

INFORME TP INTRODUCCIÓN A LA PROGRAMACION

El objetivo de este trabajo fue desarrollar una página web en Django que interactúe con la API de Rick & Morty para buscar y mostrar información de personajes, permitiendo también que los usuarios autenticados gestionen sus personajes favoritos, que personajes están vivos, qué personajes no, cuál fue su última ubicación y el primer episodio en el cual aparece el personaje. Este sistema permite practicar el manejo de APIs, renderización dinámica de datos y creación de interfaces interactivas con funcionalidades centradas en la experiencia del usuario.

Lo primero que se hizo, fue agregar y modificar algunos códigos dentro del views.py, siendo así agregado primeramente el siguiente código:

```
def getAllImages(input=None):
    url = f"https://rickandmortyapi.com/api/character/?name={input}" if input else "https://rickandmortyapi.com/api/character"
    response = requests.get(url)
    data = response.json()
    print(data) # Verifica qué datos estás obteniendo desde la API
    characters = data.get('results', [])
    cards = []
    for character in characters:
        cards.append({
            'name': character['name'],
            'image': character['image'],
            'status': character['status'],
            'location': character['location']['name'],
            'episode': character['episode'][0].split("/")[-1]
        })
    return cards
```

Este código cumple la función de Construcción del URL. La función toma un parámetro opcional input. Si input contiene un valor (por ejemplo, el nombre de un personaje), La URL incluye este parámetro para filtrar resultados. Si no, carga todos los personajes. Se utiliza el requests para realizar una solicitud GET a la API, recuperando datos en formato JSON. Luego accede a la clave results del JSON (que contiene los personajes) y crea una lista llamada cards con datos específicos de cada personaje. El nombre, la imagen del personaje, el estado del personaje, la última ubicación y por último el primer episodio donde aparece el personaje. Se ejecuta un print(data) para verificar la estructura del JSON obtenido durante el desarrollo.

Luego se modificó el comando de def home dentro del views.py.

```
def home(request):
    images = getAllImages() # Llama a la función
    print(images) # Agrega esto para verificar si se están recibiendo imágenes correctamente
    favourite_list = []
    return render(request, 'home.html', {'images': images, 'favourite_list': favourite_list})
```

Este código ejecuta enviar los datos necesarios al template home.html para renderizar la página principal de la página. El getAllImages llama a la función así se obtiene una lista de personajes desde la API de Rick & Morty. Puse el print para que esto se ejecute y verifique que los datos recibidos de la API estén correctos. Dejamos el favourite list vacía para los personajes favoritos del usuario.

Luego también hubo cambios dentro del archivo home.html

img.url → img.image

img.last_location → img.location

img.first_seen → img.episode

tuvimos que cambiar el nombre para que coincidan con el getALLImages que estaba en el views.py, ya que no había coincidencia con los nombres del home.html y el views.py, quedando esta parte del código del home.html de la siguiente manera:

```
<div class="row row-cols-1 row-cols-md-3 g-4">
  {% if images|length == 0 %}
    <h2 class="text-center">La búsqueda no arrojó resultados...</h2>
  {% else %}
    {% for img in images %}
      <div class="col">
        <div class="card mb-3 ms-5" style="max-width: 540px;">
          <div class="row g-0">
            <div class="col-md-4">
              
            </div>
            <div class="col-md-8">
              <div class="card-body">
                <h3 class="card-title">{{ img.name }}</h3>
                <p class="card-text">
                  <strong>
                    {% if img.status == 'Alive' %}  {{ img.status }}
                    {% elif img.status == 'Dead' %}  {{ img.status }}
                    {% else %}  {{ img.status }}
                    {% endif %}
                  </strong>
                </p>
                <p class="card-text"><small class="text-body-secondary">Última ubicación: {{ img.location }}</small></p>
                <p class="card-text"><small class="text-body-secondary">Episodio inicial: {{ img.episode }}</small></p>
              </div>
              {% if request.user.is_authenticated %}
                <div class="card-footer text-center">
                  <form method="post" action="{% url 'agregar-favorito' %}">
                    {% csrf_token %}
                    <input type="hidden" name="name" value="{{ img.name }}">
                    <input type="hidden" name="url" value="{{ img.image }}">
                    <input type="hidden" name="status" value="{{ img.status }}">
                    <input type="hidden" name="last_location" value="{{ img.location }}">
                    <input type="hidden" name="first_seen" value="{{ img.episode }}">
                    {% if img in favourite_list %}
                      <button type="submit" class="btn btn-primary btn-sm" disabled>✔ Ya está en favoritos</button>
                    {% else %}
                      <button type="submit" class="btn btn-primary btn-sm">❤️ Añadir a favoritos</button>
                    {% endif %}
                  </form>
                </div>
              {% endif %}
            </div>
          </div>
        </div>
      </div>
    {% endfor %}
  {% endif %}
</div>
</main>
{% endblock %}
```