Integração de Sistemas

Projeto 2

Introdução

O Java Enterprise Edition (também conhecido por Java EE) é uma plataforma de programação Java destinada a servidores de aplicações. O Java EE surge devido à necessidade de criação de aplicações empresariais multicamada em sistemas distribuídos. De entre as várias vantagens para a sua utilização, destacam-se a portabilidade, a escalabilidade e a segurança das aplicações produzidas. A plataforma fornece uma API modular e um ambiente de execução para o desenvolvimento e execução de ferramentas corporativas incluindo serviços de rede e serviços web.
Neste trabalho desenvolveu-se uma aplicação web para gerir uma plataforma de aprendizagem online, utilizando o Java Persistence API (JPA) e o Java Persistence Query Language (JPQL), o Enterprise Java Beans (EJB) e Servlets/JSP Pages. A aplicação está dividida em quatro camadas distintas, a camada de dados (JPA), a camada de negócio (CRUD), a camada de negócio (EJB) e a camada web (Servlets/JSPs). O servidor de aplicações utilizado foi o WildFly utilizando o Hibernate como mapeador relacional de objetos. A atividade dos diversos utilizadores dentro do sistema é monitorizada recorrendo ao logging (utilizando o sl4j) tanto na camada de negócio como na camada web.
Este relatório está dividido em quatro partes, em que cada uma das partes descreve as decisões de projeto tomadas para cada uma das quatro camadas.

Camada de dados (JPA)

O modelo de dados do projeto é composto por seis entidades: Administrador, Pessoa, Professor, Aluno, Disciplina e Material. Para cada uma destas entidades (ou classes) foram definidos vários atributos e identificadas as relações necessárias para responder às necessidades do problema apresentado. Cada classe contém também um conjunto de métodos getters e setters de forma a possibilitar a inserção e a obtenção dos atributos utilizando o Java Persistence API. De seguida são referidos os atributos e as relações definidas para cada uma das entidades da camada de dados.

<u>Administrador</u>

A entidade Administrador é constituída pelos seguintes atributos: id (gerado automaticamente), nome, email institucional e password. O atributo email institucional foi colocado como único (unique=true) para garantir que não sejam criadas várias contas de administrador com o mesmo email institucional, que é utilizado para a identificação do administrador no sistema.

<u>Pessoa:</u>

A entidade Pessoa é uma superclasse e é constituída pelos seguintes atributos: id (gerado automaticamente), nome, data nascimento, email institucional, password, email alternativo, morada, telefone. O atributo email institucional foi colocado como único (unique=true) para garantir que não sejam criadas várias contas de professor/aluno com o mesmo email institucional, que é utilizado para a identificação do professor/aluno no sistema.

Professor

A entidade Professor é uma classe descendente da classe pessoa e é constituída pelos seguintes atributos: categoria, gabinete, telefone interno, salario e uma lista de disciplinas.
Quanto às relações, foi aplicada uma relação OneToMany relativamente à classe disciplina, uma vez que um professor pode lecionar várias disciplinas.

Aluno

A entidade Aluno é uma classe descendente da classe pessoa e é constituída pelos seguintes atributos: ano entrada curso e uma lista de disciplinas.
Quanto às relações, foi aplicada uma relação ManyToMany relativamente à classe disciplina, uma vez que um aluno pode pertencer a várias disciplinas e uma disciplina pode ter vários alunos.

Disciplina

A entidade Disciplina é constituída pelos seguintes atributos: id (gerado automaticamente), nome, informações, um professor, uma lista de alunos e uma lista de materiais.
Quanto às relações, foi aplicada uma relação ManyToOne relativamente à classe professor, uma vez que cada disciplina tem apenas um professor, mas cada professor pode lecionar várias disciplinas. Uma relação ManyToMany relativamente à classe aluno, uma vez que uma disciplina pode pertencer a vários alunos e um aluno pode ter várias disciplinas. Por último uma relação OneToMany relativamente à classe material, uma vez que uma disciplina pode ter vários materiais.

Material

A entidade Material é constituída pelos seguintes atributos: id (gerado automaticamente), nome ficheiro, diretoria e uma disciplina.
Quanto às relações, foi aplicada uma relação ManyToOne relativamente à classe disciplina, uma vez que cada material tem apenas uma disciplina, mas cada disciplina pode ter vários materiais.
O upload dos materiais é feito diretamente para o disco, para uma pasta com o nome MateriaisIS previamente criada dentro do diretório System.getProperty("user.dir").

As passwords das classes administrador e pessoa guardadas na base de dados são um hash md5 não invertível da password utilizada no registo. Deste modo caso a base de dados seja comprometida a informação relativa aos utilizadores do sistema não é comprometida. Na figura abaixo, encontra-se um esquema sumário das entidades e das suas respetivas relações.

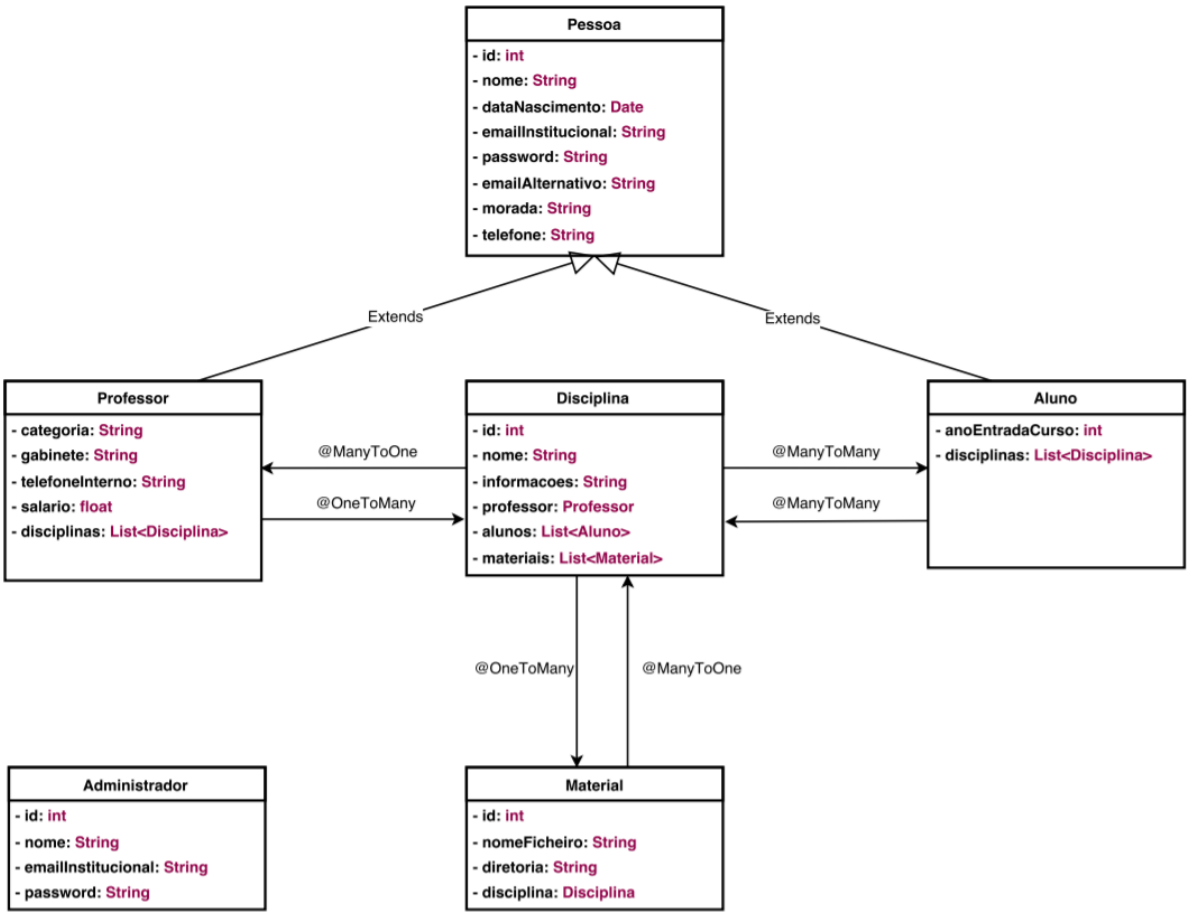

Figura 1 – Diagrama de classes da base de dados

Camada de negócio (CRUD)

O CRUD é o componente do lado do servidor que encapsula a lógica de negócio, ou seja, o código que responde ao objetivo do pedido. Optámos por criar uma bean stateless que não guarda o estado, porque como os pedidos são independentes e como a verificação do utilizador é feita na sessão da camada web (através de cookies do brownser) não foi necessário a criação de uma bean statefull. De forma a evitar o SQL Injection todos os parâmetros foram introduzidos nas queries através do método setParameter(). Deste modo asseguramos a segurança da camada de dados ao não permitir a introdução direta de parâmetros nas queries. É importante referir que a própria bean já inclui um modelo de transações, sendo que o Hibernate trata essas transações de forma transparente. Por este motivo, as transações da bean com a base de dados não foram geridas no código produzido neste trabalho. De seguida apresentamos uma listagem dos métodos definidos na interface da bean:

Métodos gerais necessários ao funcionamento do sistema

int login(), boolean loginScript(), String hashMD5(), Date getDate(), List<Material> infoAluno(), String verificarMaterialPertenceAluno()

Métodos definidos para lidar com as operações relativas ao administrador

boolean registarAdministrador(), boolean verificarEmailAdministradorExiste(), boolean removerAdministrador(), List<Administrador> obterTodosAdministradores()

Métodos definidos para lidar com as operações relativas ao aluno

boolean registarAluno(), boolean verificarEmailAlunoExiste(), Aluno obterAluno(), boolean editarAluno(), boolean removerAluno(), List<Aluno> obterTodosAlunos(), int obterIdAluno()

Métodos definidos para lidar com as operações relativas ao professor
boolean registarProfessor(), boolean verificarEmailProfessorExiste(), Professor obterProfessor(), boolean editarProfessor(), boolean removerProfessor(), List<Professor> obterTodosProfessores(), int obterIdProfessor()
Métodos definidos para lidar com as operações relativas à disciplina
boolean registarDisciplina(), Disciplina obterDisciplina(), boolean editarDisciplina(), boolean removerDisciplina(), boolean inscreverAluno(), boolean desinscreverAluno(), List<Disciplina> obterTodasDisciplinas()
Métodos definidos para lidar com as operações relativas ao material, pesquisas de alunos e listagens específicas
List<Disciplina> obterTodasDisciplinasProfessor(), List<Material> obterTodosMateriaisDisciplina(), String verificarDisciplinaPertenceProfessor(), boolean uploadMaterial(), String removerMaterial(), List<Aluno> obterTodosAlunosOrdemCrescente(), List<Aluno> obterTodosAlunosOrdemDecrescente(), String pesquisarAlunoId(), String pesquisarAlunoNome(), String pesquisarAlunoEmailInstitucional()

Camada de negócio (EJB)
O EJB é o componente do lado do servidor que faz a ponte entre a camada de dados e camada de web, evitando que o CRUD seja acedido diretamente a partir a camada web. Para além disso transforma as entidades em objetos DTO (objetos de transferência de dados) antes de serem enviados para a camada web.

Camada web (Servlets/JSPs)
A camada web foi desenvolvida recorrendo a servlets e a páginas JSP. As servlets servem para processar as ações do utilizador, redirecionar o utilizador para uma determinada página e funciona ainda como fonte de informação (backend), enquanto que as páginas JSP servem para apresentar a informação ao utilizador (frontend). Para tornar o frontend visualmente apelativo foi utilizada a framework de CSS, Bootstrap. Os EJBs são invocados a partir das servlets, e as servlets devolvem informação ás páginas JSP através da session e do request. O sistema verifica de forma robusta se um utilizador está autenticado ou não. Quando um utilizador faz login na aplicação, o id do utilizador é adicionado à sessão. Sempre que um utilizador faz um pedido às servlets ou às páginas JSP verifica-se se o id existe ou não na sessão, no caso de não existir a informação não é apresentada na página nem a servlet processa o pedido e o utilizador é redirecionado para a página de login. Para além do controlo da sessão foi feita também a validação de todos os dados de entrada inseridos pelo utilizador, como por exemplo se o email contém um @, se todos os campos do formulário foram preenchidos, ... No caso de haver dados inválidos ou um erro no processamento do pedido uma mensagem de erro apropriada é apresentada ao utilizador.

Na tabela abaixo apresentamos todas as servlets e todas as páginas JSP utilizadas. O nome de cada um dos ficheiros é esclarecedor da sua função.

Servlets (backend)	Páginas JSP (frontend)
/Login	/Login.jsp
/Logout	/MenuAdminAlunos.jsp
/GerirAluno (registar, remover, editar)	/MenuAdminDisciplinas.jsp
/GerirProfessor (registar, remover, editar)	/MenuAdminProfessores.jsp

/GerirDisciplina (registar, remover, editar, inscrever aluno, desinscrever aluno)	/MenuProfessorAlunos.jsp
/AdicionarMaterial	/MenuProfessorDisciplinas.jsp
/RemoverMaterial	/MenuProfessorPesquisas.jsp
/DownloadMaterial	/MenuAluno.jsp
/SelecionarDisciplinaMateriais	
/ SelecionarDisciplinaListagem	
/ListarAlunosDisciplina	
/PesquisarAluno	

Tabela 1 – Servlets (backend) e páginas JSP (frontend)

De seguida apresentamos 2 screenshots do frontend da aplicação web.

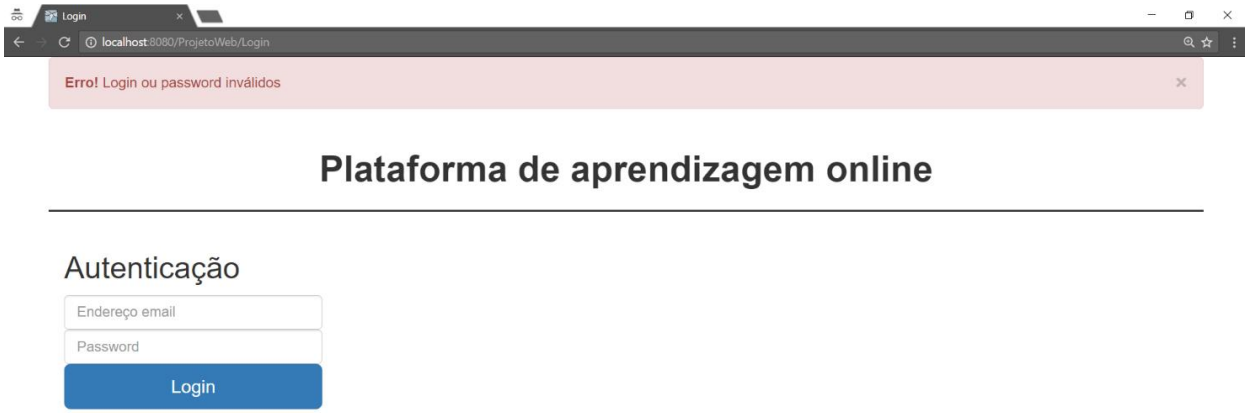

Figura 2 – Página inicial de login

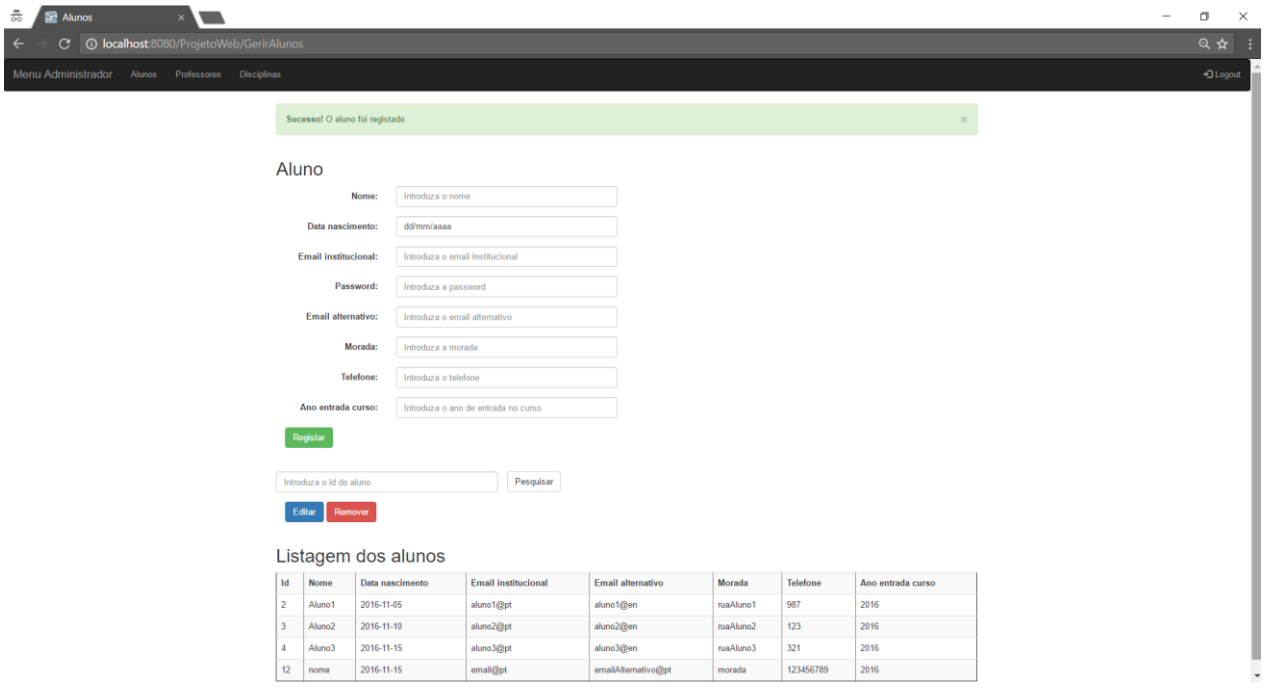

Figura 3 – Menu do administrador (registo de um novo aluno)