

Memoria del Proyecto: Pokémon Toolkit & Z-A Manager

IFP (Instituto de Formación Profesional)

Autores: Sergi Fernández y Álvaro Rodríguez

Fecha: Mayo 2024

Asignatura: Proyecto (Antonio Guil)

1. Introducción

1.1. Propósito del Proyecto

El proyecto nace de la necesidad de centralizar información técnica del ecosistema Pokémon (PokeAPI) con datos estratégicos obtenidos de la web (Web Scraping). El objetivo es ofrecer una herramienta funcional tanto para usuarios casuales como para jugadores competitivos.

1.2. Motivación

La elección de este proyecto se basa en el interés de los autores por la automatización de procesos mediante Python y la gestión de interfaces gráficas para mejorar la experiencia de usuario.

2. Definición del Proyecto

2.1. Objetivos Principales

- Implementar un sistema de autenticación seguro y persistente.
- Consumir datos en tiempo real de la API oficial de Pokémon.
- Desarrollar módulos de scraping para obtener información no disponible en APIs.
- Ofrecer una interfaz dual: CLI (Consola) y GUI (Gráfica).

2.2. Alcance

El sistema permite el registro de usuarios, la consulta de estadísticas base, la visualización de mejores movimientos ofensivos/defensivos y la gestión de tablas de megaevoluciones.

3. Análisis Técnico

3.1. Tecnologías Utilizadas

- **Lenguaje:** Python (Versión 3.x).
- **Interfaz:** Tkinter para la GUI y sistema de impresión formateada para CLI.
- **Gestión de Datos:** JSON para usuarios y CSV para megaevoluciones.
- **Bibliotecas Clave:** `requests` (API), `BeautifulSoup4` (Scraping), `Pandas` (Tablas), `Pillow` (Imágenes), `OS` (Creación de carpetas).

3.2. Arquitectura de Archivos

El proyecto sigue una estructura modular para separar la lógica de negocio de la interfaz:

- `src/auth/`: Lógica de usuarios y seguridad.
- `src/scraping/`: Extracción y consumo de datos externos.
- Raíz: Puntos de entrada y archivos de configuración.

```
APP-PROJECT-MAIN/
├── .venv/
├── data/
│   ├── megaevoluciones_general.csv
│   ├── megaevoluciones_luminaila.csv
│   ├── moveset_pokemon_6.csv
│   └── users.json
└── src/
    ├── auth/
    │   ├── __pycache__/
    │   │   ├── __init__.py
    │   │   ├── login.py
    │   │   ├── registrarUser.py
    │   │   └── user.py
    |
    ├── scraping/
    │   ├── __pycache__/
    │   │   ├── __init__.py
    │   │   ├── pokemon_api.py
    │   │   ├── scrape_pokemon_go.py
    │   │   ├── scrape_pokemon_za.py
    │   │   └── ver_clasificacion.py
    |
    ├── main_gui.py
    └── main.py

```

└── README.md
└── requirements.txt

4. Desarrollo y Metodología

4.1. Reparto de Tareas

Tarea	Responsable Principal
Diseño de Interfaz (GUI)	Sergi y Álvaro
Lógica de Autenticación y JSON	Álvaro
Módulos de Scraping (BS4)	Sergi
Integración de PokeAPI y Sprites	Sergi y Álvaro
Documentación y Pruebas	Sergi y Álvaro

4.2. Desafíos Técnicos

- **Validaciones:** Implementación de expresiones regulares para garantizar la seguridad de las contraseñas.
- **Concurrencia:** Gestión del tiempo de carga de imágenes en la interfaz gráfica para evitar bloqueos.

5. Pruebas y Validación

- **Pruebas de Usuario:** Registro con correos inválidos y contraseñas débiles para comprobar el sistema de alertas.
- **Pruebas de Conexión:** Manejo de excepciones en caso de que la API o la web de destino no estén disponibles.

6. Conclusiones

6.1. Logros Alcanzados

Hemos logrado una aplicación funcional que integra con éxito diversas fuentes de datos. La separación entre consola e interfaz gráfica permite que la herramienta sea versátil. También hemos aprendido a combinar las APIs con el web scraping para extraer datos.

6.2. Futuras Mejoras

- Implementación de base de datos SQL (SQLite) para mayor escalabilidad.
- Añadir comparación de estadísticas entre dos Pokémon.
- Generación de reportes PDF con los datos del equipo del usuario.
- Mejora del GUI (apartado gráfico)

7. Bibliografía

- Documentación oficial de Python: docs.python.org
- PokeAPI: pokeapi.co
- BeautifulSoup Documentation.