

## Relatório Projeto Curricularização – 2025\_2

**Título do Projeto:** Sistema de Gerenciamento de Doações  
Componentes da Equipe de Desenvolvimento

**RA:** 1840482412044 **Nome:** Maria Luisa Martos Montalbini  
**RA:** 1840482412048 **Nome:** Rafaella Felipe de Siqueira Silva

**Professor Orientador:** Pedro Toledo  
**Período de Realização:** de 01/11/2025 até 09/12/2025

**Descrição:** O projeto consiste em um sistema de gestão de doações e pedidos para instituições, com o objetivo de facilitar o controle e a organização de doações recebidas e pedidos feitos pelas instituições.

As principais funcionalidades incluem: cadastro e gerenciamento de doadores e instituições, criação e acompanhamento de pedidos, registro e atualização de doações, consulta de informações detalhadas por doador ou instituição, e controle de status das doações. O sistema garante consistência de dados e integração entre front-end e back-end via API REST, além de aplicar boas práticas de segurança e tratamento de vulnerabilidades.

**Link para o projeto no Github:**

**Back-End:** <https://github.com/rafaellasiqueira/POO-BACK/tree/master>

**Front-End:** <https://github.com/rafaellasiqueira/POO-FRONT/tree/master>

## Telas do projeto

Figura 1 - Tela de login da Instituição



**Login Instituição**

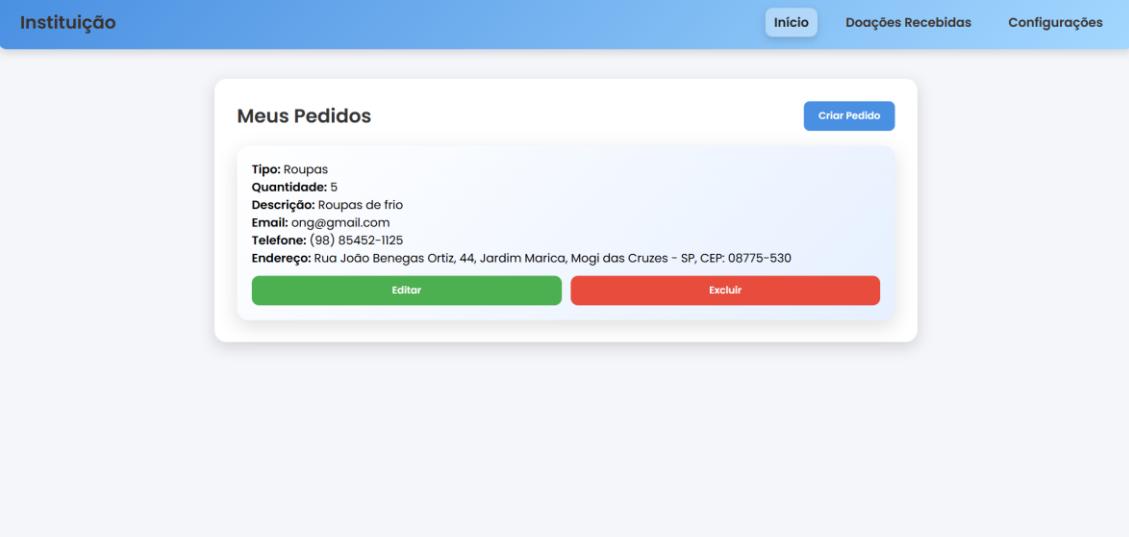
Email

Senha

**Entrar**

Não tem uma conta ainda? [Cadastre-se](#)

Figura 2 - Meus pedidos



**Instituição**

**Meus Pedidos**

**Criar Pedido**

**Dados do Pedido:**

**Tipo:** Roupas  
**Quantidade:** 5  
**Descrição:** Roupas de frio  
**Email:** ong@gmail.com  
**Telefone:** (98) 85452-1125  
**Endereço:** Rua João Benegas Ortiz, 44, Jardim Marica, Mogi das Cruzes - SP, CEP: 08775-530

**Ações:** **Editar** | **Excluir**

**Figura 3 - Doações recebidas**

Início    Doações Recebidas    Configurações

**Doações Recebidas**

**Roupas**

Quantidade: 5  
Data de entrega: 17/12/2025  
Status:  
ENTREGUE  
Observações: V  
Doador: Nicole Rodrigues  
Email do doador: nicole22@email.com  
Telefone do doador: (98) 85452-1125

**Figura 4 - Tela de login do doador**

Seja bem-vindo, doador!

Email  
Digite seu email

Senha  
Digite sua senha

Entrar

Não tem uma conta ainda? [Cadastre-se](#)

**Figura 5 - Pedidos de doações**

**Pedidos de Doações**

**Ong Vida Linda**

**Tipo do item:** Roupas  
**Quantidade:** 5  
**Descrição:** Roupas de frio  
**Email:** ong@gmail.com  
**Telefone:** (98) 85452-1125  
**Endereço:** Rua João Benegas Ortiz, 44 - Jardim Marica, Mogi das Cruzes - SP, CEP: 08775-530

**Ong Vida Nova**

**Tipo do item:** Higiene  
**Quantidade:** 4  
**Descrição:** E  
**Email:** ongvidanova@gmail.com  
**Telefone:** (98) 85452-1234  
**Endereço:** Rua João Benegas Ortiz, 44 - Jardim Marica, Mogi das Cruzes - SP, CEP: 08775-530

**Doar**

**Figura 6 - Histórico de doações**

**Histórico de Doações**

**Roupas**

**Quantidade:** 5  
**Data da entrega:** 17/12/2025  
**Status:** ENTREGUE  
**Observações:** V  
**Email da instituição:** ong@gmail.com  
**Telefone da instituição:** (98) 85452-1125

## Código fonte:

### Back-End

Figura 7 - SecurityConfig

The screenshot shows the IntelliJ IDEA interface with the 'SecurityConfig.java' file open in the central editor window. The code defines a public class 'DoacaoEntity' with various fields and annotations. It includes a status enum with values PENDENTE, EM\_ANDAMENTO, ENTREGUE, and CANCELADA. The code is part of a larger project structure visible in the left sidebar.

```
public class DoacaoEntity {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String tipoDoacao;
    private Integer quantidade;
    private LocalDate data;
    private String observacao;

    @Enumerated(EnumType.STRING)
    private StatusDoacao status;

    @ManyToOne
    @JoinColumn(name = "doador_id")
    @JsonIdentityReference(alwaysAsId = false)
    private DoadorEntity doador;

    @ManyToMany
    @JoinColumn(name = "pedido_id")
    @JsonIdentityReference(alwaysAsId = false)
    private PedidoEntity pedido;
}

public enum StatusDoacao {
    PENDENTE,
    EM_ANDAMENTO,
    ENTREGUE,
    CANCELADA
}
```

Figura 8 - DoacaoController

The screenshot shows the IntelliJ IDEA interface with the 'DoacaoController.java' file open in the central editor window. The code defines a 'DoacaoController' class with various methods using annotations like @RestController, @RequestMapping, and @DeleteMapping. It interacts with a 'DoacaoService' and a 'DoacaoRepository'. The code is part of a larger project structure visible in the left sidebar.

```
@RestController("doacao")
@RequestMapping("/doacao")
public class DoacaoController {

    private final DoacaoService doacaoService; 6 usages

    public DoacaoController(DoacaoService doacaoService) { this.doacaoService = doacaoService; }

    @PostMapping(value = "/add", consumes = MediaType.APPLICATION_JSON_VALUE) no usages &rafaellasiqueira
    public ResponseEntity<DoacaoEntity> registrarDoacao(@RequestBody DoacaoEntity doacao) {
        DoacaoEntity salva = doacaoService.registrarDoacao(doacao);
        return ResponseEntity.ok(salva);
    }

    @GetMapping("/list/{id}") no usages &rafaellasiqueira
    public ResponseEntity<List<DoacaoEntity>> listarPorDoador(@PathVariable Long id) {
        List<DoacaoEntity> doacoes = doacaoService.listarDoacoesPorIdDoador(id);
        return ResponseEntity.ok(doacoes);
    }

    @DeleteMapping("/delete/{id}") no usages &rafaellasiqueira
    public ResponseEntity<Void> excluir(@PathVariable Long id) {
        boolean excluido = doacaoService.excluirDoacao(id);
        return excluido ? ResponseEntity.noContent().build() : ResponseEntity.notFound().build();
    }

    @GetMapping("/listInstituicao/{id}") no usages &rafaellasiqueira
    public ResponseEntity<List<DoacaoEntity>> listarPorInstituicao(@PathVariable Long id) {
        List<DoacaoEntity> doacoes = doacaoService.listarDoacoesPorInstituicao(id);
        return ResponseEntity.ok(doacoes);
    }

    @PutMapping("/updateStatus/{id}") no usages &rafaellasiqueira
    public ResponseEntity<DoacaoEntity> atualizarStatus(@PathVariable Long id, @RequestBody StatusRequest statusRequest) {
}
```

**Figura 9 - DoacaoEntity**

The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays the `DoacaoEntity.java` file, which is part of the `com.gerenciamendo_daoes.gerenciamendo_daoes` package. The code defines a simple entity with attributes `id`, `data`, `descricao`, `valor`, and `status`. The `status` attribute is annotated with `@Enumerated(EnumType.STRING)`.

```

1 package com.gerenciamendo_daoes.gerenciamendo_daoes;
2
3 import javax.persistence.*;
4 import java.util.Date;
5
6 @Entity
7 @Table(name = "doacao")
8 public class DoacaoEntity {
9
10     @Id
11     @GeneratedValue(strategy = GenerationType.IDENTITY)
12     private Long id;
13
14     @Column(name = "data")
15     private Date data;
16
17     @Column(name = "descricao")
18     private String descricao;
19
20     @Column(name = "valor")
21     private Double valor;
22
23     @Column(name = "status")
24     @Enumerated(EnumType.STRING)
25     private Status status;
26
27 }
  
```

**Figura 10 - DoacaoRepository**

The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays the `DoacaoRepository.java` file, which is part of the `com.gerenciamendo_daoes.gerenciamendo_daoes` package. This is a JPA repository interface with methods for finding doações by ID, by doador, and by instituição.

```

1 package com.gerenciamendo_daoes.gerenciamendo_daoes;
2
3 import com.gerenciamendo_daoes.gerenciamendo_daoes.Entity.DoacaoEntity;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.data.jpa.repository.Query;
6 import org.springframework.data.repository.query.Param;
7 import java.util.List;
8
9 public interface DoacaoRepository extends JpaRepository<DoacaoEntity, Long> {
10
11     @Query("SELECT d FROM DoacaoEntity d " + "JOIN FETCH d.doador " +
12           "JOIN FETCH d.pedido p " +
13           "JOIN FETCH p.instituicao i " +
14           "WHERE i.id = :instituicaoId")
15     List<DoacaoEntity> findByInstituicaoId(@Param("instituicaoId") Long instituicaoId);
16
17     @Query("SELECT d FROM DoacaoEntity d " + "JOIN FETCH d.doador " +
18           "LEFT JOIN FETCH d.pedido p " +
19           "WHERE d.id = :id")
20     DoacaoEntity findByIdWithDoadorAndPedido(@Param("id") Long id);
21
22     @Query("SELECT d FROM DoacaoEntity d " + "JOIN FETCH d.doador " +
23           "LEFT JOIN FETCH d.pedido p " +
24           "LEFT JOIN FETCH p.instituicao " +
25           "WHERE d.doador.id = :idDoador")
26     List<DoacaoEntity> findByDoadorId(@Param("idDoador") Long idDoador);
27
28 }
  
```

**Figura 11 - InstituicaoService**

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Tree:** The left sidebar displays the project structure under "gerenciamento\_dacoes". It includes main packages like "gerenciamento\_dacoes", "main", "java", "Config", "Controller", "Entity", "Repository", "Service", and "resources".
- Code Editor:** The right pane shows the source code for the "InstituicaoService.java" file. The code defines a service layer for managing institutions, utilizing a repository and entity classes.
- Toolbars and Status Bar:** The top bar includes tabs for "gerenciamento\_dacoes", "master", and "com.gerenciadordedoacoes.gerenciamento\_dacoes". The status bar at the bottom shows "Start Free Trial" and other system information.

```
1 package com.gerenciadordedoacoes.gerenciamento_dacoes.Service;
2
3 import com.gerenciadordedoacoes.gerenciamento_dacoes.Entity.InstituicaoEntity;
4 import com.gerenciadordedoacoes.gerenciamento_dacoes.Repository.InstituicaoRepository;
5 import org.springframework.stereotype.Service;
6 import java.util.Optional;
7
8 @Service
9 public class InstituicaoService {
10
11     private final InstituicaoRepository instituicaoRepository; 5 usages
12
13     public InstituicaoService(InstituicaoRepository instituicaoRepository) {  no usages &rafaelisaqueira
14         this.instituicaoRepository = instituicaoRepository;
15     }
16
17     public InstituicaoEntity cadastrar(InstituicaoEntity instituicao) { 1 usage &rafaelisaqueira
18         return instituicaoRepository.save(instituicao);
19     }
20
21     public Optional<InstituicaoEntity> login(String email, String senha) { 1 usage &rafaelisaqueira
22         return instituicaoRepository.findByEmail(email)
23             .filter(instituicaoEntity i -> i.getSenha().equals(senha));
24     }
25
26     public InstituicaoEntity atualizar(InstituicaoEntity instituicao) { 1 usage &rafaelisaqueira
27         return instituicaoRepository.save(instituicao);
28     }
29
30     public Optional<InstituicaoEntity> buscarPorId(Long id) { 1 usage &rafaelisaqueira
31         return instituicaoRepository.findById(id);
32     }
33 }
```

## Front-End

**Figura 12 - Doador/Script\_login**

The screenshot shows the WebStorm IDE interface. The left sidebar displays the project structure:

- Project
- front\_end\_gerenciamento\_doadores
  - idea
  - scripts
    - Doador
      - script\_adddoador.js
      - script\_configuracoes.js
      - script\_home.js
      - script\_listhistorico.js
      - script\_Login.js
      - script\_updateadoidor.js
    - Instituicao
      - script\_adinstituicoes.js
      - script\_configuracoes.js
      - script\_home.js
      - script\_listhistorico.js
      - script\_login.js
      - script\_updateinstituicao.js
    - styles
    - Doador
      - AtualizarConta
        - updateadoidor.css
        - updateadoidor.html
      - Cadastro
        - adddoador.css
        - adddoador.html
      - Configuracoes
      - Historico
      - Home
      - Login
        - index.html
        - style.css
      - Instituicao
        - AtualizarConta
        - Cadastro
        - Configuracoes
        - Historico
        - Home
        - Login

**Figura 13 - Doacao/Script\_listhistorico**

```
front_end_gerenciamento_daoes master
Project
  front_end_gerenciamento_daoes
    -> idea
    -> scripts
      -> Doador
        -> script_Ladddor.js
        -> script_Configuracoes.js
        -> script_Home.js
        -> script_lhististorico.js
        -> script_Login.js
        -> script_Updatedoador.js
      -> Instituicao
        -> script_Administrativas.js
        -> script_Configuracoes.js
        -> script_Home.js
        -> script_lhististorico.js
        -> script_Login.js
        -> script_UpdatedoInstituicoes.js
    -> styles
      -> Doador
        -> AtualizarConta
          -> updatedoador.css
          -> updatedoador.html
        -> Cadastro
          -> adddor.css
          -> adddor.html
        -> Configuracoes
        -> Historico
        -> Home
        -> Login
          -> index.html
          -> style.css
      -> Instituicao
        -> AtualizarConta
        -> Cadastro
        -> Configuracoes
        -> Historico
        -> Home
        -> Login
    -> Current File
      -> Try WebStorm
      -> Learn more
      -> Dismiss
      -> Start Free Trial
      -> 96 %
```

```
Doador/script_lhististorico.js x script_adddor.js <> configuracoes.html <> Instituicao/script_configuracoes.js <> Instituicao/script_login.js <> script_Updatedoador.js <> Doador/script_Login.js
```

```
Try WebStorm Learn more Dismiss
```

```
34         container.appendChild(card);
35     }
36
37
38     function carregarDoacoes() {
39         if (!idDoador) {
40             container.innerHTML = '<p>Erro: usuário não está logado.</p>';
41             return;
42         }
43
44         fetch(`http://localhost:8080/doacao/list/${idDoador}`)
45             .then(response => {
46                 if (!response.ok) throw new Error('Erro ao buscar doações');
47                 return response.json();
48             })
49             .then(doacoes => {
50                 container.innerHTML = '';
51                 if (!doacoes || doacoes.length === 0) {
52                     container.innerHTML = '<p>Nenhuma doação encontrada.</p>';
53                     return;
54                 }
55
56                 doacoes.sort((a, b) => (a.tipoDoacao || '').localeCompare(b.tipoDoacao || ''));
57                 doacoes.forEach(criarCardDoacao);
58             })
59             .catch(error => {
60                 container.innerHTML = '<p>Erro ao carregar doações: ${error.message}</p>';
61                 console.error(error);
62             });
63
64
65     function excluirDoacao(idDoacao) {
66         if (!confirm("Deseja realmente excluir esta doação?")) return;
```

**Figura 14 -Instituicao/Script\_listhistorico**

The screenshot shows the WebStorm IDE interface with the following details:

- Project Tree:** The project structure is visible on the left, showing modules like `front_end_gerenciamento_doador`, `Instituicao`, and `Login`, along with their sub-modules and files.
- File List:** A list of files is shown at the top right, including `figuracoes.html`, `Instituicao/script_configuracoes.js`, `Instituicao/script_login.js`, `script_updateador.js`, `Doador/script_Login.js`, `Instituicao/script_home.js`, and `Instituicao/script_listhistorico.js`.
- Code Editor:** The main area displays the content of the `script_listhistorico.js` file. The code is written in JavaScript and includes several imports from other files like `script_updateador.js` and `script_login.js`. It defines a function `carregarDoacoes()` which makes a fetch request to `http://localhost:8080/doacao/listInstituicao/${instituicaoId}` to retrieve a list of donations. The fetched data is then processed to update the DOM.
- Toolbars and Status:** The top bar includes standard browser-like navigation buttons (Back, Forward, Home) and a search bar. On the right, there are buttons for "Start Free Trial", "Try WebStorm", "Learn more", and "Dismiss". A status bar at the bottom right indicates "84 84".