

Informe Trabajo Práctico IP

Integrantes:

Brener, Matías

Herrera, Maximiliano

Villafañe, Hugo

Comisión 05

Se trabaja en el desarrollo de una de una página web donde muestra distintas imágenes de la nasa. El código a emplear está incompleto por lo que hay que terminar el desarrollo.

Realizamos lo siguiente:

En nasa_image_gallery/views.py:

```
def home(request):  
    images = []  
    favourite_list = []  
    images = services_nasa_image_gallery.getAllImages()  
    favourite_list = getAllImagesAndFavouriteList(request)  
    return render(request, 'home.html', {'images': images, 'favourite_list': favourite_list})  
)
```

Dentro de la función home, en la variable images guardamos los resultados obtenidos al invocar a la función getAllImages() desde services_nasa_image_gallery.py

En la variable favourite_list se guarda el listado de imágenes al llamar a la función getAllImagesAndFavouriteList() con el argumento request.

En services/services_nasa_image_gallery.py:

Completamos la función getAllImages():

En la variable `json_collection` llamamos a la función `getAllImages()` desde `transport.py` y mediante un condicional, hacemos que muestre un listado de acuerdo al argumento.

Luego hacemos un recorrido de los objetos `json` y agregamos el listado en la variable `images`.

En la variable `nasa_card` se guardan las imágenes transformadas en cards con la función `fromRequestIntoNASACard()` desde `mapper.py`.

```
def getAllImages(input=None):
    json_collection = []
    images = []
    if input:
        json_collection = transport.getAllImages(input)
    else:
        json_collection = transport.getAllImages(None)

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

    return images
```

De esta manera, al clicar en el link, se muestra el listado de imágenes de la Api.

Buscador:

Para hacerlo funcionar, completamos la función `search()` en `views.py` que de acuerdo al request, llama a la función `getAllImages()` y devuelve un listado de imágenes relacionadas al tema buscado. En este caso le agregamos un condicional para que cuando el usuario no ingrese ninguna palabra en el buscador, este devuelva por defecto imágenes relacionadas al tema “Espacio”

```
def search(request):
    images, favourite_list = getAllImagesAndFavouriteList(request)
    search_msg = request.POST.get('query', '')
    if not search_msg:
        search_msg="space"
    else:
        images=services_nasa_image_gallery.getAllImages(search_msg) #

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

Login:

En templates/header.html modificamos los vínculos correspondientes al inicio/cierre de sesión para que nos permita acceder a login.html.

```
{% if request.user.is_authenticated %}
    <a class="nav-link" href="{% url 'exit' %}">Salir</a> {% else %}
    <a class="nav-link" href="{% url 'login' %}">Iniciar sesión</a>
```

En main/urls.py/ agregamos la ruta path('accounts/',include('django.contrib.auth.urls')),

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('nasa_image_gallery.urls')),
    path('accounts/',include('django.contrib.auth.urls')),
]
```