**Descrição da Arquitetura do software sisImobiliária**

Este documento possui informações e descrição da arquitetura de um sistema para controle imobiliário aqui denominado sisImobiliária. A documentação foi feita utilizando as recomendações da norma ISO/IEC IEEE 42010.

**1. Introdução**

**1.1. Stakeholders**

a) Os administradores, são os funcionários da imobiliária responsáveis por gerenciar imóveis.

b) Os compradores, são os clientes interessados obter um ou mais imóveis.

c) Os corretores, que são funcionários da imobiliária responsáveis pela analise física de imóveis.

d) Os desenvolvedores do sistema, alunos de graduação de Ciência da Computação e Engenharia da computação, matriculados na disciplina de Engenharia de Software 2.

e) Os locatários, são os clientes interessados em alugar um ou mais imóveis.

f) Os proprietários, são os clientes interessados em disponibilizar um ou mais imóveis para aluguel ou venda.

**1.2. Concerns**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Administrador | Comprador | Corretor | Desenvolvedor | Locatário | Proprietário |
| Funcionalidade | 2 | 2 | 2 | 2 | 2 | 2 |
| Viabilidade | 1 | 0 | 0 | 2 | 0 | 0 |
| Usabilidade | 2 | 0 | 2 | 2 | 0 | 0 |
| Recursos do sistema | 2 | 0 |  |  |  |  |
| Propriedades |  |  |  |  |  |  |
| Confiabilidade |  |  |  |  |  |  |
| Estrutura |  |  |  |  |  |  |
| Comportamento |  |  |  |  |  |  |
| Desempenho | 2 | 2 | 2 | 2 | 2 | 2 |
| Utilização de Recurso |  |  |  |  |  |  |
| Segurança | 1 | 2 | 1 | 2 | 2 | 2 |
| Assegurar Informação |  |  |  |  |  |  |
| Complexidade | 0 | 0 | 0 | 2 | 0 | 0 |
| Evolutividade | 0 | 0 | 0 | 2 | 0 | 0 |
| Qualidade de serviço |  |  |  |  |  |  |
| Flexibilidade |  |  |  |  |  |  |
| Agilidade |  |  |  |  |  |  |
| Modificabilidade |  |  |  |  |  |  |
| Modularidade |  |  |  |  |  |  |
| Controle |  |  |  |  |  |  |
| I.P.C. |  |  |  |  |  |  |
| Mudança de estado |  |  |  |  |  |  |
| Integração de subsistema |  |  |  |  |  |  |
| Acessibilidade de dados |  |  |  |  |  |  |
| Privacidade |  |  |  |  |  |  |
| Conformidade com o regulamento |  |  |  |  |  |  |
| Experiencia do usuário |  |  |  |  |  |  |
| Manutenibilidade |  |  |  |  |  |  |
| Acessibilidade |  |  |  |  |  |  |
| Descartabilidade |  |  |  |  |  |  |

1. Nenhum interesse
2. Pouco interesse
3. Muito interesse

**2. Visão Lógica**

**2.1. Diagramas de Classe e Caso de Uso**

**3. Visão de Processos**

**3.1 Requisitos não funcionais e as decisões tomadas**

**3.2 Decisões Arquiteturais Tecnologia**

**Nome:** Definição da Linguagem de programação

**Estado atual:** Definido

**Grupo de decisão:** Desenvolvedores (Todos os integrantes)

**Problema:** Uma linguagem que todos os integrantes tenham algum conhecimento e que se mantenha no mercado a longo prazo é fundamental para o desenvolvimento do sistema.

**Decisão/solução:** A linguagem escolhida foi Java 8.2. Pois é uma versão estável e uma linguagem conhecida por todos os desenvolvedores e a aplicação é desktop.

**Alternativas:** Foram Consideradas as linguagens JSP para o caso da aplicação ser Web com o Java Servlet.

**Decisões relacionadas:** -

**Preocupações (Concerns) relacionadas: -**

**Histórico: -**

**- Decisão Arquitetural:**

**Nome:** Escolha do Ambiente de Desenvolvimento (IDE)

**Estado atual:** Definido

**Grupo de decisão:** Desenvolvedores

**Problema:** Deve-se escolher um ambiente de desenvolvimento em que todos os desenvolvedores, independente do SO da sua máquina, possa fazer alterações no projeto.

**Decisão/solução:** NetBeans IDE 8.2. Pois além do suporte à linaguem de programação escolhida, o NetBeans possui um ambiente de criação de telas que facilita o desenvolvimento das mesmas.

**Alternativas:** Eclipse IDE 2018/12.

**Decisões relacionadas:** Definição da Linguagem de Programação

**Preocupações (Concerns) relacionados: -**

**Histórico: -**

**- Decisão Arquitetural:**

**Nome:** Definiçao da ferramenta de controle de versão

**Estado atual:** Definido

**Grupo de decisão:** Desenvolvedores

**Problema:** Todos os desenvolvedores devem possuir a mesma versão do projeto em produção. Além disso alguma ferramenta deve ser utilizada para manter salvo as alterações feitas no sistema caso necessite o retorno à uma versão mais antiga do mesmo.

**Decisão/solução:** GitHub. Pois é uma ferramente grátis e que pode incluir várias pessoas a um mesmo projeto sem custo adicional. Além disso, por ser popular, a curva de aprendizado é menor pois possui muitos materiais disponíveis.

**Alternativas: -**

**Decisões relacionadas: -**

**Preocupações (Concerns) relacionadas: -**

**Histórico: -**

**- Decisão Arquitetural:**

**Nome:** Definição do SGBD

**Estado atual:** Definido

**Grupo de decisão:** Desenvolvedores

**Problema:** O sistema a ser desenvolvido precisa manter os dados amarzenados de forma persistente, logo precisamos de um banco de dados e um sistema de gerenciamento de banco de dados.

**Decisão/solução:** PostgresSQL 11. Pois é uma ferramenta grátis e de conhecimento da equipe de desenvolvedores responsável pelo desenvolvimento das classes de acesso ao banco de dados.

**Alternativas:** MySQL

**Decisões relacionadas: -**

**Preocupações (Concerns) relacionadas: -**

**Histórico: -**

**- Decisão Arquitetural:**

**Nome:** Adição do framework Hibernate

**Estado atual:** Definido

**Grupo de decisão:** Isaquel e Welerson Melo

**Problema:** A criação das tabelas do banco de dados e acesso ao banco de dados é feitas de forma manual e podem demandar algum tempo do projeto.

**Decisão/solução:** Utilizar o framework Hibernate para auxiliar na criação das tabelas e consultas ao banco de dados, agilizando o processo de desenvolvimento. É uma ferramenta de fácil aprendizado e que não precisará ser aprendido por toda a equipe de desenvolvimento.

**Alternativas: -**

**Decisões relacionadas:** -

**Preocupações (Concerns) relacionadas: -**

**Histórico: -**

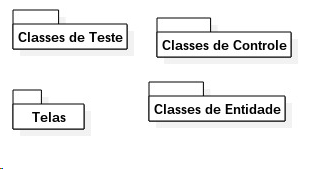
**- Decisão Arquitetural:**

**4. Visão Desenvolvimento**

**4.1 Organização dos pacotes e camadas**

O sistema foi dividido em 3 camadas. A camada View, que possui as telas. A camada de Controle, a qual está todas as classes de controle que se comunica com as telas e as classes de entidade. E a camada de Acesso, ao qual possui as classes de entidade que possuí as regras de negócio e acesso ao banco de dados.

Pacotes do Sistema



**5. Cenários**