

# Trabalho Prático 2 – Aplicação Web em Jakarta EE



Departamento de  
Informática

## T4 – Agência de aluguer de automóveis



Elaborado por:

Bernardo Claro - 46879

Tiago Barreiros - a46118

Orientadora:

Professora Doutora Paula Prata

28 de maio de 2023, Covilhã

# Índice

<b>Introdução .....</b>	<b>3</b>
<b>Modelo de Dados .....</b>	<b>4</b>
<b>Descrição da arquitetura e funcionalidades da aplicação .....</b>	<b>5</b>
<b>Manual de Configuração e Instalação.....</b>	<b>6</b>
<b>Trabalho Futuro.....</b>	<b>7</b>
<b>Conclusão .....</b>	<b>7</b>

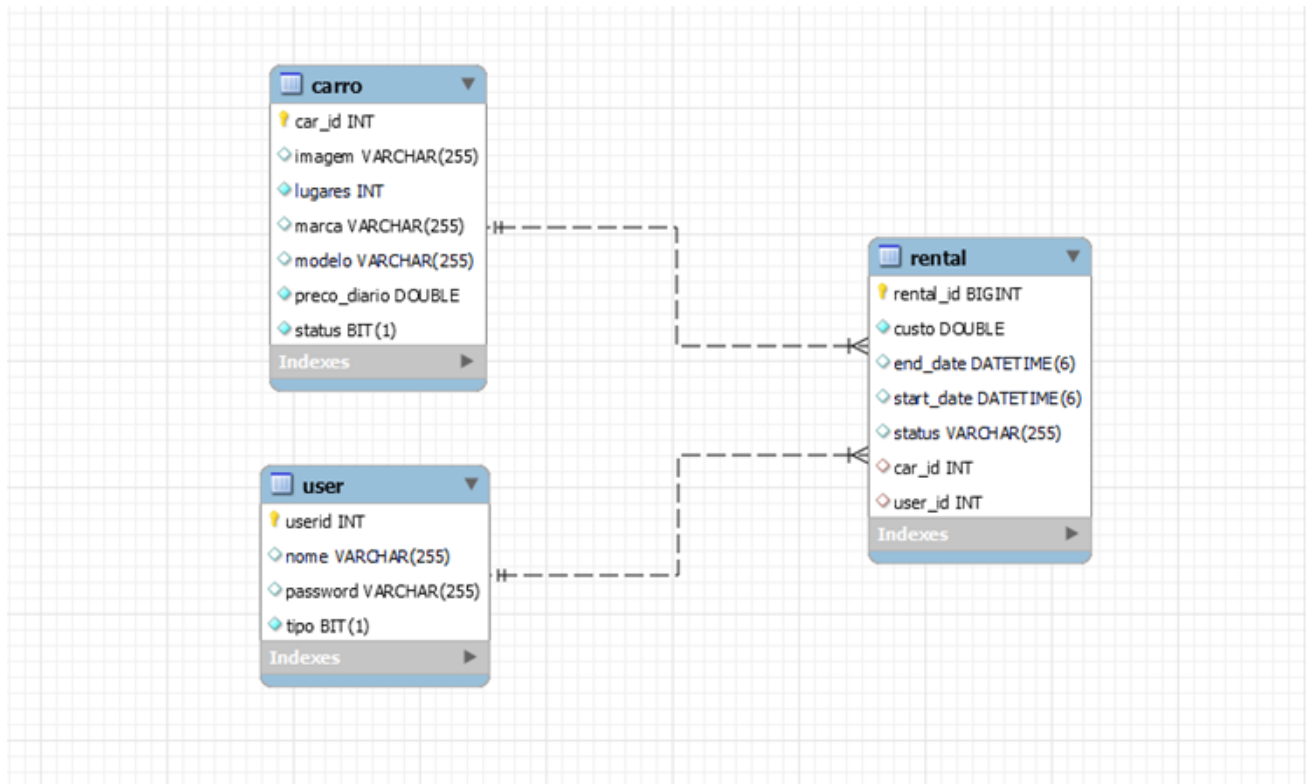
# Introdução

Este curto relatório foi elaborado por alunos do 3.º ano da Licenciatura de Engenharia Informática (EI) da Universidade da Beira Interior (UBI). Este foi desenvolvido no contexto da Unidade Curricular, Sistemas Distribuídos (SD) da UBI, através de pesquisas pessoais e material pedagógico providenciado pela Professora da própria UC.

Procuramos desta maneira refletir a forma como realizamos e procedemos à implementação deste trabalho, explicar e explorar as funcionalidades oferecidas, bem como a modelagem das mesmas.

Por fim, preparamos um pequeno manual de configuração e instalação para agilizar todo o processo.

## Modelo de Dados



- Relação 1-N entre **Carro** e **Rental**;
- Relação 1-N entre **User** e **Rental**;
- O 'car\_id' e o 'user\_id' vão ser – por consequência – chaves estrangeiras da tabela Rental.

## Os utilizadores subdividem-se nos seguintes papéis:

- O administrador pode adicionar e remover utilizadores, carros e reservas. Pode também alterar o estado de uma reserva e ver todas as estatísticas sobre a aplicação.
- O cliente apenas pode reservar os seus automóveis e cancelar a reserva em caso de a mesma ainda se encontrar em processamento.

# Descrição da Arquitetura e Funcionalidades da Aplicação

- Autenticação e criação de sessões : Para usufruir da aplicação o utilizador tem de ser registar ou fazer login no caso de já ter conta. Ao realizar o login uma sessão será criada usando o Package Httpsession que permite identificar o utilizador enquanto este navega na aplicação, além de permite guardar informação sobre o mesmo. A partir do momento que dá login e até dar logout, cada vez que aceder à aplicação não necessitará de se autenticar novamente.
- Controllers: Esta package compreende todos os controladores designados para a funcionalidade da aplicação Web.
  - Neste caso a aplicação só contém um controlador que é o mainController, este é responsável por:
    - Tratamento de operações de login;
    - Tratamento de operações de registo;
    - Tratamento de operações de Reserva;
    - Redireciona o utilizador para a página que lhe está atribuída de acordo com o seu papel na aplicação;
- Entities: Este package engloba as entities da nossa aplicação:
  - Carro: Entity que representa a tabela dos carros disponíveis para reserva bem como todos os seus atributos.
  - User: Entity que representa a tabela dos utilizadores da aplicação bem como as relações da mesma;
  - Rental: Entity que representa a tabela das reservas bem como as relações da mesma;
- Repository: Este package abarca diversas interfaces que efetuam operações CRUD(Create, Read, Update, Delete). As interfaces subdividem-se nas seguintes:
  - CarRepository: Interface que possui métodos que permitem operações relativas aos carros;
  - RentRepository: Interface que possui métodos que permitem operações relativas às reservas;
  - UsersRepository: : Interface que possui métodos que permitem operações relativas aos utilizadores;
- resources: Este package contém os ficheiros .html que são utilizados na aplicação desenvolvida. Para a elaboração dos ficheiros .html foi escolhida a framework bootstrap para fornecer um design mais consistente e equilibrado do front-end da aplicação.

# Manual de Configuração e Instalação

- Comece por extrair o zip com o projeto Java EE com as seguintes dependências base: **Spring Web, Spring Boot Dev Tools, Thymeleaf, Spring Data JPA, MySQL Driver**. Deve usar uma versão igual ou superior ao **JDK 17**.
- Crie a sua Base de Dados local e um User com privilégios, através dos seguintes comandos:
  - `mysql> create database [NOME];`
  - `mysql> create user 'NOME'@'%' identified by 'SENHA';`
  - `mysql> grant all on [NOME_db].* to 'NOME_USER'@'%';`
- Altere o ficheiro "application.properties" com os teus dados:

```
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:3306/
db_ProjetoSD
spring.datasource.username=[NOME]
spring.datasource.password=[SENHA]
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

- Deve conseguir executar a sua aplicação Java Spring Boot;
- Aceda ao website no URL de um browser: **`http://localhost:8080`**.

## Trabalho Futuro

Pretendemos melhorar este trabalho num futuro próximo. Foram poucos dias e muito trabalho acrescido.

## Conclusão

Em resumo, o projeto que desenvolvemos foi um Sistema de Reserva de Automóveis, que permite aos utilizadores efetuar reservas de carros disponíveis. Através da implementação de um controlo de acessos, foi possível garantir a autenticação e privacidade dos dados. Utilizamos tecnologias como Spring Boot, onde foi possível criar um aplicativo web dinâmico e eficiente. O sistema inclui recursos como visualização e pesquisa de carros disponíveis, reservas, atualização de informações e administração geral. Durante o desenvolvimento, aprendemos bastante sobre a estruturação de um projeto Java Spring, a sua integração com a base de dados e a criação de consultas personalizadas. Ao finalizar o trabalho, sentimos-nos satisfeitos com o resultado alcançado e confiantes nas habilidades adquiridas ao longo do processo.