

TRABAJO PRÁCTICO INTEGRADOR

Comisión: COM-04

Docentes: Omar Argañaras, Benesch Alicia

Integrantes:

Sandoval Celeste (DNI 45430026)

Índice:

introducción y funciones implementadas.....	3,4,5
--	--------------

Introducción

El trabajo consiste en implementar una aplicación web fullstack usando Django FRamework. Esta aplicación consta en un buscador central que permite consultar las imagenes de la API publica que proporciona la NASA. La informacion que provenga de esta API será renderizada por el framework en distintas card que muestras, la imagen en cuestión el título y la descripción.

Funciones implementadas:

def getAllImages

(input=None): **# obtiene un listado de imágenes desde transport.py y lo guarda en un json_collection.**

images = []

json_collection = transport.getAllImages(input)

if input == None or input == "":

 for image in json_collection:

 card_nasa=mapper.fromRequestIntoNASACard(image) **#transformó a nasa CARD**

 images.append(card_nasa)

 else:

recorre el listado de objetos JSON, lo transforma en una NASA Card y lo agrega en el listado de images.

for imagen in json_collection:

 nasaCard=mapper.fromRequestIntoNASACard(imagen)

 if str(nasaCard.title).lower().find(input.lower()) != -1: **#si el título es igual al input buscado lo agrego a las imagenes**

 images.append(nasaCard)

return images

La función **getAllimages** lo que hace es primero es obtener las imágenes de la API de la nasa y devuelve una lista de las imágenes

dificultades:

poder obtener las imágenes con la información pertinente, para lograr esto se utilizó una función ya implementada mapper que permite dar formato y obtener las imágenes con la información necesaria.

```
def getImagesBySearchInputLike(input): #tratar errores del buscador  
if input == "":  
    input="Space"  
if input.lower() == "marte":  
    input="mars"  
return getAllImages(input)
```

La función **getImagesBySearchInputLike** Solo tratamos errores posibles del buscador

dificultades:

poder mostrar una imagen por defecto en caso de que el dato buscado esté vacío, para lograrlo se verifica que el dato exista y luego se reutiliza la función `getAllImages`.

```
def getAllImagesAndFavouriteList(request): #obtener todas las imagenes  
cuando inicia la pag por defecto  
images = services_nasa_image_gallery.getAllImages(None) favourite_list = []  
return images, favourite_list
```

La función **getAllImagesAndFavouriteList(request)** obtiene la lista de las imágenes

```
def home(request): #función principal de la galeria  
Se llama a la función auxiliar getAllImagesAndFavouriteList() y obtiene 2  
listados: uno de las imágenes de la API y otro de favoritos por usuario.  
#este último, solo si se desarrolló el opcional de favoritos; caso contrario,  
será un listado vacío.  
images = getAllImagesAndFavouriteList(request)[0] #es la primer lista que te  
devuelve la función getAllImagesAndFavouriteList
```

dificultades:

poder obtener el listado de forma correcta de las imágenes a renderizar.

```
def search(request):
```

```
    favourite_list = []  
    return render(request, 'home.html', {'images': images,  
    'favourite_list': favourite_list}) #muestro todas las nasaCARD redirecciona a  
home, pasa imagen y lista de favs
```

La función **def home(request)** renderiza las cartas

```
def search(request): #buscador  
favourite_list=[]  
search_msg = request.POST.get('query', "")  
imagesSerch=services_nasa_image_gallery.getImagesBySearchInputLike(search_m  
sg) #genera las cartas buscadas por el usuario  
# si el usuario no ingresó texto alguno, debe refrescar la página; caso  
contrario, debe filtrar aquellas imágenes que posean el texto de búsqueda.  
pass  
if search_msg!=" ": #si existe muestro las imagenes buscadas por el usuario,si  
no redirecciono al home  
return render(request, 'home.html', {'images': imagesSerch, 'favourite_list':  
favourite_list,} )  
return redirect(home)
```

La función **def search(request)** es la función para el buscador central, busca la imagen ingresada por el usuario, si existe muestra las imágenes buscadas, en caso contrario redirecciona a home

dificultades:

poder renderizar correctamente el contenido pedido por el usuario, para esto se optó por dos situaciones, la primera que se hiciese un click por error al botón de búsqueda, en dicho caso se renderiza el home por defecto, en caso contrario se busca renderizar el dato solicitado.

