

## Manual: Alertas de Citas por Telegram con Python

---

### 1. Requisitos previos

**Antes de empezar, necesitas:**

- - Cuenta de Telegram (app instalada en celular o escritorio).
- - Crear un bot de Telegram usando BotFather (obtener el TOKEN).
- - Tener instalado Python 3.x en tu computadora.
- - Instalar las librerías de Python: pandas y requests.
- - Un archivo de Excel con la información de las citas (por ejemplo: citas\_clientes.xlsx).

### 2. Crear un bot en Telegram con BotFather

- 1) Abre la aplicación de Telegram (en el celular o en la computadora).
- 2) En el buscador de Telegram, escribe: BotFather y selecciona el bot verificado.
- 3) En el chat con BotFather, escribe /newbot y presiona Enter.
- 4) Sigue las instrucciones:

- Primero te pedirá un nombre visible para el bot (por ejemplo: Recordatorio Citas).
- Despues te pedirá un nombre de usuario que termine en 'bot' (por ejemplo: recordatorio\_citas\_bot).

- 5) Al final, BotFather te mostrará un mensaje con el TOKEN de tu bot, con un formato parecido a:

123456789:ABCdefGHI\_jklMNOpqrsTUVvwxyz

- 6) Copia ese TOKEN y guárdalo en un lugar seguro.

**IMPORTANTE:** No lo pubiques en GitHub ni lo compartas con nadie.

*Después de crear el bot, búscalo en Telegram por el nombre de usuario que elegiste y presiona "Iniciar" o envíale cualquier mensaje (por ejemplo: Hola). Esto es necesario para poder obtener tu chat\_id.*

### 3. Obtener tu chat\_id de Telegram

- 1) Abre el chat con tu bot en Telegram y envíale un mensaje (por ejemplo: 'Hola').
- 2) En tu navegador de internet, escribe la siguiente dirección (URL):  
[https://api.telegram.org/botTU\\_TOKEN\\_AQUI/getUpdates](https://api.telegram.org/botTU_TOKEN_AQUI/getUpdates)  
Reemplaza TU\_TOKEN\_AQUI por el TOKEN real de tu bot (sin espacios).
- 3) En la página se mostrará un texto en formato JSON (texto con llaves, corchetes, etc.).
- 4) Dentro de ese texto, busca una parte que diga algo como:  
"chat": {"id": 123456789, "first\_name": "TuNombre", ...}
- 5) El número que aparece en id (en este ejemplo 123456789) es tu chat\_id.
- 6) Anota ese número, porque lo usarás en el script de Python y/o en tu archivo de Excel.

### 4. Preparar el archivo de citas (Excel o DataFrame)

Para que el script funcione, es recomendable utilizar un archivo de Excel, por ejemplo: citas\_clientes.xlsx, con las siguientes columnas:

- - nombre\_cliente: Texto. Ejemplo: 'Juan Pérez'.
- - fecha\_cita: Fecha y hora. Ejemplo: 2025-11-30 15:30 (formato yyyy-mm-dd hh:mm).
- - chat\_id: Número entero. Ejemplo: 123456789 (el chat\_id de Telegram del cliente).
- - nota\_opcional: Texto libre, por ejemplo: 'Primera consulta', 'Control mensual'. (columna opcional).

Asegúrate de que las fechas y horas estén bien capturadas y que la zona horaria coincida con la de tu computadora o servidor donde se ejecutará el script.

### 5. Código de ejemplo en Python

**Primero instala las librerías necesarias (en la terminal):**

```
pip install pandas requests
```

### A continuación, un ejemplo de script llamado `script_alertas_telegram.py`:

```
import pandas as pd
import requests
from datetime import datetime, timedelta

# === CONFIGURACIÓN ===
TOKEN = 'TU_TOKEN_AQUI' # Reemplaza con el token de tu bot
BASE_URL = f'https://api.telegram.org/bot{TOKEN}/sendMessage'

def enviar_mensaje(chat_id: int, texto: str) -> None:
    params = {
        'chat_id': chat_id,
        'text': texto
    }
    requests.get(BASE_URL, params=params)

def revisar_y_enviar_alertas(ruta_excel: str) -> None:
    # Leer archivo de citas
    df = pd.read_excel(ruta_excel)

    # Convertir columna fecha_cita a tipo datetime
    df['fecha_cita'] = pd.to_datetime(df['fecha_cita'])

    # Momento actual
    ahora = datetime.now()

    # Ventana de 2 horas hacia adelante
    ventana_inicio = ahora
    ventana_fin = ahora + timedelta(hours=2)

    for _, fila in df.iterrows():
        fecha = fila['fecha_cita']
        chat_id = fila['chat_id']
        nombre = fila['nombre_cliente']

        # Verificar si la cita está dentro de las próximas 2 horas
        if ventana_inicio <= fecha <= ventana_fin:
            mensaje = (
                f'Hola {nombre}, recuerda que tienes una cita hoy a las '
                f'{fecha.strftime("%H:%M")}'
            )
            enviar_mensaje(chat_id, mensaje)

if __name__ == '__main__':
    revisar_y_enviar_alertas('citas_clientes.xlsx')
```

## 6. Programar la ejecución automática del script

### Opciones principales:

- - Ejecutar el script manualmente.
- - Programar la ejecución en Windows (Programador de tareas).
- - Programar la ejecución en Linux/Mac usando cron.

### 6.1 Ejecución manual

- 1) Abrir la terminal (CMD o PowerShell en Windows).
- 2) Navegar a la carpeta donde guardaste script\_alertas\_telegram.py.
- 3) Ejecutar:  
`python script_alertas_telegram.py`

### 6.2 Programar en Windows (Programador de Tareas)

- 1) Abre el 'Programador de tareas' (Task Scheduler).
- 2) Crea una tarea básica nueva.
- 3) Elige la frecuencia (por ejemplo, cada 5 o 10 minutos).
- 4) En 'Acción', selecciona 'Iniciar un programa'.
- 5) En 'Programa o script', selecciona la ruta de python.exe.
- 6) En 'Agregar argumentos', escribe la ruta del script, por ejemplo:  
`C:\ruta\al\script_alertas_telegram.py`
- 7) Guarda la tarea. A partir de ahí, Windows ejecutará el script automáticamente con la frecuencia elegida.

### 6.3 Programar en Linux/Mac (cron)

- 1) Abre la terminal.
- 2) Escribe: `crontab -e`
- 3) Agrega una línea como la siguiente (ejemplo cada 5 minutos):  
`*/5 * * * * /usr/bin/python3 /ruta/al/script_alertas_telegram.py`
- 4) Guarda el archivo. Cron ejecutará el script según la frecuencia configurada.

## 7. Buenas prácticas y recomendaciones

- - Nunca pubiques tu TOKEN en repositorios públicos (GitHub, GitLab, etc.).
- - Para mayor seguridad, guarda el TOKEN en una variable de entorno o en un archivo .env que no compartas.
- - Primero haz pruebas solo contigo (usando tu propio chat\_id).
- - Cuando confirmes que los mensajes llegan bien y en el horario correcto, agrega los chat\_id de tus clientes reales.
- - Revisa que la hora del sistema (computadora o servidor) esté bien configurada en la zona horaria correcta.
- - Si el proyecto crece, puedes agregar manejo de errores (try/except) y guardar logs de qué mensajes se enviaron.

*Con este manual tienes lo necesario para conectar Python con Telegram y enviar recordatorios automáticos de citas 2 horas antes del horario programado.*