

**INFORME TRABAJO GRUPAL**  
**GALERÍA DE IMÁGENES DE LA NASA**  
**INTRODUCCIÓN A LA PROGRAMACIÓN**  
**COM-6 | Tarde**

**Alex Ortiz – Delfina Gómez – Malena Balbi**

Tenemos un proyecto de página web, y nuestro trabajo era que se puedan renderizar las imágenes en la pantalla, con sus correspondientes títulos y descripciones. Para ello, había que arreglar las funciones que incluían el renderizado de estas cosas: “*getAllImages*”, “*getAllImagesAndFavouriteList*”, “*Home*”.

Nuestra principal problemática fue entender la lógica del código. Veíamos las funciones a arreglar y no entendíamos qué había que hacer en un principio. Luego nos dimos cuenta de que, gracias a las importaciones de otros archivos, como “*mapper.py*” o “*transport.py*”, estas ya tenían incluidas otras funciones para poder arreglar las 3 funciones principales de nuestro problema. Así que no había que hacer mucho, sino llamar a esas funciones para hacer todo el proceso más simple.

**“def getAllImages (input=None)”**

```
def getAllImages(input=None):  
    # Se importa el listado de imágenes desde transport.py y lo guarda en un json_collection.  
    # json_collection es un diccionario. El parametro 'input' indica si se debe buscar por un valor introducido en el b  
    json_collection=transport.getAllImages(input=None) ## Se importa de "Transport.py" la  
  
    images = []  
    for elem in json_collection: ##recorremos cada elemento de la lista  
        nasa_card=mapper.fromRequestIntoNASACard(elem) ##Convertimos cada elemento e  
        images.append(nasa_card) ##Agregamos esa nasa card a la lista de imagenes.  
  
    # recorre el listado de objetos JSON, lo transforma en una NASACard y lo agrega en el  
  
    return images
```

Para la función “*getAllImages*” del archivo “*services\_nasa\_image\_gallery.py*” importamos diferentes funciones, como el servicio de obtener imágenes desde la REST API de la NASA, y transformarlos en una “Nasa Card”, que incluye la imagen, su título y su descripción para luego usarlos en las siguientes funciones. Luego, estas fueron importadas en el archivo “*views.py*” (además de la función que

recolecta la lista de favoritos del usuario) para ser recolectadas por la función “*getAllImagesAndFavouriteList*”.

“**def getAllImagesAndFavouriteList (request, x)”**

```
def getAllImagesAndFavouriteList(request,x):  
  
    images=services_nasa_image_gallery.getAllImages(input=None)  ##Importamos imágenes  
  
    favourite_list=services_nasa_image_gallery.getAllFavouritesByUser(request)  ##Im  
    ##Ya que necesitaremos usar ambas listas en diferentes ocaciones, añadimos un se  
    if x==1:  
        return images  
    elif x==0:  
        return favourite_list
```

La función “*getAllImagesAndFavouriteList*” toma las dos listas, imágenes y la lista de favoritos del usuario importadas de “*services\_nasa\_image\_gallery.py*” para poder usarlas en “*views.py*” dónde podrán ser renderizadas. Esta función retorna dos listas diferentes: lista de imágenes y lista de favoritos del usuario. Para poder darle la opción al programa de utilizar una lista o la otra, le agregamos un parámetro más a la función (x) que será utilizado en la siguiente función.

“**def Home(request)”**

```
def home(request):  
    # llama a la función auxiliar getAllImagesAndFavouriteList() y obtiene 2 listados: uno de las  
    # (*) este último, solo si se desarrolló el opcional de favoritos; caso contrario, será un li  
    images = []  
    favourite_list = []  
    #Llamamos a la función anterior, y establecemos el segundo parámetro como "1", para tomar los  
    for elem in getAllImagesAndFavouriteList(request,1):  
        images.append(elem)  
    #Llamamos a la función anterior, y establecemos el segundo parámetro como "0", para tomar los  
    for elem in getAllImagesAndFavouriteList(request,0):  
        favourite_list.append(elem)  
  
    return render(request, 'home.html', {'images': images, 'favourite_list': favourite_list} )
```

Y al final, en la función “*Home*” toma estas dos listas, y dependiendo de qué valor tome (x), llamará a la función anterior, y retornará la lista correspondiente, y luego la propia función “*Home*” retornará el render.

En conclusión, así fue la lógica que llevamos durante el proyecto: llamar a otras funciones ya hechas de otros archivos y utilizarlos para poder renderizar las imágenes de la galería de la NASA.