



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE – UFRN
INSTITUTO METRÓPOLE DIGITAL – IMD
CENTRO INTEGRADO DE VOCAÇÃO TECNOLÓGICA – CIVT
DIM0547 – DESENVOLVIMENTO DE SISTEMAS WEB II

2º Entregável: implementação das camadas de apresentação (usando o framework JSF) e a camada de acesso a dados, em termos de mapeamento objeto-relacional utilizando a JPA.

Software em Análise:

Sistema de Monitoria Virtual

Equipe responsável:

Ewerton Ranielly de Sousa Cavalcante (mentor)

Fernanda Chacon Fontoura

Lucio Soares de Oliveira

Sérgio Giordanno Medeiros de Luna

NATAL/RN, 2016

1. Introdução

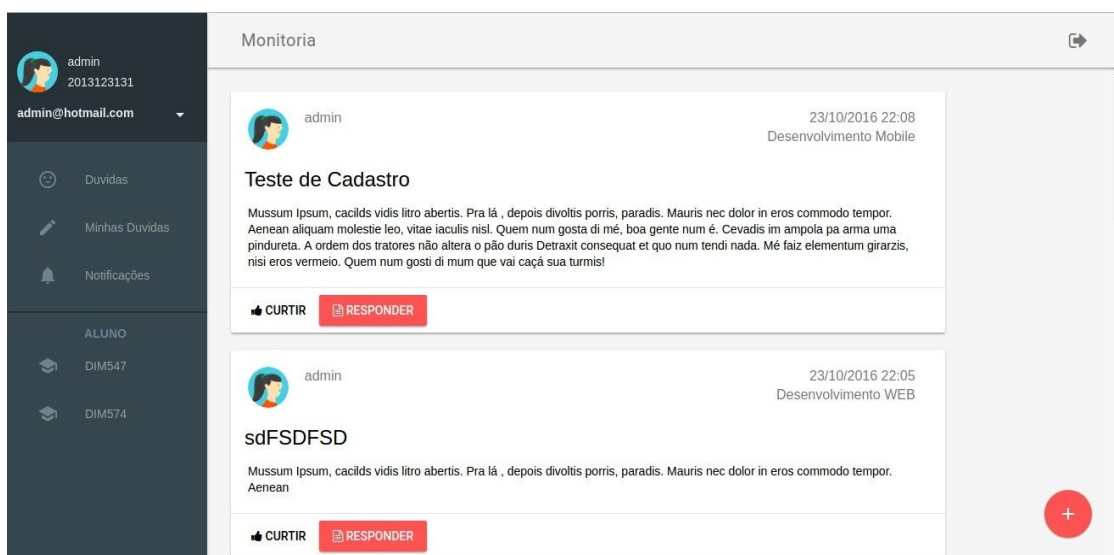
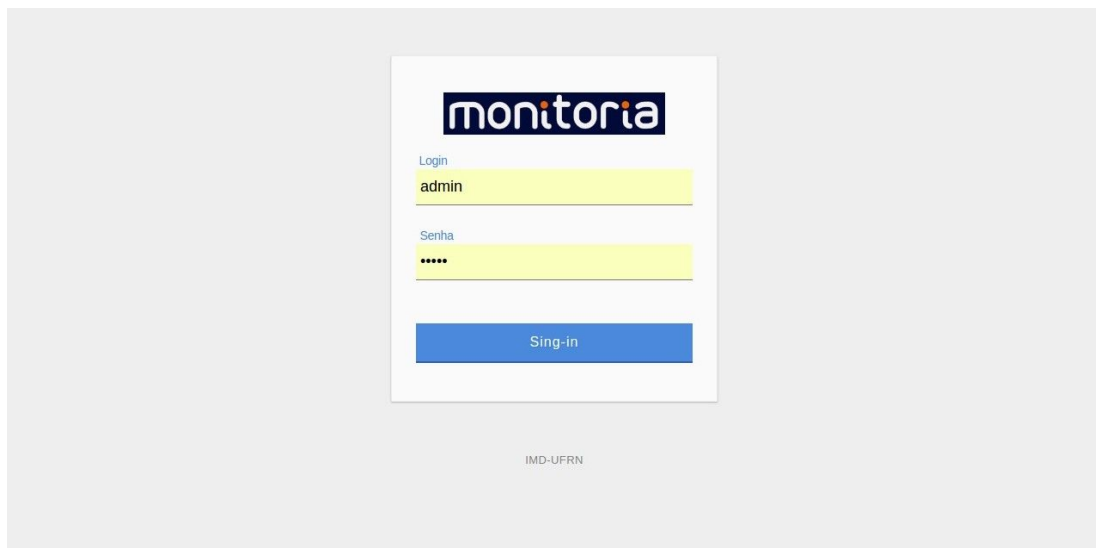
O segundo entregável consiste em implementar as camadas de apresentação, usando JSF e a camada de acesso a dados, em termos de mapeamento objeto-relacional utilizando a JPA.

Neste documento vamos apresentar como se deu essa implementação, como foi feito o mapeamento objeto-relacional com o banco de dados as classes de domínio e as relações entre elas. Vamos mostrar como foi feita a camada de apresentação com jsf e a ligação dela com as informações do banco de dados.

2. Desenvolvimento

- A estrutura da aplicação em termos de composição de páginas Web, as informações que são requeridas do usuário como entrada e as exibidas como saída, e como se dá o fluxo de navegação entre as páginas de acordo com as ações realizadas pelo usuário.

1. Primeiro passo é entrar no sistema, o usuário vai precisar de um login e uma senha (que vão ser do sigaa, quando tivermos com a conexão com a api). Depois do usuário ter feito seu login, será mostrada uma página com um menu lateral e um feed com perguntas e respostas de todas as disciplinas, e de várias pessoas. O menu tem dados do usuário e algumas opções que ele poderá escolher. A seguir temos a imagem da tela de login e em sequência a primeira tela depois de fazer o login.



2. A seguir podemos ver o menu em mais detalhes:



A primeira parte do menu são as informações pessoais do usuário.

Se o usuário clicar em “dúvidas”, vai aparecer a tela inicial que são todas as perguntas que todos os alunos fizeram em todas as disciplinas.

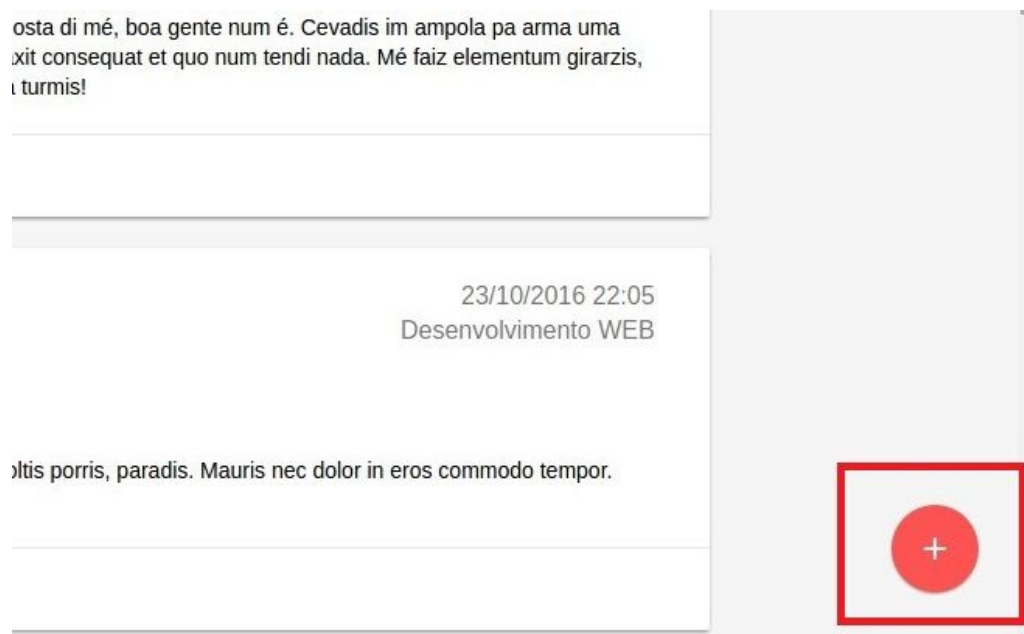
Se o usuário clicar em “minhas dúvidas”, vai mostrar uma tela do mesmo modelo do feed de dúvidas, só que vai ter apenas as dúvidas do usuário em todas as disciplinas.

Em “notificações”, vão estar as notificações de dúvidas respondidas e respostas aprovadas para os alunos, já que eles podem responder perguntas, mas antes terá que passar por uma aprovação do monitor. Para os monitores, em notificações vão estar as perguntas que foram feitas para ele responder, e as respostas que ele tem para analisar.

Na parte de baixo do menu estão listadas as disciplinas que o aluno está matriculado, se o usuário for um monitor, vão ser listadas as disciplinas que ele é monitor e no caso do professor são as que ele leciona. Clicando em cada disciplina

dessa, serão mostradas as perguntas só dessa disciplina que todos os alunos fizeram.

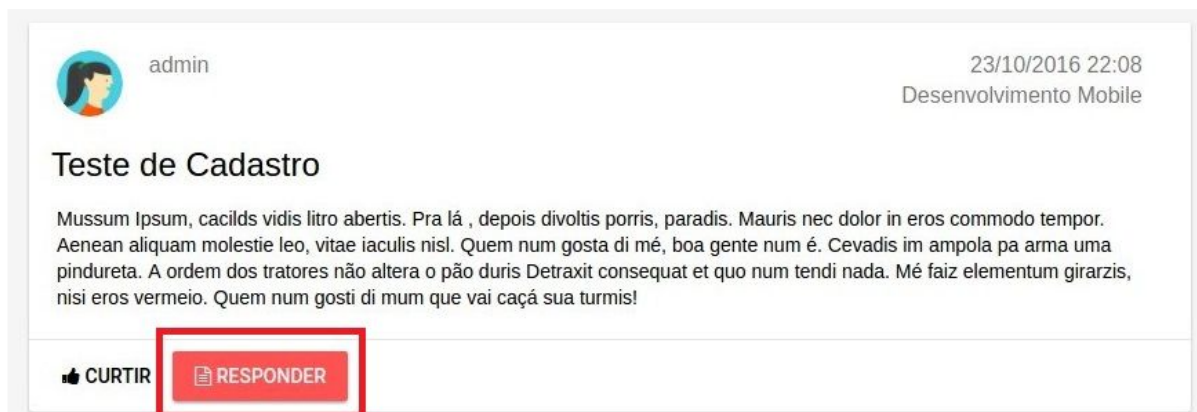
3. Clicando no botão “+” na tela inicial do sistema(feed de perguntas gerais), será mostrada a tela do usuário fazer uma pergunta.



4. A tela de fazer a pergunta, possui três campos para serem preenchidos. O primeiro é assunto, nele o usuário vai dizer qual é o assunto da dúvida dele. O segundo campo é a descrição, nele é o texto da dúvida. e o terceiro é a disciplina da dúvida. Quando ele apertar no botão “cadastrar” a dúvida será cadastrada, mostrada no feed de dúvidas e uma notificação vai chegar para o monitor da disciplina associada, avisando que ele tem uma pergunta para responder.

A screenshot of the "Cadastro de duvida" form in the "Monitoria" section. The form has three input fields: "Assunto *" (required), "Descrição *" (required), and a dropdown menu labeled "Selecione uma disciplina". A red "CADASTRAR" button is located at the bottom right of the form. On the left side of the screen, there is a sidebar with a user profile (admin, 2013123131, admin@hotmail.com) and a list of navigation items: "Duvidas", "Minhas Duvidas", "Notificações", and "ALUNO" (with sub-items "DIM547" and "DIM574").

5. Em cada pergunta tem um botão “responder”. Clicando nele é exibido um formulário para o monitor ou aluno responder a pergunta.



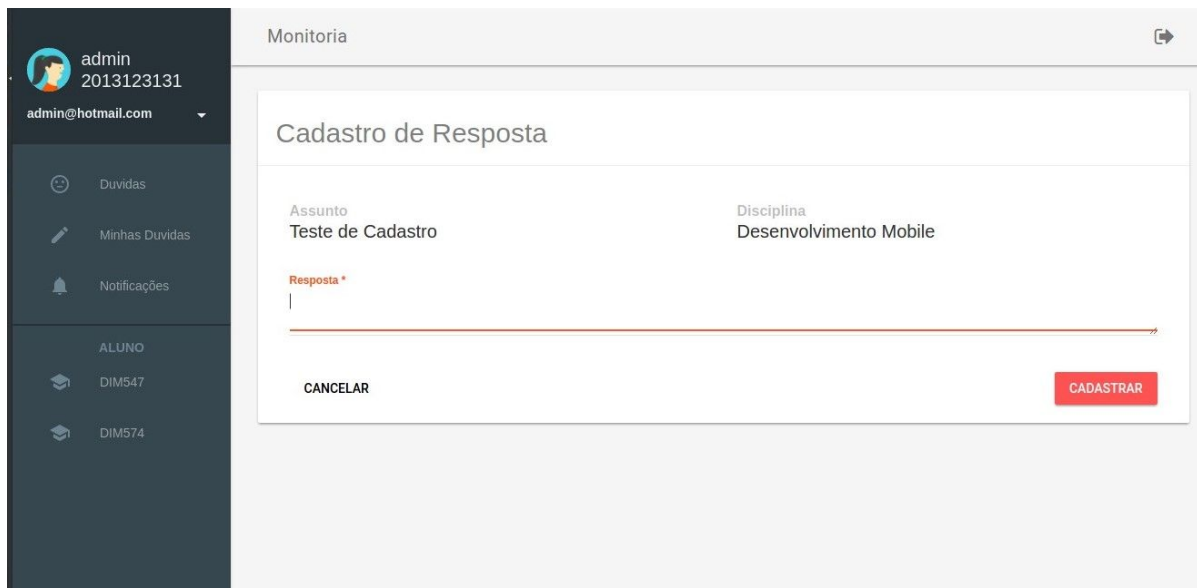
admin 23/10/2016 22:08
Desenvolvimento Mobile

Teste de Cadastro

Mussum Ipsum, cacilds vidis litro abertis. Pra lá , depois divoltis porris, paradis. Mauris nec dolor in eros commodo tempor. Aenean aliquam molestie leo, vitae iaculis nisl. Quem num gosta di mé, boa gente num é. Cevadis im ampola pa arma uma pindureta. A ordem dos tratores não altera o pão duris Detraxit consequat et quo num tendi nada. Mé faiz elementum girarzis, nisi eros vermeio. Quem num gosti di mum que vai caçá sua turmis!

👍 CURTIR 

6. A tela de responder a dúvida, possui apenas um campo para ser preenchido, que é o da própria resposta. Possui um cabeçalho com o assunto da pergunta e a disciplina. Depois de apertar no botão cadastrar a resposta, se ela foi de um monitor, uma notificação será enviada para o aluno que fez a pergunta, dizendo que ela foi respondida e uma tag será colocada na pergunta: “respondida”. Se a resposta foi de um aluno, uma notificação será enviada para o monitor, mostrando que ele tem uma resposta para avaliar. E uma tag de “pendente” será colocada na resposta. Depois que o monitor avaliar como certa ou não outra tag vai aparecer na resposta.



admin 2013123131
admin@hotmail.com

Dúvidas
Minhas Dúvidas
Notificações

ALUNO
DIM547
DIM574

Monitoria

Cadastro de Resposta

Assunto
Teste de Cadastro

Disciplina
Desenvolvimento Mobile

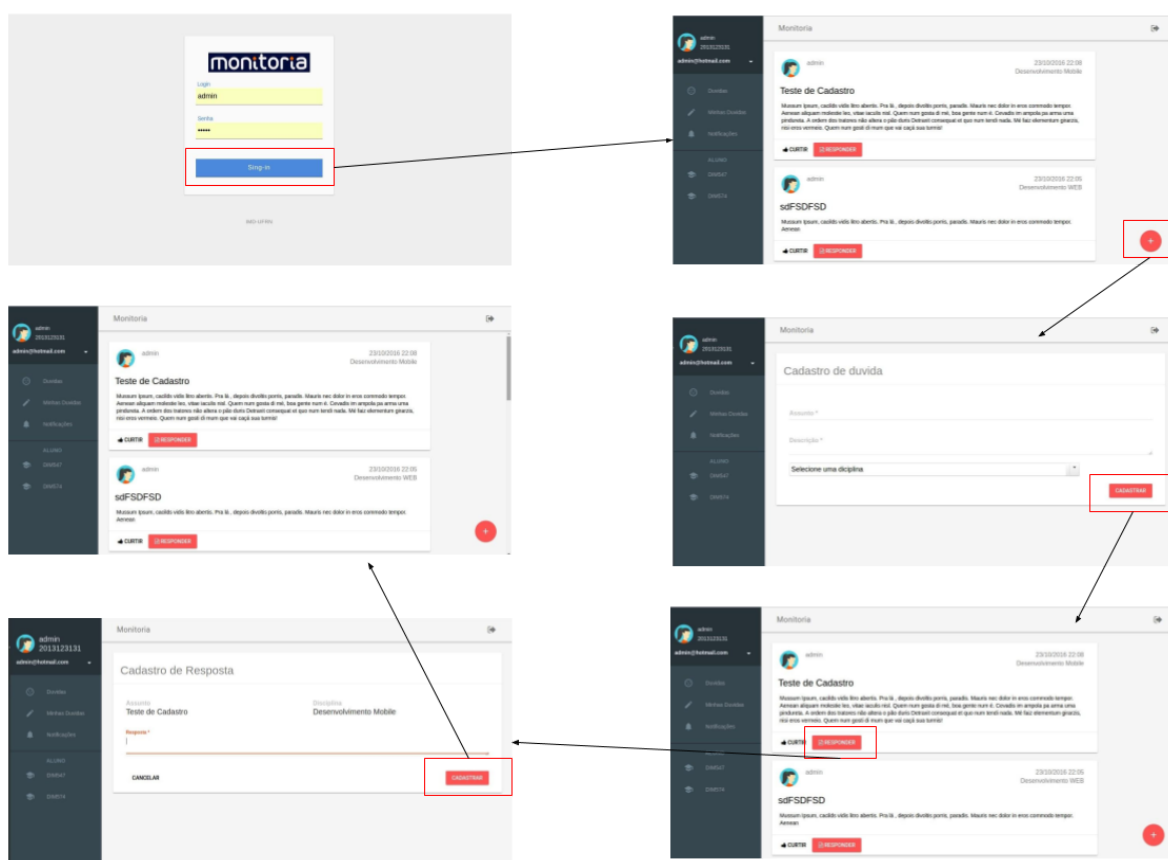
Resposta *

CANCELAR CADASTRAR

7. Em cada pergunta também possui um botão de “curtir”, essa curtida que dizer que o aluno também possui essa dúvida.



Um fluxo geral do que foi explicado acima é mostrado a seguir.



- A interface gráfica da aplicação, incluindo protótipos de tela e descrição de como os componentes do framework JSF foram utilizados na implementação.

A interface gráfica da aplicação e os protótipos das telas estão no tópico acima.

Componentes usados do framework JSF na implementação foi o primefaces, ele ajudou no layout e na conexão da apresentação com os Managed Bean, para os formulários de cadastro que temos na aplicação(pergunta, resposta). E para mostrar dados nas telas também foi usado.

- As classes utilizadas na implementação das camadas de apresentação e acesso a dados. A modelagem entidade-relacionamento descrevendo as entidades da aplicação e seus correspondentes a nível de banco de dados, feito através da JPA.

As classes utilizadas são: pessoa, usuário, perfil, papel, disciplina, dúvida e resposta. Abaixo temos uma descrição de cada uma e diagrama de classes que mostra em detalhes essas classes, seus atributos e a relação entre elas.

1. Usuário

A classe usuário é responsável pelos dados de login de uma pessoa no sistema, ela é mapeada para o banco de dados como uma entidade e nome da tabela no banco é “usuarios”. As colunas da tabela são os atributos e são eles: o id, atributo de identificação da classe que é um inteiro, mapeado como uma coluna com o nome id_usuario. O atributo login é uma string e é única. A senha do usuário também é uma string. Na classe tem um boolean com o nome ativo, ele serve para guardar se um usuário está ativo ou não no sistema. A classe usuário possui um atributo pessoa, pois todo usuário tem uma pessoa no sistema, esse relacionamento é unidirecional, ou seja, o usuário que pode ver as informações da pessoa e a pessoa não ver de usuário e é um para um, porque um usuário só pode ser de uma pessoa e uma pessoa só pode ter um usuário.

2. Pessoa

A classe pessoa é responsável pelos dados das pessoas que usam o sistema, ela é mapeada para o banco de dados com uma entidade e nome da tabela no banco é “pessoas”. As colunas da tabela são os atributos e são eles: o id, atributo de identificação da classe que é um inteiro, mapeado como uma coluna com o nome

id_pessoa. O nome, sexo, cpf, email e matrícula são string que guardam as informações pessoais.

3. Perfil

A classe perfil é responsável pela ligação da pessoa com o papel. É mapeada para o banco de dados com uma entidade e nome da tabela no banco é “perfil”. As colunas da tabela são os atributos e são eles: o id, atributo de identificação da classe que é um inteiro, mapeado como uma coluna com o nome id_perfil. Um boolean ativo, para indicar se aquele perfil está ativo ou não no sistema. O curso é uma string, que informa de qual curso é o professor, aluno ou monitor. Temos um atributo pessoa, com um relacionamento unidirecional, pois o perfil que pode ver as informações da pessoa e a pessoa não pode ver o perfil. E o relacionamento é 1 para 1, um perfil só pode ser de uma pessoa. Tem um atributo papel que é uma classe também e possui um relacionamento unidirecional, pois só o perfil pode ver qual é o papel e é de 1 para 1. Na entidade perfil temos também uma lista de disciplinas, já que tanto aluno, monitor e professor possuem disciplinas. O relacionamento unidirecional, pois é o perfil que ver a disciplina e disciplina não pode ver o perfil, é de muitos para muitos, ou seja, um perfil tem várias disciplinas e uma disciplina pode ser de vários perfis.

4. Papel

A classe papel é responsável pelos dados que guardam de qual tipo é um usuário no sistema, se é um aluno, monitor ou professor. É mapeada para o banco de dados com uma entidade e nome da tabela no banco é “papel”. As colunas da tabela são os atributos e são eles: o id, atributo de identificação da classe que é um inteiro, mapeado como uma coluna com o nome id_papel. Possui mais dois atributos que são: descrição e role, os dois são do tipo string.

5. Disciplina

A classe disciplina é responsável pelos dados da turma dos alunos, professores e monitores. É mapeada para o banco de dados com uma entidade e nome da tabela no banco é “disciplina”. As colunas da tabela são os atributos e são eles: o id, atributo de identificação da classe que é um inteiro, mapeado como uma coluna com o nome id_disciplina. Possui os atributos código e descrição que são strings.

6. Dúvida

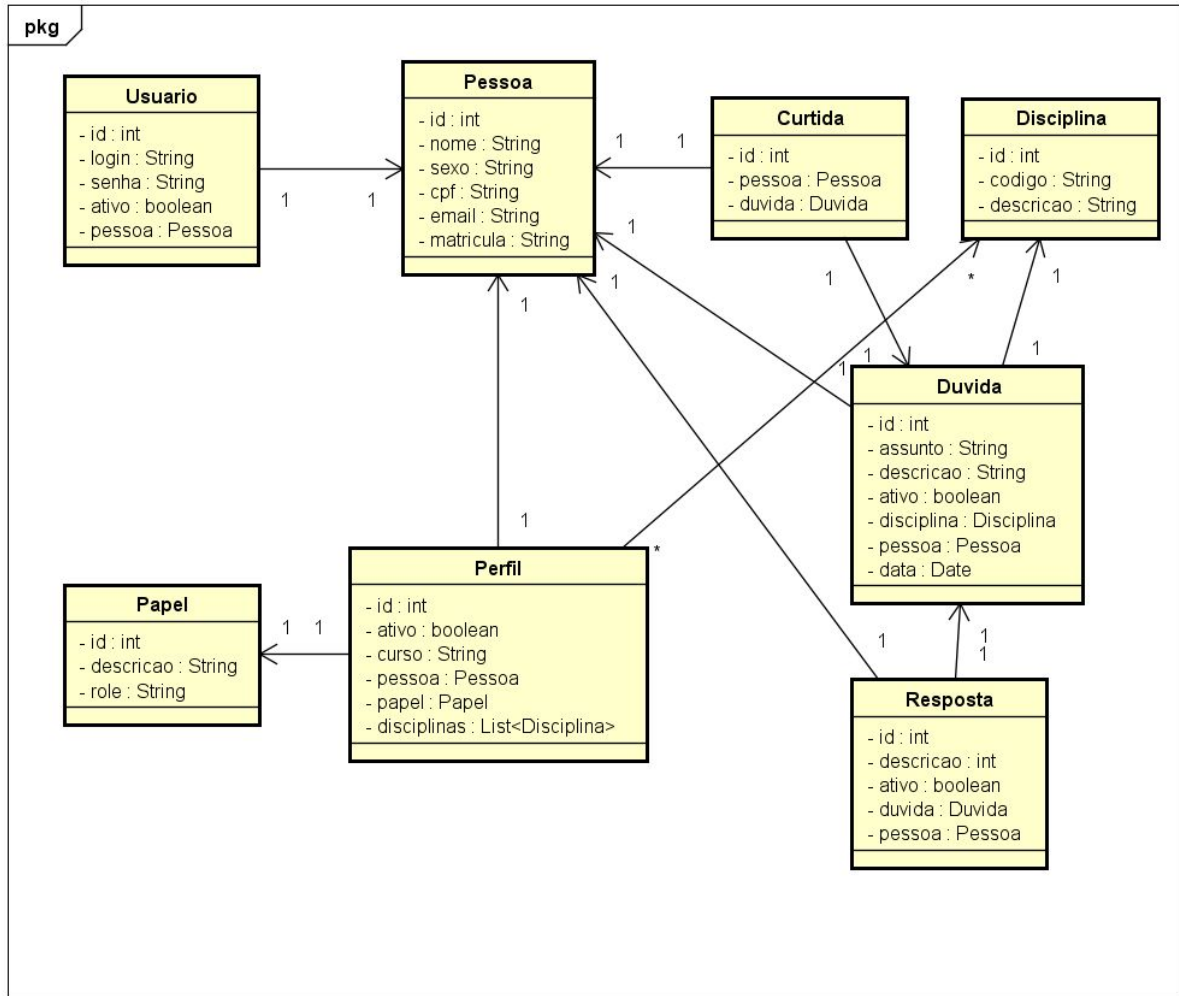
A classe dúvida é responsável pelos dados das perguntas dos alunos. Essa é uma das classes mais importantes, devido a ideia do sistema, que é fazer perguntas e pessoas responderem. É mapeada para o banco de dados com uma entidade e nome da tabela no banco é “duvidas”. As colunas da tabela são os atributos e são eles: o id, atributo de identificação da classe que é um inteiro, mapeado como uma coluna com o nome id_duvida. A dúvida tem um assunto e uma descrição que são string e um boolean que indica se foi respondida ou não. Dúvida tem um atributo disciplina, com um relacionamento unidirecional que indica que só a dúvida pode ver de qual disciplina ela é, e de um para um, pois uma dúvida só possui uma disciplina. Já que um perfil possui pessoa e disciplinas e dúvida tem disciplina, temos que fez a pergunta.

7. Resposta

A classe resposta, como o nome já diz, são as respostas das perguntas feitas nas disciplinas. Essa é uma bastante importante no sistema, devido a ideia do sistema, que é fazer perguntas e pessoas responderem. É mapeada para o banco de dados com uma entidade e nome da tabela no banco é “respostas”. As colunas da tabela são os atributos e são eles: o id, atributo de identificação da classe que é um inteiro, mapeado como uma coluna com o nome id_resposta. Possui o atributo descrição que é uma string. Tem também um boolean ativo, pois um aluno pode responder uma pergunta e ela precisará passar por uma avaliação do monitor, e esse atributo indica se ela está correta ou incorreta. Outro atributo é dúvida, pois as respostas tem relacionamento com as perguntas unidirecional, e de um para um. A resposta só pode está associada com um pergunta e com a pergunta nós temos de qual dúvida ela é.

8. Curtida

A classe curtida é responsável pelas curtidas que os usuários vão dar nas dúvidas. É mapeada para o banco de dados com uma entidade e nome da tabela no banco é “curtidas”. As colunas da tabela são os atributos e são eles: o id, atributo de identificação da classe que é um inteiro, mapeado como uma coluna com o nome id_curtida. O atributo pessoa é para salvar quem deu a curtida e temos um relacionamento unidirecional e de um para um. O atributo dúvida, guarda a dúvida da curtida.



- Tecnologias adicionais que porventura tenham sido empregadas no desenvolvimento e como isso foi feito.

As tecnologias usadas foram as seguintes:



- Possíveis pendências e dificuldades e como elas podem ser sanadas nas próximas etapas do desenvolvimento

As pendências que temos é a conexão com a api, pois precisamos de alguns dados para concluir nossa aplicação. Os dados que vamos precisar são as turmas dos alunos, monitores e professores. As turmas possuem os participantes que são os alunos, quem são os monitores e o professores delas. Também vamos precisar do login e senha dos alunos do sigaa, para ser feito o login no sistema da monitoria.

Enquanto isso, estamos usando outros dados que não são “reais” e a conexão com a api será feita na terceira e última parte de desenvolvimento do projeto. Logo, nosso projeto não tem muitos CRUDs e consulta com o banco de dados, pois a aplicação é simples e basicamente temos dois principais CRUDs que são de dúvida e resposta. E resolvemos fazer já com as três camadas: domínio, services e dao. Que os services(camada de negócio), só seria entregue na próxima etapa. Adiantamos isso para focar nos próximos passos focarmos na conexão com a api.

3. Conclusão

Concluimos a segunda parte do projeto com as entidades mapeadas no banco de dados usando JPA(domínio da aplicação), os controllers, que fazem a ligação da camada apresentação com as informações salvas no banco, usando JPA. Os service, que estão as regras de negócio(camada de negócio) e o dao, que temos as consultas do banco de dados. Na próxima etapa do projeto iremos usar serviços web para conectar nossa aplicação com a api e pegarmos as informações que faltam para finalizar.