

**Curso Técnico Superior Profissional em:** Tecnologias e Programação de Sistemas de Informação

**1.º Ano/ 2.º Semestre**

**Unidade Curricular:** Desenvolvimento Web - Back-End

**Docente:** David Jardim

---

## FICHA DE TRABALHO 11

---

### Exercícios:

O objetivo desta ficha é implementar autenticação utilizando JSON Web Tokens (JWT)

1. Efetue download do template disponibilizado no Moodle
2. Na pasta *models* defina o modelo *Users* que possui as seguintes colunas:
  - a. *id* (int auto-increment)
  - b. *password* (var char)
  - c. *email* (var char)
3. Crie um ficheiro *sequelize.js* e implemente todo o código necessário para efetuar a ligação à base de dados e inserir alguns utilizadores
4. Crie um ficheiro *.env* para armazenar as variáveis de ambiente utilizadas para criar a ligação à base de dados e altere o código para utilizar essas variáveis
5. Crie um novo ficheiro *token.js* e utilize o *crypto-js* crie um *token-secret* e guarde-o no ficheiro *.env*
6. Na rota **index** implemente os endpoints que vão efetuar o render das views (*login*, *signup*), essas views recebem um objeto contendo uma mensagem **connect-flash**
7. No **IndexController** implemente uma função para gerar o token de acesso utilizando a função **sign** do JWT
8. Na rota **index** implemente um endpoint (POST) **signup** para criação de novos utilizadores via formulário WEB que serão adicionados à base de dados
9. No **IndexController** implemente a função que será invocada pelo endpoint anterior para criar um novo utilizador na base de dados, gerar um token e armazenar esse token na sessão
10. Na rota **index** implemente um endpoint (POST) **login** para permitir que os utilizadores efetuem autenticação via formulário WEB
11. No **IndexController** implemente a função que será invocada pelo endpoint anterior para autenticar um utilizador na base de dados, gerar um token e armazenar esse token na sessão
12. Implemente um endpoint (GET) **profile** para mostrar o perfil do utilizador após ter sido autenticado. Este endpoint é de acesso restrito a pedidos com um token de acesso válido armazenado na sessão

Cofinanciado por:

13. Implemente uma nova rota com acesso protegido a nível do middleware com verificação por cabeçalho (users) e um novo controller (usersController) para disponibilizar toda a lógica de CRUD em forma de api tal como descrito na tabela abaixo

| URI        | Método HTTP | Body        | Resultado                   |
|------------|-------------|-------------|-----------------------------|
| /users     | GET         | empty       | Show list of all the users. |
| /users     | POST        | JSON String | Add details of new user.    |
| /users/:id | DELETE      | empty       | Delete an existing user.    |
| /users/:id | GET         | empty       | Show details of a user.     |
| /users/:id | PUT         | JSON String | Update details of a user    |

*Tabela 1 - Endpoints para o recurso User*

Cofinanciado por:

