

Trabajo practico final: Introducción a la programación

Alumnos: Tomás Alejandro Ledesma, Juan Cruz Rumpel

Comisión: 04

Profesores: Omar Argañaras, Benesch Alicia

Índice

Introducción	2
Desarrollo	2
Conclusión	4

Introducción

En este trabajo, se busca desarrollar una aplicación la cual se dedicará a mostrar imágenes de la Nasa con su información pertinente (título y descripción), a través de distintas tarjetas, las cuales van a poder ser filtradas por un buscador principal.

Para esto, se utilizará tecnologías tales como:

- Django framework
- Python
- HTML
- CSS
- Bootstrap

Gran parte del trabajo ya se encontraba desarrollado, pero en este informe, detallaremos todo lo que hemos desarrollado, como funcionan, y las dificultades que nos hemos encontrado.

Desarrollo

Problema principal:

La primera parte del proyecto consistía en desarrollar la lógica para que, al entrar en Galería o “este link”, muestre todas las imágenes con la búsqueda por defecto del sistema, que es la palabra “space”.



Lo que faltaba desarrollar para eso, era la función que traía la información de la Nasa, para así poder acomodarla, y convertirla en una tarjeta, para así la página web podía mostrar toda esa información por pantalla. Por eso desarrollamos las siguientes funciones:

- getAllImages:

```
def getAllImages(input=None):
    # obtiene un listado de imágenes desde transport.py y lo guarda en un json_collection.
    # ¡OJO! el parámetro 'input' indica si se debe buscar por un valor introducido en el buscador.
    json_collection = []
    json_collection = transport.getAllImages()

    images = []
    for objeto in json_collection:
        images.append(mapper.fromRequestIntoNASACard(objeto))
    # recorre el listado de objetos JSON, lo transforma en una NASACard y lo agrega en el listado de images.

    return images
```

La cual recibe de una función ya creada, una lista con la información de la Nasa relacionado a la palabra predeterminada, para luego acomodar cada elemento a través de otra función, para que todo esta pueda ser mostrado en distintas tarjetas. Al final, esta función devuelve toda esta información acomodada.

- getAllImagesAndFavouriteList(request):

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

    return images, favourite_list
```

Una función la cual recibe esta información anteriormente acomodada, y la devuelve, además de un listado vacío de imágenes favoritas, ya que no se encuentra desarrollado.

- home(request):

```
# 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 otro
    # (*) este último, solo si se desarrolló el opcional de favoritos; caso contrario, será un listado vacío [].
    images = []
    favourite_list = []
    images, favourite_list = getAllImagesAndFavouriteList(None)
    return render(request, 'home.html', {'images': images, 'favourite_list': favourite_list})
```

Por último, esta función es la encargada de recibir esta información arreglada, y dársela a la pagina web, así las muestra en forma de tarjetas.

Buscador

Uno de los agregados del programa, es la implementación de un buscador, el cual filtra la información en base a la palabra ingresada. Para eso, utilizamos las siguientes funciones:

- `getImagesBySearchInputLike(input)`

```
#filtra las imagenes de la API en base a lo que llega por parametro y mapea c/u de ellas como una NasaCard
def getImagesBySearchInputLike(input):
    json_collection = transport.getAllImages(input)
    images = []
    for objeto in json_collection:
        images.append(mapper.fromRequestIntoNASACard(objeto))
    return images
```

En este caso, la función recibe algo que ingrese el usuario, y busca información de la Nasa en base a eso, para después repetir el proceso de acomodar la información en una tarjeta y devolverla.

- `Search(request):`

```
# función utilizada en el buscador.
def search(request):
    images, favourite_list = getAllImagesAndFavouriteList(request)
    search_msg = request.POST.get('query', '')
    # si el usuario no ingresó texto alguno, debe refrescar la página; caso contrario, debe filtrar aquellas imágenes
    if search_msg != '':
        images_filtered = services_nasa_image_gallery.getImagesBySearchInputLike(search_msg)
        return render(request, 'home.html', {'images': images_filtered, 'favourite_list': favourite_list})
    else:
        return redirect('home')
    pass
```

Esta función toma lo que ingresa el usuario, y si ingreso algo, lo busca con la función hecha previamente, y muestra las tarjetas que corresponden, pero, sino ingresa nada, actualiza la página y vuelve a cargar el contenido principal.

Conclusión

Tanto la problemática principal, como la implementación del buscador, se lograron exitosamente, ambas se lograron llevar a cabo sin demasiada ayuda. Para la próxima actualización, esperamos implementar el login, además de darle una nueva estética a la pagina así la sentimos más personal.