**Documento detalhado das funcionalidades do projeto =D ~**

**📌 Visão Geral**

Esse script em Python automatiza o envio de uma mensagem de relatório diário para contatos via WhatsApp Web. Ele:

1. Exibe uma interface gráfica para o operador;
2. Lê uma planilha de Excel com nomes e números;
3. Deixa você **selecionar os contatos com o mouse**;
4. **Envia a mensagem formatada automaticamente**;
5. Pode repetir o envio com o intervalo desejado.

Agora vamos detalhar cada parte:

**🔹 Importações**

import tkinter as tk

from tkinter import simpledialog, messagebox

import pandas as pd

import pywhatkit

import pyautogui

import schedule

import datetime

import time

Essas bibliotecas fazem o seguinte:

* tkinter → cria as janelas gráficas e caixas de diálogo.
* pandas → lê a planilha Excel com contatos.
* pywhatkit → escreve a mensagem no WhatsApp Web.
* pyautogui → simula pressionar Enter no teclado para enviar.
* schedule → agenda o envio automático em intervalos.
* datetime e time → manipulam o relógio do sistema e pausas.

🔹 Etapa 1 – Interface inicial

def interface\_inicial():

Essa função abre uma janelinha e pergunta:

* Qual o nome do operador (quem está enviando o relatório).
* Quantas pessoas receberão a mensagem.

Ela garante que os dados inseridos sejam válidos, senão exibe uma mensagem de erro e encerra.

🔹 Etapa 2 – Leitura da planilha

def extrair\_contatos\_do\_excel():

Essa função:

* Acessa assets/relatorio\_diario.xlsx;
* Vai direto para a aba "Contatos";
* Considera o cabeçalho real começando na **segunda linha** (header=1);
* Extrai os pares ("Nome", "Telefone").

Se não conseguir ler, mostra o erro e fecha o programa.

🔹 Etapa 3 – Seleção dos contatos

def selecionar\_contatos\_via\_lista(contatos, quantidade):

Aqui acontece a mágica da interface:

* Abre uma janelinha com todos os nomes disponíveis;
* Permite marcar os contatos com o mouse (com múltipla seleção);
* Só permite seguir se a quantidade de selecionados for igual ao número informado.

Isso evita erro de digitação e garante que só contatos válidos sejam escolhidos.

🔹 Etapa 4 – Enviar mensagem via WhatsApp

def enviar\_mensagem\_whatsapp(numero, mensagem):

Essa função:

1. **Calcula o horário de envio** para 2 minutos à frente, porque o pywhatkit exige tempo para carregar o WhatsApp.
2. **Agenda a mensagem** usando sendwhatmsg();
3. Espera **10 segundos adicionais** para garantir que o navegador esteja carregado;
4. **Simula o Enter** para realmente enviar a mensagem com pyautogui.

Esse combo resolve o problema clássico de “a mensagem foi escrita mas não foi enviada”.

🔹 Etapa 5 – Agendamento de envio

def agendar\_envios(contatos, mensagem, intervalo\_minutos=1):

Essa função usa a lib schedule para:

* Repetir o envio a cada X minutos (no seu caso, a cada 1 minuto);
* Para cada contato, agenda a função enviar\_mensagem\_whatsapp.

Depois entra num loop infinito monitorando o relógio e executando os envios quando chega a hora.

🔹 Formatação da mensagem

def montar\_mensagem(nome\_operador):

A mensagem contém todos os campos do seu relatório, com quebras de linha, e ao final:

“ Documento enviado por Robert. ”

Ou por quem estiver operando.

🔹 Execução principal

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

Aqui está o fluxo que costura tudo:

1. Pede o nome + quantidade;
2. Lê os contatos da planilha;
3. Abre a tela de seleção dos destinatários;
4. Monta a mensagem no formato correto;
5. Inicia o agendador de envios.

## ✅ Ajustes rápidos que você pode fazer

* Trocar o tempo de envio: intervalo\_minutos=1 → troque pra 30, 60, etc.
* Ajustar mensagem: Edite dentro da função montar\_mensagem().