

## Sistema Doação Inteligente

Data	Versão	Descrição	Au- tor
06/10/2021	1.0	Criação da Documentação	
11/10/2021	1.1	Desenvolvimento dos requisitos	
14/10/2021	1.2	Desenvolvimento das regras de negócio e requisitos não funcionais.	
24/10/2021	1.3	Desenvolvimento do Caso de Uso e Desenvolvimento do Mer	
25/10/2021	1.4	Desenvolvimento da arquitetura e Cronograma de desenvolvimento do sistema	

Introdução.....	3
Termos e Abreviações .....	3
Requisitos Funcionais .....	4
Requisitos não funcionais tecnológicos .....	5
Regras de Negócio .....	5
Arquitetura do Sistema.....	6
Mer (Modelo Entidade Relacionamento).....	6
Diagrama Caso de Uso.....	7
Cronograma de desenvolvimento do sistema.....	7

## Introdução

Este documento tem como finalidade coletar, analisar e definir necessidades e recursos que serão utilizados para o desenvolvimento do projeto Doação Inteligente, estabelecer dados para a modelagem do sistema, incluindo a definição para o desenvolvimento e implementação do projeto.

Este projeto tem como finalidade conectar quem precisa de doação com seus respectivos doadores.

## Termos e Abreviações

Spring Framework	O Spring é um framework open Source que fornece modelo abrangente de programação e configuração para aplicativos empresariais modernos baseados em Java.
Spring boot	O Spring Boot é um framework Java open Source que tem como objetivo facilitar a configuração e manutenção de aplicativos Spring.
Lombok	O Lombok é uma biblioteca Java open Source que tem como objetivo aumentar a produtividade do desenvolvimento e manutenção de sistemas Java por reduzir a criação de código repetitivo através de anotações no código fonte.
PostgreSQL	PostgreSQL é um sistema gerenciador de banco de dados objeto relacional, disponível sob a licença PostgreSQL, uma licença de código aberto.
Git	Git é um sistema de controle distribuído de versão gratuito e de código aberto projetado para lidar com tudo, desde projetos pequenos a muito grandes com velocidade e eficiência.
ReactJS	O React é uma biblioteca Javascript de código aberto com foco em criar interfaces de usuário em páginas web.
Material -UI	Material-UI é uma biblioteca React que visa facilitar o desenvolvimento de interfaces gráficas de aplicações web por meio da disponibilização de componentes reutilizáveis e customizáveis que seguem os padrões do Material Design.

## Requisitos Funcionais

Código	Descrição	Caso de Uso
RF01	Deve permitir ao usuário realizar o cadastro no sistema através de um formulário, será possível criar três tipos de usuários, <b>doador, receptor e Hemonúcleo</b> . Os dados dos usuários são: identificador único, tipo de usuário, login, senha, nome completo, CPF (ou CNPJ), RG, e-mail, telefone, data_nasc, celular, tipo sanguíneo, genero, admin, <b>Endereço</b>	UC01- Cadastrar usuários
RF02	Deve permitir que os <b>usuários</b> façam alterações em seus cadastros.	UC02- Editar usuário
RF03	Deve permitir que os <b>usuários</b> excluam seu cadastro, caso não façam mais parte do grupo de doadores ou não desejam receber mais doações.	UC03- Excluir usuário
RF04	Deve permitir que os <b>usuários</b> vejam os detalhes dos seus dados.	UC04- Detalhar usuário
RF05	Deve permitir a listagem dos <b>usuários</b> . Neste caso haverá a listagem dos doadores e a listagem dos receptores.	UC05- Listar usuário
RF06	O Sistema deve permitir o acesso dos <b>usuários</b> através de <b>login e senha</b> .	UC06- Fazer login
RF07	O Sistema deve permitir que o <b>usuário receptor</b> envie por meio de mensagem um pedido ao doador.	UC07- Enviar mensagem para o doador.
RF08	O Sistema deve permitir que o <b>usuário doador</b> receba uma mensagem de pedido de doação.	UC08- Receber Mensagem do receptor.
RF9	O Sistema deve permitir informações sobre a doação de sangue.	UC09- Publicar informações sobre doações.
RF10	O Sistema deve mostrar os pontos de doação a partir de um mapa.	UC10-Mostrar ponto de coleta no mapa.
RF11	O Sistema deve permitir que o <b>usuário</b> realize o <b>agendamento</b> da doação de sangue. O <b>agendamento</b> deve conter: Identificador único, horario, local, data agendada, <b>Usuário</b> .	UC11- agendar doação de sangue.
RF12	Permitir que o usuário do tipo hemonucleo veja a listagem dos <b>agendamentos</b> .	UC12- Listar os agendamentos de sangue.
RF13	O Sistema deve permitir que os usuários ( <b>doador, receptor e hemonucleo</b> ) gerenciem endereços no sistema. Um <b>endereço</b> possui: Identificador único, cep, rua, número, complemento, bairro, cidade, estado, país.	UC13- Manter endereço.

RF14	O Sistema deve permitir que o usuário do tipo <b>receptor</b> escreva um pedido de doação para o doador. O pedido deve conter: identificador único, descrição, tipo sanguíneo, data pedido, <b>Usuario</b> .	UC14- Requisitar pedido de doação.
------	--	------------------------------------

### Requisitos não funcionais tecnológicos

Código	Descrição
RNF01	O Sistema deverá ser desenvolvido para plataforma Web.
RNF02	A Interface deve ser agradável e intuitiva para o usuário.
RNF03	O Sistema deve apresentar a documentação técnica.

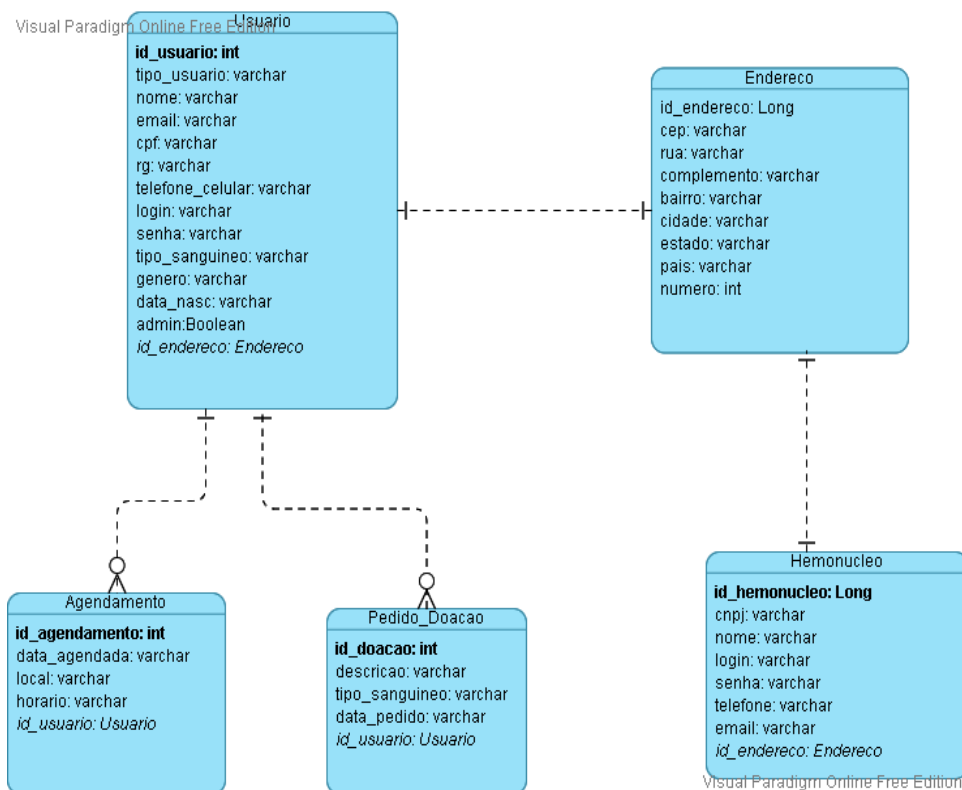
### Regras de Negócio

Código		Descrição
RN01	RF01	O cadastro não deverá ser realizado caso o usuário já esteja cadastrado no sistema.
RN02	RF02	O usuário terá permissão para alterar apenas o seu próprio cadastro.
RN03	RF06	Somente usuários cadastrados poderão realizar login no sistema.
RN04	RF05	Somente os usuários cadastrados do sistema poderão ver a lista de pessoas que precisam de doação.
RN05	RF05	Somente os usuários cadastrados do sistema poderão ver a lista de doadores.
RN06	RF06	Verificar se a senha do usuário está correta.
RN07	RF06	Caso a senha esteja errada exibir um erro de login.
RF08	RF13	Somente o Hemonucleo poderá ver a listagem de doações de sangue.
RF09		

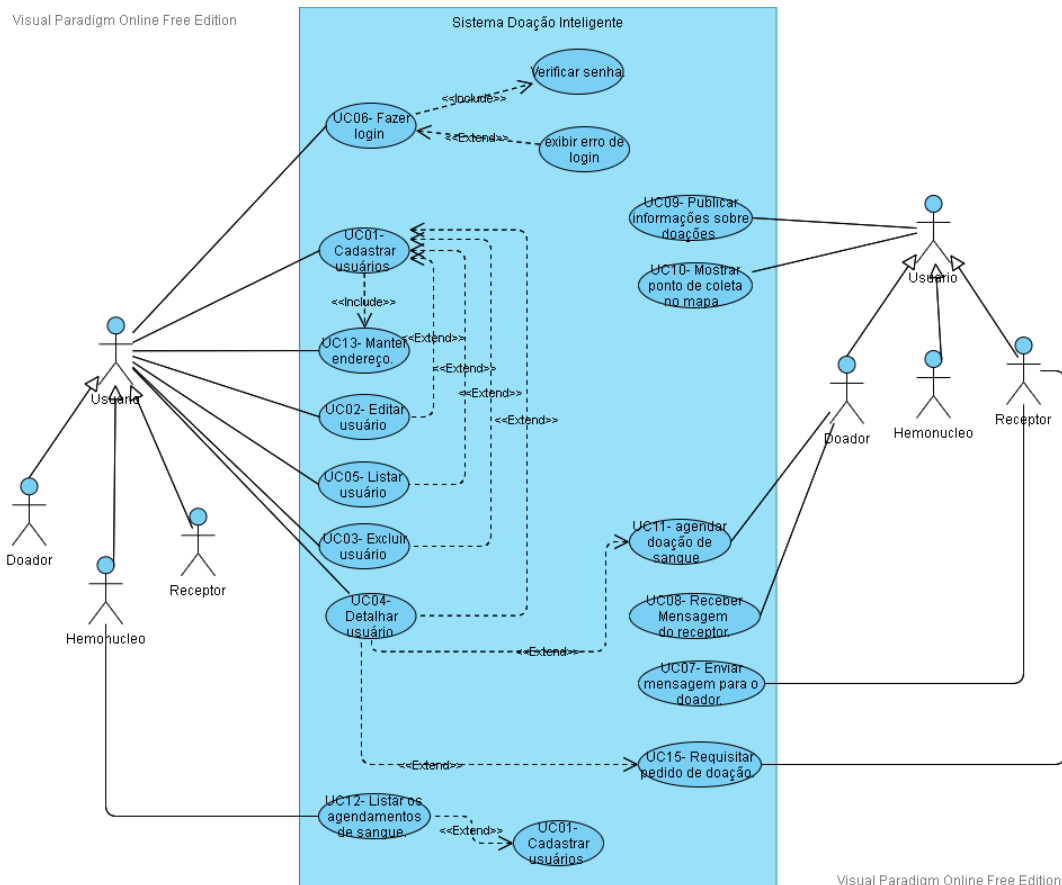
## Arquitetura do Sistema

Para este projeto ser construído foram escolhidas tecnologias de desenvolvimento tanto para o Front-End quanto para o Back-End. O Back-End encapsula a API REST construída com o framework Spring Boot e o container do banco de dados PostgreSQL. Por sua vez, o Front-End representa o lado cliente da arquitetura e pode ter múltiplas instâncias distintas, por exemplo, uma aplicação WEB construída com ReactJS, ambiente principal do projeto. Como critério de escolha foram levados em conta a experiência prévia e a facilidade no manuseio destas Tecnologias.

## Mer (Modelo Entidade Relacionamento)



## Diagrama Caso de Uso



## Cronograma de desenvolvimento do sistema

Atividade	Duração	Status	Início	Fim
1) Concepção	12 dias	Finalizado	31/08/2021	05/10/2021
2) Planejamento	21 dias	Finalizado	14/09/2021	05/10/2021
3) Análise	20 dias	Finalizado	06/10/2021	26/10/2021
4) Desenvolvimento	28 dias	Em andamento	27/10/2021	25/11/2021

