

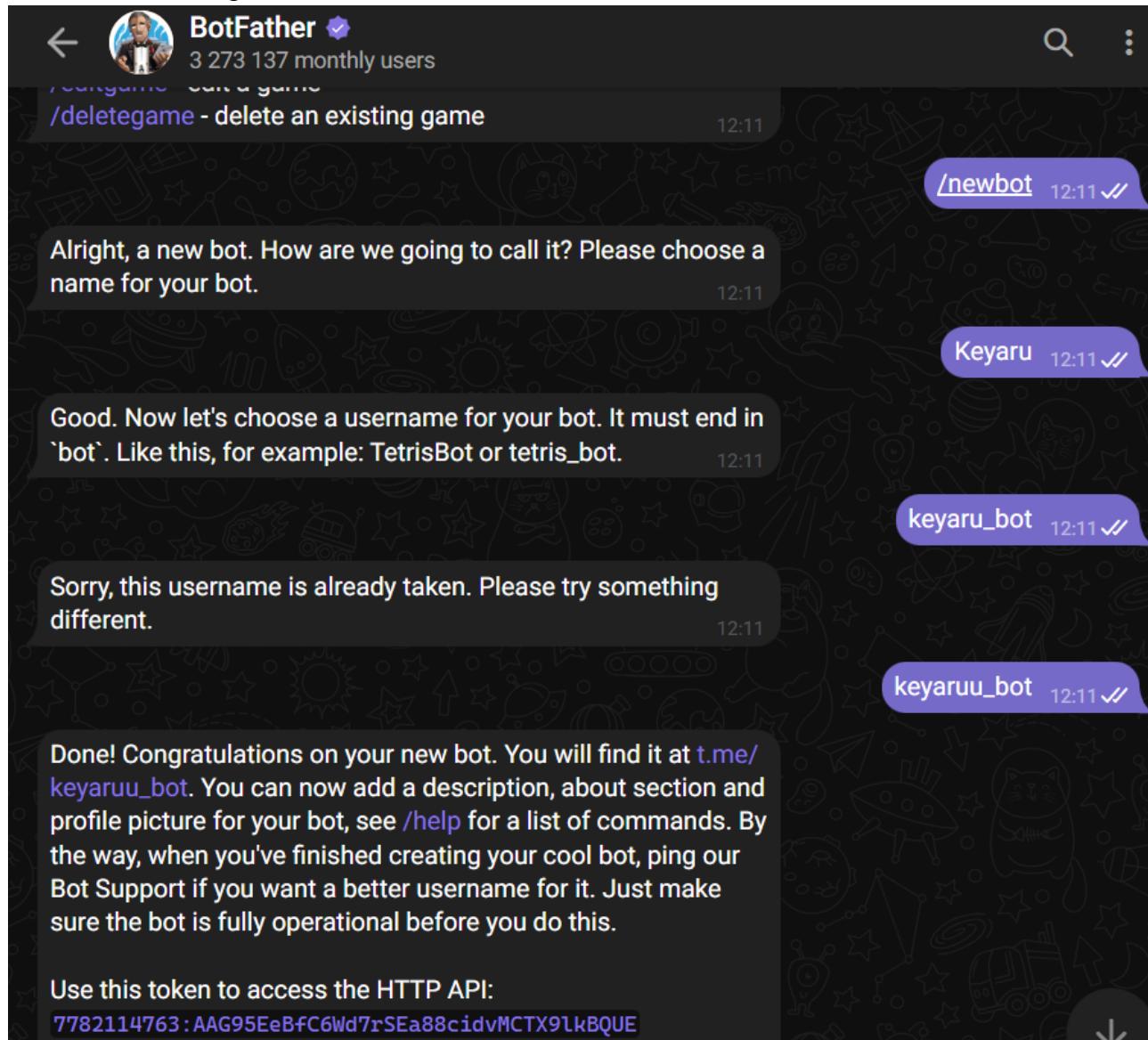
Actividad Extracurricular de Logging en Telegram

- [Repositorio](#)

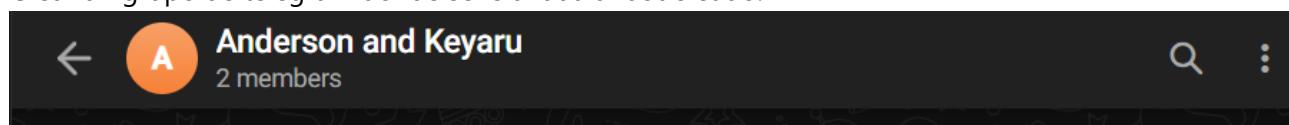
El logging ayuda a mantener un registro detallado de lo que hace tu programa, facilitando a la detección de problemas y el monitoreo del funcionamiento.

Cree un ejemplo que: realice logging en consola, en un archivo, en un mensaje de Telegram.

1. Crear un bot en telegram en el chat de @BotFather



2. Crear un grupo de telegram donde se le añada al bot creado.



3. Crear un script en python para recibir el id del chat donde se coloca el url y el token que se recibió al crear el bot.

Antes de ejecutar hay que instalar

```
pip install requests
```

```
import requests

TOKEN = '7782114763:AAG95EeBfC6Wd7rSEa88cidvMCTX91kBQUE'
URL = f'https://api.telegram.org/bot{TOKEN}/getUpdates'
response = requests.get(URL)
print(response.json())
```

4. Crear un archivo .py donde:

- Consola: Muestra logs de nivel INFO y superior
- Archivo app.log: Guarda todos los logs (DEBUG y superior)
- Telegram: Envía logs de nivel WARNING y superior al grupo

```
import logging
import requests

# ===== CONFIGURACIÓN =====
TELEGRAM_BOT_TOKEN = "7782114763:AAG95EeBfC6Wd7rSEa88cidvMCTX91kBQUE"
TELEGRAM_CHAT_ID = "-4813543477" # ID del grupo (negativo)

# ===== HANDLER PARA TELEGRAM =====
class TelegramHandler(logging.Handler):
    def emit(self, record):
        msg = self.format(record)
        # Limitar a 4000 caracteres para evitar errores
        if len(msg) > 4000:
            msg = msg[:4000] + "..."

        # URL correcta
        url = f"https://api.telegram.org/bot{TELEGRAM_BOT_TOKEN}/sendMessage"
        data = {
            'chat_id': TELEGRAM_CHAT_ID,
            'text': f"LOG: {msg}",
            'parse_mode': 'HTML'
        }
        try:
            response = requests.post(url, data=data, timeout=10)
            if response.status_code != 200:
                print(f"Error Telegram: {response.status_code} - {response.text}")
        except Exception as e:
            print(f"Error enviar Telegram: {e}")

# ===== CONFIGURACIÓN DEL LOGGER =====
def setup_logger():
```

```
logger = logging.getLogger()
logger.setLevel(logging.DEBUG)

formatter = logging.Formatter(
    '%(asctime)s - %(levelname)s - %(message)s',
    datefmt='%Y-%m-%d %H:%M:%S'
)

# Handler para consola (INFO+)
console_handler = logging.StreamHandler()
console_handler.setLevel(logging.INFO)
console_handler.setFormatter(formatter)

# Handler para archivo (DEBUG+)
file_handler = logging.FileHandler('app.log')
file_handler.setLevel(logging.DEBUG)
file_handler.setFormatter(formatter)

# Handler para Telegram (WARNING+)
telegram_handler = TelegramHandler()
telegram_handler.setLevel(logging.WARNING)
telegram_handler.setFormatter(formatter)

# Añadir handlers al logger
logger.addHandler(console_handler)
logger.addHandler(file_handler)
logger.addHandler(telegram_handler)

return logger

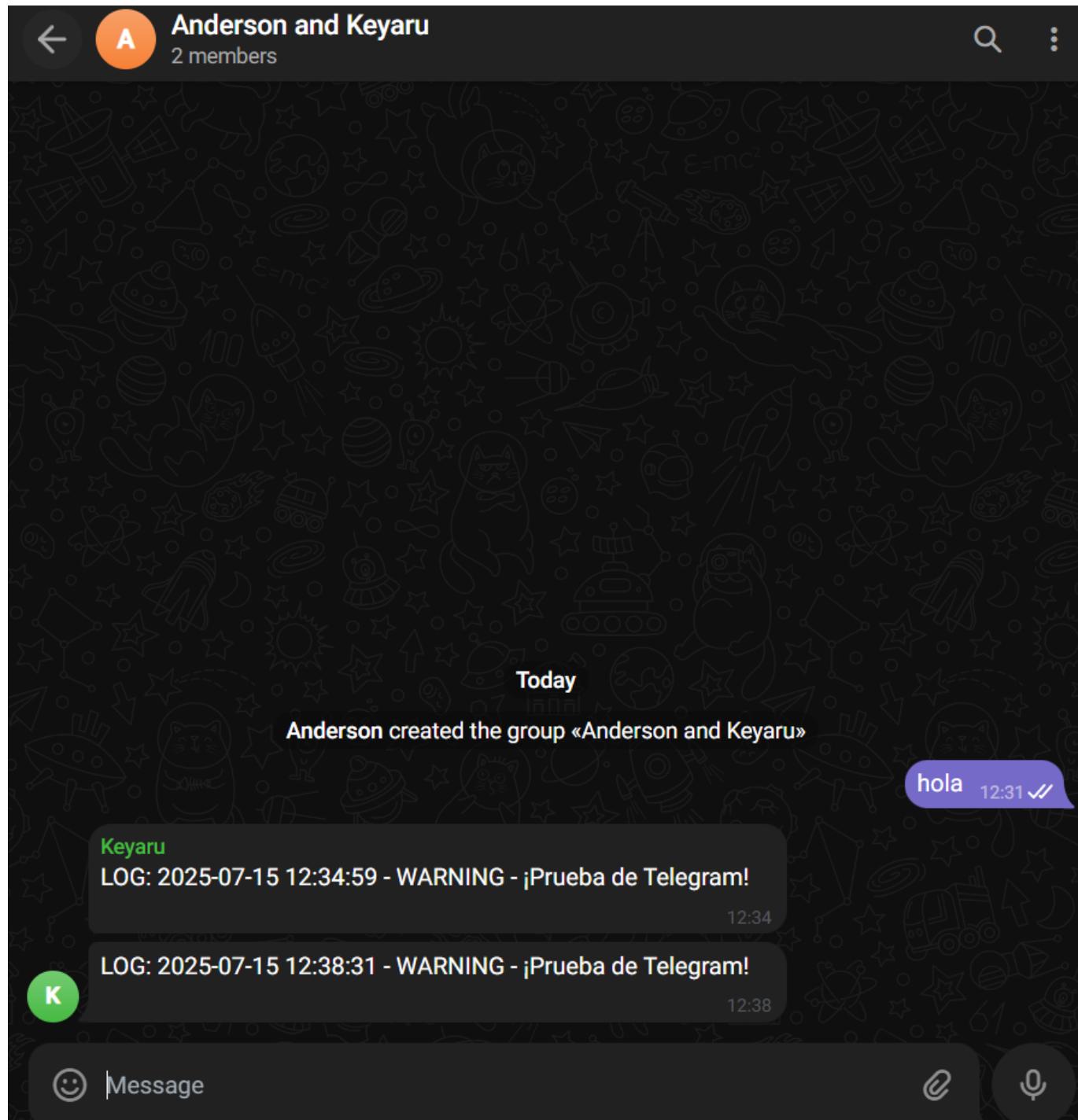
# ===== EJECUCIÓN =====
if __name__ == "__main__":
    logger = setup_logger()

    # Logs normales
    logger.info("Iniciando aplicación...")
    logger.warning("¡Prueba de Telegram!") # Se envía a Telegram

    # Log de excepción
    try:
        1/0 # Forzar error
    except Exception as e:
        logger.exception("Ocurrió una excepción") # Se envía a Telegram

    logger.info("Pruebas completadas")
```

5. Resultado



Modifique el logging para mostrar el nombre del archivo, incluir la fecha y la hora, cambiar de color, etc.

- Muestra del nombre del archivo y línea
- Colores en la consola
- Formato de fecha/hora personalizable
- Logueo a archivo con rotación

Antes de ejecutar hay que instalar

```
pip install colorama
```

```
import logging
import requests
from colorama import init, Fore, Style
from logging.handlers import RotatingFileHandler

# Inicializar colorama
init()

# ===== CONFIGURACIÓN =====
TELEGRAM_BOT_TOKEN = "7782114763:AAG95EeBfC6Wd7rSEa88cidvMCTX91kBQUE"
TELEGRAM_CHAT_ID = "-1002513850512" # Grupo de Telegram

# ===== HANDLER PARA TELEGRAM =====
class TelegramHandler(logging.Handler):
    def emit(self, record):
        msg = self.format(record)
        if len(msg) > 4000:
            msg = msg[:4000] + "..."

        url = f"https://api.telegram.org/bot {TELEGRAM_BOT_TOKEN}/sendMessage"
        data = {
            'chat_id': TELEGRAM_CHAT_ID,
            'text': f"💡 LOG: {msg}",
            'parse_mode': 'HTML'
        }
        try:
            response = requests.post(url, data=data, timeout=10)
            if response.status_code != 200:
                print(f"Error Telegram: {response.status_code} - {response.text}")
        except Exception as e:
            print(f"Error al enviar a Telegram: {e}")

# ===== FORMATO COLOREADO =====
class ColoredFormatter(logging.Formatter):
    COLORS = {
        logging.DEBUG: Fore.CYAN,
        logging.INFO: Fore.GREEN,
        logging.WARNING: Fore.YELLOW,
        logging.ERROR: Fore.RED,
        logging.CRITICAL: Fore.RED + Style.BRIGHT
    }

    def format(self, record):
        message = super().format(record)
        color = self.COLORS.get(record.levelno, Fore.WHITE)
        return f"{color}{message}{Style.RESET_ALL}"

# ===== CONFIGURACIÓN DEL LOGGER =====
def setup_logger():
    logger = logging.getLogger()
    logger.setLevel(logging.DEBUG)

    # Formato con detalles
```

```
detailed_format = '%(asctime)s | %(filename)s:%(lineno)d | %(funcName)s | %\n(levelname)s | %(message)s'\n\ndate_format = '%Y-%m-%d %H:%M:%S'\n\n# Formato coloreado para consola\nconsole_formatter = ColoredFormatter(detailed_format, datefmt=date_format)\n\n# Formato plano para archivo y Telegram\nplain_formatter = logging.Formatter(detailed_format, datefmt=date_format)\n\n# Handler para consola\nconsole_handler = logging.StreamHandler()\nconsole_handler.setLevel(logging.INFO)\nconsole_handler.setFormatter(console_formatter)\n\n# Handler para archivo (rotación)\nfile_handler = RotatingFileHandler(\n    'app.log',\n    maxBytes=1_000_000, # 1MB\n    backupCount=5,\n    encoding='utf-8'\n)\nfile_handler.setLevel(logging.DEBUG)\nfile_handler.setFormatter(plain_formatter)\n\n# Handler para Telegram\ntelegram_handler = TelegramHandler()\ntelegram_handler.setLevel(logging.WARNING)\ntelegram_handler.setFormatter(plain_formatter)\n\n# Limpiar handlers previos\nlogger.handlers.clear()\n\n# Añadir handlers\nlogger.addHandler(console_handler)\nlogger.addHandler(file_handler)\nlogger.addHandler(telegram_handler)\n\nreturn logger\n\n# ===== EJECUCIÓN ======\nif __name__ == "__main__":\n    logger = setup_logger()\n\n    logger.info("Iniciando aplicación...")\n    logger.warning("¡Prueba de Telegram!") # Este se envía a Telegram\n\n    try:\n        1 / 0 # Forzar error\n    except Exception as e:\n        logger.exception("Ocurrió una excepción")\n\n    logger.info("Pruebas completadas")
```

```
2025-07-15 13:29:32 - INFO - Iniciando aplicaci n...
2025-07-15 13:29:32 - WARNING - Prueba de Telegram!
2025-07-15 13:29:32 - ERROR - Ocurri  una excepci n
Traceback (most recent call last):
```

- ANTES

- AHORA

```
2025-07-15 13:32:00 | telegram_logger.py:97 | <module> | INFO | Iniciando aplicaci n...
2025-07-15 13:32:00 | telegram_logger.py:98 | <module> | WARNING | Prueba de Telegram!
Error Telegram: 404 - {"ok":false,"error_code":404,"description":"Not Found"}
2025-07-15 13:32:00 | telegram_logger.py:103 | <module> | ERROR | Ocurri  una excepci n
```