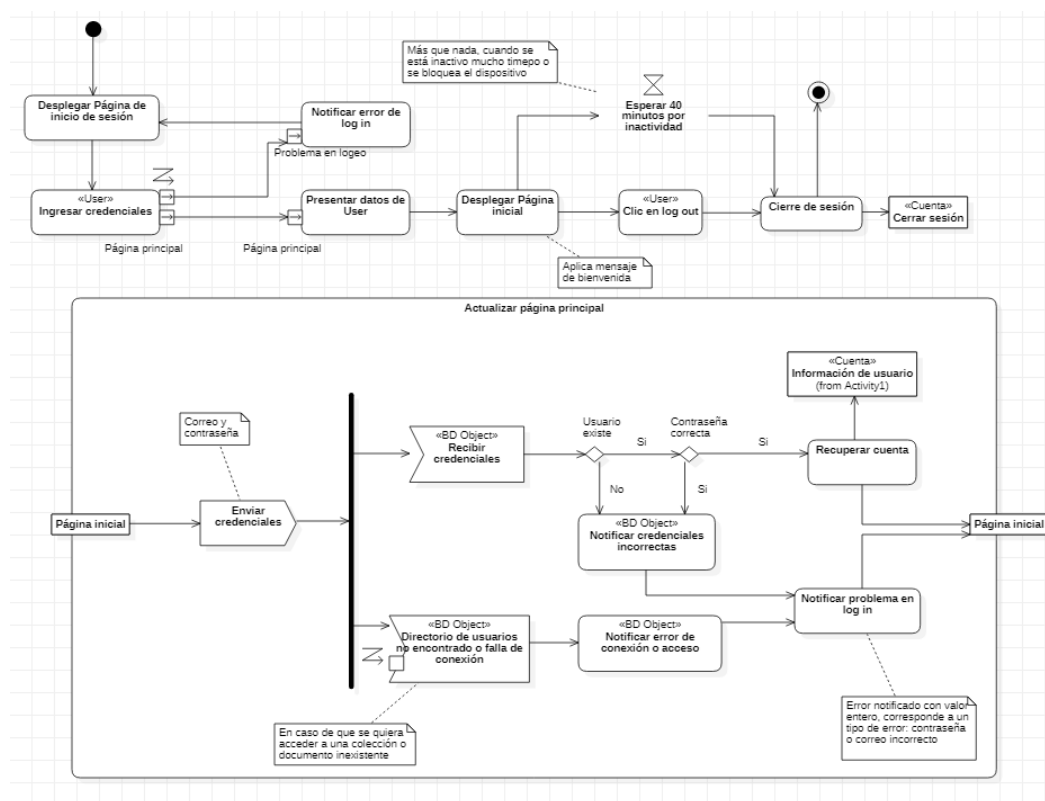


Figura 11 Diagrama de actividad de logeo

#### 4.3.3. Definición de parámetros de extracción

Para esta sección simplemente se considera si el usuario ya ingresó su credencial de acceso al correo, en caso de que no se presenta una pantalla para ello y se accede luego a almacenar ese valor, y el acceso al correo se establece como permitido (True). Luego se

pasa a una sección donde el usuario escoge entre los reportes de fallas disponibles: Matching history, DIFA, Tear Down y Pull. Al escoger uno de estos, se presenta un espacio a continuación donde ingresa el asunto del correo que le llegaría con el reporte que desea la extracción.

En la siguiente sección se sube un archivo PDF, que sería un ejemplar del reporte. Sobre este, en un panel de canvas, selecciona las áreas donde está el texto que desea extraer. De modo que pasa por cada página, sin posibilidad de retorno a la página anterior, y al momento de que llegue a la última página, se presenta un botón para guardar los parámetros de extracción. Esto aplica tanto para campos de texto como tablas, y luego da un mensaje de que la información fue guardada con éxito.

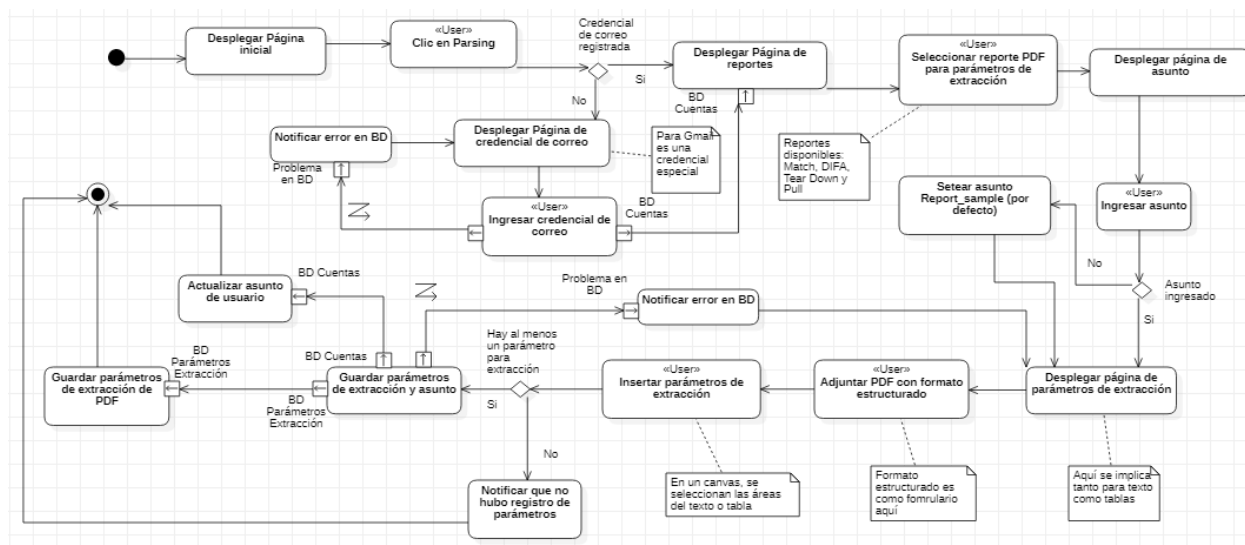


Figura 12 Diagrama de actividad de definición de parámetros de extracción

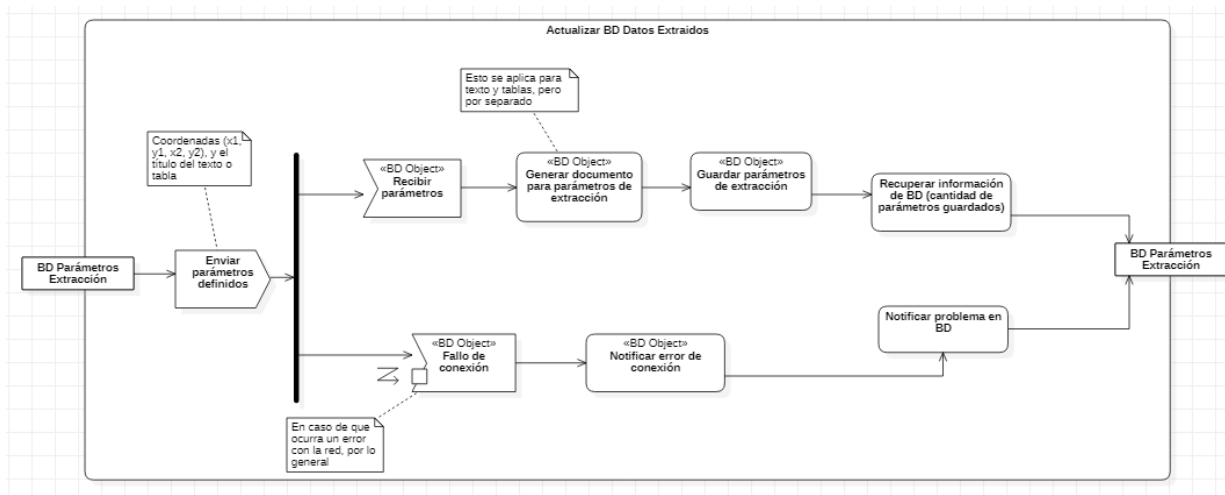


Figura 13 Diagrama de actividad de actualización de BD Datos Extraídos

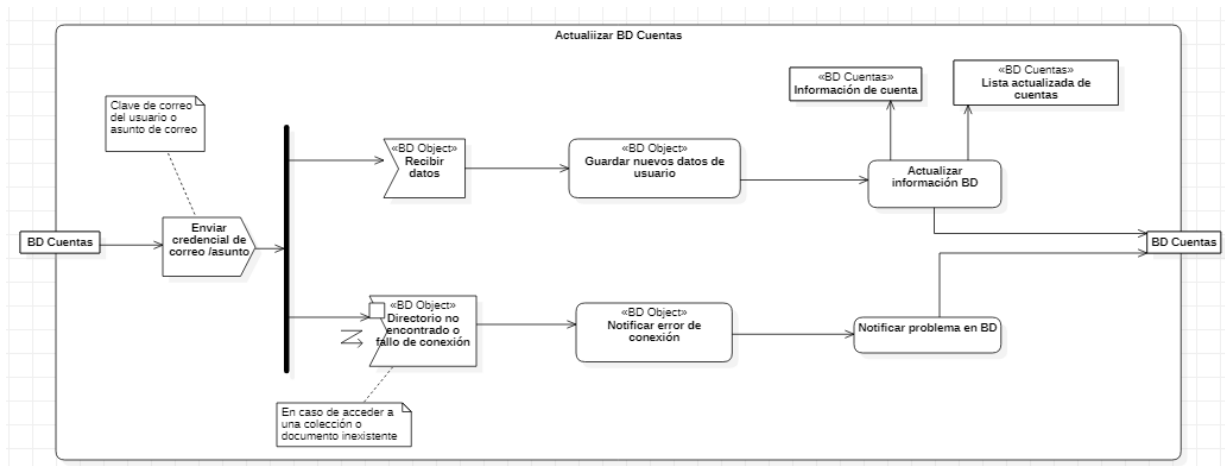


Figura 14 Diagrama de actividad de actualización de BD Cuentas

#### 4.3.4. Presentar tablas

Esta sección implica principalmente que se acceda a la sección Datasets, en la cual se extrae primero la información extraída del correo hasta la fecha. En ese instante, se divide cada grupo de datos en reportes, y así se definen los tabs en la página, y cada uno con su conjunto de tablas.

La información que se extrae se lo hace como diccionarios, o en formato JSON. Luego, se pasan a la página de Datasets y se presentan como Dataframes.

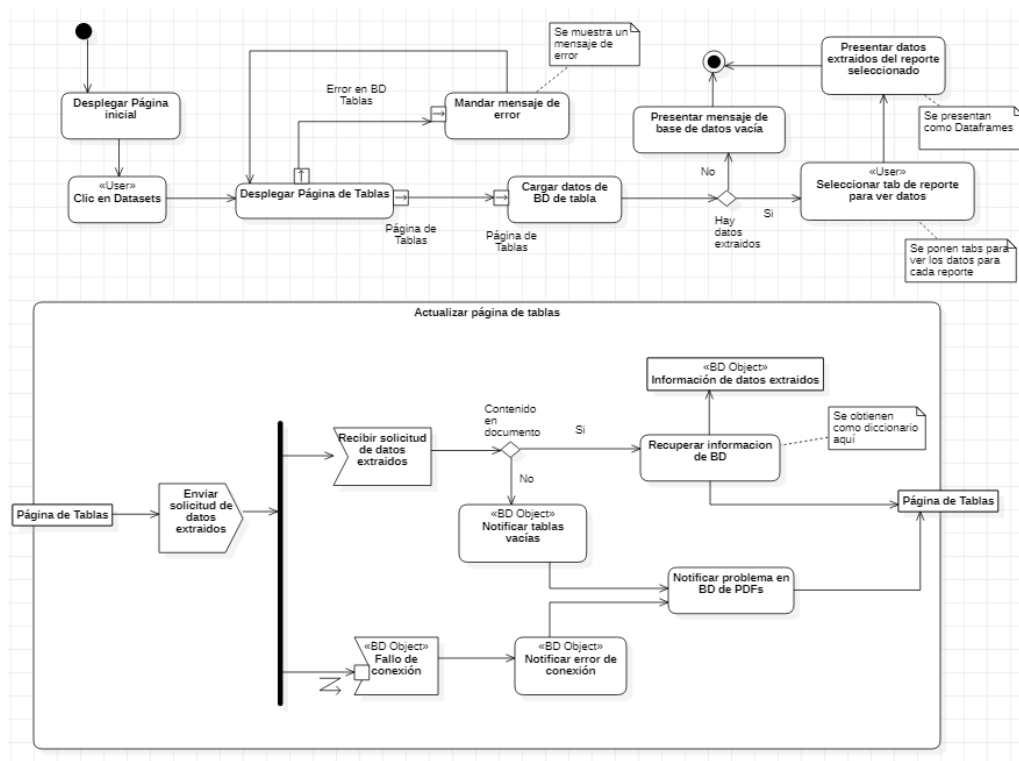


Figura 15 Diagrama de actividad de presentar tablas

#### 4.3.5. Presentar gráficos

Para esta sección, se extraen los datos extraídos de la base de datos. Luego, se filtran los datos con el fin de tener aquellos que sean necesarios para poder hacer los gráficos (en este caso de líneas para producción de DIFA y eficiencia de motor y bomba de Matching history).

En este proceso de filtrado también se toma una lista compilatoria de todos los pozos de los cuales haya datos extraídos. Estos se ponen en un selectbox, en el cual al escoger una de las opciones se actualiza la página de Dashboards (donde están los gráficos y visualizaciones), filtrando la información de las tablas, para solo tener aquellos registros del pozo seleccionado.

Finalmente, se presentan tablas de Pull, DIFA, compilaciones de Tear Down y gráficos de líneas para Matching history. En cuanto a los gráficos, se puede filtrar la información y que se grafique otro tipo de variable, dependiendo la información disponible.

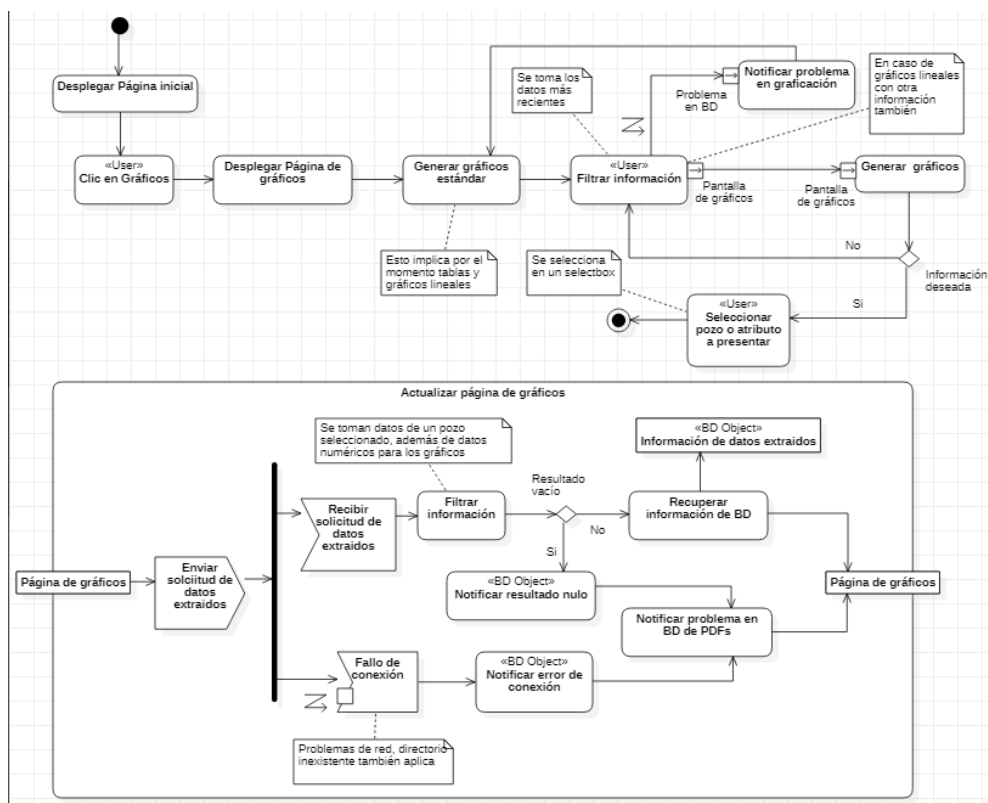


Figura 16 Diagrama de actividad de gráficos

#### 4.3.6. Proceso automatizado de extracción de datos

En este caso se toma en cuenta principalmente el funcionamiento del sistema cuando llega un correo, se considera que ya se tiene las credencial y asunto de un usuario registrado, además de ya tener accionado algún hilo que ejecute este proceso en paralelo.

Se comienza chequeando si ha llegado un correo, en la bandeja de entrada, como también correos no leídos. En caso de que el asunto de un correo concuerde con la lista de asuntos o palabras clave de asunto que el usuario ha definido en la parte de parámetros de extracción, se procede a revisarlo.

En la revisión se ve que tenga un archivo adjunto, en PDF, considerando que este tipo de archivos es muy común en las empresas para reportes. Al suceder ello, se solicitan los parámetros de extracción y se procede a extraer la información del documento.

Cuando ya se tenga la información extraída, se almacena está agrupando por el nombre de la tabla, concatenando sobre la información ya existente. Y dando un mensaje de que la información fue guardada exitosamente.

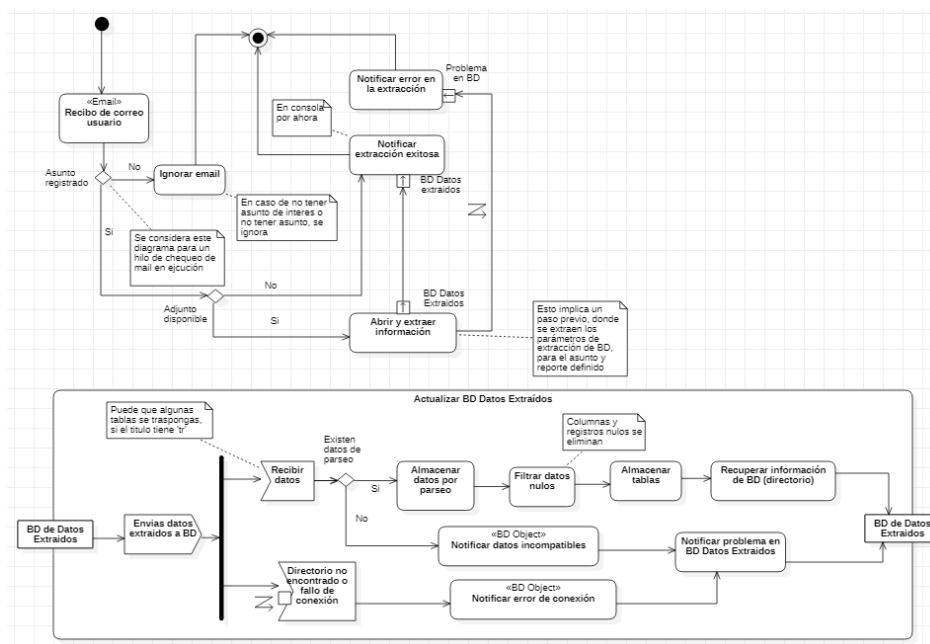


Figura 17 Diagrama de actividad de flujo automatizado para extracción

#### 4.4. Diagrama de clases.

Por medio de lo generado, se tiene entonces el siguiente diagrama de clase, el cual implementa páginas y objetos de back-end.

Dentro del diagrama se declaran las clases más importantes, que serán de ayuda para la extracción de datos, el almacenamiento de la información, chequeo del correo electrónico, la presentación del canvas para definir parámetros de extracción y el filtrado de datos extraídos, obtenidos de la base de datos.



Dentro del diagrama, se puede ver un grupo de clases con el estereotipo *page*, esto indica que son scripts con funciones para presentar la información en cada página. Al usar Streamlit, se puede solamente con la escritura de scripts, sin necesidad de definir clases, establecer el funcionamiento de cada página. Por otro lado, una clase contiene el estereotipo de script, eso indica que se corre como un script normal en Python, y este se aplica para el flujo automatizado, que muestra el estado del chequeo de correos en consola.

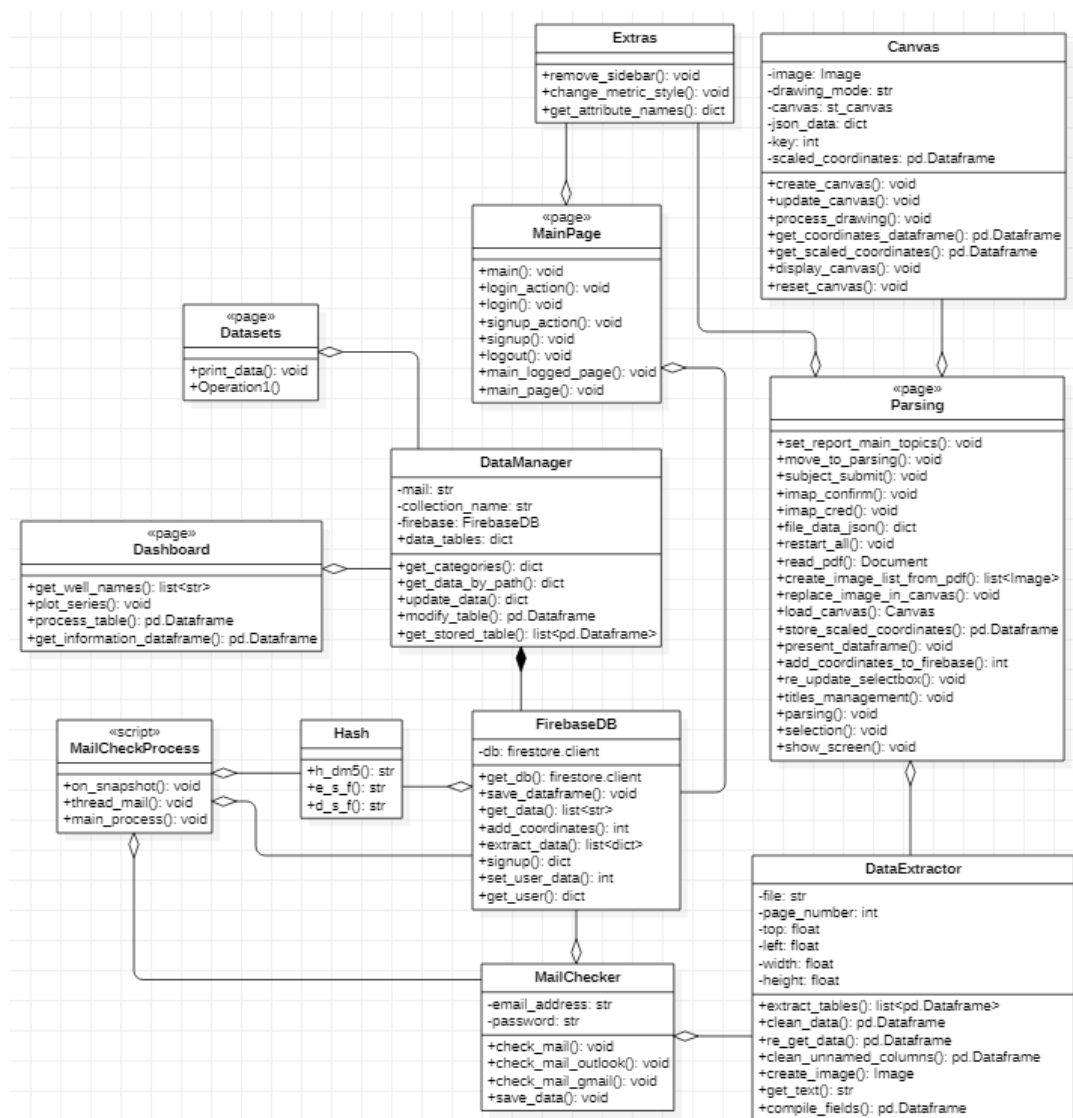


Figura 18 Diagrama de clases

## 5. DESARROLLO DE PROTOTIPO

### 5.1. Uso de parseo en el proceso de extracción de datos

#### 5.1.1. Extracción de tablas y texto

La extracción de tablas es importante en el conjunto de procesos que tiene este proyecto, puesto que se requiere implementar un proceso de extracción de datos por etapas, y en ocasiones de una forma más organizada y orientada a objetos, con el fin de poder operar sobre un grupo específico de parámetros para cada tabla.

Es por ello que se definió una clase `DataExtractor`, la cual recibe las métricas para extraer datos en un documento. Dichas métricas serían las coordenadas del área en la que se encuentra una tabla, la página en la que se ubica la tabla, la ruta del archivo para la extracción y un título de tabla, que se le adjudica para almacenarla con un identificador en la base de datos. De cierto modo, escribir una clase para la extracción de datos fue una idea clave al inicio, pero siempre ocurría que la información obtenida se componía de columnas nulas o registros vacíos, por lo que era necesario procesar la información luego de extraerse.

Debido a ello, se aplicó *pandas*, con el fin de poder pasar la tabla extraída a un `DataFrame` para su modificación, lo que implica un parseo de la extracción a *pandas DataFrame*. Esto con el fin de que se eliminen registros nulos y columnas sin nombre. De esta manera, se aplicaba primero una función para extraer tablas, con *tabula-py*, luego se convertía el resultado a `DataFrame` de *pandas*, con el cual se iba eliminando columnas sin nombre (con *loc* para columnas que contengan “Unnamed”), además de registros que contenían todos los datos nulos (uso de *dropna*).

Adicionalmente, se consideró escribir para esta clase un proceso alternativo, por el cual se extrae nuevamente una tabla si es que en el primer intento esta salía vacía. Esto se aplicó con el objetivo de tratar de extraer el contenido de todas las formas posibles, siendo en el caso alternativo removiendo la detección de columnas separadas por líneas, sino por espacios, siendo esto la limpieza de datos.

```
x1: 90.51428571428572, x2: 646.3059428571428, y1: 255.816, y2: 14.834, page: 1
[ Serie                Tipo No. Serie ... Longitud Caja Propiedad
0   418                Head PMP      - ...    0.59   - EP PETRO
1   418 Discha rge Presure AMT - ...    0.71   - EP PETRO

[2 rows x 11 columns]]

x1: 90.51428571428572, x2: 646.3059428571428, y1: 292.38912, y2: 19.37, page: 1
[ Unnamed: 0 Serie Tipo ... Longitud Caja Propiedad
0      NaN  418 D460N ... 11.08 ft S-56790 EP PETRO
1      NaN  418 D460N ... 11.08 ft S-04570 EP PETRO
```

*Figura 19 Resultados de extracción, con DataExtractor, sin limpieza de datos*

Tras extraerse la tabla y contenido de esta, se pasaba a una etapa posterior, en la cual se verificaba si era necesario transponerla o no. Este proceso se aplicó con base en el título de la tabla, y dado que en ocasiones las tablas de los documentos contenían valores de columnas variables, siendo esto un problema para el almacenamiento de los datos en la base de datos (véase anexo 19).

<b>Frecuencia</b>	61.67	61.30	61.00	60.83	59.7	58.79	54.45
<b>Caudal</b>	250.86	280.39	294.77	287.07	285.2	299.76	300.94
<b>PIP</b>	283.16	314.36	430.13	260.91	415.02	360.01	378.84
<b>Corriente</b>	30.52	32.06	30.72	32.15	30.98	27.17	30.54
<b>Eficiencia bomba</b>	23.10	11.83	21.08	29.87	22.78	20.30	23.69
<b>Carga Motor</b>	55.43	57.54	56.63	61.46	58.87	63.33	63
<b>Fluido sobre la bomba (ft)</b>	43.76	21.67	78.65	25.36	83.16	60.14	197.16
<b>Comment</b>	-	-	OK	-	-	No recomendado	OK

*Figura 20 Muestra de una tabla de Matching history, con columnas de frecuencias*

Por otro lado, se aplicó *tabula-py* también en la extracción de campos de texto, siendo este un caso exclusivo dentro de los métodos de la clase. Esto se debe a que *tabula-py* se aplica para extraer tablas y no campos de texto, por lo que era necesario que el contenido de la supuesta tabla se pase a una cadena de texto.

En esta clase, dependiendo de que tipo de datos se quieran obtener, pasa o a una función para procesar columnas o una para realizar parseo de la tabla a texto, y luego eliminar expresiones regulares, usando *re*. En el caso de texto, se convierte a cadena de caracteres el contenido de la tabla, y se va eliminando elementos de la estructura de extracción de *tabula-py* (véase anexo 20). Entre dichos elementos, se encontrarían expresiones o caracteres frecuentes, como llaves o corchetes, borrando también valores como “Columns” o “Index”, quedando solamente el texto final extraído.

```
x1: 683.3828571428571, x2: 40.73142857142857,
1
Empty DataFrame
Columns: [08/3/2023]
Index: []
```

*Figura 21 Muestra de la extracción de un campo de texto con tabula, parseado a string*

### **5.1.2. Canvas y streamlit**

Puesto que dentro del sistema integrado se tiene una aplicación web, es aquí donde el usuario define los parámetros para la extracción de datos. Y dado que se aplica *streamlit*, era necesario escribir el código necesario para presentar un espacio donde el usuario pueda seleccionar qué áreas del documento se deben revisar para obtener las tablas y campos de texto.

Pues sucede que existe una extensión de *streamlit*, que ayuda a establecer un panel o pizarra para ilustraciones o creación de figuras geométricas, que es *streamlit-drawable-canvas*. La idea entonces fue que, usando esta extensión se pueda presentar un canvas o espacio de ilustraciones, donde se pasa cada página a imagen y se selecciona un área específica de la página en el panel. Para esto, se usó la herramienta de rectángulo para encerrar un segmento de la página, y así obtener las coordenadas de dicho espacio.

Para esto, se creó una clase Canvas, la cual despliega el canvas en la página de definición de parámetros de extracción. De igual forma, también para desplegar las coordenadas de la figura creada en el canvas, ya escaladas a las dimensiones del documento. Esto último ya que el canvas genera las coordenadas de las figuras rectangulares dentro de las dimensiones de la pizarra, y es por eso que tuvo que transformar los valores para que se obtengan las tablas correctamente.

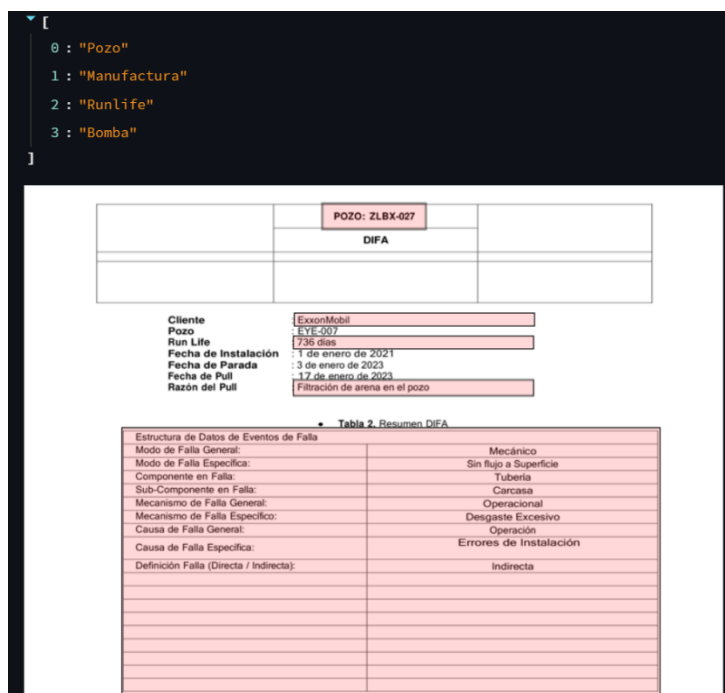


Figura 22 Vista de la interfaz de canvas para seleccionar áreas de extracción

### 5.1.3. Vista previa de la extracción

Cuando un usuario hace una selección, es necesario saber si dicha selección contiene los datos que busca que se extraigan. Dada esa situación, es importante que la persona que defina los parámetros de extracción tenga una idea sobre cómo podrían extraerse los datos.

Es ante ello que, cuando se realiza una selección en el canvas, se presenta una vista previa de la información que se va a extraer. Para esto se agregó un objeto de DataExtractor en el script de la página de parámetros de extracción. Cuando se hace una selección, se actualiza una tabla de coordenadas, ya mencionada anteriormente en la transformación y escalamiento de los valores de coordenadas.

Cuando se transforman las coordenadas, esas se usan para iniciar un objeto de DataExtractor, y luego se llama a una función que extrae la tabla del área, le quita los valores nulos y el resultado se presenta en pantalla (véase anexo 23).

Scaled Page Coordinates								
	Left	Top	Width	Height	scaleX	scaleY	Final height	Final width
0	90.5143	255.8160	652.8343	13.4640	0.9900	1.3700	14.8340	646.3059
0	90.5143	292.3891	652.8343	18.3600	0.9900	1.0100	19.3700	646.3059

Extracted Page Tables							
Table 1							
	Serie	Tipo	No. Serie	No. Parte	Rosca	Metalurgia	Condición
0	418	Head PMP	-	1,298,461	3 1/2" EUA	RLOS	Ex ternamente limpio.
1	418	Discharge Pressure AMT	-	187,144,521	FLANGA	RLOS	Ex ternamente limpio.

Table 2								
	Serie	Tipo	# Etapas	No. Serie	No. Parte	Tipo.1	Rotacion	Condición
0	418	D460N	121	2FN5K03692	101,318,389	66CRCT-AFL-INC-AA-ZZ-RIOA	Libre	Externa
1	418	D460N	121	2FN5K03693	101,318,389	66CRCT-AFL-INC-AA-ZZ-RIOA	Libre	Externa

*Figura 23 Vista previa de la extracción de tablas con canvas*

Por otro lado, cuando se quiere hacer la extracción de texto, se llama a una función diferente, la cual extrae una supuesta tabla aplicando *tabula-py*, para luego hacer parseo y eliminar expresiones regulares (véase anexo 24).

Scaled Page Coordinates									
	Left	Top	Width	Height	scaleX	scaleY	Final height	Final width	Title
0	99.5657	98.7943	26.0229	6.9943	1	1	7.9943	26.0229	Pozo
0	102.9600	104.9143	18.1029	4.3714	1	1	5.3714	18.1029	Campo

Extracted Page Text	
<b>Text Pozo</b>	
ARCA-035	
<b>Text Campo</b>	
ARCA	
Current Extracted Page Fields	
	value
Pozo	ARCA-035
Campo	ARCA

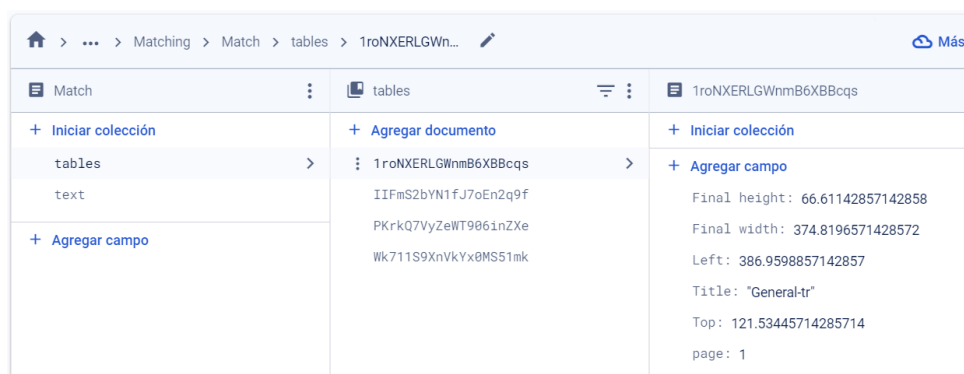
*Figura 24 Vista previa de la extracción de texto*

## 5.2. Implementación de base de datos en el proceso de extracción y almacenamiento.

### 5.2.1. Almacenamiento en Firebase

Cuando se opera con reportes de fallas, se requiere que se almacenen los parámetros para extraer la información de los documentos que lleguen al correo. Además de eso, se requiere un espacio para ir almacenando los datos extraídos, los cuales luego serán de utilidad para ser presentados ante el usuario cuando este los solicite. Esto además es importante ya que se requiere de generar gráficos y tablas para presentar la información relevante en cuanto a fallas de cada pozo que se tenga datos, y a mayor información será mejor el análisis.

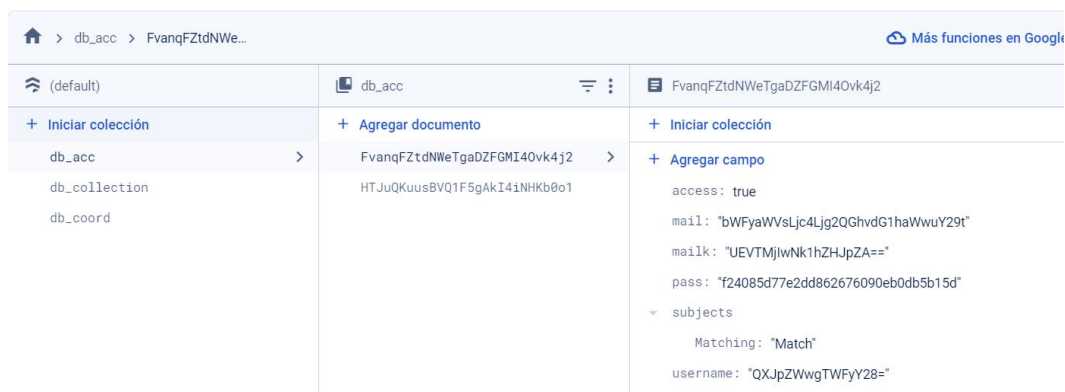
Como se requiere de una base de datos, donde se almacenen la información de parámetros de extracción o credenciales de usuario, se usó Firestore de Firebase. Al utilizar Firebase, se tuvo que comprender el funcionamiento de una base de datos por colecciones y documentos. La idea de este sistema de almacenamiento es que se tiene las colecciones para almacenar documentos en su interior, y estos en su interior la información que se desea guardar. A pesar de tener esa estructura, era posible que un documento condujera a otras colecciones en su interior, y estas a nuevos documentos. Como la información a almacenar dependía de los atributos comunes que compartía un conjunto de datos con otro, en ocasiones era necesario definir una ruta de la forma colección-documento-colección.



*Figura 25 Ruta y contenido almacenado en Firebase, para parámetros de extracción*

Fue por esto que se tuvo que crear una clase de Firebase, o que se conecta a los servicios en la nube de esta, para poder extraer datos de cada tipo de colección de forma específica. De esta manera, la información debía procesarse de maneras diferentes dada la extensión o profundidad del contenido almacenado. Esto significaba que se debía acceder para una tabla de datos extraídos por dos colecciones y dos documentos, mientras que para los usuarios solo con entrar a la colección de estos ya se podía extraer los datos directamente (véase anexo 26).





*Figura 26 Ruta y contenido almacenado en Firebase, para cuentas de usuario*

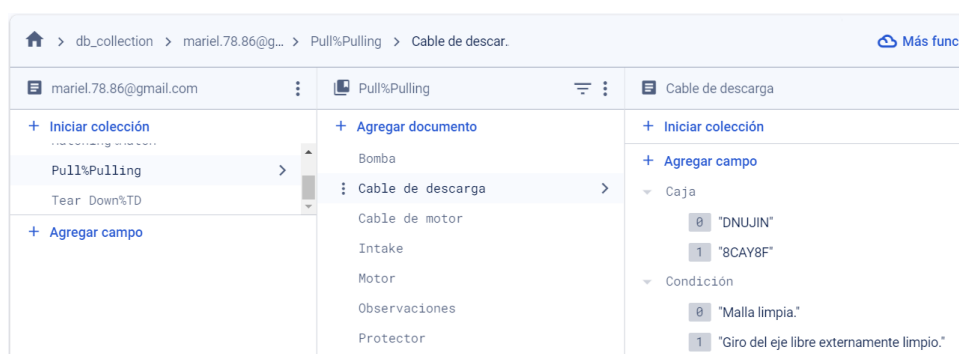
Ante esto, cada colección fue tratada de forma diferente, especialmente por la extensión en cuanto a colecciones y documentos necesarios para guardar la información de forma ordenada. Por ejemplo: para un reporte de Match, se definió la colección de datos extraídos como `db_collection` (enfaticando que se coleccionan o recopilan datos de diferentes reportes ahí), luego se accede a un documento `example@gmail` como dirección de correo del usuario, luego se tendría ahí otra colección llamada `Matching%Match` (considerando que lo primero implica el nombre del reporte y lo segundo el asunto de correo). Dentro de esta colección, se tendría un conjunto de documentos que serían para cada una de las tablas consideradas dentro de ese reporte y que el usuario seleccionó para la extracción, como podría ser una tabla de información general y una de la eficiencia de la bomba. De este modo, se tienen los datos almacenados de forma ordenada y mejor agrupada.

Por otra parte, la información del usuario se modificaba con funciones exclusivas, puesto que se decidió encriptar la misma, con el fin de brindar cierto grado de privacidad de las credenciales. Siendo para este caso más la modificación de información en cuanto a las credenciales de acceso y la adición hecha para asuntos de correo solamente.

### **5.2.2. Extracción de datos de Firebase**

La información almacenada ya era un paso importante, pero la obtención de datos es una parte importante y que se debe analizar dado cada caso. Sucede que, para la base de datos y la extensión de colecciones y documentos, se debía primero verificar los nombres de documentos y luego de colecciones almacenadas. Posterior a ello, se iteraba sobre cada colección y se extraía cada documento con datos extraídos, para así luego poder ser procesados y mostrados en la página de tablas a mostrar al usuario.

La extracción requería acceder a las colecciones internas de la base de datos por medio de una ruta, de modo que se obtengan los nombres de estas, y de estos sus datos, exclusivamente para los parámetros de extracción y los datos extraídos.



*Figura 27 Almacenamiento de datos extraídos de PDF, para reportes de Pull*

### 5.2.3. Verificación de correos

El acceso a correo fue una cuestión importante para el sistema, ya que se requería de las credenciales del usuario para poder entrar a la bandeja de entrada y verificar la existencia de algún correo de interés dado el asunto. Fue pensando en el sistema de correo como se tomó la decisión de automatizar un flujo que vaya por cada una de las cuentas de usuario, verificando si en algún momento han ingresado un asunto para realizar el chequeo y extracción, usando el protocolo IMAP o directamente accediendo al servidor de Microsoft Exchange.

La tarea de acceder al correo fue una de las más complejas de este proyecto, ya que más allá de requerir las credenciales del usuario para entrar a su correo, se debía monitorear y acceder a la base de datos constantemente para ver si había un nuevo usuario con asunto para chequear. Fue gracias a eso que se creó una clase MailChecker, la cual solicita las credenciales de usuario, y se acciona con una lista de asuntos para verificar si hay algún correo de interés, dado su asunto.

Esta clase usa *imaplib* para realizar una conexión con credenciales a un servidor de Gmail, pero también *exchangelib* para hacer lo mismo con los servicios de Outlook y Hotmail. Dentro de esta clase se establecieron las funciones para el caso de *imaplib* con Gmail y *exchangelib* para Hotmail y Outlook, considerados los dos únicos servicios de correo para este proyecto por ser los más usados y con una cantidad de documentación de apoyo.

Adicionalmente, en cada función se verifica y guarda temporalmente el reporte en PDF para ejecutar la extracción de datos con *tabula-py* y *re*. El almacenamiento temporal en el dispositivo local era necesario ya que *tabula-py* solo trabajaba con la ruta de acceso al documento, siempre y cuando esté dicho documento en el dispositivo local. Entonces, para ejecutar la extracción se llama a un método estático de la clase, que solicita los parámetros de extracción, los aplica para extraer los datos, luego parseo y limpieza de valores basura, para posteriormente guardar la información en la base de datos.

```

processing...
Nuevo documento añadido: HTJuQKuusBVQ1F5gAkI4iNHKb0o1
Datos del nuevo documento: {'mail': 'bWFyaWVsLjc4Ljg2Q6dtYWlsLmNvbQ==', 'pass'
Nuevo documento añadido: S7yQq8w6FRNs8CFZruTQhzwvtIo2
Datos del nuevo documento: {'mail': 'bWFyaWVsLjc4Ljg2Q6hvdG1haWwuy29t', 'pass'
processing...
Mail connection to mariel.78.86@gmail.com is now
Inbox check to mariel.78.86@gmail.com for DIFA and document DIFA is now
Inbox check to mariel.78.86@gmail.com for Pulling and document Pull is now
Inbox check to mariel.78.86@gmail.com for TD and document Tear Down is now
Inbox check to mariel.78.86@gmail.com for Match and document Matching is now
processing...
Mail connection to mariel.78.86@gmail.com is now
Inbox check to mariel.78.86@gmail.com for DIFA and document DIFA is now
Inbox check to mariel.78.86@gmail.com for Pulling and document Pull is now
Inbox check to mariel.78.86@gmail.com for TD and document Tear Down is now
Inbox check to mariel.78.86@gmail.com for Match and document Matching is now
processing...

```

*Figura 28 Flujo automatizado de chequeo de correo en consola*

El proceso de chequear el mail se aplica una sola vez se llama al método de chequeo de mail, dependiendo el servicio de correo electrónico. Lo cual se definió así ya que luego se buscaría realizar el chequeo para múltiples usuarios a la vez, aplicando *Pool* y monitoreo constante de la base de datos. Es en el script del flujo automatizado donde se realiza una espera de un segundo y se vuelve a chequear el correo de cada usuario registrado y con asuntos nuevamente.

En este punto, se definió una función para el monitoreo constante de la base de datos, la cual verifica si ha habido un nuevo usuario o se ha modificado uno. De esta manera, revisa en cada caso si el usuario tiene las credenciales completas para ingresar al correo y si tiene al menos un asunto por chequear. Cada usuario que cumpla esas condiciones se lo añade a un diccionario, el cual contiene de clave el id del usuario en la base de datos y sus credenciales y asunto como valores. Posteriormente, se crean hilos por cada ítem guardado en el diccionario, ejecutando el chequeo del mail de forma paralela para cada usuario, usando *Pool* de *multiprocessing*. Cuando se ha chequeado por una vez el mail en busca de algún correo para todos los usuarios se espera un segundo, se vuelve a

chequear la base de datos por cambios y se vuelve a ejecutar en diferentes hilos el chequeo del correo para cada usuario otra vez. Cabe mencionar que, de haber un usuario nuevo o modificado, que no tiene credenciales completas ni asunto, se lo descarta.

```
Mail connection to mariel.78.86@gmail.com is now
Inbox check to mariel.78.86@gmail.com for DIFA and document DIFA is now
Inbox check to mariel.78.86@gmail.com for TD and document Tear Down is now
Documento modificado: HTJuQKuusBVQ1F5gAKI4iNHKb0o1
Datos antiguos: 2
Datos nuevos: {'mail': 'bWFyaWVsLjc4Ljg2Q6dtYWlsLmNvbQ==', 'pass': 'f24085d77e
Inbox check to mariel.78.86@gmail.com for Match and document Matching is now
processing...
Mail connection to mariel.78.86@gmail.com is now
Inbox check to mariel.78.86@gmail.com for DIFA and document DIFA is now
Inbox check to mariel.78.86@gmail.com for Match and document Matching is now
Inbox check to mariel.78.86@gmail.com for TD and document Tear Down is now
Inbox check to mariel.78.86@gmail.com for Pulling and document Pull is now
processing...
```

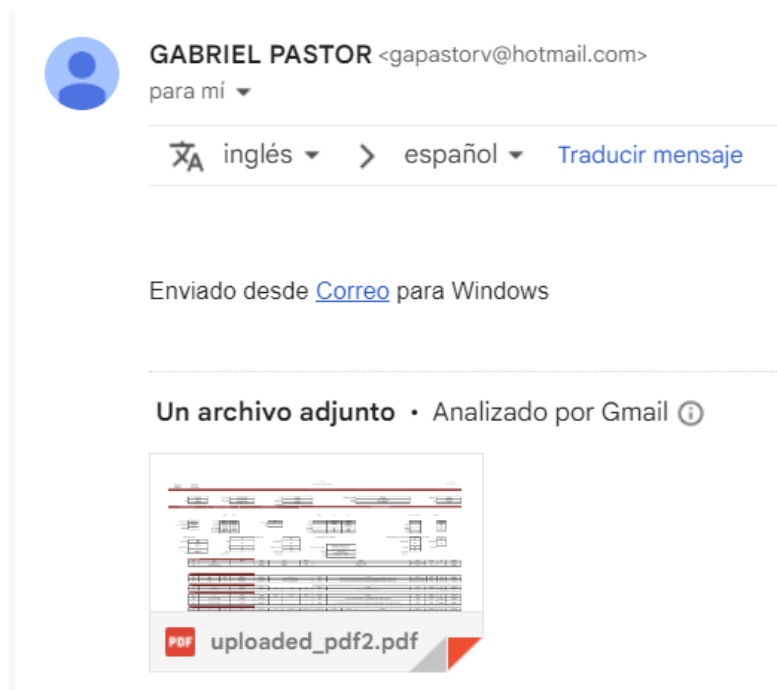
*Figura 29 Cuando un usuario nuevo aparece sin cumplir los requisitos se descarta*

Aquí, cuando se encuentra un asunto de interés en algún correo, se procede a extraer los datos importantes, según lo haya definido el usuario. Los parámetros se obtienen con un objeto de la clase de Firebase, el cual extrae dichos parámetros tanto para campos de texto como para tablas. En el caso del texto, compila cada campo en una tabla con el título ‘General’, que indica que esos son los campos de texto extraídos y son principalmente datos base y generales del reporte. En los reportes de fallas estos campos de texto serían el tiempo de vida útil (Run Life), identificador del pozo, el campo, la fecha de instalación del equipo, la fecha de parada, la fecha de pull y la razón del pulling.

En el caso de las tablas, estas se extraen y se les adiciona una columna donde se le adjudica a cada registro el identificador del pozo, lo cual sería fundamental para filtrar la información extraída para generar gráficos y tablas compiladas. Finalmente, esa información se guarda en la base de datos.

```
C:\Users\gapas\AppData\Local\Programs\Python\Python310\python.exe C:\Users\gapas\PycharmProjects\streamlit\mail.py
Subject: mailer
Sender: gapastorv@hotmail.com
Attachments:
- Saved Attachment: C:\Users\gapas\PycharmProjects\streamlit\uploaded_pdf2.pdf
```

*Figura 30 Obtención de mail bajo el asunto 'mailer'*



*Figura 31 Correo recibido con el asunto 'mailer'*

### 5.3. Presentación de datos extraídos

Cuando la información se extrae, de algún modo el usuario sentiría interés por ver que datos se han obtenido hasta el momento. Esto es importante puesto que puede haber información que luzca irrelevante, pero que a la persona encargada de la cuenta le sean de apoyo para tener datos adicionales, y en ocasiones por brindar un aporte más al análisis de causa raíz.

De esta manera, el usuario entra a la sección de Datasets, que es la página donde se presentan las tablas. Para ello, el sistema llama a una función DataManager, la cual se

encarga de extraer primero los tipos de reporte y asunto para un usuario en específico. Luego, esa información la usa para entrar a cada documento existente de cada reporte y asunto extraído hasta el momento, almacenando dicha información en un diccionario, donde la clave es el nombre del tipo de reporte, y el valor la tabla extraída y concatenada hasta la fecha.

Para este proceso se generó un objeto de la clase Firebase, el cual se usó para obtener una lista de los tipos de reportes con datos extraídos hasta el momento. Posteriormente, en un lazo se va iterando a cada uno de esos valores, y accediendo a los documentos dentro de las colecciones de estos, aquí se insertó el nombre del reporte para acceder, junto con los nombres de documentos y colecciones previas. Con los datos de cada documento, que implican las tablas que se solicitó extraer el usuario y que se han extraído, se decidió almacenar esos datos en un diccionario, donde la clave fue el nombre del reporte. Finalmente, se insertó esta información dentro de tabs para cada tipo de reporte, usando el componente de tabs de *streamlit*, y así pudiendo revisar la información existente.

**Datasets**

**intake-uploaded\_pdf2.3.pdf**

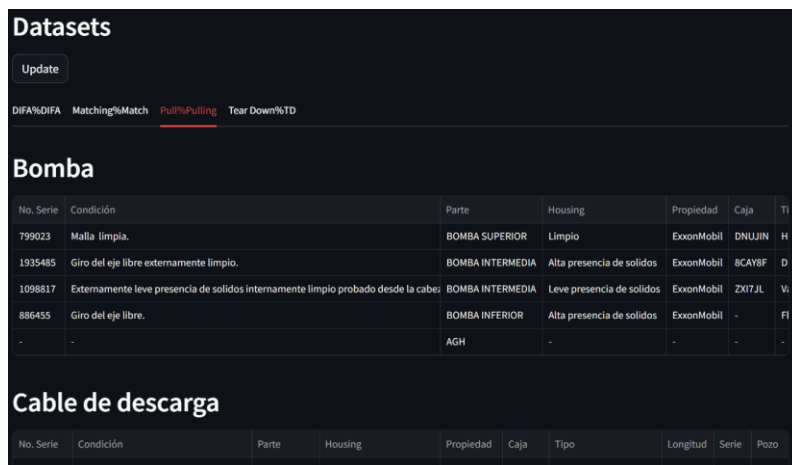
No. Serie	Serie	Tipo	No. P arte
4KN6B00917	417	VGSA D18/50	100,853,278

**protector-uploaded\_pdf2.3.pdf**

Condición	Unnamed: 0	2da Cam ara Ext/Ir
Giro del eje libre Cabe za con fluidosdelpozo,pe ndiente verificar cámara rasen Art ce	None	-
Limpio	417	-

*Figura 32 Presentación de tablas (v2)*

Adicionalmente, se incluyó la misma extracción para datos generales, es decir, datos que no implicaran tablas sino campos de texto. Esta implementación ha sido clave para poder obtener atributos generales sobre un reporte específico.



The screenshot shows a web interface titled 'Datasets' with a dark theme. At the top, there is an 'Update' button and a navigation menu with items: 'DIFA%DIFA', 'Matching%Match', 'Pull%Pulling' (highlighted in red), and 'Tear Down%TD'. Below the menu, the section 'Bomba' is displayed with a table containing the following data:

No. Serie	Condición	Parte	Housing	Propiedad	Caja	TI
799023	Malla limpia.	BOMBA SUPERIOR	Limpio	ExxonMobil	DNUJIN	H
1935485	Giro del eje libre externamente limpio.	BOMBA INTERMEDIA	Alta presencia de solidos	ExxonMobil	8CAY8F	D
1098817	Externamente leve presencia de solidos internamente limpio probado desde la cabe	BOMBA INTERMEDIA	Leve presencia de solidos	ExxonMobil	ZX17JL	V
886455	Giro del eje libre.	BOMBA INFERIOR	Alta presencia de solidos	ExxonMobil	-	FI
-	-	AGH	-	-	-	-

Below the 'Bomba' table, there is a section titled 'Cable de descarga' with a table that is partially visible, showing columns for 'No. Serie', 'Condición', 'Parte', 'Housing', 'Propiedad', 'Caja', 'Tipo', 'Longitud', 'Serie', and 'Pozo'.

*Figura 33 Presentación de tablas en la sección de Datasets, para Pull*

#### 5.4.Creación de gráficos de la información extraída

Tal como en la sección de tablas, esta sección es fundamental para que el usuario pueda ver los gráficos y tablas con información relevante sobre algún pozo específico, y con esto se pueda generar reportes de RCA. Sin embargo, esta sección es específicamente enfocada en un grupo de información, de modo que las tablas de que extraigan deben cumplir con que haya valores numéricos para presentar datos de eficiencia de bombas y motor, a diferentes frecuencias. Además, que la tabla de producción debe si o si componerse de datos numéricos, para poder realizar los gráficos de líneas.



**Dashboards**  
Web: EYE-007

No data was extracted or there is no data to show

**Pull**

**Cable de descarga**

No. Serie	Condición	Parte	Housing	Propiedad	Caja	Tipo
0	6UBAIFEI	Malla limpia.	ESP BOOH	Limpio	ExxonMobil	DNLUJIN Head PMP
1	MUJYPIN1	Giro del eje libre esternamente limpio.	CABEZA FDP	Alta presencia de sólidos	ExxonMobil	ICAYBF Discharge Presu

**Bomba**

No. Serie	Condición	Parte	Housing	
0	799023	Malla limpia.	BOMBA SUPERIOR	Limpio
1	1335485	Giro del eje libre esternamente limpio.	BOMBA INTERMEDIA	Alta presencia de
2	1098817	Esternamente leve presencia de sólidos internamente limpio probado desde la cabeza	BOMBA INTERMEDIA	Leve presencia de
3	886455	Giro del eje libre.	BOMBA INFERIOR	Alta presencia de
4			AGH	

**DIFA**

**Antecedentes**

Fecha	Detalles	Pozo	Fecha extracción	
35	01/01/2021	Actividad en el pozo el día 2021-01-01: Actividad normal	EYE-007	12/12/2023
36	02/01/2021	Actividad en el pozo el día 2021-01-02: Actividad normal	EYE-007	12/12/2023
37	03/01/2021	Actividad en el pozo el día 2021-01-03: Actividad normal	EYE-007	12/12/2023
38	04/01/2021	Actividad en el pozo el día 2021-01-04: Actividad normal	EYE-007	12/12/2023
39	05/01/2021	Actividad en el pozo el día 2021-01-05: Actividad normal	EYE-007	12/12/2023
40	06/01/2021	Actividad en el pozo el día 2021-01-06: Actividad normal	EYE-007	12/12/2023
41	07/01/2021	Actividad en el pozo el día 2021-01-07: Actividad normal	EYE-007	12/12/2023
42	08/01/2021	Actividad en el pozo el día 2021-01-08: Anomalia encontrada	EYE-007	12/12/2023
43	09/01/2021	Actividad en el pozo el día 2021-01-09: Actividad normal	EYE-007	12/12/2023
44	10/01/2021	Actividad en el pozo el día 2021-01-10: Actividad normal	EYE-007	12/12/2023

Figura 34 Muestra de visualizaciones de tablas de Pull y DIFA para un pozo

Aquí se extrae los nombres de los pozos extraídos y almacenados hasta la fecha, verificando en cada documento y tablas, de cada reporte con datos extraídos y almacenados. Posteriormente, se selecciona por defecto uno de estos pozos en un selectbox de *streamlit*. Con un pozo seleccionado, se procede a obtener los datos del equipo de los reportes de pull, en formato de dataframe y de la última fecha en la que se extrajeron datos para dicho pozo. Además de ello, se toma una compilación de antecedentes y eventos de DIFA para dicho pozo, lo cual es general y de todo lo extraído. Por otro lado, se toman los hallazgos de la última fecha de extracción para dicho pozo, como también los registros de eficiencia de bomba y motor más recientes. Finalmente, se toman los registros de producción de DIFA más recientes.

Para generar las gráficas, se utilizó la librería *plotly*, con la cual se crearon los gráficos de líneas tanto para producción como datos de eficiencia de motor y bomba, brindados por los reportes de Match.

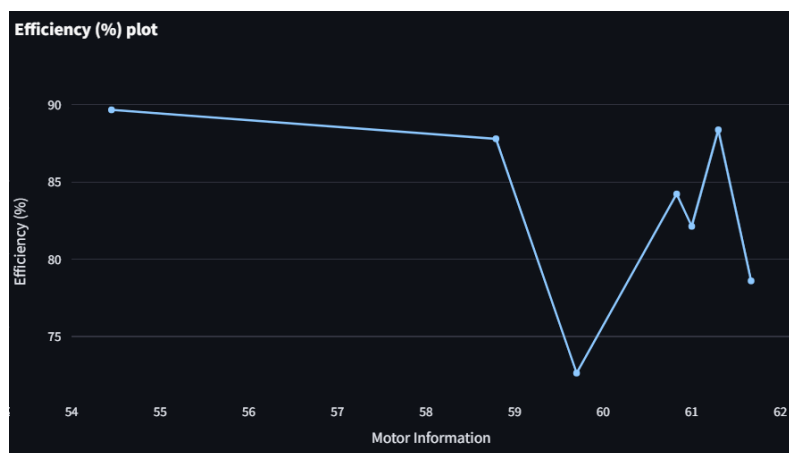


Figura 35 Gráfico de eficiencia de motor, del reporte de Match

Cabe mencionar también que para filtrar los datos se eliminaba primero los valores no numéricos de la tabla, luego los valores nulos y al final sacando solo los datos que tengan el valor de fecha máximo.

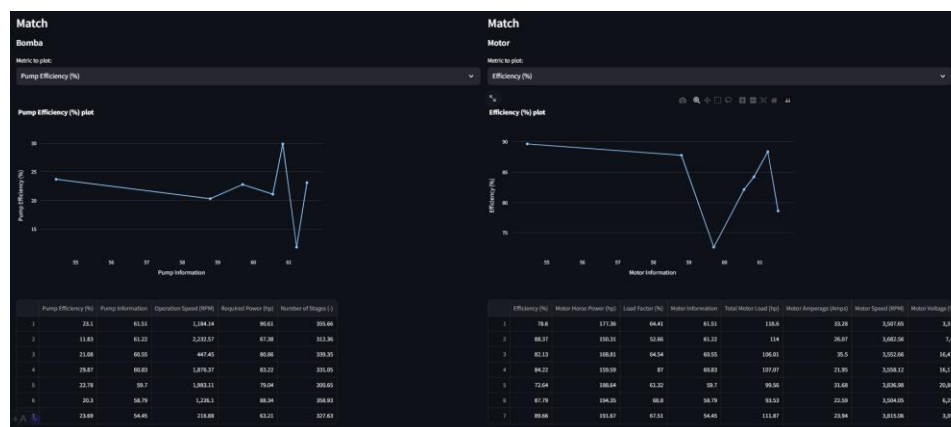
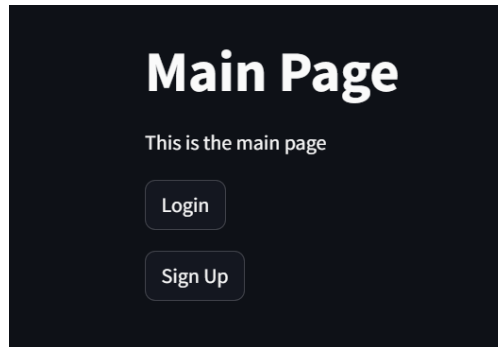


Figura 36 Muestra de gráficos lineales de motor y bomba, de Match

## 5.5. Páginas de inicio de sesión y registro

Un sistema de logeo se desarrolló para este proyecto debido a que, cuando se opera con datos de campo en el área de petróleo y gas, siempre se presentan datos confidenciales. Es por esto que se estableció una capa o página de acceso al sistema, con el fin de que solo un grupo selecto de personas puedan gestionar la información extraída y la puedan analizar para definir causas de eventos de fallas en el equipo de levantamiento artificial.

Esta sección se compone de las páginas de inicio de sesión y registro, las cuales utilizan la base de datos de Firebase para almacenar los datos de los usuarios para poder acceder al sistema.

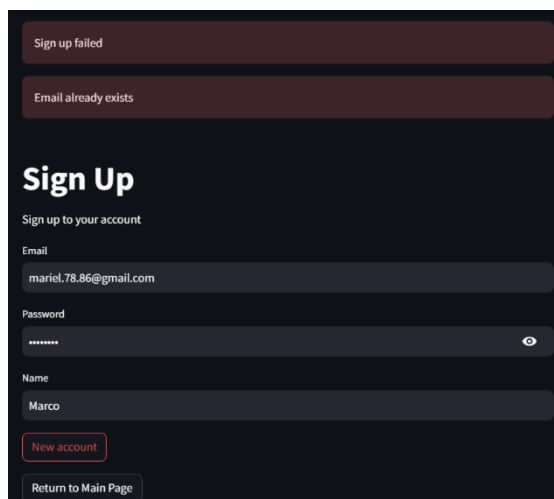


*Figura 37 Página inicial*

A dark-themed screenshot of a "Sign Up" form. The title "Sign Up" is at the top. Below it, the text "Sign up to your account" is displayed. The form has three input fields: "Email" with the value "mariel.78.86@gmail.com", "Password" with masked characters "\*\*\*\*\*" and a visibility toggle icon, and "Name" with the value "Marco". At the bottom, there are two buttons: "New account" (highlighted in red) and "Return to Main Page".

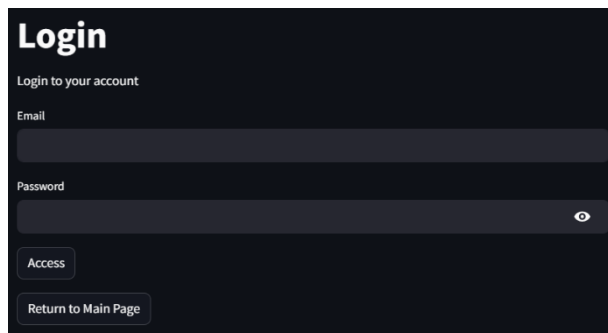
*Figura 38 Página de registro*

El funcionamiento de ambas páginas es sencillo: cuando el usuario se desea registrar ingresa el correo electrónico con nombre de usuario y contraseña incluidos. En el caso de que ya haya un usuario con nombre y correo similares, salta la notificación de usuario duplicado, pero no serlo se pasa a la pantalla de inicio de sesión.



The image shows a dark-themed 'Sign Up' form. At the top, there are two red error messages: 'Sign up failed' and 'Email already exists'. Below these, the form title 'Sign Up' is displayed in white. Underneath, the instruction 'Sign up to your account' is shown. The form contains three input fields: 'Email' with the value 'mariel.78.86@gmail.com', 'Password' with masked characters '\*\*\*\*\*' and a toggle icon, and 'Name' with the value 'Marco'. At the bottom, there are two buttons: 'New account' (highlighted in red) and 'Return to Main Page'.

*Figura 39 Notificación de correo duplicado*



The image shows a dark-themed 'Login' form. The title 'Login' is at the top in white. Below it, the instruction 'Login to your account' is shown. The form has two input fields: 'Email' and 'Password' with a toggle icon. At the bottom, there are two buttons: 'Access' and 'Return to Main Page'.

*Figura 40 Página de inicio de sesión*

## 5.6. Inconvenientes y nuevas versiones

### 5.6.1. VI

Esta versión comprende los componentes más importantes de la aplicación, que serían la creación de la clase de MailChecker y Canvas, además de la interfaz gráfica de la página de definición de parámetros de extracción por parseo. Sin embargo, no contiene todos los elementos que se establecieron dentro del diagrama de casos de uso, esto ya que aún le faltaba desarrollar el registro, el inicio de sesión y la generación de Dashboards.

La primera versión se limita básicamente a una sola página, la de parámetros de extracción de datos de un documento subido a la plataforma. Además de ello, contiene la conexión a la base de datos y el chequeo de mails bajo asunto (siendo esto importante para el flujo automatizado).

### **5.6.2. V2**

Esta versión se enfoca principalmente en agregar las páginas de logeo, que serían el inicio de sesión y registro. Esto se decidió agregar entendiendo que la información en el área de petróleo y gas es de uso exclusivo para un conjunto del personal de una empresa en el área. Dado esto, se añadió esta sección con el fin de que los datos puedan ser visualizados y asequibles solamente para usuarios que hayan definido los parámetros para la extracción.

Ante esto, también se decidió incluir la sección de presentar tablas de información, la cual implica básicamente presentar los datos extraídos por el flujo automatizado.

### **5.6.3. V3**

Esta versión mejora el flujo automatizado con el monitoreo continuo de la base de datos, de esta forma se chequea por nuevos usuarios y asuntos registrados. Se incluyó una sección de definición de parámetros de extracción para campos de texto y se mejoró la extracción de datos, incluyendo segmentos de código con *pandas*, para eliminar registros vacíos.

Por otro lado, se incluyó títulos específicos para cada tabla y campo de texto que se extraiga de un reporte dado. Esto significa que cuando se seleccione un área para extraer texto o tablas en un documento, se les atribuye un título referente a un dato o tabla del reporte al cual se le quiere definir qué información extraer. Por ejemplo, cuando el usuario

seleccione un área para extraer una tabla en la primera página, se determina que la tabla extraída es del equipo de descarga de levantamiento artificial.

#### **5.6.4. V4**

Finalmente, esta versión ya incluye un flujo automatizado mejorado y la página de gráficos y tablas relevantes (Dashboards). Adicionalmente, se modificó la atribución de nombres o títulos a las tablas a extraer, de manera que el usuario ahora puede determinar el orden en el que estas aparecen con base en las coordenadas de extracción definidas.

### **5.7.Repositorio**

Este proyecto implicó la implementación de componentes en diferentes versiones, todas ellas en GitHub. De esta forma, se presentan en diferentes carpetas los avances realizados durante el periodo de tiempo permitido. Cabe recalcar que ciertos componentes, como el archivo de credenciales para inicializar o acceder a la base de datos de Firebase, no se encuentran disponibles. Esto es debido a que puede comprometerse un espacio de pruebas, y la idea sería evitar que usuarios alteren la información del sitio accidental o intencionadamente.

El repositorio donde se almacena la información se encuentra en el siguiente link:

[https://github.com/gapastorv/st\\_rca\\_project](https://github.com/gapastorv/st_rca_project)

## **6. EVALUACIÓN Y VALIDACIÓN**

### **6.1.Pruebas de rendimiento y precisión de extracción de datos**

#### **6.1.1. Extracción de tablas**

Dentro de las implementaciones realizadas para la extracción, se obtuvo los siguientes resultados:

- V1: Se tenía problemas con las coordenadas, puesto que fue necesario redimensionar cada página para que pueda visualizarse completa dentro del canvas, fue necesario entonces aplicar tablas adicionales para convertir las coordenadas a la dimensión del archivo procesado. Por medio de esto, se tomó en cuenta las variaciones en la escala de la selección de área realizada en el canvas, con el fin de cambiar el resultado de tabula en la vista previa cuando cambie la selección.
- V2: Se ve aún ciertos inconvenientes en la extracción, lo cual se debe principalmente a columnas vacías y sin nombre. Dichas columnas se presentaban en ocasiones en las vistas previas, y complican para esta versión la presentación de los datos. Ante esto, se va a tomar en cuenta los medios posibles para eliminar columnas vacías.
- V3: Se presentó problemas para cuando se quiera extraer un campo de texto en una página posterior, especialmente por el orden en el que se definieron los títulos de cada selección de área para la extracción.
- V4: Se presenta una pequeña demora en la extracción de información, especialmente de tablas. Aunque no sería algo totalmente grave, en el caso de que la cantidad de tablas a extraer aumente progresivamente, el tiempo de extracción podría complicarse.

### ***6.1.2. Almacenamiento de datos***

Dentro de las implementaciones, se ha tenido en cuenta las siguientes características:

- V1: La clase implementada de Firebase realiza una conexión con el almacenamiento en la nube, e implementa la ruta para almacenar la información de

la forma 'data/mail/tipo-documento/tablas/'. En las primeras pruebas han ocurrido errores que implicaban el acceso a documentos o colecciones, principalmente por la cantidad de elementos para definir la ruta donde se guardaría la información. Fue gracias a esto, que los componentes de la ruta de almacenamiento serían pasados de forma individual, y no como una sola ruta.

- V2: Esta versión no implicó realmente inconvenientes en el almacenamiento, aunque fue necesario analizar como extraer la información. Puesto que Firebase, especialmente el componente de Firestore, implementa un sistema de colecciones/documentos, el poder acceder a los documentos fue un componente difícil. Dado eso, se decidió guardar las rutas de almacenamiento del contenido de las tablas y texto de cada reporte, con el fin de tener mayor facilidad para obtener la información recopilada y que sea mostrada en la sección de tablas.

### **6.1.3. Limitaciones**

Ante esto, el sistema obtuvo los siguientes inconvenientes para la versión 1.0:

- Detección de líneas: Las tablas legibles dentro del programa se caracterizaban por delimitar cada celda con líneas, siendo un problema la delimitación de celdas por espacios en ciertos documentos. A pesar de que tabular puede permitirse la delimitación por espacios, el problema radicaba principalmente en datos que puedan pertenecer a una misma celda, pero segmentados en mini párrafos, complicando la legibilidad.
- Celdas inexistentes: En el testeado de las implementaciones, ocurre que en la vista previa se pueden obtener tablas con columnas extras, las cuales no contienen datos, de la misma forma para registros vacíos. Esto puede darse principalmente al



seleccionar el área de modo que tabula supone la existencia de celdas por delimitación por líneas. Es necesario que se haga una selección correcta de la información con el fin de que en la extracción sobre otros documentos no se inserte información innecesaria.

- Área de selección: Puesto que no se puede implementar un visualizador de PDF, para ver cada página, se tuvo que recurrir a transformar cada página de un archivo subido a imagen. A pesar de ello, se han tenido dificultades en el funcionamiento del sistema, tales como imágenes de baja calidad o los archivos de imagen perdidos, de modo que el canvas se encuentra vacío.

De igual forma, el sistema obtuvo los siguientes inconvenientes para la versión

2.0:

- Extracción de campos de texto: Se planteó usar Tabula nuevamente, en conjunto con RegEx para hacer parseo sobre el texto completo extraído. Curiosamente, los resultados fueron exitosos con ello, de modo que se tuvo que cambiar la forma de operar los datos.
- Documentos estructurados: A pesar de que se trata de un sistema para extraer información clave de un reporte, implementando parseo, no puede actuar efectivamente cuando el documento vario su estructura y número de páginas. Esto implica que para que los parámetros de tablas y campos a extraer funcionen lo mejor posible, solamente se puede trabajar con reportes con diseño y estructura fija, específicamente con aquellos similares a formularios.

Por otro lado, para las versiones 3.0 y 4.0 se consideró lo siguiente:

- Aplicable solo para documentos estructurados.

- Tiempo de extracción aceptable, pero crítico a la hora de tener tantos parámetros para la extracción
- No implica un sistema de detección de tablas.
- Sin muchos gráficos aplicados en la sección de Dashboards.
- Restringido su uso para cuatro tipos de documentos.
- Alta demora en la carga de información de la base de datos.
- Área para definir parámetros de extracción con las páginas en resolución baja (puede afectar al usuario)
- Flujo automatizado corriendo en consola solamente.

## **6.2.Evaluación del flujo automatizado**

### **6.2.1. Complejidades**

Para la versión 1.0, se han tenido los siguientes inconvenientes:

- Sistema desintegrado: debido a que faltan las implementaciones de la página principal, no puede haber una conexión directa con las cuentas de usuario y sus credenciales, con el fin de chequear el mail por correos con documentos de reportes a extraer la información.
- Tiempo agotado: en la implementación hecha, se han tenido complicaciones en el chequeo de mails, principalmente debido al envío de correos por medio de la clase de mail creada. Esto básicamente implica que cuando se quiera enviar un correo por medio del sistema a un usuario (no de un usuario con el adjunto al usuario registrado, sino del sistema mismo al usuario registrado), este no llegaría y el tiempo de espera se agotaría, parando a la aplicación.

Para la versión 2.0, se presentaron algunos inconvenientes aún por el momento:

- Sistema desintegrado: a pesar de que el funcionamiento del sistema se mantuvo y no fue necesario más que solamente definir ciertos detalles (como permitir que solo se trabaje con Gmail y Outlook), aun esto se ejecuta aparte, siendo clave para la extracción de datos de reportes y el almacenamiento de la información.
- Fallos en la extracción de tablas: esto se ha visualizado mayormente en la sección de visualización de tablas, siendo un componente aún por analizar y resolver, teniendo gran relación con las columnas vacías en la definición de parámetros.

## 7. CONCLUSIONES

### 7.1. Resumen de los resultados obtenidos

Cuando se buscó desarrollar un sistema integrado, que involucrara una aplicación web y un flujo automatizado de chequeo en el sistema de correo electrónico, se quería llegar a tener buenos resultados en cada instante. En las primeras versiones se buscaba asegurar el buen funcionamiento de la extracción de datos, la buena conexión con la base de datos y que haya un funcionamiento aceptable en el chequeo del correo electrónico. Tras las primeras pruebas, se pudo constatar la necesidad de cumplir con los objetivos de extracción de datos, la correcta conexión de la aplicación web y flujo con la base de datos y el propio proceso automatizado de chequeo del correo electrónico. Sin embargo, en las primeras pruebas se verifico principalmente el buen funcionamiento de los métodos de extracción de la información y de almacenamiento de parámetros de extracción y datos extraídos en la base de datos, de manera que no se hizo una prueba directa con la conexión al correo electrónico y el chequeo constante en este punto aún. Por otra parte, se obtuvo

problemas al solo usar *tabula-py* en la primera versión, puesto que se obtenían tablas con columnas vacías y registros nulos, dificultando la buena extracción de la información.

De forma iterativa, surgieron diferentes ideas para determinar el funcionamiento del sistema de chequeo de correo electrónico, del flujo automatizado. Fue en este punto que se usó *imaplib* y se chequeaba de forma constante la bandeja de entrada estableciendo un tiempo de reposo de un segundo, para que luego se volviera a conectar y chequeara por correos de interés (esto fue lo que se añadió al método de chequeo de correo electrónico). Por otra parte, el componente de extracción de datos, ya definido en la versión 1.0 se mantuvo, para realizar pruebas sobre un archivo adjunto que llegara por correo. En este caso, el flujo automatizado comprendía de verificar un mail con asunto de interés cuando llegara, verificar el contenido de este y, si tiene adjuntos, ejecutar los métodos de extracción con *tabula-py*, pero antes extrayendo de la base de datos los parámetros de extracción (estas eran las coordenadas de las áreas de extracción junto con el número de página). Finalmente, el resultado de la extracción del documento se guardaba en la base de datos de Firebase. Aplicando esta idea de cómo debía funcionar el flujo automatizado, se pudo confirmar el correcto chequeo de mails con asuntos específicos, la extracción de datos funcionando sin inconvenientes y su posterior almacenamiento.

Una prueba importante que se hizo desde la versión 1.0 fue con el canvas en la aplicación web, aunque se puede visualizar en cierta medida el contenido de cada página, aun se tenían problemas poder leer ese mismo contenido para un documento más complejo. Esto debido a que el espacio que el canvas ocupa en la página es limitado, de modo que la resolución de la imagen se verá afectada. Fue por ello que se estableció una dimensión los

más extendida posible (700 pixeles x 700 pixeles), y aun así la resolución no pudo mejorar lo suficiente.

En la versión 2.0, hubo un enfoque en el sistema de logeo y el procesamiento de la información en relación a usuarios específicos. De igual forma, se pudo agregar una sección para extraer campos de texto, lo cual tuvo un buen nivel de eficacia, aunque esta se agregó finalmente para la versión 3.0. Esta sección ya tenía incluido *re* para descartar expresiones regulares y así limpiar el texto tomado tras usar *tabula-py*, y solo obtener el texto deseado. Por otra parte, se ha incluido la sección de datos extraídos, en la cual se presenta la información obtenida hasta el momento. Esta sección, debido a los inconvenientes de extracción de tablas anteriormente, mostraba valores nulos y registros distorsionados. Por ejemplo, para un registro de propiedades de cabeza de descarga, de un reporte de Pull, mostraba el valor de diámetro de cabezal en la columna de observaciones.

Finalmente, para las versiones 3.0 y 4.0 se ha visto una mejora significativa en el proceso de chequeo del correo, cuando se modificó dicho chequeo para más de un usuario y en paralelo. Es en estas versiones en las cuales se pudo ver mejoras en la extracción de datos aplicando *pandas* para descartar registros nulos. Por otra parte, se agregó código para trabajar con tablas que contenían los valores de columnas como filas, es decir, código para trasponer una tabla extraída. Sin embargo, se han encontrado al final problemas para limpiar los valores nulos de las tablas traspuestas, dado que, en las pruebas de extracción de datos en el flujo automatizado, se veía en consola una vista previa de cómo se guardaría la tabla traspuesta, y estas contenían al menos una columna sin nombre y con datos nulos.

Finalmente, se agregó la sección de visualizaciones (Dashboards) en la versión 4.0. Para las pruebas de funcionamiento de este componente, se pudo confirmar el correcto

funcionamiento de las tablas y gráficos lineales, aunque requiere de una limpieza extra de las tablas almacenadas en la base de datos, considerando que existe siempre la posibilidad de que haya registros nulos, y solo se deben usar los registros numéricos. Esto se debe a que, como se trabajó con reportes de Match específicos, estos contenían tablas que debían trasponerse y, tal como se mencionó para este tipo de tablas, aún se ven registros nulos.

## **7.2. Contribuciones del proyecto en el área de petróleo y gas**

Automatizar la extracción y almacenamiento de datos pertinentes a fallas que surgen en equipos para la extracción de petróleo y gas. La idea radica principalmente en que se obtenga los datos más importantes sobre reportes, y que esta información pueda almacenarse para luego poder ser representada de diferentes maneras, ya sea como gráficos de líneas o de barras que muestren valores de producción y eficiencia del equipo en operación en un yacimiento, por ejemplo. Serían estos los puntos más importantes que le dan importancia a este proyecto para que se pueda aplicar en el área de petróleo y gas.

Con este proyecto se busca no solo optimizar procesos de recolección de información, sino también reducir el tiempo de procesamiento de la información para poder ser representada y tomar decisiones sobre las operaciones que se hagan en campos de producción petrolera. Pues la cuestión de fallas en el sector de petróleo y gas es una de las más importantes, y a las cuales se les debe prestar mucha atención dada la cantidad de incidentes que aún siguen ocurriendo hoy en día en yacimientos y zonas cercanas a estos, que involucran mayormente derrames de líquido o incidentes por fallas mecánicas.

La idea de un flujo automatizado de chequeo de reportes, que lleguen por correo, vino a la luz debido a que este es el medio de intercambio de información más importante, dentro de una empresa sea cual sea, y porque este medio es clave para recibir reportes de

fallas. Para este sector el manejo de la información es primordial no solo por la confidencialidad de la misma, sino también por el aporte que puede brindar cada reporte de fallas con datos relevantes, ya sea sobre antecedentes o eventos ocurridos en las operaciones en un campo de producción petrolera o sobre los detalles y propiedades del equipo instalado. Cada uno de esos datos permite al personal especializado verificar las posibles causas de eventos de fallas en la producción, ayudando en la toma de decisiones más acertadas y en la mitigación de riesgos al operar en campos petroleros. Esto se puede lograr por medio de la regularización de normas de seguridad y protocolos de inspección de actividades. Es con base en esto que la recolección de datos importantes de reportes ayuda a mejorar el rendimiento de las actividades de producción en el área de petróleo y gas, y da paso a mejores prácticas en las operaciones en yacimiento petroleros.

### **7.3.Recomendaciones para un trabajo futuro**

La creación de una aplicación web que aplica parseo puede ser el primer paso para desarrollar todo un sistema de extracción de datos de forma automatizada. Sin embargo, este proyecto se ha visto limitado por tiempo y la ausencia de medios que faciliten que la interacción del usuario con el sistema se torne más amigable. Es por esto un paso adicional en este proyecto y a futuros proyectos implicará lo siguiente:

1. Insertar OCR sobre secciones específicas de un documento, por medio de capturas de pantalla. Este medio puede ser la clave para restringir la extracción de datos y reducir la cantidad de información a almacenar de un documento, además que permitirá trabajar sobre documentos que sean escaneados.
2. Se puede extender el número de reportes de los cuales se quiere extraer información, puesto que actualmente el proyecto solamente funciona para cuatro

tipos de reporte. Se puede tener otro tipo de reportes de los cuales se necesite extraer información importante, ya sean reportes de Rig Time, en los cuales se almacenan más detalles de la maquinaria utilizada y las actividades realizadas en campo, de forma diaria.

3. Es recomendable aplicar modelos de machine learning o deep learning para identificar los campos de texto a extraer al momento de que un reporte llegue por correo, por medio de detección de patrones específicos dentro del documento y así extienda el uso del sistema a documentos semiestructurados, dado que el sistema actual solo funciona para documentos estructurados.
4. Se podría mejorar la resolución de imágenes en el canvas, tomando en cuenta que cada página del PDF se transforma a imagen para poder renderizarse, y así se seleccionen las áreas de extracción de datos.
5. Por el momento, la sección de visualizaciones (Dashboards) solamente contiene gráficos lineales y mayormente tablas con datos extraídos y recolectados. Por lo que se puede expandir esta página de la aplicación web, con el fin de que también se puedan presentar datos en diagramas de frecuencia, para ver qué tipo de razón pull (causa de extracción del equipo instalado para inspeccionarlo) es la más frecuente, como un caso. Además, esto sería de mayor ayuda para el personal encargado ya que daría información clave de forma breve y directa.
6. Se podría modificar el flujo automatizado, de manera que se cree una interfaz gráfica por la cual se puede visualizar que hilos de cuentas de usuario están en el instante chequeando el correo por nuevos mensajes con asuntos de interés, Además



que esto podría ser clave para que un administrador tenga la capacidad de gestionar a que usuarios se debe chequear el correo.

Estas extensiones del proyecto incrementarían su utilidad y extensibilidad en el área de petróleo y gas.

## 8. REFERENCIAS BIBLIOGRÁFICAS

- Agarwal, S. (n.d.). “Regular Expression modules,” in Python re(gex)?: a magical tool for text processing (pp. 7-10). Essay.
- Bell, S., Marlow, T., Wombacher, K., Hitt, A., Parikh, N., Zsom, A., & Frickel, S. (2020). Automated data extraction from historical city directories: The rise and fall of mid-century gas stations in Providence, RI. *PLoS One*, 15(8), e0220219.
- Bhatt, A. (2022). Document Automation Using Artificial Intelligence. *International Journal for Research in Applied Science and Engineering Technology*, 10(9), 1365–13169. <https://doi.org/10.22214/ijraset.2022.46839>
- Batt, S., Grealis, T., Harmon, O., & Tomolonis, P. (2020). Learning Tableau: A data visualization tool. *The Journal of Economic Education*, 51(3-4), 317-328.
- Castro-Castro, J. D., & Cendales-Ladino, E. D. (2019). Casos aplicados del análisis de causa raíz: revisión. *Ciencia e Ingeniería Neogranadina*, 29(1), 95-134.
- Comience con Cloud Firestore | Firebase. (n.d.). Firebase. <https://firebase.google.com/docs/firestore/quickstart?hl=es>
- Chacon Cruz, T. P., & Riaño Amaya, C. A. (2020). Análisis del sector petrolero en Colombia, carga tributaria y comparación con Perú, México y Ecuador.
- de la Cruz Ventura, A. (2019). ELABORACIÓN DEL ANÁLISIS CAUSA RAÍZ POR PRESENCIA DE FUGA EN TUBERÍA FLEXIBLE 1 ½” EN EL POZO AYOCOTE 3.
- Del Pezo Yagual, J. E. (2021). Screening para selección de levantamiento artificial aplicado a un pozo en un campo maduro perteneciente al oriente ecuatoriano (Bachelor's thesis, La Libertad: Universidad Estatal Península de Santa Elena, 2021.).
- Escobar, F. H., Ramírez, A. C., & Enciso, O. L. (2010). Software para interpretar registros de producción de pozos y su aplicación en Campos Petroleros. *Ingeniería y Región*, 7, 93-101.
- exchangelib. (2023, August 22). PyPI. <https://pypi.org/project/exchangelib/>
- Figeroa, R. (2013). Empresas regionales proyectan construir el primer equipo de “pulling” nacional. *Supledesarrollo*. Recuperado 13 de diciembre de 2023, de <https://www.supledesarrollo.com.ar/?p=1497>
- Georgiostrantzias. (n.d.). Crear Flujos de Escritorio - power automate. Power Automate | Microsoft Learn. <https://learn.microsoft.com/es-es/power-automate/desktop-flows/create-flow>

- Giraldo Ramírez, M. E., Álvarez Cadavid, G. M., & Navarro Plazas, C. D. P. (2020). Usos de TIC y software especializado en la investigación cualitativa. Un panorama. *Investigación bibliotecológica*, 34(84), 33-57.
- imaplib — Protocolo del cliente IMAP4. (n.d.). Python Documentation.  
<https://docs.python.org/es/3/library/imaplib.html>
- Jiménez Moreno, M. A., Hernández Barajas, J. R., Jiménez Hernández, J. D. C., & Plazas Quiroga, J. G. (2022). Software académico de control de pozos petroleros MAROGA. *Acta universitaria*, 32.
- Karnik, S., Yenuganti, N., Jusri, B. F., Gupta, S., Nirgudkar, P., Mohajer, M., & Malik, A. (2021, September). Automated ESP Failure Root Cause Identification and Analyses Using Machine Learning and Natural Language Processing Technologies. In *SPE Gulf Coast Section Electric Submersible Pumps Symposium?* (p. D022S001R002). SPE.
- Khattak, M. A., Zareen, N., Mukhtar, A., Kazi, S., Jalil, A., Ahmed, Z., & Jan, M. M. (2016). Root cause analysis (RCA) of fractured ASTM A53 carbon steel pipe at oil & gas company. *Case Studies in Engineering Failure Analysis*, 7, 1-8.
- Lagla Paneluisa, D. F. (2023). Análisis técnico-económico para optimizar el sistema de levantamiento artificial de un campo x mediante la utilización del software decisionspace (Bachelor's thesis, Quito: EPN, 2023.).
- Lesik, S. A. (2018). *Applied statistical inference with MINITAB®*. CRC Press. maggiesMSFT. (n.d.). Generador de Informes de power bi - power Bi. Power BI | Microsoft Learn.  
<https://learn.microsoft.com/es-es/power-bi/paginated-reports/report-builder-power-bi>
- Mayorga-Mayorga, H. S., & Reyes-Bueno, F. (2022). Análisis de Derrames de Petróleo en el Campo Ancón Mediante Sistemas de Información Geográfica. *Revista Politécnica*, 49(1), 53-60.
- Norsworthy, C., & NAVAL RESEARCH LAB WASHINGTON DC. (2022). Synthetic Data Generation Project for a Document Parsing AI.
- Power, B. I., Apps, P., & Automate, P. (2020). Microsoft Power Platform.
- tabula — tabula-py documentation. (n.d.). <https://tabula-py.readthedocs.io/en/latest/tabula.html>
- Trujillo Coral, M. A. (2018). Estudio de tendencias históricas de los parámetros de operación de los equipos de bombeo eléctrico sumergible del Bloque 61–Auca (Bachelor's thesis, Quito, 2018.).
- Zumba Rosero, J. A. (2014). Desarrollo de una aplicación para generar reportes personalizados utilizando software libre (Bachelor's thesis, Quito, 2014.).