

Trabajo Práctico – Introducción a la Programación

Alumno: Federico Gómez

Grupo: sin grupo (para hablar con profesores)

Fecha: 23 de junio de 2025

1. Introducción

En este trabajo práctico desarrollamos una aplicación web que permite explorar una galería de Pokémon. Cada Pokémon se muestra en una tarjeta con su imagen, tipos, altura, peso y nivel de experiencia base. Se puede buscar por nombre, filtrar por tipo (fuego, agua o planta) y marcar como favorito si el usuario está logeado.

El trabajo se basó en una estructura previamente armada, donde nuestro rol fue completar la lógica central, tanto para mostrar los Pokémon como para aplicar filtros y gestionar los favoritos.

2. Funciones implementadas y/o utilizadas

services.py

- **getAllImages():** transforma los datos de la API en tarjetas visuales con íconos y color según tipo.
- **filterByCharacter(name):** filtra por nombre, ignorando mayúsculas.
- **filterByType(type_filter):** filtra por tipo (fuego, agua, planta).
- **saveFavourite(), getAllFavourites(), deleteFavourite():** gestionan los favoritos.
- **get_type_icon_url_by_name():** trae la URL del ícono según el tipo.

views.py

- **home(), search(), filter_by_type():** muestran los datos según filtros o búsquedas.
- **saveFavourite(), deleteFavourite(), getAllFavouritesByUser():** vistas protegidas para manejar favoritos.

home.html

- Muestra tarjetas con diseño adaptado, íconos de tipo, color de borde y botones para favoritos.

translator.py

- **fromFavouriteIntoCard(fav):** convierte un favorito desde la base de datos en una Card.

3. Dificultades y decisiones tomadas

- Tuve un problema principal donde por algún motivo no me funcionaba la api, probé distintas formas de solucionarlo: apagando firewall, antivirus, etc, utilizando datos desde el mobile, ingresando desde android, ordenador de mi vecina, utilizando vpn (proton y widscribe) pero siempre me un error similar a la captura adjunta:

ConnectionError at /home/

HTTPSConnectionPool(host='pokeapi.co', port=443): Max retries exceeded with url: /api/v2/pokemon/1 (Caused by NewConnectionError('<urllib3.connection.HTTPSConnection object at 0x0000022EB97DB770>: Failed to establish a new connection: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión'))

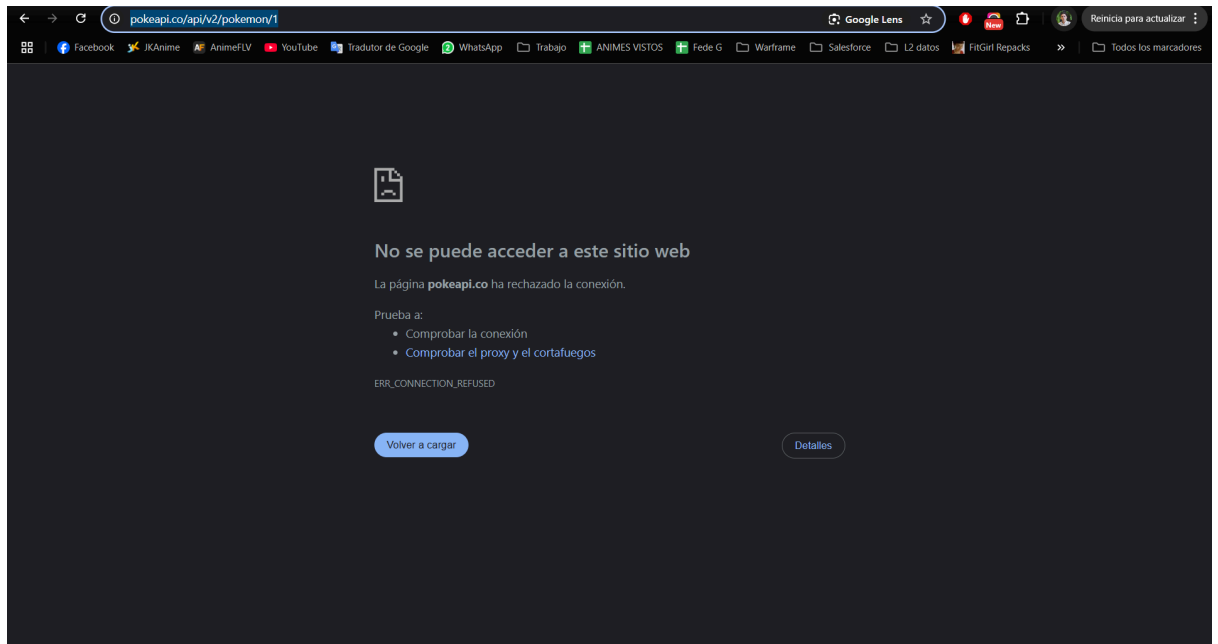
```
Request Method: GET
Request URL: http://localhost:3000/home/
Django Version: 4.2.10
Exception Type: ConnectionError
Exception Value: HTTPSConnectionPool(host='pokeapi.co', port=443): Max retries exceeded with url: /api/v2/pokemon/1 (Caused by NewConnectionError('<urllib3.connection.HTTPSConnection object at 0x0000022EB97DB770>: Failed to establish a new connection: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión'))
Exception Location: C:\Users\Feder\AppData\Local\Programs\Python\Python313\Lib\site-packages\requests\adapters.py, line 519, in send
Raised during: app.views.home
Python Executable: C:\Users\Feder\AppData\Local\Programs\Python\Python313\python.exe
Python Version: 3.13.5
Python Path: ['C:\\Users\\Feder\\OneDrive\\Escritorio\\Proyectos\\ip-1c2025-alumnos',
'C:\\Users\\Feder\\AppData\\Local\\Programs\\Python\\Python313\\python',
'C:\\Users\\Feder\\AppData\\Local\\Programs\\Python\\Python313\\DLLs',
'C:\\Users\\Feder\\AppData\\Local\\Programs\\Python\\Python313\\Lib',
'C:\\Users\\Feder\\AppData\\Local\\Programs\\Python\\Python313\\',
'C:\\Users\\Feder\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages']
Server time: Mon, 23 Jun 2025 03:30:02 +0000
```

Traceback [Switch to copy-and-paste view](#)

```
C:\Users\Feder\AppData\Local\Programs\Python\Python313\Lib\site-packages\urllib3\connection.py, line 198, in _new_conn
198.         sock = connection.create_connection(
> Local vars

C:\Users\Feder\AppData\Local\Programs\Python\Python313\Lib\site-packages\urllib3\util\connection.py, line 85, in create_connection
85.         raise err
> Local vars

C:\Users\Feder\AppData\Local\Programs\Python\Python313\Lib\site-packages\urllib3\util\connection.py, line 73, in create_connection
73.         sock.connect(sa)
> Local vars
```



para poder trabajar tuve que estar ingresando via anydesk al ordenador de un amigo de españa con mac y trabajando de esta manera (sigo sin saber si sera mi proveedor movistar o si por tener ip dinamica me tocan algunas malas a veces).

- Cambiamos el diseño de las tarjetas para mostrar íconos en vez de texto para los tipos.
- Se aplicó un color de borde diferente según el tipo principal del Pokémon.
- Se usó el filtro `safe` para poder renderizar los íconos en HTML porque por tema de seguridad no funcionaban con django.
- Se normalizaron los textos para búsquedas y filtros, comparando todo en lower.
- Creamos la variable `typetext` para comparar con los filtros de type desde botones con value.
- Se ajustó `home()` para que muestre correctamente los favoritos del usuario logueado.
- En `translator.py`, usamos `eval()` para reconstruir la lista de tipos al convertir datos desde la base de datos.