

# Projeto Final

## Programação Orientada a Objetos - MC302

Alunos: André Papoti, Bruno Falkenburg, Lucas Ramos,  
Nicolas França, Sophia Estrêla

Professora: Esther Colombini

Unicamp - Instituto de Computação

Junho de 2018

### Resumo

A intenção do projeto é ser um sistema interno de uma imobiliária, ou seja, é feito considerando os corretores como usuários e a imobiliária em si como o administrador (root). Esses usuários gerenciam imóveis, clientes, proprietários e propostas e o administrador gerencia os corretores.

É importante destacar que o usuário final são os corretores e outros possíveis funcionários da imobiliária (que é o caso do nosso administrador).

Queremos fazer buscas de imóveis do melhor agrado do cliente, assim como servir como um banco de dados afim de armazenar os dados do cliente e do proprietário em si.

Queremos enfatizar que o projeto não foi pensado para acesso direto de terceiros, como compradores e proprietários, pois a nossa regra de negócio diz que esses dados são sigilosos.

Agora em sua etapa final o sistema dá sugestões de imóveis do agrado do cliente, cruzando suas preferências com os imóveis disponíveis.

O sistema foi pensado consultando uma corretora real, de forma que este se adapte às necessidades de um usuário do mundo real. Este também trará o benefício de tornar processos do dia-a-dia de uma imobiliária, como a oficialização de uma proposta, se tornar mais fácil e ágil.

## 1 Funcionalidades

### 1.1 Funcionalidades Implementadas

Foram implementadas as seguintes funcionalidades nesta etapa do projeto:

1. Adição de novos Corretores pelo Administrador;
2. Adição de novos Imóveis;
3. Adição de novos Corretores
4. Adição de novos Clientes;
5. Adição de Propostas;
6. Adição de Proprietários dos imóveis;
7. Remoção de Corretores pelo Administrador;
8. Remoção de Imóveis;
9. Remoção de Corretores
10. Remoção de Clientes;
11. Finalização de Propostas: propostas pode estar ativas ou não;
12. Listagem dos Corretores pelo Administrador;
13. Listagem dos Imóveis relacionados as preferências de um cliente;
14. Listagem das Propostas cujo o Corretor foi intermediário;
15. Listagem dos Clientes cujo responsável é o Corretor;
16. Listagem dos Proprietários de todos os Imóveis disponíveis no sistema;
17. Salvar dados no banco de dados;
18. Realizar buscas banco de dados;

## **2 Ferramentas Utilizadas**

### **2.1 Lagom**

O Lagom Framework será utilizado para desenvolver serviços que serão consumidos pelo sistema. Entende-se Framework como uma coletânea de códigos genéricos - ou seja, que podem ser utilizados em diversos projetos - que têm por finalidade auxiliar o programador no desenvolvimento, servindo de esqueleto para o código que será escrito. Ao contrário de bibliotecas, frameworks ditam o fluxo de controle da aplicação.

O framework contém um conjunto de APIs que facilitam ao desenvolvedor o trabalho de escrever microserviços em Java, que podem fazer uso de ferramentas já incluídas no Lagom, como servidores para permanência dos dados e compartilhamento de informações com outros serviços. Todas

as ferramentas utilizadas e os serviços desenvolvidos podem ser inicializados com um único comando, ou separadamente.

Microserviço é uma abordagem que visa construir um sistema como um conjunto de pequenos serviços, com funcionalidades específicas e completas, e que se comunicam por meios leves, havendo baixa dependência entre os módulos. Este tipo de arquitetura colabora com a manutenção da modularidade, permitindo que alterações em serviços específicos não alterem o resultado do sistema como um todo. A estrutura dos serviços Lagom segue firmemente este modelo, fortalecido pela separação entre a declaração da interface do serviço e a implementação em si.

O Lagom oferece também a possibilidade dos microserviços nele construídos consumirem serviços externos, que não precisam seguir a estrutura dos microserviços Lagom. Isto será utilizado, por exemplo, quando for preciso extrair dados de rotas utilizando a API do Google Maps para descobrir uma boa rota entre os imóveis que um corretor deseja mostrar a um cliente.

Das APIs que o Lagom fornece, podem-se destacar a Service API, útil para as declarações das interfaces dos serviços desenvolvidos, bem como sua implementação, e a Persistence API, que auxilia no controle da persistência de dados. Apesar de não ser o banco de dados padrão, o Lagom tem suporte ao PostgreSQL, que será futuramente utilizado.

No momento, há escrito um conjunto de instruções básicas sobre as funcionalidades do Lagom, bem como instruções de instalação e configuração. O texto está disponível no arquivo "DocumentacaoLagom.pdf" localizado no diretório raiz do primeiro projeto ("Projeto1/Documentacao/Lagom/").

Por enquanto as classes relacionadas ao Lagom estão com algum defeito, mas como não é o foco nesse momento, decidimos adiar o tratamento dos erros.

## 2.2 PostgreSQL

PostgreSQL é um SGBD (Sistema Gerenciador de Banco de Dados) objeto-relacional. Isso significa que os dados no banco são modelados como entidades relacionadas, semelhantes a tabelas, acrescidos de estruturas típicas de orientação a objetos. A linguagem utilizada no PostgreSQL é a SQL.

Será utilizado para criação e gerenciamento de um banco de dados que armazenará os dados referentes às entidades do sistema.

## 2.3 React

Durante a conclusão do nosso projeto, além do Lagom/Maps/PostgreSQL, queremos montar uma página web baseada no Framework Javascript chamado React. O React foi criado pelo facebook principalmente para facilitar a criação de páginas que se atualizam a todo momento e que utilizam APIs

externas a todo momento. Esse tipo de trabalho sem o React precisa de tratamento de DOM e Ajax, que não é uma tarefa muito trivial.

Dessa forma, o nosso Front-end vai utilizar um serviço externo, o Lagom vai gerar a API do Backend e o React vai consumir essa API e mostrar para o usuário final através do navegador. Dessa forma, o cliente do nosso projeto vai usar um programa Backend em Java sem necessariamente precisar ter o Java instalado, trazendo maior flexibilidade para o sistema.

## 2.4 Git e GitHub

O GitHub é uma plataforma que permite hospedar e compartilhar arquivos, com foco em arquivos de código-fonte.

Por utilizar o Git para controle de versão, os programadores podem trabalhar em ramificações locais do projeto e enviar ao repositório hospedado no GitHub os arquivos que trabalharam, registrando todas alterações e permitindo que outro desenvolvedor que esteja trabalhando no projeto possa permanecer atualizado sobre o progresso do outro.

Por conta dos benefícios que esta plataforma traz, o grupo está utilizando para controle do código - garantindo que todos estejam com versões atualizadas e possam com a mesma facilidade revisar e alterar o próprio código ou o de outro membro da equipe - e dos demais arquivos relacionados ao projeto.

## 2.5 Trello

O Trello é o nosso organizador de projetos. Com ele temos um board, que tem um conjunto de listas. E cada lista tem é um conjunto de card. Como usamos o método Kanban, os cards transitam entre lists até chegar a lista "Conclusão".

A estrutura do nosso board pode ser vista na Figura 1.

O nosso board é dividido em 6 listas: Informações, Ideias, Objetivos, Execução, Revisão e Conclusão.

Informações: Tem cards com informações gerais, links, recursos e descrições e documentações das ferramentas que estamos utilizando.

Ideias: Lista para cards de ideias gerais.

Objetivos: Lista com os objetivos futuros que estamos galgando.

Execução: Lista com as tarefas que estão sendo executadas naquela fase do projeto.

Revisão: Revisão das tarefas terminadas em Execução.

Conclusão: Tarefas que foram em grande parte feitas.

## 2.6 Google Hangouts

O Google Hangouts é o serviço padrão da Google para comunicação em tempo real. Usamos muito durante nossas reuniões remotas. Normal-



Figura 1: Como o Trello é usado no nosso projeto

mente nos encontramos mais online do que fisicamente devido aos diferentes horários em comum entre os membros.

Conseguimos com ele compartilhamento de tela compartilhamento de áudio e essas features são as mais usadas entre nós.

## 2.7 Google Drive

Usamos para ter os dados mais atualizados possível de forma sincronizada. O Trello e o draw.io tem integração com o Drive que usamos constantemente entre nós.

## 2.8 draw.io

Para a criação do Diagrama de Classes UML presente neste arquivo foi utilizado o draw.io, uma ferramenta online de criação e edição de diversos tipo de diagramas. O site permite integração dos diagramas com o Google Drive, que é útil para compartilhamento dos diagramas e edição simultânea por mais de um membro.

## 2.9 Whatsapp

Usado para comunicação rápida entre os membros.

## 2.10 LaTeX

O  $\text{\LaTeX}$  é utilizado para a formatação do nosso relatório final sobre o projeto. O arquivo lido atualmente foi compilado pelo LaTeX e nos proporciona

vários benefícios relacionados a escrita. É uma ferramenta com grande uso pela comunidade acadêmica.

### **2.11 WindowsBuilder**

O WindowsBuilder é um plugin para Eclipse para facilitar a criação de interfaces gráficas (GUI). Nessa fase do projeto essa ferramenta vai nos ajudar a entregar parte de um conjunto de interfaces gráficas simples.