

THE CREATION OF A POSTING NETWORK PLATFORM

Tomás Cárdenas Benítez, 20221020021

Juan Jesus Poveda, 20202020128



INTRODUCTION

In the following report we will present the program inspired by the given Twitter platform, which is a very recognized platform and can view, publish, search for different content, be it video, text or image to share with other audiences. In addition to being a safe platform because you can eliminate or block unwanted contacts in order to improve the entire project



THE PLATFORM DESIGN

MATERIALS

We mainly use json and fake to compile the information and create test users.

METHODS

In the organization of the program we use facade thus giving the organizational part and using a general menu for execution, the actions are generated thanks to a factory which manages the functions of each type of post



Menu

This class provides a menu interface for interacting with the social media system.

This class provides the basic definition of a user.

SocialMediaFacade

User

This class provides the basic definition of a user.

This class provides methods in order to authenticate de log in of a user.

Authenticator

PostFactory

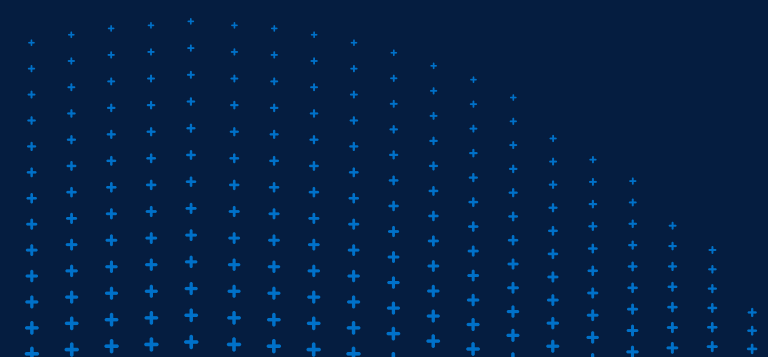
This class creates a post object based on the provided post type.

This class provides the basic definition of a user

Post

Timeline

This function initializes the Timeline object.



FAKE

The following algorithm represents the creation of a fake user and how it must be generated in the code for it to be related.

```
add_fake_users(num_users=10):
    users = []

    for _ in range(num_users):
        user = {
            "username": create_person_name().replace(" ", "_"),
            "password": create_password(),
            "profile": {
                "bio": fake.text(),
                "pfp": fake.image_url(),
                "birthday": str(create_birthday()),
                "posts": [
                    {
                        "text": create_person_post(),
                        "likes": generate_likes(),
                        "comments": [create_comments() for _ in range(random.randint(1, 5))],
                        "saved": generate_saves(),
                    } for _ in range(random.randint(1, 5))]
                ]
            }

        users.append(user)

    with open("users.json", "w", encoding="UTF-8") as file:
        json.dump(users, file, indent=4)
```

EXPERIMENTS AND RESULTS

```
Bienvenido al sistema de redes sociales!
1. Iniciar sesión
2. Crear usuario
0. salir
Seleccione una opción: 2
Ingrese un nombre de usuario: juna
Ingrese una contraseña: juan
Ingrese su biografia: juan
Ingresa la direccion de tu imagen de usuario: as
Ingrese su cumpleaños: 2000-12-23
Usuario posible
Usuario ya existente
Usuario añadido exitosamente
Usuario creado exitosamente.
Bienvenido al sistema de redes sociales!
1. Iniciar sesión
2. Crear usuario
0. salir
Seleccione una opción: 1
Ingrese su nombre de usuario: juan
Ingrese su contraseña: juan
¡Inicio de sesión exitoso!
```

```
{
  "username": "juna",
  "password": "juan",
  "profile": {
    "bio": "juan",
    "pfp": "as",
    "birthday": "2000-12-23",
    "posts": []
  }
}
```

The following images represent the process of saving and starting the user correctly

CONCLUSIONS

- The simplest relationships are necessary because everything depends on them and they cannot be altered.
- Virtual environments like GitHub make it easy to move programs and collaborate properly among programmers. also generating sufficient changes to solve problems, thus avoiding problems of loss of information since each one has a copy of their Code in their own cloud
- Not having the necessary organization of the program can generate headaches, but by managing the fundamentals of the program you can understand the process very easily.