
Donar

Donar
Arquitetura de Software
Versão <1.4>

Histórico da Revisão

Data	Versão	Descrição	Autor
27/03/18	1.0	Elaboração do documento	Matheus Lima, Samuel Souza
10/04/18	1.1	Revisão do Documento	Gabriel L. Dias, Lucas Andrade
05/06/18	1.2	Revisão do Documento	Érico Junior, Vitor Augusto
07/08/18	1.3	Revisão do Documento	Josimar Candido
14/10/18	1.4	Revisão do Documento	Gabriel L. Dias

Sumário

1.	Introdução	5
1.1	Finalidade	5
1.2	Escopo	5
1.3	Definições, Acrônimos e Abreviações	5
1.4	Visão Geral.....	5
2.	Representação Arquitetural.....	5
3.	Metas e Restrições da Arquitetura	6
4.	Visão de Casos de Uso.....	6
5.	Visão Lógica.....	7
5.1	Visão Geral.....	7
5.2	Pacotes de Design Significativos do Ponto de Vista da Arquitetura.....	8
6.	Visão de Implantação.....	8
7.	Visão da Implementação.....	8
8.	Tamanho e Desempenho.....	9
9.	Qualidade.....	9

1. Introdução

O presente documento tem como objetivo descrever o documento de arquitetura do projeto Donar. Este sistema tem como finalidade o aumento do índice de doações de sangue no Brasil, mediante disseminação dos direitos dos doadores e realização de campanhas que podem ser compartilhadas nas redes sociais.

1.1 Finalidade

Este documento oferece uma visão arquitetural geral do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do mesmo. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas no decorrer do processo de desenvolvimento.

1.2 Escopo

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

1.3 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.

1.4 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.

2. 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.

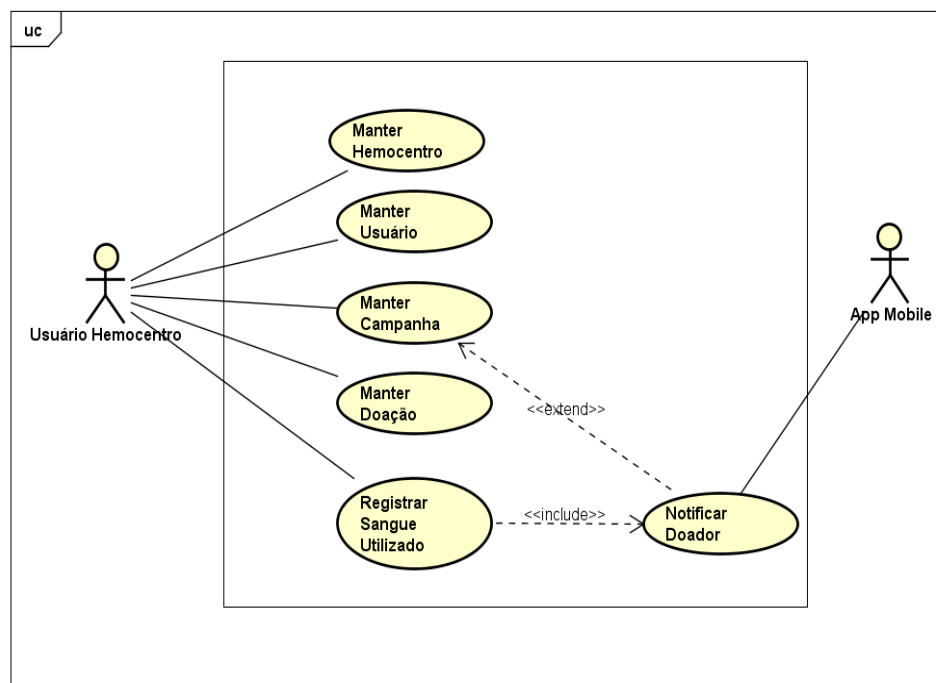
3. Metas e Restrições da Arquitetura

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 TypeScript;
- Linguagem de programação PHP;
- Framework *Ionic* e *Laravel*;
- O sistema em questão deverá ser multiplataforma;
- Bancos de dados *Firebase* e *MySql*.

4. Visão de Casos de Uso

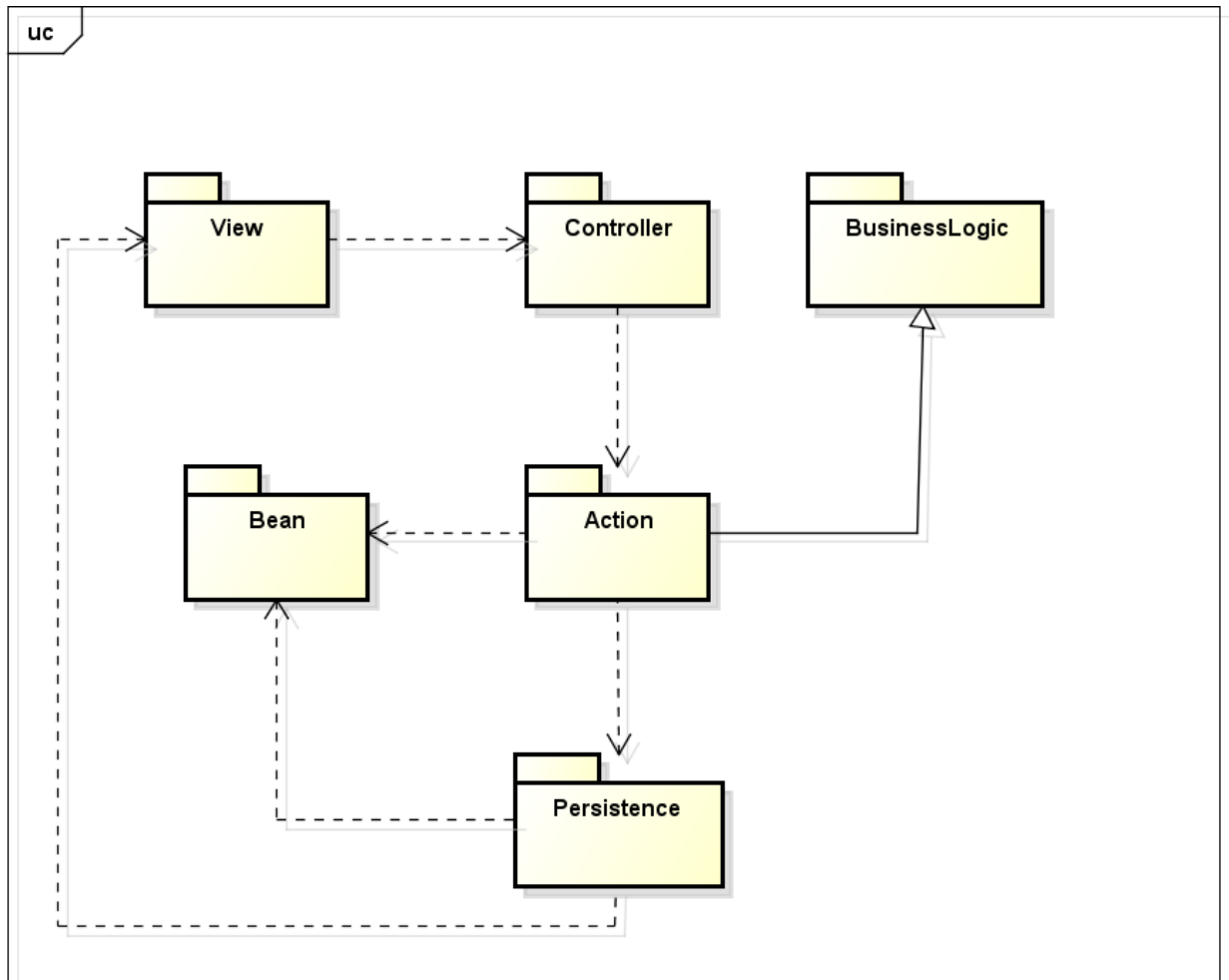
Os casos de uso do sistema Donar serão listados abaixo:



5. Visão Lógica

5.1 Visão Geral

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

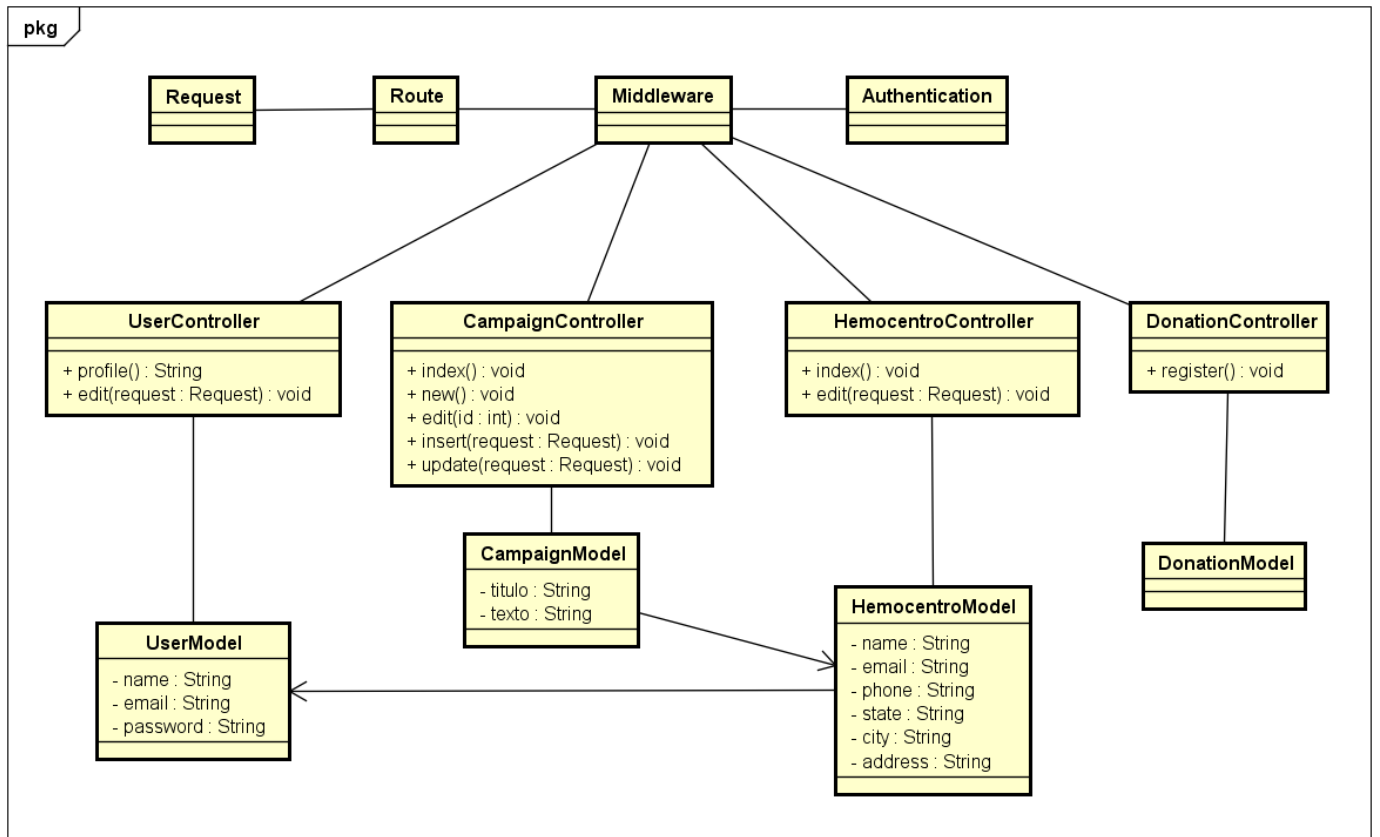


powered by astah

- *View*: Componente que contém as visões do projeto, ou seja, as interfaces, formulários e etc.;
- *Controller*: Componente que recebe as informações e requisições do pacote *view* e os despacha para devida classe de controle, o *controller* se comunica com classes *BusinessLogic* que contém as regras de negócio do sistema;
- *Action - Model*: pacote que recebe as informações e requisições do pacote *controller* e os atribui às respectivas classes do pacote *bean* e do pacote *persistence*, utilizando dados de seus modelos;

- *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*.

5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura



powered by Astah

Diagrama de Classe geral do sistema.

6. Visão de Implantação

O sistema será implantado quando for validado entre os *stakeholders*.

7. Visão da Implementação

O sistema será implementado utilizando conceitos de Programação Orientada a Objetos através do framework *Ionic* e *Laravel*, estrutura *MVC*, *TypeScript*, *Web Container Apache*, Linguagem de Programação *PHP* e Bancos de Dados *Firebase* e *MySQL*.

8. Tamanho e Desempenho

A aplicação deve rodar através de um site próprio na versão web para os hemocentros e por meio de dispositivos móveis na versão mobile que será para o uso do usuário doadores.

9. 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.