



29-11-2024

Introducción a la programación

Alumnas:

- Poblete Insaurralde Maria Fernanda
- Poblete Insaurralde Melina Soledad

Profesores:

- Sergio Santa Cruz
- Nahuel Sauma
- Nancy Nores

Comisión: 07



Introducción

El presente informe tiene como objetivo explicar cómo se realizó el trabajo de introducción a la programación. Detallar cuales fueron los códigos de las funciones implementadas, las dificultades que surgieron y las soluciones encontradas.

**Objetivo:**

El objetivo del trabajo practico consiste en implementar una aplicación web usando Django que permita buscar imágenes de la serie Rick & Morty, usando su API homónima. La información que proviene de esta API es renderizada por el framework en distintas cards y deben mostrar la imagen del personaje, el estado, la última ubicación y el episodio inicial.

Funciones implementadas:

- **Views.py**

Obtiene 2 listados que corresponden a las imágenes de la API y los favoritos del usuario, y los usa para dibujar el correspondiente template. Si el opcional de favoritos no está desarrollado, devuelve un listado vacío.

```
def home(request):
    images=[]
    services.getAllImages(None)
    lista_fav=[]

    for i in range(len(images)):
        if images[i]==lista_fav[i]:
            print(images[i])
```

- **Services.py**

Obtiene un listado, con formato de Card, de imágenes de la API. El parámetro input, si está presente, indica sobre qué imágenes/personajes debe filtrar/traer.

```
from ..persistence import repositories
from ..utilities import translator
from django.contrib.auth import get_user
from ..transport import transport

def getAllImages(input=None):
    # obtiene un listado de datos "crudos" desde la API, usando a transport.py.
    json_collection = []
    json_collection = transport.getAllImages(input)
    # recorre cada dato crudo de la colección anterior, lo convierte en una Card y lo agrega a images.
    images = []
    for imagenes in json_collection:
        images.append(translator.fromRequestIntoCard(imagenes))
    return images
```



- **Home.html**

La card debe cambiar su border color dependiendo del estado del personaje. Si está vivo (alive), mostrará un borde verde; si está muerto (dead) mostrará rojo y si es desconocido (unknown), será naranja.

```
{% img.status == 'Alive' %} ● {{ img.status }}  
{% img.status == 'Dead' %} ● {{ img.status }}  
{% else %} ● {{ img.status }}  
{% endif %}  
/strong>
```