

# Bitácora de Desarrollo: Proyecto MarathonBot

**Equipo de Desarrollo:** Justo Pérez Viceconte, Giuliana Bressi Vaglivello, Federico Solanes, Martina Roldán.

**Materia:** Taller de Programación II

## Entrada 1: Definición de la Arquitectura

**Objetivo:** Establecer la estructura del código para cumplir con la consigna de Programación Orientada a Objetos (POO).

**Decisión:** Se decidió no escribir todo el código en un solo archivo para facilitar el mantenimiento. Dividimos el sistema en tres módulos principales:

1. `bot.py`: Módulo de Coordinación (Interfaz con Telegram).
2. `data_fetcher.py`: Módulo de Extracción (Conexión con API TMDb).
3. `data_processor.py`: Módulo de Lógica (Cálculos matemáticos y generación de gráficos).

## Entrada 2: Implementación de la API de TMDb y Manejo de Errores

**Problema Encontrado:** Error de Autenticación (401 Unauthorized).

**Descripción del problema:** Al realizar las primeras pruebas de conexión con el módulo `data_fetcher.py` para buscar la serie "Breaking Bad", la consola arrojó un error `401 Client Error: Unauthorized`.

**Análisis:** Revisando la documentación de la API de The Movie Database (TMDb) y comparando con las credenciales utilizadas en nuestro archivo `config.py`, detectamos que estábamos utilizando el "API Read Access Token" (una cadena muy larga diseñada para autenticación Bearer) en lugar de la "API Key (v3 Auth)" que requiere la librería `requests` para consultas simples.

**Solución:** Se procedió a generar una nueva API Key v3 desde el panel de desarrollador de TMDb, se actualizó el archivo de configuración y la conexión fue exitosa, permitiendo recuperar los metadatos de las series.

## Entrada 3: Desarrollo del Valor Agregado (Gráficos)

**Desafío:** Cumplir con el requisito de procesar datos para comunicar valor agregado visual.

**Implementación:** Integramos la librería `Matplotlib`. Se programó la clase `DataProcessor` para que, una vez calculados los tiempos de "Pantalla" vs "Pausas", genere dinámicamente un archivo de imagen (`.png`).

**Resultado:** El bot ahora no solo envía texto, sino que adjunta un Gráfico de Pastel (Pie Chart) que facilita la lectura del plan de maratón.

## Entrada 4: Integración Final y Despliegue

**Acción:** Se unificaron los módulos en el archivo principal `bot.py` y se realizaron pruebas de estrés con diferentes títulos (películas y series). Se subió el código final al repositorio de GitHub, asegurando mediante `.gitignore` (o exclusión manual) que el archivo `config.py` no fuera público.