

Informe

Pokedex

Introducción a la Programación

Profesor: German Lucero

Alumna: María Luz Genre

Comisión: 13

Junio 2025

Este trabajo se refiere a los Pokemons, nombre que proviene de Pocket Monsters (monstruos de bolsillo).

La historia, de origen japonés, trata de un mundo ficticio lleno de criaturas llamadas Pokemons, los cuales tienen distintos tipos de características y habilidades.

En este mundo existen personas que los entrenan y los hacen combatir entre ellos.

Una de estas personas es el entrenador llamado Ash, que viaja por diferentes regiones recolectando Pokemones, y se crean historias alrededor de este grupo.

La historias de estos personajes dieron lugar a juegos, películas, presentaciones en eventos, ropa, merchandising.

La Pokédex es la que se relaciona con este trabajo. Intenta que los fans de los Pokemons (que llamamos “user”: usuarios) accedan a través de una página web, a las tarjetas (cards) conteniendo imagen y características de cada Pokemon, y también puedan seleccionar a sus Pokémons favoritos, pues los que existen son muchos.

Para ello se realizó una aplicación sobre una estructura que se utiliza para la creación de páginas web llamada Django.

Esta estructura está escrita principalmente en lenguaje Python.

En los módulos: view, service y home contienen:

- View contiene funciones que reciben pedidos desde el navegador, y responden los mismos.
- Service contiene funciones que se conectan con la API (acrónimo de Application Programing Interface) para acceder a los lugares donde la información original de los Pokemons se encuentra alojada. También la transforma en cards.
- Home dónde está codificada la parte central de la página, en el formato HTML que se usa para visualizar las páginas de internet.

Veamos lo completado en cada módulo:

En módulo view.py:

Función home:

- Se encarga de mostrar la totalidad de las imágenes de los pokemons.
- Images es la variable con qué llamamos a la función getAllImages del módulo services.

```
def home(request):
    images = services.getAllImages()
    favourite_list = []

    return render(request, 'home.html', { 'images': images, 'favourite_list': favourite_list })
```

Función search:

- Se encarga de encontrar todas las tarjetas y filtrar cada pokémon por su nombre, lo hace llamando a la función filterByCharacter (personaje) del módulo services.

```
def search(request):
    name = request.POST.get('query', '')

    # si el usuario ingresó algo en el buscador, se deben filtrar las imágenes por dicho ingreso.
    if (name != ''):
        images = services.filterByCharacter(name)
        favourite_list = []

        return render(request, 'home.html', { 'images': images, 'favourite_list': favourite_list })
    else:
        return redirect('home')
```

Función filter_by_type:

- Filtra los pokemons por tipo (Ejemplo, si se elige tipo agua se muestra a squirtle, wartortle y blastoise), a través de la función filterByType del módulo services.

```
def filter_by_type(request):
    type = request.POST.get('type', '')

    if type != '':
        images = services.filterByType(type) # debe traer un listado filtrado de imágenes, según si
        favourite_list = []

        return render(request, 'home.html', { 'images': images, 'favourite_list': favourite_list })
    else:
        return redirect('home')
```

En el módulo services:

Función getAllImages:

- Devuelve el listado completo de tarjetas.
- Primero la variable imagenescrudas llama a la función getAllImages del módulo transport.
- Segundo, cada una de las imágenes crudas (Formato JSON) pasa a formar una card (tarjeta).
- Finalmente esa tarjeta se añade a la variable cards.

```
def getAllImages():  
  
    # debe ejecutar los siguientes pasos:  
    # 1) traer un listado de imágenes crudas desde la  
    imagenescrudas = transport.getAllImages()  
    #  
    cards = []  
  
    # 2) convertir cada img. en una card.  
    for crudo in imagenescrudas:  
        card = translator.fromRequestIntoCard(crudo)  
  
    # 3) añadirlas a un nuevo listado que, finalmente  
    cards.append(card)  
    return cards
```

Función filterByCharacter:

- Esta función filtra a los Pokemons según su nombre
- Define un bucle for para cada tarjeta, pregunta si el nombre consultado coincide con el nombre de la tarjeta y si es así lo suma a la variable filtered_cards.

```
# función que filtra según el nombre del pokemon.
def filterByCharacter(name):
    filtered_cards = []

    for card in getAllImages():
        if name.lower() in card.name.lower():
            filtered_cards.append(card)
    return filtered_cards
```

Función filterByType:

- Esta función filtra los Pokemons por su tipo (agua, fuego, etc) para que luego sea mostrada por views
- Se inicia un bucle for, que para cada tarjeta, primero analiza si el tipo que figura en la tarjeta a filtrar coincide con alguno de los tipos existentes. En caso de que efectivamente coincida, lo agrega a la variable filtered_cards.

```
def filterByType(type_filter):
    filtered_cards = []

    for card in getAllImages():
        if type_filter.lower() in [t.lower() for t in card.types]:
            filtered_cards.append(card)
    return filtered_cards
```

En home:

- Se cambiaron los colores de los recuadros de las cards de acuerdo a lo solicitado:
Border-Danger (rojo) para tipo fuego.
Border-Primary (azul) para tipo agua.
Border-Success (verde) para tipo planta.
Border-Warning (naranja) para los demás tipos.

```
<!-- evaluar si la imagen pertenece al tipo fuego, agua o planta -->
<!--div class="card border-success mb-3 ms-5" style="max-width: 540px; "-->
<div class="card
  {% if 'fire' in img.types %}
    border-danger
  {% elif 'water' in img.types %}
    border-primary
  {% elif 'grass' in img.types %}
    border-success
  {% else %}
    border-warning
  {% endif %}
  mb-3 ms-5" style="max-width: 540px;">
```

Dificultades y toma de decisiones:

Para ingresar a favoritos es necesario primero registrar un usuario.

Miré repetidamente el tutorial de login y logout que está en inglés y no pude comprender como hacerlo.

Intenté con información de la web e inteligencia artificial y logré colocar la solapa registro, también cargar un usuario y que lo aceptara.

La primera vez se veía la página principal con el botón favoritos, pero cuando probé para marcar favoritos me envió a la página de error. A partir de ahí tampoco cargó nuevamente la página principal home. No pude resolver el conflicto.

Al no poder ingresar un usuario no pude avanzar con el opcional de favoritos.