**UNIVERSIDAD NACIONAL DE GENERAL SARMIENTO**

**Materia:** Introducción a la programación – Comisión 11  
**Profesores:** Omar Argañaras y Luca Velázquez  
**Alumno:** Kilian Alfaro  
**Fecha de entrega:** 23/06/2025

**Trabajo Práctico – Galería de Imágenes con API**

**Índice**

**1. Introducción**

Este trabajo consistió en desarrollar una galería web que muestra imágenes de una base de datos en línea, permitiendo a los usuarios registrados guardar sus imágenes favoritas. El objetivo era construir una interfaz interactiva y visualmente clara para navegar entre los personajes obtenidos desde una API pública.

Se utilizaron las siguientes tecnologías:

* **Django** (framework web en Python)
* **HTML y CSS** (para el diseño visual)
* **Sistema de autenticación de usuarios**
* **Caché** para mejorar el rendimiento
* **PokéAPI**, como fuente de datos para las imágenes y atributos

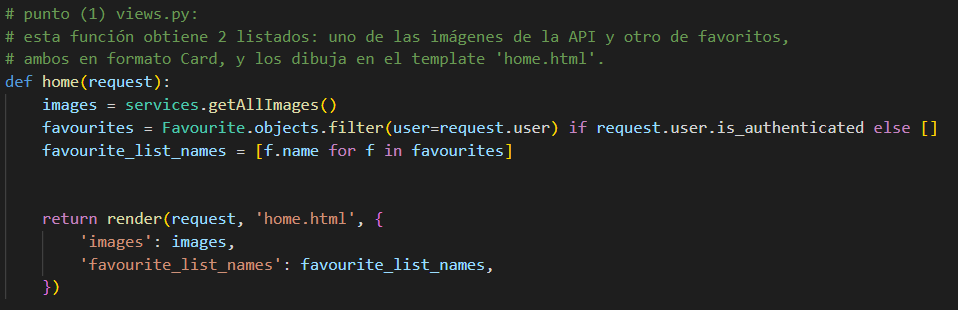
**2. Desarrollo**

El código desarrollado se encarga de:

* Obtener los datos de la API y procesarlos
* Generar una card para cada pokemon
* Detectar el tipo del personaje y asignarle un color visual distintivo
* Permitir que los usuarios guarden favoritos
* Agilizar mediante cache

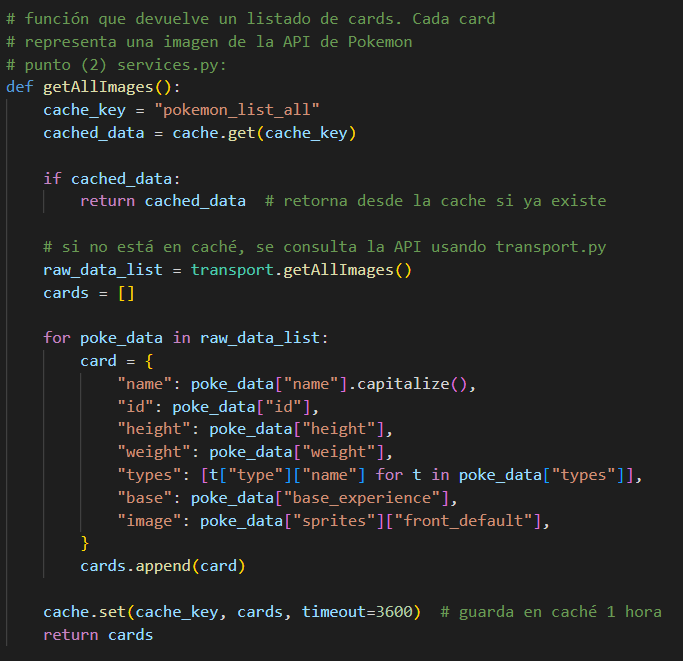
2.1 Función home (views.py)

Esta función recupera todas las imágenes desde la API y, si el usuario está autenticado, también obtiene su lista de favoritos. Luego, ambas listas se envían al template home.html.



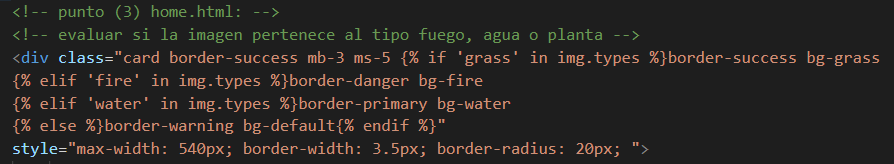
2.2 Función getAllImages (services.py)

Esta función obtiene todos los datos desde la API y los transforma en cards. También implementa cache para evitar hacer pedidos repetidos y reducir el tiempo de espera.



2.3 Estilo visual en home.html

En el template, se asigna un color de borde distinto a cada tarjeta en función del tipo del personaje:



2.4 Buscador por nombre del personaje

Se implementó una funcionalidad que permite filtrar los personajes de la galería según el nombre ingresado por el usuario.

## 

2.5 Buscador por tipo (fuego, agua o planta)

Se implementaron tres botones que permiten filtrar la galería según el tipo principal del personaje: fuego, agua o planta. Esta funcionalidad se desarrolló para facilitar el acceso a ciertos grupos característicos de personajes, utilizando como base sus tipos.



2.6 Funcionalidad de favoritos

Una de las funcionalidades principales del proyecto es permitir que un usuario **logueado** pueda guardar personajes como favoritos. Esto mejora la personalización de la experiencia, permitiendo volver rápidamente a los personajes que más le interesan al usuario.**Dificultades y decisiones**

Dificultades encontradas:

El manejo de favoritosme costó más que nada por el hecho de que era un proceso más complejo que los anteriores puntos

Decisiones tomadas:

Se implementó cache por 1 hora para almacenar los resultados de la API y mejorar la velocidad de carga.

Se agregó lógica condicional para desactivar el botón de "Agregar a favoritos" si la imagen ya estaba en la lista.

Cosas que me gustaría poder haber hecho:

La pantalla de loading

Interfaz responsive o animaciones

Modo oscuro

**3. Conclusión**

Durante el desarrollo del proyecto se afianzaron conocimientos sobre separación de capas (vistas, lógica y plantillas), manejo de APIs y personalización visual en función de datos dinámicos.

Este trabajo resultó muy útil para entender cómo unir el backend con la interfaz y cómo optimizar el rendimiento de una aplicación real. Lo más desafiante fue interpretar correctamente los datos de la API y reflejarlos visualmente de manera clara.