```
# Importación de librerías necesarias
from flask import Flask, render_template, request, redirect, url_for, session
from flask_pymongo import PyMongo
from bson.objectid import ObjectId
from datetime import datetime
from werkzeug.security import generate_password_hash, check_password_hash

# Inicialización de la aplicación Flask
app = Flask(__name__)
app.secret_key = "supersecretkey"  # Clave secreta para manejar sesiones

# Configuración de conexión con la base de datos MongoDB
app.config["MONGO_URI"] = "mongodb://localhost:27017/blog_db"
mongo = PyMongo(app)  # Inicialización del objeto Mongo

# Ruta principal: muestra los artículos publicados
@app.route("/")
def index():
    posts = mongo.db.articles.find()
    return render_template("index.html", posts=posts)

# Registro de nuevos usuarios
@app.route("/register", methods=["GET", "POST"])
def register():
    if request.method == "POST":
        email = request.form["email"]
        password = request.form["password"]
        existing_user = mongo.db.users.find_one({"email": email})
        if existing_user:
            return "El usuario ya existe"
        hashpass = generate_password_hash(password)
        mongo.db.users.insert_one({"email": email, "password": hashpass})
        return redirect(url_for("login"))
    return render_template("register.html")

# Inicio de sesión
@app.route("/login", methods=["GET", "POST"])
def login():
    if request.method == "POST":
        email = request.form["email"]
        password = request.form["password"]
        user = mongo.db.users.find_one({"email": email})
        if user and check_password_hash(user["password"], password):
            session["user"] = user["email"]
            return redirect(url_for("index"))
        return "Credenciales incorrectas"
    return render_template("login.html")

# Cierre de sesión
@app.route("/logout")
def logout():
    session.pop("user", None)
    return redirect(url_for("index"))

# Crear nuevo post
@app.route("/post/new", methods=["GET", "POST"])
```

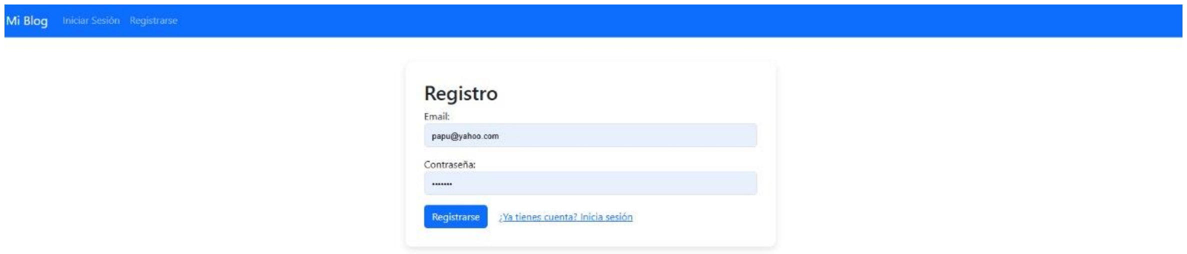
```
def create_post():
    if "user" not in session:
        return redirect(url_for("login"))
    if request.method == "POST":
        title = request.form["title"]
        content = request.form["content"]
        category = request.form["category"]
        mongo.db.articles.insert_one({
            "title": title,
            "content": content,
            "author": session["user"],
            "date": datetime.now().strftime("%Y-%m-%d %H:%M"),
            "category": category
        })
        return redirect(url_for("index"))

    categories = mongo.db.categories.find()
    return render_template("create_post.html", categories=categories)

# Ver un post específico con sus comentarios
@app.route("/post/<post_id>")
def view_post(post_id):
    post = mongo.db.articles.find_one({"_id": ObjectId(post_id)})
    comments = list(mongo.db.comments.find({"post_id": ObjectId(post_id)}))
    return render_template("view_post.html", post=post, comments=comments)

# Editar un post propio
@app.route("/post/<post_id>/edit", methods=["GET", "POST"])
```

```
def edit_post(post_id):
    post = mongo.db.articles.find_one({"_id": ObjectId(post_id)})
    if "user" not in session or session["user"] != post["author"]:
        return redirect(url_for("login"))
    if request.method == "POST":
        mongo.db.articles.update_one(
            {"_id": ObjectId(post_id)},
            {"$set": {
                "title": request.form["title"],
                "content": request.form["content"]
            }}
        )
        return redirect(url_for("view_post", post_id=post_id))
    return render_template("edit_post.html", post=post)

# Eliminar un post propio
@app.route("/post/<post_id>/delete", methods=["POST"])
def delete_post(post_id):
    post = mongo.db.articles.find_one({"_id": ObjectId(post_id)})
    if "user" in session and session["user"] == post["author"]:
        mongo.db.articles.delete_one({"_id": ObjectId(post_id)})
    return redirect(url_for("index"))

# Agregar un comentario a un post
@app.route("/post/<post_id>/comment", methods=["POST"])
def add_comment(post_id):
    if "user" not in session:
        return redirect(url_for("login"))
    content = request.form["content"]
    mongo.db.comments.insert_one({
        "post_id": ObjectId(post_id),
        "content": content,
        "author": session["user"],
        "date": datetime.now().strftime("%Y-%m-%d %H:%M")
    })
    return redirect(url_for("view_post", post_id=post_id))

# Editar un comentario propio
@app.route("/comment/<comment_id>/edit", methods=["GET", "POST"])
def edit_comment(comment_id):
    comment = mongo.db.comments.find_one({"_id": ObjectId(comment_id)})
    if "user" not in session or session["user"] != comment["author"]:
        return redirect(url_for("login"))
    if request.method == "POST":
        mongo.db.comments.update_one(
            {"_id": ObjectId(comment_id)},
            {"$set": {"content": request.form["content"]}}
        )
        return redirect(url_for("view_post", post_id=comment["post_id"]))
    return render_template("edit_comment.html", comment=comment)

# Eliminar un comentario propio
@app.route("/comment/<comment_id>/delete", methods=["POST"])
def delete_comment(comment_id):
    comment = mongo.db.comments.find_one({"_id": ObjectId(comment_id)})
    if "user" in session and session["user"] == comment["author"]:
        mongo.db.comments.delete_one({"_id": ObjectId(comment_id)})
    return redirect(url_for("view_post", post_id=comment["post_id"]))

# Listar categorías existentes
@app.route("/categories")
def list_categories():
    categories = mongo.db.categories.find()
    return render_template("categories.html", categories=categories)

# Crear una nueva categoría
@app.route("/category/new", methods=["GET", "POST"])
def create_category():
    if "user" not in session:
        return redirect(url_for("login"))
    if request.method == "POST":
        name = request.form["name"]
        if mongo.db.categories.find_one({"name": name}):
            return "La categoría ya existe"
        mongo.db.categories.insert_one({"name": name})
        return redirect(url_for("list_categories"))
    return render_template("create_category.html")

# Ver todos los posts de una categoría
@app.route("/category/<name>")
def posts_by_category(name):
    posts = mongo.db.articles.find({"category": name})
    return render_template("index.html", posts=posts)
```

```
# Perfil de usuario con sus posts y comentarios
@app.route("/profile/<email>")
def user_profile(email):
    user_posts = list(mongo.db.articles.find({"author": email}))
    user_comments = list(mongo.db.comments.find({"author": email}))
    return render_template("profile.html", email=email, posts=user_posts,
comments=user_comments)


# Ejecutar la app si este archivo es el principal
if __name__ == "__main__":
    app.run(debug=True)
```

Fotos del blog en uso

Inicio/registro

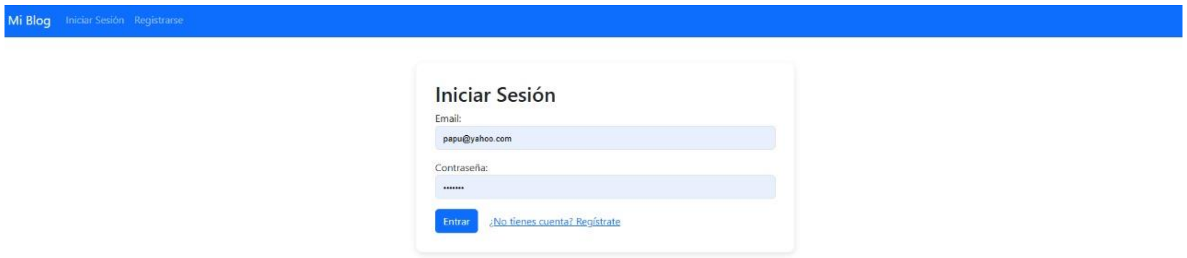

Mi Blog Iniciar Sesión Registrarse

Registro

Email:
papu@yahoo.com

Contraseña:
•••••••

Registrarse ¿Ya tienes cuenta? Inicia sesión


Mi Blog Iniciar Sesión Registrarse

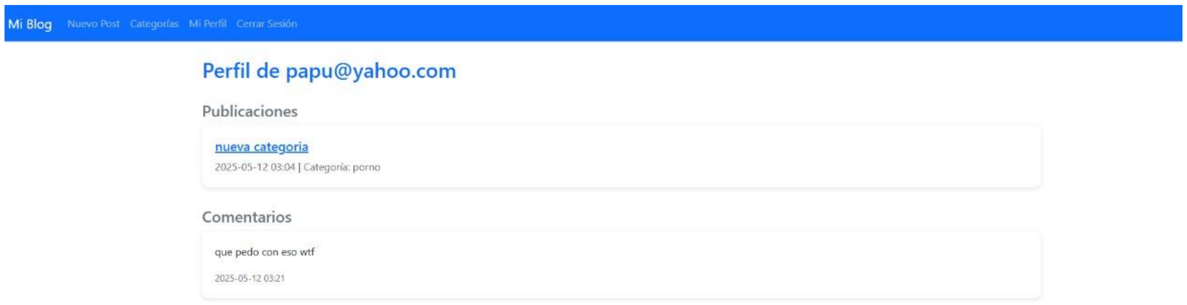
Iniciar Sesión

Email:
papu@yahoo.com

Contraseña:
•••••••

Entrar ¿No tienes cuenta? Regístrate

Perfil propio


Mi Blog Nuevo Post Categorías Mi Perfil Cerrar Sesión

Perfil de papu@yahoo.com

Publicaciones

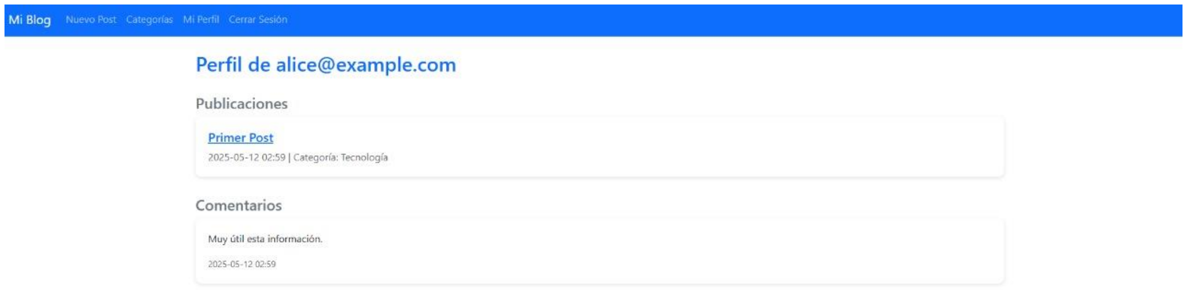
nueva categoria
2025-05-12 03:04 | Categoría: porno

Comentarios

que pedo con eso wtf
2025-05-12 03:21

Diego Jassiel Ovalle Reynoso 348795

Perfil ajeno


Mi Blog Nuevo Post Categorías Mi Perfil Cerrar Sesión

Perfil de alice@example.com

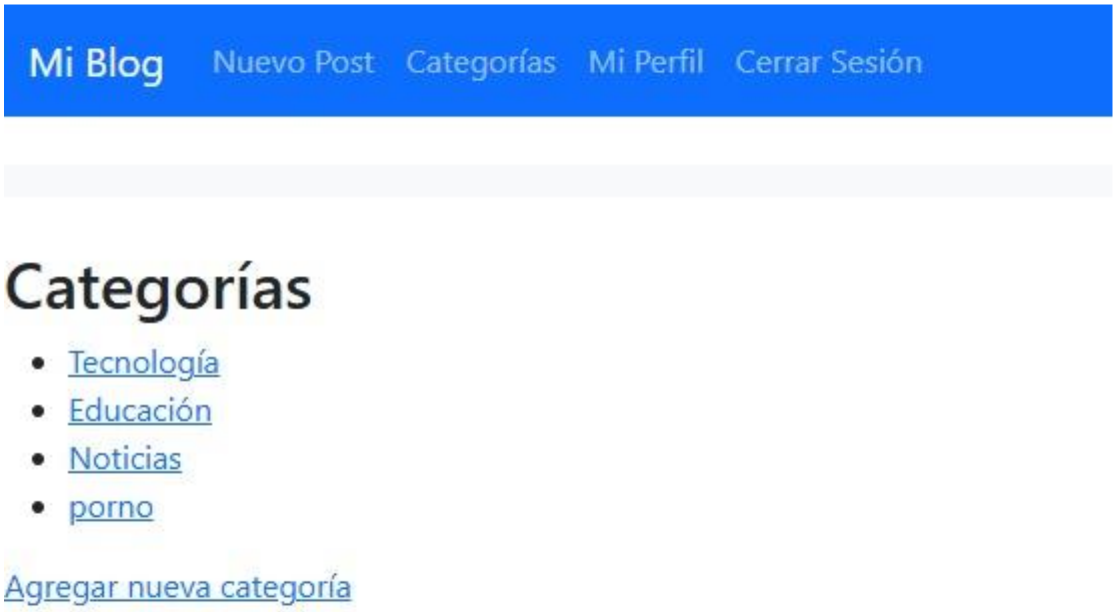
Publicaciones

Primer Post
2025-05-12 02:59 | Categoría: Tecnología

Comentarios

Muy útil esta información.
2025-05-12 02:59

Categorías

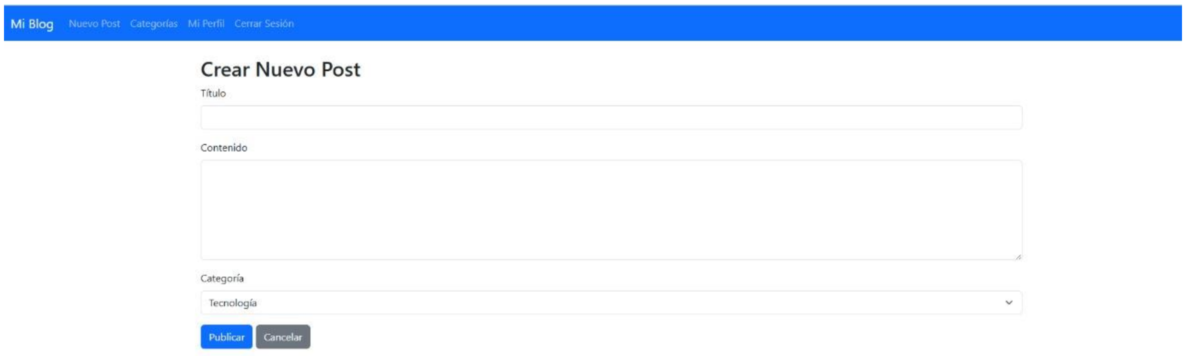

Mi Blog Nuevo Post Categorías Mi Perfil Cerrar Sesión

Categorías

- Tecnología
- Educación
- Noticias
- porno

Agregar nueva categoría

Crear post



Crear Nuevo Post

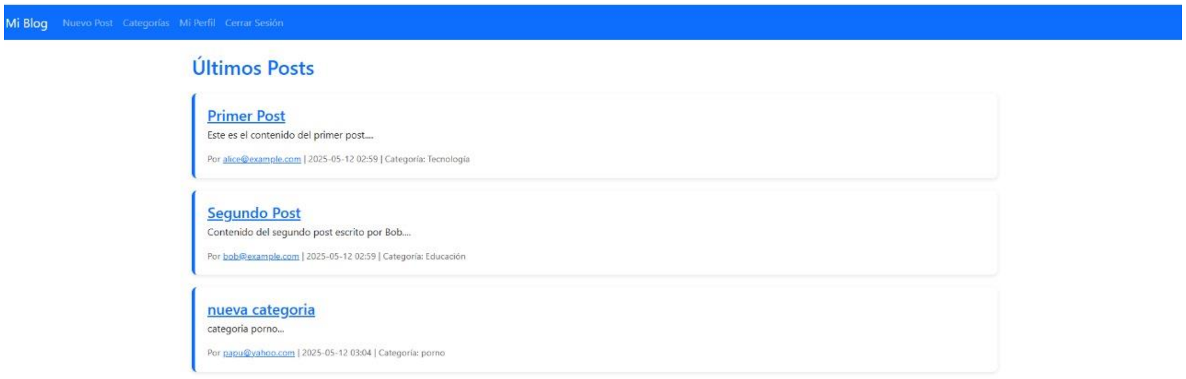
Título

Contenido

Categoría

Tecnología

Pagina principal



Últimos Posts

Primer Post
Este es el contenido del primer post....
Por alice@example.com | 2025-05-12 02:59 | Categoría: Tecnología

Segundo Post
Contenido del segundo post escrito por Bob....
Por bob@example.com | 2025-05-12 02:59 | Categoría: Educación

nueva categoria
categoria porno...
Por papu@yahoo.com | 2025-05-12 03:04 | Categoría: porno

Git hub

https://github.com/DiegoOvalle348795/blog.git