

MOIHUB

ESEI / ALS

Alexandre Moinelo Rodríguez

ÍNDICE

- **INTRODUCCIÓN**
- **INSTALACIÓN Y CONFIGURACIÓN**
- **MODELOS**
- **VISTAS**
- **CONTROLADOR PRINCIPAL**
- **INTERFACES**
- **DIAGRAMAS**
- **CONCLUSIÓN**

INTRODUCCIÓN

Moihub es una aplicación web diseñada para gestionar repositorios de código, similar a GitHub. Permite a los usuarios registrarse, subir archivos de código, comentar, y gestionar sus repositorios.

Como objetivos tenemos:

- Crear una plataforma para compartir y gestionar código.
- Facilitar la colaboración entre usuarios.
- Implementar funcionalidades de comentarios y likes para el feedback del código.

INSTALACIÓN Y CONFIGURACIÓN

Tecnologías utilizadas:

- Python
- Flask
- Sirope (para la persistencia)
- Flask-Login (para la gestión de usuarios)

Para la instalación se requiere

- Python 3.7 o superior
- Pip (gestor de paquetes de Python)

Las instrucciones para ejecutar la página web son:

1. Clonar el repositorio del proyecto
2. Instalar dependencias
 - `pip install -r requirements.txt`
3. Crear un archivo config.json en el directorio instance con la configuración necesaria.

4. Ejecutar en la terminal
 - python app.py

MODELOS

En la carpeta models tenemos:

USER:

```
class User(flask_login.UserMixin):
    def __init__(self, email, pswd):
        self.email = email
        self.pswd = safe.generate_password_hash(pswd)
    # Métodos principales
```

Code_File:

```
class CodeFile:
    def __init__(self, user_email, name, content, language,
is_favorite=False, likes=0, is_liked=False):
        self.user_email = user_email
        self.name = name
        self.content = content
        self.language = language
    # Métodos principales
```

Comment:

```
class Comment:
    def __init__(self, user_email, code_file_id, content):
        self.user_email = user_email
        self.code_file_id = code_file_id
        self.content = content
    # Métodos principales
```

VISTAS

Pondré las funciones de cada una de los views:

Code_File:

```
def my_code_files():  
def add_code_file()  
def edit_code_file(safe_code_id)  
def delete_code_file(safe_code_id)  
def view_code_file(safe_code_id)  
def toggle_favorite(safe_code_id)  
def favorite_code_files()  
def like_code_file(safe_code_id)
```

User.py:

```
def user_add():  
    srp = sirope.Sirope()  
    email = request.form.get('edEmail', "").strip()  
    pswd_hash = request.form.get('edPswd', "").strip()  
  
    if not email or not pswd_hash:  
        flash('Email y contraseña requeridos')  
        return redirect('/')  
  
    usr = User(email, pswd_hash)  
    if User.find(srp, email):  
        flash('Usuario ya existe')  
        return redirect('/')  
  
    srp.save(usr)  
    login_user(usr)  
    flash('Usuario registrado y logueado')  
    return redirect('/')
```

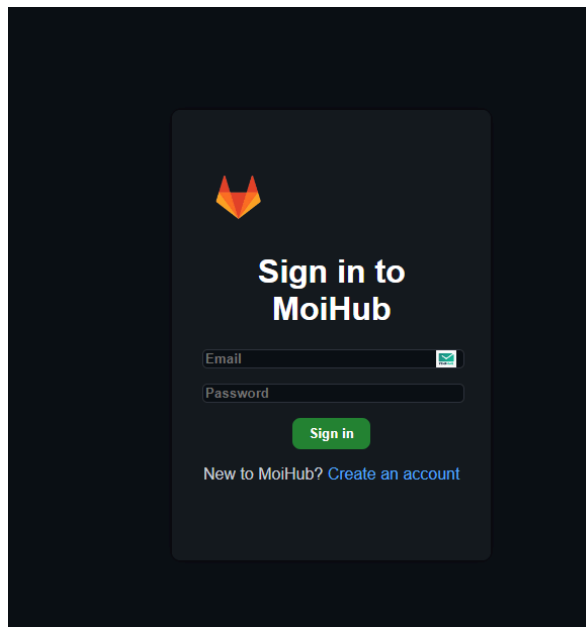
CONTROLADOR PRINCIPAL

app.py:

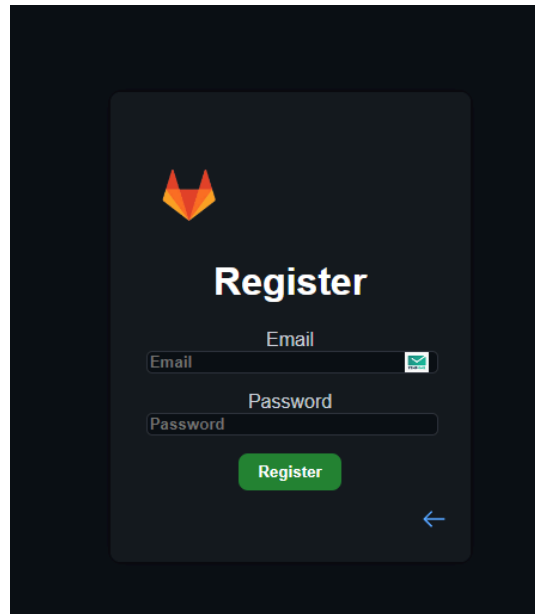
```
def create_app():  
    flapp = flask.Flask(__name__)  
    sirop = sirope.Sirope()  
    login = flask_login.LoginManager()  
    flapp.config.from_file("instance/config.json", json.load)  
    flapp.secret_key = flapp.config["SECRET_KEY"]  
    login.init_app(flapp)  
    flapp.register_blueprint(user_blpr)  
    flapp.register_blueprint(code_file_blpr, url_prefix='/code_file')  
    return flapp, sirop, login
```

INTERFACES

Como pestaña de login tenemos:

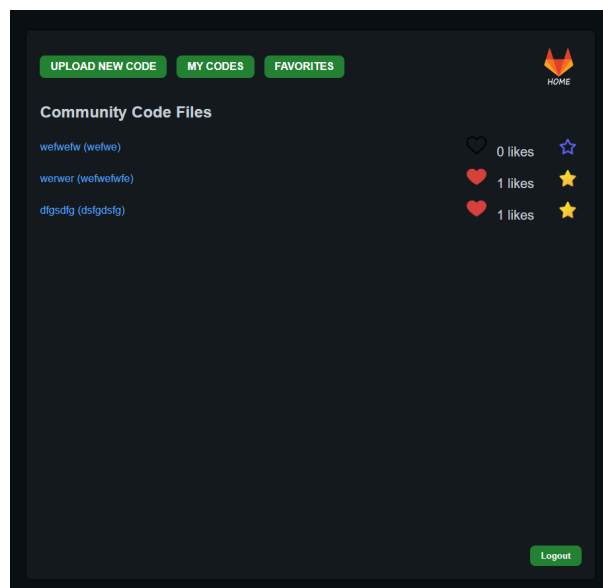


A la pestaña de registro accederemos pulsando en el hipervínculo “Create an account”



The image shows a dark-themed registration form. At the top is the GitHub logo. Below it is the word "Register" in a large, bold font. There are two input fields: "Email" and "Password". The "Email" field has a small envelope icon on the right. Below the "Password" field is a green "Register" button. In the bottom right corner of the form area is a blue back arrow.

Al acceder a nuestra cuenta tendrás una interfaz con diferentes botones y una sección de códigos de la comunidad, que podremos tanto valorar, marcar como favorito o comentar en ellos.



The image shows a user interface with a dark background. At the top, there are three green buttons: "UPLOAD NEW CODE", "MY CODES", and "FAVORITES". To the right of these buttons is the GitHub logo and the word "HOME". Below this is the section "Community Code Files". It contains a list of three items, each with a username, a code snippet, a heart icon, a number of likes, and a star icon. The items are: "wefwefw (wefwefw)" with 0 likes, "worwer (wefwefw)" with 1 like, and "dsgsdg (dsgsdg)" with 1 like. At the bottom right is a green "Logout" button.

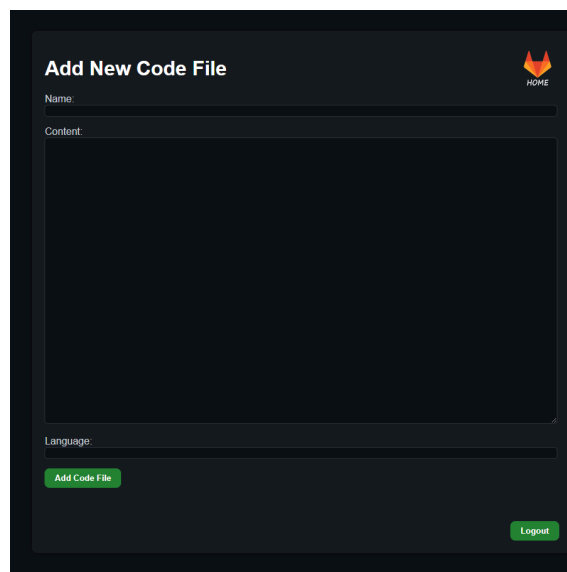
Username	Code Snippet	Likes	Favorites
wefwefw (wefwefw)		0 likes	
worwer (wefwefw)		1 likes	
dsgsdg (dsgsdg)		1 likes	

Cada vez que se vea este botón:

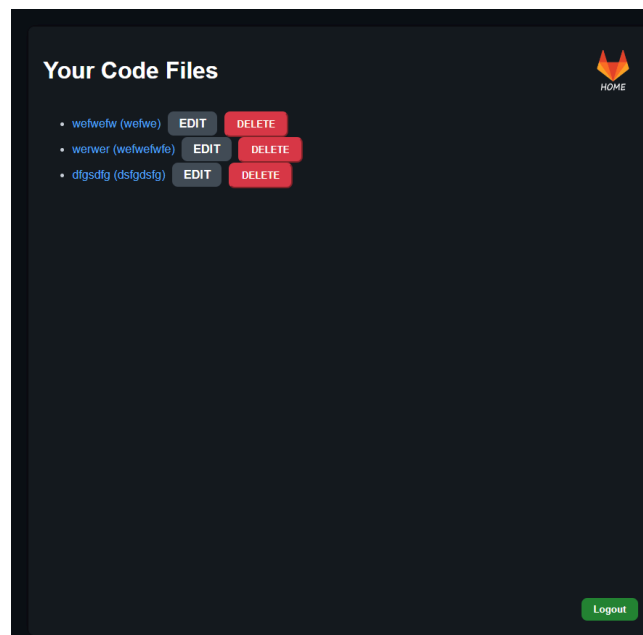


Si hacemos click en él, nos llevará a la ventana principal de nuestra cuenta, para una mayor fluidez.

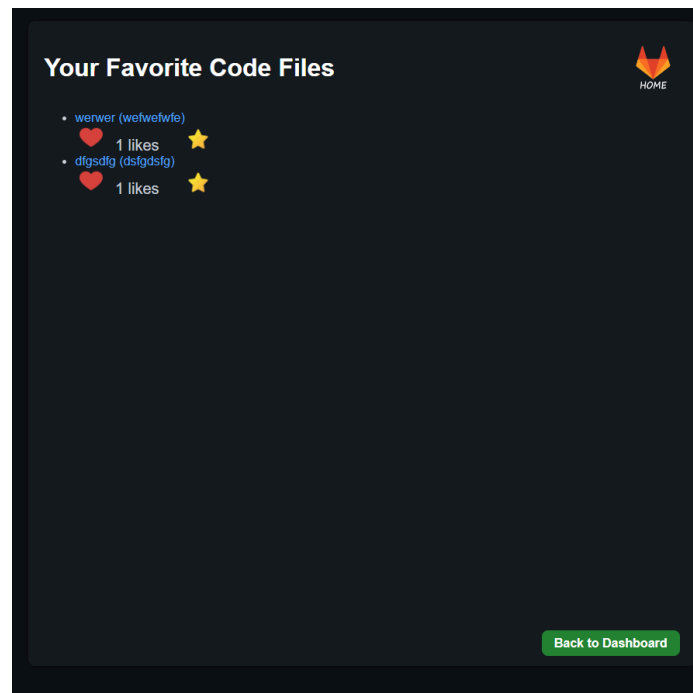
Podemos tanto subir un código:



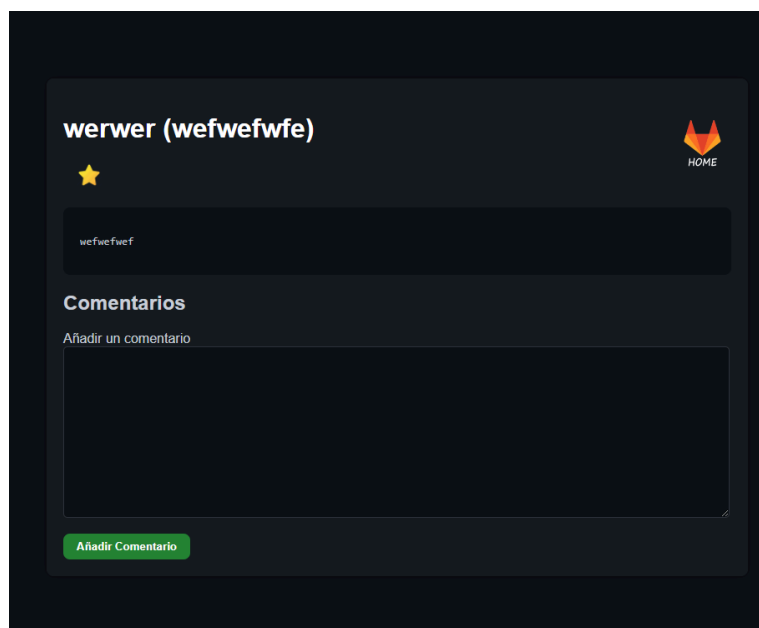
Como ver la lista de tus códigos y además editarlos o eliminarlos:



Además de eso tenemos una sección donde están listados tus códigos favoritos para acceder más fáciles a ellos:

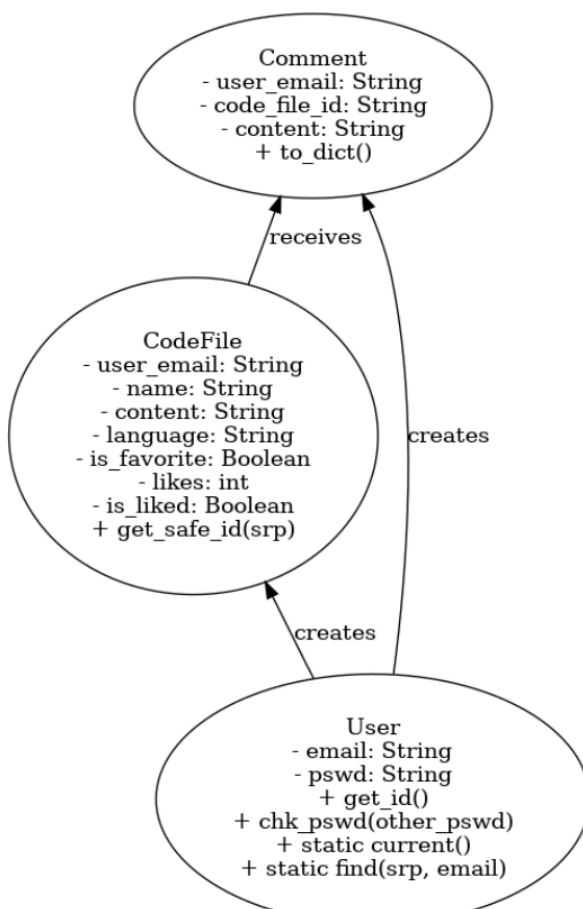
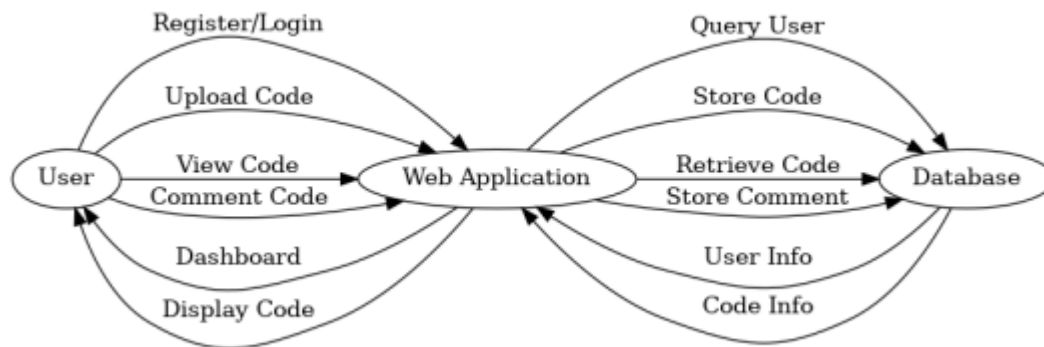


En cuanto a la sección de códigos de la comunidad tenemos tanto botones de like, con su recuento de likes y un botón de favoritos. Si se quiere ver la información más en detalle de alguno de los códigos, solamente hay q clickarlos:



En esta sección un usuario puede comentar lo que le parezca el código y que se quede guardado el comentario en la publicación del código.

DIAGRAMAS



CONCLUSIÓN

El proyecto MoiHub ha logrado desarrollar una plataforma funcional similar a GitHub, permitiendo a los usuarios registrarse, iniciar sesión, subir archivos de código, comentarlos y gestionarlos. La implementación de funcionalidades básicas de autenticación, autorización y persistencia de datos ha sido exitosa, y se ha creado una interfaz de usuario intuitiva. En el futuro, se podrían mejorar aspectos como la seguridad, el rendimiento y añadir funcionalidades de colaboración y análisis para fortalecer y expandir la plataforma.