

Trabajo Práctico Galería de Imágenes de la NASA

Introducción a la Programación

Primer Semestre 2024



Resumen: Este proyecto es una galería de imágenes de la NASA que utiliza la API de la NASA para obtener imágenes de su base de datos.

Integrantes: Franco Damián Pacheco francodamianpacheco@gmail.com

Santiago Nicolas Orlando orlandosantiagonicolos@mail.com

Yamila Aldana Villalba yamilavillalba03@gmail.com

1. Introducción: El proyecto consiste en desarrollar una galería de imágenes de la NASA que utiliza la API de la NASA para obtener imágenes de su base de datos. El objetivo es crear una aplicación que permita a los usuarios buscar y visualizar imágenes de la NASA de manera sencilla y eficiente; de igual manera se implementaron funciones que le permitirán al usuario poder registrarse en la aplicación.

2. Desarrollo:

2.1 Descripción general: En primera instancia, se buscó resolver el problema base para que la aplicación cumpla su función principal, la visualización de imágenes en Home. Para lograr la funcionalidad, se trabajó sobre los archivos *views.py* y *services_nasa_image_gallery.py*; modificando las funciones encargadas de hacer que las imágenes de la galería se muestren.

Además, se logró implementar la función *Search* para que la galería filtre las imágenes y devuelva la solicitada por el usuario, y, se completo la *Feature* para el inicio de sesión del usuario.

2.2 Funcionalidades principales: Las primeras funciones implementadas se encuentran en el archivo *views.py*:

- ***getAllImagesAndFavouriteList(request)*:** Esta es una función auxiliar, la cual invoca al servicio correspondiente para obtener 2 listados, uno de las imágenes de la API y otro de los favoritos del usuario. En nuestro caso, no se logró el desarrollo del listado con los favoritos del usuario, por lo que, la función final, solo devuelve un listado con las imágenes de la API. Puede visualizarse en el siguiente código:

```
14 def getAllImagesAndFavouriteList(request):
15     images = []
16     favourite_list = []
17
18     images= services_nasa_image_gallery.
getAllImages()
19
20     return images, favourite_list
21
```

- ***home(request)*:** Esta es la función principal de la galería. Invoca a *getAllImagesAndFavouriteList(request)* para obtener los listados que renderizan el template. Como consecuencia de no poder desarrollar la lista de favoritos en la función anterior, esta función solo devuelve 1 listado con las imágenes de la API.

```

22 # función principal de la galería.
23 def home(request):
24     # llama a la función auxiliar getAllImagesAndFavouriteList() y obtiene 2 listados: uno de las imágenes de la API y otro de favoritos por usuario*.
25     # (*) este último, solo si se desarrolló el opcional de favoritos; caso contrario, será un listado vacío [].
26     images = []
27     favourite_list = []
28
29     images, favourite_list =
    getAllImagesAndFavouriteList(request)
30
31     return render(request, 'home.html'
    , {'images': images, 'favourite_list':
    favourite_list} )

```

Dentro del mismo archivo se desarrolló la función *Search*:

- **Search:** Es la función utilizada por el buscador para filtrar las imágenes y mostrar las que el usuario solicita.

```

34 # función utilizada en el buscador.
35 def search(request):
36     images, favourite_list =
    getAllImagesAndFavouriteList(request)
37     search_msg = request.POST.get('query', '')
38
39     if search_msg:
40         images = services_nasa_image_gallery.
    getAllImages(search_msg)
41     else:
42         images = services_nasa_image_gallery.
    getAllImages()
43     return render(request, 'home.html'
    , {'images': images, 'favourite_list':
    favourite_list})

```

Por otra parte, en el archivo *services_nasa_image_gallery.py*, se desarrolló la siguiente función:

- **getAllImages(input=None):** Esta función obtiene una lista de imágenes desde el archivo *transport.py* y las guarda en una variable llamada *json_collection*. Luego, la función itera sobre la lista de imágenes y crea una *NASACard* para cada una. Finalmente, la función devuelve una lista de las *NASACards*.

```

def getAllImages(input=None):
    # obtiene un listado de imágenes desde transport.py y lo guarda en un json_collection.
    # ¡OJO! el parámetro 'input' indica si se debe buscar por un valor introducido en el buscador.
    json_collection = []

    images = []

    json_collection = transport.getAllImages()

    for json in json_collection:
        images.append(mapper.fromRequestIntoNASACard(json))

    # recorre el listado de objetos JSON, lo transforma en una NASACard y lo agrega en el listado de images. Ayuda: ver mapper.py.

    return images

```

Finalmente, dentro de los archivos *nasa_image_gallery/templates*, *nasa_image_gallery/template/registration* y *mail/urls.py*, se modificaron los archivos *header.html* y *login.html* para completar la autenticación del usuario.

3. Conclusiones: Como conclusión a este proyecto podemos decir que, este trabajo, representó un reto para nosotros como equipo, no solo porque tuvimos que equilibrar el tiempo que le dedicamos a este trabajo junto con otras responsabilidades personales, sino que también nos encontramos en un ambiente nuevo al que debíamos descifrar; a pesar de eso, pudimos organizarnos de manera eficiente para llevar a cabo el trabajo de forma exitosa. Estamos seguros de que con un poco más de tiempo y esfuerzo lograríamos que todas las funciones que conforman la aplicación, funcionen correctamente.

Anexo: [ungs-ip/ip-public-repo \(github.com\)](https://github.com/ungs-ip/ip-public-repo)