CENTRO UNIVERSITÁRIO UNIEVANGÉLICA

ENGENHARIA DE COMPUTAÇÃO

Real Imóveis

Documento de Arquitetura de Software

Versão <1.0>

Autor:

Elson Bento

Karina Hellen

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 03/05/2017 | 1.0 | Início do documento. | Elson Bento |
| 08/05/2017 | 1.1 | Alteração no campo 5.2 | Karina Hellen |

# **Sumário**

[1. Sumário 3](#_Toc469146331)

[1. Introdução 4](#_Toc469146332)

[1.1 Finalidade 4](#_Toc469146333)

[1.2 Escopo 4](#_Toc469146334)

[1.3 Definições, acrônimos e abreviações 4](#_Toc469146335)

[1.4 Visão geral 5](#_Toc469146336)

[2. Representação arquitetural 5](#_Toc469146337)

[3. Metas e restrições da arquitetura. 6](#_Toc469146338)

[4. Visão de casos de uso 6](#_Toc469146339)

[4.1 Realizações de casos de uso 6](#_Toc469146340)

[5. Visão lógica. 7](#_Toc469146341)

[5.1. Visão geral. 7](#_Toc469146347)

[5.2. Pacotes de design significativos do ponto de vista da arquitetura. 8](#_Toc469146348)

[6. Visão de implantação. 9](#_Toc469146349)

[7. Visão de implementação. 9](#_Toc469146350)

# Introdução

É um sistema web que possibilita que os locadores divulguem, através de um cadastro na plataforma, seus estabelecimentos de forma detalhada. O objetivo do sistema é aproximar o locador do locatário, além de gerenciar e controlar as informações em relação ao aluguel.

## Finalidade

Este documento oferece uma visão geral arquitetural de todo o sistema, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema. Projetistas e desenvolvedores devem fazer uso deste documento para obter detalhes sobre a visão lógica do sistema, ou seja, seu esquema de pacotes, subsistema, organização em camadas e demais componentes.

## Escopo

Desenvolver um sistema que possibilite ao usuário visualizar quais são as opções de aluguel disponível mais próximas. O contato entre locador e locatário será realizado através do sistema. Ao fechar contrato, as informações sobre pagamento poderão ser gerenciadas pelo sistema.

## Definições, acrônimos e abreviações

Abaixo são apresentadas definições que consideramos importantes para o entendimento do documento de arquitetura de *software*:

|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| Manter | Cadastrar, alterar, consultar e excluir/desativar. |
| Usabilidade | Que se compreende bem, que é fácil de entender; claro, compreensível. |
| Manutenibilidade | Capacidade do produto de software ser modificado. |
| Implantação | Disponibilização do processo ou rotina de trabalho para o usuário final. |
| Implementação | Introduzir algum processo ou protocolo em um determinado local. |
| MVC | Padrão de arquitetura de software onde M significa modelo *(Model)* sendo responsável pela parte de regras de negócio, V *(View)* a visualização responsável pela parte de interfaces e C *(Controler)* a parte de controle dos dados. |
| PHP | Linguagem de programação cujo acrônimo é: Hypertext Preprocessor. |

## Visão geral

A visão lógica define a estrutura lógica da arquitetura. Ela consiste em descrever as camadas que compõem a aplicação definindo as responsabilidades de cada camada. O modelo que será usado é *Model-view-controller (MVC)* é um padrão de arquitetura que divide a aplicação em controladores que tratam as entradas dos usuários no modelo que prove as funcionalidades principais, e nas visões que mostram as informações para os usuários.

# Representação arquitetural

A arquitetura do sistema é representada por modelos que mostram a visão do caso de uso, visão lógica e implantação. Esses modelos mostram como os artefatos estão distribuídos pela arquitetura e como que eles interagem entre si.

# Metas e restrições da arquitetura.

As restrições arquiteturais são:

* Utilização do paradigma Orientado a Objetos para o desenvolvimento;
* Estrutura MVC;
* Linguagem de programação PHP;
* O sistema deverá ser Web;
* Banco de dados MySQL;

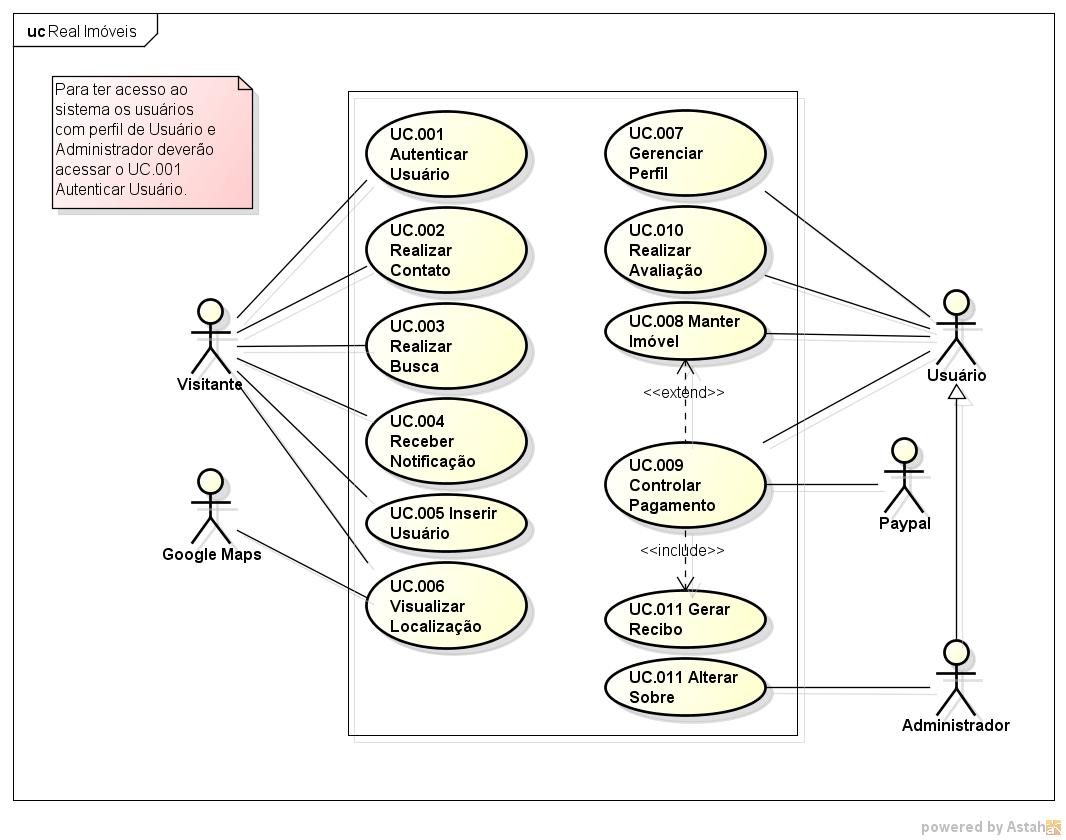
# Visão de casos de uso

A visão de casos de uso mostra um subconjunto do modelo de casos de uso e atores significativos para a arquitetura. Através dessa visão conseguimos representar as funcionalidades central e significativa do sistema, e os pontos importantes da arquitetura do software.

## Realizações de casos de uso

**Figura 01:** Diagrama de Caso de Uso**.**

**Ferramenta:** Astah**.**

****

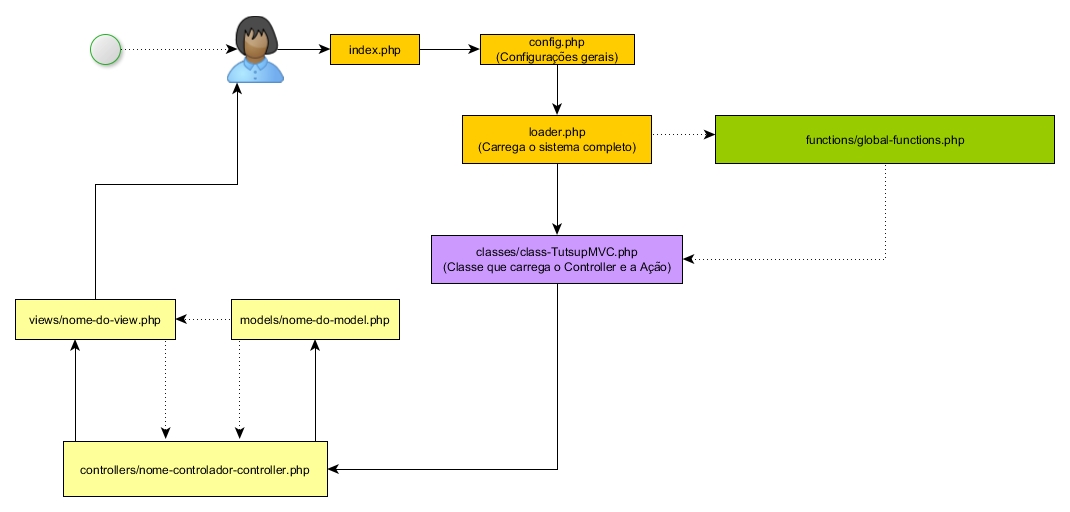
# Visão lógica.



## Visão geral.

A visão lógica define a estrutura da arquitetura. Abaixo será especificado o padrão utilizado para o desenvolvimento do sistema, no caso, MVC.

**Figura 02: Aplicação MVC em PHP.**



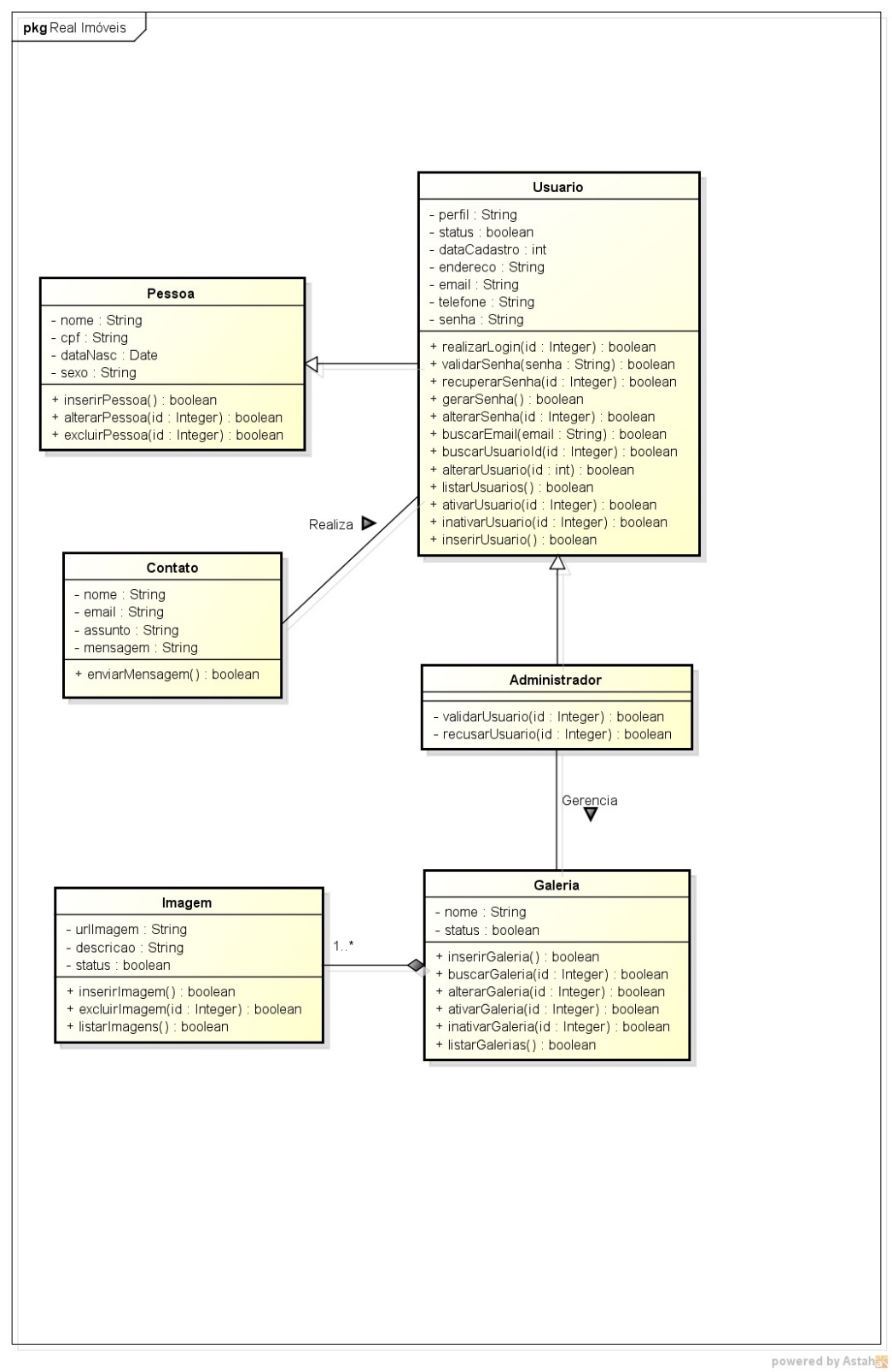
Onde:

* *View*: pacote que contém as visões do projeto, ou seja, as interfaces, formulários e etc.;
* *Controller*: pacote que recebe as informações e requisições do pacote *view* e os despacha para devida classe de controle, o pacote *controller* implementa o pacote *BusinessLogic:*o qual contém as regras de negócio do sistema;
* *Action*: pacote que recebe as informações e requisições do pacote *controller* e os atribui às respectivas classes do pacote *bean* e do pacote *persistence*;
* *Bean*: pacote que recebe e armazena as informações referentes a seus respectivos objetos;
* *Persistence*: pacote que recebe as requisições e realiza as operações relacionadas ao Banco de Dados, utilizando dados provindos do pacote *bean*. O pacote *persistence* também pode enviar dados para o pacote *view*;

## Pacotes de design significativos do ponto de vista da arquitetura.

**Figura 03: Diagrama de Classes.**

**Ferramenta: Astah.**

****

# Visão de implantação.

Para implantação do Real Imóveis escolhemos um servidor web que é responsável por armazenar e trocar informações com outras máquinas. Portanto, pelo menos dois participantes são envolvidos em cada troca de informações: um cliente, que solicita informações, e um servidor, que atende a esses pedidos.

# Visão de implementação.

O sistema será implementado utilizando conceitos de Programação Orientada a Objetos e Estrutura MVC. As ferramentas utilizadas serão: AngularJS, Laravel framework PHPe Banco de dados MySQL.

**8. Tamanho e desempenho.**

Não há restrições explicitas relacionada a tamanho ou desempenho.

**9. Qualidade.**

Para a qualidade do sistema deverá atender às seguintes características:

* *Usabilidade:* O sistema deverá ser inteligível contendo uma interface amigável e intuitiva com todas as informações necessárias. O acesso deve ser o mais apreensivo possível, de maneira que uma pessoa com poucos conhecimentos em informática consiga acessar as informações sem dificuldades. O usuário não deverá ter que passar por muitos menus, a busca poderá ser realizada indicando apenas a localização ou então os filtros desejados pelo cliente. O menu constará opções visíveis e atrativas para o cliente.
* *Eficiência:* Tempo de resposta terá de ser de no máximo 05 segundos para carregamento do aplicativo e retornará resultados da busca em no máximo 07 segundos. O software deverá estar dentro dos padrões de conformidade da norma ISO.
* *Segurança:* O produto de software deverá ter a capacidade de apresentar níveis aceitáveis de riscos de danos a pessoas, negócios, software, propriedades ou ao ambiente, em um contexto de uso especificado.