# Documentación del Bot de Automatización para Call Center

## Introducción

Este bot ha sido diseñado para automatizar la recolección, análisis y envío de alertas sobre el rendimiento del call center. Su función principal es extraer datos de reportes, analizarlos y generar notificaciones cuando se detectan condiciones críticas. Esto permite a los supervisores tomar decisiones en tiempo real para optimizar la operación y mejorar la eficiencia del servicio.

## Dependencias

Para su correcto funcionamiento, el bot requiere las siguientes librerías:

import pandas as pd

from datetime import datetime

import os

import pyautogui

import time

import mysql.connector

import pyperclip

from selenium import webdriver

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.common.by import By

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

import tkinter as tk

import customtkinter as ctk

import math

import dotenv as env

## Configuraciones Iniciales

Antes de ejecutar el bot, se definen varios parámetros clave:

Condicional\_NDS = 1

tiempoEntrePasos = 1.5

grupoWSP = "Lideres - GTR (customer)"

publicar = 0

dbTableName = "sadasd"

contactoError = "2604053233"

Estos parámetros configuran aspectos como el intervalo de tiempo entre pasos, el grupo de WhatsApp donde se enviarán las alertas y la base de datos donde se almacenan los datos procesados.

Se cargan datos desde un archivo de nómina para asociar información de asesores con los reportes:

df\_nomina = pd.read\_excel("nomina.xlsx")

Además, se crean DataFrames vacíos para almacenar los datos de conexión y tiempos de los asesores:

columnsDfConexion = ["ID", "inicioConexion", "tiempoTranscurrido", "proximoRefrigerio", "Skill"]

df\_conexiones = pd.DataFrame(columns=columnsDfConexion)

columnas = ['Nombre', 'ID', 'Extension', 'Motivo\_AUX', 'Estado', 'Tiempo', 'VDN', "Skill", "Fecha", "Hora"]

df\_tiempos = pd.DataFrame(columns=columnas)

## Funcionalidades Principales

### 1. Temporizador para la Ejecución

El bot despliega un temporizador visual antes de ejecutar ciertos procesos, lo que permite a los supervisores conocer el tiempo restante para la próxima evaluación:

def iniciar\_temporizador(tiempo\_inicial):

...

### 2. Extracción y Procesamiento de Datos

El bot automatiza la recolección de datos mediante scraping y control de la interfaz:

* click\_on\_image\_when\_visible(image\_path): Detecta y hace clic en una imagen en pantalla.
* ejecutarReporte(reporte): Abre y ejecuta un reporte específico.
* consultarConcatenarTiempoConexion(): Extrae y procesa datos de conexión de los agentes.
* navegarCalcularConcatenarReporteConexiones(): Consulta y almacena los reportes de conexiones.
* navegarCalcularConcatenarPlanaTiempoReal(): Obtiene y procesa información sobre tiempos reales de los asesores.

### 3. Evaluación y Generación de Alertas

El bot evalúa condiciones críticas basadas en:

* Nivel de servicio (NDS)
* Llamadas en espera
* Tiempo promedio de manejo (TMO)
* Estado de los asesores (disponibles, auxiliares, ocupación, etc.)

Si se detectan problemas, se generan alertas automáticas para que los supervisores tomen decisiones en tiempo real.

El bot también alerta cuando hay asesores que exceden el tiempo permitido en auxiliar, lo que puede afectar la disponibilidad general del equipo. Si un asesor permanece demasiado tiempo en auxiliar, se envía una notificación para su revisión.

### 4. Envio de Alertas por WhatsApp

El bot usa Selenium para enviar mensajes automáticamente a un grupo de WhatsApp:

def envio\_wsp(argumento, grupoWhatsapp):

...

Esto permite mantener informados a los líderes en tiempo real sin necesidad de monitoreo manual constante.

### 5. Bucle de Ejecución Continua

El bot opera en un bucle infinito que:

1. Extrae datos de los sistemas.
2. Evalúa las condiciones del call center.
3. Genera y envía alertas cuando es necesario.
4. Espera un tiempo antes de volver a ejecutar el proceso.

while True:

try:

...

except Exception as error:

print("Error de sistema: ", error)

envio\_wsp(f"Error detectado: {error}", contactoError)

## Conclusión

Este bot optimiza la gestión del call center al automatizar la extracción de datos, el análisis y el envío de alertas. Con ello, permite una toma de decisiones más rápida y eficiente, asegurando que el equipo de supervisión pueda actuar de inmediato ante cualquier problema detectado. Además, mejora el control del uso de auxiliares para garantizar que los asesores estén disponibles cuando se los necesite.