

Desenvolvimento Web II

Atividade 1

Índice

Resultados a serem entregues.....	2
Composição da nota	2
Descrição do Trabalho	3
Opção de sistema 1 – Portal de Notícias	4
Sistema.....	4
Requisitos funcionais	4
Requisitos mínimos.....	4
Requisitos extras	5
Protótipos	6
Modelo de dados	7
Opção de sistema 2 – Catálogo de Filmes	8
Sistema.....	8
Requisitos funcionais	8
Requisitos mínimos.....	8
Requisitos extras	9
Protótipos	10
Modelo de dados	11
Opção de sistema 3 – Postagens	12
Sistema.....	12
Requisitos funcionais	12
Requisitos mínimos.....	13
Protótipos	14
Modelo de dados	15
Dicas.....	16

Resultados a serem entregues

1. O trabalho pode ser desenvolvido em grupos de **até (no máximo) 4 pessoas**, podendo ser individual;
2. O trabalho escrito deve ser publicado no Teams por 1 pessoa do grupo;
3. Código fonte publicado no github (ou similar como gitlab);
4. O projeto no github deverá conter um README.md contendo a descrição do projeto e o passo a passo para subir a aplicação e realizar os testes. Lembrete: você deveria subir os projetos de back e front em repositórios distintos;
5. No mínimo 2 requisitos extras (não solicitadas pelo professor), que serão o diferencial do seu trabalho comparativamente aos demais
6. Documento escrito, contendo, no mínimo:
 - a. Descritivo da aplicação, funcionalidades, destacando as funcionalidades extras;
 - b. Descrição da arquitetura de aplicação utilizada;
 - c. Descrição sobre os frameworks utilizados;
 - d. MER da base de dados;
 - e. Diagrama de classes;
 - f. Prints da aplicação funcionando
7. Apresentação
 - a. Arquitetura, framework e diagramas
 - b. Explicar quais funcionalidades extras foram desenvolvidas
 - c. Demonstração da aplicação funcionando

Composição da nota

1. Código fonte publicado – 1 ponto
2. Aplicação que rode – 1 ponto
3. Requisitos mínimos atendidos – 2.5 pontos
4. Requisitos extras – 2.5 pontos
5. Documento escrito – 1 ponto
6. Apresentação – 2 pontos

Descrição do Trabalho

O trabalho a ser desenvolvido consiste na construção de um sistema web completo (front, back e banco), de forma que o aluno se desafiará a utilizar os conhecimentos acumulados do primeiro e segundo semestres.

Especificarei 3 sistemas diferentes, onde o grupo deverá escolher somente 1 deles. É esperado que exista uma distribuição uniforme dentro da turma entre os sistemas escolhidos.

O grupo deve desenvolver o sistema atendendo obrigatoriamente todos os requisitos mínimos especificados, e também, 2 requisitos extras à escolha do grupo.

Opção de sistema 1 – Portal de Notícias

Sistema

O objetivo é permitir que, do ponto de vista do Publicador (ator do sistema) publique notícias, edite ou exclua através do subsistema Gerenciador e do ponto de vista do Visitante (ator do sistema) leia notícias de acordo com suas preferências de categorias

Requisitos funcionais

1. Como um visitante da home page, quero poder efetuar o login ou cadastro, para que eu possa acessar o sistema
2. Como um Visitante logado, quero poder ler as notícias de minha preferência de categoria da mais atual para a mais antiga, para que eu leia as notícias mais relevantes para mim
3. Como um Publicador logado, quero poder publicar uma notícia, para que os clientes do portal tenham notícias atuais para serem lidas.

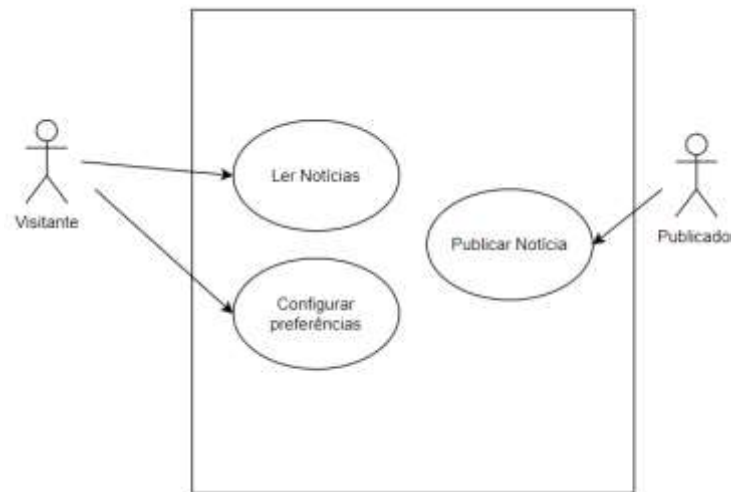


Figura 1. Diagrama de Caso de Uso do Portal de Notícias

Requisitos mínimos

Descrevo aqui os mesmos requisitos funcionais já citados acima, porém com maiores detalhes técnicos.

Os requisitos mínimos deste projeto serão descritos a seguir:

1. Inscrição - deverá ser desenvolvida uma tela para inscrição (cadastro) de novos usuários. Os campos de e-mail, senha e apelido devem ser obrigatórios. Após a inscrição o usuário deve ser redirecionado para a tela de login. Ao ser inscrito, o usuário deve receber por padrão o perfil de Visitante. Você poderá alterar o perfil para Publicador diretamente no banco para alguns usuários, pois não está sendo prevista uma tela para essa gestão, porém sinta-se livre para construir essa tela.

2. Login - a tela de login deverá possuir uma descrição do sistema; link para inscrição de novo usuário e formulário para login. Em caso de insucesso, deverá ocorrer uma mensagem de “Usuário e/ou senha estão incorretos”. Em caso de sucesso, o usuário deverá ser redirecionado para a tela principal de publicações. Obs.: atualmente o sistema não usa criptografia de senha, sintá-se livre para adicionar caso deseje.

3. Leitura de notícias – o Visitante verá na tela principal de publicações todas as publicações em ordem da mais recente para a mais antiga. Caso ele não possua uma configuração de categorias de preferência, não deve ser aplicado filtros. Caso possua, somente as notícias das categorias de sua preferência devem ser exibidas.

4. Publicação de notícias – o Publicador terá acesso à uma tela de publicação de notícias. Nela o Publicador preencherá um formulário com os campos necessários como, por exemplo: título, imagem, texto e categoria; e submeterá para criar a nova notícia.

Requisitos extras

Para esse projeto você deverá incluir, no mínimo, 2 requisitos extras (sendo funcionais ou não-funcionais). Você tem liberdade para decidir o que irá fazer. As sugestões abaixo são somente sugestões.

Sugestões de requisitos extras - Funcionais

1. Confirmação de existência do e-mail utilizado no cadastro;
2. Filtro dinâmico por categoria (nesse cenário o filtro prevalece sobre as preferências);
3. Compartilhar uma notícia;
4. Tela para configuração de Preferência de Categorias;
5. Gestão de notícias pelo Publicador, permitindo editar ou excluir notícias já existentes;
6. Efetuar o upload da foto da notícia via sistema (ao invés de inserir um link);
7. Lazyload no carregando das notícias e de forma que gere uma rolagem infinita, ou seja, na tela de notícias você só vê algumas, mas ao rolar para baixo (scroll) mais algumas postagens são carregadas.

Sugestões de requisitos extras – Não-Funcionais

1. Criptografia BCrypt da senha na base;
2. Autenticação usando JWT;
3. Utilização de Docker;
4. Utilização de framework de front como React, Angular ou Vue (não vale o Bootstrap CSS);
5. Utilização de framework diferente de Spring, como Micronaut ou Jooby;
6. Inclusão de testes unitários ou de carga;
7. Utilização de um banco de dados por Docker, ou seja, para subir a aplicação posso subir um banco local, sem nenhuma outra instalação exceto o Docker.

Protótipos

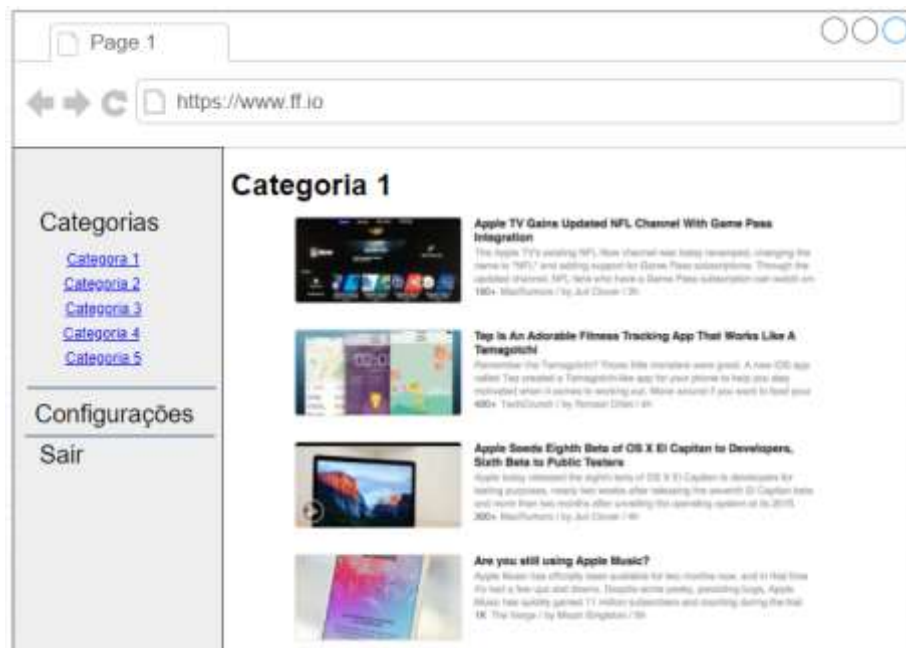


Figura 2. Tela de leitura de notícias

Este protótipo representa a interface de uma aplicação web para publicação de notícias. No topo, há uma barra de navegação com o texto "Page 1" e um endereço de URL "https://www.ff.io". À esquerda, um menu lateral contém as opções "Publicar", "Gerenciar" e "Sair". O menu "Publicar" está selecionado. O formulário de publicação contém os seguintes campos: "Título:" com um campo de texto "título da notícia"; "Imagem:" com um campo de texto "endereço da imagem"; "Notícia:" com um campo de texto "texto da notícia"; "Categorias:" com uma lista de seleção contendo "Categoria 1", "Categoria 2", "Categoria 3" (selecionada) e "Categoria 4"; "Publicada:" com um botão "ON" e um botão "Salvar".

Figura 3. Tela de publicação de notícias

Modelo de dados

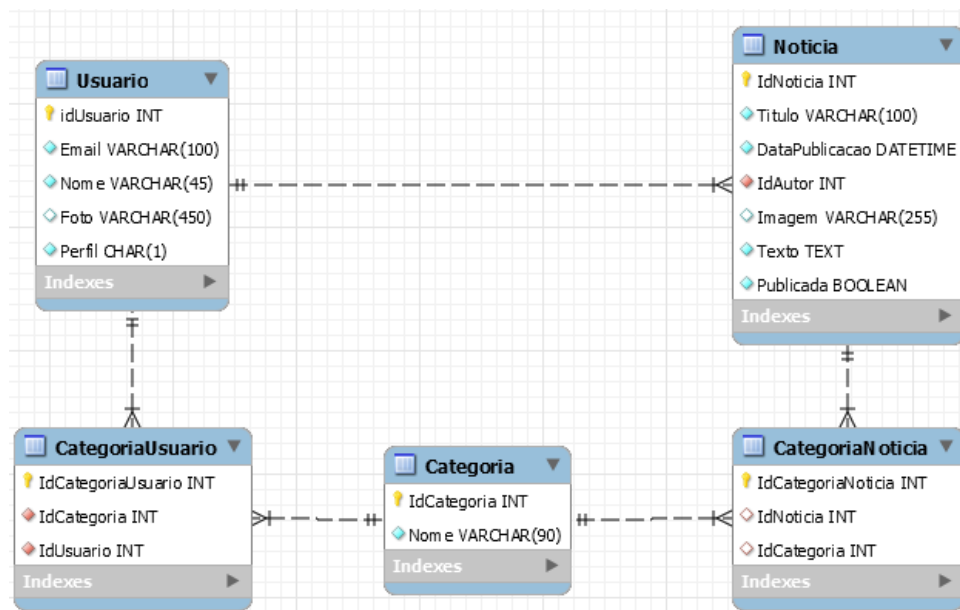


Figura 4. Modelo de dados do Portal de Notícias

Opção de sistema 2 – Catálogo de Filmes

Sistema

O objetivo é permitir que um Usuário visualize, publique e ranquee filmes dentro de um catálogo online.

Requisitos funcionais

1. Como um visitante da home page, quero poder efetuar o login ou cadastro, para que eu possa acessar o sistema
2. Como um Usuário, quero poder visualizar todos os filmes cadastrados com a possibilidade de filtrar a lista através do gênero e/ou nota média do filme
3. Como um Usuário logado, quero poder publicar um filme, para que outros usuários possam ser informados sobre o lançamento.
4. Como um Usuário logado, quero poder dar uma nota para o filme entre 0 e 5, e que essa minha nota individual acarrete em um re-cálculo da nota média do filme, para que os outros usuários possam fazer pesquisas mais relevantes.

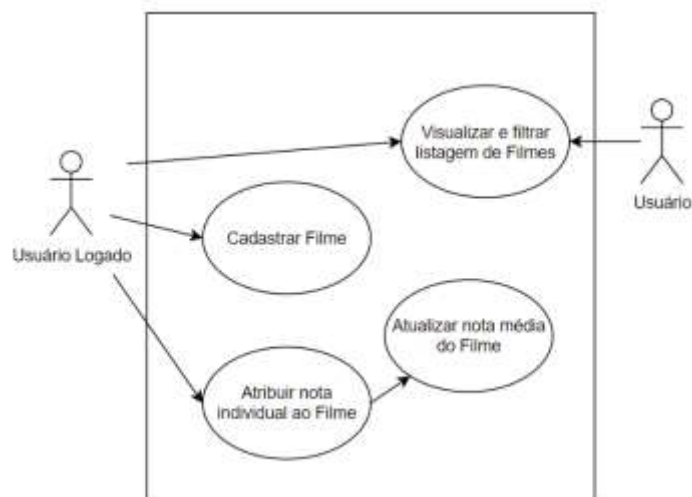


Figura 5. Diagrama de Caso de Uso do Catálogo de Filmes

Requisitos mínimos

Descrevo aqui os mesmos requisitos funcionais já citados acima, porém com maiores detalhes técnicos.

Os requisitos mínimos deste projeto serão descritos a seguir:

1. Inscrição - deverá ser desenvolvida uma tela para inscrição (cadastro) de novos usuários. Os campos de e-mail, senha e apelido devem ser obrigatórios. Após a inscrição o usuário deve ser redirecionado para a tela de login.

2. Login - a tela de login deverá possuir uma descrição do sistema; link para inscrição de novo usuário e formulário para login. Em caso de insucesso, deverá ocorrer uma mensagem de “Usuário e/ou senha estão incorretos”. Em caso de sucesso, o usuário deverá ser redirecionado para a tela principal de publicações. Obs.: atualmente o sistema não usa criptografia de senha, sintase livre para adicionar caso deseje.

3. Visualizar o catálogo – o Usuário verá na tela principal do catálogo todos os filmes em ordem conjunta (2 atributos sendo considerados) de pontuação mais alta para mais baixa e do lançamento mais recente para o mais antigo. Caso ele utilize algum filtro como, pontuação ou gênero, o filtro deve ser respeitado.

4. Cadastrar filme – o Usuário logado terá acesso à uma tela de cadastro de filmes. Nela o Publicador preencherá um formulário com os campos necessários como, por exemplo: título, imagem, texto e categoria; e submeterá para criar o novo filme.

5. Atribuir nota ao filme – o Usuário logado poderá dar nota à um filme. Tal nota somente poderá ser dada 1 (uma) vez por mesmo usuário ao mesmo filme. Ao atribuir essa nota individual, a nota média do filme deverá ser atualizada.

Requisitos extras

Para esse projeto você deverá incluir, no mínimo, 2 requisitos extras (sendo funcionais ou não-funcionais). Você tem liberdade para decidir o que irá fazer. As sugestões abaixo são somente sugestões.

Sugestões de requisitos extras - Funcionais

1. Confirmação de existência do e-mail utilizado no cadastro;
2. Pré-visualização do filme ou trailer;
3. Compartilhar o filme;
4. Após o login, apresentar ao Usuário uma listagem com os filmes recém-cadastrados;
5. Permitir comentário textual ao atribuir nota ao filme;
6. Efetuar o upload da foto da notícia via sistema (ao invés de inserir um link);
7. Lazyload no carregando das notícias e de forma que gere uma rolagem infinita, ou seja, na tela de notícias você só vê algumas, mas ao rolar para baixo (scroll) mais algumas postagens são carregadas.
8. Galeria de imagens do filme;

Sugestões de requisitos extras – Não-Funcionais

1. Criptografia BCrypt da senha na base;
2. Autenticação usando JWT;
3. Utilização de Docker;
4. Utilização de framework de front como React, Angular ou Vue (não vale o Bootstrap CSS);
5. Utilização de framework diferente de Spring, como Micronaut ou Jooby;
6. Inclusão de testes unitários ou de carga;

- Utilização de um banco de dados por Docker, ou seja, para subir a aplicação posso subir um banco local, sem nenhuma outra instalação exceto o Docker.

Protótipos

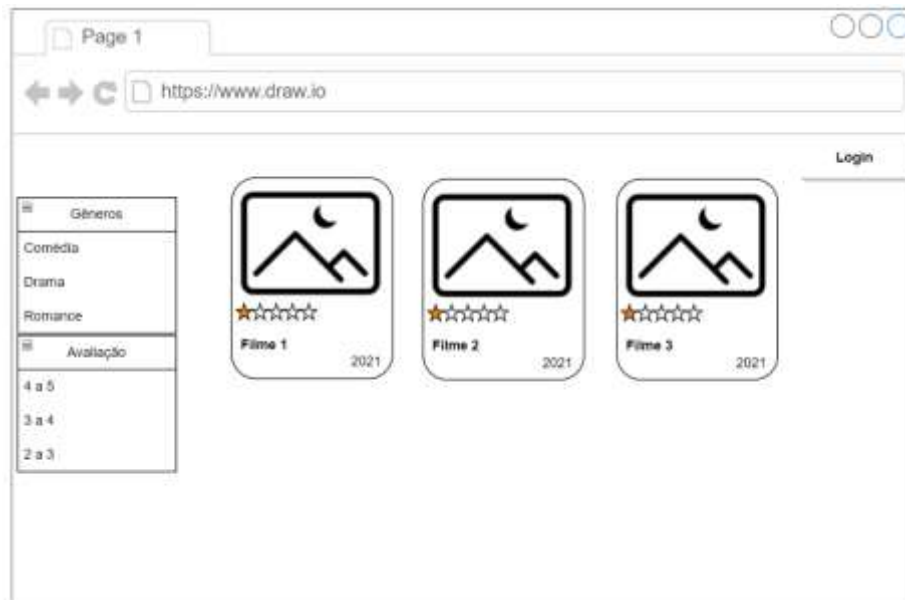


Figura 6. Tela de visualização e filtro do catálogo

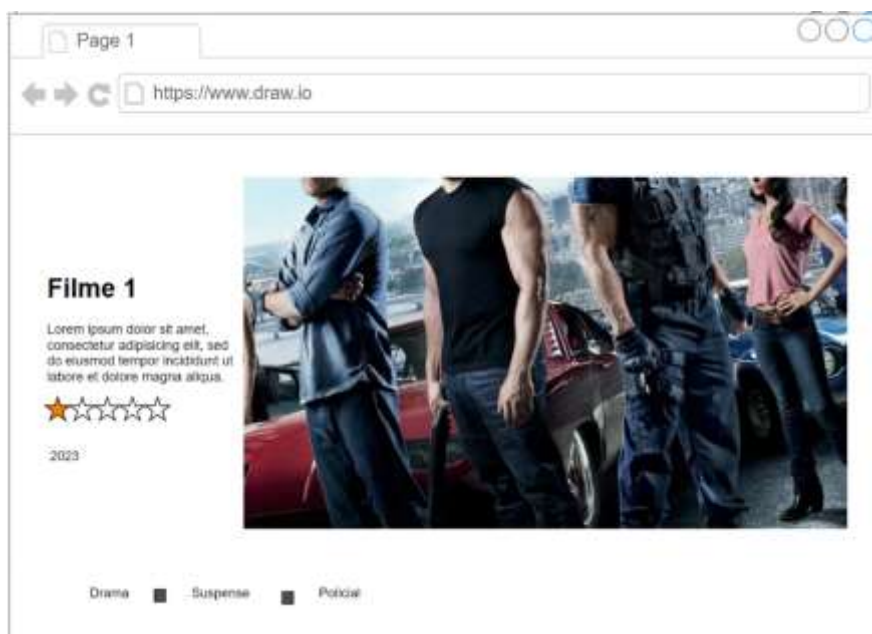


Figura 7. Tela de detalhes do filme

Modelo de dados

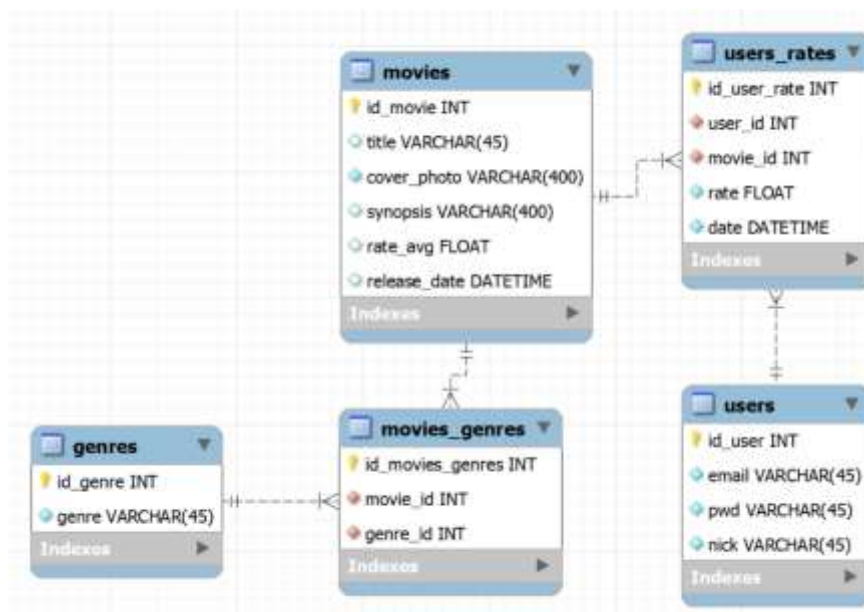


Figura 8. Modelo de dados do Portal de Notícias

Opção de sistema 3 – Postagens

Sistema

Sistema de publicação de postagens e reações, onde o usuário cadastrado e logado pode fazer postagens, e essas serão visualizadas pelos demais usuários que poderão reagir à postagem com “gostei” por exemplo.

Requisitos funcionais

5. Como um visitante da home page, quero poder efetuar o login ou cadastro, para que eu possa acessar o sistema
6. Como um usuário logado, quero poder efetuar uma postagem, para que essa minha postagem apareça para os demais usuários e eu possa expressar minhas opiniões e senti-me parte de um grupo (pirâmide de Maslow)
7. Como um usuário ao ver uma postagem poderá "reagir" gostando ou desgostando do post.

Requisitos extras

Para esse projeto você deverá incluir, no mínimo, 2 requisitos extras (sendo funcionais ou não-funcionais). Você tem liberdade para decidir o que irá fazer. As sugestões abaixo são somente sugestões.

Sugestões de requisitos extras - Funcionais

9. Confirmação de existência do e-mail utilizado no cadastro;
10. Tagueamento de postagem, como um “#programacao”;
11. Seguir um usuário, onde as postagens desse usuário seguido terão algum destaque;
12. Responder uma postagem;
13. Marcação de usuários na postagem, como um “@marcinha”;
14. Efetuar o upload da foto do perfil via sistema;
15. Lazyload no carregando das postagens e de forma que gere uma rolagem infinita, ou seja, na tela de postagens você só vê algumas, mas ao rolar para baixo (scroll) mais algumas postagens são carregadas.

Sugestões de requisitos extras – Não-Funcionais

8. Criptografia BCrypt da senha na base;
9. Autenticação usando JWT;
10. Utilização de Docker;
11. Utilização de framework de front como React, Angular ou Vue (não vale o Bootstrap CSS);
12. Utilização de framework diferente de Spring, como Micronaut ou Jooby;
13. Inclusão de testes unitários ou de carga;
14. Utilização de um banco de dados por Docker, ou seja, para subir a aplicação posso subir um banco local, sem nenhuma outra instalação exceto o Docker.

Requisitos mínimos

Descrevo aqui os mesmos requisitos funcionais já citados acima, porém com maiores detalhes técnicos.

Os requisitos mínimos deste projeto serão descritos a seguir:

1. Inscrição - deverá ser desenvolvida uma tela para inscrição (cadastro) de novos usuários. Os campos de e-mail, senha e apelido devem ser obrigatórios. Após a inscrição o usuário deve ser redirecionado para a tela de login.

2. Login - a tela de login deverá possuir uma descrição do sistema; link para inscrição de novo usuário e formulário para login. Em caso de insucesso, deverá ocorrer uma mensagem de “Usuário e/ou senha estão incorretos”. Em caso de sucesso, o usuário deverá ser redirecionado para a tela principal de postagens. Obs.: atualmente o sistema não usa criptografia de senha, sintase livre para adicionar caso deseje.

3. Postagens - esta tela deve suportar as funcionalidades de postar; ler as últimas postagens (as 10 últimas); e reagir.

No formulário de postar o usuário poderá preencher um campo com o texto desejado e submeter.

A lista das últimas postagens estará visível para este usuário. Cada post listado deve exibir: foto, apelido e status do usuário que postou; data e texto do post; total de "gostei" e "não gostei"; e links para "gostar" e "não gostar".



Figura 9. Demonstração do fluxo de informações no sistema.

Protótipos

The screenshot shows a web browser window with the title "Login - Falaê". The page content is as follows:

Falaê

Falaê é um sistema de comunicação desenvolvido pelos alunos do Falaê da Oitavo.

Através do Falaê é possível postar suas mensagens, ler o que seus amigos estão dizendo e comentar/editar as mensagens.

Entre agora para sua comunidade!

On the right side, there is a login form with the following fields and buttons:

- E-mail:** A text input field containing the word "text".
- Senha:** A password input field with a masked password "XXXXXXXXXX".
- Entrar:** A button to submit the login form.
- [Esqueceu-se](#)**: A link for users who forgot their password.

Figura 10. Login



Fasto - Falae

Falae

Cadastro de Usuário

E-mail
Text

Senha
Text

Apellido
Text

Cancelar

Figura 11. Inscrição

Figura 12. Postagens

Modelo de dados

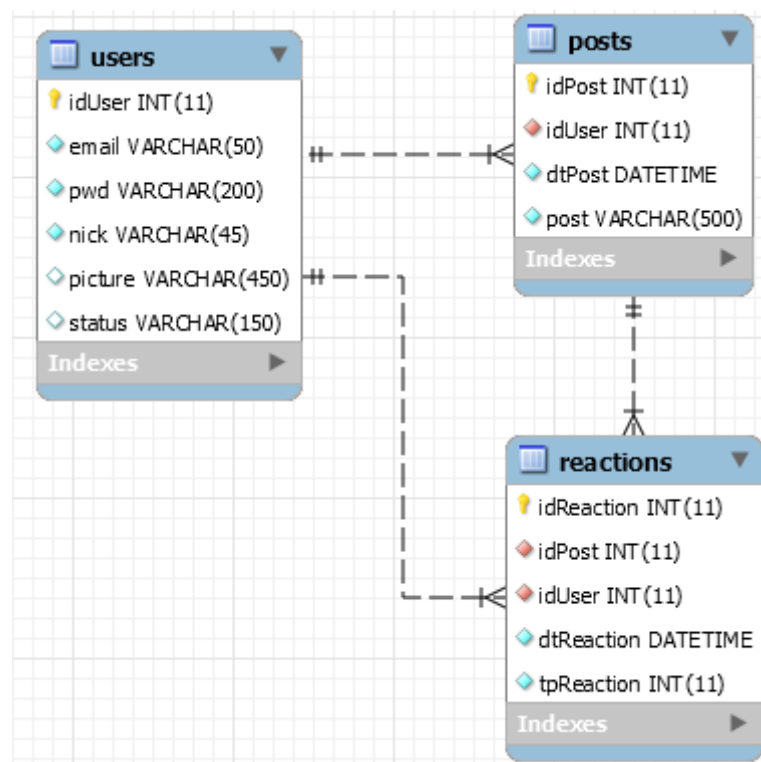


Figura 13. Modelo de dados do Portal de Postagens

A tabela “users” guardará os dados dos usuários;

A tabela “posts” guardará as postagens feitas por cada “user” (fk);

Na tabela “reactions”, outros “users” (fk) poderão reagir à um “post” (fk), a reação pode ser do tipo (tpReaction) “gostei” ou “não gostei”.

Dicas

1. Para trabalhar em time, como em uma empresa qualquer, organize sua equipe dividindo em responsáveis pela interface; funcionalidade; e dados, ou seja, pessoas focadas em deixar o site bonito e organizado (interface); pessoas focadas em deixar as páginas funcionando, os formulários submetendo (login, cadastro, postagem) (funcionalidade), e pessoas focadas nas consultas e inserções no banco de dados (dados);
2. Sugiro o uso de alguma ferramenta para organização da equipe, por exemplo o Trello (<http://trello.com>) ou o ClickUp (<https://clickup.com/>);
3. Não se preocupe com criptografia de senha, salva a senha como se fosse um dado comum (a não ser que esse seja seu requisito extra);
4. A imagem do usuário a ser guardada é na realidade o **endereço** físico da imagem, podendo ser o endereço dentro do seu site (ex: `"/img/pic_user_1.jpg"`), ou um endereço externo (ex: `"http://www.externo.com/pic_user_1.jpg"`). Não se preocupe com salvar a **imagem** dentro do banco (blob) (a não ser que esse seja seu requisito extra);
5. O modelo de dados criado permite qualquer tipo de reação, mas para facilitar o desenvolvimento, use somente 2 tipos (gostei e não gostei). Porém, caso deseje crie novos tipos, será um desafio a mais;