

Licenciatura en Sistemas

# Trabajos Práctico

## Nombre del TP

Introducción a la Programación  
2Do semestre 2024

### Resumen:

El trabajo consiste en reparar una página web la cual contiene un listado de personajes de la serie Rick and Morty y unas características determinadas de cada uno de estos.

Integrantes: David Uriel Guaraz, Valentino Muto, Franco Emmanuel Rios

## 1. Introducción

El trabajo consiste en una página web la cual se encuentra malfuncionando debido a varios errores en su código. Los componentes de la página y su información se encuentran armados, pero no se logran ver en pantalla. Nuestro trabajo consistió en ver en código, identificar cual es el problema y buscar la manera de arreglarlo.



## 2. Desarrollo

Al revisar el código, en un inicio no logramos comprender nada, por suerte los que sí tenían conocimientos previos le explicaron a los demás y con eso pudimos comenzar a trabajar. Dentro del código había varias funciones que no estaban colocadas correctamente, tuvimos que comprender cómo es que el programa funcionaba e ir viendo paso a paso donde fallaba la cadena de funciones.

Comenzamos revisando la carpeta "views", en la misma encontramos lo que aparentaba ser una lista vacía llamada "images" y una línea que decía `from .layers.services import services`. La cual no estaba funcionando, por lo que decidimos buscar en el archivo llamado "services". Dentro de la carpeta "service" había una función incompleta llamada `getAllImages`, cuya descripción decía que obtenía datos crudos desde la API usando a otra carpeta llamada "transport". Notamos que no había nada conectando "transport" con "services", así que tuvimos que armar una línea de código que lo hiciera, `from ..transport import transport`. Después cambiamos el interior de la función para que esta pueda trabajar con la data traída, `json_collection = transport.getAllImages(input=None)`. Luego creamos un ciclo for, que llama a la data y dentro del cual colocamos una variable nombrada "card". Esta variable, mediante la función `translator.fromRequestIntoCard(object)`, obtiene una a una las imágenes y procede a agregarlas a una lista llamada "images", para al final retornar dicha lista, dando terminada la función.

Una vez hecho eso, volvimos a la carpeta views y cambiamos `images = []` y lo reemplazamos por `images = services.getAllImages(input=None)`. De forma tal que llamara los datos desde services.

Una vez aparecieron las imágenes, nos dimos cuenta que a pesar de que la información decía cuáles personajes estaban vivos y cuáles no, todos aparecían en color naranja. Investigando, encontramos que dentro de la carpeta templates, en un archivo nombrado `home.html` se encontraba la condicional que indicaba que color mostrar. Resulta que la condicional decía `if true == 'Alive'`, la cambiamos por `if img.status == 'Alive'` y eso arregló el problema.

Faltaba algo más, los bordes de cada casilla se encontraban en color gris, cuando deberían estar en un color específico según el estado de los personajes. Dentro de

home.html vimos que había otra variable que se encargaba de definir eso, `<div class="card mb-3 ms-5" style="max-width: 540px;">`.

Al principio intentamos agregar una condicional debajo de esa línea, copiando a la que habíamos usado para el estado de los personajes, pero no funcionó. Entonces le preguntamos a Chat gpt cómo agregar condicionales correctamente en esa línea, con su respuesta se pudo agregar la condicional `border: 3px solid {% if img.status == 'Alive' %}green{% elif img.status == 'Dead' %}red{% else %}orangered{% endif %};`. La cual ahora variaba el borde de verde o rojo si el estado era vivo o muerto y cambiaba el color gris del estado unknown a naranja. Con eso se podría dar por arreglada presentación de las imágenes de la página.

### 3. Conclusiones:

En conclusión, podemos afirmar que la programación se trata de una práctica muy amplia, la cual eleva su complejidad a medida que varios lenguajes y formatos se van entremezclando para trabajar en conjunto. Este trabajo es un ejemplo de ello y de lo necesario que es tener un conocimiento amplio o al menos básico de varias formas de trabajar dentro de la programación. O de la necesidad de aprender a buscar información, por cuenta propia, adaptándose a nuevas tecnologías como el uso de las IAs en el ámbito de la programación.

### Anexo:

El trabajo consiste en implementar una aplicación web usando Django que permita buscar imágenes de los personajes de la serie Rick & Morty, usando su API homónima. La información que provenga de esta API será renderizada por el framework en distintas cards que mostrarán -como mínimo- la imagen del personaje, el estado, la última ubicación y el episodio inicial. Adicionalmente -y para enriquecerla- se prevee que los estudiantes desarrollen la lógica necesaria para hacer funcionar el buscador central y un módulo de autenticación básica (usuario/contraseña) para almacenar uno o más resultados como favoritos, que luego podrán ser consultados por el usuario al loguearse. En este último, la app deberá tener la lógica suficiente para verificar cuándo una imagen fue marcada en favoritos.