

Projeto - SI 1 2016.2

Especificação

Sistema de anúncios ad-extreme: pessoas ou empresas possam publicar vários tipos de anúncios, tanto para venda de imóveis ou móveis bem como serviços e empregos. Os usuários pesquisam por anúncios por meio de *tag/categoria* ou pelo título do mesmo e ordena os mesmos por: data, preço, ou classificação do anunciante. Eles podem classificar o anunciante e contactar os mesmos por meio da plataforma ou por meio dos contatos disponibilizados pelos anunciantes. Os anunciantes podem determinar o tempo dos anúncios bem como determinar a data do início da publicação do anúncio (similar: olx, mercado livre).

Funcionalidades já implementadas:

- É possível cadastrar um usuário, pessoa física ou jurídica;
- É possível para o usuário cadastrar anúncios de móveis, imóveis e emprego, definindo um valor;
- O usuário faz login para poder usar o sistema;
- É possível listar todos os anúncios postados, independente do usuário.

Cada equipe deve escolher cinco (5) user stories para implementar, sendo que (1) delas é obrigatoriamente:

- O projeto deve ser colocado no Heroku ou outro serviço de hospedagem, para facilitar o acesso.

As quatro stories restantes podem ser escolhidas a partir destas abaixo:

- Cada usuário que comprar algo de um anúncio fica com débito, e se vender algo fica com crédito. Toda vez que o usuário logar, deve ser mostrado o saldo devedor/credor na tela;
- Um anúncio pode ser comprado por qualquer usuário (se forem móveis ou imóveis, apenas). Quando comprar, o usuário fica com o débito do valor, e o usuário que anunciou fica com o crédito. Neste caso, o anúncio já não existe mais no sistema, assim que for comprado/vendido;
- Cada usuário deve ter disponível seu saldo de compra e venda, para visualizar a qualquer momento;
- Cada usuário pode fazer busca de anúncios por tipo e data; cada usuário pode também visualizar seus próprios anúncios;
- Usuário pode cadastrar um novo tipo de anúncio: serviço (por exemplo, encanador, ou eletricista). O usuário que comprar o serviço pode agendar um dia e horário. No caso apenas usuários Pessoa Jurídica podem colocar anúncios de Empregos e Serviços.
- Vendedores e compradores podem avaliar anúncio (pelo comprador) ou comprador (pelo vendedor), com estrelas e comentários;
- O usuário quer visualizar um ranking de vendedores, baseado na avaliação dos seus anúncios;
- Usuário pode visualizar perfil dos anunciantes, e definir seus anunciantes favoritos;
- Vendedores podem ser notificados sobre mudanças nos seus anúncios (compras).

Além das obrigatórias, outras podem ser implementadas, opcionalmente, podendo trazer pontuação extra na correção.

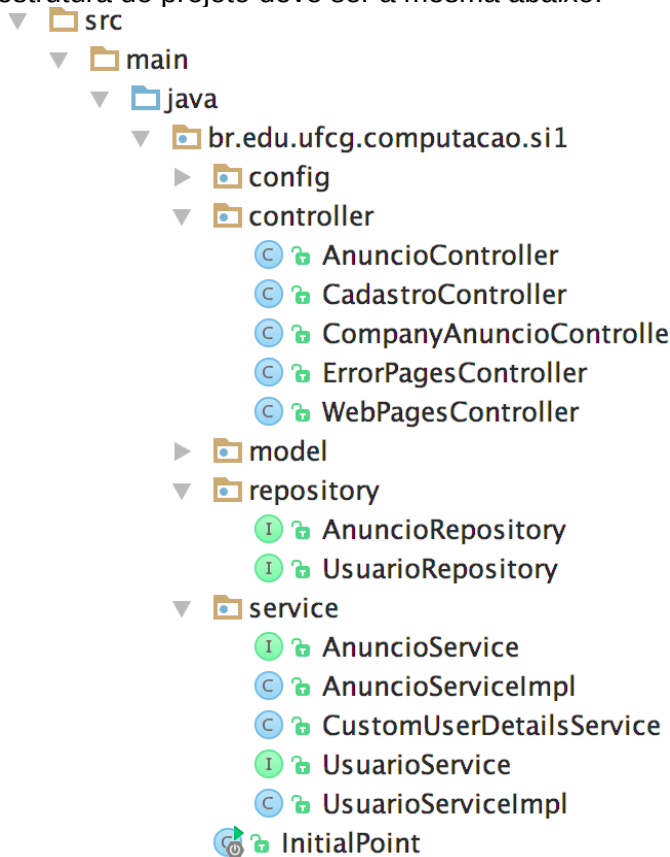
Instruções e Recomendações

- Cada equipe deve ter no máximo quatro integrantes;
- Uma equipe pode ter integrantes de ambas as turmas, no entanto esta equipe precisa definir a turma na qual serão feitos os acompanhamentos;

- A entrega será o link do projeto no repositório github, **junto com o link de um documento público, no google docs, com as principais decisões de design, documentadas, e os problemas de design (*smells*) corrigidos durante a implementação, indicando as user stories escolhidas pela equipe para implementar;**
- **Vocês podem fazer as modificações que acham necessárias tanto no backend quanto no frontend (pode usar Angular , por exemplo), contanto que usem o Spring Boot no backend;**
- **Façam o frontend mais simples possível que satisfaça as stories, evitando perder muito tempo na interface com o usuário (vocês terão disciplinas no curso que terão foco no backend);**
- A nota máxima é 10,0, atribuída a partir dos seguintes critérios:
 - adequação aos requisitos (stories obrigatórias/opcionais);
 - defeitos detectados;
 - design arquitetural (separação de interesses, integridade conceitual);
 - design detalhado (princípios de projeto, padrões);
 - participação individual;

Guia para instalação/configuração do projeto

1. Descompacte o arquivo zip fornecido, com as classes do projeto.
2. Ao incluir o projeto na IDE (IntelliJ ou Eclipse, mas pode ser qualquer uma com suporte a Maven), a estrutura de projeto deve ser a mesma abaixo:



2. Instale o banco de dados PostGres 9.5 (<https://www.postgresql.org>), criando uma conta com login **postgres** e senha **admin** (esta conta já é criada na instalação, basta definir a senha) . Inicie o banco de dados na porta 5432 (configurada automaticamente).
3. Vá ao arquivo application.properties e retire o comentário da seguinte linha:
`spring.jpa.hibernate.ddl-auto=create`
4. Execute o main da classe InitialPoint
5. Acesse a aplicação no navegador , através da URL: `http://localhost:8080/ad-extreme`
6. Comente novamente a linha indicada no arquivo application.properties, para que os dados não sejam recriados e perdidos na próxima reexecução.

Como adicionar uma nova página HTML usando os templates do THYMELEAF

- A página propriamente dita deverá ficar no diretório /templates localizado em resources, e deverá seguir [thymeleaf](#) (template engine adotada).
- Para que a página fique acessível, deve haver uma configuração no controller . Por exemplo:

```
@RequestMapping(value = "/user", method = RequestMethod.GET)
public ModelAndView getPageIndexUser(){
    ModelAndView model = new ModelAndView();
    model.setViewName("user/index");

    return model;
}
```

- O método getPageIndexUser() faz com que quando o usuário fizer uma requisição do tipo GET no endpoint **"/user"**, a página que será retornada ao usuário é **user/index.html** que está localizada dentro de de resources/templates.