Informe del grupo 2
Trabajo Práctico Grupal
Introducción a la programación

Gino Ciancia / Ramiro Paz / Agostina Ozorio / Isaac Vazquez

Comisión 02
Profesoras: Bottino Flavia y Winograd Natalia
Primer Cuatrimestre
Año de cursada: 2024

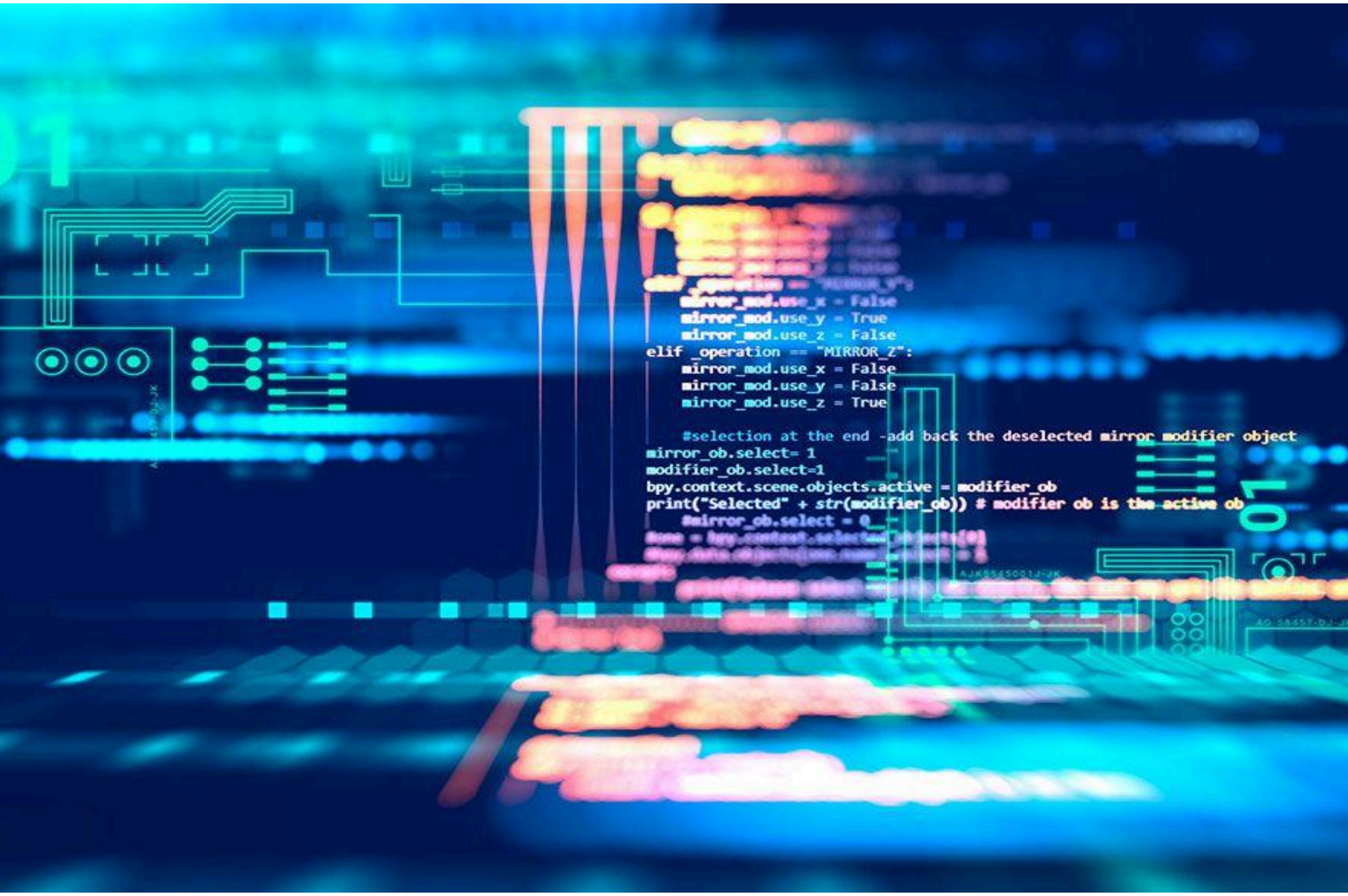

Tabla de contenido:

Introducción

En el presente informe pasaremos a desarrollar sobre las dificultades y el proceso de realizar el trabajo final. Debíamos implementar una aplicación web fullstack usando Django Framework que nos dejaría llamar a las imágenes de la API pública de proporciona la NASAi. Lo que debíamos hacer eran completar 3 funciones, para que se puedan visualizar las imágenes llamadas NasaCard, en una galería de imágenes de una página de la Nasa. Al definir las funciones nos debíamos asegurar que estas NasaCard aparecieran en el siguiente orden: la imagen, su título y su descripción.
Las 3 funciones se encontraban distribuidas de la siguiente manera, 2 estaban en la carpeta view.py, y la otra estaba en la carpeta service_nasa_image_gallery.py, estaba dentro de la capa llamada “sevices”.
La página al finalizar nos quedaría como vemos en la siguiente imágen.

Funciones

Empezamos por modificar la función def getallimages en service_nasa_image_gallery.py, para que presente el listado con las imágenes.

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

Completamos json_collection para así obtener un listado de imágenes utilizando transport.getAllImages(input). Luego, mapea los resultados a objetos NasaCard usando mapper.fromRequestIntoNASACard(object) y devuelve la lista de imágenes. La función nos quedaría de la siguiente manera:

```
def getAllImages(input=None):
    # obtiene un listado de imágenes desde transport.py y lo guarda en un json_collection.
    # el parámetro 'input' indica si se debe buscar por un valor introducido en el buscador.
    json_collection = transport.getAllImages(input)
#getAllimages = todas las imágenes con lo dispuesto en el parámetro.
    images = []
    for object in json_collection:
        nasaCard = mapper.fromRequestIntoNASACard(object)
        images.append(nasaCard)
    # recorre el listado de objetos JSON, lo transforma en una NASACard y lo agrega en el listado de images. Ayuda: ver mapper.py.
    return images
```

Seguimos con las 2 funciones de la carpeta View.py, primero con la función def getAllImagesAndFavouriteList(request):

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

Esta función obtiene todas las imágenes llamando a getAllFavouritesByUser.
Si el usuario está autenticado, también obtiene la lista de sus favoritos desde la base de datos, filtrando por el mismo, pero la dejamos vacía ya que no configuramos los usuarios. Devuelve una lista de las imágenes.
En la siguiente imagen demuestra el como quedaría:

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

    return images, favourite_list
```

Luego procedimos con la función def Home, la función home se encarga de obtener la lista de imágenes y favoritos, y renderizar la página principal ('home.html') con esta información.

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

En esta función debemos agregar el getAllImagesAndFavouriteList(request) y utilizar el render para enviar las listas images y favourite_list al template home.html, que luego se encarga de mostrar esta información en la interfaz de usuario.

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

Además modificamos la def Search, la cual debía ser completada para manejar adecuadamente las búsquedas realizadas por el usuario y renderizar los resultados correspondientes en home.html.
Se utiliza el request.POST.get('query','') para obtener el texto de búsqueda ingresado por el usuario. Si search_msg está vacío, nos lleva a la página principal home utilizando redirect('home'), sino, procede a llamar a services.getImagesBySearchInputLike(search_msg) para obtener las imágenes que coinciden con el texto de búsqueda. Por último utiliza render para enviar las listas images y favourite_list al template home.html, que luego se encarga de mostrar los resultados de la búsqueda en la interfaz de usuario.

```
def search(request):
    search_msg = request.POST.get('query', '')
    images =
    favourite_list = []
    return render(request, 'home.html', {'images': images,
'favourite_list': favourite_list} )
```

Con los cambios quedo de la siguiente forma:

```
def search(request):
    search_msg = request.POST.get('query', '')
    if not search_msg:
        return redirect('home') #si no hay texto de búsqueda,vuelve
a home
    images = services.getImagesBySearchInputLike(search_msg)
    favourite_list = []
    # si el usuario no ingresó texto alguno, debe refrescar la
página; caso contrario, debe filtrar aquellas imágenes que posean
el texto de búsqueda.
    return render(request, 'home.html', {'images': images,
'favourite_list': favourite_list} )
```

Opcionales:
Para agregar decidimos modificar el aspecto de la página respetando las condiciones de presentación de imagen con título y descripción.
Inicio: Se agregó un background con un linear gradient. Para mejorar el aspecto de visualización.
Secciones de “Cards”: Se modificó el estilo de las “Cards” y se agregó una animación de la pseudoclase “hover”. Y, el template (solo el background).
Para esto, se utilizaron algunas clases en los archivos HTML disponibles. Y se agregaron estilos en el archivo “../styles.css”.

Dificultades

Desafíos y decisiones (dificultades encontradas y decisiones técnicas):

Uno de los primeros desafíos fue acercarnos al formato del trabajo, que a simple vista era distinto a lo trabajado en clases programando solo desde python. Además estaba el factor de que al ser un nuevo trabajo práctico, todo se entornaba a ser más experimental y eso jugaba en contra por el nerviosismo y preocupaciones de cada uno de los integrantes del grupo. También cabe resaltar que algunos de nosotros lamentablemente no contábamos con las herramientas, tales como computadora con la capacidad para soportar los programas a instalar, necesarias para proseguir por su cuenta. Pero para solucionar estas problemáticas, realizamos con el grupo reuniones virtuales para que en el ida y vuelta del trabajo o en otros casos, nos reunimos en una casa para trabajar en equipo, con el fin de completar/modificar las funciones. Y con la ayuda de cada miembro, despejamos las dudas que surgían a medida que avanzábamos, resultando más dinámico el desarrollo del mismo y su comprensión.
Durante el desarrollo de las funciones, una de las dificultades fue asegurarnos de cumplir con el trabajo principal de que todos los objetos obtenidos fueran mapeados correctamente a objetos NasaCard utilizando los comandos ya dichos anteriormente. Para finalmente lograr que aparecieran en el orden indicado de presentación: imagen con el título y descripción debajo. También cabe recalcar que fue complicado corregir cada “error” de los códigos, lo cual tomó tiempo y dedicación en encontrar lo que faltaba para poder arreglarlo y hacer que funcione correctamente.

Conclusión

Fue un trabajo que requería que nos informaramos de temas relativamente nuevos de los ya aprendidos en clase. Los cuales nos ayudarán y utilizaremos a lo largo de la carrera, tales como visualcode, github, django, entre otros. Aunque parecía complicado al principio ya que era todo relativamente nuevo y rápido, fue un buen acercamiento a lo que nos espera si en el futuro queremos desempeñarnos en el ámbito de la programación. Es importante esforzarse por practicar para poder dominar estas herramientas, e intentar seguir descubriendo más contenidos para así poder lograr un mejor desempeño en nuestra carrera como programadores.Y no menos importante, este trabajo nos enseñó la importancia y utilidad del trabajo en equipo, lo cual nos podrá facilitar en nuestro rendimiento laboral.