

Universidad Nacional de General Sarmiento

MATERIA: INTRODUCCIÓN A LA PROGRAMACIÓN

**Fecha Límite de entrega 25 de junio de 2025**

**Nombre de los profesores: Omar Javier Arganaras y Nancy Johana Nores**

**Nombre de los alumnos: Joel Ayllon y Lucas Cristaldo**

**Comisión: COM-04**

Contenido

[Introducción 4](#_Toc201691402)

[Desarrollo 4](#_Toc201691403)

[Funcionamiento general 4](#_Toc201691404)

[Conclusión 4](#_Toc201691405)

# Introducción

Este trabajo practico pide el desarrollo de una aplicación web llamada Pokedéx, que permite consultar, visualizar y filtrar imágenes de Pokémon obtenidas mediante una API y presentarlas en tarjetas (cards) con información como el nombre, los tipos, la altura, el peso y el nivel base.

Para su implementación utilizamos:

* Lenguaje: Python
* Framework: Django 4.2.10
* Herramientas: Visual Studio Code, GitHub

# Desarrollo

## Funcionamiento general

La aplicación funciona consumiendo los datos de una API externa sobre Pokémon mediante la capa *transport.py.* Estos datos son transformados por *Translator.py* en objetos **Card** que contiene la información necesaria para renderizar las tarjetas en el template *home.html*.

Al ingresar a la galería, se muestran todas las imágenes y sus datos. Los usuarios pueden filtrar los pokemones por nombre o tipo (fuego, agua, planta).

Cada **Card** cambia su color de borde según el tipo:

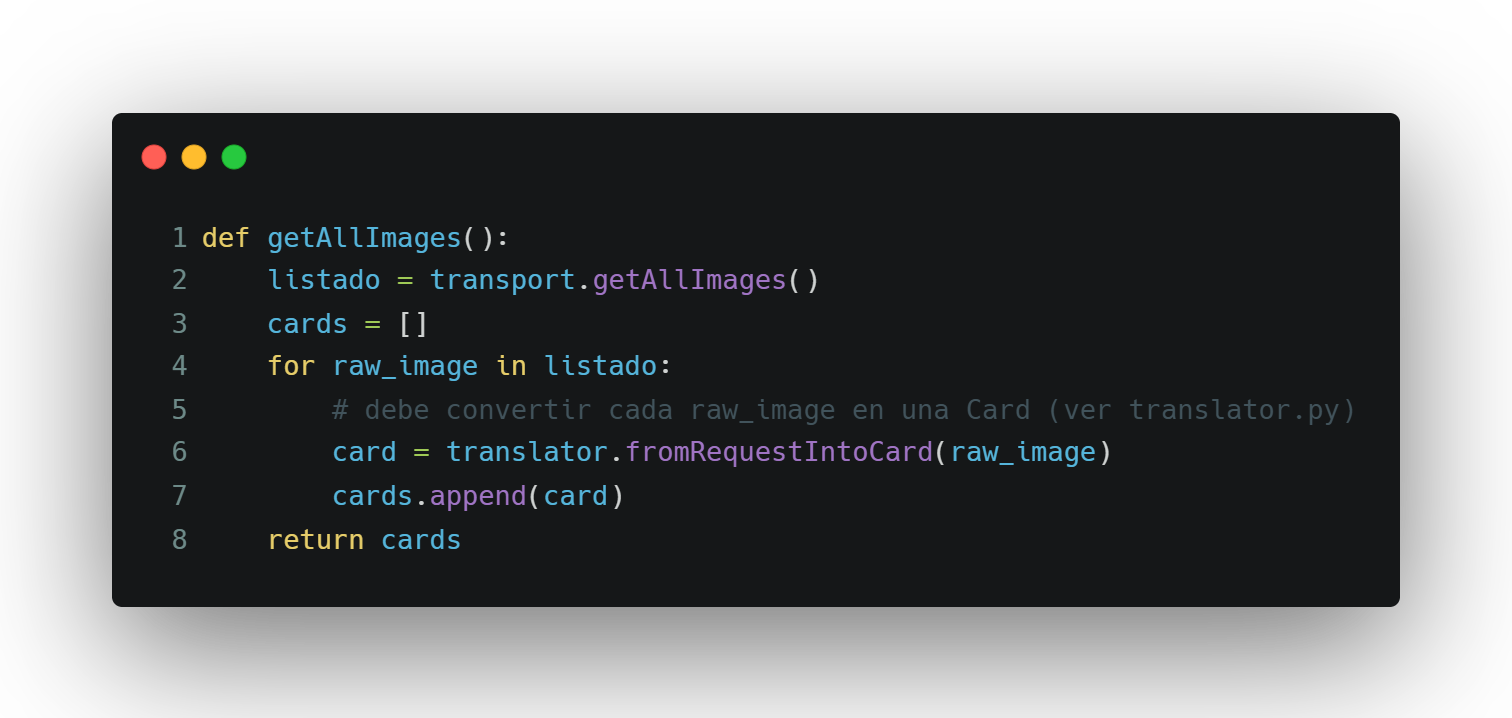
* Planta: borde verde
* Fuego: borde rojo
* Agua: borde azul
* Otros: borde naranja

## Funciones implementadas

Texto

El contenido generado por IA puede ser incorrecto.

La función home (vista.py) renderiza la vista de *home.html* obteniendo la lista de todas las imágenes y favoritos del usuario, aunque en este caso la lista de favoritos está vacía.



La función getAllImages (services.py) obtiene una lista con las imágenes desde la API y las transforma en objetos del tipo Card para guárdalo en una lista nueva y retornarla.

  
Si bien no es una función, aca se usa unos if anidados en *home.html* con el fin de cambiar los colores del borde según el tipo del Pokémon.



La función search (views.py) recibe como parámetro el *request* que contiene lo escrito por el usuario en el cuadro de búsqueda y filtra las imágenes según el nombre del Pokémon, retorna únicamente el Pokémon que el usuario busco. Si el usuario no puso nada, mostrara todos los pokemones.

Texto

El contenido generado por IA puede ser incorrecto.

La función filter\_by\_type (views.py) recibe como parámetro el tipo de Pokémon que el usuario quiere buscar (fuego, agua y planta) para poder mostrar solamente las imágenes del Pokémon de ese tipo.

Texto

El contenido generado por IA puede ser incorrecto.

La función filterByCharacter (services.py) usa la función *getAllImages()* para obtener una lista de todos los pokemones transformadas en **card** para luego recorrer dicha lista en busca del Pokémon identificado por el nombre (parámetro) y devolverlo en forma de lista.



La función filterByType (service.py) hace prácticamente los mismo que la función *filterByChapter* pero con el tipo del Pokémon en vez de con el nombre.



La función getAllImages() de la pestaña *transport.py* viene implementada de base en el proyecto pero que se reutiliza en algunas funciones desarrolladas, básicamente esta función se conecta con la API para obtener los pokemones en formato json.



La función fromRequestIntoCard de la pestaña *translator.py* tambien viene implementada de base y tiene la función de recibir el Pokémon en su formato json y la transforma en una **card** para se pueda visualizar en la página web.

## Dificultades y decisiones

La principal dificultad fue la cantidad abrumadora de información que contiene el trabajo practico tanto el código como los links externos, pero al ir avanzando con el proyecto se logró comprender lo necesario.

## Aspectos Pendientes

Nos hubiera encantado renovar la interfaz grafica y agregarle el símbolo de carga para que se vea mejor estéticamente, pero por falta de tiempo no se logró.

Intentamos implementar la función de los “favoritos” pero tampoco se pudo avanzar en ello.

# Conclusión

En este trabajo practico se pudo apreciar una curva de dificultad bastante marcada en cuanto a los puntos a resolver, lo esencial y la primera funcionalidad tuvieron su dificultad en su resolución pero en comparación a los puntos finales resultaron ser mas fáciles.