



Relatório da fase 5 do projeto de CSS

Grupo CSS 03:

Carlota Filipe nº 51027

Maria Fernandes nº 51097

Leonor Candeias nº 51057

Camada de negocio

Definimos os handlers chamados pelos services e os catalogos chamados pelos handlers como EJB.

Fizemos interfaces remotas para cada service, que tínhamos da fase anterior. Neste mudámos o tipo de retorno de void para uma das classes DTO que nós criámos nesta fase.

Quanto a essas classes DTO, que estão no pacote facade.dto, foram definidas uma para cada classe das nossas tabelas (realizadas na fase 3) menos a classe Utente, ou seja, AulaDTO, InscricaoDTO, InstalacaoDTO e ModalidadeDTO. Fizemos, ainda, interfaces para as mesmas classes, ou seja, IAula, IInscricao, IInstalacao e IModalidade.

Para haver persistência dentro do projeto, retiramos nos handlers os construtores, as linhas de código em.getTransaction().begin() e commit(). Nas classes catalogos, anotamos o EntityManager como @PersistenceContext e @Transactional(Transaction.TxType.REQUIRES_NEW) nas funções que em fases anteriores tinham entity manager begin e commit.

WEB-CLIENT

Criamos 2 models, um para cada servico remoto e 4 actions, 2 para cada um dos model.

Ligamos as actions a um path no app.properties:

- appRoot/action/inscricoes/novaInscricao = java:module/NewInscreverAulaAction
- appRoot/action/inscricoes/criarInscricao = java:module/InscreverAulaAction
- appRoot/action/visualizar/newVisualizar=java:module/NewVisualizarInstalacaoAction
- appRoot/action/visualizar/visualizar=java:module/VisualizarInstalacaoAction

Adaptamos o Front Controller para chamar as nossas Actions e Models.

Criamos 2 jsp, um para cada model, e estes forem chamados nas Actions correspondentes.

GUI-CLIENT

Fizemos 2 ficheiros fxml, com a ajuda do Scene Builder, um para cada service remote.


Criar uma nova aula:

AulaGes

Modalidade	<input type="text"/>
%label.Nome	<input type="text"/>
Dias da semana	<input type="text"/>
Hora de Inicio	<input type="text"/>
Duração	<input type="text"/>

%button.CriarAula

Ativar uma aula:



The screenshot shows a window titled "AulaGes" with a light gray background. Inside the window, there is a form with the following elements:

- A label "%label.NomeDaAula" followed by a text input field.
- A label "Data Início" followed by a date picker control (text field with a calendar icon).
- A label "Data Fina:" followed by a date picker control (text field with a calendar icon).
- A label "Nº Máximo de Alunos" followed by a text input field.
- A label "Instalação" followed by a dropdown menu.
- A button labeled "%button.AtivarAula" located at the bottom right of the form.

Adaptamos o Startup para chamar os nossos 2 controllers, um para cada service remote. Nos controllers pusemos as variáveis do tipo String e int como TextField, as do tipo Date como DatePicker, à exceção da duração e da hora de início de uma aula em que usamos TextField e depois no código do model convertemos para Date, e quando existe uma lista de modalidades ou instalações, dependendo do controller, por onde se tem de escolher entre vários dados já existente, utilizamos uma ComboBox. Fizemos os models de duas classes: Instalacao e Modalidade, e também 2 models um para cada controller.