



# MESA COMPARTILHADA

3 ° SEMESTRE



# PROJETO INTEGRADOR

Professor orientador  
Bruno Zolotareff

Disciplina Chave  
Gestão Ágil de Projetos



The background features several food-related illustrations. In the top left, a bowl of soup with dumplings and green vegetables. In the top right, a yellow circular shape with green wavy lines. In the middle left, a cluster of mushrooms. In the middle right, two broccoli florets. In the bottom left, a green circle and a yellow shape with green wavy lines. In the bottom right, a bowl of food with a fried egg, green onions, and other toppings.

# INTEGRANTES

Bruna Brito Santiago  
Bruno Henrique Gusmão  
Guilherme Cardoso Pessoa  
Nicole Aparecida Teixeira  
Willian Otavio de Oliveira


# ODS's






# TÓPICOS

- 1 Introdução
- 2 Problemática
- 3 SMART
- 4 EAP
- 5 Canva
- 6 BPMN
- 7 Caso de uso
- 8 Tecnologias Utilizadas
- 9 Código Fonte
- 10 Sprints
- 11 Vídeo do Projeto




# INTRODUÇÃO

## A FOME




No Brasil a fome persiste , com milhões de pessoas em insegurança alimentar, especialmente após a pandemia. Desigualdade social é um fator contribuinte significativo.

## O DESPERDÍCIO



Apesar do Brasil ser um grande produtor e exportador de alimentos, o desperdício é um problema importante devido a más condições de transporte, clima e armazenamento inadequado.

## DESAFIOS GLOBAIS



Apesar do aumento da riqueza global, mais de 1 bilhão de pessoas vivem em extrema pobreza e enfrentam privações em áreas essenciais como saúde, educação e padrões de vida.


# PROBLEMÁTICA

Como viabilizar o acesso de pessoas vulneráveis a alimentos produzidos por instituições fornecedoras de alimento que atualmente são descartados em bom estado?





# SMART



**SPECIFIC**  
**MEASURABLE**  
**ACHIEVABLE**  
**RELEVANT**  
**TIME-BOUND**

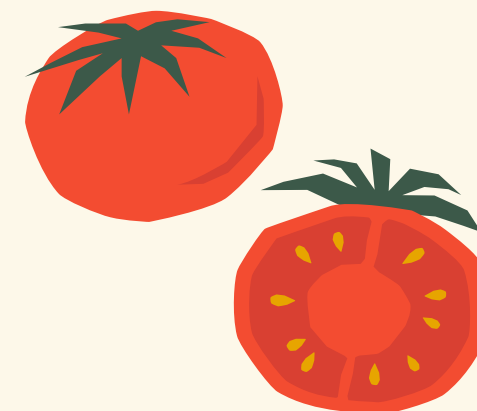
○ nosso objetivo é saciar a fome das pessoas em situações precárias, utilizando os alimentos em bom estado que seriam descartados em conformidade com a lei

○ nosso projeto se mostraria qualificado quando 10% da fome em diadema fosse erradicada, demonstrando progresso na nossa finalidade

○ nosso objetivo abrange toda a cidade de Diadema, com o propósito de viabilizar a conexão de empresas e ong's, e para alcançar isso precisamos de visibilidade

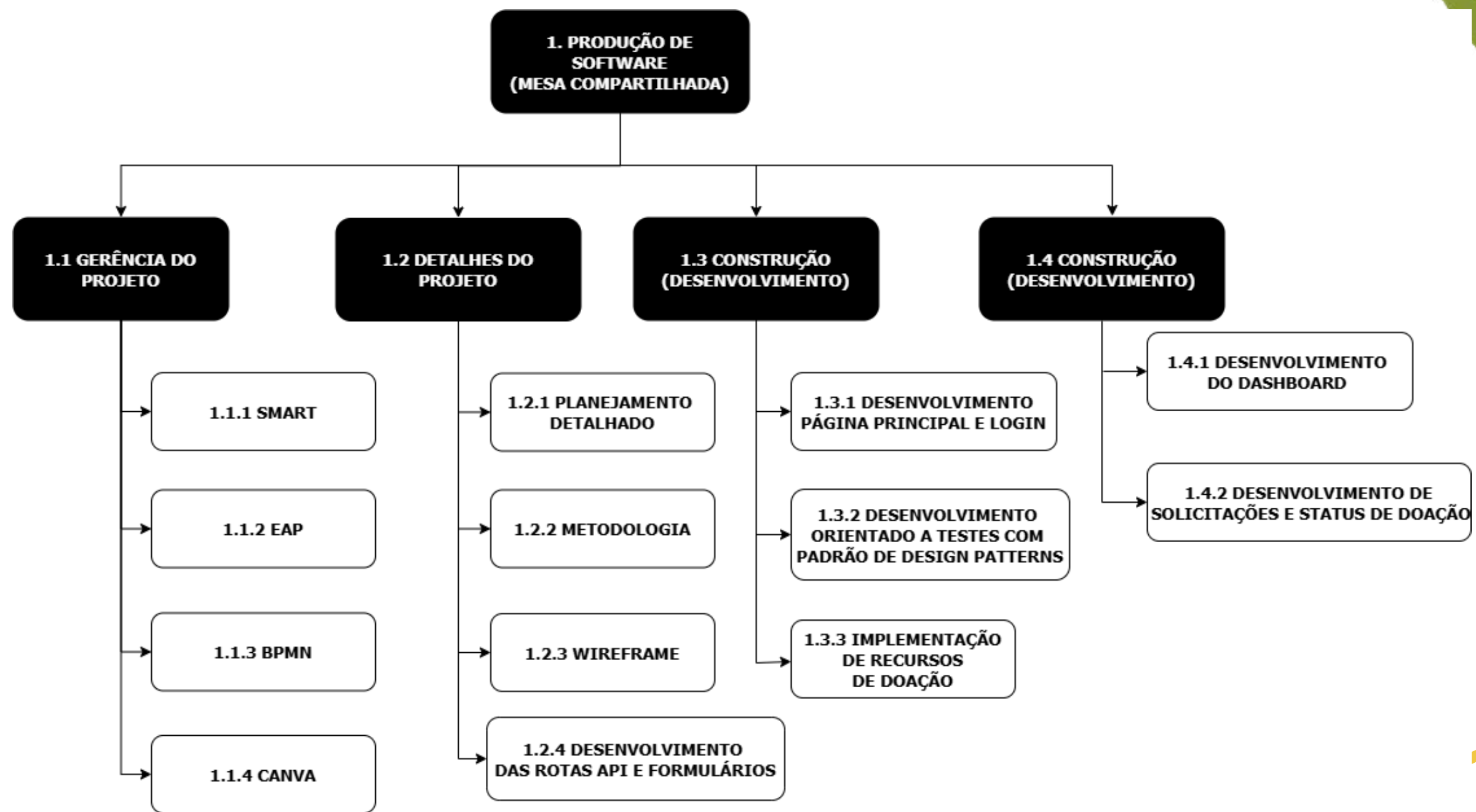
○ nosso projeto contruibui significativamente para a população Diademense e instituições sociais responsáveis pois o auxilio aos carentes contribui para uma sociedade solidaria e conciente

○ nosso objetivo é atingir juntamente com a Agenda da ONU os resultados de um mundo melhor em 2030 aplicando em ações eficazes as ODS

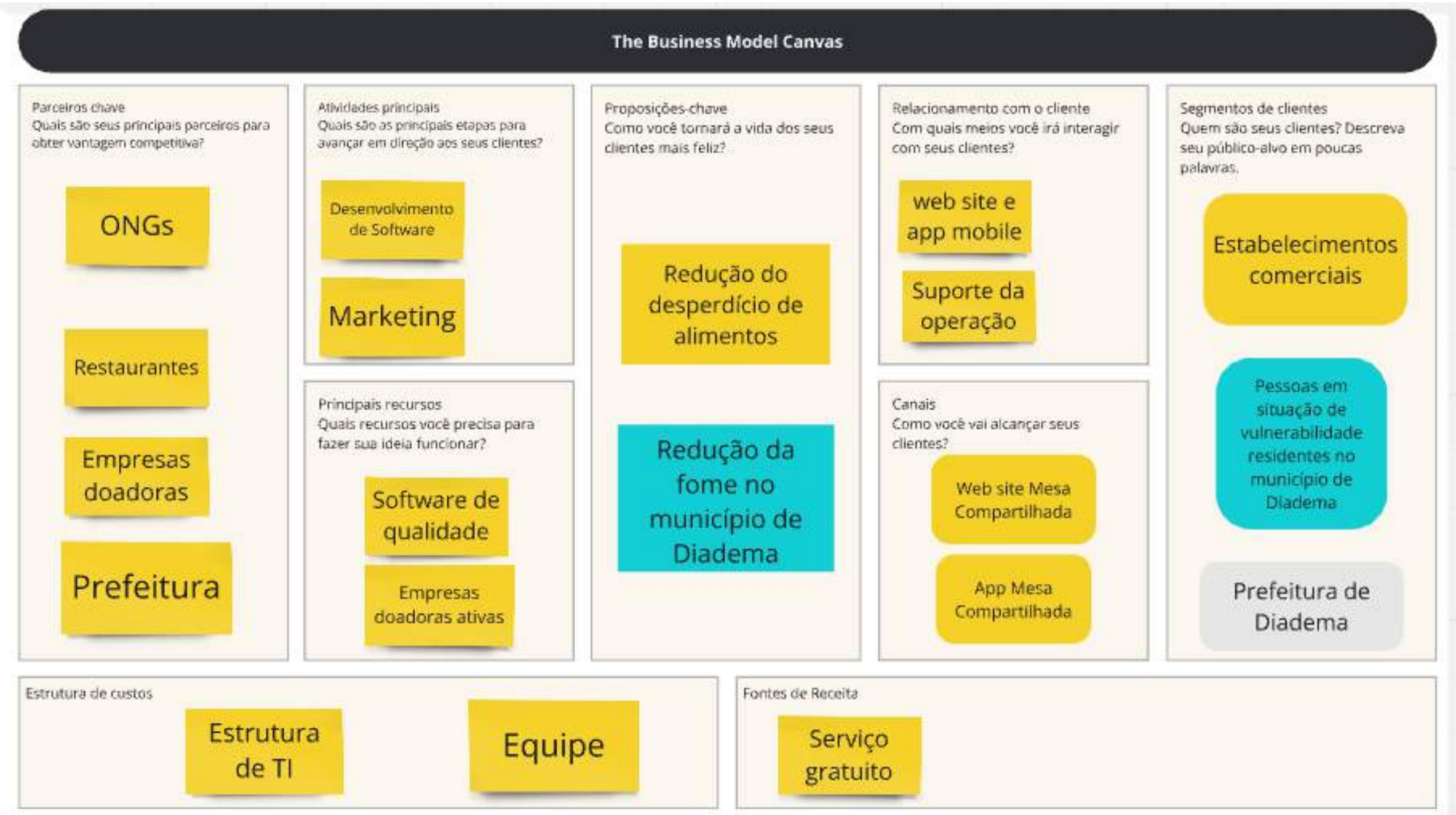




# EAP

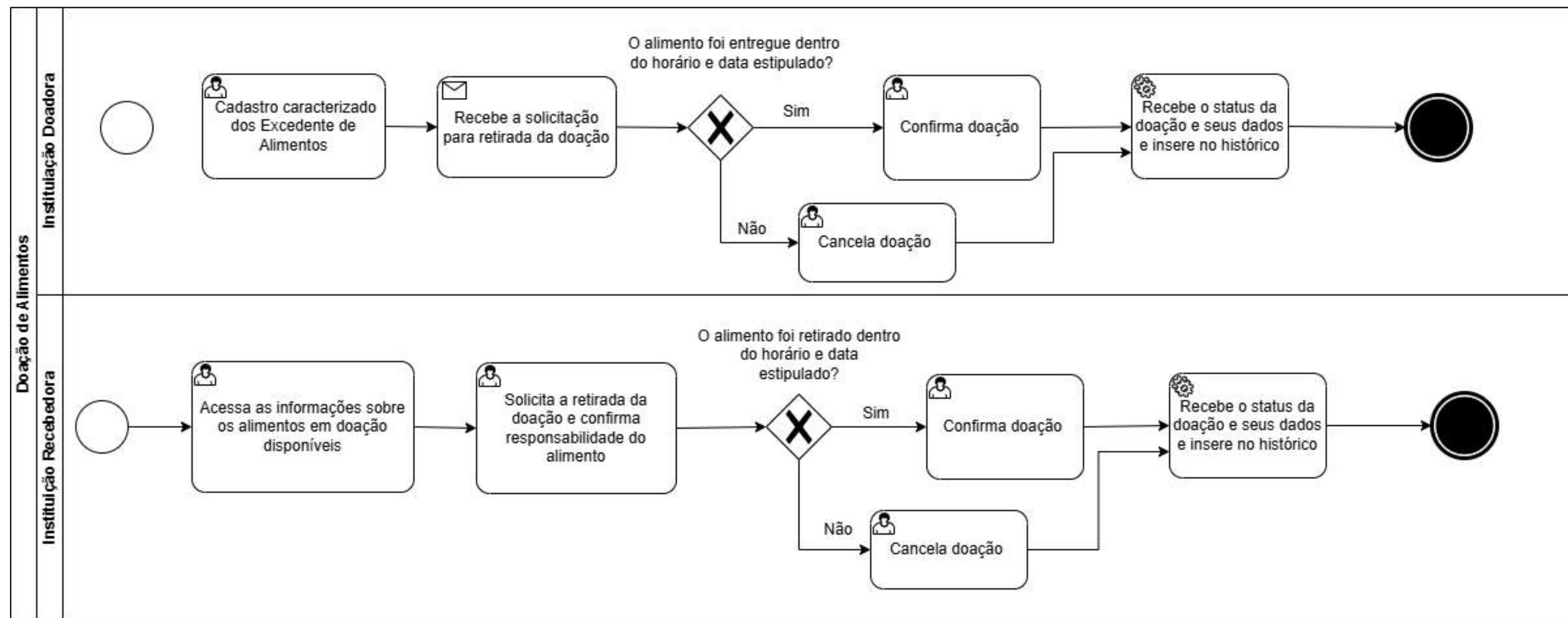


# CANVA

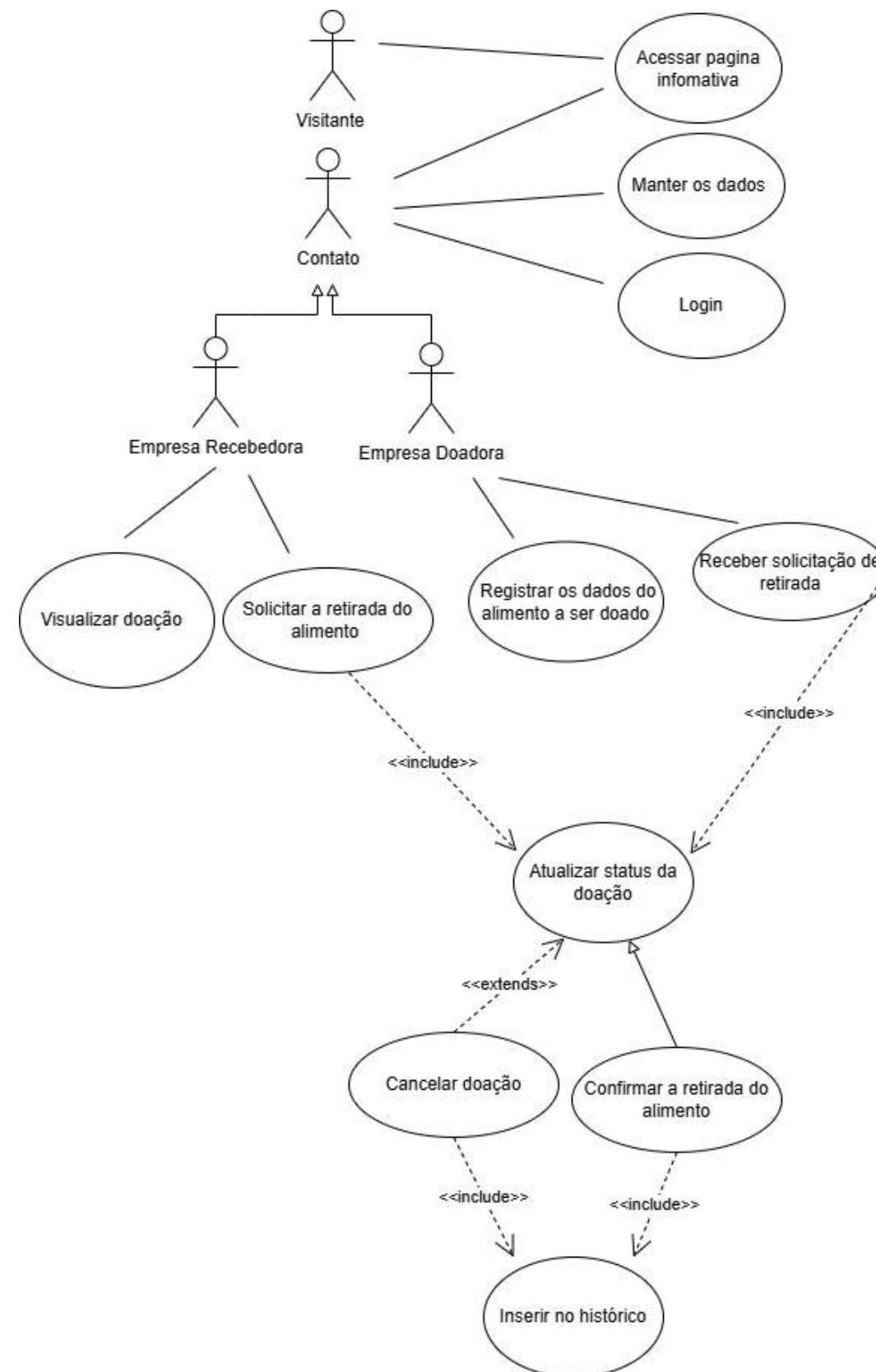




# BPMN



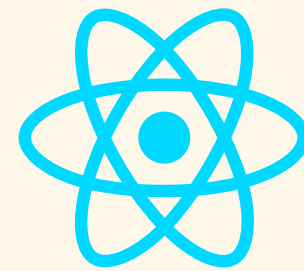
# CASO DE USO





# TECNOLOGIAS UTILIZADAS

FRONT-END



BACK-END



BANCO DE DADOS





# CÓDIGO FONTE

**FRONT-END**





# FRONTEND

```
1 import { getEmpresaById } from "../empresa"
2
3 export async function getDoacoes() {
4   try {
5     > let result = await fetch("http://localhost:8080/api/c/doacao", (...
6   } catch (error) {
7     console.log(error)
8     return false
9   }
10 }
11
12 > export async function getDoacaoById(id) { ...
13 }
14
15 > export async function getDoacoesByEmpresaDoadoraId(id) { ...
16 }
17
18 > export async function getDoacoesByEmpresaRecebadoraId(id) { ...
19 }
20
21 > export async function getDoacoesByStatus(status) { ...
22 }
23
24 > export async function addDoacao(doacao) { ...
25 }
26
27 > export async function updateStatusDoacao(status, doacaoId, empresaRecebadoraId) { ...
28 }
29
30 > export async function getDoacoesEmpresa(id) { ...
31 }
32
33 111 |
```

```
export function ListaDoacoes({ getDoacoes, params = null }) {
  const { doacoesAlteradas, setDoacoesAlteradas } = useContext(DashboardContext)

  const [doacoes, setDoacoes] = useState(null)

  useEffect(() => {
    getListaDeDoacoes()
  }, [doacoesAlteradas])

  const getListaDeDoacoes = async () => {
    let result = !params ? await getDoacoes() : await getDoacoes(params)
    setDoacoes(result)
  }

  return(
    <>
      {
        doacoes !== null && doacoes.length > 0
        ? <div className="flex gap-2">
          { doacoes.map((doacao) =>
            <CardDoacao key={doacao.id} doacao={ doacao } />
          ) }
        : </div>
      }
    </div>
  )
}
```

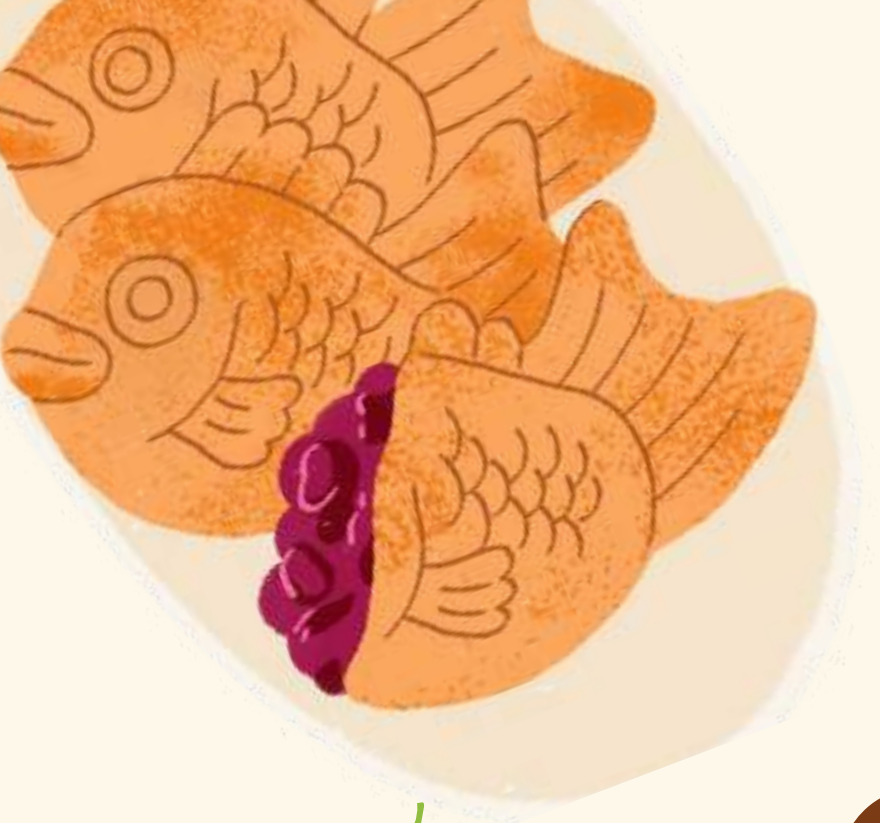
# TAILWIND CSS

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: ["./src/**/*.{js,jsx,ts,tsx}"],
  theme: {
    container: {
      center: true,
    },
    extend: {
      colors: {
        "azul": "#62C0C0",
        "l-Abobora": "#FF6600",
        "l-pessego": "#FEEAC2",
        "branco": "#FFFAF0",
        "azul-escuro": "#003B5D",
      },
      backgroundImage: {
        fundo: "url('/src/assets/onda.svg')",
      },
    },
    corePlugins: {
      preflight: false,
    },
  },
  plugins: [],
};
```

```
export default function Home() {
  return (
    <>
      <div className="bg-white h-full">
        <Header />
        <div className="pt-5 lg:p-0">
          <div className="container text-l-Abobora flex flex-col lg:flex-row justify-center lg:
            <div className="self-center lg:max-w-lg">
              <h1 className="mb-4 text-4xl font-extrabold leading-none tracking-tight md:text-5xl
                Reduzir o desperdício e{" "}
                <span className="underline underline-offset-3 decoration-8 decoration-azul">
                  compartilhar
                </span>{" "}
                com quem mais precisa
              {" "}
```

```
.btn-primary{
  @apply bg-l-Abobora text-white p-1 px-2 ml-auto rounded-lg hover:cursor-p
}
.btn-red{
  @apply bg-red-700 text-white p-1 px-2 ml-auto rounded-lg hover:cursor-poi
}
.input-default{
  @apply rounded-lg p-2 bg-neutral-100
}
.input-error{
  @apply rounded-lg p-2 outline outline-red-500
}
.label-default{
  @apply absolute top-2.5 left-2 text-neutral-800 text-xs -translate-y-7 dura
  peer-placeholder-shown:-translate-y-0 peer-focus:-translate-y-7 peer-placeh
  peer-focus:text-opacity-100 select-none hover:cursor-text
}
```





# CÓDIGO FONTE

**BACK-END**





# SERVICES E CONTROLLERS

```
public class EmpresaController {

    private final EmpresaService empresaService; 8 usages

    > public EmpresaController(EmpresaService empresaService) { this.empresaService = empresaService; }

    @GetMapping(path="/empresa", produces = MediaType.APPLICATION_JSON_VALUE) no usages 1 willotavio
    > public List<EmpresaResponseDto> getAllEmpresas(HttpServletResponse response) {...}

    @GetMapping(path="/empresa/{empresaId}") no usages 1 willotavio
    > public ResponseEntity<Object> getEmpresaById(@PathVariable(value="empresaId") String empresaId) {...}

    @GetMapping(path="/empresa/email/{email}") no usages 1 willotavio
    > public ResponseEntity<Object> getEmpresaByEmail(@PathVariable(value="email") String email) {...}

    @PostMapping(path="/empresa") no usages 1 willotavio
    > public ResponseEntity<Object> addEmpresa(@RequestBody @Valid EmpresaRequestDto empresaRequestDto) {...}

    @PutMapping(path="/empresa/{empresaId}") no usages 1 willotavio
    > public ResponseEntity<Object> updateEmpresa(@PathVariable(value="empresaId") String empresaId,
        @RequestBody @Valid EmpresaRequestDto empresaRequestDto) {...}

    @DeleteMapping(path="/empresa/{empresaId}") no usages 1 willotavio
    > public ResponseEntity<Object> deleteEmpresa(@PathVariable(value="empresaId") String empresaId) {...}

    @PostMapping(path="/empresa/login") no usages 1 willotavio
    > public ResponseEntity<Object> login(@RequestBody @Valid EmpresaLoginRequestDto empresaLoginRequestDto)
```

```
public class EmpresaService {

    private final EmpresaRepository empresaRepository; 10 usages
    private final EnderecoRepository enderecoRepository; 3 usages
    private final PasswordEncoder passwordEncoder; 4 usages

    @Autowired no usages 1 willotavio
    public EmpresaService(EmpresaRepository empresaRepository, EnderecoRepository enderecoRepository) {...}

    public List<EmpresaResponseDto> getAllEmpresas() {...}

    public Optional<EmpresaResponseDto> getEmpresaById(String empresaId) {...}

    public Optional<EmpresaResponseDto> getEmpresaByEmail(String email) {...}

    public Optional<EmpresaResponseDto> addEmpresa(EmpresaRequestDto empresaRequestDto) {...}

    public Optional<EmpresaResponseDto> updateEmpresa(String empresaId, EmpresaRequestDto empresaRequestDto) {...}

    public Optional<EmpresaResponseDto> deleteEmpresa(String empresaId) {...}

    public Optional<EmpresaResponseDto> login(EmpresaLoginRequestDto empresaLoginRequestDto) {...}

}
```



# DESIGN PATTERN STATE

```
public abstract class StateDoacao { 5 usages 4 inheritors 1 willotavio *  
  
    protected Doacao doacao;  
  
    public StateDoacao(Doacao doacao) { this.doacao = doacao; }  
  
    public abstract String getStateName(); 6 usages 4 implementations 1 willotavio  
  
    public abstract void solicitar(Empresa empresaRecebedora) throws DoacaoStatusOperationNotSupportedException,  
        DoacaoStatusIllegalArgumentException;  
  
    public abstract void concluir(Empresa empresaSolicitante) throws DoacaoStatusOperationNotSupportedException,  
        DoacaoStatusIllegalArgumentException;  
  
    public abstract void cancelarSolicitacao(Empresa empresaRecebedora) throws DoacaoStatusOperationNotSupportedException,  
        DoacaoStatusIllegalArgumentException;  
  
    public abstract void cancelar() throws DoacaoStatusOperationNotSupportedException; 1 usage 4 implementations
```

```
public class Disponivel extends StateDoacao { 2 usages 1 willotavio *  
  
    public Disponivel(Doacao doacao) { super(doacao); }  
  
    @Override 0 usages 1 willotavio  
    public String getStateName() { return "DISPONIVEL"; }  
  
    @Override 1 usage 1 willotavio *  
    public void solicitar(Empresa empresaRecebedora) throws DoacaoStatusOperationNotSupportedException,  
        DoacaoStatusIllegalArgumentException {  
        if(empresaRecebedora == null) {...}  
        // tornar doacao solicitada em andamento  
        super.doacao.setEmpresaRecebedora(empresaRecebedora);  
        List<Doacao> doacoesAtualizadas = empresaRecebedora.getDoacoes();  
        doacoesAtualizadas.add(doacao);  
        empresaRecebedora.setDoacoes(doacoesAtualizadas);  
        doacao.setStatus(new Andamento(doacao).getStateName());  
    }  
  
    @Override 1 usage 1 willotavio  
    public void concluir(Empresa empresaSolicitante) throws DoacaoStatusOperationNotSupportedException {  
        throw new DoacaoStatusOperationNotSupportedException("Uma doação disponível não pode ser concluída");  
    }  
  
    @Override 1 usage 1 willotavio  
    public void cancelarSolicitacao(Empresa empresaRecebedora) throws DoacaoStatusOperationNotSupportedException {...}  
  
    @Override 1 usage 1 willotavio  
    public void cancelar() throws DoacaoStatusOperationNotSupportedException {...}
```

```
public class Andamento extends StateDoacao { 2 usages 1 willotavio  
  
    @Override 1 usage 1 willotavio  
    public void solicitar(Empresa empresaRecebedora) throws DoacaoStatusOperationNotSupportedException {...}  
  
    @Override 1 usage 1 willotavio  
    public void concluir(Empresa empresaSolicitante) throws DoacaoStatusOperationNotSupportedException, DoacaoStat  
        if(empresaSolicitante == null  
            || !empresaSolicitante.getId().equals(super.doacao.getEmpresaDoadora().getId())  
            && (super.doacao.getEmpresaRecebedora() != null  
                && !empresaSolicitante.getId().equals(super.doacao.getEmpresaRecebedora().getId())) {...}  
        if(empresaSolicitante.getTipo() == TipoEmpresa.DOADORA.getCodigo()) {...}  
        else {...}  
        if(super.doacao.isEmpresaDoadoraConcluida() && super.doacao.isEmpresaRecebedoraConcluida()) {...}  
    }  
}
```

# MAPEAMENTO DAS ROTAS

## endereco-controller

GET /api/mc/endereco/{enderecoId}

PUT /api/mc/endereco/{enderecoId}

DELETE /api/mc/endereco/{enderecoId}

GET /api/mc/endereco

POST /api/mc/endereco

## empresa-controller

GET /api/mc/empresa/{empresaId}

PUT /api/mc/empresa/{empresaId}

DELETE /api/mc/empresa/{empresaId}

GET /api/mc/empresa

POST /api/mc/empresa

POST /api/mc/empresa/login

GET /api/mc/empresa/email/{email}

## doacao-controller

GET /api/mc/doacao/{doacaoId}

PUT /api/mc/doacao/{doacaoId}

DELETE /api/mc/doacao/{doacaoId}

PUT /api/mc/doacao/status/{doacaoId}

GET /api/mc/doacao

POST /api/mc/doacao

GET /api/mc/doacao/status/{status}

GET /api/mc/doacao/empresa-recebedora/{empresaRecebedoraId}

GET /api/mc/doacao/empresa-doadora/{empresaDoadoraId}





# CÓDIGO FONTE

**BANCO DE DADOS**



# DOCUMENTOS MONGODB

```
apimc> db.doacao.find()
[
  {
    _id: ObjectId("670a8b708dc16e25f25a6e6d"),
    nome: 'Torta de Frango',
    descricao: 'Torta salgada com massa macia e recheio de frango desfiado',
    status: 'ANDAMENTO',
    observacao: 'Manter refrigerado e consumir em até 3 dias',
    dataFabricacao: ISODate("2024-10-12T03:00:00.000Z"),
    dataValidade: ISODate("2024-10-15T03:00:00.000Z"),
    dataCriada: ISODate("2024-10-12T03:00:00.000Z"),
    dataMaxRetirada: ISODate("2024-10-15T03:00:00.000Z"),
    horarioMin: ISODate("2024-10-12T10:20:00.000Z"),
    horarioMax: ISODate("2024-10-13T00:50:00.000Z"),
    tipoAlimento: 1,
    tipoArmazenamento: 3,
    empresaDoadora: DBRef("empresa", ObjectId("6709ddfdf4eaf5460e17cb6c")),
    empresaRecebedora: DBRef("empresa", ObjectId("670ab3beb016f913691f41d9")),
    empresaDoadoraConcluida: false,
    empresaRecebedoraConcluida: false,
    _class: 'com.pi.mesacompartilhada.models.Doacao'
  },
]
```

```
apimc> db.empresa.find()
[
  {
    _id: ObjectId("6709ddfdf4eaf5460e17cb6c"),
    cnpj: '22.527.789/0001-76',
    tipo: 1,
    categoria: 1,
    nome: 'Lancheria Prato Cheio',
    email: 'lancheriapratocheio@gmail.com',
    senha: '$2a$10$lhq8GZgE6lkstPmKLkiwTegDlx06.9bzStIWeOJAUNnwKZs9cWAFW',
    status: 1,
    endereco: DBRef("endereco", ObjectId("6709ddfdf4eaf5460e17cb6b")),
    doacoes: [
      DBRef("doacao", ObjectId("670a8b708dc16e25f25a6e6d")),
      DBRef("doacao", ObjectId("670a8c208dc16e25f25a6e6e"))
    ],
    _class: 'com.pi.mesacompartilhada.models.Empresa'
  },
]
```

```
apimc> db.endereco.find()
[
  {
    _id: ObjectId("6709ddfdf4eaf5460e17cb6b"),
    cep: '09960010',
    numero: '11',
    logradouro: 'Avenida Encarnação',
    bairro: 'Piraporinha',
    cidade: 'Diadema',
    estado: 'SP',
    pais: 'Brasil',
    _class: 'com.pi.mesacompartilhada.models.Endereco'
  },
]
```



# SPRINT

**Mesa Compartilhada 2024.2** ★ Particular Quadro

Power-Ups Automação Filtros BS NT BG BS GP +1

**Sprint - 05/08 à 19/08**

Reunião Inicial  
5 de ago.

Iniciando Atividade - Grupo  
12 de ago.

Mesa Compartilhada Logotipo

**SMART GOALS**

- S** - Nosso objetivo é melhorar a qualidade das parcerias entre as instituições e a comunidade local, criando uma rede sustentável e de qualidade de vida.
- M** - Nosso objetivo é melhorar a qualidade da comunicação entre as instituições e a comunidade local, criando uma rede sustentável e de qualidade de vida.
- A** - Nosso objetivo é melhorar a qualidade da comunicação entre as instituições e a comunidade local, criando uma rede sustentável e de qualidade de vida.
- R** - Nosso objetivo é melhorar a qualidade da comunicação entre as instituições e a comunidade local, criando uma rede sustentável e de qualidade de vida.
- T** - Nosso objetivo é melhorar a qualidade da comunicação entre as instituições e a comunidade local, criando uma rede sustentável e de qualidade de vida.

+ Adicionar um cartão

**Sprint - 26/08 à 09/09**

Reunião Inicial  
26 de ago.

Apresentação do Projeto

Documentação Arquitetura  
21/21

Wireframe  
26 de ago.

Desenvolvimento do Front-End. Formulários de cadastro  
9 de set.

+ Adicionar um cartão

**Sprint - 16/09 à 30/09**

Reunião Inicial  
16 de set.

Apresentação do Projeto

Desenvolvimento do Front-End. Site principal  
1

Desenvolvimento do Back-End. Padrão de Projeto  
1

Definição das partes jurídicas  
1

+ Adicionar um cartão

**Sprint - 07/10 à 21/10**

Reunião Inicial  
7 de out.

Desenvolvimento do Front-End. Plataforma  
0/3

Desenvolvimento do Back-End. Plataforma  
3

Desenvolvimento do Back-End. Criação de doação - Doador  
1

Desenvolvimento do Back-End. Solicitação Recebedor  
1

+ Adicionar um cartão

+ Adicionar outra lista







# REPOSITÓRIO



<https://github.com/mesa-compartilhada>





The background features a light cream color with abstract yellow and orange wavy shapes in the corners. On the left, a bowl of food is partially visible, containing white rice, a hard-boiled egg, a slice of cheese, and a dark brown sauce with orange sticks and green herbs. On the right, a bowl of soup or stew is shown, with chunks of meat, vegetables, and green herbs. Two green starburst shapes are also present, one in the top right and one in the bottom left.

**OBRIGADA(O)  
PELA  
ATENÇÃO!**