PROYECTO TP

TRABAJO PRACTICO FINAL DE I.P.

POR ZABALA DAMIAN Y
FERNANDEZ EZEQUIEL

Profesores: Fernando y Claudia

Comision: 05

2° semestre 2024

RESUMEN DE INFORME

INTRODUCCION

CAMBIOS EN VIEWS.PY

CAMBIOS EN SERVICES.PY

CAMBIOS EN HOME.HTML

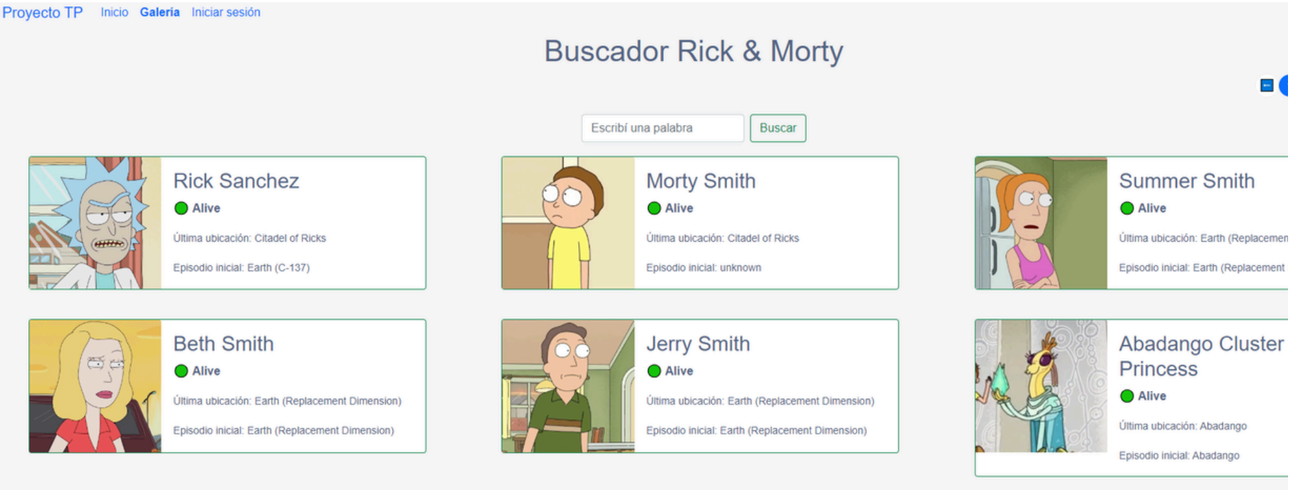
OTROS CAMBIOS

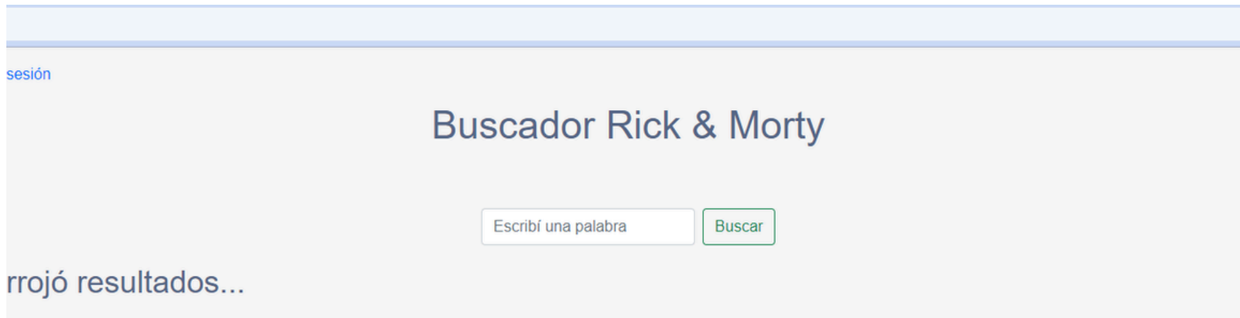
PRESENTACION

INTRODUCCIÓN

Informe sobre el proyecto TP, la pagina de rick and morty. El trabajo practico consiste en hacer los arreglos necesarios para que la pagina funcione de forma adecuada, y lo que se espera al finalizar es que la pagina se vea de la siguiente manera.

Se espera lograr que una pagina que no muestra nada mas que unos items, luego de las modificaciones necesarias muestre las imagenes de los personajes de la serie rick and morty, desde este punto de partida.





MODIFICACIONES

ARCHIVO VIEWS.PY

En el archivo views.py modificamos la funcion home para que mostrara las imagenes de los personajes.
Logrado llamando a la funcion getAllImages desde services.py

```python
def home(request):
    images = services.getAllImages()
    favourite_list = []
    return render(request, 'home.html', { 'images': images, 'favourite_list': favourite_list })
```

ARCHIVO SERVICES.PY

Las modificaciones de el archivo services.py fueron necesarias para que la funcion en views funcione correctamente, primero importamos la libreria "transport" y luego modificamos la funcion getAllImages para ver en pantalla las cards de los personajes.

```
me.html M   services.py   views.py
```

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

```python
def getAllImages(input=None):
    # obtiene un listado de datos "crudos" desde la API, usando a transport
    # brindamos a un valor a la variable para usarlo en un futuro programa
    json_collection = transport.getAllImages()

    # recorre cada dato crudo de la colección anterior, lo convierte en una
    images = []
    # programa que recorre los archivos de json_collection y los convierte
    for i in range(len(json_collection)):
        image = translator.fromRequestIntoCard(json_collection[i])
        images.append(image)
    return images
```

ARCHIVO HOME.HTML

Por ultimo para conseguir los bordes agregamos codigo al archivo home.html y modificamos unos valores para que los bordes y el estado del personaje se mostraran correctamente.

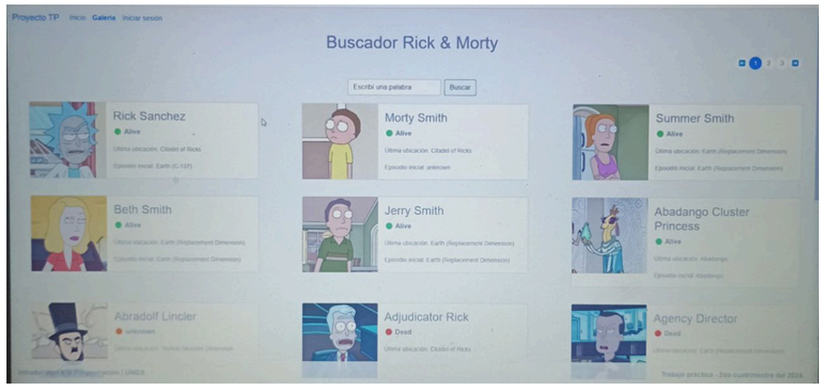
```
<!-- esta parte esta agregada para el TP -->
<div class="card mb-3 ms-5
    {% if img.status == 'Alive' %}border border-1 border-success
    {% elif img.status == 'Dead' %}border border-1 border-danger
    {% else %}border border-1 border-warning
    {% endif %}"
    style="max-width: 540px;">
    <!-- hasta aqui lo agregado (los bordes) -->
```

```
<strong>
    <!-- modificamos la parte if True por if img.status
    {% if img.status == 'Alive' %} 🟢 {{ img.status }}
    {% elif img.status == 'Dead' %} 🔴 {{ img.status }}
    {% else %} 🟠 {{ img.status }}
    {% endif %}
    <!--tenia dos identacion y al sacarlas no afecta al
</strong>
```

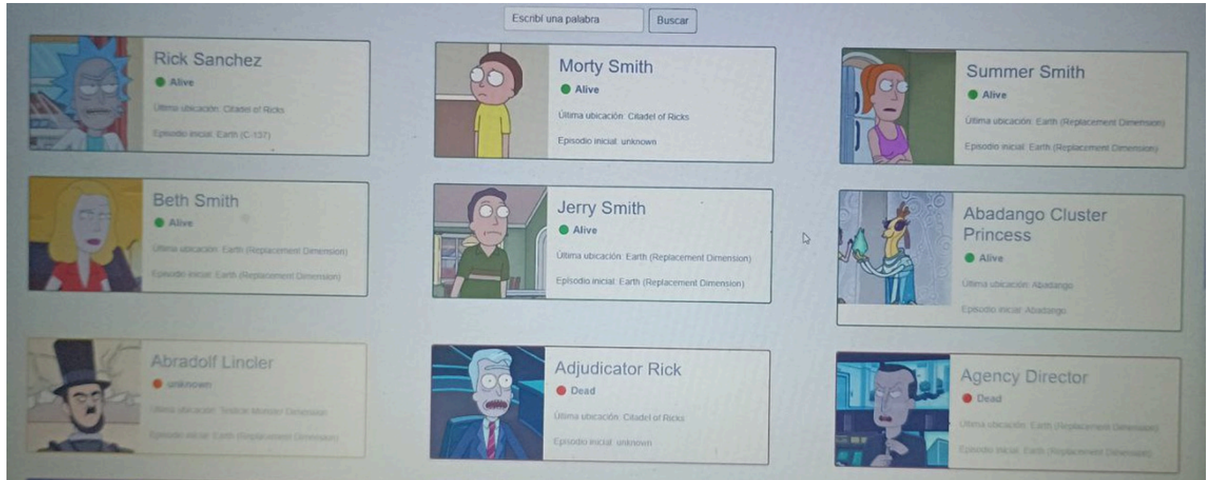
MODIFICACIONES P.2

Vista de pagina luego de modificar el archivo views.py

Vista de pagina luego de modificar el archivo services.py



Vista de pagina luego de modificar el archivo home.html



EXTRAS

ARREGLO DEL BUSCADOR Y LOGOUT

Arreglamos el buscador el buscador modificando la funcion getAllImages del archivo services.py agregando la variable (input),que muestra las imagenes con la palabra del buscador.
en el archivo views.py arreglamos la funcion search, para que el buscador funcione y devuelva las cards que se estan buscando.

```python
def search(request):
    search_msg = request.POST.get('query', '')

    # si el texto ingresado no es vacío, trae las imágenes y favoritos desd
    # y luego renderiza el template (similar a home).
    if (search_msg != ''):
        # llamamos a la funcion getAllImages nuevamente desde services, con
        images = services.getAllImages(search_msg)
        favourite_list = []
        # devolvemos lo mismo que la funcion home pero con las limitaciones
        return render(request, 'home.html', {'images': images, 'favourite_l
    else:
        return redirect('home')
```

En cuanto al logout es dificil mostrarlo en funcionamiento , llamamos a la funcion “logout" dentro de la funcion “exit" y redirigimos al usuario al inicio “index.html". Dentro del archivo views.py.