

---

**Donar**

---

**Donar**  
**Arquitetura de Software**  
**Versão <1.2>**

## Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
04/08/2021	1.0	Versão inicial	Luís Lucilandio de Queiroz Mesquita e Leonardo Fabrício da Silva Costa	Raphael C. Muniz	
21/08/2021	2.0	Segunda versão	Luís Lucilandio de Queiroz Mesquita e Leonardo Fabrício da Silva Costa	Raphael C. Muniz	

## Sumário

1. Introdução .....	4
1.1 Escopo .....	4
1.2 Definições, Acrônimos e Abreviações.....	4
1.3 Visão Geral.....	4
1.4 Posicionamento.....	5
1.4.1 Instruções do problema.....	5
1.4.2 Resumo da parte interessada.....	5
1.5 Documento de requisitos.....	6
1.6 Funcionalidades.....	6
1.7 Modelagem do negócio.....	7
1.7 Modelagem do aspecto do negócio.....	7
2. Representação Arquitetural.....	8
3. Metas e Restrições da Arquitetura .....	8
4. Visão Lógica.....	9
4.1 Diagrama de classes.....	9
4.2 Modelo lógico.....	9
4.3 Diagrama entidade relacionamento.....	10
5. Visão de Implantação.....	10
6. Visão de desenvolvimento.....	11
7. Qualidade.....	11
8. Responsáveis.....	11
9. Arquitetura de Software.....	12
10. Necessidades e funcionalidades.....	13

## 1. Introdução

O propósito deste documento é expor as necessidades e funcionalidades gerais do sistema, tais como a metodologia utilizada, ferramentas para produção e levantamento de requisitos.

### 1.1 Escopo

Gerenciar o controle de pessoas em determinados locais, tais como: igrejas, academias, restaurantes, entre muitos outros que geram aglomeração de pessoas.

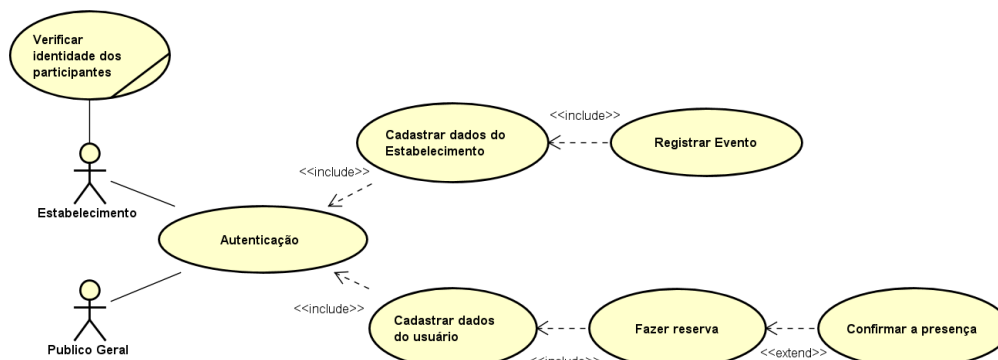
### 1.2 Definições, Acrônimos e Abreviações

- MTV - Padrão de arquitetura de software. A arquitetura MTV é uma derivação deste padrão de projeto. Onde M significa model responsável pela parte de regras de negócio e funciona como um intermediário para manipular os dados. T significa template, que é responsável pela parte de visualização. E por último a sigla V, que faz referência a view, na qual é responsável por formatar os dados que serão mostrados no template.

### 1.3 Visão Geral

Será apresentado ainda neste documento os posicionamentos quanto à parte interessada, a visão geral do produto na forma de: casos de uso, modelagem de negócio e modelo de estado do negócio e ainda as visões funcionais e lógicas.

#### Perspectiva do Produto



## 1.4. Posicionamento

### 1.4.1. Instrução do Problema

Este sistema tem o intuito de diminuir as aglomerações em locais públicos ou privados, uma vez que possa diminuir os riscos de contaminação da covid-19, ou apenas gerenciar o controle de pessoas desses estabelecimentos, ou ainda beneficiar quem queira criar evento privado.

### 1.4.2 Resumo da Parte Interessada

Esta seção fornece um perfil das partes interessadas e usuários que estão envolvidos no projeto. Esta seção também identifica os principais problemas que as partes interessadas e os usuários consideram que a solução proposta deva tratar.

Nome	Representa	Função
Estabelecimentos (igrejas, academias, crossfit, restaurantes, etc)	Consumidores finais do produto.	Cadastrar sua atividade ou evento.
Público em geral	Consumidores finais do sistema.	Realizar check in, ou seja, se inscrever na plataforma para tal serviço.
Sistema	Canal entre as entidades e o público.	Guardar informações de reservas, fornecer número de vagas pendentes, e cadastro de novas entidades.

## 1.5. Documento de Requisitos

### **RF01. Cadastrar Dados dos Estabelecimentos.**

#### **Descrição:**

- Após a autenticação no sistema, a instituição preenche um formulário passando alguns dados de sua empresa, tais como nome, endereço e etc.

#### **Usuário:**

- Estabelecimentos.

### **RF02. Cadastrar Eventos.**

#### **Descrição:**

- As instituições informaram os dados do evento, tais como: quantidade de pessoas, horário inicial e final e etc.

#### **Usuário:**

- Estabelecimentos.

### **RF03. Cadastrar Dados do Público.**

#### **Descrição:**

- Os usuários realizam login através da autenticação e preenchem um formulário para concluir cadastro.

#### **Usuário:**

- Público em Geral

### **RF04. Fazer Reserva.**

#### **Descrição:**

- Os usuários escolhem o agendamento de seu interesse e criam uma reserva.

#### **Usuário:**

- Público em Geral

### **RF0. Realizar Check-in.**

#### **Descrição:**

- No dia do evento o usuário entra no sistema e confirma sua participação no evento.

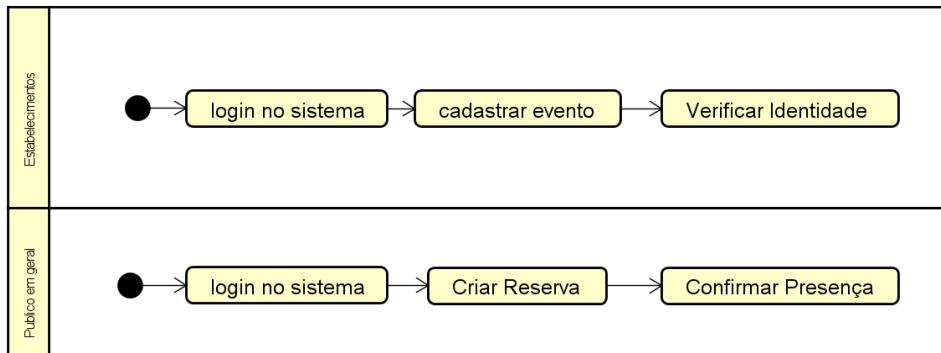
#### **Usuário:**

- Público em geral.

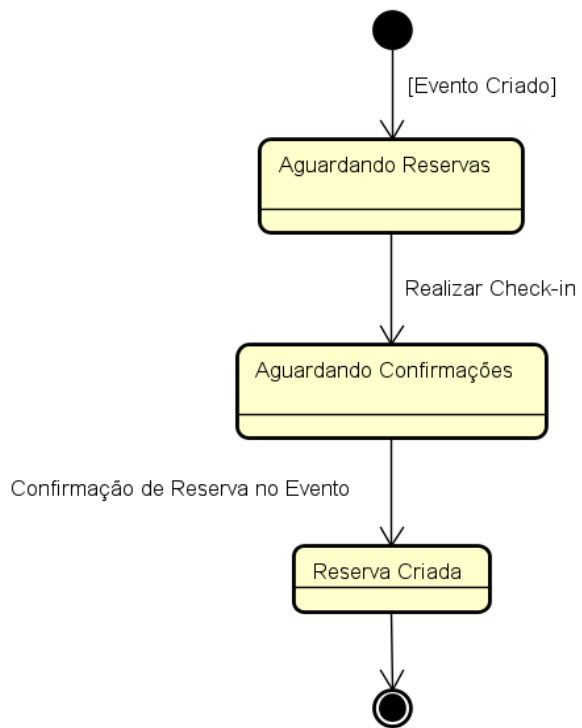
## 1.6. Funcionalidades

Os usuários terão acesso a todos os recursos de seu interesse que estejam disponibilizados pelo sistema de gerenciamento de reservas. Lá eles poderão cadastrar suas programações, ter acesso a listagem de agendamentos, realizar reserva ou check-in, ou seja, confirmar sua participação em eventos.

## 1.7. Modelagem do negócio



## 1.8. Modelagem do aspecto do negócio



## 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 implantação
  - Descreve a estrutura do ambiente onde o software será instalado.
- Visão de Desenvolvimento
  - Visão responsável por dividir o sistema em partes menores, chamadas de módulos e ainda explicar como o sistema é organizado.

## 3. Metas e Restrições da Arquitetura

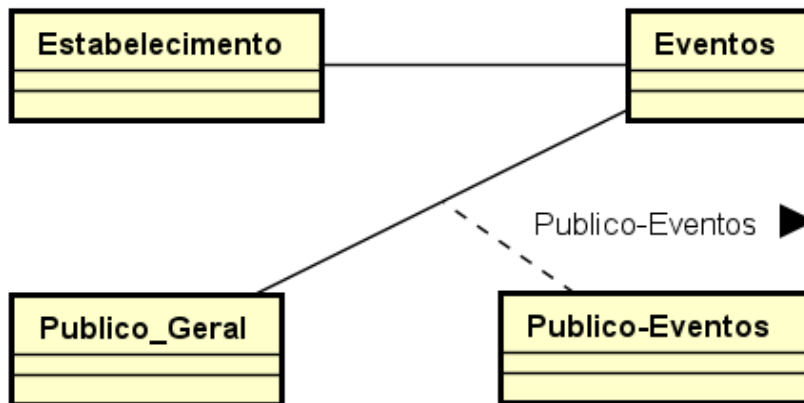
Existem algumas restrições de requisito e de sistema principais que têm uma relação significativa com a arquitetura, sendo elas:

- Utilização do paradigma Orientado a Objetos para o desenvolvimento;
- Estrutura MTV;
- Linguagem de marcação HTML;
- Linguagem de programação Python;
- Framework *Bootstrap* e *Django*;
- O sistema em questão deverá ser implementado apenas para Web;
- Bancos de dados *PostgreSQL*.

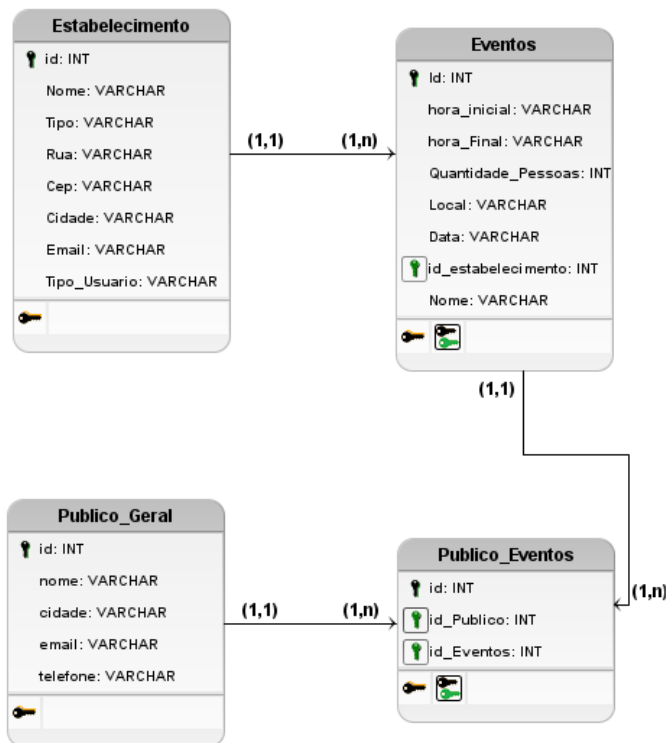


## 4. Visão Lógica

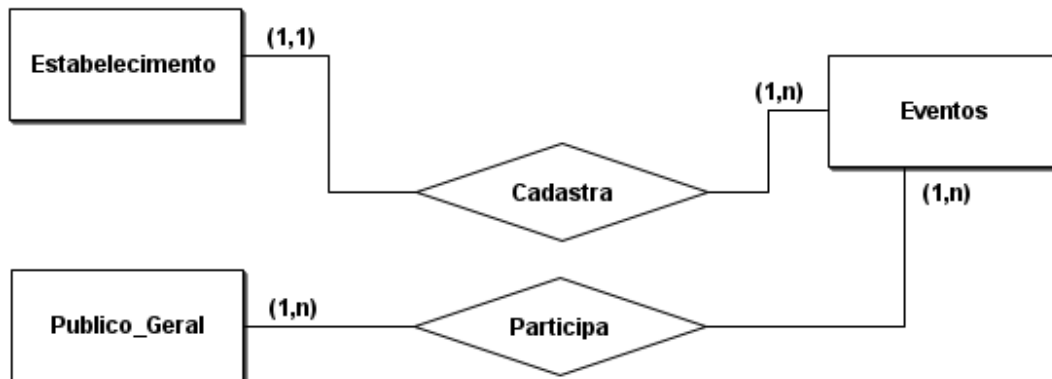
### 4.1 Diagrama de Classe geral do sistema.



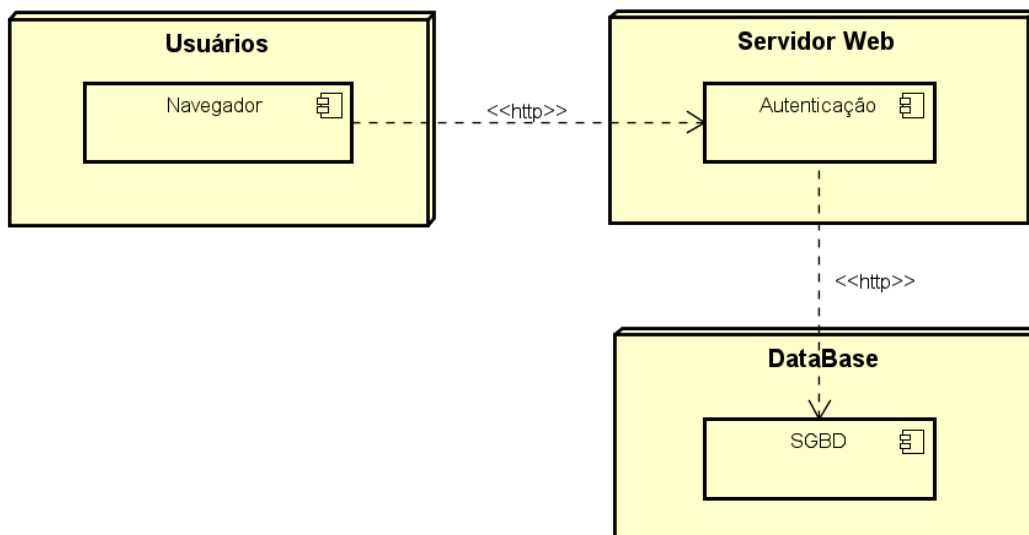
### 4.2 Modelo Lógico



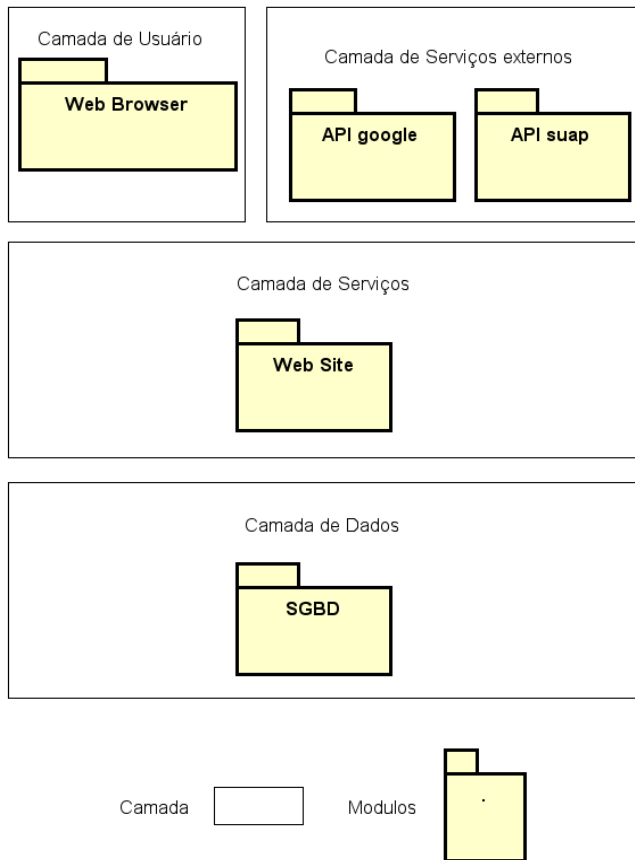
### 4.3 Diagrama Entidade e Relacionamento



### 5. Visão de Implantação



## 6. Visão de Desenvolvimento



## 7. Qualidade

O padrão de arquitetura adotado no projeto tem como finalidade garantir uma melhor organização do código-fonte, o que auxilia na manutenção do software, bem como na portabilidade do mesmo.

## 8. Responsáveis

**Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande Do Norte – Campus Pau dos Ferros.**

### **Reitor**

José Arnóbio de Araújo Filho

### **Coordenador-Geral**

Emanuel Neto Alves de Oliveira

## Sistema de Gerenciamento de Reservas.

### **Presidente**

Lucas Vieira de Souza

### **Diretor Superintendente**

Raphael de Carvalho Muniz

### **Superintendência de Negócio**

Jefferson Queiroga Pereira

### **Líder de Projeto**

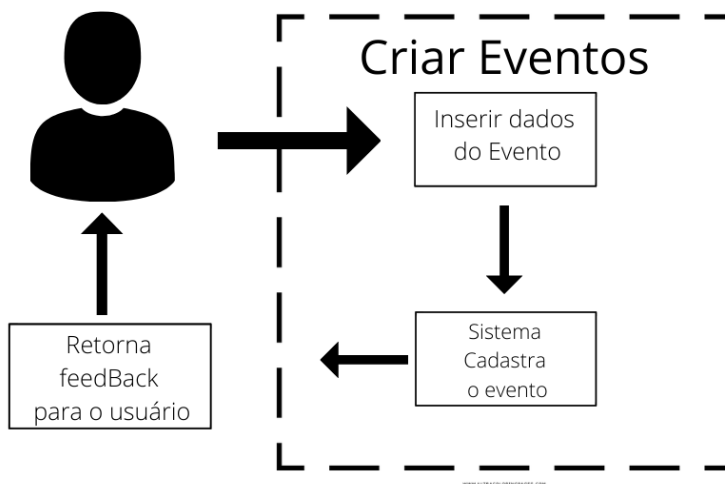
Lucas Vieira de Souza

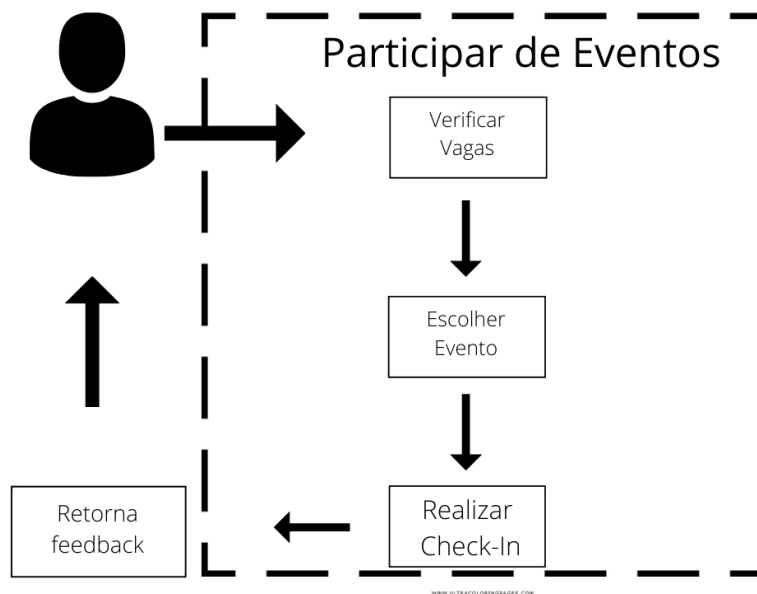
### **Desenvolvimento**

Luís Lucilândio de Queiroz Mesquita

Leonardo Fabrício da Silva Costa

## 9. Arquitetura de Software





## 10. Necessidades e Funcionalidades

Necessidade 1		Benefício
Gerenciar controle de pessoas em eventos.		Crítico
Id Func.	Descrição das Funcionalidades/atores envolvidos	
F 1.1	Cadastrar Dados dos Estabelecimentos.	
	Estabelecimentos.	
F 1.2	Cadastrar Eventos	
	Estabelecimento	
F 1.3	Cadastrar dados do público	
	Público em Geral	
F 1.4	Realizar Reserva	
	Público em Geral	
F 1.5	Realizar Check-in	
	Público em Geral	

### Proposta de Solução Tecnológica Escolhida

O sistema será implementado em Django, Javascript e HTML e será criado um banco de dados utilizando o PostgreSQL.