Informe TP Introducción a la Programación

Las funcionalidades que hemos desarrollado durante el trabajo incluyen la visualización de imágenes en la galería de imágenes, iniciando con las relacionadas al espacio en caso de que no se hiciera una búsqueda puntual. Las ilustraciones son presentadas en forma de Cards, mostrando el título y la descripción de cada una de ellas.

Para lograr esto, llamamos a la función getAllImages() desde el archivo *transport.py*, obteniendo todas las imágenes y almacenándolas en una lista llamada json_collection. Luego, iteramos sobre cada imagen en la lista utilizando un ciclo for y enviamos cada una a la función fromRequestIntoNASACard() del archivo *mapper.py*. Esta función convierte cada imagen en una Card, que almacenamos en una lista llamada images utilizando la variable auxiliar nasa_card. Este enfoque se utilizó para mantener el código más organizado y legible.

El código que hemos implementado en la vista *services_nasa_image_gallery.py* para esta característica es el siguiente:

```
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
```

En la vista *views.py*, desde la función home() guardamos en la variable favourite_list el listado de imágenes cuando llamamos a la función getAllImagesAndFavouriteList() con el argumento request, y obtenemos dos listas, una de imágenes de la NASA y otra con los favoritos.

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

En la función getAllImagesAndFavouriteList() llamamos a la función getAllImages() de la vista *services_nasa_image_gallery.py* que guardamos la información recibida en la variable images:

```
def getAllImagesAndFavouriteList(request):
    images = []
    favourite_list = []
    images = services_nasa_image_gallery.getAllImages()
    return images, favourite_list
```

De esta manera obtenemos todas las imágenes en la galería con el título y la descripción en forma de cards.

Buscador:

Otra funcionalidad que hemos añadido al código de la vista *views.py* es la búsqueda de imágenes dentro de la galería, permitiendo obtener imágenes específicas según los criterios de búsqueda ingresados. Para implementar esta característica, hemos utilizado el siguiente código en la función search():

```
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:

Esta funcionalidad permite a los usuarios autenticarse, proporcionando sus credenciales (usuario y contraseña) para acceder a su cuenta personal. Una vez autenticados, los usuarios pueden añadir imágenes a favoritos acceder a la lista de sus imágenes guardadas.

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')),]
```

Favoritos:

La funcionalidad de "Favoritos" permite a los usuarios crear una lista personalizada de sus imágenes preferidas. Los usuarios pueden añadir imágenes a esta lista mientras navegan por la galería, guardando aquellas que les resulten más interesantes o significativas. Esta lista de favoritos es accesible en cualquier momento, siempre que el usuario esté logueado.

Las siguientes funciones se utilizan para implementar la sección de favoritos: traer los favoritos de un usuario, guardarlos, eliminarlos y desloguearse de la app.

La función getAllFavouritesByUser() trae las imágenes guardadas como "favorito", se utiliza el modelo favourite con la accion "filter" basado en el dato "user" y retorna la lista de favoritos utilizada en el template:

```
def getAllFavouritesByUser(request):
    favourite_list = Favourite.objects.filter(user=request.user)
    return render(request, 'favourites.html', {'favourite_list':
favourite_list})
```

```
def saveFavourite(request):
    if request.method == 'POST':
        saved_favourite =
services_nasa_image_gallery.save_Favourite(request)
        if saved_favourite:
            return redirect('favoritos')  # Redirige a la vista de favoritos
si el favorito se guarda correctamente
        else:
            return redirect('home')  # Redirige a la vista principal si hay
un error
```

Eliminar:

Función para eliminar favoritos:

```
def deleteFavourite(request):
    if request.method == 'POST':
        success = services_nasa_image_gallery.deleteFavourite(request)
        return redirect('favoritos')
    return redirect('favoritos')
```

Las dificultades que enfrentamos como equipo se centraron en el desarrollo del trabajo en general. Al principio, nos resultó complicado entender la lógica del proyecto, ya que no teníamos experiencia previa en un proyecto de esta magnitud. Aunque nuestra tarea en el trabajo práctico era desarrollar funciones, variables, ciclos, etc., nuestro desconocimiento del entorno Django era total y esto nos llevó a invertir muchas horas en tutoriales para comprender su funcionamiento. Posteriormente, en el desarrollo específico, lo que más nos costó fueron los puntos adicionales, especialmente la implementación de la lista de favoritos. Para lograrlo, dedicamos muchas horas a ver videos, enfrentamos frustraciones e incluso enojos, pero finalmente, con esfuerzo colectivo, logramos completar la tarea con éxito.

Comisión: 05

Alumnos:

- Brener Matias
- Herrera Maximiliano
- Villafañe Hugo