**Orpha**

**Orpha Documento de Arquitetura de Software**

**Versão <1.0>**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 19/10/17 | 1.0 | Elaboração do documento | Marcos Natã Santos |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Índice Analítico**

1. [Introdução 4](#_TOC_250015)
   1. [Finalidade 4](#_TOC_250014)
   2. [Escopo 4](#_TOC_250013)
   3. [Definições, Acrônimos e Abreviações 4](#_TOC_250012)
   4. [Visão Geral 4](#_TOC_250011)
2. [Representação Arquitetural 4](#_TOC_250010)
3. [Metas e Restrições da Arquitetura 5](#_TOC_250009)
4. [Visão de Casos de Uso 5](#_TOC_250008)
   1. [Realizações de Casos de Uso 6](#_TOC_250007)
5. [Visão Lógica 6](#_TOC_250006)
   1. [Visão Geral 6](#_TOC_250005)
   2. [Pacotes de Design Significativos do Ponto de Vista da Arquitetura 8](#_TOC_250004)
6. [Visão de Implantação 8](#_TOC_250003)
7. [Visão da Implementação 8](#_TOC_250002)
8. [Tamanho e Desempenho 8](#_TOC_250001)
9. [Qualidade 8](#_TOC_250000)

**Documento de Arquitetura de Software**

# Introdução

O sistema Orpha tem como finalidade gerir o processo de entrada e saídas de crianças/adolescentes em orfanatos, gerando relatórios durante seu ciclo de vida.

## Finalidade

Este documento oferece uma visão geral arquitetural abrangente do 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.

## Escopo

Este documento auxilia os envolvidos no projeto a captar aspectos arquiteturais do sistema que são necessários para o desenvolvimento de um sistema que atenda às necessidades dos usuários finais. Além de auxiliar no entendimento do sistema por novos membros da equipe.

## Definições, Acrônimos e Abreviações

* + - MVC – Padrão de arquitetura de software onde M significa modelo sendo responsável pela parte de regras de negócio, V a visualização responsável pela parte de interfaces e C a parte de controle dos dados.

## Visão Geral

São apresentados ainda neste documento diferentes visões arquiteturais de como o sistema deve se comportar em diferentes processos, como deve ser implantado é implementado e restrições de desempenho e qualidade.

# Representação Arquitetural

* Visão de caso de uso
  + Apresenta as funcionalidades arquiteturais importantes e os usuários do sistema.
* Visão lógica
  + Descreve as classes e sua organização e apresenta o padrão de arquitetura que deverá ser utilizado para o desenvolvimento do sistema.
* Visão de processos
  + Mostra o padrão de comportamento do sistema diante de diferentes ações do usuário.
* Visão de implantação
  + Descreve a estrutura do ambiente onde o software será instalado.
* Visão de implementação
  + Ilustra a distribuição do processamento em um conjunto de nós do sistema, incluindo a distribuição física dos processos e threads.

# Metas e Restrições da Arquitetura

Existem algumas restrições de requisito e de sistema principais que têm uma relação significativa com a arquitetura, sendo elas:

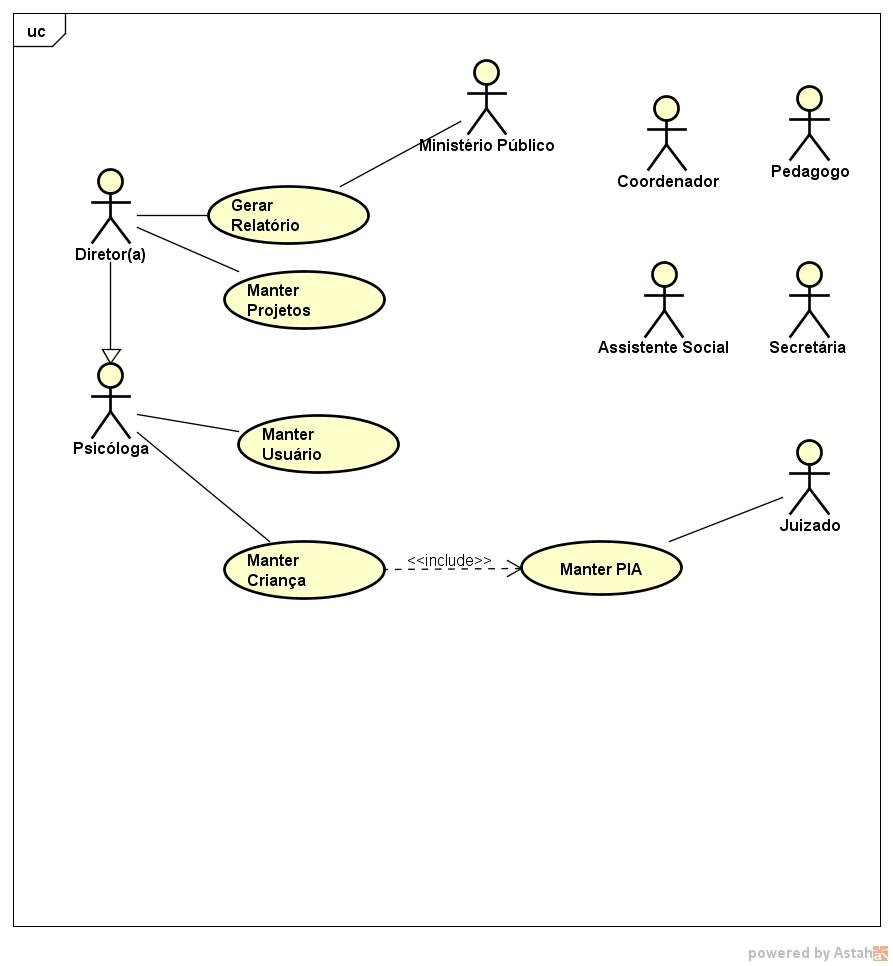
* Utilização do paradigma Orientado a Objetos para o desenvolvimento;
* Estrutura MVC;
* Linguagem de programação PHP/Laravel/Angular;
* O sistema em questão deverá ser Web/Responsivo;
* Banco de dados MySQL;

# Visão de Casos de Uso

Os casos de uso do sistema Orpha serão listados abaixo:

* Autenticar Usuário;
* Consultar Dashboard;
* Manter Usuário;
* Manter Criança/Adolescente;

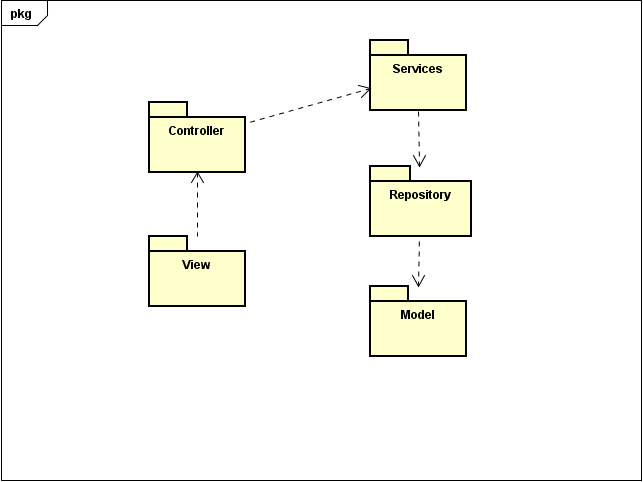
## Realizações de Casos de Uso



# 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.



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 complementa* o pacote serviceo qual contém as regras de negócio do sistema;
    - *Repository*: pacote que recebe as requisições e realiza as operações relacionadas ao Banco de Dados, utilizando dados provindos do pacote *Model*. O pacote *repository* também pode enviar dados para o pacote *view*;
    - *Service*: pacote que implementa toda a regra de negócio da aplicação.

## Pacotes de Design Significativos do Ponto de Vista da Arquitetura

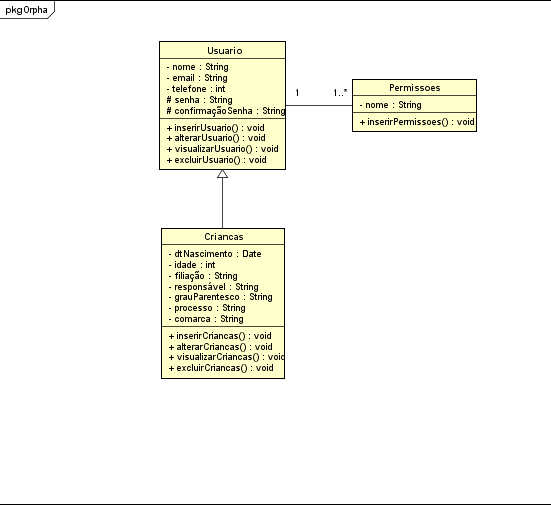


Diagrama de Classe geral do sistema.

# Visão de Implantação

O sistema será implantado no servidor da Heroku, no qual será disponibilizado para acesso através da web.

# Visão da Implementação

O sistema será implementado utilizando conceitos de Programação Orientada a Objetos, Estrutura MVC, *Laravel*, *Web Container Apache* e Linguagem de Programação PHP e Banco de Dados MySQL.

# Tamanho e Desempenho

Não há restrições explicitas relacionadas a tamanho ou desempenho.

# Qualidade

O padrão de arquitetura adotado no projeto tem como finalidade garantir uma melhor organização do código-fonte, o que auxilia na manutenibilidade do software, bem como a portabilidade do mesmo.