



# INTRODUCCIÓN A LA PROGRAMACIÓN

TRABAJO PRÁCTICO: GALERÍA DE IMÁGENES LA NASA

ALUMNOS: RAMIREZ LARA MARTINA

DNI: 45.994.695

RAMIREZ GALA MORENA

DNI: 45.994.696

*COMISION 07*

PROFESORES: SERGIO SANTA CRUZ

NAZARENO AVALOS

NAHUEL SAHUMA

# INTRODUCCIÓN

Principalmente, nosotras queríamos aclarar, que hicimos ambas el trabajo practico en una misma computadora. Somos hermanas y vivimos juntas, por lo tanto, la computadora es de ambas.

Como primer paso lo que hicimo fue crear cada una su propia cuenta de github. Luego procedimos a descargar e instalar lo que se nos pidió en el TP (VS Code, Python, Django, DB Browser SQLite). También, instalamos las extensiones pedidas, en VS Code (After Dark, Prettier – Code formatter, Pylance, Python, Python Debugger).

Hicimos un fork del proyecto, ip-public-repo, hacia nuestro repositorio local, y lo clonamos (git clone) a este mismo. Luego instalamos Django mediante una terminal en VS Code, con el código que se nos fue brindado. Procedimos a instalar las dependencias necesarias. Luego ejecutamos el servidor Django, y finalmente hicimos click en el protocolo que nos apareció en la terminal, después de haber ejecutado el paso anterior. Y de esa manera entramos al servidor Django, donde nosotros tenemos que presentar el proyecto haciéndolo funcionar.

La idea principal, es poder terminar de completar las funciones, que nos presentan en la explicación del trabajo practico, que aun no están terminadas. Dentro del servidor, tienen que cargarse las imágenes que fueron extraídas desde una API de la NASA. De esta manera también hacer funcionar la barra de búsqueda, y la opción de agregar a favoritos, Eso también es uno de los puntos extras que realizamos, que sería la opción de poder loguearse, De esa manera podríamos agregar las fotos a favoritos, y si quisiésemos también eliminarlas de ahí. También la opción de poder cerrar la sesión. Y que todos estos anteriores pasos comentados, nos redirijan nuevamente a donde se nos pide, de manera correcta.

# DESARROLLO

El Framework **Django**, nos permite crear el proyecto general, y a través de ese proyecto, una vez creado, podemos crear las apps, en su caso sería la página la cual vamos a trabajar.

**1-** `manage.py`, es el archivo que se conecta con la API de la NASA, y genera las imágenes.

**2-** Todo empieza en `urls.py`, ahí estaría el proyecto en general. En esta, pedimos que, si el path está vacío, incluya las `urls` de `nasa_image_gallery`

**3-** Dentro de `nasa_image_gallery`, vamos a tener otro `urls.py` las cuáles son las rutas de la app.

**4-** Lo que hicimos al entrar a la app fue definir los path y mediante ese path definimos a que views vamos a llamar, y junto a ellos, los names, que vendrían a ser los "Alias" que les colocamos para no tener que escribir todo el path, y al momento de llamarlos, los llamamos mediante su name.

**5-** En `models.py`, se definen todos los objetos que se conectan a una base de datos, en este caso solo tenemos el de la **clase favoritos**, que guarda las urls (el título, la descripción, la fecha, etc) ósea las cards.

**6-** Dentro de `views.py`, el `index_page` es una función que invoca el template del índice de la aplicación, ósea `index.html`.

`busqueda_var`, busca el parámetro `query`, si viene una petición por post. Si no se ingresó nada en la búsqueda, que cargue directamente todas las imágenes, de lo contrario, que busque de estas imágenes, las que estén relacionadas con la palabra ingresada en el buscador, eso lo podemos ver en `transport.py`

**7-** En `transport.py` en la función `GetAllImages`, también recorre la variable `json_response`, y busca los objetos, ósea las cards, que tengan links. Si es así, que las guarde en `json_collection`

**8-** En `transport.py`, si queremos ver los datos que guardamos en la lista `json_collection`, descomentamos el `print`, lo guardamos con `control+s`, y al ir a la página y entrar a galería, cuando volvemos a la terminal, nos aparecen los datos que se retornan de `json_collection`.

**9-** En `service nasa gallery`, en la función `getallimages`, ahora recorre los objetos (las cards) que están en `json_collection`. Tiene una variable llamada `card`, que captura todos los datos de cada card, las cuales guardan en la lista `images`, para luego retornarla.

**10-** Recién ahí guarda todos esos datos en el archivo **views.py** , y ahora si puede retornar **images**, **favourite\_list**. Luego de eso, podemos usar estas variables en la función **home** y renderizarlas a **home.html**.

En el caso de que se precione el botón buscar, y no se allá ingresado nada en el buscador, nos redirige al **home** nuevamente, de caso contrario, filtra las imágenes solicitadas en el buscador, con sus respectivos datos

**11-** Dentro de **template**, en **home.html** , podemos ver que hay un **for imagen in images**, que recorre las cards e imprime las que tienen valores, en el caso contrario no imprime nada. En esta misma, nos aparece la opción de que si el usuario está ingresado, que le dé la opción de agregar a favoritos. Tenemos unos **input** que están ocultos, son los datos que se guardan en favoritos, si están en la lista de favoritos nos aparecerá que ya están añadidos, si no, nos aparecerá la opción para agregar a favoritos.

**12-**En **views.py**, en la función **login**, toma los **POST**, ose los datos ingresados en el usuario y contraseña, y los guarda en variables (**username**, **password**), **Try** lo que hace es buscar los errores, es decir, controla que el usuario ingresado sea correcto (estos datos están en la base de datos, el usuario y contraseña son ambos "**admin**"), en este caso, se guardara en la variable **user**. **except** seria que, si captura el error, que nos dé un aviso de que no se encontró un usuario con ese nombre, en este caso, nos redirige a **login**. Pedimos que autentique los datos ingresados, y si esta todo correcto, que los guarde en la variable **user**, y que imprima que el usuario está autorizado. Si es así, que lo renderice a **home**, y ahí ya podemos seleccionar las fotos a favoritos, o borrarlas de favoritos también. **@login\_required** nos ayuda a dar permiso de ciertas funciones, solo a los usuarios logeados, en este caso, las fotos guardadas en favoritos.

**13-** En **home.html**, nos muestra la condición, que le mande un **request** solo a los usuarios autenticados (ósea lo que hablábamos recién, el permiso de la lista de favoritos, y las demás opciones de esta lista.

**14-** Volvemos a **views.py** y vemos que en una variable con el nombre **card**, guarda las **nasa card**, ósea los datos de las cards seleccionadas para ingresarlas a favoritos, luego las guarda en la función **guardar favoritos** y nos redirige a **home**. También esta la función **deleteFavourite**, que nos da la opción de borrar una card de favoritos. Luego tenemos la función **exit**, que nos redirige a **login**, en caso de que cerremos la sesión , y ya no nos aparecera la opción de guardar a favoritos.

## **PROBLEMAS ENCONTRADOS**

Nosotras, principalmente tuvimos problemas, con la invocación de funciones desde varios módulos. Nos mareemos un poco en ese momento, pero pudimos solucionarlo mediante videos informativos de usuarios de YouTube como: Soy Dalto, UskoKruM2010, Deivchoi. También con algunas funciones o variables nuevas que no conocíamos, Pero luego pudimos entender varias invocaciones, formas que ya venían implementadas en el proyecto, gracias a algunos apuntes de la clase. No se nos complico tanto, mas que al principio, a pesar de que pudimos solucionar esas dudas.

## **CONCLUSION**

Aprendimos a llamar a variables y funciones desde varios módulos a otros, también a usar VS Code, GIT, GitHub, DB Browser SQLite. A familiarizarnos mas con Python y todos sus códigos. Y También a poder simplificar códigos y llamados a funciones mediante Names (Alias). Pudimos arreglarnos para poder hacer el trabajo en conjuntos y compartirnos ideas capturadas de las clases de Introducción a La Programación. Gracias a varios videos vistos, también nos fuimos fluyendo por otros videos explicativos que nos brindaban ideas muy importantes.

## **ANEXO**

[https://www.youtube.com/watch?v=vlCXdvciE0&ab\\_channel=deivchoi](https://www.youtube.com/watch?v=vlCXdvciE0&ab_channel=deivchoi)

[https://www.youtube.com/watch?v=nKPbfIU442g&t=89s&ab\\_channel=SoyDalto](https://www.youtube.com/watch?v=nKPbfIU442g&t=89s&ab_channel=SoyDalto)

[https://www.youtube.com/watch?v=UDad8Vwz6N0&ab\\_channel=Roelc ode](https://www.youtube.com/watch?v=UDad8Vwz6N0&ab_channel=Roelc ode)

[https://www.youtube.com/watch?v=rwUVwxh5N74&ab\\_channel=UskoKruM2010](https://www.youtube.com/watch?v=rwUVwxh5N74&ab_channel=UskoKruM2010)

