

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

MESTRADO EM ENGENHARIA INFORMÁTICA

Assignment #2

INTEGRAÇÃO DE SISTEMAS

Diogo Cruz - 2018306709

e

Fábio Ferreira - 2015229929



15 de novembro de 2021

Conteúdo

Índice	I
1 Introduction	1
2 Presentation Tier	2
3 Business Tier	5
4 Data tier	6
5 Project Management and Packaging	8

1 Introduction

No âmbito da cadeira de *Integração de Sistemas* fizemos uma aplicação web que consistia em gerir uma companhia de autocarros. Cumprimos todos os requisitos enunciados pelo professor, incluindo ainda os recursos adicionais (hash passwords, login tool, good structure, etc).

Para correr o projeto é necessário correr dentro do container command-line:

```
1 (cmd) cd workspace/assignment2\_is/jeeapp
2 (cmd) mvn clean package install wildfly:deploy
```

2 Presentation Tier

Para a presentation tier colocamos tudo dentro de um módulo ao que chamamos web.

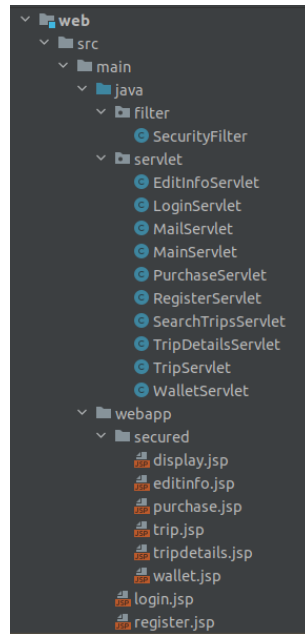


Figura 2.1: Estrutura da presentation layer.

Dentro desse módulo dividimos a estrutura pelas tecnologias que utilizamos, servlet e jsp.

No que concerne ao login e registo utilizamos um filtro para filtrar todo o tipo de acesso sem sessão para estas paginas, ou seja, quem não tinha sessão ativa não conseguia acessar as paginas que se encontram dentro da diretoria secured.

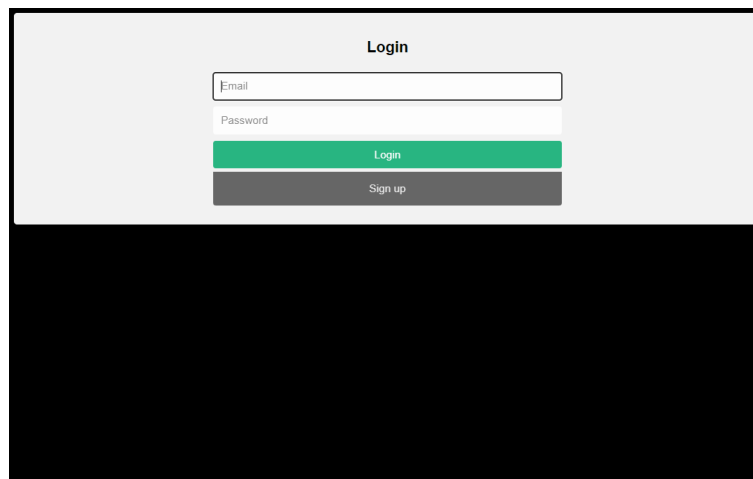
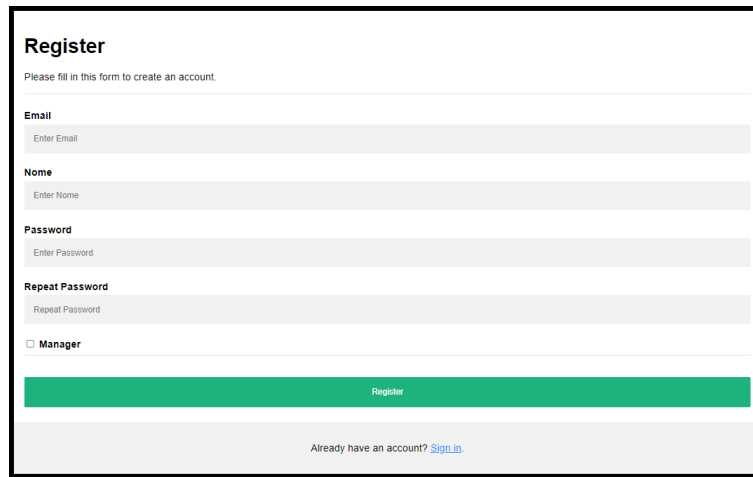


Figura 2.2: Página do Login.



Register

Please fill in this form to create an account.

Email
Enter Email

Nome
Enter Nome

Password
Enter Password

Repeat Password
Repeat Password

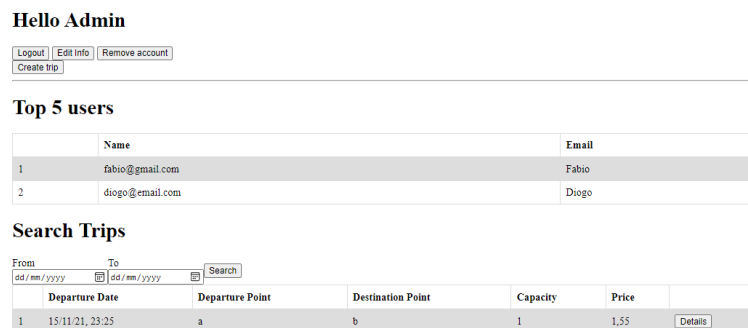
☐ **Manager**

Register

Already have an account? [Sign in](#)

Figura 2.3: Página do Registo.

Após o login feito é atribuída uma sessão ao user, onde inclui o atributo de este ser manager ou não. Utilizamos uma biblioteca de jsp,jstl. Esta biblioteca serve para utilizarmos foreach para iterar dentro de listas, scope para mostrar os componentes que desejávamos, por exemplo ao manager aparece create trip ao utilizador normal não aparece, entre outras funcionalidades.



Hello Admin

Logout | Edit info | Remove account
Create trip

Top 5 users

	Name	Email
1	fabio@gmail.com	Fabio
2	diogo@email.com	Diogo

Search Trips

From To Search
dd/mm/yyyy dd/mm/yyyy

	Departure Date	Departure Point	Destination Point	Capacity	Price	
1	15/11/21, 23:25	a	b	1	1,55	Details

Figura 2.4: Interface do manager.

Hello Fábio

[Logout](#) [Edit Viallet](#) [Edit info](#) [Remove account](#)

My Trips

	Departure Date	Departure Point	Destination Point	Capacity	Place	Price	
1	15/11/21, 23:25	a	b	1	1	1,55	Delete
2	15/11/21, 23:25	a	b	1	1	1,55	Delete

Search Trips

From

To

dd/mm/yyyy

dd/mm/yyyy

Search

Figura 2.5: Interface do cliente.

Estas sao as paginas principais que sao demonstradas aos tipos de user.

Penso que as maiores dificuldades foram em certos bugs de codigo como por exemplo a falta de uma chaveta dentro de um jsp.Depois dividir em servlets diferentes e a sua maneira de acessar esses survlets.

Relativamente as passwords,estas são armazenadas na base de dados sobre um mecanismo que as encripta logo que o registo é feito.

O envio de emails tambem e feito nesta camada dentro dos servlets,utilizando o protocolo smtp com ssl para o efeito.

Esta camada acede a camada business utilizado os ejbs e permitindo mostrar assim todo o conteúdo ao utilizador.

3 Business Tier

A Business Tier é a camada responsável por fazer a ponte entre a Data Tier e a Presentation Tier.

Esta camada foi desenvolvida com recurso a EJBs (*Enterprise JavaBeans*). Para transferência de dados entre as camadas de negócio e apresentação recorremos a DTOs (*Data Transfer Objects*).

A estrutura deste módulo é apresentada na figura seguinte:

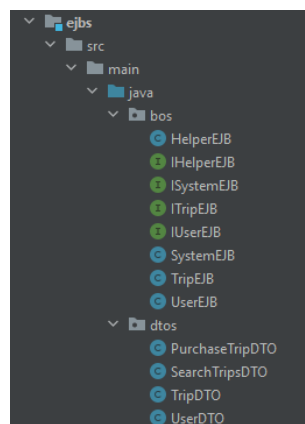


Figura 3.1: Estrutura da business tier.

Entre os vários EJBs presentes, temos o TripEJB e o UserEJB. Estes *java beans* são *Stateless* para que sejam instanciados a cada pedido feito ao servidor. Neles estão presentes os métodos de CRUD das várias entidades bem como outros métodos adicionar de consulta e manipulação de dados. Em cada um desses métodos é instanciado um DAO (*Data Access Object*) presentes na Data Tier (Ver secção 4), que por sua vez contem os métodos manipulação das entidades. Em todos os métodos são retornados DTOs.

Para além dos EJBs descritos anteriormente, foram desenvolvidos dois EJBs adicionais, SystemEJB e HelperEJB, responsáveis por notificar via email os utilizadores do tipo *manager* sobre o sumário de vendas do dia e encriptar as passwords dos managers criados via script no primeiro pedido feito ao servidor, respectivamente. Estes EJBs são *Singleton* uma vez que desempenham tarefas do sistema e são transversais a todo o runtime da aplicação.

Todos os EJBs implementam uma interface, usada para injetar estes objectos na camada de apresentação.

4 Data tier

A Data Tier foi desenvolvida com recurso à framework *Hibernate*, uma implementação da JPA (*Java Persistence API*), utilizada para gestão da base de dados.

Segue o diagrama ER da base de dados desenvolvida e a estrutura de pastas deste módulo:

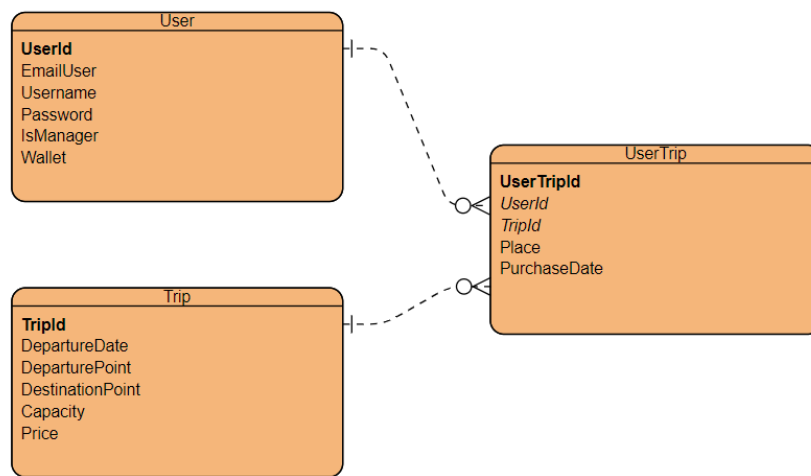


Figura 4.1: Diagrama ER.

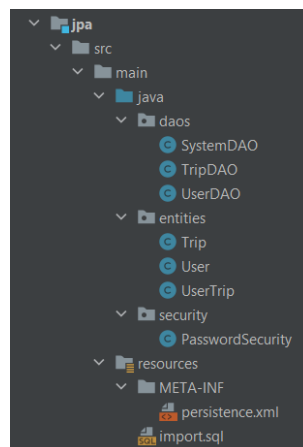


Figura 4.2: Estrutura da data tier.

Como demonstrado na figura acima, para além das entidades, foram também desenvolvidos DAOs (*Data Access Objects*) que contêm todos os métodos com as *queries* de consumo e manipulação das entidades. Estes são utilizados pelos EJBs da business tier (Ver Secção 3).

Para a criação dos *managers* na BD, recorreremos ao script `import.sql` que é executado via SQL Injection no momento do arranque criação da base de dados.

5 Project Management and Packaging

Dividimos o projeto em 4 módulos, este inclui um módulo ear que é responsável por interligar os restantes módulos, inclui um módulo de ejbs que é responsável pela camada de negócio, inclui um módulo de jpa que é responsável pela camada de dados e por fim inclui um módulo web que concerne à camada de apresentação.

Dentro destes módulos e fora encontra-se os pom.xml que contêm as dependências necessárias para o bom funcionamento do projeto.

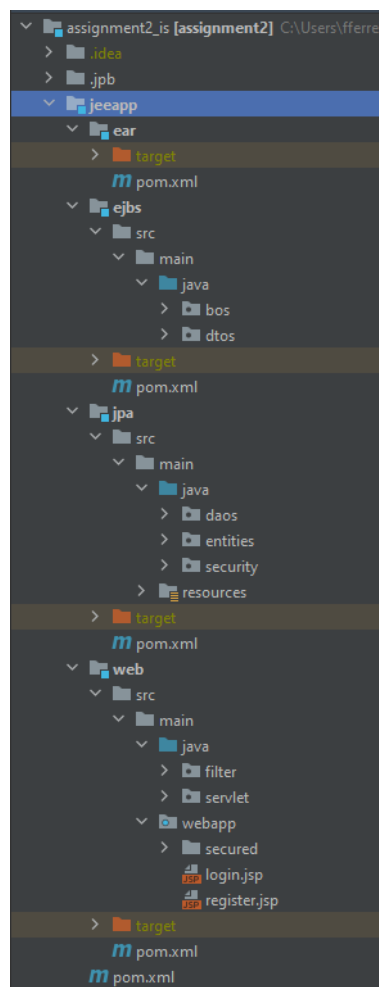


Figura 5.1: Estrutura do projeto.