

Licenciatura en Sistemas

Trabajos Práctico

Buscador: Rick & Morty

Introducción a la Programación

(2do Semestre - 2024)

Resumen: Página web que muestra personajes de Rick & Morty desde su API, con búsqueda, favoritos y login/logout, usando Python y Django.

Integrantes: *Mirko Stadelman mirkostadelman@outlook.com*
 Homero Martínez homero.cresta@gmail.com
 Marcos Benítez benitezmark43@gmail.com

1. Introducción (Resumen explicando de qué se trata el trabajo enfocándose en el problema que se intenta resolver de ser necesario se pueden incluir dibujos o capturas de pantalla)

2. Desarrollo (Esta sección está destinada a explicar cómo se resolvió el problema presentado en la introducción, aquí se deben incluir las principales dificultades encontradas, las soluciones a ellas y las decisiones tomadas)

2.1 Descripción general: (Resumen general de la solución completa)

2.2 Funcionalidades principales: (Breve descripción de qué se trata y cómo se resolvió la funcionalidad. En las siguientes secciones detallar las funciones implementadas incluyendo por cada función: La idea general de la función, por qué se hizo, el código comentado (utilizando tipografía adecuada) y una descripción de los parámetros que toma y los valores que devuelve/modifica)

3. Conclusiones: (Comentarios puntuales pensados finalizada la realización del trabajo)

Anexo: Enunciado (Enunciado del trabajo copiado textual)

(Nota: Se valora ser breve y conciso, incluir sólo las cuestiones relevantes que ayuden a comprender el trabajo)

1. Se nos asignó el trabajo de completar un proyecto de una página web en la cual se toman los datos provistos desde la api de Rick&Morty para generar una serie de tarjetas que contengan la imagen y el nombre de cada uno de los personajes de la serie, además, también incluye el primer episodio en el que fue visto, la última ubicación conocida y el estado de cada uno, siendo el color del borde de las tarjetas la forma más rápida de conocer el estado del personaje, verde si está vivo, rojo si está muerto o naranja si es desconocido.

2. Para cumplir con las consignas indicadas en el trabajo hicimos uso de una serie de funciones en Python que se encontraban ya implementadas en el proyecto pero que debían ser modificadas para su correcto funcionamiento. La problemática principal que afrontamos a la hora de realizar el trabajo fue la de comprender el código que se nos fue provisto, pero gracias a los comentarios implementados en todo el proyecto, a la documentación y las capacidades explicativas de la herramienta chatgpt pudimos solucionar esta dificultad.

En primer lugar, modificamos la función `home()` en el archivo `views.py`. Esta función recibe "None" como parámetro ya que renderiza las tarjetas de los personajes sin condición en la galería, haciendo uso de una lista entregada por la función `getAllImages()`, importada desde el archivo `services.py`. Además, tuvimos que completar la función `getAllImages()` de `services.py` para que funcione adecuadamente y pueda entregar una lista con las tarjetas de los personajes a la función anterior. Como la tarea de esta función es entregar la lista de tarjetas sin condición alguna, recibe "None" como parámetro. Primero obtiene una lista con los datos crudos de la api haciendo uso de la función `getAllImages()` que es importada desde el archivo `transport.py`, para luego transformar cada elemento de la lista en una carta con la función `fromRequestIntoCard()` importada del archivo `translator.py`. Por último, para completar los requisitos mínimos de aprobación, se modificó el archivo

home.html para que el borde de las tarjetas cambie según el estado del personaje, para esto usamos condicionales if/else con la estructura propia de Django, el framework que fue utilizado para realizar el proyecto.

Como segundo punto, incluimos la funcionalidad de la barra de búsqueda, la cual permite filtrar los personajes por su nombre. Para esto se utiliza la función `search()` del archivo `views.py`, que toma el texto ingresado por el usuario en la barra de búsqueda y lo procesa para realizar un filtro en la lista de tarjetas entregadas por `getAllImages()` de `services.py`, pasando como parámetro el texto que se ingresó para generar una nueva lista con los resultados coincidentes y renderizarlos en pantalla.

Continuando, completamos el inicio y cierre de sesión como usuario, para de esta manera poder incluir la funcionalidad de agregar personajes a una lista de favoritos. Como el inicio de sesión ya estaba implementado, solo tuvimos que completar la función `exit()` en el archivo `views.py` con la función `logout()`, nativa de Django, y retornar ruta, en este caso, nuevamente a la pestaña de inicio de sesión.

En última instancia, fue implementada la posibilidad de agregar personajes a una lista de favoritos, para esto se modificó el archivo `home.html` para que cada tarjeta incluya un botón para agregarla a la lista de favoritos y el mismo sería modificado una vez el personaje se encontrase en la lista de favoritos mostrando un mensaje diciendo que ya se encuentra el personaje en el listado. Se aplicó lo usado anteriormente de login y logout, siendo estos necesarios para poder agregar un personaje a la lista. Por último, se implementó la opción de poder eliminar a un personaje de la lista utilizando la función `deleteFavourites()`

3. Conclusión:

El trabajo nos permitió trabajar sobre nuestros conocimientos en distintas áreas de la programación como lo serían el programar en el lenguaje de Python, html, también nos ayudó a comprender como podría ser un proyecto profesional en el cual están involucradas varias personas. Por otra parte, el código ya existente facilitó mucho el trabajo y ayudó a mejorar la comprensión del código y cómo funciona el programa, para así hacerlo funcional y añadirle algunas funciones extras, como el login/logout, el poder añadir y eliminar tarjetas de la lista de favoritos o la barra de búsqueda.