

# Universidad Nacional de General Sarmiento



## **Introducción a la programación**

### **Trabajo práctico grupal**

**Estudiantes:** Sorroche Agustina y Greco Lucas

**Comisión:** 03

**Docentes:** Daniel Bressky, Esteban Fassio y Santiago Velázquez

**Año:** 2024

**Carrera:** Lic. Sistemas

**Nota:**

## Introducción

El presente informe detalla el desarrollo de un proyecto práctico. El proyecto se centra en la creación de una galería de imágenes utilizando el framework Django, diseñada para mostrar imágenes provenientes de la API de la NASA y hacer un login de usuario. Este informe proporciona una visión general de las funcionalidades implementadas, las dificultades que se tuvieron durante el proceso de desarrollo.

## Views.py

La función `getAllImagesAndFavouriteList(request)` es una función auxiliar que obtiene una lista de imágenes de la API y la lista de los favoritos del usuario y estos datos se devuelven en una tupla para luego ser utilizada en la función `home(request)`.

```
# auxiliar: retorna 2 listados -> uno de las imágenes de la API y otro de los favoritos del usuario.
def getAllImagesAndFavouriteList(request):
    favourite_list = get_Favourites_User(request)
    images = getAllImagesfromTransport()

    return images, favourite_list
```

La función `home(request)` actúa como el controlador principal para la página de inicio de la galería. Utiliza una función auxiliar para obtener listas de imágenes y favoritos, luego renderiza el contenido de 'home.html' pasando estos datos como contexto. Esto permite que 'home.html' muestre las imágenes recuperadas y los favoritos del usuario

```
# función principal de la galería.
def home(request):
    # llama a la función auxiliar getAllImagesAndFavouriteList() y obtiene 2 listados: uno de las imágenes de la API y
    images, favourite_list = getAllImagesAndFavouriteList(request)
    return render(request, 'home.html', {'images': images, 'favourite_list': favourite_list})
```

la función `search(request)` obtiene un mensaje de búsqueda ingresado por el usuario, si encuentra un mensaje de búsqueda válido, realiza la búsqueda de imágenes correspondientes para luego renderizar `home.html` con los resultados de la misma. Si no encuentra ningún mensaje de búsqueda refresca la página.

```
# función utilizada en el buscador.
def search(request):
    images = []
    search_msg = request.POST.get('query', '')
    if search_msg != "":
        images = getImagesBySearchInputLike(search_msg)
        return render(request, 'home.html', {'images': images })
    # si el usuario no ingresó texto alguno, debe refrescar la página
    return redirect(home)
```

La función `get_user_favourites(request)` obtiene la lista de favoritos del usuario llamando a la función que se encuentra en `services_nasa_image_gallery.py` y renderiza el contenido de `favourites.html` con el contenido de la lista de favoritos.

```
@login_required
def get_user_favourites(request):
    favourite_list = get_Favourites_User(request)
    return render(request, 'favourites.html', {'favourite_list': favourite_list})
```

La función `user_login(request)` obtiene los datos del usuario y contraseña del formulario de inicio de sesión, luego verifica las credenciales del usuario, si las credenciales son correctas y el usuario existe en la base de datos la función inicia sesión y redirige al usuario a la página de inicio, si no existe el usuario refresca la página para que el usuario vuelva a intentar

```
def user_login(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')
        user = authenticate(request, username = username, password = password)
        if user is not None:
            login(request, user)
            # Redirige a la página de inicio o a alguna otra página después del inicio de sesión exitoso
            return redirect(index_page)

    return render(request, 'registration/login.html')
```

La función `exit(request)` se encarga de cerrar la sesión del usuario autenticado, para eso llama a la función `logout(request)` para eliminar la sesión actual del usuario y luego lo redirige a la pagina de inicio

```
@login_required
def exit(request):
    logout(request)
    return redirect(index_page)
```

## Services\_nasa\_image\_gallery.py

La función `getALLImagesfromTransport(input=None)` obtiene las imágenes de la API y los guarda en `json_collection`, luego se recorre cada imagen para transformarla en una NASA Card y lo agrega al listado de imágenes. La función devuelve el listado de imágenes transformadas en NASA Card

```
def getALLImagesfromTransport(input=None):
    # obtiene un listado de imágenes desde transport.py y lo guarda en un json_collection.
    images = []
    json_collection = getAllImages(input)
    for obj in json_collection:
        nasa_card = fromRequestIntoNASACard(obj)
        images.append(nasa_card)
    # recorre el listado de objetos JSON, lo transforma en una NASACard y lo agrega en el listado de images.
    return images
```

La función `getImagesBySearchInputLike(input)` es la función que utiliza el buscador para conseguir las imágenes a partir del input del usuario y luego transformarlas en NASA Card para después mostrarlas

```
def getImagesBySearchInputLike(input):
    json_collection = getAllImages(input)
    images = []
    for obj in json_collection:
        nasa_card = fromRequestIntoNASACard(obj)
        images.append(nasa_card)
    return images
```

La función `get_Favourites_User(request)` verifica si el usuario esta autenticado, si no esta

```
# usados en el template 'favourites.html'
def get_Favourites_User(request):
    if not request.user.is_authenticated:
        return []
    else:
        user = get_user(request)
        favourite_list = GetAllFavouritesByUser(user) # buscamos desde el repositorio TODOS los favoritos del usuario
        mapped_favourites = []

        for favourite in favourite_list:
            nasa_card = fromRepositoryIntoNASACard(favourite) # transformamos cada favorito en una NASACard
            mapped_favourites.append(nasa_card) # lo almacenamos en nasa_card.

        return mapped_favourites
```

autenticado devuelve una lista de favoritos vacía, si el usuario si esta identificado obtiene todos los favoritos del mismo, luego convierte cada favorito en una NASA Card para luego devolver una lista de NASA Cards que representan los favoritos del usuario y que se muestren cuando se ejecuta la función de favoritos en el archivo views.

Luego de estas funciones se hicieron pequeños ajustes a los archivos `urls.py` y `header.html`, para poder desarrollar el inicio de sesión, ya que en `urls.py` se llamaba a la función `index(request)` en lugar de llamar a la función `user_login(request)` y en el `header.html` los botones de inicio de sesión y salir no redirigían a ningún lado

En el presente trabajo se desarrollaron los condicionales “barra de búsqueda” y “inicio de sesión”, se desarrollo una parte del condicional de favoritos, pero no se pudo llevar a cabo, se pueden mostrar los favoritos del usuario, pero no se pueden agregar o borrar.