

O+: UMA APLICAÇÃO PARA GESTÃO DE HEMOCENTROS E AGENDAMENTO DE DOAÇÕES DE SANGUE

Aluno(a): Silvino Brigido de Souza

Orientador: Francisco Adell Péricas

Roteiro

- Introdução;
- Objetivos;
- Fundamentação Teórica;
- Trabalhos correlatos;
- Requisitos Funcionais;
- Requisitos Não Funcionais;
- Especificação;
- Implementação;
- Análise dos Resultados;
- Conclusões e Sugestões.

Introdução

- Uma das consequências *COrona Vlrus Disease* (COVID) 19 é a queda na doação de sangue;
- A necessidade dos bancos de sangue em manter os níveis de estoque de sangue;
- Auxiliar na gestão dos estoques de sangue e promover agendamentos e fidelizar doadores.

Objetivos

- O objetivo do trabalho é disponibilizar uma aplicação web para gestão de estoques de hemocentros e agendamento de doações de sangue
- Os objetivos específicos são:
 - a) desenvolver uma aplicação de fácil utilização seguindo as boas práticas e padrões de usabilidade para aplicações web;
 - b) automatizar o processo de agendamento de doações de sangue diretamente pelo cliente;
 - c) automatizar o processo de gestão de estoque pelos funcionários do hemocentro.

Fundamentação Teórica (1/3)

- Doação de Sangue:



Fundamentação Teórica (2/3)

- Gerenciamento de Estoque:

Vantagens dos Estoques	Desvantagens dos Estoques
Melhoram o nível de serviço	Ocupam grandes espaços, gerando gastos
Incentivam economias na produção e permitem economia de escala nas compras e no transporte	Podem encobrir problemas de qualidade
Agem como proteção contra aumentos de preços e servem como segurança contra contingências	Podem promover isolamento entre os canais da cadeia
Protegem a empresa de incertezas na demanda e no tempo de ressuprimento	Há riscos de obsolescência e deterioração

Fundamentação Teórica (3/3)

- Boas Práticas em Aplicações Web:
 - design Flexível;
 - reduzir a utilização da rede;
 - explorar funções específicas;
 - controle ao usuário.

Trabalhos Correlatos (1/3)

- Título: DOE+: Um aplicativo móvel de cunho social para agendamento de doação de sangue no hemocentro público de alagoas
- Características principais:
 - a) Sistema apenas para plataforma Android;
 - b) Permite apenas o agendamento e visualização de campanhas.

Trabalhos Correlatos (2/3)

- DOAR: Sistema web para otimização do processo de coleta de sangue
- Características principais:
 - a) Sistema web;
 - b) Sistema com integração com a plataforma do SUS (Hemovida);
 - c) Possui gerenciamento de estoque das doações e gestão de agendamento apenas (não cadastro).

Trabalhos Correlatos (3/3)

- BLOODSYS: Controlando o processo de doação de sangue para hemocentros
- Características principais:
 - a) Sistema web;
 - b) Apenas de cadastro de doadores e triagem feito diretamente pelo hemocentro.

Requisitos Funcionais

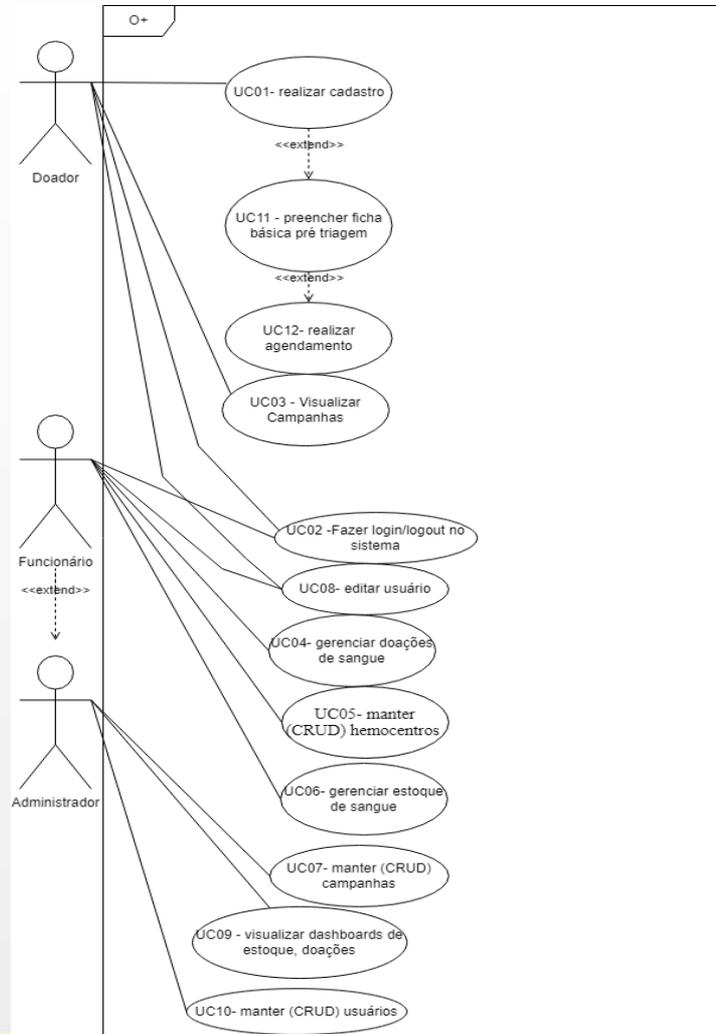
- A aplicação web proposta deve:
 - a) permitir manter (*Create, Read, Update, Delete* - CRUD) doadores;
 - b) permitir manter (CRUD) hemocentros;
 - c) permitir gerenciar doações de sangue;
 - d) permitir agendar doações de sangue;
 - e) gerenciar estoque de sangue;
 - f) emitir uma listagem de usuários aptos a doação, baseado em um tipo de sangue necessitado;
 - g) permitir o usuário com perfil funcionário visualizar os níveis de estoque de sangue em uma *dashboard*.

Requisitos Não Funcionais

- A aplicação web proposta deve:
 - a) respeitar a permissão de cada tela, de acordo com o tipo de usuário (doador, funcionário, administrador);
 - b) utilizar a plataforma Angular para o desenvolvimento da aplicação *front-end*;
 - c) utilizar a linguagem de programação Java para desenvolvimento do *backend* da aplicação;
 - d) utilizar os padrões de interface baseados no princípio do Material Design;
 - e) utilizar o banco de dados MySQL.

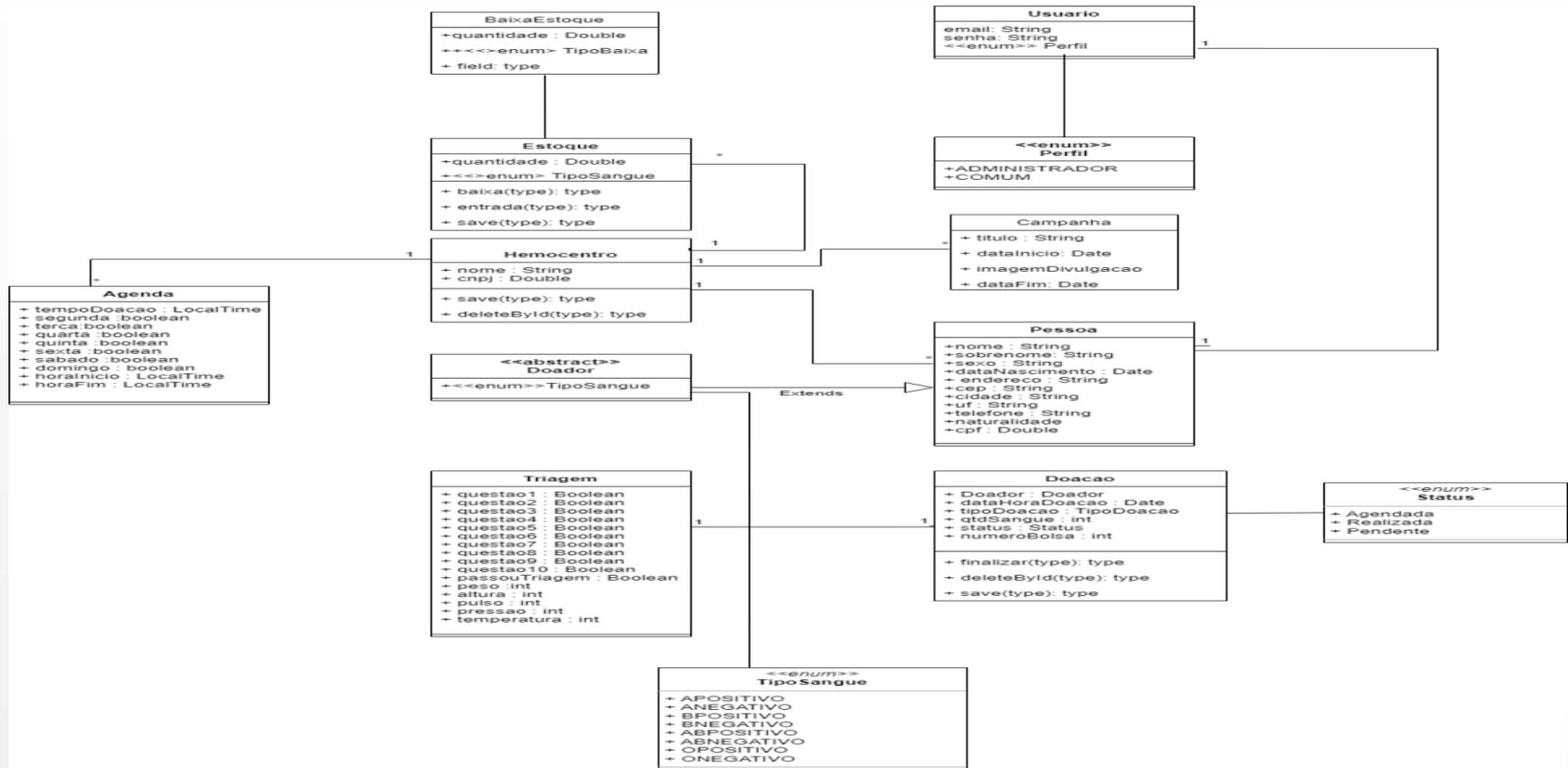
Especificação (1/4)

- Diagrama de Caso de Uso:



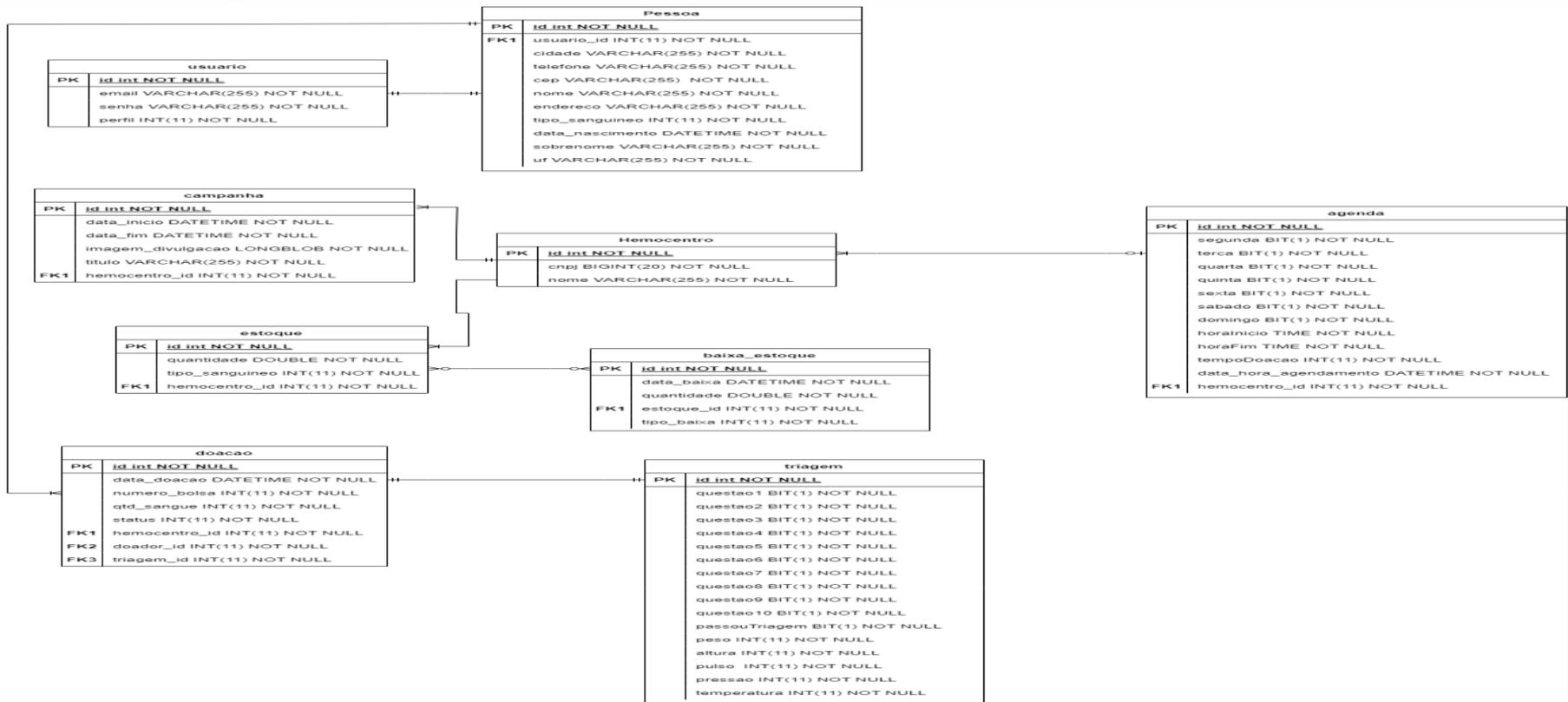
Especificação (2/4)

- Diagrama de Classe:



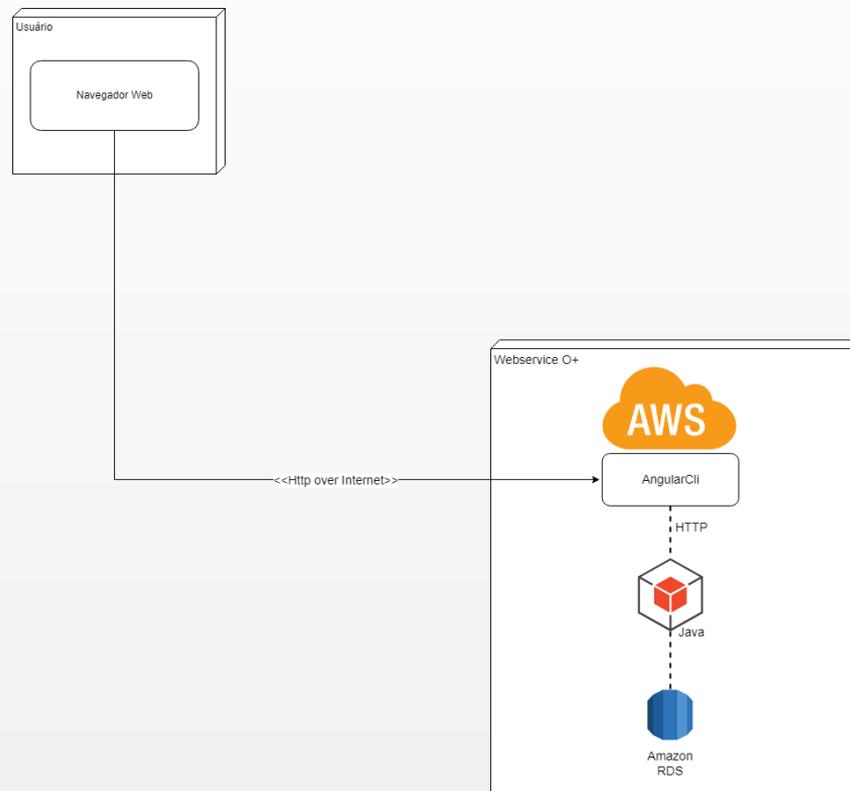
Especificação (3/4)

- Diagrama MER:



Especificação (4/4)

- Diagrama de Implantação:



Implementação (1/4)

- Função de cadastro da doação:

```
67 @Override
68 public ResponseEntity<?> save(@RequestBody DoacaoPojo doacao) {
69     // Procura o doador
70     Optional<DoadorEntidade> doador = doadorRepo.findById(doacao.getDoadorID());
71     // Procura o hemocentro
72     Optional<HemocentroEntidade> hemocentro = institutionRepo.findById(doacao.getHemocentroID());
73     // Procura o triagem
74     Optional<TriagemEntidade> triagem = triagemRepo.findById(doacao.getTriagemID());
75
76     if (doador.isPresent() && hemocentro.isPresent()) {
77         DoacaoEntidade savedDoacao = doacaoRepo.save(
78             new DoacaoEntidade(doacao.getDataHoraDoacao(), doador.get(), hemocentro.get(), triagem.get()));
79         return new ResponseEntity<>(savedDoacao, HttpStatus.OK);
80     }
81     return new ResponseEntity<>(HttpStatus.INTERNAL_SERVER_ERROR);
82 }
```

Implementação (2/4)

- Função de entrada de estoque:

```
123@ @RequestMapping(method = RequestMethod.POST, value = "entrada", consumes = MediaType.APPLICATION_JSON_VALUE)
124 public ResponseEntity<?> entrada(@RequestBody BaixaPojo pojo) {
125     Optional<EstoqueEntidade> estoque = estoqueRepo.findById(pojo.getId());
126     if (estoque.isPresent()) {
127         //salva baixa estoque
128         Date data = new Date();
129         BaixaEstoqueEntidade baixaEstoque = new BaixaEstoqueEntidade(pojo.getQuantidade(),data, estoque.get());
130         baixaEstoque.setTipoBaixa(TipoBaixa.ENTRADA);
131         baixaEstoqueRepo.save(baixaEstoque);
132
133         estoque.get().setQuantidade(estoque.get().getQuantidade() + pojo.getQuantidade());
134         return new ResponseEntity<>(estoqueRepo.save(estoque.get()), HttpStatus.OK);
135     }
136     return new ResponseEntity<>(HttpStatus.NOT_FOUND);
137 }
138
139 }
140 }
...
```

- Função de baixa de estoque:

```
107@ @RequestMapping(method = RequestMethod.POST, value = "baixa", consumes = MediaType.APPLICATION_JSON_VALUE)
108 public ResponseEntity<?> baixa(@RequestBody BaixaPojo pojo) {
109     Optional<EstoqueEntidade> estoque = estoqueRepo.findById(pojo.getId());
110     if (estoque.isPresent()) {
111         //salva baixa estoque
112         Date data = new Date();
113         BaixaEstoqueEntidade baixaEstoque = new BaixaEstoqueEntidade(-pojo.getQuantidade(),data, estoque.get());
114         baixaEstoque.setTipoBaixa(TipoBaixa.SAIDA);
115         baixaEstoqueRepo.save(baixaEstoque);
116
117         estoque.get().setQuantidade(estoque.get().getQuantidade() - pojo.getQuantidade());
118         return new ResponseEntity<>(estoqueRepo.save(estoque.get()), HttpStatus.OK);
119     }
120     return new ResponseEntity<>(HttpStatus.NOT_FOUND);
121 }
...
```

Implementação (3/4)

- Tela de gerenciamento de doação:

No.	Hemocentro	Data	Triagem	Status			
43	Hemocentro 1	29/04/22	42	CANCELADA			
45	Hemocentro 1	02/05/22	44	CANCELADA			
68	Hemocentro 2	07/11/22	67	CANCELADA			
70	Hemocentro 1	07/11/22	69	AGENDADA	Finalizar	Cancelar	Editar Triagem
76	Hemocentro 1	22/11/22	75	AGENDADA	Finalizar	Cancelar	Editar Triagem

Implementação (4/4)

- Tela de gestão de estoque:

O+ Aplicação para gestão de Hemocentros

Pesquisar em:

No.	Tipo Sanguíneo	QTD	Hemocentro			
22	B+	10	Hemocentro 1			Dar Baixa Dar Entrada
24	A+	25	Hemocentro 1			Dar Baixa Dar Entrada
73	O+	0	Hemocentro 2			Dar Baixa Dar Entrada

Análise dos Resultados (1/2)

- Em demonstração e acompanhamento para um usuário comum, o sistema demonstrou significativa facilidade em utilizar o sistema em procedimentos como agendamento de doação e cancelamento;
- Segundo esse usuário, um ponto a ser melhorado no sistema diz respeito a alguma notificação de agendamento, podendo esta ser feita via e-mail ou SMS, o que hoje o sistema não faz.

Análise dos Resultados (2/2)

- Comparativo entre os correlatos:

Correlatos	Souza Junior (2020)	Lira (2020)	Severo e Santos (2018)	O+
Características				
Cadastro de Doadores	Sim	Não	Sim	Sim
Cadastro de Hemocentros	Sim	Não	Não	Sim
Gerenciar Doações de Sangue	Sim	Sim	Sim	Sim
Gerenciar Estoque de sangue	Não	Sim	Não	Sim
Divulgação de campanhas de coleta de sangue	Sim	Não	Sim	Sim
Permitir o usuário realizar agendamento	Sim	Não	Sim	Sim
Exibir informações de estoques do hemocentro.	Não	Sim	Não	Sim
Integração com outros Sistemas	Não	Sim	Não	Não
É responsivo	Sim	Sim	Sim	Sim
Linguagem	Java	Javascript, NodeJS	Java	Java, Angular
Banco de Dados	SQL	SQL	PostgreSQL	mySQL
Plataforma	Android	Web	Web	Web

Conclusões e Sugestões

- Para atingir os objetivos propostos, foi implementada a funcionalidade de agendamento realizada pelo doador e gerenciada pelo funcionário ou administrador do sistema;
- Desenvolvida a funcionalidade de estoque podendo ser utilizada pelos usuários de perfil funcionário e administrador do sistema. Para auxiliar no controle de níveis do estoque foram criados dashboards auxiliando o administrador na tomada decisões.

Extensões

- criar mais gráficos estatísticos, como produtividade por usuário;
- possibilitar a integração com sistema do SUS;
- implementar o login via redes sociais;
- implementar lembrete de agendamento via e-mail e/ou SMS.