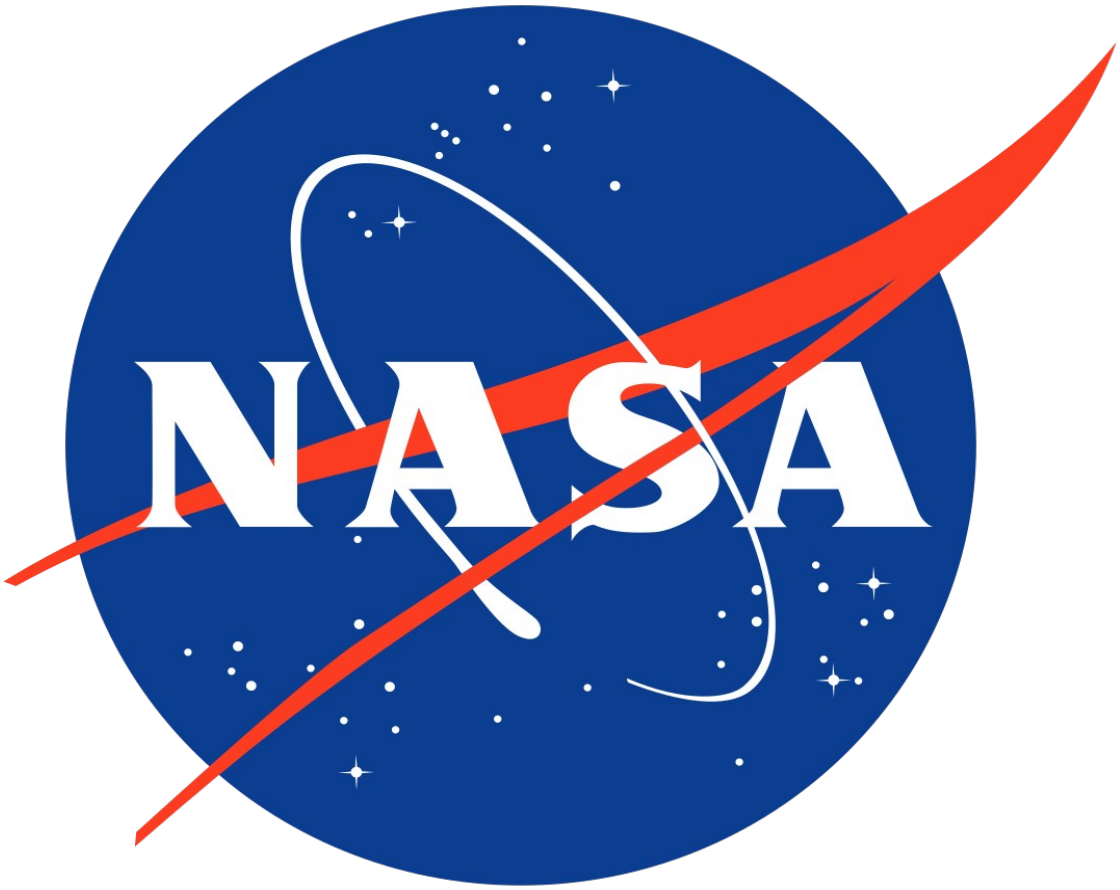
Proyecto
Galería de imágenes de la NASA



Integrantes:
Lucero, Ramiro Gonzalo.
Pereyra, Nicolas Ricardo.
Ferreyra, Gian Franco.
Dias, Quimey.

Este proyecto tiene como objetivo implementar el correcto funcionamiento de una página web, que recibe, procesa y visualiza datos publicados por la NASA. Este sitio web está diseñado para recoger información actualizada de diversas fuentes de la NASA, transformando los datos en algo más fácil de interpretar y ver. Para lograr esto, hemos completado el código de la página, que inicialmente estaba incompleto. Este proceso implicó también la adición de funcionalidades necesarias y la implementación de varias funciones útiles.

Ahora, comenzaremos a enseñar las funciones que modificamos o creamos para luego explicar de forma simple su uso. Y para, a continuación comentar, las dificultades a la hora de pensarlas, escribirlas y cómo sobrepasamos dichas dificultades
.
Para empezar tenemos que explicar brevemente la función que hace la mayoría del trabajo funcione:

La Función “getAllImages”

```
def getAllImages(input=None):
    json_collection = transport.getAllImages()
    images = []
    for objeto in json_collection:
        nuevo_nasacard = mapper.fromRequestIntoNASACard(objeto)
        images.append(nuevo_nasacard)
    return images
```

Obtiene información de otra función y las guarda en una lista con un formato de texto sencillo para el intercambio de datos.
Luego recorre la lista por cada objeto para convertirlas en el formato de cartilla (información, imagen, enlace y título) los cuales agrega a una lista. Al final de la función devuelve la lista de información que ahora están en formato “cartilla”.

Esta fue sin duda la parte más complicada, no teníamos idea de cómo convertir la información de la base de datos de la NASA a cartillas de información legibles e interpretables por las demás funciones, luego de investigar y probar constantemente, logramos hacer que mostrará lo que necesitábamos en la página.

La función “getAllImagesAndFavouriteList”

```
def getAllImagesAndFavouriteList(request):
    images = services_nasa_image_gallery.getAllImages(input=None)
    favourite_list = [services_nasa_image_gallery.getAllFavouritesByUser(request)]

    return images, favourite_list
```

Esta función recibe la lista de información y devuelve las imagenes a renderizar en la página, así como las favoritas del usuario al iniciar sesión.

Se nos complicó el cómo importar funciones e información de otras carpetas y archivos. Por lo que empezamos a revisar tutoriales y blogs.
Luego de consultar a los profesores, logramos comprender y completar la función. Dejando de lado la funcionalidad de imágenes favoritas ya que nuestro principal enfoque era que la pagina cargara las cartillas de información.

La función de la pagina principal

```
def home(request):
    images, favourite_list = getAllImagesAndFavouriteList(request)

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

La siguiente función pide el listado de imágenes en formato cartilla a la anterior función, envía y muestra las cartillas en pantalla al entrar en el apartado galería.

Esta función fue la primera que intentamos hacer, habíamos pasado mucho tiempo configurando y descargando lo necesario para empezar y nos desorientó mucho por la cantidad de carpetas y archivos, los cuales teníamos que investigar uno a uno para averiguar cómo funcionaba el todo. Luego de un descanso de 2 días comenzamos a experimentar con las funciones, creando carpetas de pruebas y cargando imágenes que usamos como borrador. O eso creíamos hasta que con un poco más de investigación nos dimos cuenta que las imágenes vendrían gracias a una función y para que las “home” sirvan, necesitábamos primero hacer las otras dos funciones.

La función del buscador

```
def search(request):

    images, favourite_list = getAllImagesAndFavouriteList(request)
    search_msg = request.POST.get('query', '').lower()

    if search_msg:
        filtered_images = [
            image for image in images
            if search_msg in image.title.lower() or search_msg in
image.description.lower()
        ]
    else:
        search_msg = "moon"
        filtered_images = [
            image for image in images
            if search_msg in image.title.lower() or search_msg in
image.description.lower()
        ]

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

    pass
```

La función de búsqueda se encarga de mostrar en pantalla las cartillas que tenga similitud con lo escrito en el buscador por el usuario, esta misma pide 2 datos, la lista de cartillas y una solicitud que se encargará de filtrarlas en base a palabras claves, tal cual serian “moon” (luna), “sun” (sol), “earth” (tierra), entre otras.
Si no se ingresa ningún dato o palabra a filtrar se buscará y mostrará por defecto los resultados de la búsqueda “moon”

Pensamos muchas ideas para cómo filtrar las cartillas, aunque entendíamos el concepto, el cual era recorrer una lista, verificar y comparar elementos dentro de esta, no entendíamos cómo podríamos implementarlo con el formato que utilizaba. Al completar la función “getallimages”, entendimos que

podíamos que filtrar por título y descripción. Por lo que tras un par de intentos logramos hacer que funcione correctamente.

La función de inicio de sesion

path('accounts/', include('django.contrib.auth.urls')),

Esta línea la usamos para que cuando el usuario inicie sesión puedan autentificarse los datos.

def login_page(request):

return render(request, 'registration/login.html')

Esta función se encarga de llevarte a la pantalla de inicio de sesión al pulsar en el botón “Iniciar Sesión”

def logout_page(request):

return render(request,'index.html/accounts/logout')

Esta función se encarga de salir de la sesión del usuario en el momento en que pulse el botón “Salir”

@login_required
def getAllFavouritesByUser(request):

favourite_list=services_nasa_image_gallery.getAllFavouritesByUser(request)

return render(request, 'favourites.html', {'favourite_list': favourite_list})

def getAllFavouritesByUser(request):
if not request.user.is_authenticated:
return []
else:
user = get_user(request)

favourite_list = repositories.getAllFavouritesByUser(usermapped_favourites []
for favourite in favourite_list:
nasa_card = mapper.fromRepositoryIntoNASACard(favourite)
mapped_favourites.append(nasa_card)
return mapped_favourites

Estas dos funciones reciben todas las cartillas que se hayan marcado como favoritas del usuario. Las mismas se muestran en la tabla de la sección “Favoritos”

@login_required
def saveFavourite(request):

services_nasa_image_gallery.saveFavourite(request)

return HttpResponseRedirect(request.META.get('HTTP_REFERER', '/'))

def saveFavourite(request):
fav = mapper.fromTemplateIntoNASACard(request)
fav.user = get_user(request)
return repositories.saveFavourite(fav)

Estas dos funciones se encargan de guardar en favoritos del usuario cuando aprete en ‘Agregar a Favoritos’

def deleteFavourite(request):
favId = request.POST.get('id')
return repositories.deleteFavourite(favId)

@login_required
def deleteFavourite(request):

services_nasa_image_gallery.deleteFavourite(request)

return HttpResponseRedirect(request.META.get('HTTP_REFERER', '/'))

Estas dos funciones se encargan de borrar la cartilla seleccionada con el botón de “borrar” de la lista de favoritos.

En conclusión, fue un trabajo bastante divertido de realizar ya que pudimos interactuar con nuevas funcionalidades dentro de la programación que nos servirán en un futuro para desarrollar programas más avanzados.