Criptografia

O **MD5** (Message-Digest Algorithm 5) é um algoritmo de hash amplamente usado que converte dados em um hash de 128 bits (geralmente representado como uma sequência hexadecimal de 32 caracteres). Ele foi projetado para ser rápido e é comumente utilizado para gerar checksums para verificar a integridade de arquivos.

Neste projeto o MD5 foi implementado para criptografar as senhas dos usuários.

Na rota ***Register***, onde o usuário pode criar uma conta, o md5 foi utilizado para criptografar a senha que o usuário inserir e armazenar no banco de dados.

Na rota ***Login***, a senha que o usuário inserir será criptografada e comparada com a senha do banco de dados para garantir que a senha inserida é a mesma para aquele respectivo usuário.

```python
@app.route('/login', methods=['POST'])
def login():
    data = request.json
    email = data.get("email")
    password = data.get("password")

    db = connect_db()
    cursor = db.cursor()

    cursor.execute('SELECT * FROM adminUser WHERE email = ?', (email,))
    result = cursor.fetchone()

    if result == None:
        return jsonify({"message": "email não cadastrado"}), 400

    email_db = result[1]
    password_db = result[2]
    password_md5 = hashlib.md5(password.encode()).hexdigest()

    db.close()

    if password_db == password_md5:
        return jsonify({"message": "Login realizado com sucesso"}), 200
    else:
        return jsonify({"message": "Login não realizado, verifique suas credenciais"}), 400

@app.route('/register', methods=['POST'])
def login_register():
    data = request.json
    email = data.get("email")
    password = data.get("password")

    password_md5 = hashlib.md5(password.encode()).hexdigest()

    db = connect_db()
    cursor = db.cursor()
    cursor.execute('INSERT INTO adminUser (email, password) VALUES (?, ?)', (email, password_md5))
    db.commit()
    db.close()

    return jsonify({"message": "Registro concluído"}), 200
```