



UNIVERSIDADE DE ÉVORA
CURSO DE ENGENHARIA INFORMÁTICA
2024/2025

Tecnologias Web

→ Relatório ←

Carpooling Évora

Miguel Aleixo - I51653
Mariana Cavaco - I51820

Objetivo do Trabalho

Este trabalho foi desenvolvido como resposta ao enunciado do segundo trabalho prático da unidade curricular de Tecnologias Web. O objetivo foi criar uma plataforma de carpooling/boleias para a comunidade académica da Universidade de Évora, com gestão de viagens e pedidos de transporte.

O sistema implementado inclui funcionalidades de autenticação, gestão de utilizadores, viagens e pedidos, com uma interface responsiva e integração com uma base de dados PostgreSQL.

Solução Implementada

A solução foi desenvolvida utilizando as seguintes tecnologias:

- **Spring Boot:** Framework principal para o backend.
- **Spring JPA:** Gestão das interações com a base de dados.
- **PostgreSQL:** Base de dados para armazenamento persistente.
- **HTML/CSS:** Interface do utilizador.

Funcionalidades Implementadas

A nossa solução dispõe das seguintes classes:

- **AdminController** → Gere as ações administrativas, como ativar e desativar os utilizadores.
- **AdminInitializer** → Inicializa configurações específicas do administrador.
- **AdminRepo** → Interage com a base de dados para operações relacionadas ao administrador.
- **AcessControllers** → Gere a autenticação e as permissões dos utilizadores.
- **Admin** → Representa um administrador da aplicação.
- **Pedido** → Representa um pedido de transporte associado a uma viagem.
- **Rotas** → Modela as rotas disponíveis para as viagens.
- **Utilizador** → Representa os utilizadores da aplicação, podendo ser passageiros ou condutores.
- **Viagem** → Representa uma viagem criada por um condutor.
- **UserAlreadyExistsException** → Exceção lançada quando um utilizador já existe no sistema.
- **UsernameNotFoundException** → Exceção lançada quando o username não é encontrado.
- **PedidoRepo** → Gerencia as operações relacionadas aos pedidos na base de dados.
- **RotasRepo** → Manipula as rotas armazenadas na base de dados.
- **UtilizadorRepo** → Gerencia dados dos utilizadores.
- **ViagemRepo** → Gerencia as viagens criadas e armazenadas.

- **PedidoService** → Lida com a lógica de negócios para pedidos de transporte.
- **RotasService** → Gere operações das rotas.
- **UtilizadorService** → Gere operações relacionadas aos utilizadores.
- **ViagemService** → Lida com a lógica dos negócios para a criação e gerenciamento de viagens.
- **SecurityConfig** → Configura o Spring Security para autenticação e autorização.
- **UserAuthService** → Carrega os dados do utilizador durante o processo de autenticação

Páginas HTML

- **admin** → Página onde são mostradas as opções de ativar e desativar contas de utilizadores que se auto registam, possuindo também a visualização da lista de utilizadores existentes na aplicação.
- **condutor** → Página dos recursos que o condutor pode acessar.
- **entrar** → Página de login do site.
- **index** → Página inicial do Passageiro
- **newuser** → Página inicial antes da autenticação.
- **passageiro** → Página dos recursos que o passageiro pode acessar.
- **perfilcondutor** → Página do perfil do condutor.
- **perfilpassafeiro** → Página do perfil do passageiro
- **register** → Página registo do site.

Descrição do Funcionamento do Programa

Configuração Inicial

- Editar os ficheiros `application.properties` para conectar à base de dados.

Execução

- Executar o servidor com `mvn spring-boot:run`.
- Aceder à aplicação em `http://localhost:8080`.

Conclusão

A aplicação desenvolvida cumpre os objetivos principais, oferecendo uma solução funcional para gestão de carpooling, com autenticação segura, gestão de utilizadores, viagens e pedidos. A integração com PostgreSQL garantiu a persistência dos dados, e o uso de Spring Security reforçou a segurança da aplicação.