



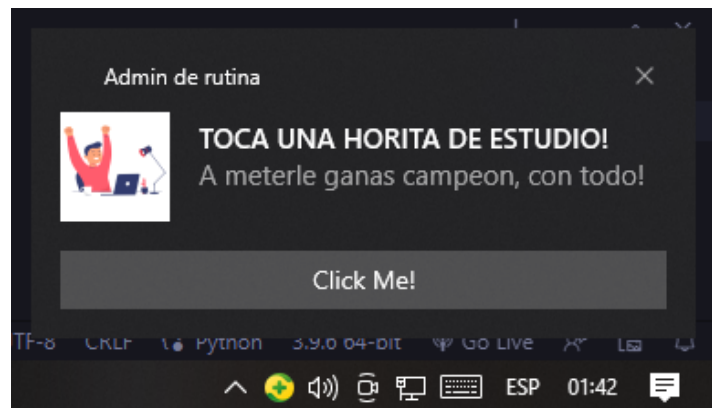
ADVISER

🕒 Created	@May 6, 2023 12:11 AM
📁 Tipo	PROPIO
☑ Reviewed	✓
📎 Materials	
🔗 URL	https://github.com/iatorres/Adviser

ADVISER

Aplicación para el manejo de rutinas, se encarga de notificarte en la computadora cuando según tu rutina es momento de realizar tal actividad, muy funcional para estudiantes o trabajadores FreeLancer, **sabemos que un orden en las actividades es igual a un mejor desarrollo de las mismas.**

La aplicación consiste en **poppear** una notificación en el escritorio, tal como lo haría cualquier otra, mediante el pop y la música poder dar esa sensación de *alarma*.



CLICK ME! se puede seleccionar un link, donde te enviara la notificación, en caso de necesitarlo (web campus, Google, etc.)

Actualmente solo posee mi rutina instalada, falta implementar una interfaz gráfica y una base de datos para poder almacenar la información de las actividades por horario. Todo esto para poder ser adaptada a un usuario ajeno a la programación.

Alguien con conocimientos de básicos en Python podría instalar su propia rutina y sería totalmente funcional.

▼ PROBLEMAS SOLUCIONADOS.

Mucho consumo de energía, necesito darle inteligencia al programa para saber cuándo pausar de correr el ciclo y saber a qué hora tirar las notificaciones.

```
'''Programa corriendo aprox desde 8 am hasta 15 pm y reanuda 23 pm hasta las 00am.

1) primero necesito obtener las variables de Hora, Minuto y Dia. --
2) bucle que vaya viendo las horas, en el momento que lo haga arrancar debería ser mayor a las 8.
3) ¿Como romper el bucle, como inicializarlo?

4) Una vez en el bucle, ¿Como hago q mi pc no se rompa?
   Hago funcion que reconozco dia hora minuto

17:20

while se pause unas 35

18:00

now=datetime.now()
dia=datetime.today().weekday()
hora=now.hour
minuto=now.minute'''
```

El programa es capaz de reconocer cuando es mi hora de trabajo y cuando no estoy. Es capaz de saber en qué hora y minutos arranca, para calcular cuánto falta para el próximo pop de la notificación así mediante el ***time.sleep*** pausa el bucle. También reconoce si es sábado o domingo y no está programado por ser días libres.

```
if (hora>=8 and hora<15) or (hora>=22):
    popOut(diaLista, hora)

while (hora>=8 and hora<16) or (hora>=22):

    time.sleep( (3600 - (minuto*60) )) #el programa descansa hasta la siguiente hora
    #os.pause
    horaN=actual.hour
    #17:00

    if (horaN != hora):
        hora=horaN
```

```
        minuto=actual.minute
        popOut(diaLista,hora)

if hora < 8 or hora < 23:
    toast=Notification(

        app_id="Admin de rutina",
        title="FUERA DE HORARIO",
        msg="No hay actividades seleccionadas para este horario",
        duration= 'long',
        icon=r"icon.png")

    toast.set_audio(audio.LoopingCall, loop=True)
    toast.show()
```

▼ PROBLEMAS A SOLUCIONAR.