



***MANUAL TÉCNICO PARA LA AUTOMATIZACIÓN DE INFORMES
TRIMESTRALES***

QUITO, 7 DE MARZO DE 2024

1. Introducción

1.1 Objetivo del Manual

Este manual técnico tiene como objetivo principal proporcionar una guía exhaustiva para la comprensión, uso y mantenimiento del sistema automatizado de gestión documental y reporte de operaciones de pozos petroleros. A través de este sistema, los usuarios podrán automatizar la generación de informes trimestrales, manejar eficientemente documentos relacionados con las operaciones de pozos, y automatizar la copia y gestión de documentos desde y hacia OneDrive y sistemas de archivos locales.

1.2 Alcance del Sistema

El sistema cubierto por este manual se ha diseñado para automatizar y optimizar varias tareas críticas dentro de la gestión de documentos y reportes de operaciones de pozos petroleros. Las capacidades del sistema incluyen:

- Conexión con Bases de Datos: Permite la extracción automatizada de datos operativos de pozos petroleros desde bases de datos corporativas para su análisis y reporte.
- Generación Automatizada de Informes: Facilita la creación de informes trimestrales de operaciones de pozos, incluyendo la selección de trimestres específicos, y el manejo de la información asociada a los pozos.
- Gestión Documental Avanzada: Automatiza la copia de documentos necesarios para el soporte de operaciones de pozos desde ubicaciones de OneDrive, directorios de la empresa y sistemas de archivos de red a destinos específicos, siguiendo reglas de nomenclatura y estructuras de directorios definidas.
- Reporte y Seguimiento: Ofrece herramientas para el seguimiento del estado de los documentos, incluyendo la generación de hojas de cálculo que resumen el progreso y completitud de los documentos necesarios para cada operación de pozo.
- Notificación por Correo Electrónico: Se procede con la creación de un flujo en Power Automate para la notificación automática del reporte generado en la parte de seguimiento.

Este manual está dirigido a ingenieros de producción, técnicos de documentación y cualquier personal involucrado en la gestión de operaciones y documentación de pozos petroleros que utilice o mantenga este sistema.

1.3 Importancia del Sistema

La implementación y uso adecuado de este sistema contribuyen significativamente a la eficiencia operativa, reduciendo el tiempo y esfuerzo requeridos para la gestión documental y la generación de informes. Al asegurar una documentación completa y oportuna, el sistema apoya la toma de decisiones informadas y la gestión efectiva de los recursos y operaciones de pozos petroleros.

2. Uso del Sistema

2.1. Estructura General del Sistema

Utilizando Python como lenguaje de programación base, el sistema se apoya en diversas librerías especializadas para conectar con bases de datos, procesar datos, manejar archivos y generar informes detallados.

Componentes Clave del Sistema

- **Conexión a Bases de Datos con pyodbc:**

pyodbc es una librería de Python que proporciona conectividad con bases de datos SQL Server. Dentro de nuestro programa, se utiliza para ejecutar consultas que extraen datos operacionales de pozos, facilitando su posterior análisis.

El siguiente código establece la conexión a la base de datos SHAYA, donde se almacenan los registros de operaciones de pozos:

```
# Información de conexión

server = 'ec0038app05'
database = 'SHAYA'
username = 'python_user'
password = 'python_user'

conn_str = f'DRIVER={{SQL Server}};SERVER={server};DATABASE={database};UID={username};PWD={password}'
```

El sistema comienza estableciendo una conexión segura con la base de datos corporativa. Utilizando *pyodbc*, se ejecutan consultas SQL predefinidas para extraer información específica sobre las operaciones de pozos, como fechas de operación, tipos de intervención y estados de documentación.

En el sistema, se utilizan diversas consultas SQL para extraer datos específicos necesarios para la generación de informes y la gestión documental de las operaciones de pozos. A continuación, se describen y explican las consultas proporcionadas en el código:

Consulta para Operaciones Opex

```
# Conexión a la base de datos y obtención de los DataFrames
conn = pyodbc.connect(conn_str)

# Consulta SQL
query0 = f"""
SELECT VENDOR_ID_TEXT, ITEM_NAME, START_WO, END_WO, ESTIMATED_DATE, PLAN_TYPE_TEXT, WO_NUMBER, WO_OPEX_TEXT
FROM VT_WELLJOBLOG_en_US
WHERE YEAR(END_WO) = {año_interes} AND PLAN_TYPE_TEXT = 'Opex' AND VERIFIED = 'True'
"""
```

Descripción: Esta consulta selecciona varias columnas relevantes para las operaciones Opex (gastos operativos) de la vista VT_WELLJOBLOG_en_US. Filtra las operaciones basadas en el año de finalización (END_WO), específicamente para el año 2024, y se enfoca solo en las entradas que han sido verificadas como True. El filtrado por PLAN_TYPE_TEXT = 'Opex' asegura que solo se recuperen las operaciones relacionadas con gastos operativos, excluyendo otros tipos de planificaciones financieras.

Consulta para Operaciones Capex

```
queryC = f"""
SELECT VENDOR_ID_TEXT, ITEM_NAME, START_WO, END_WO, ESTIMATED_DATE, WO_NUMBER
FROM VT_WELLJOBLOG_en_US
WHERE YEAR(END_WO) = {año_interes} AND PLAN_TYPE_TEXT = 'Capex' AND VERIFIED = 'True'
"""
```

Descripción: Similar a la consulta Opex, esta consulta se dirige a operaciones Capex (gastos de capital), extrayendo datos relevantes de la misma vista. La condición PLAN_TYPE_TEXT = 'Capex' diferencia esta consulta al centrarse en actividades que implican inversiones de capital, como la construcción de infraestructura de pozos.

Consulta para Operaciones de Completación (CPI)

```
queryCompletacion = f"""
SELECT VENDOR_ID_TEXT, ITEM_NAME, START_WO, END_WO, ESTIMATED_DATE, WELL_STATUS_TEXT, WO_NUMBER
FROM VT_WELLJOBLOG_en_US
WHERE YEAR(END_WO) = {año_interes} AND WELL_STATUS_TEXT = 'CPI' AND VERIFIED = 'True'
"""
```

Descripción: Esta consulta se enfoca en recuperar datos de operaciones de completación de pozos (CPI), seleccionando registros que tienen el estado de pozo marcado como CPI y que han sido verificados. La especificidad del estado WELL_STATUS_TEXT = 'CPI' permite identificar las operaciones que se relacionan directamente con la finalización y preparación de pozos para la producción.

- **Procesamiento de Datos con pandas:**

pandas es una librería de Python que ofrece funciones de análisis de datos, permitiendo manipular tablas de datos y series temporales. En este programa se emplea para filtrar, ordenar y agrupar los datos extraídos, preparándolos para el análisis y la generación de informes.

Una vez extraídos mediante las consultas sql previamente descritas, los datos se cargan en DataFrames de *pandas* para su manipulación y análisis. Este paso implica filtrar los datos según el trimestre y el tipo de operación, limpiar los datos para eliminar entradas incompletas o erróneas, y realizar cálculos como duraciones de operaciones y estadísticas de rendimiento. La flexibilidad de *pandas* permite un ajuste fino del proceso de análisis, asegurando que solo los datos relevantes sean presentados en los informes finales. El código adjunto en la imagen demuestra este comportamiento:

```
dfO = obtener_dataframe(queryO, conn)
dfC = obtener_dataframe(queryC, conn)
dfCo = obtener_dataframe(queryCompletacion, conn)
conn.close()
```

```
# Filtramos los DataFrames por el trimestre actual
dfO_trimestre = filtrar_por_trimestreCPI_Opex(dfO, meses_trimestre)
dfC_trimestre = filtrar_por_trimestreCapex(dfC, meses_trimestre)
dfCo_trimestre = filtrar_por_trimestreCPI_Opex(dfCo, meses_trimestre)

# Dataframe combinado que representa todas las actividades de los pozos en el trimestre
df_combinado = pd.concat([dfO_trimestre, dfC_trimestre, dfCo_trimestre])
```

- **Creación de Estructuras de Directorios para Informes y Gestión de Archivos con *shutil* y *os*:**

Las librerías *shutil* y *os* se utilizan para operaciones relacionadas con el sistema de archivos, como la creación de directorios, la copia de archivos y la manipulación de rutas de archivos. Estas librerías son esenciales para organizar los documentos según la estructura deseada y para preparar los archivos necesarios para los informes.

La generación de informes comienza con la creación de una estructura de directorios bien organizada. La función `crear_directorios` en el código facilita la creación de un entorno de archivo coherente donde los informes generados y los documentos relevantes se almacenan y categorizan sistemáticamente.

Implementación en el código:

```
def crear_directorios(df, ruta_trimestre, tipo_actividad):
    os.makedirs(ruta_trimestre, exist_ok=True)
    # Lógica adicional para crear subdirectorios y organizar archivos
```

Este fragmento de código muestra cómo el sistema automatiza la creación de directorios y subdirectorios basados en las actividades operativas y los datos específicos del pozo. Por cada actividad operativa y pozo enumerado en el DataFrame `df`, se crea una estructura de directorios correspondiente en la ubicación definida por `ruta_trimestre`. Esto proporciona un acceso organizado y directo a los informes y otros documentos de apoyo para las operaciones de cada trimestre.

Paralelamente al análisis de datos obtenidos desde las consultas sql, el sistema utiliza *shutil* y *os* para organizar los documentos de soporte de las operaciones de los pozos. Esto incluye buscar en directorios locales y de OneDrive los archivos relevantes, copiarlos a una estructura de directorios específica para cada informe, y renombrarlos según convenciones preestablecidas para facilitar su identificación y acceso.

El proceso comienza con la identificación de los documentos necesarios. En el contexto de las operaciones de pozos petroleros, estos documentos pueden incluir archivos PDF, presentaciones PPTX, y hojas de

cálculo, entre otros. Los nombres de archivos y rutas son críticos para esta operación, y el sistema está configurado para reconocer patrones de nombres de archivo específicos. A continuación, implementación en el código:

```
import re
import os

# Ejemplo para identificar y copiar archivos de Post Mortem Opex
patron_opex = re.compile(rf'.*{pozo}.{Wo_number}.*OK', re.IGNORECASE)

# Búsqueda de archivos en los directorios relevantes para el año de interés y el siguiente
for año in años:
    ruta_origen_onedrive = f'C:\Users\{usuario}\OneDrive - SLB\General - Post mortem\{año}'
    for carpeta in os.listdir(ruta_origen_onedrive):
        # Filtrar por el patrón de búsqueda y copiar
        archivos = [archivo for archivo in os.listdir(os.path.join(ruta_origen_onedrive, carpeta)) if archivo.endswith('.pptx')]
        for archivo in archivos:
            if patron_opex.search(archivo):
                # Logica de copia aquí
```

Por ejemplo, para el caso de archivos Post Mortem, el sistema busca en los directorios especificados para el año en curso y el siguiente para localizar documentos de Post Mortem asociados con operaciones Opex, utilizando un nombre (logrado mediante el uso de expresiones regulares) que incluye el nombre del pozo y el número WO en el nombre del archivo junto con la verificación extra “OK”. Esto asegura que solo se identifiquen y procesen los documentos pertinentes a la operación específica ya revisados por superiores y listos para ser agregados al reporte trimestral. Dentro del sistema existen varias funciones con este fin, cada una dirigida a un archivo en específico del reporte trimestral. Sin embargo, todas siguen el mismo proceso de búsqueda, validación copiado y pegado. La lista completa de estas funciones se adjunta en la sección de Anexos, sección 3.1 y 3.2.

Una vez identificados, los documentos se copian a la ubicación designada. La organización de estos documentos es crucial, ya que afecta directamente la eficiencia con la que los usuarios pueden acceder y utilizar la información. Esto se repite para cada los distintos documentos de interés vinculados al reporte trimestral, lo que garantiza que todos los archivos relevantes se recolecten en una ubicación central para su análisis y referencia futura. Usando la librería *shutil* el sistema copia los archivos filtrados a la nueva ubicación, preservando su integridad y estructura original. A continuación, se muestra un ejemplo de este proceso:

```
import shutil

destino = '/ruta/a/informe/documentos/'
for archivo in os.listdir(ruta_origen):
    if fnmatch.fnmatch(archivo, patron_busqueda):
        shutil.copy(os.path.join(ruta_origen, archivo), destino)
        print(f'Documento copiado: {archivo} -> {destino}')
```

```

# Crea un nuevo documento con un archivo excel (Sumarios Finales) para CPI
def copiar_sumario_cpi(pozo, ruta_destino):
    ruta_base_origen = obtener_ruta_origen(pozo)
    ruta_cpi = obtener_directorio_paraCPI(ruta_base_origen)

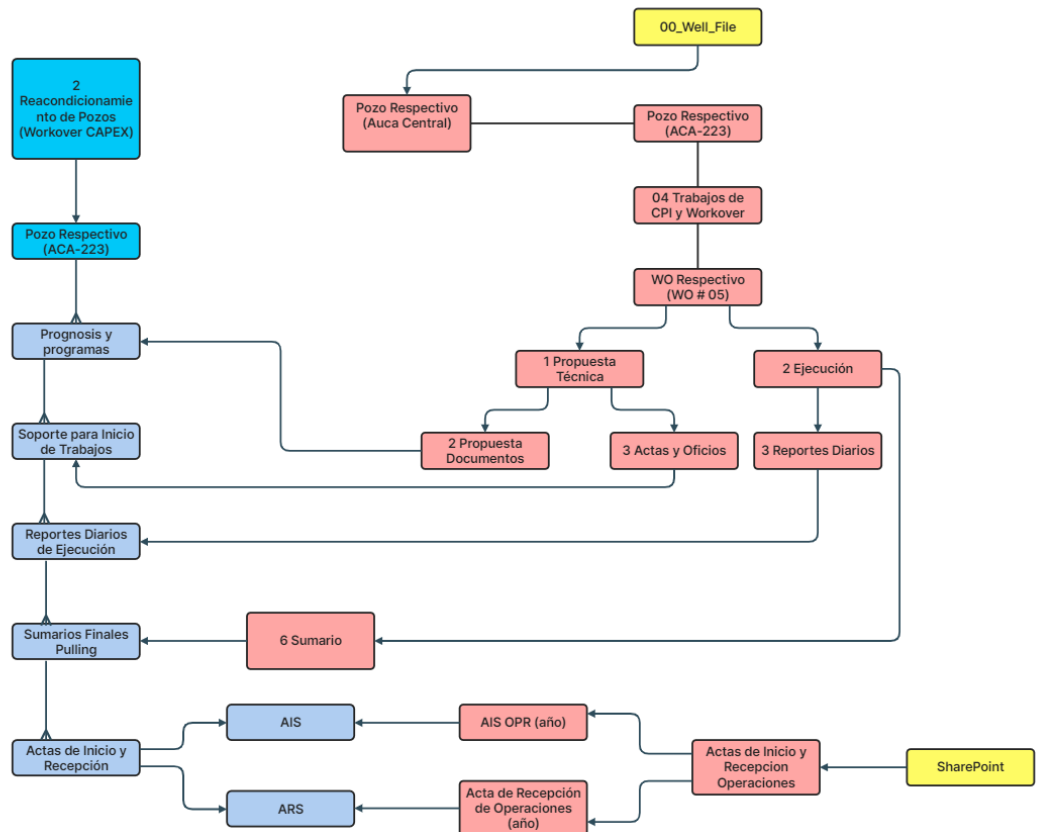
    estado_documento = 'X' # Estado por defecto
    if pozo not in estado_documentos:
        estado_documentos[pozo] = {'CPI': {}}

    if ruta_cpi:
        # Encuentra la primera subcarpeta que comienza con "2."
        ruta_primera_subcarpeta = encontrar_carpetas_por_prefijo(ruta_cpi, '2.')
        if ruta_primera_subcarpeta:
            # Encuentra la segunda subcarpeta que comienza con "6." dentro de la primera subcarpeta
            ruta_segunda_subcarpeta = encontrar_carpetas_por_prefijo(ruta_primera_subcarpeta, '6.')
            if ruta_segunda_subcarpeta:
                for archivo in os.listdir(ruta_segunda_subcarpeta):
                    ruta_origen = os.path.join(ruta_segunda_subcarpeta, archivo)
                    if os.path.isfile(ruta_origen) and (archivo.endswith('.xlsx') or archivo.endswith('.xls') or archivo.endswith('.xlsm') or archivo.endswith('.xlsb')) and "sumario" in archivo.lower():
                        shutil.copy(os.path.join(ruta_segunda_subcarpeta, archivo), ruta_destino)
                        estado_documento = 'OK'
                        break

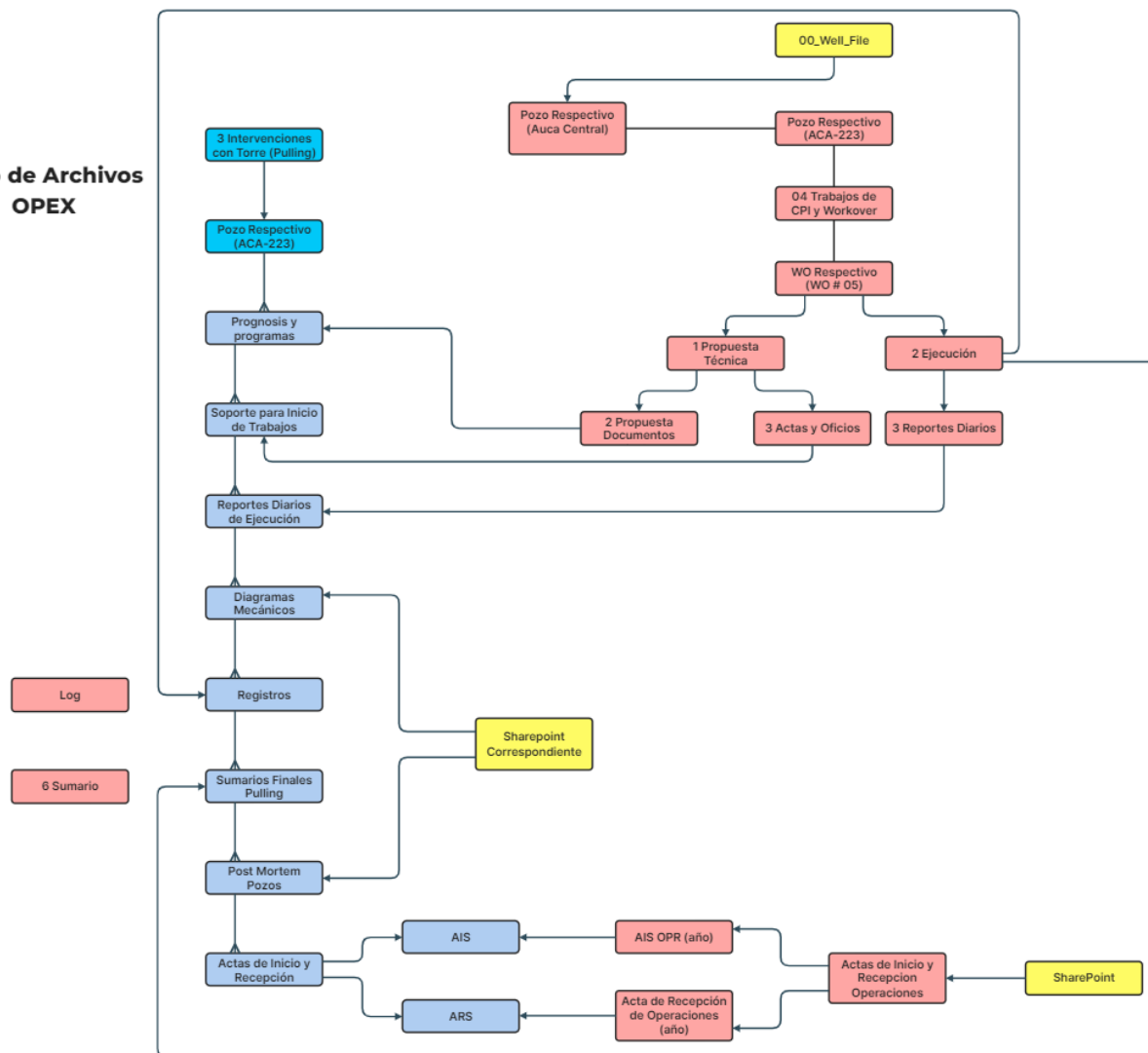
```

El flujo que este reporte sigue para la búsqueda de archivos se ve en detalle, por documento y por directorio, en Anexos, sección 3.2. A continuación, se muestra una gráfica del mismo:

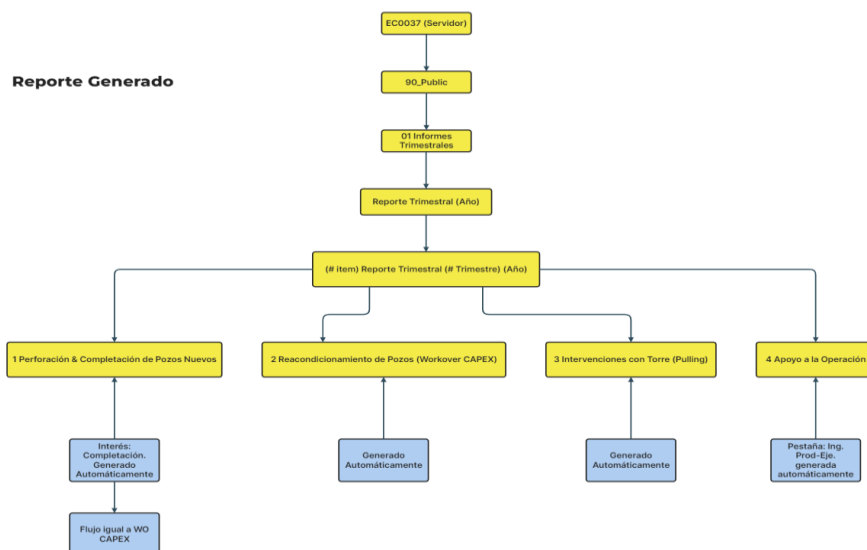
Flujo de Archivos CAPEX



Flujo de Archivos OPEX



La estructura creada por el alcance de esta automatización corresponde a la mostrada a continuación:



- **Creación de Informes con openpyxl:**

openpyxl permite leer y escribir archivos Excel, facilitando la generación de informes personalizados que resumen los resultados del análisis. Se utiliza para crear hojas de cálculo que contienen datos operativos y estados de documentación de los pozos. Este proceso está comprendido dentro de la etapa final del sistema de gestión documental, donde se sintetizan los datos recopilados y procesados en informes formateados que son tanto comprensibles como presentables a las partes interesadas.

El sistema define la función “generar_informe_excel” para crear un informe Excel con los datos de estado de los documentos relacionados con cada pozo y actividad operativa. A continuación, se muestra cómo se implementa esta función:

```
def generar_informe_excel(estados_documentos, año_interes, trimestre_actual, directorio_base):  
    # se crea un dataframe a partir del diccionario  
    data = []  
  
    # Se recorre el diccionario para obtener los datos  
    for pozo, pozo_info in estados_documentos.items():  
        for tipo, tipo_info in pozo_info.items():  
            # Se separa el caso de CPI, ya que no tiene WO #  
            if tipo == 'CPI':  
                cpi_info = {k: v for k, v in tipo_info.items() if k is not None}  
                data.append([pozo, tipo, None] + list(cpi_info.values()))  
            else:  
                for wo_number, values in tipo_info.items():  
                    row = [pozo, tipo, wo_number] + list(values.values())  
                    data.append(row)  
  
    # se crea el dataframe  
    columns = ["Pozo", "Tipo", "WO #", "Propuesta Técnica", "Notificación", "Reportes Diarios", "Sumario", "AIS", "ARS", "Post Mortem", "Diagramas Mecánicos", "Registros"]  
    df = pd.DataFrame(data, columns=columns)  
    df.insert(len(df.columns), "Apoyo a la Operación", "OK, Para Pestaña de Control Documental")
```

Aquí, **estados_documentos** es un diccionario que contiene información sobre varios documentos asociados con cada pozo y su estado actual. La función organiza esta información en filas y columnas dentro del informe.

Además, el sistema tiene una función llamada **calcular_porcentaje_completitud** que toma como entrada el registro de documentos de un pozo y determina qué porcentaje de la documentación necesaria está completa y disponible. Implementación en el código:

```
def calcular_porcentaje_completitud(row):  
    campos_relevantes = ['Propuesta Técnica', 'Notificación', 'Reportes Diarios', 'Sumario', 'AIS', 'ARS', 'Post Mortem', 'Diagramas Mecánicos', 'Registros']  
    total_campos = 0  
    campos_completados = 0  
  
    for campo in campos_relevantes:  
        # Nos aseguramos de que sólo se cuenten los campos con datos  
        estado = row.get(campo, "")  
        if estado != "" and estado != "N/A" and estado is not None:  
            total_campos += 1  
            if "OK" in estado:  
                campos_completados += 1  
  
    # Se calcula el porcentaje si hay campos a considerar  
    if total_campos > 0:  
        porcentaje = (campos_completados / total_campos) * 100  
        return f"{porcentaje:.2f}%" # Se redondea a dos decimales  
    else:  
        return "N/A" # Si no hay campos a considerar, se retorna N/A
```

Esta función itera a través de una lista predefinida de campos de documentos relevantes, contando cuántos están completos. El resultado es un porcentaje que refleja la completitud de la documentación.

Adicionalmente, la función `calcular_porcentaje_completitud` evalúa cada tipo de documento por su estado (por ejemplo, si se ha marcado como 'OK', que indica que el documento está copiado y verificado). Los campos que no aplican o no están disponibles se excluyen del cálculo para asegurar una métrica precisa. El cálculo del porcentaje de completitud proporciona una visión valiosa para los informes, ayudando a identificar áreas donde la documentación puede estar faltante o incompleta y necesite atención adicional. La función se aplica dentro del contexto de la generación de informes, integrándose en la lógica que prepara el resumen de datos operacionales:

```
# Calcular el porcentaje de completitud y añadirlo como una columna al dataframe
df['Porcentaje de Avance'] = df.apply(calcular_porcentaje_completitud, axis=1)

# Guardar en un archivo excel
with pd.ExcelWriter(ruta_guardado, engine='openpyxl') as writer:
    df.to_excel(writer, sheet_name='Ing. Prod-Eje.', index=False, startrow=1)

    # Obtenemos la hoja de trabajo de openpyxl para poder editarla
    workbook = writer.book
    worksheet = writer.sheets['Ing. Prod-Eje.']
```

Cada entrada en el informe detalla el estado de los documentos relacionados con las operaciones del pozo, proporcionando una visión integral del progreso de las actividades. El informe también estiliza los datos para mejorar la visualización y la legibilidad. *openpyxl* proporciona herramientas para ajustar el ancho de las columnas, aplicar filtros y estilos, y agregar gráficos si es necesario. La aplicación de negritas a los títulos y el ajuste del ancho de las columnas son ejemplos de cómo el sistema automatiza la presentación de los informes para su revisión final.

```
def generar_informe_excel(estado_documentos, año_interes, trimestre_actual, directorio_base):

    # Aplicar formato a las celdas con "X"
    red_fill = PatternFill(start_color='FFFF0000', end_color='FFFF0000', fill_type='solid')
    for row in worksheet.iter_rows(min_row=3, max_col=len(df.columns), max_row=worksheet.max_row):
        for cell in row:
            if cell.value == "X":
                cell.fill = red_fill
```

```
def generar_informe_excel(estados_documentos, año_interes, trimestre_actual, directorio_base):

    # Aplicar formato a las celdas
    for col in worksheet.columns:
        max_length = max((len(str(cell.value)) for cell in col)) + 2
        worksheet.column_dimensions[col[0].column_letter].width = max_length

    # Aplicar negrita a los títulos de las columnas y centrar todo el texto
    for cell in worksheet["2:2"]:
        cell.font = Font(bold=True)
        cell.alignment = Alignment(horizontal="center")



    # Centrar todas las celdas
    for row in worksheet.iter_rows(min_row=1, max_col=len(df.columns), max_row=len(df)+2):
        for cell in row:
            cell.alignment = Alignment(horizontal="center")
```

- **Flujo en Power Automate – Notificación para Seguimiento**



Power Automate permite realizar procesos automatizados respetando las políticas de ciberseguridad conforme a la visión de la empresa. Para el caso de este proyecto, esta herramienta fue utilizada con dos propósitos:

- Realizar una conexión a un Sharepoint de interés: **General - DIGITALIZACION ING. PROD. & EJEC_**
- Notificar a destinatarios definidos, por medio de correo electrónico, el informe de seguimiento generado para el trimestre de interés.

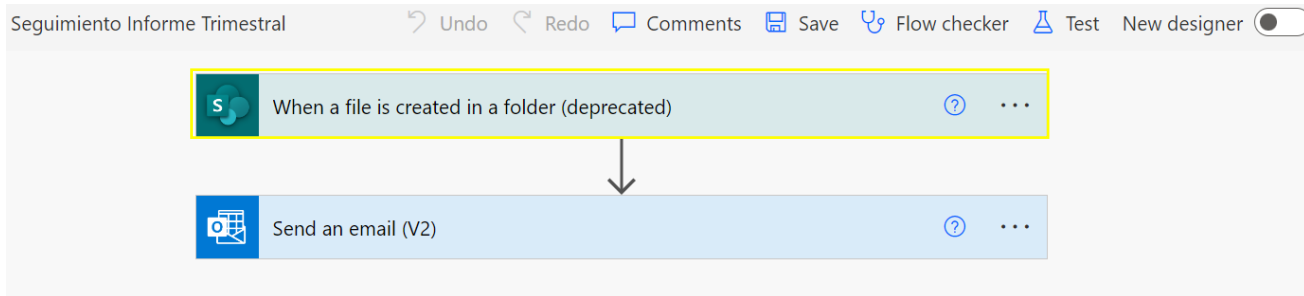
Para esto, se realizó el flujo mostrado a continuación:

Flows Install ▾			
Cloud flows	Desktop flows	Shared with me	
	Name	Modified	Type
	Seguimiento Informe Trimestral	1 d ago	Automated

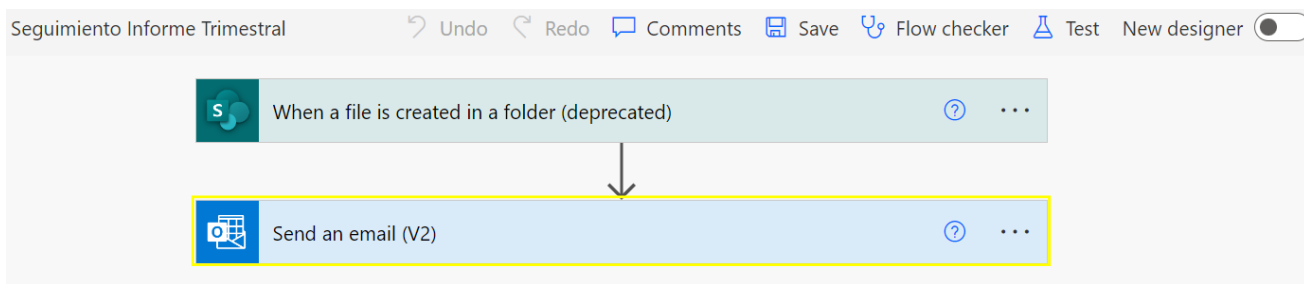
Este flujo está compuesto por un trigger y una acción. El trigger es una acción que se realiza siempre y cuando una condición sea cumplida. Para el propósito de este proyecto, la condición a cumplirse es la generación del reporte del informe trimestral dentro del Sharepoint: **General - DIGITALIZACION ING. PROD. & EJEC_**. Un ejemplo de este producto, es decir el reporte generado, se puede ver a continuación:

OneDrive - SLB > General - DIGITALIZACION ING. PROD. & EJEC_ > 01 Seguimiento Informe Trimestral						
Quick access	Name	Status	Date modified	Type	Size	
Desktop	 Informe Trimestral Q1 2024 (040659)		3/19/2024 5:42 AM	Microsoft Excel Work...	12 KB	

Una vez este reporte de informe sea generado, es decir, una vez que el proyecto automatizado sea ejecutado con éxito, el flujo es activado. La acción **When a file is created in a folder** representa el trigger de este flujo, puesto que reconoce la creación del reporte luego de haber sido ejecutado el proyecto automatizado.



La acción resultante implica el envío de un correo electrónico, con un mensaje (body) establecido, un asunto (subject) programado para contener el nombre del archivo de reporte generado, y una fecha límite de carga de archivos incluida en el mensaje (body) del correo. Para la fecha en cuestión, se estableció que el límite sería de 5 días después de la ejecución del proyecto automatizado y generación del reporte de seguimiento. Esto se puede ver a continuación:



The screenshot shows the configuration window for the 'Send an email (V2)' action. It includes the following fields:


- * To:** A list of email addresses: NHerrera6@slb.com, ECiencia@slb.com, and APerez143@slb.com.
- * Subject:** The text 'Seguimiento de' followed by a function 'decodeUriCom...'.
- * Body:** A rich text editor containing the following content:
 - Font: 12, Bold, Italic, Underline, and other formatting options.
 - Estimado equipo,
 - Remitimos estado de Informe trimestral para su revisión. Su acostumbrado apoyo para completar la información pendiente, para la entrega del informe del Q correspondiente.
 - Fecha límite para carga de información: addDays(...) (yyyy-mm-dd)
 - Por su ayuda, anticipamos nuestro agradecimiento.
 - Saludos
 - Control Documental


Como adición, dentro del campo de opciones avanzadas de la acción, se establece un emisor del correo y además el reporte de seguimiento como adjunto. Esto se puede ver a continuación:

From (Send as) N NHerrera6@slb.com X

CC

BCC

Attachments Name - 1  x-ms-file-name... X

Attachments Content - 1  File Content X

+ Add new item

Sensitivity ▼

Reply To

Importance ▼

[Hide advanced options](#) ^

3. ANEXOS

3.1 Funciones de copiado de archivos

```
# Funciones para copiar archivos de origen a destino

# Caso en que se desea copiar un archivo pdf (Soporte para Inicio de Trabajos) para Capex y Opex
> def copiar_pdf(pozo, wo_number, ruta_destino, tipo_pozo):...

# Caso en que se desea copiar un archivo pdf (Soporte para Inicio de Trabajos) para CPI
> def copiar_pdf_cpi(pozo, ruta_destino):...

# Caso en que se desea copiar un archivo word (Prognosis y Programas) para Capex y Opex
> def copiar_prognosis_capex(pozo, wo_number, ruta_destino, tipo_pozo):...

# Caso en que se desea copiar un archivo word (Prognosis y Programas) para CPI
> def copiar_prognosis_cpi(pozo, ruta_destino):...

# Caso en que se desea copiar todos los archivos pdf (Reportes Diarios de Ejecución) para Capex y Opex
> def copiar_reportes_diarios(pozo, wo_number, ruta_destino, tipo_pozo):...

# Caso en que se desea copiar todos los archivos pdf (Reportes Diarios de Ejecución) para CPI
> def copiar_reportes_diarios_cpi(pozo, ruta_destino):...

# Caso en que se desea copiar un archivo excel (Sumarios Finales) para Capex y Opex
> def copiar_sumario(pozo, wo_number, ruta_destino, tipo_pozo):...

# Caso en que se desea copiar un archivo excel (Sumarios Finales) para CPI
> def copiar_sumario_cpi(pozo, ruta_destino):...

# Caso en que se desea copiar archivos pdf de AIS desde OneDrive
> def copiar_AIS_Opex(pozo, año_inicio, wo_number, ruta_destino, tipo_pozo, abandono):...
```

```

> def copiar_AIS(pozo, año_inicio, wo_number, ruta_destino, tipo_pozo):...

# Caso en que se desea copiar archivos de Diagramas Mecanicos
> def copiar_diagramas_mecanicos(pozo, wo_number, ruta_destino, tipo_pozo, abandono):...

# Caso en que se desea copiar archivos de registro
> def copiar_registros(pozo, wo_number, ruta_destino, tipo_pozo, abandono):...

# Caso en que se desea copiar un archivo pptx de Post Mortem desde OneDrive, caso Opex (y caso especial abandono, tipo de Opex)
> def copiar_PM_Opex(pozo, año_fin, wo_number, ruta_destino, tipo_pozo, abandono):...

# Caso en que se desea copiar archivos pdf de ARS desde OneDrive
> def copiar_ARS_Opex(pozo, año_fin, wo_number, ruta_destino, tipo_pozo, abandono):...

> def copiar_ARS(pozo, año_fin, wo_number, ruta_destino, tipo_pozo):...

```

3.2 Detalle de archivos y directorios utilizado en el flujo

A continuación, se detalla el flujo de archivos manejados para el Reporte Trimestral.

La mayoría de los archivos se obtienen del directorio que será denominado **Directorio base**:

\\dir.slb.com\NSA\SAM_Collaborate\EC0037\00_Well_File\NOMBRE GENERAL DEL POZO\NOMBRE RESPECTIVO DEL POZO

Para la Notificación, aplicada a CPI, OPEX y CAPEX:

Para CPI, el directorio de búsqueda es:

Directorio base\04 Trabajos de CPI y Workover\CPI\1. Propuesta Tecnica\3. Actas y Oficios\1. Permisos PAM – SHE

Para OPEX y CAPEX, el directorio de búsqueda es:

Directorio base\04 Trabajos de CPI y Workover\# WO\1. Propuesta Técnica\3. Actas y Oficios\1. Permisos PAM-SHE

Archivo a Copiar: **Todo archivo PDF o Todo archivo con la palabra “Notificacion” en su nombre (sin importar mayúsculas o minúsculas).**

Para la Propuesta Técnica, aplicada a CPI, OPEX y CAPEX:

Para CPI, el directorio de búsqueda es:

Directorio base\04 Trabajos de CPI y Workover\CPI\1. Propuesta Tecnica\2. Propuesta Documentos Finales

Para OPEX y CAPEX, el directorio de búsqueda es:

Directorio base\04 Trabajos de CPI y Workover\# WO\1. Propuesta Tecnica\2. Propuesta Documentos Finales

Archivo a Copiar: **Todo archivo Word o todo archivo con la palabra “Propuesta” en su nombre (sin importar mayúsculas o minúsculas).**

Para los Reportes Diarios, aplicados a CPI, OPEX y CAPEX:

Para CPI, el directorio de búsqueda es:

Directorio base\04 Trabajos de CPI y Workover\CPI\2. Ejecucion\3. Reportes Diarios

Para OPEX y CAPEX, el directorio de búsqueda es:

Directorio base\04 Trabajos de CPI y Workover\# WO\2. Ejecucion\3. Reportes Diarios

Archivos a Copiar: **Todos los archivos PDF encontrados en el directorio (representan todos los reportes diarios).**

Para el Sumario, aplicado a CPI, OPEX y CAPEX:

Para CPI, el directorio de búsqueda es:

Directorio base\04 Trabajos de CPI y Workover\CPI\2. Ejecucion\6. Sumario

Para OPEX y CAPEX, el directorio de búsqueda es:

Directorio base\04 Trabajos de CPI y Workover\# WO\2. Ejecucion\6. Sumario

Archivo a Copiar: **Archivo Excel que contenga la palabra “Sumario” en su nombre (sin importar mayúsculas o minúsculas).**

Para Diagramas Mecánicos, aplicado a OPEX:

El directorio de búsqueda es:

Directorio base\08 Diagramas Mecánicos

El archivo a copiar es: **Archivo (no importa el formato) que contenga el nombre específico del pozo y el número de workover en su nombre (sin importar mayúsculas o minúsculas).**

Para Registros, aplicado a OPEX:

El directorio de búsqueda es:

Directorio base\04 Trabajos de CPI y Workover\# WO\2. Ejecucion\Log

El archivo a copiar es: **Archivo (no importa el formato) que contenga el nombre específico del pozo y el número de workover en su nombre (sin importar mayúsculas o minúsculas).**

El porcentaje restante de archivos necesarios para el reporte trimestral se obtienen de distintos Sharepoints/OneDrive detallados a continuación:

Para Post Mortem, aplicado a OPEX:

Directorio de búsqueda:

OneDrive - SLB\General - Post mortem\AÑO RESPECTIVO

El archivo a copiar es: **Archivo (no importa el formato) que contenga el nombre específico del pozo, el número de workover y la palabra “OK” en su nombre (sin importar mayúsculas o minúsculas).**

Para AIS, aplicado a CPI, CAPEX y OPEX:

Directorio de búsqueda para CPI:

OneDrive - SLB\Actas de Inicio y Recepcion Operaciones\AIS OPR (año respectivo)\2. Completación\CARPETA POZO RESPECTIVO CPI

Directorio de búsqueda para CAPEX:

OneDrive - SLB\Actas de Inicio y Recepcion Operaciones\AIS OPR (año respectivo)\3. Workover\CARPETA POZO RESPECTIVO Y NÚMERO DE WO

Directorio de búsqueda para OPEX:

OneDrive - SLB\Actas de Inicio y Recepcion Operaciones\AIS OPR (año respectivo)\4. Pulling (OPEX)\CARPETA POZO RESPECTIVO Y NÚMERO DE WO

O

OneDrive - SLB\Actas de Inicio y Recepcion Operaciones\AIS OPR (año respectivo)\4. Pulling (OPEX)\CARPETA POZO RESPECTIVO ABANDONO

El archivo a copiar es: **Archivo (no importa el formato) que contenga la palabra “consolidado” y con la mayor cantidad de veces de la palabra “signed” en su nombre (sin importar mayúsculas o minúsculas).**

Para ARS, aplicado a CPI, CAPEX y OPEX:

Directorio de búsqueda para CPI:

OneDrive - SLB\Actas de Inicio y Recepcion Operaciones\ACTA DE RECEPCION DE OPERACIONES (Año respectivo)\CARPETA POZO RESPECTIVO CPI

Directorio de búsqueda para CAPEX:

OneDrive - SLB\Actas de Inicio y Recepcion Operaciones\ACTA DE RECEPCION DE OPERACIONES (Año respectivo)\ CARPETA POZO RESPECTIVO Y NÚMERO DE WO

Directorio de búsqueda para OPEX:

OneDrive - SLB\Actas de Inicio y Recepcion Operaciones\ACTA DE RECEPCION DE OPERACIONES (Año respectivo)\ CARPETA POZO RESPECTIVO Y NÚMERO DE WO

O

OneDrive - SLB\Actas de Inicio y Recepcion Operaciones\ACTA DE RECEPCION DE OPERACIONES (Año respectivo)\ CARPETA POZO RESPECTIVO ABANDONO

El archivo a copiar es: **Archivo (no importa el formato) que contenga la palabra “consolidado” y con la mayor cantidad de veces de la palabra “signed” en su nombre (sin importar mayúsculas o minúsculas).**