Para implementar el monitoreo solicitado, se pueden seguir los siguientes pasos:

1. Crear un script SQL que ejecute la consulta y cuente los registros con el estado 6.

2. Crear un script en Python que ejecute el script SQL, procese los resultados y envíe correos electrónicos según las condiciones especificadas.

3. Configurar una tarea programada para ejecutar el script de Python periódicamente.

### Paso 1: Crear el script SQL

Crear un archivo SQL que contenga la consulta para contar los registros con el estado 6

SELECT COUNT(\*) AS count

FROM CE\_SOLICITUD

WHERE TRUNC(SOLI\_FECSOL) >= TO\_DATE(TO\_CHAR(SYSDATE, 'dd/mm/yyyy'), 'dd/mm/yyyy') - 10

AND SOLI\_ESTADO = 6;

### Paso 2: Crear el script en Python

Crear un archivo Python que ejecute el script SQL, procese los resultados y envíe correos electrónicos según las condiciones especificadas.

### [monitoring\_script.py](file:///c%3A/Users/Analis15/OneDrive%20-%20SISTEMAS%20INTEGRALES%20DE%20INFORMATICA%20S.A.S/monitoring\_script.py)

```python

Made changes.

import smtplib

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

import cx\_Oracle

def send\_email(subject, body, color):

sender\_email = "your\_email@example.com"

receiver\_email = "receiver\_email@example.com"

password = "your\_password"

msg = MIMEMultipart()

msg['From'] = sender\_email

msg['To'] = receiver\_email

msg['Subject'] = subject

body = f'<p style="color:{color};">{body}</p>'

msg.attach(MIMEText(body, 'html'))

with smtplib.SMTP('smtp.example.com', 587) as server:

server.starttls()

server.login(sender\_email, password)

server.sendmail(sender\_email, receiver\_email, msg.as\_string())

def main():

dsn\_tns = cx\_Oracle.makedsn('hostname', 'port', service\_name='service\_name')

conn = cx\_Oracle.connect(user='username', password='password', dsn=dsn\_tns)

cursor = conn.cursor()

cursor.execute(open('monitoring\_query.sql', 'r').read())

result = cursor.fetchone()

count = result[0]

if count > 2:

send\_email('Alerta', f'Se encontraron {count} registros con estado 6.', 'red')

elif count > 0:

send\_email('Warning', f'Se encontraron {count} registros con estado 6.', 'yellow')

cursor.close()

conn.close()

if \_\_name\_\_ == "\_\_main\_\_":

main()

```

### Paso 3: Configurar una tarea programada

Configurar una tarea programada en el sistema operativo para ejecutar el script de Python periódicamente. Esto se puede hacer utilizando el Programador de tareas en Windows.

1. Abrir el Programador de tareas.

2. Crear una nueva tarea básica.

3. Configurar la tarea para que se ejecute diariamente o según la frecuencia deseada.

4. En la acción, seleccionar "Iniciar un programa" y buscar el script de Python creado.

Con estos pasos, se implementará el monitoreo solicitado que ejecuta la consulta en la base de datos y envía correos electrónicos según las condiciones especificadas.

Made changes.