

Informe del Trabajo Práctico

Martin Godoy y Carolina Allende

Introducción

El Trabajo Práctico de la materia Introducción a la Programación consiste en implementar distintas funcionalidades para el correcto desarrollo de una aplicación web utilizando Django Framework. La página inicialmente no presenta un código que le permita un correcto funcionamiento, por lo que el trabajo consistirá en completar el mismo.

Se entiende a que la página funcione correctamente, como mínimo, que muestra distintas imágenes provistas de una API que da la NASA, además se pueden agregar distintas funcionalidades opcionales que permiten un funcionamiento aún mejor.

Desarrollo

Función: “getAllImagesAndFavouriteList()”

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

Se llamó desde el `services_nasa_image_gallery.py` a la función `getAllImages()` para obtener todas las imágenes que proporciona la NASA y retornarlas, al igual que la de los favoritos, que pide las imágenes que el usuario guardó, este último no está del todo desarrollado pero fue necesario su implementación para que funcione la página principal y el `return` tenga sentido.

La función devuelve dos valores, pero `favourite_list` no será utilizado.

Esta función nos causó problemas ya que nosotros eliminaremos el `favourite_list` al no utilizarlo, pero esto generaba un error a la hora de mostrar la página, por lo que decidimos dejarlo al concluir que debe ser útil en algún retorno que no hemos trabajado de los puntos opcionales.

Función: “home”

```
def home(request):
    images = []

    images, favourite_list = getAllImagesAndFavouriteList(request)

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

Esta función llama a “getAllImagesAndFavouriteList(request)” que obtiene las imágenes de la API y la de los favoritos, solo se podrá ver las de la página principal ya que no desarrollamos la de favorito, luego renderiza las imágenes en el archivo home.

Se nos pide invocar una función y que guarde los resultados, por lo que no hubo mucho motivo de discusión a la hora de abordar esta función en base a lo directa que es la consigna.

Función: “getAllImages()”

```
def getAllImages(input=None):
    json_collection = []

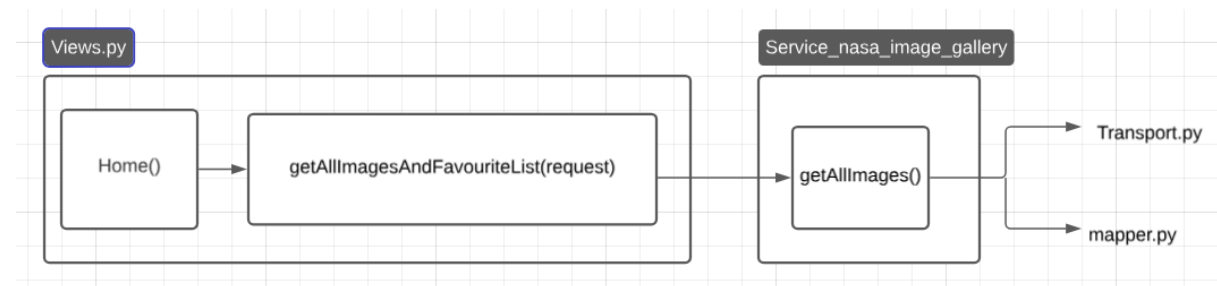
    for items in transport.getAllImages(input):
        json_collection.append(items)

    images = []
    for items in json_collection:
        images.append(mapper.fromRequestIntoNASACard(items))
    return images
```

A la función “getAllImages” se le llama para obtener desde transport.py todas las imágenes de la API y guardarlas en una variable, en este caso json_collection. Luego se la recorre, las transforma en una nasa card desde mapper.py para que tomen el formato necesario, con la

función `fromRequestIntoNASACard(items)` y las guarda en otra variable que es la que se retorna.

Esta fue una de las funciones que más nos costó implementar a pesar de entender la lógica de lo que debíamos hacer, ya que para su funcionamiento teníamos que buscar de donde venían los datos que necesitaba y agregarlos a una lista. Para ordenar los datos nos valimos de un diagrama, ya que no logramos hacerlo en el aire.



Función: “search()”

```

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

    if not search_msg:
        return redirect('home')
    else:
        filtrar_images = services_nasa_image_gallery.getAllImages(search_msg)
        return render(request, 'home.html', {'images': filtrar_images, 'favourite_list':
favourite_list})
  
```

A esta función se le agregó un condicional para filtrar la búsqueda, si el usuario no escribe nada, redirecciona a la página principal, caso contrario filtró la palabra llamando a la función “getAllImágenesAndFavouriteList” para obtener la lista de imágenes que proporciona la API de la NASA y retorna basándose en un término de búsqueda ingresado por el usuario, renderizar la página ‘home.html’, con las imágenes filtradas y la lista de favoritos.

Para esta función no fue tanta la dificultad a comparación de las otras, ya que teníamos más noción en general, seguimos las instrucciones (los comentarios) agregados en el trabajo, e implementamos varias opciones hasta obtener una que funcione correctamente.

Para las demás funciones opcionales, quisimos realizar la de inicio de sesión, agregamos varias modificaciones y con ninguna se logró que funcione, pero no lo logramos a pesar de ver el tutorial de autenticación.