Informe Trabajo Practico – IP

En este informe, explicaré el trabajo práctico de Introducción a la Programación, sobre la página web de la NASA. Donde también veremos las funciones que fueron implementadas y modificadas a lo largo de este trabajo.

En primer lugar, le agregué la siguiente linea a la funcion "home":

images=services_nasa_image_gallery.getAllImages()

```
def home(request):
    # llama a la función auxiliar getAllImagesAndFavouriteList() y
obtiene 2 listados: uno de las imágenes de la API y otro de favoritos por
usuario*.
    # (*) este último, solo si se desarrolló el opcional de favoritos;
caso contrario, será un listado vacío [].
    images = []
    favourite_list = []

    images=services_nasa_image_gallery.getAllImages()

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

Cuya función es hacer aparecer en la pagina principal todas las imagenes que vienen por defecto.

En segundo lugar, se implementó el suguiente condicional a la funcion "getAllImagesAndFavouriteList":

if request.user.is_authenticated:

user = request.user

favourite_list = services_nasa_image_gallery.getAllFavouritesByUser(user)

else:

favourite_list = []

```
def getAllImagesAndFavouriteList(request):
    images = []
    favourite_list = []

    if request.user.is_authenticated:
        user = request.user
```

```
        favourite_list =
services_nasa_image_gallery.getAllFavouritesByUser(user)
    else:
        favourite_list = []

    return images, favourite_list
```

Ese es el condicional implementado, el cual verifica si el usuario está autenticado, y devuelve una lista de las imagenes favoritas de dicho usuario llamando a la función "getAllFavouritesByUser" utilizando al usuario como parametro.

En tercer lugar, se agregó lo siguiente a la funcion "search":

if search_msg !='':

images = services_nasa_image_gallery.getAllImages(search_msg)

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

```
def search(request):
    images, favourite_list = getAllImagesAndFavouriteList(request)
    search_msg = request.POST.get('query', '')

    # si el usuario no ingresó texto alguno, debe refrescar la página;
caso contrario, debe filtrar aquellas imágenes que posean el texto de
búsqueda.
    if search_msg !='':
        images = services_nasa_image_gallery.getAllImages(search_msg)
    return render(request, 'home.html', {'images': images,
'favourite_list': favourite_list} )
```

Esto hace que si el usuario busca una palabra, la página busque las tarjetas de imagenes utilizando la palabra utilizada por el usuario como parametro.

Por ultimo, dentro de la ruta "nasa_image_gallery>layers>services>services_nasa_image_gallery" implemetamos el siguiente ciclo dentro de la funcion "getAllImages":

for object in json_collection:

nasa_card=mapper.fromRequestIntoNASACard(object)

images.append(nasa_card)

```
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 = []
```

```
    # recorre el listado de objetos JSON, lo transforma en una NASACard y
lo agrega en el listado de images. Ayuda: ver mapper.py.
    json_collection = transport.getAllImages(input)

    for object in json_collection:
        nasa_card=mapper.fromRequestIntoNASACard(object)
        images.append(nasa_card)

    return images
```

Donde se busca y ordena las imagenes por defecto dentro de la lista json_collection y lo transforma en una NASACard, que luego es agregada a la lista de "images".