

Login e Rexistro con contrasinais escriptadas con Flask

1. Creamos a estrutura do proxecto

- `python -m venv venv`
- `./venv/Scripts/activate`
- `pip install flask`
- `pip install flask-marshmallow`
- `pip install flask-sqlalchemy`
- `pip install marshmallow-sqlalchemy`
- `pip install pymysql`

2. Creamos o arquivo `main.py`

3. Realizamos as importacións necesarias

```
from flask import Flask, request, jsonify
from flask_sqlalchemy import SQLAlchemy
import flask_marshmallow import Marshmallow
from werkzeug.security import generate_password_hash, check_password_hash

app = Flask(__name__)

if __name__ == '__main__':
    app.run(debug=True)
```

4. Configuramos a base de datos

...

```
app.config['SQLALCHEMY_DATABASE_URI'] = "mysql+pymysql://root@localhost/login"
```

```
db = SQLAlchemy(app)
```

```
ma = Marshmallow(app)
```

```
with app.app_context():  
    db.create_all()
```

```
class User(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    username = db.Column(db.String(150))  
    password = db.Column(db.String(160))  
  
    def __init__(self, username, password):  
        self.username = username  
        self.password = password
```

```
class UserSchema(ma.Schema):  
    class Meta:  
        fields = ('id', 'username', 'password')
```

```
user_schema = UserSchema()  
users_schema = UserSchema(many=True)
```

```
if __name__ == '__main__':  
    app.run(debug=True)
```

5. Realizamos as petições POST

...

```
@app.post("/save-user")  
def save_user():  
    username = request.json['username']  
    password = request.json['password']  
    user = User(username, password)  
    db.session.add(user)  
    db.session.commit()  
    return user_schema.dump(user)
```

```
if __name__ == '__main__':  
    app.run(debug=True)
```

6. Creamos a base de datos en pypmyadmin chamada login. Executamos o noso código e diriximonos a Postman para probar que funcione.

7. Añadimos na clase User a liña de *tablename* para definir o nome da nosa táboa

```
class User(db.Model):
    __tablename__ = 'users'
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(150))
    password = db.Column(db.String(160))

    def __init__(self, username, password):
        self.username = username
        self.password = password
```

8. Diriximonos a pypmyadmin e creamos a táboa users con 3 campos (id, username e password)

9. Hasheamos o contrasinal

```
@app.post("/save-user")
def save_user():
    username = request.json['username']
    password = request.json['password']

    password_hash = generate_password_hash(password)

    user = User(username, password)
    db.session.add(user)
    db.session.commit()
    return user_schema.dump(user)
```

10. Probamos en Postman que funcione e mostre o contrasinal hasheado.

11. Creamos o POST de login

```
@app.post("/login")
def login():
    username = request.json['username']
    password = request.json['password']

    user = User.query.filter_by(username = username).one_or_none()

    if user is not None and check_password_hash(user.password, password):
        return jsonify({"success": "USUARIO AUTORIZADO"})
    else:
        return jsonify({"error": "UNAUTHORIZED"})
```

12. Ejecutamos. Probamos en Postman que funcione correctamente.